

CRANFIELD UNIVERSITY
DEFENCE COLLEGE OF MANAGEMENT AND TECHNOLOGY
ENGINEERING SYSTEMS DEPARTMENT

PhD Thesis

Academic Year 2007-2008

Stuart Arnold

**TRANSFORMING SYSTEMS ENGINEERING PRINCIPLES
INTO INTEGRATED PROJECT TEAM PRACTICE**

Supervised by Prof. Phil John

January 2008

[Blank]

Abstract

This investigation considers the composition, status, principles and defence acquisition setting of systems engineering. From these some opportunities for enhancement of its practice are considered.

It opens with a re-assessment of the disciplinary essence of systems engineering. Systems engineering is considered as an amalgam of three components – systems reasoning, engineering and management – that form a coherent and distinctive discipline. It is advanced that a fresh balance of system-related factors, characterised in this text as systems reasoning, is the distinguishing essence of systems engineering. It conveys a rationale for present-day practice and provides a basis for advancements.

Consideration is given to the construction of a systems engineering framework, built from a re-interpretation of engineering and management science constructs. A triptych of viewpoints of systems engineering, comprising connected representations of business process, organisational capability and individual competence, is proposed and outlined. These three essential views define a paradigm of systems engineering able to structure present-day engineering complexities and risks, and permit project and enterprise control of business achievement and risk exposure.

An analysis of the UK MOD acquisition setting for systems engineering, and an Integrated Project Team Leader survey of prevailing system engineering attitudes, experiences, expectations and concerns, set the scene for practice advancements. The first of these is based on a rigorous view of what capability means and how this impacts IPT technical contributions and responsibilities. The effectiveness of the current MOD acquisition cycle is then considered. An alternative, that might better serve the changing nature of investment constraints and effective capability delivery, is presented. Approaches to systems engineering planning are then analysed and a conclusion drawn regarding a planning instrument for IPTs that balances prescription, guidance and didacticism.

An assessment of how requirements assist and hinder working with customers and suppliers dissects the intent and content of requirements, including their contrasting technical and commercial purposes. System descriptions, their relationship and their concordance are then considered in a detailed look inside the technical processes, and this includes the principles and methods employed to design architecture. The resolution of current conflicts and confusions over architecture is seen to lie in observance of disciplined systems engineering principles. Finally the systems engineering views of humans inside and outside the system boundary are explored, and the investigation closes with a consideration of the degree to which systems engineering may reasonably address social influences.

[Blank]

The Order of Things

Big systems contain little ones,
from which properties are emergent.
Little systems then hold lesser ones;
thus, to elements are convergent.

And, big systems serve environments
that progressively extend,
to a celebrated restaurant
at the universe's end.

S. Arnold
[after Augustus De Morgan, *A Budget of Paradoxes*, 1872,
after Jonathan Swift 1667-1746,
and recognising Douglas Adams 1952 -2001]

With gratitude to Linda Arnold,
who both endured and supported this part-time study throughout its eight years.

[Blank]

Contents

	ACKNOWLEDGEMENTS	xi
	ACRONYMS	xii
1	INTRODUCTION	
1.1	Rationale for research	1-1
1.1.1	Time and Tide	1-1
1.1.2	Responding to Change	1-2
1.1.3	The Subject of Enquiry	1-3
1.1.4	The Domain of Consideration	1-3
1.2	Thesis Structure	1-4
1.2.1	Overview	1-4
1.2.2	Systems Engineering Essence	1-5
1.2.3	Codified Systems Engineering Principles	1-5
1.2.4	MOD Context for Systems Engineering	1-6
1.2.5	Systems Engineering Practice in MOD IPTs	1-7
1.2.6	Themes	1-8
2	AN ENDURING DISCIPLINE	
2.1	Prologue	2-1
2.2	Synopsis	2-2
2.3	The Rise of a Discipline	2-3
2.3.1	The Emergence of Systems Engineering	2-3
2.3.2	Influences and Interpretations	2-8
2.3.3	A Faltering Journey	2-10
2.4	Identity and Recognition	2-13
2.4.1	Scope and Definition	2-13
2.4.2	Practice, Discipline or Profession	2-16
2.4.3	Delivering Value	2-23
2.5	Formalising the Systems Engineering Discipline	2-25
2.5.1	Systems in an Engineering Setting	2-25
2.5.2	Systems in an Management Setting	2-27
2.5.3	A Context of Others	2-30
2.6	Outcomes	2-34
3	REASONING ABOUT SYSTEMS	
3.1	Prologue	3-1
3.2	Synopsis	3-2
3.3	Systems Reasoning and Philosophy	3-4
3.3.1	Classical Order and Rationality	3-4
3.3.2	Reductionism and Holism	3-7
3.3.3	Latter-Day Philosophy	3-09
3.4	Systems Reasoning and Science	3-11
3.4.1	A Common Scientific Insight	3-11
3.4.2	A Classification of Systems	3-13
3.4.3	The Science of Order	3-21
3.5	Systems Reasoning and Psychology	3-24
3.5.1	Dealing with Complexity	3-24
3.5.2	Perceiving Reality	3-30
3.5.3	Modelling Systems	3-35
3.6	Coda	3-38
3.7	Outcomes	3-38

4	THE FORMULATION OF SYSTEMS ENGINEERING	
4.1	Prologue	4-1
4.2	Synopsis	4-1
4.3	Codifying Systems Engineering	4-3
4.3.1	The Role of Standards	4-3
4.3.2	Models Founded on Business Process	4-6
4.3.3	Synthesising an International Solution	4-11
4.4	The Engineering Viewpoint	4-16
4.4.1	A Basis of Systems Reasoning	4-16
4.4.2	Minimising Model Complexity	4-21
4.4.3	Process Model Limitations	4-26
4.5	The Management Viewpoint	4-31
4.5.1	A Project Focus	4-31
4.5.2	Corporate Responsibilities	4-34
4.5.3	Trading and Agreements	4-35
4.6	Outcomes	4-37
5	LIFE CYCLE MANAGEMENT	
5.1	Prologue	5-1
5.2	Synopsis	5-2
5.3	Life Cycle and Management	5-3
5.3.1	A System of Processes	5-4
5.3.2	States, Stages and Gates	5-6
5.3.3	The System's View	5-9
5.4	Life Cycle Classification	5-12
5.4.1	Sequential Forms	5-12
5.4.2	Canonical Forms	5-14
5.4.3	Hierarchical Forms	5-16
5.4.4	Cyclic Forms	5-17
5.4.5	Organic Life Cycles	5-19
5.5	Life Cycle Morphology	5-20
5.5.1	Life Cycles of Life Cycles	5-20
5.5.2	Metamorphosis of Life Cycles	5-22
5.6	Outcomes	5-23
6	CAPABILITY AND COMPETENCE	
6.1	Prologue	6-1
6.2	Synopsis	6-3
6.3	Systems Engineering Capability	6-4
6.3.1	From Business Process to Organisational Capability	6-4
6.3.2	Discrete or Integrated Capability Models	6-8
6.3.3	Indicators	6-10
6.3.4	Work Product Identification	6-14
6.4	Systems Engineering Competence	6-18
6.4.1	From Business Process to Professional Competence	6-18
6.4.2	Competence Attributes	6-21
6.4.3	The Attainment Scale	6-25
6.4.4	Professional and Organisational Contexts	6-30
6.5	Outcomes	6-32

7	THE ACQUISITION SETTING FOR DEFENCE SYSTEMS	
7.1	Prologue	7-1
7.2	Synopsis	7-2
7.3	The March of Acquisition Reform	7-3
7.3.1	The Drivers for Change	7-3
7.3.2	A Smart Response	7-6
7.3.3	An Uncertain Start	7-11
7.4	The Paths of Others	7-13
7.4.1	Aligning Acquisition Views of Systems Engineering	7-13
7.4.2	Systems Engineering in DoD	7-14
7.4.3	Swedish Approach to Systems Engineering	7-16
7.5	The Nature of IPTs	7-20
7.5.1	Whole Life and Trans-functional	7-20
7.5.2	360 Degree Interactions	7-24
7.5.3	Culture	7-28
7.6	Outcomes	7-30
8	GAUGING THE PROGRESS	
8.1	Prologue	8-1
8.2	Synopsis	8-1
8.3	Sampling IPTL Opinions	8-2
8.3.1	Shaping the Survey	8-2
8.3.2	The Survey Structure	8-4
8.3.3	Gather Data and Eliciting Issues	8-6
8.4	Systems Engineering in the IPT	8-8
8.4.1	The Prevailing Perception of Systems Engineering	8-8
8.4.2	Systems Engineering's Contribution to the IPT	8-10
8.4.3	Team Capability and Individual Competence	8-12
8.5	The IPT's Systems Engineering Relationships	8-15
8.5.1	Working with the Customer	8-15
8.5.2	Interfacing with the Supplier Base	8-16
8.5.3	Technical and Program Interactions between IPTs	8-17
8.6	Observations	8-18
8.7	Outcomes	8-19
9	ADVANCING AGREEMENT, ENTERPRISE AND PROJECT PRACTICE IN IPTs	
9.1	Prologue	9-1
9.2	Synopsis	9-1
9.3	Supplying and Acquiring Systems	9-3
9.3.1	The Ontology of Defence Capability	9-3
9.3.2	Elements of Capability	9-6
9.3.3	IPT Accountability for Capability	9-10
9.4	An Enterprise View of Life Cycle	9-12
9.4.1	Life Cycles of Life Cycles	9-12
9.4.2	Life Cycle Reference Models	9-14
9.4.3	Whole-Life Life Cycle	9-18
9.5	A System Engineering Focus for Projects	9-23
9.5.1	A DoD Approach	9-23
9.5.2	Through Life Management and Systems Engineering	9-26
9.5.3	Augmenting TLMPs	9-27
9.6	Outcomes	9-31

Transforming Systems Engineering Principles Into Integrated Project Team Practice

10	ADVANCING TECHNICAL PRACTICES IN IPTs	
10.1	Prologue	10-1
10.2	Synopsis	10-3
10.3	Defining Requirements	10-5
10.3.1	The Nature of Requirements	10-5
10.3.2	Technical, Management and Agreement Influences	10-10
10.3.3	Requirements and the Trading Culture	10-17
10.4	Designing Architecture	10-19
10.4.1	The Nature and Description of Architecture	10-19
10.4.2	The Pattern in Design	10-27
10.4.3	System Model Frameworks	10-32
10.5	Factoring in the Humans	10-43
10.5.1	The Nature of Human Factors Engineering	10-43
10.5.2	Humans Inside and Outside systems	10-47
10.5.3	Human-Intensive Systems	10-53
10.6	Outcomes	10-56
11	CONCLUSIONS	
11.1	Summary	11-1
11.1.1	A Recapitulation	11-1
11.1.2	Principles and Engineering Knowledge	11-2
11.1.3	Practice and Management Understanding	11-4
11.2	Future Lines of Enquiry	11-6
11.3	Reflections	11-7
12	REFERENCES	12-1
Appendix A	SYSTEMS ENGINEERING CAPABILITY INDICATORS	A-1
Appendix B	SYSTEMS ENGINEERING COMPETENCE	B-1
B.1	SFIA 2 Skill Component Mapped to 5 Levels	B-1
B.2	ISO/IEC 15288 to SFIA Mapping	B-5

Acknowledgements

At the organisational level, the primary recognition is to QinetiQ and its award of a Fellowship that permitted research into the systems engineering capability models and competence models that are described in §6. This is an indication of QinetiQ's desire to advance the discipline of systems engineering, to progress its application across its project teams and to develop the competence of its systems engineering practitioners.

The opportunities afforded by MOD in its former Research Package 3a to examine how systems engineering is conducted by its international partner government organisations brought a valuable dimension to the research, and this is much appreciated.

At the individual level, the principal acknowledgement is to the support of Professor Phil John, who shared the pressures of conducting this research part-time and who provided an essential ingredient: the counselling ear. His contributions to clarity of direction and purpose over an extended and disjointed period of part-time research, and his stated criteria of achievement – the advancement of knowledge, new insights, value to society, and formulation of questions yet to be answered – have helped channel the research effort towards academic and business goals.

Dr. Raghu Singh, who on his retirement was the FAA's Chief Software Engineer, deserves particular mention as a catalyst for this research. His commitment to the subject area, and the opportunity he created for some of the ideas to be robustly debated in the international arena of ISO, immensely added to steps forward in comprehension and insight.

Discussions with Dr Jonathan Earthy helped to develop deep-seated issues in the relationship between inanimate systems and humans, and the influence these have on the systems engineering and human factors disciplines. In addition, his encouragement throughout the course of this investigation is most gratefully acknowledged.

Dr. Harold 'Bud' Lawson provided a sounding board for ideas. In particular, his contribution to our joint efforts in exploring the comparative situation in Swedish and UK defence acquisition practice (§7.4.3) is recognised.

I am specifically indebted to Professor Ken Jackson for the insights he brought to the development of ideas related to Figure 4.14. These were a powerful catalyst for new thinking about system life cycle processes and from this much has come internationally. Likewise, Dr. James Martin is thanked for many stimulating discussions on some of the profound issues faced in systems engineering, including the existence of systems in the real world, and for his views on requirements in Figure 10.6 and his understanding of systems engineering standards.

In his role as DE&S Systems Engineering Development Partner, Hillary Sillitto's sponsorship of the IPTL survey enabled the primary empirical part of this research to be conducted and this is acknowledged with much gratitude. Thanks go to Dr Dave Hawken of DE&S for arranging IPTL interviews, and for assisting in their conduct and in the documentation of responses. Special thanks go to the generosity of many hard-pressed Integrated Project Team Leaders who contributed their valuable time, candid and constructive observations, and positive support for advancing an understanding of how systems engineering may be applied better in their IPTs.

Specific texts and sources are acknowledged in the conventional manner, with one exception. Some initial outcomes from this research, including figures as indicated, were contributed from this research to the 2002 publication of the International Standard on systems engineering: ISO/IEC 15288 '*Systems engineering – System life cycle processes*'. Consequently, this International Standard is not identified as the source of some now-recognised ideas; rather, the rationale behind these unpublished Editorial influences is presented for the first time.

These acknowledgements conclude with reference to an essential, though indirect contribution. The analyses, propositions and arguments that follow were written with Prof. Mike Woodhead in mind. This was not simply because he was to have been the External Examiner but because he was a respected champion of systems engineering. He was a major influence in the discipline's advancement and standing in the United Kingdom. His personal approval of this Thesis would have been greatly valued. His death a few weeks before the completion of this period of study is a loss to the systems engineering community.

Acronyms

AMS	Acquisition Management System (of the UK Ministry of Defence)
AOF	Acquisition Operating Framework
CAD MID	Concept, Assessment, Demonstration, Manufacture, In-Service, Disposal (life cycle stages)
CMMI	Capability Maturity Model, Integrated
COTS	Commercial off-the-shelf
CRD	Capability Requirements Document
C4	Command, Control, Communications and Computers
C4ISR	Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance
DCDS(EC)	Deputy Chief of the Defence Staff, Equipment Capability
DE&S	Defence Equipment and Support
DEC	Director of Equipment Capability
DIS	Defence Industrial Strategy
DLO	Defence Logistics Organisation
DLoD	Defence Lines of Development
DoD	Department of Defense (of the United States of America)
DODAF	DoD Acquisition Framework
DPA	Defence Procurement Agency
EAC	Enabling Acquisition Change
EIA	Engineering Industries Alliance
ESA	European Space Agency
FAA	Federal Aviation Administration
FMV	Försvarets Materielverk (Swedish Defence Materiel Administration)
ICBM	Inter-continental Ballistic Missile
iCMM	Integrated Capability Maturity Model (of the FAA)
IEC	International Electrotechnical Commission
IEE	Institution of Electrical Engineers
IEEE	Institute of Electrical & Electronics Engineers
IET	Institution of Engineering and Technology (formally IEE)
IPT	Integrated Project Team
IPTL	Integrated Project Team Leader
INCOSE	International Council On Systems Engineering
ISO	International Organization for Standardization
MOD	Ministry of Defence
MODAF	MOD Architecture Framework
NAO	National Audit Office
NASA	National Aeronautics and Space Administration
NATO	North Atlantic Treaty Organisation
PDCA	Plan, Do, Check, Act
OA	Operational Analysis
OUSD(AT&L)	Office of the Under Secretary of Defense (Acquisition, Technology, and Logistics) (US DoD)
SECO	Synthetic Environment Coordination Office
SEI	Software Engineering Institute (Carnegie Mellon University)
SEMP	Systems Engineering Management Plan
SEP	Systems Engineering Plan
SFIA	Skills Framework for the Information Age
SoS	Systems-of-Systems
SRD	System Requirements Document
TLMP	Through Life Management Plan
TOGAF	The Open Group Architecture Framework
UOR	Urgent Operational Requirement
URD	User Requirements Document
UML	Unified Modeling Language

1 INTRODUCTION

1.1 Rationale for research

1.1.1 Time and Tide

In the face of a changing world, relevance, correctness and value are attributes always open to challenge in any field of science or management. In this regard systems engineering is no exception. These three criteria – relevance, as a measure of pertinence to challenges; correctness, in the sense of 'to adjust or make conform, especially to a standard' [*Collins, 1991*, discipline, meaning 5]; and value, being delivered benefit to individuals and society – have provided the purpose, direction and structure for this research.

Much soul-searching has characterised the last decade and a half of systems engineering. This angst has largely arisen from perceived failings in the level of its appreciation, uptake and successful application. This has been due in no small measure to a perennial question of what body of principles and practice is conveyed by the term systems engineering across communities of interest.

Like any evolving branch of philosophy, systems engineering depends on re-evaluations that bring fresh insight and steps of understanding, and these can lead to new theory and areas of application. The interplay between improved appreciation of principles, based on technical and business organisation theory, and application experience, derived from the pressures of everyday engineering practice, has characterised the subject since its inception. This has been shaping an identifiable, coherent and increasingly relevant discipline for over half a century.

This trend appears set to continue, and systems engineering's course will doubtless depend on regular re-appraisals, challenges, new theories and reformulations of its body of knowledge. It is a necessary dynamic that drives any discipline, and it ensures its continuing relevance, correctness and value [*Kuhn, 1962*].

Looking back permits an objective, critical analysis and assessment of old theories. It revisits old challenges, sees them in new light, and this can then offer insight into technical and business challenges of the moment. Looking around allows a more effective fit of systems engineering into the present order. Looking forward anticipates the possibilities of a new and shifting order, and points to change actions necessary for greater benefit and more widespread use.

This research may thus be seen as a contributor to these essential, periodic evaluations, augmentations and re-adjustments of systems engineering – and it is especially timely. Political, cultural, economic and technology factors continuously evolve, and they present an ever changing social and commercial environment to which engineering must respond in order to best serve the interests of society. Taken together, these factors become a catalyst for periods of rapid and profound change. Thresholds in needs and modes of application have to be crossed, and steps of radical response are called for.

In the case of systems engineering's major and traditional application domain of defence systems acquisition, a near step function change occurred during the 90s. The demands this placed on governments, industry, project teams and individual systems engineering practitioners have been profound.

The change was a result of a complex set of circumstances that arose from:

- the structural change and volatility of international security threats;
- the increasing management and use of information;
- the development of technologies that offered richer solution opportunities and options;
- the greater competitiveness in a business environment strongly influenced by technology and mass market dynamics;

- the need for much shorter periods between problem recognition and military capability deployment;
- the openness of global defence markets to high-technology system trade;
- the growth and internationalisation of synergistic supply-chain relationships.

It is in this changed environment of application that systems engineering has had to adapt and to deliver even greater value.

1.1.2 Responding to Change

Change is an outcome not only of new challenges but also of new opportunities. New openings to meet changing technical and business needs arise, and new, relevant engineering and management principles and practices step forward to match them. The scope and structure of traditional engineering discipline stovepipes are not individually well fitted to meet the consequences of complexity in today's military systems. Prof John O'Reilly, IEE President 2004-5, in his Presidential Address stated that 'real world problems do not respect the boundaries of established academic disciplines – nor indeed the traditional boundaries of engineering. Interdisciplinary, inclusive, global dimensions are core, inherent considerations for the engineering enterprise of C21st [O'Reilly, 2004].

This then is the opportunity space, and systems engineering appears particularly well poised to equip C21st enterprise to face the defence business challenges that government and industry are presented with. This opportunity has been recognised and recent UK strategies to counter failings in defence acquisition have included an emphasis on systems engineering to resolve evident difficulty in delivering effective military capability to the operational user, to time and to cost [MOD, 1999], [DIS, 2005].

Stretching back to the early days of applying a coherent, system-based engineering approach to complex defence situations there is evidence of remarkable achievement. History, however, shows us that achieving engineering excellence in military systems, or indeed in other domains, on one occasion is no guarantee of repeated success. Yet business and society nonetheless rightly expect predictability, assurance and understood risk from engineering. So it is to this – to repeatedly bridge the gaps between technology's promise and society's expectations – that systems engineering needs to be applied with ever greater effectiveness.

Systems engineering is certainly a broad-ranging and compelling strategy for the fusing of engineering and management in order to provide a concerted response to these challenges. However, it does not offer any silver bullet solution to the hugely complex socio-technical problems that continue to be presented by defence capability provision. What it does offer is the prospect of steady advancements in tackling these issues, based on continual re-evaluation of its principles and refinement of its practice. It is by this means that systems engineering has been evolving into a coherent discipline; one that is now internationally viewed as a cardinal part of defence business practice.

Nevertheless, systems engineering's qualification as an enduring, widely-accepted discipline depends on it continuing to meet criteria by which all disciplines are judged. Existing, established disciplines will not readily be displaced by a relative newcomer, even less supplanted or subsumed by it. Systems engineering's place in society has to be earned and this is judged by how effectively its principles are translated into worthwhile practices with evident benefit to society. Any long term business strategy based on systems engineering, such as that advocated by MOD, should be predicated on the discipline's evident and lasting value.

As in all competitive situations, positioning is vital. Present day circumstances in the domain of systems engineering's genesis – military systems – mean that time and tide favour the discipline's more committed adoption. In response, its interpretation and modes of application have been evolving quickly in the opening years of C21st. This investigation is a contribution to this evolving scene.

1.1.3 The Subject of Enquiry

Systems engineering is concerned with combining the technological dissimilar in order to achieve functional unity, purpose and benefit. It arises out of the wide variety of interventions and interactions that different parties make over a range of levels of structural detail throughout the life cycle of an engineered solution. It depends on the co-operation of diverse contributors – in time, location, skills, interest and technical knowledge – and it is most commonly conducted in competitive trading environments. It is, according to Ramo, a supreme outcome of "techno-political-econo-socio-experts" [Ramo, 1998, p 58]. It is therefore not in the least surprising that there are many viewpoints and interpretations of systems engineering.

The diversity of views of these many participants has led to different, often conflicting descriptions of systems engineering. Many who employ its approach do not even identify with it, either as a singular discipline or by its title. The cross-cutting strength of systems engineering is therefore to be seen also as a weakness; an inhibitor to its adoption and an obstacle to its consistent, across-the-board application. Common consent on the composition and scope of systems engineering, and how it sits in the context of existing engineering and management order, has been a long-standing matter of contention. Under its title of systems engineering it has flourished in some quarters and yet been ignored in others.

This research advances that, in a coming of age, systems engineering can be seen clearly as an amalgam from three necessary disciplinary components. Firstly, it is composed of systems reasoning: a fundamental stratagem of the human mind and an underpinning element of many branches of philosophy. To this base is added engineering: the profession and the endeavour that applies science and technology to create useful artefacts, possessing desirable and typically novel properties. Lastly, management principles bring the planning, monitoring and control of resources, including skilled engineering practitioners, and organisational and investor assets; these are directed to meet the interests of all parties who have a stake in the outcome.

Applied holistically, these three components lead to a branch of engineering that focuses on ensuring that man-made entities can better meet the needs of society according to the technological possibilities available. It has resulted in the emergence of a distinct intellectual and empirical tool. Trans-disciplinary in its nature, it is a technology-based subject that has been crafted to fulfil present-day needs of society to create and use complex artefacts, according to the constraining conventions of trade and commerce.

The chapters that follow examine the status, composition, principles, defence domain and associated application advancements of systems engineering. They explore some fresh interpretations of and revisions to existing views of systems engineering. They advance some models that apply to businesses employing systems engineering, and to the individuals and teams practicing it.

1.1.4 The Domain of Consideration

Systems engineering is approaching workable levels of codification, yet its formulations can still be a journey into the abstract. Such 'pure' systems engineering is a necessary step from systems reasoning to engineering practice. However, achieving the goal of context-dependent, 'applied' systems engineering signifies its qualification as a truly mature engineering discipline. This is not confined to a concern with the management of science and technology to create technical products and services that are useful in particular application domains. It is also about successfully executing complex engineering in highly competitive business environments.

This means that the codification of systems engineering must align with the challenges of commerce; that models of action can be effectively instantiated in the context of trading organisations. These models must describe how to reliably meet stakeholder expectations for engineering solutions that are capable of contributing effectively, routinely, soundly, safely and responsibly, and that can be created and operated according to demanding business criteria and time scales.

Systems engineering evolved as a discernable, self-consistent set of engineering and management principles and practices in the defence community. It is thus apposite that this technical and business re-examination should be in its contextual origins of defence. Moreover, in the arena of defence acquisition, changing need and the need for changing responses are currently important national issues and are worthy of pressing investigation. They are subjects of immense concern, presently being considered from many viewpoints. National security, economics and industrial wellbeing have all fuelled a continuing strategic debate with systems engineering a cardinal part of this [DIS, 2005].

This investigation therefore explores the application of a codified framework of systems reasoning, engineering and management tuned to meet the new UK defence acquisition situation. Though its selected operative context is the Integrated Project Team (IPT) within MOD's Defence Equipment and Support (DE&S), and more specifically its predecessor Defence Procurement Agency (DPA) constituency, this is nevertheless considered to be an important indicative use case for systems engineering. The political and commercial environment that characterises defence systems has parallels in other areas of application: transport, civil security, health, emergency services, even financial trading. Thus, whilst conducted in the context of a particular community in a defined field of application, this enquiry into systems engineering may be seen as having far greater generality.

1.2 Thesis Structure

1.2.1 Overview

As implied above, and in accordance with the 'Law of Requisite Variety' [Ashby, 1956, Ch 11] systems engineering is an extremely complex structure in its own right. Just as with the complex systems it enables and controls, it is only understood through related views that convey its perspectives from a variety of concerns. It does not therefore lend itself well to a linear, sequential description. There is no unique or preferred order to present this interlocked network of concepts and images; no ideal unfolding of a sequence of arguments.

The following chapters are nevertheless constrained by textual and graphical sequence. The rationale for the chosen structure is one of steady progression from abstraction to pragmatism; from theory to day-to-day practice. Accordingly, the structure of this thesis is built up in four layers, see Figure 1.

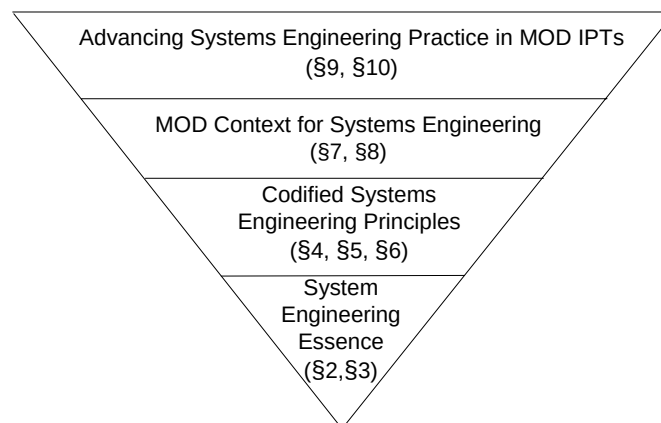


Figure 1 Thesis Structure

In §2 and §3, the argument commences by conducting the groundwork that re-assesses the disciplinary essence of systems engineering. This is followed in §4, §5 and §6 by the construction of a technical framework, built from a re-interpretation of particular engineering and management science constructs on which systems engineering is generally accepted to depend. Finally, considering systems engineering practice in MOD IPTs, §7 and §8 present an analysis of prevailing acquisition management challenges faced by MOD; and §9 and §10

propose some engineering practice enhancements, derived from the preceding principles, that should enable IPTs to better address some of these challenges.

1.2.2 Systems Engineering Essence

In §2, *An Enduring Discipline*, this thesis commences by resolving a crucial question: whether the discipline of systems engineering is indeed an adequate and reliable predicate from which to define effective business change actions.

It reviews how systems engineering entered the technical and managerial consciousness of organisations, and evolving into its present-day status. It then postulates steps of discipline maturity against which it can be appraised; now and as it continues to evolve. It calibrates present day systems engineering against these criteria and argues that, despite persistent and often-voiced concerns, its current status gives rise to reassurance and optimism.

A powerful and rational association of three essential disciplinary elements – systems reasoning, engineering, management, is seen to have been progressively synthesised into a consistent and distinguishable whole. How they inter-mesh, one with another and with the organisational business environment they inhabit, completes an assessment of the discipline's dependability and value. It is concluded that systems engineering has evolved into a discipline with levels of business relevance, recognition and pragmatism that merit its explicit adoption as an enabler of MOD acquisition changes.

Given this sufficient measure of foundational assurance, §3, *Reasoning About Systems*, then analyses the least well codified of this trio of essential elements: the cognitive hallmark of systems engineering – advanced here under the title of systems reasoning. It defines systems reasoning to be an explicit region of cognition that is intimately employed in the engineering and management of man-made systems, and for that matter in many human pursuits. Systems reasoning is presented according to three primary contributors: a long-standing body of intellectual argument – termed systems philosophy; the C20th 'discovery' of a substantially common system-based contribution to a range of natural and human science disciplines – systems science; and hypotheses about a region of cognition that leads to the perception of systems – termed systems psychology. From this amalgam some natural and underpinning systems engineering principles become evident.

Reasoning About Systems, reframes the distinguishing essence of systems engineering by advancing a fresh balance in the system-related factors that govern the systems engineering practitioner's conduct of engineering and management.

1.2.3 Codified Systems Engineering Principles

The second layer of Figure 1, *The Formulation of Systems Engineering*, §4, commences the shift from principles towards the practicalities of systems engineering, towards a setting of competitive defence business.

It opens with a proposition for defining a coherent triptych of systems engineering views. This comprises related descriptions of business process, organisational capability and individual competence that trace to a common, underlying human activity model. These three essential views define a paradigm for systems engineering which enables trading communities, project teams and individuals to collectively harness complexity. Their definition is a requisite maturity step if systems engineering is to be established as a widespread, cardinal element of business outside its founding defence/aerospace community.

The first of these views – a business process model – is presented in §4. It is a model that is explicitly related here to primary axioms of systems reasoning. It embodies the essential transformations that link problem definition, through abstract solution definitions, to its construction as material reality, and ultimately to intervention in an operational environment. During the period of this investigation, this process model has become internationally accepted as the most useful representation of systems engineering. The model conveys that the human-driven actions that effect these life cycle transformations and expound system engineering as a discipline are themselves a system; a dynamic network of human activities capable of being reasoned about in systems terms.

The organisational and trading context into which systems engineering must deliver value is an important part of this consideration. It lays the basis for introducing advances in the practical interpretation of systems engineering principles. It shows how strata of transformations associated with different responsibilities are bonded into a hierarchy of inter-organisational trading interactions. Process work products serve to integrate the interfaces at commercial/agreement boundaries and this underlines the business concerns that systems engineering must satisfy. At each trading interface between different realms of responsibility a compatible, yet inverse work product representation of the technical transformations is evident.

In §5, titled *Life Cycles and Management*, the microscopic, almost chaotic real-time execution of the network of processes in §4 is abstracted to reveal macroscopic patterns that enable the flow of through-life management. These life cycle forms help structure the engineering complexities and risks, and synchronise requisite project and enterprise assessments in order that business achievement and exposure is controlled. This equates to the supervision and periodic scrutiny of projects, and is performed by technical, project and enterprise managers during the lifetime of a military capability.

In preparation for later analysis of defence acquisition cycles, §5 also examines how different system attributes influence the fashioning and management of a system's life cycle. A hierarchy of life cycle forms is categorised in order to disentangle the complexity embodied, and often concealed, in individual life cycles. The underlying morphology that links these forms is illustrated. It conveys how in practice they resolve one into the other at different levels of abstraction of life cycle complexity.

Under the heading of *Capability and Competence*, §6 completes the two remaining faces of the triptych of systems engineering. Each is derived from the business process model described in §4. It describes the relationships and detailed mapping required to build these two other views; one for an organisation's systems engineering capability, the other for systems engineering practitioner competence.

The model for organisational capability complements ISO's well-proved capability maturity assessment method. Drawing from information in the transformation-rich model of system life cycle processes in §5, it provides the engineering and associated management work product inventory that signifies the conduct of effective systems engineering across an organisation. The detail of the resulting model is presented in Appendix A, *Systems Engineering Capability Work Products*, and this has been ratified for incorporation in a forthcoming International Standard.

To complement this capability model, a schema for defining and developing practitioner competence is also derived from §5. It presents arguments for the particular approach that is followed and illustrates key components of the competence model. In its detail it benefits from a widely employed skills framework that evolved from information technology/software practices. Some detail of this approach is presented in Appendix B, *Systems Engineering Competence Framework*. This completes the development and presentation of elements of a codified systems engineering paradigm on which defence acquisition practices may be reliably based.

1.2.4 MOD Context for Systems Engineering

Two chapters set the context for the conduct of systems engineering in MOD. This commences in §7, *The Acquisition Setting for Defence Systems*, with a description of the MOD acquisition context for systems engineering. It presents an analysis of recent, formative factors and events that have moulded current UK defence acquisition, and how these have both favoured and hindered the acceptance of systems engineering as a cardinal part of acquisition practice.

In order to gain comparative understanding, and because military capability acquisition is increasingly coupled to practices in other nations, the wider picture of the international defence acquisition scene is considered next. This is achieved through reference to two nations with contrasting ambitions and constraints – USA and Sweden – and this assists an understanding of UK progress, and where opportunities for further enhancement of UK defence systems engineering practices may lie.

The IPT setting is specifically considered in preparation for subsequent, more focussed exploration of how systems engineering is being, and could yet more fully be adopted by MOD. It examines features that characterise IPTs, particularly their integrated organisational function nature, and their key role and responsibilities in the chain of military acquisition interactions that link operational user and industrial technology provider.

In §8 this investigation follows an empirical path. Under the title of *Gauging the Progress*, this chapter presents evidence of prevailing IPT attitudes, experiences, expectations and concerns about systems engineering. This was gained through face-to-face interviews with IPT Leaders (IPTLs). Analysis of their views illustrates the degree to which systems engineering is already embedded in IPTs, how this manifests itself and areas in which this may be strengthened.

Importantly, the IPTL opinions convey how systems engineering is perceived at the very heart of MOD acquisition. These views point to where systems engineering could assist the creative interactions between IPTs and their customer, and also strengthen the relationship with the provider of key elements of capability: the industrial suppliers. This survey also assesses the systems engineering asset base available to IPTs, including knowledge, guidance, skills and behaviours.

1.2.5 Systems Engineering Practice in MOD IPTs

Finally, the uppermost layer of Figure 1 is considered. It investigates how foregoing theory and codification of system engineering can be applied in order to increase the effectiveness of IPTs. Given this setting, and in the light of preceding refinement of principles, §9 and §10 proceed to investigate advancements in IPT systems engineering practice. These two chapters consider how day-to-day systems engineering might be augmented or refined. This is illustrated according to the now-justified International Standard's pattern of agreement, enterprise, project and technical dimensions [ISO, 2002]. Necessarily limited to a few key, representative challenges from across systems engineering's province of application, the two chapters take selected IPTL concerns and show that aspirations declared in recent defence acquisition reform initiatives could benefit from the re-evaluation of principles in this investigation.

In *Advancing Agreement, Enterprise and Project Practice*, §9 addresses contributions that systems engineering can make to the organisation's acquisition/supply of capability, to the cycle management of systems, and to project planning. It starts with a consideration of the nature of capability and what IPTs are actually contracted to supply and acquire. This requires a more rigorous view to be taken of what capability means as a systems engineering concept. The analysis sharpens the meaning of a term that has moved into trite and sometimes glib use in defence acquisition circles. The argument is developed ontologically in terms of systems engineering's international codification [*ibid*] and MOD's Capability Management Handbook [MOD, 2007]. It contrasts UK and overseas approaches to the acquisition of what will frequently prove to be complementary components of coalition capability.

An international comparison also opens the section on life cycle management. This explores the enterprise approach to controlling the quasi-sequential stages of progression through the military system life cycle. It leads to consideration of the detail in, and relationship of, two intermeshing cycles: the system creation cycle and the system utilisation cycle. With MOD's 2007 amalgamation of 'procurement' IPTs and 'support' IPTs under a single enterprise management regime termed DE&S, an 'infinity' life cycle model rationalises the technical synergy in this procurement/support unity and conveys how system life cycle responsibility maps to this changed MOD grouping. It shows the value of IPT continuity throughout the life cycle, with team capability and competences metamorphosing in response to the changing nature of the prevailing systems engineering and project management challenges.

For each IPT the delivered value of systems engineering starts with effective planning, followed by continual assessment and control using the plans produced. Particularly for large projects, planning for project management and systems engineering management have been the traditional, co-existent mechanisms for providing project direction. Given Smart Acquisition's evolved corporate frame of reference for the procedures that are to be followed, the information work products to be produced and the project scrutiny criteria that will be

applied, an explicit systems engineering management plan may not be the most appropriate information product for directing systems engineering. The pros and cons of different approaches are analysed, and conclusions are drawn regarding how to establish the most acceptable and least disruptive planning instrument, and how within this to balance prescription, guidance and didacticism.

The penultimate chapter considers some advances that systems engineering can bring to technical practice. §10, *Advancing Technical Practice in IPTs*, follows the notional sequence of requirements, design and implementation. It begins with a re-assessment of what requirements are and what they do through the system life cycle. It examines how they can both help and hinder communal working between and across teams. The catch-all term 'requirements' is applied to a category of information items that is here dissected into its several constituents. Their technical role of system design and build documentation is contrasted with their role in intra- and inter-organisation agreements and transactions. Weak appreciation of the nature and distinction of these different contributions has dogged requirements management and some of Smart Acquisition/DIS's sought-for changes in government/industry trading. Issues that underlie this technical/trading dichotomy are exposed and approaches to their resolution suggested.

Two distinct classes of system description lie at the heart of design: function and form. The interplay between them and their resolution is for many the very essence, the focus of systems engineering. The analytical and compositional design strategies and tactics, employed at different stages in the lifetime of a system and at multiple levels of system structural resolution, are at the heart of system design and architecture description. These architecture descriptions, their relationship and their consistency are considered in a detailed look beneath the technical processes, and the principles and methods employed to design architecture. Building from the earlier foundations in systems reasoning, a creative discipline and logic is apparent in the art and intuition that many associate with design. The prevailing conflicts and confusions about architecture that trace back to software and information technology-intensive systems are assessed. The resolution of this is seen to lie in a more disciplined interpretation of fundamental systems engineering principles.

Lastly, whether considered to be part of a system solution or a system problem, the human component is shown to be an essential consideration in many system implementations and all analyses of need. Accordingly, a perceived weakness of the systems engineering discipline – an appreciation of human factors – is given final prominence in this investigation report. This establishes systems engineering views of humans, both inside and outside the system boundary. The often essential contribution they make to system performance and the part they play in forming stakeholder needs are shown to lie on a continuum of properties that systems engineering practitioners must be cognisant of. This understanding is also important for appreciating how far systems engineering may reasonably go toward addressing the more subjective needs of social systems. Past, questionable approaches to this might have benefited from a deeper understanding of systems engineering's principles, with popularised 'soft' problem approaches still able to learn from their originating discipline of systems engineering.

This critique of systems engineering principles and practices, and the path followed during its programme of investigation, draws to a close with *Conclusions*, §11. This summarises the salient outcomes, issues raised and possible directions of future enquiry. Some personal reflections finally close this text.

1.2.6 Themes

The individual chapters of this thesis address as far as possible self-contained topics. Each is introduced through Prologues that provide selected seminal observations on the subject of the chapter. This is followed by a synopsis of the chapter's structure and content. Chapters conclude with a set of outcomes that highlight issues or conclusions that are outputs of the research, or/and are used as input to other chapters.

Forward and backward references within each chapter establish threads that form themes that emerge across several chapters. They are indicative of the issues most likely to influence the continuing evolution of systems engineering. Principal amongst these are:

- *System life cycle management*. This appears in §2 and remains as a theme through to the end, where in an MOD context it is described as Through Life Capability Management;
- *Capability viewpoint of a system* – this also starts in §2; it is developed under process and life cycle management; it takes the spotlight in defence acquisition in §9.3, where the capability-based approach is an important dimension of defence acquisition;
- *Architecture* – systems reasoning first considers concepts related to architecture; it is built on in §4 and taken up under design in §10.4, where it is considered in a defence context according to the scope and contributions of architecture frameworks;
- *Systems engineering work products* – they are an intrinsic attribute of process, appearing first in codification of systems engineering in §4; they contribute to decision gate criteria in life cycle management, contribute to maturity of an organisation's systems engineering capability in §6 and ultimately to Through Life Management Plans in §9.5;
- *Systems engineering organisational capability* – this is first presented in the emergence of systems engineering; it influences system element selection at trading interfaces, in §6 is a cardinal facet of the systems engineering triptych; is apparent in the MOD context for systems engineering in §7 and is fundamental to a supportive enterprise context for IPTs in §9;
- *Systems engineering competence* – the importance of systems engineering competence starts in §2 with the origins of systems engineering; it is developed as an element of discipline formulation, and appears repeatedly throughout the defence acquisition setting in §7 and §8, and IPT systems engineering advancement actions in §9 and §10.

2 AN ENDURING DISCIPLINE

2.1 Prologue

At the outset of this research the IEE President, Prof John Midwinter asserted in his inaugural address in 2000 that looking back to 1960s engineering 'the challenge was usually to try to find a technical solution to a well defined problem ... Innovation ... led to new components, a few of which combined to solve the problem. Today, one finds that the major activity is much more market focused ... The solution is likely to involve a complex system that draws upon a wide range of different disciplines for its implementation ... The level of complexity now possible has also triggered a new approach to the design of complex systems ... with most of the system design being carried out in an abstract, almost mathematical way and expressed through generalised ... system upper layers' [Midwinter, 2000]. This analysis was in support of an argument for fresh views on engineering and management education and training. It was illustrated by a three-layer model. This comprised society and business, overlaying and fed by technical and social structures, and underpinned by a foundation of technology. It showed the hierarchical spans of liberal arts, business and law, social sciences, engineering and physical sciences. Only engineering spanned all three of these levels.

As a précis of a shift from 'bottom-up' towards 'top-down' engineering – a move from a technology-opportunity approach to one driven by the needs of society – and as a synopsis of how systems engineering acts to meet this challenge, it is exemplary. As explicit advocacy for a discipline of systems engineering, it was silent.

Despite there now being a more widely recognised set of practices, even an academic and industrial discipline, termed systems engineering, would this have been written differently seven years on? This chapter reflects on whether at the close of this investigation a strategic engineering address of this nature would have explicitly referenced systems engineering, and whether its role, qualities, and commercial and social benefits now single it out as a widely-recognised discipline, or even as a profession.

Systems engineering as a discipline tantalisingly offers what no other branch of science, technology or management has yet delivered: a coherent, well-integrated body of business-related action that fuses technology, engineering, human sciences and management, and that brings a holistic approach to the creation of artefacts used to beneficially intervene in the real world. Such a trans-disciplinary model is not novel; 'subjects ... studied in no set order must now bring together to form a unified view (*sunopsis*) ... with one another and with the nature of what is' [Plato, 375BC, 537c1-3, 6-7]. Plato's *sunopsis* is a central dictum of systems engineering: a holistic understanding and capacity united from diverse technical communities and viewpoints that enable the creation of coherent designs intended to meaningfully interact with what is.

This bringing together to form a unified view challenges not just the individual; for systems engineering is, in its unification of subjects, a team discipline. As Ramo put it, 'a good systems-engineering team combines individuals who have specialized variously in mathematics, physics, chemistry, biology, other branches of physical science, the many branches of engineering, economics, political science, psychology, sociology, business finance, government, and so on. The systems engineering team attacks the interaction problems among these specialties that characterize any practical problem' [Ramo, 1998, p.20]. Across team members this coordinated attack requires a sound knowledge of engineering, a refined skill for team management and a gift for systems reasoning. At its most succinct, systems engineering may thus be summarised as 'the generalship of engineering' [Gordon, 1978].

At the *Boeing Lecture 2007*, NASA Administrator Michael Griffin said of this generalship that it 'is the art and science of developing an operable system capable of meeting requirements within imposed constraints ... Many more [engineering] disciplines are weighted and considered and balanced, one against another, to produce a coherent whole that is not dominated by the view from the perspective of a single discipline. System engineering [sic] is about tradeoffs and compromises, about generalists rather than specialists ... System

engineering is not about the details ... No amount of care in such matters can make a poor design concept better. System engineering is about getting the right design' [Griffin, 2007].

Systems engineering would thus seem to be a well-tuned disciplinary product of its time. But in any discipline, whether it is established or still finding its place in the social and technical order of things, change and adaptation are necessary. So this chapter is about change; about how systems engineering emerged from a plethora of contributing factors and pre-existent, disjointed principles and elements of practice; about reassessing whether the catalytic effects of the Cold War assembled and conveyed the discipline to best effect; and, with the benefit of hindsight, it is about the balance of the composition and contributions that now characterise systems engineering.

This changing complexion of systems engineering faces the inertia of 'traditional' approaches. Max Planck somewhat cynically observed that "a new scientific truth does not triumph by convincing its opponents and making them see the light, but rather because its opponents eventually die, and a new generation grows up that is familiar with it" [quoted in Kuhn, 1962, p151]. However, whilst the greatest champion of recent systems engineering re-codification and revitalised application may be time – time for practitioners and beneficiaries to recognise the technical and business value of change – it is like any business change programme, it still needs the directive hand of its designers to ensure that it is deployed and employed effectively. Without this impetus it may stumble in a disciplinary contest in which it faces displacement or being subsumed by other approaches.

These matters – whether systems engineering will stay the course – are thus considered in this chapter in order to analyse whether as a discipline systems engineering is indeed a sound foundation on which to build MOD revitalisation actions, as advocated in Smart Acquisition [MOD, 1998], in the *Defence Industrial Strategy* [DIS, 2005] and in *Defence Acquisition Change* [EAC, 2006].

2.2 Synopsis

This chapter's title could have been cast as an interrogative. There is a body of opinion that sees systems engineering to be a C20th anachronism, owing it origins to a short-lived state of technical, political and economic affairs that are now history. If this rang true, if the discipline were truly in its declining years, then the propositions advanced in this investigation would merely be of passing interest and of little practical importance. They would be of limited value in promoting or advancing IPT practices under a banner of systems engineering.

This chapter thus begs the question it may appear to imply. It omits the interrogative and opens this research with an affirmative. Systems engineering does, judged against the conventional ebbs and flows in science, technology and commerce, appear to be an enduring discipline – one set to substantially impact businesses beyond its past homeland of defence and aerospace.

Throughout this thesis it is advanced and repeatedly demonstrated that systems engineering is a composite discipline constituted from three realms – systems reasoning, engineering and management – that unite to form a discipline of mounting significance across many areas of trade and commerce. This chapter looks at aspects of how these three components combine; what novel advances their combination brings; how this is codified and promulgated; and how in the particular, yet exemplary instance of MOD IPTs its principles can be translated into more effective practice.

The ground for this is laid by first examining how systems engineering came together during the latter half of the C20th. This begins with an essay on why and how systems engineering arose. It shows that it has been an uneven ride for systems engineering. It has seen unquestionable successes but also evident failures; these stem from recognised strengths but also unforeseen weakness. They have led to successive waves of recognition and disillusionment, and attendant, fluctuating progress.

This then leads into how systems engineering has been seen by different parties: a diversity of viewpoints on a complex, holistic discipline. To calibrate progress, important issues concerning systems engineering's identity and recognition are considered. An assessment

against these criteria then resolves whether systems engineering has truly attained the status of accepted discipline, and whether it may ascend yet further to be a recognised profession. This analysis of its standing and prospects reflects on systems engineering's perceived value and the degree to which it is accepted by society and organisations.

To close this review of systems engineering as a dependable springboard for business advancement in general and for MOD's improved acquisition of military capability in particular, the way it meshes with and serves other organisational and commercial functions is considered. First its engineering and then its management contexts are defined, with illustrations of how they act to shape systems engineering. This results in models that rationalise its harmonisation with these two longer-established regions of practice. It shows that systems engineering actively addresses weaknesses that have existed in past views of business process; weaknesses still unrecognised by many organisations or that are being poorly resolved by others.

Some of these weaknesses exist in MOD's Smart Acquisition practice and it is seen in this investigation that, with some confidence, they may be countered using refined expressions of systems engineering principles and their translation into IPT practice.

2.3 The Rise of a Discipline

2.3.1 The Emergence of Systems Engineering

Since its inception over fifty years ago, systems engineering has been considered by many as a practice-based activity rather than an academically-based discipline; as a methodology rather than as a discipline with a foundation in scholastic texts and associated formalisms.

It is true that unlike traditional engineering disciplines it is not intimately governed by a set of fundamental phenomena based on physical properties and relationships, although its role is most certainly to knowledgeably orchestrate such phenomena. Thus, whilst academically-specialised implementation technologies and branches of engineering (including human factors engineering) are an essential foundation for systems engineering, the intellectual focus of systems engineering is on their skilled trading, one for the other, to create original properties and behaviours that transcend the limitations of individual branches of engineering. Systems engineering epitomises engineering in the round.

Conventionally, these implementation technology-based branches of engineering have largely been treated in academia and research as self-standing directions of intellectual pursuit. Each is a mechanism for translating the laws of physics into practical and profitable elements of some greater value. Drawing on system elements implemented according to diverse technologies, and set in the management context of business enterprise and its project teams, it is systems engineering's role to institute and conduct practices that achieve the vision, understanding, assembly and sustainment of this greater engineering value. With this role in mind, this report opens by considering how the one major branch of engineering that does not depend on an individual class of science or technology phenomena came about.

Many research studies lay their ground by referring to a survey of the subject's literature. Considering systems engineering as a discrete academic subject, its literature provides limited pickings from over half a century of its seminal texts or from the subsequent elaboration of their themes. This may be due to its practice-based dimension; for systems engineering emerged *de facto*, it was not assembled from a basis of theory. It was synthesised naturally from existing practices that exhibited a mutual dependence for far longer than systems engineering's brief, recognised existence.

There have been many accounts of how systems engineering came into being, [Hughes, 1998], [Ramo, 1998], [Brill, 1998], [Checkland, 1999]. For some practitioners these events are familiar and lie substantially within their career spans. Most accounts neutrally convey a litany of events or, each in their way, are an *a priori* recounting of history in preparation for particular ensuing arguments. This section conforms to the latter category. The emphasis in its partial repetition is intended as a setting out of a stall of systems engineering issues and opportunities considered during this investigation.

Looking at the last half century, if there is any individual worthy of the title “father of systems engineering” then it is most probably Simon Ramo. It is his summary of the engineering profession that this account of systems engineering commences with: ‘engineering is generally defined by the dictionary and by members of the profession as the application of science and technology to the needs of society’ [Ramo, 1998, p 20]. Defined in this manner, engineering has self-evidently been present for countless millennia. It has existed since the dawn of tool-making man. However, once engineering’s conceptual span crossed the threshold of the single mind, and the ensuing diversity of structure and assembly of fabricated technological elements exceeded the knowledge and skills of the individual artisan, engineering changed. It then passed from the engineering polymath into the realm of social structure and separated contributions. At this point technical architecture and social architecture became interdependent. It was still not, however, systems engineering. It was probably what has become arcanelly referred to as ‘engineering of systems’, and it essentially remained like that until the mid C20th.

The increasing challenges of organised group action and common purpose, of clarity of communication and common understanding, and of coordinated modes of activity and value exchange have progressively placed engineering action in the realm of social and trading structures.

With growing social expectations of evermore creative interventions in the world, engineering had to draw on fresh strategies. These needed to more effectively achieve common purpose and action, and provide value and optimum solutions from an ever-wider range of technological opportunity. Out of this, systems engineering emerged as a fundamental and formative engineering influence, and an agent of change for enterprise and project management.

Over many millennia ‘engineering’ and ‘management’ had harnessed technology and human resources to establish agricultural schemes, natural resource structures, urban complexes and transportation networks that contributed to population increase and dispersal, and as a consequence to the viability of diverse regimes of social and political structure. In the uncertain, competitive and opportunistic circumstances that arose from this, it was the security of nations and the welfare of souls that in particular drove man to apply engineering with increasing creativity and vigour. Medieval fortifications, European Renaissance cathedrals, Cold War missile systems, Mesopotamian ziggurats, a succession of Chinese defensive walls, Middle Age Mayan temples, Iron Age forts and much else provide evidence of the cutting edge engineering and management of their times.

With this history, it is not surprising that a supreme threat to security and welfare – the Cold War missile defence systems – should have been the cradle for the explicit union of engineering and management, and for the emergence of systems engineering. Unprecedented steps of political, communal and technological change had finally conspired to cross a complexity threshold in man-made systems. They precipitated a defence acquisition crisis that was the catalyst for the inception and rapid evolution of a new discipline in the early post-war period.

A history of ‘systems engineering’, summarised in Figure 2.1, shows evolving engineering achievements and particular events that have led to the present-day discipline. Presented on a roughly logarithmic time scale, it shows the watershed of systems engineering’s recognition to be around 1950; the time when cutting-edge science minds at the Macy Conferences [§3.4.1] were articulating the ubiquitous nature of systems science. Prior to this, systems engineering, such as it was, had been conducted implicitly and indiscriminately. After this, structure, discipline and a specific identity increasingly characterised its conduct.

Recorded history has left limited information on the engineering that led to many past achievements, but the surviving artefacts frequently speak for them. It is clear that ‘engineering’s’ application of technology and ‘management’s’ marshalling of human and natural resources underlie the design and building of the great civil and military engineering structures such as the Babylonian, Saharan and Mayan irrigation canals and water cisterns; the Chinese, Egyptian, European, South American and Sumerian temple complexes and burial structures; and the harbours, roadways and fortifications across the planet. Within the knowledge, resources, technology and transportation limitations of their time, these man-made entities were pushing the boundaries of human ingenuity and discipline. Clarity of

purpose, complexity of structure, range of technological contributions, creation of novel characteristics, magnitude and management of the task should all be seen as fully the equal of present-day systems engineering achievements.

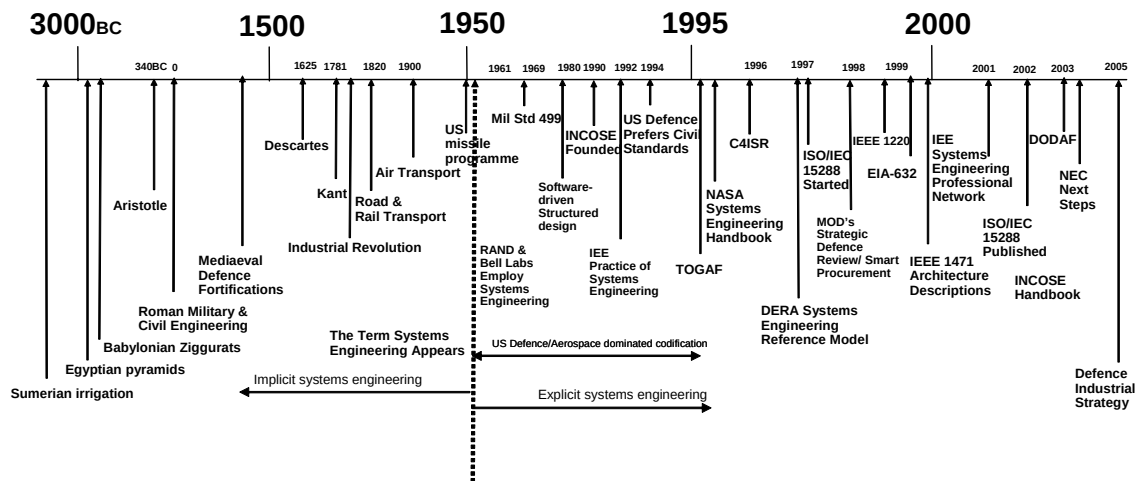


Figure 2.1 A 'Logarithmic' Timeline of Systems Engineering

Detail on how and by what principles yesterday's 'systems engineering' was conducted was well within the capability of many of these designers and builders, but little survives. It was either never recognised as needing to be recorded or was not deemed sufficiently important for it to have survived. Engineering was probably more communal craft than discipline, led by master practitioners who drew on examples (good and bad) of predecessor implementations. Engineering as such appears to have depended on processes and practices that were largely handed down by rote and apprenticeship, and in some cases bound into rituals of procedure, or protected within closed professional societies.

The earliest surviving records show engineering to have been an important contributor to warfare. Defensive state barriers, and city and harbour defences still exist as testament to remarkable engineering achievements of their time. However, military engineering was not concerned only with static structures. Over a long period siege engines, mobile fortifications and ballistic engines were elements of strategic capability. Direct evidence of the refined skills required to design and supervise such military artefacts is hard to find, though some examples, such as da Vince cartoons, reveal the designer's mind as it modelled and communicated military system concepts and construction.

Before major civilian undertakings made their mark on the 'engineering' term at the start of the C19th, the connotation of engineering was entirely in the military application domain. In 1797 in *Encyclopaedia Britannica* it was the only sense of use offered for the term: 'ENGINEER, in the military art, an able expert man, who, by a perfect knowledge in mathematics, delineates on paper, or marks upon the ground, all sorts of forts, and other works proper for offence and defence... Engineers are extremely necessary for these purposes: wherefore it is requisite, that besides being ingenious, they should be brave in proportion... Their business is also to delineate the lines of circumvallation and contravallation ... After making a faithful report to the general of what is a-doing, the engineers are to demand a sufficient number of workmen and utensils, and whatever is necessary' [EB, 1797]. In the language of the day this sets out the engineer's attention to stakeholder communication, model-based design, the planning of enabling resources, and the dependence of all this on distinctive knowledge and skills.

For the most part major engineering artefacts had been static spatial entities, not needing to possess time-critical features. However, by the mid C19th timing, related to critical interactions with the environment and to internal mechanism, changed the nature of systems and the way they were designed. Systems, structured as time-dependent networks with parallel paths and feedback loops, required fresh approaches, and new understanding. Previously external human elements had controlled internal synchronisation and supervised temporal criticalities

of in-service behaviour. Now transportation and communication – roads, postal networks, canals and railways – depended on the planning and implementation of time critical system features. Temporal as well as spatial systems understanding and modelling became key to effective operation.

The interconnected elements in a railway system/network required routing logic, control structures, critical timing, safety analysis, technical standards, service delivery requirements and maintenance regimes that prognosticated the inevitability of a unifying discipline that could orchestrate the engineering of systems. By 1866 the newly instituted journal *Engineering* announced that 'the title of this journal has been chosen ...as typifying the business, art, and profession of the Engineer'. However, despite possessing features in common with today's systems engineering teams, Victorian engineering entrepreneurs lived in an era of heroic inventors and charismatic leaders, where individual and gifted minds could conceive of, design and lead the construction of large-scale complexity. This faded with the growth of major industrial corporations and commerce structures that depended on the very transportation and communication structures that engineering was building, and also on the growth of implementation options offered by expanding scientific knowledge and its pursuant technology skill base.

Nonetheless, heroic inventors and charismatic engineering leaders still held sway in both World Wars but they became evermore dependent on the integration of diverse skills in the growing engineering teams they led.

Post-World War II defence engineering faced daunting problems as approaches to the control of increasing size and heterogeneity in individual development, manufacture and deployment programmes had to change. Notably across the US, the need to marshal teams of scientists, engineers and managers changed the complexion of defence industry projects. Attention shifted to engineering management as much as to the technical engineering being managed [Hughes, 1998].

Out of the attendant "military-industrial-academic complex" [Fulbright, 1966] in the USA, systems engineering emerged in a recognisable form. The pressure from long-range nuclear attack led to the Atlas intercontinental ballistic missile program. It resulted in a national imperative predicated on extraordinary technical complexity, an almost unparallel array of technology and disciplines, and interacting echelons of systems-of-systems. Monolithic, vertical organisations could no longer meet the technical and timescale challenges.

An influential catalyst for the ensuing change was the so-called Teapot Committee. It was a strategic US response tasked with assessing the capability of prime supplier organisations to provide to government, principally the DoD, system complexity unlike previous system developments. In 1954 it recommended that there should be a systems engineering contractor comprising "unusually competent" engineers and scientists to technically and managerially effect a whole programme. It was the clearest statement that system complexity and stakeholder need had conspired to push business and technology across a barrier that required a reformation in the organisational principles and practices associated with defence system contracts.

The foundation of a new paradigm had been laid; one in which engineering practice and business management practice could no longer be considered in isolation. It signified a coming together of two major areas of discipline in a way that had no precedence and from which would emerge a radically different approach to handling complex, large, high-risk technical ventures. It offered a new way of approaching unparalleled technical complexity.

Although individually much of the required principle and practice in engineering and management was already in place a new, common influence was brought to bear on them. It arose from a mood of Post War cross-discipline technical advancement and intellectual liberation. It was based on the attributes and characteristics of systems in general, both theoretical and actual. It led to related ways of analysing problems and synthesising solutions, and it impacted and drew from many disciplines. The consequences of this sea change in scientific thinking, its application and its influence on systems engineering are considered further in §3.4.1.

Without this providential 'lateral-thinking' about systems, and the recognition of its value over a range of natural and social sciences, it is unlikely that the cross-disciplinary practice of

systems engineering would have emerged as quickly as it did. It was not just external pressures that were encouraging engineering and management towards a new, holistic relationship. There was also an appreciation of how a common systems-oriented approach led toward a more powerful, more structured and more coherent body of principles and practice: systems engineering.

The essential catalyst of systems engineering – ‘system-reasoning’ – was a low key, yet hugely pervasive scientific revolution. A ‘Systems Age’ had arrived without any of the drama that accompanied other C20th advances of equal merit and less widespread influence. The surge of etymological sources for the word ‘system’ [OED, 2004] from 1951 onwards is probably the most revealing testimony to this muted revolution. ‘System’ was now a commonplace mode of viewing natural and social sciences, and with time much else as well.

Both engineering and management drew from this reasoning in terms of systems. It established a ‘cross-engineering’ perspective that readily meshed with evolving ‘scientific management’ practice. A disciplining system perspective brought insights and fresh interpretations to existing practices. Systems engineering rapidly offered relevance, coherence and a distinguishable body of practice that had not previously existed with any semblance of an institutionalised discipline. It was ideally suited to meet the composite of post-war technical and business challenges.

With the escalation of Cold War imperatives there had never been such economic weight, national security concerns and political urgency driving matters, nor such industrial and technological variety to draw on. Time and tide were indeed ready for engineering and management change. It was an era of scientific liberation when many began to think ‘systems’.

Thus, three essential components – systems reasoning, engineering and management– were now alloyed and systems engineering entered the discipline scene.

The landmark that signified systems engineering’s arrival is generally cited as the Atlas intercontinental ballistic missile programme. Its scale of enterprise was unprecedented as, according to its political masters, was its security implications. It is reputed to have drawn on 18,000 scientists, engineers and technical experts; on 170,000 people from office to factory floor across twenty-two industries; and on a supply chain of 17 primary contractors, trading with 200 subcontractors, who in turn depended on a supply base of 200,000 businesses. The customer was represented by 500 military officers. It marked a new regime of political, business and engineering amalgamation. From Atlas, and an era of similar, staggeringly progressive military programmes, the ‘mode of management known as systems engineering emerged’ [Hughes, 1998].

The company of Ramo-Wooldridge led the systems engineering of the Atlas program and it was Simon Ramo, a member of the formative ‘Teapot Committee’, and his staff who helped establish systems engineering as a discipline. They establish an organisation capable of concurrently developing the designs of the system-of-interest, its subsystems, the related command and control systems, an infrastructure of training, and the necessary enabling systems of development, manufacturing and test to undertake this.

As was recognised at the time, the engineering management often proved more difficult and complex to solve than specific technical problems. In response to this, the 1960’s began to witness the progressive codification and rationalisation of the systems engineering practices seen in Atlas and in other military-industrial-academic programmes. A cohesive discipline of systems engineering was forming. This led to systems engineering appearing in the 1960’s as a subject of military standards [AFSC, 1966].

Systems engineering quickly started to be applied outside defence and by 1965 ‘the term ‘systems’ was becoming popular for describing large-scale, non-military industrial projects’ [Ansoff, 1965]. As NASA’s Project Mercury launch vehicle, Atlas also ensured systems engineering’s entry into the ‘space race’ and a recognised role in the practices associated with international aerospace ever since.

Its defence/aerospace powerbase, however, had established a potential weakness. Systems engineering had inadvertently and perhaps unavoidably become typecast for large project performance undertaken by large organisations, typically with government as the overall

customer. The particular 'techno-socio-economic model' that evolved conferred limitations on systems engineering's application domains and modes of use, and inhibited its natural transfer into other areas where comparable complex system challenges were becoming commonplace.

Thus, although U.S. President Johnson issued an Executive Order in 1965 that required all federal agencies to adopt the Pentagon's systems engineering approach [NASA, 1972], outside of the military and aerospace domains its track record was, and to this day remains, less successful. Its early advocates such as Schriever, 'the architect of the US Air Force's ICBM and military space program' [USAF, 2005], and Ramo found that attempts to apply the principles embodied in systems engineering to solve major social system challenges were frustrated [Hughes, 1998].

The step into the life cycle management of dynamic, ill-defined and loosely understood social systems-of-systems was beyond the learning and experiences that had been gained from time-pressured, essentially linear defence developments with their clear-cut, if complex technical goals. To compound matters, the negation, compromise and consultation necessary for 'soft' civil problems, such as housing, health care, social amenities and education, were a world apart from the military authority that had propelled the introduction of systems engineering techniques into defence.

Despite championing systems engineering – promoting it as “the design of the whole from the design of the parts” – Ramo and others found their integrating discipline did not, as hoped, bring science and engineering more fully into the solution of society's problems. It faltered and retreated back to defence and aerospace. Ramo's *cri de coeur* at its failure to live up to expectations can be read as a subtext of *The Systems Approach*, [Ramo, 1998], which he subtitled *Fresh Solutions to Complex Civil Problems Through Combining Science and Practical Common Sense*.

Nonetheless, systems engineering had illustrated that challenging technology projects depend as much on management as on the engineering and technology being managed. It had shown the relationship between technical systems, business management and inter-organisational trading. It set a new, enduring pattern for acquisition and a new era of government/industry relationship, at least in the defence, aeronautics and space application domains.

2.3.2 Influences and Interpretations

Systems engineering was thus an ideology that grew out of the inability of mechanistic engineering and management to cope with complexity. It evolved from these more long-standing disciplines to meet the challenges seen in many present-day systems. These are characterised as being technological diversity, functional and compositional complexity, imprecise needs, multiplicity of contributors and, more recently, a multinational diversity in the associated acquirer and supply structures.

Systems engineering proved equal to these challenges and progressively became acknowledged as a definable and teachable discipline. It was evidently different in nature to other engineering disciplines. It did not immediately align with traditional engineering disciplines regarding its intellectual content. Its approach to education and training was more practice based, and the skills profile of its practitioners and its career structure were hardly comparable. Success in systems engineering evidently benefited from aptitudes and ways of thinking in its practitioners that distinguished it from other engineering disciplines. In these and other respects, systems engineering was a distinguishable and different engineering discipline.

Yet despite these distinctions systems engineering is concerned with actions and objectives that are discernable in most, if not all, engineering disciplines. In this regard it can be interpreted as a common denominator of engineering thinking, providing a conceptual and structural essence above the technological detail. It binds the many engineering 'stovepipes' – as taught by academia, promoted by professional bodies and frequently practiced in industry – bringing them together in a mutually-contributive way.

Systems engineering is thus subject to an engineering dichotomy. From one perspective, it is a common, bounded, self-consistent body of systems reasoning and pan-engineering practice,

standing in its own right as a body of knowledge and discipline to be practiced. In contrast, viewed from a wide range of engineering and technology concerns, it is also an indivisible and natural aspect of each and every specialist engineering discipline.

These two views are only reconciled if systems engineering is accepted as a common and unifying layer of engineering action that intimately intermeshes with, and contributes to, these 'stovepipes' of engineering discipline. As Ramo is quoted; "systems engineering is inherently interdisciplinary because its function is to integrate the specialized separate pieces of a complex of apparatus and people – the system – into a harmonious ensemble that optimally achieves the desired end" [Hughes, 1998, p67]. Thus, engineering specialisms, including human factors engineering, depend on the integrative nature of a distinct systems engineering discipline, which is at the same time part and parcel of each and every engineering discipline [Figure 2.5].

This essential duality of systems engineering's existence – being a separate, overarching (or, less emotively, underpinning) discipline – yet also being an integral part of any implementation technology-based engineering discipline is a profound, subtle and sometimes sensitive state of affairs. Establishing a culture in which a cohesive juxtaposition with individual technical specialisms exists has been, and even remains, a distinct challenge for systems engineering. Its resolution is considered further in §10.5.1 in the exemplary context of human factors.

If systems engineering's role as interdisciplinary engineering *primus inter pares* is to be effective, it has to be conferred with a degree of managerial authority. In no respect can it focus on solely technical principles and practices. To do that would restrict its value delivery. For systems engineering to fully contribute to the organisation, it must be an intimate part of a management regimen for delivering product and service complexity in commercially complex contexts.

Thus, of necessity, systems engineering sits alongside project management to oversee the harnessing of diverse technologies and guide them towards business goals. In the wider organisational management context it influences and should be nurtured by enterprise management, with its responsibility to establish, provision and mature an organisation's systems engineering capability.

The engine of US post-war economic and scientific advancements encouraged a bottom-up, technology-led approach to design and development. Technological excellence and performance supremacy were all important. If technologically it could be done, then it would be done – a matter of national power and authority. Subjected to this influence and with the self-confidence it brought to developers, systems engineering was firmly associated with the development stage of a system's life cycle; with systems engineering practitioners seen as a class of development or product engineer.

This development-centric view of systems engineering had its consequences on language, models of action and methods. Despite systems engineering's avowed principles of harmonious, through-life decision making and avoidance of 'over-the-wall' attitudes, many through-life system decisions were still being made in the comparative isolation of development teams, who presumed to impose through-life system judgements on other responsibility groups for the system-of-interest and for its enabling systems at all stages of a system's life cycle.

This development-centric view of systems is betrayed in IEEE 1220 [IEEE, 1998], see Figure 4.5, in which a system's physical structure is defined from a product design-dominated perspective, with through-life considerations handled as labyrinthine, adjunct parts to the 'development product'. This completely missed a service or capability viewpoint of the system. It conditioned a generation of system designers, not just in the US but across the international community, to thinking product and it hindered acceptance of systems engineering as a set of whole life practices.

The US dominance, seen to the left of Figure 4.4, therefore meant that systems engineering conformed to expectations regarding culture, perceptions of risk, goal setting, business strategies, economic inertia, availability of resources and success criteria that strongly characterise the wealth, ambitions and power of the USA. These did not, however, necessarily translate into the circumstances and constraints faced in other nations.

In addition, the publicly-funded, government customer view, e.g. DoD, NASA, that accompanied systems engineering's rise unavoidably emphasised a model of action driven from clear (hard) contractual requirements. This proved to be a powerful influence on the definition, vocabulary and public perception of what systems engineering is and to whom it applies.

The national acquisition authority-dominated style of application limited attempts to relate systems engineering to small and medium enterprises, SMEs. The tenor of its descriptions, terminology, training, methods and tools did not favour smaller organisations. Rather than presenting a set of widely-applicable technical and management processes that characterised the conduct of smaller businesses, they were couched in a style that conveyed a resourcing overhead and threshold of applicability that appeared inappropriate for SMEs. Little attempt was made to accommodate the lower orders of the supply chain, despite the obvious value of achieving coherent, well-integrated vertical acquirer/supplier chains. This situation remains an impediment to this day to systems engineering's wider acceptance.

These and other systems engineering failings were seized on by detractors and competing disciplines. Individual implementation technology-based engineering disciplines, seeing the 'territory' to be part of their concern, also defined common system viewpoints. Software engineering in particular found that its natural involvement in functional abstractions invited definition of a common, technology-independent system perspective. Indeed, as each engineering specialism moved towards need-driven, service-orient considerations of its purpose, the more it found itself defining the common systems engineering perspective and methods.

Each of these interpretations suffered the same weakness: that the conditioning of their technology confines, culture, language, implementation characteristics, methods and tools appeared (and differently so) in each definition of a systems viewpoint. The software engineering, the electronic engineering, the mechanical engineering, the human factors engineering viewpoints did not align with each other, nor with any recognised common systems engineering viewpoint. Thus essentially replacing the term 'software' in the software engineering viewpoint by the term 'system' misled rather than assisted.

Just as individual engineering specialisms attempted to define systems engineering, so did project management. The synergy between the two disciplines made this an obvious re-scoping opportunity for project management. During the 90s, project management defined much of the technical control detail of engineered artefacts, e.g. configuration management, the information (documentation) management, and defined the life cycle management of stages. Again, the result was an incomplete and partisan view.

From all this systems engineering found itself influenced as much by international tensions, political imperatives, government trading conventions and industrial business climate as by the technical efforts of its practitioners and academia. As a result its path has been somewhat erratic and uncertain – a factor considered next.

2.3.3 A Faltering Journey

After a promising and prominent start in the 1950's and 1960's, systems engineering has had a difficult journey and many see it continuing in this vein. It has not yet fulfilled expectations within its roots of defence/aerospace and it has failed to live up to ambitions when applied to other industrial and social domains. Shortcomings in its effective application have been documented by several authors, [*Petroski, 1994*], [*Le Lann, 1997*], [*Bar-Yam, 2003*].

The following observations on why systems engineering appears to have had such a roller coaster existence point to areas that this investigation's re-interpretations of principles and re-statement of practice have addressed. Accordingly, this section details some of systems engineering's ill-fortunes and adverse outcomes.

The following US Defense Acquisition University 'eulogy' [*DAU, 2004*] offers a prognostication on what has been called the "traditional systems engineering" approach [*Rhodes, 2004*]; and what failure to change with the times (which it is has done) would have amounted to. In its sardonic style, it records some milestones in systems engineering's first half century:

' – Military development of Ops Research in the 1940s was instrumental in shaping its youth

- First formal education attempted at MIT in 1950
- SE became the darling of NASA & military systems in the 1960s; its processes imposed on prime contractors
- Its many offspring were Chief Systems Engineers, Directors of Engineering, VPs of Systems Engineering as well as many Systems Engineers
- It ... appeared as Mil-Std-499A; grew in stature & acceptance
- SE became check & balance in massive engineering efforts
- It suffered an identity crisis ... [and] in the 1980s SE attempted to re-identify itself as Mil-Std-499B
- Later cashiered out of the military, it found civilian employment as an IEEE standard
- It wasted away with an illness brought on by rejection of its processes & bastardizing of its techniques & methods ... died a lingering death in the 1990s & was replaced by IPTs...
- Systems engineering is survived by & remembered by a cult of old systems engineers who still practice their now meaningless rituals
- It was almost 70 years old at the time of its death; no services are planned.'

There are some basic truths in this eulogy. Elements of the "cult" do linger on and still cast adverse influences on an otherwise changing and youthful discipline. However, overall this litany of events is fortunately anachronistic and blinkered. It may summarise systems engineering to its "end of the beginning" around 1996: the period that equates to systems engineering's 'infant', 'whining school-boy' and 'lover' ages [*Shakespeare, 1599*] but as a balanced history it omits that the "IEEE standard" [*IEEE, 1998*] was complemented by an equivalent civil standard [*EIA, 1998*] and that both these have been supplemented by a non-development-centric, non-US-culture-dominated ISO Standard [*ISO, 2002*].

In the event systems engineering has metamorphosed into a healthy, forward-looking and commercially relevant discipline. However, it needed influence from outside the US to overcome its "identity crisis", to shake it from a stereotyped approach and to be equal to its fast-changing environment of origin. It did move on; ceasing its introspective "whining" about underappreciated value, and is now concerned with how better to serve the business community from which it sprang.

In fairness, it is not unusual for arrivals on the discipline scene to suffer "identity crisis". Continual refinements of approach, territorial expansion, overconfident problem seeking, rebuttal by other disciplines, and too-slow accommodation by existing social, professional and academic order can cause early success to be questioned. As systems engineering's "darling" status waned, it attempted to better resolve and assert its identity, purpose and scope. Through the 90's its perceived enigmatic nature, accompanied by endless debate to resolve its identity, was particularly detrimental and was highlighted by detractors. Under challenge, the systems engineering community failed to authoritatively and ambiguously define an identity or scope with acceptable clarity. This failure of presentation has profoundly handicapped the discipline and it is only just recovering from it.

Nor has the ready applicability of systems engineering to issues of detail, including many particulars of technology, helped matters. It enjoys highly versatile applicability of its systematic approach to a considerable range of compositional detail, yet this leads many minds to see its goal to be the design of engineering/technology detail, with strategic technical decisions something that a class of 'architects' pursue; a matter considered in §10.4.1.

Whilst there is a need for different technical and management perspectives of systems engineering, undue diversity in its models and terminology has arisen. Against a mix of cultural interactions and changing paradigms, the drive to harmonise systems engineering process standards is underway but has far to go. This has yet to be complemented by the continuing alignment of standards between disciplines – initially systems engineering with software engineering, followed by project management and human factors. There is also an interface with industry sector standards, and there is a clear need to provide systems engineering relevance in specific product markets and application sectors.

This purpose/identity/scope weakness has encouraged those engaged in organisational improvement initiative and in method-development initiatives to variously commandeer areas of systems engineering ground. Using good branding, strong promotion and by engendering

organisational mandates, they have successfully made inroads. The resulting methods have made contributions to the wider discipline but are not adequate substitutes. Quality Function Deployment, Six Sigma, Systems Dynamics, Design for Six Sigma, Computer Integration of Requirements Procurement, Logistics and Support (CALS), Lean Systems, House of Quality, organisational maturity attainment models, e.g. CMMI, Management by Objectives, Total Quality Management, Lean Enterprise are all examples of past and current initiatives that have a strong relationship with systems engineering. Many flourish and fade. They do, however, exemplify that to 'promote' a subject it needs to be 'attractively packaged', be recognisable and readily comprehended, and have almost immediate benefit to its consumers – society, businesses, academia and individual practitioners. Systems engineering has found the going hard in this regards; a point made by several authors [Emes, 2005].

The mindset of defence acquirers encouraged top-down, hierarchical system views that echoed management control in military commands. Echelons of decision making favoured and promoted sequential and waterfall representation of life cycles, and concealed the richer approaches required for overall management of system complexity. This was evident in the 'objective rigour' that McNamara brought into DoD in 1961 planning and design, with his penchant for system analysis. It appealed to the political controllers of DoD, but it imbued complex defence system reality with a highly optimised simplicity that was frequently misleading.

Early success also brought unwanted baggage. In its vision of the future, INCOSE observes that 'the formalism of systems engineering process has brought some untoward consequences. It has engendered a perception of burdensome, heavyweight efforts, leading to unjustified cost and time overheads' [INCOSE, 2020]. Faced with systems engineering's association with overbearing defence/aerospace approaches, many businesses and government departments turned to methodologies that benefit from partial implementations of its totality. In this way they broke free from the perceived peremptory, fastidious and unwarranted prescription of defence application, and gained immediate and seemingly adequate benefit with reduced effort.

Non-defence users in particular evolved important areas of systems engineering in more palatable ways. Some were badged as systems engineering synonyms and applied to a de-scoped systems engineering. Their titles were expedients: culturally acceptable synonyms that avoided the term systems engineering in non-defence/aerospace business situations, e.g. product management, requirements management, system architecting, life cycle management, or variations on supply chain management and risk management – the prestige of 'management' being evident in the 'branding'

Many systems engineering practitioners' interpretations have been incompatible and unconvincing to those outside the discipline. Individuals have vied to assert personal and distinctive views. In this they have benefited from inconsistent models and an attendant lack of uniform terminology and definitions. Seductive terminology has both flourished and confused.

Too frequently 'the systems approach has quickly attracted overly zealous proponents and, as often, misinformed detractors' [Ramo, 1989, p2]. Silver bullet solutions have been sold and over-expectation was the result. 'Some hail it as magic, a new all-powerful tool that can demolish any tough problem, engineering or human' [ibid, p3]. There are, alas, no silver bullets to "kill the werewolves" of system complexity.

Some practitioners see the evident, though elusive-to-prove benefits of systems engineering as a matter of faith and the discipline has seen bouts of evangelical presentation. It has variously been extolled by aficionados and technical enthusiasts, but only infrequently by business leaders. The promises of its champions have generally outpaced realistic delivery of benefits and secure answers to real-world complexity.

Presented as a solution to crises being faced, systems engineering has too often been mobilised in critical situations as a last resort – a rescue mission mobilised retrospectively – without appropriate regard to the attendant timescales and effort necessary to introduce its skills, methods, discipline and culture changes.

Early standards in particular produced stereotyped and ritualistic sequence, hidebound by a top-down, sequential, closed-system avoidance of real world disturbances, with the result that

the long time scales encountered in major programmes and changes in operational reality led to suboptimal deployments – sometimes obsolete deployments. “Late-to-need”, “unresponsive” and “too expensive” have been common descriptors of its results.

‘Hard core’ intellectual disciplines see systems engineering as lacking an intellectual basis: as offering only a purely commonsense approach to necessary practicalities. ‘Ultimately, systems engineering relies on consideration of controlling qualitative factors and on judgments, intuitions and experiences that are not quantifiable’ [Ramo, 1998].

The assumption that, because systems engineering is a recipe for success in defence and aerospace, it would naturally apply in other fields without significant adaptation proved to be gross optimism. Ill-judged over-optimism did systems engineering no favours. Particularly where humans form substantial parts of the resulting solution, the inanimate system-oriented approach was misleading. As described in §3, systems reasoning translates to other application domains, but applying en bloc systems engineering methodologies successful in one application domain to others, without adaptation to account for context and implementation constraints, borders on the naïve.

Over its existence there has been a tendency for systems engineering to grasp beyond its sensible reach. It has much yet to resolve on its home turf. For some practitioners, there is an almost imperious attitude that systems engineering has a proprietorial understanding of any systems; universally meeting challenges from the socio-economic to detailed technology. This misreads that it is its foundation of systems reasoning that is its gateway to enter other fields of study. For the most part, other fields already have implicit and even explicit, practical interpretations of systems reasoning. Whilst some of the discipline associated with engineering might not go amiss, many of its precepts are too foreign and ill-translated to be of sound practical value. Systems engineering’s borders with human-intensive social systems, where Ramo and successors encountered disappointment, is a case in point. This is an issue addressed at the close of this report in §10.5.3.

Given such a chequered history, many organisations have been slow to acknowledge systems engineering importance in their businesses and application domains. They have been reticent to promote it as a distinct discipline and, as a consequence, to foster its practice in a disciplined and professional way. The smaller the undertaking, the less relevant it has appeared. Yet theory and mounting evidence show that, whether explicit or implicit, what is known as systems engineering is unavoidably conducted in some form or other. If explicitly conducted, then its recognition is much more likely to lead to effectiveness and successes; if implicitly conducted, then more errors, rework and possible failure may result.

The message is, that by explicitly conveying systems engineering’s identity and recognised value, benefits accrue. These issues of identity and recognition are considered in the next sections.

2.4 Identity and Recognition

2.4.1 Scope and Definition

The identity crisis that dogged systems engineering throughout the 90s impeded its advance by practitioners and its take up outside its community of existing use. Although the question “what is it?” is hardly more clearly answered by broad ranging disciplines such as sociology or economics, they proceed satisfactorily nonetheless. Identity crisis is an indicator of a discipline’s lack of maturity, foundation and self-assurance.

Pinning down a meaning to systems engineering has been a necessary, though often sterile debate. The prevailing attention to a comparable and more recent discourse regarding ‘architecture’ has thankfully deflected the long-standing preoccupation with systems engineering definitions.

Systems engineering’s name has been a particular barrier to wider acceptance. “What’s in a name? That which we call a rose; By any other word would smell as sweet” [Shakespeare, 1597] is a stanza that well conveys the difficulties that stem from the ‘engineering’ family name, and its acceptability and cachet in its potential constituency of use. Its associated

connotations with wholly technological detail and its weak portrayal of management contribution mean that systems engineering's name has been more of a hindrance to establishing the discipline than a help.

In 1973 Ramo is quoted by Rechten as saying that systems engineering is a branch of engineering that "concentrates on the design and applications of the whole as distinct from the parts ... looking at a problem in its entirety, taking into account all the facets and all the variables and relating the social to the technological" [Rechten, 1991]. However, Ramo's late-in-life views are indicative of a persistent definition problem and its clear resolution. In his most recent of dozens of books, it still causes him difficulty [Ramo, 1998]. In it he stumbles over terminology, employing the terms 'systems engineering' and 'systems approach' almost synonymously. He applies discipline to the latter, but uses it in the sense of 'a system of rules for behaviour, methods of practice, etc' rather than the formalisation of 'a branch of learning or instruction', which would better fit 'systems engineering'. [Collins, 1991, discipline, meanings 4 and 5].

In this text Ramo never satisfactorily defines systems engineering. The closest he gets can be inferred from his observation that 'a substantially greater number of professionals exist today who are well seasoned in interdisciplinary problems. They know how to relate the many facets of one technology to another and to relate these in turn to all the non-technological factors that characterize practical problems. They are "systems engineers" ' [Ramo, 1998, p 19].

However, he defines 'systems approach' as 'an intellectual discipline for mobilizing science and technology to attack complex, large-scale problems in an objective, logical, complete, and thoroughly professional way....[It] includes the use of sophisticated techniques for assembling and processing the necessary data, comparing alternative approaches as to their relative benefits and shortcomings, making sensible compromises, producing quantitative analyses and predictions where they are appropriate, seeking out judgments from experience of the past, and introducing creative innovations where they are indicated'. In other words, he sees it unencumbered by 'the technological methods to which the term "systems engineering" applies' [ibid, p4].

Ramo thus regards his systems approach more in the style of the 'system thinking' of Checkland [Checkland, 1999]; something applicable to 'choos[ing] proper options in the way we design our cities, transportation systems, communication networks, educational and medical facilities'. This suggests lingering doubts about the "engineering" word, and its unnecessarily restrictive and unhelpful boundary to the discipline.

Many others faced with this semantic difficulty, have searched for "any other word" and so far failed. 'System management', an attractive but unrevealing alternative, was usurped by IT practitioners. The DoD college/university carrying that name was eventually re-titled (sensibly) to communicate its mainstream 'acquisition' function. The lure of 'system architect' has cachet, but is confined to a role circumscribed well within systems engineering's scope.

In systems engineering's formal codification, definition of term has fared no better. IEEE 1220, despite using the term 70 times and defining how to manage it, then omits a definition and leaves its readership to surmise the precise nature of this prime element in its ontological map [IEEE, 1998]. EIA-632:1998 sidesteps a definition altogether and does not use the term [EIA, 1998]. Likewise, faced with the daunting problem of gaining international agreement on a definition, the ISO systems engineering standard deliberately avoided the term other than on its title page [ISO, 2002].

To compound matters IEEE 1220 and EIA 632 both emphasise systems engineering as a 'left hand side of the engineering V' practice, focussing on the development stage of a system's life cycle. It thus became associated with 'development engineers' work' rather than with the strategic application and management of engineering throughout the life cycle, with a commensurate concern for concept, development, manufacture, service sustainment and evolution.

INCOSE, the international professional body for systems engineering, is faced with resolving a federation of views from a wide range of cognisant practitioners, some possessing strong individual opinions. It has consistently failed to arrive at a single definition and presently

offers a progression of alternatives, each of which fails in the eyes of some; requiring yet one more definition to correct its deficiencies.

At the close of this investigation INCOSE's website [INCOSE, 2007a] blandly states that 'systems engineering is an interdisciplinary approach and means to enable the realization of successful systems'. The website digs into some deeper substance, stating that systems engineering 'focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, then proceeding with design synthesis and system validation while considering the complete problem... [it] integrates all the disciplines and specialty groups into a team effort forming a structured development process that proceeds from concept to production to operation. Systems engineering considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs.'

With the opportunity to draw back from this labyrinth of definitions, the OED makes a brave attempt to briefly give its own meaning to systems engineering. Given the difficulty in reaching a single definition that has so far eluded the community of practitioners, it arrives at a succinct and not incorrect '*systems engineering* the investigation of complex, man-made systems in relation to the apparatus that is or might be involved in them; so *systems engineer* [OED, 2004].

Like INCOSE, Smart Acquisition was faced with a division of views. In its AMS Glossary at the start of this research it provided two views:

'The application of scientific and engineering efforts to: (a) transform an operational need into a description of system capability together with a system configuration through the iterative process of definition, synthesis, analysis, test and evaluation (b) integrate related technical parameters and ensure compatibility of all physical, functional and programme interfaces in a manner that optimises the total system definition and design (c) integrate reliability, maintainability, survivability, safety, cost, human factors (and other such factors) into the total engineering effort to meet cost, schedule and technical capability objectives.

A structured, inter-disciplinary process which delivers and supports integrated, whole systems to meet user need through-life; or the set of activities which control the overall design, implementation and integration of a complex set of interacting components or systems in order to meet the needs of all users and other stakeholders within the constraints arising from the system's operational and development environment.' [AMS, 2000].

The Defence Industrial Strategy, DIS, quotes INCOSE to have defined systems engineering as "an engineering discipline whose responsibility is creating and executing an interdisciplinary process to ensure that the customer and stakeholder's needs are satisfied in a high quality, trustworthy, cost efficient and schedule compliant manner throughout a system's entire lifecycle" [DIS, 2005, p59]. Aligning itself with some measure of international endorsement would seem to be a sound move; however, DIS then immediately proceeds to offer its own definition. In it, it curiously relegates systems engineering from the status of discipline to that of method: 'systems engineering is the general term for the methods used to provide optimally engineered, operationally effective, complex systems. Systems engineering balances capability, risk, complexity, cost and technological choices in order to provide a solution which best meets the customer's needs'.

Each of these definitions is the result of a particular viewpoint and a particular juncture in the evolving understanding of systems engineering principles and their extending region of application. Very many more definitions exist to emphasise the nuances in systems engineering's 'stakeholder' concerns. This variety of definitions is no more than confirmation of systems engineering's complexity and growing community of use. It is also confirmation, as described in §3, that linguistic models of systems engineering will always fail to adequately model its real-world complexity.

If linguistic definitions of systems engineering are destined to difficulty, then graphical ones offer more hope (see Figure 3.8). §2.2 points to a high-abstraction image of systems

engineering conveying its three primary discipline constituents: systems reasoning, engineering and management.

Based on their concerns, different observers centre their focus of holistic approach at different points in the trine shown in Figure 2.2 according to their:

- problem domain;
- span and level of responsibility;
- point in the life cycle;
- particular partiality or constraint of method;
- adopted tools or teaming influences.

These factors confer particular foci, spreads of engagement and often distinguishing titles to their concerns, with each conveying a particular emphasis on the management of complexity in systems. Systems engineering is depicted as being a neutral centre of this model, but with many creeds of practice bound up with individual, team and organisation concerns, there is no definitive centre or coverage.

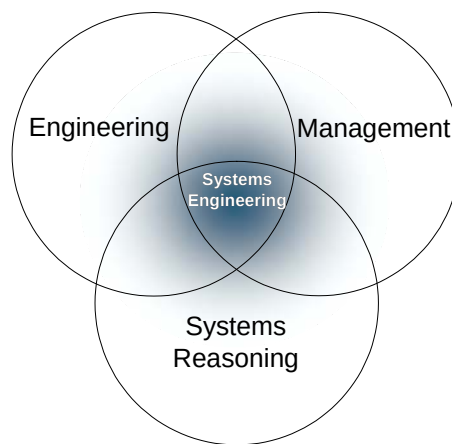


Figure 2.2 The Emergence of Systems Engineering from its Three Primary Discipline Constituents

Ramo's 'systems approach' tends towards in the 6 o'clock direction of the centre point; Checkland's is in the 4 o'clock direction; many technologically inclined practitioners in industry are biased towards 11 o'clock; chief engineers towards 1 o'clock. In competitive industrial situations differentiation and positioning are significant factors, and Figure 2.2 offers a wealth of opportunity regarding methodologies and the portrayal of systems engineering influence.

Striking a balance between the technical dimension of engineering and the social dimension of management is crucial. Too far in either direction and the emergence of systems engineering's unique properties can be dominated by the characteristics and culture of existing discipline. This balance was re-struck in the ISO systems engineering model according to its requirement "to provide a basis for international trade in system products and services". It influences the positioning of systems engineering towards the centre of Figure 2.2 in the minds of many practitioners.

Given this basis of focus and scope, the dynamics of why a discipline such as systems engineering is successful or not, and how it advances, is now analysed.

2.4.2 Practice, Discipline or Profession

This section examines the criteria that define any discipline, and considers how well systems engineering measures against them and how far it may still progress. If systems engineering is to be a secure and cardinal basis for MOD acquisition revitalisation, or be adopted confidently by a wider business community, then it must qualify against criteria of coherence, comprehension, value and positioning.

Discipline titles and shapes this chapter, and is now defined. In its common usage a discipline is:

- a branch of learning or instruction,
- a system of rules for behaviour, methods of practice, etc,
- systematic training in obedience to regulations and authority,
- the state of improved behaviour, etc., resulting from such training or conditions;

[Collins, 1991, discipline: meanings 5, 6] and it is these features that describe the term's use throughout this report.

By title this investigation addresses systems engineering both in its principles and its practices. In it, 'principles' refers to the developed body of systems engineering theory, abstract knowledge and instruction which have been contributed to by leading theoreticians (many also being practitioners), industrial pathfinders, academic bodies and research institutions. 'Practices' refers to a range of application approaches that employs this body of knowledge so as to diagnose, advise, intervene, treat and control system creation and utilisation within enterprise or social settings. Practice also invokes the connotation of a consistent quality of service delivery.

A practice is established through a set of received beliefs [Kline, 1995, p. 4], whereas a discipline requires at least some implicit body of theoretical and methodological principles intertwined with practice. Discipline therefore combines a segment of formal knowledge with the consequence of its application to the affairs of others. Defined in this manner, the compatible pairing of systems engineering principles and practice equates to the "discipline of systems engineering".

Kline's *The Conceptual Foundations for Multi-Disciplinary Thinking* [ibid, p3] emphasises four points that characterise a discipline. They may be summarised as being:

- a specific area of study,
- a literature that embodies a framework of ideas that underpin the discipline,
- an agreed methodology, and
- a working community of rewarded, and therefore valued scholars and/or practitioners.

This implies an intellectual convergence towards some communally accepted underlying principles that are articulated and distributed (defined, taught and trained), and a common mastery of these by practitioners who can demonstrate their attained proficiency through assessment against academically and/or empirically established criteria (criteria of service delivery).

The model of discipline by this text is that of a set of principles and practices conducted according to a paradigm. The term paradigm is employed in its strict philosophy of science sense as 'a very general conception of the nature of scientific endeavour within which a given enquiry is undertaken' [Collins, 1991, paradigm, meaning 4]. This accords with Kuhn's usage in his text on *The Structure of Scientific Revolutions*, where it taken to be a body of scientific achievement (laws, theory, application and instrumentation) that models problems and solutions for a community of practitioners [Kuhn, 1962]. 'Paradigm's' descent into somewhat hackneyed use is testimony to Kuhn's influence on the understanding of scientific disciplines.

In his analysis of scientific history Kuhn advances in that working within the boundaries of a scientific discipline, practitioners find themselves faced with anomalies and inconsistencies of the agreed-upon body of intellectual and practical knowledge. Accordingly, they act to extend, adapt, modify or revise this knowledge to account for these challenges, and in doing so change the nature of the discipline. As a result, science and engineering cannot simply be considered as accumulations of individual discoveries, lining neatly in a steady progression of knowledge, they evolve to new states with new, more powerful characteristics.

Despite having become a cliché, Kuhn's term "paradigm shift" completely captures this issue: radical steps in the 'nature of scientific endeavour'. These shifts require reinterpretation of

prior theory, re-evaluation of prior fact. Reconstruction of these are then applied to new situations and re-assessed in previous ones.

A scientific community, working in equilibrium with a shared set of values, suddenly finds the assumptions of the field change. A new paradigm that provides greater explanatory powers than the old emerges. The field then investigates the questions and problems presented by the new paradigm. It works in a new state of equilibrium, until yet another challenge comes along to provide another paradigm transition. These are transitions comparable to nature's evolutionary condition of 'punctuated equilibrium' [Gould, 1972].

Though its transition has taken nearly a decade, such a change has been at work in systems engineering. Arcanely referred to as 'traditional systems engineering', and adhered to by practitioners in substantially well-precedented regions of problem or technology detail, past approaches still deliver solutions. However, major changes in the nature of problems, the options of technology and international trading opportunities have unsettled past equilibrium, forcing a transition to more fluid, exploratory approaches to technical trade off and design decision making.

A paradigm 'transforms a [scientific] group into ...a discipline' and is the essential criterion that proclaims a field of scientific endeavour a discipline [Kline, 1995, p19]. A paradigm must therefore articulate simultaneously both the theoretical and the empirical content of that field.

Based on principles, paradigms act as maps that chart the direction through which technical problems may be solved. 'Adherents observe paradigms in order to become members of a community of practitioners, learning the bases of their field from the same concrete models' [ibid, p 11]. This requires communal agreement over fundamentals, and a shared set of rules and standards of action for its practice. In this way a paradigm helps a scientific community to bound its discipline, distinguishing it from others.

A paradigm must synthesise a coherent approach able to attract a substantial following of practitioners. This is likely to entail:

- specialised journals;
- specialised groups within societies, such as 'special interest groups';
- academic curricula;
- new principles, justifications of concepts, lines of enquiry and methods;
- scholarly articles 'addressed only to professional colleagues, whose knowledge of a shared paradigm can be assumed' [ibid, p 20].

'Paradigms gain their status because they are more successful than their competitors in solving a few problems that are recognize as acute' [ibid, p 23]. A paradigm offers the promise of success, creates confidence and, with time, must deliver that promise. This is effected typically by:

- adherents committing to extending the knowledge that is revealed by the paradigm;
- increasing the extent of the paradigm's predictions by matching facts with theory, and by developing ways to apply the paradigm to related areas of interest;
- further articulation of the paradigm (a topic of this investigation);
- refinement of those phenomena and theories that the paradigm supplants, and the predication of phenomena that had been entirely unsuspected while a predecessor held sway.

Even secure, deep-rooted paradigms are continually subject to transformations. This is part and parcel of the reformulation of a discipline in response to changes in the problems, opportunities and circumstances they address. 'Normally the reformulations are based on achievements of a previous order that supplies the foundation for further practice' [ibid, p 10]. Systemisation of both engineering and of management was the reformulation that paved the way for systems engineering.

Transition from one paradigm to another via revolution is the usual developmental pattern, and is a result of persistent scientific failure brought about by limitations of existing theory or

from changes in social/cultural climates. Crises are commonly a precondition for the emergence of wholly novel paradigms. The crisis of post-Cold War ICBM threat was the catalyst for systems engineering revolution.

As a new candidate paradigm emerges, a battle over its acceptance ensues. No theory ever solves all the problems with which it is confronted at a given time. Fundamentally, it is a model of knowledge, application and action, always subject to limitations on the valid range and fidelity of that model. The yardstick is whether it fits the facts better; whether it is the most viable among alternatives.

Adoption comes when a candidate is seen to resolve some outstanding and generally recognised problem or weakness that can be met in no other way, such as requirements management or architecture descriptions in the case of systems engineering, or when it predicts phenomena that had been unsuspected while the old paradigm prevailed, such as the impact of whole life cycle management thinking.

Such transitions are typically not cumulative processes. They may reconstruct a field from fundamentals, with foundational theory changed, and rules, methods and modes of application altered. When transition is completed, the view of the field, its goals and its approaches have changed. Practitioners see new and different things where they have looked before.

Much of this will resonate with systems engineering practitioners whose experience spans the last decade. Such a shift from 'traditional' systems engineering is increasingly evident.

Since new paradigms are born of old ones, they may incorporate vocabulary and apparatus that the traditional paradigm previously employed, but now in different ways. This requires practitioners to see familiar objects in a different light. The baggage of past semantics and concepts can hinder this understanding. A perceptual transformation is called for – hierarchical models for sequential models, capability for product views, business-driven decisions for science-driven decisions approach illustrate a decade's worth of the mind-set conversion in systems engineering.

A choice between alternative approaches may be called for and, rather than a single group conversion, what occurs is an increasing shift in the distribution of professional allegiances: EIA-632 or ISO/IEC 15288. In this sort of gestalt adoption, alternate perceptions may simultaneously be valid, reasonable and even necessary. Different views – new and old interpretations – having to coexist for a time. This accommodates the investment in past approaches; an important issue given the duration of many systems engineering programmes. Eventually, 'older views ... are simply read out of the profession'.

Transfer of allegiance from paradigm to paradigm is largely a proselytisation that cannot be forced. Proponents and adherents of a paradigm devote lives and careers to a paradigm. Lifelong resistance comes from deep-felt assurance that the older paradigm will ultimately solve all its problems; the same assurance that first reinforced it. In this way, the progressive ascendancy of the ISO model-based paradigm of systems engineering has not been the result of evangelism but of exposure, localised discovery and cumulative adoption.

Some resist a new paradigm indefinitely. Planck's observation, that "a new scientific truth ... [flourishes because] a new generation grows up that is familiar with it", suggests why the value of distinguishing between service and product requirements, of applying common transformations recursively, of separating enabling systems from systems-of-interest, all take time to be accepted. It indicates the receptiveness of a "new generation" towards present-day systems engineering; a new business generation not only of practitioners but also of leaders of project teams. The current generation of IPT Leaders has been conditioned by Smart Acquisition's policy and benefits, and is increasingly committed to systems engineering as an integral component of their business world. The evidence of this is presented in §8.

Beyond disciplines and their defining paradigms come professions.

Ramo is forthright about the status of systems engineering. He considers it to be the 'use of techniques which must be called professional because it can only be done well by people who prepare themselves well and build up experiences in the approach ... professionals exist today who ... are [called] "systems engineers"' [Ramo, 1998, p.19]. Whether his use of professional is emblematic or a judgment on some social scale is unclear from his text.

In this text 'profession' is seen to signify a particular regard for quality and recognition of status, accorded both to a discipline and to its individual practitioners. Professions are identified with accreditation of a discipline's practitioners, usually by an award of licentiate status directly or indirectly by legislative authority, or certification by a professional association recognised by society, or sometimes the accolade of a well-regarded occupational grouping or business federation. Profession is the highest status achieved by a discipline and its practitioners. It conventionally attracts an appellation and a warrant to act in society in a defined region of service provision under the auspices of some counselling authority.

Professionalism, as a recognised status, enshrines attributes that support a mutually beneficial relationship between the individual practitioner and society:

- for the individual professional, professional status provides practice development and tutelage, occupational shelter (security, safeguard against error), recognised framework of employment, reserved career structure, social standing, and autonomy and discretion of action within the confines of the profession;
- for society it provides declared standards of service level, the identification and endorsement of service providers accredited to deliver this, regulation of service quality and redress in the event of failings in this quality.

To earn recognition as a profession it is suggested that a branch of endeavour needs to be measured against key criteria:

- a generally accepted body of learning and knowledge;
- a mechanism for advancing that knowledge;
- a formalised education and tutelage structure that imparts accumulated understanding and practice;
- a community of regulated practitioners that provide society with a service distinguished by value and integrity.

This requires a discipline to be regulated according to an intellectual tradition and an established body of knowledge. This has not yet been achieved in systems engineering. Recognised shortcomings, instability and limited endorsement seen in successive publications of the INCOSE Systems Engineering Handbook show there to be some way to go. Alignment of this in 2006 with the ISO systems engineering model have greatly strengthened the claim, and this will be much sounder if and when this handbook is placed in the public domain for its ratification. A proposed move to be an ISO Technical Report should satisfy many concerns.

Technical skills need to be acquired not just through approved academic routes but through a concerted and mutually recognised scheme of professional training across organisations. This requires a framework that covers different organisations and application sectors. Systems engineering may appear a reusable skill but it needs to be promoted as such, and not merely absorbed in application specific situations. The lack of cross border agreement – organisation to organisation, industry to industry, and state to state – means that this has some way away.

Any profession needs a bought-into set of professional values. These are not simply the ethics of behaviour but the demonstrable assurance of individual practitioners' services within a cultural framework. International agreement here is problematic and it is likely that, in the foreseeable future, strong and rather diverse national constraints will hold sway. However, the multinational dimension of systems engineering is likely to work in its favour and it is the growing dependence of a trading dimension in systems engineering that will force the issue. Consultation will play its part, but business and economics will hold sway.

Most profoundly, though, the stride from discipline to profession in any field is attributed to the latter's exclusive jurisdiction over a region of discipline [Abbott, 1988 p 86], [Freidson, 1994]. The true power of professions lies in their de jure or de facto prerogative that sets the boundaries of what their occupation's work embraces. In order to claim such jurisdiction of a discipline, a profession must ask "society to recognize its cognitive structure through exclusive rights" [Abbott, 1988, p 59].

Professions have assumed a range of formalisations, which by and large can be traced to four motivating factors [*ibid*]. These factors having a subjective foundation result either from social synergies or from monopolistic interests; most traditional professions arose from one or both of these motivations. Conversely an objective foundation for a professions stems from its functional role in a multi-functional whole, or from a structural distinction that arose through diversification (theoretical, methodological or/and applicability) in a previously unified discipline. These four motivators are not mutually exclusive and elements of each can be seen in systems engineering.

Social synergies trace to the tradition of accumulated, slow-changing cultural structures, e.g. professional or ceremonial bodies, or latterly to change which is a reaction to critical shifts in social values or wellbeing that usher in communal acceptance without formal ratification, e.g. technology revolutions and threats. They involve accepted standards of action and best practice that society cede to a professional community on the basis of their special expertise, objectivity and moral probity, e.g. clergy, law, medicine, marine and air safety, electrical safety and, perhaps to come, information security.

Many fields go through developmental stages to attain dominance over a region of discipline jurisdiction and monopoly is a compelling strategy to achieve this. 'One profession's jurisdiction pre-empts another's'. Those coming later find territory already occupied and well defended. 'Highly organized internal structures are more resilient to attacks by less organized professions'. Failure to treat existing weakness is the main attacking point for invading professions [*ibid*, p 49]. In changing situations, and with weakly defined and unclear areas of tasks, professions struggle and compete against each other to gain control and expand their jurisdictional strength [*ibid*, p90]. Systems engineering resides in such territorial uncertainty.

Change can result from the development of new knowledge and corresponding expansion of jurisdiction. Some problems live in constantly shifting classifications and may lead to intervention or competition by other professions who want to assimilate the unclear problem into their own professional repertoire (*ibid*, p 44), e.g., information technology and project management in the case of systems engineering.

Monopoly has a dark side that seeks a privileged place in the labour market. An outcome can be the organisation of esoteric training, tutelage and credential structure. This may be self defeating in changing times, as adherents take refuge in the increasingly detached and irrelevant. Arguably, system analysts and operational researchers suffered this fate [Ackoff, 1979] and in doing so inadvertently cleared some of the field for systems engineering.

The functional claim to jurisdiction relies on mastery of the knowledge and skills of a discipline. Professions provide expert services that address the problems of individuals, groups and society at large, and Abbott advances that the history of professions maps to the history of personal and social problems [*ibid*, 1988, p33, p285]. Latterly, two particular circumstances have helped the advancement of functionally-based jurisdiction: the rise of skill sets in large-scale organisations and the origination of technological imperatives [*ibid*, 1988, p144], both of which favour the advancement of systems engineering towards being a profession.

Abbott specifically identifies management, engineers, operations researchers, and systems analysts as exemplifying the functional jurisdiction situation. 'Members [of such professions] conceive of themselves by their occupation first and by their "class", if at all, only second' [Freidson, 1994, p17] – what Elliott refers to as an "occupational profession" as opposed to a "status profession" [Elliott 1972, pp.14, 32]. Special expertise and intellectual probity drive commitment to professionalism. For redress in the event of professional failing, practitioners must submit to the authority of a professional body; a sanction that systems engineering is long way off.

Abbott points out that the structure of a discipline or established profession is 'constantly subdividing under the various pressures of market demands, specialization, and inter-professional competition' [Abbott 1988, p.84]. New factions make a play for independent recognition and may in time even subsume their origins, e.g., systems engineering and systems analysis. Thus the structural factor is ever present in the establishment of new claims, with academic leaders, consulting 'gurus' and enterprise competition constantly driving to position themselves and their platform to advantage.

The influence of these four profession motivators has come into play at different points in the maturation of systems engineering. It evidently began as a practice-driven discipline in response to political and military pressures. First and foremost, the functional dimension has been the driver of systems engineering.

From this, and in the Cold War tensions that gripped society, it was quickly understood that systems engineering possessed a strong social synergy through its delivery of new ways of working in the novel, complex government/industrial relationship that arose in strategic defence acquisition.

From the resulting growth of systems understanding, experience and techniques, alternative approaches and application opportunities developed. These have presented, and still do present, threats and opportunities that are not yet being managed in systems engineering according to the precepts of an aware and coherent profession. Enterprise architecting, soft systems approaches, flavours of quality management methodology, and service and capability regimes are all acting to commandeer ground that was previously undisputed or has become under-recognised as systems engineering 'territory'.

More recently, a rush towards establishing monopolistic interests, motivated both by raised standards of discipline execution and by the need to proclaim systems engineering's standing and jurisdiction, have characterised on-going actions. The push to establish certification or accreditation to existing, recognise, licensed or de facto jurisdiction bodies, e.g. ABET, the UK Engineering Council, is evidence of this. If and when carried through they may raise systems engineering to the level of a profession, but they face an uphill struggle in a context of fragmented national/state authorities on such matters.

A jurisdictional claim by a profession can be achieved in different ways: assent by state, sanction by the legal system, common consent in the realm of public opinion, or in the workplace the power and natural authority of the trading structure. "Each profession is bound to a set of tasks by ties of jurisdiction". This is the ultimate step that transforms discipline into professional status [ibid, p33].

The US DoD's wavering valuation of systems engineering and the UK MOD's diffidence towards it have failed to provide a consistent or wholehearted proclamation of its value and standing within its heartland of defence. NASA and ESA have been more steadfast, and can point to its benefits and the dangers of disregarding it; for many, though, they are in a world away from the general run of competitive enterprise. Boeing has been forthright but not proselytising; Airbus has dressed matters up using in-house methodological euphemisms. Generally, the wider world has not seen unreserved commitment to, or persistent examples of, systems engineering such that society should see it as the unchallengeable route to resolving regions of prevailing technical and managerial challenge.

Step by step these criteria of professional status are being met for systems engineering. For the present, it must be accepted that systems engineering is a distinctly subordinate discipline of the engineering profession, albeit one that contributes to and influences probably all other engineering disciplines or specialisms.

Codified principles and practices publicly assert a discipline's territory, provide stable models for cooperative action and offer criteria that when observed lead to mutual confidence in ventures. Though the discipline was slow to respond, ISO/IEC 15288 has provided a professional "standard to rally around". Despite the International Standards ascendancy over its US national predecessors, lingering internecine discord and divisive software engineering influences have impeded the establishment of a clear-cut and stable international systems engineering instrument of jurisdiction. INCOSE certification, now based on ISO/IEC 15288, may partially and belatedly recover this weakened opportunity for advancing the jurisdiction of systems engineering, and for subsequent attainment of the status of profession.

It is thus proffered that ISO codification offers identity and jurisdiction – continuing weaknesses of systems engineering. §4 therefore re-assesses its bases in some detail. Next comes the obligation to deliver value.

2.4.3 Delivering Value

A recurrent theme in the analysis of professions is explored by Freidson with his association of a profession's work, success and exercise of influence [Freidson, 1986]. He stresses that for professions 'it is necessary to understand how knowledge gets translated into action, which means understanding the human institutions that mediate between knowledge and power'. In other words, 'pure' principles are fine, valid 'applied' practice is what matters. Does systems engineering follow such a maxim?

A misquote – ask not what your nation can do for systems engineering, ask what systems engineering can do for your nation, [Kennedy, 1961] – could be used as a mild rebuke against past practitioner introspection and lamentations over systems engineering. Systems engineering's brooding needs to be consigned to the past; there is no place for indulging in self-analysis and self-justification. It must concentrate on delivering value and be recognised for it. If it does not, others will certainly seize the ground

Increasingly systems engineering is looking outward at its wealth creation contribution to society and how it constructively complements, and builds on, a wide range of technical disciplines. Although its intermeshing with the other engineering disciplines and with project management has been a source of past tension, if not turf war, this relationship is becoming a more comfortable matter of fact – one reason for a growing interest in systems engineering.

One of the earliest, most often quoted, though still questioned value assessments related to systems engineering was performed by Gruhl at NASA [Gruhl, 1992]. It compared upfront expenditures (largely recognisable as present-day systems engineering action) with eventual cost growth in programmes. Management data showed an inverse relationship between programme definition costs and development cost overruns. The validity of the demonstrated correlation has been queried [Schaffer, 2003] and the whole question of such metrics-based assessments had already been called into question by Sheard, who argued that there will be no hard return-on-investment numbers in the foreseeable future and that, if there were such numbers, applying them would be so situation specific that the results would not be credible [Sheard, 2000]. Honour points out the difficulty: 'management choices are made that set the parameters for the stochastic [systems engineering] process ... many factors influence the actual outcome ... any given project provides a single sample of the values ... from interrelated stochastic processes'. This highly varied population may thus defeat rational analysis.

All this leads INCOSE to conclude that 'a rigorous business case for systems engineering has yet to be made but is under development' [INCOSE, '2020']. The task is in hand and metrics have been chosen: 'technical "size", technical complexity, technical quality, project schedule or duration, project cost, risks and systems engineering effort expended. The last is the hardest to measure. It is taken to be systems engineering quality (a somewhat subjective measure) x systems engineering cost (cross-project estimated contribution from all quarters) / project cost [Honour, 2004]. Boehm is researching a constructive cost model for systems engineering akin to the COCOMO model applied to systems engineering. It uses EIA 632 as a guide for identifying the systems engineering actions and ISO/IEC 15288 for identifying system life cycle stages [Boehm, 2003]. Both of these approaches are likely to require a long period of collection of much data to overcome the wide variations in key measures. Systems engineering is strategic and only long-timescale management data is likely to look credible.

To achieve results quickly, the 'measurement' of conjectured loss ascribed to not investing in systems engineering or not conducting it explicitly, may be followed. This is often used as an argument, but the search for instances of non-existence of a problem that would otherwise have existed is somewhat subjective and almost un-provable. An accumulation of past failure is perhaps the best indication of future failures and the need for ameliorating action, though mutual dependencies undermine this argument.

In general it is difficult to establish the value of systems engineering. This arises because of:

- the strong action-and-payback interdependency with other business perspectives; one value statement cannot be disentangle from a highly complex set of factors;

- the overlapping scope and timing of other asynchronous and independently motivated improvement actions, meaning that assessment in isolation is near impossible; spurious correlations get in the way or multiple related change programmes act as uncoordinated external agents;
- the inability to define and objectively assess against meaningful metrics; recognised business indicators are not designed to isolate systems engineering's impact.

Time, rationale and rising complexity are the surest orators for systems engineering's value; metrics are unlikely to persuade. It is the business conviction of leaders that will be the most powerful influence for the foreseeable future – a hypothesis explored in §8.

The business focus of the organisation is the profitable provision of products and services that satisfy customers. This is the business excellence goal portrayed in Figure 2.3 and it is the value that systems engineering contributes that will ultimately determine the discipline's future. Its foundation is a well-integrated set of business processes that govern the technical issues and the (labour, material, equipment and financial) asset utilisation of the organisation.

The organisation constructs an operating environment – a systems engineering capability – that enables its business processes to be conducted. This may need to work in harmony with other environments of trading partners, sharing approaches, resources and suppliers to ensure that at the trading interfaces consistency in delivered products or services is maximised. In the case of government trading with industry, this is a headline message [DIS, 2006].

The most powerful of the organisation resources are the competent professionals who cause the processes to be conducted and that manage their overall synchronisation and direction. These stand apart in the business model in Figure 2.3 and are generally the subject of distinct organisational interest. The battery of individual competences, that constitute an organisation proficiency profile tuned to the execution and control of business process transformations, is correctly stated as the organisation's 'most valuable asset'. Individual competence cannot be fully benefited from unless set in a team whose members' complementary competences build a well-designed profile. Team competence has to be managed and orchestrated to form a holistic synergy equal to the project team goals.

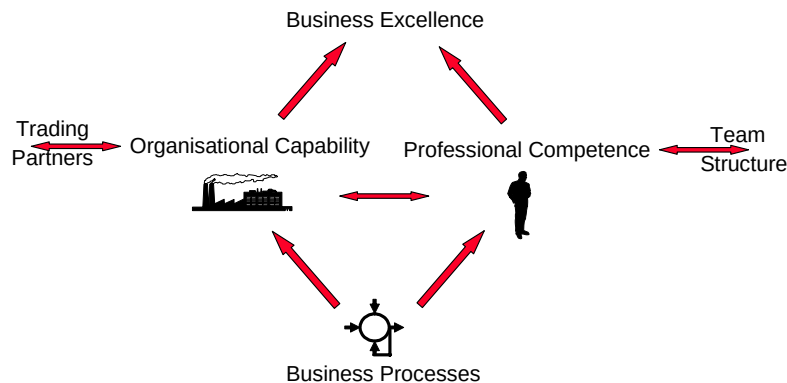


Figure 2.3 Process: A Basis for Business Excellence

Capability and competence do not stand in isolation; they are interdependent, with each other and with others. The development of competence relates to the organisational infrastructure available to develop skill and experience. Conversely, excellent facilities and tools are of little value without the competences tuned to complement them. Trading partners are most effective when matched by complementary capability; individual professionals when integrated with complementary competences.

This simple but profound model is the motivation for the triptych of views of systems engineering presented in §4 and §6.

2.5 Formalising the Systems Engineering Discipline

2.5.1 Systems in an Engineering Setting

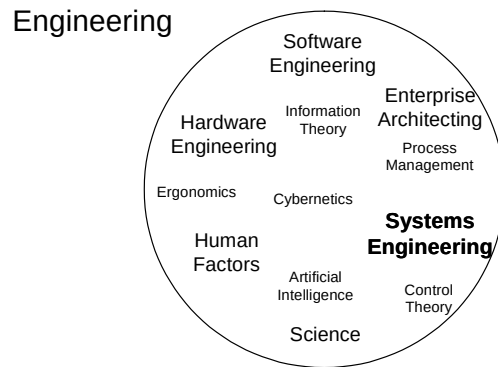


Figure 2.4 The Engineering Context of Systems Engineering

The systems engineering ‘family name’ may present drawbacks, but engineering is unquestionably systems engineering’s strongest jurisdictional territory. It shares its engineering heritage with a diversity of other engineering disciplines that contribute to the integration of ‘implemented’ elements according to holistic design. It is a member of an engineering community, yet it must act as unifier, orchestrator and arbiter across the rest.

This role regulates the level of engineering abstraction and the granularity of structural detail that it must manage relative to other contributing engineering disciplines. Its focus of concern is necessarily always “one level up” from that employed by the implementation technology-specific engineering family members it acts to unify, irrespective of the structural granularity being considered.

Examples of other members of the engineering family are shown in Figure 2.4. Software, hardware and (the now quite commonly termed) human factors engineering are the three examples of implementation technology presented by ISO/IEC 15288. They emphasise the strongly contrasting implementation media features that have characterised the formulation and descriptions of their individual principles and practices, the mind sets, language, methods and tools of their respective practitioners. They illustrate the distinctly separate ways in which engineering disciplines have described and conducted engineering in their individual bailiwicks. Though traceable to common principles and conventions in systems engineering, they have formed into stovepipes of relative isolation, having some understandable division and some incidental diversity.

Amongst major engineering players illustrated in Figure 2.4 science is included as a catch-all to represent contributions from basic sciences. More specialised regions of practice such as ergonomics, artificial intelligence, control engineering and information theory serve to exemplify the wealth of supporting detail that contributes to the engineering family.

The individual distinctions across these engineering disciplines are important and necessary. The nature of their implementation media is radically different. They rely on the conduct of radically different material, energy and information transformations, and employ radically different enabling environments. They are also based on bodies of knowledge, skills and experience that can be incomprehensibly different, one discipline to another.

This diversity is essential, for it is from this heterogeneity that novel, holistic homogeneity comes. Technological disparity is the essence of new system properties; it is also the principal source of engineering challenge.

Building a united view, the selection and arbitration between competing and cooperating candidate implementation technologies, and the technical leadership to achieve common purpose is what distinguishes systems engineering from the other engineering disciplines. It is an engineering discipline meta-model.

It is seen by some as no different to general or generic engineering, but this is to deny its profound orientation towards systems reasoning. Perhaps, if the significance of systems in science and engineering thinking has occurred in the industrial revolution, then engineering's scope and etymology would have incorporated it from the outset. Systems engineering would never have existed as a discipline or term; there would simply be a meta-level engineering concept that included all we now know as systems engineering. The dichotomy of it being a part of holistic engineering, yet a part of each implementation-oriented specialism would have been accommodated.

Thus, like many things in systems engineering, one's point of view is important. From each discipline's point of view, systems engineering deals in a somewhat abstract view of concepts that are expressed in a sufficiently pragmatic and practical manner to be directly relevant to one or more implementation media. From the systems engineering point of view, common concepts are diversely and distinctly expressed in the implementation specialism of individual engineering disciplines.

Systems engineering is thus a cross-cutting common component of all engineering. It dissipates into the stovepipes of engineering specialism as conveyed in Figure 2.5. The holistic abstraction progressively absorbs the specific implementation technology features as emergent properties become the focus of engineering attention. Systems engineering may in this way be seen as not just an integrator of products but also an integrator of engineering disciplines.

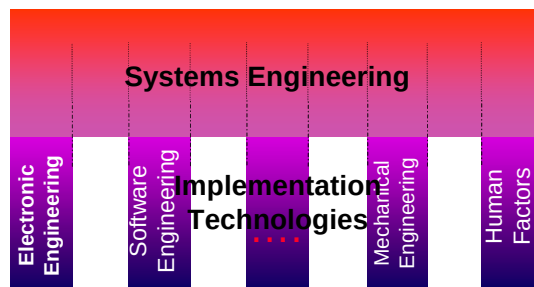


Figure 2.5 The Relationship of Systems Engineering to Branches of Engineering based on Specific Implementation Media

Figure 2.6 conveys this in a different manner. System product/service consideration equates to 100% systems engineering responsibility, whereas fabrication instructions reside at the 100% implementation technology consideration level. The climb is from total technology to total engineering, rising in columns of technological independence that progressively are assimilated into one another as the architecture binds different elements created from different branches of engineering into greater and greater unity.

The holistic system properties that emerge can also be considered according to distinct classes of property, integrating diverse phenomena that contribute to their emergence. At the holistic level of consideration, particular system oriented specialisms are typically drawn into considering the safety, security, reliability, maintainability, environmental impact and many other overall system aspects. The result is a complex engineering interplay between a 'horizontal' united system view at the top of Figure 2.5 and 'vertical' individual technology views at its base. Echelons of systems engineering thinking metamorphose into columns of implementation technology-driven engineering thinking.

This emergence of properties is also seen as an emergence of meaningful, application-oriented components. The build up from technology is therefore also a fusing into components or parts that exhibit practical functions with individual and recognisable purpose that may be incorporated in different systems. The hierarchy of granularity that ascends from natural

technology structure, through assemblies that create artificial technological properties, to composite technological products, and on through enterprise complexities that provide complete capabilities (and ultimately the systems of social order) is therefore a ranking of functional autonomy, structural intricacy and growing complexity. A hierarchy of ascending system complexity conveyed in Figure 3.2 spans this scale from implementation medium to social order.

As this aggregation proceeds, it does so according to ever greater product/service identity, and this is conveyed lower right in Figure 2.6. First the technology, such as metallurgy, is assembled to independent function units, such as fuel systems; which are then integrated to independent functional units, such as engines; which are built into autonomous purposeful systems, such as aircraft; that then are employed in holistic capabilities, such as an air transport system; as illustrated in Figure 3.2. At each level of aggregation, originality and innovation in system functional and physical performance emerges.

This progressive merging and transformation from the implementation technology/application engineering disciplines into the engineering of application domain capabilities is the 'bottom-up' view of disciplines summarised lower right in Figure 2.6.

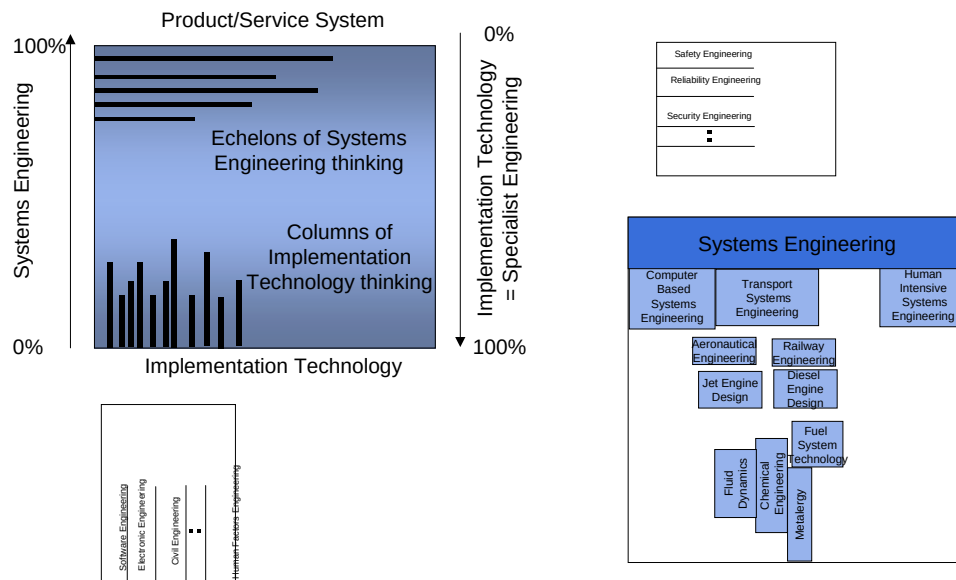


Figure 2.6 Merging Influences of Systems Engineering and the Implementation Technology/Application Engineering Disciplines

How the organisation's resources are planned, achievement is assessed and direction is controlled in order to achieve this raw material to changed-order-of-actuality transformation are considered next in the management contribution to systems engineering.

2.5.2 Systems in a Management Setting

The role of management in organisations has traditionally been to plan for the employment of assets to meet agreed goals, to monitor achievement against these goals and to direct the diverse contributors that develop products, and apply and sustain services. Since the rise of multi-disciplinary teams, especially in order to cope with the diversity of technological specialisms that contribute to highly complex systems, these management functions have been assimilated into a distinguishable and discrete set of practices termed project management.

Sitting at a responsibility level above this is the corporate enterprise management, conducted in accordance with some chosen sphere of trade and commerce. It is tasked with directing resources and assets in a competitive environment to meet the needs of specific sectors of

society: the customers/stakeholders who invest in systems-of-interest and the shareholders who invest in the creator/user organisations.

Figure 2.7 shows the management setting in which systems engineering is situated. It complements the engineering context shown in Figure 2.4, with systems engineering an integral part of each. Its 60° axis defines the emphasis of project management. Off this axis, enterprise management addresses policy, business direction, resource allocation and the investment in assets in one direction (150°). In the other, enterprise architecture is concerned with the technicalities of specifying and implementing the functional coherence and provision of information technology-related aspects of enterprise and project operation.

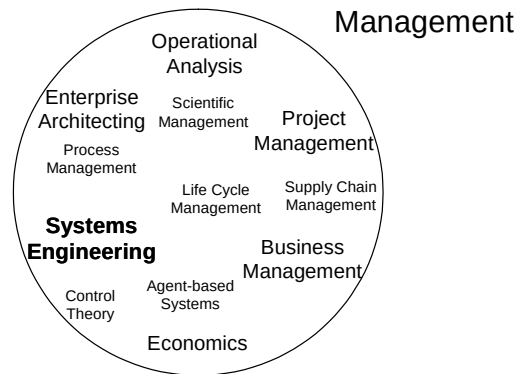


Figure 2.7 The Management Context of Systems Engineering

Just as with systems engineering, the discipline of project management evolved from different organisational management practices. In the 1930's, Schell took the view that 'the work of the engineers in most departments is not sufficiently routinized to allow process control. The most satisfactory policy appears to be that of employing competent men and then holding them [responsible] for results ... leaving ways and means largely to their individual discretion' [Yates, 2000].

This sentiment may trace to Taylor's more sociologically-based advances in management early in the C20th. However, Shewhart made a major engineering-related breakthrough when he published his text on the control of industrial process based on statistical variations, and this presented the 'plan, do, check, act' Shewhart Cycle [Shewhart, 1931], elucidated and promoted by Demming with huge effect from 1950 onwards. This had profound influence on management thinking in general, and quality levels in particular. It was built on the negative feedback control principle of establishing a value criteria, acting to achieve it, assessing the deviation from the criteria and then introducing corrective control action. This had been emphasised by Rosenblueth in his 1943 paper on *Behavior, Purpose and Teleology*, and was a basis for Weiner's work on cybernetics. But it was Gantt and others in the Post War period that evolved management practice, and project management specifically, into a distinct business function that merited study and discipline.

Gantt introduced rigorous planning and control techniques using the now familiar graphical and logical models developed from his detailed study of the order of work operations in US Navy shipyard management. In so doing he built a highly practical engineering-related outcome from what had been a sociology-oriented approach followed in Taylor's earlier 'scientific management' [§ 3.4.1]. The complexities of projects and the shrinking war-time labour supply demanded new organisational structures, and systemisation of an increasingly scarce resource led to essential advances in efficiency and effectiveness.

Prior to the 50s projects had still been managed on an ad hoc basis, and it was the same circumstances that catalysed systems engineering that also led to present-day project management. Project management really took off in the early 1960s as businesses fully realised the benefit of organising work around projects, and understood the critical need to communicate and integrate planning of resources, assess achievement and apply corrective control actions across multiple organisational units and disciplines.

Methodologies such as Critical Path Method for managing plant maintenance and Program Evaluation and Review Technique, PERT, from the Polaris missile submarine program highlighted the importance of structured, highly disciplined planning and control methods. They fitted well with the structure and discipline being introduced into engineering. These mathematically inclined management techniques quickly spread into many industrial organisations and ultimately into government control of defence programmes.

Kerzner's 2003 assessment of project management is that, whereas 'thirty years ago project management was confined to DoD ...today [it is] applied in diverse industries and organizations'. It is now 'highly organic and can respond very rapidly as situations develop inside and outside the company' [Kerzner, 2003]

The evolutionary path taken by project management is a valuable indicator for systems engineering advancement. In 1969, the Project Management Institute (PMI) was formed in the USA. In 1981 *A Guide to the Project Management Body of Knowledge* (the PMBOK Guide) was published containing the standards and guidelines of practice (largely process oriented) that are widely used throughout the profession. In the wake of this industry and commerce learned (often the hard way) that the good structure and discipline of project management invariably led to business success. This recognition was translated into competence development, supporting infrastructure of capabilities (methods and tools) and reward structures that encouraged professionalism, personal development and a culture shift. Most recently, and with the recognition by businesses of the PMI's system of accreditation for project management, the expectation that project management should be conducted by accredited staff has taken hold, reinforcing the value and status of the discipline. This is a pattern that, though still in its earlier stages, is being followed in systems engineering – similar motivations, similar goals. Systems engineering has been trailing project management by around 15 years and could recover that lost ground by learning from its advancement as a discipline.

The explanation for the lag behind project management is that project management's linkage with profit and business success is more evident and judged by shorter, more easily measured criteria. Project management could break the bounds of defence and aerospace easily, and be applied to non-technical undertakings, thereby enjoying a much wider level of use. Also, on average, there are far more 'project managers' around than systems engineering practitioners; the highly skilled systems engineering resource is but one of many that need to be managed according to disciplined project team conventions.

Management profoundly influences engineering conduct as today's complex systems grow in their dependency on widely diverse skills and assets that are distributed across multiple, international organisations needing to behave as single enterprises. These organisations see themselves as a part of a single global economy. They need to co-operate in a framework of common concepts and ways of inter-working. They need to conduct the management of complex systems with acceptable levels of commercial risk throughout their lifetimes. It is, however, a mutual influence for, as seen in §4, the ISO standard model of system technical processes confers obligations on key aspects of the tactical management at a project level, the strategic management at the enterprise level and the management of agreements with other organisations.

The relationship between the responsibility of enterprise management and project management is shown in Figure 2.8. It has similarity to the whole/part relationship of engineering in Figure 2.5. The two regions of management overlap since much of the management of enterprise is conducted according to time-boxed, target-focussed, integrated team action of projects. Almost any management may be approached according to the precepts for conducting a project.

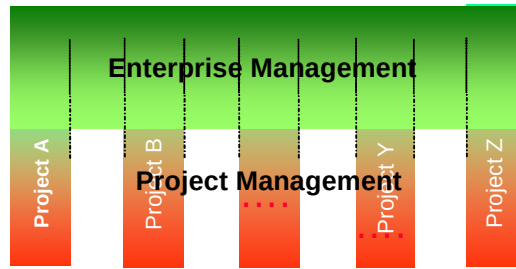


Figure 2.8 The Relationship Between the Management of the Enterprise and Management Responsibility in Individual Projects

Overarching in Figure 2.8 is the enterprise level management with its commercial responsibility to customers of and stakeholders in the products and service that are traded, and with its investment obligations to the shareholders in the organization. Here, against the prevailing socio-economic environment for the organisation:

- the policy followed for trading products and services is laid down;
- business assets required to conduct their creation and/or application are defined and established, e.g., technology base, organisational process descriptions, enabling systems, workforce skills; much of this equates to the organisation's capabilities and competences according to §6;
- project management is empowered according to agreed responsibilities;
- overall resources (budget, manpower, infrastructure assets) are deployed amongst projects in order to meet their conferred responsibilities.
- corporate monitoring and control of projects is conducted through life, i.e., the executive responsibility for system life cycle management.

Subordinate to the enterprise level in Figure 2.8 is the project level. Here management is concerned with achievement and progress against technical, schedule and cost requirements, whether product or service, and demonstrating this achievement at the major decision (stage) gates established by the enterprise, and used to monitor achievement.

2.5.3 A Context of Others

From Figure 2.4 and Figure 2.7 it may be seen that systems engineering sits in a complex mix of business practices, technical disciplines and specialised subjects. Within business organisations it is found at a conjunction of longer-established management and technical disciplines.

A relative newcomer, it has not always sat comfortably within compartmentalised technical functions of organisations, nor with the stovepipes of engineering and scientific specialism in academia. Nor has it meshed easily with traditional business and management schooling, notably project management. It is in essence an integrative discipline and thus of necessity actively spreads into the territory of others.

The processes that comprise systems engineering have always existed and have been implicitly applied, though not uniformly or repeatably, by others. Past weaknesses in establishing identity and clear definition have left its region of business process open to neighbouring disciplines to treat according to their particular interpretations. The processes that constitute systems engineering have in particular been separately described by 'adjacent' disciplines, principally project management or engineering fields such as software engineering. They have provided some influential if partisan definitions of what would otherwise be considered as systems engineering principles and practice. This proxy representation (or as Abbott might put it, disciplinary attack) has frequently led to unsatisfactorily definitions and mutually incompatible outcomes, leading to unrepeatable achievement and sometimes to failure.

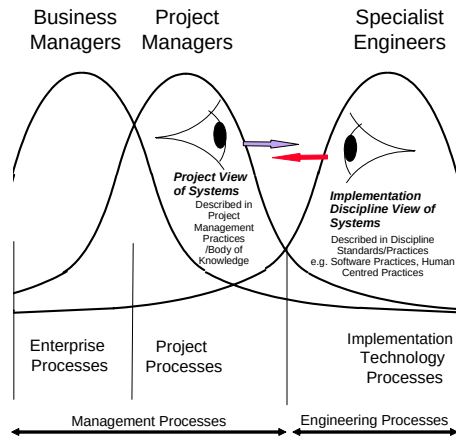


Figure 2.9 Business Process Responsibility and Influence in Organisations with No Explicit Systems Engineering

Before systems engineering established its ground effectively and securely, the picture of business processes looked somewhat like that shown in Figure 2.9. Enterprise processes served the business managers well, and project management had become sufficiently established and valued to exercise jurisdiction over a key part of this management region.

On the right side of Figure 2.9 the many technology-based engineering disciplines made their diversity of contributions to product creation. Between management and engineering lay a less coherently defined, less stable and often shifting region of systems engineering: an ill-coordinated but necessary orchestrator of technology-based engineering disciplines, and a pivotal interface between management and engineering.

Before a coherent systems engineering viewpoint of business process could be define and agreed, its neighbours sought to fill the 'gap' in Figure 2.9. This was addressed from the management direction by the project management discipline and its view of relevant systems engineering practice, expressed in its Book(s) of Knowledge [PMI, 1998], [PMBOK, 2004]. From the engineering disciplines the same region was described in software engineering terms [ISO, 1995] and to an extent also by many technical disciplines – though differently by each. The outcome seen from each concern appeared workable; the overall outcome seen in the resulting systems often was not.

The need was for the shaded region in Figure 2.10, [Arnold, 2000], to be effectively and stably defined in technical/business process terms. As shown in Figure 4.4 it has been a long haul to arrive at anything approaching a universal outcome but this largely has now been achieved. The resulting systems engineering profile defined completes a 'joined up' view of the processes employed across the organisation.

Each organisational function's view is an interpretation that applies to its particular discipline(s) and is limited by the ability to interpret and represent the complexity of the whole. The best that can be achieved is segments or distributions of interest. Each overlapping profile is faced with inconsistency, albeit containable inconsistency, with it neighbouring distribution of process concern. This is not particularly a matter of right and wrong, more a matter of contrasting viewpoints of a complex structure: a matter of different models and 'architecture descriptions' of a complex arrangement that arise from a diverse concerns and diverse organisational responsibilities.

The well-integrated organisation needs to behave as a seamless span of 'joined up' enterprise management, project management, systems engineering and technology-dependent engineering specialisms. Accordingly, its business processes should exhibit similar seamlessness in the way they assemble to form a continuum of action based on complementary capabilities and compatible competences.

Such seamlessness is not achieved by actors in these organisational functions operating in isolation. Each function has spans of influence and knowledge that overlap to produce both depth and integrated breadth, as portrayed in Figure 2.10. The unity of a well-integrated, coherent delivered system is the outcome of its ultimate enabler – the well-integrated

organisation using a coherent span of processes (managerial and technical) to perform its business.

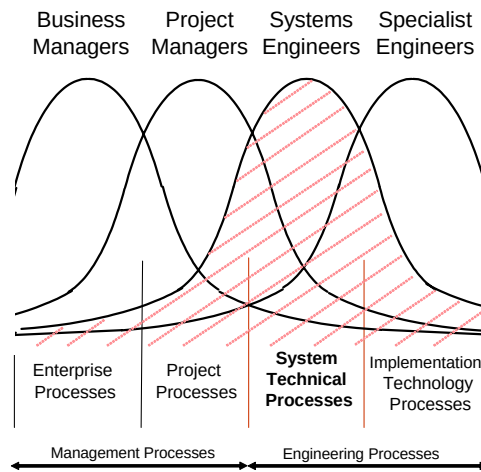


Figure 2.10 The Profile of Systems Engineering Accountability and Influence in Organisations

Systems engineering may thus be described as a profile of business process, centred on non-implementation-specific technical processes and expressed to match the context of complementary organisational processes. It is defined generically in terms of the organisation's actions during the whole life cycle of a system. Such a frame of reference can effectively guide the system contributions of diverse organisational functions at different stages of the life cycle. It is a view that sees the system technical processes as a band in a continuum of business processes within any organization.

Figure 2.10 conveys some valuable messages:-

- To some extent any function or role in an organisation spans many system life processes. For each there is a focus of responsibility that decays gradually as these idealised profiles convey.
- Everyone in the organisation benefits from some level of understanding of systems engineering. At the very least they need an awareness of it, up to and including the most senior members of management to the left of Figure 2.10.
- There is significant shared execution responsibility of life cycle processes between and across disciplines. This is not an issue of encroachment but is the natural state of affairs in a healthy organization. No functional group should operate in isolation.
- All engineering disciplines, e.g. software engineers, human factors specialists, electronics engineers, rely on a distinct level of proficiency in systems technical processes. Without this they live in their silos of technical specialism and systems engineering is their route to interacting with other branches of engineering, see Figure 2.5.
- Systems engineering's mutual dependence on, and relationship with, management is as great as it is on implementation technology-based branches of engineering. If 'joined-up' engineering and management is to be achieved, then the advancement of systems engineering as an organisational discipline is pivotal – business excellence will be very difficult, if not impossible, to achieve without it.
- Project managers in particular should have a significant level of systems engineering proficiency; there is much shared common ground between the two disciplines. A long-running debate about the separate roles of project management and systems engineering, partly driven by desires to establish identity and jurisdiction, has deflected the more important issue of commonality and consistency. They are "two sides of the same coin".

- The level of attainment in any one process may for some only merit awareness rather than a practical understanding. The way processes are defined and presented should thus be abstracted according to the level of proficiency expected, see Appendix B.
- The value to systems engineering practitioners of experience in one or more engineering specialisms and, through this, the benefit of their being able to relate across all engineering disciplines is clearly evident in the overlap of profiles. It underlines the benefit of a solid and in-depth foundation in at least one branch of engineering, either as a precursor or a complementary component of systems engineering competence development.

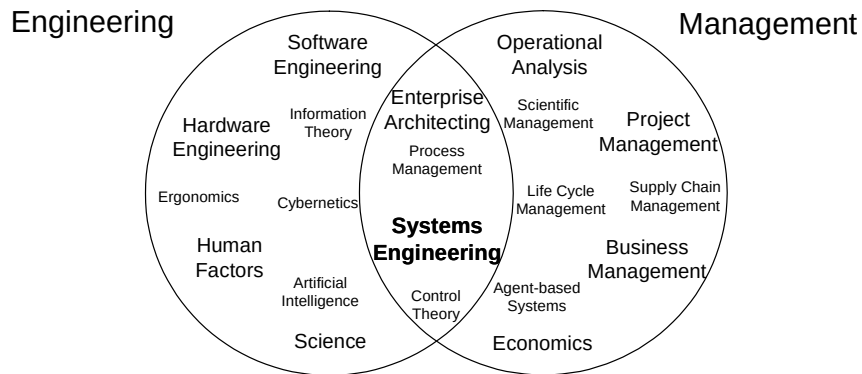


Figure 2.11 The Engineering and Management Context for Systems Engineering

An alternative picture of these overlaps of responsibility and influence is shown in Figure 2.11, in which Figure 2.4 and Figure 2.7 are superimposed. It provides a simplified map of to the engineering and management practice detail that forms the context for systems engineering. The relatively recently defined discipline of 'enterprise architecting', that has arisen in the IT community and is shown as straddling software engineering and operational analysis in this representation, addresses some concerns in common with systems engineering. It presents inconsistencies and opportunities with regard to systems engineering that are important to IPTs, and these are explored further in §10.4.

In the well-integrated organisation the models in Figures 2.5 and 2.8 mesh together to form Figure 2.12. With some adjustment to account for existing standards [Figure 4.13] and with the addition of explicit agreement processes that govern the fulfilment of trading commitments between cooperating organisations and that act as the 'glue logic' for the confederated responsibilities, this Figure defines the scopes and relationships of the regions of business process that determine the provisions of the ISO systems engineering standard considered in §4.

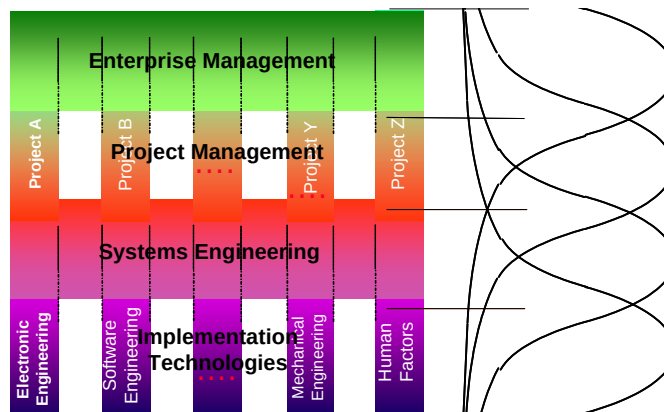


Figure 2.12 The Disciplines that Define Organisational Structure (left) with their Profiles of Influence Merging to Form the Well-integrated Business (right)

2.6 Outcomes

Whilst the individual intellectual and practical elements of present-day systems engineering may have existing over countless generations, it is only in the last half century that a coherent, business-relevant and socially advantageous branch of philosophy has gelled and, taken shape. It now appears to have completed its passage through adolescence. Its protagonists are thankfully ceasing to indulge in that combination of evangelical justification and introspective analysis that often characterises fledgling disciplines. They now look outward to systems engineering's wealth creating contribution to society, and how it constructively meshes with a range of longer established technical disciplines.

The immediate post-war years saw major change and dislocations in global social order; a by-product of which was a new engineering discipline. Since then the time and tide of political, economic and scientific situations has been forging systems engineering. The systems engineering of the 1960s differed in emphasis and some practicalities from that seen today. , Technology options, models of action, methods, teaming responsibilities, training, standards, tools, and user and society expectations have all shifted, and so has the paradigm of systems engineering in response.

Fresh interpretation, driven by this changing context, has brought shifts of emphasis to the scope and content of the discipline so that it may better fit prevailing business circumstances. This investigation is a contribution to the attendant re-interpretation and re-assessment of systems engineering.

This chapter has considered whether systems engineering is a sufficiently distinct, stable and valued platform on which to build changes to organisational operation and behaviour, particularly in MOD. It is concluded that systems engineering has without question reached the status of an international discipline, but falls short of being a profession.

With this maturity comes recognition. Systems engineering is now a generally accepted branch of engineering – accepted by academia, accepted by professional bodies and, most significantly, accepted by organisations as a cardinal element of a disciplined approach to business. Discipline is now a keyword in the systems engineering lexicon, as are two of its synonyms: profession and competence.

Systems engineering as a discipline is, it is contended, coming of age. It is moving quickly from the adolescent, self-justifying introspection of an emerging discipline to the purposeful responsibility of making a sustained socio-economic contribution. This is a necessary transition for any long-lived technical discipline. It is an essential step in maturity before propositions such as those offered in §9 and §10 are assuredly seized by MOD and its IPTs, and absorbed into the practices and culture of defence systems acquisition.

It is from a crafted, holistic amalgam of systems reasoning, engineering and management – with the whole amounting to more than the sum – that the full potential and relevance of systems engineering emerges. The progressive re-balancing of these three constituent components is the source of shifting boundaries, emphasis and applicability of systems engineering. This rebalancing is an important message from this research and the next chapter begins its delivery.

3 REASONING ABOUT SYSTEMS

3.1 Prologue

Addressing the mechanism of design, Dawkins observes that 'design explains the efficient complexity of microphones. Natural selection explains the efficient complexity of bat ears. Ultimately, natural selection explains microphones and everything designed too because the designers of microphones are themselves evolved engineers by natural selection' [Dawkins, 2005, p457]. The power of systems reasoning that nature has caused these evolved engineers to possess, and that enables them to create, refine and adapt designs, is advanced in this investigation as a cardinal constituent of systems engineering. Together with engineering and management, systems reasoning is one of three essential components of the discipline of systems engineering.

The evolution of the "Systems Movement" is illustrated by Checkland as having come from three areas of thinking; philosophy, natural sciences and social sciences [Checkland, 1999, p 97, Fig 4]. In this investigation, however, the balance is shifted, and so is the term used in order to reflect this. 'Systems reasoning' is offered as a step towards assimilating the 'mechanisms' of thinking about systems in a way that has particular relevance for systems engineering.

Without question, the social sciences en bloc have played their part. However, they took a prominent role in Checkland's considerations of 'unstructured', 'subjective problems'. In systems reasoning there is no such emphasis, and the congruous natural and social science are seen as having contributed similarly to a unified, widely-applicable understanding of systems.

The radical shift of emphasis in this text is the role of psychology. It has been distinguished from other sciences for it brings its own particular systems understanding. Psychology currently offers the best explanatory generalisations of how the human designer thinks about complex systems. It begins to pull back the veil concealing how engineers formulate and resolve designs, and then communicate these ideas and decisions in the necessary environment of the multi-disciplinary, integrated team.

In 2007 NASA Administrator Michael Griffin reflected that 'the development of formal methods has not altered in any way the fundamental nature of design, which still depends, as it did in antiquity, upon the generation of a concept for a process, technique, or device by which a given problem might be solved. The engineering sciences have provided better, and certainly quicker, insight for the designer ... but a human being must still intuit the concept. We have no idea how we do that. And until we do, we have little hope of developing a formal method by which it can be accomplished' [Griffin, 2007].

Thus, creativity is commonly seen as a mystic step between problem perception and the notion of a solution. It may appear to be transcendental but systems reasoning suggests otherwise. The adage "95% perspiration and 5% inspiration" is a fairer assessment of the relationship between disciplined foot soldiering and generalship from which creative design springs. It hints at there being a well-structured, even laborious mechanism employed to synthesise novelty and innovation. This hypothesis is presented in this chapter and the clues it gives about the working of the mental machinery of design are followed up in §4 and §10.4.

Well-structured and mechanistic all this may be, but it is no less remarkable for that. The capability to build a workable model of the semi-infinity of reality within the confines of the human mind, and then to find that it equips humans not just with a survival aid but with a predictive power that allows us individually and as teams to purposefully intervene in actuality, is a staggering outcome of natural selection. As de Broglie put it, 'this general agreement between external things and human reason is no small miracle, though it is to be observed that if it did not exist human life would be impossible, since in the absence of any relation between our understanding and facts, we should be incapable of foreseeing the consequences of our actions' [de Broglie, 1939].

It is therefore postulated that systems reasoning is genetically inbuilt. It is the basis of the analytic and synthetic reasoning employed by every scientist and engineer in order to unravel and compose complex things; it is a basis for creative thought.

Engineering is a technical discipline concerned with that part of the world created by man. It is fundamentally about modelling that world, predicting the effect of novel, intervening entities and then making this conjecture happen. It is a truly supreme faculty of mankind. As Theodore von Karme the aeronautics physicist and engineer put it: whereas 'scientists study the world as it is; engineers create the world that has never been'. Checkland makes the same point: science implies that the highest value attaches to the advancement of knowledge whereas engineering prizes most highly the efficient accomplishment of some defined purpose. The principal question that scientists ask is "have we learned anything" whereas the engineers and technologists ask "does it work" [Checkland, 1999], to which might be added "does it purposefully intervene".

If systems reasoning is a genetic inheritance then it might be expected that Dawkins has a view on the matter. His message, in a montage assembled from the *Blind Watchmaker*, is at one with the engineer's perception what we conceive a system to be and how systems reasoning relates to it. 'A necessary attribute of a complex thing, is that it has a heterogeneous structure'. 'Heterogeneity, or "many partedness", may be a necessary condition, but it is not sufficient'. 'A complex thing is something whose constituent parts are arranged in a way that is unlikely to have arisen by chance alone'. 'If we wish to understand how a machine (or a living body) works, we will look to its component parts and ask how they interact with each other'. 'A satisfying explanation has to be in terms of a manageably small number of interactions'. 'If there is a complex thing that we do not yet understand, we can come to understand it in terms of simpler parts that we do already understand'. 'The aptest name....is probably 'hierarchical reductionism'. 'The hierarchical reductionist ... explains a complex entity only one level down the hierarchy; entities which, themselves, are unlikely to be complex enough to need further reducing to their own component parts; and so on'. 'The behaviour of a complicated thing should be explained in terms of interactions between its component parts, considered as successive layers of an orderly hierarchy'. 'There is a hierarchy of sub components within components'. 'The behaviour will emerge as a consequence of interactions of the parts'. [Dawkins, 1986].

It should not be overlooked that this innate ability to conduct systems reasoning impinges on mankind's undertakings in general. It is far from being peculiar to engineering. Senge 'rediscovered' aspects of it in the management of organisations and curiously designating it the "*Fifth Discipline*", binding four, more obvious others that contribute to 'learning organisations'. He could well have built his management arguments starting from systems reasoning as his "First Discipline" [Senge, 1990]. Many disciplines could similarly benefit.

Philosophy, science and psychology comprise the essential intellectual amalgam of the systems reasoning advanced by this investigation, and this is the order in which they are considered.

3.2 Synopsis

Engineering has always implicitly drawn on systems-oriented principles and practices. However, a distinguishing characteristic of systems engineering is its continual reference and orientation towards an explicit, developed body of systems theory, knowledge, experience and practice.

Over much of the C20th pioneering workers made defining contributions from their respective areas of expertise to a body of knowledge having a common core of systems principles and knowledge, and these appeared to be based on an epistemological universality. Much of this body of knowledge is scientific theory and received wisdom gained from studies of control engineering, information science, biology, cybernetics, social science and the like. All this was particularly enriched by systematised heuristics distilled from many engineering and management careers. Often termed systems thinking, it is readily recognised by most systems engineering practitioners.

A truly holistic approach to the engineering and management of complex products and service thus has a necessary dependence in a systems approach. This is especially the case when the non-linearity arising from the tangled interactions of diverse technologies and inter-organisation trading conspire to present an otherwise intractable level of information and/or intolerable level of risk.

The unconventional emphasis conveyed in this text by the term ‘systems reasoning’ is intentional. Systems reasoning is not seen to be just an assembly of culturally-instilled, systems-oriented arguments concerning presuppositions, implications and interrelationships in the nature and structure of reality (systems philosophy – specifically metaphysics). Nor is it purely confined to a tailored body of system-based science that contributes to useful, working systems (systems science). It is not limited to the systematic reasoning strategy that explores intricate or messy problems, and is contingent on the strengths and the limitations of the working of the human mind (systems psychology). It is all of these – each feeding the other to furnish a coherent rationale for the strategic patterns of approach and the detailed tactics of method that underpin systems engineering.

‘Systems reasoning’ confers an identity on a particular framework of knowledge and intellectual strategy that (restrictively developed here) ties together in a consistent way important systems-oriented aspects of philosophy, science and psychology. It is seen to be a foundational layer for systems engineering. On it, the fabric of systems engineering is then assembled from components across engineering and from selected regions of management.

The term ‘systems reasoning’ is used to avoid any typecasting that may be conveyed by existing stereotype approaches, schemes of application, and methodological assemblages of principle and practice. Examples of these are ‘Systems Thinking’, ‘Systems Theory’, ‘Systems Approach’, ‘Systems Science’, ‘Systems Movement’, ‘General Systems Theory’, ‘Cybernetics’, ‘Systems Dynamics’, ‘Systemics’, ‘Total Systems Intervention’, ‘Complexity Theory’, ‘Catastrophe Theory’, ‘Operations Research’, ‘Soft Systems Methodology’, ‘Systems Analysis’ and the like. They are all contributors to some generality often termed “systems sciences” and each brings value, but they share a common weakness. No matter what knowledge they have marshalled, or what actions they advocate to unify, discipline or proceduralise, they do not explicitly capitalise on the individual mind’s reasoning about systems and how it translates human ‘intentionality’ into actuality.

The ‘reasoning’ in systems reasoning is therefore a deliberate and apposite descriptor. It acts as a reminder that it is not just a matter of system principles and theories. Importantly, it is also about an intrinsic, naturally-evolved region of cognition that enables humans to undertake purposeful, structured, technologically-based interventions. Critically, it is one of nature’s gifts to the competent systems engineer.

The pervasive nature of this systems reasoning has been subliminally absorbed by, adapted into and advanced by many ideologies. This is a strength and a weakness. The diverse application of its common doctrine across different communities is one of its most distinguishing and powerful features. However, scant individuality means that a growing medley of academics and everyday practitioners discover and re-discover its common essence from different viewpoints, and do so with limited consistency. In this regard, engineering, management, and systems engineering itself, are no exceptions.

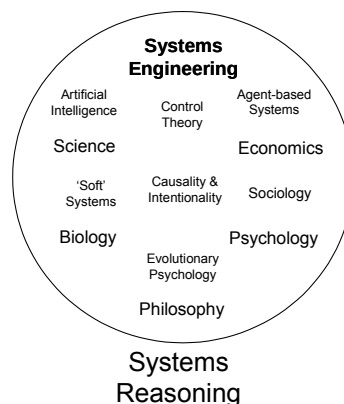


Figure 3.1 Some Disciplines that Have Contributed to and Employ Systems Reasoning

Systems reasoning is ideologically discipline-independent; assembled from a set of postulations, concepts and models from diverse sources. Largely concerned with order, structure, interactions and outcomes it is expressed here in a manner capable of interpretation by systems engineering. A representative set of its (non-engineering and non-management) contributing disciplines and methods is shown in Figure 3.1. This places detail in the region beneath Figure 2.2.

This then is the rationale for this chapter's reformulation and re-balancing of a common system-oriented foundation for systems engineering. It looks at systems reasoning from the standpoint of each of its three primary contributors: philosophy, science and psychology. There are, as should be the case, no sharp boundaries between each of these. Consequently, this categorisation may be invidious. However, it fittingly brings structure and order to what is a systems-centric chapter.

The opening topic – systems reasoning and philosophy – begins by illustrating how man has intellectually enquired about the real world and built an understanding of how this is perceived. From classical philosophy to modern-day thinkers, the first section picks out the cardinal influences on how the mind conceives of the real world and the systems it sees in it.

The next topic – systems reasoning and science – then assess the science-based contributions to systems reasoning. Out of this, and by way of considering the span of systems reasoning's influence and application, a simple model classifies the changing nature of systems as their perceived complexity increases. This is repeatedly used in later considerations. The model exposes differences and similarities between animate and inanimate systems, and considers common components in their through-life management.

The third contributor to systems reasoning – systems psychology – then explores how the mind has evolved to cope with complexity. This provides a cognitive basis for engineering and management, and provides clues as to the intuitive approaches that the able systems engineering mind follows. Abstracting the common techniques that cognition employs is a pointer as to how the mental models of reality are communicated between team members, each bringing their particular concerns and knowledge to the execution of a project by an integrated team.

3.3 Systems Reasoning and Philosophy

3.3.1 Classical Order and Rationality

If systems reasoning is indeed an innate constituent of the human condition, then it is only natural that philosophers should have grappled with its repercussions from the very earliest rational investigation into the nature and structure of reality (metaphysics) and the relationship between that reality and language, including non-linguistic models (semantics).

The accumulated body of systems-oriented philosophy that contributes to systems reasoning is an outcome of centuries of rationalism and empiricism. From a physical and then a metaphysical basis it has evolved through steps of development in both Eastern and Western philosophy.

In the earliest records of a systems philosophy Confucius considered that harmony, in its most concrete forms or manifestations, can be described as order and be expressed in terms of 'creative movement' [*Confucius, 470BC*]. Cosmic order and creativity are thus natural and intrinsic, or are established by mankind's intervention. They are products of the fundamental dynamics of the cosmos, and mankind contributes to this through procreation, formation, transformation and development. Definitions of this order convey a relationship between all things. They are expressed in terms of novelty, creativity, integration and value [*Tran Van Doan, 1997*]. They are cardinal attributes in systems reasoning and are primary qualities of C21st engineering.

Almost contemporaneously with Confucius, though quite independently, Western culture was addressing similar conceptual challenges. From its outset, Western philosophy was concerned with order and harmony, and with unity and rationality. System structures, their meaning, properties, dynamics and consequence were addressed in many directions of intellectual

enquiry. Over two millennia later some of this early philosophy looks anachronistic or mistaken in its offerings as science furnishes the knowledge that explains former mystery. Nevertheless, in fundamental areas of rationality, science has as yet no convincing answers to many questions on natural or man-made order that are as relevant today as when philosophy first raised and posited enduring responses.

Thales led the attempts to find naturalistic, physics-based explanations of the world, without resorting to prevalent supernatural or mystical beliefs of old. Though a philosopher in the Greek tradition, many now consider him the "father of science". According to Aristotle, he asked what was the nature (*physis*, Gr., *natura* Lat.) of objects such that they exhibited their characteristic behaviours. His rational views had a profound influence on other Greek thinkers and consequently on Western culture.

In this tradition Greek thinking steadily led toward a view that an object's goodness, in the Stoic sense of value judged across categories of knowledge, e.g. physics, logic, ethics, is characterised by order, structure, harmony and wholeness, and that these link naturally to rationality and benefit. This relationship of objects with rationality and benefit implies the notion of cause or motive, and to either the hand of divinity or man as the enabler of purposeful creation.

Plato's ideas about harmony were heavily influenced by the rationality of Pythagorean thought. Pythagorean philosophy took numeric principles and, by association, rules of reality or laws of physics to be the ordering and structuring principles of the whole universe [Plato, *Tim.*, 36 a-d]. Plato's notion of an extensive interconnectivity, leading to a universal whole, intimates the concept of universal order – the universal system – as signified by the Greek *kosmos* (order, world or universe) and present-day cosmos: 'the world or universe considered as an ordered system; any ordered system; harmony, order.' [Collins, 1991, cosmos, meanings 1, 2, & 3].

Even within the limits of scientific understanding in the classical world, it was seen to be the structure of the universal system that confers structure and stability at the micro-level of living systems. [Plato, *Tim.*, 44c, 82a-b]. This concept of levels of cosmic detail, influencing the structure and rationality of all subordinate levels, is both rational and, as argued in later philosophy, manifest.

Plato considered that the order, structure and rationality that is embodied in the physical universe is interpreted by the human mind at virtually every level of structural detail [*ibid*]. His dialectics stress that ontological concepts and mathematical relations represent different but complementary ways of defining structure throughout the natural world. Taken together they enable models to represent the architecture of the real world, and offer different degrees of fidelity, resolution and information about reality. It was an early declaration of present-day architecture viewpoints in systems engineering.

Originating from Plato, substance was divided into an important, co-existent dichotomy of matter and form; a principle termed hylomorphism. Stated in systems engineering (and classical physics) terms, this equates to separate models of implementation medium and structural relationship in an object's composition and architecture description.

The Greeks looked into the causes of natural phenomena to give understanding to the actuality in which man lives. Aristotle developed his theory of causality from earlier works that had lacked coherence and the systematic relationship that his insight brought [Aristotle, *Met.*, I 3-7, 985, 988]. It was from Plato he inherited the emphasis on form as a key to knowledge and also a belief that natural order is teleological - the result of design for purpose. Matter he saw as a passive principle, 'the operative principle is not matter but form'. According to the Stagirite, 'matter is the element that gives a substance its individuality [from all else that exists]; form is the element that gives it universality [recognisable structure]' [Aristotle, 350BC].

Hylomorphism is, however, only half of Aristotle's picture of object causation. 'We know a thing only when we can say why it is as it is – which in fact means grasping its primary causes – plainly we must try to achieve this with regard to the way things come into existence and pass away out of it'. 'Cause' here is a weak translation of Gk. αἰτιον (*aition*) which is perhaps better understood as 'factor responsible' or 'explanatory determinant'.

From these beliefs he bound hylomorphism with teleology to construct his doctrine of causation [Aristotle, *Phys.*, II.3], [Aristotle, *Met.*, V.2]

In it Aristotle advanced four very different kinds of explanatory principles or causes for the existence of an object. [Aristotle, *Phys.*, II.3, 194b24 ff]. There are two static explanatory principles of an object in isolation, i.e. what it is made of and how this is configured, and there are two dynamic and interactive explanations of an object, i.e. what they are for and how they came to be, the latter two being dependent on external influences.

Firstly came his Material Cause. In systems engineering terms the 'Implementation Technology Cause'. As he put it 'that from which, <as a constituent> present in it, a thing comes to be', or the 'material out of which a thing comes into being'. It encompasses the substance, material properties, functional possibilities, fabrication limitations of the material of a system that is worked upon (as by an artisan or technologist) – the *substratum*. The material cause is the elementary raw material of which a thing is produced, as from its constituents.

Next comes the Formal Cause, Gk. εἶδος (*eidos*), and this equates to the 'Architecture Design Cause': 'the form, i.e., the pattern ... the form is the account of the essence ... and the parts of the account' [*ibid*], which describes the architecture or associations that structure the parts (the elements, constituents, ingredients) to form the whole (the holistic system, composite or combination), thereby providing the part-whole causation or rationale of a system and the emergence of properties. It establishes the reference, design or architecture description against which the "matter" or *substratum* is to be worked in order to satisfy the moving agent of change, i.e. the goal, plan, end, or good.

These two, the material cause and the formal cause, establish the static explanations of an object.

The Final Cause, Gk. τέλος (*telos*), may be seen as the 'Service Capability Cause' and describes 'something's end, that for the sake of which a thing is done'. It establishes the purpose, useful effect or end state of affairs that an object is intended to serve. It is a natural outcome of its material and form. It is the effect on the real world that is manifest from the presence of the object's material and form at a particular location on the space/time manifold. For man-made systems it includes psychological causation, such as need and intent that give purpose to object behaviour.

To complete the four causes (though third in Aristotle's sequence of presentation), the Efficient Cause is of different kind, being the 'third party' agent of change: 'the source of the primary principle of change or stability'. It is the force, nonliving or living, immediately responsible for bringing the matter and form 'together' in the creation of the object. It is 'what makes of what is made and what causes change of what is changed'.

In the case of man-made systems, Aristotle sees the Efficient Cause to not just to be simply the human agent. More profoundly it is the causative principle that is manifest in specific knowledge, e.g. to design and produce. It is the body of knowledge (principle plus practice) rather than the individual practitioner's desires and beliefs that are salient. In systems engineering terms, the efficient cause encompasses the agent, e.g., the systems engineering practitioner and the relevant steps of change, e.g., the systems engineering processes. In terms of this text, it equates to the 'Systems Engineering Discipline Cause'.

The four causes are not exclusive but, in differing degrees, combinatorial and teleologically consistent. Thus an adequate explanation of real world change rests on all four.

Ontological causality does not suggest a temporal relation of before and after; of cause and effect. The primacy and associative patterns of cause are reactive or discretionary; there is no waterfall structure in the Efficient Cause, no reductionist strategy in the Formal Cause. Causality can follow different strategies – requirements (Final Cause) led, e.g. top-down; implementation (Material Cause) led, e.g. bottom-up; architecture (Formal Cause) led, e.g. middle-out; or any combination of these.

Aristotle's text is worded to apply to both animate and man-made systems and this (together with translation) makes its joint ontological structure awkward, a factor encountered in §3.4.2 in the comparison of natural and man-made system life cycles. However, when his doctrine is applied to the realm of man-made systems it provides a clear-cut, 3-layer model of causation or transformation, i.e., service capability, architecture design, implementation technology, that is conducted according to a through life sequence governed by the efficient cause. This layer

model of causality emerges as a central axiom in ISO's C21st codification of systems engineering.

The work of Aristotle and others handed down a formalism regarding any object's 'being' that has largely remained intact, despite numerous philosophical, scientific and psychological reassessments. A rational and enduring foundation of systems reasoning was laid by this early Greek enquiry into the order and meaning of reality. Though the term system, in its current usage, was not employed in this classical reasoning, its notions and semantic influences are evident in the Greek *σύστημα*, (*sustema*). Etymologically this stems from *syn-*, meaning together or fusion (denoting integrated), plus *histanai*, to cause to stand (denoting to put in place a structure). This commences a linguistic thread through Late Latin *systema* and French *système* (C17th) to the present English language word [Collins, 1991, system].

This linguistic heritage signifies Greek society's recognition of the importance of the principles of order in form or construction, as well as purpose. It gave powerful vitality to Greek and subsequent philosophical attention to the order and nature of reality. It is recognisable in Greek ordering and disciplining of human skills, where it can be readily seen in architecture and in their main branches of scientific thinking, and was often equated to essential rules or standards of action associated with human endeavour.

With this appreciation of the skills required to define and construct system form, which the Greeks termed *tekhnē* meaning skill or art, and from its association with the importance of civil skills such as building, came an acknowledgement of the need for leadership and direction of 'technical' teams. Endeavours needed to be led and coordinated by a highly competent and regarded practitioner, by an individual who was in Greek *archi*, meaning first in position or rank. From this notion comes the term *arkhitekton*, leading through Latin and French to 'architecture' and the present-day engineering term. The association of 'architecture' with systems skills is thus long standing.

The Romans acquired the writings of Aristotle and other Greek philosophers, though for them it was the practical outcomes of engineering that was the important factor in their view of, and dominance over the world. Thus, though the Greek for skill, *tekhnē*, ultimately traces to 'technical', it is from the Latin for skill, *ingenium*, and the strong Roman association of skill with engineering, that 'engineering' owes its linguistic origin.

Time and evolving society values have seemingly transformed the classical etymology for a "skill for integrated structuring" into "systems engineering".

3.3.2 Reductionism and Holism

It may be culturally significant that the disciplined, subjugating Romans bequeathed the motto *divide et impera*, divide and conquer, whereas the partisan Greeks left us the quotation 'united we stand, divided we fall' [Aesop]. Each of these radically different systemic strategies of conflict and problem resolution have figured down time. They present a philosophical seesaw that persistently dogs approaches to systems reasoning: the balance between, or doctrinal precedence of, reductionism and holism. Repeatedly, a favouring or subjugation of one over the other is the source of blame for the failings of systems reasoning and system solution deficiencies.

The Ages of Reason and of Enlightenment started an eclipse of scholastic philosophy and the destruction of Aristotelian physics [Butterfield, 1957, p.viii]. Amongst the thought leaders of this was the "father of modern philosophy", Descarte, and it was his fusing of philosophy and science that brought reductionism and holism to the fore.

Descarte advanced 21 general rules for scientific method [Descarte, 1625]. He addressed reductionism in Rule V. He expounded that 'we reduce complex and obscure propositions step by step to simpler ones and then try to advance by the same gradual process from the intuitive understanding of the very simplest to the knowledge of all the rest'. This analytical reduction of complexity into separately tractable components, in order to gain a greater understanding, has remained a powerful method of natural and applied sciences, and of mathematics. It is one foundation of modern scientific method: systematic inquiry and empirical observation.

Of the present and from the standpoint of an evolutionary philosopher, Dawkins summarised Descarte's rule. 'If there is a complex thing that we do not yet understand, we can come to

understand it in terms of simpler parts that we do already understand. This ‘hierarchical reductionism’ can provide satisfying explanation in terms of interactions between its parts, considered as successive layers of an orderly hierarchy of manageably small [to humans] number of interactions’ [Dawkins, 1986].

Reductionism therefore approaches the analysis of complex things or ideas in terms of less complex constituents: simpler parts or components that can be treated in isolation and so rigorously understood. Despite its power and indisputable ability to make tractable the intractable, reductionism is often linked to disparaging condemnations of systems engineering. Donne lamented ‘‘Tis all in pieces, all coherence gone’ and with this short stanza (on the passing of The Age of Enlightenment) [Donne, 1621] he pinpoints a weakness of reductionism – that in the non-linearity of reality, a coherent whole offers more than the sum of its parts. Again of our time and by Dawkins in his *Eulogy to Douglas Adams*, ‘if you try and take a cat apart to see how it works, the first thing you have on your hands is a non-working cat’ [Dawkins, 2001].

The antidote to reductionism is holism: a philosophical leaning that advances that the significance of the parts can only be understood in terms of their contribution to the significance of the whole. In consequence, it holds that the whole must be epistemologically prior – a corollary of which is that the whole may indeed have properties over and above those of its parts.

Descartes’s reductionism is tempered by commentary to Rule XII in which he elucidates ‘all human knowledge consists of this one thing, that we perceive distinctly how these simple natures combine to produce other things’ – in system terms, a message of holism and emergence. However, Descartes Rules do not satisfactorily resolve that complex phenomena are not logically consequent from ‘simple’ phenomena. Despite pejorative implications in “Cartesian reductionism”, Descartes did hint at a satisfactory ideological basis for systems reasoning; one in which holistic synthesis plays its part – his bias just happened to be towards analytic reduction.

This reductionism/holism balance presents systems reasoning with a paradox: the primacy of the whole opposed to the primacy of the parts. Neither whole nor part has epistemological precedence. Depending on an observer’s specific concerns and motivations, the mind has a propensity to apply one or other of these reasoning modes to the domination of the other.

The observer is trapped in this paradox, for the whole/part relationship is recursive. Any whole-of-interest is a contributor in some greater whole and thus only fully understood in terms of its context of existence. At any instant, the sublime significance of an entity can only be understood in the context of every other entity and, if any one entity were fully understood, all others would have to be also. The information handling would be untenable.

A third philosophical doctrine, atomism, is related but distinct. It too plays an important role in systems reasoning. Atomism adheres to the concept that reality is constructed out of a small number of distinct types of simple indivisible entities. In contrast to holism, it holds that an understanding of the parts is logically prior to an understanding of the whole. Ignoring the inherent tension between views of the whole over parts and visa versa, the power of the atomistic theory lies in its ‘black box’ ability to encapsulate low level detail within a closed boundary. Whether axiom or expediency, it places a lower bound on the level of detail considered in a particular situation. Linked to effective boundary definition, it is a powerful manoeuvre for information handling and containment, and also for encapsulating delegated engineering responsibility.

Given the pejorative accusations levelled at systems engineering, it is scientists who are the reductionists. They see their role as deconstructing our world to its elemental workings and from these to advance knowledge by postulating and testing its other elemental workings. Technologists on the other hand are inherently holists, taking these elemental workings to create novel workings, unthought-of by nature.

Where does this place engineers and systems engineering? Of necessity, in both camps: reductionist and holist.

For systems engineering the synthesis of holistic solution at one level of system detail is founded on a synthesis of contributory, detailed holism arising from its immediately subordinate level, and so forth ascending from ever finer detail. In no way can this be construed as a

reductionist strategy – it is every bit a scheme for holism. Most certainly, system creation employs reductionism, but it perforce incorporates holism, ultimately in equal measure – a ‘V’ has two equal sides. Both are essential, distinguishable contributors; each depends on different orientations of enquiry and skills.

This reductionism/holist tension has been a fundamental and far reaching antinomy of systems engineering. It has spawned recurrent pointless invective, yet there is barely a debate to be had. Holism and reductionism are necessary, complementary components of systems reasoning. On this matter Kant was unequivocal: ‘there are things whose parts exist for the sake of their whole and their whole for the sake of their parts’ [*Kant, 1781*] – that, in the philosophy of systems, is how it is.

3.3.3 Latter-Day Philosophy

Through to the C20th the ontological branch of philosophy has been concerned with what a physical object is, what its existence means and what constitutes its identity. Although Plato and Aristotle studied the conceptions of reality and the nature of ‘being’, it was Heraclitus who had earlier advanced the importance of change and transformations of objects – ‘everything flows and nothing is left [unchanged]’ [*Plato, Crat.*, 402a]. For the next two and a half millennia much philosophical attention was directed to understanding these essential attributes of an object; its material properties and its relation to other objects, and temporal issues associated with its state and state changes, including when it goes out of existence. Philosophical in their style, these questions nonetheless have considerable importance to the engineering perception of systems in the real world.

Since the C18th many have addressed this link between mind and the external world – the relationship between awareness and reality, and the tension between the subjective and objective. In this regard Kant made a massively important breakthrough [*Kant, 1781*] when he related cause and effect without relying on the precedence of sensory empirical knowledge. This had been seen as a particular weakness of Hume’s conclusions [*Hume, 1739*] and in Leibniz’s earlier rationalism. Hume, had advanced that propositions must be either analytic and thus *a priori* or synthetic and *a posteriori*. This places man-made design in a quandary: either design is the outcome of an emergence of order or properties already known, with unprecedented properties always being predictable, or design is the outcome of empiricism and rationalisation (and that trial-and-value is the determinant of advancement). Given the improbability of predicting the unprecedented, Hume’s philosophy would consign man-made creativity to the same footing as evolution in nature. Kant dispelled this paradox by positing a third category, namely propositions that are synthetic *a priori* – a supreme characteristic of originality. This opens the door to creative design that is not contingent on empiricism.

Kant’s work also led to his postulation that the real world of experience can only ever be an appearance or phenomenon. What things are in themselves are completely unknowable by the mind.

To account for these cognitive capabilities, Kant invoked synthetic reasoning and a set of ‘hardwired’ truths: a system of mental transformations that when executed purport to represent some aspect of the world. ‘Since ... [the observer’s] capacity to be affected by objects must necessarily precede all intuitions of these objects, it can readily be understood how the form of all appearances can be given prior to all actual perceptions, and so exist in the mind *a priori*’ [*Kant, 1781, A26/B42*]. That is, it is an *a priori* cognitive faculty allows us to see the spatial and temporal conditions of sensibility. Space and time are conditions of perception furnished by the mind. They are what we might regard as being the subjective (and commonly shared) representation of the objective phenomenal world. As Einstein put it, ‘time and space are modes by which we think, not conditions under which we live’.

Paraphrasing Kant: immediate knowledge of experienced objects, i.e., intuition, in the absence of a notion of the characteristic or essential features of that class of objects, i.e. concepts, is visionless; whereas concepts without intuitions are meaningless [*ibid, A 51/B 75*].

Likewise, causality is a conceptual organising principle that we impose upon nature. However, it is a paradox of Kant’s *Critique* that, while we are prohibited from absolute knowledge of the thing-in-itself, the observer may impute a cause to it.

Through these arguments Kant established the elements of a systematic framework used to structure experience. These spatial, temporal and causal dimensions of object understanding are associated with an identity or referent of the object under consideration – here labelled ‘substantive’, as in ‘relating to, containing, or being the essential element of a thing [that has] ... independent function, resources, or existence’ [Collins, 1991, substantive: meanings 2, 3].

Thus, it is through a pre-structured activity of the mind, that a composite model of reality is built. It draws ontologically on substantive, spatial and temporal aspects of objects, and on an imputed causal aspect. Objects appear to an observer via a faculty of transcendental imagination that is grounded systematically in accordance with these four aspects of understanding. Innate cognitive faculties of the observer build this composite model of reality.

Whilst the observer’s empirical knowledge is real, he is blind to the operations of the cognitive faculties that transcend it. Through their mental transformations, a model reveals the known world and also the knowable world – in the case of systems engineering, a conceived-of, yet-to-be-built system in a projected operational situation.

Kant’s mental model framework thus comprises four constituents by which an object is known and comprehended:

1. a substantive aspect of the object of knowledge;
2. a spatial aspect of the perception of it;
3. a temporal aspect of the perception of it;
4. a concept of causality in nature into which the observer fits experience.

Brentano, who had a special interest in Aristotle, advanced that every mental phenomenon, every psychological act, has a content directed at objects and events in the external world. In a shift from causality, he ascribed intentionality to objects by identifying them as ‘intentional objects’, though his philosophy does not confer anthropomorphism on the inanimate.

For Brentano, every belief, desire, need or requirement of man relates to an object that is ‘about’ the believed, wanted or required. ‘Every mental phenomenon is characterized by what the Scholastics of the Middle Ages called the intentional (or mental) inexistence of an object’ [Brentano, 1874, p. 88-89]. Since physical phenomena lack intentionality, Brentano argued, that being intentional is a primary distinguishing feature between mental phenomena and physical phenomena, i.e. between a concept of system and a real world system.

For a time, Jean-Paul Sartre influentially challenged this, equating intentionality directly with consciousness and stating that the two are indistinguishable from one another [Satre, 1943]. However, most present-day theories now align with Brentano’s thesis of the irreducibility of the intentional meaning.

Some philosophers still consider intentionality as a problem that is intractable to science. They adhere to a realist view, imputing that entities actually intrinsically possess intentionality. This would imbue inanimate man-made systems with the otherwise exclusively animate property of cognition – systems would have to actually comprise physical realisation of mental representations [Dennett, 1987, p345]. Even in the creation of the most recent and advanced artificial intelligence systems, where intentional viewpoints are now being effectively employed in information systems engineering [Vilkomir, 2004], this is not palatable to engineering reasoning.

Although Quine posited that ontologically objects are not intentional, he nonetheless used the instrumentalist argument to assert that the language of intentionality is indispensable for relating real entities and mental constructs of them [Quine, 1960].

Sitting on the fence of this dichotomy of realism and instrumentalism is Dennett. Building on intentionality, his views transpose philosophical aspects of reasoning about objects or systems into the realm of psychology, and this is considered in §3.5.2

It is Dennett’s judgement that ‘Brentano ... proposed intentionality as the mark that sunders the universe in the most fundamental way: dividing the mental from the physical’ [Dennett, 1971, p.87-106], [Dennett, 1978]. In so doing, it distinguishes the rationale in a system’s design (description) from the actuality that is the realised system and, intriguingly as discussed in §10, a glimpse of the rationale behind the actions of design.

3.4 Systems Reasoning and Science

3.4.1 A Common Scientific Insight

Whilst the philosophy of systems has spent much of its history reflecting on a reality that is the result of a natural order of things and of mankind within it, it has paid much less attention to how man contributes to the construction and controlling of order. During the 20th century, however, philosophers increasingly tried to reframe questions about 'being' in terms of man-made objects in an environment. Successively, this was frustrated by and then drew from insights derived from scientific research into animate systems in natural and artificial settings, and from mathematics and physics exploring fundamental understanding about 'being' of the inanimate, including particle physics, quantum effects and uncertainty.

Systems science took a more pragmatic view than philosophy. Its objectivity and discipline towards a systems-based body of knowledge emerged from a progression of 'revolutions' in science and engineering that profoundly shaped understanding of the physical world and the nature of mankind's relationship with it. For science, and for Penrose, the 'physical world is depicted according to mathematical laws, by equations, or perhaps by some future mathematical notation' [Penrose, 2004, p18].

Thus to complement the system philosophy view, this section on systems reasoning reflects on some guiding systems contributions from different sciences. These are chosen for their bearing on the way systems reasoning is applied in systems engineering.

Throughout the C20th there was a string of revelations in many branches of academic discipline regarding the relevance and contribution of the "Systems Movement" to thinking in these fields. As each field 'discovered' and resolved an explicit component of systems thinking, analysed its significance and saw its generality, a growing body of systems theory and understanding began to be assembled. Diverse fields both contributed to and benefited from this transferable set of notions. Systems concepts such as complexity handling strategies, boundary definition, hierarchy relationships, abstraction, state definitions, feedback and regulation, and information theory were seen to have wider relevance than their originating discipline. Notions in one regime of science were recognised in others. By mid-century the realisation of this triggered an evolutionary step based on 'systems science'. The post war impetus brought a new age of system-based views of technical and management sciences.

In his treatise on scientific revolutions [Kuhn, 1962] Kuhn observes that Aristotle's *Physics* was astonishingly unlike Isaac Newton's *Philosophiae Naturalis Principia Mathematica* [Newton, 1687] in its concepts of matter and motion. He argues that ideas, intellectual options, strategies, lexicons and terminology have their age, and that the evolution of scientific theory is an emergence from a set of changing intellectual circumstances and possibilities, not an accumulation of knowledge or method. Nonetheless, Kuhn concludes that the concepts of Aristotle and other philosophers are not to be seen as 'bad Newton', simply contrasting and different models of reality that empathise with different epochs, frontiers of knowledge and cultures.

A regime of explicit systems science was one such intellectual circumstance, but it took over one and a half centuries to emerge and become established. By the start of the C19th the notion of system was entering common usage. It was described in *Encyclopaedia Britannica*: 'SYSTEM, in general, denotes an assemblage or chain of principles and conclusions, or the whole of any doctrine, the several parts whereof are bound together, and follow or depend on each other, in which sense we say a *system of philosophy*, a *system of divinity*, etc. The word is formed from the Greek [for] "composition, compages"' [EB, 1797, entry for system]. Whilst this definition is directed toward conceptual systems, the term was also being applied to the physical world. An observation by the theologian William Paley, that "the universe itself is a system; each part either depending on other parts, or being connected with other parts by some common law of motion" [Paley, 1819, p398], indicates the growing appreciation across intellectual society of a system view of the world and universe.

Thus, whereas some system axioms from philosophy were appearing esoteric or anachronistic, science was setting about establishing a firm, pragmatic footing of for systems reasoning during C20th. This led to solid and practical results with broad applicability. Evidence for the soundness of these scientific advances can be seen in the crossover and the interpretive power

they offer between different branches of science and engineering. Biology, sociology, economics, mathematics, informatics, electronics and computer science were all contributors to the pool of transferable knowledge on systems, their descriptions, distinguishing patterns, behaviour and influences.

Early thinking came from economics, politics and humanitarian studies with Hume [Hume, 1752] and Malthus [Malthus, 1798] examining socio-economic systems and the effects of the properties they possess.

By the early C20th it is not surprising that rigorous insight into formalised systems reasoning was coming from animate systems – biological and social – for the pickings of complexity were much richer and more pressing than for inanimate system. Biology had grown up with vitalism and it took Cannon on homeostasis [Cannon, 1927] and Koehler's work on open and closed systems [Koehler, 1938 pp. 314-328] to illustrate how complex biological structures operated in terms of organised, deterministic forces.

These were important precursors for v. Bertalanffy who from 1932 onwards developed reasoned systems arguments on the theory of open systems in biology and physics, ultimately producing a seminal work in 1963 [von Bertalanffy, 1963]. This text extended Boulding's work [Boulding, 1953, 1956a p 11-17, 1956b, p66-75] and from it came across-the-board insights into systems. This led to the formalisations in which he developed the vocabulary and popularised the title 'General System Theory' [von Bertalanffy, 1950, p.23-29, 1968]. With this he attempted to develop a mathematically exact theory in the non-physical sciences, emphasising holism over reductionism, and organism over mechanism [v. Bertalanffy, 1975].

It is indicative that the Macy Conferences between 1946 and 1953, which started as an experiment in multidisciplinary science, continually alluded to the need for a 'meta-language' that could bind the different sciences represented. The handful of attendees included luminaries Margaret Mead, John von Neumann, Norbert Wiener, Claude Shannon, Ross Ashby and Talcott Parsons. Considerable attention was given to subjective perception and its relationship with neural mechanisms. Given the mood of the age, a strong 'systems reasoning' flavour quickly surfaced from the multidisciplinary exchanges and, though no unified theory or meta-discipline spun out of it, the conferences endorsed the term cybernetics – a Platonic appropriation of κυβερνήτης, steersman – which Wiener went on to develop formally. In it he drew on analogies between organisms, social systems and computer systems, and he considered the flow of information round a system, examining how this establishes the means of system 'self' control.

Some highly practical methods emerged from the theory being developed. Systems Dynamics evolved around 1956 as an offshoot of the SAGE (Semi-automatic Ground Environment) US military programme. Forrester, with his electrical engineering background, described this method for understanding dynamic behaviour of components from the feedbacks and time delays that arose from their individual functions and their overall structuring. He recognised ways of analysing social systems by introducing engineering descriptions of systems into other domains of application, including the design and control of an industrial organisation, which he saw to be a particular instantiation of an "information feedback system". [Forrester, 1961].

With the increasing formalism, connectedness and predictability seen in natural science the social sciences joined the quest for a systems science basis and in turn it was the social sciences that made a contribution. Evolutionary theories of economics constructed directly from the theory of evolution and systems theory were developed by Alchian [Alchian, 1950, pp.211-21], and later by Georgescu-Roegen, who also showed the significance of entropy in the field of economics [Georgescu-Roegen, 1971], and Boulding [Boulding, 1981]. This theory places economic systems more or less into a biological model of evolution.

With its focus on social systems and the social action of the individual humans within them, sociology concerned itself with the axioms, rules and processes that bind and distinguish people as individuals, members of associations, groups, communities and institutions. Parsons' sociology ideas were developed during the period when systems theory and cybernetics were very much in the spotlight of science. Applying systems reasoning, he advanced that the systems treated in social and behavioural science were open systems residing in an environment consisting of other systems. Attempting to make sociology a social and behavioural science, he treated society as the largest system, comprising the interrelated behaviours of human beings in a physical-organic environment.

His system-based approach began by examining the individual and their actions, and then the interaction between two individuals [Parsons, 1961, p.41]. In this way he started to build equivalent models to Figure 3.2 starting from its second-level animate layer upwards to the top. In it, behaviours are repeated as more and more interactions occur, finally appearing as entrenched roles in enterprise structures. In turn these roles become bound up in institutions and social structures, such as political, legal and educational.

From this he produced a general theoretical system for the analysis of society that in time became called 'structural functionalism'. Ultimately, this hierarchy of functionalism becomes a sociological paradigm that then explains strategic social institutions as the collective means to fill the needs of the individual.

Management science too made its contributions and these started early in the C20th building from a systemic management movement that arose in the wake of railway building. Taylor, "the father of scientific management", infused scientific principles into management [Taylor, 1911]. He applied scientific reasoning, or more specifically systems reasoning, to industrial work and showed that labour can be analysed and organised by focusing on its elementary parts. He demonstrated his thinking in the tasks found in steel mills. "In the past the man was first, in the future the system must be first" is a quip attributed to him that foretold of 'conditioning' humans to be integrated into the design of complete work systems.

Returning to a biological metaphor, *The Theory and Management of Systems* [Johnson, 1973] described how a modern business is like a human organism, with a skeletal system, a muscular system, circulatory system, nervous system, etc. However, the effectiveness of the 'hard approaches' to systems was seriously challenged in the 1970s in regard to its ability to resolve organisational management problems. The 'failure' of Management Science and Operational Research was strongly debated by Churchman [Churchman, 1971] and Ackoff [Ackoff, 1979].

Checkland [Checkland, 1981] examined the relationship between 'hard' problem thinking, which used 'traditional systems engineering' ideas as a mechanism for choosing between alternative ways of achieving a well defined goal, and 'soft' problem thinking, in which methodologies or systemic approaches use system models as mental constructs for perceiving problematic 'situations' with the view of bringing improvement and learning by this means. The claimed dichotomy in hard and soft system problems is reflected on in §10.5.3 from a common systems reasoning position, with the design principles and strategies developed in §4.4.1 and §10.4 seen to be capable of spanning a wide range of hard to soft problems.

Thus it has been, that many branches of natural and social sciences have contributed to a body of system-oriented knowledge and understanding which has in turn influenced the development and control of systems in these and other individual domains of interest. A classification of the systems-of-interest from across this foundation of scientific systems reasoning can offer some helpful system generalisation and insights. It has a direct bearing on the range and nature of systems engineering's application, and is considered next.

3.4.2 A Classification of Systems

Classification is about distinguishing the same and the different; about order, structure and understanding. Indeed, classification itself is integral to systems reasoning, cf. 'system: any scheme of classification or arrangement' [Collins, 1991, system: meaning 2]. In general, the grouping or ordering or classifications designed to share or linearise characteristics or qualities in member elements so as to convey particular understanding of a whole population. Accordingly, a range of classification schemes of systems (many linear hierarchies) has been used by workers from different branches of science to gain insight into the nature and working of systems.

The variety seen in these classification schemes emphasise that 'order can be different for different human beings, having their separate concerns, cultures and mental constructs. Classification conveys significance to the classifier' [Foucault, 1966]. Borge starkly illustrated this by the bizarre and anachronistic animal classification found in an old Chinese encyclopaedia [Borges, 2000]. Latterly, even the rather more rational Linnaean taxonomy of living things has changed since its inception to convey present-day knowledge, interpretations and concerns.

According to Magee there are three inter-related reasons for classifying systems [Magee, 2004]. Paraphrasing him, classification:

- delineates the intellectual boundaries of system-orientated disciplines;
- structures academic fields of study;
- contributes to the engineering and management of systems.

Magee offers a framework that itself classifies system classifications according to different independent variables of ordering, and which adapted is presented in Table 3.1. Other classifications have been developed that employ more than one independent variable [Jordan, 1968], [Ackoff, 1999, p.20-25].

Attributes	Reference(s)	Specific Qualitative Scale
Degree of Complexity	(Hubka, Eder 1988), (Haberfellner, 1992) (Pahl, Beitz 1996)	Increasing technical complexity
Branch of Economy	(Hubka, Eder 1988)	Business domain grouping
Realm of Existence	(Haberfellner, 1992)	Real vs. virtual
Boundary	(Haberfellner, 1992), (Bertalanffy 1968), (Boulding 1953)	Open vs. Closed
Origin	(Haberfellner, 1992), (Bertalanffy 1968), (Boulding 1953)	Natural vs. Artificial
Time Dependence	(Haberfellner, 1992), (Bertalanffy, 1968), (Boulding, 1953)	Static vs. Dynamic
System States	(Haberfellner et al. 1992)	Continuous, discrete and hybrid
Human/Control	(Ashby, 1963)	Autonomous/human in the loop/mixed

Table 3.1 System Classification Schemes (Adapted from [Magee, 2004])

There have been several helpful classification structures for systems, grouping like phenomena along a scale of significance in order to promote understanding. Many comprise a succession of classes that possess an incremental increase in complexity, and which in some degree incorporate all subordinate entries. As Polanyi describes, 'each level relies for its operations on all the levels below it. Each reduces the scope of the one immediately below it by imposing on it a boundary that harnesses it to the service of the next higher level' [Polanyi, 1968, pp.1308-1311]. Aggregations at any level may act to produce systems in a higher class and these attract a greater order of complexity. At each level in the hierarchy novel attributes or phenomena, not seen in the subordinate classes, occur.

Table 3.2 is selected to represent some system classifications spread over a half century and two continents. They are structured according to different disciplines and to correspondingly different arguments being presented. No attempt is made to establish equivalences between them since, as Faucault advances, each classification exists to serve the concerns or arguments of the classifier, and there are no collective dimensions or common scale to their class structure. There is nevertheless a clear ascending level of complexity and problem-solving challenge in each column. They broadly reach down to the deterministic and rise to the indefinite and unspecifiable. Each echelon in the structure encompasses the contributions of all those beneath and new characteristics emerge at each new level that did not exist at subordinate levels. It is the emergent properties that primarily define the strata in each of these hierarchies. Each hierarchy is (literally) a system of systems.

In the first column of Table 3.2 Boulding, an economist by training, first postulated a system classification structure based on theoretical models that had been developed from different disciplines. In doing so, he was partly attempting to show the status and unifying power of science, calibrating each class of system against a common theory of systems [Boulding, 1956a].

Next, von Bertalanffy came with his biology background, seeking to develop a mathematically exact theory in the non-physical sciences [v.Bertalanffy, 1968]. He drew on other sciences to build his structure, which sympathises with that of Boulding.

Boulding 1956	Bertalanffy 1968	Checkland 1981	Hitchins 2003
Economics	Biology/Physics	Management Science	Systems Engineering
Social organisation	Symbolic	Transcendental	Socio-economic systems engineering
Human	Socio-cultural	Human activity	Industrial systems engineering
Animal	Man	Natural	Business systems engineering
Genetic/Societal	Animals	Designed physical	Project systems engineering
Open Systems	Lower Organisms	Designed abstract	Product/subsystems engineering
Cybernetics	Open Systems		
Clockworks	Control Mechanisms		
Frameworks	Static Structures		

Table 3.2 Classification Structures for System

Following this, Checkland offers his view as a management sciences academic with practical experience and a keen sensitivity for sociological issues [Checkland, 1981]. His five classes of system serve to structure the universe according to a coarser calibration; which in practice is really four with its uppermost 'too difficult' box.

Lastly in this selection of classifications, Hitchins examines the complete world of man-made systems categorised according to their 'engineering'. At each level is a flavour of a ubiquitous discipline of systems engineering. His classification thus advances that systems engineering applies to any man-made system irrespective of its predominant implementation medium/media. Each systems engineering prenominal in the hierarchy segregates regions of the discipline's application according to the environment that systems engineering is applied to.

Hitchins' hierarchy spans five levels: socioeconomic structure, the stuff of regulation and government control (Layer 5); industrial structures including complete supply chains/circles (Layer 4); individual business organisations, where systems engineering seeks to optimise performance somewhat independent of other businesses (Layer 3); projects where engineer-managers create complex artefacts (Layer 2); products, the traditional ground of systems engineering (Layer 1) [Hitchins, 2003, p.312]. Though the model is strictly a classification of systems engineering application domains, it also equates to a system class structure.

As a systems engineering thought leader, Hitchin's is widely referred to and his classification serves some systems engineering purposes well. However, it does make clear that human/social factors contributions may be present at any level of system complexity, it emphasises business organisations, and it compresses the conduct of engineering and basic technology into one class. In short, it biases systems engineering toward social and business concerns, and away from engineering.

The chronology of the classifications in Table 3.2 shows evidence of a shift from a 1950's concern with scientific understanding of systems as mechanisms towards a C21st need to understand how systems are created in a business-based society. The early post-war concerns were on refined scientific distinction of increasing system complexity. Attention has clearly moved towards placing systems in a business and economic context, with a mapping to the people-oriented structures that contribute to their effective provision.

None of the classifications in Table 3.2 convey a mutually consistent dimension or scale in their progression of classes. Nor do any of them clearly draw out the relationship between natural and man-made systems, despite their human and social content emphasising that consideration of inanimate systems in isolation is ineffective. This underlines the weakness of viewing systems

engineering without due regard to humans as a part of the system-of-interest and part of its lifetime of environments, as considered in §10.5.

Given the limited generality of any classification of systems, this investigation offers one more scheme, see Figure 3.2, with the justification that it resolves weaknesses in other schemes and supports ensuing arguments.

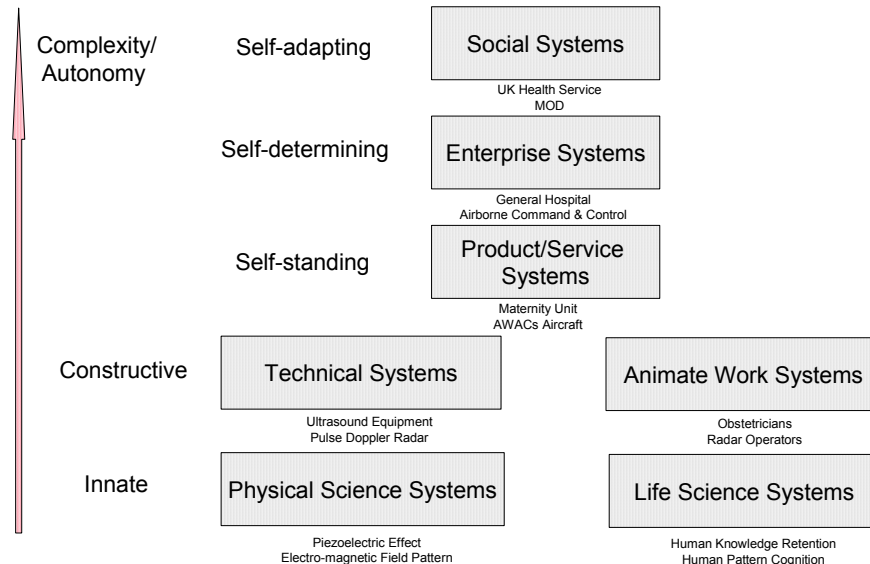


Figure 3.2 A Classification of Natural and Man-made Systems

The system developed to support this investigation has a base layer of natural phenomena, rising to the complexity of social systems at its vertex. Under each class there are typifying instances that build two consistent, illustrative threads; one social services, the other military capability.

The parallel strands in the lower levels of Figure 3.2 separate the inanimate and animate lines of increasing complexity before they coalesce into system classes that increment toward socially-dominated complexity. The classification thus has to some extent two orthogonal independent variables: complexity and creative origin (man and nature).

Innate systems (as in 'being an essential part of the character of a person or thing' [Collins, 1991, innate, meaning 2]) form the base layer of this classification. The physical science systems (bottom left) comprise the basic structure of physical phenomena. They arise from the existence of matter and energy, and from their conversion or transformation, e.g. nuclear processes, crystal formations, electromagnetic fields, energy transducers. They possess naturally emergent phenomena that are determinable by reductionist approaches.

At the level of emergent property/complexity above this, natural phenomena are purposefully combined to create artificial technical phenomena synthesised from different technology foundations. This is the domain of technology advancement that initiates, or that is inserted through life, into products such as nuclear reactors, mechatronics, radar, medical equipment. This level is termed constructive, as in 'to shape or influence, to give direction to'.

An equivalent pair of innate animate system levels is represented at the bottom right of Figure 3.2. At the innate level of complexity are the individual life science systems that contribute to biological structures, e.g., respiratory system, pattern recognition processes, circulatory system, memory, immune system. In their turn these are coalesced to form holistic, constructive systems – skilled individuals or teams, domesticated animals, wild herds, synergistic aggregated organisms – exhibiting 'intentionality' and/or particular intervention properties or skills, e.g. radar operator, nuclear engineer, obstetrics team, dairy herd, energy-generating biomass, in which physical and/or mental associations and conditioning lead to constructive outcomes.

Composites of animate and/or inanimate technical systems lead to self-standing capabilities that possess significant self-sufficiency. Synergistic integrations of heterogeneous constructive

structures result in systems characterised by their self-standing nature. This includes multi-skilled, adaptable human work teams (business departments), manufacturing facilities, military platforms, hospital specialist units, space vehicles, transportation platforms. They are commonly viewed as being products or, when active in their intended operational environment, as services. They are designed and built according to the needs of multiple stakeholders, including sponsor commitments, typically according to predefined scenarios and defined service delivery rules or operating procedures.

The next level of complexity is enterprise systems. These exhibit self-determining cognition or agent status. They are controlled real-time by animate/human intelligence residing in individuals or multifunctional teams according to endeavour or enterprise goals in a dynamic social environment. Enterprise is here not applied restrictively as a synonym for a business operation but rather to describe the characteristic of self-determining endeavour in a competitive environment conducted according to agreed expectations. Generally, such systems are goal-seeking, self-directing, self-organising and capable of learning. Specific examples of enterprise systems are business organisations, government departments, hospitals, strategic military capabilities, wolf packs, space missions.

Members of the enterprise system class are generally comprised of deployed assets and resources, controlled according to some authority structure in order to achieve an intentionality that is directly or perhaps indirectly shared with stakeholders. In the case of business situations, these stakeholders may also be shareholders in the enterprise asset/resource.

At the highest level of system complexity, at the geopolitical scale, come social systems. These combine multiple enterprise systems and thereby make up the social structures that serve the strategic needs, co-existence, survival strategies and communal achievements seen in society. Such systems are co-evolving and adaptive. They have no single supreme directive source and behave according to organic conventions. These systems may find themselves substantially at, or even beyond the limits of self-determination and an ability to compensate for disruptive external and internal influences. They are capable of transitioning into chaotic and even terminal states.

The framework of institutions that comprise the societal structures of this uppermost level of system classification comprise elements that employ democratic, economic and coercive power to control them according to regimes of political, ethical and/or religious intentionality. Examples of social systems are military defence forces, commodity and financial trading systems, social development agencies, humanitarian agencies, national health services, space exploration agencies and business supply chains.

The names in the boxes in Figure 3.2 are intended to convey the general nature of each system class and provide descriptors that apply to concepts developed later. The ascending scale of the classes is associated with two interrelated parameters: complexity, briefly considered in the next section, and its association with autonomy – a relationship long observed and which for nature was encapsulated by Dobzhansky: ‘evolution as a whole doubtless had a general direction, from simple to complex, from dependence on to relative independence of the environment, to greater and greater autonomy’ [Dobzhansky, 1974, p.311].

As with many classification structures, some members unavoidably straddle class divides and their allocation is determined by their dominant characteristics. Each class in Table 3.2 inherits the properties of all its subordinate classes.

Two specific examples of ascending system complexity are conveyed in Figure 3.2. One is military, one civil; both show the span from social structure of national government to a science and technology base. Uppermost, UK political policy and defence chiefs analyse actual and representative international scenarios. These drive the direction of and decisions on UK defence systems acquisition, and lead to primary areas of defence enterprise managed by DECs, such as an airborne command and control system. From this, and in accordance with the anticipated science and technology options, and the capabilities of competing industrial suppliers, IPTs define the major products and services that comprise regions of military enterprise are identified. This leads to largely self-standing elements of capability such as an AWACS aircraft. These in turn contain radars employing antennas design according to microwave field pattern technology. The aircraft are crewed and the radar is operated by trained specialists who blend innate pattern cognition with a body of knowledge and experience.

A corresponding civilian example path leads from the UK Health Service, through a hospital enterprise structure, to maternity facilities that depend upon ultrasound equipment using piezoelectric technology whose results are interpreted by trained, certified obstetrics specialist using foetal development knowledge and image recognition skills.

There are many messages in the class structure in Table 3.2 and they are brought out in later text. Overall, it is held that this model offers better incremental linearity, provides more apposite classes in the face of current systems engineering issues, and conveys a better representation of inanimate and animate contributions to system structure than has so far been offered.

The characteristics of the classes in Table 3.2 are shown to have a bearing on system life cycles in §5, on enterprises in §10.5 and social systems in §10.5. However, in this section it is the distinction and equivalence between the animate and inanimate in Fig 3.2 that is addressed. The focus is on processes that govern the working of life cycles – organic life cycles, termed so because of their continuous, creative cycle of life spans of organisms that respond to shifting environments, and of man-made system life cycles, so called because of a comparable, continual round of man-made systems that deliver a succession of changing interventions into ever transforming circumstances.

There have been important comparisons that draw attention to the equivalence of the mechanism and implementation of animate and man-made systems in the upper four levels of Figure 3.2, [Malthus, 1798], [v.Bertalanffy, 1950, 1952], [Miller, 1978, 1992, p.1-38]. This investigation has a focus of systems engineering processes and this comparison of the creation mechanism for animate and man-made systems is based correspondingly. Acknowledging the almost unavoidable poetry of anthropomorphism, the life cycles of animate and engineered systems have much in common.

Both organic natural systems and man-made systems may, at a high level of abstraction, be seen to comprise a common set of ten operations:

1. a problem/opportunity for an intervention in the flow of actuality presents itself, leading to the inception/conception of a man-made/animate system;
2. founded on engineering principles/biological principles, a controlled method for synthesising a solution to a problem/exploitation of an opportunity is devised/exists, and this is executed;
3. appropriate technologies/organic processes for implementing the system that use technology fabrication techniques/organic reactions are available;
4. based on empirical assessment of the effectiveness of the resulting system properties, the most appropriate inorganic/organic fabrication and assembly sequences are employed;
5. a rigorous and complete sequence of assembly/fabrication instructions is recorded for building the system in the selected fabrication media/medium;
6. the selected fabrication material is secured for processing;
7. the recorded fabrication and assembly sequence is executed;
8. the resulting fully constructed system is subjected to installation conditioning, as appropriate to the operational circumstance, and transitioned into an operational status;
9. the system is operated/operates in its target operational environment according to the problem/opportunity that originally stimulated its creation;
10. information/data that led to well-suited, maintainable/self-sustainable systems is recorded in the pool of information/data on fabrication and assembly instructions, and this is the source of the next design/generation of a system.

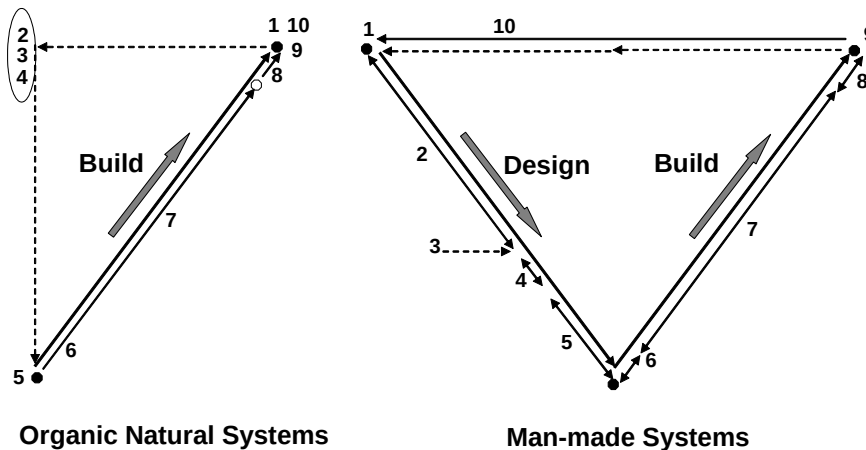


Figure 3.3 Organic System and Man-made System Life Cycles

This flow of ten operations is represented in Figure 3.3 for both organic natural systems and man-made systems. They are stylised according the familiar 'V' representation widely employed in systems engineering.

The man-made system cycle is readily distinguishable from the animate, having three tell-tale differences. Their similarities are nonetheless striking and suggest some common system truths that apply to systems engineering's management of life cycle.

Animate systems have no mechanism for predictive change. There is no model of a current or future operational environment against which to compare alternative solutions. The progress of change is not governed by randomness, or even by directed randomness. It works according to conditioned randomness, fashioned by the positive feedback path of fitness to survive and replicate. Without this strategy the odds of creating biological complexity on earth in the time available are unrealistically too high.

Man-made systems on the other hand depend on prescience to establish and select against goodness criteria. Decisions made according to these guide future action. They require the existence of a projection of the future, i.e. a world model. Even on the occasions when man-made design is advanced by serendipity, recognising its future fitness still needs a world model.

Unlike the animate cycle, changes in man-made systems are the result of discontinuities without intermediate forms; discontinuity far in excess of even the punctuated advancement driven by macromutational saltation (*Dawkins, 1998, p197*) – the nearest equivalent to engineering's 'disruptive technology'. They are the result of the human mind's ability to design using a model building capability that can leap the intermediate. Engineering is about visualising or recognising the unprecedented.

In the animate system cycle there is thus no decision-making or trade-off to influence the outcome; no 'left hand side' design to the 'V'. Consequently, there is no declaration of architecture and no derived integration instructions. The mechanism for recording information, the genetic coding, does not define a set of instructions to assemble parts: a blueprint approach to construction. 'DNA is not a description, in any language, of what the finished [system] ... should look like' [*Dawkins, 2005, p 344*]. Rather, it is a specification of a sequence of fabrication instructions as opposed to a blueprint for assembling. These fabrication instructions do not 'see' a solution or an operational environment. Nature has no predictive power, only memory in the form of past configurations of fabrication instruction: its tried-and-tested, digital (strictly quaternary) heritage.

Through proteins, the genetic code controls complicated fabrication actions that first establish the topographical structure: the management of the system's architecture. Having revealed the coded structural plan, the genes then drive fabrication actions for individual heterogeneous elements. Each element has its individual and characteristic functional contributions in order to complete the implementation of every diverse constituent part of the whole. Relative positions and interconnectivity emerge from the DNA's specification of timing, with later implementation actions being massively concurrent.

To an extent the environment contributes its part by introducing phenotypical manifestations on the fabricated outcome. In addition the 'operational stage' of higher life forms depends on survival issues that arise from self-determinism and autonomy. Operational self-adaptation, effected through cognitive mechanisms to reason, store knowledge and build world models, assists survival and contributes to the viability of the cycle.

Man can, of course, intervene by defining and applying success criteria for the evolving direction, for example for food production, aesthetic merit, domestication – a sort of rapid application development method for natural evolution – but this then elevates matters from the self-standing level in Fig 3.2.

The 'hand' of nature and the hand of man respectively control the cycles in Figure 3.3, and in doing so draw on four management operations essential to any life cycle.

In evolution the stress by Dawkins and others on the management of genetic information, as opposed to species, emphasises a common basis of information in both man-made and organic systems. Though 'evolution is a set sequence of design information steps that result in better solutions to new situations' [Dawkins, 1998, p 192], the 'driver' for the cycle of natural creation is the long term survivability of information itself, albeit with occasional errors, and not on the individual species or family of systems. Information management, its storage medium, replication and dissemination, is paramount to biological systems.

Even though some man-made systems appear to be highly original, like natural organisms they are not created *ex nihilo*. They are augmentations or variations based on accumulated information describing the function and form of all preceding systems, and on empirical information regarding their qualities of performance and behaviour. For animate and man-made systems alike, information management is a fundamental enabler of creation. In the man-made system cycle, as Appendix A conveys, the vast majority of process work products are information/data items.

The outcome of information management is that it 'generates, collects, transforms, retains, retrieves, disseminates and disposes of information' [ISO, 2002, 5.4.8.2]. Information is the means of communicating between processes and, for man-made systems, the minds of an integrated team of creators. By assembling a genetic code or by documenting a design, animate and man-made systems depend on information management and on the 'semantics, formats and medium for the representation, retention, transmission and retrieval of information' [ibid, 5.4.8.3 d].

The information content that is being managed is the result of unique paths through a labyrinth of alternative opportunities. Nature randomly selects all its options according to incremental change rules. Man follows a reasoned selection strategy; continuously comparing progress with intended outcome, until all options are resolved in terms of a system's complete fabrication and assembly sequence. Irrespective of how opportunities are selected, the underlying scheme is one of cumulative decision-making that achieves a 'balance of consequences of alternative actions ... to arrive at an optimization of, or an improvement' [ibid, 5.4.5.3 e]. Its outcome is to 'select the most beneficial course of ... action where alternatives exist ... to reach specified, desirable or optimized outcomes' [ibid, 5.4.5.2].

Whether by the hand of nature or man, the configuration of each individual system produced is managed through information baselines with rigorously controlled change or variation strategy. These baselines are communicated by high-reliability information replication that then informs reliable, repeatable fabrication and assembly sequences. Checking or auditing followed as appropriate by rejection may form part of this configuration control strategy. The result is a well-controlled product build. The essence of this configuration management is that 'configuration baselines are established ... [and] changes to items under configuration management are controlled' [ibid, 5.4.7.2]. Following this, Successor sequences of instructions configuration manage steps of incremental advancement that amount to a series of related product 'family' changes.

To 'reduce the effects of uncertain events that may result in changes to quality ... [and] characteristics' [ibid, 5.4.6.2] of the resulting system, risk management of some form is complied with. Both animate and man-made creation cycles each observe a risk management strategy, but they are based on radically different logic. Being blind to the future, nature counters uncertain events with uncertain events by intentionally varying the information for each

successive configuration. There are no continuous product runs, rather a batch-of-one strategy is employed, leading to close similarity in individual units and high interoperability in order to derive group benefit (as in some military product families). In the case of man-made systems, unintended errors, changes or potential failings in the design definition, build operations and service environment are postulated and actions to minimise or counter their effects, should they arise, are planned for. It is a reactive strategy predicated on conjectured failure or change, and timely detection and rectification. For its part, nature has 'planned for' and discards failed or unfitted systems, replacing them with a continual flow of potential upgrades.

A crucial message from Figure 3.3 is that these four processes – information management, configuration management, decision making and risk management – are fundamental operations in the management of animate and inanimate system life cycles. Together with the ubiquitous project management trio of planning, assessment and control (plan, check and act), they are the team-managed processes seen in systems engineering's codification in §4.

3.4.3 The Science of Order

The understanding of what a system is 'about' and why the ordering of reality or cognitive entities should be so important has been widely covered in literature. As described above, philosophers and scientists have addressed these issues repeatedly. With such a rich set of interpretations, the lexicographer is faced with an excess of options when defining what a system is.

Dictionary definitions abound, yet Kline observed of eighteen meanings for 'system' listed in *Webster's Unabridged Dictionary* [Webster, 1995] and thirteen in the impending First Edition of *Random House Unabridged Dictionary* [RH, 1996], that none of these meanings satisfactorily corresponded to the concept of system used as a basis for engineering and science [Kline, 1995].

Collins manages 'a group or combination of interrelated, interdependent, or interacting elements forming a collective entity' and 'any assembly of electronic, electrical, or mechanical components with interdependent functions, usually forming a self-contained unit' [Collins, 1991, system: meanings 1,10].

The OED does rather better with 'an organized or connected group of objects. A set or assemblage of things connected, associated, or interdependent, so as to form a complex unity; a whole composed of parts in orderly arrangement according to some scheme or plan; rarely applied to a simple or small assemblage of things... A group, set, or aggregate of things, natural or artificial, forming a connected or complex whole ...artificial objects or appliances arranged or organized for some special purpose ... An organized scheme or plan of action, esp. one of a complex or comprehensive kind; an orderly or regular method of procedure ... A formal, definite, or established scheme or method (of classification, notation, or the like) ... In the abstract (without a or pl.): Orderly arrangement or method; systematic form or order [OED, 2004, system, meanings 1,4,9,10].

The theme is of order and unity juxtaposed with complexity and division; of homogeneity related to heterogeneity; of comprehensibility coupled with abstruseness. Fundamentally, it is about the whole/part relationship and its significance.

Scientifically a system is viewed to be a holistic zone of hypothesis or reality that is identified by a limiting, closed boundary. This has a composition described in terms of elements, order and interactions that give rise to properties that identifiably belong to the whole. In this regard:

- holistic refers to a continuous region of space/time, that is viewed as a single entity identifiable by properties manifest at its boundary, and is identified generically as the system-of-interest or specifically by a meaningful descriptor;
- closed boundary refers to the terminating surface that limits the region of consideration from the space/time continuum that it exists without, i.e. its environment, and across which flow interactions between the system and its environment,
- elements refer to the complete set of discrete subordinate entities that comprise the whole, each having a different homogeneous nature and identity relative to all other members of the set;

- order refers to the arrangements of elements, their functioning and their relationships and their precedence in a hierarchy of consideration;
- interaction refers to all the mutual influences that each element has with all other elements;
- properties refer to all qualities that emerge at the level of the whole in all degrees of freedom as a result of the combinatorial effect of each individual entity, one on another.

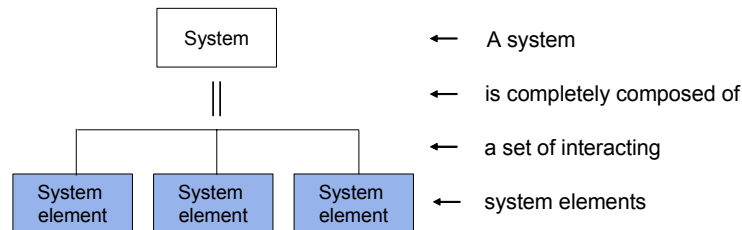


Figure 3.4 System and system element relationship

The notation used in this investigation, Figure 3.4 [contributed to *ISO, 2002* as Figure D.2], attempts to avoid weakness in past portrayals of the system concept. Boxes signify bounded objects, white for a holistic system, grey for a system element. Single lines convey a group of objects that mutually interact, and || means 'is completely composed of'. Each of the two separate rows of boxes in Figure 3.4 therefore portrays exactly the same entity expressed at a different level of structural resolution or granularity, the upper level being the holistic system-of-interest and the lower a set (one of many possible sets) of system elements. This notation avoids misleading connectedness between rows implied in hierarchical compositional trees, such as in IEEE 1220 [IEEE, 1998] and IEC 632 [EIA, 1998], as seen in Figure 4.5.

A man-made system, whether viewed in isolation as a product or in some environment operating to deliver a service, exhibits particular characteristics and many of these are conveyed in Figure 3.4 [contributed to *ISO, 2002*, after Singh, R.].

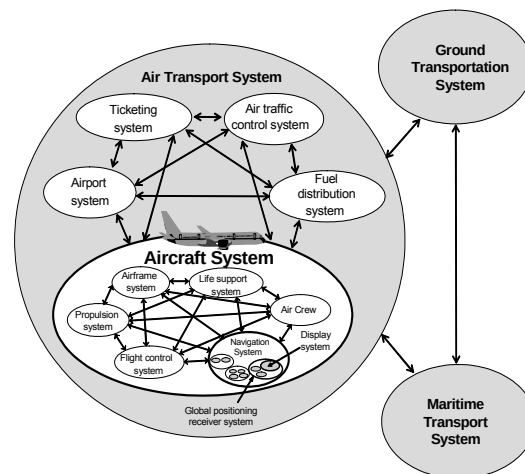


Figure 3.5 Typical system view of an aircraft in its environment of use

This exemplifies the multitude of perceivable systems seen from different structural levels or granularity that can be associated with an aircraft set in its environment of operation and service provision. It illustrates:

- the environmental situation that gives rise to a perceived need for a man-made system to fulfil some beneficial change in the order of reality – a teleological motivation arising from stakeholders, in this case people seeking a means of transportation;
- the importance of a closed boundary definition that establish a holistic manifestation and equates to some ontological identity;

- boundaries that encapsulate meaningful need and practicable solutions, and establish the extent of the system-of-interest (in Figure 3.4, the aircraft), its composition and its interface with notional systems the surrounding environment;
- the perception of hierarchy that arises from recursive consideration of object definition and their relationships, irrespective of the span or detail of concern;
- that a system can be considered in isolation as an entity, i.e. a product (in Figure 3.4 the aircraft);
- that a system comprises a fully integrated, defined set of subordinate systems or system elements (which in the case of the aircraft in Fig 3.4 has an exemplary representation of six elements and in which the navigation system is illustrated to comprise its subordinate structure, terminating in its selected elements)
- that a system can be considered to be an ordered collection of functions and properties capable of interacting real time with its surrounding environment, i.e. a set of services (in Fig 3.4 the air journey that transports.
- a system's concomitant product and service existence;
- that humans can be viewed as both users external to a system-of-interest (in Fig 3.4 this would be passengers and thus a particular class of stakeholder) and as system elements within a system-of interest acting in the role of operators, e.g. air crew.

In such composition, what is it that is actually ordered? Thermodynamic entropy, as a spread of energy, was propounded by Clausius [*Clausius, 1864*) and thirteen years later, through the concept of mixing, interpreted by Boltzman to be a measure of energy disorder in any region of the universe [*Boltzmann, 1877, p.188-93*]. The often quoted Second Laws of Thermodynamics states that, without intervention, entropy always increases.

The Second Law is not however a law of unidirectional decrease of order with time towards irretrievable (though not total, according the Third Law) disorder. Even though the Second Law is conserved and overall disorder increases, in isolated locations or regions order can increase. With the microscopic attention to space and time paid by humans, such regions are commonplace in our world, and any man-made system exemplifies this concept. In subordinate, closed boundary regions of universal continuance, man-made and also natural order can increase – at the expense of other regions of the cosmic whole. Without this expedient neither man's design nor nature's ordering of systems could occur.

Order is thus detectable as islands in a sea of entropy; as an arrangement that is not likely to have occurred by chance; as the presence of patterns and regions of heterogeneity set in comparative homogeneity. Such order is a key characteristic of the notion of system. Systems, with their order, and entropy, with its disorder, face in opposition. Disorder eventually wins in a decaying cosmic condition.

A system is thus a state of energy and matter with distinguishable arrangement. A human's conceptualisation of a system even conforms to this axiom, as neural patterns build a picture of a present or future ordered reality, see §3.5.2.

The reasoning mind is tuned to defined regions and to degrees of ordering within them. In nature, just as with man-made systems, increased order leads to novelty; new properties emerge. Order establishes associations and, within a bounded region and compliant with the laws of physics, this generates properties not detectable in any subordinate region; in systems engineering vernacular, properties emerge from a less ordered complexity that is intrinsically unable to exhibit these phenomena.

Human awareness is predisposed to impute intention to increased order and in turn to the concept of design: coherent structuring to achieve a goal, purpose or fitness. In man-made systems this is manifest, for natural systems it is philosophical or theological, for biological systems it may 'simply' serve information 'immortality'.

Irrespective of its formative agent, order is a fundamental notion in reasoning about systems. For man-made systems it is about understanding how, when and why systems work as capsules of reality that appear to buck the trend of the Second Law.

Thermodynamic entropy, however, cannot be equated to physical order (or rather disorder) in a system in isolation of its energy state. Order 'hidden' in material and energy is interpretable as

information. When the transmission of order has the purpose of being a messenger, and the coding of intention in that ordering is common to both creator (source/transmitter) and a user (receiver) able to correspondingly decode it, then information flows.

Through meaningful flows of order that tunnel through background 'entropy', actual or imputed intention present in one system can inform and control another system provided that it shares symmetric conventions for constructing and deconstructing that order. Though it is material and/or energy that is actually exchanged and transformed in such encounters between systems, intentionality is ciphered in material and energy interaction. The exchange is interpreted as one of information transfer. Information is thus manifest in the physical world – a point made by Landauer, [Landauer, 1991], and reviewed by Bawden, [Bawden, 2007].

From the mid C20th, this sense of structured order or 'logical entropy' has come to be used in the fields of information science and communications theory, which depend on the physical 'implementation' of information through, for example, the implementation technology medium of electronics and the functional logic of software engineering.

This gave an information meaning to entropy, (apocryphally?) attributed to Von Neumann addressing Shannon – both Macy attendees [Tribus, 1971, p. 179-188]. Viewed in this way neither Shannon nor subsequently Feynman assigned physical units to entropy, only value and a logarithmic form, and they use no equation to describe it. Its 'measurement is relative to the coordinate system' [Shannon, 1948, pp. 379-423, 623-656] [Feynman, 1964].

Thus across the boundaries of a system the flow is not simply of matter and energy but also of relative order, and therefore of information relative to the background environment. This may be interpreted as complex instructions, operations or transformations that construct functional or physical order, or as the logic of controlling signals that synchronise these operations. Either way, they are information encoded in the system input and output flows. To reflect this and to accommodate the interpretation of a system as an information processing complex, the simple functional representation (IDEF) of a system's input/output dynamics shown in Figure 4.1 also models information transformation.

A particularly important example of inter-system information flow, and specifically of design messages within it, is that of one human neural system communicating with another and this is explored further in architecture descriptions of §10.4. To prepare for this, and also for the process model of systems engineering in §4, the 'systems psychology' component of systems reasoning is considered in the next section. It hypothesises about the cognitive mechanisms and intrinsic coding that act to model systems in reality, or that could exist in reality, and that fashion the mind's attribution of intentionality.

3.5 Systems Reasoning and Psychology

3.5.1 Dealing with Complexity

Do systems exist in actuality?

Philosophers would probably tell they no more exist than any other perception of the real world. It is beyond our knowing.

Scientists would be inclined to say yes; information theory and thermodynamics point to the existence of order and to systematic characteristics in the universe. In addition, the knowledge accumulation and predictive power of mathematics and physics are partially predicated on analytic and holistic system principles.

Psychologist, however, may respond with "does it matter"? For them it is how the subject perceives, interfaces and deals with reality that is of ultimately practical importance, not what absolute truth lies on the other side of man's frontier with actuality. Humanity could (and may very well) operate using a 'black box' of too-complicated reality, around which a homespun cognitive model is wrapped. Ultimately, individuals either succeed or fail, or in other ways intervene in the world, by following a particular verisimilitude that just happens to have sufficient day-to-day effectiveness and success. Others might have evolved. They may have in worlds elsewhere, or in other life forms in this one.

This section addresses the third component of systems reasoning's postulated trio: 'systems psychology'. It considers the psychological stratagems and psychological devices by which the human mind deals with reality and with its perception of systems in it. It is the psychological machinery of systems reasoning, and it is why systems engineering is instinctively, if not always objectively, approached as it is. With some emphasis or other, it appears to be common to all humans and probably all human pursuits and disciplines.

Psychology's contribution to systems engineering thus lies in explanatory generalisations of mental functions that underpin awareness of and deductions about systems. An appreciation of them helps explain individual functions and group behaviour that stem from cognitive mechanisms, including the formalisms of team communication about systems. Despite some materialist scepticism regarding the possibility of ever successfully describing the underlying structure, mechanism and dynamics of the mind, some psychological hypotheses do deliver credible interpretations of why and how it works as it apparently does. Whether they are actually founded in scientific realism or in instrumentalism, they nevertheless enable tenable, psychology-based rationalisations. The cognitive models advanced here, and their development in §10.4, rest on this syllogism.

Systems psychology is offered here as a term to embrace the evolved and refined solution that counters the neurophysiologic limitations of human intellect and mental capacity that handle complex real world situations over a range of scales. A 30W, 1Khz neural system with a network of around 10^{11} functional elements and 10^{15} interconnections, powerful as it is, is seriously limited in its capability and capacity to model the real world as an extended set of systems. It is a meagre match for the real world's complexity.

It is argued that an armoury of systems reasoning mechanisms are the evolved biological response to the existence of systemic order in the real world. Refined mental models adequately facilitate the resolution of real world problems and tasks, such as anticipation of danger, forecasting of food availability, location of water, reproduction opportunities, interaction with social structure, and ultimately creation of tools, and in so doing preferentially favour continuance of the individual and the group [Dawkins, 2005]. The resulting pre-coded neurological network patterns that have evolved allow effective prediction and favoured survival of any species with this capability. In addition, this 'Darwinian-driven' solution to handling complexity can be reinforced by experiential and educational conditioning, i.e., supplementary coding according to communal knowledge bases, training and mentorship.

Conceiving of and manipulating objects as systems according to particular cognitive schema is thus a necessary and instinctive stratagem to accommodate the limitations of the mind. Without this, the task of engineering novelty would be overwhelming for the individual or even, assuming effective communication, the integrated team. Although some favoured individuals are patently equipped and predisposed better than others, system-oriented cognitive mechanisms are a natural way by which any human mind contends with the complexity of reality, understands its significance, decides how to react and then intervenes in it.

Though there are mathematical and scientific qualities to complexity [§3.4.3], there is also a subjective dimension. Cognitively, complexity is perceived as non-intuitive, non-cumulative, non-extrapolatable and non-linear aggregation of properties, characteristics or behaviours. An instinctive assessment of the nature of this complexity in specific situations is called into play to gauge which elements of the cognitive armoury should be employed to undertake particular modelling and decision-making tasks, and when, or in what order or pattern to use them. Each has as its target a contribution to the task of resolving an overwhelming level of information. Each has the goal of either reducing the magnitude of data to a manageable level or précising only the salient relationships for decision-making purposes.

The rationale for familiar and instinctive strategic approaches and tactical manoeuvres of engineering design derive from this pre-programmed mental machinery of systems reasoning, and these are considered further in §10.4. In preparation, the foundational working of this machinery is hypothesised according to five dominant mechanisms: boundary building, viewpoint selection, scaling, abstraction and pattern recognition. Each contributes in a different way to selectively reducing the information present in the real world in order to arrive at a workable reconciliation with the limitations of the mind.

Boundaries are the system reasoner's primary weapon. A boundary is a termination of the extent of consideration; it is not in any sense a presumed termination in the space/time manifold

of actuality. It is a limitation applied by the observer's interests or concerns in the real world regarding the objects it is conceived to be composed of, and the interactions that exist between each and every other of these object. It is a conceptual protection barrier from the overwhelming complexity of reality.

For the systems reasoner, it is the potency of the closed boundary that has greatest benefit. It is through closed boundaries that entities or objects are known, delineated from all else. A real-world object exists as a bounded region of consideration that has material existence in space and time, and by that existence engenders causality on all else.

A closed boundary is the mind's means by which the complexity of that 'all else' may be considered not to exist. It is an isolating mechanism: a surface circumscribing all internal causes and effects of an object's composition. Conversely, it screens out all causes and effect from the complexity of the almost illimitable surrounding of mutual interactions and influences.

Such closed boundaries are mental constructs; in the real-world the closed system is untenable [Flood, 1993]. Based on classical science and the second law of thermodynamics, a closed, bounded region of space and time is a fallacy. Reality's continuum of interactions and mutual co-existence cannot be suspended, only considered as if it were not present for the purposes of understanding and conjecture. At every level of classical real-world consideration, everything is connected to everything else no matter how tenuously; fundamental particles to fundamental particles, galaxies to galaxies.

Nevertheless, hypothetical as they may be, closed boundaries are a formidable and successful systems reasoning artifice; in practice they work – contingent on the ever-present cognisance that they are a modelling device and therefore a simplification of reality that may harbour pitfalls. In engineering a closed boundary should always be recognised as being an artefact of supposition and convenience, and thus be subjected to continuous review to ensure that no significant interior or exterior objects and interactions have been excluded, and no interactions of significance across its surface have been omitted.

The success of such boundaries arises because, in most situations, order or entropy in the universe can be mapped effectively to an arrangement of closed boundaries. Notwithstanding the 'Second Law', there is a natural, if localised in space and time, order to the matter and energy of the cosmos (hence its name). That is, boundary conditions can be specified to a workable level of descriptive trustworthiness. These conditions permit the construction of a mental model of what exists, or (even without the benefit of immediate sensory perception) what is conjectured could exist.

Such closed boundary definitions should lead to objects that are maximally cohesive (assessed in terms of how effectively they denote a single abstraction), i.e., convey at some level of detail the appearance of homogeneity. They should also be chosen to minimise interactions between the objects so defined. The existence of highly ordered or well-understood interactions, or of weak material or energy exchanges, expedites effective boundary selection. However, the rationale for choosing what is within and what is outside each closed boundary is a key skills of systems reasoning. In engineering, just as in society, the adage "bad boundaries make bad government" is an apt metaphor.

A closed boundary delineates an object but does not define a system. It is the expedient of re-applying the closed boundary convention recursively within a chosen closed boundary, in such a manner that their surfaces do not intersect and are contiguous one with another or with the chosen boundary that defines a system. A complete, contiguous set of non-intersecting, subordinate boundaries now exists. Applied recursively by an observer, this operation delineates a set of selected elements and relationships that fully comprise a particular object at a particular level of granularity. Many useful, different and complete sets of elements may be defined to occupy the original, encapsulating boundary. Each is a separate instance of a system characterised by a different architecture.

Used recursively, the closed boundary becomes the mind's mechanism of hierarchical reduction. It bounds 'problem' spaces into meaningful regions described and makes detailed understanding or control of the whole more tractable.

The outermost closed boundary of a man-made system thus equates to an observer's system-of-interest. Looking inward is a product comprised of system elements whose holistic properties

are manifest at the inside of the boundary, to the right of Figure 3.6. Considered in isolation, it is an island of relative space and relative time. Looking outward from the boundary those same properties are open to influence reality and can be interpreted as an intentional set of interventions that change the course of actuality according to some intended purpose. Thus the system-of-interest's services (and its behaviour) only have meaning when it is set at an absolute point in a contextual expanse of the space/time manifold.

Service is used throughout this text to mean a man-made intervention of a system-of-interest in the timeline of space/time actuality that results in a desired change to the course that it would have otherwise run; shown to the left of Figure 3.6. It may be described in terms of an outcome or end result. There is however no end state, rather a desired perturbation or shift at some temporal and spatial point due to the presence of an 'intentional' object. In consumer and in military terms, services are typically contingent on an environmental state, i.e. circumstances that trigger or sustain an intervention, as in a military mission or the provision of a public utility.

Systems engineering therefore approaches a system-of-interest according to two cardinal boundary viewpoints: the system as a product and the system as a service, see Figure 3.6. One is a viewpoint at the outside surface of the system boundary, the other from the inner surface. They are both essential during the system life cycle, and the distinctive resolution and definitions of system views, described according to these two separate viewpoints (by work products of systems engineering processes), is an important advance in the ISO model of systems engineering [ISO, 2002].

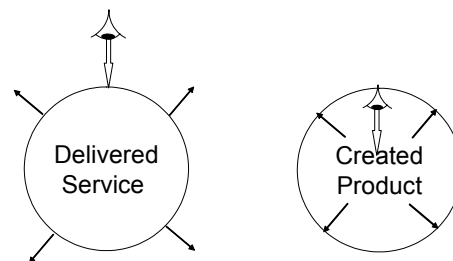


Figure 3.6 The Product and Service Viewpoints at a System's Boundary

Each subordinate closed boundary defines a new object: a new module to be considered in a relative isolation. The mind invokes a principle of modularity, assuming that details of one part of the system may be designed independently of details in other parts. This eminently powerful strategy, however, introduces an inherent danger: local optimisation at the expense of holistic optimisation. Continually switching between considerations with the boundary existing and not existing ensures that decisions at one level of detail are not to the detriment of those above. Living only within the comfort zone of one level of structural detail, and within that one region of consideration, is not in part of the repertoire of the effective system reasoner. A propensity and ability to give similar consideration to multiple levels of structural detail is a systems engineering skill that merits development.

In a comparable way, the consideration or optimisation of any one element in a domain such as in Figure 3.4 to the disadvantage to an other may miss the balance of effectiveness, resilience and adaptability across a region of consideration.

The message is that switching bounded considerations or concerns, and developing a discipline regarding when and why this is done, is a key technique of systems reasoning and an important part of design strategy.

If boundaries are the systems reasoning scheme for limiting information content by constructing limits that circumscribe an observer's interest in an object (as opposed to any other object), then viewpoints are the scheme for limiting information according to particular concerns about the object-of-interest. It is a filtering or masking mechanism that offers views of the bounded whole according to different and separable concerns. In this way only that information relevant to an issue of concern is made visible.

This partial representation of an object is at one with models of objects for, in the multidimensional representation of every aspect of their reality, each model that is employed to

convey this is limited by its modelling notations and conventions. Each aspect of an object's reality is constrained to being a facet of the total information it contains.

Observer's viewpoints are thus an expedient and a necessity. The mind cannot handle the totality of information in an object, the models of whatever form that describe it cannot represent its full reality. The nature of modelling – language, images, mathematics, mock-ups – and the nature of the information sought – function, form, parametrics, logic – are locked in mutual dependence.

In their partial representation of reality, a web of views is woven around an object to build an evermore comprehensive understanding that is assembled from manageable increments. Each facet is built separately and, by establishing their mutual correspondence, the partial images are locked together to create an adequately faithful description the whole. Adequately faithful depends on observer concerns, modelling fidelity and a judgement on the gaps of knowledge and the degree of inconsistency between views.

Each observer has a set of concerns about an object – threat, benefit, performance – and this set may apply across different objects, whether real systems inter-operating in concert or prospective future objects each offering their respective merits. Either way, well-conditioned viewpoints bring discipline to the decision making of prediction or analysis.

The systems reasoning stratagem of abstraction is a sophisticated information reducing mechanism. It is 'the process of formulating generalised ideas or concepts by extracting common qualities' [Collins, 1991, meaning 2] from a specific object of consideration. In engineering terms, its interpretation is consistent with this but more specific: a 'simplified description, or specification, of a system that emphasizes some of the system's details or properties while suppressing others' [Shaw, 1984, p.10]. Abstraction thus attends only to what is relevant for gaining understanding, and this clearly benefits from skill and experience to discern what is relevant from the irrelevant according to a given concern.

Abstraction serves to reduce the information content of observable or conceived phenomena by retaining only information that is essential for a particular purpose of analysis or synthesis. It is an approach to information reduction in which regions of interest are not masked, as in boundary encapsulation or rendered according to perspectives, but are mentally filtered out according to relevance.

Abstraction takes the whole and represents it at a particular level of significance. It has the effect of reducing heterogeneity by reducing spurious detail and drawing out characteristic similarity. In consequence the descriptive content of a whole region or object is reduced, as is its corresponding detail of interactions at its boundary.

'We employ abstractions daily and tend to develop models of reality by identifying the objects and operations that exist at each level of interaction' [Booch G. 1987, p.11]. Once an understanding is gained at some level of compositional detail, previously irrelevant detail that was suppressed or unexplored may then be recognised as relevant. This triggers deeper exploration and more detailed understanding. In this way successive, nested levels of abstraction tackle manageable steps of information reduction to reveal progressively more detailed understanding.

Conducted along with encapsulation of neighbouring regions, e.g., other system elements, exploration down detailed paths of enquiry can drill into detail that, for a particular purpose, exerts disproportionately high significance to the whole, i.e., decompositional paths leading to critical areas requiring particular attention. This is a risk reducing strategy that only terminates when information and understanding pass cross a threshold of tolerability, and deeper understanding becomes immaterial to a project's success, see Figure 4.17.

By building of effective boundaries in conjunction with different scales of observation the mind can analytically expose or holistically compose new order at different levels of abstraction. It resolves homogeneous composition and interconnectivity from comparative heterogeneity, and thereby resolves complexity down to elementary technological sources.

One level of abstraction dissolves into another as different patterns are resolved and then disappear through increased or reduced detail. Patterns are cognition's access mechanism to previously-encountered, useful-for-prediction relationships or interconnectedness. From their ability to reveal in one situation, they are stored as a reference library of spatial or/and temporal

structures built according to the way the mind interprets order, cause and purpose in reality. 'Seeing' a correspondence between a distinctive arrangement and useful behaviour in a particular settings allows repeated use of the arrangement, or allows a transposition of its essence into novel situations bearing the same underlying ordered relationship. e.g., an equivalence between patterns of structure and behaviour sequence in animate and in man-made systems.

Particular scaling of information content presents the mental faculties with a limited number of separate objects to represent, a limited number of relationships to build and a tractable amount of properties to consider. Switch scale of the totality, and with it resolution, and a new regime of order occurs (in both senses of 'order' – relationships and size). New information is revealed, e.g. new patterns and new properties, and the net information content to reason about is maintained at a tractable level. Rescale and re-scope by defining new boundaries, and the mind can again masters the problem solving. Overall Information content and the scale of consideration are tradable phenomena.

Explicit or observer-implicit classification and description schemes, based on patterns, such as sequences, stars, trees and highways, confer *a priori* knowledge. Known holistic characteristics are associated with particular patterns and this again speeds the identification of patterns. Transposed from analysis to synthesis, this classification structure suggests architectural forms that are likely to confer particular system qualities. Such architecture patterns have the benefit of being understood and proven, they may be re-applied and they can express complex solutions succinctly.

Subject to the rules of its various interacting forces, nature operates simultaneously on all scales: subatomic particles to subatomic particle; galaxy to galaxy. The capability to apply scale to observations of objects is essential to offset the limitations of mental processing in order to understand reality. The field of view and the granularity of detail are traded in a dynamic visualisation of reality. Such cognitive scaling provides systems reasoning with two related strategies.

Applying a reductionist strategy, the extent of the boundary of consideration drives a change in the level of abstraction. The boundary of consideration is employed as an independent variable; the dependent variable becomes the level of detail. Using a mental quid pro quo, it is possible to keep the information content maintained at roughly constant levels whatever of the extent of the systems boundary being observed.

These independent and dependent roles can be reversed. The level of detail is adjusted in order to detect properties not discernable at other levels of detail and the scope of consideration is reciprocally adjusted accordingly to maintain a tractable information level.

Either way, this enables phenomena and patterns that only manifest themselves on some characteristic scale of observation to be detected. Moving up in scale, new properties emerge replacing those recognisable in finer detail. Moving to finer scale, the reduced boundaries of consideration cause higher order properties to be dismantled to their more basic contributing phenomena. Scaling depends on coordinated boundary redefinition and steps of abstraction. Thus scaling is in practice not a continuously adjustable parameter of observation, rather it is switched in increments according to the resolution of new patterns of information.

The perception of patterns is an association mechanism. Salient and repeated orderliness is abstracted from a labyrinth of detail by selecting repeated forms of connectedness. The mind is naturally attuned to resolving and remembering significant order, and then identifying homological relationships in disparate situations. In essence, an abstraction that removes 'noise' to reveal order leads to the discernment of spatial and temporal patterns – and the information or meaning signified by that ordered structure and/or events. It learns to recognize common, underlying themes or features that recur. They recur because they have value according to purpose or intent. Order with value in one successful situations is remembered and when matched infers significance and merit in another.

These powerful cognitive strategies for designing systems in a manner that relates to our reasoning about real world objects are traditional and almost subliminal in systems engineering. Being natural they have not always been well-disciplined and uniformly practiced, particularly in a coherent way across organisations and project teams. Their 'discovery' by software engineering in 60's programming was eventually accorded eminence as the 'object-oriented'

analysis/design approach by the 80's, and with this came some powerful methods and supporting tools that applied beyond software implementation technology to systems reasoning generally. The methods that were (sometimes haphazardly) empirically developed and defined trace to the rationality of systems reasoning, and they have helped to impose some external methodological disciplining and structuring influences on the instinctive mechanisms of cognition that govern design analysis and synthesis. They help shift systems engineering from the mysteries of art form towards disciplined practice.

Whereas much of the foregoing object thinking about the real and conceptual world is ingrained in humans, its formalisation in most disciplines has not been well developed. Its adoption and codification by software engineering, and its refinement of methods and tools under the 'object-oriented' banner, is beginning to transfer into other disciplines.

Ingrained systems reasoning has been considerably strengthened and disciplined by software engineering's methodologies and automated environments, and has considerably strengthened systems engineering team cooperation. Raising it out of the subconscious, and speculating on what cerebrally drives matters, opens up possibilities for improved training and assessment of practitioners, as reflected on in §6.4.2 and system design methods as seen in §10.4.2. This depends on understanding some of the underlying cognition. In the next section, systems psychology begins to offer a window into the systems engineering mind.

3.5.2 Perceiving Reality

The foregoing advances that systems psychology entails a repertoire of complexity-reducing, innate mental strategies that all individuals possess and that are employed to understand the real world. This accords with the proposition that Dennett and others have advanced: our minds have evolved to work so as to best fit us for reality and that it is genetically pre-wired to provide an enduring systems-oriented reasoning mechanism (capable of 'in-service' development).

Dennett goes beyond this proposition as he attempts to build the psychological bridge between predictive cognition (a necessity of design) and physical objects in the real world. He posits that intentionality profoundly governs the way we think about the real world and how we assemble models that permit us to predict the course of actuality. 'We are biologically programmed to impute intentions to entities whose behaviour matters to us' [Dennett, 1987]. Humans' teleological predisposition, he advances, is an outcome of evolutionary advantage. In other words, natural selection has given man the supreme advantage to predict with a requisite level of success in the purpose-/outcome-/effects-based interventions of objects in the real world. These 'objects' may be animate entities, such as other humans and animals, or they may be inanimate entities, such as elements of the natural environment, or they may be engineered artefacts, or indeed any combinations of these. The same set of cognitive mechanisms – modelling conventions, their mapping and mutual resolution – deals with them all.

Faced with threat or competitive opportunity circumstances, this innate mental machinery has advanced the individual or group fitness to perpetuate. Its selected benefit, Dennett argues, has furnished individuals with mental constructs for survival, which have evolved to be a basis for refined social and intellectual conduct, extending beyond the phenotype to the regulation of scientific and cultural conduct [Dennett, 1995].

Based on arguments of evolutionary psychology, Dennett posits there are three different model conventions that we use when confronted with objects or systems. He terms them stances; his term is followed in this text. We all use stances individually and in common to explain and predict the behaviour of an object in the course of its interaction with its environment.

Stances may not yet be widely accepted as axiomatic but they have definite instrumentalist merit in different fields and are being increasingly cited, e.g., [Dawkins, 1996], [Ross, 2000], [Brook, 2002], [Mindpapers, 2007]. In the context of the systems engineering propositions that are advanced in §4 and §10 they bring forcible relevance. Dennett's innate cognitive modelling constructs appear to map well into actual design strategies for engineered systems, and specifically into the application and management of system models, now often referred to as architecture descriptions.

Dennett advances that, in order to provide the fastest interpretation of threats and of opportunities, we hold beliefs about real world objects in three distinctly different ways:

- as a physical system with material characteristics, operating according to the laws of physics, e.g. as a physical structure
- as a designed system consisting of parts with specific functions that interact to produce certain characteristic functions, e.g. as a functional design;
- as an intentional system acting rationally relative to a certain set of beliefs and goals, e.g. as a real world intervention according to intended effects.

‘Stances’ are quite distinct from viewpoints and may be understood to be ‘attitudes’ towards a system, using attitude in its foremost sense of ‘the way a person views something or tends to behave towards it, often in an evaluative way’ [*Collins, 1991*, attitude, meaning 1]. The evaluative quality common to each stance is intentionality and this translates into three cognitive models that underpin systems reasoning.

Intentionality was seen by Brentano as the property of the mind by which it is directed at, about or ‘of’ objects and events in the world. It is a characteristic of mental phenomena, and may be précised as the mechanism that configures models of the relationship between mental acts and the external world. Dennett argues that the intentionality expressed in these models is the mind’s supreme predictive strategy [*Dennett, 1996*, p191]: the essence of systems engineering’s dimension of prescience. Accordingly, stances can be interpreted as the mind’s primitive modelling elements of actual objects. They are the basic building blocks of our vision of reality. They are inherent in our system model building and in our externalisation of those models in teaming and social situations.

Clearly, when dealing with objects having awareness, particularly other humans, an observer can impute intention to a subject in particular circumstances. By this means a subject’s behaviour can be anticipated. With regard to non-animate objects, intentionality is facilitated in a second-hand manner, i.e., a system is intentional only in relation to someone who is trying to explain and predict its behaviour. Humans, or more specifically their minds, are in either of these cases the ultimate intentional systems and man-made systems become an agent for this intention.

Dennett’s terms for each stance come from the language of psychology and evolutionary biology. Had they been coined by the systems engineering community they might have been termed the ‘physical form stance’, the ‘functional design stance’ and the ‘service effect stance’.

The physical stance is the most reliable of these cognitive models since ultimately all real world entities comply with the laws of physics. Within the limitations of knowledge, reasoning can be developed from basic principles, through knowledge of particular implementation media and their associated technology characteristics, to establish the physical stance. It is thus a science and technology, knowledge-based modelling approach; a bottom-up, first principles strategy that is likely to require considerable and lengthy reasoning from basics to understand emergent properties and their opportunity to deliver effect.

Physical stance predictions are based on the actual physical state and constitution of the particular system and are deduced by applying the observer’s extent of physical knowledge. Repeated or similar situations, e.g., a sound experience of one or more technologies, offer the benefit of memorable examples of this kind of model. This is a recognisable contributor to systems engineering competence.

In contrast to this, the design stance employs inferential models of function that are an intermediate link between physics and the intended (or imputed) intervention of an object in real-world order. The essential feature of the design stance is that predictions are made solely from knowledge or assumptions about the system’s functional design, independent of its physical constitution.

Developing the design stance, it is assumed that the entity of observation has been ‘logically designed’ in a directed manner, and it is thus predicted to behave in accordance with a conjectured design. Dennett points out one fact so obvious that it is easily overlooked: our intuitive explanations of behaviour and intention in both natural and man-made systems’ design are founded in an assumption of rationality or logical function. Predictions therefore naturally depend on suppositions of how an object rationally functions. This deduction of rationality benefits from experience, a body of precedence and a skill to interpret. The ability to construct

this stance thus benefits from observations of diverse systems; a breadth or diversity of experience counts more heavily than duration.

The design stance's short-cut, predictive strategy is based on causal deduction about probable intent. It follows a 'middle-out' strategy in which skilled reasoning about function is dominant. When considered in the context of a developed functional appreciation of surrounding reality, it can prognosticate events and change. As Dennett puts it; 'design-stance prediction, when applicable, is a low-cost, low-risk shortcut, enabling me to finesse the tedious application of my limited knowledge of physics' [Dennett 1996]. Nonetheless, though the design stance considerably speeds prediction over the physical stance and avoids "knowledge of physics", it does so at the expense of conjecture introducing risk in its predictive accuracy.

Just as it is possible for the mind to improve predictive speed when moving from the physical stance to the design stance, albeit subject to a higher level of empirically-based skills and incorrect inference risks, it is possible to improve yet further by adopting the intentional stance.

The intentional stance is the fastest but the least reliable predictor of an object's behaviour and effect in a specific real world situation. It second-guesses situational intent or need by drawing heavily on experience and precedence of observed behaviour in context.

Models of intended effect, in which fast decision making can pay off in survival or opportunistic situations, mean that in evolutionary terms a needs-driven, top-down mental modelling capability has been refined and furnishes systems reasoning with a purpose-oriented strategy of predictive awareness. It relies on extracting behaviour patterns from experience. It is the most skilled and most remarkable of the three survival models, providing a composite of speed and workable accuracy that is critical in the evolutionary formula. In the earliest stages of man-made system creation (and also in young children's cognitive development) it is likely to outrank the what-does-it-do and the how-is-it-done interpretations.

This fastest of prediction strategies has the distinct penalty of inaccuracy. Imputing intent relies on a range of unreliable factors, including assumptions and validity of precedence. Conversely, the most accurate prediction may be fatally slow.

The optimal stratagem to simultaneously counter error and minimise delay is to develop all three types of cognitive models (probably in parallel according to Dennett and other psychologists) and to resolving inconsistencies in their descriptive features through trade-off and reconciliation of all three stances (probably iteratively according to Dennett) in order to maximise the benefit/risk ratio.

This triad of mental constructs – form, function and effect – is taken to be axiomatic in this investigation. It should be noted that form is no longer being used in its earlier philosophy sense, as in Formal Cause, i.e. in its sense of the 'structure of anything as opposed to its constitution' [Collins, 1991, form, meaning 17. *philosophy*], but in its sense as the holistic essence of an object's physical existence, comprising its material substance and its structure. In support of the ubiquity of this triad of inherent constructs, they are evident in many guises; for example, in software as data, algorithm, control; in business transformation as means, ways, ends; in system modelling (SYSML) as structure diagram, behaviour diagram and requirements diagram. As an innate way of reasoning, this pattern is detectable in many schemes of thinking,

Stances lead to the model in Figure 3.7. In it, a system-of-interest sits in a real-world environment – its defined context of operation or use – shown as having a notional boundary that limits its extent of consideration to only the significant interactions with the system-of-interest. The result of those interactions is a change in the course of actuality.

For either system-of-interest or environment the optimal functional and physical boundaries, that permit adequately rigorous and compact descriptions at a particular level of structure detail, i.e., abstraction, may or may not be congruent. In well-designed systems form and function have evident correlation, but only in the simplest of systems does this even approach being one to one. Changing the level of structure detail and abstraction, this form/function non-congruency repeats and the task is to maintain the least tangled relationship between them. There is invariably a complexity of threads between one level of detail and the next. It is a systems engineering misrepresentation that in a 'flow down' of requirements, implying mapping functional and physical needs/constraints through successive levels of detail, they are 'allocated' to objects at a subordinate level of detail, [IEEE, 2005, pp22,29]

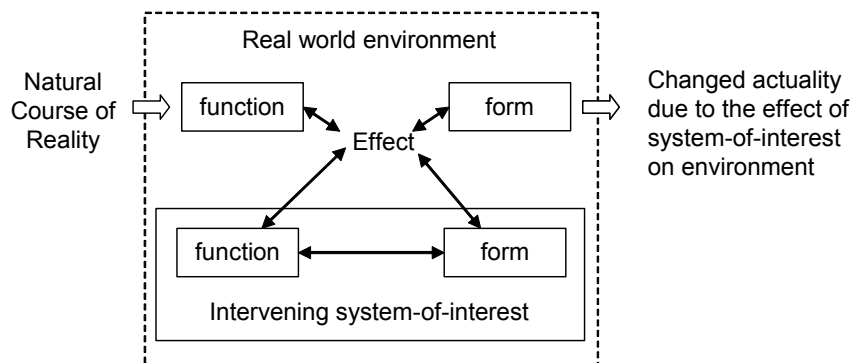


Figure 3.7 System-of-interest Intervention in the Real World

Aristotle advocated that form follows function in natural systems [Aristotle, *Anim.*]. Writing on atomism and psychology, Lucretius on the other hand identifies that what happens naturally to exist is the cause of its use [Lucretius, *Nat.*, 4.823-57]. Neither had the benefit of Darwinian thinking. For man-made systems, either may be the genesis for intentionality, but neither should be developed far without the other.

Correlating function and form, mapping and expressing their relationship, and resolving inconsistencies between them would seem to be a concurrent/strongly-iterative task. The engineering processes involved in the building, mapping and resolution of these two stances, and of their manifestation as communicable models, must perform have a high level of mutual dependence and interaction.

Resolving consistency between physical objects in the real world and functional objects in a cognitive world may require a high degree of skill – and, for the design of significantly complex systems, a high degree of team working and/or tool support, both of which depend on rigorous communication conventions, i.e. semantics/tangible model conventions. With the degree of complexity in present day systems, an individual with successful models of intentionality in their heads has restricted value unless these can be communicated to others in a team – how this is done is taken up in architecture descriptions [§10.4.3].

Function is taken to be ‘the natural action ... of a thing’; natural meaning ‘being so through innate qualities’, action being ‘the way in which something operates or works’ [Collins, 1991, function meaning 1; natural, meaning 5, action; meaning 11]; i.e. ‘the natural way in which a thing works as a result of its innate workings’.

Function is commonly and inappropriately synonymised with behaviour. Behaviour is a response to context or circumstance as in ‘the aggregate of all the responses ... to a specific stimulus or group of stimuli’ [Collins, 1991, behaviour, meaning 3]. Thus function is intrinsic and fixed. Behaviour is dependent of external interaction, is dynamic and cannot be applied to an object considered in isolation.

This text therefore uses:

- form as the intrinsic composition and structure of an object;
- function as the intrinsic transforming actions it is able to perform;
- effect as the induced material and state changes (the outcome of intervention) resulting from interaction of an object’s form and function with a particular environment.

As seen in the design strategies of §10.4.2, their emphasis of use characterises design strategies, for example:

- form/function-driven design: “this is what technology can implement and this is how it might change the world”

- effect-driven design “this is how society wants the world changed, so what will satisfy this?”

Effect is applied to the unified phenomenon of change caused by the intervention of an object at a location in the space/time manifold. It is a composite of functional and physical intervention in the perceived functional and physical representation of the real world.

These then are the primary teleological mental attitudes used to rationalise objects and, used iteratively, systems. They do not appear in any way remarkable because they are innate and they manifest themselves in a multiplicity of (maybe all) creative actions. As stressed, systems reasoning is not peculiar to systems engineering, though it may (or at least should) be applied in systems engineering in a better disciplined and more explicit way, and be supported by better methodology, than in most other branches of science, arts and humanities.

Some form of mental semiotics establishes tokens of information, e.g., system model semantics and syntax, which signify and relate objects in both the physical world and in the world of ideas. The machinery of systems psychology gathers and transmits data on the real world according to an ontological schema that gives it meaning. That is, we perceive the world according to a set of tried-and-tested conventions: experiential dimensions that allow us to extract a workable and useful level of information such that we can survive in the world. A primary component of that survival strategy is prediction. It is also a cardinal feature of systems engineering.

The data bank of models built from this data undoubtedly serves many purposes. Understanding how these models are built, other than as the connection of meaningful neural patterns, may at present be sublime but we are offered some clues from Kant’s experiential dimensions and Dennett’s object intentionality reasoning. Connection from both of these cognitive processes to the resulting bank of models gives rise to notional intersects that form a matrix (at least when tangibly expressed) of modelling conventions that must allow ‘writes, reads and updates’ compatible with the ontological and teleological schemas of the mind; see Figure 3.8.

				Teleological Schema		
				Object Intervention Reasoning		
				Stance		
				Intentional	Design	Physical
Ontological Schema	Experiential Dimensions	Percept	Substantive	Cognitive Models of Systems in Reality		
			Spatial			
			Temporal			
			Causative			

Table 3.8 The Ontological and Teleological Schema of Systems Cognitive Models

Nature would of course not see any of this modelling as sublime or even surprising. It spent eons of trial-and-fitness exploring the matter. Attending only to the configuration-managed package *genus Homo* and (maybe optimistically) assuming on average two reproducing offspring, it has had around 10^{200000} individual do-check-acts to refine a solution (7 million generations [Dawkins, 2005, p.136]). Given the benefit of the spadework done with *class Mammalia* (around 163 million prior generations) and then *subphylum Vertebrata* before that, whatever the actual figure, nature has the time to evolve the brain and pre-wire it with a workable way of dealing with reality.

As seen in §3.3, an understanding of the cardinal dimensions of physical entities and their conceptual counterparts have been distilled from over two millennia of philosophy – rather less from science. They are interpreted to be a system's spatial and temporal values, and its ontological and teleological values. These four semiotic values appear to encapsulate the dominant conceptual and communication dimensions of man-made systems and they are shown to the left in Figure 3.8. They signify four fundamental classes of information, two cognitive, two real-world, that are *sine qua non* of the system life cycle and its management. From them more refined models of a system may be built.

The rows to the top of Figure 3.8 also constitute a schema, in the terms psychological meaning of a 'mental model of aspects of the world ... structured in such a way as to facilitate the processes of cognition and perception' [Collins, 1991, schema: meaning 3]. Also, appropriately as in the philosophy of Kant, schema refers to 'principle that enables the understanding to apply its categories and unify experience', here taken as the categories of *a priori* understanding he expounds [§ 3.3.3].

Definitions of purpose, boundaries and rational structure are substantially influenced by the concerns of the observer. With many individuals and teams interacting to common purpose, this observer dependency gives rise to many fundamental challenges in systems engineering – common perceptions of boundary definition, quantised structural detail and architecture descriptions – due to observer dependency, relative viewpoints and mutual interpretations

The rebalancing of contributions in systems reasoning has the benefit of reducing the traditional distinction between hard systems and soft systems regimes. With an increased stress on philosophy's ontological underpinnings and psychology's instruments that address how the mind interprets the world, the notion of an environment having deterministic existence or need is blurred. The distinction of hard and soft blurs, and associated problem-solving strategies become less polarised. The exploration and interpretation of required service interventions is fundamentally no different, other than the degree of chaotic behaviour seen in socio-economic and social systems compared with the conventional trade-offs between service need and product solutions seen in 'hard' systems engineering.

3.5.3 Modelling Systems

The truism that "if it's not reality, then it's a model" describes a balance in a system's state at any point in its life cycle. As solution evolves from need, so an end-state system progressively is transformed from all model to all reality. Models are implicitly a prime instrument of the individual reasoner and explicit an enabling mechanism of team reasoning.

There is (we believe) one reality and there are countless models of it. How the mind manipulates these models, perceives environmental information and transmits information that represents reality from one mind to another is fundamental to systems reasoning, and it has particular consequences for how systems engineering is conducted.

Whether a physical system intervening in an operational location at a particular instant, or perhaps a human activity system of life cycle processes controlling such an intervention regime, the reality of man-made systems can only be represented to the extent of the internal variety in the mechanisms (system life cycle process transformations) and the artefacts (system life cycle process work products) involved. The well-known corollary is that models are approximations of reality that serve particular contextual circumstances well and may not transpose usefully into unfamiliar or novel situations.

'In developing models there is always a trade off. A model is a simplification of reality, and as such, certain details are excluded from it. The question is always what to include and what to exclude. If relevant components are excluded there is a chance that the model will be too simple in nature and will not support the development of the understanding desired. On the other hand, if too much detail is included, the model may become so complicated that, again, it fails to promote the development of the deeper levels of understanding one seeks' [].

Models can take many forms and exist in many media. They all have the purpose of allowing the mind to predict action, to impute intentionality, what is yet to become. Models start in the mind; the genesis of a conception about real world objects and events that might surround its existence and intervention or effect in a particular real world setting. If the physical creation and use of such an object is totally within the capability of individual, then the formation and

formulation of models may proceed no further. It is a purely cognitive matter. But if things are, as in the case of complex engineered artefacts, a matter of multi-disciplinary team involvement, then mental models have to be given tangible form.

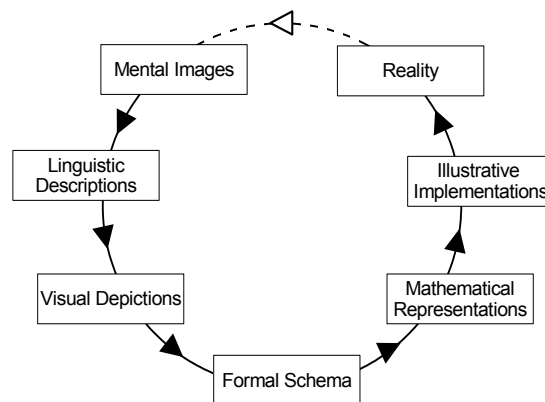


Figure 3.9 The Progression of System Models According to their Rigour

Figure 3.9 conveys some of the principle tangible forms that models may take. They still observe the need to convey intentionality and do so in a manner that preserves their originating thoughts and transmits ideas, and allows them to be refined. 'Models are deemed appropriate if they survive rigorous examination and if, in addition, they are internally consistent structures. Internal consistency of a model requires it to be precisely defined' [Penrose, 2004, p.12]. This internal order is where their information lies. Transmission of this information requires sender and receiver to possess the same conventions regarding meaning, as observed in §3.4.3. The rigour of those conventions increases around the cycle shown in Figure 3.9.

The semiotics, semantics, grammar of language; the nuances of shapes, spaces, arrows and the dotted line in pictures; the meaning of diamonds, circles, lines and naming conventions in formal notations; the logic, operators, symbols, structuring and dialectics of mathematical representations and coding; and the partial, imitative likenesses of mock-ups and prototypes that are the pathfinders into reality: all require the conventions of their coding to be shared by cooperating parties, by members of a team, and its rigour, richness and effectiveness count heavily for relaying models of systems in one mind to form sufficient useful corresponding models in the mind of another. To complicate matters, this information transfer may not enjoy reciprocity to iteratively resolve any weakness, particularly if the passage of time acts as a barrier to this.

The order in models thus foretells the order in systems, and the building of these models is a necessary prelude to the building of real systems. This sequence is in practice a cycle, for it is the final existence of reality, and cognitive models that the mind builds of it, that is the source of future conceptions of reality. Reality came first, human thought followed it.

Ambiguity and imprecision in a model is not necessary detrimental. Linguistic descriptions and visual depictions, as used throughout this text to convey some intended notion of systems engineering, have particular benefits. They avoid unnecessary detail and offer multiple interpretations or viewpoints to coexist. They avoid the need for tedious and possibly disadvantageous conceptual constraints. Their flexibility can avoid the necessity to aggregate an excessive number of alternative model views in order to build a representative picture of holistic complexity. However, their creative strength is also their prescriptive weakness.

To gain greater formality and rigour in specifying future reality, the rigour of mathematics and logic are called into play. Understanding and accessibility requires massively more knowledge about the coding conventions and the translation of one model form to another. It ceased to be intelligible to the problem owner or to business overseer and resides in an environment of technical specialists. Repeated imploratory trips around the cycle in Figure 3.9 help to communicate between the minds of the technical problem solvers and the minds of the real world problem conceivers – their ability to travel along the lower portion of this cycle may be limited. The feedback of tangible, testable, representative examples – prototypes and

demonstrators – provides their reassurance and insight into prospective systems solution acceptability. It is the stepwise (or repeated spiralling) building of confidence from models – starting with groups to work **Concept**, through **Assessment** of the benefit and viability of linguistic models of requirement and engineering/mathematical models, through representative models that provide **Demonstration** of future utility, and the **Manufacture** that replicates its real world existence – that is the basis of transforming from mental model to installed capability, and the basis of life cycle management dealt with in §5 and the CADMID ‘cycle’ in §9.4.1

Considered in these terms, service requirements and product requirements are system models; architecture descriptions are systems models; test specifications are systems models. Building models is a fundamental systems engineering activity. ‘Model-based systems engineering’ is tautology in the extreme.

Models are a route to intentionality and its effects, reasoned through mental images and handled according to stances. ‘The general form of the mind model is an adaptation of the animal’s way of life The nature of the model is governed by how it is to be used [Dawkins, 2006]

The data bank of information that the teleological machinery of stances draws on is itself drawn from perceptions of reality. This Kant and others tell us is categorised according to four percepts of reality: the structuring of information gathered by the senses’ that gives recognition and meaning to objects and phenomena. The resulting ontological map of reality is built according to models, perhaps gross approximations, of the real world. These are both interpreted and externally expressed according to these percepts, which acts as communal conventions and allow us to conceive of and communicate what things are, where and when they are, and what they do.

It is hypothesised here that the cerebral data bank of models transmits and receives according to two (and perhaps many more) key schemas: teleological and ontological. At their notional intersection lie models with qualities conditioned by the demands of percept and stance. These cognitive ‘objects’ or models (neurological orderings) possess the mark of both.

The nature of these models is shown in Figure 3.10 and they are seen to represent familiar, communicable qualities of being and purpose. They are the essence of shareable understanding used in systems engineering with different degrees of formality according to progress through the life cycle. They represent current or future existence and influences of real world systems.

		Teleologic Schema		
		Stance (Psychological / Engineering)		
		Intentional / Effect	Design / Function	Physical / Form
Ontological Schema	Percept	Substantive	<i>Intervening Entity</i>	<i>Logical Object</i>
		Spatial	<i>Absolute Location</i>	<i>Object Association</i>
		Temporal	<i>Absolute Time</i>	<i>Event Relationship</i>
		Causative	<i>Situational Transformation</i>	<i>State Transition</i>
				<i>Material Transformation</i>

Table 3.10 Model Qualities According to Stance and Percept

Issues of systems reasoning in Table 3.10 are considered further as the practices of system design, and its dependency on architecture description for team communication and team contributions, are examined in §10.4.

3.6 Coda

Through rational argument, presuppositions and axioms, and the physical and metaphysical views of the nature and structure of our universe, reality can be described in terms of systems. But are systems just a metaphor of reality; a powerful instrumentalism that is simply an expedient? Do systems exist in reality? Is any system nothing more than a product of human artifice – a mental convention that is a counterfeit of reality – or is a system a mental replica that faithfully conveys physical reality itself?

Martin advances that systems do not exist [Martin, 2007]. Philosophy says we cannot tell. Science offers the most credible evidence that they do. Psychology presently avoids the question.

Nonetheless, abductive reasoning strongly implies the real world existence of systems: that mankind's cognitive notions of system are a wholly natural consequence of a counterpart physical order in the real world. This hypothesis has been involuntarily tested as rigorously as mankind can through the rigorous empiricism of Darwinian evolution. Supported by the scientific corroboration of thermodynamics and information theory, the argument 'for' is cogent. Ultimately, though, the polemics philosophically appear irresolvable.

On such affairs of reality and complexity, Haldane sublimely summarised the matter. "Reality is not only more complex than we conceive, it is more complex than we can conceive". It is proffered that this axiom may be seen as a corollary of the 'Law of Requisite Variety'. Subject to the assumptions that Ashby furnished a universal truth, and (denying Cartesian dualism) that cognition is of and therefore limited within the realms of reality, we lack the cognitive capability to conceive its complexity. If this holds good, then systems reasoning binds Ashby and Aristotle, Boulding and Brentano, and Kline and Kant in common purpose.

3.7 Outcomes

Systems engineering is widely held to be a multi-disciplinary topic and thus can be traced to a scheme of complementary, contributing disciplines. In this reassessment of systems engineering it is propounded that it is effectively considered according to a scheme that fuses three complementary elements of discipline:

- a substantially subject-independent federation of systems reasoning.
- a strategic appreciation of technology-based engineering;
- selective regions of goal-orient, team-management science;

The first of these is the least considered and least well developed aspect of systems engineering. This chapter has addressed this by considering systems reasoning from the standpoints of philosophy, natural and social science, and psychology.

It has considered the physical and metaphysical existence of systems, and interpreted how this influences the way human's deal with and treat systems.

Building from these intellectual origins of philosophy, the chapter introduced some of the pragmatic advances by the natural and social sciences as they considered what system are and how they relate to the pursuit of established disciplines. It then considered hypotheses in psychology that appear to have direct relevance to the mechanisms governing how humans employ systems reasoning, and why it so important them. It is advanced that this is a cardinal aspect of how engineers tackle otherwise intractable complexity in systems.

Though systems reasoning may apply in most subjects, its influences and consequences have a particular relevance to engineered artefacts and to the body of practice that expedites them. The information items, technical transformations, decision-making schemes and management strategy that characterise systems engineering would appear to be predicated on systems reasoning. The psychological dimension of systems reasoning goes some way towards resolving some of the long-proclaimed, subconscious dimension of 'art' in the discipline of systems engineering.

Although systems reasoning has a foundation in cognitive matters, its explicit use in systems engineering depends on the assistance of methods and tools – on spoken, graphical and logical language formalisms, and on the paraphernalia of information technology in order to more handle the rigour and consistency of system models. A clearer appreciation of the fundamentals of systems reasoning may thus bring insight to how all this is entailed in systems engineering principles and is then manifest in effective practices.

The far-from-comprehensive presentation of systems reasoning in this chapter resonates with aspects of the ‘thinking’, ‘movements’ and ‘theories’ of several system-oriented thought leaders over the last half of the C20th. However, the balance of their respective approaches does not readily offer sufficiently consistent or appropriately-directed descriptions and explanations of how systems are reasoned about in order to permit some prevailing systems engineering challenges to be satisfactorily addressed.

Observing Horace’s maxim ‘either follow tradition or invent what is self-consistent’ [*Flaccus, 56BC*] this chapter has departed from custom and drawn on some separately, well-trodden, substantially parallel paths of thinking to offer a rebalanced whole.

A particular ideological framework has thus been offered that assembles systems-oriented aspects of philosophy, sciences and psychology. Though necessarily very restricted in its considered here, it may merit refinement and further integration; Horace again; ‘he has half the deed done who has made a beginning’ [*ibid*].

Some of systems reasoning’s consequence on how systems engineering might be visualised, codified, educated and practiced are dwelt on in the remaining chapters. In particular, aspect of this systems reasoning are applied later to develop clarifications of traditional views, specifically business process models and competence models of systems engineering in §4 and §6, and to advance some fresh ways of tackling the challenges seen in UK defence acquisition, especially in regard to architecture descriptions and human factors in §10.

Treating real world entities or situations as systems is far from peculiar to systems engineering. However, the formality of treatment required in systems engineering – to coalesce a broad range of scientific and technological phenomena, at the same time to dig deep into detail, and to ensure that in every regard the dimensions of man-made intervention are harmoniously matched one-for-one with the dimensions of the environment being intervening in – is why systems reasoning has such prominence and influence in the make-up of systems engineering. It is why systems engineering is over-and-above an engineering discipline combined with the executive influence of management discipline, and why it is becoming recognised for the “superior tool” that it is.

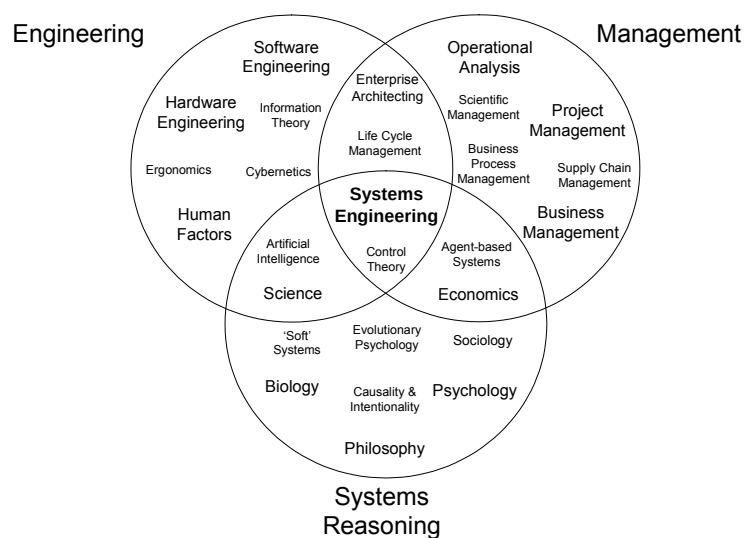


Figure 3.9 The Meshing of Systems Reasoning, Engineering and Management that Establishes the Context for Systems Engineering’s Focus

Superimposing the three contexts conveyed in Figure 2.4, Figure 2.7 and Figure 3.1 leads to the an overall contextual description for systems engineering. It suggests how it has been assembled; why it may be seen as a distinguishable yet cross-disciplinary subject; who most immediately benefits from it; the sort of individual competences that contribute to it; and the organisations/functional units it is likely to be found in.

This chapter advanced that we interpret ordered and structured behaviour in the world in terms of discrete systems because our brains most usefully serve us by constructing bounded, systematic models, about which we can rationalise and by which we can predict. For all, this is a day-to-day subliminal act, prone to imperfection; it is subliminally that most apply it in systems engineering. For some favoured individuals it is a more structured, better developed and highly beneficial cognitive faculty, and a basis of emphatic, disciplined systems engineering. For project teams, it has to be explicit and thus more disciplined. Without an explicit and communal declaration of its nature, team contributions are impracticable. Codified guidance is a necessary foundation of team action and for the communication required to create 'team cognition' about the systems they are responsible for.

How the ramifications of all this are codified in order to permit consistent and repeatable systems engineering application across and between business organisations is considered in the chapter that follows.

4 THE FORMULATION OF SYSTEMS ENGINEERING

4.1 Prologue

In *Rescuing Prometheus*, Hughes recounts how the system builders presiding over the SAGE and Atlas programs referred to in §2.3.1, used an empirical or cut-and-try approach for solving engineering and managerial problems [Hughes, 1998]. Out of it, patterns of dependable, coherent action emerged. From this a growing body of practitioners began to advocate the putative nature of such actions. Spurred by this recognition, engineers, academics and social scientists articulated, rationalised and fed back the experience amassed during these ground-breaking projects. They presumed to create a codified engineering and managerial science which they called systems engineering. A relevant, meaningful and coherent discipline was borne and, for better or worse, the name stuck.

For some the resulting discipline merited no distinction from what had preceded it. A frequent commentary on systems engineering was, and indeed has persisted, that it is “nothing more than common sense” – common sense conveying the concept of to-be-expected, basic (common) practical judgement or reasoning (sense) in the domain of engineering and its management.

This “common sense” label could perhaps with limited justification be levelled against the systems reasoning component of systems engineering, for, as advanced in §2, systems reasoning is an innate human attribute. Even so, employing systems reasoning to select, structure and coordinate a swathe of technology-based engineering disciplines, so as to create holistic systems, no more merits this “common sense” accusation than any other branch of philosophy that draws naturally on a system perspective.

Common sense, however, has a double meaning. It also conveys the notion of a uniform, widespread (common) awareness or understanding (sense) by individuals. In this regard there is no justification for past disparaging common sense accusations about systems engineering. For, whether systems engineering is considered in terms of a constituent of business process, or as an organisational capability, or a professional discipline, there has certainly been a weakness in its common interpretation. This common sense of systems engineering may reasonably be considered to still be its Achilles heel.

In his retrospective view of systems engineering in 1998, Brill’s response to the “common sense” criticism was that ‘in the context of the present knowledge base and the complexities of today’s and tomorrow’s systems, a “common sense” view would, as a minimum, encompass the application and management of a defined systems engineering methodology by integrated teams supported by an appropriate computer aided tool set’ [Brill, 1998] – advocacy respectively for establishing a system of coherent business processes, for complementary professional competence in individuals and teams, and for a supporting organisational capability.

As highlighted in §2.4.2, for recognition as a discipline systems engineering needs to be founded first and foremost on the formality of uniform, widespread technical actions and managerial controls – on a common sense of process. Only from such a foundation is it likely that there could be any prospect of exercising the jurisdictional authority to ever become a recognised profession. Only from such a foundation will organisations’ project and enterprise environments be conditioned so as to enable the effective conduct of systems engineering.

The Law of Requisite Variety postulates that the actions necessary to create and control a real world system of a given complexity themselves constitute a system of at least equal complexity. For the discipline of systems engineering the task therefore is to create a dynamic action and control model capable of exhibiting massive complexity; that comprises material, energy and information transformations according to declared intention; that generates related work products; that is subject to the provision of control actions and resources; and can then be commonly interpreted and applied by teams of people operating in separate, though interconnected organisational functions.

To achieve this, Tully’s characteristically lucid explanation states that what is required is a model for ‘managing and controlling such complexity ... [having] the dual attributes of

simplicity and power. Simplicity is achieved through conceptual economy – a few concepts clearly expressed and related; power is achieved through being able to apply a model recursively to match the complexity engendered in real situations'. Such a model, he hypothesises, will meet these twin requirements if it is '(a) systemic (based on sound systems principles), (b) realistic (a sound basis for doing real-life engineering), (c) rigorous (logically consistent)' [Tully, 1993, p45, using his parentheses].

The formulation of systems engineering in terms of process models that meet these aims has therefore not surprisingly been an intimate part of the evolution of the discipline from its origins in the mid-C20th. As systems engineering theory and practice have evolved, there has been a strong dependency on process codification to convey this wavefront of understanding to a growing community of practitioners. Without such codification, systems engineering would lack the necessary coherence to meet its technical and commercial challenges. Without it, systems engineering would have far less claim to be called a discipline, much less so a profession.

The supreme form of codification – standardisation – thus appears set to remain an integral part of systems engineering's continuing evolution. As systems engineering is refined, adapted and re-formulated (following Abbott's arguments in §2.4.2) in order to meet existing and new, more stringent challenges, both its effectiveness and wellbeing will be in part determined by the evolution of the standards that define it.

4.2 Synopsis

The value of process definition in systems engineering is deep rooted and its formulation over decades has given rise to models of systems engineering that are a basis for its common sense of technical cooperative action. These models provide a language for clarity of communication and understanding. As seen in §9 and §10 they enable communal appreciation within and between project teams, as they and their members collaborate to meet intra-organisational agreements and inter-organisational commitments. Accordingly, this chapter rationalises and develops a business process view of systems engineering that in its practicality is gaining wide acceptance.

It firstly considers the value of engendering a common perception of systems engineering across and between business organisations. The failure to satisfactorily resolve this goal has been a fundamental inhibitor of the effective application and wider take-up of systems engineering. Its resolution is a prerequisite for MOD, or indeed any other undertaking, if it is to build a policy of business enhancement around systems engineering.

Addressing this issue, this chapter lays the process foundation for the business excellence vision conveyed in Figure 2.3. Three primary viewpoints of systems engineering are advocated: a profiled set of business processes, a model to ascertain corresponding organisational capability and a mutually consistent description of individual competence.

In conjunction with §6, this chapter demonstrates that systems engineering can be represented according to a triptych of viewpoints of an internationally ratified baseline. This is undertaken in a manner that allows each resulting model to be unified to the other two in an intimate, consistent, traceable manner. In architecture description terms [IEEE, 2000], these three models equate to views arising from the primary concerns of:

- consistent, coordinated, quality-focussed business action;
- an effective resource base provided by organisations to support this action;
- a dependable, communally-endorsed body of practitioners to execute and manage these actions.

All three facets of this triptych are necessary in order to convey a comprehensive picture of systems engineering able to meet Abbott's requirements [§2.4.2]

The process model developed in this chapter is subsequently transformed into complementary representations of capability and competence in §6. Although the concept of relating business process to capability and to competence is not in itself radical, expressing them as a mutually traceable trio, based on a single model of systems engineering, is. It is

considered timely to undertake such an approach. The attention to systems engineering by ISO, bringing its customary focus of trade and commerce, has engendered a favourable mood for the adoption of a truly international model; one able for the first time to offer a credible, non-partisan approach to a definition and description of the discipline.

An analysis of the benefits of such a systems engineering model commences with a justification for the 'process approach'. From this a survey is made of past standards – milestones along the road – that have followed this process approach. This succession of standards has incrementally evolved, each building on its predecessors, finally to emerge as ISO/IEC 15288, to which National Standards, concerned with narrower but more detailed issues, now refer.

The survey highlights that that no standard model has sufficed for long, and that ever changing business circumstances and organisational responses will ensure that no solution remains immutable. However, the current convergence to an International Standard has brought a degree of much needed stability and consensus to what has been a contentious aspect of systems engineering. The background to this situation is reviewed.

ISO/IEC 15288 is a model with many flavours of interpretation to suit a wide range of concerns of its users. The rest of the chapter thus provides one such interpretation; it serves the purpose of informing later discussion on MOD acquisition advancement and systems engineering in IPTs. This particular interpretation summarises some key elements of Editorial rationale and structuring that helped lead to the International Standard's publication in 2002 [ISO, 2002].

This analysis of the ISO model is undertaken in reverse order to the sequence of presentation in the International Standard's text. It first builds an argument for the logic of the technical process definitions that constitute the engineering focus of the model. They are the kernel of the standard, as illustrated in Figure 4.3. A rationale for their composition is shown to flow from the systems reasoning themes advanced in §3.

Some of the strengths and weaknesses of the resulting model are critiqued. The former underpin arguments developed in this investigation. The latter act as pointers to the direction of evolution of the International Standard's inevitable successors.

As described in §2.5.2, systems engineering also draws from management discipline, and it is the management contribution to the process model that completes this analysis of the International Standard. It is project and enterprise management that set systems engineering firmly in the business organisation context. The deliberate project orientation of the standard is examined; this prepares for a consideration of the role and nature of IPTs in §7.5. Each project is subject to the authority, control and support of corporate enterprise management, and it is the influences and responsibilities of this enterprise management context that are then examined.

Finally, the value and use of agreement processes to bind the elements of organisation responsibility for systems, and to govern the trading relationships between organisations, is considered. This makes plain that systems engineering's ultimate purpose is to meet the needs of society and of individuals in it through the conventions and practicalities of trade, and it illustrates how the technical processes are viewed at the inter-project 'interfaces'.

4.3 Codifying Systems Engineering

4.3.1 The Role of Standards

Broadly, standards serve three distinct, though related purposes:

- to be an authoritative or recognised exemplar of correctness, perfection or quality, against which others are compared or measured, and to which ultimate reference may be made;
- to establish principles of integrity, propriety and trustworthiness that establish the confidence to cooperate and trade;

- to be a rallying emblem, or focus of allegiance to a special cause, so that mutually held beliefs or commitments may be furthered.

In roughly similar measure, each of these purposes has figured in systems engineering's codification over the last 40 years. They have led to ratified models of action, propriety, attainment, technical communication and discipline identity.

Advancing through a succession of models, systems engineering best practice has been on a journey of progressive refinement, codification and application that now serves to unify multi-party teams in intra- and inter-organisational situations. Initially it predominantly addressed the creative focus of technical transformations and their outcomes. It then moved toward quality enhancements and organisational assessment. With experience came a balance of attention to standardised modelling and communication conventions, and finally to considerations across complete engineering and system life cycles.

As a result of standards the systems engineering community now can draw on more stable, logical models of action, objects and outcomes. Arising from this is a less capricious and more rational technical vocabulary. This vocabulary does not just communicate a holistic engineering perspective to systems specialist. It also offers a unifying basis for translating to and between the implementation technology-based engineering specialisms that systems engineering is founded on. Standardisation is thus building a stable ontological map that brings discipline to collaborative endeavour and necessary formalisms to engineering.

As cross-discipline uniformity from standardised systems models takes hold, more objective decision making and selection of competing technologies follows. In turn this is likely to have more influence on previously disparate engineering and technical disciplines. It will help extend a systems' lingua franca that can be understood by evermore diverse fields of science and technology – an outcome considered in §10.5 with regard to human factors.

The foremost direction that systems engineering standardisation is moving in is toward greater business orientation. Trading agreements, new markets and new sources of technically skilled labour have combined to encouraged global markets and global partnering in order to provide and sustain systems. The upshot is the need for more common understanding of systems engineering expressed as a cardinal element of internationally recognised business process, and thereby to complete the gap in Figure 2.9. In this regard much remains to be resolved. Before systems engineering can sit comfortably and acceptably alongside existing cardinal organisational roles and responsibilities, its inconsistencies with longer-standing branches of engineering, technology and management need to be resolved and expressed through a more compatible set of standards.

Consequently, systems engineering is increasingly being defined as part of a continuum of business processes. Its positioning as an integral part of the industrial trading scene, comprising multi-disciplinary teams and networked-enterprises, has been of paramount importance for the discipline. Fundamentally, a widely-regarded systems engineering discipline needs to be presented in terms of codified, de-conflicted views in order to unlock its value across businesses. Beyond this, interpretation, customisation and adaptation across cooperating responsibility structures, commercial sectors and national interests are ensuing requirements. Ultimately, a better harmonised set of related standards will emerge.

Translating sound technical principles into good business practice first and foremost depends on a foundation of recognised processes that:

- codify a widely-accepted model for cooperative action;
- designate common language for clarity of communication;
- establish common understanding to underpin agreement and consent;
- permit actions and responsibilities to be cited in formal or informal acquirer/supplier transactions;
- define a basis for determining organisational capability;
- assist the development of practitioner competence and business team awareness.

International financing, communications and transportation have encouraged global markets and this has opened opportunities to spread the development costs and risks of evermore

complex systems. With it has come global partnering in order to supply and sustain these systems. The payoff will be a more common, more international understanding – a common sense – of systems engineering as a cardinal, unifying element of business process. This will fit systems engineering into a business scene that itself has become a hugely more complex system of multi-disciplinary teams and networked-enterprises.

Such standardisation is, however, not without its detractors. For them it is perceived as constraining, authoritarian and inflexible. Striking the balance between prescription, guidance and didacticism is a delicate judgement. Moreover, it is a shifting pivot, as the body of knowledge moves and the problem set becomes more exacting. Too often, procedure becomes a palliative, with competence development the real deficiency. This misjudgement is at the heart of past systems engineering standardisation failings and disappointments in derived organisational procedures. There is a vital and complementary relationship between process prescription and systems engineering competence levels is a hugely important and sensitive trade-off that all too often unreasonably looks for documents to compensate for development of practitioner skills. One is not a substitute for the other.

Constraint, authoritarianism and inflexibility have been giving way to constructive assistance. There has been a shift from overly detailed, one-size-fits-all definitions towards necessary and sufficient frameworks that can be intelligently modified in traceable, structured ways. The range of complexity, criticality, size and risk characteristics of systems, and of the organisations creating them, all change and need to be accommodated in tailorable standards that offer best practice and dependability.

Interpretation, customisation and adaptation are thus important factors in the acceptance and influence of systems engineering across cooperative responsibility structures, commercial sectors and national interests. This calls for business process standards in particular to undergo a maturity metamorphosis, moving from mechanistic institutionalisation towards competence-based judgment; one that is matched by a corresponding advance in systems engineering skill levels – the two necessarily go hand in hand. The complementary systems engineering faces of normative business process and professional competence lay either side of a fulcrum whose position defines where prescription should give way to informed adaptation.

Nonetheless, this adaptation in systems business process standards elsewhere has to be met by increased formality and rigour. Past systems engineering standards can be seen to have strongly relied on localised experiential interpretations, favoured methods, even prejudices and idiosyncrasies, and not sufficiently on disciplined models. An outcome of increased formality here should be better consistency and rationale in the terminology, mental models and communicated descriptions used by practitioners working as teams – something that has long been an inhibitor to systems engineering advancement.

Such moves toward better models, formalisms and structure in systems engineering standards will undoubtedly assist the presentation of some of the discipline's profound complexities. In truth, applying systems engineering principles to standardisation practice would hardly go amiss.

It is to be expected that standardisation of systems engineering work products, particularly data, will continue to rest strongly on formal detail and machine execution. Without this, the cooperative tools and virtual collaborative working environments of tomorrow's distributed systems engineering paradigm will be distinctly restricted, as will the opportunity to integrate or re-use the diverse contents of distributed model repositories.

As a result, codified views of system function and physical form are set to increasingly characterise effective system design and assessment, with more unified, standardised architecture descriptions, capable of offering greater accessibility and relevance to a wider community of engineering and management interested parties. Such descriptions will inevitably evolve with experience so as to better serve the interests of these parties, each having their respective concerns and viewpoints, and set in their different domains of application. With better communication and interpretation of system views will come greater familiarity, more widespread use and enhanced uniformity in system modelling. These trends, as they apply to architecture description standardisation, is considered further in §10.4.3.

Coherent codification is needed not only to serve across and between trading organisations but also across whole life cycles and multi-scale system interests. So-called whole-life, whole-system models still speak primarily to confined communities of practitioners and managers. Standardised decision gates, cost models, technical achievement criteria, readiness measures, and other indicators of life cycle advancement and design resolution are obvious candidates to benefit from normative criteria and metrics.

Thus it is that systems engineering process standards reflect, and will continue to reflect, a shifting baseline of formally accepted best practice as it is evolved by the discipline's community of practitioners and academics. Periodically, new standards proactively influence the scope, understanding and effectiveness of systems engineering and so steer a future path that arises from new insight into systems engineering principles and new refinements to its effective practice.

Shortly after the emergence and strategic employment of the fledgling systems engineering discipline, it was US military standards that formalised it in the eyes of government and major contractors. These standards provided a framework of codified knowledge for a technical community that was struggling with growing product complexity and challenging timescales. Latterly, it has been the international standardisation arena of ISO, where trade and commerce are its *raison d'être*, that has taken up the challenge of translating the technical dimensions of systems engineering more clearly into a business context, and thereby clarifying its wider relevance to society.

4.3.2 Models Founded on Business Process

Translating sound technical and management principles into good business practice fundamentally depends on recognised processes – on 'a series of actions that produce a change or development; a method of doing or producing something' [Collins, 1991, process, meanings 1,2]. Not surprisingly, the value of process definition in systems engineering is long-standing and has marked the course of the discipline. The goals over this time have been, and remain, clear-cut. They are to build common models of cooperative technical action, that use common language to communicate a common understanding in those collaborating to meet inter-organisational agreements and intra-organisational commitments. To meet these goals, systems engineering process models need to present widely-applicable, profiled sets of business processes, as conveyed in Figure 2.10.

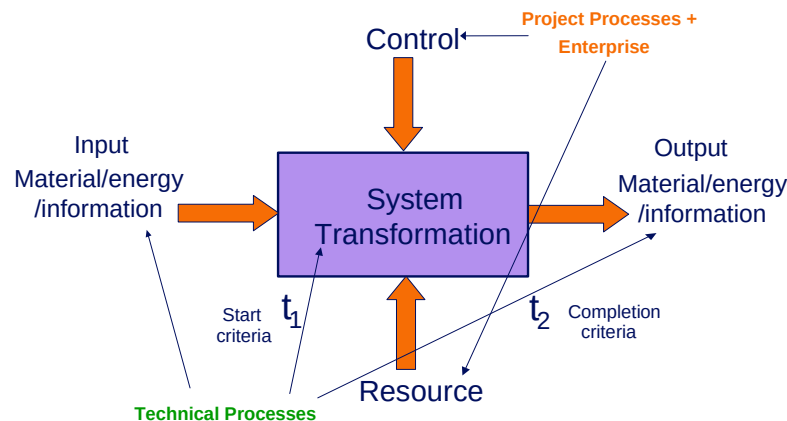


Figure 4.1 Typical Structure of a Process with ISO/IEC 15288 Process Classes Shown

ISO 9001 states that 'the application of a system of processes within an organisation ... can be referred to as the "process approach"'. Process is fundamental to the precepts presented in ISO 9001' [ISO, 2000]. This cardinal quality management International Standard defines a process as 'a set of interrelated or interacting activities which transforms inputs into outputs Inputs to a process are generally outputs of other processes'. This "process approach" is fundamental to the concepts behind the ISO systems engineering standard. Its stress on a 'system of processes', as opposed to a sequence, cycle, hierarchy or other logical relationship, is a hugely revealing of an important and powerful mechanism. 'Network of

processes' would perhaps be a better description of the approach to the management of systems complexity taken in ISO/IEC 15288's object-oriented logic.

This is not to advance that pre-ordained patterns or procedures of process relationship are misleading or of limited value. To the contrary, the next chapter considers why, in a fluid and adaptable system of processes, it is valuable for project and enterprise management to understand how simple patterns emerge from the instantiated, labyrinthine detail of unique flows around the network of processes. Moreover, it is the logic of systems reasoning that is used to introduce the individual elements in the system of life cycle process in §4.4.1 and is then used to synthesise these into a quintessential systems engineering model.

Within a system of processes, each technical process acts to transform the materials, energy and information that constitute a system into 'processed' material/energy/ information. It is the transformational relationship between inputs and outputs, and the transcription of outputs to become inputs, which structures business process models according to timelines and patterns. By this means, the data used to model a system in its conceptual state is progressively transformed through innumerable, successive executions of processes (iteratively, concurrently and recursively) to ultimately be materialised as artefacts embedded in the real world.

There are two orthogonal but wholly complementary views of a process. It can be viewed either in terms of the transformation that it performs, or in terms of its input and associated output characteristics – that is the box or the arrows in Figure 4.1. It may more easily be described in terms of both these viewpoints. In addition, descriptions of this kind may be expressed independently of time (a static model), or as a function of time (a time line or dynamic model).

Expressed purely as a transformation, a process is defined at its boundary as a 'white box' representation. In such a description the internal structure of interacting activities transforms an input in some purposeful way in order to deliver an output with benefit over the input. This transformation detail may be composed of similarly defined, finer-level detail processes, and so on, until the transformations are so specific that they become distinctive methods or become characterised by mechanistic constraints of particular tools employed to conduct them. In this manner, a totally rigorous white box representation of process can be defined independent of input/output definitions.

Conversely, a process can be defined at its boundary purely in terms of a 'black box' representation. Such a description knows nothing of transformation and defines the output response when a specified input stimulus is presented to the process. This black box view defines only valid products associated with the process. They result from conducting the 'work' that the transformation performs – in process terms therefore usually referred to as 'work products'. Again, ever-finer levels of detail in the work products can define input/output relationships with increasing rigour.

For any process, these two orthogonal process representations must be utterly consistent. They exhibit duality and are mutually dependent – any re-definition of one leads to a corresponding change to the other. However, completely rigorous transformation descriptions are, just as with completely rigorous work product descriptions, extremely difficult if not impractical to define. The middle ground offers a way forward.

In practice, a definition of life cycle processes needs to be meaningful to the people in organisations who are responsible for conducting and managing them. This has two consequences. Firstly, natural language is the most commonly used medium to model and communicate processes to humans, and this limits the rigour of either a white box or a black box representation. Secondly, humans naturally rationalise what is being done in terms of what it is being done to, and vice versa – a consequence of evolution and stances. The one description informs and enriches the understanding of the other. Linguistic process models are therefore limited in their value unless they contain elements of both representations.

Generally these mixed representations lead with either the transformation or the work product definition, and then draw on the other to enrich the description. This helps to give a more rounded understanding and to cover up weaknesses in detail. A resulting description is therefore likely to contain a degree of redundancy across the two definitions. This means that

if either description lacks accuracy, and particularly if they evolved with any degree of independence, then there will be inconsistency between them.

The business process models of systems engineering have traditionally been strongly biased towards transformation descriptions, with varying levels of detail seen in the work product descriptions, e.g., naming, structure, content. Latterly systems engineering models of capability, architecture descriptions and data exchanges have tilted the balance back towards input/output descriptions, i.e., work products; in the case of some architecture frameworks to the exclusion of even any implied transformation information.

In ISO/IEC 15288 it is the white box view of process that is the predominant description. No process inputs or outputs at all are explicitly defined. Each process's description of outcomes is the closest the International Standard comes to a normative statement of output work products. Nonetheless, together with the outcome, the purpose of each transformation and the description of their internal activity structure, do imply considerable output work product information. To a lesser extent, input work product information is also implied in descriptions. This is a matter exploited in this investigation's formulation of work products in §6.3.4.

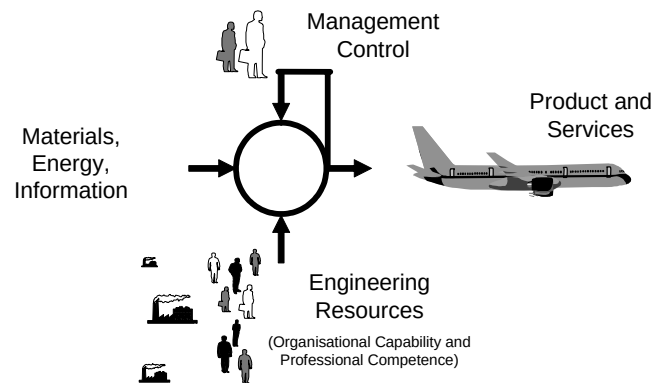


**Figure 4.2 A Process-based View of Systems Engineering:
People (in Organisations) are the Agents of System Life Cycle Processes**

Figure 4.2 conveys the notion that people in organisations perform processes in order to apply structured operations on a system, and this applies whether it is descriptions of the system or on its physical form. These progressively change the system's state and move it through its life cycle.

In turn, the processes need to be practiced in a compatible environment of methods and tools by competent professionals operating within an appropriate organisational responsibility and resource structure. Systems engineering thus contributes from a foundation set of business processes that underpin, through capability and competence, the business excellence of organisations, as shown in 2.3.

The steps of human action that transform models into actuality by employing technology, and thereby intervening in what is, to become what is wanted, are the essence of engineering. The structuring of those steps brings an essential ingredient to the systems engineering mix from which a discipline emerges.

As already advanced, the Law of Requisite Variety decrees that the discipline of systems engineering is unavoidably complex. Any model that attempts to describe it needs to reduce that complexity significantly. Since every model is a simplification to meet specific purposes, different people with different roles, responsibilities and concerns see systems engineering in terms of different models.

This presents a major challenge. Conflicting simplifications lead to models that emphasise particular concerns or approaches, such as architecture descriptions, requirements management, quality management, service delivery, risk control, safety attributes. Such diversity of concern can disperse the essential holism of systems engineering. It leads to factions of the discipline assuming progressively independent and unrelated practices. These

can be bolstered by competitive methodologies, technical fashion or proprietary supporting tools. In consequence, retaining and evolving a unity of principles, and offering consistent interpretation yet achieving broad-ranging practicability, has been a perennial problem for systems engineering.

It is therefore to be fully expected that a common, universal view of systems engineering, modelled as a dynamic system of processes, is difficult for practitioners to agree on or align with.

Steps along this path have been, and continue to be, contentious and partisan. The challenge is to arrive at comprehensible and practical meta-models of systems engineering action, that are self-consistent and sufficiently comprehensive to meet the overwhelming majority of situations.

At the highest levels of systems engineering process definition, these process standards continue to evolve:

- the US national standards (IEEE 1220 and EIA-632) are now three generations old,
- the international standardisation arena has responded to the recognised importance of systems engineering to international trade and commerce (ISO/IEC 15288 and AP-233),
- the SEI has provided essential models for raising the standards of (software-intensive) systems engineering execution within and between organisations (CMMI).

Individually and jointly these exert a strong influence on the conduct of systems engineering.

Such process models perforce emphasise the prevailing importance and concerns at the time of their creation; they advance according to the necessities and fashions of their time, and thus continually evolve. Process models are thus snapshots of current practice. They are axiomatic, in that they are propositions and empirically supported accounts of 'best practice'. In this regard they cumulatively recount a sequence of advancing baselines, typically linked to an element of didactic rationale. The USAF model of 1966 [AFSCM, 1966], though containing immutable 'truths' therefore looks very different to the ISO systems engineering model of 2002.

Such process standards attempt to convey essential reality, but as with any model they have problems of representing the complexity of that reality. They address some features well whilst misrepresenting others to the point of being misleading. Particular models are useful in some situations, and their scope, detail, application circumstances, and context of user knowledge and experience all contribute to their acceptability and popularity. They each have their focus of attention and they exhibit limitations of validity as model use moves outward from this focus. Matters are compounded by this not being a static scene; for experience and feedback from application bring change. The models must track an evolving picture of use, environment of organisational support and user competence.

The goal therefore is to define a set of technical processes that can systemically and adequately describe countless of instances of systems engineering conduct, governed by an intricate responsibility structure and ultimately set in a social context.

The organisational context in which systems engineering processes are typically conducted was concisely laid out in IEEE 1220-1994, and again in its 1998 versions. They each hold in Appendix A a concept with considerable value but little evident contribution to the systems engineering model in the main body of their texts; namely, that the 'systems engineering process' (in this National Standard aggregated to form the left-hand side of the 'V' model) is executed within a project environment that 'defines the objectives, success criteria, project milestones, and associated project priorities which will govern the integrated technical activities' [IEEE, 1998, pp.64-66]. It is associated with plans, teams, tools, controls and metrics.

In turn this project environment resides within an enterprise environment that 'establishes the policies and procedures which govern project activities ... [and] directives representing enterprise guidelines for establishing a viable product in a competitive marketplace' [*ibid*, p.66]. The enterprise is conveyed as the managed environment concerned with competitive provision of products, technology constraints, natural constraints, standards and laws.

This was a pointer to a categorical definition of systems engineering in terms of processes that could relate these two regions of organisational management. It was an approach

followed in the *DERA Systems Engineering Reference Model*, augmented by the social environment, which was the contextual setting of an enterprise 'shell', and at its technical centre was the implementation technology environment embedded within the systems engineering context. As Ramo put it, 'relate the technology to the objectives, the social environment, the available resources, the time constraints, and the economics' [Ramo, 1991, p17].

Figure 4.3 [Arnold, 1997, p8] shows this nested set of environments in which systems engineering sits, and this relates basically to the profiles in Figure 2.10. The hierarchical layering seen in this context view of processes exists by virtue of managerial responsibility for process. The sequence or precedence of use is independent of this structure.

This succession of nested environments therefore sets the scene for systems engineering's process codification. The overall context of the social domain creates the need for systems and establishes the laws, trading conventions and social mores that govern acceptable ways of creating and using systems.

Within this social domain, business organisations exist. Their role, as in the enterprise systems in Figure 3.2, is 'self-determining endeavour in a competitive environment'.

Each organisation's outermost layer of enterprise management looks outward to its customers responsibilities and its stakeholders in the organisation. Looking inward, the enterprise management is concerned with the strategic deployment of assets, including implementation technologies, enabling system resources, and trained staff, to the individual projects engaged in conceiving of, creating, utilising and retiring systems. The overall organisational responsibility for life cycle management resides at this enterprise level.

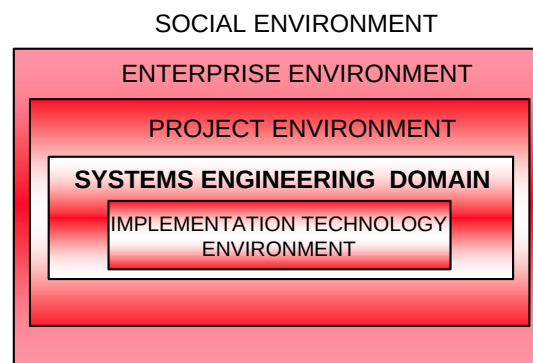


Figure 4.3 Context for Conducting Systems Engineering

Subordinate to the enterprise level is the project level. Here management is concerned at a more tactical level with achievement criteria that ensure continual progress against technical, schedule and cost requirements. This is contingent on renewed enterprise support at the major decision (stage) gates, where agreed allocation of assets, e.g., budget, skilled staff, enabling systems, is sanctioned. The project is the immediate and thus dominant contextual influence on how systems engineering is executed.

The system technical processes are conducted within a project (or a succession of projects) to achieve the engineering actions across the whole life cycle. They may be employed, commonly with tailoring, at any point in time and according to any level of technical detail.

Underpinning this system technical level are the implementation technologies, where the desired system elements are designed, built and tested by employing processes that apply to the implementation technology-based engineering disciplines, e.g. software engineering, human factors.

There has been an understandable desire to present systems engineering as well-ordered, sequential models of technical and management action. For those less familiar with systems engineering this may be no bad thing, but it sits uncomfortably with the massively concurrent reality of creating and using complex systems. The middle ground is a holistic system of

processes that can be systemically arranged to model primary patterns of systems engineering conduct that is capable of describing intricate detail when needed and can be related to corresponding responsibility structures.

Ideally, process patterns or models convey workable abstractions of highly complex actions. They need to accommodate countless situations, yet be simple enough to be a 'head full' for individuals, and also be capable of uniform interpretation by all practitioners.

This requires a process reference model to:

- have memorable structure with discrete, rational parts with logical relationships, and thereby form a coherent and complete system of processes (Tully's criteria) – this favours a model built according to sound systems engineering principles (and paradoxically, since it is a system, in accordance with the provisions it embodies);
- be capable of manipulation in order to manage extreme complexity when faced with any instance of man-made system creation and use – this favours an object-oriented model of real world actions;
- align with natural processes of creative cognition, and thus be practicable, instinctual and communally acceptable – this favours alignment with systems reasoning.

These then were some of the needs that ISO faced as it entered the systems engineering standards scene in 1996. The next section looks at how ISO's engagement progressed.

4.3.3 Synthesising an International Solution

ISO/IEC 15288 *Systems engineering - System life cycle processes* [ISO, 2002] was developed over the period 1996 to 2002. It thus evolved over the most formative period of Smart Acquisition. With its funding of the Editor, MOD was a significant, if somewhat unaware contributor to the thinking and development behind this first international codification of what systems engineering is, and how it fits into the conduct of business across organisations. As a result, aspects of Smart Acquisition naturally embody some key ISO/IEC 15288 concepts, as applied to defence business.

Whether ISO/IEC 15288 is viewed as a system life cycle management best practice model or a systems engineering best practice model (either is valid), it has made an important step in that it defines long-standing systems engineering principles and practice in a way that is more clearly related to need, service delivery, acquisition and supply, and organisational responsibility. It strikes a balance between engineering and management, and it conforms well to systems reasoning presumptions.

Process descriptions have been central to the codification of systems engineering since the mid-'60s. They have contributed strongly to its identity, and to awareness of it as an essential element of business practice, and to its recognition as a formal discipline. Refinements of these descriptions have, through to the present day, kept pace with the evolving understanding and the widening application of systems engineering. It is to be expected that process models will remain an important part of the formalisation of systems engineering.

Figure 4.4 is a customisation of a familiar representation of that history and it presents a "brief history" of systems engineering process standards and relevant software engineering influences. It conveys some important message about systems engineering's emergence and evolution. It shows that:

- as the technological and engineering powerhouse of the world, the United States synthesised the discipline of systems engineering and presided over its formulation until the turn of the C20th century;
- the defence programme origins of systems engineering are clearly evident from the sequence of DoD sponsored standards that span almost 30 years;

- there has been a steady shift from standards that address a sector of a business domain (US Air Force), to the whole domain (US DoD), to overall national interests (US trade association and professional bodies), to international codification (ISO and IEC) marking the ascendancy and pervasive influence of systems engineering in the international business scene;
- continual refinement and rebalancing of process representation has been necessary in order to reflect progressively better understanding of systems engineering and the changing imperatives of its application;
- the ‘interim’ (IS) and ‘provisional’ (P) designations of 1994 standards illustrate tensions and vested interests that threaten new systems engineering models then and for several years following, needing the expedient of these “under development” caveats to progressively gain acceptance and ratification;
- excursions by the implementation technology-based engineering into systems engineering formulation led to software engineering making major contributions, as shown in the lower region of the figure;
- fresh lines of thinking emerged from Europe via software engineering in the early 90’s and took a decade to penetrate the somewhat proprietorial and protectionist position the US had taken to defining the discipline;
- as Abbott points out, new paradigms are borne of old paradigms, and some measure of backwards traceability and co-existence of models is necessary for the protection of past investments and long-lead developments; it has taken nearly 40 years to achieve a clear cut international point of reference.

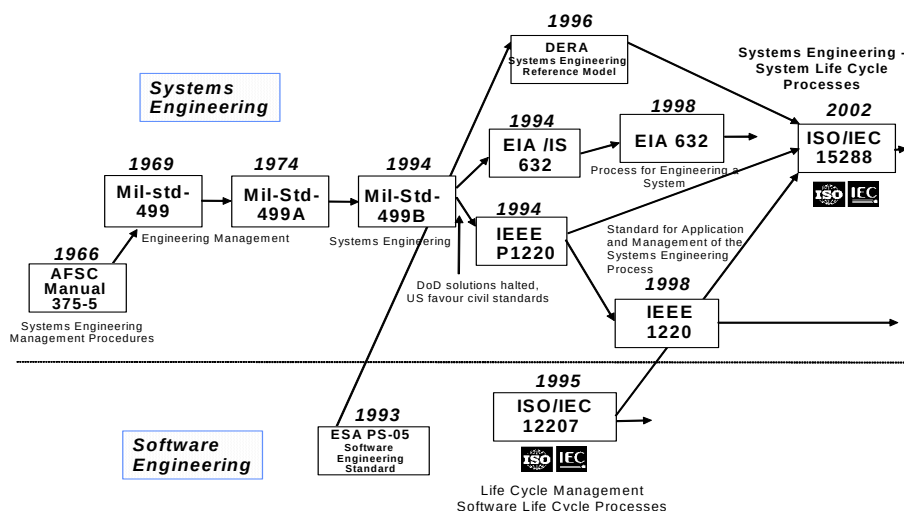


Figure 4.4 A Timeline of Influential Systems Engineering Standards

Figure 4.4 starts in the wake of the systematic approach to total system analysis that evolved in the US Air Force Systems Command at the height of its ICBM programmes. *System Engineering Management Procedures* [AFSCM, 1966] prescribed the management policies and procedures to be followed during the establishment of requirements, the design, development, test, operation, and maintenance of US Air Force systems. This landmark step described the life cycle of systems, prescribed a systems engineering management process and discussed significant document work products in detail.

The value of this formality was recognised at the pan-DoD level and three versions of Mil-Std-499 were subsequently developed; the first two were titled *Engineering Management* and the last assumed the title and a clearer identity of *Systems Engineering*. They have had a profound influence on systems engineering, not least because they were increasingly cited in US defence systems contracts.

Mil-Std-499A on Engineering Management defined systems engineering to be 'a logical sequence of activities and decisions transforming an operational need into a description of systems performance parameters and a preferred system configuration' [DoD, 1974, p3]. Its composite, left-side-of-a-'V' view of 'requirements', 'logical design' and 'physical design' remained central to the US presentation of systems engineering for the following three decades. Its restatement in the two US National Standards in 1994 ensured that it dominated international thinking on systems engineering until well after the International Standard's publication in 2002. The last version of Mil-Std-499, though widely used, was never formally issued since it coincided with the DoD's move to civil standards wherever practicable.

This shift to civilian systems engineering (and other) standardisation came in 1994 when US Defense Secretary Perry introduced a major acquisition reform directive that reversed the traditional DoD preference for Military Specifications in favour of commercial standards [Perry, 1994]. The transfer of responsibility to (US) non-defence bodies to codify systems engineering marked an important watershed for systems engineering that has had long-term consequences, in that hitherto military-only practices began to receive wider attention.

Work in Europe in the early 80's led to Philips in the Netherlands exploring technical process models, including the role of User Requirements. By the late 80's a standardised approach for software systems had been defined by ESA in the Netherlands that was similar to Mil-Std 499 model, though it resolved requirements into distinct classes by separating User Requirements from System Requirements [Mazza, 1994].

The two US National Standards that arose from Mil-Std-499 – EIA 632 [EIA, 1999] and IEEE 1220 [IEEE, 1998] – vied with each other to present superior models, but in both cases they suffered a degree of national, in-built inertia arising from massive, buoyant and therefore vested US defence organisation interests.

A reduced US standards domination traces to ISO's publication in 1995 of a process standard for the life cycle management standard of software-intensive systems, ISO/IEC 12207 [ISO, 1995]. In the absence of a satisfactory system life cycle management standard, this software International Standard was faced with building its own umbrella of systems engineering processes under which it then addressed the management of the software life cycle. A foundation for system life cycle management had been laid.

It was a natural step for ISO to fill the evident need for a complementary systems engineering standard. The US proposed the move assuming it would edit the solution, however competing US camps and a recognition that elsewhere other ideas were emerging, ultimately resulted in Editorial control being given as a compromise to the UK (and specifically to MOD-funded Editorship).

This turn of events opened the door for MOD's embryonic interest in systems engineering to become an influence on the direction of systems engineering codification. Through MOD's Defence Evaluation and Research Agency, the ESA standards had been elevated from purely software systems to systems in general with the publication of the *DERA Systems Engineering Reference Model* [Arnold, 1997]. This model found favour with several Member Bodies of ISO. It matched well with progressive concepts in ISO/IEC 12207 and was easily able to adopt some aspects of IEEE 1220. Though these events were 'in the noise' for MOD as it moved into its major Strategic Defence Review, they proved to be a judicious, though not fully recognised investment return when Smart Acquisition subsequently burst onto the UK defence acquisition scene.

Thus, from the outset of ISO's interests in systems engineering, MOD was a significant, if unintentional contributor to the thinking and development of the ISO systems engineering standard. The ESA model, via the DERA model, exerted a marked influence on the development of ISO/IEC 15288; in the event it proved to be one of three primary sources for the International Standard's development, see Figure 4.4.

It is relevant to this investigation that the development of ISO/IEC 15288 spanned 1996 to 2002 for it coincided with much of the formative evolution of MOD's new approaches to acquisition described in §7.3. The outcome was a measure of interaction between MOD's practical understanding of how to apply newly evolving systems engineering concepts and ISO's intention of expressing systems engineering best practice.

The key to ISO/IEC 15288's superior modelling of process is its explicit treatment of a system throughout its whole life cycle. Importantly, its statement of service requirement and service achievement is separated from product requirements and achievement, and this creates two cardinal viewpoints of a system. As presented in §9.3.2, it also leads to a clarification of the technical meaning of capability.

ISO/IEC 15288's model defines the primary actions of an organisation that are fundamental to the effective creation of products and/or the provision of services in order to satisfy customers – the motivating goal of most business undertakings. These actions stem from an underlying, powerful, universal set of system concepts founded in systems reasoning that imply as much about how management should be conducted and organisations should be structured as they say explicitly about engineering.

ISO/IEC 15288 presents a widely applicable, profiled set of business processes that complement ISO 9001:2000. It provides a model of excellence that defines the system-driven actions of organisations engaged in the creation of products and the provision of services that meet stakeholder needs that ultimately lead to customer satisfaction. When applied with understanding and experience, it leads to a disciplined linking of needs for services to the satisfactory provision of those services. In this sense, it may be viewed as a pragmatic and detailed exposition on the management and engineering of quality.

While several aspects of the standard draw from the thinking and experience of earlier standards, the architecture of ISO/IEC 15288 provides a unique collection of system life cycle processes based on fundamental system principles and a set of concepts that govern their application.

Typically, organisations distinguish different areas of managerial responsibility and action; in concert they govern the organisation's overall capability to trade. The International Standard employs a process model based on the three organisational areas (or levels) of responsibility set out in IEEE 1220: enterprise, project and technical. Within any organisation, a co-ordinated set of enterprise, project and technical processes contribute to the effective creation and use of systems, and therefore to achieving the organisation's goals.

As shown in the contextual layers of Figure 4.3, the outermost business organisation layer is the enterprise level, with its business responsibility to customers and stakeholders inside and outside of the organisation. Here, against the prevailing socio-economic environment of the organisation, the strategic deployment of assets is decided. This results in the organisation's infrastructural capability, e.g. technologies and enabling resources, and in selected projects that engage in conceiving, creating, utilising and retiring systems. The overall responsibility for modelling and managing the life cycle resides at this enterprise level.

Subordinate to the enterprise level is the project level. Here management is concerned with achievement and progress against technical, schedule and cost requirements, whether product or service, and with demonstrating this achievement at the major decision (stage) gates established by the enterprise throughout the life cycle.

Within the project, the technical processes are conducted. These are described in terms of an overarching set of system technical processes that define a common set of technical actions and establish a frame of technical reference. They guide the engineering effort from a systems thinking perspective across the whole life cycle and are the focus of systems engineering action.

Different organisations, and also different areas of responsibility within the same organisation, mutually establish their working relationships and acknowledge their respective responsibilities by making agreements. These agreements unify and co-ordinate the contributions made by separate areas of responsibility in order that they can meet a common business purpose. Agreement processes are the glue in life cycle building and essential components of the International Standard.

This business organisation-oriented frame of reference more effectively guides the system contributions of diverse organisational functions at different stages of the system life cycle. It is a multi-disciplinary view of systems that sees the system technical processes set within a span of business processes that apply to a wide range of organisations, as conveyed in Figure 2.10.

Process execution throughout the life cycle, and over levels of system detail, is dependent on a diversity of assets and skills that are distributed across multiple, international organisations. These organisations see themselves as a part of a single global economy. They need to cooperate in a framework of common concepts in order to manage their contributions to complex products and services with acceptable levels of commercial risk.

ISO/IEC 15288 is also defined to apply at any level of detail in a system's structure. Its infrastructure of system technical processes is therefore expressed in a level-independent way that is capable of being translated into the language and techniques recognisable by parties responsible for different levels of a system's hierarchical structure. The system technical processes thus act to unify the engineering actions of technical specialists who employ implementation-related technologies to build the diversity of heterogeneous elements from whose interaction the overall behaviour emerges.

Viewed in terms of their physical implementation, the system's elements are designed, fabricated and tested using processes that apply to particular fields of engineering technology, e.g. software engineering, human factors engineering, electronic engineering, biological engineering, and these must harmoniously blend with the 'umbrella' of 'technology-agnostic' system technical processes. ISO/IEC 15288 illustrates its unification of the contributions from implementation technology specialisms by reference to three major, characteristic classes of implementation media: hardware, software and humans. A comprehensive set of mappings of system processes into individual implementation technology processes would be unwieldy and, moreover, unnecessary. Other standards and treatises exist to describe the engineering concepts, language, techniques and tools of the individual technologies employed to implement system elements.

The ISO/IEC 15288 system life cycle processes standard [ISO, 2002] was proposed to ISO/IEC JTC1 by the US National Body in 1994. The primary goal specified for the standard was 'to provide a basis for international trade in system products and services'. It proved to be a slow and contentious journey to gain agreement, first with regard to the fundamentals, and then on a raft of detail seen from the different perspectives and concerns of the many contributors and reviewers.

Fundamentally, it recognises that the practice of systems engineering is not confined to a particular stage within a system's life cycle or to a particular level of system detail, nor is it purely the preserve of a specific class of technical actors. The Technical Processes are concerned with technical actions throughout the life cycle. They transform the needs of stakeholders first into a product and then, by applying that product, provide a sustainable service when and where needed in order to achieve customer satisfaction. The Technical Processes are thus employed to create and use a system whether it is in the form of a model or is a finished product, and at any level in a hierarchy of system structure.

Every system is an outcome of an individual life cycle. Consequently, the International Standard does not attempt to describe even the major patterns that emerge from different application domains, business communities, trading circumstances or attributes of a system. It provides a single exemplar model to illustrate the common stages in a simple linear systems life cycle – a view that serves the organisation's enterprise management particularly well. As detailed in Table 9.3, this life cycle model is also well suited to the strategic management of defence acquisition.

The life cycle processes are based on principles of modularity (maximal cohesiveness of the functions of a process and minimal coupling among processes) and ownership (a process is associated with a distinct responsibility in an organisational structure). The functions these processes perform are defined in terms of specific purposes, outcomes and the set of activities that constitute the process.

ISO/IEC 15288 became an international standard in June 2002. After an initial weak showing, ISO/IEC 15288's presence and influence has grown – naturally through recognised benefit, rather than as a championed policy. This has avoided practitioner aversion to any sense of standards indoctrination. Nor has it actively been promoted to displace existing standards; they are treated as alternative views having greater emphasis and detail in some regions of the life cycle. The standard is the basis for the 3rd Edition of the INCOSE handbook on Systems Engineering. This handbook follows the Enterprise, Agreement,

Project and System Technical process structure provided in ISO/IEC 15288. IEEE 1220 moved to better alignment with ISO/IEC 15288, referencing it 88 times in its 2005 revision.

Despite an impending 2008 revision to help accommodate the gap with its software life cycle process companion, there is some justification that the 2002 publication of ISO/IEC 15288 will stand the test of time. This is not blindness to its failings (which are acknowledged in §4.4.3) but to 6 year's experience of beneficial contributions, and an often-voiced opinion that 'revisions' of standards are as likely to bend and bias them to partisan territory as to elevate their merit and value (- the Second Law?).

4.4 The Engineering Viewpoint

4.4.1 A Basis of Systems Reasoning

The order of presentation of a dynamic network of processes could be approached many ways, for instance alphabetically, but it is the logical sequence of information and material transformations, seen with the clarity of retrospection when the life cycle has run its course, that is most frequently employed. This daisy chaining from real world need to satisfactory termination of its fulfilment is the most easily assimilated life cycle representation. It gives rise to the classic waterfall of process representation in Figure 5.10. For ease of comprehension, though at the risk of misleading its users into overlooking the true complexity in life cycles, this is the chosen order of technical process presentation in ISO/IEC 15288. It is not, however, the order adopted in the following analysis of why this International Standard's particular set of technical life cycle processes of a system is so powerful and adaptable.

The crux of this investigation's unfolding presentation of systems engineering processes is its observance of two key issues:

- the manifestation of systems reasoning, as presented in §3, in the logic of a functional models that describes the system life cycle, and that are recognisable in everyday engineering and management conventions;
- the structural simplicity of a realised physical model – in this case the text of the systems engineering International Standard – that can be intellectually mastered yet remain open to dealing with extraordinarily high levels of system complexity;

From their foundation in systems reasoning the logic of technical processes is covered in this section; their corresponding instantiation, and how this is simplified in today's principal point of reference, ISO/IEC 15288:2002, is addressed in the next section; to conclude this engineering view of systems engineering's formulation, key strengths and weaknesses of this International Standard are reflected on.

The engineering of man-made systems revolves around the problem solving of design. Design is founded on the consistent, directed resolution of diverse, partial views of systems, and on the viability of translating these models into reality. The resolution of view consistency is in turn founded on the decision making that leads to concordance between dissimilar, heterogeneous models of intended real-world intervention [*Figure 4.9*] – in Dennett's evolutionary psychology, between dissimilar (though when resolved) complementary stances.

Kant's philosophy of reality is that we only interact with the real world through cognitive models: tractable simplifications that can be manipulated within the limitations of the imagination and be adequately informed within the limitations of the senses. These models reduce reality to a perceived space/time manifold: a basic workable approximation to the dimensions of the real world. Identifiable objects, their spatial presence, temporal existence and mutual causal influences govern how these models play out.

It is to give meaning to this complex, and thereby to predict the likely course of the wavefront of actuality, that Dennett argues for the existence of a primary tripartite cognitive modelling convention. The effectiveness of the mental mechanism that distils and resolves these three models is key to maximising human survival chances. Evolution has therefore filtered and fed back a natural refinement of this mechanism to reason about intention and outcome, and cause and effect, in the system of perceived objects that comprises man's environment.

Though Kant, limited by the scientific knowledge of his age, resorted to a 'mystical' transcendental logic to explain apparent prescience and intuition in predicting actuality and man's interventions in it, it is advanced in §3 that the *a priori* logical component this implies may be no more than 'hard-wired', set-piece reasoning. This has been genetically handed on from ancestral tried-and-tested empiricism (though is undoubtedly further refined and updated according to each individual's path of experience and knowledge gathering).

Based on Dennett's premise, each member of a system of real world objects is represented by three exclusive interpretations: a 'physical form stance', a 'functional design stance' and the 'service effect stance' [§3.5.2]. Causation of an intervention— the processes of creating and using a product – is thus founded on reasoning about object function and about form. The resulting effect of this function and form then derives from their joint interaction with a corresponding composite of function and form of the real world being intervened in [Figure 3.6].

In this way, we use the two stances of function and form to reason about objects in isolation; whether existing objects, i.e., observed present reality, or future objects, i.e., concepts of some anticipated reality. The mental conventions of function and form are paramount to evolving presumptions of a future system (system modelling) and to observing its incremental and verifiable piecing together and existence in reality (system building). Function and form correspond in conception and in construction. The stances that attend to function and form, and the transformations and mapping that reconcile them, thus constitute a primary mechanism of the creative mind.

Models of a rational, projected reality and our perceptions of its built reality thus mirror each other in a complementarity of 'functional design stance' and a 'physical form stance' [§3.5.3]. These two stances are central to the design and also to the build of real world objects, as shown in Figure 4.5.

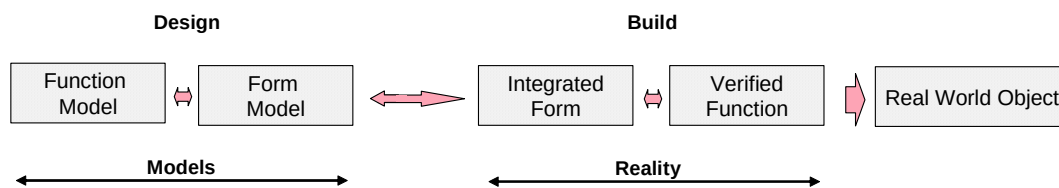


Figure 4.5 The Complementarity of Function and Form Models Mirror Reality

Descriptions of the synthesised cognitive primitives and their actualisation as real-world duals are respectively the foundational information and material work products of systems engineering. They lead to an outcome that is the intended real-world system-of-interest. Devising, relating and resolving these systems reasoning primitives is paramount in systems engineering. A description of how self or team execution of this basic block of information transformation and model building in Figure 4.5 is conducted is at the heart of systems engineering process codification.

Expressed as the transformations that give rise to the objects or work products of Figure 4.5, four corresponding processes are paramount to the conduct of systems engineering, as shown in Figure 4.6. The primary interactions that logically bind these four processes are schematically conveyed by the emphasis of the arrows. In line with the practices advanced in this text, the ISO systems engineering process terminology is used in this figure, though as considered later the rationality of the design process names to the left is open to question.

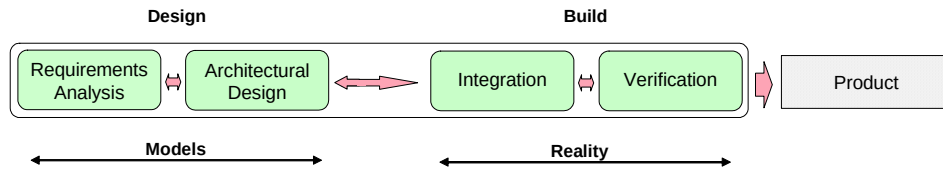


Figure 4.6 The 'Architecture Design Cause' Processes

The four processes (two pairs) constitute a fundamental activity group in which design analysis interacts with design synthesis and then feeds into, and responds to the constraints of, build synthesis interacting with build analysis. The design processes are concerned with creating system models; the build processes with creating system reality.

The processes in Figure 4.6 'cause' a product or real-world object to come into being according to a designed form or architecture. They are for systems engineering the essence of Aristotle's Formal Cause, equated in §3.3.2 to an 'Architecture Design Cause'. This group of processes brings order and structure to the system and can be regarded as a layer of system architecture design and build: the quintessence of systems engineering activity.

The Final Cause in process terms arises from a strikingly similar set of processes. These are the transformations that address service rather than product. They arise from the teleological drivers of need, intention and intervention. They are a transposition of the processes in Figure 4.6 according to the different concerns or viewpoint of service. It is convenient to omit the leftmost process, and to thereby assume that the recognition and analytical justification of need (akin to service/operational analysis) is an incipient engineering trigger. Originating from this, the required intervention is formalised according to a defined environment, which is commonly described in terms of an architecture of notional systems (many of which will comprise the system-of-interest stakeholders; see Figure 10.17). This transformation is termed the definition of stakeholder interests and requirements, i.e. the Stakeholder Requirement Definition Process.

With regard to assembling the capability to sustainably deliver the product's services, Integration and Verification in Figure 4.6 are given new identity according to their service role, being titled Transition and Validation respectively. They install and test the coming together of every element of a capability that is necessary in order to deliver intervention over the required duration.

These Final Cause processes, likened to a 'Service Capability Cause' in §3.3.2, are illustrated in Figure 4.7 on the top row. They provide as their output the capability to provide the required service.

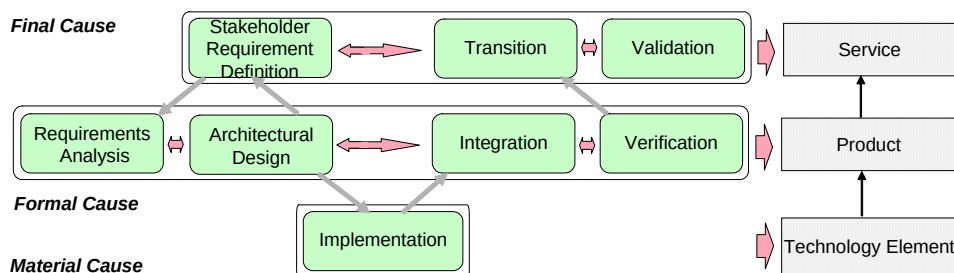


Figure 4.7 The '3 Causes' Structure of Systems Engineering

As emphasised in §3.5.1, a system may be viewed either as a product whose internal composition interacts to create novel emergent properties at the system's boundary or it may be viewed as the services delivered at the system's boundary into an operational environment. This important separation of viewpoints applies throughout the processes of ISO/IEC 15288 and distinguishes the service capability (Final Cause) layer and the

architecture design (Formal Cause) layer in Figure 4.7. It has profound consequences on the purpose, ownership and definition of systems engineering activity detail that is aggregated to form a coherent, rational and utilitarian set of processes.

This duality of viewpoints of a system leads to clear distinctions respectively between:

- stakeholder requirements and system requirements;
- validation and verification;
- transition and integration;
- measures of effectiveness and measures of performance;
- acquirer concerns and supplier responsibilities.

The separation of stakeholder requirement in the users'/acquisition authority domain from the system technical requirement in the system creators'/supplier domain is essential if pre-emptive definition of solutions is to be avoided. In ISO/IEC 15288, this leads to requirement work product outputs from the Stakeholder Requirement Definition Process and the Requirements Analysis Process having distinctly different natures. Respectively, these outputs clearly discriminate between the service need in an operational environment and the requirement for a conformant product solution in the domain of the creator organisation.

Similarly, the responsibilities for the verification of correctly integrated and supplied product, according to defined measures of performance as opposed to the validation of correctly installed service capability in conformity with measures of effectiveness, are also separated. This better matches intra- and inter-organisation responsibilities as systems move through development and production to their in-service lives.

Leaving aside the Efficient Cause (the agent of process execution), Aristotle's Causes are completed by the Material Cause. This is served by a third level of process associated with the 'raw material' from which a thing is produced 'as from its parts, constituents, substratum, or materials'. Again the four basic processes are employed but they are abstracted according to systems engineering concerns to form a single composite, implementation technology-oriented process. They sit astride the detailed fabrication actions that 'process' raw materials. Fabrication is thus the fulcrum of the systems engineering "V" that divides design from build, as conveyed in Figure 4.14 and then in Figure 5.9.

The design and build of product architecture in Figure 4.6 is thus instantiated in two different ways to represent service delivery viewpoints and element implementation viewpoints. In this way the '3 Causes' are completely represented by a minim but sufficient set of system life cycle processes, with systems engineering practitioners effecting their execution according to the Efficient Cause.

Figure 4.7 therefore conveys three layers that correspond to different viewpoints or concerns of 'effecting parties'. Each layer generates functional models that explain the logic and relationships of the 'hows', and models of form that explain the composition and structure of the 'whats'. For the whole system-of-interest, this amounts to its architecture described and rationalised down to a particular level of structural detail and beneath which implementation detail ceases to be of due interest to the 'effecting party'.

Whatever the degree of abstraction of system detail, these same conventions apply. With each higher order of detail (that is downwards in Figure 4.8) there is a geometric growth of model information and a need to employ the cognitive defence mechanism of systems reasoning that counters this, i.e., by simultaneously re-scoping and commensurately re-scaling to form contiguous packages of information. Successive steps of reduced abstraction rapidly become unworkable if a "divide and conquer" strategy of modularity is not rigorously pursued.

Figure 4.7, being a matter of viewpoint, is thus a composite of comparative descriptions or views. 'Looking' to a coarser level of structural detail is looking towards the source of requirements. Looking to finer detail is looking towards implementation technology detail. No matter what the scope and scale of consideration, this is the rule of consideration.

This recursive 'downward' consideration leads to ever finer levels of detail in a system's structure. It is a primary mechanism by which system complexity can be reduced to

comprehensible and manageable proportions. Conversely, the route to synthesising novel, overall system-of-interest properties from a set of dissimilar system elements is performed by applying processes recursively 'upward' to achieve a fully integrated entity.

In common with fundamental system notions, the design of architecture is a recursive concept. It leads to a concise, mutually consistent set of description of function and form (and, in some books, a description of rationale also, [IEEE, 2000]) that may be expressed according to any degree of granularity and scaling. In practical terms (and arising from systems reasoning) different scales, and thus different layers of architecture consideration, discriminate different emergent properties. As §10.4.1 points out, the degree of recursion is governed by responsibility, risk and due interest.

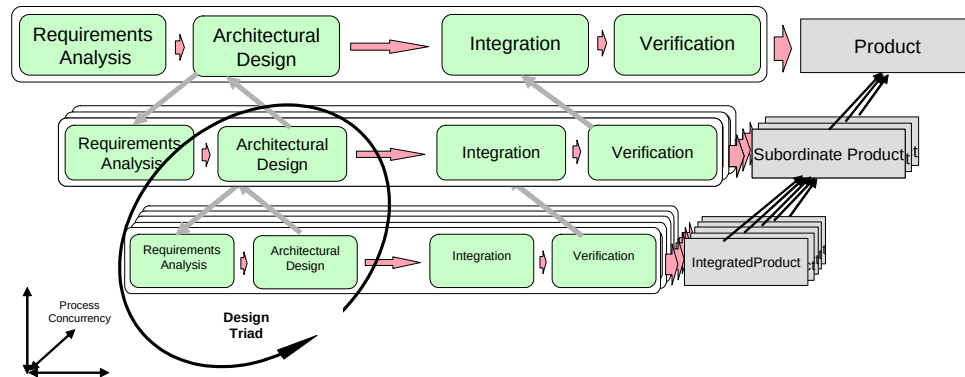


Figure 4.8 Reduction through Iteration and the Design Triad

Figure 4.8 conveys this concept of recursion, reduction and synthesis that has generally been weakly portrayed in most systems engineering representations. It shows how reduction and synthesis is achieved through downward or upward recursive application of transformation; to the left on the models, to the right on their implemented reality.

A powerful relationship stands out from this (shown circled). It is the triad of strongly associated transformations that provide the engine of design in this network of technical processes. Advanced as the System Design Triad, it is conveyed in Figure 4.9 as it appears at the top left of the '3 Causes' representation of systems engineering in Figure 4.7, i.e., with the super-ordinate architectural design process shown according to its instantiation from the service viewpoint.

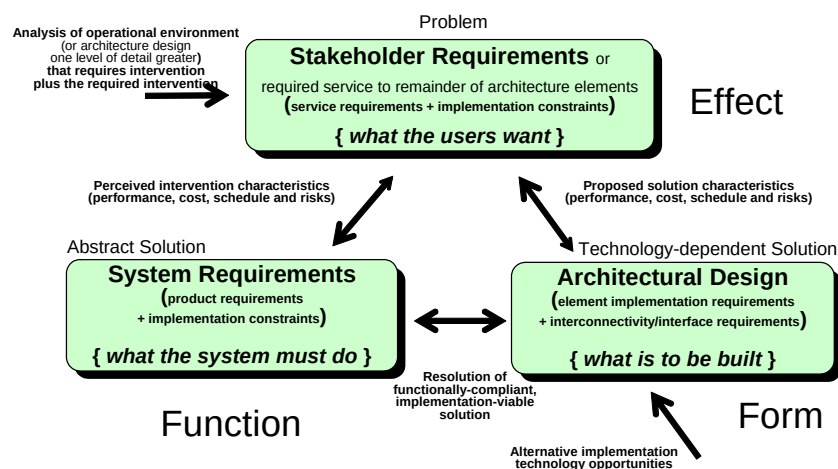


Figure 4.9 The System Design Triad

Figure 4.9 iconically conveys a notion that runs through this investigation. It is the distilled systems engineering representation of the three stances.

As seen in the consideration of practices in later chapters, the design triad is the powerhouse of man-made system creation and in varying forms is paramount at all stages of the life cycle and over all scales of interest. It is the quintessence of the technical transformations and work products. It is the creative engine of engineering.

Since, as emphasised in Figure 4.6, models and built reality are mirror images, so it is only natural that the design triad is reflected in reality by the correspondingly related threesome of assembly and confirmation transformations shown in Figure 4.10. As in Figure 4.9, the brackets indicate the nature of completed work product produced by the transformations, and the arrows show the emphasis of the logical relationships between the transformations.

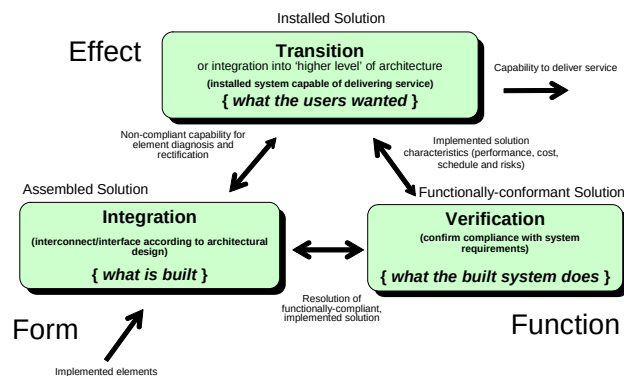


Figure 4.10 Materialising the Stances of the Design Triad

4.4.2 Minimising Model Complexity

A difficulty encountered when dealing with any system, and therefore also with a system of standardised systems engineering processes that governs them, is how to bound descriptions so that they do not digress into repetition of the same common concepts, and ultimately into unnecessary and potentially confusing detail. ISO/IEC 15288 tackles this four ways.

Firstly, it focuses on a single, bounded object which it views as a system during its complete lifetime. To uniquely clarify what is being referred to in an infinity of possibilities, the term system-of-interest is used to identify the object that the International Standard is applied to. A system-of-interest is most effectively defined by boundaries that encapsulate meaningful need and practical solution.

Secondly, it does not attempt to describe the system-of-interest at all levels of compositional detail or granularity. Rather, it employs a single set of transformations that apply at any level of structural detail. It then defines a risk-based terminating criterion that avoids the necessity for concern with ever finer detail.

Thirdly, it considers the system-of-interest life cycle in terms of distinguishable periods, termed stages. The common set of life cycle processes is described in a manner such that each process is applicable in any stage, irrespective of which organisation or party may be exercising responsibility for that stage.

Fourthly, it identifies the interactions with other systems that are essential to the system-of-interest's progress through its life cycle but are not part of the environment in which the system-of-interest's services are required. These systems, although necessary, are ancillary to the purpose of the system-of-interest's services and are viewed as enabling systems.

In his *Methodology for Systems Engineering*, Hall [Hall, 1962] outlined the concept that systems engineering has relevance to the whole life cycle; defined in his terms to be 'problem detection', 'problem definition', 'design', 'manufacture', 'use' and 'obsolescence'. Even though this was reiterated by others [Blanchard, 1981], a mind set that systems engineering was a

development-based discipline became instilled in the minds of many of its practitioners and also in DoD standards. Progressive steps of US codification re-affirmed this notion and it led to a development-centred representation of a system that could be confusing, if not logically questionable.

It was fully recognised that whole-life application of systems engineering was important, but this was predicated on a view that it was the preserve of systems engineering practitioners who masterminded the whole life cycle from the 'design stage'. To overcome limitations of this development-centric view yet accommodate whole-life thinking, the boundary of the system under consideration was extended beyond a system-of-interest to also include its integration with the different environments (with the exception of its operational environment) encountered during its lifetime, e.g. its manufacturing system, its support system, etc..

Defining a system in this way, for example as in IEEE 1220 [IEEE, 1998], had two adverse effects. First, it enforced the notion that systems engineering is the preserve of a specific class of engineer working within the confines of one stage of the life cycle. This is a deeply rooted view. The contrary notion, that to some extent most engineers undertake systems engineering tasks throughout the life cycle, has been taking time to instil. Second, it led to an excessively complex representation of system composition and structure. This then required a matching complexity in the codified description of its associated systems engineering actions. As a result standards could be unnecessarily convoluted and somewhat tedious. It made systems engineering so much the harder to readily define.

The resulting models in these standards are not wrong, but they are built around a design focus that governs actors concerned with other regions of the system life cycle. As described in §3.6.4, the positioning of the hub of a model and its degree of self-centeredness then impacts its region of usefulness.

To achieve the supremacy of this development focus and yet accommodate the importance of the whole life cycle, the definition of a system becomes complicated, unnecessarily labyrinthine and for many confusing. One example of this can be seen in Figure 4.11, [ibid, Figure 2].

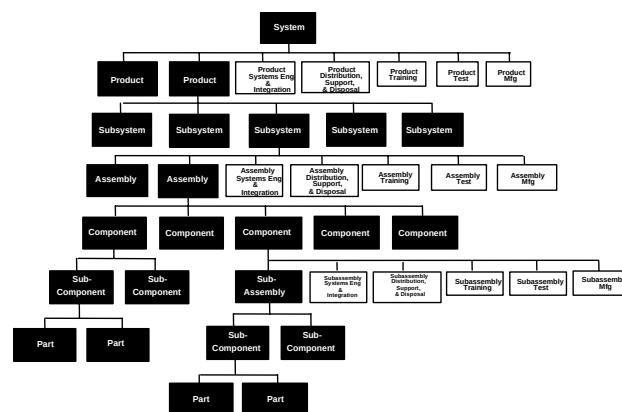


Figure 4.11 IEEE 1220 :1994 Typical System Composition

A system representation of this type falls between two stools. It conveys (in dark boxes) an instantaneous view of ever-reducing compositional detail of the system as a product together with its temporal dependency on other (non-concurrent) products (white boxes) that constitute the successive environments encountered throughout the life cycle (excluding the operational service environment). It is a consequence of a development-dominated view of systems engineering. It contains notational ambiguities that have misled rather than clarified system structure and through life management. Despite this shortcoming, it has internationally strongly influence systems engineering procedures and agreements for more than a decade, and its legacy lives on in many major organisations' procedures.

Given the origins and principal thrust of systems engineering, this development dominance is understandable, but it has introduced an obstacle to comprehensively defining systems engineering. It sets out boundaries and constraints that distort holistic contributions to holistic system solutions.

This has failed to serve systems engineering's best interests, making it appear confined and arcane in its application. It required the state-invariant definition of system-of-interest, as used in the International Standard, to moderate this influence.

Matters were compounded by the packaging of systems engineering into a unified process – the 'systems engineering process' – rather than considering it as a network of technical processes dynamically interacting to create high-complexity systems. This unified process packaged the discipline into a discrete and potentially discretionary engineering action employed 'when the going got tough', and conducted by practitioners termed systems engineers who otherwise did not impinge on the general design community.

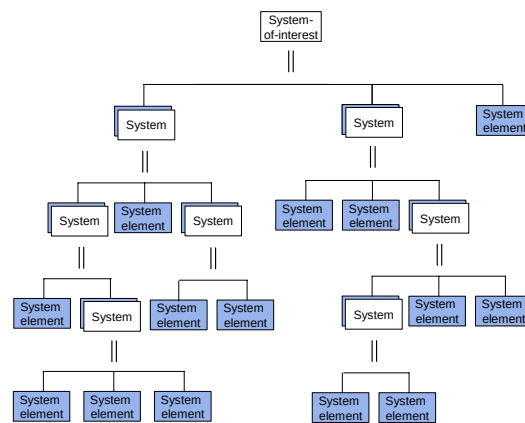


Figure 4.12 The Recursive Composition System Model
[Contributed to ISO/IEC 15288]

A distinctly different approach accommodates life cycle interactions with the system representation in Figure 4.12. It applies to all stages of a system-of-interest as it is created, used, modified or disposed of. Relationships with each environment it will encounter during its lifetime are modelled by a sequence of interactions at the system-of-interest boundary. This permits a much simpler, state-invariant composition to represent matters. Viewed either as a functional or physical composition, it can be employed to describe a hierarchy of detail down to selected system elements. The repeated superimposition of Figure 3.4 to create Figure 4.12 illustrates the recursive nature of system architecture.

At the foot of each path of decomposition/composition is a discrete, essentially homogeneous part of a system-of-interest that mutually interacts with all other elements of the system, thereby contributing to the system properties and characteristics. A system element conforms to two criteria:

- for the party responsible for the whole system-of-interest, it is the lowest level of immediate interest and direct responsibility in any branch of the architecture;
- from a system-of-interest perspective, it appears to have an implementation technology specific nature.

The relationship between the system-of-interest and its complete set of system elements can typically only be resolved in a single step for the simplest, most highly precedented systems-of-interest. For more complex systems-of-interest, a prospective system element may itself need to be considered as a system (that in turn is seen to be comprised of its system elements defined in yet greater detail) until a complete set of system elements can be eventually be defined with confidence, i.e., the grey boxes in Figure 4.12. In this manner, the system life cycle processes are applied recursively to a system-of-interest to resolve its structure to the point where understandable and manageable system elements can be

implemented or reused, or acquired from another party, and then integrated to form the complete system-of-interest.

Two important issues are incorporated in the schema of Figure 4.12 (and 3.4) by the || sign. The first is an epistemological issue; namely, that if the whole were to be identical (=) with a summation of the constituent elements, then knowing the meaning of the whole would be indistinguishable from any meaning gained through observations individually of all the constituent elements. This would violate a prime axiom of systems.

The second is a physical issue that relates to the recursive application of Figure 3.4. That is, the || sign conveys a discrete transition in the level of abstraction. In Figure 4.12 this equates to progressive, incremental resolution of structural detail down individual paths of composition. System composition thus equates to the aggregation of the lowest order member in every path of decomposition/composition. Ultimately, system-of-interest composition is reduced to all system elements in every path. System detail can be represented down to whatever level of architecture detail is required by one party in order that it may confidently communicate a rigorous understanding to other parties.

This resolution of system structure results from dynamic application of the set of system life cycles processes and, despite the existence of strategic patterns, the specific path followed differs on each occasion. Each life cycle process can be invoked, as required, at any time throughout the life cycle and there is no definitive order in their use. The detailed purpose and timing of use of processes are influenced by multiple factors, including social, trading, organisational and technical considerations, each of which may vary during the life of a system. An individual system life cycle is thus a complex system of processes that possess concurrent, iterative and recursive characteristics.

The concurrent use of processes permits different paths of decomposition to be explored with some measure of independence and concomitance (the z-axis of Figure 4.8). Design actions and preparatory actions for building a system may also be conducted with substantial autonomy (the x-axis of Figure 4.8). Once defined, system elements are then designed in parallel under different project responsibilities.

Autonomous, concurrent execution of processes in a mutually dependent system of processes is, however, a relative concept. Overall, congruence and synchronisation depends on repeated comparisons and adjustments. It is this iterative application of processes that allows the minimal process set to achieve model concordance across at same level of structural detail. This is essential for the progressive refinement of work products at a particular scale of consideration.

Recursive use of processes at adjacent layers, as in Figure 4.8, enables system models to be resolved at successive levels of structural detail. This hierarchical resolution largely revolves around the achievement of concordance sought in the triads of Figure 4.9 and 4.10. In effect, the output(s) of executing of the processes in Figure 4.6 have the same set of transformations applied to them, and so on, whether conducted according to the precepts of analytical decomposition or holistic synthesis.

In this way the minimal meta-model of a system life cycle processes, through dynamic connectivity and through concurrent, iterative and recursive interaction, can geometrically expand to form an avalanche of process complexity, as Figure 4.14 suggests. There are as many individual instantiations of this time-dependent minimal process model as there are man-made systems created. Given such diversity and complexity, it is fundamentally the patterns of process structure detectable at different scales that permit this Gordian challenge to be cut through. It is these patterns that are the subject of *Life Cycle Classification* in §5.4.

The changing nature of the influences on the system (e.g. operational environment changes, new opportunities for system element implementation, modified structure and responsibilities in organisations) requires continual review of the selection and timing of process use. Process use in the life cycle is necessarily dynamic, responding to the many influences external to the system.

ISO/IEC 15288 is not intended to treat the whole span of business processes shown in the abscissa of Figure 2.10 with similar emphasis. Its scope omits many enterprise processes and those project processes that are only weakly linked to the system nature of the products

and services being traded. The detail of process treatment is akin to Figure 4.13. This is notable by its almost complete omission of technical processes that describe implementation technologies such as electronics engineering, software engineering, mechanical engineering and human factors. These implementation technology-dependent engineering disciplines are responsible respectively for implementing the varied elements, and thus for contributing the functional diversity from which complex system characteristics emerge.

The descriptions and provisions in the International Standard that relate to implementation technology-dependent engineering disciplines are, by and large, left to the growing number of other standards – though as yet these are not always treated in a manner that is compatible with the balanced, pan-system viewpoint of ISO/IEC 15288. Nevertheless, through its definition of a single Implementation Process (using hardware, software and humans to exemplify all possible implementation media open to the system creator), the International Standard acts as a common link to the multiplicity of specialist disciplines, each of which is characterised by terminology and concepts that apply to specific implementation media, to their associated technical skills, detailed practices, fabrication techniques, and engineering methods and tools.

The profile of ISO/IEC 15288's scope shown in Figure 4.13 illustrates this deliberate hand-off to dedicated implementation technology standards. It has important benefits, for:

- it avoids the danger of duplication or inconsistent overlap of content with a multiplicity of technology specific standards,
- it constrains the size of the model and thus enhances its effective application,
- it defines a standard interface to implementation technologies that provides generality for other standards developers,
- it accommodates the n:1 relationship looking from each individual implementation technology towards systems engineering by defining the simpler generality of a 1:n relationship looking in the opposite direction.

ISO/IEC 15288's technical processes are thus concerned with system detail down to the level of acquisition/supply risk/benefit that often equates to when individual implementation media practices (such as software, electronics, biology, hydraulics, mechanics and human sciences) are employed.

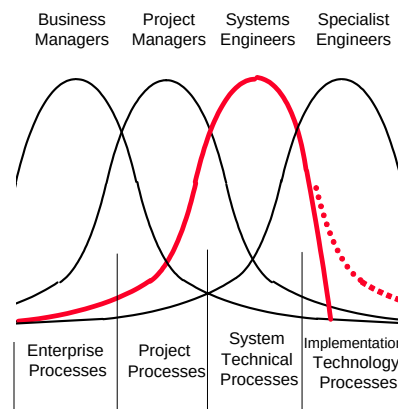


Figure 4.13 The Scope and Profile of Business Processes in ISO/IEC 15288

In a complementary way engineering standards in which overall system properties are considered with particular emphasis throughout the life cycle and all levels of structural detail can also be linked to this overarching process model. Thus standards dealing with properties such as safety, for example IEC 61508 '*Functional Safety - Safety related systems*' [IEC, 2005], can in their future revisions align more closely with other system views.

Seeing themselves as distinct, many technologies and engineering disciplines have felt obliged to develop their own life cycle and process set, thereby contributing to stovepipe thinking and actions. ISO/IEC 15288 offers a framework of common system-level thinking and action, and meta-processes that can be employed with interpretation by any technical specialist. They can be transformed to levels of functional and physical structure at which methods and constraints of implementation technology or application domain begin to dominate the ways of thinking, perceiving and acting.

The International Standard thus establishes a yardstick for standards that define the life cycle management of system elements that are created using specific engineering technologies and disciplines, i.e., the transformations/work products that apply to the elements at the base of Figure 4.14

This lowest-order set of processes comprises the four transformations in Figure 4.6 (but treated as activities in ISO/IEC 15288) that constitute an Implementation Process. This figure conveys how these four fundamental processes are employed recursively, starting from the initiating case of the system-of-interest and eventually individually terminating down each path when the conditions for defining a system element have been met.

The delivered objects from each layer of transformation are shown to the right of the figure and these have a relationship that builds the architecture representation of Figure 4.12.

A compatible setting of technology-specific descriptions of life cycle processes in the overall systems engineering context leads to implementation technology-oriented standards such as the International Standard on software life cycle process management [ISO, 1995].

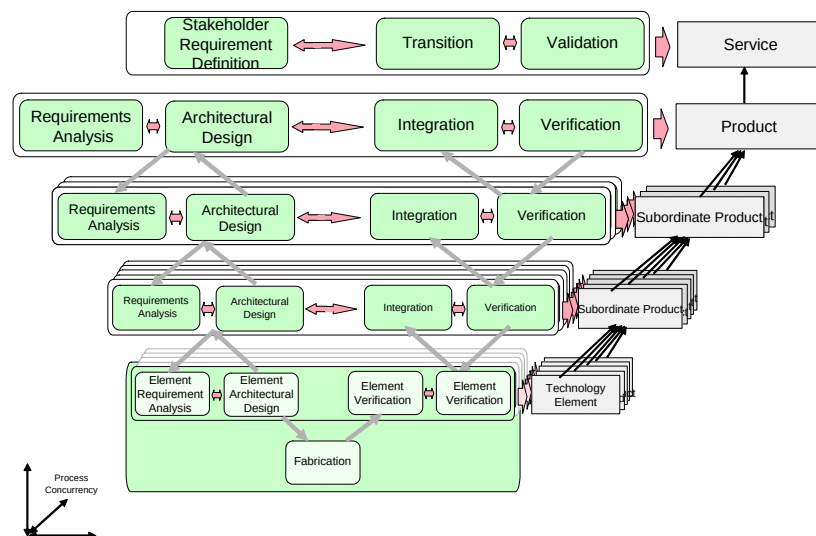


Figure 4.14 Concurrent, Iterative, Recursive System Life Cycle Process Model

The simplifying power of the enabling system is considered further in §5.3.3. However, its value in minimising the complexity of the process model arises from the ability to apply the International Standard to each enabling system when, in its own right, it is viewed a system-of-interest. In this way, the labyrinthine complexity of layers of system-of-interest/enabling system interdependencies, such as in Figure 4.11, can be encapsulated within units of interacting organisational responsibilities, each unit's system have a defined interactions with the system-of-interest.

4.4.3 Process Model Limitations

As emphasised in §3.5.3 models are just that – models; they only approximate to the reality that they are seeking to describe. The challenges faced in codifying systems engineering

according to the conventions and limitations faced in international standardisation illustrate why knowledge, experience and skill are essential adjuncts to the provisions and guidance found in standards.

The acceptable presentation form, development protocol and ratification conventions that characterise most publicly developed standards all place limitations on what can be achieved. ISO/IEC 15288 was no exception. It was subject not just to a stock set of limitations, but also to its own particular set of constraints and obstructions.

Any international standard that sets out to define best practice is faced with building a model that will always fall short of the ideal. The goal is thus not an idealised model but a workable model – a practicable set of compromises that are usable and interpretable in day-to-day application. Compromise and constructive ambiguity are obligatory ingredients of a successful outcome.

Process standards are reference models struggling to represent the full complexity of reality and therefore exhibit distinct limits to their validity. They work well in some situations and, pushed outside the region of their main focus, they rapidly display weaknesses. The technical metaphors they contain need to be understood and respected. Other technical metaphors exist and need to be used in conjunction. Understanding which horses suit which courses is a skill of judiciously applying engineering models.

To compound matters, the ISO/IEC 15288 model was the product of fragmented, often discordant contributions. In varying detail, the users of any business standard are all functions in the organisation – one story, many audiences; a complex combination of viewpoints. This multiplicity of viewpoints is both an enriching feature and a bane of systems engineering. Arriving at a single international view that could accommodate all involved parties was destined to dissatisfy everyone to some extent. Designers, reviewers and users all wanted to see themselves in the model – and ideally at its centre. Cynically, the task then regressed to “spreading unhappiness as evenly as possible”.

Superficially ISO/IEC 15288's process model resembles the IEEE 1220 and EIA-632 models, with some of the basic transformations that are defined appearing similar. This led to many practitioners approaching a new model preconditioned by progenitors in other systems engineering standards. However, there are several radical introductions of principle in ISO/IEC 15288 compared with its predecessors. It could have contained more, but international voting led to compromises, distortions and omissions (as it often does) that fell short of the advances and rigour that in theory might have been attainable in the 2002 publication.

A reality emerged during the rounds of voting about how radical the model could be. Too large a step for the general practitioner mind set to accommodate or for organisations to adapt to in their procedures and its widespread adoption would fail. Too small an advancement and the existing US National Standards would continue as an unchallenged and comfortable way forward. The evidence of the International Standards unanimous ratification suggests that the compromise was about right. Nonetheless, weaknesses are evident and they require attention:

- In common with most ISO standards, ISO/IEC 15288 is principally implemented using language – in this case, a linear sequence of words trying to capture an iterative, recursive, dynamic reality. The order of presentation and build up of concepts was a delicate and experimental feature of the development.

Language brought other constraints. Most of the good words had been used in other models, often with different and (in the context of ISO/IEC 15288) misleading meaning – misconceptions often took time to unearth. Even achieving compatibility between British English and American English presented problems as a divided common language and cultural differences led to cryptic misunderstanding and erroneous interpretation. Ultimately the International Standard also required unambiguous translation into the obligatory French and Russian and was also translated into German, Japanese, Korean, Portuguese, Spanish, and Swedish. To counter some of these language difficulties, the selection of neutral and translatable terms required the adoption of some initially unfamiliar terms, and (more challenging) a degree of redefinition for some traditional words. The presentation

and resolution of these niceties in the Glossary was a drafting/commenting/redrafting ping-pong that ran the full six years of the International Standard's development.

Furthermore, graphical/schematic representations may appear to offer rich modelling opportunities, but they bring with them implied notations and conventions which have to be interpreted or tediously defined. In the event, ambiguity knowingly and constructively left in the text was matched by constructive ambiguity in the graphics notational.

- No tool-based modelling to discipline the standard formulation and detect its inconsistencies;
 - The most condemnatory factor in the development of ISO/IEC 15288 was the complete failure to employ a discipline modelling approach during its development. Immaturity of concepts and flexibility during the definition of its development are perhaps justifiable, mitigating circumstances. This left a trail of minor structural, content and lexical distortions; none serious, all having a negative impact on perceived quality. The failure of systems engineering practitioner representatives to rectify this in follow-on work is inexcusable.
- No codification of work products and their correlation with transformations;
 - Unwisely observing the precedence of ISO/IEC 12207, there was no attempt to define the input and output work product of processes; not even as 'work in the margin' in order to check the consistency of the defined transformation. Work products were implied throughout but never formalised. The consequent inability to cross-check the transformation definition against the dual view of work products inevitably reduced the resulting quality. Again, the mitigating argument was the entry into territory that could risk contention, and thus the International Standard's development.
- No clear presentation of the controls that regulate the execution of process;
 - The mutual dependencies and logic of process execution follow patterns of system control were only cursorily expressed. This logic could have been described in advisory text in order to illustrate features of life cycle model building. The guidance in the subsequently published Technical Report [ISO, 2003] concentrates on one traditional pattern of process interaction and life cycle management, but provides little on the theory or principles that assist the building of life cycle models – the ultimate purpose of the International Standard. There needs to be much better relationship with control criteria:
 - value delivery;
 - technical achievement criteria;
 - readiness measures;
 - decision criteria;
 - cost models;
- Weak accommodation of the service viewpoint;
 - Partly because the software viewpoint taken in ISO/IEC 12207 essentially saw no need to address the service delivery period of the life cycle, a balanced and effective presentation of 'the other half' of the system life cycle was only seen by many reviewers as important late on in the development. In addition, a majority of involved parties did not identify with service as fundamental to their roles. The outcome was many compromises necessary in order to get an acceptable balance of product and service in the provisions of the International Standard.

Whilst a system viewed as a product is an essential focus for engineering, for most organisations the comprehensive response to user need is service: actual intervention in real-world situations. It is essential to see the span of system existence as both product and service (and to refine this distinction further in §9.3 to

include capability) in order to gain a comprehensive, through-life understanding of engineering endeavours. A life cycle management culture breakthrough, partially achieved first with the commencement of ISO/IEC 12207 for software engineering in 1989 and again with ISO/IEC 15288 in 1996, has had a big influence on systems engineering thinking. Whilst not fully wholly satisfactory in its treatment of coherent system life cycle (through life) management, ISO/IEC 15288 has made a permanent change is the balance of emphasis regarding systems engineering's application.

This life cycle management thinking was particularly emphasised in the UK in 1998 as a result of MOD's stress on through-life management for effective defence systems acquisition. Its shift from the equipment-centric view of a system to encompass its capability and service states has opened an opportunity for systems engineering application in MOD that has yet to be fully seized. Nevertheless, its impact on the formalisation of requirements according to service and product descriptions has been a dominant technical aspect of Smart Acquisition. This distinction in ISO/IEC 15288, and the consequential related shifts in how other aspects of systems engineering must be viewed, has reverberated around national views.

- No recognition of a system state of capability;
 - Contributors from outside the defence community saw no significance in the concept of capability, and many from within it had not read the changes that were occurring in the thinking of the defence acquisition authorities. The clandestine but disciplined introduction of capability in the text, as summarised in §9.3.2, was largely overlooked by reviewers and subsequent users.
- Inconsistent, even conflicting and confusing interpretations and expression of sequential function and concurrent object viewpoints of the processes model;
 - Many (probably the majority) of systems engineers think of systems engineering in terms of a functional decomposition model rather than an object-oriented model. All ISO/IEC 15288's predecessors were biased towards interpretation as functional models. In contrast, ISO/IEC 15288 is presented in a manner that far better accommodates an object-oriented interpretation. Neither is wrong, neither right, but as in all models, each is appropriate to situations. Without object orientation, extreme complexity cannot be accommodated by the model.

The object-oriented paradigm considers a collection of interacting objects (in this case processes), with each object viewed as an independent entity with a distinct role or responsibility, whereas a functional paradigm considers a sequence of stateless function evaluations.

The mapping between behaviour and structure in systems engineering has always depended on object-oriented thinking for this is how real-world objects exist. As Booch put it, it offers 'build abstractions of the problem space that permit a more balanced treatment between the objects and operations that exist in a model of reality ... functional methods do not adequately offer data abstraction and information hiding, are inadequate for problems with natural concurrency and are not dynamically responsive to changes in the problem space' [Booch, 1987, p.13].

Object-oriented decomposition closely matches our model of reality. On the other hand, the functional decomposition is only achieved through a transformation of the problem space... [and is thus] imperative in nature. It concentrates on the actions of a system and is silent about the agents that perform or suffer these actions [ibid, p.16]. It is the prevalence of users to declare their required effect in terms of real-world, sequential action that biases much early function thinking towards sequence. Indeed, Figure 4.9 suggests that systems engineering has an equal dependence on sequential functional analysis expressions and on concurrent object-oriented synthesis.

A single textual model that could of necessity accommodate both of these viewpoints was only possible by the use of ambiguous terminology and text. The dominant, legacy viewpoint during development was of sequence according to the

terminating logic of the waterfall model of system processes. The resulting text was most readily interpretable as sequence; the actual execution of instantiated complexity was less evident. This remains a difficult challenge to resolve using just one textual model with negligible graphical support, as can be inferred from the investigation into models that describe the architecture of any system makes clear in §10.4.3.

- A failure to accommodate architecture, ontologically, semantically or pragmatically into process descriptions, such that it aligns with authoritative, prevailing views;
 - The failure to employ concepts of architecture to best effect in the process description was perhaps an understandable oversight. The sense of architecture inherited from ISO/IEC 12207, and for that matter from predecessor systems engineering process standards, was restrictive compared with the architecture modelling techniques in the IT and related software communities that were co-evolving during the period of the International Standard's development. The not unreasonably focussed development activity did not pick up on alternative ways to consider the meaning of architecture. Accommodating these software/IT developments would certainly have been a 'bridge too far', and the resulting weakness has to be attributed to the inevitability of cloistered, concurrent technical developments. The failure to pick up on this in ensuing work on the International Standard has no such legitimacy.
- Misleading process titles;
 - ISO project requirements for maximum compatibility with ISO/IEC 12207 and the legacy of titles that had served their purpose for software life cycle modelling both placed tight constraints on options available. Early adoption of much of the process titling used in ISO/IEC 12207 (which has negligible service content) quite reasonably went unchallenged during development. The comfort factor of familiarity was an attraction, but it also predisposed practitioners to past ways of thinking that inhibited some of the options that could have been followed in the process modelling. The continuance of this titling in ongoing developments, again because of ISO/IEC 12207, demonstrates the difficulty of shaking off unwelcome legacies.

CMMI users expected to see CMMI processes in it; for example, those with quality assurance and data gathering responsibilities expected to see a Measurement Process.
- Avoidable inconsistencies in the structure and abstraction of activities across processes;
 - The lack of a disciplined model of the standard's system of processes – something seasoned systems engineering practitioners should advocate whatever the nature of the system – and also the 'death by a thousand cuts' reconciliation in the text as a result of repeated cycles of international comment dispositioning, led to incongruities in scope, content and presentation style, and distortions of emphasis across the processes set. These compromises bought international voting support at the expense of model cohesion.
- Few criteria to frame the structuring and assembly of life cycle models.
 - The ultimate purpose of the International Standard is to build life cycle models that guide projects in their creation and application of systems. How this is actually achieved is almost a blind spot in the provisions and descriptions. The single exemplar model of a life cycle is a valuable case study, but the theory and principles behind life cycle models has to be inferred rather than referred to. More advisory text could have been added but was seen to be in the realm of guidance and thus out of scope. The minimal set of principles obliquely referred to throughout the text could have been marshalled and presented to better effect.

The exemplar stage model has weaknesses. In-service would bring linearity and avoid the awkwardness of concurrency. Support is a major aspect and highly important to MOD (and many other undertakings) but the model is confusing.

Transition could be a stage, which may then explain the quiescent state of service as capability and the transitioned state of product(s) + services also as capability. It would also handle the weakness of service continuity/migration. Weakness of transition, should refer to migration of service for continuity (e.g. EA/IT)

In the face of these unfulfilled ambitions, several key next-steps were left to a revision to be conducted after a sufficient and stable period of experience of use. In the event, a rush to change a number of second-order prejudices and to incorporate concessions that narrowed the gap with its predecessor software life cycle cousin pushed most of these pressing enhancements to one side. This haste has barred taking advantage of consensus feedback from representative, widespread or institutionalised use of the International Standard. Weaknesses in the 2002 publication appear set for repeated dissemination.

However, notwithstanding its shortcomings, the 2002 publication has been a welcome and major step forward for systems engineering. Its deficiencies take a back seat to its advances. Its influence has been profound, not least on how management perceives systems engineering in the context of projects and the enterprise. The setting of the project and its explicit linking into the endeavours of interests of enterprise have been recognised as important features in the advancement of the systems engineering discipline, and in its recognition as a cardinal element of business process.

Given the potential diversity for systems engineering interpretations, it is perhaps the International Standard's stabilising baseline that is its most important contribution to the discipline; its imperfections are of a second-order, tolerable nature. There is for the first time in 60 years, a single, international source statement of basic systems engineering process definition on which to build. It provides a now widely-accepted model of systems engineering that assists cooperative technical action, that uses common language for clarity of communication, and that enables common understanding by those collaborating to meet inter-organisational agreements and intra-organisational commitments. Its definitions, terminology, and actual and implied models provide the current best approximation to a systems engineering lingua franca.

However, international standards, certainly those addressing business or technical process-oriented codification, are moving baselines of good practice. They make headway and prosper through greater understanding from use; from steps forward in theory and from the influence of an environment of improvements in related practice. As a body, standards are never static and, being somewhat asynchronous in their advancements, they are never completely self-consistent.

Perfection and idealism is therefore not what ISO/IEC 15288 is about. Its underlying engineering power is not that it just looks and feels right, but that it can also be demonstrated (within the limits of present knowledge) to be well-founded in its theoretical base. This technical theory, as seen in §3, derives from a longstanding and surprisingly stable dimension of human knowledge. The International Standard's detail may fluctuate with eddies of interpretive nuances, fashion and engineering politics, but its general course is unlikely to be substantially unsettled for a long time, if ever. Its intellectual bases, just like the outcomes they lead to, are the product of fundamental human faculties that, whilst they may be conditioned to an extent, are ultimately in-built.

4.5 The Management Viewpoint

4.5.1 A Project Focus

This section places the technical processes that lie at the heart of systems engineering within a business organisation setting in which they are typically conducted. This setting is conveyed in Figure 4.3 and in the business process influence/responsibility model of Figure 2.10. This management context conveys the close mutual dependence in particular between systems engineering and project management, and a primary interface that binds management to engineering in any organisation.

The dominant viewpoint that provides a shape of the whole ISO/IEC 15288 model is that seen by the project. This project-centric perspective was considered to be the most useful, pragmatic and advantageous way of presenting systems engineering concepts and provisions.

A close correspondence between the system boundaries and related boundaries of project responsibility is an important notion that runs through the standard. The definition of the system-of-interest's boundary, its likely composition and its interactions with other systems has a major influence on the partitioning of work through the life cycle of a system – the so-called work package structure of project management. It drives the planning, the execution, the assessment and the control of a system of hierarchically and temporally related projects.

The goals for a project are largely driven by the agreement established between the organisation and its trading partners, in line with policies set by enterprise management. Accordingly, the Agreement Process is applied inter-organisationally to establish this trading relationship. Ultimately, satisfactory delivery of a system, whether defined in terms of product or service characteristics, will be assessed against the agreements.

Across the whole organisation, enterprise management acts to provide a business framework that is equal to the trading obligations it enters into, and also to the needs of its own stakeholders. The primary system-related actions of enterprise management are defined in the Enterprise Processes. Key aspects of this organisation-wide policy relate to the creation, support and monitoring of the organisation's portfolio of projects.

The responsibility for the creation of a system or the provision of a service is assigned to a project within this framework of organisational policies. Authority is conferred by enterprise management to project management to employ organisational resources as well as to acquire externally provided assets and services. Again, this is accomplished using the agreement processes, though in this case it is applied intra-organisationally and without the legal formality of inter-organisational trading.

Each project incorporates, and tailors as necessary, those Project Processes that are required in order to employ and to direct its resources and assets to achieve its particular goals. They enable a project to perform the technical transformations which are applied in accordance with the level(s) of compositional detail and the stage(s) in the life cycle for which it has responsibility. The nature of the process work products generally vary with compositional detail and stage, as may the methods and tools used to produce them. Each project therefore is likely to tailor the Technical Processes in order to meaningfully place them in context and convey their relevance to team members.

The technical processes are influenced by, and in turn confer constraints on key aspects of the tactical management at a project level, on the strategic management at the enterprise level and on trading interfaces with other organisations. For example, in the early life cycle stages, systems engineering is about fuzzy boundaries that explore the practicability of solutions and about accepting fluid change. In contrast the management of projects, enterprises and certainly agreements is inclined towards hard boundaries that align with contract legalities and favour infrequent, discrete changes.

Individual organisation's in a hierarchy or network of trading therefore can each employ the same set of processes to bind them in common enterprise. Inter-organisational agreements build an arrangement of supply chain hierarchy and through-life transfers of system responsibility. In principle it is possible for one project to have complete through-life responsibility for a system-of-interest and this is the image presented by the DE&S, though changes in team members, locations and immediate management structure weaken the reality of such continuity.

The Project Processes are intimately concerned with managing the resources and assets allocated by enterprise management, and with applying them to fulfil the agreements into which that organisation enters. They apply to the management of any project, whether to create tradable systems or enabling resources, e.g., organisations' infrastructures, facilities, enabling services and their distinguishing foundations of progressive technology. This management of a project entails planning, in terms of resources (including budgets), timescales and achievements; the checking of actions, to ensure that they comply with plans and performance criteria; and the identification and selection of corrective actions that recover shortfalls in progress and

achievement. It is in essence the 'plan', 'check' and 'act' of the Shewhart cycle that regulates the 'do'; the 'do' being the execution of the technical processes that 'engineer products'. This control cycle is shown to the left of Figure 4.15 using the International Standard's process titles.

The four processes shown to exist in the life cycle of any system [§3.4.3] bring an essential discipline to the management of any project. They steer and record project progress in an orderly, well-trying manner. Together with the 'plan', 'check' and 'act' processes, they form the minimal set of Project Management Processes at any point during a life cycle and at any level of system detail. Shown schematically in Figure 4.15 they are the management essence of ISO/IEC 15288.

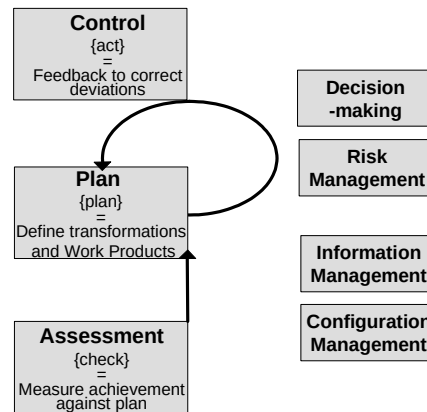


Figure 4.15 Project Management Processes

Any particular project views its system to be a system-of-interest, and whilst it can influence higher system levels it does not have responsibility for them. However, it does retain ultimate responsibility for the system elements that constitute its system-of-interest, and consequently for the output of all subordinate projects, see Figure 4.16.

In practice, the risks associated with responsibility for a system diminish as feasibility and viability are confirmed according to devised architecture and implementation approaches that are explored and selected. Eventually finer detail is no longer of undue interest to a project. At this level (not necessarily the same level down different paths of system-of-interest decomposition as shown in Figure 4.12) each system element may then be acquired with acceptable risk. The detail of, and the rationale for, element composition can acceptably remain hidden from the acquiring project. The risk/benefit balance has been conferred on a trading partner. From the system-of-interest viewpoint, this delegation of responsibility for each system elements may appear as being where specialist disciplines or particular implementation technology practices are engaged in more pragmatic detailed design and build.

Thus, from the perspective of each project having responsibility for its own system-of-interest, a level of ever decreasing system detail is reached down any path of decomposition that is no longer of undue concern. For each path, this level is defined in terms of a judgment that balances the risks and benefits of conferring continuing responsibility on another party with appropriate capability and competence. That is, a level of system detail is ultimately reached at which the provision of a system element, having specified performance characteristics and necessary implementation constraints, can be delegated to advantage to another party. Such a system element may already exist, in which case it is an off-the-shelf acquisition or a re-use item, or it may need to be designed and built, possibly in-house for critical elements; either way it becomes the conferred responsibility of a subordinate project/supplier. A system element, as viewed by one project, thus becomes a system-of-interest when viewed by another project.

The starting condition for this conferred (not transferred) responsibility is the reaching of an agreement. The stopping condition is transfer of ownership of the resulting goods (in exchange for some consideration). It should be a mutually advantageous risk/benefit judgement by both parties, ideally with a sharing of the consequences of risks and benefits – a government/industry trading message of Smart Acquisition.

By identifying system elements, a project can thus confer authority on subordinate projects in supplier organisations for the supply of parts of a system, and draw on knowledge, skills and assets that lie outside its core capability. This assigns bounds on a ranking of project responsibilities for specific ranges of detail in the system's structural hierarchy, i.e., for successive descriptions of architecture that span from each subordinate project's unique, holistic system-of-interest down to a level of detail at which, respectively, its traded system elements are defined.

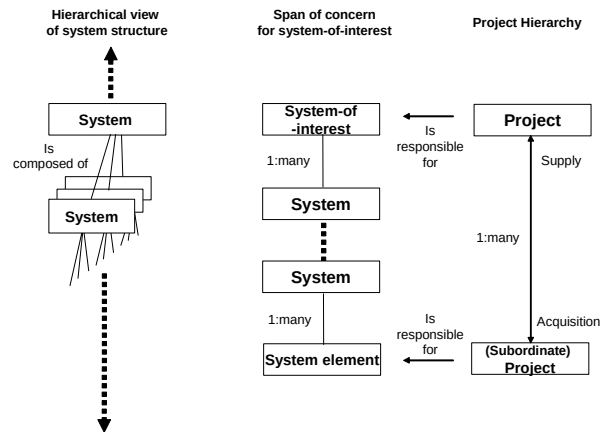


Figure 4.16 System and Project Hierarchies
[Contributed to ISO/IEC 15288]

For a temporal bounding of project responsibility, the stages of the life cycle offer the most containable and manageable start and stop points. Using enterprise management's control via decision gates, the initiation and termination conditions for a particular span of life cycle responsibility can in this way be defined. These are typically expressed in terms of work product maturity or levels of achievement.

By using the level of interest (measured in terms of risk/benefit) to terminate structural responsibility for the system-of-interest detail and using the cardinal conditions of system state, i.e., life cycle stage, to commence and terminate that responsibility, projects arise and then close; each playing their part in a dynamic, continuous flux of asset deployment and employment. The hand of authority that sanctions all this is the strategic management that governs the whole organisation. It is responsible for corporately ensuring the effective conduct of systems engineering, and it is considered next.

4.5.2 Corporate Responsibilities

Enterprise is that part of an organisation with responsibility for mission, goals, and objectives, so as to supply products and/or services according to agreements. It is set in the geo-political surroundings of self-adapting social systems [Figure 3.2], with responsibilities to and influences on society at large.

The Enterprise Processes are concerned with ensuring that the needs and expectations of the organisation's interested parties are met. Dealt with here largely for completeness and not in detail, they are an integral part of the International Standard and are the overriding determinant of a corporate environment of systems engineering capability and competence, as considered in §6.

The Enterprise Processes are typically concerned at a strategic level with the management and improvement of the organisation's business or undertaking, with the provision and deployment of resources and assets, and with its management of risks in competitive or uncertain situations. Responsibility for these Enterprise Processes is typically at the highest level in the organisation. Overall responsibility for life cycle management resides at this enterprise level.

A single organisation may be involved in several enterprises. Conversely, an enterprise may span one or more organisations and be governed by segregated management having substantial autonomy, and separate and individual procedural approaches.

It has the strategic responsibility to create an organisational environment with the purpose of ensuring the satisfaction of the needs of organisation stakeholders. This leads to an organisation's infrastructural capability, e.g., a distinguishing, progressive technology base and enabling resources, and to instituted projects that engage in conceiving, creating, utilising and retiring systems.

It achieves this through its management of organisational resources to common benefit, according to the scope of the organisation's trading (or enterprise) purpose, e.g., return-on-investment to shareholders, responsibilities to society and employees, business risk and continuity, deployment of assets.

Many of the effects of the enterprise environment on systems engineering are seen as constraints and additional requirements imposed on the options for creating a system. Examples of this may be seen in:

- the business scope of the enterprise that determines the markets, application areas and opportunities that systems' ventures pursue;
- business policy that defines how, on behalf of the stakeholders in the enterprise, resources will be invested; it influences for example the decisions to bid, invest or proceed with a product development, it determines the nature and allocation of corporate resources capacity;
- market strategy that influences product system families, intended product lifetimes and support policy;
- investment strategy; this impacts on the product novelty, introduction or use of technology, supporting infrastructure for system design, the capabilities and training of personnel, the manufacturing locations and capacities, and so on;
- business practices that lead to mandated or recommended process standards, business and technical process improvement actions, common methods and tools.

The concurrency of process use also applies to Acquisition and Supply processes, and it emphasises the reality that any organisation is simultaneously and/or sequentially both an acquirer and a supplier of products and services. In the limit in Figure 3.2, the upper reach of society is the only acquirer without being a supplier; fundamental science the only supplier without being an acquirer. A message here is that systems engineering enhancements actions apply equally to both acquirer and supplier roles; they are reciprocally related at each trading boundary. Smarter acquisition should be matched by smarter supply within and across any organisation – DE&S must also practice 'Smart Supply'. Its industrial supplier organisations employ CMMI and many other systems engineering devices to do just that.

4.5.3 Trading and Agreements

Systems engineering does not just structure well-integrated of systems, it also introduces coherence across trading organisations. System life cycle processes thus serve to harmonise information interfaces and artefact transfers at each trading boundary in a synergistic federation of businesses. In this regard, enterprises may be viewed as complex systems established for the purpose of trading, and it is the contracts they enter into that specify interactions at their commercial borders.

A hierarchy of projects, generally spanning different organisations, is likely to characterise the trading environment that creates systems with any appreciable degree of complexity. Similarly, sequences of projects, again in potentially different organisation, may be responsible for the changing nature of the system as it moves from stage to stage through its life cycle. This is thus a matrix of information and material exchanges across internal and external boundaries, as shown in the 2-dimensional simplification in Figure 4.17.

This viewpoint and its concerns for trade see organisations as producers and consumers of systems, i.e. their *raison d'être* is to trade products and services (with the caveat of profitability for their investors). One organisation can, acting as an acquirer, task another, acting as a supplier, for products or services.

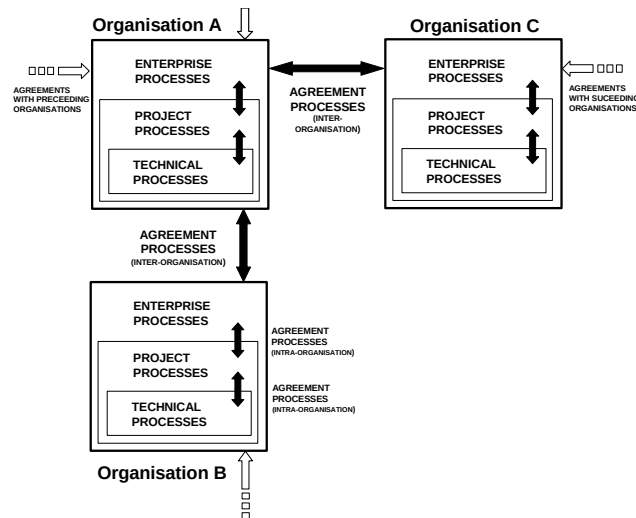


Figure 4.17 Agreements Between and Within Organisations
[Contributed to ISO/IEC 15288]

Business trading is facilitated via agreements that lead to the exchange of system information/artefacts. Tangible work products become the obligations of trading agreements. They may range from information items, to finished product, to delivered service. This disciplined shuttling back and forth in a trading complex is the most concrete manifestation of a tumult of underlying business process transformations. Purposefully conducted, it is the overriding goal of systems engineering (and it is also the ultimate remit of ISO to enable such trade and commerce).

Thus the purpose of the last processes in this presentation order of ISO/IEC 15288 is to provide a basis for establishing relationships between organisations, enterprises and projects as they trade in systems. Agreements formed by utilising the Acquisition and Supply Processes can be used for both formal (contractual) agreements between organisations as well as for less formal agreements internal to an organisation, e.g. between enterprise and project, and between projects.

The glue of agreements binds the trading entities in common purpose: the creation and application of complex man-made systems. Generally, organisations act simultaneously or successively as both acquirers and suppliers of systems. For example, in Figure 4.17 the vertical relationship of Organisations A and B can be considered to represent organisations in a supply chain, trading during a stage in a life cycle. Similarly, the horizontal relationship of Organisations A and C can be considered to represent organisations with successive responsibility for stages in a life cycle.

The Agreement Processes can be used with less formality when the acquirer and the supplier are in the same organisation. Similarly, they can be used within the organisation to agree on the respective responsibilities of enterprise, project and technical functions.

As the *INCOSE Vision 2020* puts it, 'the need for application of coherent systems engineering process across multi-party teams has never been higher. Trading in intra-organisation and inter-organisation situations depends on good common understanding of process. Global partnering and complex structures of subcontracts and suppliers make common understanding of process evermore important' [INCOSE, 2007a].

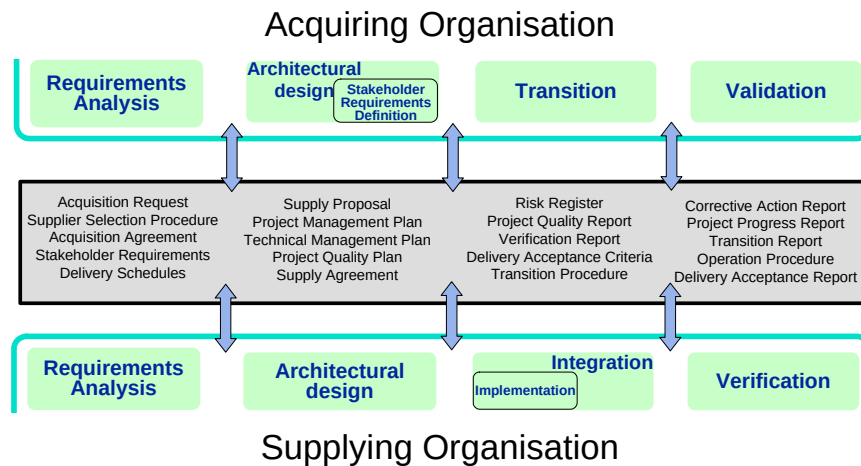


Figure 4.18 Typical Information Work Product Transfers at Trading Interfaces

Trading concerns thus give rise to the inter-organisational exchange view of systems engineering. The viewpoint is that of an interchange of work products across trading interfaces with a system of organisational entities performing process transformations in order to effect this.

ISO/IEC 15288 takes an important step in defining systems engineering in a way that is more clearly related to organisational responsibility. Its description of technical processes is set in the context of a spectrum of system-related business processes and explained in terms of an organisation's actions during the whole life cycle of a system. The traditional view of a sequential life cycle has given way to process representations that effectively convey hierarchy, not just in a system's architectural detail, but in the organisational and project structures that execute systems engineering.

4.6 Outcomes

This chapter has shown why and how, out of a continuum of tasks and activities, a set of coherent, pervasive system life cycle processes emerges. Their delineation is rationalised in terms of work products that relate to fundamental cognitive models and to the transformations that effect them. They are evident as three distinct layers of transformations regulated by a complex interplay that resolves them. These three layers are the trinity of causality that, together with their 'effective' causation of systems engineering practitioner skills, stand out as one Greek philosophy's enduring postulations.

An undeclared but nonetheless rigorous materialisation of this trinity is to be found in ISO/IEC 15288. Though it is commonly perceived as a body of evolved best practice that has been progressively refined and formalised, and has culminated an internationally ratified model of systems engineering, it should also be seen as a disciplined embodiment of systems reasoning applied to engineering and management – as a manifestation of the essence of systems engineering.

Whether it is viewed as a system life cycle management best practice model or a systems engineering best practice model, ISO/IEC 15288 takes an important step by defining systems engineering principles and practice in a way that is more clearly related to need, acquisition and supply, service delivery and organisational responsibility. As a result it is implicitly reshaping the profile and definition of systems engineering so that it can better serve today's international, technology-based business environment. ISO/IEC 15288 is a major milestone in reaching a consensus statement on system concepts and their practical application to trade and commerce.

For these reasons this investigation has employed the International Standard's conventions and terminology in its consideration of defence acquisition challenges and IPT practices.

This chapter has presented for the first time some key elements of Editorial rationale that lie behind the architecture of ISO/IEC 15288. This logic was unnecessary during review and ratification of the International Standard for it did not materially influence the key goals of relevance, acceptability and practicability of the provisions in the published document. To the contrary, it would have been a didactic hindrance to the introduction of some radical advances. Headway depended on issues of continuity, association with current practice and adherence to a 'comfort zone'. A transparent alignment with the instinctive stances of the design triad or philosophical precepts was decidedly not on the agenda.

A somewhat diffident launch of the new International standard largely avoided confrontation with the interests of established commerce and academia. However, this has had its price. With growing acceptance has come a 'bandwagon' rush to develop the model in order to further partisan interests. The lost US national driving influence and the backseat role of software engineering have led to a power base of specific rather than community interests. Thus, a somewhat traditional interpretation, based on industrial practitioner custom and practice without the enriching evidence that systems reasoning brings, augurs that the model may revert towards a style of systems engineering seen in its predecessors.

For the moment, ISO/IEC 15288:2002 is the prevailing model and it is on this that the rest of this investigation turns. It does this by taking the International Standard's life cycle processes and in §5 examining the power in the patterns that they can form. In §6 it builds the complementary elements of the triptych of images. In §9 and §10 it explores how the processes, and the reasoning from which they emerged, can be used to interpret IPT challenges, and to structure advancements of understanding and responses to the imperatives of defence acquisition enhancement considered in §7 and §8.

The foregoing indicates how a ratified model of action, propriety, attainment and technical communication can bring discipline to collaborative endeavour and to engineering creativity. From it, a technical vocabulary is being built. This not only communicates a systems engineering perspective but also offers a unifying foundation able to translates to and from the implementation technologies. This kinship between systems engineering and technical specialisms is demonstrated in §10.5 with regard to human factors engineering.

Systems engineering's identity and characterisation in terms of business process is beginning to gel. The crucial breakthrough for the ISO model will be if and when it is recognised as a natural and indispensable tranche of business process, not just in defence and aerospace but in any business sector where complex products and services are created.

5 LIFE CYCLES AND THEIR MANAGEMENT

5.1 Prologue

Engineering is defined by Collins as 'the profession of applying scientific principles to the design, construction, and maintenance of machines ... aircraft ... military fortifications and communications' [Collins, 1991]. A minimal, underlying sequence of an engineering life cycle is fused into this definition. Engineering, and systems engineering in particular, is indivisible from life cycle management.

Logically, systems, as seen in § 3 and § 4, may be viewed spatially in terms of their relative composition and arrangement, and also according to a dimension of relative time. However, in the physical world, natural and man-made systems 'live' juxtaposed with a succession of manifolds of actuality, as they pass through a sequence of life cycle states in which the relative has become the actual. Any instance of system is characterised by its actual unidirectional life flow of transformations that changes material and energy, and information carried by them, from an initial to a terminating order. The human activity system engaged in effecting and resourcing this – the Effective Cause – sees holistic patterns in its actions. It uses them to structure schemes of action and provide waypoints of navigation so that there is a shared rationality and a coordinated way forward for the integrated team responsible for periods or for all of the life cycle.

The transience of animate and man-made systems alike serves a purpose. It is an essential enabler of renewal and change. In both cases it solves the problem of changed circumstance, failure of fittedness, and the march of wear and tear. Whether it is the accumulated inheritance of biological fabrication information or the accumulated factual and experiential technical knowledge of good design, all systems ultimately depend on a common theme: tried and tested, repetitive patterns of creation.

These patterns have been long recognised in the cycles of animate life. More recently, they were seen during the systematic enquiry into many fields of natural and human science. It became clear that patterns governed the technical and social artefacts that life itself had evolved the power to build and to use. The metaphor of life cycle is well-founded and, as seen in §3.4.2, can be instructive in understanding how man-made order comes to be.

As pointed out in §3.4.2, organic systems are repetitiously constructed and managed by nature, forming unbroken, drifting and dividing channels of information flow in response to environmental conditioning. Not unlike this, man-made systems are the result of unique, intentional flows of communal decision making, directed purposefully and responsively toward a diversity of sanctioned goals. Hence, the management of these life-flow patterns in man-made systems and the rationale behind them is aptly termed system life cycle management. It is evident as a strategic, team-based structuring and control that is a top-level concern of systems engineering. In MOD, it has gained currency in the term 'through life capability management'

Royce is attributed with ushering a disciplined yet creative life cycle understanding into modern engineering when he addressed the challenge of managing the development of large software systems. Although he advocated iteration within a series of sequential processes, his contribution was (mis)interpreted and widely adopted as the now archetypal stepped waterfall model [Royce, 1970, A/1 pp 1-9]; some of his deeper wisdom was overshadowed by his model's capability to convey life cycle logic at its termination. The power in the simplicity of this essentially linear, unidirectional model has served engineering well. Models must meet the needs of the moment and the waterfall representation did just that; its more profound iterative message being a step beyond that required at the time. It continues to serve well, for it conveys a classical order of emphasis and, at the conclusion of design, an inescapable sequential logic in material, energy and information transformations.

It remains possible to hide much of the labyrinthine complexity of technical and management actions that lies behind present-day systems, families-of-systems and systems-of-systems by recourse to a linear life cycle. It is after all how the strategic management minds in

organisations abstract order and progression from the profound complexity that underlies their responsibilities; it is how, by and large, much of industry was successfully overseen during the C20th. But problems change, solutions diminish in effectiveness, technology evolves, the social environment becomes less predictable and budget profiles are squeezed. This encourages more calculated and better risk-managed approaches to be followed. Cycles of feedback and learning, and controlled increments of commitment mean that simple linearity may retain a didactic logic, but this becomes increasingly detached from the complex practicalities of trade and commerce.

Chapter 4 explained that life cycle management was the catalytic focus of ISO/IEC 15288. It is revealing that the original '*life cycle*' titling of the International Standard ISO/IEC 15288 was only at a late point in its development re-branded by a replacement containing '*systems engineering*' – the International Standard's requirements and implementation strategy had been from the outset firmly driven by life cycle management motives. It says much about the perceived strength of correlation between systems engineering and life cycle life management by the international commenting community that a title change without any content change could so easily have been internationally ratified.

Fundamentally, life cycles are the natural abstraction of a dynamic network of systems engineering processes. A life cycle is itself the paramount instance of process; holistically assembled from lesser transformations and from echelons of progressive detail. At particular levels of abstraction, significant, purposeful patterns of transformation appear out of a mosaic of stock-in-trade processes and the primary pictures of system life cycle management emerge. Macroscopic order and information is built out of an overwhelming microscopic 'disorder' that is the day-to-day, project-based, background 'noise' of systems engineering. Without the controlling hand of life cycle management, team action, set within organisations that perforce operate in global trading situations, would become impossible.

Life cycle management is therefore for some an unimpaired embodiment of systems engineering principles and practice. Through-life management and systems life cycle management are synonyms for sound business management. In MOD, effective through-life management is acclaimed; aggressively pursued; keenly rewarded. Yet in contrast systems engineering has been largely cloistered, nearly to obscurity, in the AMS and even now in the AOF. Greater visibility and understanding of through-life management's dependence on systems engineering and its patterns of process is to the benefit of all – engineering and management. This is the theme of this chapter.

5.2 Synopsis

The logic that rationalises a system of transformations is perforce constrained by the unidirectional march of time that governs its physical enactment. To the systems reasoning mind this reveals fresh order at different scales of consideration. The appearance and rationale behind this order is the subject of this chapter.

Any man-made system progresses through its life cycle as the result of actions – transformations and controls – performed and managed by people in organisations, typically according to predefined processes. This is how unity of contributions and uniformity of quality is attained within and between organisations. These individual actions may be encoded according to the system life cycle processes described in §4 and, as seen, from these models can be built that are far removed from the linearity of bucket-brigade simplicity.

The step from life cycle processes to holistic life cycle is one that is upward in simplifying abstraction, toward an apparent reduction in complexity. The nature of this transition is first considered in terms of a hierarchy of process granularity. The uppermost level of this hierarchy is the holistic life cycle. Its lower levels gravitate into the detail of method and tool dependency. Between them lie the stages and the generic life cycle processes: the primary instruments of life cycle description.

Next, the inanimate viewpoint, that of the system itself, is used to illustrate the need for coherent thinking of the whole lifetime of a system. Distinguishable system states within this lifetime map to first-order management decision making, in which a combination of business

strategy, resource investment and project risk determines the business direction taken; this begins to establish the primary structures of business life cycles. The different external enablers that are required to effect the transformations from one state to another are conveniently set outside the system-of-interest. Designated 'enabling systems', their crucial role of progressing the system through its life cycle is described.

The animate views of life cycle are those of the key organisational contributors: the enterprise managers, the project managers and the engineers. Each of these views is considered in terms of the decision-making responsibilities, the characteristics of the decisions to be made and the way in which they are structured to mitigate threats to technical and, ultimately, business success. The time-structured action and decision making are then considered according to a classification of life cycle patterns, with each class member tackling a step in a progression of challenges that are characterised predominantly by system complexity, novelty and degree of change. They present idealised patterns of process, cyclic phase and quasi-linear stage structure that provide templates for managing project and enterprise actions. In particular they serve to orchestrate the essential decision making flows of engineering creativity and management risk containment.

A classification hierarchy of five life cycle archetypes is proposed. They are idealised structures that stand out as cardinal references in a continuum of increasingly complex instantiations of life cycle model.

The morphology that transforms one pattern into another demonstrates the existence of the common, minimal, underlying set of generic systems life cycle process, as defined in §4 according to a consistent and manageable level of detail. Life cycle complexity can be seen to be a non-linear meshing of echelons of many life cycles that arise from the individual life cycles of all the elements and partial assemblies that comprise a system. This massive interdependency highlights the threat to delivered service that can arise when a single element reaches its termination, and it emphasises the need to manage the multiplicity of life cycles to avoid the adverse impact of obsolescence.

The life cycle management classification arrived at in this chapter basically accords with the cardinality in the classification of Figure 3.2. It prepares the ground for the defence acquisition setting described in §7, is a prelude to §9.4's critique of defence system's life cycle modelling and is a foundation for the questioning of continued CADMID supremacy in §9.4.3.

5.3 Life Cycle and Management

5.3.1 A System of Processes

Every system has a life cycle; ultimately, some hypothesise, even the cosmos. The diverse nature of man-made systems means that there is a vast variety in life cycles, and hence the individual models that can represent them. These life cycle models are the basis for planning or formulating the intent, assessing or measuring the passage, and controlling or guiding the progression of a system from its origin to its extinction. In short they shape the life cycle management of man-made systems.

§4 is concerned with how patterns of process build to form logical flows of action that can achieve coherent outcomes at successive level of structural detail. These outcomes represent major steps or, more fundamentally, state changes in the life flow of a system. These patterns of process act as frameworks for considering lower-order maps of life cycle action. This points to the importance of being able to build generally comprehensible life cycle models from individual processes so that the actions of diversely skilled individuals then contribute in concert to the whole.

Whole life thinking means that the set of system life cycle processes must be capable of modelling the information and the material and energy transformations, and the associated enabling actions and outcomes, throughout the system's life span. This set must be capable of being assembled with different order and usage both across the life cycle and at different levels of system detail. The diversity of technical and managerial contributions throughout a system's life also means that system life cycle processes must possess flexibility and

generality. Their modularity, cohesion and granularity have been carefully refined in order to permit the building of an almost limitless range of life cycles that are themselves complex transformation systems. The selection of the process set granularity and, of importance, retention of its uniformity of abstraction across its individual constituents have been of fundamental importance to the codification and practice of systems engineering. Decades of trial and error have struck a practical balance.

The granularity must be able to build life cycle representations that convey meaningful actions to different parties, equate to recognisable responsibility, and assist planning and control. It should avoid descriptions that are tortuous, near irrelevant networks of highly concurrent detail. The resolution of a broadly acceptable set of generic process descriptions from which to build life cycle models, as rationalised in §4, has been a key step in systems engineering's procedural maturity and, as seen in §6, also in its definitions of capability and competence.

The generic life cycle processes must be defined so that they may coherently contribute to the building of any life cycle model. Their individual purposes, outcomes, activities and input/output work products are specifically selected to interact as a coherent and complete system of processes such that they permit unique combinations of transformations and overall delivered purpose to be assembled. This generic process set is the basis for the technical, project and enterprise management of system life cycles and ISO/IEC 15288 provides such a set.

Life cycles may be viewed as the structure of events, actions and work products that mark the passage through the whole life of a system. For a system to be effectively managed there is a basic dependence on holistic or whole-life thinking at all times by all parties who are responsible for individual process execution; the antithesis of a past 'over-the-wall' culture that distinguished and rewarded one period and organisational unit of responsibility from its successors. Fundamentally, effective life cycle management depends on continual looking forward and, of comparable importance, looking backward in order to establish coherence in decision making. Such whole-life thinking means, at the most strategic level, concomitant consideration of a system as both a product and service.

This reduction to a minimal, generic set of modelling elements demands the use of process patterns to help cloak the complexity of process detail and thereby to reveal different strategies employed in life cycle management. This leads to transformation/work product structure and purposes expressed in terms of more discrete units of transformation and outcome: super-process descriptions that establish higher levels of purpose and activity outcome. From this, higher-order processes emerge. They define the actions and outcomes at a strategic level of system life cycle description according to a high order of organisational management responsibility. At these levels of action and responsibility, characteristic, intuitively distinguishable major steps in system lifetime progression can be seen. They are frequently termed life cycle stages (in MOD, phases). Aggregations of these life cycle stages, sequentially and concurrently, lead to a complete life cycle: birth to death.

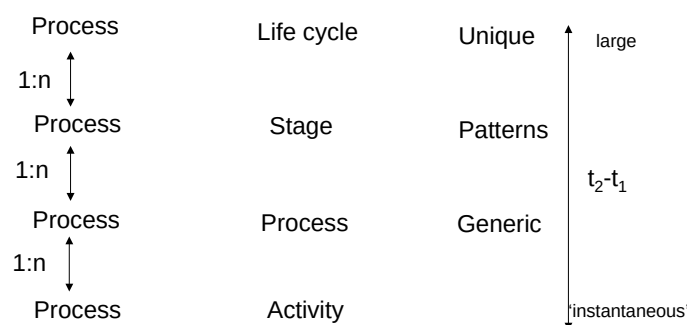


Figure 5.1 The Hierarchy of Process from Life Cycle to Activity

Just as the general class of system is comprised of subordinate systems, so in a categorisation of processes the 'generic level' of process is also comprised of subordinate processes, here

termed activities. Figure 5.1 conveys this notion of generic, stock-in-trade system life cycle processes set in a hierarchy of super- and sub-ordinate processes.

At the highest level in Figure 5.1 is the undivided life cycle. Whilst not normally regarded in terms of a process, it possesses all the attributes defined in Figure 4.1. A discrete life cycle is subject to:

- a set of starting or input conditions that arise from circumstance and environment;
- an initiating concept and input of resources to create a system;
- a transformation whose outcome is a service intervention that affects the conditions in its surroundings;
- a termination or restoration state of the environment, typically at system disposal;
- start and finish times of this lifetime of events;
- responsibility and resources for its execution.

Beneath life cycle, stage is the next level of process structure down. Subordinate to that come the generic system life cycle processes. The decomposition of process continues, with terms such as activity, as in ISO/IEC 15288:2002, and in some models with the term task applied to ever finer process detail.

This means that each process level in Figure 5.1 can in principle be defined in the same manner. However, in practice the different ways these layers of processes definition are used to convey different levels of life cycle granularity do not favour this. Basically, the purposes, outcomes, activity detail, work products and dynamics all change in their nature, and they benefit from shifting forms of description. Thus each level in Figure 5.1 is able to build different, though equivalent models of a complete system life cycle, with different echelons of detail of life cycle responsibility favouring different representations to give meaning to their respective roles.

Beneath the top, unitary level of system's life cycle model the increasing detail of subordinate contributions give rise to increasing and eventually limitless variety. From recognition of need to termination of solution existence, instances of complete life cycles detail differ one to another, and ultimately resolve into the unique.

At the level of stage the span of the life cycle is divided into shorter periods and the models of transformation and work product begin to describe mutually dependent flows. They give rise to life cycle representations with often familiar and strategically meaningful patterns. These shorter time spans often equate to the duration of individual projects, or at least to major approval waypoints of a continuous project. Sequence predominates, though concurrency may be evident.

At the next level come the system life cycle processes. Their purpose and nature are so familiar that they form a symbolic layer of generic process. They enable the familiar, well-practiced, readily planned and commonly executed set pieces of engineering and management action. Times scales are even shorter, with the result that process concurrency has equal sway with process sequence.

At a level of process granularity beneath system life cycle process, activities (the ISO/IEC 15288 term used in Figure 5.1) serve to give descriptive detail that adds meaning, finer action structure and potentially greater regulation. Here, a high degree of invocation, iteration and pervasiveness exists, and ever greater concurrency means that rigid sequence holds even less sway here. To introduce structure and sequence here it is to rapidly stray into the elective realm of method.

Beneath activity, task might identify the next level of process detail, however, the transformations and work products are so basic that the term process barely applies to the formulation of this ubiquity. This process detail tends towards such instinctive engineering and management actions that its codification becomes trivial and unnecessary. In addition, the power law growth of description militates against tomes of further detail.

5.3.2 States, Stages and Gates

Despite an apparently limitless variety in system life cycles, an underlying, essential set of characteristic life cycle periods exists in the complete life cycle of any system.

Stages are an important construct of enterprise and project management and provide the framework for the highest level of control. They establish a pattern of strategic risk management that relates to temporal responsibility spans and progression criteria. From the view point of the system itself stages are recognisable technically by state changes. They are synthesised from a consideration of system states, organisational responsibility changes and time profile of business risk.

Any system exists in several primary states during its lifetime and these relate to the status of the system description or the system's manifestation. Periods during which the system's state remains stable are terminated by cardinal transitions.

The most dominant of system state changes is that from product to service – it is the necessary pivot around which the creation and utilisation periods in any life cycle balance. Without this essential transition, a system's lifetime would be prematurely terminated and be void.

Other state changes are less sharply defined but are nonetheless essential waypoints along a system's life path. For example, the step from conception to committed development is one of changing gear model-wise and also resource commitment-wise. Similarly, the state transition from "assessment" of feasibility to the "demonstration" of viable product is a crucial step, and one that merit close scrutiny and documented evidence to justify subsequent major commitments of resource to "manufacture". These are particular state changes that govern MOD's CADMID life cycle flow and stage titles.

Any re-focusing of responsibility according to a set of subordinate elements also gives rise to a transition in system description detail and hence a new state. In practice these are often accompanied by changes of organisational responsibility for the system and/or its parts – project to project, business unit to business unit or organisation to organisation – and thus normally governed by intra- or inter-organisational agreements.

When viewed from the organisation's responsible for a system, state changes appear as stages of management in the life path of the system. Thus, in a rational life cycle modelling approach, system states and life cycle stages are intimately related, if not exactly equivalent. To the responsible parties these stages represent major system life cycle periods, and they designate the cardinal progress and achievement milestones of the system through its life cycle. The characteristic, intuitively distinguishable stages bring particularity and significance to how common processes are conducted, leading to different emphasis of use and effecting moves from one primary system state to another.

Different industry and application sectors have their characteristic system states: in pharmaceuticals, safety demands that trials and testing comprise a major later state of drug development; in public transportation, persuading stakeholders as to the net technical and commercial benefits of new solutions and to tolerate the disruption of introducing replacement solutions is frequently an essential public consultation state; in IT systems services, it is the migration to new generation assets and configurations in order to increase performance capability, and to avoid obsolescence, lead to a 'roll-out' or changeover state.

As with state, each stage has a distinct purpose and contribution to the whole system life cycle, and also to the interests of the organisation. Stages span from the earliest abstract images of a future system to the termination of its physical existence. The structure of life cycle process usage is not defined within the life cycle stages; they encapsulate this detail within a boundary of strategic purpose and outcomes. In this way and with appropriate tailoring, stages establish a framework for enterprise assessment and control of any life cycle.

Stages of a life cycle thus serve quite different purposes to the generic life cycle processes, and have different significance to different parties across the organisation. A process may be enacted in any stage in the life cycle and it is this separation of quasi-continuous and therefore largely concurrent processes application throughout a life cycle (largely serving the interests of the project) from the more strategic stage patterns (that largely serve the interests of enterprise management and organisational trading) that is a strong, distinguishing feature

of the ISO/IEC 15288 model. The unguarded mixing of the generic process and stages, as if they apply to the same level of technical abstraction and management control, has led to marked structural unevenness and unhelpful distortions in many process standards, and from this on into many organisation's life cycle reference models.

The precepts of life cycle management dictate that each system should have an appropriate life cycle stage model from a very early point in the system's lifetime, and that this should be explicitly defined and employed in a disciplined way in order to manage progression during the life cycle. For many systems this is frequently achieved by early adoption of a standard life cycle models to help structure planning, and this model is refined through a growing understanding of problem and solution. Each and every stage needs to be considered individually and jointly when planning and executing the system life cycle.

Though often used synonymously, the terms stage and phase do not trace to the same ontological origin. They allude to quite different concepts, both important to life cycle management

Stage (as used in this text and ISO/IEC 15288) connotes the image of renewal of allocated resources that enable a venture to run its course, as in predetermined staging points to continue a journey or venture. This metaphor conveys an essentially linear path of engineering and management with stopping points for decisions that lead to the sanctioning of fresh assets, including budget.

Phase (as used in this text) represents a distinguishable aspect or sector of a repetitively changing situation, as in the recurrence of phases of the moon. It is a feature of cyclic model forms, as shown in Figure 5.7, and is a metaphor that suggests reiteration of identical or similar situations. As such it is apt for conveying sectors of corresponding action that repeatedly evolve and progress achievement from stage to stage. The classic spiral life cycle representation can thus separately embody both the notions of stage and phase, as is shown in Figure 5.9.

MOD's use of the term phase in the CADMID life cycle may betray blinkered life cycle management logic, but is more likely to have been drawn from the longstanding DOD misnomer.

Stages therefore frame the detail of project and engineering action, providing decision gates that act as scrutiny points for authorising life cycle progression. These stage work products are primary evidence of achievement and of a satisfactory, ongoing benefit/risk relationship. Decision gates are thus typically synchronised with the commencement and termination of the system state changes. Quantification of the maturity of the system state at each decision gate, and of the resources consumed to achieve this, provide visibility of progress and achievement to enterprise management.

Decision gates are characterised by pre-defined criteria for the enterprise's commitment to continuation, and may signify complete transfer of project or organisational responsibility. They lock technical and management decisions and controls into a single model. The resulting enterprise management decision-making, e.g. Initial Gate Approval in Smart Acquisition, leads to the strategic control actions, such as allocation of continued organisational assets, adjustment of requirements or partnership commitments, that characterise sequential stage life cycles.

With some emphasis or profile, each stage completes a full cycle of the technical processes as conveyed in Figure 5.8, but each successive stage exhibiting a progressive shift (anticlockwise) in its focus of attention. Each such (epi)cycle strives to arrive at a 'minimal uncertainty', 'highly defined' condition at which point the decision criteria, such as the continuing need for the system, the solution viability and the risk, can be most easily determined by assessing the compliance and quality of the delivered product/service in a notional or actual operational environment.

Table 5.1 shows a hypothetical sequence of detailed system life cycle stages that could be employed in the creation of a mass-produced system. The names illustrate the principal purposes of each stage. The resulting output system-of-interest progressively changes, with the nature of successive delivered products following a path of increasingly faithful representations, much as illustrated in the steps shown in Figure 3.9. Likewise, the nature of the operational medium in which achievement is demonstrated correspondingly changes according to

progression through the life cycle. In the column to the right, the decision factors that primarily indicate stage success are listed

STAGE	FORM OF DELIVERED PRODUCT	OPERATIONAL ENVIRONMENT	DECISION FACTORS
Feasibility Study	Technical papers, commercial analysis, engineering calculations.	Reviews, consultations, discussions, presentations.	Confirmed need, proved concept, feasible solution business viability
Modelling	Scale/space models, simulations, mock ups, animations.	Software simulation environment, physical test situation.	Performance criteria, selected technologies, technical risks.
Demonstration	Partial implementations of key/risk features, pilots.	Development environment, emulated real-world.	Technical viability, technology maturity, business confidence
Prototyping	Representative implementation of all characteristics.	Instances or representations of operational environment.	Implementation viability, service characteristics, operational viability.
Initial Production	One or few fully integrated, validated, operational systems.	Actual location or circumstances of through-life operation.	Solution validity, manufacturing viability, in-service needs.
Mass Production	Batch or continuous manufacture of Full Development product	Actual location or circumstances of through-life operation	Service conformance, quality, service continuity.

Table 5.1 An Example of Stages, Corresponding Products, Operational Environments and Decision Factors for Volume Manufacture

From the projects perspective, stage gates provide well-ordered rites of passage and mark, in as far as possible, objective waypoints of advancement and readiness for future achievement. Although they may be signified by occasions of management ritual, they are rarely instantaneous actions and normally depend on a steady build-up of a dossier of information that is an indicator of current and prospective success. They should not be competitions, rather a platform for informed debate. Stage models are therefore essentially predicate-based representations, as compared with the predominantly action-based representation of processes.

At each gate, several decision options are open. The most common are:

- proceed to execute the next stage;
- continue the current stage until the designated exit criteria are satisfactorily met;
- return to a previous stage in order to conform to a revised purpose, or a new or preferred solution option(s);
- hold the project activity until evident uncertainties or shortcomings are resolved;
- terminate the project due to critical changes or an inability/excessive risk to complete.

System stages therefore need to be terminated by well-defined, objectively assessable achievement states. As a result, stages are predominantly overseen according to their work product status, and generally by a profile of achievement across a searching range of these work products. This then relies on meaningful and consistent definitions of increments of maturity in the evolving process work products considered in §6.3.4.

Globally, national defence acquisition has to date been founded on sequential stage models and on their decision waypoints, at which achievement is recompensed with a renewal of conferred authority and resources, as described in §7.3. CADMID is MOD's selected sequential stage model and it has guided IPTs since the introduction of Smart Acquisition. Its predecessor Downey Cycle model, so-named despite being a linear model, reinforces the point that stage models are driven by the thinking of enterprise business management – Downey was first and foremost an economist.

Although ISO/IEC 15288 uses a sequential stage model as its exemplar form to convey life cycle management concepts, it does not advocate it in preference to any other life cycle forms. The strong correlation between the CADMID life cycle model and ISO/IEC 15288's equivalent "CDPUSR" exemplar is shown in Table 9.3.

The defence industry is by no means alone in defining idealised, strategic life cycle models. A multiplicity of industrial sectors, consumer and professional markets, and government acquirers has laid down the patterns of their business. However, even within a single domain of application, such as defence systems, the concept that one sequence could describe all life cycles is misleading – CADMID is a variation on a traditional sequential pattern that the benefit of providing a grossly simplified reference model to which many parties across organisations can relate.

Unlike other national defence communities, the fact that the acronym CADMID is readily pronounceable has caused it to become a synonym in the UK for defence system life cycle. Attractive as this seems, it may harbour risks for Smart Acquisition: the long term institutionalisation of CADMID and the closing of eyes to other, more useful options. As CADMID begins to look less appropriate to many of today's defence acquisition traits, its reassessment becomes more pressing; an issue that is followed up in §9.4.2 under *Life Cycle Reference Models*.

5.3.3 The System's View

The abstraction of life cycle patterns from a dynamic system of life cycle processes may, in accordance with the International Standard, be described from three primary organisational viewpoints. These equate to the major classes or actor engaged in its execution and control:

- looking from the technical role, it appears as a pattern of system life cycle processes that assist navigation through the highly complex structure of information and material transformations;
- looking from the enterprise role, it is viewed as the primary business risk and exposure management model, offering macroscopic visibility and control that serve the strategic interests of the organisation, its investors and its primary relationships with trading partners, acquirers, collaborators and suppliers;
- faced with reconciling these two viewpoints is project management, with its concern for the day-to-day control and achievement and the periodic demonstrations of achievement and justifications for continuation.

Nevertheless, one revealing viewpoint of the system life cycle is not that of a responsible party at all, but that of the inanimate system itself. In the International Standard, it is this that leads to a life cycle perception of a system that is simpler and more powerful than in the predecessor standards that tackled through-life system issues.

In IEEE 1220 and EIA-632 it is the lack of a truly life cycle-oriented view of the system that resulted in the system being defined in terms of a composite whole-life model rather than a through-life discrete entity interacting with enabling systems. The complexity this then leads to can be judged in Figure 4.11.

In contrast to this, in ISO/IEC 15288 the system-of-interest is defined as an entity distinct from the succession of environments it encounters throughout the stages of its life cycle. For the International Standard, any system that aids transition of the system-of-interest from one state to the next (other than its service delivery state) is viewed as an enabling system, e.g. a test system used during manufacturing.

These enabling systems are essential to the provision of the system-of-interest services. Although they lie outside the boundary that defines the system-of-interest, just as with systems in the immediate operational environment, they act on it – though in very purposeful ways. However, they not only project influence on it purposefully but they sometimes do so in unavoidably constraining ways. This is particularly evident in the case where an enabling system pre-exists the system-of-interest, e.g. a production line influencing a new member of a family of products.

A power of the way in which the International Standard is formulated is that it can be applied to any enabling system when, in its own right, it becomes another party's system-of-interest. Using this stratagem, the labyrinthine complexity of layers of system-of-interest/enabling system interdependencies (such as in Figure 4.11) are resolved in a clearly-structured hierarchical structure and time ordering of organisational responsibilities.

Accordingly, the larger organisation re-applies the standard wherever it elects to create, operate and manage the enabling system services in-house, whether within or outside the responsibility of the system-of-interest project. For the smaller organisation, that may not possess the enabling system assets and needs to acquire these services from third parties, this approach enables it to apply the standard in a simpler, more focused way, linking reciprocal actions through agreements.

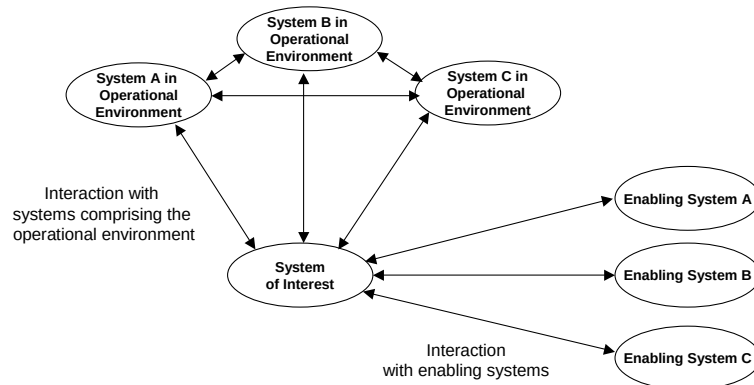


Figure 5.2 The Interaction of System-of-interest with Systems During its Lifetime
[Contributed to ISO/IEC 15288]

The overall relationship between the services delivered to the operational environment by the system-of-interest and the services delivered by the enabling systems to the system-of-interest are shown in Figure 5.2. In this manner, enabling systems can be seen to be necessary indirect contributors to the services provided by the system-of-interest.

When each enabling system is considered as a system-of-interest, in turn it too can be related to its own enabling systems. In this way, complex interdependence ripples out across an expanse of mutually related systems, each with its own particular purpose and context. This structural and temporal bounding that defines a system-of-interest is a particular and powerful employment of systems reasoning's reconciliation of information overload.

As with any system, each enabling system has its own life cycle. Each enabling system life cycle is linked and synchronised to that of the system-of-interest. The two major synchronising interactions are when (in the case that it does not already exist) the enabling system requirement is defined during the Concept Stage of the system-of-interest (or later if lead times permit) and when the enabling system is utilised to provide its particular service to the system-of-interest. This notion is conveyed in Figure 5.3. A series of interactions (as shown) meshes enabling systems and system-of-interest to form a super-system of immense potential complexity.

When an enabling system pre-exists the system-of-interest, i.e. is an existing part of the infrastructure of the organisation responsible for the system-of-interest or is a chosen service from a supplier, additional constraints on the system-of-interest generally ensue.

The reuse of, or contracting for, enabling systems is a major concern of enterprise and project management. In many instances it extends to bilateral and multi-lateral cooperation to form synergistic, long-term acquirer/supplier relationships. Information and material exchanges, synchronised trade-off of constraints, enabling resource capacity and availability, and many other factors demand a carefully orchestrated harmony in the life cycles of multiple enabling systems. It is a key role that is comparable to the forward planning necessary to make availability a technology base from which systems are created.

Consequently, during a stage in the system life cycle, the relevant enabling systems and the system-of-interest are considered together. Project responsibility for a stage in the system-of-interest life cycle extends to responsibility for acquiring services from the relevant enabling system. If a suitable enabling system does not already exist, the project responsible for the system-of-interest may find itself directly responsible for creating and applying that enabling

system. In the case of a specialised enabling system, this may be highly desirable and be where a major commercial differentiator lays.

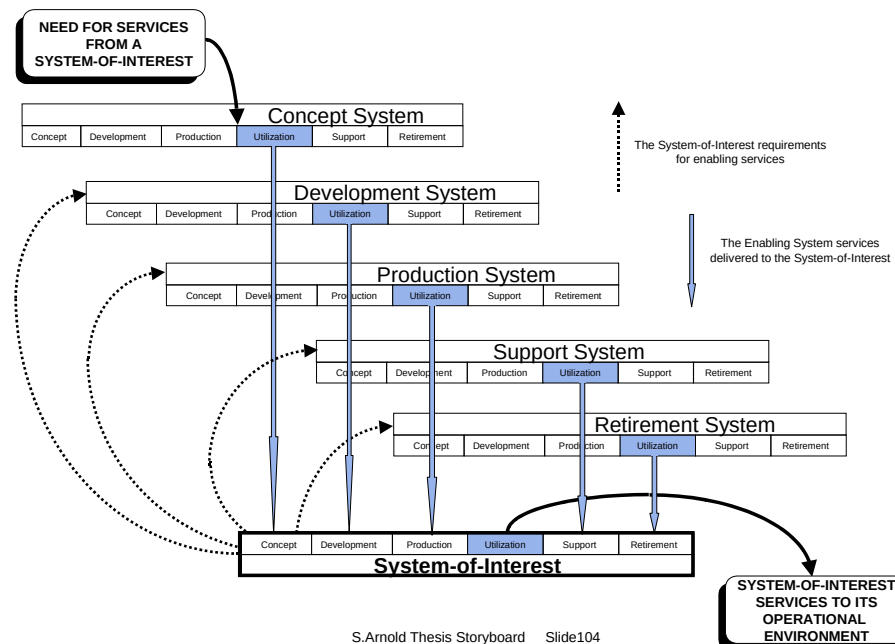


Figure 5.3 The Interaction of Enabling System Life Cycles with the System-of-interest Life Cycle [Contributed to ISO/IEC 15288]

From all this, the life cycle stages are seen to be key the planning, execution and management of life cycle processes in the face of this complexity. They provide the necessity of comprehensible and recognisable high-level purpose and structure. Precedence, particularly in similar market and product sectors, also plays its part in the selection of stages and the application of life cycle processes in order to build an appropriate and effective life cycle model for any system.

Just as each system element contribute to the system as a whole, so each stage of the life cycle needs to be considered during any other stage of the life cycle, and the contributing parties need to co-ordinate and co-operate with each other throughout the life cycle. This synergism of the life cycle stages and of functional contributors is essential for successful project actions. Close communication with the different functions and organisations responsible for other life cycle stages leads to consistency in the life cycle and in the system-of-interest.

The stages of the life cycle offer containable and manageable start and stop points for temporal bounding of a project's responsibility. Using the initiation and termination conditions for a stage, the span of project life cycle responsibility can be clearly defined and its delivery criteria into follow-on projects made clear. These transitions define major project progress and achievement milestones. They provide DECs, scrutineers, and DE&S business management with high-level visibility and control of advancement and accomplishment.

In the dynamic detail beneath the stages, processes can be applied at any level in the hierarchy of a system's structure. Each life cycle process may be invoked, as required, at any time throughout the life cycle and there is no definitive order in their use. Any life cycle process may be executed concurrently with any other life cycle process. The execution of life cycle processes may be quasi-continuous at any level of system structural detail.

Each process may be enacted in any stage in the life cycle, with the extent (and nature) of this enactment having a quite different emphasis in different stages. This is illustrated in Figure 5.4, in which an example execution profile of specific technical processes is presented as a function of life cycle stage. Its organ pipe representation exemplifies the different level of effort in different stages across a life cycle, here divided according to the linear pattern of

CADMID. This graphic could be modified to represent the cumulative effort as a percentage of the total required to complete the overall contribution from each process. Either way, it defines progressively accumulating steps of achievement from each process. The pattern of achievement varies as a result of novelty, stability of need and technology options, and generic figures are only indicative.

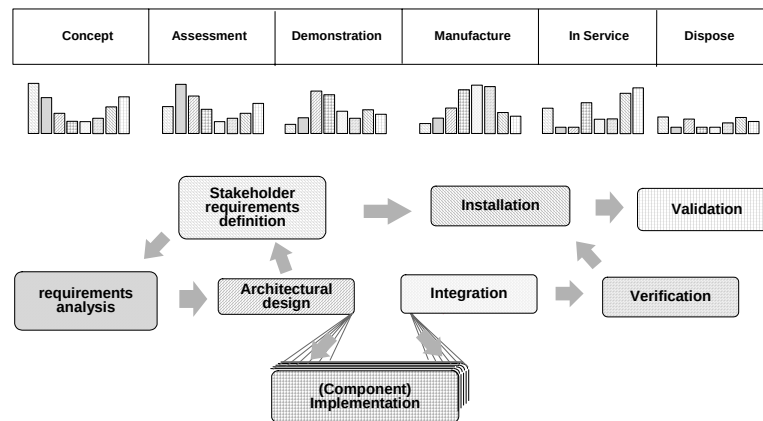


Figure 5.4 Process Concurrency and Continuity Expressed in Terms of CADMID

Despite the extent of process execution varying dependent on the stage in which it is performed, the logical nature of the transformation remains essentially the same. However, the physical characteristics of the material/information being acted on will vary, the methods and tools that effect transformations could well change, and the responsibility/skill profile and culture is likely to be different. The same system life cycle processes can appear somewhat different when instantiated in the different practical settings of the stages of the life cycle.

From this diversity of technical, project, enterprise and agreement process application, it is not surprising that life cycles vary with 'apparently limitless variety' [ISO, 2002, p.56]. In their detail, they are unique to each system in accordance with the nature, purpose, timing, use and prevailing circumstances of the organisations creating and using them. They employ stages differently to satisfy widely contrasting business and risk mitigation strategies.

Using stages concurrently and in different orders can lead to life cycle forms with distinctly different characteristics. Sequential, incremental or evolutionary life cycle forms are frequently used or, alternatively, a suitable hybrid of these can be developed. The selection and development of such life cycle forms by an organisation depend on several factors, including the business context, the nature and complexity of the system, the stability of requirements, the technology opportunities, the need for different system capabilities at different times and the availability of budget and resources. Primary options encountered in life cycle management are considered and categorised in the following sections.

5.4 Life Cycle Classification

5.4.1 Sequential Forms

Unlike DoD-Std-499, IEEE 1220 and EIA-632, which are predominantly time-sequenced functional models, ISO/IEC15288 was intended to be (though is often not seen as) a substantially time-independent, object-oriented model, offering the opportunity to model virtually any life cycle form. Despite assertions to the contrary in the International Standard, the technical processes and the conditions of their application are most readily conceptualised and rationalised by many practitioners as a sequential representation. The International Standards complementary guidance report pays little attention to any other pattern of interaction that can emerge from the systems of life cycle processes [ISO, 2003].

The premise of sequential life cycle models is that an idealised chain of transformations advances a system from abstract need, through material solution, to activated service provision, ultimately to service expiry, retirement and disassembly.

At their most rudimentary, such life cycles are represented as piecewise linear progressions, each process transforming information, and inanimate and animate resources in a daisy chain of inputs and outputs from 'lust to dust'. This piecewise linear instantiation is thus the most basic life cycle form: the protozoan model of life cycles. When every process has been fully enacted, the information/material/energy flow and dependency then additively appears to have been a linearly-dependent sequence: a logical succession model that has total validity at its completion.

Ultimately, this logical sequence of information/materiel transformation steps must be conducted in order to create any system; each transformation step being dependent on the completion of its logical predecessor. At closure of all transformations, signified by complete equivalence between problem statement and compliant solution delivery, the unidirectional flow of information/materiel will have been punctuated by intermediate, logically significant and practically necessary information/materiel entities. These are the many work products that signify disciplined life cycle execution.

It was a compelling early step in the formalisation of life cycle management to adopt this logical sequence in order to share engineering reality across project teams – this was the part of Royce's message that fell on most fertile ground. Though the piecewise linear representation, and its depiction as a waterfall, is limited to the fundamental logic of unidirectional steps of system creation, the discipline that it brought (and continues to bring) has served engineering, and software engineering in particular, well. However, engineering actuality is subject to technological, problem-solving, resource and social limitations and constraints. Set in this reality, the instantiation of the logic of transformations and work products can appear and feel different – almost unrecognisably different.

Nonetheless, this sequential coupling of life cycle processes is compelling, for its expression of the terminating logic of a contiguously arranged set of generic system life cycle process is logically rigorous and technically simple. Its rational nature conditions practitioners to reason about progress and to measure this against linked indicators of achievement.

The re-expression of a horizontal linear sequential as a 'diagonal' linear sequence of the waterfall representation – as in Figure 5.9 – appears to add little. However, the waterfall model does subliminally imply an important property. Its fundamental cardinal steps of piecewise linearity are conveyed in the technical logic of the 'abscissa', whilst at the same time the 'ordinate' implies a down-flowing tally of managed project achievement. This may offer little information-wise but psychologically it is powerful, and this memorable icon has had more influence on life cycle thinking and management than any other model.

In addition, the waterfall can, through varying horizontal length, convey the relative duration of process execution and, by vertical displacement of process, convey a schedule of progress. It then moves toward presenting the instantiation of engineering process and the parametrics of project management. Graphically it begins to tie together engineering principles and project practice.

The irreversible downward flow in the waterfall metaphor has obvious practical limitations. Uni-directional process flow is at odds with the iteration, feedback and concurrency necessary to achieve concordance the models of Figure 4.9 and Figure 4.10. A simple enhancement is made by allowing process outputs (work products) to be fed back as inputs to the preceding processes: Royce's unsung contribution. This enables iterative refinement between adjacent processes and significantly improves the value of the waterfall model. In essence, this comes from the re-visitation, incremental refinement and mutual resolution across models.

With or without this enhancement the waterfall, or any quasi-linear model, depends on a sound understanding of need at the outset, on no technology surprises, on selection of a stable, acceptable solution, and on constancy in the operational environment. Individually, and more so jointly, these are improbable events. Certainly, weak prediction of behaviour of the inanimate – the human system elements – strains this approach.

Nonetheless, from its immense power to convey a logical functional representation of life cycle processes, this linear, sequential model continues to fundamentally influence the management of life cycles. For stable, persistent problems with significant precedence and mature technology, it works acceptably well and it brings helpful comprehension in more complex situations. Sequential models are criticised, yet widely employed. It is their simplicity that is both their merit and their deficiency.

5.4.2 Canonical Forms

Sequence is an inescapable feature of system state changes, and of steps in decision management and risk management. It is also a restriction of some engineering and project management methods/tools. However, in the face of uncertain or changing situations, linear chronological patterns are distinctly weak representations of all but the simplest or most abstract model.

A simple and very powerful step from the sequential life cycle model is achieved by introducing concomitance or punctuated delays in stage and process execution. This simple construct leads to a range of potentially complicated models that generally reflect risk and constraint handling strategies employed in the face of changing need, constraining resource profiles (including budget) or evolving technical opportunity. Used in combination, these two variations on strict linearity give rise to canonical patterns of life cycle execution. Though harder to plan, assess and control, the concurrency and repetition of similar but staggered sequence seen in canonical life cycles provides a more realistic representation of many life cycles where the supposed one-size-fits-all management expedient of linear sequence is, quite simply, a misrepresentation.

The most commonly encountered form of canonical life cycle is that associated with incremental delivery life cycles. Given the uncertainties and constraints of many defence acquisitions, this form of life cycle has been strongly advocated by Smart Acquisition, though it has been less strongly supported by related business procedure and practice.

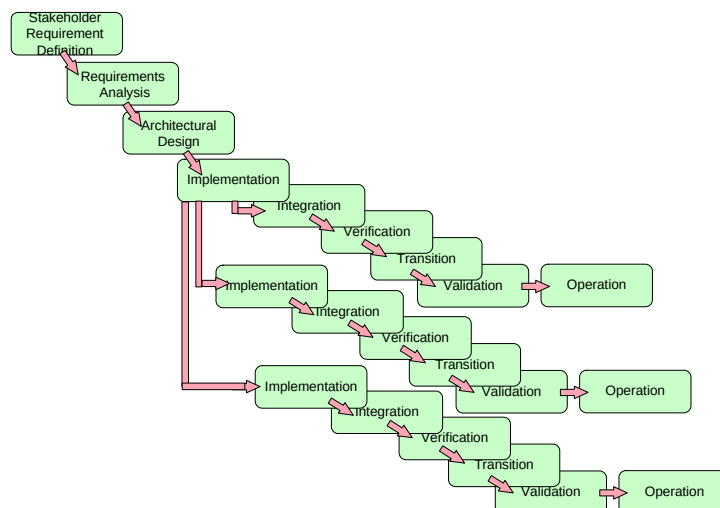


Figure 5.5 Canonical Life Cycle Models Enable Incremental Deliver

At its simplest the canonical life cycle model takes the form shown in Figure 5.5. It is particularly effective at rapid delivery of a limited though useful level of system capability, which may lack some ultimately required, but postponable service functionality altogether. As it progresses it allows some adaptability that can accommodate better levels of understanding or certainty regarding both problem and solution before commitment feedback to enhance ensuing increments of solution. It is based on a planned upgrade sequence aimed at:

- steps in value and comprehensiveness regarding overall need;
- minimum service disruption;

- scheduling to fit financial constraints.

This comes at the price of defining need and acceptance criteria that can be expressed in an increments fashion. As a strategy it requires commitment to a core architecture that can enable incremental upgrade, and time-profiled steps in the expenditure of resource.

The break from strict linear sequence arises from the necessity to bound risk-managed increments of life cycles. It leads to a pattern of staged delivery, in which a progression of parametric or functional augmentations extends earlier deliveries/deployments. This approach is characterised by solution advancement through accrual of capability or/and capacity.

Incremental life cycles may result from necessity or prudence; either way the life cycle form has similar features. It is characterised by increments of approval and delivery that equate to commitment steps in required (or available) enabling resources, labour and/or budget. In incremental system life cycle plans these steps may be governed by waypoints of enterprise management approval that sanction further tranches of funding and allocated resource in exactly the same way that stages in the life cycle are risk managed.

A benefit of time-profiled steps in the expenditure of resource or the settlement on a complete solution is that:

- uncertainties or changes in system services can be approached in a less risk-prone way;
- immature implementation risks or emerging technology opportunities can be progressively reconciled.

This strategy tends towards a managed, 'wait and see' strategy that is able to avoid undue and unnecessary exposure to commercial or technical risk. Progressive implementation improvements can be accommodated, lagging technology readiness/maturity can be reconciled, advantage can be taken of new technology breakthroughs, and new commercial offerings may emerge and be adapted to satisfy increments of required performance.

The result is that achievement of system functionality or performance through an incremental system life cycle is regulated primarily through the precedence associated with system need and the degree of confidence in the availability of assets and supplied technology. This permits early delivery of baseline capability – less than full capability, but a quicker route to achieve essential, pressing capability.

Clearly, the service that can be delivered by each step has to be of value and relevance to the overall need. This requires a succession of requirement and design models that are defined in a consistent, often accumulative fashion – a traceable progression of solution viewpoints – defined as steps towards a comprehensive, ultimate service solution, and not expressed as increments of solution quality.

A statement of stakeholder need that is expressed in an incremental fashion introduces constraints on a solution as a whole, and on the permissible increments of deployed capability. Principal among these is the identification of an architecture that can enable incremental upgrade to equipment function and performance with the necessary levels of backwards compatibility (including multi-variant interoperability), and that can be installed and transitioned to use in a manner consistent with agreed (and usually continuous) levels of service availability. To meet this, the design of a resilient architecture, possessing some degree of adaptability, extensibility and flexibility, thus presents a major systems engineering challenge. Viewed at a high level of abstraction, network capability acquisition is akin to this incremental life cycle strategy and is architected in line with these attributes.

The incremental life cycle form, and the architectures that suite it, generally permit managed technology insertion rather than forced reactions to implementation change, and thus permit a planned approach to system element obsolescence issues. Lower-order change does not ripple through echelons of interaction to result in top-level

Incremental delivery requires some measure of regression testing and overall regressive validation. Synchronised increments of testing, acceptance and release to deliver service is essential if uninterrupted, seamless capability is to be achieved. Successive deliveries of

progressive configuration may lead to the interaction and the support of multiple concurrent variants – and a heightened need for rigorous configuration control.

A technical threat arising from the increased opportunity to incorporate increments of approval in a canonical life cycle is the susceptibility to investment cutting. For this reason legitimate approval points may be deliberately downplayed by a conspiracy of acquirer's and supplier's protecting budget.

5.4.3 Hierarchical Forms

Figure 4.12 conveys the notion of hierarchy in system architecture. This hierarchy can be hugely influential in the representation of life cycles. The introduction of hierarchical patterns into life cycle forms is able to convey two system fundamentals: layers of organisational responsibility, with an attendant implication of viewpoints, and echelons of structural detail. The more complex model structure that presents this also allows greater opportunity to convey more complex activity flows and feedback paths.

The seemingly simple morphological step of symmetrically bending the waterfall model to form a 'V' shape about a fabrication activity 'midpoint' institutes a massively powerful change. The 'V' model conveys new attributes far beyond its parent waterfall representation.

Just as with the waterfall model it arose out of software engineering as a benefit seen by NASA in 1988 during its development of a Software Management and Assurance Programme [Forsberg, 1996, p.35]. It is for many systems engineering practitioners the archetypal life cycle form and it represents key life cycle features in a way that is accessible to engineers and enterprise management alike.

The left hand side of the V broadly speaking comprises design actions and the right hand side contains the build actions. It demonstrates the symmetry and equivalences between the design modelling and the build reality that is pivoted around implementation, or specifically fabrication. In the 'V' model illustrated in Figure 5.6, the horizontal linking feeds requirements and design forward to build, and build constraints backward to requirements and design. The introduction of these horizontal information paths opens up major feedback loops.

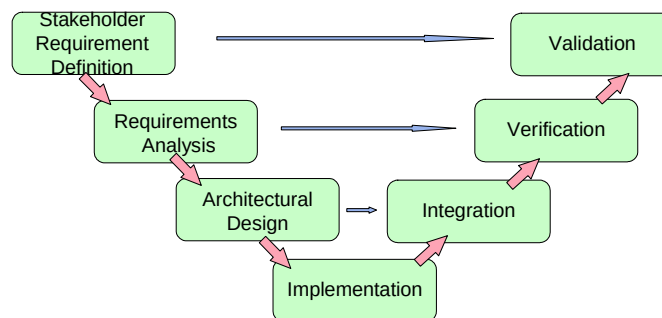


Figure 5.6 The 'V' Model Representation of System Life Cycle Processes

Firstly, hierarchical models, as conveyed in '3 Causes' structure of Figure 4.7 and in the contrasting 'V' representation of Figure 5.6, more easily depict exchanges between non-successive life cycle processes in the sequential waterfall. Secondly, it emphasises the essential symmetry that exists between design models and the corresponding integration and testing of constituent real elements. Thirdly, echelons of organisational responsibility begin to materialise from the hierarchy, allowing clearer assignment of (often) organisationally diffuse system responsibilities. These features make the 'V' model a highly illuminating and influential engineering representation of life cycle; in fact it is still the most common depiction used in systems engineering.

The basis of the 'V' model representation, in which the LHS is concerned with the system description (design) and the RHS is concerned with the material system (construction), results in an elegant and revealing symmetry of transformations pivoted on an axis that divides descriptions of reality and constructed reality. It is an engineering consequence of the

systems reasoning in §4.4.1. This symmetry applies to process transformations (as in Figure 4.6), work product exchanges, resources, controls and responsibilities. It also distinguishes the life cycles of man-made systems from organic natural systems, as shown in Figure 3.3.

The appearance of influential horizontal information paths arises because the system's description on the left and its built reality on the right possess an equivalent level of detail or granularity. This introduces the notion of hierarchical layers in system structure that is evident in Figure 4.14. It also introduces the associated notion of layers of organisations or of organisation function responsibilities conforming to a hierarchy of inter- or intra-organisation agreements that govern the vertical flows of information/material between.

5.4.4 Cyclic Forms

Natural evolution, the biological life cycles that drive it and the metaphors it provides us with, have strongly influenced thinking about man's management of a continuous sequence of changes to a technical system. The term system life cycle is itself a telltale metaphor. If the 'V' model is the most influential systems engineering life cycle representation, the spiral model is next.

Both animate and man-made systems beget, by different strategies as seen in §2, successor systems that are akin to their immediate predecessors. The cyclic nature of a succession of man-governed life flows controls, improves and manages generation after generation of evolving system.

Scientists and engineers have gained insight into many cyclic natural events and have seen that an understanding of the drivers and detail behind the repetition in their patterns offers a degree of predictability, risk mitigation and, ultimately, control of complex systems – natural and man-made. The essence of cyclic process structures is advancement along a well-trodden path of constructive transformations followed by feedback to provide an equivalent but refined starting point allowing repeated transformation at a greater degree of precision, and so on until the degree of exactness meets desired criteria. This repetition of patterns of transformation, the learning accrued from such cyclic process execution, and the natural coordinating influence of the various parties that in concert are responsible each cycle, is a cardinal feature of engineering and a focus of attention for systems engineering.

Because of its feedback of operational understanding into design so as to improve solutions, cyclic life cycle forms attract the biological connotation of evolution: a gradual change in the characteristics over successive generations or stages of development. This basic cycle of engineering creation is conveyed in Figure 5.7. Taken at its highest level of abstraction, it can represent the full life cycle of a man-made system, with the new starting point taken to be a fresh or replacement solution. In this regard, it reduces to four periods of a regenerative, sequential life cycle:

- problem appreciation;
- solution abstraction;
- solution definition;
- solution realisation.

The upper half of the figure addresses service; the lower half, the product that can deliver it. On the two axes shown lie the three definitions of effect, function and form, plus the resultant realised system capable of delivering service. For a single circuit of this representation, phase and stage display no distinction

In Figure 5.7, the counter-clockwise sense of direction, the skewed orientation, the angular emphasis of information/material transformation and the naming conventions chosen differ from traditional representations that trace to Boehm [Boehm, 1988]. Here, they are chosen to convey equivalent weighting to service and product viewpoints of a life cycle and to aid morphological rearrangement into other life cycle forms.

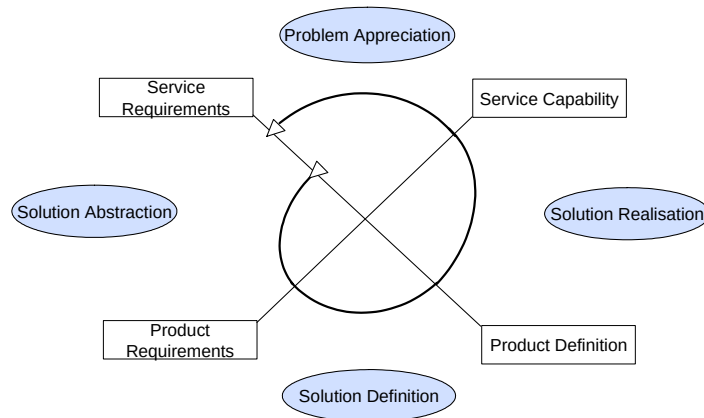


Figure 5.7 The Four Phases of the Product Creation Cycle

Natural evolution is a process of cyclical feedback that builds complexity in a risk-controlled way. It is about very small change in a large population over extended periods of time. Nature's evolutionary strategy is mimicked in man-made systems in the situation of mass consumer situations, where different companies simultaneously offer competing systems in the same free markets. This leads to market determination of the fittest system solutions, and overall in which direction the characteristics of these survivors will progress with time.

In this type of large volume situation evolutionary changes can radically impact architecture. For this reason, evolutionary life cycles are often considered in terms of generations, each having comparatively stable architecture (sometimes called epochs). Successive architectural configurations may need to co-exist and to inter-work, and this has implications on backwards compatibility and introduces resultant solution constraints.

However, for low volume markets and/or high value systems this strategy is not a practical model given the constraints of finite resources and limited exploration of fitness data. In contrast, a quite different employment of evolution principles is applied. Here, the benefits of successive refinement from cyclic re-visitation of life cycle processes is not so much gained from one delivered solution to a different delivered solution. Rather is through a succession of solution refinements within the span of a single life cycle that change is marked out; the evolutionary metaphor then aptly applies to the design and build of a solitary system.

Progressive cycles of ever more similar and more comprehensive solution definitions and problem expositions iteratively converge to acceptability.

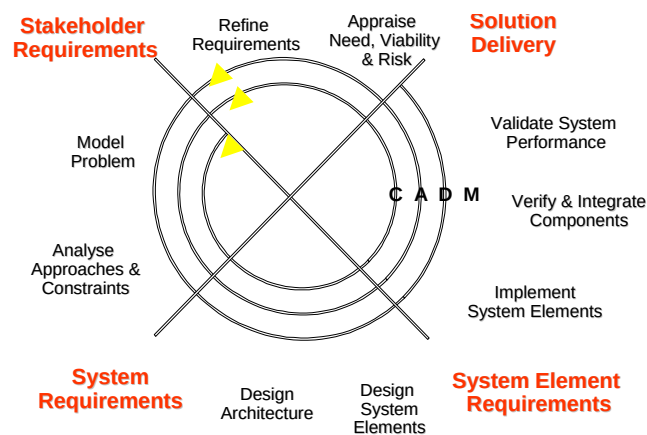


Figure 5.8 The Evolutionary or Learning Cycle Form of System Life Cycle Processes

This persistence of need means that continuous and seamless service delivery is required over long periods of time, often greater than the lifetimes of creator or service provider organisational regimes, or the continuity of an individual implementation technology.

This life cycle form is typically associated with the need for strategic capability. There is rarely a final solution and solutions continuously evolve. The idea of a final solution is not clearly defined and finish when the need for services finally terminates.

Evolutionary life cycles are thus associated with:

- a persistent underlying set of stakeholder requirements;
- routine user-developer contact during operation;
- revisiting needs, risks and opportunities;
- expensive architectural changes;
- planning horizon that are restricted by shifting goalposts;
- cyclic re-visitation of *all* the systems engineering processes.

Life cycle stages cycle through four phases, moving progressively outward in Figure 5.8 as solution entropy decreases and accumulated cost increases.

With sufficiently long timescales architecture is considered in terms of a number of discrete increments (epochs). These epochs, are essentially snapshots of an evolving technology base that periodically crosses thresholds of opportunity and manageable risk. They are characterised by step changes in architecture associated with major patterns or modes of material or data exchange or storage. They describe the evolutionary movement from current fielded and committed-to systems towards a future managed architecture of federated systems.

Evolutionary models may incorporate elements of incremental advancement. These jumps may arise as a result of:

- accumulated changes in operational environments;
- scalability limitations on existing solutions;
- new modes of system application;
- revolutionary implementation opportunities;
- breakthroughs in technology or performance thresholds;
- gross instances of technology obsolescence.

As a result of the cyclic repetition, concurrent stages within cycles, changes in organisation structure and shifts of influence at multiple levels of responsibility, the management of evolutionary life cycles depends on complex structures of interaction, consultation and decision-making that themselves evolve over time.

5.4.5 Organic Life Cycles

If natural evolution had not demonstrated it adequately, the Internet has shown how organic life cycles spawn incredible complexity, hugely powerful, and almost always unexpected and unprepared for results.

'Organic life cycle' systems are not designed, they evolve. The rules, ordinances or conventions that regulate their elaboration provide a continuous force that conditions advancement with a degree of blindness as to the outcome.

Yet within that continuousness of influence, in that drift of a mass of systems according to an overall organic life cycle, can be seen order and patterns against a background of illusory total chaos. As Ramos claims, 'the systems approach, if it is used wisely, is, at the least, a cure for chaos'. [*Ramo, 1998, p6*]

Organic life cycles are most commonly associated with levels of strategic service provision, typically to society as a whole, and are normally associated with long-lived or persistent need, with collective or mass user needs and acquisitions.

Management of organic life cycles is thus essentially one of consensus, even compromise, with organisations co-operating to form federated responsibility for system solutions. Success/failure is judged at high levels, with subordinate inefficiencies and sub-optimisation at discrete equipment level being viewed as being of little importance.

Organic life cycles emerge from the cooperation of many semiautonomous groups not accustomed or inclined to cooperate. It is difficult to arrange an operational structure to provide the resources, authority, responsibility, and funding to implement the solution. No honest-to-goodness customer-to-supplier relationship exists. 'No group can be said to be in a fully accepted position to define these needs, command the funds, and assume the responsibility for implementation and operation'. [*ibid*, p13]

Organic life cycle management becomes submerged in a detail of structure with no clearly identifiable hand of control. It results from the net effect of a multiplicity of managerial influences whose apparent co-ordination traces as much to common environmental influence and common technical opportunity as from closely-coupled decision making

Resources are not planned against capability objectives, they occur more by natural availability driven by prevailing social and economic factors. They may not be equitably contributed and system solutions become linked to the parochial interests of available resources.

Where self-imposed constraints can be agreed, e.g. global goals, collaborative regulation, international treaties, their influence is accepted as long as benefits accrue to a majority of parties and none are unacceptably compromised. In time federations fall, localised interest surface, new alliances rise and the dynamic tensions of cooperation and competition shift.

The flow of organic life cycles thus approximates to the characteristics of an ecosystem in a steady, typically drifting state, driven by the dynamics of a myriad of instances of operations that mimic those that drive evolutionary drift in biological systems, as seen in §3.4.2. Models of ecology and biological evolution have much relevance, and the scientific exploration of biological and social structures has offered many clues, and provided many metaphors.

The difficulty with multiple enterprise systems networked together is to avoid them slipping into organic life cycles

Any impression that organic life cycles are disordered is incorrect. Specifically they should be seen as chaotic with clear, long-term pattern of direction and purpose. Thus in organic life cycles system solutions progress naturally as an aggregated response to evolving environmental circumstances and to the continual advance of implementation opportunities. Nevertheless, discontinuities can and do occur in both need and solution. These are responded to most quickly and effectively by strong, though possibly short-term, single-point managerial authority that emerges from consultation and common consent.

These system life cycle characteristics can be recognised in military coalitions, and they influence and impact the formation and development of coalition capabilities. They can be planned for through a purposeful balance of proactive direction setting and reactive critical action, and in so doing can lay a basis of stability and plan for crisis states of required system service

5.5 Life Cycle Morphology

5.5.1 Life Cycles of Life Cycles

Because all system life cycles are composed of the same fundamental set of life cycle processes it is the emphasis of their interrelation and the meaningful patterns this creates distinction. All tracks back to the same underlying source of transformation and work products. It should thus be expected that each pattern, arising from a common logical and

physical origin, can be transmuted one to the other. A degree of equivalence between all process-based representations of all life cycle should exist, traceable to a dynamic concurrency of executed life cycle processes.

The nature of life cycles (and system architecture for identical systems reasoning) change according to the resolution employed by the observer. Resolved order and scale go hand in hand. As a consequence, the patterns and sequences of action, the organisational controls and influence, and the methods and tools employed appear differently, function differently and serve different purpose across a diversity of levels.

Because of the inherent complexity in most life cycles, different patterns can thus be abstracted from the same life cycle depending on the interests and responsibilities of different parties.

Enterprise management predominantly looks to linear sequences of stages and gates, preferably without canonical complications. Engineering tends towards and cyclic structures from which learning or understanding progressively emerges. Project management has to resolve this contention. It tends towards hierarchical views of responsibility and is influenced by task-based project methodology representations, which if anything favours canonical forms.

Systems generically are, as presented in Figure 4.16, composed of systems. Each system is associated with a life cycle. Thus it must be that, as a system-of-interest is described by a life cycle, this life cycle is composed of evermore detail life cycles. The form of a life cycle associated with any level of detail in the system-of-interests structure could conform to any of the classes described in the progression of forms in §5.4. Higher order systems contain and conceal greater complexity, and are more likely to be more effectively described by higher order models, lower order models applying to less complex systems which are subsumed within.

At the lowest levels of life cycle come those systems associated with science and basic technology. Here linear sequence is a highly workable approximation. Having a propensity to be a bottom-up driver for systems and their life cycles, this lowest layer has greatest opportunity to be decoupled from service delivery goals and to run asynchronously from other layers. To bring technology and products/services onto the same roadmap, NASA, with the maturity term in vogue, defined a technology maturity model. This is in effect a sequential, gate-regulated technology life cycle model that manages the lowest level in the classification structure in section §5.4, and gives visibility to potential or designated higher-order system projects. Science and technology research is largely predicated on this life cycle model.

Though it should only be taken as a guideline, starting from this base the hierarchical classification in §5.4 essentially corresponds to the system classification in Figure 3.2. A latent logic drives both.

Next up, canonical life cycle forms relate to the artificially created phenomena in technology systems. Parallel paths of development based on a level of common architecture detail favour the multiple technologies, trade-off exploration, higher risks, longer timescales and evolving application opportunities that characterise the a commercialised scene of tradable artefacts.

Mid-range come the hierarchical life cycle models that apply so well to products/services well. The 'V' model takes centre stage here, with echelons of technical and managerial being even more evident on other models. The emphasis on delivered service gives a better overall balance to these representations.

Cyclic models describe the directed evolution of enterprise systems in a win/lose environments. Fundamentally a structured learning and exploration approach, these life cycles are able to accommodate the unpredictabilities and non-linearities of human teams. Long periods of continual evolution, including controlled steps of architecture epochs can be accommodated in different cycles

Finally, the organic life cycle representation dominates the drift and occasional discontinuities in socio-politically influenced social systems. Co-evolution and self-adaptation conveys no overall control – patterns are the product of circumstance, not intention.

Just as in the classification of Figure 3.2 in which 'each level relies for its operation on all the levels below it' [Polanyi, 1968, pp1308-1311], so each level of life cycle complexity draws on the presence of subordinate forms and integrates them to evolve ever greater complexity within the next higher life cycle class. There are no sharp divisions between classes, and their properties and structural forms gradually metamorphose from one strata of complexity to another, resolving according to the scale of consideration and the degree of complexity.

5.5.2 Metamorphosis of Life Cycles

This metamorphosis of life cycle forms comes as each patterns formed by the system life cycle processes dissolve and re-arrange to create a new, higher- or lower-order patterns. Though difficult to convey statically, this is graphically presented by mappings that show the equivalence between the cardinal patterns. Each pattern is made up from the generic set of system life cycle process considered in §4.

The most traditional of the life cycle forms is the linear sequence of processes that describe the terminating logic of transformations and work product relationship. Unbending the diagonal stroke of the waterfall to a 'V' as in Figure 5.6 leads to what is almost an emblematic representation of systems engineering. The memorable symmetry of the 'V' and its forcible disclosure of the design/build equivalence demonstrate how different, closely related patterns bring home quite different features of life cycle structure and management.

A far more complicated and powerful transformation from the waterfall representation is achieved by associating system life cycle processes with the major responsibilities during the life cycle. This transposition is illustrated by circumscribing processes in a waterfall representation to the left of Figure 5.9 according to organisational responsibility for service, its quiescent state of capability, product and elements. This rearranges the dominant process associations into strata according to these responsibilities. The result is a mapping into the '3 Causes' in Figure 4.7. The dotted line boundaries to the left of Figure 5.9 map to the '3-Causes' hierarchy to the right, with operation lying beyond and to the right of this. This demonstrates how sequential is manifest as hierarchy.

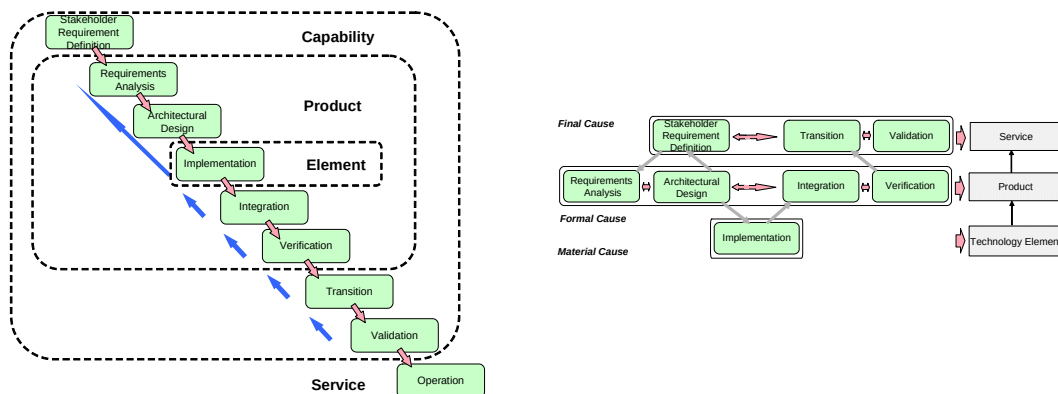


Figure 5.9 Sequential to Hierarchical Model Mapping

The hierarchical structure the right of Figure 5.9, and shown in greater detail and with more levels in Figure 4.14, emerges as a V by abstracting the design and build patterns to the left and right to create the general 'V' shape. This is shown in the superposition of these two models in Figure 5.10

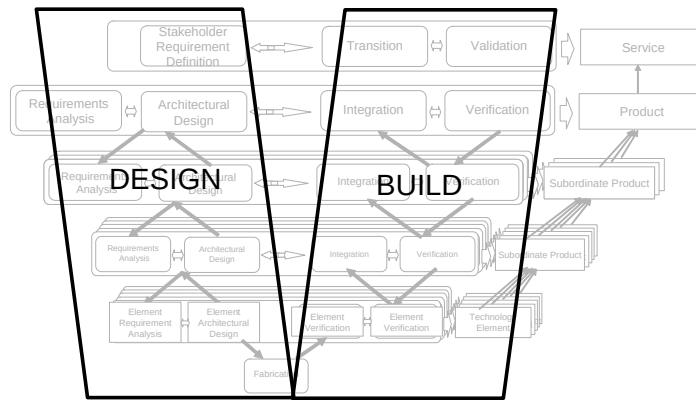


Figure 5.10 Overlaying a V Model on the Hierarchical Model

Finally, the cyclic or spiral representation can similarly be morphed from a hierarchical model, to bring out a phase structure that is concealed in its 3 Causes structure. Once again it is the hierarchical presentation of the life cycle process to the left of Figure 5.11 that provides the basis for mapping to the cyclic form. This also demonstrates the value of showing the cycle/spiral as an anticlockwise representation.

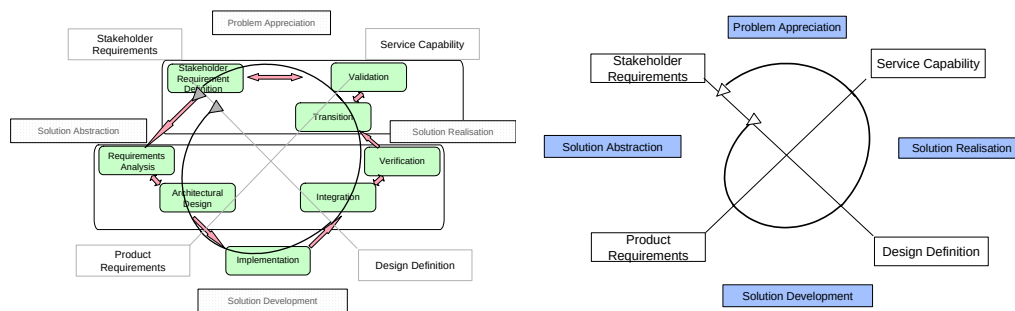


Figure 5.11 Overlaying a V Model on the Hierarchical Model

The engineering generality of the '3 Causes Model' of systems engineering provides an immensely powerful way of visualising and rationalising life cycles. Figures 5.9, 5.10 and 5.11 all trace to it. This means that any of the models in the classification of §5.4 can, with the exception of the organic life cycle, be discernibly morphed to any other. The organic life cycle lacks this type of system life cycle process rearrangement; its patterns comprise the ebb and flow of aggregate process execution and the patterns of social systems, as discussed in §10.5, are the most revealing.

5.6 Outcomes

The primary purpose of this chapter has been to demonstrate that the system life cycle processes of §4 themselves comprise a dynamic system: a system of processes with emphases of interaction and primacy that convey to the systems reasoning mind different patterns and different properties associated with these patterns.

At particular levels of abstraction the minutiae of life cycles resolve into distinct patterns with rationales that convey the goals, constraints, assets and dynamics that apply to that level of process structure. This is no surprise to the systems reasoning mind; it would expect no less.

It is less obvious and the rationality that is revealed is less familiar to non-technical members of business organisations. Life cycle management is thus where systems engineering and strategic business management most closely come together.

Broadly speaking, different life cycle patterns have been shown to predominate at each of the echelons in the complexity classification of man-made systems in Figure 3.2. This generality is indicative but not an absolute. These relationship points to the value of a hierarchy of life cycle pattern to resolve systemic order in systems engineering action, and to gain understanding from an overwhelming level of detail according to the interests and responsibility of the observer. It also emphasises that to achieve governance, attention needs to be focussed on those primary points in a system's life cycle that are most effectively reveal technical advancement and comprehension of risk. They mark out the principal staging points of a system's progression through its life cycle, and they relate to its evolving technical strategy and decision making.

Simple patterns make projects readily governable by enterprise management. They delineate periods of ostensibly lower risk and stability for projects, and this avoids the necessity for continuous involvement and 'micro-management' from above. They provide the headroom for engineering creativity without an overbearing structure of risk-controlling checks and balances.

Life cycle models are, nonetheless, as titled: models. Their simplification attract all the advantages but also all the attendant weaknesses detailed in §3.5.3. They are abstractions formulated in order to avoid the labyrinthine complexity of the actual, time-ordered threads of each individual life cycle. This abstraction therefore requires customary caution, and the complexity hiding its effects inevitably harbours risks. Iconic but idealised defence life cycle models such as CADMID are no exception, and the benefits and dangers of life cycle models of this type are analysed further in §9.4.2.

The simple life cycle classification scheme offered in this chapter provides insight into important issues of life cycle management. It coalesces technical thinking with that of project and business management. It assists the formulation of system life cycles/through life management plans and, based on this chapter, these are matters considered further in §9.5. In particular, the basic patterns presented here point to new arrangements that are more in keeping with MOD's present-day defence acquisition challenges.

6 CAPABILITY AND COMPETENCE

6.1 Prologue

A telling and unequivocal observation in the DIS augers well for systems engineering's future in MOD. 'It is little use investing in cutting-edge science unless systems engineering capability and vital long-term knowledge is maintained ... This demands a high level of systems engineering skills ... [and] is generally very important for maintaining our control of how we operate our Armed Forces' [DIS, 2005, p.59].

DIS goes on to say that 'ensuring the retention of systems engineering capability in the UK is in general important but needs, for implementation, to be underpinned by different behaviours in both us [MOD] and industry ... amongst other things, it requires clear leadership and investing in maintaining and refreshing skills and knowledge, and a marketing mindset' [*ibid*, p.64]. A message is that systems engineering skills and knowledge – basic ingredients of systems engineering competence – will depend upon a shift in business culture.

In support of this the DIS announces that 'along with industry, [MOD] needs to invest in maintaining and growing high quality systems engineering capability' [*ibid*, B1.3]. What this entails is not developed in the DIS, and how it could or should impact IPTs is left as work-in-progress. This growing of capability is the subject of DE&S initiatives, MOD/industry consultation and, doubtless, industry positioning.

What is made clear, however, is that advancements in systems engineering capability and competence go hand in hand. Accordingly, they are treated in a complementary fashion in this chapter. They form the two remaining images of the systems engineering triptych.

As presented in §4, the system life cycle processes are fundamental to a comprehensive and effective business environment in which systems engineering can be conducted. They help to build the bedrock of business excellence, and are a natural foundation for capability and competence, as portrayed in Figure 2.3.

Business processes are, however, not static. They exist in the dynamics of organisations that are continually subject to changing operational challenges and trading circumstance. Deming's quote that "process improvement is necessary as long as you are in competition" is a truism about the influence of an ever-changing trading environment; a business principle that Japan helped him demonstrate to the rest of the world.

This concept of continuous quality improvement is fundamentally built on the notion that processes, their identification, their codification and their managed improvement, are an essential element of business success. The worldwide impact of ISO 9001 is testament to this. As a result, capability improvement programs have swept through organisations. Relevant models and assessments of how processes are performed and effectively employed are not simply a vogue; they have been recognised in many quarters to be a business imperative. Systems engineering process improvement has received particular attention in the defence arena, and a strong interest in capability maturity models has in turn influenced systems engineering process definition and conduct.

These processes do not, however, stand in isolation. They should be mirrored in a compatible organisational structure. 'The systems approach suggests organizational innovation, and such innovation is usually required in our social structure if we are to handle our unsolved problems ... :the idea of modifying existing organizations or creating new ones' [Ramo, 1998, p.148]. Thus systems engineering, its processes and organisational structures co-evolve. The business organisation is the ultimate enabling system for systems engineering and it is commonly developed, too often trailing reactively, in concert with systems-of-interest.

Filling in the detail of Figure 2.3 is important in order to effectively establish a corporate, strategic systems engineering capability, and a corresponding individual practitioner and project team capability. In keeping with this, initial changes to organisation structure and responsibilities accompanied the introduction of MOD's Smart Acquisition, and these have

evolved in a manner that can be readily rationalised in terms of the system life cycle management model of ISO/IEC 15288.

The presence of systems engineering capability in individual organisations is clearly important. External visibility, recognition and proof that it exists are key to safeguarding acquirers' interest and to engendering mutual trust in collaborative ventures. They are germane to the relationship between government and industry, and especially so in the distinctly risk-prone defence business sector. All this was implicit in much that DIS stated. It was repeatedly emphasised throughout DIS's development of a policy that is intended be mutually advantageous to MOD and to the various echelons of the industrial supply chain.

It is therefore natural that systems engineering capability should be seen to lie at the heart of a change programme that affects both sides of the trading interface. 'The Department [the DE&S] is committed to driving this change agenda. We will be looking for parallel commitment from industry to ... invest in growing and maintaining a high-quality systems engineering capability within the UK, at all levels in the supply chain' [DIS, 2005, C1.50]. Systems engineering is rightly seen not just as the preserve of prime suppliers but relevant to the whole supply chain, as Figure 4.10 illustrates. The UK MOD emphasises this and its importance merited reiteration in DIS: 'along with industry, [MOD] needs to invest in maintaining and growing high quality systems engineering capability, at all levels in the supply chain' [ibid, B1.3]. 'It is fundamental to the partner relationship envisioned for two mutually dependent communities, and for the health of 'UK plc' [ibid, C1.50].

Ultimately, however, process, procedures, tools and integrated environments only go so far towards meeting the challenges inherent in complex systems. They establish a preordained stereotypy in a world where each and every problem to be solved is in actuality different in its detail, and often in its structure and context. Just as in many man-made systems, it is the individual operator or operative that in the end finely matches apparatus to purpose. This is where organisational process and enabling infrastructure capability give way to practitioner competence; where a regime of education, training, skill development and mentoring takes over to deliver the cutting edge of achievement. Selecting the point of this handover is a delicate, shifting and hugely profound decision. It has repeatedly been misjudged by many organisations and it may still be seen as a major point of failure in instituting effective systems engineering practice and culture.

As the value of systems engineering discipline becomes increasingly recognised by society, so there is a corresponding demand firstly to attract able systems engineering staff into business organisations, then to develop them according to best principles and practice, and ultimately to ensure that they can employ their competence to meet evermore demanding technical system responsibilities as they arise out of societies needs. Systems engineering competence is virtually indispensable for commercial success of organisations that trade in complex products and services, and/or that rely on complex systems in order to conduct their businesses – not least on the ubiquitous in-house information systems. The project is the principal context for conducting systems engineering best practice. Consequently, individual systems engineering competence needs to be shared and aggregated to become team competence, which for MOD's business especially means in the IPTs.

As the retiring Chief of Defence Procurement, Sir Peter Spencer, said in March 2007, 'we must also recognise that we still have a long way to go particularly in consistent compliance with best practice. Above all, we must invest more heavily and systematically in the professional skills of our people' [Preview, 2007].

In support of this, and despite still being diversely formulated across businesses and professional bodies, competence frameworks look set to be a catalyst for more effective conduct of systems engineering by the individual, by the project team, by the organisation and by the 'profession' as a whole. Just as capability models have multiplied [Sheard, 1997], so competency frameworks are in vogue. Accordingly, educational attainment, relevant experience, accreditation and certification are issues of ascending importance for systems engineering.

In considering the talents of those engaged in the creation of complex systems, Ramo reflected on 'why systems work [i.e. why systems engineering] is not likely to be done well by amateurs ... In describing the nature of the systems approach, we ... think of it as a multi-parameter problem ... we seek to be logical and objective, to consider all of the interactions

among the various parts of the problems ... we give attention ... to setting down the goals with clarity and using [them] to create criteria for evaluation of alternative approaches. All of this ... sounds like just ordinary common sense' [Ramo, 1998, p.57]. His rhetoric could be taken as an admonition targeted at a habitual accusation levelled against systems engineering: "no more than common sense".

Ramo drives the point home: 'what, then, distinguishes the professional systems engineer, the team of 'techno-political-econo-socio-experts' who will approach the problem in a 'professional' way?' He concludes that it 'requires individuals who are highly trained, who have great experience, and who by the practical selection process of success or failure emerge as especially gifted in the kind of analysis and design involved'. This is a prescription for systems engineering aptitude, knowledge, skill and experience, and it is a formula for systems engineering competence that is advanced in this chapter.

6.2 Synopsis

Starting from ISO's process-based bedrock description of systems engineering, this chapter considers how the provisions of the systems engineering International Standard may be transformed into a compatible definition of an organisation's capability to conduct systems-oriented business. It then turns to the less mature and presently more fluid topic of a corresponding codification of individual systems engineering competence.

These are the remaining two faces of the systems engineering triptych proposed in this research – capability and competence – and they are addressed in this order.

The paramount enabling system of any product is the business organisation that creates it. Because the system principles embodied in ISO/IEC 15288 are styled and presented to be relevant not just to the delivered system-of-interest but, by a re-focus of concerns, to also apply to enabling systems, it is therefore natural to judge the organisation itself against the ISO model. A route to effecting such assessments of capability is outlined. It requires an essential step that fills a current gap: the definition of process work product indicators of capability that match the transformations of the ISO model. Consequently, a mapping is described between the transformations of the ISO process descriptions and a complementary set of work products – predominantly, the information items and material/product objects of systems engineering.

The technique employed in this investigation to define these work products is outlined and the detail of the resulting inventory is presented in Appendix A, *Systems Engineering Capability Indicators*. From this it becomes possible for assessors to tally observed day-to-day indicators of mature systems business action against a model of expected systems engineering process work products, and from this together with other indicators to then assign recognised levels of capability.

The set of work products defined in Appendix A is also a step toward improving the longer term quality of codified system life cycle processes, for it tackles a major weakness in current ISO life cycle process models. This is the failing to describe processes by models that exhibit concordance between a rational, practical set of transformations and a rational, practical set of inputs/output work products. Appendix A takes the first step in a necessary direction of evolution that ISO/IEC 15288 must face if it is to overcome its many lower-order, irritating but tolerable inconsistencies.

Moving to the third and final face of the systems engineering triptych, that of competence, this investigation offers indications as to why formulations of systems engineering competence have not been widely accepted and why the definition of a framework that tallies with ratified system life cycle processes remains a far-from-resolved exercise. Taking account of these difficulties, this investigation explored one approach and then built a limited prototype framework to demonstrate its viability.

The approach taken advocates an underlying data set – in essence a populated meta-model of systems engineering competence – that can be transformed according to different, consistent viewpoints in order to meet the needs of the primary stakeholder classes. For practitioners it leads to a proficiency-oriented view, for the organisation a role-oriented view,

for professional bodies an accreditation-oriented view and for business management a procedure-oriented view.

Models of systems engineering competence are multifarious by nature – a reflection of the diversity of human factors, some of which that are briefly considered in §10.5.1. Because the dimensions and parameters that frame this competence space need to be open to multiple routes of navigation according to the purpose and circumstance of user classes, the term framework aptly applies to the codification of competence.

The competence framework proposed by this investigation is based on the interaction of four substantially orthogonal attributes of competence. Individually present in other representations of competence, these attributes are functionally combined here in a manner that exploits the logic, structure and maturity of the ISO system life cycle processes. In its detail it is then able to leverage a considerable body of authenticated data that is embodied in other representations of systems engineering competence.

Starting from the ISO system life cycle processes, which act in the manner of a primary independent variable of the framework, a mapping is conducted first with the activity/role structure and then to the supporting data of the long-standing, well-proven approach taken by SFIA, the Skills Framework for the Information Age [SFIA, 2003]. Though SFIA is intended to apply to the building and use of IT-intensive systems, it is seen to contain valuable systems generality. This quality both aids and substantiates the approach taken.

In addition, from an analysis of how competence value can be metricated, SFIA is shown to provide an appropriately defined scale of competence attainment for systems engineering. This scale then acts as the primary dependent variable of the framework. Each intersection of dependent and independent variable establishes cells of competence space that are constructed from the four-attribute representation of competence. In this way, multiple influences are drawn together in a logical framework that can be populated with ever finer attribute detail.

A partial prototype of this competence framework is presented in Appendix B, *Systems Engineering Competence Framework*. Limited to a tabulation of the primary dependent and independent dimensions of the framework, and corroborated by the detail that can be referenced within the SFIA data, Appendix B illustrates the viability of this approach.

6.3 Systems Engineering Capability

6.3.1 From Business Process to Organisational Capability

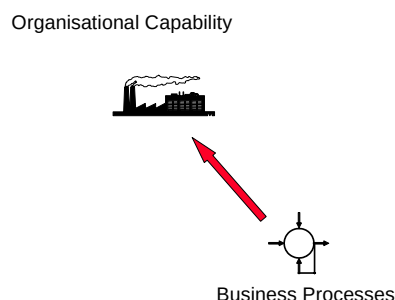


Figure 6.1 Deriving Systems Engineering Organisational Capability from Business Process

The means for assessing the maturity of an organisation's capability to perform systems engineering, as defined in the provisions of ISO/IEC 15288, is the focus of the first half of this chapter. It is based on the translation of business process into criteria for organisational capability, as shown schematically in Figure 6.1 (a part of Figure 2.3).

Essentially, an organisation's systems engineering capability may be equated to its ability to dependably conduct a trade in products and services that traceably flows from:

- a knowledge/experience base;
- competent staff;
- enabling systems;
- a set of relevant organisational assets.

Formalising the development and institutionalisation of a systems engineering capability serves the organisation by providing:

- an influence on systems engineering policies, practices and procedures;
- a driver for process implementation - method, tools, resources;
- a basis for measurement and continuous improvement;
- a yardstick for performance evaluation of an organisation;
- a common approach for determining supplier acceptability.

An organisation's systems engineering capability may come into being by the happenstance of repeated trading obligations but it rarely becomes a refined and effective enabler, that is continuously and reliably available to meet future business commitments, without the highly structured and committed drive of organisational improvement. Like any project in the organisation, this requires planning, assessment and control to effect a successful outcome. The model against which planning is conducted, the criteria against which assessments are made, and the management actions to secure steady, well-directed progress all depend on a reference of recognised good practice.

To date, the DoD-promoted CMMI model is the most widely used, systems engineering-oriented capability reference model used in the defence community. Nevertheless, with its foundation of systems reasoning and its compliance with an inter-/intra-organisational business trading model, the ISO systems engineering International Standard presently provides the best structured, most consistent, and perhaps most influential reference model to be compliant with. Coupled with the well-developed and proved ISO approach to process assessment [ISO, 2002b], few steps need to be completed to provide an internationally ratified, commercial provider-independent instrument for the assessment of any organisation's capability to perform systems engineering.

The commercial benefits of compliance to this type of reference model are widely upheld, particularly in the face of DoD acquisition authorities mandating it in one form or another. The DoD's funding of the development of a range of assessment instruments, including CMMI, is evidence of their belief in the long term benefits. Compliance is demonstrated according to a objective approach to process assessment by an organisation itself, or by a second or third party, with the objective of:

- understanding the state of its own processes for process improvement;
- determining the suitability of its own processes for a particular requirement or class of requirements;
- determining the suitability of another organisation's processes for a particular contract or class of contracts.

Most approaches to organisational capability assessment, whatever the nature of the processes that are the focus of attention, have followed a five-level grading structure. Levels 1 to 3 generally apply to active, declared improvement, Level 4 is reactive, Level 5 is proactive. Generally, these steps approximate to the Shewhart Cycle in accordance with:

1. do;
2. plan, do;
3. plan, do, check;
4. plan, do, check, act (to correct the delivered system);
5. plan, do, check, act (to correct) plus act (to improve business process).

This last step revisits the underlying model employed. This usually entails a tailoring of an institutionalised approach, in which the ritualistic observance of procedure has given way to informed, context responsive (though still declared) application. In this last step, the disciplined execution of rote has given way to disciplined adaptation according to prevailing circumstance – a totally different regime to that of heroic individuals at Level 1.

Elements of each of these levels may co-exist within and across all processes under consideration. However, different assessment techniques each arrive at some mean value that signifies, as objectively as possible, the prevailing status of capability maturity. This may be consolidated as steps of maturity for pre-selected groups of process (a stepped model) or be a continuous profile over all or chosen processes (a continuous model)

As with any model, each capability assessment reference model has its strength, each has its limitations. The overlaps in capability process models clearly illustrate that boundaries drawn around areas of interest can be both difficult to define, unhelpfully constraining and lead to distortions of abstraction level. In the continuum of options a plethora of process maturity models has led to many perspectives of broadly similar process models, each offering subtly differing levels of abstraction and differing foci of interest. Duplication, contention and competition are an almost inevitable consequence of what has been a rush to provide capability models that serve particular organisational perspectives.

The inset to the left of Figure 6.2 illustrates the profusion of initiatives to address process improvement in the areas of software engineering, and more generally in systems engineering. Out of this detail, the particular time threads relevant to this investigation are shown. CMMI is shown with a foot in each camp, whereas EIA 731 and the impending ISO/IEC 15504 Part 6 target systems engineering generality.

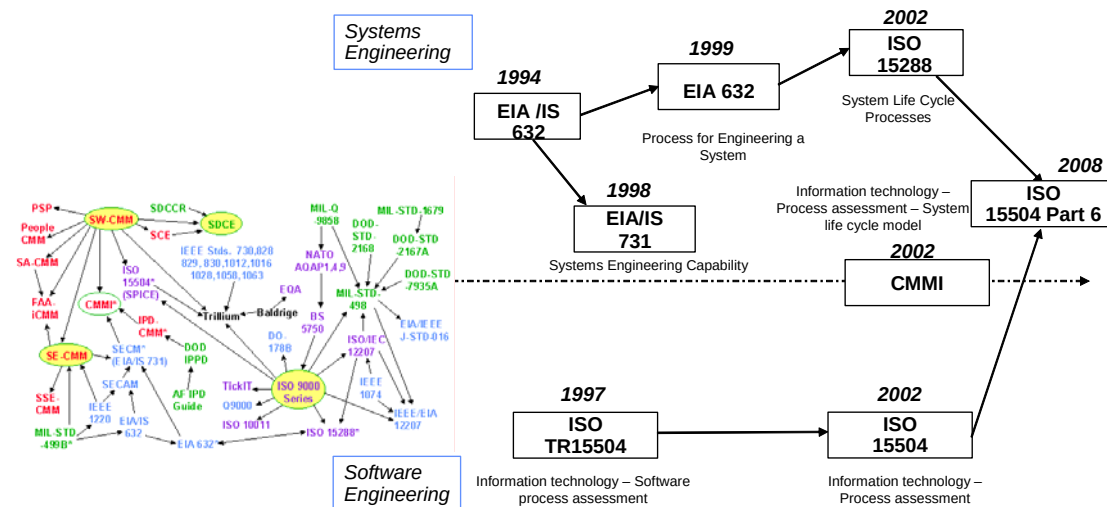


Figure 6.2 A Timeline of Influential Systems Engineering Capability Assessment Standards

A determination of the level and extent of systems engineering capability may take the form of a first party assessment for the purposes of self-improvement or a second/third party assessment to assist a current or prospective business relationship. To this end, ISO is acting to define an assessment reference model for systems engineering that conforms to ISO/IEC 15504 and is based on ISO/IEC 15288. This may assist in transforming ISO/IEC 15288 into an organisational imperative.

It is important that capability assessment actions are evidence-based to achieve objectivity, uniformity and confidence. This requires tangible corroboration of business process action and achievement. Ultimately, this evidence will take the form of the delivered product or service, but prior to this, and indeed throughout the lifetime of any project, documented information is key to project success. This information relates to the conduct and achievements of the project, to the resultant products that are created, and/or to the services delivered. Disciplined information recording and project-generated documentation underlie all effective projects and the instruments for their assessment.

Assessment findings therefore necessarily draw on the evidence of data collected from objective, documented reviews and from deliverables to clients and other stakeholders. In addition, well-structured dialogue with management and project team members and, as appropriate, relevant project stakeholders can contribute valuable information

Project documentation is a general term for the records of information that convey meaning, significance or intention in a project. These information items may be in any form and recorded using any medium. They will generally be long-lasting, though are likely to change with time, and should be capable of being read by humans or machines and interpreted in context.

Information items are the means by which information is conveyed in a consistent and dependable way between cooperating parties and individuals in a project. They are the basis for shared, common project action, responsibility and achievement. They define future action (plans or procedures) and past action (decisions, accomplishments and realisations), and so lays a foundation for repeatability of project action and for assessments. In particular, they provide the tangible information thread that links client need to validated solution.

They are thus essential to the effective and efficient operation of a project. The larger and longer the project is, so the greater the need for the unifying power of cardinal information items. The greater the business risk, the greater the need for the formality and audit of documents and records. However, ISO 9000:2000 is explicit regarding the level of documented information: 'the generation of documentation should not be an end in itself but should be a value-adding activity'. This requires a prudent balancing of short-term project tasking, and ongoing maintenance, against long-term benefit and risk reduction. For this reason, systems engineering capability reference models do not normally mandate which information items shall be prepared or when. Nor do they stipulate the information item contents or the formality of document control that should be applied. Rather they exemplify, with each project making a judgment on the extent and formality of documentation required, and how this evolves with time.

Therefore, to assist assessment of the outcomes, maturity reference models commonly define expected work products. These are not intended to be all-inclusive nor applied in their entirety; subsets appropriate to the context may be selected, and possibly augmented with additional indicators. Thus reference model work products and their characteristics should be seen as a starting point for considering whether, given the context, the necessary artefacts (information items and product objects) are indicating the intended purpose of the process, not merely acting as a slavish, tick-in-the-box check list of what every organisation must have.

In ISO/IEC 15288, work products are evident at all levels in the process hierarchy of Figure 2. Stages, processes and activities each have the same class properties, and thus have inputs and outputs that may be viewed as work products. Though capability assessment has predominantly has a generic process-based focus, the evidence of managed life cycle stage introduces associated work product descriptions is important and often distinctive, with the particular constraints, methods and resources introducing leading to different characteristics.

As in ISO/IEC 15288, processes may be defined in terms of discrete, complete transformational objects. When and how these transformations are conducted is independent of their definition. Repeatedly invocation and quasi-continuous execution both lead to work products that may only be considered complete when the life cycle itself has run its course. Until then, partial steps of completeness characterise progress through the life cycle. Consequently, the cardinal stages of continuously evolving, partially complete work products apply. These stages are typically functions of planned progression, resource use, actual progress, decision-making and risk management, i.e. they are stage gate related, and need to be clearly defined, not least because of their importance to enterprise management's involvement at decision gates.

Partially complete states of work products are therefore a feature of stages, and of progression thorough the life cycle. For example, using CADMID, work products associated with the Implementation Process may only be outlined at the Concept Stage. They may be understood at a greater, defined level of detail in the Assessment Stage and only rigorously defined in a Manufacture Stage. Similarly, the requirement for a disposal enabling system may steadily evolve in a controlled way over much of the life cycle.

The composition and characteristics of system life cycle process work products are thus functions of when they are viewed in the life cycle. This definition and management of the evolving work product states is key to time line or dynamic models of system life cycles, and is an important (and challenging) factor in define the achievement criteria at successive stage gates.

6.3.2 Discrete or Integrated Capability Models

The original Capability Maturity Models (CMMs) were developed from quality management principles to improve organisational performance in the domain of a single (usually business critical) discipline or functional contributor. Although many aspects of organisational best practices are generic by nature, business process models typically focussed on individual engineering disciplines, and were organised and expressed with a view towards specific technical practices. These related to an individual implementation technology/medium, such as software, or to a generic system attribute that needed to be addressed with particular emphasis to ensure overall quality, such as safety.

The development of technology-oriented process models, and the experience of assessing an organisation's maturity when practising them, highlights that it is neither easy, and can be detrimental, to set artificial boundaries to assessed purpose and outcomes for what is a generic set of processes. A specific discipline focus can weaken the wider, holistic system view that is important if the products and services that organisations trade in are to fully meet customer needs and expected quality levels.

The experience of developing the ISO/IEC 12207 software life cycle process model illustrated the necessity of the overarching holistic systems life cycle model in order to establish the technical and business context for software engineering within an organisation. This is evidenced by the 140+ references to system in the ISO/IEC 12207 standard and its definition of particular processes with explicit system-related titles [ISO, 1995].

This complementary system/technology relationship in process definition was explored by several bodies, including the Software Engineering Institute (SEI), as they moved to integrate implementation technology process models, particularly software engineering, with systems engineering process models.

A difficulty encountered in separating an implementation technology viewpoint from a system viewpoint arises from the lack of well-defined boundary to distinguish them; a point illustrated in Figure 10 with regard to the systems engineering/human factors relationship. Looking upward from a software engineering view toward a system perspective is technically necessary in order to achieve holistic product or service quality. Thus the move to integrate software engineering process assessment models, for example, and systems engineering process assessment models has merit.

Such integration, however, has a price. The bringing together of models from different communities of technical interest has led to results that err toward amalgamating independent models rather than rationalising new order. The result has tended towards large models and inherited inconsistencies. Such models can:

- carry decreasing returns, even reduced returns, for increasing size;
- be inclined toward detailed prescriptiveness;
- be applied outside the range of validity of their originating parts;
- have a tendency to continue growing.

In varying measure, these characteristics can be detected in assessment models, for example, the SEI's CMMI model, and notably the Federal Aviation Administration (FAA) in its iCMM model. Thus far ISO and the IEC have largely only followed this system-plus-technology route for models that address particular holistic system attributes, e.g. security, safety, environmental issues. Such models address the technical emphasis in processes and attendant life cycle management issues in as far as they lead to the specific system attributes. Some encroach into the detail of specified method and can be unnecessarily prescriptive.

Both systems engineering process models/assessments and their software engineering counterparts are concerned fundamentally with the execution of technical processes that are conducted in the context of projects subject to the direction and support of enterprise management.

Though substantially common underlying engineering principles apply to multi-technology systems and to software, their interpretation can lead to distinct practical consequences. This is a result of:

- different levels in the hierarchy of product or service structure,
- the state of the system/software in its life cycle, e.g. life cycle stage,
- the nature and characteristics of the sector or market in the which the product or service is being traded,

The assessment models and the assessor experience, knowledge and skills must accommodate these differences. For example, assessment methods and assessor skills need to be able to cope with the gradation from system to implementation technology.

This requires more than an appreciation of how system principles are expressed in the context of the diverse physical characteristics of particular implementation media. Different constraints clearly arise from the radically different nature and fabrication actions associated with different implementation medium (exemplified in ISO/IEC 15288's Implementation Process by hardware, software or humans). However, the implementation technology gives rise to many more constraints: the descriptive forms and models, practicalities of methods, specialist community terminology, the established education and training of disciplines, the custom and practice of skills, the tools that address relevant functional and physical attributes, measurement technology, diagnostic strategies, the social obligations and risks of use and disposal. They can present radically different landscapes for assessments of what otherwise is a largely common set of technical processes. The process indicators they give rise to can convey very different appearances.

The lower the level in the hierarchy of product or service structure (and the closer to the influences of implementation technology), the more these technical differences distinguish individual assessments. The higher the level in the hierarchy of product or service structure (and the closer to systems capability and its services in the environment of operation), the more customer need, market sector factors and trading influences will shape assessment.

These contrasting assessment attributes mean that ISO/IEC 15288 and ISO/IEC 12207 do not exhibit a neat symmetry. System assessments and software assessments depend on complementary rather than congruent interpretation in order to describe the continuum of concepts that span system to the implementation technology focus of software. This relies on differences in the knowledge, experience and empathies of assessment teams and individual members.

Whilst the span of project team disciplines rather than project team size impacts assessment, team size alone can influence process assessment. Systems tend toward larger project teams; software tends towards small teams and the autonomy of the specialist. Disciplined process execution is key to large teams and system life cycles, whereas creativity, responsiveness and greater self-governance may characterise software processes and software life cycle management. The resulting cultural biases may lead to corresponding quality regimes, and these their differences require appropriate assessment appreciation and knowledge.

In addition, the wider range of process interactions and life cycle forms encountered in systems engineering means that ISO/IEC 15288 is a more object oriented model in contrast to the with the more functionally oriented and more sequential process descriptions in ISO/IEC 12207. This requires assessors to recognise how objects map into the dynamics of system life cycle management and how this influences coherent, risk-managed project team.

There are different flavours of harmonisation of implementation technology process assessment models and systems engineering process assessment models. Size is one factor. The ISO approach has been to maintain comparatively small standards, for example the 15504 Part 5 software model at 206 pages compared with SEI's CMMI software/system

model at 726 pages in order to counter the adverse attributes of size and complexity. This also makes ratification easier to achieve, particularly since ISO's development procedures are international and more stringent (and thus more electively endorsed) than those conducted by commercially driven organisations such as the SEI.

Compound (or integrated) capability maturity models have evolved to provide the guidance from more than one single-discipline model; their intent is to relate the disciplines, remove redundancy, and integrate best practices into a common reference model with a common structure and common terminology. In this way, a compound model is intended to span multiple disciplines. Process improvement efforts can then be aligned across the multi-disciplinary aspects of projects or of enterprises. Thus, despite the dangers of model size and complexity, many assessment models have moved in this integrative direction.

One of the widest scopes of an integrated model is offered by the FAA-iCMM [iCMM, 2001]. At 480 pages Version 2 is more compact than the SEI's CMMI. As a result, iCMM purpose is to guide improvement of FAA's engineering, management and acquisition processes in an integrated, effective and efficient way.

The portmanteau model was assembled to include best practices and concepts from several standards that relate to process improvement. iCMM is an amalgamation of 12 standards, including ISO/IEC 15288, ISO/IEC 12207 and ISO/IEC 15504 Part 5. It shows evidence of 'cherry picking' rather than of a designed, balanced, coherent model– a selective coalescence rather than a cohesion strategy.

The strategy of its future continuing growth is hinted at from the statement that "other models and standards of interest to the FAA also offer guidance for organisational improvement". Additional extensions have been proposed for future releases of the iCMM. Already extensions have been completed that bring together the practices of 4 safety standards and 4 security standards, and they are "packaged for use" with existing guidance found in iCMM.

The FAA claims that "having an integrated model on which to base its process improvements is a real advantage. The alternative approach of using multiple standards is simply too expensive, too confusing, and too difficult". However, the iCMM Version 2 model is still being evaluated and this claim has yet to be objectively substantiated.

Further, the FAA sees that, as it moves to using integrated product teams comprising various disciplines, the interrelationship of their work requires better-integrated process guidance. The discipline models that are integrated at present fall well short of those required for integrated project teams dealing with multiple implementation technology disciplines and with through life functional contributions. The model would need to grow very much to meet these needs.

6.3.3 Indicators

ISO/IEC 15504 Part 2 defines the capability dimension of process assessment and this forms a basis for any compliant Process Assessment Model. The measure of capability in Part 2 is defined in terms of a set of given levels as shown in Table 6.1.

Level 1	<i>Performed Process:</i> An implemented process achieves its process purpose.
Level 2	<i>Managed Process:</i> A process is implemented in a managed fashion and its work products are appropriately established, controlled and maintained.
Level 3	<i>Established Process:</i> A process is implemented using a defined process is capable of achieving its process outcomes.
Level 4	<i>Predictable Process:</i> A process operates within defined limits to achieve its process outcomes.
Level 5	<i>Optimising Process:</i> A process is continuously improved to meet relevant current and projected business goals.

Table 6.1 ISO/IEC 15504 Process Compliance Levels [ISO, 2004]

For Levels 2 to 5, the process attributes are defined in terms of indicators. These indicators convey the extent of achievement of an implemented process and are concerned with significant activities (the practices), resources and results (the work products).

Process work products are particularly important in process assessment models since they are tangible evidence of the execution of a process transformation.

A process assessment model describes an approach that permits the collection of evidence and the rating of process capability in a disciplined and consistent way. Each model is based on a set of indicators that ensure consistency and repeatability in the demonstration of the achievement of the process attributes associated with capability.

Process assessment models employ work products as a fundamental set of indicators. Work products are described in terms of recognisable characteristics, and the presence or absence of these characteristics assist assessors:

- to recognise and reason about work products;
- to associate evidence with a particular process;
- to ultimately judge and assess the attained capability rating of a process.

Process assessment has two dimensions, a process dimension and a capability dimension.

The process dimension is provided by a process reference model that defines a set of processes in a life cycle. A process assessment model is based on the process descriptions from an ISO/IEC 15504-compliant process reference model. The process reference model acts as the reference source for the process(es) of interest and it may be the basis for one or more process assessment models. These processes are described in terms of process purpose and outcomes, and information that establishes the relationships between them.

Unlike its predecessor Technical Report, the International Standard ISO/IEC 15504 does not define a process reference model. It does, however, define the common requirements for any process assessment model. This enables consistent comparison of assessments using different process assessment models based upon the same process reference model. ISO/IEC 15288 is the process reference model for system life cycle processes.

The capability dimension of process assessment is defined in ISO/IEC 15504 Part 2. The measure of capability is based upon a set of process attributes that are used to indicate whether a process has reached a given level. Work products are part of this set of process attributes. There are nine process attributes and each attribute measures a particular aspect of process capability. They are grouped into a five level scale of process capability in Table 6.2:

There are 8 process attributes that span Levels 2 to 5, and each attribute measures a particular aspect of process capability, as shown in Table 6.2. These attributes are associated with determining process capability at Levels 2 through to 5.

Level 2:	Performance management attribute Work product management attribute
Level 3:	Process definition attribute Process deployment attribute
Level 4:	Process measurement attribute Process control attribute
Level 5:	Process innovation attribute Process optimisation attribute

Table 6.2 Process Attributes According to Level [ISO, 2004]

To assist assessment of the degree of achievement of the process capability using work product evidence, the characteristics of the 8 classes of generic work product are associated with the 8 process attributes (levels 2 to 5 of the capability dimension), [ISO,2008, Clause 6]

Stage work products, being more sequence related, include individual or composites of system life cycle process work products, defined in conformity with stages (literally) of their completion/refinement. Stage work products are thus dependent on the particular nature of the life cycle. They are application/business domain dependent and enterprise management custom and practice dictated. Defined as a function of achievement, risk, resources and decision making information according to a progression of cardinal states of system creation and use, they are intimately coupled to stage gates and provide baselines of evidence.

These stages of continuously evolving, partially complete work products can and should be defined across similar lines of business and, as appropriate, tailored for individual system life cycles. They are importance for consistency in technical reviews, management of a project and enterprise management's involvement at key decision gates. They are defined in terms of a planned progression of progress, resource use, risk and decision-making. It is a feature not readily evident in capability assessment models. It was not expressed maturely in ISO/IEC 15288 and was thus not included in ISO/IEC 15504 Part6. It remains an area for further advancement.

Work products are primary, tangible indicators of process performance. The hierarchy of attributes in Table 6.2 is based on work product management from Level 2 upward; Level 1 being the lacking the evident formalism of their construction. They aid defined class recognition, description of common attributes and comprehension of their properties. They are not a necessity for the objects and information items created during the system life cycle, but they greatly facilitate their recognition, permissible adaptations and general quality. This is true for both for their creators and those assessing the execution and outcome of their creation.

It is helpful to describe work products in terms of a classification structure in which members inherit common characteristics according to their class. Classification schemes, as described in §3, are an ordering technique that allows interested parties to reason about the common attributes of object classes or information classes in a total population. The value of a work product classification scheme directly relates to the effectiveness by which the assessment model creator may define work products and the effectiveness by which the assessment model user may recognise them and interpret what they indicate.

WP ID	Generic Work Product Class	Generic Work Product Description	Generic Work Product Typical Characteristics
1.00	object	An entity created to serve a purpose, or created in the course of serving that purpose. Its existence is observable and rationalised by its material or behavioural characteristics. It may exist as a complete, partial or exemplifying realisation of a product, be a subordinate part of a product, be a by-product or be a part of an enabling system	<ul style="list-style-type: none"> – identity, name of object – purpose, value that caused its creation – ownership and responsibility for object – status, state and classification of object – distinguishing observable qualities and properties – functional and behavioural characteristics – dimensional and parametric characteristics – relationship with and dependencies on surroundings – observable interactions or effects on other objects – interfaces, connections to surroundings – location, position in surroundings – safety, security, privacy and environmental regulations
2.00	description	An account or representation of a proposed or actual object or concept. It may be a textual, pictorial, graphical or mathematical representation. It may be in a standardised form for human or machine interpretation. It may be a static or dynamic model or a simulation representing reality. It	<ul style="list-style-type: none"> – object, subject or class represented – purpose and applicability of description – concerned parties, viewpoints, views – range of use, and validity of description – accuracy, detail and abstraction level – model dimensions, degrees of freedom – description language, notation, nomenclature – applicable standards, formats and styles

		may establish order, structure, grouping or classification.	<ul style="list-style-type: none"> – representations of function, attributes, properties – descriptions of architecture, arrangement, interfaces – depiction of composition or form – definition of classification, category, ranking, type
3.00	plan	A proposed scheme or systematic course of action for achieving a declared purpose. It predicts how to successfully accomplish objectives in terms of specific actions, undertaken at defined times and employing defined resources. It may apply to technical, project or enterprise actions. At a high level of abstraction it may be a policy or, with reference to assets and their disposition, a strategy.	<ul style="list-style-type: none"> – definition of undertaking, purpose and objectives of plan – strategy and policy guiding plan – plan owner, stakeholders, responsible parties and their authorities – plan status, version, reviews and modifications – proposed events, actions and tasks – predicted timescales, durations, dates of actions – assumed dependencies, conditions, constraints, risks – allocated resources, labour, facilities, materials – planned budget, cost, expenditures – defined milestones, results and progress targets – decision points and authorisation gates – options and contingency actions
4.00	procedure	A declared way of formally conducting a customary course of action. It defines an established and approved way or mode of conducting business in an organisation. It may detail permissible or recommended method in order to achieve technical or managerial goals or outcomes.	<ul style="list-style-type: none"> – purpose, outcomes and results of performing actions – issuing authority and controls – roles, responsibilities and duties – actors, their competence and proficiency – dependency on requirements, standards and directives – achievement, goals, completion criteria – definition of transformations and their products – work definitions, instructions to act – progression and dependencies of action – guiding method and practices – enabling tools and infrastructure
5.00	record	A permanent, readable form of data, information or knowledge. Accessible and maintained evidence of the existence or occurrence of facts, events or transactions. It may take the form of a journal chronicle, register or archive. It may contain the information to confirm achievement of performance, fiscal or legal conditions or obligations.	<ul style="list-style-type: none"> – record identity or title – content, description and reason for record – ownership, origin and authorship – practices, agreements, commitments and regulations applying to record – authorities and condition of storage, retrieval, replication and deletion – medium and format of record – location, conditions and periods of storage – applicable information privacy, security and integrity – declaration of status, configuration and baseline information – information on audit, validity and history
6.00	report	An account prepared for interested parties in order to communicate status, results or outcomes. It is a result of information gathering, observation, investigation or assessments, and it may impart situation, affects, progress or achievement. It serves to inform so that decisions or subsequent actions can be taken.	<ul style="list-style-type: none"> – purpose or benefit of report – source, author and authority to report – interested parties, recipients, distribution – knowledge, understanding communicated – information, data, facts and evidence contained – analysis, inspections and audits employed – timing, validity, condition of information use – dependence on circumstances, constraints and assumptions – reported status, results, achievements, conformance, compliance or outcomes – identified faults, failings or errors – inferred patterns, trends or predications – conclusions, recommendations, rationale
7.00	request	A communication that initiates a defined course of action or change in order to fulfil a need. This may originate or control on-going action based on an agreed plan or	<ul style="list-style-type: none"> – objective, purpose or outcome of request – expression of a demand, need or desire – communication of enquiry, solicitation or an order to provide

		procedure. It may result in a proposal or plan of action. It may take the form of a solicitation, requisition, instruction or demand for a resource, product, service or an approval to act.	<ul style="list-style-type: none"> – initiation of supply, provision or support – definition of action, change or exchange – identification of required products, services, capability or resources – authorisation of tasking or commitments – specified terms, conditions to act, agreement conveyed – required availability of requested provision communicated
8.00	specification	Criteria or conditions that place limits or restrictions on actions, attributes or qualities. It establishes measures or qualities for determining acceptability, conformance or merit. It may be required as part of an agreement or contract.	<ul style="list-style-type: none"> – definition of needs, wishes and circumstances – statement of requirements – definition of constraints and conditions – standards and regulations invoked – dimensions of achievement and outcome – criteria of conformance, correctness and compliance – definition of measures, indicators, limitations, values, and thresholds – statements of action and conduct – required functions, performance, behaviour or service levels – definitions of interfaces, interaction, location and connection – conditions of acceptance, permissible exceptions and deviations – conditions of change and variation

Table 6.3 The Work Product Classification of System Life Cycle Processes

This classification scheme took as a starting point the seven work product types of ISO/IEC 15289 [ISO, 2006a]. The generic work product types defined in ISO/IEC 15289 are intended to apply to ISO/IEC 12207 and it was predominantly concerned with information work products. In its later stages of development this was broadened to include ISO/IEC 15288. However, from the ISO/IEC 15288 standpoint the set is too restrictive. Even for software life cycle management, it was immediately extended to include the class Record in the publication of IEEE/EIA 12207.1 [IEEE, 1998a]. This is the refinement introduced in US National Standards in order to create their domestic ANSI Software Life Cycle Process standard of ISO/IEC 12207 in which, unlike its International Standard source, work products were sensibly presented in an Annex to augment the model.

By following the feedback loop at the bottom of work product identification scheme used in this investigation, shown in Figure 6.3, the progressive refinement of Generic Work Product descriptions and typical characteristics that were extracted from the ISO/IEC 15288 model led to the most pronounced yet coherent discrimination occurring with one further extension of the classification scheme. The inclusion of the Generic Work Product type Artefact accommodates man-made entities that have a life cycle purpose and role only when they are manifest in a tangible or corporeal form.

These two additions extend the ISO/IEC 15289 generic work product set to eight classes, arriving at the categories in Table 6.3

6.3.4 Work Product Identification

Since 15288 does not formally name or explicitly define work products, and therefore offer no referential information on these key indicators, it cannot directly act as a process reference model for systems engineering capability assessment. Nevertheless, information on systems engineering process work product identity and characteristics is woven into the text of ISO/IEC 15288, with work products a tacit part of the International Standard's natural language modelling of processes' purposes, outcomes and activities. As described in §4.3.2, it would have been difficult to meaningfully describe the International Standard's transformations to the exclusion of its input and output work products.

A rigorous approach was followed in this investigation to extract this work product information and this is described in Figure 6.3. This investigation did not conduct Actions 3, 4, 5 and 6 in great detail due to the magnitude of the tasks involved. They remain enhancement actions that could improve the consistency and coherence of future reference and assessment models alike.

Actions 5 and 6, and to some extent Action 4, will inevitably introduce changes and rely on judgements that diminish the traceability to the ISO/IEC 15288's text. Conversely, and highly beneficially, such changes stem from the consistency in a fresh and distinct process perspective, and should proactively be employed to modify the existing text that describes/models the transformations. A disciplined audit trail was therefore used to record the decisions taken to make the foundational steps taken in this investigation, thereby provide a secure base from which to continue this refinement.

The steps of decision making were as follows:

Step # 1

All candidate work products that can be inferred from the text of ISO/IEC 15288 were extracted by inspection. This information was traceably drawn to the letter from the text of the International Standard. These fragments of verbatim text were necessarily though minimally extended in order to be meaningful and clear when seen out of context of its broader source text, also to assist understanding when work products do not immediately or obviously relate to specific processes.

Step # 2

Without the benefit of the feedback loops, initially judgement had to be exercised in the identification of all work products and, respectively, all of their attributes and characteristics alluded to in the text. In some cases particular individual characteristics equate to other work products, and ceding this information to bolster other work products generally assisted the levelling of the work product abstraction.

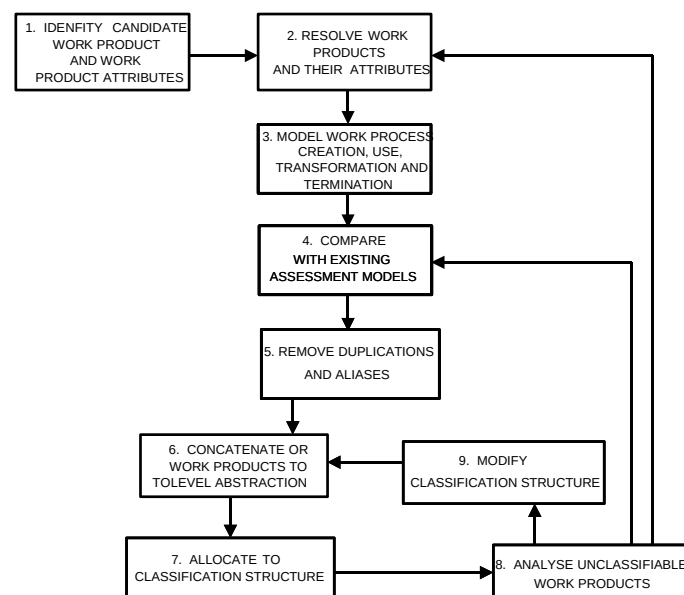


Figure 6.3 The ISO/IEC 15288 Work Product Identification Procedure

Other than to avoid misunderstanding, no text was added beyond that in ISO/IEC 15288. Nor was any removed other than in isolated cases of gross duplication. This led to more work product detail than may be appropriate to complement the abstraction level of the life cycle processes, with this retention of detail leading to pockets of apparently superfluous

information (mostly where international commenting had driven the model to serve localised interests). However, overall the transformation detail and the input/output detail of processes individually, if not with reciprocal equivalence, displayed a generally consistent abstraction across processes.

In total 152 work products were identified to be associated with the 25 system life cycle processes – an average of 6 for a process. This distribution, shown in Figure 2, directs attention in Step # 6. A similar analysis of the level of work product characteristics information would contribute to abstraction levelling.

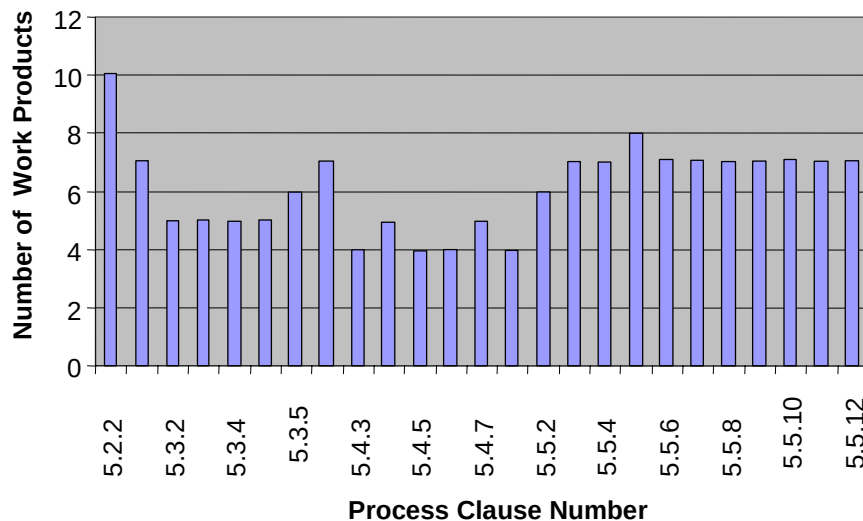


Figure 6.3 Distribution of Work Products Across Life Cycle Process

Step # 3

This step was not conducted due to the magnitude of building and resolving the logical model of processes – a task that bizarrely was not undertaken by the ostensibly disciplined systems engineering community during the \$9.5M development of ISO/IEC 15288.

Step # 4

Again, comparison with other models to identify omissions, errors and distortions was a major step that was only cursorily undertaken. The seemingly obvious reference of the CMMI model highlighted the quality assurance-driven, software-orient approach in CMMI, resulting in a model that appears distorted from a systems engineering viewpoint. An analysis of the CMMI model showed it to contain 403 work products, i.e. over 2.5 times the detail with some of this descending into obscurity.

Of more value was ISO/IEC 15289 with 94 essentially information work product, although the classification of these led to work product distribution by class that had three times the variation seen in the final ISO/IEC 15288 classification shown in Figure 6.4.

Step # 5

Removal of duplications and aliases was undertaken to correct lax terminology and comment-driven biases in the system life cycle reference model text.

Step # 6

An interventionist levelling of work product abstraction was not taken to avoid deviation from the reference model. This action has most value in feeding back proposed changes to the reference model.

Step # 7

The allocation of work products to the classification structure is for the most part straightforward. However in cases where the distinction was marginal, judgement based on

the context or intent of the activities in which a work product is implied was employed. Style variations and constructive ambiguity in the descriptions that acceptably accommodated many National Body comments (a total of 3470 on the Committee Drafts alone) contributed to this uncertainty.

Step # 8

Where Step # 7 failed to convincingly classify an individual work product uniquely or convincingly to one class, the need for an additional classes or a redefinition of the existing classes was considered.

Step # 9

Notwithstanding the classification uncertainty seen in Step # 8, the variation across the work product classes is around 3:1. This indicates that the introduction of new classes, or the redefinition of class boundaries, could further benefit work product definition and identification, and thereby aid assessment model developers and assessors. Possible additional classes were not evident in other classification schemes and thus not introduced in this study. The logic for balancing and sharpening the classification scheme would benefit from an objective modelling of processes, as observed to in Step # 3.



Figure 6.4 Work Product Distribution by Class

Work products, like the process transformation that use and create them, exhibit varying degrees of abstraction or granularity. Whilst the processes of ISO/IEC 15288 received much attention during their development in regard to a constant abstraction level, no explicit attention was given to the structuring, coherence, self-consistency and mutual consistency across the activities. In consequence, the implicit definition of process work products that arises from activity descriptions is a source of variation in work product detail and importance.

An overriding message from this analysis, at least for ISO/IEC 15288 (though it would seem relevant to any organisational process), is that, given the intrinsic duality of transformation and input/output product descriptions, the quality of life cycle process definitions depends on the reconciliation of these two process viewpoints. Each representation of process should be simultaneously and progressively developed, with each view mapped to the other and the inconsistencies iteratively resolved. This suggests that both descriptions should be concomitant in a one International Standard, that defines a single reference model and that defines the same processes from different points of reference. Stated more succinctly, ISO/IEC 15288 would benefit from the inclusion of work products in order to comprehensively and usefully describe its life cycle processes.

Appendix A presents all instances of expected work products from each life cycle processes that an organisation may have implemented. These are described in terms of a work product identifier number, name and the characteristics that provide guidance for assessors.

Many information items evolve as a system evolves through its life cycle. They are thus functions of when they are viewed and are typically a function of planned progression, resources profiles, decision-making and risk management, i.e., for IPTs they are CADMID stage related. These cardinal partially complete states may also need to be defined, not least because of their importance to enterprise decisions at stage gates.

6.4 Systems Engineering Competence

6.4.1 From Business Process to Professional Competence

To complement the preceding treatment of capability, this section begins a consideration of systems engineering competence by examining the corresponding transformation from business process, shown schematically in Figure 6.5 (again a part of the business excellence concept in Figure 2.3).

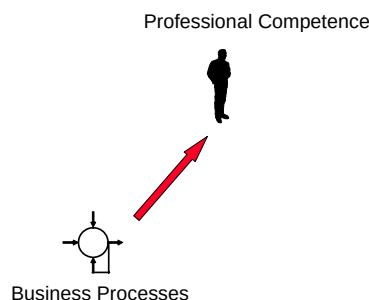


Figure 6.5 Deriving Individual Systems Engineering Competence from Business Process

Chapter 2 considered the nature of a discipline and saw it to be defined [Collins, 1991] as:

- a branch of learning or instruction;
- a system of rules for behaviour, methods of practice, etc.;
- systematic training in obedience to regulations and authority.

However, this is a somewhat 'product-oriented' description of discipline. Recasting it in 'service-oriented' terms leads to a more outward looking definition; namely, the actions and behaviours concerned with a branch of learning and practice that is conducted to the benefit of the community by trained practitioners according to a regulated or authorised model.

Systems engineering competence is in the round concerned with practitioners, their development and how they make skilled engineering contributions to meet society's demanding needs for quality system products and services. ISO's regulated and authorised set of system life cycle processes is the natural candidate for the 'system of ... methods of practice'. Accordingly, it is employed here as the basis for developing a systems engineering competency framework. The value of this approach lies in its adoption of a widely accepted model of processes, whose scope equates to systems engineering's delivery of service into business, and thereby society.

Ramo asserts that 'professional, qualified competence in systems [engineering is]... partly mathematical, partly experience, partly common sense, partly detailed specialised knowledge; partly the setting up of the right kinds of scientific experiments. It is not easy, and [repeating his words in this Prologue] not for amateurs'. From this, he argues that '[what] stands in the way of the fullest application of the systems approach is that we lack enough trained professionals' [Ramo, 1998, p.67]. This is an important and still wholly valid message, for it identifies a prevailing barrier to the more widespread adoption of systems engineering. It also affirms the value of trained professional competence development, downplaying the often-voiced notion that it is an essentially empirical discipline that comes only from a significant level of experience.

With justification, there are particular challenges when defining systems engineering competence. Systems engineering is concerned with a comparatively immature and only recently codified and recognised element of strategic business practice. It meshes with, and must thus harmonise with elements of much longer-established management and technology discipline. It sits, as seen in Figure 2.10, at a key organisational interface between engineering and management; an interface that has long harboured the potential for cultural dislocation and mutual oversights

To compound matters, systems engineering teaching has evolved in different academic centres from different combinations of related subjects/departments, each bringing different bias. Different seats of learning have focussed on different application sectors, again bringing different emphasis to the principles employed.

In many First Degrees there has been a reluctance to introduce a common element of systems engineering into the curriculum due to a perceived competition with traditional course material whose primacy eclipses systems engineering. Commonly systems engineering has been academically taught as a post graduate subject, frequently for students who have benefited from a period of industrial or other practical experience. Debates about how many years of career experience are needed before studying a second degree on systems engineering have churned for years, and whilst a foundation in one implementation technology-based engineering subject is hugely valuable, it is not on evidence essential. In the United States there has been much interest in degrees of systems engineering combined with a specific engineering application or technology domain. This mixture is able to emphasise the application relevance that releases the full power of systems engineering.

Beyond academia and into industry, the application domains and technologies diverge and bring further diversity. A single, coherent body of systems engineering becomes harder to isolate, and its day-to-day relevance and use harder to recognise or interpret – “tell me the practice to follow, I haven’t time to understand the principles” is a call from many of its users that fails to recognise the consequences of systems engineering complexity and the adaptive nature of its application. Despite, or perhaps because of, the challenges and a poor track record to date, a sound, rational and widely endorsed codification of systems engineering competence remains a valuable goal. A commonly respected competence model offers many benefits and as Abbott advances [§2.4.2] it would be a cardinal step in the jurisdictional standing of the discipline.

In various guises, formalisms of systems engineering competence have been employed by practitioners, businesses and academia to:

- encourage proficiency and professionalism;
- establish performance criteria, permitting demonstration of competence;
- imply codes of professional conduct and approach;
- accreditation bodies;
- prescribe social responsibilities, e.g. safety, environment;
- profile training and education schemes;
- lead toward norms of attainment and qualification;
- career development decisions.

With this range of benefits, there are many stakeholders and many viewpoints.

Four important and widely used viewpoints stand out from all others, and they are conveyed in Figure 6.6. The notion in this figure is of a common body of competence information that can be described according to four different frames of reference. Each one these four viewpoints has been described separately many times with differing rigour and quality – by academics, institutions and business organisations.

Arguably it has been the past lack of a truly well-accepted ‘system of rules for behaviour, methods of practice, etc.’ for the competence information at the centre of Figure 6.6, and the multiplicity of stakeholder concerns and possible viewpoints of it, that have led to limited agreement on how to describe systems engineering competence. Therefore, by way of

completing the triptych started in §4, this and the following section offer a route that could serve the concerns of individual practitioners, academia, project teams and business organisations alike.

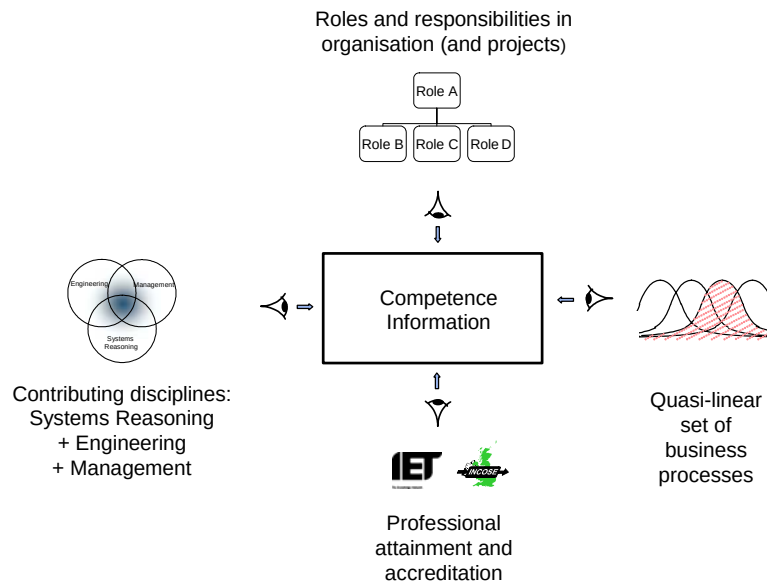


Figure 6.6 Viewpoints of Competence

The approach advanced in this investigation is thus of a common body of competence information filtered into the four distinct perspectives shown in Figure 6.6. That is, of a coherent system of processes performed and managed by practitioners, expressed according to the viewpoints that satisfy systems engineering's major stakeholder classes.

These viewpoints give rise to (clockwise in Figure 6.6):

- the business process view as presented in ISO/IEC 15288 and Figure 2.10;
- the professional standing view, using UK-SPEC to provide the yardstick for all engineering competence;
- the three primary discipline constituents of systems engineering conveyed in Figure 2.2;
- the roles and responsibilities in organisations view, and particularly that seen by the typical project team

Competence in many branches of human pursuit is seen as the relevant qualities that enable individuals (and teams) to beneficially employ a set of processes to change events and outcomes; for example, the creation of an aircraft and/or the delivery of transportation services as in Figure 3.5. Expressing the competence model proposed is, at its most abstract, shown in Figure 6.7. It advances that systems engineering competence is the natural and developed ability of humans to perform system life cycle processes in order to act on systems, and thereby to achieve beneficial change/interventions.

The crux of this approach to representing competence is the formulation of a model that defines the major categories of human ability required to conduct of system life cycle processes and their expression in terms of attained level of performance. These two features – components of ability and attainment level – are treated in this order.

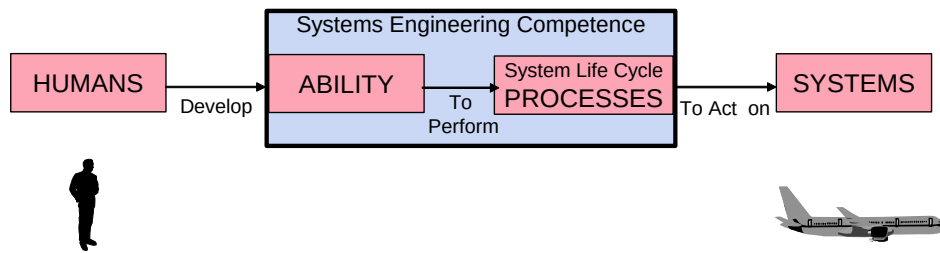


Figure 6.7 Humans Act on Systems Using Systems Engineering Competence

The characteristics of competence are complex for they are:

- a function of:
 - the level of individual attainment,
 - the characteristics of a range of system life cycle processes;
- overall, a non-linear, multi-featured set;
- difficult to objectively quantify as a single measurable value;
- based on values that are heavily dependent on national tradition, norms and accreditation legacy.

Being a cross-cutting field of engineering and management, systems engineering and its codification dip into the traditions of others. Indeed for many systems engineering is no longer seen to be solely the preserve of those who are styled systems engineers. Systems engineering principles and practices now impact and influence the actions of a majority of technical, managerial and executive staff in organisations. Thus, whether roles require high levels of systems engineering professional attainment or, more generally, a lesser awareness that complements existing disciplines [Figure 2.10], a solid framework of system technical competences is an across-the-board necessity. The wide range of stakeholders presents a challenging spectrum of needs for a competence framework.

Any solution needs to comply with requirements of recognition, business utility and practical acceptability and these derive from the practitioners, academia, the professional bodies and Institutions, and employers. It is not surprising that systems engineering competence, almost archetypically, is a battleground of diverse (often zealous) opinion. Add to this organisational tradition and, even more ingrained, the national *modus operandi* built on separate cultural traditions, and the codification of systems engineering competence for widespread use understandably remains elusive.

Systems engineering capability is thus an overdue maturer in a quarter century of quality-based improvement actions. During this time the quality improvement ‘plan, do, check, act’ cycle has remained both slogan and model. So it is that the ability to plan using a competence reference model, to check the conduct (the do) against this, and to act in accordance with a development path through such a framework has governed the development of competence attributes and attainment scales.

6.4.2 Competence Attributes

As has been the case for capability, codifications of competence are in vogue across organisations and professional bodies. The twin drivers of maturity attainment and assessment seen in organisational capability advancement have likewise been driving the development and appraisal of competence.

Also, just as with capability, competence is expressed most comprehensibly in terms of an ordinate attainment level and an abscissa relating to a codified range of actions or processes that are seen to effect successful systems engineering outcomes. Further, and again as with

capability, due to a dependence on several parameters with non-linearity relationships, both of these axes are difficult to express linearly, and each is best represented as a discontinuous or banded scale.

Competence in systems engineering (and many other pursuits) is a function of several factors or variables and this has opened the door to a diversity of models. This research has focussed on four strongly influential components with substantially, though not wholly, orthogonal relationships. They are aptitude, knowledge, experience and skill, and the way these relate to ability, and hence to competence, is put forward in Figure 6.8.

This figure illustrates how systems engineering ability of an individual (or for that matter a team) relates to:

- aptitude: the natural or personal qualities of the individual (or the net qualities of a team of individuals) that are relevance to systems engineering conduct, such as the faculty to effectively employ systems reasoning, the interpersonal and team sensitivities, and the approaches to enquiry;
- knowledge: the information gathered from others, often in the form of academic teaching, formal tutelage or the assimilation of a recognised body of knowledge that is concerned with schooling in engineering and management;
- experience: the direct personal participation in the practicality of systems engineering conduct through which the individual can benefit from the judgement, wisdom and mentored of others in situ;
- skills: the trained practical execution and craftsmanship using recognised methods, techniques and tools.

Maximising attribute orthogonality is a compromise and these four variables, in common with other competence models, do not contribute to the degrees of freedom with complete independence. For example, knowledge is a function of experience that provides contexts, relevance and insights; also knowledge is dependent on aptitude, in that memory and information interpretation are dependent on aptitude. Despite this, these four attributes are to be seen individually or in partial combinations in competence representations. In this combination they define a common denominator set of attributes that for the purposes of systems engineering is substantially orthogonal.

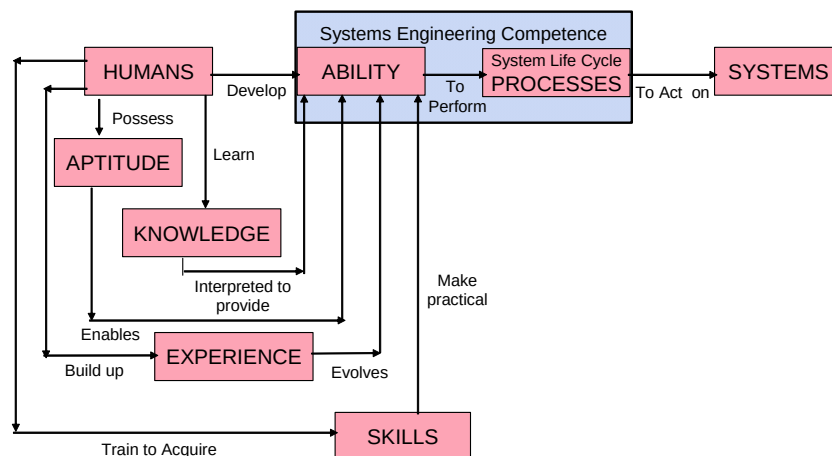


Figure 6.8 Components of Systems Engineering Competence

Aptitude is the quality that humans possess that acts as the prime enabler of competence. For systems engineering it is strongly contingent on the natural propensity of an individual to conduct systems reasoning effectively. It is through the existence of systems reasoning's

cognitive mechanisms and self-discipline that the individual effectively delivers the other components of competence.

This requires a concerted application of the armoury of cognitive primitives; for example, abstraction handling, pattern awareness, structured exploration, contention resolution and many more tactics of cognition that together empower the systems reasoner. §3 advanced that it is innate, evolutionarily-developed cognitive mechanisms that govern an individual's systems reasoning. Drawing on personality traits that favour or otherwise its effective application, it is that part of a person's make-up that predisposes them to be a natural 'systems engineer' – to be made of the 'right stuff'.

Nevertheless, the phylogeny that configuration controlled the fabrication of these specialised cognitive modules and contributes to each individual's system psychology are not an end point. For an individual they are a highly influential, but mutable beginning. Ontogeny then takes over, and it is conditioning and development that places the final cutting edge on cognition.

Individuals are born with an intrinsic aptitude for systems reasoning – and also for that matter with the inquiring and problem-solving instincts of engineering, and the competitive and organising instincts of management. Figure 2.2 does not start with a blank sheet.

Nonetheless, systems reasoning can and should be explicitly developed. True that through genetic good fortune good systems reasoners may be born, but a managed path of exposure to situations and challenges designed to develop the 'right stuff' can make any individual better, especially in the way they contribute to team systems engineering situations. This was recognised early on. Hughes recounts [Hughes, 1998] that, following the ICBM programme, it could be 'argued that academically, or scientifically, trained persons had especial expertise for implementing the systems approach'.

The way in which some of this 'especial expertise' in systems reasoning is developed and conditioned was empirically determined in the '60s, though the difficult-to-calibrate results obtained are not without contention. Nonetheless, real-world modelling according to linguistic and symbolic conventions, the spatial rules of geometry, the conferring or imputing of intentionality, the logic of decision making and conflict resolution, the skills of pattern recognition, the treatment of abstraction handling, and much else, are capable of being developed individually and, even better, as a coherent package during a lifetime.

Knowledge gathering and retention, experience to map to different contexts, and skills that leverage methods and tools *do* appear to feed on and into the cognitive raw material of systems reasoning. Natural aptitude *is* open to advancement and redirection during an individual's career.

Categorising 'the right stuff' for systems thinkers helps to structure and manage this advancement. It is a complex set and it includes:

- holistic thinking;
- model visualisation;
- abstraction handling;
- information hiding and elision;
- structured reasoning and decision making;
- risk recognition;
- comparative analysis;
- speculative, conjectural, imaginative and adventurous enquiry;
- ambiguity accommodation and resolution;
- multi-scale visualisation;
- ability to create and manipulate models;
- dealing with uncertainty, indefiniteness, indeterminateness;
- "thinking outside the box", i.e. beyond conventional or comfortable boundaries;

- resolution of compound relationships, influences and dependencies;
- empathy with different viewpoints, resolving and reconciling the differences;
- pattern matching.

Knowledge is the second component in Figure 6.8. It is the accumulated body of theory, facts, principles, rules, standards, scholarship and awareness that has been distilled from researchers, practitioners and end-user beneficiaries. It depends on information management comprising:

- gathering;
- structuring;
- retention;
- access;
- update;
- application.

For the individual, knowledge is the assembled body of information contributed by the academic and practitioner community, together with the knowledge management techniques that mobilise this. It links strongly to formal education and training, often with a theory bias. So called Books of Knowledge bring together that which is seen as essential to individual disciplines; systems engineering has seen attempts (with mixed results) to provide this information [INCOSE, 2003].

Experience, the third component of the model, has been a traditional route to systems engineering competence. Ideally, it is not something that should be acquired by happenstance but should be a crafted regime of context exposures.

The systems engineering misquote “jack of all trades and master of one” is an apposite observation on how the best of systems engineers often develop. Building first on the mastery of a specific implementation technology-dependent engineering discipline, an individual can establish the relationship between reality, the effect of interventions in it, and how a specific technology can be employed to effect those interventions. An understanding of the fundamental correspondence between form, function and effect is thus forged in the mind according to its innate mechanisms, the laws of physics and one, thoroughly familiar implementation medium. The rules and strategies for manipulating the design triad in Figure 4.9 are then firmly established ready to be mapped to fresh, less familiar situations and implementation media. Mastering this sublime task without the benefit of a training ground of at least one implementation technology is a demanding mission.

Experience is accumulated from personal participation in the day-to-day conduct of systems engineering. It benefits from exposure to different life cycles and to systems having varying levels of complexity and size, and subject to different levels of engineering formality. The personal navigation through a string of career moves that are matched to competence gaps, that involve ‘challenging’ situations and responsibilities, and that can benefit from observation of and mentoring by those already experienced, all go to building an explanatory and interpretive framework that eventually becomes the first line of attack on complex problems.

To complete the model in Figure 6.8, practitioners train to acquire skills that then bring practicality to the other capability components. Skill and engineering are traditional partners. As seen, the term engineer comes to us ultimately from *ingenium*, Lat., meaning skill or talent. In broad terms it is the effectiveness and reliability of process execution or service delivery achieved by an individual. It is developed through repeated refinement of execution, feedback and correction.

In the case of engineering it depends heavily on methods and tools. It is a practitioner’s ability to optimally execute practical methods and procedures, to proficiently employ techniques and to leverage the power of tools. It commonly equates to the peaks of craftsmanship and dedication that distinguishing a level of quality that is prized, promoted and often professionally protected. It may be dependent as much on the individual’s selection and

employment of particular methods and tools as on the subsequent execution, and thus requires situation judgement and empathy with the particular problem and solution opportunities.

The UK Engineering Council exercises authority for a wide range of branches of engineering, and to each of these applies the Engineering Council's UK Standards for Professional Engineering Competence (UK-SPEC). In it, the Council defines competence to 'include the knowledge, understanding and skills which underpin performance' [UK-SPEC, 2004]. This competence is predominantly attained by virtue of an individual's education, training and experience. These are effectively delivered according to prescribed behaviours, and can be demonstrated by indicators that confirm threshold criteria.

UK-SPEC provides criteria for education (to indicate knowledge and understanding), for training (to indicate skilled application of knowledge and understanding) and for experience (bringing practitioner judgement to the whole). Two of its four requirements areas are predominantly concerned with the application of technical processes, the third with project and enterprise management processes and fourth with interpersonal skills (behaviour both as an individual and in teams).

In general terms, the UK-SPEC aligns with this model, with education substantially equating to knowledge, training substantially equating to skills, understanding substantially equating to experience and interpersonal skills substantially equating to aptitude.

This competence model does not detail behavioural attributes that favour the conduct of systems engineering and many other creative, social and competitive undertakings. Behavioural attributes of this type are recited in many models and are not repeated here.

6.4.3 The Attainment Scale

As Proust is quoted as saying "we don't receive wisdom; we must discover it for ourselves after a journey that no one can take for us or spare us". This is a précis of the nature of each and every journey up the scales of attainment that constitute the ordinate of most competence frameworks.

Such a scale delineates directly:

- the quality function being conveyed;
- its cardinal graduation of professional influence, achievement and merit graduations;
- descriptions of competence bands that convey progression in a relevant way to a community of stakeholders.

For well over a decade, the British Computer Society's has refined the levels and titles that metricate organisations' IT roles. It has defined steps of personal competence using a scale that covers 9 levels of organisational authority/job value and these have been mapped to 7 so-called skill levels in SFIA, its widely regarded IT skill model.

Nevertheless, for many practical purposes, 5 levels appear to have considerable merit. 5 levels are compact and yet providing a richness of scale, and appear to be cognitively comfortable. This may arise from some deep seated cognitive agency that supports the mechanisms of categorisation and leads to aphorisms such as "7, + or – 2" being used to guide the practicalities of structured decomposition. Certainly, a 5-step structuring aligns with the quality thinking that has resulted in practical capability maturity models.

Condensed to 5 levels, SFIA has been selected in this investigation to structure an exemplar framework. It maps, using SFIA terminology, to the characteristics in Table 6.4. The small population at SFIA Level 7 and the low relevance at Level 1 to mainstream systems engineering make this 5-band scale a legitimate level compression of SFIA for the purposes of systems engineering.

SFIA Level 2	Assist
SFIA Level 3	Apply
SFIA Level 4	Enable
SFIA Level 5	Ensure, Advise
SFIA Level 6/7	Set Strategy, Initiate, Influence

Table 6.4 SFIA Skill Levels

Approaching matters from a different line of reasoning, the research of the Dreyfus's postulated what has now become a widely-accepted model of how individuals progress through various steps in their acquisition of skill [Dreyfus, 1986]. Their model defines a five-stage progression which they labelled: novice; advanced beginner; competent; proficient; expert. Artificial intelligence (sic), i.e. established or customary routes of reasoning, and rote content, i.e. facts, are considered in the representation to be useful in the first three stages, to barely assist in progression to the fourth, but are not contributors in advancement to the fifth stage. The characterisation and rationale for the Dreyfus's progression is summaries on the right of Table 6.4.

Novice	Facts + Rules
Advanced Beginner	Fact + Rules + situation
Competent	Facts + Rules + selected contexts + accountable
Proficient	Accountable + intuitive, immediately sees what
Expert	Immediately sees how

Table 6.4 The Dreyfus Model of Learning

Whereas the titles to the left of Figure 6.4 should be seen as signifying domain-related lexical preferences; what is of utmost important is the bands of characteristics on the right. Titles, particularly when bound up with professional or organisational status, are emotive and the subject of much irresolvable wrangling.

To avoid the typecasting of any particular model and chosen number of levels, this investigation lexically conveys competence according to five levels that are designated by titles that attempt to convey the nature and team/social recognition of competence. In ascending level of attainment they are:

basic understanding;
 working knowledge;
 team leadership;
 professional direction;
 recognised authority.

These are shown in the ordinate of Figure 6.9, which conveys how this model relates bands of attainment to the linear ranking of ISO/IEC 15288's system life cycle process that forms the abscissa. Every cell formed by the conjunction of intervals on each these primary dimensions of the competence equates to descriptions of aptitude, knowledge, experience and skill. These then spawns further dimensions formed from yet more detailed modelling of each component.

Competence is not, however, a point or cell in the framework formed. It is a quality that spreads across the multi-dimensions of competence space. For the individual, this detail is important. However, it builds a complex picture and, as with capability, two simplifications are used to summarise it: a single, all embracing value (for capability a stage) and a profile (for capability a continuous maturity contour) as a function of the abscissa.

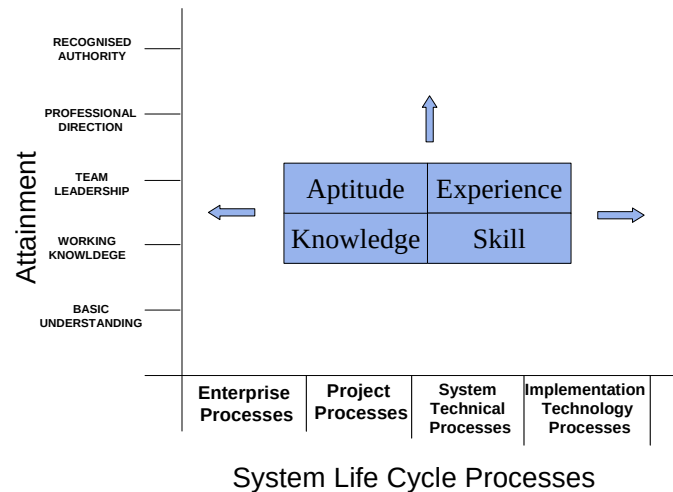


Figure 6.9 Building a Matrix of Competence Attributes

Figure 6.9 readily relates to attainment as a dependent function – a profile – of values according to the independent abscissa of system life cycle processes. The illustrative symmetric distribution in Figure 6.10 advances the notion of an individual having a focus of attainment, with a range of supporting contributions from contiguous regions of business process. The representation in this figure borrows from the idealised systems engineering profile in Figure 2.10 and illustrates how its focus may shift over time for one individual, or at the same time be different for different individuals who may be contributing to the overall competence of a single team. It also conveys how any profile may move up the attainment scale.

Change with time is described by a third axis; one that for the individual would be most qualitatively represented as career path. This for some would have a general leftward and upward trend, for some would become a flatter profile that may suit team leadership, and for others would lead to the sharpening of specialisation seen in the expert.

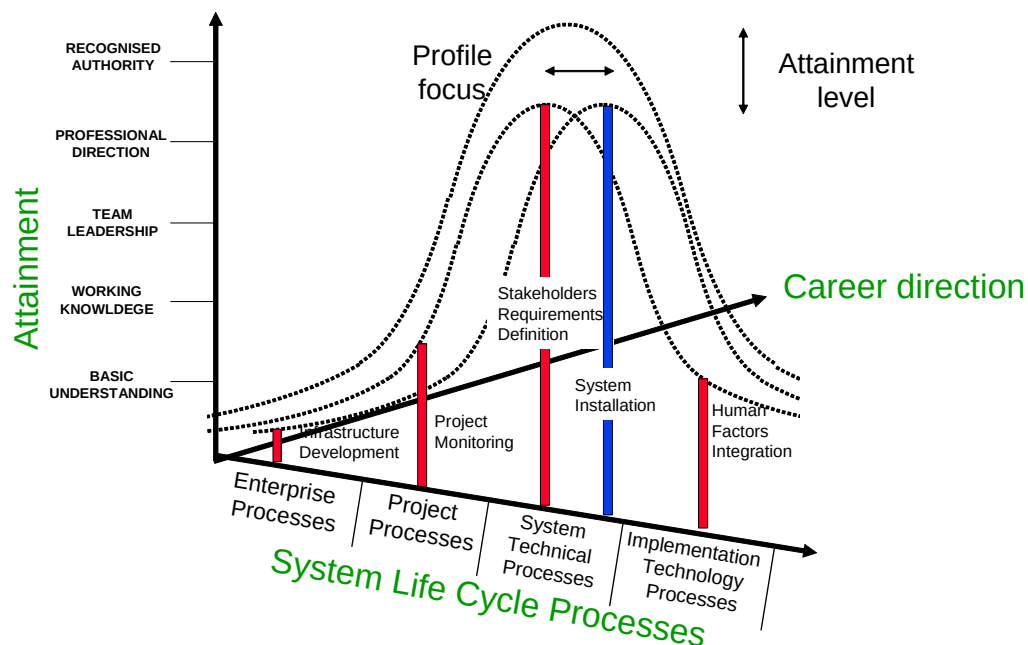


Figure 6.10 Competence Profile and Career Progression

Whatever the shape of the attainment profile, the correspondence of levels in Table 6.4 with the five maturity stages of capability detailed in Table 6.1 and Table 6.2 is marked: first, three stages of institutionalisation ascend toward progressively more established and dependable outcomes up to Level 3, capped by two situation sensitive stages of regulation that depend on contextual recognition and response. The nature of the quality/value function in both cases is similar.

The pragmatic human factors that make up competence have been the subject of much analysis. Various educational taxonomies have been developed in recognition of the observations that an individual's learning occurs in distinct dimensions or domains of human performance, e.g., cognitive, affective, psychomotor. Commonly learning taxonomies have been employed to explore and structure cognitive learning. Although Bloom [Bloom, 1956] and a team of nationally assembled US colleagues developed the most well-known taxonomy, other workers have proposed alternatives that organise cognitive knowledge in different ways – the complexity in the dimensions of competence is open to multiple, credible viewpoints.

The value of learning taxonomies lies in their structuring of educational development objectives, and how they related to achievement and assessment. These taxonomies typically use particular verbs to specify desired learner behaviours, and give rise to designators that (ostensibly) identify and distinguish levels. However, the varied interpretation and recognition due to cultures, academic bodies and professional institutions inevitably leads to considerable imprecision.

Professional competence assessment, such as that undertaken by registration authorities, depends on commonly interpreted codification. This is particularly important when responsibility for this is delegated to subordinate authorities who specialise in branches of learning, practice and traded services. Strong structure and definitions to counter non-repeatability and deviations of interpretation count for much.

In general, advancement up a scale of professionalism requires a practitioner to hone their abilities through repeated and diverse self-application that develops experience-based understanding of what it means to exercise care and attention. But in advance of this an understanding of abstract concepts is necessary, and this fundamentally derives from formal education, teaching books and a clear structure of subject classification.

The learner's movement towards the expert stage is conventionally described through developmental stages in which the most basic – novice, advanced beginner and competent practitioner – are characterised by the use of substantially context-free rules. These are abstract generalisations that in the higher stages of experience and skill are open to understanding in new situations or subject areas. Competent practitioners, thus, take abstract rules as their starting point and develop to reach independent decisions on which strategy to optimally pursue in a given situation.

At these proficient and expert levels of the skills model, developed expertise permits deep situational understanding. Analytical principles are relegated and tacit knowledge is brought to the fore, with knowledge translated into refined, purposeful action in each given context. From experience, the expert is in a position to act intuitively and with focus when confronted with a fresh problem. In this way, behaviours of professional judgement shift from rule-based action, independent of context, to context dependent, reasoned conduct.

Bloom's taxonomy for cognitive skills was originally developed by a US committee of educational psychologists to categorise steps of corporate training and development. It is summarised in Figure 6.5 and is presented as degrees of cognitive attainment. It starts from the simplest cognitive skills and ascends to the most complex, each depending on subordinate contributions: in effect a hierarchical classification of competence outcomes. It remains probably the most widely used taxonomy today, having stood the test of time since its definition in 1956.

Evaluation

Make judgments about the value of ideas or materials.

Synthesis	Builds a structure or pattern from diverse elements. Put parts together to form a whole, with emphasis on creating a new meaning or structure.
Analysis	Separates material or concepts into component parts so that its organisational structure may be understood. Distinguishes between facts and inferences.
Application	Use a concept in a new situation or unprompted use of an abstraction. Applies what was learned in the classroom into novel situations in the work place.
Comprehension	Understand the meaning, translation, interpolation, and interpretation of instructions and problems. State a problem in one's own words.
Knowledge	Recall data or information.

Table 6.5 Blooms Taxonomy for Cognitive Skills

Bloom's taxonomy provides pointers to at least some of the cognitive mechanisms at work in competence development without attempting to directly equate these to any professionally recognised or business role banding of individual or team attainment. The taxonomy provides a clear indication that knowledge (the foundation on which Table 6.5 is built) is unsurprisingly of primary importance. It strongly supports the argument that it is context-based experience that aids the interpretation of novel situation. It also implies that systems reasoning aptitude plays a key role in the analytical separation of conceptual and material objects (reinforcing the systems psychology arguments of §3.5.3), and the pattern-enabled synthesis of structure.

This mapping of knowledge, experience and aptitude to stages of attainment, whilst complex, is consistent with the model in Figure 6.8. Skill, however, is not brought out with any distinction by the Dreyfuses, Bloom et al. or most learning/cognition development models, certainly not as defined in this investigation's model, i.e. as specialised proficiency in the use of techniques, methods and tools. Notwithstanding, it is advanced as a facet of competence that intimately meshes personal accomplishment with an organisation's provision of intellectual and material assets. This linkage is illustrated in Figure 6.11, where the relationship between professional competence and organisational competence (rather more simply depicted in Figure 2.3) is made.

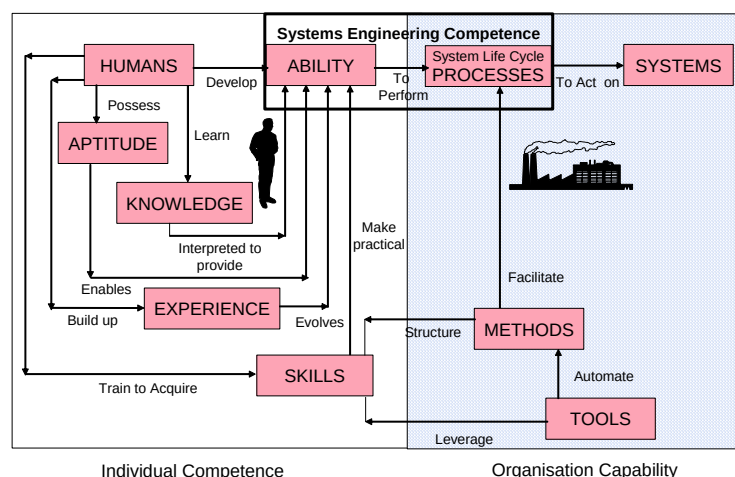


Figure 6.11 Interaction Between Individual Professional Competence and Organisational Capability

6.4.4 Professional and Organisational Contexts

It is the perspectives of the professional bodies, whose role is to govern the development of systems engineering practitioners, and of the business organisations, who employ these practitioners, that conclude this consideration of competence.

The professional institutions are the agents of regulation and authority that place discipline in the realm of professionalism. They conduct endorsed self-regulation of their discipline, often as a result of conferred authority from the state (crown or government in the UK). For engineering in the UK crown authority is vested in the UK Engineering Council and, by conferment, on its accredited member engineering and technology bodies. Two such bodies are the Institution of Engineering and Technology, IET, concerned with engineering and technology generally, and the British Computer Society, BCS, having an information technology focus. These and other affiliated bodies are heavily concerned with establishing and maintaining engineering professionalism, and with monitoring its propriety and effectiveness.

At the heart of the Engineering Council's *UK Standard for Professional Engineering Competence* [UK-SPEC, 2004], is a structured, minimal but sufficient definition of engineering professionalism. This is expressed in general terms that define threshold standards of competence and commitment. The expectation is that this generality is then translated by individual professional bodies accredited by the Engineering Council into the language and context of their branches of engineering. It is therefore paramount in any UK competence framework that traceable compliance with the Engineering Council's requirements for professionalism is demonstrated. For this reason the professional attainment and accreditation viewpoint of Figure 6.6 is essential.

Whilst systems engineering is in principle the preserve of many Engineering Council accredited bodies, there has been limited registration of candidates presenting themselves explicitly in a systems engineering category. The number is reportedly on the increase.

Where a well-ordered and established tradition does not apply, such as across boundaries of nations, then it is through good standing, strategic associations and a record of valued outcome that regulatory authority becomes recognised. This is a path along which INCOSE is moving.

Despite being the 'home' of systems engineering, professional registration in the USA is subject to serious constraints. Professional standing is regulated at the individual State level and this has handicapped a unified approach. It has inhibited individuals locally styling themselves as a 'Systems Engineer' in many States. INCOSE's actions are assisting this situation by developing a certification styling for systems engineering. It is based on a set of criteria, partly attained by examination and partly by experience and commendation [INCOSE, 2007]. The certification scheme may have limited relevance in countries and cultures outside the USA, however, its examination could be a valuable contribution in some, if not all, professional development schemes. It is based on the INCOSE Systems Engineering Handbook [INCOSE, 2007b], and this in turn had been based on the ISO systems engineering International Standard.

More specifically, the UK Chapter of INCOSE advisory body comprised of representatives from UK organisations have developed a guide to a set of systems engineering core competencies to fill the perceived gap [INCOSE, 2005]. Limited consultation and weak traceability to explicit stakeholder needs, or to national/international models of systems engineering and professionalism, mean that endorsement has been limited. However, it is the product of informed and relevant contributions, and is a valuable contribution to the developing competence scene.

An underlying drive of competence model development is towards standards of attainment and conduct that can meet the needs of business. Competence definitions that directly relate to business, and to the roles and responsibilities in 'systems organisations', are considered essential. Any framework that cannot serve individual professionalism and organisational business alike is unlikely to gain recognition.

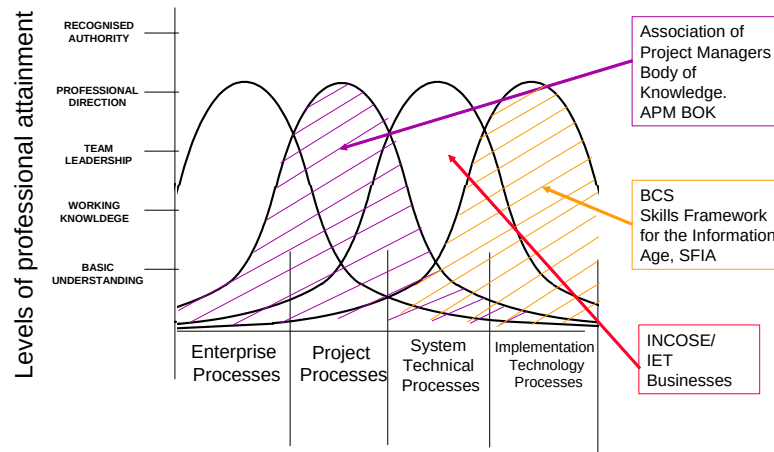


Figure 6.12 The Relationship Between Systems Engineering Competence and Related Frameworks that Have Been Published

Following the pan-organisational business process model of Figure 2.10 and with reference to competence models for related disciplines, it can be seen in Figure 6.12 that there should be valuable precedents and indicators of direction that systems engineering could align with or at least draw from.

This investigation assessed the approaches taken by project management professional to be useful but hardly definitive, whereas that taken by the BCS is well-design and well-proved, and strongly relevant to systems engineering. SFIA was constructed to describe the typical roles and responsibilities, together with the supporting competence attributes associated with them, that are found in organisations employing and/or creating information-based systems. It has its origins in the British Computer Society's Industry Standard Model (ISM), and this has been used and validated since 1996 for the purposes of software engineering professional development. SFIA has a provenance, intellectual content, practicality and standing that merits attention by the systems engineering community.

From Figure 6.6, an ideal approach to systems engineering competence would have traceability to the International Standard (as the internationally ratified model of systems engineering), to UK-SPEC (as the nationally regulated model of engineering professionalism) and to SFIA (as the pre-eminent UK model of relevant organisational roles and skills). In light of this, this research has mapped SFIA to ISO/IEC 15288 to examine the value of harmonising a process-based and a role-based approach to defining a competence framework. In essence the body of competence information at the centre of Figure 6.6 is built from these two viewpoints.

The text of SFIA has been transposed (about 90% mechanistically and 10% judgement) into the life cycle process structure of ISO/IEC 15288. The detail of the outcome is approximately an 85% word-for-word mapping, with changes made to remove the technology bias of IT and with the ordinate reduced to a 5-level band structure. This transposition is presented in Appendix B. The emphases of the mapping morphology between SFIA skills and ISO/IEC 15288 processes can be seen to the top of this tabular representation. Regions of compression, such as resource management, implementation and maintenance, reflect the service management predominance in SFIA. Regions of weak or no clear mapping, such as the management of life cycle stages, demonstrate the better through-life product/service balance in the ISO International Standard.

Because SFIA traces to a 9-level organisational role model, some roles as not considered to exist in it at some levels, i.e. in either upper or lower reaches of the table. These null-entries are retained in this table to help illustrate the mapping, although appropriate higher and lower level cells would require completion to satisfactorily present the full picture of the shaded systems engineering curve in Figure 2.2.

SFIA 2.0 thus provided a fast track opportunity to prototype and explore a partial competence framework implementation. Its replacement model, SFIA 3.0 introduced later in this research has no significant influence on the result. The UK-SPEC on inspection appears to be fully

capable of an equivalent mapping but would require a mapping of next-level SFIA detail and this lies beyond the prototype concept demonstration undertaken. This is corroborated in Table B-2 in which the correspondence between SFIA responsibility and accountability detail is equivalenced to a verbatim transposition of DERA/QinetiQ's staff development scheme that is UK-SPEC accredited.

Overall, the outcome of this model-building exercise is that with a 10% content change and a 15% text change the resulting framework detailed in Appendix B-3 appears by inspection to be a highly credible, top-level representation of systems engineering competence. It offers a well-grounded prototype systems engineering competence framework that is a traceable baseline from which the next steps of refinement could be conducted. Initially, this would follow the path of a systems engineering augmentation and structuring of the SFIA Skill Components according to the model in Figure 6.8.

6.5 Outcomes

This chapter has completed the systems engineering triptych proposed in §2.4.3. It demonstrates that it is feasible to take a single process model founded on the rationale of systems reasoning, on engineering discipline and on management practice, and out of this to build a self-consistent set of systems engineering business procedures, organisational capability and practitioner competence. Subject to the condition that this process model has widespread acceptance, it is a powerful route for a united advancement of systems engineering. ISO/IEC 15288 has a legitimate claim to be such a model; accordingly, §4 and this chapter have built their arguments using the International Standard to demonstrate their practicability.

The growing application experience of ISO/IEC 15288, and the continually evolving challenges of international systems business, will suggest many detailed opportunities for future enhancements to the Standard – regular review is an underlying principle applied to all ISO standards – but for the present there are some key steps forward that are already overdue. It has been argued that principal amongst these is the development of the International Standard evolution to act as a reference model for organisational process assessment.

An essential augmentation to achieve this is extension of the International Standard's transformation-oriented descriptions to include a consistent set of work products. This investigation has taken a disciplined first step along this path of evolution. Its definition of the Generic Work Product categories and the formulation of individual work products and their individual characteristics, as detailed in Appendix A, have been adopted by ISO for inclusion in a forthcoming Part 6 of its process assessment suite ISO/IEC 15504. This will result in a model that describes systems engineering capability maturity, and when published will be a companion to the Part 5, concerned with software-intensive systems [ISO, 2006].

Concise definition of work products is also important for the description of architecture, and consequently for disciplined comparison between candidate system solutions and for resolving the interoperation complexities between different systems. The transformation dominated definitions of process, as employed in ISO/IEC 15288, still fall well short of what is required for the disciplined modelling of system architecture. The coalescing of mainstream systems engineering with the rapidly growing and influential architecture practices is likely to depend on resolving the prevailing, unsettling dichotomy in present-day codification of systems engineering transformations and work products; a matter taken up §10.4.3.

It has been shown that the counterpart to a capability model is a competence model. This too has been demonstrated to be a viable development from a common, unifying process model. The prototype described in Appendix B, and the proposed structuring of a body of supporting competence categories, aligns well with the UK's SFIA model. Because of this compatibility with SFIA, the competence model described has relevance to the IET and the BCS (the two primary SFIA consortium members). Its further development should therefore offer benefits for self- and team-development to systems engineering practitioners, and assist issues of career development, organisational role definition and professional registration.

Figure 2.3 does not, however, just convey an objective mapping of information or action relationships. It is also a model of a massively important investment balance between capability and competence that every organisation encounters, whether in the form of proactive strategy or, as too often, an accommodation of default. This balance depends on many factors, and it is a trade-off that should be carefully judged, rigorously managed and regularly reappraised. It is a trade-off between the prescription of procedure and the adaptation enabled by individual and team competence. The well-judged striking of this balance between institutionalised prescription and competent adaptation is strongly indicative of an organisation's maturity. It is hugely important to business competitiveness. Its ramifications are woven into the regimes of defence acquisition practices that are the subjects of the following chapters.

Together with §4, this chapter has illustrated an underlying message: a cardinal systems engineering process model should as a *sine qua non* be simultaneously developed from three viewpoints of the discipline, namely, business procedure, organisational capability and professional competence. Without harmonisation of the images of this fundamental triptych of systems engineering, the different strands currently engaged in advancing the discipline of systems engineering will be ignoring the very principle of holistic thinking they espouse.

7 THE ACQUISITION SETTING FOR DEFENCE SYSTEMS

7.1 Prologue

This thesis now turns to systems engineering applied to the acquisition of UK military systems. It is a context of novel international security threats and responses, high-technology international cooperation, tightening defence budgets, and changing MOD and industry structures and relationships.

This chapter reviews the tides of change in recent defence systems acquisition, and the ebb and flow of systems engineering's fortune within it. Whilst present-day acquisition change appears exceptional, with hindsight it is likely to prove little more than another step in an inexorable march of business reforms. As Proust 'summarised' it, "the only thing that does not change is that at any and every time it appears that there have been 'great changes'." However, if to-day's changes prove to be unexceptional in defence acquisition terms, they could turn out to be of more moment when viewed by the systems engineering historian.

Colin McInnes, giving evidence to the Defence Committee on the background to the Government's Strategic Defence Review, in October 1997, surmised that "seeking efficient defence procurement has become the philosophers' stone of defence policy". Nevertheless, the wide ranging SDR made clear references as to how this transmutation might come about and that one of its bases was to be systems engineering. "A Through-Life Systems Approach is founded on the proven principles of Systems Engineering.....This is a central theme of Smart Procurement" – the Strategic Defence Review of 1998 [SDR, 1998, Study 2F3/7] heralded one of the foundations on which it was to be built, or so it seemed. SDR was the genesis of Smart Procurement (to be later styled Smart Acquisition) and it had explicitly accepted systems engineering to be an essential part of UK defence acquisition.

If Smart Procurement was unequivocal about the nature of acquisition changes; it was no less forthright, but perhaps less judicious, about payback timescales: 20% in 3 years [MOD, 1999]. Coming from the business process improvement perspective, Demming summarised it: "it takes two to three years to change a project, it takes a 'generation' to change a culture". A few of the newly formed Integrated Project Teams, IPTs, the selected torchbearers of Smart Procurement policy, had indeed radically changed in this timescale. However, due to competing claims on resource, inescapable legacies and uncertain guidance, most had not and MOD's acquisition stream as a whole certainly had not changed its culture. That took, and is still taking, much longer.

In October 2005, the retiring MOD PUS, Sir Kevin Tebbit, who oversaw Smart Acquisition, expressed the opinion that the "Smart Procurement Initiative did a great deal, but I think we did not spread the culture of improved practice widely enough. There were pockets of excellence but it clearly was not enough" [Tebbit, 2005]. This underscores the reality of the long-term, cultural nature of business process change associated with acquisition reforms and, within this, changes to engineering's technical transformations and work products.

Ultimately, acquisition reform must be seen as a never-ending action. It is a succession of reform initiatives: successive responses to the inevitable change in the "techno-political-econo-socio" system [Ramo, 1998] that defence acquisition serves. A successor to Smart Acquisition was thus assured. In December 2005 the Defence Industrial Strategy, DIS, was published with full Ministerial backing. Whether the result of providence in the wake of Sir Kevin's views or serendipity, it has provided a replacement engine of acquisition change – not eclipsing but hugely augmenting Smart Acquisition.

"DIS isn't just a policy document – it is a framework for real change that will deliver better results and better value for money.....we need more consistency in [defence procurement]" stated Lord Drayson, the then Defence Procurement Minister [Preview, 2006]. Systems engineering is a prominently declared part of that framework.

Furthermore, under its heading of *'Intelligent customers - intelligent suppliers: the importance of systems engineering'* its systems engineering pronouncement is emphatic. 'We must

maintain the appropriate degree of sovereignty over industrial skills.....where systems engineering skills amongst others may be important ... to support in-service equipment.’ [DIS, 2005p.17] and ‘systems engineering at various levels is critical for the successful acquisition of complex projects and programmes; and in-life upgrades, including UORs, often require deep systems engineering skills and knowledge’.[*ibid*, p.18]

DIS is concerned with industrial and technology policy, and does match the pan organisational scope of Smart Acquisition. Not surprisingly, hard on its heels has come Enabling Acquisition Change, EAC, with its complementary influences of structural re-organisation, re-assigned roles and shifts in budget practice. Together, the DIS and EAC initiatives are recasting the shape of MOD acquisition and re-directing its advancement ‘post-Smart Acquisition’.

A primary instrument in that recasting is the IPT. As the AMS put it “the heart of Smart Procurement is the Integrated Project Team (IPT). We have had integrated teams, sometimes called multi-disciplinary groups, in the past. The Smart Procurement IPT is a much more comprehensive and powerful concept”. It is this concept and its resulting apparatus of change that is considered in this investigation. As the title implies, teaming and effective integration are the essence of IPTs and of their ability to conduct systems engineering. The pivotal role of the IPT is captured at the close of 2007 in MOD’s website: ‘the IPT is the team responsible for the research, development, procurement, support and final disposal of all our defence equipment’ [MOD, 2007].

7.2 Synopsis

This Chapter considers the MOD acquisition scene from a systems engineering perspective. It argues that systems engineering principles underpin many features of the current defence acquisition improvement initiatives and it is surmised that they are likely to remain an explicit feature in continuing acquisition practice advancements for many years to come.

The circumstances surrounding UK defence acquisition improvement initiatives over the last fifty years are examined to build up a picture of present-day defence acquisition concerns and responses. It is a critique principally of implicit, and only recently explicit, systems engineering content in a continual sequence of reform initiatives. It sets the scene for the remaining chapters, in which firstly the impact of systems engineering on IPTs is empirically determined in the next chapter and then, in the course of two subsequent chapters, selected ambitions of these acquisition initiatives are discussed according to each of the ISO categories of system life cycle process. These analyses and projections are rationalised in terms of the systems engineering principles already examined and translated into examples of systems engineering-based acquisition practice advancement.

From the stereotyped post-Cold War approaches, defence acquisition reforms are traced through the watershed of Smart Acquisition, in which responsibility structure, geographical location, management roles and engineering practices all changed. Smart Acquisition was explicitly to depend in part on the principles of systems engineering in order to support revised acquisition practices. Its impact on requirements was without doubt substantial, though not always well-directed; elsewhere its employment of systems engineering was patchy. The systems engineering optimism that had accompanied the introduction of Smart Acquisition reforms was tempered by engineering and management realities, and the challenges of translating principles into practice is accordingly considered.

A follow on step, that of DIS, had amongst its many goals the revival of a, by then, flagging systems engineering message in Smart Acquisition. DIS more explicitly presented defence-oriented principles of systems engineering than any predecessor defence acquisition improvement initiative. It is an initiative that has yet to run its course but its systems engineering edict to both MOD and UK industry has delivered an influential message.

This journey of defence acquisition reform was not taken alone by MOD. Changed security threats and economic climates nowadays are international in nature and international in effect. Subject to localised national circumstances, other nations found themselves having to make corresponding adjustments to acquisition policy and practice. More than ever, these

had to be heedful of the climate of coalition force cooperation and interoperability, of globalisation and of the ensuing rationalisation that has swept through the international defence marketplace. Coordinated international acquisition practices are part and parcel of future coherent coalition capabilities, and common international systems engineering approaches a key component of those practices.

Accordingly, systems engineering-related influences in overseas defence acquisition organisation are analysed to see what messages they may offer. Illustrated by two cases, they underscore themes that have features in common with the UK, but they also expose some contrasting elements of approach. The USA is taken as an almost obligatory end-of-scale reference for size, complexity and international influence. This is balanced with a European slant. Sweden is taken to represent a nation whose defence forces and defence industry face many economic, industrial and cultural challenges that approximate to those seen in the UK.

Despite its trans-Atlantic acronym roots being in 'product' rather than 'project', the IPT is a concept shared with the USA as a response to a similar set of defence acquisition challenges. SDR/Smart Acquisition explicitly saw the IPT as an essential agent of change and it features as a hub of much reform. All this is in keeping with the systems engineering International Standard, which sees the project to be the principle vehicle for systems engineering-based business and the focus of improvement and maturity advancement. Together, this points to the IPT as the primary instrument for defence systems engineering-based transformation. For this reason, MOD's IPTs are taken as the centre of practical attention in this research and serve as an exemplar of its general applicability across many industrial and commercial domains.

The last section of the chapter therefore explores the structure, interactions, dynamics and opportunities of this important organisational unit of defence acquisition. From a system life cycle-based analysis of its structure and functions, a model of responsibility and actions point to some essential, defining characteristics of truly integrated teaming. The responsibilities, difficulties and unfulfilled opportunities seen in IPTs and related to the application of systems engineering are discussed. These factors serve to frame the remaining chapters as they calibrate the prevailing situation, and then explore how systems engineering can advance some crucial, immediate challenges faced by IPTs and the defence acquisition/supply community in general.

7.3 The March of Acquisition Reform

7.3.1 The Drivers for Change

During the Cold War the defence equipment acquisition model had been comparatively fixed. Strategically, the problem was persistent and well defined, and this produced recognisable equipment solutions with substantial precedence. Solutions advanced to reflect new technology opportunities, with the most fundamental of these influences being the evolution of nuclear capability and missile technology. From the latter systems engineering emerged in the 1950s in a form recognisable today.

Given the essentially fixed set of geopolitical threats, the direction of defence acquisition had over time been broad predictability. The required operational concepts were substantially stable and deterministic, and performance edge offered by science, technology and engineering had a huge hand in driving rather than following defence options and military doctrine. A prime tenet of the Cold War culture, and a difficult legacy to subsequently cast off, had been to strive for the best level of weapon system characteristics at almost any cost, by drawing on advanced design and leading-edge technologies to confer a margin of combat superiority.

The essentially symmetric military power structure that distinguished the Cold War brought stability, but at a price. The procurement of defence systems established itself as a complex and protracted set of processes, prone to escalating costs and late delivery. Increasing sophistication and complexity of solutions led to expenditure that could not be sustained and,

always subject to budgetary and political pressures, equipment acquisitions moved forward into their development and production stages even though major risks remained.

A Ministry of Supply [MOS, 1958] study in 1958 reported that on average actual costs were 2.8 times those forecast, and (apart from fixed-price contracting) two main causes were identified:

- changes to operational requirements;
- inadequate understanding by the MoD and its contractors of the scope of the work needed to deliver systems successfully;

To tackle these two issues, the Gibb-Zuckerman recommendations in 1961 proposed an improved acquisition cycle, driven by Staff Targets that defined required military need and Staff Requirements (added subsequently) that defined key performance characteristics of system solutions. A system creation cycle was defined: a Feasibility Study of technical problems; a Project Study to resolve the risks, cost and timescale of an equipment acquisition; and a Full Development phase, proved by trials, ready for Equipment Production.

Nonetheless, continuing cost overruns led Downey, an economist by background, to recommend a more detailed and complete life cycle, upper Fig 7.2., including a Project Definition phase and decision gates controlling entry as each life phase began [MOT, 1968]. An appreciation of the general form of the system life cycle's risk exposure profile, Figure 6.3, prompted an advised 15% of the total development cost to be a risk-reducing outlay in the Feasibility Study and Project Definition phases.

Following Rayner's report on Defence Procurement [DPCA, 1971] the Procurement Executive was born in 1971 and under his leadership, and with an emphasis on "Efficiency Strategy", a new concern with management and "scrutinies" appeared in UK defence acquisition. Rayner's "functional costing" – a focus upon controlling what is spent in relation to the system that is want – was extensively applied. It was also a first step in a new breed of project leader.

It was 1980 when the National Audit Office commenced annual reviews of major defence projects and 1982 when the associated publications of MoD Major Projects Reports to the Public Accounts Committee began. Since then these have presented a continuing litany of problems in defence procurement that relate to budget overspends, delivery overruns or inadequate performance, sometimes all of these.

Levene, as Chief of Defence Procurement in 1985, introduced significant changes of a commercial nature, with competition, including overseas contractors, and fixed-price contracts. This placed defence system acquisition more firmly in the realm of prevailing business conventions. Though defence systems and commodity trading retain distinctions, this profoundly influenced the management and conduct of engineering. Prime contractors were to manage requirements and integrate acquired system solutions against key 'Cardinal Points' in order to clarify the essential criteria of contracts and acceptance.

These principles, however, continued to be flouted. Military, political or industrial reasons were repeatedly advanced for accelerating particular projects before their requirements were fully understood or technologies were well established. This led in 1987 to Jordan recommending professional MoD project managers and rigorous, renewed implementation of the Downey life cycle and procedures, with full risk analysis to be conducted before a Full Development phase [PE, 1998].

With the demise of the Cold War, and the less predictable threat situation that evolved in its wake, highly tuned solutions and long creation periods were rapidly becoming no longer acceptable. The leviathans of post war defence capability now needed to be replaced by adaptable and flexible equipment to meet radically different and often uncertain types of threat.

Whole-of-life management and costing became a focus in the 1990s, with an emphasis on the reliability and maintainability of defence equipment, and a push for integrated logistics support that introduced greater cost, performance and timescale considerations throughout the system life cycle.

However, despite this succession of refinements there was a consistent pattern of continuing cost overruns and delays in introducing equipment into service. A required sea change in defence procurement was, however, already beginning to happen. Able project managers in the Procurement Executive were already flexing the written and unwritten rules that governed traditional civil service behaviour. They were beginning to act outside the 'comfort zone' of time-honoured government procurement procedure to tackle problems that were well recognised but were not being systematically or radically changed in regard to approach. Instances of able, individual action were not, without a clear strategy and commitment from the top, going to effect the required business and culture changes.

Over this period and driven by similar considerations, others too had been addressing similar challenges. As a result of US defence reforms, US equipment was becoming cheaper and, given the open overseas competition introduced by Levene, the protection of UK industrial assets and a UK strategic technology base became an issue. As the Seventh Report of the Select Committee on Defence put it, "The risks of [overseas] dependency are high...and we must still be willing to fund the necessary capabilities single-handed. [*HOC*, 1998, p331]. The strategic management of engineering and technology in defence systems acquisition was being recognised as a weakness.

Following the General Election of 1997, the new government embarked on a review of UK defence. It led to the Strategic Defence Review that re-assessed all aspects of Britain's security interests and defence needs. Published in July 1998, this review examined the roles, missions and capabilities of UK Armed Forces and how these should be adjusted to meet new strategic realities. It signalled the mood of the next round of wide-ranging procurement changes. Many of its defence system acquisition assertions could be traced to a systems engineering-oriented approach to requirements, complex system design and technology introduction, and it explicitly ushered systems engineering into MOD policy.

Primary amongst these systems-related notions was a new 'through-life approach' to procurement decisions that addressed an integrated, whole-of-life cycle view of system acquisition. A 'new' Acquisition Cycle was to reduce the excessive enterprise-controlled hurdles along the system life cycle to just two. The first led to the evolution of concepts and the assessment of feasibility, with targeted early 'investment' to reduce risk; the second led to approval for production of a demonstrator or full manufacture and thus major expenditure.

The vehicle for this revised, complete life cycle was to be the IPT: a discrete acquisition organisation unit with strongly delegated authority and an owned, rather than functionally contributed, 'integrated team' of resources. The project team leaders, fully accountable for their actions, would be conferred with full decision-making privileges subject to the minimum number of enterprise interventions.

The all-or-nothing procurement, a legacy of the cold war's most-capable solution approach, was to give way to equipments with initially less ambitious capability that could be upgraded incrementally, as in Fig 5.4. This would also tackle the cost and time overruns being seen in large mid-life upgrades.

A clearer customer/supplier relationship was to be achieved by separating the agency of acquisition, both organisationally and physically, from the defence customer located in London.

Thus it was that, since 1998, the DPA became MOD's agent for acquiring the majority of products and many services that contribute to the formation of military capability, i.e., the commercially supplied constituents of the UK's Armed Forces. Its mission has been "to equip the Armed Forces" [*MOD*, 2006]. It has undertaken this by procuring new equipment in response to approved requirements and by providing other procurement-related services to its customers. Even with the advent of the DE&S in April 2007 and the dissolution of the DPA title, this remains a clear-cut responsibility.

The role of the 'DPA component' of DE&S is to manage the transformation of military need into requirements against which industry and service providers can be contracted to supply system solutions. It then oversees the acceptability of those solutions from contract placement to acceptance, and supports their transition into the service and support environment. Seeing the world heavily in terms of its acquisition responsibility, 'acquisition cycle' is a synonym in its vocabulary for systems engineering's 'system life cycle'.

With an annual budget of £6 billion, around 700 individual projects and 13,000 contracts, it is the single biggest acquirer of manufactured goods from UK industry. It thus has not only a crucial security role but carries a significant national economic, engineering and technology responsibility. It has thus been at the heart of changes in system life cycle management techniques and in the introduction of systems engineering into the MOD mind set.

7.3.2 A Smart Response

The Smart Acquisition initiative was a joint exercise. The MOD lead was complemented by inputs from the UK defence industry and this was in clear recognition of the engineering business interdependency and synergy between government and UK industry. In addition, McKinsey & Co, who had extensive experience of developments in the US where the DoD had been introducing a range of similar measures under its Defense Reform Initiatives, brought experience of 'reform of procurement practice in other countries and in the private sector, to ensure that [the MoD] are at the forefront ... of international best practice' [SDR, 1998].

Smart procurement involves a raft of many measures, described in the SDR's *Supporting Essays* in three main areas:

- adopting a 'through-life systems approach'.
- 'integrated project teams' to manage each project.
- simplifying procedures, and tailoring them according to projects

A supporting study declared one of the principle aims of defence acquisition to be the adoption of a "through-life Systems Approach founded on the proven principles of Systems Engineering" [SDR, 1998, Study 2F3/7 5, p.5]. This helped set the scene for systems engineering to be a central theme of Smart Procurement.

Figure 7.1 is from the Strategic Defence Review Study [*ibid*, p19] and highlights how systems engineering was seen to address defence equipment and capability acquisition challenges. It specifically points to systems engineering contributions being required at both the project (equating to product) level of responsibility and the capability (resulting in service) level of responsibility.

Systems engineering was thus broadcast to be a cardinal element in the introduction of the acquisition enhancements. Whilst not new to UK defence procurement, as a discipline or set of practices it was not widely, methodically or explicitly practiced in MOD, unlike industry. Suddenly, the pressure was present to introduce systems engineering practices.

From the systems engineering process point of view, the almost concomitant publication of a report on a *Systems Engineering Practices Reference Model* by the Defence Evaluation and Research Agency [DERA, 1997] was both fortuitous and significant. It conveyed approaches that aligned with key SDR's needs, most notably its advancement of the use of User Requirements (roughly Stakeholder Requirement in the ISO standard) and System Requirements throughout the system life cycle. These were adopted almost instantaneously by MOD. They very quickly became a fundamental, even flagship aspect of MOD's acquisition practice, displacing the Staff Targets and Staff Requirements overnight in new projects.

However, the extreme rush to introduce these systems engineering-inspired requirements concepts, together with the dismantling of part of the new DPA's corporate advisory structure to reduce costs and confer greater autonomy on IPTs, led to poorly coordinated and poorly informed implementations of systems engineering practice, notably in the case of requirements. The 'mentoring avalanche' from Waves of Pilot IPT 'breakthrough' led to independent paths of discovery. With hindsight, it is arguable that years of weak and confusing (and highly costly) requirements management practice stem from these circumstances.

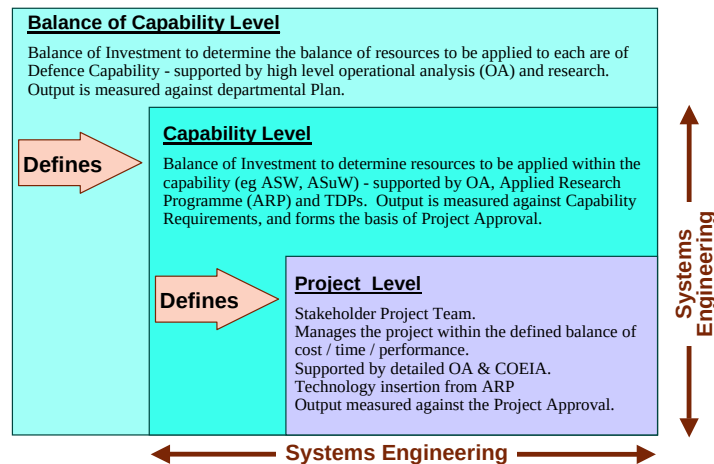


Figure 7.1 The SDR View of Systems Engineering in the Hierarchy of Defence, [SDR,1998]

Nevertheless, MOD's adoption of systems engineering processes that recognisably aligned with the then emerging ISO/IEC 15288 brought greater clarity to responsibility and ownership, with technical processes that distinguished the acquirer and supplier roles more effectively – one goal of the change programme. They also held open implementation decisions that prior to Smart Acquisition were unnecessarily and ill-advisedly being pre-emptively forced down the procurement chain by the user or customer.

Quoting MOD, the definition of Through Life Management is “building on the ‘whole life approach’ and taking into account all Lines of Development to deliver a fully integrated defence capability”. The key features of this approach remain:

- using an appropriate acquisition cycle,
- developing and using a realistic, costed whole-life plan known as the Through Life Management Plan (TLMP) to manage the project throughout the life cycle,
- considering the integration of all the Lines of Development [meaning all elements – products and services – that comprise military capability].

Selecting an appropriate acquisition cycle was a two step approach.

Starting from Downey's cycle, McKinsey & Co recommended the reduced number of stages, shifting more attention to Demonstration in order to de-risk prior to undertaking industrial commitments. In addition, to control progress through the life cycle the major decision gates were reduced to two cost, schedule, and performance parameters, thereby avoiding the over-intrusive enterprise supervision seen to impede project progress.

Changing both stages and approval gates in this way would avoid past, harmful interstitial disruption and resource the discontinuities that had plagued industrial resource profiling (mid Figure 7.2). Two primary decision gates also offered the far greater degree of empowerment being sought at project level. From the initial letters of the stage names of the new life cycle, the acronym CADMID was quickly adopted as an insignia for Smart Acquisition.

MOD's Acquisition Organisation Reform team readily accepted the stage definitions and the acronym from McKinsey & Co but moved both decision gates forward, lower Figure 7.2, to better ensure leverage of early investment and address the imbalance between the commitment curve and the spend curve seen in Figure 7.3 [based on Forsberg, 1996, p80]. The twin systems engineering-related goals of ‘early identification of acceptable performance trade-offs to deliver projects to time and cost’ and the ‘investment of sufficient time and resources during the early stages of a project, with a view to rapid execution thereafter’ were reiterated in the EAC [EAC, 2006, p18].

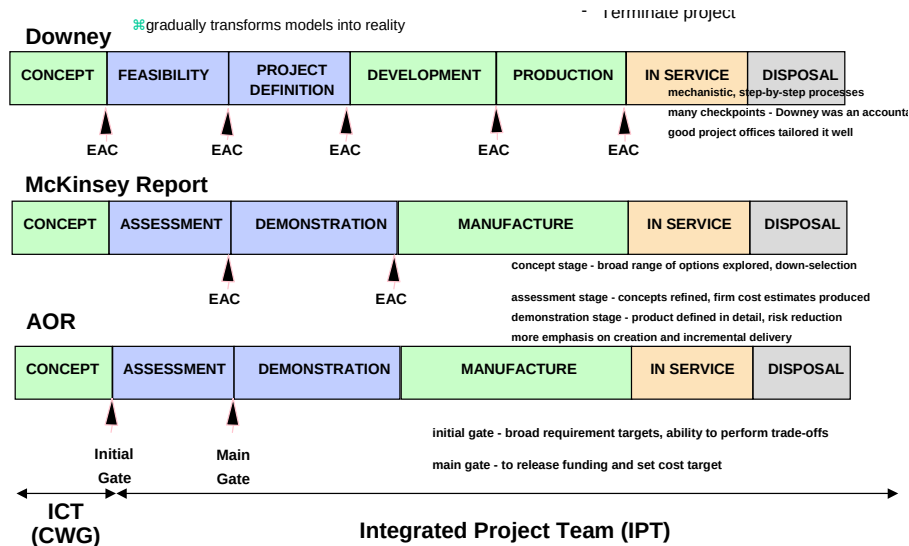


Figure 7.2 Re-profiling the Risk and Commitment Gates through the Life Cycle

Initial Gate approval is a relatively low hurdle at which a wide range of options have been explored. The Assessment stage is characterized by its use of investment to achieve early risk reduction. Main Gate approval of the proposed solution (and also the supplier organization) is required before money can be spent in the Demonstration Stage and Manufacture Stage. At each gate there is a submission that takes the form of a Business Case. This is not intended to contain the unexpected but to have been built up with early engagement with the scrutiny community; it typically takes the form of a joint production with customer representatives.

Although the transformational issues of life cycle processes are vital, just as important are the work product outputs. Typically, scrutiny lays stress on the identity and content of the information work products and any artefacts produced as evidence of achievement.

By May 2007 the Smart Acquisition seven principles, as annunciated in the AMS, remained largely those at the outset; clearly systems engineering-related but not explicitly so:

- a whole-life approach, typified by applying Through Life Costing techniques.
- Integrated Project Teams (IPTs) with clearly identified customers.
- a better, more open relationship with industry.
- more investment during early project phases.
- effective trade-offs between system performance, through-life costs and time.
- new procurement approaches, including incremental acquisition
- a streamlined process for project approvals', [MOD, 2007b].

SDR's Forward acknowledged that 'complex, technologically challenging and high-value systems ... will last for many years.. [with] increasing emphasis on an ability to support and upgrade them through life, as well as having implications for the level of industrial capability ... [including] specialist skills and systems engineering capabilities required to manage military capability on a through life basis' [SDR, 1998]. In section B.1 it went on, 'this demands a high level of systems engineering skills, at all levels of the supply chain ... sustained through the life of the equipment. The significance of this capability varies by sector, but it is generally very important for maintaining our control of how we operate our Armed Forces'. It was a reaffirmation of Smart Acquisition's commitment to systems engineering.

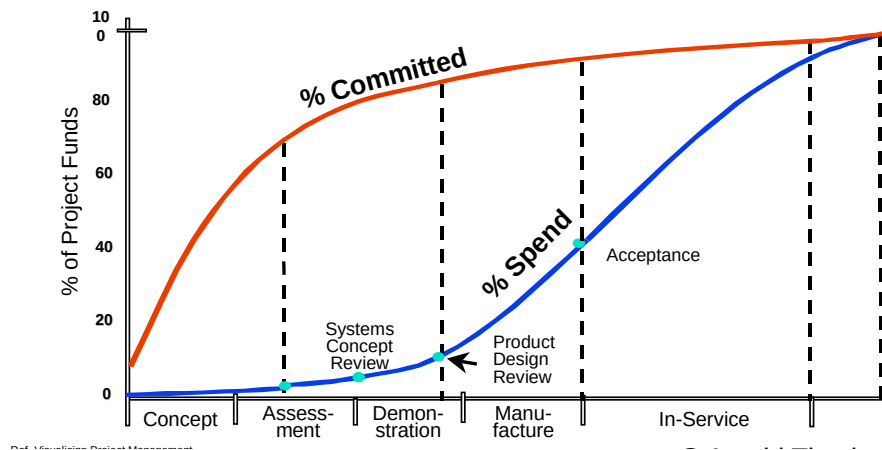


Figure 7.3 Life cycle management and enterprise exposure

The need for a more rationalised approach to the provision of defence capability, for a proactive approach to defining industry's role in that provisioning and for the promotion of a sustainable defence industrial base together led to the publication of the White Paper on DIS. The new government/industry cooperation, that had been heralded in Smart Acquisition, developed hesitantly in many sectors and required more assertive whole system, whole life cycle leadership from MOD. This was expressed in DIS in terms of nine industry sectors, of which eight made specific reference to systems engineering in their policy. Seen as a 'cross-cutting capability', systems engineering itself was given substantial coverage. [DIS, 2005 p.57] 'The DIS has been developed using Key Principles....one of these was recognising the importance of systems engineering, to ensure that our equipment is procured and continues to develop through-life as a result of mature discussions between intelligent customers and intelligent suppliers' [ibid, p.59]

From MOD's point of view a healthy and competitive industrial base is crucial to ensuring that it can continue to procure the right equipment for UK forces at competitive prices. Competing with the attractions of the US defence market provisions, the UK defence industry has a comparative advantage due to its 'growing expertise in the combination of systems engineering skills, agility and supply chain management required to deliver through-life capability management'. It remains MOD policy to maintain an appropriate degree of sovereignty over industrial skills, capabilities and technology, and be willing to build and sustain effective partnering relationships with industry.

The systems engineering revitalisation message of DIS rests on there being 'little use investing in cutting-edge science unless systems engineering capability and vital long-term knowledge is maintained' [ibid, p.7].

The Smart Procurement Initiative Implementation Report committed to an Acquisition Management System (AMS) that would provide a 'one-stop shop' for information [MOD, 1999, para 79] on MOD's acquisition business and processes, and communicate many of the principles and practices arising from Smart Acquisition. It was to be aimed at members of IPTs, at other MOD Staff involved in acquisition, and also at the industry supply base. It enables understanding of the evolving acquisition processes and is updated monthly to deliver new or revised content.

The AMS was initially (1999) populated with material from existing information sources and, in the opinion of many (see for example 9.4.7) still betrays its origins of a hurriedly-assembled, compendium of information at the commencement of Smart Acquisition, rather than a structured, pan-MOD source of reference.

Although the through life message is announced strongly in the AMS, it contains little in the way of explicit systems engineering content. The exceptions were a *Systems Engineering Guide* and a *Systems Engineering Glossary*. These were early DERA contributions that were neither owned nor maintained, became inconsistent with later AMS texts and were eventually withdrawn. As a result, for most of its existence, the AMS has contained a fragmented and

implicit presentation of systems engineering that weakly relates to a structured definition of systems engineering practice.

Top-level navigation is conducted partly through CADMID stages, with links to established patterns of more detailed action and behaviour that match this life cycle model. This has had the benefit of emphasizing an underlying through life management concept, though it has also influenced a sequential, one-size-fits-all mentality to life cycle management.

The AMS provides users with a standard, largely pre-formulated pattern of system life cycle management and life cycle process application, and this eclipses underlying information on which users could base more advanced adaptation of process and decision-making. Nevertheless, systems engineering principles and practices are present in the AMS, though not with the coherence or structuring of ISO/IEC 15288. With the ratification of ISO/IEC 15288 in 2002, the AMS did begin to move towards recognition of the International Standard's framework of systems thinking and business processes.

Building on this and to address the structure and access issues, the Acquisition Policy and Process Framework, APPF, was to have offered 'a single, coherent view of pan-acquisition processes and their owners, derived through involvement of all key stakeholders, and based on ISO/IEC 15288' (AMS, 2004, Version 8.4) and would provide a consistent and coherent overlay on the AMS.

This initiative was to have firmly set systems engineering in the AMS. Its high-level view was to act as a portal into a more detailed, lower-level catalogue of acquisition policies and processes, with links to policy documents, process maps, stakeholder web pages etc. The framework was to be based on ISO/IEC 15288's provision of processes against which the acquisition community could be managed and (significantly) assessed.

The APPF was potentially a powerful mechanism for the strategic presentation of systems engineering to the whole acquisition community and would have been a vehicle for conveying a more explicit systems engineering message throughout the AMS. It was intended to be a management tool through which policy development and process improvement could be instigated, monitored and measured. The completed catalogue was to have provided a definitive list of the high-level acquisition policies and corresponding business processes that make up the system life cycle.

The APPF never became reality. The only evident legacy of this initiative is that 'Key Processes are grouped by ISO 15288 category' [AMS, 2007, Version 9.1] providing the top-level categorisation for the presentation of acquisition processes, and reference to ISO 15288 terminology in the ECC Handbook, Feb. 2005, [MOD, 2007c]. The APPF was a step that, if carried through, could have structured key areas of acquisition according to the internationally ratified systems engineering best practice model.

The result is that for the most part the term systems engineering is given lip service in the AMS and is not presented as a distinct and influential component of business process. The impact of systems engineering is translated into largely isolated and sometimes incongruous descriptions of the many technical and management viewpoints that comprise systems engineering. This fragmented presentation leads to weakly-related definitions of practice that miss the essential unity that systems engineering can bring to complex system acquisition and to the IPTs responsible for this.

This raises the question as to whether MOD can reach level 3 institutionalisation or surpass it as judged by CMMI [SEI, 2002] or ISO/IEC 15505 Part 5 (and 6) [ISO, 2004] without having the benefit of a clearly and logically well-structured presentation of the business processes that constitute systems engineering.

With the appearance of the AOF there is little indication of an advancement on the AMS situation. The Tactical (bottom) layer will - in the future - be the source of the most detailed guidance and instruction within the AOF. Whilst this is being developed it will exist alongside the AMS for a transition period of 12 months. Relevant AMS content will be migrated.

An AOF Systems Engineering Handbook [AOF, 2007] replaces that which fell by the wayside in AMS but like its predecessor has a didactic tendency which may not attract a following of the 'non-converted', despite the reminder that 'Systems Engineering competence is

recognised as a major enabling capability in the Defence Industrial Strategy'. In this context, the term "systems engineering" is defined as:

- An expertise on the techniques and steps for managing the engineering process of complex projects from requirements, through design and manufacture to final acceptance.

Such expertise is expressly linked to Project Planning [project management] and addresses issues such as:

- Technology insertion.
- Development and upgrade strategies.
- Incremental acquisition.

All engineering staff with senior responsibilities must ensure they have sufficient understanding of engineering management (or systems engineering).

Systems engineering must be linked to a specific domain engineering discipline such as air, maritime, land or Communication and Information Systems (CIS) to provide a technical understanding of how to deal with the design and upgrade issues affecting particular types of system - be they information, electronic or mechanical in nature.'

7.3.3 An Uncertain Start

There was a clear expectation of the defence-related intellectual and empirical tool that could responsively tackle the rapidly changing military acquisition challenges seen in a new generation of defence systems' solutions. However, any sense of euphoria over Smart Acquisition's implantation of systems engineering discipline in UK defence acquisition was premature. It faltered as early promise soon lacked leadership, did not identifiably deliver proclaimed returns or proved difficult to harness.

The early promotion of systems engineering heralded a groundbreaking shift to User Requirements and Systems Requirements and by implication other system life cycle-related features. However, despite there being support for the wind of acquisition change, many sensed rhetoric without corroboration; restated common sense not a new paradigm. A somewhat silver bullet status that Smart Acquisition afforded systems engineering was partly self-defeating. Ultimately, there are no simple remedies for overcoming the complexity challenges of defence systems acquisition. New processes (such as provided by systems engineering business-based models), more relevant work products (such as new definitions of requirements), new methods (such as through life costing) and new tools (such as object-oriented data bases for through life requirements management) remain dependent on high levels of technical and management knowledge, skills and experience in order to apply them effectively.

This management of complexity was over-emphasised as a top-down, sequential action and deprecated, with some justification, the prevailing bias towards premature solution influence. Past solutions to new problems were an anathema in the wave of reforms. An excessive focus on URDs being established ahead, and almost in isolation of, solution dependence acted to downplay the reality that systems engineering is fundamentally a bottom-up as well as top-down approach. Dogma licenced a unidirectional ride through a sequence of life cycle processes. The legacy from linear life cycles and the sequential implications of CADMID did not sit with creative complexity of concurrent process.

Smart Acquisition's stress on top-down User Requirements, followed by System Requirements and then Architectural Design, obscured the essential viability and practicability of solutions that is driven up from technology. Excessive time was spent in the early flush of Smart Acquisition with abstract requirements development that paid insufficient heed to the presence of the inevitable constraints of fielded legacies, technology practicalities and interoperability with existing, well-advanced developments. The perception of how these requirements related to contracting issues raised by the "is responsible for" in Figure 4.17 were from the outset ill-understood, poorly defined and lacked training. To an extent this has not been satisfactorily corrected and the accumulated costs after nearly a decade of weak

deployment of this aspect of systems engineering discipline have adversely impacted acquisition costs and the recognition of systems engineering's value.

The early Smart Procurement introduction almost neglected Architectural Design and largely relegated it to be 'industry's problem'. The high level of interaction between Requirements Analysis and Architectural Design shown in Figure 4.9 shows the danger of considering these two processes to be related by a unidirectional information flow. The responsibility is delegated by the IPT, not abdicated to industry, and continual attention to architecture is an imperative of IPT accountability.

There were few 'green field' opportunities in which wholehearted change could be introduced at the commencement of Smart Acquisition. The constraints of existing process, design decisions and industry contracts inhibited changed ways, and this limited the opportunity for change in the majority of IPTs. Most were handling legacy situations in which aspects of a revised systems engineering approach, notably requirements management, could not be introduced without disruption. Too much re-documentation and revisiting of past decisions would have been entailed, and retrospective changes to established projects introduce cost and risks. Legacy and latency New approaches can only readily be introduced into new projects.

Whatever the priority order in Smart Acquisition's 'faster, cheaper, better' slogan, cheaper was a strategic goal. With regard to demonstrating a return on investment, systems engineering has experienced a longstanding international challenge. Being a cross-cutting and integrative discipline, it is a pervasive and complex part of business process. It is thus virtually impossible to prove a dependency of outcomes on specific systems engineering change actions. Benefits may correlate well with systems engineering enhancement actions, but they generally are not explicitly traceable to them and organisations rarely await the controlled effects of one change initiative in isolation before starting the next. In addition, systems engineering enhancement consequences relate to specific improvement initiatives (methodologies, requirements management, contractor management, acceptance), and attributing benefit to action is a matter of competitive conjecture. Explicit proof of systems engineering success is difficult, and measures attributable to effect appear too difficult to conclusively prove; businesses, and business change programs, exist in a complex, multivariate problem space in which cause and effect are difficult to correlate.

Thus it is that a culture shift to entrench systems engineering in the influence curves of Figure is as much a matter of managerial conviction as business metrics. In this regard an informed advocate and a committed champion existed from the outset. The then Deputy Chief of the Defence Staff (Equipment Capability), Lt Gen Sir Edmund Burton, was knowledgeable and insightful about what systems engineering could offer. Under his leadership there had been impetus. Time scales for identifiable returns are, however, long compared with senior MOD appointees or ministerial tenure.

Demming's timescale prognosis is at odds with the tenure of political and military leadership posts. A 'hearts and minds' change, in which the organization's culture has been transformed, only comes when actors behave instinctively in response to new challenges that lie outside the ranges of predetermined procedure. Received wisdom on the most optimistic progression up maturity levels in a change program is 2 years per step (in a 5 step maturity model). For many large organizations, particularly where advancement is dependent on complementary changes in their trading partners, this means change programs of up to a decade.

The implementation of a major process change in a large organization usually causes turmoil and some initial inefficiency. At the outset business changes are a net cost, and the initial pulse of cost and with no measurable quick benefit introduced pressures for delivered benefit. Already launched savings actions were offered to justify progress [MOD,1999, para 39] but only postponed the pressure.

Despite attempts to sweep it away, annuality remained a frustration to the essentially asynchronous enterprise management and project management control of the CADMID system life cycle model. Budget rounds, resource planning, in-year cost saving measures, technology research programmes and test facility cycles are substantially coupled to the annual cycle. In addition, wholly asynchronous events such as politically inspired policy changes, revised accounting practices, MOD planning patterns, cost savings to offset Front

Line expenditure and MOD organisation changes all disturbed the management of individual system life cycles.

Thus, a mixture of unrealistic expectations, slow pace of delivery, some false trails and new leadership faces to invigorate weaker aspects of the business change forced a re-badging. After two years, it was considered expedient to transform the original title of Smart Procurement into the now familiar Smart Acquisition, burying finally any adverse connotations that the term procurement may have carried from the DPA's Procurement Executive predecessor.

There is a given rate at which change can be accomplished. Key military champions and politicians are often not in post long enough to provide required continuity, and focus can be quickly lost. With hindsight Gen Fulton's retirement had a serious impact on the rate of take up and permeation of systems engineering through the ECC and in turn the DPA. Since then, and despite the DPA having no evident champion, the DPA 'supplier' not the ECC 'Customer 1' has led the way in conveying systems engineering to the capability management community. Whatever the extent of this leadership setback, it provided stark evidence of the value of a champion for instilling systems engineering in the MOD culture.

Whereas the principles and some of the practices of systems engineering are plainly evident in a succession of Smart Acquisition initiatives, and can be interpreted as addressing specific areas of a holistic systems approach, they have derived from different camps of influence and have not presented, or mostly even alluded to, any clear systems engineering identity. This subliminal introduction of systems engineering practice has obscured its underpinning and unifying role. The result has been a range of disconnected, though individually helpful steps that still do not emerge in a readily recognisable, coherent stratum of systems practice that is intrinsic to MOD business actions.

7.4 The Paths of Others

7.4.1 Aligning Acquisition Views of Systems Engineering

Defence is an international undertaking. From the operational perspective it is rarely conducted in national isolation. With few exceptions, MOD contributes to coalition forces, finding itself in fresh theatres of operation according to new operational circumstances and novel combinations of capability. It is required to contribute to international federations of systems, in sometimes previously untested deployments that may have to be adapted real-time. This enquiry into defence acquisition practice would therefore be incomplete without some consideration of that international setting,

If the circumstances of capability deployment are international, the circumstances of capability acquisition are no less so. The competing factors of more military capability and lower national budget are naturally leading to international consortia of defence customers and international supplier collaborations. An outcome is increasing international defence trade; one of the factors that encouraged ISO's engagement in systems engineering codification.

For these reasons an autonomous UK national defence business is no longer viable. The need to trade internationally has led to international company acquisitions by UK business and to a growing overseas presence in ownership of previously indigenous UK design and manufacturing capability. This changing industrial climate is a theme of the DIS

Any international systems engineering identity crisis in defence acquisition is steadily being resolved as a result of the recognition of a codified and internationally ratified systems engineering standard. Following through from the DoD Std 499's lead shown in Figure 4.4, better interpretation, presently spearheaded by the ISO systems engineering standard, is helping to serve the international collaborative interests in defence systems.

In the area of *de jure* codification of systems engineering, the two existing major direction-setters for systems engineering in DoD, IEEE 1220 and EIA-632, both of which remain US national standards, are moving to align with ISO/IEC 15288. In recognition of this, the DoD adopted ISO/IEC 15288 on 1 May 2003 for use under the direction of the Under Secretary for Defense, Acquisition, Technology And Logistics, defined in Project SESS-0040. The

International Standard is now acknowledged by DoD 'to provide corporate/enterprise/project level system life cycle process descriptions appropriate for defence acquisition programmes'.

As yet this remains essentially a DoD policy move rather than a substantive introduction into any new procedures, but the ISO/IEC 15288 concepts are increasingly being referenced and used in training. It is a first step towards a more harmonised view of systems engineering across defence acquisition organisation. Where DoD leads, others commonly follow.

The most direct effect of the DoD actions can be seen in the NATO Conference of National Armaments Directors. It has established a policy that ISO/IEC 15288 is to be utilized as a basis for implementing system life cycle management across Nato defence organisations [NATO, 2004].

7.4.2 Systems Engineering in DoD

Looking to national defence acquisition organisation comparators, the most obvious baseline for the UK or other nations to take remains the US DoD. However, comparisons with the systems engineering approach taken in the US DoD have benefits and disadvantages.

The clearest benefit provided is a view of the largest and most technologically advanced national defence force. It was, after all, the environment from which systems engineering emerged. The US offers a breadth and complexity of application that gives credence to approaches adopted. It acts as a point of reference that generally establishes an end of the scale in most dimensions of comparison.

In addition, many nations look to US industry as a supplier of their defence assets and towards DoD with whom they undertake, or anticipate a future need for, collaborative deployments. Aligning with US approaches to systems is assumed to ultimately minimise interoperability issues.

However there are disadvantages. The size and fragmentation of the US DoD leaves observations regarding its approach to systems engineering open to question, with a diversity of emphasis still evident in approaches across the Services.

Further, despite being the UK's closest ally, the order of magnitude difference in budgets and force size do not necessarily commend US approaches as being those most appropriate for the UK. Nor has the US been quick to modify the traditions of systems engineering it gave the world, carrying the inertia of past approaches and thus tending to lag with some of its views.

Nonetheless, no international comparison would be complete with reference to the USA.

Sitting at the head of the DoD acquisition management system is DoD Directive 5000.1, the US DoD Defense Acquisition System, [DoD 2003]. It is complemented by DoD Instruction 5000.2, [DoD, 2003b], which is an evolving Directive that provides the management principles and mandatory policies and procedures for all defence acquisition programs across the whole of DoD.

The Directive and Instruction pay major attention to systems issues, though systems engineering is generally not identified as such. It does however contain an unequivocal directive that 'acquisition programs shall be managed through the application of a systems engineering approach that optimises total system performance and minimizes total ownership costs'. Systems are characterised by hardware, software, and human elements (an ISO/IEC 15288 emphasis) in which human/systems integration is emphasised to be an integral part of the optimised total system.

Though issued subsequent to Smart Acquisition, the Directive traces to Defense Reform Initiatives that were marginally precursors of the UK actions. The Directive emphasises the close working relationship between acquirers and users in order to define acquired capability needs. This is stressed by using Integrated Product Teams (IPTs) that begin during definition of the capability needs. IPTs are required to provide only essential program information, consistent with establishing the program baseline, plans, status and enabling strategic decision-making. The move is toward decentralised responsibility for acquisition to the maximum extent practicable: the mantra of empowerment.

The Program Manager (PM) and the Milestone Decision Authority are permitted to exercise discretion and prudent business judgment to structure a tailored, responsive, and innovative program. Programme managers are made accountable for total life cycle management of a system. The DoD reference life cycle comprising six stages is defined and shown in comparison to the CADMID cycle and the ISO/IEC 15288 exemplar stages in Figure 9.3.

Two life cycle adaptations are promoted. Incremental Development, just as in Smart Acquisition, is the life cycle to be employed where the end-state requirements are known and increments of capability depend on available, mature technology and agreed rates of funding. The so-called Spiral Development, in which end-state requirements are unclear at programme initiation, leads to a life cycle in which future increments depend on continuous user feedback and technology maturation. As stated in §5, the reality of a specific life cycle is likely to be a fusion of the linear, incremental and spiral.

Despite being a 2003 revision, DoD 5000 Conveys an 'old guard' systems engineering mind set that has not rebuilt system models to fully accommodate capability. Generally, past models of systems engineering are stretched beyond useful limits, including clear distinctions between the purpose and use of equivalent stakeholder and systems requirements.

The influence of DoD's almost unavoidable fragmentation flows into the presentation of systems engineering, for 5000.1 is not a single source of direction, in fact its treatment of defence capability as opposed to equipment is a distinct weakness. Capability is better addressed, as its title suggests, by the Joint Capabilities Integration and Development System [DoD, 2003c] re-issued in 24 June 2003. In keeping with the shift of emphasis in DoD acquisition, the Instruction presents a joint perspective of a capability-driven, top-down approach, comparable to that adopted in Smart Acquisition.

Interestingly, and perhaps competitively, the JCIDS Instruction does not mandate DOD 5000. Rather it obliquely refers to it by stating that this Instruction does not preclude the use of the (almost contemporaneously issued) DoD 5000 documents.

This Instruction is principally concerned with identifying, assessing and prioritising US joint military capability needs. Its drive is towards a better-structured, commonly-applied set of processes that permit the definition and the development of an integrated set of joint capabilities. It is also a response to the increased influence of networked, distributed operations that need to possess flexibility and responsiveness to changing operational circumstances.

The JCIDS Instruction is expressed in terms of a strongly process-oriented, sequentially-stepped view of capability creation. The processes are expressed in terms of transfer functions that achieve a logical, end-to-end sequence of information transformations and the associated organisational decision-making mechanisms. They define a sequence that moves from needs identification through to the establishment and control of programmes that will deliver capability, including the delivery of new equipment where selected. Delivery includes other elements of an SoS or common elements of a FoS, and/or changes to complementary doctrine, organization, training, leadership and education, personnel and facilities systems in the existing operational environment.

Recognising the realities of international trade, the Instruction states that capability proposals are to consider the unique attributes of international systems from allies and related cooperative opportunities. This is an element of the decision-making and approvals aspect of the JCIDS processes and responsibilities.

The JCIDS Instruction acknowledged that cultural change is required within DoD for the new JCIDS process to fully work. It is envisioned that future revisions of the Instruction will be required to complete what is seen as a major transition to effective acquisition management of joint capability.

A further fall out from DoD fragmentation can be detected in the emergence of architectures, or more specifically architecture descriptions. Traditionally, the DoD has followed a threat-based force-planning construct to drive system acquisition. As a result requirements have often driven stand-alone solutions to specific scenarios and this has led to uncoordinated solutions and also a focus on materiel solutions to joint warfighting. To control this, funding at

decision gates now depend on compliance with validated and approved integrated architectures.

The lead-in to this commenced in the early 1990's when DoD began to address its difficulties in comparing C4ISR architectures developed by competing contractors (and DoD commands). Different parties designed and developed their own independent capabilities and solutions despite strong similarities amongst the requirements, performance criteria and contexts of application. The outcome was disparate and un-relatable descriptions of systems that led to non-integratable, non-interoperable and cost-ineffective SoS. As a result, the Office of the Secretary of Defense (OSD) chartered an approach for comparing C4ISR system architectures and this established action substantially independent of any wider systems engineering approach. Its legacy continues.

The goal of the C4ISR Architecture Framework initiative, and its subsequent refinement as DoDAF, is to enable system descriptions developed by the various Commands, Services, and Agencies within DoD to be standardised in their form. The outcome is descriptions of systems that can be compared and correlated to:

- assure that systems are integratable across the defence community
- provide the basis for an audit trail that relates current (as is) and postulated (to be) systems to measures of effectiveness for mission operations.

With diverse, vested Service and Command interests, each wishing to see their existing approach in the result, there was and still is evidence of inconsistency arising from overlaps, incompatibilities, gaps, unnecessary prescriptiveness. When released in 1996 it was recognised that significant work was still needed and in the light of experience that situation continues to the present

Rather than see the Framework as a potential panacea for SoS, it is more helpful to see the architecture as a key piece of systems engineering methodology that, set in a wider context of other systems practices, makes a major step in discipline and uniformity that can lead to integrable, interoperable SoS.

The Defense Acquisition Guidebook is designed to complement DoD Directive 5000.1 and DoD Instruction 5000.2 by providing the acquisition workforce with discretionary best practice that should be tailored to the needs of each program. A substantial part (Chapter 6) of it is devoted to systems engineering and its first reference to models of systems engineering is to ISO/IEC 15288 [DAU, 2006]

7.4.3 Swedish Approach to Systems Engineering

If a coherent presentation of systems engineering is difficult to interpret from MOD's AMS, and is little clearer in DoD 5000/CJCSI 3170 and their supporting documents, other approaches having greater clarity and coherence already exist. A noticeably different balance in the explicit/implicit approach to systems engineering is evident in Swedish Defence Materiel Administration (FMV) business practices.

The legacy of past approaches and the prevailing circumstance in FMV have contributed to differences of emphasis in their approach compared with counterpart organisations in the UK and USA. For instance, it is policy that, given the level of influence that FMV finds itself able to bear on the wider world, it is predisposed to align with internationally endorsed approaches that can leverage best effect from its efforts.

It is also the case that FMV traditionally has had a through life technical responsibility. This is due to stronger controls of production and logistics than DPA, including the design of often complex interaction of logistics with other elements of capability, and also its frequent role as an integrator for systems. FMV's systems engineering approach thus may be a useful pointer for DE&S.

The following assessment is based on a review conducted in conjunction with Dr H.Lawson [Arnold, 2005] of FMV documentation and discussions conducted in February 2005 at FMV with Staffan Näström (Chief of Operations), Eddie Lindqvist (Director of Enterprise System) and Olle Bååthe (Director Technical Coordination), and at the Swedish Defence (FM) Headquarters with Colonel Björn Ekstedt, Supreme Commander's Comptroller.

After adverse experiences with an overbearing and intractable approach to defining engineering life cycle process, FMV decided in 2000 to follow a modular process approach in its business operations. In so doing, it was determined that certification according to three standards; namely ISO 9001, ISO 14001 and the Rules for Military Aircraft was to be achieved.

FMV is an organization with multiple “customer” relationships. FMV provides both acquisition and in many cases systems integration services to their primary customer, the Swedish Armed Forces (Försvarmakten or FM). At the same time, the Department of Defence (Försvarsdepartementet or FD) is also a customer of the services of FMV. The FD as a governmental department has both the authority and responsibility for military matters and establishes the goals and missions for both FM and FMV.

FMV as an acquirer organisation requests and evaluates offers, and prepares and administers contracts with military equipment and services suppliers. In addition and where appropriate, FMV conducts the integration of products to be delivered to FM. As a result, FMV has thus established and continues to develop technical competences appropriate to the acquisition and to the integration of capabilities. This has depended on a close relationship with the Swedish defence industry staff moving from FMV and to defence suppliers and vice-versa.

FMV has established a set of policies and regulations that guide the organization called in Swedish (Verksamhetsordning (VO)). One such document deals with “Production” that FMV provides, primarily to FM. The term Production describes the value added by FMV for its customers and roughly translates to ‘business management’ in English.

As the embedding of ISO/IEC 15288 in procedures grew, a deeper understanding of its utilization transpired and the contents of VO Production documentation also evolved to define in a clear manner the policies and regulations under which FMV production takes place. Guidance is provided by procedures on the responsibilities and role regarding the customer, on the system life cycle management of product and service delivery to the customer, and on the relationship to suppliers of platforms, subsystems and networks elements.

These relationships are presented in the VO-Production document in terms of three cardinal life cycle models that map to the responsibility hierarchy as presented in ISO/IEC 15288.

At the level of FMV’s enterprise management the annual planning cycle is conducted in response to FM ‘Equipment Plan’ requests. This draws on a set of business planning processes that broadly equate to the Enterprise Processes of ISO/IEC 15288; the left side of Figure 8.6. This plan initiates, monitors and controls the complete life cycle of ‘products’ or systems.

Each product is managed according to a generic framework through its complete life cycle using a stage framework similar to CADMID. This model explicitly relates to FMV responsibilities; the MOD Manufacturing stage is viewed as the period of acquisition, and the in-service period is not directly conveyed and is assumed to be largely contemporaneous with maintenance period that is contracted for. Decisions gates separate each stage, and permit the evaluation of progress and risks associated with development, production and maintenance of the system. Although the model implies a sequential flow, iterations and concurrency of stages are performed.

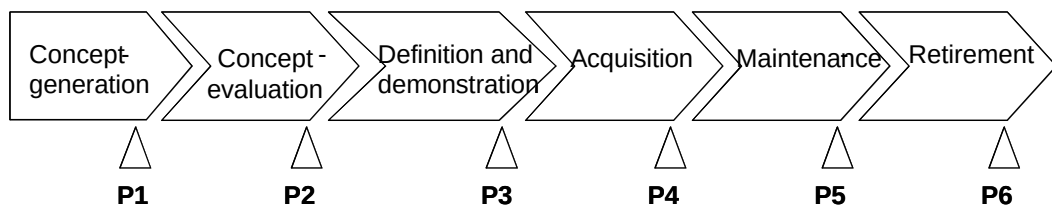


Figure 7.4 Life Cycle Model for a System Product

During this life cycle model for a system product, actual work to be performed is executed in the form of projects (in Swedish uppdrag). This second model provided in the VO-Production

relates to the cyclic nature of project as portrayed in Figure 7.5 and this relates to a minimal set of project process in Figure 7.6. However, a fuller set of Agreement Processes than is defined in ISO/IEC 15288 governs agreements, reflecting the significance and nature of international trade. A project may be defined for an entire life cycle or apply to one or more stages within the life cycle of a system. The division into projects enables FMV to continually evaluate progress and make decisions as to continued execution of plans, modification to plans or termination according to the model in Figure 7.4.

Given the policies, regulations and guidance of the VO-Production document, a set of processes has been developed for usage within the contexts of the life cycle models. These process descriptions have been produced and are maintained via the usage of the Qualiware software system. The entire system of processes is available on an intranet within FMV. Those processes that have been implemented in relationship to production are indicated in Figure A-5.

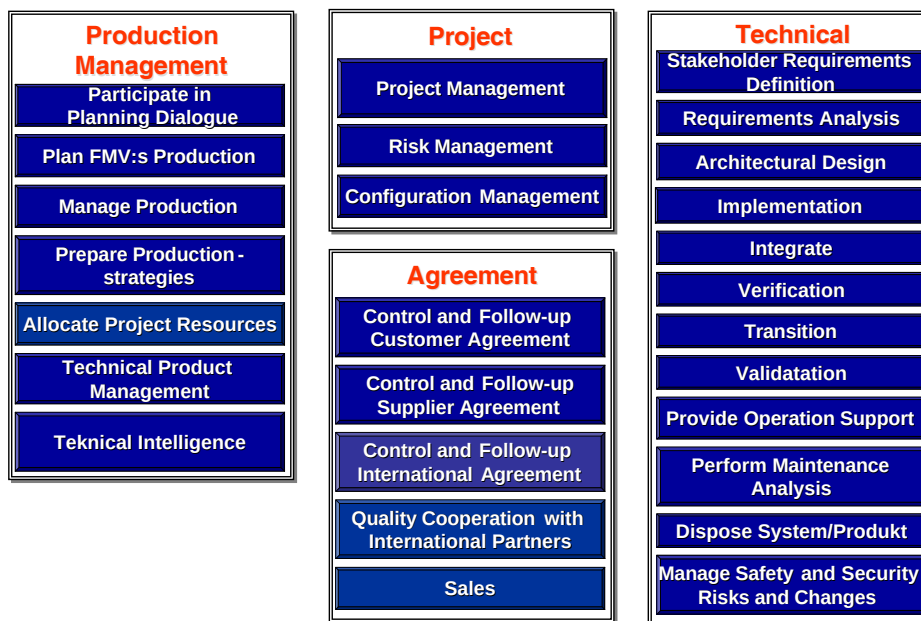


Figure 7.6 The FMV System of Production Processes

The influence of the process categories of ISO/IEC 15288 is quite evident. Some processes have been introduced by "tailoring" the International Standard, notably in 'Production (Business) Management' and in Agreement, and other have been concatenated or duly relegated. In each case the conventions of the International Standard have been observed [ISO, 2002, Annex A]. This tailoring has provided particular support for the three models provided in VO-Production above.

In effect FMV has modified the system life cycle processes to describe three interlocking cyclic elements and these equate to the hierarchy of organizational responsibility and control defined in Fig 4.18: an enterprise cycle, a project cycle and a technical cycle, with agreements building the trading links to partnering and internal organizational units.

The processes required for supporting the ISO 9001, 14001 and RMA quality management system requirements as well as the tailored ISO/IEC 15288 processes are documented on an intranet web. The structure of flows that arise from collective sets of activities are also described on the web. The web therefore provides a structured roadmap for the business of FMV. Two aspects of the intranet web implementation are called the Process Web and the Flow Web.

In the Process Web each process has both a process owner and a process leader that have authorities and responsibilities in respect to the process. Available in this Web is the definition of roles, checklists, requirements documents, and document schema. This is an indication of where the APPF may have taken the AMS.

In the Flow Web there are four typical flows that define a large part of the production business of FMV. These are:

- Offer Project
- Execute Project
- Planning Dialogue and Trade-offs
- Planning Premises

Like processes these flows have owners and leaders with corresponding authorities and responsibilities. The flows are described in a manner similar to ISO/IEC 15288 processes; that is, purpose, outcomes, and description. The typical roles that are involved in executing the activities are also described. In detailing the flows activity steps (taken from processes) are connected. Work products generated by the flow activities are also described.

It is important to note that the processes and flows are meant to provide detailed guidance to specific projects and activities. They can be applied verbatim (which is the easiest and most compelling course of project action) or they can be subject to further tailoring as required. FMV has developed a related training course in which it expects all personnel to participate.

Evidence of the external influence of this approach can be seen, in that during 2000, the Swedish Armed Forces (FM) prepared a new Directive for Information Technology (DIT) for use across Swedish Defence. In this it decided to employ the concepts and principles and processes of (at that time unratified) ISO/IEC 15288, as well as ISO/IEC 12207, as a basis for the directive. This evolved into a 2004 version of the directive called DIT-04. In implementing DIT the various stages of the life cycle are actually viewed as processes (which is consistent with the concepts conveyed in Figure 5.1). The usage of DIT has led to an improved understanding of the business of implementing and integrating IT infrastructure systems in the Swedish Armed Forces and has helped in communications between FM and FMV.

In summary, FMV's systems engineering policy and project conduct are intimately underpinned by ISO/IEC 15288 adoption. For FMV, ISO/IEC 15288 is the definitive exposition of systems engineering principles and a basis of its practice.

Whereas the principles and some of the practices of systems engineering intended to meet anticipated defence acquisition situations have been assimilated into the AMS and its procedures, a clearer, more structured and prescriptive approach has been followed by FMV. As with the AMS, FMV's Process Management System is a publicly available business system and it is exerting influences on their primary trading partners, primarily the Swedish Defence Force and Swedish industry alike. Specifically, a number of defence-related suppliers in Sweden have begun to examine ISO/IEC 15288 and how they can constructively employ it in their relationship with FMV, as well as for their own internal operations

The FMV business approach is built on a documented, compliant and tailored use of the systems engineering International Standard. It is leading towards systems engineering becoming an intrinsic and explicit business driver. This is not to say that systems engineering is a doctrinaire FMV approach; FMV staff are as creative and resourceful as DPA in their undertakings, and they equally value empowerment at the project level. Their practice is, however, more firmly guided by systems engineering principles and they are closer than DPA to instilling a systems engineering culture that, to quote the survey in §9, is something the "team lives" and a "way of thinking". Whether by design or chance, FMV is moving towards a systems engineering capability maturity level of Institutionalization (Level 3) and is likely to arrive at higher level before DPA or its successor DE&S

The development of the organizational policies at FMV according to the Verksamhetsordning (VO) for 'Production' stipulates a life cycle for managing customer interactions (including short and long term planning), a life cycle for projects to be carried out in agreements between the customer and FMV, and a life cycle for products which stipulates the stages and decision gates. In order to achieve this, ISO/IEC 15288 has been explicitly adopted as the guiding instrument for policy and process. In its approach to educating and training systems engineers, FMV has utilized the system technical processes provided in ISO/IEC 15288.

Business change programmes require unshakable leadership and in interview (1st February 2005, Stockholm) Staffan Näström, FMV's Chief of Operations, set a resolute tenor. Paraphrasing him - the introduction of such changes requires the provisioning of adequate time for introduction, the training of personnel at all levels, and then persistence with frequent measurement and improvement efforts; the introduction of process change and its effective application takes time and organisations have to persevere. His long-term championing and his team's committed support were evident contributors to a growing FMV systems engineering culture.

In summary, at FMV the policy has been to move towards the internationally ratified model of systems engineering. Strategic documents that drive their value-added production have been developed according to systems engineering thinking and this evolved from 2000 onwards as ISO/IEC 15288 was being consolidated. The International Standard is therefore a recognized and authoritative element of business policy, and is employed as a token of FMV adherence to practices that conform to the expectations of its trading partners and stakeholders.

7.5 The Nature of IPTs

7.5.1 Whole Life and Trans-functional

The empirical element of this research described in the following chapter questions how from a management perspective systems engineering is viewed in IPTs. As a prelude, the reasons why IPTs were established by Smart Acquisition, the business role they perform and the contributions they make to through life management are analysed.

The move to IPTs first emerged in the USA. The Federal Acquisition Streamlining Act of 1994 simplified acquisition of commercial items and allowed DoD to explore innovative acquisition procedures. From this the Secretary of Defense directed 'immediate implementation' of IPT approach, including the manufacturers concerned. [DoD, 1998]. The concept was introduced by McKinsey and was intended to transform past adversarial relationships with industry into productive partnerships. The approach taken was to place renewed emphasis on the importance of working as a cross-functional team to maximise overall performance. [OUSD, 1995]. Process was seen as instrumental in this reform [DUSD, 1994]. As the DoD Guide to Integrated Product and Process Development put it [DoD, 1996], 'Integrated Product Teams are cross-functional teams that are formed for the specific purpose of delivering a product for an external or internal customer. IPT members should have complementary skills and be committed to a common purpose, performance objectives, and approach for which they hold themselves mutually accountable'.

Apart from the title though not acronym change, the Integrated Project Teams that McKinsey & Co proposed for Smart Acquisition followed this concept. It led to much structural and responsibility change. As Sir John Bourn, head of the National Audit Office, noted: 'MoD adopted a pragmatic strategy to achieve early implementation of IPTs, re-allocating around 10,000 personnel into some 130 IPTs in 18 months' [NAO, 2002a].

IPTs were a central enabler of Smart Acquisition intended to improve acquisition by moving from the functionally based management and reporting structure of the Procurement Executive to a project-based organisational structure in the new DPA. They removed lines of traditional enterprise management influence that fed into the values and behaviours of individuals in procurement teams, and established a high degree of team independence: empowerment was a Smart Acquisition strapline. The co-operation engendered by this team approach was, on US experience, expected to lead to earlier problem solving and more efficient development

Critical factors for the formation of a successful IPT are seen to be;

- all functional disciplines influencing the product throughout its lifetime should be represented on the team;
- a clear understanding of the team's goals, responsibilities, and authority during that lifetime.

The team's composition is made up of all those with a stakeholding in the success of a project. This comprises those with direct or 'dotted-line' responsibilities for operational requirements, scientific contributions, procurement management, contracts, finance, training and logistics functions. It also includes industry personnel, except during the assessment of competitive bids. The IPT thus attempts to include all the skills necessary to undertake and manage a project and they operate according to processes and practices that are team-oriented.

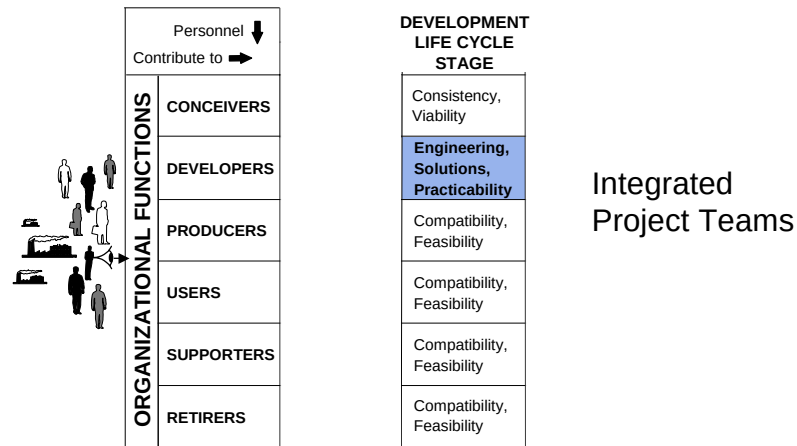


Figure 7.7 The Functional Contributions of the IPT to a Life Cycle Stage

The principle of an integrated teaming operating at a stage of the system life cycle is conveyed in Figure 7.7. General life cycle concepts are presented rather than CADMID and the terms are derived from the exemplar life cycle used in ISO/IEC 15288. It shows how five areas of acquirer/supplier function representation are assembled as a single team. Each brings particular skills and makes their primary contribution at different stages of the life cycle. In addition a sixth contributor represents the interest of the user in the team and either comes from, or has a 'dotted line' responsibility to the user/operator/customer organisation. The equivalent to this role in an IPT is the Requirements Manager, described as "the DEC's man in the team" during S7's interviews.

Figure 7.7 shows the integrated team operating in the Development Stage of the ISO/IEC 15288 exemplar life cycle. It identifies in the shaded box the key role of the developer function in the team to be engineering, with its focus on solutions and attention directed towards the practicability of that solution.

Those team members that have already made their primary contribution, in this case the conceiver function that has previously been engaged in establishing the system concept, continues to actively contribute by examining that the engineering solution maintains consistency with their vision and has viability to provide a useful outcome.

Those team members yet to deliver their primary contribution assess the work of the Development Stage for its compatibility with their anticipated contributions, such as the application of designated enabling systems, e.g. production or support. They assess the feasibility that the evolving solution can indeed be produced, supported and ultimately retired from use. Though their own primary contribution is yet to come, they are fully interacting with the development actions.

The vertical columns of Figure 7.7 convey the cross-functional nature of an IPT, with its common focus on the prevailing role of the team (in the shaded box). The unity of purpose, shared responsibility, and team identity and loyalty lead to effective decision-making regarding trade-offs between system performance, through life costs and time. This outcome of the IPT concept is an important property of McInnes's philosophers' stone.

IPT are integrations, not just of complementary skills (responsible for implementation technologies and technical specialisms such as Command and Control) but also of complementary concerns or views of the life cycle.

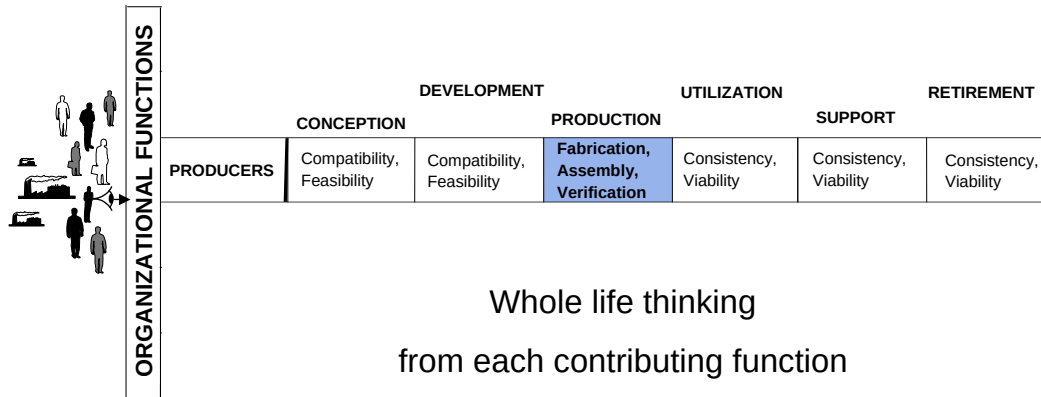


Figure 7.8 The Whole Life Contributions of an Organisational Function

Figure 7.8 conveys the essential continuity of functional, through-life contribution that characterises the IPT teaming concept. As defined in the Capability Management Handbook, "The Smart Acquisition IPT is characterised by its 'cradle to grave' responsibility" [CMH, 2007] As shown, a function that has already made its primary contribution, in this case the producer, continues to exercise a responsibility for ensuring that the system remains consistent with its existing decisions and that it advances, according the judgements of their functional perspective, towards a viable outcome. Decisions, intentions and understanding attained in preceding stages are fully taken account of.

Likewise, prior to making their primary (shaded box) contribution in the life cycle, their contribution is one of analysing the compatibility of the current system state with their anticipated future actions and assumptions, and of assessing that a feasible approach is emerging. The task is to continuously look ahead, rehearse down-stream problems, reduce risk, increase product definition and refine service definition, reassess costs and ownership issues,

This ensures that forward planning takes account of future resources, skills, enabling system and provisioning needs, as seen from the concerns of all stakeholders during the life cycle, crucially including those of the user, who in previous acquisition regimes in the UK and overseas had been treated as somewhat distant observers. Each particular function has an acknowledged through life responsibility and this enhances the application of through life costing techniques: a stated goal of Smart Acquisition reforms.

In the case of the teaming view of the IPT in Figure 7.7 the shaded box represents the prevalent function being performed by the whole team, whereas in a life cycle view the shaded box indicates the point through the life cycle that functions make their primary contribution.

It is the attributes of design feasibility and solution viability brought by the two dimensions presented in Figures 7.7 and 7.8:

- compatible integrated team contributions;
- consistent through life responsibility;

that are central to the IPT concept.

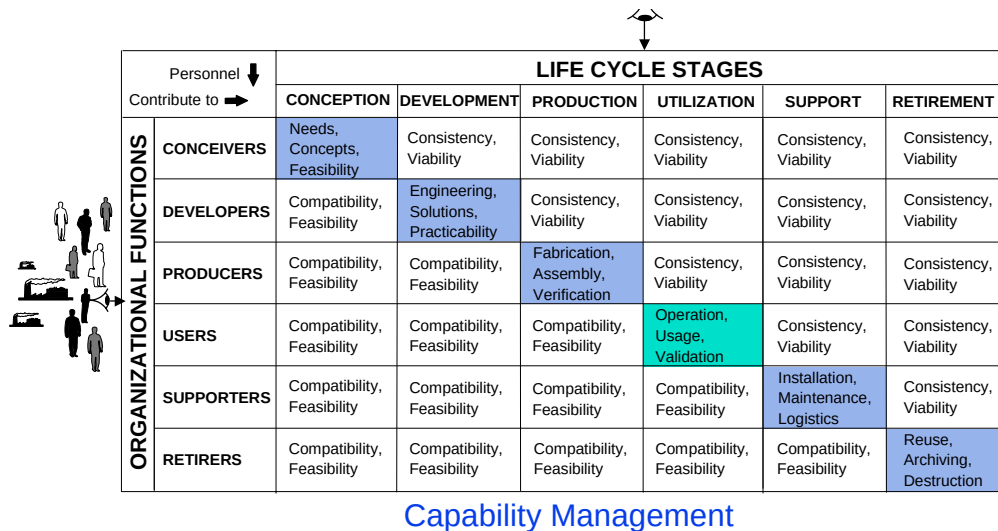


Figure 7.9 The Whole Life Contributions of IPTs

Combining both dimensions of an IPT, a picture is built of all functional contributions over all stages of a systems life cycle, as shown in Figure 7.9

Team actions thus progressively build up, and continue to monitor, design consistency and solution viability. As the system evolves through its life cycle, it moves from models of a feasible solution to a viable operational solution that eventually can be withdrawn in an appropriate manner from service and decommissioned.

Figure 7.9 illustrates that the true IPT consists of a fully integrated set of actors who represent all aspects of the life cycle and are active concurrently and continuously. It achieves the goal to “apply functional expertise in a team-oriented manner on a global-optimisation basis” [IPPD p1.3].

As the system passes through stages, the focus of contribution shifts from one functional role to another and with it the whole team’s focus of activity. For simplicity of presentation, the functions here are chosen to align with the nature of the life cycle stages. In practice a vastly more complex matrix, comprising management, technical, commercial and administrative functions, with their respective pattern of contributions, would arise during the life of an IPT.

The shaded boxes in Figure 7.9 more faithfully present the familiar waterfall life cycle model. Firstly, it conveys that the waterfall model is not a sequence of life cycle processes but stages through a life cycle. Progression through the waterfall steps is confirmed by satisfying business achievement criteria, not sequential completion of work products.

Secondly, it conveys ‘feedback’ from the concurrency and the forward and backward looking decision-making. At any stage the interests of all subsequent stages are represented to the best of the team’s ability. It avoids any through-life division of responsibility that threatens coherence of the product creation and service provision and inhibits any tendency towards an ‘over-the-wall’ life cycle management culture of the past.

With the exception of the operation/usage/validation focus in-service, the diagonal line of shaded boxes describe the primary thread of life cycle responsibility now held by the DEC and his team through life.

As SDR put it, “we should adopt a through-life approach to projects covering both acquisition and in-service support” and the move to create the DE&S follows naturally from this. Prior to the DE&S, when an IPT simultaneously spent funds from both the DPA and DLO Top Level Budgets dual accountability existed.

If ‘product’ inaccurately captures a system through life view in the US acronym IPT, ‘project’ in MOD’s acronym is unconventional for a “cradle to grave” activity. Whilst in theory it is possible for a single project to have complete responsibility for a system-of-interest throughout its complete life cycle, this definition is at variance with conventional interpretations, including that of the Association of Project Managers (APM) on whose

practices MOD bases its staff development and certification. Their definition, repeated in successive editions of The Acquisition Handbook, e.g. [MOD,2006], is a 'unique set of co-ordinated activities, with definite starting and finishing points, undertaken by an individual or organisation to meet specific objectives within defined time, cost and performance parameters'.

Conventionally, a project is considered to operate according to time-segmented responsibilities conferred or renewed in stages by enterprise management according to decision gates defined by initiation and termination conditions; this is the basis of Table 5.1. Balanced teaming and the breaking of functional reporting lines, not continuous project, are the essential mechanism by which IPTs ensure that system cohesion, integrity and feasibility is retained, and that system consistency, practicability and viability is carried forward. It is more realistic to envision a metamorphosis of the vertical columns in Fig 7.7, the integrated team, through successive stages, transferring knowledge and even specific staff to convey vital experience and skills.

The significantly different functional focus (shaded boxes Figure 7.9), skill profiles, emphasis of life cycle process application, service/civilian levels, procedures, geographical locations, enabling systems, infrastructure, immediate reporting lines and cultural influences that distinguish system conception, realisation, in-service support or disposal inappropriately stretch the meaning of a discrete project. Despite the establishment of the DE&S, these distinctions will remain and Figure 7.10, a tailoring of Figure 4.18, conveys that despite a unified enterprise management level, snapshots of a project as it metamorphoses according to changing system state and stage in its life cycle. Overall control is exercised by a unified DE&S enterprise management through the primary mechanism of decision gates; variously, Gates 1 and 2, gates in incremental life cycle models, periodic MOD budget rounds and government edicts.

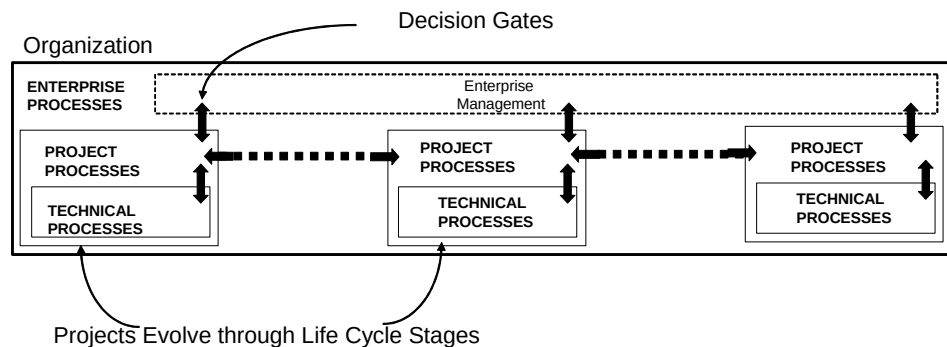


Figure 7.10 The Project Metamorphosis through the Life Cycle

In changing the original US name Integrated Product Team to Integrated Project Team, MOD addressed two issues. 'Marking spin' encouraged a UK stamp of origin on a concept that had emanated from the USA; there are notable precedents for this strategy. More insightfully, the move towards capability that lies at the heart of Smart Acquisition does not align well with the stress on 'product' in the US acronym. However, the term Integrated System Team would have more accurately conveyed the team's focus and given more impetus to the drive for the 'proven principles of Systems Engineering...[to be] a central theme of Smart Procurement' – a missed opportunity to assist a systems-oriented business change.

7.5.2 360 Degree Interactions

MOD's terse definition of an IPT is 'the body responsible for developing the System Requirement Document, devising equipment solutions to meet that requirement, and managing the procurement and in-service support of the equipment' (*AMS Glossary definition of IPT*) [AMS, 2007], [CMH, 2007, Glossary]. They are teams are created within the DE&S for the purpose of acquiring from industry and delivering through the Customer 1 sponsor to Customer 2 a specified equipment and an increasing proportion of other elements or DLoDs that compose defence capability.

This glossary definition of an IPT, however, downplays the breadth and depth of role of a defence system's principal enabling mechanism: the integrated team that is ultimately responsible for the conduct of its effective formulation, realisation and support. As conveyed in the ISO/IEC 15288 model, its top-level project team responsibility for system supply should expressly be the paramount influence on how systems engineering is conducted down the defence enterprise alliance/supply chain. Within its teaming must exist the resources, competence and controls that enable the execution of engineering processes, their effective application based on systems reasoning skills in order to reduce complexity and risk to tractable and containable levels, and the controlling and coordinating actions of management necessary to guide this engineering toward achieving customer goals; in short exhibiting capability to direct and conduct systems engineering. Being a bystander or an industry follower in this role is a failure of IPT responsibility and an invitation to risk accrual.

The IPT is in many regards the business organisational corollary of the technical system they are responsible for: an entity that must act, interface, communicate, decision make, externally influence and deliver outcomes, with a comparable order of complexity and the intricacy – an ordinance of requisite variety. It is tenable that it is more complex and challenging, for it operates in the complexities of the social environment. Just as in a hierarchy of system structure and interaction, IPTs must conduct these functions at the super-ordinate customer service level, at the peer IPT product interoperability level, and at the subordinate supplier implementation level.

Like any link in the complete span of a supply chain, and as illustrated in Figure 4.18, IPTs are equally as much suppliers as acquirers, and the team is faced with the responsibilities of upwards and downwards trading interaction. Coupled with this is the need to interact with sister IPTs, whose developing solution outputs impact an evolving future target operating space. The IPT is faced with continual management of these 360 degree trading boundaries.

Equipment programmes are driven by Directors of Equipment Capability (DEC), supported by their Capability Working Groups. This requires each DEC to define qualitatively and quantitatively the strategic capability that must be generated over periods of time in order to satisfy the operational needs in their area of responsibility, and this is presented in Capability Area Plans (CAPs).

The development of a CAP must be responsive to anticipated operational needs, ranging from civil humanitarian relief to wartime conflict situations. These plans distil envelopes of anticipated operational capability, ultimately assembled from flexible federations of equipments and other DLoDs, into carefully structured investment plans that make demands on a range of top-level MOD budget holders. They have much in keeping with the management of systems-of-systems in which materially autonomous assets must be brought together as federations to achieve the sort of coherence expected of linear hierarchies rather than parallel and concurrent lines of resource control.

The CAPs define an unfolding capability with time. These plans and their management need to accommodate the practicality of successive versions of SoSs co-existing in the battlespace. Constraints, in the form of pre-existent, deployed solution (legacy systems) influence options, and individual systems may (at different times) form part of different operational capabilities.

Over the period of Smart Acquisition the categorisation and definition of cardinal capabilities according to the responsibility spans of Capability Managers and DECs in the ECC organisation has continually evolved. The capability areas and their relationship to specific equipments has been subject to a measure of fluidity as the relationship between, and responsibility for, services and products has been tested and refined through experience.

Establishing consistency across all areas of capability is founded on the measurement of capability and the identification of gaps, surpluses and interdependencies between (and within) areas. The community engaged in defining these gaps, the operational analysts, rightly see it as capability planning, and describe their responsibilities and actions in the terms of service delivery, in the language of their customers (Customer 2), and according to the practices, methods, customs and tools that enable relevant modelling and planning.

Despite Smart Acquisition's downward impetus from need, when translating these service gaps into expressions of product – equipment and other DLoD – a full awareness of the

upward drive of solution feasibility and viability offered by technology and industry suppliers must be continuously observed. DEC's are thus tasked with overseeing a system (a capability) trade-off; in effect, systems engineering at the military capability level. The cognoscenti of technology supply opportunities and limitations that essentially contribute to this trade-off are the IPTs. Their engagement in capability planning is a mutually recognised necessity, bringing sponsor and agent together from the early conception of need.

In this manner a capability gap may translate into diverse, focussed product agreements between DEC's and IPTs.

However, the relationship between the military capabilities employed by the operational users, acquired through their DEC agents, and the equipments supplied by industry can be complex; it is commonly far from a one to one relationship. MOD acknowledges that 'in most cases the military capability sought can only be delivered through the inter-operation of two or more projects, and some projects exist solely to contribute to a system capability.....These interfaces need to be identified, understood and managed' [AMS, 2007].

This unavoidable fragmentation of responsibility for cooperating elements, which must eventually coalesce in operational situations to form a coherent capability, presents each IPT with a continually changing background. IPTs are confronted with a negotiation and trade-off environment in which individually they must manage and direct the evolution of their own equipments or capability elements in a network of shifting linkages and influences. They must constantly be aware to balance the metrics of individual IPT responsibilities and achievement with the often un-stated, and sometime unforeseen criteria of quid pro quo working relationships with their peers.

This natural concurrency of action across projects introduces a fundamental threat; a risk that the effect of one system change will ripple across, impacting functionality and interface definitions in neighbouring systems. New design necessities, options and trade-off decisions in one IPT can force re-designs, compromises and new risk profiles in a web of SoS interactions.

Day-to-day this places many IPTs firmly in the classic 'systems-of-systems' dilemma. The nature of the problem – that no direct relation exist between the structure of elements that comprise operational capability and the organisational responsibility structure for those elements, leading to no overall definition of need, accountability or funding – has been long recognised. The Smart Procurement Final Report Figure 2, titled 'The System of Systems Model', alludes to this challenge of recursive associations that transcribe equipments into a balanced set of complementary capabilities (Figure 7.1 in this text).

Nor is the problem one of acquisition 'command structure' alone, it is compounded by time. Federations that form SoS commonly relate to IPT responsibilities that commence at different starting dates, some of which may even predate the recognition of an SoS relationship. An SoS may thus result in a dynamic programme of projects across the DE&S that comprises contributions from individual IPTs vested with heavy individual responsibilities and significant autonomy. The term system-of-systems might be better caste as confederation-of-systems to convey the essential attribute of executive independence or autonomy balanced by cooperative benefit.

IPT autonomy has been a double-edged sword. MOD's Smart Acquisition drive toward balancing accountability and empowerment of IPTs offer the potential for regions of engineering action that harbour inward focus and inhibit outward collaboration. There is a tendency to compromise interoperability in preference for other, more readily specified and determinable equipment attributes. Unhelpfully, the incentivisation and reward structure is insufficiently sophisticated or oriented towards interoperability challenges.

Project interdependencies can thus profoundly impact the effectiveness of deployed capability. They are mostly the result of sound but focussed design and implementation decisions taken with only a limited view of a wider systems-of-systems (SoS) picture. Self-interest and headless decision making are not major drivers. Customer agreements, lacking foresight, blinkered through a lack of effective communication by overstretched resources, containing weak interface requirements and equally weak acceptance tests, and technically blind to the existence and consequences of aspects of SoS interoperation, are the real sources of the problem.

Actions and mechanisms to reduce the adverse impact of one project on another are an ever present IPT responsibility. Mechanisms to ameliorate these impacts range from dialogue in collaborative communities of interest, through mutual consistency checks in architecture descriptions (through MODAF), to integration plans directed towards exposing inter-IPT dependencies.

In this manner, IPT relationships within MOD are predominantly upward to sponsors and laterally to other IPTs. Although these exchanges observe distinct, even mandated levels of formality and documented agreement, they do not lead to relationships that must stand the independent social and legal scrutiny of government/industry trading conventions. Such conventions, however, strongly govern the IPT downward facing interactions.

The MOD relationship with industry is hugely delicate. Ever-present risks in defence systems and the necessity for even-handedness in government/private sector trading make the IPT's task of working with research, manufacturing and support organisations formidable.

This trading interface has additionally suffered from the dogmas of value-for-money and so-called risk transference. The Smart Procurement Report observed that an "emphasis on passing risk, both commercial and technical, to the contractor has prevented the acquirer and the supplier from working together to manage and control risk".

One of Smart Acquisitions guiding principles therefore was that IPTs must establish a better, more open relationship with industry; one that looks more like a partnership, sharing common challenges, see Figure 7.11. Establishing the DPA was itself intended to symbolically contribute to diminishing the prevailing adversarial relationship between MOD and industry, brushing away any taint of Procurement Executive mind-set, and ushering in a closer 'partnering' and, in as far as practicable, inclusion of industry in project teams.

The relationship with industry has been complicated by reductions in defence equipment budgets and a corresponding downsizing of the supply base. Lost defence technology and manufacturing capability are a threat. A viable, healthy supply base, indigenous wherever feasible and expedient, means that "the prime contractor for the capability and the MOD both have a clear interest in managing the supply chain very closely. This is not only to drive out unnecessary cost, but also to ensure key underlying industrial capabilities are maintained, and sub-system/partial systems engineers share the motivation to improve equipment through-life" [DIS, 2005, p62]

The DIS took this message further by necessitating a structured government/industry relationship and by presenting its proposed strategy for how this may play out in the primary sectors of defence equipment supply. Despite its overt technology messages, the communiqué embedded in DIS is 'mutual dependence and partnership', and IPTs are of necessity in the vanguard of making that happen.

Finally, this message was re-announced in the EAC [EAC, 2006, p18] which calls for "a strong relationship with industry partners to deliver long term value for money based on trust, openness and a clear alignment of incentives".

Smart Acquisition principles of mutual trust, confidence and shared risk identification are based on visibility, accessibility and openness. The message, however, is dependent on the medium. Collaborative working alliances with industry rely on techniques and technologies for project and system life cycle information exchange. Many IPTs have seen the indispensability of SDEs, Shared Data Environments, not just as effective, integrating solutions for system and project information, but as integrators of an extended enterprise that follows common or complementary processes. The SDE is an important technology contributor to the IPT/industry collaborative relationship.

Collaboration and openness present their own problems, legally-binding agreements governing SDE usage not withstanding. Different levels of design formality, with different levels of configuration control, help promote sharing cultures that identify risk and opportunity, but are difficult to manage and maintain. Echelons of control (and access) and liability for the consequences of using design data are not fully resolved. The problem is compounded in international situations where national legal positions of data as an asset, national security policies and mandated technical standards have acted as barriers for IPTs.

Long term ownership of system descriptions and design decisions, and of tool continuity to handle this, must be clear. IPT experience is showing that their system life cycle often exceeds the life cycle of the commercial SDE tools. Transference of SDE responsibility through the life cycle and migration of IT applications has, since Smart Acquisition's inception, presented issues.

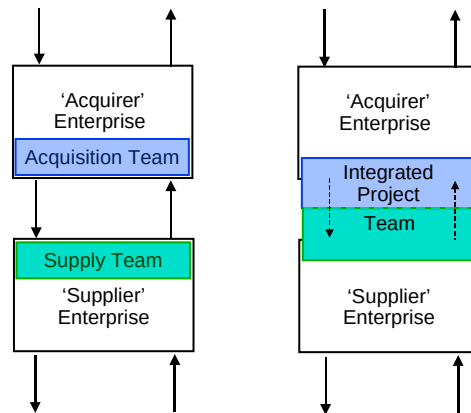


Figure 7.11 Integrating the Interfaces in Supplier Chains: an IPT Role

The greater the difference in the characteristics of the organizations either side of a trading interface, the greater the need for effective intra-organizational integration across the business, cultural and competence dimensions of the boundary. This is hardly more evident than in trading conducted between industry and government. Effective bridging of the MOD/industry trading interface is thus heavily dependent on a sharing of common system-of-interest risks and opportunities according to commonly held systems engineering practices.

7.5.3 Culture

A change in culture - the ideas, beliefs, values and knowledge which constitute a shared team basis for acting – has been paramount in IPT's response to the changed, post-Cold War defence system scene. Being a sociological and psychological construct, culture is perhaps the hardest IPT characteristic that Smart Acquisition sought to modify.

The Comptroller and Auditor General in a 2002 report on the implementation of IPTs, drew attention to the fact that "the introduction of Integrated Project Teams has involved a major change of culture" [NAO, 2002]. This transformation remains a top-level commitment; as MOD's website puts it 'Smart Acquisition is not just about organisation and processes. It means a fundamental change in the behaviour of all those involved in acquiring defence capability' [MOD, 2007d]. This is in keeping with most major business change initiatives

The NAO assessment subsequently underlined this in its analysis of the successful delivery of major defence projects by observing in IPTs that 'successful working relationships are characterised by soft factors such as team working, trust and honesty' in its analysis [NAO, 2005]. The NAO's assessment examined IPT alignment with its 'gold standard' model, the foundation layer of which is "establishing and maintaining the right cultural environment".

Most recently, the importance of culture was further emphasised in the EAC recommendations. Its first recommendation titled 'culture, behaviours, skills and training' [EAC, 2005, Para 5.14] once more gave prominence to the 'soft' factors of business change.

Soft factors that are articulated by MOD include 'an empathy with the customer that supports commitment to providing capability that meets the user's needs, on time and budget', and a motivation to "challenge convention and improve processes rather than hide behind 'the rules' and be satisfied with current performance 'norms'" [MOD, 2007d]. There is no longer an IPT refuge behind traditional government/industry modus operandi regarding diligence, risk and propriety. A different set of responses to the uncertainties and opportunities thrown up by a changing defence scene is expected.

IPT empowerment never equated to unconstrained freedom to act, it was and remains synonymous with responsibility. Smart Acquisition committed to teams having the authority, responsibility, and resources to manage their product acquisition with minimal enterprise interference, and a minimum of mandatory enterprise reviews and approvals. IPTs are conferred with substantial authority to act without recourse to executive endorsement of decisions. Decision making has been driven to the lowest possible level commensurate with risk. The whole team shares responsibility and is held accountable for the results of its efforts.

The hackneyed acronym interpretation of *Independent Project Team* may have reflected a brief, early, euphoric liberation in some minds, but not in those of the IPTs who were held accountable. Empowerment brought obligations to seek, find and apply better ways of working. This required independent approaches to explore options, not least to ways of working more effectively with suppliers. It did not open the door to rampant diversification of localised action.

Ironically IPTs suffered a weakness with empowerment. Decentralisation, a payback of IPT empowerment, was used as a cost reduction exercise to reduce corporate overheads. An ensuring loss of central direction resulted in a confused passing on of new messages. The torch of enlightenment was to be handed on from the initial group of pathfinder IPTs to others in an ordained avalanche change process of 'Breakthrough'. The rapidly constructed training and the self-development exercises in IPTs led to a variety of torches of truth, depending on interpretation and the particular context of the 'mentoring' IPT.

Smart Acquisition's maxim of "the right people at the right place at the right time to make timely team-orient decisions to the benefit of customer" placed a spin on actions and learning that lacked pragmatic, proven devices to support them. DoD's trailblazing IPT venture had not run a sufficient course to indicate what they should be; as a result concurrent experimentation by IPTs played a role in the DPA learning process.

Further, it was incorrectly assumed that the AMS would quickly establish a definitive framework, providing a coherent repository of 'good practice' for all to adopt or tailor as appropriate. It did not. Over a long period it remained a rag bag of immaturely developed and not-communally-agreed-to new approaches, with legacy material in need of update and, in some cases, legacy material that no-one needed. It left IPTs with limited progressive material of value to adopt as they sought to find their place in a new acquisition regime.

A proved, coherent systems engineering discipline had in particular been heralded as a fresh approach to dealing with creative and technical complexities that had long dogged defence procurement. However, Smart Acquisition was ill prepared to support this. It failed to gain appropriate such assistance as existed to annunciate with confidence a coherent, holistic corporate approach to systems engineering, cherry picked new (to MOD) approaches to requirements and otherwise relied on image to bolster the substance that did not exist.

Empowerment also signified effective leadership and the paramount member of any IPT is its Leader. IPT Leaders were delegated with the authority, within the boundaries set by enterprise approvals, to make cost-effective trade-offs between performance, whole-life cost and time. The authority of the team leader within the enterprise and the project team, and the authority this brings to the relationship established with the customer and with industry, are paramount to the projection of Smart Acquisition's new culture.

A high calibre IPTL is an essential dimension of this new strategy and selection on merit by competition of Service, civilian and industry personnel for each post is based on their leadership, as well as their technical understanding. Selection is specifically subject to their being able to apply the behavioural competence of a 'Systems Approach', defined by a 'Through Life, Whole Systems' consideration to asset deployment and decision making, taking into account all DLoDs.

Skill selection and development across the whole project team is now seen as vital. MOD set itself the task of developing a special 'acquisition stream' of personnel: a cadre of both civil and military staff, who would be specifically trained and spend much of their careers in the procurement field. It was acknowledged that career development action was required to retain skilled civil service procurement personnel in the face of demand in the private sector. Weaknesses of systems engineering understanding were recognised early on in Smart Acquisition.

A Skill Development Programme gradually emerged, in which IPT duties were 'punctuated by a series of high impact workshops and forums' to share the growing body of learning and best practice and program of mentoring, with some joint training and interchanges with industry reinforce this. Out of this a set of acquisition-based competences evolved to help in the development of team members and selection of acquisition [MOD, 2007e]. MOD's Acquisition Competency Framework identifies first and in most detail the competency of 'Applying a Through Life, Whole Systems Approach'.

The rubric to this 'Systems Approach' does not, however, mention systems engineering or point to any widely-recognised set of practices that support such an approach. The D&ES Systems Engineering Development Partner action has attempted to redress this by employing the INCOSE UK Chapter Competence Framework [INCOSE, 2007] in the Development Partner scheme.

Perhaps the most demanding cultural change in IPTs has been associated with the development of a new working relationship with industry. A tradition of adversarial posturing has suppressed creativity, trade-offs and optimisation opportunities. Hidden agendas and power struggle have led to locked-in, polarized positions, sometimes with relevant information suppressed or distorted. The outcome was invariably compromise according to a win-lose negotiation.

The failure of this strategy is that it is founded on the assumption of a zero sum game: that one 'player's' benefits are gained at the expense of another's. A resulting competitive struggle between government and industry has swung back and forth, from industry zeniths characterized by the Ferranti's over-costing affair to government having the whip hand in early 1990s defence contract negotiations as the result of radical defence budget reductions and supplier competition.

As Elliott, in a consideration of MOD/industry contracting accountabilities, analysed it, "Organisational fragmentation can lead to competitive targets and win-lose strategies. Public servants thrust into the competitive marketplace tend to over-react. They assume that being commercial means being aggressive and confrontational at all times. After a few years they realise that private industry is much more about cooperation than confrontation – today's competitor may be tomorrow's partner ... and customer and supplier have a shared interest in success." [Elliott, 2005] That realisation is now driving MOD and IPT culture.

The trading reality is that defence procurement, in common with many commercial situations, is a non-zero sum relationship. The 'banker' in the non-zero sum game of defence acquisition is 'systems engineering's' ability to manage system complexity and mitigate technology risks. Both trading parties are making decisions and taking actions according to a single, shared, time, cost and performance uncertainty envelope that depends on the delivered system solution.

In addition, defence procurement is an atypical trading situation, with a security-driven necessity to acquire and a very small number of suppliers having the capability to manage and supply such complexity and technical systems. A mutual dependence is at the heart of the relationship. A collaborative association is vital. This requires a culture that permits freedom to openly explore options, with creativity encouraged. Relevant information from all parties needs be considered and all agendas articulated openly, with mutually agreed trade-offs and optimisation driving the strategic decision making. The resolution sought is a global optimum, win-win outcome.

To achieve this, it is essential that the IPT is sufficiently informed about system trade-offs and systems engineering practice to act as an informed partner in the relationship with industry. The DIS strapline emphasises the point – 'Intelligent customers - intelligent suppliers: the importance of systems engineering' [DIS, 2005, p17]

7.6 Outcomes

This chapter is more than just a scene setter for C21st defence systems acquisition. The nature and pattern of its evolution over 50 years are likely to prognosticate defence acquisition for many years to come; possibly decades of the prevailing 'Anti-terrorist, Anti-insurgent War'. The present drive of successive recent initiatives to place systems

engineering prominently on the change agenda, and the supporting arguments in §8 and §9, forecast that systems engineering is highly likely to be an explicit dimension of military asset acquisition. If its name changes, its principles and ensuing practices will not

This chapter hints at why past 'systems engineering' challenges were met with fluctuating degrees of success in the absence of an evident conceptual or methodological approach that unified contributing technical specialists.

Today, higher expectations, a growing level of systems engineering discipline and fewer uncontained failures appear to be interdependent. Absolute proof remains elusive but MOD has gained sufficient conviction to advocate that systems engineering is essential to defence acquisition.

The systems engineering of the early Cold War, its founding era, is equivalent though clearly distinguishable from C21st systems engineering. Systems engineering is being adapted to meet rapidly changing technological, economic and political demands, and will continue to be subject to change. MOD, and IPTs in particular, have been and will be instrumental in these changes. The views of IPTs that follow carry weight in where and how systems engineering will and must change.

From MOD's side, IPTs are in the front line of this crucial trading interface. Their actions are therefore a major factor in how effectively this acquirer/supplier relationship can be conducted, and they are intrinsically and inescapably dependent on sound, widely-accepted systems engineering practice. The translations of systems engineering principles into IPT practice is a major determinant of effective defence acquisition, with high-return improvement opportunities.

The early prominence originally given to the explicit presentation of systems engineering by Smart Acquisition has lost some impetus. This may trace to a lack of proponents in key role who have sufficient empathy with, and awareness of, its contribution to defence acquisition, and also to a lack of systems engineering identity, coherence and unification in the messages of the AMS.

This chapter has reviewed how Systems engineering's foundational marketplace – military systems – is at the forefront of seeking to harness business improvements from a revitalised and more relevant systems engineering discipline. The goal is a win-win outcome for defence force customers, for capability developers, for equipment developers and for industrial suppliers alike.

'The Smart Acquisition IPT is characterised by its 'cradle to grave' responsibility, its inclusion of all the skills necessary to manage its project, and its effective and empowered leader' [AMS 2007]. This chapter ended with IPT culture: the values, beliefs and opinions of the acquisition community. The values, beliefs and opinions of these empowered leaders' are crucial to the introduction of systems engineering's principles and the application of its practice. The next chapter examines how this appears to those at the heart of MOD's adoption of systems engineering: the IPTs leading Abbey Wood-based IPTs.

8 Gauging the Progress

8.1 Prologue

As Machiavelli stated it, 'there is nothing more difficult to take in hand, more perilous to conduct, or more uncertain in its success, than to take the lead in the introduction of a new order to things' [*Machiavelli, 1515*]. This is what has faced IPTs since the inception of Smart Acquisition, and this chapter explores how systems engineering has played a part in the business transformation that they have been central to.

Disraeli's observation that "change is inevitable, change is constant" conceals the reality that for social systems change is usually continual rather than continuous. Periodically subject to reinvigoration and re-focussing; change follows a pattern of punctuated evolution. Thus it is been, that defence acquisition reform is characterised by continual steps of revitalisation. §7 traced the recent history of this, and its most recent steps – Smart Acquisition, Defence Industrial Strategy, Enabling Acquisition Change – exemplify the situation. Undoubtedly other initiatives will come.

§6 postulated that in the UK defence system acquisition community, partly because of the catalytic effect of these initiatives, the discipline of systems engineering is poised to break its accustomed technical bounds. A new landscape exists in which systems engineering principles can be translated more effectively into business organisation practice. Whether this happens is heavily dependent on the recognition and worth of systems engineering as seen by IPTs.

Operating on the customer side of the key interface between government and industry, IPTs are pivotal in the application of systems engineering principles across MOD. Moreover, as the face of major government investment in industry, they manifestly influence the systems engineering stance taken by UK and, to an extent, overseas defence suppliers. Thus, the advancement and health of systems engineering across MOD, and across its supply base, rests heavily on the understanding, attitudes and motivations of IPTs, and of their leaders the IPTLs.

'The DPA's strategy is to achieve its objectives by means of highly motivated, skilled and empowered Integrated Project Teams using Smart Acquisition methods' [*MOD, 2006*]. They are the engine of systems engineering reinvigoration and influence.

Ramo recognised the dependence of systems engineering on well-balanced, competent teams. 'A good systems-engineering team will include many individual specialists who have learned how to work their areas into sensible interfaces with the contributions of the other specialists. It is the team that must include the total intelligence, background, experience, wisdom, and creative ability to cover all aspects of the problem of applying science and technology, and particularly, who must integrate the overall intelligence ... to reach real-life solutions to real-life problems' [*Ramo, 1998, p.20*].

IPT attitudes towards systems engineering, their support for it, their commitment to its effective practice, will all have a marked influence on systems engineering in MOD and, because of the MOD's major purchasing power, the UK generally. Gauging the prevailing situation and the areas of challenge that IPT leadership sees lying ahead is therefore central to advancing IPT systems engineering practices.

8.2 Synopsis

The business management dimension of this investigation now turns to issues of the effective application of systems engineering in MOD. As seen in §7, this context of application is an outcome of successive acquisition improvement initiatives that resulted in 1998 in the establishment of the IPT structure and culture.

These initiatives have amounted to a policy of commitment to systems engineering as a cardinal contributor to required improvements. They have substantially conditioned the

systems engineering expectations, interpretation, opportunities and constraints now seen in IPTs. This chapter takes stock of the prevailing situation. It presents the approach followed in order to gauge the nature, value and shortcomings of the current status, it consolidates and analyses the result of sampling IPTL views, and it identifies some specific challenges that the foregoing principles bring fresh insight to and that suggest possible advancements of IPT practice.

An important factor in this investigation was to gain dependable indications of the extent to which systems engineering is being practiced in IPTs, the manner of its execution, resulting consequences and tangible benefits. As seen, business metrics have struggle to provide credible or reliable information on such matters [§2.4.3]. Accordingly, within the time and resource limitations of this investigation, an empirical determination was seen as the quickest and most dependable route to understanding the prevailing situation in IPTs

The vehicle for this empirical focus of the research was a survey of DPA-based IPTs undertaken during September, October and November 2006 – seven years on from the systems engineering assertions made by Smart Acquisition and one year on from the publication of DIS. It explored some of the IPT factors likely to have been impacted by their degree of adherence to a systems engineering approach. Specifically, it was a survey of the views of leaders of IPTs, purposely intended to provide a management and business view of systems engineering execution and contributions in IPTs.

The survey was conducted according to three primary lines of enquiry. These broadly equate to the three formulations of systems engineering described in §4 and §6; that is, the prevailing systems engineering business practices (business process), the infrastructure that supports them (capability) and the skills base that enables them (competence).

This chapter presents the survey techniques employed to minimise subjectivity and the practicalities that were observed to avoid bias. The responses gathered were subject to a two stage analysis. The first stage had the goal of eliciting the major subjects, observations and concerns prompted by the questions. These were categorised according to the significant, collective issues that were identified. The second step of the analysis took all the issues elicited and regrouped them into cogent themes that were essentially independent of the originating questions. The resulting emergent themes conveyed, as was hoped, as much about the business view of systems engineering as the engineering management or technical view.

Finally, drawing on the themes and issues identified, observations are made on areas of systems engineering practice that could be positively influenced by a re-interpretation of principles. These areas for advancement, taken as representative rather than exhaustive, then lead into a more detailed consideration in §9 and §10.

8.3 Sampling IPTL Opinions

8.3.1 Shaping the Survey

The line of this investigation moves to determining how effectively Smart Acquisition's systems engineering mantra and DIS's support of it has been translated into good DPA IPT practice, and as a result to identifying areas that continuing improvement actions could be directed towards.

Attempts worldwide to objectively isolate the benefits of systems engineering have underlined the complex relationship between performance/success metrics and the effective employment of systems engineering. The standard business and performance metrics gathered across DPA offer little in the way of acknowledged dependency on systems engineering, and any correlation has to be viewed as indirect. A fully objective assessment of the state of systems engineering across the DPA is a difficult, possibly impossible task.

A subjective assessment, despite inherent weaknesses, is thus the most reliable and quickest way to build a picture of the conduct of systems engineering across DPA IPTs. It also has the

benefit of gauging not only the prevailing situation, but also management projections and expectations.

Informal dialogue with IPT members in 2004 readily evoked a range of views, hypotheses and anecdotal observations on the conduct of systems engineering in IPTs. However, credible as they seemed, the lack of systematic presentation of information or objective supporting evidence diminished the credence in the picture that was forming. In addition, responses appeared too influenced by respondent's position in the responsibility structure of an IPT, on the dependence their role had on systems engineering, and on the most immediate challenges they were facing at the time of enquiry.

The investigation therefore turned to establishing an interview technique that could lead to a more objective survey of IPTs. A considered, strategic picture was required and attention shifted to seeking IPTL views only. This had the goal of gaining a more dispassionate view, and of achieving uniformity in the level of seniority and viewpoint of interviewees.

There is a plethora of interview methods that can underpin surveys, with some approaches better suited to particular situations. A study of interview strategies and techniques led to the selection and customisation of a method evolved by Ratcliffe [Ratcliffe, 2002] based on the work of Douglas [Douglas, 1985]. This had evolved from the development of strategic plans to guide the organic evolution of complex organisations. The approach is designed to establish a uniform context across all respondents by using the same set of questions to draw out anticipated elements of information and also to act as the trigger to develop an understanding of remaining major concerns.

The pre-determined questions are viewed as an active instrument in a 'conversation' with the interviewee. The technique employed is non-directive and avoids leading or biasing respondents. Its goal is to elicit responses through a managed combination of structured and semi-structured enquiry, woven by the interviewer to appear as a seamless thread of "strategic interview and conversation". Combined, these two modes of elicitation can establish a comprehensive and ordered gathering of free-form, 'conversational' responses from a representative sample of the target population.

A framework of common enquiry was established by a fixed set of questions designed to guide responses through the three subject areas that established the scope of the survey. They essentially align with the three related views of systems engineering advanced in §7. They explored:

- the prevailing systems engineering business practices in IPTs, i.e., how, and to what extent, is systems engineering being conducted, explicitly or implicitly, in individual IPTs?
- the infrastructure that supports this, i.e., how relevant and effective is the DPA corporate systems engineering information, guidance and practical support for IPTs?
- the skills base that enables this, i.e., do projects have appropriate skill profiles and supporting skill development/acquisition actions to enable them to undertake systems engineering?

This framed a discourse that, typically with only minimal, open-ended prompting, elicited their opinions, expectations and views on systems engineering, building on their responses to the set questions.

The specific questions are chosen to avoid any direct implication of a suspected trend or issue and they elicit the cardinal facts. They encourage respondents to fill in the intervening opinion space through natural expansion of detail or by triggering related topics. Active development of significant emerging themes or identification of apparent omissions is permitted by interviewers, which they conduct through informed dialogue aimed at enriching but not leading the views expressed. The ratio between eliciting the common, cardinal point information and the development of emerging themes in this technique is intended to be evenly balanced.

The approach was trialled mid 2004 with one IPTL at 1* level. The resulting information elicitation was considered to be effective and, set as a 1 hour interview, observant of the utilisation of a valuable resource: IPTL time. Prevailing circumstances did not favour the

follow up of this pilot IPTL interview until late 2006. When repeated across the sample, the same question set and technique were followed. One outcome of the two year interruption was that most issues raised in the trial interview were repeated in 2006 survey responses.

The practicalities applied to the conduct of the IPTL interviews conformed totally to those recommended in the work of Ratcliffe, with the exception that results were recorded by hand (~12,500 words).

The survey was undertaken according to the activity flow shown in Figure 1. The formulation of questions in the first box took place in the 2004 pilot exercise.

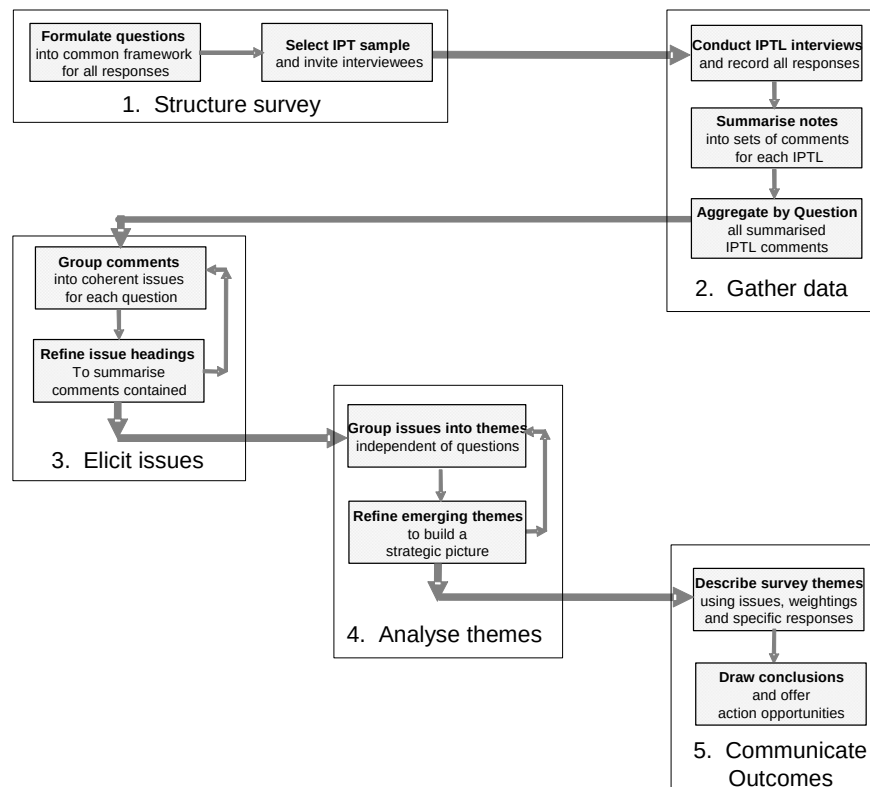


Figure 8.1 Activity Flow in the Survey and Results Analysis of IPTL Views on Systems Engineering in IPTs

8.3.2 The Survey Structure

The same ten questions were used throughout the survey and the rationale behind how each of them contributes to the framework of enquiry is as follows:

1. What do you understand by the terms:
 - a. Systems engineering
 - b. Architecture
 - c. System Integration
 - d. Capability
 - e. Integrated Project Team?

This opening question was designed to be an ice-breaker to establish a style of open opinions. It invites a sequence of descriptions for which there can be no definitive result, just a set of personal views. Its primary purpose was to assess the extent and consistency of systems engineering appreciation across the sample. Its five sub-questions were designed to establish:

- understanding of the subject of the interviews (systems engineering);
- views on a diversely understood and applied aspect of systems engineering (architecture);
- the perception of a traditional MOD interpretation of systems engineering (systems integration) that could indicate how understanding is evolving;
- the interpretation of, and insight into, a dominant influence on systems engineering in the acquisition community (capability);
- the role and dynamics of the organisational element that is the primary environment for conducting systems engineering (the Integrated Project Team).

2. How do you perceive a systems dimension in your project?

This question was designed to gauge the degree to which IPTs were performing systems reasoning and bringing system structure into their day-to-day actions.

3. What relevance does systems engineering have to your project?

This question was intended to establish the level of recognition of systems engineering as an IPT activity and to what degree it is an explicit contributor to team capability.

4. How does your project supply, acquire or contribute to capability?

This question seeks the perception of how and to what extent IPTs are moving from an equipment-centric view of their role in MOD towards dealing with holistic, operational systems that are the source of service.

5. What are the primary mechanisms of your project's interaction with:

- a. Customers
- b. Related projects
- c. Suppliers?

This question explored how the 360° relationships across trading boundaries may be influencing systems engineering approaches to bridging boundaries.

6. How do you see the relationship between project management and systems engineering?

This question helps establish the prominence of systems engineering in the profile of business roles and skills across IPTs

7. What are your views on the CADMID life cycle?

This question looked to how Smart Acquisition's modified life cycle is impacting system life cycle management.

8. What systems engineering guidance and support do you receive from MOD?

This question established the need, implementation approaches and assessed effectiveness of systems engineering information availability, procedures and tools, and related support functions.

9. Do members of your team style themselves as systems engineers or incline themselves towards systems engineering as a career development path?

This questions how IPT members are believed to view systems engineering as a primary or transitional role in their careers.

10. What personal attributes do you think benefit the conduct of systems engineering?

This is the closing, sit-back-and-relax question that opened the opportunity for IPTLs to use their experience to identify what they thought distinguished the systems engineer, and also to gain a view of what they would look for from systems engineering when building a balanced set of team skills.

A covering letter sent to interviewees listed the questions and outlined the form and style the interview would follow.

The size and selection of the composition of the IPTL sample in the survey was, as is commonly the case, a compromise between a reasonable minimum that avoids disruption to the conduct of business and that is labour effective, and the need for a sufficiently large, statistically significant sample that covers a range of anticipated independent variables. IPTL engagement time and business disturbance were primary consideration. A secondary consideration, as experienced by Ratcliffe, was the recognition that in this 'open response' interview approach the results analysis "can be a very time-consuming process".

The technique chosen holds that "it is generally found that between 12 and 20 [interviews] normally suffice". The actual sample was determined by the practicalities of IPTL availability (in two cases last-minute pressures forced a deputy) and the importance of achieving a representative mix of independent variables in the population. Ratcliffe's technique recommends the inclusion of "at least one or two" interviewees who link creatively and differently to the study. One Support Group Leader and one Executive Director (who had been an IPTL) were selected to introduce this additional insight. Consequently, the minimum recommended valid sample of one trial IPTL, 12 IPTLs and 2 'associated' managerial leaders formed the whole sample from which information was gathered.

Given that there are about 100 IPTs and Support Groups based in the DPA, a 15% 'selectively structured' sample should lead to the emergence of statistically dependable indicators, though the survey strays towards the sampling risks of small populations.

In selecting the sample, the IPT independent variables considered were:

- Strategic position in the hierarchy of system (capability) structure, e.g. major mission capability, enabling infrastructure, platform, sensor, effector;
- size of the IPT, including its relationship with a programme of related IPTs;
- the Service or defence capability sector that constitutes the end user community;
- the predominant phase in the life cycle and, somewhat related, the legacy constraints that exist;
- required levels of interoperability with existing deployments and with evolving solutions;
- the span of responsibility through the life cycle, including relationship or prior combination with logistics and upgrade responsibilities.

Since a sample of IPTs biased towards any of these independent variable would skew the observation, the selection of IPTs invited to participate was, in as far as possible, a managed balance of these factors. The survey population thus depended on deliberate selection, and ultimately on availability and willingness of IPTLs to take part. The interviewees were positive regarding the survey, keen to learn its results and some expressed a wish to be engaged in follow-up actions. The composition of the sample is detailed in Appendix C.

8.3.3 Gathering Data and Eliciting Issues

Interviews were strictly limited to 1 hour in the interests of resource usage and uniformity of approach. Other than in unavoidable instances, all interviews were conducted with a lead interviewer setting the pace, gently assisting exploration of issues according to scope and time, expanding or clarifying as necessary, and hand recording information as fully as these tasks permitted. The second interviewer captured all points made and assisted the lead interviewer as appropriate to explore issues.

The way in which the predetermined, cardinal question was augmented by free form responses was dependent mostly on the personality, fluency and experience of respondents.

With prior knowledge of the questions, respondents were free to prepare or consult as they chose. The framework of common questions established the scope of the subject prior to the interview. In their invitation letter, IPTLs were advised of the desire to achieve a balance between structured question content and their own development of issues. How each respondent approached this very quickly became apparent in the course of an interview and the interviewers reacted accordingly. For the most part, due to the high levels of competence and experience of IPTLs, they needed little prompting to build the balance and the richness of inputs sought.

For each IPTL, the recorded notes from both interviewers were consolidated to provide a single script which typically was in the order of 1000 words. From this, the salient content was summarised to improve visibility and subsequent sorting of issues. The text was modified to remove identity of the IPTL source. Each set of IPTL interview summaries was coded according a different font. This enabled avoidance of later double counting of the same (burning) issue raised by the same IPTL as a result of two different questions (several instances of which occurred). It also provided traceability that enabled resolution of queries and more accurate analysis.

Next, all summarised responses from each IPTL were aggregated by question number. Some points made in summary at the end of the interview were treated as an additional question category.

The first of a two stage analysis was then undertaken. This had the goal of categorising the major issues from all IPTL responses for each question and was performed through repeated inspection and association. It followed a progressive abstraction and refinement of salient issues, which were then given identity through a heading that summarised the common point being raised. Wherever possible, the issue heading drew on the most common terms used by respondents.

This categorisation process was repeatedly cycled for each question. The dual goals were to identify, as far as possible, a set of distinctive, independent issue headings for each question, and at the same time to minimise the number of uncategorised comments. 88% of all responses were categorised in this manner according to 107 issue headings. The remaining 12% were in effect noise. They were either isolated observations or out of scope, and were not used further in the analysis.

The second step of the analysis was to take all issues elicited and regroup them into cogent themes. In effect, this step was in the nature of a synthesis that re-aggregated the categorised issues according key, emergent messages.

This approach to the information elicitation and presentation minimised any question bias that may have presupposed the results. Substantially independent of the initiating questions, it constructed an emerging set of themes. Some of these themes were steadily becoming apparent during the course of the survey, though guidance towards reinforcing them was avoided. The themes identification was repeatedly refined in the analysis. It progressively built towards a strategic picture, traceably constructed from each and every individual response (neglecting the 12% of uncategorised issues). The themes may be interpreted as being answers to questions that were not explicitly asked in order to avoid leading the respondents.

The primary outcome of this survey is a description of its themes. These are expressed in terms of the issues raised, the weighting given to issues by the IPTL sample and, in order to reinforce the analysis, selected exemplification by way of verbatim, though un-attributed responses.

The survey themes flow in the following order:

- The Prevailing Perception of Systems Engineering
- Systems Engineering's Contribution to the IPT
- The IPT's Systems Engineering Capability Profile
- IPTs and the Relationship with Customer 1
- IPT Systems Engineering Relationship with the Supplier Base

- Technical and Program Interactions between IPTs
- Systems Engineering Competencies in IPTs

In an open question survey, due to the continuous, free form nature of responses, any numerical conclusions drawn have to be treated with caution. However, the technique of 'category refinement' that was applied during the analysis does have the effect of quantising the information continuum into discrete issues for the vast majority of responses. Using this analysis strategy, only 12% of the information gathered failed to be usefully categorised. Despite this 12% offering useful information in other contexts, it either lay out of scope or was comprised of isolated views and was discarded. The remaining 88% of all IPTL responses contributed to meaningful categorisations that could be weighed or ranked.

These should be seen as understatement of support for any particular issue. IPTLs could only contribute to a level of agreement on an issue if they raised it of their own volition; they were not led or prompted to make any points. This situation was compounded by the tempo of questions (resulting from a strictly limited 1 hour for each interview). The more pressing an issue that an IPTL raised, the fuller was their presentation of it and the more likely that lesser issues were ignored or overlooked.

The sample size limited identification of relationships between independent variables in the population and the observed variables i.e. the identified issues. The analysis that follows therefore applies to the whole population and avoids conjecture on underlying dependencies buried in the responses.

8.4 Systems Engineering in the IPT

8.4.1 The Prevailing Perception of Systems Engineering

The IPTLs presumptions of the subject and scope of the survey may be gauged from their descriptions of

- what they understand systems engineering to mean,
- how they see systems engineering contributing to the fulfilment of IPT responsibilities,
- what qualities go to making systems engineering an effective business contributor.

Though as a term systems engineering is now widely used in many IPTs, it is still felt by 30% of IPTLs to convey a sufficient diversity of meanings as to leave it open to inappropriate or misleading usage. This stems in part from the way it is seen to combine a diverse mixture of characteristics, including "science, art and skills". Despite this, all the systems engineering descriptions, which were not guided in any way, fell into just 8 categories, with a weighting shown in Figure 9.2. The category titles are immediately recognisable as being indicative of up-to-date, even liberal systems engineering thinking. They were:

- is a disciplined approach;
- spans need to solution;
- applies at multiple levels of system structure;
- applies to capability;
- establishes context dependency;
- structures composition;
- is needs driven;
- builds on technology.

This suggests that there is substantially a common perception of systems engineering across IPTLs, whether or not it is an explicit responsibility in their IPT's structure and roles.

Overall IPTLs see systems engineering as a "disciplined approach" to "delivering solutions" based on engineering and on "a systems way of thinking". This formalism is seen to assist project team members to understand life cycles, and it contributes to management actions throughout the whole life time of a system.

Systems engineering is considered to provide an holistic view that impacts, with virtually equal weighting, the whole life cycle and the whole of a system's structure.

Taking the system's whole life cycle view, systems engineering is seen as a process or set of processes that continuously transform the need for a system to deliver effect into a solution that can be demonstrated to have satisfied that need. It applies to a system's capability to deliver military services and, as expressed by 35%, to each contributing element of a system's capability, as defined by DLoDs. This conveyed an expectation that the disciplined mode of working that systems engineering brings to the equipment element of capability has validity and value in other lines of development – we “need to ensure that systems engineering covers all DLoD”.

Systems engineering was seen to apply at multiple levels of systems structure. It may be applied at successive levels of detail, ranging from strategic capability, through an individual discrete system contribution, to the detailed implementations of industry suppliers.

This multi-scale system approach, from capability effect through to detailed composition, was evident in the equal weighting given by IPTLs to,

- understanding a system's context, including interactions with the surrounding operational systems some of which are outputs of concurrent IPTs,
- the internal interfacing and integration of its components in order for it to optimally work as an assembly.

A structured progression through the life cycle, which comes from a drive down of stakeholders' needs and defined performance criteria, was recognised to be matched by the contribution of a managed, upward drive from technology. This suggests that the dominant, even dogmatic, top-down, requirements-constrained, sequential defence image of systems engineering is in fact moderated by pragmatic reconciliation with an upward-influenced, technological practicability.

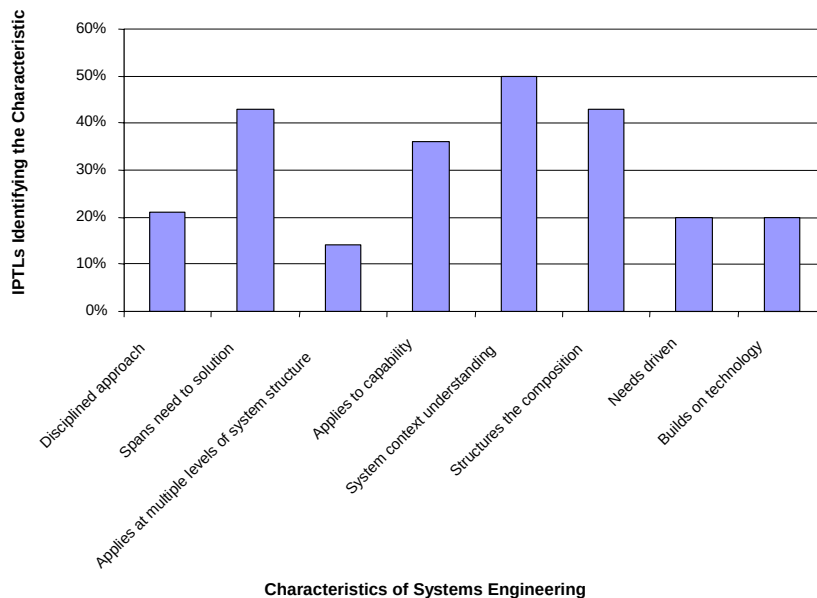


Figure 9.2 IPTL's Description of Systems Engineering Characteristics

It is reassuring that the aggregated IPTL understanding of systems engineering accords well with a scope and nature that most systems engineering professionals would identify with. Taking all the headings in Figure 1 and combining them:

Systems engineering is a “disciplined approach” “applied to [system] capability” that “spans need to solution” and “applies at multiple levels of system structure”. It is a “needs driven”

approach that establishes “understanding of the system’s context” and “builds on technology” to “structure its composition”.

This may not satisfy all as a description, but as a composite from the IPTL sample it catches the tenor of systems engineering in the defence domain.

Overall, the management perception of bringing discipline to multiple levels of effect and detail, throughout a system’s life cycle, conveyed a mature appreciation of systems engineering that aligns well with views of experienced systems engineering practitioners. It may be evidence that IPTLs have not reached their levels of responsibility without absorbing a significant level of systems engineering appreciation.

This suggests a favourable environment for wider and more effective application of systems engineering in IPTs. However this is moderated by the 30% that remain of view that systems engineering as a term carries baggage, is a “bad name” and outmoded. It is conveyed as not accessible and being difficult, even “threatening”, and needs marketing and re-branding, for example as “capability engineering”.

Systems engineering’s scope may also be inferred from IPTL views on what is conveyed by the term systems integration. Prior to Smart Acquisition the two terms were freely used as synonyms, ‘Systems Integration is another term sometimes used synonymously with Systems Engineering’ [*DIS, 2005*, p59, Note 1] and a decade ago systems integration was possibly a more familiar and even a more acceptable defence term. Now, systems integration is generally seen as a subset, albeit a major subset, of systems engineering.

Systems integration was described as the progressive bringing together of a working whole, according to definitions of integrated structure and interface specifications that are outcomes of system design. It includes testing and trials that demonstrate correct product working and appropriate delivered effect.

As with systems engineering, systems integration is associated with structural detail from NEC downward. 20% saw the outcome of systems integration as demonstration that need has been met, with one view that it “delivers the output of systems engineering”.

Notably, 60% stressed the creative nature of systems integration. Although the preceding, creative design action may appear complete, there is still a need for “experimentation, changes”, “adaptations”, optimisations and “adjustments”. These activities characterise the continuing problem-solving content of integration. This creative challenge is seen as compounded by programmatic factors that arise from actual rather than planned availability of solution elements.

8.4.2 Systems Engineering’s Contribution to the IPT

For 20% of IPTLs a systems-based approach was specifically seen as a fundamental or dominant success factor for their IPT. The balance of their attention was biased 3:2 toward the challenges of external system environment, rather than the internal system composition or the inter-working of an individual IPTs deliverables. This may reflect their lower control on external factors and the need to apply discipline in order to understand them.

A prominent concern was interfaces at the boundary of their system of responsibility and this prompted a stress on a good contextual perspective. It underlined the benefits of looking “outside the box when exploring solutions”.

Regarding internal composition of their systems, just 15% expressed that it was related to a people component within the boundary of their system-of-interest. One definition of capability – “joining up people and equipment; concept, plus physical, plus morale, plus training to give force level elements of readiness” – revealed a some appreciation of the human factor in systems. Nevertheless, the generally low level of identification with the human component is entirely consistent with the survey by the Human Factors Integration Defence Technology Centre published in April 2007 [*DTC, 2007*]

Whether inside or outside the system of responsibility, formalism and the management of detail were associated with systems engineering. From the technical management perspective they contribute to route maps and associated checks of progress, and for the

engineering they assist the handling of technical information. 30% of responses referred to the benefit of methods and tools to achieve this.

The major system-oriented concern of IPTs (57%) was a continuing shift from involvement in equipment toward military capability. The degree of implicit, if not actual responsibility for the whole capability is increasing. This requires a bigger system view in order to understand services and their effects across the user community, and to ensure that in-service solutions have planned sustainment.

The IPTL view of the contributions of systems engineering to an IPT's capability profile is partially revealed by the attributes they thought benefited the conduct of systems engineering, i.e., the competence make-up of an able systems engineer. This is a management view, as opposed to a practitioner view, and is noteworthy because it is the perception of MOD's most influential user and deployer of systems engineering.

87% of all the attributes described by IPTLs fell into 11 distinguishable categories. The remainder possessed negligible commonality or distinguishing features and were classified as Miscellaneous Attributes. They constituted run-of-the-mill characteristics that might be expected across a wide range of technical or management responsibilities.

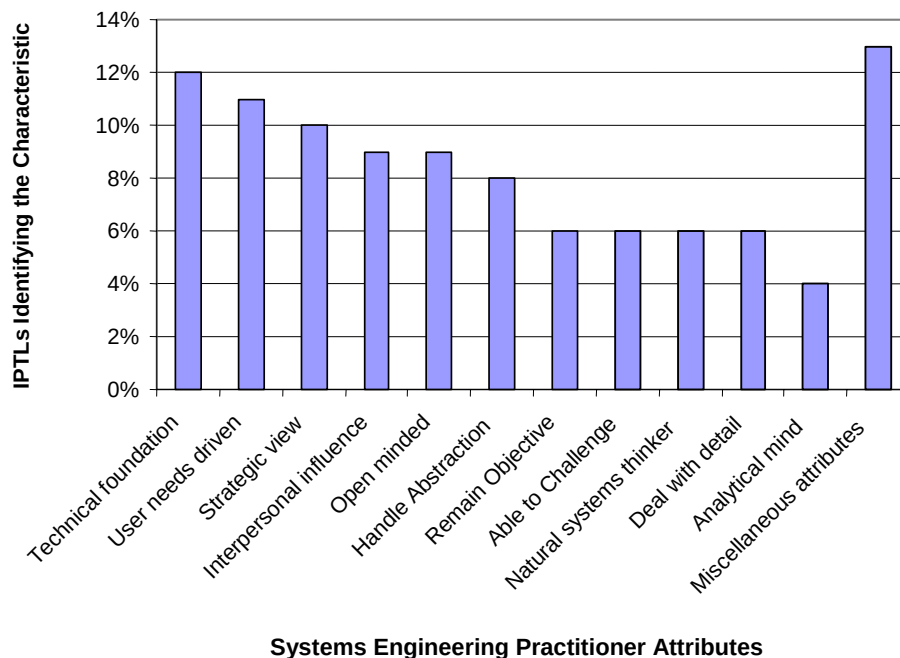


Figure 9.3 IPTL's Views on Personal Attributes that Favour Effective Conduct of Systems Engineering

The unprompted systems engineering attributes described, presented in order of precedence, are:

1. Technical foundation and understanding
2. User needs driven
3. Strategic view
4. Interpersonal influence
5. Open minded
6. Handle Abstraction
7. Remain Objective
8. Able to Challenge
9. Natural systems thinker
10. Deal with detail
11. Analytical mind

With no pre-defined attribute structure offered, this result has the merit of being a conviction based on management experience, delivered with an immediacy of response that favoured primary factors. There is some mutual dependency in these categories, though given the interrelated nature of many human attributes this would probably be a limitation of any proposed categorisation structure.

The profile of attribute categories (Figure 9.3) was substantially linear, with a residual component arising from the Miscellaneous Attributes. The summation of all miscellaneous responses is equal in weight to the principal attribute category (Technical Foundation) and may reasonably be discarded as of secondary importance.

The profile implies that systems engineering conducted within or for an IPT is broad ranging, strategic and cross-cutting in its nature and contribution. It conveys some evidence that the three discipline components advanced in Figure 2.2 underlie in roughly similar proportion the IPTL perception of what constitutes systems engineering:

- Systems Reasoning (33%):
 - Strategic view (10%);
 - Open Minded (9%);
 - Handle abstraction (8%);
 - Natural systems thinker (6%).
- Engineering (33%)
 - Technical foundation and understanding (12%);
 - User needs driven (11%);
 - Deal with detail (6%);
 - Analytical mind (4%).
- Management (21%):
 - Interpersonal influence (9%);
 - Remain objective (6%);
 - Able to challenge (6%).

The balance of 13% was distributed across miscellaneous, general attributes.

8.4.3 Team Capability and Individual Competence

The IPT concept, and for the most part its implementation, was viewed positively. Its integration of the management, commercial, financial and technical functions was associated (35%) with the ability to deliver. Team integration “works particularly well in the smaller IPT”. In the case of larger teams it may be more appropriate to view it as a “joint project office”.

One summary of an IPT’s composition as being “core expertise and project management” conveyed the dominance of the project management function in the role of the IPT. Despite this, 80% of IPTLs saw systems engineering to be a categorical part of the capability profile of their IPT and all of them saw it as having a bearing on the conduct of the team.

Systems engineering is clearly evident in some IPTs through defined lines of reporting structure. In these, specific systems management or systems engineering sections are responsible for planning and dealing with the “cross-cutting” impact of system attributes, including “technology route mapping”, overall testing, safety and security.

There was a roughly even balance in the sample between IPTs that had some form of explicit systems engineering function or section and those who did not. Each IPT’s strategy for this appeared to depend on individual project factors, such as the nature of the systems (e.g. network, platform or sensor/effector), the technology (e.g. its maturity, security level) and the supplier base (e.g. capability/equipment, UK/overseas).

Even where systems engineering was not explicitly conducted, it was perceived to be an embedded part of team capability. 36% equated systems engineering with being a team behaviour; something the “team lives” and a “way of thinking”. It is both engineering process/methods and it is team culture. It entails awareness or appreciation across the whole team, including commercial and financial staff. Systems engineering skills “need to be built into a team” and “woven in as a core element”. In one response, isolating systems engineering into a specific function was expressed as a risk.

30% of IPTs saw systems engineering to be part of project management. However, 64% of IPTs recognised the two roles or functions to be different, though mutually dependent to a large extent. Stovepiping them is wrong and they should be a continuum of complementary actions: “each needs the experience of the other”. This essentially aligns with the concepts conveyed in Figure 6.1.

Project management’s responsibilities are seen as coordination of delivery, value flow and resolving risks, including technical risks. In contrast, the responsibilities of systems engineering are seen to be the delivery of a viable solution. Even in the DPA’s project management dominated environment, IPTs considered project management to be dependent on the “results of systems engineering”, including its contribution to the definition of work packages in the project plan. There was a view that “vanilla flavoured” project management, UK Project Managers’ Association style, without the compensation of systems engineering to address the technical complexity of DPA projects, was a weakness of the skill profile.

Systems engineering is considered to be a “hybrid engineering discipline” that encompasses all IPT technical aspect and roles, across which it “needs to be *primus inter pares*”. This offers an apposite description of the concept conveyed in Figure 2.5. Systems engineering is seen to provide the technical overview that coordinates cross-cutting issues between projects and suppliers. For the IPT it is “essential to have systems engineering expertise with [problem] domain knowledge and knowledge of the systems we work with”.

There was assertion of project management’s precedence over systems engineering within the organisational responsibility structure. If the emotive observation that “DPA sees project management as the core competence and the engineering is a side show” has the ring of truth, it is little more than a reflection of DPA’s business role in the wider MOD enterprise.

29% saw the Requirements Managers as closely related to a systems engineering function and performing, or acting as a focus for, the IPT’s day-to-day systems engineering.

25% of IPTs had non-MOD staff “on the floor plate” and this aligns with the ideal cross-functional view of Figure 8.7. Prime suppliers (the producers) necessarily conduct aspects of systems engineering, such as application domain-based architectural design, supplier-specific technology integration into the whole system and international security constraints.

However, 75% of IPTs do not comprise members of industry. For those in the Concept and Assessment phase of the life cycle an industrial presence “gets in the way”, bringing supplier selection influences. Though joint MOD/industry teams could contribute to a closer and more open relationship with industry, “governance needs to be sorted” before “IPT integration with industry would bring expertise back that has been given away”. Easier, more downstream integration with industry is undertaken in 15% of cases through the presence of IPT members in prime supplier teams.

20% of IPTs stated that their SRDs were substantially developed by industry, though far more are likely to be employing this strategy.

There was a distinct mood in many IPTs that “systems engineering is put out to industry” and “something Dstl/QinetiQ/industry does.” This probably arises from a combination of the dominant project management role of an IPT, the headcount/budget constraints that force skills choices, and threshold criteria that impede establishing/retaining a critical mass of IPT systems engineering skills.

Views such as “systems engineering intelligence is gained from industry sending in systems engineers to solve problems”, “systems engineering [is] presently in industry partners”, “from the lead supplier company” and “[we] depend on SE from engineers outside the IPT”, convey an acceptance, if not a willingness, to confer systems engineering responsibility on others.

This attitude may also be a reflection of the view (35%) that MOD staff with explicit systems engineering skills are difficult to attract, reward and retain. 20% cited salary as an obstacle to establishing permanent systems engineering skills within the IPT: “[Lord] Drayson would have to sanction payments”.

Tellingly, the observation that “DPA acts as a management entity and does not produce products, thus systems engineering [in the DPA] is conducted differently to industry and product suppliers” underlines that systems engineering is applied with different emphasis depending on level/span of responsibility. IPTs sit ever closer to service delivery considerations and industry has its roots in technological solutions and their fabrication. Whatever each of these two parties may wish, this is a reality of their respective responsibilities in the defence trading hierarchy.

Where the acquisition model favours bought in skills, “the tendency is to contract out for systems engineering skills” rather than gain them from the prime supplier/direct supply chain. This affords a measure of business independence through the selection of a ‘customer friend’, in the form either of embedded services - Dstl and QinetiQ were referenced - or contracting out to external specialist systems engineering services. In the case of “safety and security responsibilities [that] equate generally to systems engineering, this is considered specialist support”.

Most IPTs did not believe IPT members “styled themselves as systems engineers or thought in those terms”. Most “would not see systems engineering as a career anchor”. “Getting from C1 to B2 is not seen as a result of following a systems engineering career path”.

The view that “Smart Acquisition’s drive for systems engineering was not followed up by incentivisation to make it happen” applies to careers as well as projects. Project management is the valued career path. Not surprisingly, therefore, many “systems engineering staff see themselves as project managers”.

This emphasis is reinforced through training. “DPA puts a lot of training on project management and should match this with training on systems engineering”. 45% of IPTs were clear about “a lack of training and civil service career management for systems engineering”. “Systems engineering training should be trained at a strategic level”.

The training need is being compounded by the drive towards capability thinking and the system skills required to enable this – “IPTs will need to build systems engineering skills as they transform from equipment to capability”. This problem stretches beyond the DPA. The observation that “we have to educate our own staff and the DEC’s to the systems engineering approach” underlines that systems engineering up-skilling is required on both sides of an IPT/DEC relationship as it evolves to tackle capability acquisition challenges.

The balance of views on the systems engineering guidance provided by MOD was heavily towards there being insufficient (80%). “You can’t find it. There is precious little information – it is supposed to be there, but isn’t”. Even in the 20% who saw there to be sufficient, it was not in adequately “digestible steps”. The limited guidance in the AMS lives in a chaotic archive that is “not easy to navigate or search” and “not totally up-to-date”. 45% were concerned with the degree of prescriptiveness of information and, interestingly, this concern was seen roughly equally as too much and too little. This dichotomy may arise from the stated problem that it is not possible to resolve whether information “is mandatory, advisory and anecdotal”.

“Systems engineering top tips and formal courses” are important, but careers and skills require more than training. “Training only scratches the surface” and a balanced staff development approach that draws on such contributions as “Learning from Experience workshops” and periods in Support Groups to “broaden the experience before going back into an IPT after 2-3 years as part of their development” is needed.

Recognising attained systems engineering competence through “some form of professional systems engineering position” was a view countered by “not anxious for systems engineering licensing”. Greater reflection on the corporate value and the individual desire for systems engineering professional recognition may be a necessary prelude before this is resolved.

8.5 The IPT's Systems Engineering Relationships

8.5.1 Working with the Customer

The message of capability across the acquisition community has translated variously into IPTs' perceived responsibilities for delivering a system. Their understanding of what is meant by capability was highly consistent, if not stereotyped. "Military effect" or "military task" summarises 100% of the responses, and for 70% this was qualified as being all DLoDs. There was an appreciation that capability is a hierarchically structured concept and that it should be viewed as evolving during the acquisition life cycle.

The formal mechanism of influence - the agreement between sponsor and IPT - gave the impression that one third of IPTs specifically identify with being contracted for the supply of capability by the sponsor, one third do not and one third appear equivocal on this issue. There was ambiguity in almost all of the responses, irrespective of the nature and detail of the agreements that were in place between DEC and IPT. Responses suggested that this arises from a mixture of:

- conflict between MOD's stated ideals of capability decision making and influence, and the actual authority/budget control conferred on DPA/IPTs;
- uniformly or effectively specifying capability and evolving need given prevailing requirements conventions.

The ability to control and deliver the outputs of DLoDs other than equipment is seen to depend on negotiation and influence, not on management authority. Despite this, disciplined, formal monitoring and reporting on other DLoDs is broadly in place, such as through balanced scorecards. Generally, there was IPTL willingness to manage the supply of a larger proportion of the overall make-up of capability and steady moves towards this end were referred to.

The uncertainty over how, or indeed whether, URDs satisfactorily convey military capability need remains a difficulty, despite the IPTs – or at least their Requirements Managers – being the engine of their creation. The contractual basis for supply was described as ranging across Customer/Supplier Agreements, URDs, Single Statements of User Need, Statements of Work, UORs, SRDs, and even a legacy of Staff Requirements.

If and where capability requirements truly exist, they evoked the view that they "are as much art as science". They depend on military judgement/SME involvement that is complemented by, even facilitated through, IPT advice. This advice contributes to the balance of "understanding what is possible, what is wanted, what is affordable and what is achievable". Cost and performance are key factors and "DECs do not have the tools to undertake this".

The progressive resolution of this uncertainty in contracting for capability would appear to lie in how the relationship between IPTs and the DEC's team continues to develop; "capability is in part a business process change". Responding to these business changes, the relationship between IPTs and their customers is interactive and cooperative. They are discovering the rules of a new game: one of customer advisor/friend with a close supportive role to "help set the capability scene and define the future".

In contributing to a realistic translation of military expectations into timely, cost-effective systems, IPTs can appear to be "the DEC's best friend and worst enemy". IPTs described their teams' involvement in bringing better understanding into business planning and capability road mapping from the outset, how this could be aided by responsive operational analysis, and how they pursue an on-going "hand-in-glove" relationship with CWGs and CIWGs.

There is unease about how well the CWGs have worked. In particular they are seen as the genesis of a fragmentation that comes from them "working in splendid isolation". With different CWGs exhibiting a "range of different behaviours, even across one DEC", they do not interact uniformly with IPTs. Whilst over 70% of IPTs cited CWGs as important, their failings evoked in one instance the view that they are stopgaps that are little more than "talking shops". CIWGs came over as commanding more respect, perhaps because they

offer practical routes to managing capability delivery. Sponsor fragmentation was further seen across Desk Officers who also lacked a “joined up” approach.

Routine customer reporting to DECs, CIWGs and partly to Customer 2 uses tools such as scorecards, progression matrix and roadmaps, and these include interactions with non-equipment DLoDs.

8.5.2 Interfacing with the Supplier Base

This very early life cycle attention to capability definition with the Sponsor community, coupled with headcount/budget constraints on IPTLs, is leaving some IPTs exposed as not being “intelligent customers” (36% explicitly stated so, though it was an undercurrent in the responses of others). This exposure, given the context of the interviews, relates to the IPT emphasis on project management to achieve a delivery to time and cost “of something”, compared with systems engineering to enable a holistic system that realistically satisfies Sponsor/Customer 2 needs

Despite the project management domination in IPTs, the construction and overseeing of a good commercial package with industry, one capable of giving visibility, assessment and control of suppliers, was seen to “need a high degree intelligence in the systems engineering role”. It is “embarrassing when suppliers’ project managers have better systems engineering understanding [than the IPT]”. However, more pragmatically there is the distinct sense of threat that an IPT can have “the wool pulled over its eyes”. “Industry recognises the IPT systems engineering weakness”, and there is a dangerous “opportunity to be taken for a ride” and to find the “tail wagging the dog”.

Taking into account that the ultimate “integration risk is held by the IPT, not the prime despite PFI” there is a clear need for continued development of a partnership with industry that counters weakness in “underpinning systems engineering knowledge and skills”. As IPTs’ roles change according to the drive towards capability acquisition, and with headcounts remaining constrained by overall MOD business strategy and budgetary controls, IPTLs are looking to adjust their relationships with suppliers. New rules of the game apply to the customer and supplier interfaces alike.

The DIS talks about partnering, and IPTLs are prepared to go there - “an MoD/Industry alliance with sharing of risk and an integrated risk budget. This is where we should be moving to”. However, their vision of the potential advantages of a relationship with greater ‘openness’ between IPTs and industry depends on cultural issues, and on building complex checks and balances to establish mutual, beneficial safeguards, including avoiding any “conspiracy of optimism established between DECs and industry”. Reference was made to existing enabling mechanisms that advance a culture of openness, such as shared data environments and the deployment of IPT staff in industry teams.

The acquisition models described ranged “from buying elements of the solution”, through complete equipment contracting, adapting supplier solutions and, supplied solutions with support services, to “a commission for the supply of capability”. Although the contractual relationship with industry still largely focuses on equipment, as it is on the IPT’s customer side, this is moving towards acquisition of capability. Service acquisitions, linked to PFI contracting, are being employed with the benefit of “paying for what works not what is delivered” and leasing solutions from which “the user will then spin out their need” by gaining hands-on understanding.

Whether IPTs “hand over to a prime systems integrator” or are “very much into service acquisition”, the need remains “to be able to influence and maintain discipline in a contract of what is delivered and received”. MOD needs to be “clear about need, acceptance arrangements and control arrangements”. The left side of the systems engineering ‘V’ model opens with DEC’s stated capability need, the right side finishes with IPT-ratified solution acceptance; the two interact and are controlled through continual steps of IPT monitoring and assurance. This is conducted through frequent IPTL interactions with key suppliers at a range of levels of supplier seniority, with the system goal being “to acquire an optimal whole from industry”.

Beyond the contractual relationship with prime contractors, 35% of IPTLs described their actions to craft future systems as relating to management of the supply base. It is an IPTL

contribution to “maintaining a viable industrial capability”. This is conducted by IPTs balancing the experiences suppliers offer, incentivising them to cooperate and “acting as a ‘dating agency’ to get suppliers to understand what they could/should do”.

8.5.3 Technical and Program Interactions between IPTs

“IPTs in DPA are still very stovepiped” and “boundaries around projects can be defensive”. Striving to unify fragmentation, 95 % of IPTs referred to bi-lateral agreements with other IPTs. Mostly in the form of Service Level Agreements or System Integration Agreements with key projects, they address key influences between systems that will ultimately lead to interactions in the user’s operational environment. Despite this, there were concerns about “no good formal mechanism for interacting with other projects”, and even when systems’ interoperability was technically feasible this could be “frustrated by the programmatic and commercial issues”. Predecessor projects are “not contracted to and do not change to fit subsequent projects”, and small projects have to “fit with larger projects”. In this regard, the standardisation influences of system architectures were viewed positively.

Internal Business Agreements exist with Support Groups and comparable arrangements define external interaction elsewhere in MOD, though this was not described as an influence outside the equipment DLoD.

Common interest groups have formed, but are formally “not defined, funded or articulated” and do not include all interested parties. Cluster arrangements are “just a talking shop and do not focus on project issues”.

IPTs generally appear to structure equipment “to minimise the number of interactions with other projects”, nevertheless things still “go wrong at the mundane level with interface relationships”. Commonly applied management of systems’ architectures and disciplined descriptions of them are gaining a wider, if cautious following.

67% identified architecture with structure, with 50% referring to product structure and 17% to project structure. 50% of IPTs equated architecture to a high level abstraction of system information that conveys an holistic view, such as NEC, and its translation to lower level detail.

35% advanced the point that architecture relates and communicates systems views and concerns as seen by different parties, and 45% that MODAF provides formal rules and techniques that can link the interests of different parties.

45% saw architecture to be a system attribute with distinguishable properties. The characteristics exhibited by architecture’s pattern of system structure, e.g. openness, or by its interaction strategy, e.g. service orientation, were seen to strongly influence the technical effectiveness and de-risking of solutions. Accordingly, there was recognition (35%) that within a wider set of systems engineering competencies, there are particular skills that contribute to working with issues of system architecture.

There is an appreciation (30%) that, being an attribute of any system, architecture descriptions apply to a business organisation when they are themselves considered as the system-of-interest. In this way, analysing and synthesising business services according to systems engineering discipline provides enterprise architectures that structure understanding of, for example, “Concept of Operations” and “multi-Service information flow”.

Despite reservations, there is 100% support for CADMID; 15% expressed this in terms of CADMIT. Although for some it is only “useful to a point” and its “practice is less good than the theory”, CADMID is seen as a business model that “gives a defined project life cycle”. It is “good for major capital equipment projects with long life acquisitions”, is “easy and digestible”. Somewhat less praiseworthy, it was also seen to be “as good as any other [life cycle]” and “a mediocre process that is well documented is better than a great process that is less formal”.

On balance CADMID “does not handle incremental delivery well”. Difficulties of increments of funding release mean it “does not handle....big, complex acquisitions” and its approach is “weak with a host of linked CADMID cycles”. Concerns were voiced that “it takes too long and is too bureaucratic”, and its “heavy systems engineering processes need to be light and agile”.

De-risking through investment of “15% pre-Main Gate is not being followed”. “In the early Concept Stage, systems engineering relates to vision” and should ask “is your journey really necessary” or viable? Too frequently “the C phase [of CADMID] is the ‘Conspiracy’ of optimism”.

8.6 Observations

The extent of IPTL recognition of the role and the contribution of systems engineering to the capability of IPTs, whether explicit or a part of implicit behaviours, suggests that it is accepted as a distinct part of an IPTL's competence profile. Given the widely-held view of a synergistic relationship between project management and systems engineering, excellence in defence acquisition project management must be seen as dependent on a degree of sound systems engineering knowledge and experience. Systems engineering competence may thus merit specific attention in the selection and development of senior appointees.

The strong emphasis rightly given to project management development in IPTs needs be accompanied by a proportionate and well-judged measure of systems engineering development. This is a message that derives not only from those IPTs whose acquisition strategy depends on explicit systems engineering lines of responsibility but also from the IPTs who see a culture of systems engineering inherent in any IPT. The extent to which IPTs are expected to conduct systems engineering as a matter of culture suggests that systems engineering skills development is as much a team capability development issue as it is a matter for individual practitioner or manager development.

There is a close relationship between IPTs and DEC teams as they together develop an understanding of required capability and then translate it into viable, cost-effective defence acquisitions. There is an evident role for systems engineering as a contributor to this mutually dependent relationship. However, with their separate lines of organisation authority there remains a potential for disconnects in systems engineering practices, behaviours and training in these two communities. The momentum of systems engineering skills development presently building in the DE&S should be extended into, or as a minimum reciprocated by, the ECC.

The dominance of and value placed on project management in IPTs suggest that, for the majority of team members, systems engineering will have most relevance and be received best when it is communicated and applied in a manner fully consistent with prevailing project management practice and training. This relates not simply to the ‘awareness level’ but to a level of proficiency that enables projects to act as systems engineering-able customers. A high-proficiency technology perspective of systems engineering, essential for some, may remain a minority approach, with budget and headcount controls increasing the possibility of a growing dependence on the external provision of such skills. Striking an appropriate balance between the systems reasoning, engineering and management that was evident in the IPTL perception of systems engineering, will be an influential profiling issue.

The steady shift towards capability acquisition across both trading interfaces - with DECs and with industrial suppliers - is alerting IPTs to the relevance of systems engineering to the conception, expression and management of capability through-life. Adaptations of systems engineering procedures, methods and even tools that address the challenges of equipment development and integration may merit consideration for tackling and harmonising aspects of other DLoDs.

The systems engineering public relations issue – the perennial disaffection with systems engineering's image – still lingers in the minds of some IPTs. The DIS is quite clear about systems engineering's identity, purpose and value in the defence acquisition community. This is a message that should be unequivocally conveyed, and may need to be explicitly addressed in the course of awareness, training, staff reviews and policy development in order to assist the requisite culture change.

8.7 Outcomes

The series of interviews described in this chapter assessed the systems engineering attitudes, actions and expectations of IPTs, and did so with an acceptable level of objectivity. It offered a project management and, to an extent, a business management view of systems engineering in the 'DPA constituency' of DE&S. Given the resource and disturbance constraints that accompany any research survey, the approach taken was felt to distil the views of a sufficiently representative (size and independent project variables) sample of the group that has the single most influence on how systems engineering has and could yet change the acquisition scene in MOD.

It would be a concern if the results had revealed radical or alarmist surprises: for the most part it provided the confirmation of assumptions that had motivated and justified this scheme of investigation in the first place. It was nonetheless a necessary, and in just a few regards unexpectedly revealing, exploration of an important dimension of UK defence acquisition. The candidness sensed in the responses and the profoundness of the concerns shared gives confidence that this may represent perhaps the best snapshot to date of systems engineering practice in IPTs.

Many of the concerns raised relate to systems engineering as it applies across the wider MOD. They are complex and far-reaching, and, since they are evident from other business perspectives, their existence has been detected and their treatment has been approached from other viewpoints. The systems engineering perspective is, however, able to bring additional appreciation and clarity to a range of issues, not all of them technical, presently faced by IPTs. Seen through the lens of a better-understood, codified and 'battle hardened' set of systems engineering principles, some of their causes are more clearly revealed. From standpoint the route to fresh remedies is more evident.

Thus, in the light of the re-assessment of systems engineering theory and principles undertaken in this investigation, and with a view to possible advancements in IPT systems engineering practice, some selected problem areas have been drawn out of this survey as worthy of deeper consideration. Many could have been chosen, but six have been selected to exemplify and provide a balanced view across the primary regions of systems business enhancement opportunity. Each is seen to be a continuing and pressing challenge in the wake of Smart Acquisition/DIS imperatives; each is selected to illustrate a part of the profile of systems engineering according the ISO model of it [Figure 4.13].

The first issue considered arises from IPTL responses regarding capability. Whether acquisition was seen to be "buying elements of the solution" or was "a commission for the supply of capability" it was often synonymised with capability acquisition/supply, despite interviewees being fully aware of capabilities incorporation of all DLoDs. This may well be a conditioning outcome of Smart Acquisition rhetoric calling for the 'acquisition' of capability and a continual exposure to a fashionable term. The majority of IPTs did, nonetheless, see a shift of involvement from equipment toward holistic military capability and a changing team profile that reflected this. Whilst in the interviews they shared some of what this shift means in the cut and thrust of authority/budget control, and how the IPTs are adapting to exercise new cross-organisational influences, they expressed far less clearly what capability entails from a systems engineering perspective. The ontology of capability, and how it relates to responsibility structure and agreement processes, is the first of the more detailed considerations of systems engineering practice, beginning in §9.3.

Capability, it was fully recognised, is a through life management matter – this is a maxim of Smart Acquisition. For all IPTs, life cycle management was most immediately expressed in terms of a CADMID (or CADMIT) "project life cycle". But the recognition that it is "useful to a point" and its "practice is less good than the theory", and that it is "a mediocre process that is well documented" suggests that some of the deeper lessons of managing throughout a life cycle have yet to be shared. Ways of enriching a view of life cycle management based on systems engineering principles are thus the subject of §9.4.

The general feeling that the AMS has poor provision of systems engineering guidance to project teams means that planning, conduct and control of systems engineering is substantially down to individual project initiatives. This issue is compounded by the stress on a project management culture in IPTs that completely outranks systems engineering in the

thinking and career paths of most DE&S staff. However, the fact that 30% of IPTLs saw systems engineering to be part of project management suggest that the complementary nature of these disciplines could be a strong card to play in resolving this failing. Since project management is seen to be dependent on the “results of systems engineering”, there is the suggestion of an open door waiting to be entered. In §9.5, the last of the predominantly management issues – planning the management of systems engineering – is analysed and a solution strategy proposed.

The first of the technical issues considered is appropriately requirements [§10.3]. The IPTL uncertainty over how, or if, URDs satisfactorily convey military capability raises the question as to whether after nearly a decade the appreciation and use of URDs, SRDs and other requirement documents conforms to some of the insights that systems engineering brought to this area of engineering and management in the mid 90s. The observation that requirements still reside in a world that is “as much art as science” suggests that the opportunity for improvement still exists. It is suggested in §10.3 that the single biggest misdirection of defence acquisition effort and nugatory expenditure can be traced to aspects of requirements, and that a more rigorous, systems engineering-based understanding could improve many ritualised and inaccurate approaches to requirements that are prevalent across MOD, and for that matter industry.

The conceptual understanding amongst IPTLs regarding system architecture appeared to be at least equal, if not better than many technical specialists. For them architecture is the comprehensible abstraction of a complexity of systems technical detail for which they are responsible. They need to effectively gain and communicate the big picture. Approaching the design of systems using collective architecture descriptions as a systems engineering weapon is starting to become palatable to them. It looks less like a technocrat jungle and more like a risk-reducing way to coordinate within and between systems, that for the most part will be deployed in inter-operated configurations that are fittingly described as ‘network enabled’. The dangerous clash between architecture as viewed from an IT/software implementation technology approach to business infrastructure and a systems engineering approach to military system-of-interest design and use is presently unhelpful, and requires rapid resolution. If ‘collateral damage’ to IPTs from inconsistent, rival technical approach is to be avoided, some of the muddled, even charlatan thinking about architecture and its profound theoretical foundation in systems reasoning needs to be quickly and effectively translated into project team practice. The thinking in §3 is thus extended in §10.4 to provide a framework with clearer rationale and structure in architecture descriptions.

The final issue, considered in §10.5, is chosen more because of its absence in IPTL discussions than its perceived problem level. Though one definition of capability offered was “joining up people and equipment”, for the most part people did not figure in the dialogues about systems. This generally low level of identification of systems with the human component is a concern; it was identified in an almost concomitant set of interviews of IPT staff undertaken by the Human Factors Integration Defence Technology Centre [DTC, 2007]. The attention of early proponents of systems engineering to human and social dimensions of systems, that was substantially a failure, and that was partially though somewhat ineffectively revived by a soft systems movement, remains a dimension of systems to which repeated failures can be traced. How systems engineering could and should view the human element and how this successively compounds, through individual stakeholders to eventually become the social dimension of most system’s environments is the last of the principles-to-practices subjects considered in detail.

This particular sequence of chosen issues follows the course that ISO/IEC 15288 takes as it presents its model of systems engineering. Hence, it is in this order – three issues in §9, three in §10 – that the consideration of how IPTs might better cope with these prevailing problems is now presented. This subject area forms the last part of this investigation, and it completes a picture of how the systems engineering’s principles presented in earlier chapters may be advantageously and usefully be transformed into IPT practice.

9 Advancing Agreement, Enterprise and Project Practice in IPTs

9.1 Prologue

Much remains to be achieved in applying systems engineering effectively in the changing and broad-ranging situation of post-Cold War defence acquisition. As Ramo summarised the situation in 1998, 'we are not yet at the point where a systems approach is being broadly used. ... there is a technological ingredient in most problems, and we are recognizing that if technology can be properly synthesized with considerations of a non-technological nature, with the factors of economics, sociology, and political forces, then we might have a new and superior tool for going after these problems' [*Ramo, 1998*].

There is a clear message; systems engineering is intimately enmeshed with the business and social environment in which it is conducted, and into which it must deliver. It is impoverished if it is seen as a self-standing technical discipline, confined to a purely engineering arena. Rather, it is an essential bridge connecting science and technology to management and commerce. As presented in §4, the international codification of technical processes, expressed in a frame of reference of a business organisation's agreement, enterprise and project structure, is bringing just such relevance and influence to systems engineering.

There is now business recognition of the need for this 'superior tool for going after problems' in defence capability acquisition, and that it must weigh technology opportunities against the 'economics, sociology, and political forces' that govern the role of government in the defence acquisition supply chain. The Defence Industrial Strategy Defence White Paper asserts in its Forward that 'industry will have to reshape itself, to improve productivity and to adjust to lower production levels ... while at the same time retaining the specialist skills and systems engineering capabilities required to manage military capability on a through life basis' [*DIS, 2005, p.2*]. It emphasises that 'new technologies will have less benefit if the knowledge of how they might best be exploited and inserted into existing equipment has been lost. This demands a high level of systems engineering skills, at all levels of the supply chain' [*ibid, p.7*].

At the head of this supply chain sit public servants and IPT members in particular. They are responsible for orchestration of teams in MOD's acquisition stream and cooperation with industrial contributors. The contribution of systems engineering to this strategically important trading arena has fortunately been acknowledged; 'the DIS has been developed using Key Principles....One of these was recognising the importance of systems engineering, to ensure that our equipment is procured and continues to develop through-life as a result of mature discussions between intelligent customers and intelligent suppliers'. [*ibid, p.59*] Systems engineering and the acquirer/supplier relationship are intimately bound up.

With regard to the drumbeat of system-related expositions that has sounded through many disciplines during the last half of C20th, Strogatz hypothesised that 'every decade or so, a grandiose theory comes along, bearing similar aspirations and often brandishing an ominous-sounding C-name. In the 1960 it was cybernetics. In the '70s it was catastrophe theory. Then came chaos theory in the '80s and complexity theory in the '90s' [*Strogatz, 2003*]. For the defence community at least, the system-related C-name of the 00s is unquestionably capability.

The term capability was given prominence in Smart Acquisition with the direction-setting title given to the Equipment Capability Customer. This signified to the defence acquisition/supply community that 'the provision of the equipment capability is only one element therefore of military capability, and the development and operation of military capability is a system in its own right. Those charged with delivering the equipment element have to ensure that it is consistent with what the other Lines of Development can deliver, in a dynamic relationship. This coordination is part of our core business' [*DIS, 2005, p.60*].

Systems engineering is thus integral to this approach: 'the emphasis [in defence acquisition] will increasingly be on through life capability management, developing open architectures that facilitate this and maintaining - and possibly enhancing - the systems engineering

competencies that underpin it' [*ibid*, p.17]. In trading terms this means that 'growing expertise in the combination of systems engineering skills, agility and supply chain management required to deliver through-life capability management gives the UK defence industry a comparative advantage'; [*ibid*, p.6]

The 2004 National Audit Office report on MOD Major Project Reports emphasised an original tenet of Smart Acquisition, that of through-life management. In pointing out that 'the Through-Life Management Plan should bring together key themes of Integrated Project Teams, Systems Engineering and improved commercial practices' [NAO, 2004] it proposed a mechanism for unifying these three cardinal elements of Smart Acquisition. The desirability and viability of an explicit role for systems engineering in TLMPs, however, still awaits attention by MOD.

Building on the model of systems engineering developed in §4 this chapter examines, in order: the supply of military capability; the management of life cycle using through-life systems engineering representations; and, mindful of the NAO's assertion of a systems engineering presence in Through-Life Management Plans, the IPT's planning for through-life systems acquisition.

As individual systems engineering-related initiatives, each of these three areas of advancement is valuable. However, together with the technically-focussed advances outlined in §10, they amount to a systemic change: the establishment of a systems engineering culture. 'Systems engineering capability in the UK ... needs, for implementation, to be underpinned by different behaviours in both us [MOD] and industry... amongst other things, it requires clear leadership and investing in maintaining and refreshing skills and knowledge' [DIS, 2005, p.64]. The need for cognisant, systems-oriented business leadership to embed a systems engineering culture in MOD has been acknowledged. As seen in §8, many IPTs already see systems engineering as a matter of team behaviour; something the "team lives" and a "way of thinking": a composite of engineering process/methods and team culture.

9.2 Synopsis

This chapter in its structure and in its arguments draws from the rising influence of ISO/IEC 15288, and on its integrative line of reasoning with regard to whole life cycle management. The chapter addresses specific issues raised in the course of the IPTL interviews that relate to the first three of the International Standard's process categories: agreement, enterprises and project. This logic is repeated in §10 but according to IPTL issues concerned with the fourth of ISO/IEC's categories: the technical process foundation of systems engineering.

Concerning Ramo's 'new and superior tool', this research advances that a reassessment of systems engineering principles offers fresh insight and can inject greater relevance in the engineering and management practices of IPTs. A point given prominence in the DIS is that this 'tool' equally impacts government and industry. With their respective responsibilities of needs-pull and technology-push, the effect on both these parties is different.

For their part, the IPTs are faced with the need to simultaneously cross two immensely difficult trading/agreement boundaries. Upwards, when dealing with its Customers 1 (sponsors) and 2 (users), there is the major transition of product to service, passing through a transitional state of capability. Downwards is the inter-organisational trading that deals with industrial suppliers, with the attendant obligations of even-handedness, propriety, openness and a host of other government niceties. Ultimately in this regard, 'the Integrated Project Team Leader is MOD's commercial interface with industry' [EAC 2006, p5].

Neither of these IPT interfaces has been fully satisfactorily resolved. Worse, they are subject to change in the face of continuously evolving political commitments, policy statements and organisational responsibility structures. This presents IPTs with daunting challenges. They have to understand their role and conduct their responsibilities in the supply of elements of capability, and also their responsibility for the acquisition of industrial products and the attendant influence this has on the behaviour of the industrial supply base.

It is possible that, without the impact of major shifts in the nature of security threats and spending reductions in real terms, capability would not have emerged as such a formative and

rapid shaper of policy in defence acquisition, and would not have become such an immediate concern of IPTs. The speed of its influences has not, however, been matched by a clarity of conceptual understanding, or by some of the organisational practicalities that might be expected to accompany such a radical shift of approach. In part the blame for this is that capability has proved to be a nebulous notion; in part this is because its make up and the associated MOD responsibility structure contain evident ambiguities. Accordingly, this chapter highlights differences of interpretation of military capability and it attempts to implant systems engineering rigour into its definition and into a matching rationality of through-life organisational responsibilities.

A vulnerability of any readily pronounceable acronym is that it can become a hostage to linguistic fortune. To an extent, this has become a fate of CAMID. Across MOD and its suppliers, it is virtually a synonym for any system life cycle. Whereas CAMID's linear simplicity is attractive to enterprise management, it conceals important defence system life cycle information. It informs strategic decision making yet hides detailed patterns of system life cycle process that are the key to IPT tactical management.

Building from the life cycle considerations in 5, this chapter explores issues of whole life cycle management and how they may be represented in recognisable systems engineering process detail yet remain relevant to enterprise managements control. Attention is first given to better presentation of the service portion of the life cycle. From this the 'infinite' life cycle model offers a view of life cycle continuity that is relevant to incremental delivery and to the evolutionary acquisition of networked systems. It matches the progressive changes of responsibility of an IPT, now emphasised in the DPA and DLO merger to form a single DE&S enterprise having control of a system throughout key periods in its lifetime.

In its consideration of the day-to-day project management of IPTs, this chapter offers a response to perceived systems engineering failures of the AMS, or for that matter of other MOD strategy or procedural directives (including the Acquisition Operating Framework as at January 2008 [AOF, 2008]). Many shortcomings in the unequivocal introduction of project practicalities that arise from the systems engineering policy laid down in Smart Acquisition and DIS do in fact have a focal point for their solution: the TLMP. Notwithstanding, alternative mechanisms that could convey the systems engineering message with immediacy, authority and compatibility are also considered. From this a case is argued for the adaptation of TLMPs to convey disciplined systems engineering across IPTs. This offers a possible route for establishing a systems engineering underpinning for IPT planning, and for the ensuing assessment and control of projects.

9.3 Supplying and Acquiring Systems

9.3.1 The Ontology of Defence Capability

The strongest set of concerns raised in the IPTL survey related to the IPTs' move towards delivery into capability. The acquisition models that were described by IPTs ranged "from buying elements of the solution" to "a commission for the supply of capability". At the time of the interviews, one third of IPTs specifically identified with being contracted for the supply of capability by the sponsor, one third explicitly did not. However, almost 60% saw a continuing shift from involvement in equipment toward military capability.

There was some ambiguity in almost all responses over whether an IPTs was, or even could be contracted to supply capability [§8.5.1]. This appeared to trace to two sources:

- contradictions between the aspirations of acquisition policy and the pragmatism of actual authority/budget controls across MOD;
- the difficulty of unequivocally specifying holistic capability according to prevailing requirements conventions.

Factors relating to the first of these are considered in this section; the issue of requirements for capability is addressed under the technical considerations of §10.3.1.

The profound observation that “capability is in part a business process change” points to its major impact on the business culture and practice changes being faced by IPTs. Smart Acquisition’s drive to a service-oriented culture from a product acquisition regime, the so-called top-down, capability-led approach, has been driven by:

- the shifting national balance of investment away from defence, emphatically requiring defence expenditure to be viewed with greater scrutiny and to be seen in business terms; that is, the delivery of more effective defence services rather than a focus on efficient acquisition of equipment;
- rapid changes in threat, compounded by long creation cycles, making pre-ordained equipment solutions anachronistic by their in-service dates; capability is seen to be a service envelope that can accommodate some measure of the unknown and the changing;
- downsizing of the whole defence organisation, which acts to counter fragmentation of responsibility that has in the past distanced equipment acquisition from in-service and support responsibilities.

As the IPTs’ role shifts according to these drivers towards capability, IPTs increasingly “help set the capability scene and define the future” by their understanding of business planning and capability road mapping. They actively contribute to the tasks of capability-based planning and balancing of investment across elements of required capability. As “the DEC’s best friend and worst enemy” they furnish both design and technology opportunities, balanced by the injection of the constraining realism of viability and risks.

From the outset of Smart Acquisition, SDR enunciated in Figure 7.1 that systems engineering was a contributor to the ‘capability level’, but it fell short of clearly explaining the composition of that ‘level’ or how systems engineering contributed to it. Capability was conveyed as a purposeful, integrated whole of diverse constituents, of which equipment was by implication one. It was viewed as sitting above the ‘project level’ which then, and substantially now still is, equated to equipment. From April 2007 IPTs have sat in the Defence Equipment and Support domain of MOD, and this drives home a reality in the remits, responsibilities and resource boundaries in the overall provision of holistic capability. Certainly the viewpoint in the Capability Managers Handbook is quite clear; ‘the DE&S role is to acquire, deliver, support and sustain materiel’ [CMH, 2007, p.42].

A decade on from SDR and capability has become a key part of today’s defence language, yet whilst the term is routinely used, it is also routinely misused and even abused. Linked with existing terminology and concepts, it can bring fashion with no extra insight. It is also applied in markedly different ways across and between organisations and across cooperating defence nations. The ‘capability’ word has joined long-standing systems engineering hot spots of diverse interpretation.

The term capability has been stereotypically implanted in the MOD psyche and this was reflected in IPTL responses. Linguistically, it is now familiar, comfortable, progressive and de rigueur, and with good reason. Capability encompasses key concepts necessary for the effective creation and utilisation of systems, especially in a defence context. Nonetheless, it has taken successive refinements to reach its present definition and this has evolved more from MOD’s structure of budget holding as it has from a disciplined approach to systems reasoning and engineering.

At the commencement of Smart Acquisition, ‘capability’ was treated somewhat ambivalently. In the Acquisition Handbook of April 1999 it was seen as ‘an operational outcome or effect that users of equipment need to achieve’. Nevertheless, despite the prominence given to the term, Smart Acquisition’s initial bias in the acquisition community remained directed toward acquisition of equipment – ‘to enhance defence capability by acquiring and supporting equipment more effectively in terms of time, cost and performance’ [MOD, 1999a]. Likewise, the equipment emphasis given at that time to systems engineering served to emphasise this; it carried a legacy of the DoD approach to equipment-centric systems engineering. The Smart Acquisition use of ‘equipment capability’ (still evident in the title Equipment Capability Customer) verges on the self-contradictory and underscores the equipment-dominated responsibility hierarchy that existed. However SDR heralded that the ‘customer function

will be more directly accountable for the delivery of defence capability' [SDR, 1998]. It was policy that rested on an indeterminate and to some extent exploratory accountability shift.

By January 2004 the MOD definition of capability had changed to "an operational outcome or effect that users of assets or services need to achieve". Significantly, equipment had changed to assets (implying non-equipment products as well) and services had also been added [MOD, 2004]. Albeit awkwardly, this definition emphasised the shift towards a richer set of elements that also comprised services at a 'capability level'. Perhaps because of its uncomfortable detail, this more enlightened definition was not recognised for its value. It soon returned to the original April 1999 form and it is this present dictum that conditions the uniform, laconic meaning of capability commonly quoted and also offered during the IPTL interviews.

If January 2004's definition saw a step backward it was offset by a propitious AMS definition of capability that offered the technical community a more systems-oriented view – 'as a Systems Engineering Term: The operational need which is satisfied by the deployment of an operational system integrated with other cooperating systems' [AMS, 2004]. At December 2007 this has continued to be the definition and, assuming 'operational system' here refers to equipment, it provides a clearer sense of a 'whole system' or holistic capability as seen from an equipment provider perspective.

The service-definer perspective on the other hand provides an operationally-based or capability planner-based definition. Thus the Capability Management Handbook in 2007 gives the capability management view as being 'the enduring ability to generate a desired operational outcome or effect, and is relative to the threat, physical environment and the contributions of coalition partners, Figure 9.1 [CMH, 2007]. The perspective here is one of service delivery, and capability is seen to comprise operational force elements, rather than elements resulting from separate responsibility lines of development. This highlights the different service thinker and developer/acquirer views of capability.

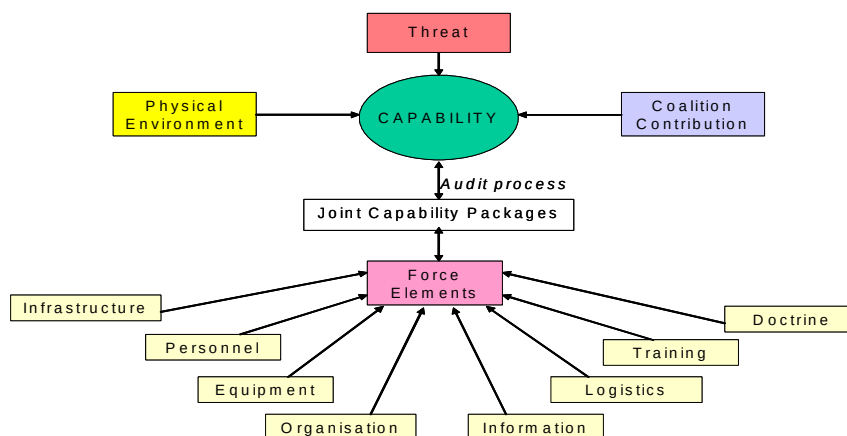


Figure 9.1 Composition and Context of Capability [CMH, 2007]

Figure 9.1 includes the obligatory influence of coalition partners on capability, which is now a precursor to considering the flow of national capability planning in Figure 10.2. Using the language of the capability community, this figure conveys the decomposition and trade-offs conducted in capability space in order to arrive at a level of capability detail that may be meaningfully communicated to capability element developers.

This coalition dimension raises the question as to how MOD's defence partners view capability [Arnold, 2005a]. Inevitably, similar capability considerations and lines of reasoning have been evolving in other nations' defence organisations. In each case they have been flavoured by differences in organisational structures, practices and culture. Three nations are used to exemplify this.

An Australian definition of capability in a defence context is 'the power to achieve a desired operational effect in a nominated environment within a specified time, and to sustain that effect for a designated period [COA, 2002].

A Canadian Force view states defence capability to be a function of the ability of a force to pre-plan a mission and its capacity to do so, seeing this generally to be a function of force structure (organization and equipment) plus training and logistic support [DND, 2002a].

The US DoD sees it as the ability to effectively execute a specified course of military action or tasks, defined by the operational user and expressed in broad operational terms, using a set of synergistic resources [DoD, 2003a].

All of these definitions highlight a range of common features. Capability is interpreted as bringing together of a diverse yet synergistic set of resources or assets to achieve a holistic system. The result of this is not measured simply in terms of compositional integrity, but is judged by the effect of the system's delivered services against pre-planned need in a defined operational environment or against specified operational missions. Further, this service delivery is to be timely, responsive and sustainable.

Fundamentally, capability is a system's inherent ability to deliver required services as and when operated in its working environment. It may be viewed not only as a concern-driven viewpoint of a system but also as a state of a system at a point or period during its life cycle.

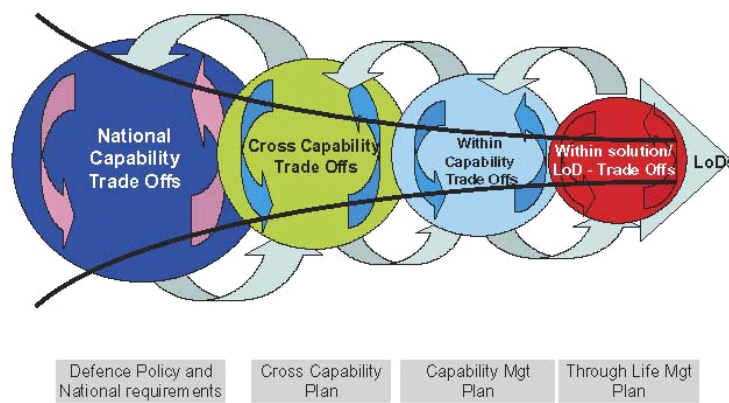


Figure 9.2 The Capability Management View of Capability Planning

Capability is therefore a cardinal system state. The capability term when correctly used is indeed a synonym for system. It is the life cycle transitional state between product and service. How capability is transitioned from constituent elements of product and service is considered next.

9.3.2 Elements of Capability

A more detailed exploration of capability exposes its constituent parts, which here are termed 'capability elements' to align with the use of the term element in the systems engineering International Standard and normalise the international inconsistency in the definitions that follow. In formulating capability, customary systems engineering practice should be applied – concisely bounded composition, completeness of composition, responsibility for the creation of the parts, and (hardest of all) collective responsibility for holistic interaction and trade-off. The delineation and ownership of this structure in defence organisations is key to effective capability, and it is here that diversity appears.

In the UK, MOD initially designated six capability elements [MOD, 2003], termed Lines of Development to signify the parallel paths of separately managed creation across MOD that converge to form coherent capability. These were Concepts and Doctrine, Personnel, Equipment and Technology, Structures and Estates, Sustainability, and Training. Experience with this particular set of capability elements led to the ratification of a more useful set, now termed Defence Lines of Development, and these included Information and Organisation. Sequenced to the acronym, TEPIDOIL, they are shown in Figure 10.1 and listed in Table 10.1.

The UK's coalition partners have their own interpretation. In Australia capability elements are termed Fundamental Inputs to Capability (FICs), [ADFP, 1994]. The FICs comprise eight capability elements that interact to generate holistic capability. They are listed in Table 1.

The Canadian Department of National Defense and the Canadian Forces use the term Functional Components of Capability to describe their six capability elements [DND, 2000b]. These 'building blocks of capability' provide a standard way of planning and developing capability elements. They have been sequenced to spell the acronym PRICIE. Importantly, each element is defined in some detail and explicitly associated with its respective responsible party in the Canadian defence organisation.

For its part, DoD has introduced processes that assess existing and proposed military capability in terms of their contribution to future joint war fighting concepts. Capability proposals must demonstrate that a full range of solution options, addressing all capability elements, have been considered. Seven capability elements have been defined. As a set, they are not identified by a term other than the almost unpronounceable acronym they form – DOTMLPF, see Table 9.1.

A fairly light hand to capability elements has been taken in the US Defense Acquisition System [DoD, 2003b], which still conveys a strong equipment perspective. However Joint Chiefs of Staff Instructions present a more robust treatment of capability, with CJCSI 3170 [DoD, 2003a] predominantly addressing the materiel approach and a complementary Instruction, CJCSI 3180 [DoD, 2002a], giving instruction on the balance of the DOTMLPF options.

As shown in Table 9.1, despite the inconsistent number of contributing elements, some elements align well across the four nations, others are difficult to map.

Australia	Canada	UK	USA
Fundamental Inputs to Capability	Functional Components of Capability	Defence Lines of Development	DOTMLPF
8 capability elements	6 capability elements	8 capability elements	7 capability elements
Organisation	Infrastructure and Organisation	Organisation	Organisation
Personnel	Personnel	Personnel	Personnel
Command and Management	Concepts, Doctrine & Collective Training	Doctrine & Concepts	Doctrine
Collective Training		Training	Training
Major systems	Equipment, Supplies & Services	Equipment	Materiel
Facilities		Infrastructure	Facilities
	Information Technology Infrastructure	Information	
Supplies		Logistics	
Support			
	Research & Development /Operational Research		
			Leadership

Table 9.1 Comparison of Capability Elements across Four National Defence Organisations

With growing experience, each nation has evolved progressively better capability taxonomies to meet their own management needs. Typically, across these nations it has been Army approaches to force formation that have evolved into more generalised Joint Service capability taxonomies.

From this whole system point of view, the capability elements do not equate directly to functional or physical system decomposition, nor even to national domains of organisational structure. For each nation capability is formed from a workable and acceptable set of elements or products that are an accommodation of separate responsibilities. At its heart is the necessity to unambiguously and completely decompose capability such that it can in practice be attributed as a whole across all constituent organisational contributors, throughout the system lifetime. Cynically, it could be seen as a 'rice bowl' decomposition of capability that reveals national defence organisation's power structures rather than objective systems reasoning.

Military capability thus arises from the effective combination of a synergistic, though not unique set of capability elements. Multiple parties across defence organisations have identified diverse lines of organisational responsibility and foci of contributions at different points in the system life cycle, i.e. lines of development, and charge them with cooperatively managing the interactions and trade-offs between these capability elements.

At face value, capability appears to be a conventional systems engineering issue. So, do these multi-national views conform to a common systems engineering model of capability?

ISO/IEC 15288 [ISO, 2002] has entered the defence scene in each of these nations and is, in different degrees, used as a reference model. In the US the International Standard is acknowledged by DoD to provide corporate/enterprise/project level system life cycle process descriptions appropriate for defence acquisition programs, [DoD, 2003]. In the UK, MOD has been using the International Standard's framework to provide high-level structuring in its Acquisition Management System, and to support the interpretation of key acquisition processes. In NATO, it is a basis for defence system life cycle management [NATO, 2004].

During the formative development period of ISO/IEC 15288 (1997-9) the value of visualising a system as capability was not well appreciated by its future users. Thus, whilst the term was used in a concise and consistent way in the process model it was given a distinct back seat, not even making it into the Glossary.

The international standard is commonly interpreted to be a statement about product or equipment. Whilst its presentation style deliberately assists the product creation community, two key messages it contains are that a system is 'composed of hardware, software, humans, processes, procedures, facilities and naturally occurring entities' and that a system 'may be considered as a product or as the services it provides'.

The first of these messages emphasises that a system is created from more than equipment product(s) and is likely to comprise other products, such as trained human operators, statements of doctrine (that operators follow to effect real-time decision making) and information products that hold historical or service delivery data.

The second message is a reminder that the *raison d'être* of products is to be operated to intervene in reality: to deliver service. This may entail not just activated products but also the integrated, dynamic delivery of complementary services in order to comply with agreed levels of service. Elements will typically include logistics to sustain and provision products, collective training to condition the holistic system and the adaptation of a range of commonly available infrastructure services.

This set of quiescent products and dynamic services are deployed and transitioned into a state of readiness to deliver sustainable service. This quiescent service state – capability – is then able to deliver the service as and when needed. In the case of intermittent service delivery, e.g. specific missions, this state may represent major periods of a system's lifetime.

By executing the Transition Process, these constituent elements of holistic capability are integrated, installed, and set ready for operation in the designated operational environment/location.

The process model in ISO/IEC 15288 uses the term capability in accordance with this intermediate, transitional system condition. The life cycle process most closely focused on achieving this transition is termed the Transition Process. Its purpose is 'to establish a capability to provide services specified by stakeholder requirements in the operational environment. This process installs a verified system [of products], together with relevant enabling systems' [ISO, 2002, p.32]. This capability is then poised ready to be activated by the Operation Process to deliver services as and when required.

Figure 9.3 illustrates a typical set of defence products and enabling systems required to establish a capability. It is broadly consistent with Table 10.1. It shows how diverse lines of product development and customised lines of service delivery are brought together ready to be operated.

A system thus may be viewed not only in its states of product and service but also in a key, transitional, intermediate capability state between them. This may be transitory or may account for major quiescent periods of the system's lifetime, for example when the availability of the product and the need for service are not contiguous, or when service is intermittent and

successive quiescent periods exist. This transitional nature of establishing a system capability to deliver sustained service lies at the interface between the concerns/responsibilities of the systems as a product, i.e., the viewpoint of system creator, and as a service during operation, i.e. the viewpoint of the user.

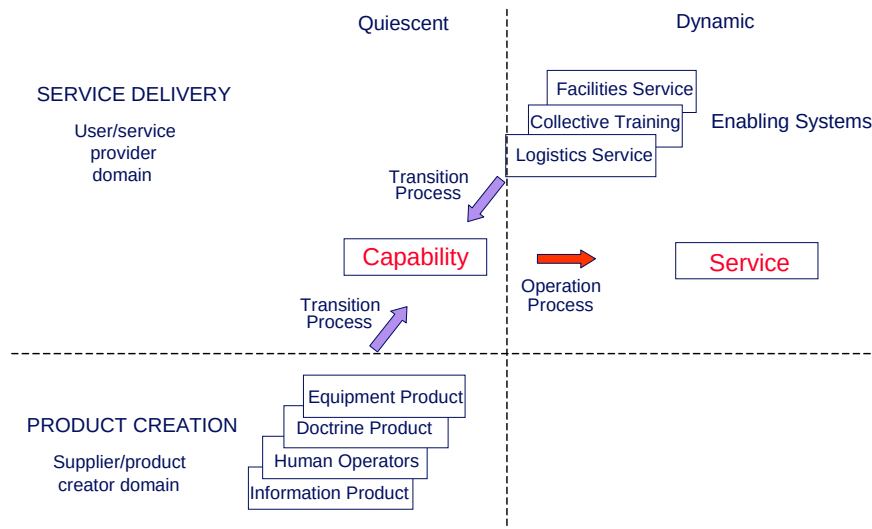


Figure 9.3 Typical System Product/Enabling Systems Transitioned to Capability which when Operated Delivers Service

The responsibility for transfer from product to service is typically aligned with acceptance. In ISO/IEC 15288 acceptance tests, as specified in agreements, are seen to 'define the criteria that demonstrate that the system entity possesses the capability to deliver the required services when installed in its operational location and staffed by operators' [ISO, 2002, p.33].

Accountability for this transitioning of all the elements of capability may fall to either or both product providers or support service providers. It is a matter of precedence, working relationships and budget holding. Of paramount importance, however, it is an issue of clear definition in agreements and responsibility structures.

During the in-service period, 'the purpose of the Maintenance Process is to sustain the capability of the system to provide a service. [It] monitors the system's capability to deliver services, records problems for analysis, takes corrective, adaptive, perfective and preventive actions and confirms restored capability'. [ibid, p.36]

Once transitioned into a contextual situation, a capability can remain essentially quiescent until its utilisation is called for.

In contrast, delivery of service is active, wholly dependent on context and, whilst it may be routine or be individual, pre-planned instances, it is unique in space and time. At any instant it is one of the many possibilities or threads contained within a capability envelope. Tully draws the distinction that 'system behaviour can be considered in two ways: *potential behaviour* is the set of all behaviours of which a system is generically capable, and *actual behaviour* is the set of behaviours actually instantiated in the life history of a system' [Tully, 1993]. This distinction stresses the difference between system as a capability, described as an envelope of designed-for possibilities, and system as specific service(s) intervening in reality. The former is the responsibility of the holistic system creator/sustainer, whomever this is designated to be (SRO, DEC), who sees the potential behaviour of an integrated capability as their goal. The latter is the concern of the system user (the operational military user), who sees actual instances or periods of system behaviour in prevailing circumstances as their ambition.

In the defence domain, capability has conventionally become associated with a strategic level of systems structure. The coming together in Figure 10.3 is typically only considered to occur at high levels of system structure and in close association with operational delivery. Nevertheless, every level of structural detail equates to a system in its own right and, since

capability is a state in the life cycle of any system, *ipso facto* capability applies at any level of structural detail. In the recursive nature of systems, capability is composed of subordinate elements of capability. This notion is implicit in the capability planning graphic of Figure 10.2. It is a matter of focus of interest and responsibility, with some form of through life capability management existing at any level of systems structural hierarchy.

Thus, in the hierarchical structuring in Figure 3.5, the global positioning receiver system provides the capability to deliver location information as required to a navigation system, which in turn provides a capability to deliver navigation information and guidance signals to the aircraft system, which in turn may contribute to an anti-submarine warfare capability. To each owner, whatever the level of detail, the same model applies.

In the hierarchical structuring of ISO/IEC 15288 processes in Figure 4.14, one or more system products (ignoring services for simplicity) are transitioned to capability which is then operated to deliver service. At the highest level of responsibility in this hierarchy, a capability is dominated by the execution of three system processes: Stakeholder Requirements Definition (User Requirement Definition in MOD), Transition and Validation. In the case of defence capability these processes weigh most heavily at higher levels of system structural detail and organisational responsibility.

Thus, despite systems engineering principles stating that capability exists at every level in a hierarchy of structure, the military mind sees it as a high-level system concept. The exact accountability for this high-level capability presents its own challenges, and this is the next consideration.

9.3.3 IPT Accountability for Capability

From all this, it may be seen that the terms 'product', 'service' and 'capability' are system viewpoints and system states that apply during the system life cycle. Depending on the concerns of the observer and the context of use, these terms are applied synonymously for the term 'system'. This somewhat lax inter-changeability also means that 'capability' is a vague and loosely-used synonym for system, product or service. The positive and negative sides of linguistic ambiguity and obfuscation have thus attended 'capabilities' rise to fame, and with it a trail of misunderstandings on the way.

If the term itself can be unclear, so can responsibilities associated with it. This is the issue that underlies this section. In MOD, clarity of accountability for capability has been an evolving situation, governed by legacies of past responsibilities, prevailing power structures and not always well-thought-through policy statements.

Whichever party is responsible for the establishment and sustainment of capability is also responsible for equipment and every other element of capability, e.g. doctrine definition or adaptations to existing general service infrastructures, any of which may have a backdrop of evolving baselines that follow communal needs. Put in these terms, the IPT and specifically the IPTL does not have the authority or the budget to anywhere near fully effect responsibility for capability; nor even does DCDS(EC) or Head of DE&S. Capability is by its nature the result of a federation of semi-autonomously managed organisational elements. It has features in common with the nature of system-of-systems, and these characteristics are evident in the management of capability.

Military capability, as shown in Figure 10.3 arises from the effective combination of a synergistic set of capability elements. Each capability element is developed by managerially, financially and locationally different parts of the organisation, each with responsibility for their own characteristic contribution to defence system service. Each is thus the result of a distinctly different defence organisation line of development – hence the DLoD acronym.

Multiple parties across defence organisations not only have diverse organisational lines of responsibility but also different foci of contributions at different points in the system life cycle. All this must be cooperatively managed for the interactions and trade-offs between these capability elements to be effective.

The capability boundary is seen from two sides: the capability-based planners (in effect the service based planners) and the product creators (systems engineering teams in IPTs and prime suppliers). The responsibility and information interface at this boundary continues to be

a dislocational weakness in MOD's organisational structure and its technical action. It is a responsibility interface that was not strengthened at the outset of Smart Acquisition by greater physical separation between ECC and DPA staff. Despite modern communications, distance still does not lend coherence.

The term 'capability engineering' suggested as a re-branding by one IPTL, may emphasise the service perspective of defence systems engineering, but whether it could help to advance systems engineering or would be an addition to a raft of already imprecise uses of the term capability is presently an open issue, see [IET, 2006]. However, Figure 10.3 conveys that capability is a concept that can be defined in strict agreement with well-accepted principles and models of systems engineering. If systems engineering can rigorously address capability does this mean the IPTL view that we "need to ensure that systems engineering covers all DLoD" [§8] is a viable approach to capability management, and that models, methods and even tools already proved in the equipment DLoD extend across the whole capability? The answer would appear to be "no, but".

The AMS enigmatically places an indefinite boundary of responsibility on IPTs with regard to capability, frequently equating capability with equipment, and repeatedly falling back on the equipment capability term [AMS, 2007a].

This dilemma is clarified, if not resolved, in the DIS; 'the provision of the equipment capability is only one element therefore of military capability, and ... military capability is a system in its own right. Those charged with delivering the equipment element have to ensure that it is consistent with what the other Lines of Development can deliver, in a dynamic relationship. This coordination is part of our core business' [DIS, 2005, p.60]. However, 'have to ensure' is neither supported by line authority nor mandatory procedure.

The pragmatic response to this is a set of cooperative, dynamic working relationships amounting to a shared responsibility across DLoDs, leading to decision making, trade-offs and outcomes that are collaborative, recorded and auditable. From this the IPT's ability to control the outputs of DLoDs other than equipment is dependent on rational argument, negotiation and influence, not on management authority. The objectivity of systems engineering provides the soundest foundation for negotiating such outcomes. "Ensure that systems engineering covers all DLoD" was the most categorical expression of a more general implication: that adaptations of systems engineering discipline may have merit for managing aspects of other DLoDs in order to achieve well-integrated capability.

The IPTL interviews conveyed that a degree of disciplined, formal monitoring and reporting across DLoDs has been in place for some time. The use of balanced scorecards, progression matrix and roadmaps that include IPT interactions with non-equipment DLoDs is evidence of this. Clearly IPT membership from other 'lines of responsibility' also helps, with Requirement Managers linking to doctrine and Customer 2 needs, and logistics support Service staff drawing on their homeland of logistics and training.

So, whilst an IPT may not have authority over all capability elements, it does bear responsibility for the successful integration of its own Military Capability Element and thus for the formal exchange of information on constraints/opportunities/trade offs that influence the capability elements and interfaces of others. Stating the IPT assumptions regarding the nature of these interfaces is a necessity, and as a minimum should expose issues that are unclear, not agreed to or not being attended to. IPTs outputs are therefore not delivery plans for other responsibility holders but are the design assumptions and decisions that inform the plans and actions of others. In return, IPTs receive inputs that are the constraints from others that, if accepted, lead to a managed trade-off between elements of capability and a optimised overall solutions.

Thus, whereas some unrealistic policy statement may have alluded to IPTs being tasked with the delivery of capability, and given that their budget and line management authority precludes this, the IPT, through the TLMP, does have a responsibility to define:

- where it sees the boundaries between equipment and other DLoDs to lie,
- what the technical information exchanges across them are,
- how the influences and trade-offs across these boundaries should be managed

- who specifically is responsible for them
- how these are reported and can be audited.

The well-integrated capability is the product of a well-integrated pan-MOD structure that is creating it.

Capability thus depends on a culture of liaison and understanding with others. Such cooperation depends on clarity of organisational interfaces; what the organisational boundaries are, with whom, what is the technical boundary, what data is exchanged. The disciplining influence and methods of systems engineering could, with suitable transcription, be ported to others to “cover all DLoD”

The TLMP's are expected to formalise such coverage with respect to other DLoDs through their somewhat minimal Responsibility Matrix. This identifies who has the lead responsibility for each element. However, it falls short of satisfactorily defining the extent of IPT's responsibilities and requires much fuller definition in Section 3 (Strategies) [*TLMP, 2007a*] and Section 4 (Plans and Processes) [*TLMP, 2007b*] of a TLMP.

Level 3 TLMP maturity requires that some accountability for delivery of Military Capability across all LoDs has been defined and agreed [*TLMP, 2007c*]. Advancement to Level 4 requires clear, accountable ownership of the delivery of Military Capability across all LoDs, with critical dependencies and interfaces across all LoDs being actively managed up to the Out-of-Service date [*AMS, 2007b*].

These capability element interfaces should enjoy the sort of systemic and systematic discipline that is applied to achieve equipment interoperability according to systems engineering practices; the aspiration voiced in the IPTL interviews. Providing that this does not unduly stretch into the social system reaches of Figure 3.4, then systems engineering and the provisions of ISO/IEC 15288 appear appropriate to address not just the equipment element of capability but much to do with the non-equipment elements. Different culture, community terminology, methods and tools of other ‘lines of responsibility’ presently frustrate this, but these are not insuperable hurdles. Systems engineering is providing a lingua franca for the engineering implementation specialisms, and its basic models and methods could similarly apply to the communities responsible for the other product and service-based elements of capability

For its part, despite Smart Acquisition, DIS and EAC, the Capability Management Handbook is totally silent about systems engineering [*CMH, 2007*].

9.4 An Enterprise View of Life Cycle

9.4.1 Life Cycles of Life Cycles

In the IPTL interviews there was total support by IPTs for the CADMID life cycle (including the 15% who identified with CADMIT). This was nonetheless tempered with reservations.

“Useful to a point”, “practice is less good than the theory” and “a mediocre process” suggests that, whilst CADMID may serve its strategic purpose of enterprise supervision of projects, it fails to fully support IPTs and the management of them. The IPTL concerns appeared to lay in the growing inappropriateness of the substantially linear acquisition pattern seen in traditional (and now only infrequently called for) single-shot, next-generation equipment acquisitions. “Does not handle incremental delivery well”, “good for major capital equipment projects with long life acquisitions” and “does not handle....big, complex acquisitions” all emphasise the changing methodological nature of how the agents of national defence acquisition will have to approach equipping their armed forces of the future – or even of today. The days of the piecewise-linear view of life cycles may be drawing to a close.

This calls into question the resignation in the observation that CADMID is “as good as any other [life cycle]”. It may not be. Just because leading, if not all defence acquisition organisations have become conditioned to this approach, see Figure 9.3, does not mean that it

is the best future or even present-day model for communicating common purpose to all concerned parties.

The politicoeconomic drivers for Downey, Jordan and Rayner, with one-size-fits-all budget control models in mind, have probably fuelled the major, set-piece linear investment-and-approval regime beyond its most gainful days, and may have bequeathed a legacy that IPTLs and even Chiefs of the Defence Staff alike might rue. Powerful management models tuned to national and international drivers of yesteryear could now have started to outlive their usefulness.

With this in mind, and building from the elemental system life cycle modelling in §4 and the patterns identified in §5, this section re-examines some issues of life cycle reference models. It speculates on other representations that could be tomorrow's CADMID.

§5.4 examined a life cycle classification structure of ascending structural complexity in which each level inherits characteristics from lower order forms. This means that any of these 'basic' life cycles is a top-level view appropriate to the nature of system, e.g. as in Figure 3.4, and the circumstances of acquisition; it is an abstraction from subordinate complexity.

Systems comprise subordinate systems (Figure 4.12). 'Each system has a life cycle' [ISO, 2002, p.56]. Life cycles therefore comprise (subordinate) life cycles. The more complex the internal detail of a system, the more complex its life cycle detail is likely to be (Requisite Variety). The extent to which subordinate detail is evident or influential at the system-of-interest level of consideration is largely a matter of how well the system's architecture encapsulates and decouples elements at progressive levels of structural decomposition. Good architecture affords simple life cycles; simple life cycles demand good architecture.

Other than a trivial, linear life cycle situation, any life cycle therefore contains a multiplicity of individual forms out of which some dominant, top-level pattern will emerge. This life cycle is how the holistic system-of-interest is most easily viewed and managed by enterprise management. Buried within its progressively finer detail architecture, and ultimately in its totality of constituent elements, examples of any through-life representation may be found.

An organic form is less likely to characterise a low-level element but this may nevertheless occur. An inversion in the complexity hierarchies of Figure 3.4 and §5.4 may trigger untoward consequences in the management of the top-level life cycle. It is most evident in elements that are the outcome of fast-moving commercial product situations, where solutions are driven almost chaotically by hard-to-predict market forces. In particular, information technology elements (hardware and/or software) may tend toward these organic life cycle characteristics. The lack of poor predictability regarding rate of change and its direction can introduce low-level disruption with major high-level life cycle consequences. It is in such situations that system architecture encapsulation and element decoupling become important. This strategy has its limitations but standardised interfaces and modular functions offer a secure if constricting line of defence against new nails losing wars.

The designer's task is as far as possible to build a system architecture in which the top-level life cycle is as simple as possible, ideally characterised by a linear life cycle form. If such a goal were not possible, then the linear patterns prevalent in past defence acquisition (and also many other complex technical markets situations) would not have been as effective as they have been. However, such top-level simplicity is of late less than fully satisfactory. As system complexity and risks are driven upward by operational needs and tight budgets, the linear life cycle more frequently fails as a strategic management tool, and it increasingly imposes unwarranted constraints on project teams trying to present simplicity upwards and grapple with complexity downwards.

Since there is no discrete transition between any of the life cycle forms and their attributes, the characteristics of the patterns in §5.4 are perhaps better characterised by a continuum with limiting case definitions, see Table 9.2. By interpolating between the attributes of these limits, i.e. the Sequential Process and the Organic life cycle forms, the properties associated with intermediate life cycle forms may usefully be inferred.

Characteristic	Sequential Process		Organic
Need	Persistent needs, Stable for definable periods	↔	Continuously evolving needs with steps of uncertainty.
Business opportunity	Routine and readily-definable market, often consumer oriented	↔	Disruptive features against a background of changing patterns
System definition	Precedence for stakeholders and requirements	↔	No explicit statement of requirement or purpose
Success criteria	Defined stakeholders, related to stated agreements	↔	Aggregated social with judgement-based or retrospective agreed parameters
Achievement Responsibility	Well correlated with creator/user organisation structure	↔	Diffuse across business markets or sectors of society
Managerial	Clear focus of managerial control or guidance	↔	No identifiable, co-ordinating hand of control
Duration	Short compared with changes of need, implementation technology changes or organisational regimes,	↔	Very long, often a persistent need
Structural resolution	Lowest hierarchical levels of architectural detail	↔	Levels of paramount architectural significance
Complexity	Definable and predictable resulting characteristics with identifiable interactions	↔	Many non deterministic characteristics comprehended through heuristics and recognised patterns
Compliancy	Optimised solution within defined boundary	↔	Driven by net behaviour, overriding sub optimisation
Adaptability	Tuned solutions with limited responsiveness to environment shifts	↔	Intrinsic mechanisms for changing characteristics and self organisation
Implementation	Stable or well-established technology	↔	Background of continuous technology advancement

Table 9.2 Life Cycle Characteristics

9.4.2 Life Cycle Reference Models

The linear life cycle pattern used by many national defence organisations has much to commend it. In stringent management circumstances it conveys an impression of good enterprise management control, the appearance of ready audit of achievement and seemingly good assessment of a nation's return of investment. However, in situations of complex platforms or networked enterprises [Figure 3.4] it can inadvertently cloak risks and inhibit more revealing investment strategies.

The rise of the capability perspective of systems in the defence sector was a portent for a shift from traditional product or equipment focus towards system service delivery and, indivisible from this, whole system, whole of life management: a cardinal message of Smart Acquisition.

Despite this evident shift in life cycle management, the importance of the service view of systems was not wholly embraced by the international community of reviewers as the ISO/IEC 15288 was refined, still less the recognition that capability should be explicitly accommodated in the model. The failure to better emphasise or recognise service now looks to be a primary weakness of the current version of the International Standard. With a surge of standardisation activity in service-based business models, this failure to give due regard to the delivery and sustainment of service in what has become the premier systems engineering model is a major threat to the penetration of systems engineering into areas of the business community. A service to meet a purpose is the ultimate driver, and a product is only created and valued in as far as it serves the end of delivering the required service interventions.

This section therefore re-assesses the International Standard to examine what it currently offers (and for some time to come will offer) to the service view. Although it may exhibit weaknesses the picture is not wholly bleak.

The intention of ISO/IEC 15288 was that it should provide a single set of processes that apply to and unify product and service views. The rationale for this was that each life cycle stage cannot be considered in isolation of the others and thus the two principle technical states of a system are interdependent (a line of reasoning that has implicitly driven architecture descriptions).

Considered from the engineering perspective, system service rests on sustainment of its functional and physical composition, e.g. renewal of outworn, malfunctioning or obsolete components, replenishment of fuel, handling of waste material, perpetuation of competent operators. A support dimension of a life cycle has to address these issues, whether defined separately from utilisation, as in ISO/IEC 15288 stages, or concurrently, as in the in-service period of CADMID.

In the mid 1980s CALS, Computer-Aided Logistics Support, focussed on the issue of 'information regarding both the product and the project, throughout its full life duration' [MOD, 2007a] and provided a through life cycle information management model. It was a defence sector initiative to assist equipment design information feed and respond to planned support actions and resources. It offered its own support-centric life cycle 'stages' which commenced with a 'Define a Military Need' and terminated with 'Dispose'. Re-branded Product Life Cycle Support in the AMS October 2000, it advanced that 'a support engineering activity is conducted, in parallel with, or as part of the product design activity, to develop the strategy, policy and procedures for maintaining the product in service' [MOD, 2007b] Methods and tools to assist logistics remain hugely important, as evidenced by the accumulated life cycle costs estimated by SECO as being acquisition (i.e., everything before operation) 28%, operations 12%, logistics 60%.

Despite the recognised value by IPTLs of the discipline and structure offered by CADMID, it is acknowledged to be a product of classical large, new equipment defence acquisitions. With the emphasis now moving towards incremental life cycles, evolving cycles of existing designs, and networked solutions, will CADMID continue to adequately guide acquisition? The indications are increasingly not.

CADMID is for the equipment approvals community an acquisition control cycle rather than an engineering or system life cycle

The adoption of a single reference life cycle model by defence organisations has been based on:

- a legacy of acquisition of major equipments or platforms;
- the way this hides life cycle complexity when viewed by enterprise management, e.g. by only presenting cardinal approval gates.
- the desire to institutionalise life cycle management across organisations in a readily assimilable way;

The perception of a one-size-fits-all life cycle model 'handed down' from enterprise management is having to change as incremental deliveries, networked capability provision, capability acquisition from industry change the nature of life cycles

CADMID is a pattern that assists in the management of enterprise risk but does little to structure the building of the technical detail of system's life cycle model. As a dogmatic and all-pervading pattern for life cycle models, CADMID has clear dangers. It attempts to address widely different risk profiles and conveys the illusion of a common, strategic risk management model, irrespective of the nature of the system. The DPA is faced with a wide variety of systems in terms of their sector of defence application, level of precedence, life spans, evolution state, complexity, size, unproved novelty, adaptation needs, quantities and deployment conditions. Complexity controlled by simplicity violates a fundamental systems axiom.

§5.4 illustrates that a linear life cycle reference model contributes to all higher order patterns. Forced such linear models into use outside their valid range of applicability and usefulness in Figure 3.4 attracts technical risks. The more strategic the nature of the system being considered, the less suitable a simple linear sequence life cycle model is likely to be to guide the conduct of systems engineering. At high-order system structure, e.g. substantial parts of networked defence capability, a linear model is likely to be least effective.

Despite the fact that ISO/IEC 15288, like Smart Acquisition, exemplifies life cycles in terms of a single, off-the-shelf, staged model, it goes to lengths to stress the concomitant existence of stages. A principal requirement of compliance to this International Standard is that an individual life cycle is defined for each system. ISO/IEC 15288 guides its readers to the existence of 'apparently limitless variety in system life cycles' and 'life cycle forms with

distinctly different characteristics'. It makes clear that "sequential, incremental or evolutionary life cycle forms are frequently used, alternatively, a suitable hybrid of these can be developed ... Stages may overlap and/or iterate, as appropriate for the system-of-interest's scope, magnitude, complexity, changing needs and opportunities".

Its provisions state that 'a life cycle model that is comprised of stages shall be established ... It is assembled as a sequence of stages that may overlap and/or iterate, as appropriate for the system-of-interest's scope, magnitude, complexity, changing needs and opportunities' [ISO,2002, p.39]

With a measure of self-contradiction, Smart Acquisition does specifically promotes incremental life cycles in order to mitigate acquisition risks. It therefore contains by implication its own missive: that reference of CADMID is just that – a reference. From this, a shift to defining more complex life cycle models is natural and inevitable.



DoD Phases 5000.2	UK MOD Smart Acquisition Stages	FMV Product Stages	15288 LIFE CYCLE STAGES	PURPOSE	DECISION GATES
CONCEPT REFINEMENT	CONCEPT	CONCEPT GENERATION	CONCEPT	Identify stakeholders' needs Explore concepts Propose viable solutions	Decision Options: - Execute next stage. - Continue this stage - Go to previous stage - Hold project activity - Terminate project
TECHNOLOGY DEVELOPMENT	ASSESSMENT	CONCEPT EVALUATION	DEVELOPMENT	Refine system requirements Create solution description Build system Verify and validate system	
SYSTEM DEVELOPMENT & DEMONSTRATION	DEMONSTRATION	DEFINITION & DEMONSTRATION		Mass produce system Inspect and test	
PRODUCTION & DEPLOYMENT	MANUFACTURE	ACQUISITION	PRODUCTION	Operate system to satisfy users' needs	
OPERATIONS & SUPPORT	IN-SERVICE	MAINTENANCE	UTILIZATION	Provide sustained system capability	
	DISPOSAL	RETIREMENT	RETIREMENT	Store, archive or dispose the system	

Table 9.3 National Defence System Life Cycle Models and the ISO/IEC 15288 Example

Table 9.3 presents the stages, purpose and decision gates of the ISO/IEC 15288 exemplar model [15288, 2002, p.57]. The figure is extended to show the strong similarity with the reference life cycles of three nations engaged in the acquisition and supply of defence systems. The US DoD life cycle defined in DoD 5000.2 [DoD, 2003] offers no lead on the disposal of systems, otherwise it essentially matches that of the UK. The UK's CADMID model is identical other than in terminology to the FMV model that supports acquisition for the Swedish Defence Force.

With regard to the enterprise decision gate approvals in Table 9.3, whilst IPTs may be held at their current stage, at face value defence life cycles are not normally considered to return to previous stages. This is not so. It is a fully reasonable decision, not indicative of failure but of new situations, technology or solution understanding. Invariably, project name changes grace such decisions. Similarly project termination can be positive and constructive, and be an indicator of progress.

Each of the defence models focus more on the early creation stages and less on the utilisation stages as compared with the ISO model (which is intended to be domain neutral). The US model does draw attention to the different, concurrent roles of operation and support.

Systems engineering applies through life, and important service delivery actions revolve around principles and practice. These are notably:

- introduction and/or service changeover with possible concurrent operation;
- response to modified operational need, with resulting adaptive action to amend services;
- opportunity for conducting in-operation service enhancement.

The need for equipment adaptations or modified operator action may revisit design decisions and/or implementation actions. This may involve changes to operating procedures, human-machine interfaces and operator training, support strategy and information handling.

The identification of problems or deficiencies to restore or remedy failings, and informing appropriate organisations (user, procurer, developer, producer) of the need for corrective action is separate from this. Corrective/sustainment actions of in-service support are fundamentally different in their origins, planning, resourcing, required skills, line management responsibility and (in principle) funding lines. However, because of geographical isolation, the same individuals may be involved in both service adaptation and sustainment roles and the distinction blurs.

Like life cycle processes, stages have object and functional model interpretation, and the apparently enigmatic lack of Utilization and Support Stage concurrency in the ISO/IEC 15288 life cycle exemplar stems from its object-oriented messages and not from a functional time-flow interpretation of the model. However, this distinction between utilisation and support states of a system was largely 'commented-out' of the International Standard's text before publication, and the rationale behind separating utilisation and support was largely obscured.

During the operational deployment of military systems, some threshold of change may occur and Urgent Operational Requirements kick in, and matters return to the system's acquisition management home. Notwithstanding, operation of a system continues to draw on through-life systems engineering. Its conduct should be wholly consistent with the systems engineering principles used throughout the life cycle. The practices, though tuned to meet the overriding demands of effective, successful location/time-dependent interventions, should complement those employed elsewhere in the life cycle. For example, services can evolve and give rise to different product configurations, requiring management of the status and descriptions of the various versions and configurations of the holistic capability and its services, or design decisions may be revisited and changed, needing associated change to recorded architecture rationale.

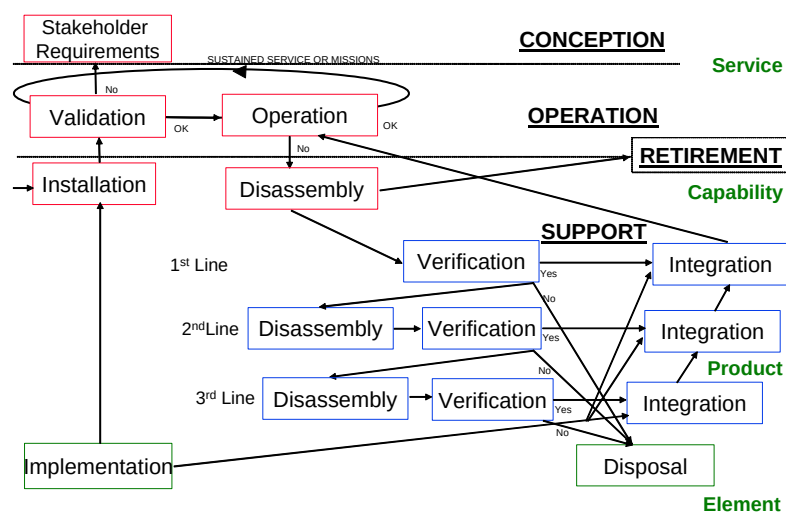


Figure 9.4 An Example of a Utilisation Cycle Model Based on System Life Cycle Processes

Figure 9.4 illustrates how a model built from ISO/IEC 15288 life cycle processes may be assembled to convey a high-level pattern of action according to systems engineering principles. This model is only concerned with repair/preventative servicing but since it employs system life cycle processes it opens consideration of operation and support from a systems engineering standpoint. The continual cycling of operation and validation highlight non-conformance situations and iterations of diagnosis and repair eventually tap into the source of implemented system elements. The familiar hierarchy of service, product and element has capability interposed in it.

With MOD finding itself increasingly contributing to force structures in new theatres of operation, with previously untested deployments, and collation systems present novel and complex systems challenges can be raised during operation. Fresh interoperability situations,

changing threat circumstances and unforeseen interactions and properties all mean that demanding systems engineering has to be conducted in the field. Many demanding systems engineering tasks occur in-service away from the creative homeland of DE&S and industry resources. Systems engineering conducted during system creation and system utilisation need to complement to ensure coherence throughout longer and more changeable lifetimes.

Systems engineering has been traditionally associated with product creation. It needs to move out of this comfort zone to a fuller consideration of life cycle management. This will impact 'DLO IPTs' and how they relate the systems engineering that is conducted in 'DPA IPTs'. The next section looks at how a fuller system life cycle representation could be approached.

9.4.3 Whole-Life Life Cycle

Figure 3.6 presented product and service viewpoints of systems, and this was augmented by a capability viewpoint in §9.3.2. This emphasises the synergistic, through-life approach required in systems engineering practices. These viewpoints now come together in a consideration of a balanced, generalised life cycle representation.

This section considers the general weakness of life cycle process descriptions of the 'service period'. It considers how processes that evolved from models that represent product creation can describe the service viewpoint of systems. It considers how the through-life management role of the IPT links to through life models, explores the systems engineering synergy that exists between the 'DPA' IPTs and 'DLO' IPTs in the DE&S.

Acquisition 'cycles', despite their name, actually track a system's through-life existence and this is by definition a linear passage through time. The term life cycle is borrowed from the necessary repetitive sequence seen in organic systems and it was a natural simile to adopt when many natural and human sciences were subject to common systems attention in the 50s. The analogy to biological cycles was justified in §3.4.2, however even the spiral learning models do not represent a retracing. Nonetheless there is an underlying cyclic nature present in the life cycle of many man-made systems and the reasoning behind this is seen in §5.4.4.

As outlined, there are conflicting requirements to be met by life cycles. From the enterprise view they should provide a sequence of waypoints that set a path of progress and define increments of achievement, accountability and permission to continue. At the same time they should provide sufficient detail to guide engineering and project management through the technical maze of transformations and decision making. Top-level 'architectural' patterns of life cycle are thus massively influential. In the case of defence acquisition, their checks and balances govern major portions of national investment and they condition the operation of large sections of industrial activity. They are dominant and they are pervasive. They have become an almost subliminal backdrop that shapes strategic business and project team decisions and actions. They are also worthy of continual re-appraisal, as presented in §7.3.1.

As it is usually presented, the archetypal image of systems engineering, the V model, only presents the design and build of systems. As a complete life cycle model this is inadequate. Without its extension into the service period it confines systems engineering to a development engineering focus. As a 'traditional' representation of systems engineering this is effective and, for many, wholly sufficient.

In order to address service the delivery nature of system life cycles, the V model requires refinement. This may be achieved by using a further downward 'bend' to represent additional aspects of a system during its utilisation. The resulting shape gives the 'Zorro' model presented in Figure 9.5. This approximates to two overlapping V models in which the first familiar product delivery V model contributes its right side to become the left hand side of a service delivery Δ , with this middle region applying to both creation and utilisation.

The service/product/element hierarchy is, as in the V model, clearly evident. Validation is shown as subsequent to operations. The rationale for this is that in-service validation follows operation (as opposed to the generally limited confirmation of conformant service capability prior to actual operation). The symmetry and logic of this model requires as a minimum a disassembly process. This was rejected in ISO/IEC 15288 as a necessary cardinal process, and from the viewpoint of product this is a reasonable decision. However, the system issue of "can it satisfactorily be taken apart?" in order to diagnose or replace elements is a crucial

design constraint for sustainable services; a constraint comparable with those of integration, verification, transition and validation that in the International standard do impose constraints on the design decisions. In short, a strategy for disassembly to meet through life service considerations is a systems engineering whole life cycle blind spot. The support service regime depends on reasoned approaches to this essential system feature.

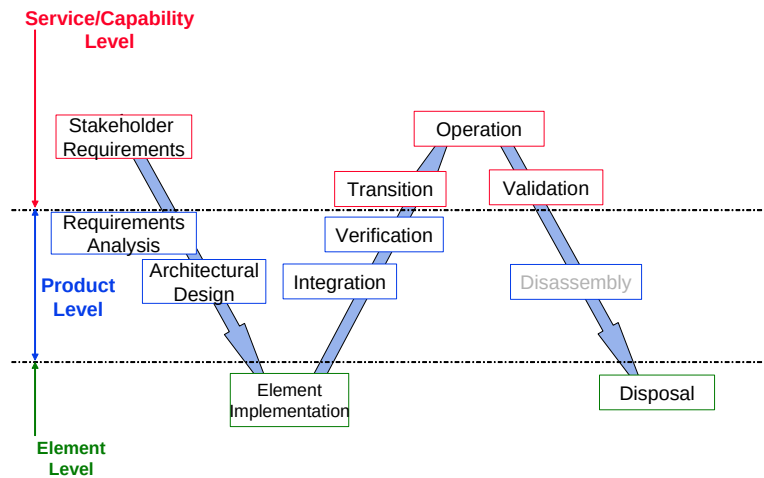


Figure 9.5 An Extension to the 'V' Model of System Creation to Include Utilisation

Accepting the presence of at least a disassembly process, the "Zorro" model traces a path from need to restoration of the environment following service delivery. Some systems, notable examples being those with long-lived consequences such as nuclear power generation, nuclear vessels, toxic waste generating systems, require consideration of disassembly, refurbishment/mid-life upgrade and disposal issues to be their earliest conceptual considerations. Thus systems engineering is incomplete as a discipline, and as a model of human activity (or business) processes, without representations of system-orient service delivery processes. Figure 9.5 is the bare minimum for this and with the exception of the disassembly process, which was a necessity in order to build the model in Figure 9.4, the International Standard is an adequate starting point.

The pattern morphology techniques used in §5.5.1 may be applied to Figure 9.5 to expose underlying cyclic features in the life cycles of more complex systems. The cyclic nature in a creating cycle was presented in Figure 5.8 and by analogy the same morphological transformation can be applied to the Λ of service delivery. This leads to the interlocking twin cycle of Figure 9.6.

Because of its (in principle) ever cycling nature and its shape, it is referred to here as the 'infinity' cycle. It introduces the concept of a life cycle form that can:

- evolve solutions, essentially in the manner of the spiral representation,
- continuously renew or rectify products to sustain service delivery,
- revisit development to enhance or extend the system properties, or take advantage of implementation opportunities,
- increment, or further evolve products so as to repeatedly refine service delivery in the face of changed or continuously situations.

It is a model that applies up to the enterprise level of Figure 3.4, permitting a life cycle management rationale that can respond to the nature of organic environments and in theory at least offering a simplified management model for continuously evolving, 'immortal' systems faced with meeting persistent needs. Though a small step from the ubiquitous V representation it addresses issue that can only be handled by a (minimal) through-life representation.

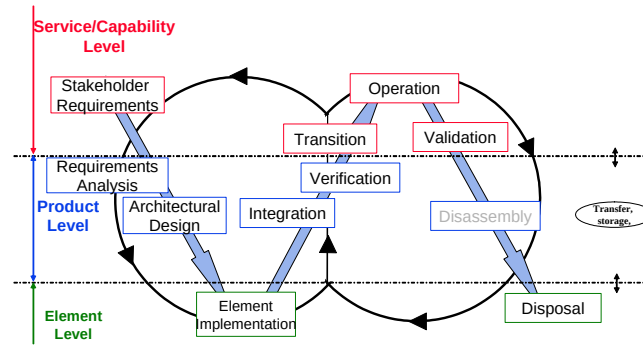


Figure 9.6 The Morphological Relationship Between Figure 9.6 and the 'Infinity' Life Cycle Model of Figure 9.8

The morphology is completed and simplified in Figure 9.7. The service/product/element hierarchy is maintained and the integration-to-transition path spans the aggregation of both product and capability. The unidirectional flow is preserved according to the direction of the arrows and, though individual life cycle processes are each a member of a networked, coherent set in which any one can interact with any other, the dominant flow pattern now traces two contiguous cycles. To the left is a define – design – build – prove cycle, and to the right is a use – sustain – discard – (re)build cycle. Drawing on the biological simile, the infinity life cycle combines a (left hand) bearing cycle and a (right hand) caring cycle. This more closely mimics nature and may thus qualify that bit more as a life cycle model.

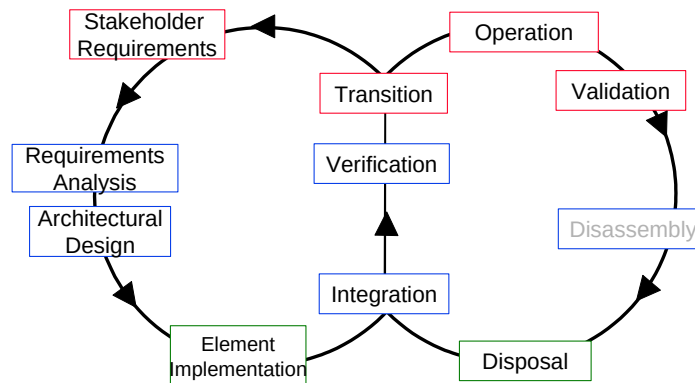


Figure 9.7 The Infinity Life Cycle Model

The infinity life cycle, however, fails the enterprise management criterion. It contains technical processes and systems engineering logic, but falls short of being a strategic model with waypoints that may be selected by enterprise management to oversee progression/funding decisions, as in the column to the right of Table 9.3. This is because it is essentially a process model and not a stage model in which detailed transformation detail has been masked.

By restating Figure 9.7 in a manner similar to the four phase representation of the spiral [Figure 5.4.4] the process detail may be suppressed in favour of phase-based view. The emphasis on the term stage – the outcome of a linear representation's journey-like pick up of permissions, new budgetary provisions, tangible assets and new team members at successive stage posts – gives way to the term phase. This metaphor more faithfully signifies repetition of identical periods in a cyclic pattern.

Starting from the thinking behind the system creation cycle in Figure 5.4.4, its phase titles are modified in order to capture service, capability and product distinctions; in other respects the intent of the titling is unchanged. The four phase representation of this cycle is emphasised by

representing it as a box shape that conveys the notion of discrete changes of focus in life cycle actions. The emphasis of the corner marks exaggerate phase changes and these now become waypoints and may thus be selected to signify enterprise decision gates.

An equivalent phase-related box is formed for the system utilisation cycle. Matching titles encapsulate equivalent life cycle actions and the building or rather re-building of capability is shared.

It illustrates that for “immortal” or more correctly regenerative systems the life cycle can be thought of as comprising two intermeshing cycles; the basis of the “infinity” life cycle model. The resulting model can be transited indefinitely and according to whether origination, incremental extension, mid-life upgrade or whether service, maintenance or service transfer is being considered. Being a more object-oriented model there is a natural expectation that phases will be concurrently active.

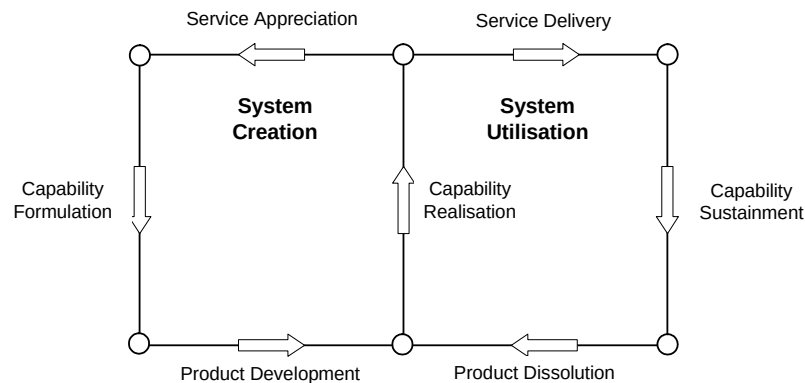


Figure 9.8 The Phases of the Infinity Model

This rectangular framework of phases suppresses any of the detail of bi-directionality. The left hand box is broadly an anti-clockwise cycle of creation, the right a clockwise cycle of utilisation. It illustrates that for “immortal” or more correctly regenerative systems that the life cycle can be thought of as comprising two intermeshing cycles; the basis of the “infinity” life cycle model.

The model has a particular advantage over CADMID for it explicitly addresses capability, as opposed to DLoDs or operational services, directing attention and responsibility towards a capability ‘level’ of action that supersedes a product or equipment ‘level’. An associated benefit is that this exposes assignment of ownership for periods of the life cycle in a way that CADMID (which as a model with an equipment heritage) does not.

As shown in 9.9 the infinity model loosely approximates to CADMID when ‘spelled out’ linearly. However, it is as much an object-oriented representation as a functional description and this complementarity does not make it easy to draw direct equivalence to the purely functional CADMID. Vaguely: Service Appreciation can be compared with Concept; Capability Formulation with Assessment; Product Development with Demonstration, a part of Capability Realisation with Manufacture; Service Delivery plus Capability Sustainment with In-service; and Product Dissolution partially with Disposal. Capability Realisation has no significant equivalent.

Tracing the simplest and most obvious path around the infinity model thus leads to a linear progression that, at face value, has a familiar feel.

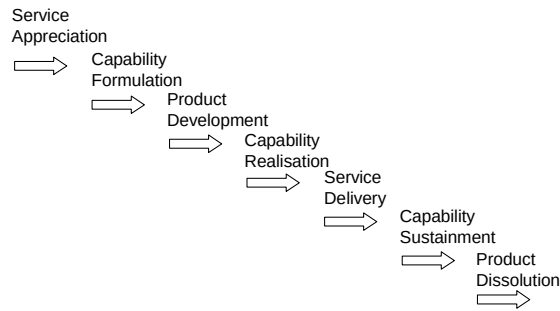


Figure 9.9 The 'Seven Phases' of System

Just as it is argued there are seven distinct ages (characteristic states) in man's life [Shakespeare, 1623] so it can be visualised that there are seven ages (or phases) in a system's life flow. This linkage of life cycle 'ages' with the barely emerged term 'system' was prophetically made by Mede in 1638 when he observed that 'man's life is a system of diverse ages' [Mede, 1641].

The cardinal points of the frame are potential review points and investment confirmation opportunities. None of them quite align with the current Initial or Main Gate criteria and the views of the scrutineers, users and the DEC's may suggest obvious gates and titles.

The two cycles amount to seven phases, with solution realisation serving a common purpose. Expressed in a circular form, the creation cycle appears as an anticlockwise single transit of a spiral, as in Figure 5.8. With the four phases given equal angular weighting, four cardinal work products can be associated with the way points. For the systems creation cycle these are the stakeholder (user) requirements, the system requirements, the realised product and the overall capability able to deliver service. For the system utilisation cycle these last two act as common way points for both cycles; the two remaining ones being non-conformance/non-compliance reports and upgrade/disposal decisions.

Many would see service delivery and capability sustainment to be largely concurrent and thus the same. This is a criticism of the stage model in ISO/IEC. However Figure 9.8 may be interpreted to distinguish systems states and management responsibility for them as much as it represents a chart to navigate a course round the buoys of system life cycle.

Like any system life cycle model it has its limitations and failings. Nonetheless, in one model, Figure 9.8 draws together a number of life cycle model features:

- it is a very simple life cycle model, yet (or because of this) it contains the power to represent and relate different life cycle concerns – it is not remarkable, but the strategically important CADMID is even less so;
- it contains a service, capability, product hierarchy that is layered from top to bottom;
- it stresses the difference between capability formulation and product development, and the responsibility structure for their conduct
- it is explicit about the amalgamation and transitioning of capability elements into capability
- it conceptually morphs into linear, hierarchical and cyclic/spiral representations;
- it contains and meshes a creation and a utilisation cycle;
- it traces and relates the contributions of DE&S IPTs in their contrasting 'DPA' and DLO' modes;
- it is not radically different to the 6 stages of CADMID;
- it employs the term phase, already well established in MOD, using it in a more appropriate way;
- it (positively) does not form a pronounceable acronym that preconditions a one-solution sequence.

With the reduction in major, set piece acquisitions and the move to evolving existing solutions and so a step nearer to organic life cycles, the linear CADMID may not be the most appropriate visualisation for enterprise management or for guiding IPTs. The cynical and doubtless

apocryphal US General's observation that "the only thing that is harder than putting a new idea into the military mind is to get rid of an old one" is as much a statement about cultural change as technical comprehension. It will take much to dislodge CADMID from its dominant role.

It will not be the supremacy of a logically argued replacement that sees the disappearance of CADMID. It will be CADMID's increasing failure to 'fit the bill' that forced change. With evidence of such failure, it is already being customised by CADMID-allegiant adaptations that bend rather than complement its model. Towing the line, these adaptations disguise themselves as being CADMID conformant but little by little they are stretching the principles and the practice consequences of MOD's life cycle model to its limit. CADMID is but one example.

This section has shown that there are other ways of strategically modelling life cycle of defence systems that these may better represent some of the acquisition issues that are rapidly becoming the norm for the DE&S 'equipment IPTs'. Moreover, the *infinit* approach explored conforms to a familiar pattern of action seen in 'support IPTs'. In addition, it offers a model that could be applied across other DLoDs and bring the discipline of systems engineering principles and practice to military practitioners in theatres of operational service delivery.

The foregoing argument demonstrates that it is possible to define a simple but powerful alternative (to CADMID) 'acquisition' models can that, in the heads of enterprise management, enables strategic control and, in the heads of engineering practitioners, permits the purpose and detail of day-to-day of technical project action to be organised.

Use of this or any other model of course rests on customisation; on the outcome of project-specific tuning. To use the language of business process management, a standardised approach needs to be 'tailored' to meet local circumstances. How this tailoring is planned, assessed and controlled is the remit of project management. The instruments that enable these actions are the project plans. How well systems engineering is coherently and categorically presented and practiced in IPTs, and conforms to Smart Acquisition and DIS policy, therefore depends on how effectively individual project policies and plans state them. Here, lacking clear corporate guidance, both commitment and achievement are patchy. The next section considers remedies to introduce greater consistency and quality into this.

9.5 A Systems Engineering Focus for Projects

9.5.1 A DoD Approach

Over 60% of IPTs recognised systems engineering and project management to be mutually dependent; "each needs the experience of the other" as illustrated in Figure 2.10. Moreover, without systems engineering to address the technical complexity of DE&S acquisitions, project management was felt to be weakened.

The corollary of this, advanced in §8.6, is that systems engineering is likely to be seen as most relevant to, and be best received by IPTs when it is communicated and applied in a manner consistent with project management practice of the DE&S. Project management is after all seen as the mainstream role of its DPA constituency.

This section thus presents views developed during this investigation on how a synergistic relationship between project management and systems engineering might best be advanced to the benefit of IPT practice.

Just as with DoD-Std-499, IEEE 1220 has a formalised structure for a Systems Engineering Management Plan, SEMP, and its approach has been widely followed internationally, both in defence and civil applications. Without further qualification, the term SEMP is essentially synonymous with the provisions of IEEE 1220 Appendix B, in which the structure is defined.

The INCOSE Systems Engineering Handbook Version 2.0 was published as in July 2000 as a "How To" Guide for all engineers. It remains the best guidance on how to develop a SEMP according to the structure presented in IEEE 1220.

To counter criticism of the standardisation for non-defence application of systems engineering practice, and thus developers of SEMP for civil product application, the Engineering Industries Alliance developed the 'non-defence' equivalent, EIA-632-1998, titled *Processes for Engineering a System*. The term SEMP was changed in EIA-632 to reflect that it addressed many general engineering issues, and it was termed an "Engineering Plan". This engineering plan provides project personnel and acquirers with a plan of the technical effort that executes the processes 'for engineering a system'. The engineering plan provides:

- an understanding of the problem to be solved;
- what is planned to be accomplished;
- how it will be done;
- who will do it;
- where and when things will be done;
- resources required, including when, how much, and characteristics.

The focus of this plan is on risk reduction. This plan need not be a stand-alone document but can be part of the project plan. EIA-632 specifically acknowledges that in military projects this plan is often called the Systems Engineering Management Plan.

As conveyed in §8, the AMS is felt to have failed to provide satisfactory leadership of a set of systems engineering-pervasive acquisition practices and the necessary culture to complement this (voiced in the survey by 80% of IPTs). This appears from the early style and content of the AOF to be continuing despite DIS, with the exception that requirements management guidance now acknowledges ISO/IEC 15288 as its source model [AOF, 2007a]. The provision of a discrete *Systems Engineering Handbook "Principles, Practices and Techniques"* [AOF, 2007b] echoes the add-on nature of the original *Systems Engineering Guide* in the AMS. This situation leaves individual IPTs who see systems engineering to be a valuable and in some cases essential contributor to their projects to devise local solutions to a long-standing failing.

The ISO model of quality [ISO, 2000], and the SEI [SEI, 2000] and ISO [ISO, 1998] models of organisational capability maturity are based on formalised process enhancement that have existed for years. They address the necessary, explicit actions of an organisation in order to ascend capability steps are thus well known and in many MOD suppliers well established. However, despite the DIS announcing that 'along with industry, we [MOD] need to invest in maintaining and growing high quality systems engineering capability', unlike key parts of UK defence industry, MOD has corporately provided scant follow-up in the form of auditable, embedded, process-based systems engineering capability. Arguing that the AMS and AOF are implicitly 'systems engineering' would fail the first test of scrutiny and be readily exposed as evidence of ill-coordinated and non-institutionalised systems engineering.

Thus, if MOD enterprise management is failing to set the direction, how might a practical remedy be effected quickly? The 2004 National Audit Office report on MOD Major Project Reports offered a highly credible solution. Is it the best approach? The alternatives are considered, starting overseas.

This question had already been first asked and answered by DoD in 1969 in Mil-Std-499 on Engineering Management, and then successively answered in successive standards, see Figure 4.4. By 1974 the answer had been refined in Mil-Std-499A and was to 'provide the program manager...a means for establishing an engineering effort and a Systems Engineering Management Plan (SEMP)'. The solution was to develop an explicit systems engineering-motivated model of project action. 'The plan shall be comprehensive and describe how a fully integrated engineering effort shall be managed and conducted' [USAF, 1974]. Its primary goals were to describe technical responsibilities, reviews and controls; the processes to be used including the procedures to implement it; and "the integration and coordination...of engineering specialty areas' akin to Figure 2.5.

For the last three decades the influence of Mil-Std-499 and its IEEE 1220 successors have accustomed projects and acquirers to the systems engineering guidance contained in the SEMP. Progressive refinements of the SEMP led to its most recent definition in IEEE Std 1220-2005, Appendix B, in which a template detailing structure and content is provided. Its mode of application advances that 'typically, a project has a project-specific engineering plan

for each stage...[which] is a living document and needs to be structured to allow ease of updating of changes and progress through a stage of the life cycle' [IEEE, 1998].

The pressure of 5000.1 and 5000.2 for systems engineering across DoD derives its impact not from detailed direction by the enterprise level of management, as in the case of FMV, but from their seeding of a solution at the project management level.

Systems engineering revitalisation in DoD was focussed on three key areas: technical planning, technical leadership and technical execution. This initiative was partly predicated on an NDIA Study on systems engineering issues in 2003 [NDIA, 2003] that concluded the top five concerns were:

- a lack of awareness of the importance, value, timing, accountability, and organisational structure of SE on programs;
- adequate, qualified resources are generally not available within government and industry for allocation on major programs;
- insufficient SE tools and environments to effectively execute SE on programs;
- poor initial program formulation;
- requirements definition, development, and management is not applied consistently and effectively.

In a re-branding, re-balancing and re-vitalisation of SEMP, a DoD guidance memorandum advised that that a 'SEP [Systems Engineering Plan] will address the integration of the technical aspects of the program with the overall program planning, systems engineering activities, and execution tracking to ... describe how systems engineering activities will be integrated within and coordinated across IPTs'. [DoD, 2006]

The SEP thus became the 'blueprint for the conduct, management, and control of the technical aspects of an acquisition program from concept to disposal, i.e., how the systems engineering process is applied and tailored to meet each acquisition phase objectives'.

In 2004, it was mandated that all programmes apply a 'robust systems engineering approach' to be documented in a Systems Engineering Plan (SEP). 'All programs ... regardless of acquisition category, shall apply a robust SE approach that balances total system performance and total ownership costs ... Programs shall develop a Systems Engineering Plan (SEP) for Milestone Decision Authority (MDA) ...[which] shall describe the program's overall technical approach, including processes, resources, metrics, and applicable performance incentives. It shall also detail the timing, conduct, and success criteria of technical reviews'. [DoD, 2004]

In October of that year, the OUSD[AT&L] issued a policy that 'an addendum to this policy directs that a lead or chief systems engineer oversee the application of systems engineering in programs, supported by event-based technical reviews and assessment technical maturity and risk mitigation'.

Policy and practice was to be supported by the 'requirement for a SEP in DODI 2000.2 and provide specific content guidance ...in the Defense Acquisition Guidebook', and this is now the case with a chapter (Chapter 4) on Systems Engineering.

In DoD IPTs the SEP now provides insight into every aspect of a program's technical plan [Shaeffer, 2005], with particular concern regarding:

- what the project requirements are;
- who has responsibility and authority for managing technical issues, and generally detailing what staffing and organisation supports the effort;
- how will the technical baseline be managed and controlled;
- what the technical review process is;
- how is that technical effort is linked to overall management of project;
- it being a living document with use, application, and updates clearly evident.

Guidance on how to undertake this technical planning was provided in the *Systems Engineering Plan Preparation Guide, Version 2* in October 2007, which addresses five principle areas: program requirements, systems engineering resourcing, technical baseline management, technical reviews, and integration of systems engineering with programme management [DoD, 2007]

9.5.2 Through Life Management and Systems Engineering

“Through-Life Management of the delivery of military capability is complex and requires a forward-looking, long-term perspective to be applied to decision-making across the whole acquisition cycle. It involves a major change in culture and processes, systems and relationships for the Department and those doing business with it. Integrated Project Teams are responsible for managing an equipment throughout its lifecycle and are the focal point for Through-Life Management....New tools and information sources are part of Through-Life Management, such as Through-Life Management Plans.” [NAO, 2003]

80% of the IPTs considered that the systems engineering guidance provided by MOD insufficient. “There is precious little information” and “you can’t find it” summarised the situation. Given that Smart Acquisition and DIS gave emphasis to systems engineering, these judgments on enterprise’s level of support to projects are concerning.

Across the Acquisition Community, the TLMP has become a well-established element of IPT management and technical direction setting and decision recording. It is the primary element of planning and process documentation in IPTs. It is the common single source of reference for objectives, strategies, plans and project processes, and it provides a record of assumptions, risks and decisions. It records the evolving descriptions of the processes used day-to-day by the IPT, and shared with its stakeholders. With appropriate adaptation, it is thus a powerful vehicle to convey systems engineering planning and practice to the IPT.

TLMPs address the primary management issues: better informed, more balanced decision-making, and trade-offs regarding timescales and resources (budgets, materials and manpower). To these are added technical issues concerned with deliverable system properties, whole life cycle implications, and (in principle) the whole capability view of system. Following the NAO suggestion, systems engineering enhancements to TLMPs could strengthen both management and technical planning.

Though systems engineering is not explicit or recognised throughout its presentation, many aspects of the TLMP Guidance Web in the AMS align with the scope, purpose, concepts and content of ISO/IEC 15288 (which is by title a good practice model). Since ISO/IEC 15288 presents systems engineering throughout its text yet politically avoided the term in favour of system life cycle management, this lack of recognition is not unreasonable. Whatever the right and wrongs, TLMPs are currently the best approximation to IPT plans for systems engineering in an IPT context. Given the assertions of Smart Acquisition and DIS, the fact that TLMPs do not, nor are not urged to follow a systems engineering theme must be seen as a lost DE&S opportunity, one that the NAO publicly point to.

The AMS’s TLMP guidance associates project management with enterprise management, as in Figure 2.10, but does not explicitly present the equivalent linking to systems engineering. Within the guidance text, systems engineering (which is mentioned just once) is related to enabling strategies and linked to project resource, rather than being related to a planning and controlling strategy.

‘Through-Life Management Plans offer benefits but are not always used to manage programmes...[they] are central to the achievement of Through-Life Management ...[and] should lead to better delivery of integrated military capability rather than individual items of equipment’ [NAO, 2003].

TLMP hits rates on the ‘DPA’ website suggest they have acted substantially as ‘tick-in-the-box’ documents. It is not surprising that the EAC recommended renewed efforts to ‘review the effectiveness and application of Customer Supplier Agreements, Through Life Management Plans and Through Life Maturity Models’ [EAC, 2006, p.28] – a clear opportunity to revitalise and advance the practice of systems engineering in IPTs was being offered. The NAO and EAC have laid the ground; the opportunity has either not been recognised or found wanting.

The term plan is often a misnomer in many project team management situations; the 'plan' contains far more than planning information. This is certainly the case for TLMPs, which are a collection of interrelated information items that apply to why and how a system is created through its life cycle. Viewed in these terms, the web-based approach to the TLMP is a highly practical implementation that can handle the complex and evolving relationships that link these system life cycle information items.

The TLMP describes an IPT's *modus operandi* and define its code of practice for all who contribute to its delivery of capability. It provides the basis for common understanding and coherence of approach across the IPT, its stakeholders, and its external suppliers of goods and services. It should also act as an agreed baseline against which the project can be assessed.

It is required to define the purpose, goals, authorities, resources, execution strategy, administration and codes of conduct that characterise an IPT. It should define future action (plans or procedures) and past actions (records, reports), cooperative action (requests), and accomplishments (descriptions, specifications) – see Table 6.3.

Overall, the TLMP should provide a tangible and auditable information thread that describes how the users need is being transformed into delivered military capability using designated MOD resources.

A Project Management Plan is rarely a stand-alone document. It could, and probably should, be viewed more as a project information library comprising separate information items. This separation of project information into different documents is necessary. It arises because of the variation in:

- the nature of information is encompasses,
- the way in which information is used by different parties,
- the ownership and control of information,
- the user access privilege to information, which is of necessity not uniform across all TLMP information
- the timeframes over which information needs to be refreshed.

The TLMP is therefore unlikely to contain all information items relevant to the system life cycle. Nevertheless, it should reference all other plans and project information items that are directly relevant to the capability being delivered, some of which will be outside the authority of the IPT.

9.5.3 Augmenting TLMPs

A decision tree, shown in Figure 5 shows the decision logic associate with the 5 options identified is. It has four main decision branches:

- the first branch determines the merits of having a discrete document that has a form broadly traceable to either a SEMP or a comparable self-standing document, or alternatively is incorporated in some manner within the structure of a TLMP,
- the second branch determines whether a discrete solution should be seen as a SEMP of standardised form or whether it should break this tradition and be presented as an SEP that opens new content opportunities,
- the next option branch selects between a SEMP of conventional IEEE format or a more discretionary format that suits the purposes of the IPT and its Alliance partners,
- the final branch determines whether material within a TLMP would be better as discrete material or distributed through the structure of the TLMP's composition.

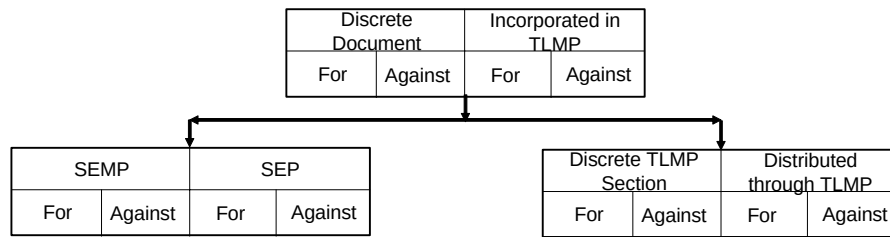


Figure 9.10 Systems Engineering Project Documentation Decision Tree

This decision structure leads to 5 possible outcomes that can then be followed. Strictly they are not mutually exclusive and taken together they describe a spread of possible approaches.

The selected approach followed could thus:

- be one of the 5 outcomes,
- combine more than one approach into a single solution,
- be one approach to meet short-term goals and transform into another approach long-term.

The options ‘trade-off’ is:

1. Discrete SEMP Against Incorporated within TLPM

This is the primary decision branch that would either explicitly introduce a new plan into the documentation structure or absorb the systems engineering planning into the established plan of a TLMP.

For	Against	For	Against
discrete, self-consistent presentation of principles, terminology, practise guidance	Opportunity for divergence from implied systems engineering in TLMP	Standard TLMP obliquely already contains some systems engineering content	TLMP not routinely consulted by IPTs
recognised identity as the source of systems engineering policy and practice	Emphasis of systems engineering as discrete add on to project action	TLMP maturity assessment can readily incorporate systems engineering factors	Not naturally seen as the source of systems engineering action
Distinct emphasis, change or purpose to existing practise	SEMP is yet another project document	Places systems engineering in the whole life project context	TLMP is already a weak presentation of systems engineering that needs radical change
concrete symbol of policy	Overlaps with standard SEMP content	Systems engineering content would raise the interest and usefulness of TLMP	Explicit systems engineering approach swamped by other information
Promotes an explicit approach to systems engineering	SEMP production perceived as an extra burden	Other planning, process definition in the TLMP	TLMP not strong on whole system approach

Table 9.4 Merits and Disadvantages of a Discrete Systems Engineering Planning Document and an Enhanced Through Life Management Plan

2. SEMP Against SEP

This decision branch leads to either a conventional name for systems engineering planning or breaks the standardised content and role within project plans by adopting the recent DoD term of SEP, opening the opportunity for change of scope and purpose.

SEMP	SEP
------	-----

For	Against	For	Against
Precedence of SEMP in project means that its existence is unlikely to be challenged	Tired term with stayed contents	Emphasises systems engineering thinking rather than project management plan	Non-traditional term that needs to be sold
Alliance System Integration Team 'initiated' and supported	Conveys a North American dominant culture	DoD initiative and views influence MOD explicitly and implicitly	Not explicitly defined in its composition
Concept and name been associated with defence work since Mil Std 499	One-solution-fits-all approach without tailoring guidance	Central to DoD systems engineering regeneration and an inevitable influence on UK approaches.	Systems engineering's lack of a clear identity compounded by a new document and emphasis
Term and indicative contents presented in the INCOSE Handbook Version 2	SEMP has evolved from a traditional equipment-based approach rather than capability	Systems engineering has been shifting in its role, purpose and nature and new information source would reinforce this.	Systems engineering instantiated differently during life cycle and at
Explicitly identified in the terms of this contract	"Tailoring" of SEMP has resulted in wide diversity	Many traditional benefits of SEMP will benefit from new ideas, e.g. ISO/IEC 15288	OSD Directives have not always turned into everyday project practice
Familiar to many industrial suppliers - Alliance partners	1990s thinking whereas MOD and DoD and US standards moving to ISO/IEC 15288 view	Opportunity for change; new name, new emphasis	
Derived from DoD 499 definition of systems engineering and thus defence oriented	Blocks the opportunity to emphasise ISO/IEC 15288 system life cycle management views	Outline SEMP already used by two FOAS Alliance partners (BAES Logica GMC)	Supplier SEMP's are statements of individual approach rather than cooperative convergence
IEEE 1220 still exhibits huge influence on systems engineering definition and practice	Weak in its through life views		
Considerable experience of IEEE 1220 in some of the biggest UK and US programmes	IEEE1220 weak in its system hierarchy and stakeholder views		
A prescriptive, sequential approach suits the defence life cycle approach	IEEE1220 weak in its capability and 'family of systems' views		

Table 9.5 Merits and Disadvantages of an IEEE 1220-Style Systems Engineering Management Plan and a DoD Systems Engineering Plan

3. Discrete Section TLMP Against Distributed Material TLMP

The selection of a TLMP approach leaves the option as to whether the systems engineering material is separated to give effect and identity, or is distributed throughout existing material to cover the scope without explicitly being systems engineering material.

Discrete Section TLMP		Distributed Material TLMP	
For	Against	For	Against
TLMP is a linked document repository that can readily incorporate additional elements	Could duplicate similar information in different parts	A low key introduction of systems engineering into existing documentation	Complex linking of content in order to see a coherent systems engineering message

Accommodates increments of change to systems engineering maturity in project with minimum interdependency on rest of TLMP	Can miss impact on related, or equivalent part of existing structure	Avoids conventional PMP material (in TLMP) and focuses on systems engineering	Failure to make systems engineering evident and thus disguises required change
Self contained and more manageable evolution of existing document	Emphasis of strategic nature and endorsement	Defines the systems engineering information only once	An amorphous, labyrinthine document which does not usefully convey systems engineering principles or coherent practice
Tailoring of TLMP is or will be the norm	Distorts a commonly used structure	Evolving systems engineering viewpoint influences all relevant parts of TLMP	Existing material may not be modified or biased to meet the systems engineering need without damaging existing purpose
Structure of systems engineering content can be borrowed	Evolutionary change to an as yet immature document concept	Incremental change can be readily managed	PDG changes to TLMP may not harmonise with systems engineering need

Table 9.6 Merits and Disadvantages of Systems Engineering Information in a Discrete Section or Distributed Throughout the Through Life Management Plan.

Following through this decision tree, the conclusion is that distributing systems engineering planning information throughout the current approach to TLMPs is the best approach. The rationale behind this selected approach is that it:

- places systems engineering in the whole life project context, and emphasises and endorses its strategic nature, The TLMP shall promote an evident approach to systems engineering with and explicit presentation of systems engineering process and practice.
- places systems engineering planning and management alongside other planning and process definition in the TLMP, The TLMP shall present systems engineering as a fundamental and intrinsic component of project action
- avoids emphasis of systems engineering as a discrete add on to project action, introducing it as an evolutionary development of existing project documentation through a change approach already in place,
- avoids divergence from implied systems engineering material already in TLMP and the duplication of similar information in different parts of FOAS documentation,
- allows the evolving systems engineering understanding to influence all relevant parts of TLMP as new material is placed coherently in the context of the existing TLMP,
- avoids another project document whose production and use may be perceived as a burden,
- permits TLMP maturity assessment already in place to be extended to incorporate explicit systems engineering factors and thus supports their committed introduction,
- offers raised interest in, and usefulness of, the TLMP through the introduction of explicit, internationally ratified systems engineering content.

From this a shift in TLMP should asset that:

- a) The TLMP shall achieve a shift of existing practise towards a system thinking and practise culture
- b) The TLMP shall be an IPT symbol of systems engineering policy
- c) The TLMP shall be developed to be the recognised first point of reference and guidance for systems engineering policy and practice
- d) The TLMP's day-to-day relevance shall be raised by its systems engineering content, through its increased usefulness and routine consultation.

- e) The TLMP shall encapsulate a self-consistent presentation of systems engineering principles, terminology and practice guidance.
- f) The TLMP maturity assessment shall incorporate systems engineering factors to assess the effectiveness of systems engineering enhancements permits and support its committed introduction,
- g) The TLMP shall place systems engineering planning and management alongside other planning and process definition in the TLMP, allowing it to be set in, and harmonise with, the existing policy and planning. The TLMP shall emphasise systems engineering thinking that strikes a balance with project management

The approach to effecting such a change should follow the comply with a number of implementation constraints

- No new structure shall be introduced, rather enhancement shall be through new emphasis and additions to the existing commonly used TLMP structure with some shifting of documentation home for information permissible;
- The implementation shall leverage off existing framework of material;
- Complex linking of content shall make visible a coherent and evident systems engineering message that accords with Smart Acquisition, DIS and EAC;
- Existing, implied systems engineering material distributed throughout the TLMP shall be re-presented in a coherent and consistent manner and its relationship to systems engineering made evident;
- A controlled introduction of systems engineering into existing documentation shall minimise IPT disruption and the web based approach generally being followed enables this;
- Descriptions and definitions shall be in accordance with the model of systems engineering practice described in ISO/IEC 15288 and many aspects of work products defined in ISO/IEC 15504 Part 6;
- The interfaces to other DLoDs shall be formulated according to systems engineering conventions to help drive its discipline into other lines of responsibility for capability;
- Reference to AOF/AMS material shall where appropriate be placed in a systems engineering context (assuming the continuing failure of AOF/AMS to do so)

The systems engineering work products described in §6 and detailed in Appendix A are the primary check list for aiding decision making about what needs to be brought into TLMPs and how they may be titled, described or rationalised in terms of systems engineering processes.

The TLMP Maturity Model is a tool to aid continuous improvement of Through Life Management. Its redefinition according to these implementation criteria could act as the vehicle for TLMP enhancements and definition of definition of project review criteria. This would bring TLMP upgrades into current DE&S practice definition and project auditing practice. The TLMP maturity assessments are synchronised to individual project schedules, thus any corporate resource required to assist IPTs prepare for such audits (as was found necessary for DoD IPTs during the DoD SEP Revitalisation actions and provided by USDL(A&T) staff) would be spread over time. The practicalities of augmenting TLMPs would appear containable.

9.6 Outcomes

This chapter has completed the product and service viewpoints of a system considered in §3.5.1, by examining an intermediate system state termed capability. The traditional product focus of systems engineering has already been impacted by an orientation towards seeing 'transitioned product' or 'quiescent service' as two aspect of the same notion: capability.

A technical definition of capability shows it to have been embedded, though not made explicit, in the systems engineering International Standard. It also shows it to be integral, though

generally not well-discriminated in frameworks of system models that describe architecture. This clearer technical meaning for a much used term in defence systems needs to be better reflected in systems engineering's codification.

The stronger model of what is meant by capability, and its relationship to organisational responsibilities through the life cycle, can help bring systems engineering discipline and coordination to other contributors to military capability.

A critique of the CADMID acquisition life cycle employed by MOD has drawn attention to weaknesses that are exposed by networked capability acquisition. The change from major, new-build equipment programmes to incremental acquisition goes hand-in-hand with this. This requires a reference life cycle model that offers a better balance between their regions of product creation and service delivery.

Such a model has been developed from systems engineering principles presented earlier. Termed the **Infinity** model to provide it with some distinction and to convey its double cyclic shape and its potential for management of semi-continuous life cycle, it superficially has sufficient in common with CADMID to offer a smooth transition to its employment. It confirms that other life cycle reference models can, and most probably will be developed to meet the changing defence acquisition scene.

The advocacy in Smart Acquisition and DIS for systems engineering to enhance MOD's acquisition capability does not appear to have been translated into effective guidance and explicit support for IPTs. Systems engineering practice could be better planned, conducted and managed IPTs by evolving the TLMPs into also being the explicit instrument of project systems engineering. The argument for this is made and some of its implementation criteria are discussed. The outcome would be to augment and more closely align existing practice and project procedure with systems engineering discipline, with limited change to existing management methods and artefacts.

10 Advancing Technical Practices in IPTs

10.1 Prologue

From the perspective of an ergonomics expert, Tom Singleton offered in 1974 a perceptive view of systems engineering. 'Systems-engineering then is concerned with man-made systems ... It is sometimes argued that systems-engineering is a new name for the traditional general engineer. This [is a] somewhat parochial view ... Systems-engineering ... re-emphasizes that good design ... requires a broad but balanced approach ... no longer confined within the limits set by the traditional engineering professions' [Singleton, 1974, p.10].

Singleton supported his advocacy for a distinguishable, broadly-based engineering discipline of systems engineering by pointing to its fulfilment of three canons. As he put it, 'the change to systems-engineering ... has three characteristics:

- a detailed attention to objectives;
- a consideration of function independently of mechanisms;
- a consideration of man as an integral part of the system.'

Over three decades later these three technical challenges still strongly characterise the discipline.

Singleton's trio of disciplinary quests are more readily recognisable in present-day systems engineering when expressed as:

- to effectively manage and define requirements according to need;
- to understand the complementary function and form viewpoints of a system's architecture throughout its design and evolution;
- to effectively accommodate human factors and the human element in the engineering of system solutions.

These three topics, considered in this order, establish the course of this Chapter. They have proved to be enduring technical issues. They are used here to exemplify how the preceding re-interpretation of systems engineering principles can revitalise IPT technical practices, enhance their effectiveness to acquire systems and contribute to military capability.

Singleton's 'detailed attention to objectives' refers to a disciplined, needs-driven approach to requirements management. It is an approach that depends on building a clear understanding of operational need and practicability at the holistic system or, for MOD, capability level. Rigorous and detailed requirements for the fabrication, the assembly, the test and the acceptance of a solution are then resolved according to these criteria through a succession of design transformations and decisions. This builds a requirements chain that links strategic need to fabrication detail and then back up to holistic unity.

Viewed in this way, systems engineering processes enable the definition of requirements that apply at all levels of system structure. This exemplifies the recursive nature of systems engineering principles and practices in keeping with the DIS's affirmation that 'systems engineering is as relevant in designing a software chip as it is for considering development of the military strategy for a particular conflict or the future force structure' [DIS 2005, p.60].

Singleton's second canon – 'a consideration of function independently of mechanism', meaning function distinct from form in this text – is concerned with a fundamental of design that stems from the systems reasoning presented in §3 and the formulations of process and work product in §4 & §6. This duality of appropriate functioning of a system solution and viability of realisable, compliant form depends on the mutual resolution of function and form stances and the effectiveness of their real world descriptions to communicate evolving architecture to designers and to subsequent builders.

The technique for designing architecture was concisely summarised in a quote of Marx: “what distinguishes the worst architect from the best of bees is this, that the architect raises his structure in imagination before he erects it in reality” [Jones, 1999]. The design of architecture thus depends on the raising of cognitive models that separately accommodate and resolve function as well as form, and do so in conjunction with models of effect. Whilst this relies on profound reasoning and cognitive modelling skills [§3.5] it increasingly depends also on effective methodological support in order to “raise” and harmonise a rich range of contrasting models in the imagination. Outside the imagination of the individual comes the integrated design or project team, and a need to communicate these models across diverse viewpoints, concerns and contexts of understanding and appreciation.

The project team responsible for the most infamous of ziggurats encountered the ultimate systems engineering obstacle (in mitigation, allegedly divine in origin): the difficulty of effectively communicating the design and build information between concerned parties – users, designers and ‘migrant’ manufacturers. It is probably the first well-documented engineering failure in history. Whilst it is tempting to speculate its cause may have been an inspired, culturally-irreproachable project management explanation for project failure, it illustrates well that system models, whether linguistic or of whatever form (see Figure 3.9), are indispensable for designing and building. It emphasises the *à la mode* term ‘model-based systems engineering’ to be patently tautological. Systems engineering simply cannot be conducted any other way. Models are not discretionary; only the concerns they address, and the way they are expressed through pragmatics, semantics and syntactics, are open to choice.

The formulation of frameworks that govern these system descriptions, their codification approaches and the semiotics employed, and how all this contributes to design strategies and methodologies, is key IPT practice. Present approaches are, however, somewhat immature, recondite and unavailing. A more effective assimilation of these frameworks of system models into design strategy, processes and methodology is thus an important task for the discipline of systems engineering.

It was 1983 when Norris, as Editor of the IEE Proceedings, surmised that in Learned Societies and their journals ‘the design process has only in the last 10 to 20 years become a topic suitable on its own for discussion ... Partly because there was no substantial design methodology ... process could be effectively described quite briefly: guess a design, assess it, modify it, reassess it and repeat the last two stages until you are satisfied ... Good guessers often based their guesses on their experience and knowledge of the design of similar existing products. Assessment also involves comparing a proposed design point by point with a fairly banal if crucial check list. The success of design ... rests on the ability to manage teams engaged in this process’ [Norris, 1983]. If this conveys a hint of cynicism about past approaches, it is tempered by some inescapable and fundamental realities of a design process: solution conjecture; its correlation with value criteria; and the role of the design or project team and its leadership. Just like Midwinter seven years later [§2.1], Norris did not identify such a method for experienced, knowledgeable ‘guessers’ and team managers as systems engineering, though his portrayal of basic aspects of today’s systems engineering design is clear cut.

In §2 correspondence and differentiation between how processes fashion man-made systems and natural systems was considered. On this point Dawkins observed that ‘design and natural selection are both processes of gradual, step-by-step, progressive improvement. Natural selection, at least, could not be anything else. In the case of design it may or may not be a matter of principle, but it is an observed fact’ [Dawkins, 2005, p457]. This Chapter addresses why this observed step-by-step progression in the design of man-made systems may indeed be a matter of principle. It is postulated how it most probably derives from the systems reasoning presented in §3 and, as a result, shapes design strategies and life cycle patterns [§5].

When standing before some of the earliest depictions of weaponry, and of how they were employed and to what effect, Jacob Bronowski observed of the cave paintings of Altamira [Bronowski, 1973, p56] that ‘the men who made the weapons ... [were] anticipating a future as only man can do, inferring what is to come from what is here. There are many gifts that are unique to man; but at the centre of them all ... lies the ability to draw conclusions from what we see to what we do not see’: from present reality, through cognitive models, to future

reality. Bronowski continued; 'art and science derive from the same human faculty: the ability to visualise the future, to foresee what may happen and plan to anticipate it, and to represent to ourselves images that we project and move about inside our head ... [these images] direct the mind from what is seen to what can be inferred or conjectured'. These are images – our cognitive stances – at the crux of design. It is advanced in this chapter that they condition the strategies and tactics we employ throughout design, and likewise our perception of architecture.

With regard to Singleton's third canon – man as an integral part of the system – it is Ramo's insight that captures the point perfectly, 'it is nevertheless possible for systems engineers to become wise about people as systems components. This results from the unique experience systems engineers get in designing and studying the workings of systems with people as members. From the beginning, the human components must be specified and evaluated as to cost, performance, stability, and time for development and training ... just as adequately as we list the specifications for the inanimate portions of the system and subject them to analysis' [Ramo, 1998, p.59].

This human component is fundamental to the systems portrayed in the cave paintings of Altamira. These tell of man's operational objectives (hunting animals), the functioning of man-made mechanisms (spears and bows) in this endeavour, and the roll of men as the operatives that control and contribute to the functioning of a system of hunting employed over 6000 years ago. The cave paintings are thought most probably to be instructional; more part of an enabling training programme than art. They convey the role of the individual in the total system operation and they contribute to the conditioning – the programming according to an empirically refined procedure – of the most sophisticated, the most complex and behaviourally the most unpredictable element of this hunting system, or indeed any other system: the human. Harnessing that complexity to best overall effect is a sophisticated, error-prone task. It presents a rich, contrary array of 'implementation technology' properties and interface incongruities that, above all other technical specialisms, requires a well-integrated team approach.

This chapter therefore closes this investigation by considering some generalities of how implementation technology-based engineering specialisms and systems engineering interact, and it explores these in terms of the most challenging of the 'implementation' media. Understanding humans as parts of a system-of-interest and also as parts of an environment of operation addresses some common ground. It begins to open a door for engineering and management, conducted in accordance with systems reasoning, to be applied to the wider world: to an operating reality of 'soft' challenges that ultimately resolve into that complex of human activity systems that is the social environment – that region where as observed in §2 Ramo, and many others since, appeared to stumble.

10.2 Synopsis

This chapter considers three areas of detail in the technical view of system life cycle process. It relates these to IPT concerns and risks identified in the IPTL interviews.

The three technical topics are investigated to determine what role the transformations in Figure 4.7 perform in the dynamic system of life cycle processes, how in more detail they inter-relate sequentially, iteratively and recursively, and what purpose of their output work products serve. From this, existing weaknesses and new opportunities to advance systems engineering technical practice are considered.

Being a holistic system of processes, the mutual dependence of the system life cycle processes make a clear-cut separation of technical concerns impossible. However, it is possible to take viewpoints of this process system such that the three selected topics may be treated substantially separately. The trio of viewpoints concerns employed here is:

- *defining requirements*: describing what pre-exists in the real-world, or is assumed will exist and how it should change, along with the design and implementation constraints adopted to govern a technically-based intervention in that real-world;
- *designing architecture*: conducting decision making and rationalisation that strategically resolves a path of progressive, evermore detailed system technical

options in order to formulate the structure of a solution in accordance with defined or evolving product or service value criteria or to exploit an opportunity.

- *implementing technology*: selecting the most appropriate medium and defining its formation, fabrication and conditioning actions, so that the properties of a system element comply with the intent of an architecture.

The chapter's first focus is on requirements. Requirements correspond to a class of life cycle work product that includes several distinct instantiations, each fulfilling markedly different purposes at different steps along the path in Fig 5.5 and at different layers in the recursive service/product/ technology model in Fig 4.7. The role and characteristics of each type of requirement serve the interests of IPTs in their support of capability-based planning, their creative contributions to solution development, and their supervision and ratification of the implementations provided by industrial suppliers.

The technical work products generally identified by the term 'requirement' contribute heavily to the overall technical information flows through the life cycle. They are also key to defining agreements at the inter-/intra-organisational trading interfaces between managerially and/or commercially autonomous organisational entities. Requirements, therefore, are not just a technical matter. They are a focus of the cooperative relationship between IPTs and their suppliers, and also their customers. It is thus essential that IPTs manage requirements in order to simultaneously serve the two purposes: building models and reality, and binding trading organisations.

Singleton's "consideration of function independently of mechanisms" is taken up in the subject of design and in its profound association with issues of architecture. The separation of function and form is a hugely powerful stratagem for engineering design. Its foundation in systems reasoning is implicit in the earliest, inchoate codification of systems engineering through to the now refined, present-day emphasis on architecture frameworks, and their organisation of system models and their communication formalisms.

The recursive application of a stratagem for resolving 'logical' and 'physical' models over the entire range of system detail is the basis of systems engineering design strategy and of architecture definition and description. Such stratagems relate to the individual's instinctive accommodation of inconsistency between stances and, together with the necessary description conventions, for integrated teams.

Propositions are made as to why architecture is presently a term with such diverse interpretations that common understanding is threatened across communities who should naturally share a unified ontology. In the realm of defence systems at least, the current dichotomy in systems engineering and information technology needs to be acknowledged and reconciled. It is suggested that contrasting systems engineering and IT/software engineering architecting interpretations can be resolved within a common set of systems engineering principles as advanced in §2, §3 and §4. This is argued by extending the systems reasoning in §3.5 to provide an explanation for the somewhat subjective and inconsistent outcomes seen in enterprise architecture frameworks, including those that have evolved for defence systems.

The last topic considered examines the lowest layer in Figure 4.7 in which systems engineering as a discipline must interface with multiple engineering specialisms. Their separate natures and modes of conduct are substantially characterised by the individual media they employ in order to effect the function and form they contribute to holistic system structures. The medium selected here to illustrate this is people. In contrast to §6, here people are not actors in the system life cycle but components of the problem and solution space; that is, voices of need and acceptability, and also elements of created solution.

This topic addresses what has been a traditional weakness of systems engineering – its overriding preoccupation with the inanimate – and examines how the principles and practices of systems engineering and human factors engineering need to be integrated in order to fully understand systems-of-interest inside and outside their boundaries.

The enterprise systems and social systems in the upper levels of Figure 3.2 often require interventions to remedy 'situations' whose cause may relate to the 'soft' complexities of human work systems. Not having readily stateable 'hard' requirements from the outset, a

distinctly empirical approach to exploring need is often required in these situation; a situation familiar to human factors specialists. It is seen that problem 'hardness' and design strategy it is a matter of degree, and that the systems engineering model in §4 can be applied to human activity systems, applying the same transformation rules and work product classes as in 'hard' system problems but resolved according to different patterns..

10.3 Defining Requirements

10.3.1 The Nature of Requirements

In the interviews IPTLs strongly associated Requirements Managers with the day-to-day conduct of systems engineering [§8.4.3]. This emphasises the profound importance of requirements in the IPTs' role of acquiring military systems from industry. The AOF top-level, somewhat enigmatic definition of an IPT confirms this requirement-dominated role: 'the body responsible for developing the System Requirement Document, devising equipment solutions to meet that requirement'.

However, despite Smart Acquisition's introduction of Requirements Managers as knowledgeable interlocutors between DECAs and IPTLs, the IPTL interviews conveyed uncertainty about how satisfactorily requirements do convey a need for military capability, and how the equipment need and solution definitions are appropriately resolved across different system life cycle process work products.

As one senior IPTL put it, requirements "are as much art as science", and this suggests that, even after an expensive decade of learning new approaches to requirements, some aspects of systems' principles and practice do not yet appear to have been wholly assimilated in MOD requirement management guidance and practice.

Whereas requirements may still appear to be art they are, and should be subject to, disciplined engineering practice. Innate skills certainly do assist in the social context of requirements elicitation, but at the heart of requirements definition lies the semiotics, semantics, syntactics and pragmatics of modelling. It is out-and-out disciplined engineering, governed by management goals and conducted according to systems reasoning, i.e., indivisibly it is systems engineering.

In its 2007 guidance on Requirements Management the AOF supports this, asserting that 'maximum benefit (to both the supplier and MOD) is gained when the expression of user need, and its evolution into the system requirements, is consistent with systems engineering principles and practices. Smart Requirements is therefore designed to be the initiating stage in a systems engineering life-cycle' [AMS, 2007].

Beginning with the work products delivered by the technical transformations in §4 and from their generic characteristics described in §6 and their individual characteristics in Appendix B, this section opens its consideration of some opportunities for advancing requirements practice. It does so by reviewing how requirements are employed to model yet-to-be systems. This then becomes the basis for considering (in §10.3.2) how requirements should be used throughout the life cycle, and how (in §10.3.3) they can be 'misused' in inter-organisational trading. These analytical steps suggest why, where and how systems engineering might strengthen the IPTs' approach to managing requirements.

The IPTL responsibility for "understanding what is possible, what is wanted, what is affordable and what is achievable" [§8.5.1] rests heavily on a user needs-driven approach that reconciles the problem space with technological options offered by industry. The drive of Smart Acquisition for MOD to clearly state the requirements for military need before becoming engaged in solution was motivated by:

- radically changing defence capability needs that should be met better by following an approach that actively avoids a past propensity to define solutions and by open-mindedly considering opportunities emerging from industry's offerings;
- the diminishing capacity and capability of MOD's downsized level of technical expertise to define and, in any reasonable way, manage solution detail;

- a policy move towards a new, cooperative, risk/benefit-sharing relationship between MOD and its industrial supply base in an increasingly volatile military threat situation.

From its outset, Smart Acquisition's attention to requirements switched to the development of two cardinal work products: URDs and SRDs, each having distinct purpose and characteristic form. Their importance lies in a separation of concerns (or viewpoints) that had been seen to greatly benefit organisations adopting this approach, particularly in a government body/agency and industrial supplier trading situation, e.g. European Space Agency. As a technique for defining and managing requirements it had been transposed from its software engineering origins [Mazza, 1994] into the generality of systems engineering [Arnold, 1997], [Stevens, 1998]. In addition, at the start of Smart Acquisition these ideas were being incorporated into the ISO systems engineering model and thus were set to have a wide influence on systems engineering discipline

During the interviews, IPTs referred to their use of and contributions to Customer/Supplier Agreements, User Requirements Definitions, Single Statements of User Need, Capability Requirements Documents, Statements of Work, Urgent Operational Requirements, System Requirements Definitions, Equipment Capability Requirements, Staff Requirements, Key User Requirements. Each of these requirement constructs appears to present its own particular message, style and role in the overall scheme of the life cycle, and to exhibit an individuality that results from local interpretations, accommodations and conventions.

Requirements, however, do not contribute solely to the overall technical information flow through the system life cycle. They are also key to defining agreements at the inter-/intra-organisational trading interfaces between commercially and/or managerially autonomous organisational entities. Their influence thus goes well beyond technical matters, for they are a focus of the cooperative relationship between IPTs and their suppliers, and equally between IPTs and their customers/sponsors.

It is thus essential that IPTs manage requirements so as to simultaneously serve two purposes: binding engineering models with future reality and binding trading organisations in common enterprise.

Although in the MOD acquisition community the term requirement is predominantly associated with URDs and SRDs, there are in any engineering undertaking several technical work products that act as requirements, and they address a wide range of concerns. They all fall within the Generic Work Product class Specification in the scheme of Table 6.3. The members of this class are various, varying, variable and varied. That is, they are:

- various, because there are several members within the category 'requirement'; each serves a different technical purpose, is associated with different technical transformations, and thus is distinct in kind;
- varying, because throughout the life cycle any member of this category is subject to change regarding its extent of completion, and its state of maturity in the light of evolving understanding and developing opportunities;
- variable, because each instantiation of a requirement contains characteristics that depend on the nature of the system-of-interest, its ranking in the hierarchy of structural detail conveyed in Figure 3.2 and therefore the degree of implementation constraints it contains;
- varied, because requirements attract forms, styles and titles as a result of (usually valid) local tailoring that accommodates the specifics of system use, inconsistency in corporate guidance, dominant technology influences, community-of-interest practicalities, and proven custom and practice.

The challenges that are presented by this diversity are considered in this order.

Of the 33 ISO/IEC 15288:2002 work products designated to the Generic Work Product class Specification, 15 relate to various system-of-interest technical specifications (with 7 relating to enabling systems, 8 to management and 3 to trading agreements) [Appendix A, 8.0]. Each member of this class of work product conforms to characteristics that:

- ‘ – [are] criteria or conditions that place limits or restrictions on actions, attributes or

qualities;.

- establish measures or qualities for determining acceptability, conformance or merit;
- may be required as part of an agreement or contract' [Table 6.3, 8.0].

Because of this diversity there is no single or best practice proforma for requirements; even for one type of requirement. There are, nonetheless, technical and agreement ordinances that need to be kept in mind. When these are strayed outside of, it is the loss of their high leverage early in the life cycle that can lead to major adverse impacts on defence acquisition outcomes and to nugatory expenditure, as conveyed in Figure 7.3. The early decisions in the life cycle shown in Figure 7.3 become the foundation of the evolving URD and the initial formulations of an SRD, and these two bodies of requirement description are major influencers of benefit/cost throughout the life cycle.

Given this situation, a Smart Acquisition technical aspiration was to introduce systems engineering discipline into the building of effective requirements models, and from this to improve the associated, integrated definitions of test, evaluation and acceptance.

Fundamentally, the various members of technical requirements define a range of necessities, problems, directives or limitations, according to past decision making and assumptions. They act to limit future decision making associated with:

- service need;
- viable modes of use and operation;
- problem and solution assumptions;
- valid user/acquisition agent constraints on solution function, structure and implementation;
- product architecture design definition;
- element implementation definition;
- fabrication instructions;
- assembly information;
- conditions of conformance with design;
- criteria of compliance with need;
- contractual acceptance stipulations.

This list illustrates that the term 'requirement' can cover multiple technical issues. For any concerned party it applies to their system-of-interest when it is viewed as:

- a set of services in a defined environment, or as a capability envelope;
- a holistic product, or specifically as an equipment;
- the constituent technology elements or fabrication entities.

Depending on the observer's point of view, the various technical work products generated during the life cycle act as input requirements to companion life cycle processes. Thus, in one guise they appear as design information outputs, in another as requirement definitions. The same technical work product may be viewed as (and therefore be termed) either a requirement model/document or a design model/document depending on the responsibilities and other concerns of the observer.

Figure 10.1 shows how work products can act as requirement inputs into neighbouring technical transformations in a network of life cycle processes, and how each arises as the result of a transformation that has logical precedence. Each input work product acts as a 'requirement' on the transformation to be performed. It defines both the input conditions and the constraints on the decision making that is permitted. This figure shows a minimum set of technical requirements according to the ISO/IEC 15288 model.

A work product's most meaningful title thus depends on the viewpoint of the observer. Some work products are explicitly referred to as requirements (notably stakeholder/user and system/product) because they are system models that are typically started early in the execution of technical processes. They act as cardinal points of reference for subsequent decision making and they are therefore most commonly viewed as requirements. In this way, the design-centred perspective that has dominated much of systems engineering's formulation conventionally associates 'requirement' with definitions of service and of product. Conversely, it has associated 'design' with descriptions of the outputs from architecture design and technology implementation. This is a convention that has been followed in Smart Acquisition.

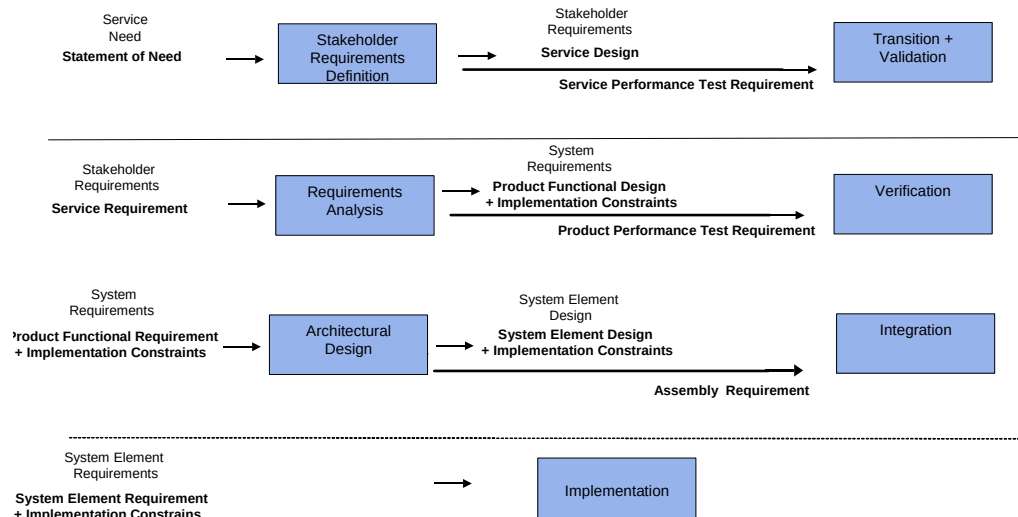


Figure 10.1 Technical Requirements Corresponding to ISO/IEC 15288 Process Transformations

Despite Smart Acquisition's stress on a needs-driven, top-down approach to through-life management, an effects-only definition of need actually runs counter to the natural inclination of systems reasoning. Individuals as a matter of course think about problems according to pre-existent solutions. The three cognitive stances are built naturally in concert; albeit with a pattern or emphasis that in individual instances of design is influenced by the evolving problem understanding, and by the prediction/presumption/precedence of a solution's function and form. Each of the stances benefit from – indeed may rely on – understanding of the others, and the models in Table 3.4 are cooperatively resolved through progressive accommodation, one stance with the others.

Given this natural interdependence of stances, it requires emphatic discipline to isolate expressions of the intentional/effect stance of a system from corresponding design/functional and physical/form stances. This unnatural task is exacerbated by the need to simultaneously build to some extent the function and form models of a solution in order to explore necessary constraints, whether designated by the stakeholders or arising from the limitations of knowledge and technology.

Different strategies have been followed during Smart Acquisition to tackle this problem/solution viewpoint distinction. The initial (and still valuable) approach of inspection of textual requirement models against criteria of user need or of solution design was advocated from the outset. At its very simplest, for URDs a recommended form of 'the user shall ...' (or more correctly, 'the service/capability shall ...') as opposed to 'the system shall ...' for SRDs (again more correctly, 'the product shall ...') was advanced in the AMS, following Stevens [Stevens, 1998, p.33, p.81].

However, an influential and effective strategy has been to separate the concerns for problem from concerns for solution by a physical separation of mindsets. This has been achieved either through enlisting consultant 'requirements engineers', who sit outside the design focus of a project, and/or (as subsequently introduced) through the role of Requirements Managers,

who as military assignees bring an operationally-immersed viewpoint rather than a design or technology viewpoint. Either or both of these approaches act to counter the problem/solution melding that is natural to systems reasoning.

Most recently, clearer separation is becoming possible through the methodological approach of a multi-perspective, codified framework that incorporates views of a system according to the different viewpoints/concerns (as described in the systems reasoning of §3.5.1) of the problem owner and the solution definer. Requirements defined in this way (typically supported by discipline and rigour of enabling tools) conform to a framework of standardised system models (which in the case of MOD is MODAF, and which defines a Strategic Viewpoint (capability definitions) and an Operational and a System Viewpoint (solution function and form definitions)). Standardised system model definitions of this type are thus exerting a new and disciplining influence on requirements definition by specifying tightly defined work product attributes, modelling conventions and contents, and this is considered further in §10.4.3.

At any instant in the life cycle each requirement is the outcome of an accumulation of decisions about what the intended intervention in reality (the service) will be, and how this will be achieved in terms of the function and form of an intervening object (the product). Ultimately, these decisions unfold down to the most detailed level of fabrication and interconnection of the intervening object, i.e., the entire description of what the system-of-interest is.

System creation may be characterised as a continually advancing boundary of decision making, and this can be seen in retrospect as the limiting surface of 'requirements space', i.e., the view of the cumulative decisions already made. However, this same boundary also appears as the leading surface of the 'design space' of future decision opportunities. In common with many features of systems, as indicated this dichotomy is a matter of observer concerns or viewpoint. Thus, one observer's problem (requirement) is another observer's solution (design), or as conveyed in Figure 10.1 one concerned party's solution gives rise to another's problem.

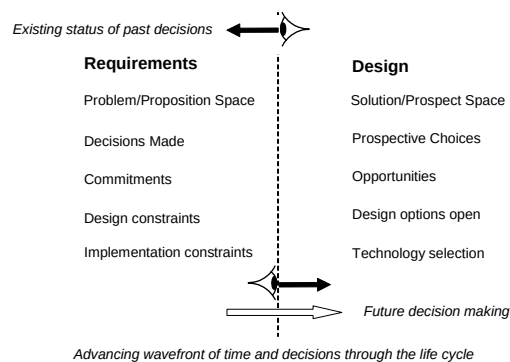


Figure 10.2 Boundary of Decision Making through the Life Cycle

This decision boundary is represented in Figure 10.2. It shows to the left the region of problem definition in which decisions have been made and commitments built up. The advancing wavefront defines a consolidation of technical decisions and progress (with possible re-visitations of earlier decisions through the life cycle). The accumulating body of commitments, design decisions and implementation constraints (to the left of Figure 10.2) shrink the region of decision options as solution development proceeds. Ahead of this wavefront of decision making (to the right of Figure 10.2) lays the solution space of design options and prospective choices. Here design decisions remain open, opportunities exist and the selection of technology specifics is yet to be taken.

This progressive decision making means that system models are evolving with time and that each time they are (re-)defined or baselined, whether as a function of event, elapsed time,

responsibility change or level of detail, they are re-expressed according to their extent of completeness, rigour or maturity.

As design decisions accumulate and the (design space) options progressively give way to (requirements space) definitions, so design options close down with convergence on a fully defined solution; that is, to a description of the detailed fabrication/technology processing instructions and detailed integration scheme/procedures.

As Figure 4.8 conveys, an unfolding resolution of all possible design decisions is achieved through the recursive application of process transformations at successive levels of structural resolution. Thus the technical requirement forms described in Figure 10.1 can in principle be defined at any level of structural composition – a URD begets subordinate URDs, for example (for a system element) as structural and technological detail that ‘delivers service’ into its environment of super-ordinate system-of-interest structure, and so on.

In practice, these recursive definitions of requirement are generally associated with changes in responsibility. They corresponds to new circumscriptions and fresh levels of structural detail, and (as considered in the following next sections) this establishes the trading boundaries between business partners and the links in a supply chain. However, in the case of major programmes within large prime supplier organisations, it can still be beneficial to have intermediate restatements of requirements, and hence fresh descriptions of design. This provides clearer visibility and control of what otherwise could become risk-hiding monolithic system developments.

Definitions of a user (service) requirement, or a system (product) requirement, or a requirement for an element of (product) architecture or a (system element) implementation requirement each serve their respective purpose at successive levels of structure and composition – for example, in the structure in Figure 3.2, from military capabilities (enterprise systems), to equipments (product systems), to sensors (technical systems), to the technology of components (physical science systems). Each requirement form has its counterpart at any chosen level. At each level they reflect the changing nature of the system-of-interest; they may adopt different notations/language in accordance with the changing nature of function and form, and may attract different names. For MOD, the form and titling of URDs and SRDs were largely based on descriptions at the equipment level, as described by Stevens [Stevens,1998], and in this regard they are most closely identified with the mid-region of Figure 3.2.

Thus in summary, the overriding distinction between requirement definition and design description is seen to be viewpoint. The same system model can contribute to the view seen from either viewpoint [IEEE, 2000]. This notion substantiates the observation by Finklestein that ‘the formulation of requirements can be viewed as a creative design process’ [Finklestein, 1983, p.217]: whether verb or noun, requirements and design are in many regards two sides of the same coin.

10.3.2 Technical, Management and Agreement Influences

This section considers why the system life cycle work products that MOD and most other organisations refer to as requirements are subject to conflicting influences. Three key determinants are considered here:

- the need to effectively communicate technical models ultimately created by a concurrency of cognitive mechanisms that regulate systems reasoning;
- a logic of sequential transformations and work products that simplify procedure, and with this management’s assessment and supervision of project progress;
- the need to comprehensively and unambiguously specify a system or its parts such that delegated responsibility can be agreed and trading risk minimised.

Together, these present a ‘trichotomy’ that contribute to differences – in fact an unduly wide spectrum – of interpretation in the purpose, composition and titling of requirements. From an analysis of this situation, a strategy is suggested to resolve apparent inconsistencies in the management of requirements by MOD and many others.

The first two of these three determinants have already been considered in detail. They are summarised respectively in Figure 4.9, *The System Design Triad*, in which the concurrency of stances is brought out, and in Figure 5.8 (the 'Waterfall Model') in which a logical sequence of life cycle processes (and hence requirements work products) is clearly evident. The third determinant is implied in Figure 4.19, *The Typical Information Transfers at Trading Interfaces*, and is now developed further.

IPTs face the need to communicate across two immensely difficult information transfer boundaries. This following examines how requirement are used to convey technical information across the IPT trading interfaces: 'upwards', when interacting with their sponsor and their users, and 'downwards', when conducting inter-organisational trading with their industrial suppliers.

Despite being a part of the same organisation, the IPT must meet the challenge of effective technical dialogue 'upward' with its DEC sponsor in order to establish agreements. It must be confident that the information flow at this internal trading boundary is adequate to perform the service-to-product transformation from a URD to the corresponding SRD. During the course of acquisition, the IPT must be able to exchange solution concepts and decision options with its sponsor/users, and to contribute to establishing and modifying the Customer/Supplier Agreement drawn up between them.

'Downwards' across the MOD/industry trading boundary, the IPT is faced with two regions of difficulty in the exchange of information. Technically, the challenge is the comprehensive and concise transfer of accumulated decisions, and the avoidance of unwarranted solution definition. In addition, there is a trading challenge: requirements also contribute to bridging business boundaries. They must enable the expression of legal agreements and government/industry trading criteria, allowing the formulation of contracts that leave open opportunities and encourage value-for-money solutions.

Technical requirements are thus crucial to the agreements at inter-organisations trading interfaces or, as in the case of DEC and IPTs, between major organisational elements. As conveyed in Figure 10.1, comprehensive technical communication at these responsibility boundaries depends on information from more than one type of requirement.

In order to consider these issues and to prescribe a solution strategy, this text establishes a clear distinction between the terms 'requirement' and 'specification'. Although the two terms are frequently used synonymously, there is an essential semantic distinction. This distinction is drawn in areas of existing defence acquisition practice.

In its common usage requirement refers to 'something demanded or imposed as an obligation, a thing desired or needed' [*Collins, 1991*, requirement, meanings 1, 2], whereas a specification refers to a 'detailed description of the criteria for the constituents, construction, appearance, performance, etc., of a material, apparatus, etc., or of the standard of workmanship required in its manufacture' [*ibid*, specification, meaning 3].

Taking a systems engineering-specific view, the *ISO Software and Systems Engineering Vocabulary* defines a requirement to be 'a condition or capability needed by a user to solve a problem or achieve an objective'. It distinguishes a specification as being 'a document that specifies, in a complete, precise, verifiable manner, the requirements, design, behaviour, or other characteristics of a system or component and, often, the procedures for determining whether these provisions have been satisfied' [*ISO, 2007*].

The general sense from these two pairs of definitions is that a requirement describes a need or obligation in which the exact nature of the outcome is open to resolution. The desired outcome is subject to imposed limits and constraints, but within these, freedom exists regarding solution. In contrast, a specification is a sufficiently complete, detailed and concise description of an object so as to assure its compliance with provisions. To demonstrate that this concise description has been complied with, the criteria that demonstrate compliance are also commonly defined.

This underlying distinction exposes that requirements apply to technical achievement, whereas a specification applies to trading achievement. The implication is that requirements are an obligatory part of a specification, with many attributes being common to both.

This semantic distinction basically conforms to long-standing US defence practice as expressed in Mil-Std 961. First published in 1981 as *'Military Specifications and Associated Documents, Preparation of'*, successive revisions of Mil-Std 961 have provided the most coherent, stable and practical guidance on defence specifications. In the Foreword of its present text [DoD, 2003e, p.iii] three statements encapsulate the nature of defence systems specifications.

Firstly, 'the purpose of a specification is to provide a basis for obtaining a product or service at an economical cost and to invite maximum reasonable competition'. This emphasises that, as distinct from requirements, specifications explicitly address business trading issues. Two particular commercial considerations are highlighted: a concern with the time, effort and resource drivers in order to minimise cost, and the importance of communicating military needs in a way that encourages competing alternatives – hence repeated stress on requirements that avoid solutions.

Secondly, this purpose is achieved if a specification meets four criteria. These are to:

- identify minimum requirements;
- list reproducible test methods to be used in testing for compliance with specifications,
- allow for a competitive bid, and
- provide for an equitable award at the lowest possible cost.'

The criterion of 'minimum requirements' reduces the chance of "solutioneering", and so encourages creative, competitive bids and avoids the cost of unnecessary constraints.

Since specifications contribute to agreement and trade, the establishment of trading partner consensus on *how* compliance shall be demonstrated can be of equal importance to *what* must be complied with. Specifying how verification is to be conducted establish the conditions of compliance and, allowing for discretionary concessions, these are conventionally expressed in terms of succeed/fail outcomes – not least to avoid contractual/legal equivocalty.

Despite over a quarter of a century of its application, the third of the Mil-Std 961 statements on the nature of specifications tellingly still sees the need to advise that the 'proper preparation and use of defence specifications is a difficult task'.

Why, with so many years of experience, is this warning still necessary?

It is advanced that, in the case of MOD and widely elsewhere, it arises from:

- the extent to which the definitions of requirements balance the concurrency of the stances that govern the resolution of system models [Figure 4.9], as opposed to being a succession of work products that emerge from an organisation's daisy-chained instantiation of event-sequenced procedure that is fashioned to permit ready, if not always accurate, gauging of project progress and achievement, i.e., a sequential, logical view of 'documents';
- the need to create specifications at trading interfaces that coalesce different requirement forms and have a tendency to bias and blur their individual purpose, ownership and definition.

From this three contrasting influences arise, as illustrated in Figure 10.3. They are, left to right in the figure:

1. Cognitively-based principles – the conventions of a logical relationship between technical transformations and work products that respectively define a system's attributes of effect, function, form. The grey boxes equate to an effect model, a function model and a form model. They are built and mutually resolved according to cognitive mechanisms having a fluidity of temporal order. This representation derives from Figure 4.9 and conveys the concordance of models that are the basis of co-existent stances/observer viewpoints.
2. Team/organisational practices – the technical information artefacts that are developed according to management procedures and time-driven plans that govern

their availability. They depend on an idealised progression of decision making and build up of information. The grey boxes thus convey a logical sequence of requirement/design decisions. This representation derives from Figure 5.10 and conveys the logical rationale of sequentially-instantiated procedure.

3. Business agreement provisions – the aggregated technical requirements (with redundancy removed) expressed at particular levels of system detail such that responsibilities may be conferred from one party to another, i.e., a hierarchy of technical specifications sufficient to conduct trade. At these levels of structural detail, it becomes possible for a project/organisation to confidently establish an agreement for the acquisition/supply of system elements. As Figure 10.1 shows, up to 6 types of requirements may contribute to a specification at a trading (acquisition) boundary. The grey boxes are thus composites of information assembled from different requirement forms. This representation derives from Figures 4.17 and 4.18.

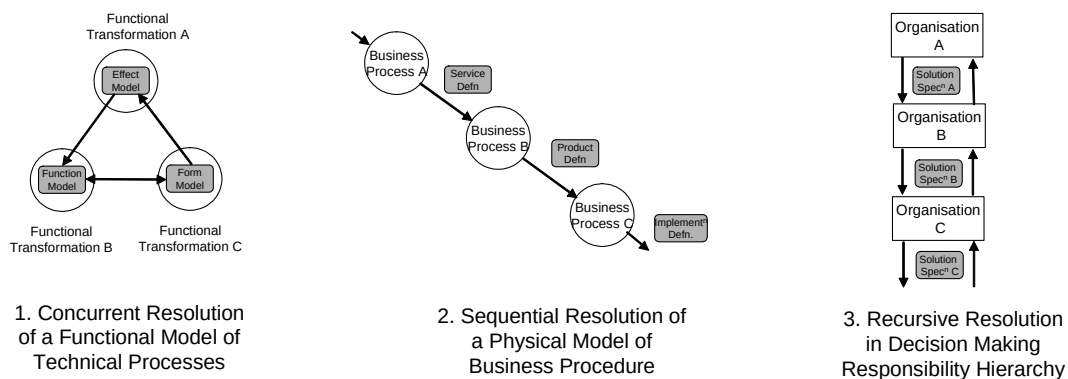


Figure 10.3 Concurrent Process, Sequential Procedure and Recursive Agreement Influences on Requirements

Figure 10.3 therefore illustrates three ways in which requirements can be interpreted and variedly applied: as models of an object governed by stance, i.e. cognitive models of reality; as work products governed by business procedures, i.e., information artefacts of system life cycle processes; as specifications that govern trading relationships, i.e., operative agreements that govern a hierarchy of responsibility and design decision making.

This figure emphasises the ontological difference between three classes of grey object which in practice are often weakly distinguished regarding their origin, purpose, names, nature or ownership. Not surprisingly, requirements can easily, almost subliminally become entangled in these three interpretations or viewpoints. In MOD, and widely elsewhere, there is generally insufficient corporate formalism and institutionalised guidance to distinguish between and resolve them.

In fairness to MOD, ISO/IEC 15288 harbours a linguistic compromise in its combined description of a logical state model and a physical real-time model of system life cycle processes, as outlined in §4.4.1. Its (constructively ambiguous) description is intended to convey and harmonise either interpretation, allowing it to be construed to meet users' circumstances and inclinations. A consequence of this linguistic compromise is that it then handed on the difficulty of reconciling these two viewpoints that arise from the 'complementarity' of function and form evident in any system – which paradoxically in the case of the International Standard is a system of systems engineering processes. Eventually these two contrasting descriptions of a system must be resolved. Each describes an identical reality. Each uses its own modelling conventions. Each describes particular attributes and properties but is unable to explain others. Revelations and changes in one have to be reconciled in the other; both are necessary to convey a rigorous description.

How does MOD guide IPTs to approach these issues?

The AOF presents the User Requirements Definition Process as the 'MOD customised version of the ISO 15288 'Stakeholder Requirements Definition Process'' [AOF, 2007a].

Likewise, the System Requirements Definition Process is the 'customised version of the ISO 15288 'Requirements Analysis Process'' [AOF, 2007b]. This leads to a persistent, underlying lack of clarity in the exact nature and purposes that URDs and SRDs serve, and what is and is not included in them. ISO/IEC 15288's compromise is inherited and not resolved by MOD.

It is therefore not surprising that some of Smart Acquisition's approach to requirements management, documentation and ownership has been consistently vague throughout the period of the AMS, and is perpetuated in the two revisions of 'Requirement and Acceptance' guidance in the AOL (April and October 2007). At the close of this investigation a lack of conciseness remains, as evidenced in statements such as 'System Requirements Document (SRD) or contract Statement of Requirement (SOR) or Equipment Contract Specification (ECS) [shall be] finalised and agreed with contractor' [AOF, 2007d]: a linguistic manifestation of the 'trichotomy' conveyed in Figure 10.3.

Each IPT appears to pitch its approach according to local factors, custom and inclination. The outcome is a multiplicity of styles and content, graced by a catch-all notion of "tailoring". This introduces a degree of unwarranted non-uniformity across the acquisition community, with the potential to adversely impact system interoperability due to localised practices.

Figure 10.4 shows the contribution of logical requirements forms to a specification (abscissa) as a function of design decisions/design triad resolution (ordinate). The flow of design decisions and information progressively transforms the perceived needs (at the top) into fabrication and assembly definition (at the bottom), at which point the design has become a completely constrained specification of solution.

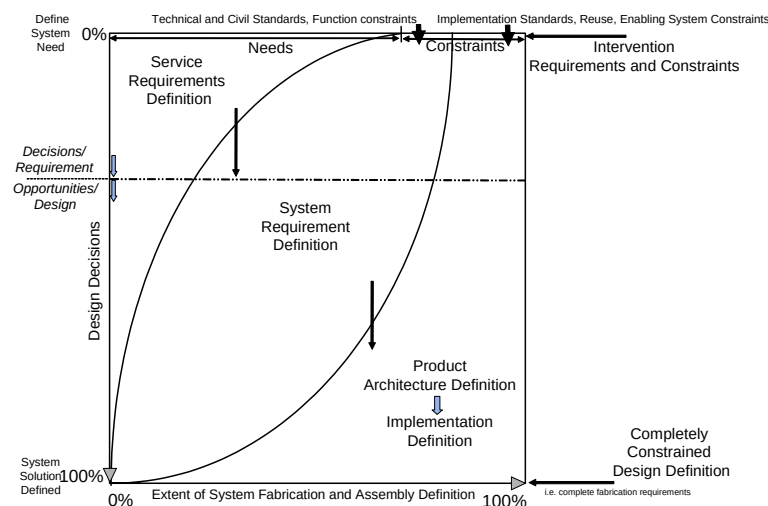


Figure 10.4 Requirements Contribution to Specification as a Function of Design Decisions

The definition of what is required at the top of Figure 10.4 is not wholly confined to service. Practicalities and business drivers introduce stakeholder constraints:

- on the solution function (technical and civil standards on functions, behaviour, protocols);
- on the form (re-use, enabling system constraints, a product family solution);
- on the implementation detail (technology standards, fabrication codes).

These constraints confirm the illusion of a solution-independent statement of need, even at strategic levels of system capability. The realities of operating environment, technology readiness and legacies of past solutions unavoidably channel viable solutions into some solution necessities.

Thus it is that Smart Acquisition's solution-averse culture was a pendulum that initially swung too far; its doctrine became a qualifying goal that could be an encumbrance to project

progress. Description of effect without some presumption of solution can become abstract to the point of inconsequentiality.

The abscissa of Figure 10.4 conveys the changing balance of contributions from requirements definitions of service, product function and product form as they contribute to a contract-enabling system specification at any selected level of system detail. As the wavefront of decisions advances according to Figure 10.2, the service need is transformed into product function. From this product function gives way to product form containing ever greater implementation detail. At any instant in this decision-making flow (exemplified by the horizontal dotted line) the status of the design derives from all three regions of requirement/design definitions shown and a specification of the systems is an aggregate of the three.

The ordinate implies but does not represent time as either a sequence and/or linear scale. It is an axis of design decisions, and these decisions may be subject to re-visitation. They are continually open to fresh examination and to re-appraisal. The dotted line can move up as well as down as options are explored and alternatives weighed against one another. The information transformation arrows are thus shown as bi-directional, vertically exchanges having an overall downward emphasis.

Formal titles such as URD or SRD convey no rigorous logic regarding content or purpose. Certainly they are not defined by MOD with sufficient clarity to satisfactorily resolve the interpretations of Figure 10.3-1 and -2. Where and how the constraints at the top of Figure 10.4 find their way into requirements and design models, and how the logical function models and physical form models of processes work products coexist and relate is open to a spectrum of interpretations.

Superficially, Figure 10.5 appears the same as 10.4; however, it stresses the impact of the authority, agreements and hierarchical approval conveyed in Figure 10.3-3, and also a relationship with cardinal life cycle control gates.

Figure 10.5 presents the balance of requirements responsibility as DEC, IPT and industry responsibility for the definition of requirements is conferred from one to the other through agreements and the specifications that these rely on. Whilst this is not the only strategy for creating effective man-made systems, it does presently dominate the way in which complex systems are commissioned, designed and constructed in the government funded arena. Whereas bottom-up, technology-led creation of systems can be a valid strategy for the lower reaches of Figure 3.2, it is the complexity of solutions, the constraints of the operational environment and the attendant business risks that strongly mitigate against this approach in the higher reaches of Figure 3.2.

Thus a drive down from service need, through product function, then product form and finally a solution resolved into elements of implementation technologies, is the strategy that governs much of the engineering of high-complexity defence equipments. Responsibility is conferred down the chain of contributions in Figure 10.5, with influence associated with the three bands of DEC, IPT and industry responsibility. The horizontal dotted lines here represent agreement interfaces. Specifications at each of these interfaces assemble the result of decisions about need and solution from the stratum of responsibility above.

Each specification at each agreement interface is ideally comprised of non-redundant, complete and contiguous descriptions of service, product function and product form. The balance of their contribution is a function of the progress through the life cycle. The particular dotted lines in Figure 10.5 correspond to MOD's cardinal life cycle trading/agreement milestones of Initial Gate, Main Gate, Equipment Contract Specification (beneath which is the detail of industrial supply chain specifications).

Although DEC's are predominantly associated with service, and IPTs with function, and industry with form, Figure 10.5 shows that DEC's, IPTs and industry each simultaneously hold direct or conferred responsibility for effect, function and form, though with markedly differing emphasise.

Thus the focus on SRD (confirmed by the AOF Glossary's definition of the IPT as 'the body responsible for developing the System Requirement Document') fails to give due credence to

the IPT contribution to either refining service need or defining product form. The implication is that, by attending to an SRD, definitions of implementation are somehow being avoided.

This is a fallacy. The complementary relationship of function and form in the design triad of Figure 4.9, and the necessity for product form to contribute to a specification that enables IPTs to trade with industry, illustrate that IPTs cannot avoid responsibility for physical views of architecture and implementation; at least not until they are contractually delegated to industry. IPTs may, and indeed do, draw on industry to assist those decisions (for example through technology development contracts), but take them they still have to take them, as Figure 10.5 conveys.

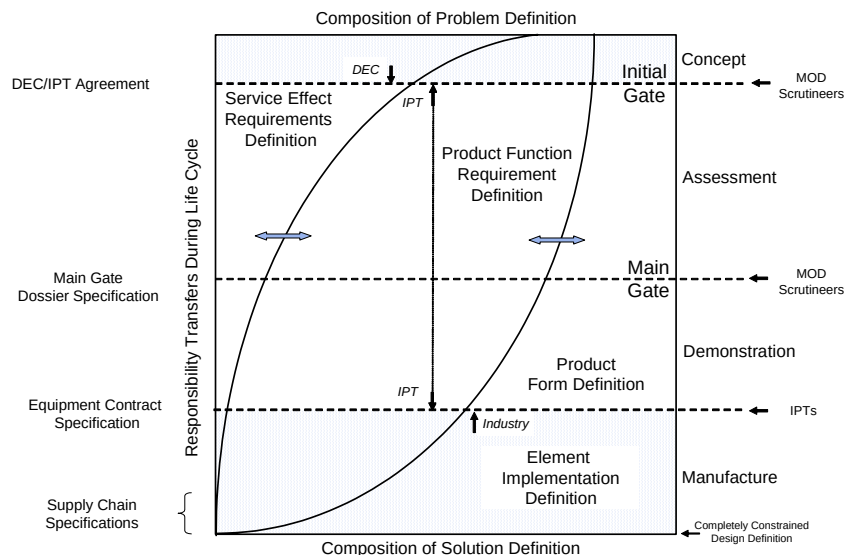


Figure 10.5 Balance of Contributions from Different Requirements to Specification

From the outset of Smart Acquisition, despite the inescapable reality in Figure 10.5, any IPT-owned, formulated description of physical architecture has largely been ignored in the AMS and AOF, and has been seen very largely as an industrial responsibility. This is despite descriptions of the role of an Architectural Design Document (or Definition) in the *DERA Systems Engineering Reference Model* [Arnold, 1997], in Steven's *Systems Engineering* text [Steven, 1998], and obliquely in ISO/IEC 15288; all of which have otherwise influenced Smart Acquisition's incorporation of systems engineering practice.

Whether the IPT explicitly assumes responsibility for physical solution definition or not, the reality is that system models and design decisions about product form have to be built and taken in concert with the function-oriented SRD models. These design decision about the real-world product form (the physical manifestation of architecture) establish a strategic, mutually consistent set of requirements that complement the SRD. They are design decision about physical structure, e.g. a crew of 110 personnel, the use of particular tactical communications equipment, the adoption of particular power plant technology. Their subsequent imposition on industrial suppliers ranges from strong implementation constraints to regions of outright solution.

It is a grave source of risk to split responsibility for product function and product form at the same strata of responsibility and decision making, i.e., at any horizontal level in Figure 10.5. Any situation that in any way splinters the responsibility for system functional models from responsibility for corresponding physical models is a threat to solution coherence. It is a separation that has profound implications for government/industry partnership.

A failure to properly recognise the IPT's responsibility for a physical architecture during Assessment and Demonstration points to a weakness in systems engineering approach. Guidance such as the 'IPT should submit the complete SRD to potential suppliers with an invitation to each supplier to proffer a complete solution architecture, and associated draft

Equipment Contract Specification as the basis of their offer ... [for] the equipment supply contract' [AOF, 2007c] is at least confusing and at worst misguided.

Other guidance appears better founded. For example, 'concurrent engineering of the URD / SRD / ECS set (and development of associated architectures / models) is the norm. The evolution of each is iterative and recursive. The schedule needs to accommodate this 'systems engineering' approach. Centre requirements activity in the Concept Stage on the evolution of the URD, in the Assessment Stage on the evolution of the SRD and in the Demonstration Stage on the evolution of the Equipment Contract Specification' [MOD, 2007f]. However even this pays no regard to the early ADD, despite the fact that the technical content of an ECS absolutely contains ADD information, as illustrated in Figure 10.5.

Thus an emphasis on the IPT's role and responsibility with regard to an ADD is not just a desirable but a necessity. Explicit definition of ADDs and IPTs' early responsibility for them would strengthen the trading interface with industry.

This is an advancement of IPT technical practice that appears inevitable. Indeed, it is an issue that may already be being ushered in, in the guise of architecture frameworks. Their recognition of the complementarity of system functional and physical viewpoints dictates the need for models of a physical views that comprise an ADD. This is considered again in §10.4's investigation into *Designing Architecture*.

10.3.3 Requirements and the Trading Culture

This section closes the consideration of requirements advancements by reflecting on how, beyond their technical role of achieving unity of technical purpose and trading agreement, the techniques and methods used in requirements definition might also influence the commercial culture between trading partners.

Enabling Acquisition Change recognised that the 'IPTL is the commercial interface with industry' [EAC, 2006, p5] and a need to amend MOD instructions to 'reinforce' this. Thus, beyond defining necessary and sufficient technical requirements and enabling trade with industry, requirements must help establish conditions for competition, and engender commercial relationships that are equitable and incur the lowest possible costs.

The management of such a relationships owes much to the way in which requirements are employed during the system life cycle. The definition and evolution of requirements is a reflection of acquisition culture and not simply an outcome of engineering process conditioned by the characteristics of enabling requirements tools.

The evident trading purpose of requirements – to contribute to the construction of a thread of information and decision-making across organisational interfaces, and so bind the needs of the stakeholders to the delivery of an optimal system solution – is complemented by the need of acquirer and supplier organisations to protect the interests of their stakeholders; the taxpayer in the case of government, the investor in the case of industry. In the midst of many stakeholders, the ultimate beneficiary of the system service, the military user, has been partially relegated to the role of bystander as requirements are used to build commercial defences in a competition of government/industry contracting. Faced with a situation of commercial accountability, the IPT as customer agent can be drawn into defensive commercial strategies that impede rather than enable effective outcomes.

A prevalent defensive requirements strategy has been to build a wall of mutual protection by both trading partners. This game plan arises from a zero-sum culture, giving rise to trading boundaries that are largely governed by immutable specifications and subject to change-resistant, highly protective modification practice.

The win/lose mindset has employed requirements as commercial weapons on both side of the trading boundary. This has harmed the effectiveness of a systems engineering approach, and Smart Acquisitions and DIS have both emphasised the need for a change in this culture. Thus, whilst 'ensuring the retention of systems engineering capability in the UK is in general important ... [it] needs, for implementation, to be underpinned by different behaviours in both us [MOD] and industry ... It also means demonstrating that we can be disciplined enough not to ask for overly demanding specifications at first delivery, while being agile and flexible

enough to be able to react when industry does produce sensible proposals for rapidly taking advantage of new technology' [DIS, 2005, p.64].

Periods of confrontational procurement behaviour (from both sides) have characterised the environment in which systems engineering has evolved. However, government/industry trading often leads to programmes that are developed by few major primes who oversee chains of suppliers to meet the specialised requirements of one or (at best) very few dominant customers. These circumstances produce a distinctive relationship of mutual interdependence between customer and supplier that is characterised by reward structures, risk management cultures, commitments profiles, accommodations and constructive deviations that would be an anathema in other trading relationships.

All this requires both formality yet flexibility. They are required in agreements, communication and information handling throughout the system life cycle if complex, integrated interaction between a wide community of stakeholders/interested parties and their supplier base is to be achieved. In this complex of commercial agreements and risks it is important to establish appropriate team behaviours and culture. They are crucial to effectiveness in the competitive, yet fundamentally cooperative environments that IPTs sit in the midst of.

Past action has seen commercial battle lines drawn, with evermore requirements thrown into the defences. Requirements became a "weapon of choice", used offensively and defensively by acquirers and suppliers alike.

Every requirement increases costs – its obligations having to be first met and then proved. Some requirements become unnecessary as fresh one supersede or influence them. Progressively design options can diminish rapidly. Unreasonably and unnecessarily, solution trade space contracts – sometimes disappears. Ultimately, excessive fortifications of requirements can deadlock the commercial trading situation.

In general, requirements grow in number with mistrust; they reduce with risk- and gain-sharing. A regime of rigid, defensive specification-dominant acquisition is a typical indicator of a failed culture of cooperation. It is at odds with the nature of risks in many military equipment acquisitions, and with the assertions in Smart Acquisition and DIS.

Time has exposed the weakness of this strategy. The autocratic requirements bandwagon is over, and inflexible and defensive requirements definition, by either government or industry, no longer suits the defence equipment trading situation. Requirements are, and only should be, enablers – never prescriptive impediments.

One consequence of all this has been that requirements are conventionally constructed from words and numbers – the raw material of contract law – in a form that supports pass/fail demonstration of conformance. It has resulted in text-based statements of 'singular/atomised' specification that 'describe a single characteristic' [AOF, Oct 2007, Requirement and Acceptance, Review Checks] and these can unnecessarily harbour ill-understood dependencies.

A prescriptive text-based requirements are an outcome (even a questionable legacy) of a specification-driven acquisition strategy rather than of a requirements-enabled design approach. Its win-lose nature pushes systems engineering toward threshold representations of system utility and pass/fail criteria [Arnold, 2005]. The need to have enforceable definitives in a contract between customer and developer leads to a 'binary' nature in specifications. Slightly above the defined threshold the system has fully acceptable utility (or value) for that particular parameter. Slightly below, and the system (ostensibly) has no utility. This is shown to the left in Figure 10.6.

This binary characterisation of systems that intervene in an essentially analogue real world is both unrealistic and (legal conventions of society aside) unnecessary. It overly constrains the 'acceptance space' and impedes agreement resolution between the users, who need an acceptable system solution, and the developers, who are often pushing the boundaries of technology and wish to provide a solution without being drawn into undue risk. Mil Std 961 suggests no alternative approach: 'requirements shall be worded to provide a definitive basis for acceptance or rejection'. Similarly, the AOF advocates the use of 'objective Measures of Effectiveness, setting threshold and objective values', and it advances criteria such as 'is it

numerically quantified or is its satisfaction ‘measurable’? [AOF, 2007c, Requirements and Acceptance] without qualifying how this might be achieved or what its consequence might be.

A longstanding augmentation that softens text-based specification has been to grade and weight individual requirements according to their importance. The ISO/IEC 15288 trio of ‘needs, wants and desires’ conveys this notion (centre of Figure 10.6) and begins to soften the discontinuity in the utility/measure relationship. Such weighting is the basis of MOD ‘s advocacy for distinguishing Key User Requirements in order to emphasise obligatory need from inspirational ‘wants’.

Text-based models with their threshold criteria weakness are, however, but one approach to specifying future systems reality, as indicated in the model cycle of Figure 3.9. The system performance measures of the engineer and the value domain of the user can be related by other modelling notations that more closely characterise the analogue reality of continuous function utility measures. This is illustrated by the simple utility function to the right side of Figure 6. Not all utility curves are smooth; some possess discontinuities driven by nonlinearities in the problem or technology spaces. Nevertheless, specifications expressed in terms of utility functions are consistent with requirement management practice able to ‘underpin different behaviours’.

The more effective modelling that is required, the tools to manage this and practitioner understanding of multi-variate analysis all present their own problems, but influences on requirements of this nature are evident.

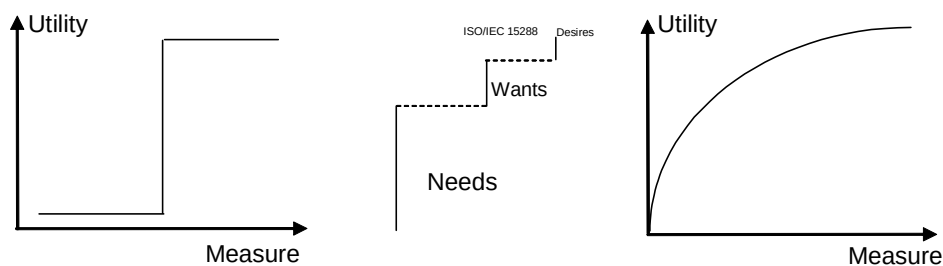


Figure 10.6 System Utility as a Function of Threshold and Continuous Function Representations

From the foregoing, it becomes evident that requirements models can be ‘executed’ to explore trade-offs, compromises and alternative solution tactics will be a powerful step. The distinction between requirement models and design models again blurs. It then becomes a short step from requirements to the models defined in architecture frameworks, and these figure prominently in the next sections on the design of architecture.

10.4 Designing Architecture

10.4.1 The Nature of Architecture

The second of Singleton’s cannons – ‘a consideration of function independently of mechanisms’ (mechanisms in his human factors text meaning inanimate, physical equipment) – is concerned with how the creative bridge is built between required service and physical implementation; that is, with the design of system or, more specifically, product architecture. It is concerned with the middle level in Figure 4.7 (the Formal Cause)

In 1974 Singleton’s argument was that, in a climate of abundance in technological advancement and offerings, the ‘development of [diverse] engineering technology.... [has] provided a whole variety of ways of solving a given problem ..[and has] given rise to the need for functional thinking.’ The reasoning behind this is that a rich range of implementation technologies opens opportunities for alternative candidate implementations strategies and

'increases versatility to do the initial thinking about a design problem in functional terms ... Too early a switch to thinking in physical terms ... may unnecessarily restrict the solution of the problem' [Singleton, 1974] – a 'solutioneering' bias that Smart Acquisition sought to counter nearly a quarter of a century later, particularly through the viewpoint discipline required in the development of URDs and SRDs, described in §10.3.

The definition of system viewpoints, and the way they lead to the construction of views comprising well-specified system models, is an area of methodology (method, rules and tools) that has been refined since the early 90s, especially by the IT/software communities. It is exerting an increasing influence on the discipline of systems engineering. At the same time, it is opening up opportunities and also presenting challenges for IPTs.

The IT/software engineering approach to system design that has been evolving has unhelpfully (from a systems engineering perspective) been referred to as 'architecting' by its communities of practitioners. The 'architecture approach' has evolved rapidly and with a marked degree of independence from mainstream systems engineering thinking. A result is that IT/software communities have mapped out a similar ontological landscape having a different set of lexical signposts and with a different emphasis on the attention to technical transformation and to work product definition. This has led to IPTs being faced with two maps with which to navigate systems – one from systems engineering and one from architecting, particularly 'enterprise architecting' – and these present areas of inconsistency and even contradiction. Already IPTs, particularly those engaged in information-intensive C4I/NEC projects, are operating in a region of some methodological discord; on in which systems engineering practices (as advocated by DIS) and architecting practices (as promoted by MODAF and, in 2006, DGIInfo) are not yet well aligned or integrated.

The "traditional" systems engineering discipline and the more recent, implementation technology-driven IT/software engineering disciplines are both concerned with common systems reasoning issues. From their different regions of application, each brings important understanding and practicalities. However many of the differences and emphases of approach are undesirable and unnecessary. A workable resolution of disconnects that have arisen from this technical and cultural segregation of two engineering communities is important for future IPT practices. It augurs well that, being cross-disciplinary, systems engineering is familiar with adopting from the implementation technology-based branches of engineering, for 'many of the formalisms now used in systems engineering had their roots in software engineering' [Rechtin, 1997, p.89].

The critique in this section advances that architecture sits naturally in an established picture of systems engineering. Nonetheless, for this relationship to be fully effective issues of technical inconsistency must be resolved. It is advocated that accommodations need to be brokered with both communities in order that different approaches to common problems are harmonised. IPTs should see no distinction or disconnects in their conduct of systems engineering and their application of an enterprise architecture-oriented approach.

The basis of the challenge for IPTs is illustrated in Figure 10.7. It presents the complete span of military system structure categorised in accordance with the classification in Figure 3.2. It shows that, over the region of networked military capabilities and the C4I assets they depend on, IT techniques and tools are becoming influential. In some quarters they are a mandated part of defence acquisition practice. For example, the US information technology management reform in the 'Clinger-Cohen Act' was a catalyst for DODAF becoming a part of DoD acquisition planning [DoD, 2007, Executive Summary].

Networked military capability now increasingly looks like, and has to be designed according to, principles and practices evident in business/enterprise systems; that is in systems that are 'goal-seeking, self-directing, self-organising and capable of learning' [§3.4.2]. This is clear to many IPTs; during the interviews 30% expressed the view that the discipline of enterprise architectures used for IT-based business organisations is directly applicable to the business services of the military capability they acquire, including the "multi-Service information flow" of NEC. They believed that their analysis and synthesis of solutions according to systems engineering discipline would benefit from disciplined modelling of enterprise architectures and thereby provide IPTs with better understanding.

Thus, given the major impact of NEC on MOD acquisition, the ‘architecting’ of information technology-intensive military (enterprise) systems needs to sit consistently with the systems engineering discipline that IPT are expected to observe according to Smart Acquisition and DIS. Any dichotomy cannot be resolved by rejecting one approach in favour of the other. Both are necessary, and the increasing overlap of systems engineering and architecting/enterprise architecting in the mid- to upper-region of Figure 10.7, makes a mutually consistent assimilation of their principles and practice an urgent necessity for the defence acquisition community.

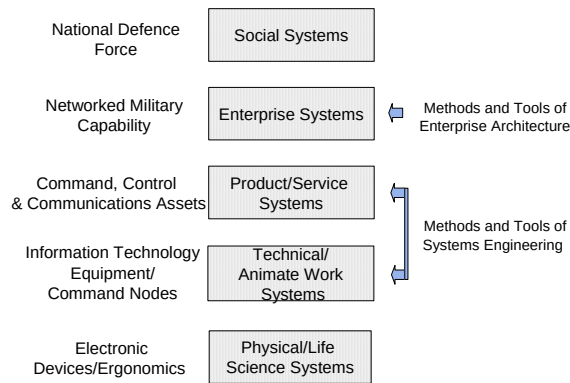


Figure 10.7 Defence Systems Hierarchy According to Fig 3.2

Although disciplined descriptions of systems' architecture descriptions are gaining wider, if cautious support by IPTs, the maturity and delivered value from current architecture standardisation and methodology remains in question. As observed in §2 there is still an international debate about “what actually is architecture?” Some basic ontological issues are still confused, and the lack of a stable foundation has led to variations of interpretation and marked differences in associated methodologies and standards.

Drawing on this investigation's model of systems engineering, and particularly on its basis of systems reasoning, a harmonisation of the current systems engineering and IT/software engineering dichotomy is proposed. It ‘attempts some conceptual cleansing of the confused field of information systems and information technology’ (to quote Checkland's chastisement on other related matters [*Checkland, 1999, p.A4*]).

Notwithstanding the ontological differences, one of the seeds of discord is straightforwardly lexical. In the '80s the IT/software engineering communities drifted into a close association of the term ‘system’ with the computer, its resident software and its network elements. In doing so they lexically painted themselves into a corner. For them, the hugely rich and powerful concepts associated with the term ‘system’ were eroded and devalued by an almost indivisible association with a specific, albeit important, class of objects. (In mitigation the systems engineering community has done much the same with its exclusive association of ‘system’ with ‘product’ or ‘equipment’). IT practitioners were faced with the need for a rich lexical replacement for system and by the end of the '80s ‘architecture’ was appearing as the ‘get-out-of-jail’ expedient. ‘Architecture’ provided the IT community with a term that it could rally around and use as a banner for its evolving principles and practices. For IT and software engineering, the word architecture would now assume a breadth of meaning to rival that of system. As a result, architecture has now become a fashionable and overused term; one that is proving elusive to distinguish from the semantic richness of ‘system’.

This lexical influence from IT/software engineering is evident in the Collins definition of architecture (in which this investigation's interpretations are set in parentheses):

‘architecture *n.*

1. the art and science of designing and superintending the erection of buildings and similar structures
 {the creativity, heuristics and engineering practice of design and technical supervision resulting in man-made systems}
2. a style of building or structure

- {a recognisable pattern or pro forma of system composition and arrangement}
- 3. buildings or structures collectively
 {a quality or attribute of systems that conveys composition and order}
- 4. the structure or design of anything
 {the composition and rational arrangement of a system}
- 5. the internal organization of a computer's components with particular reference to the way in which data is transmitted
 {the information technology viewpoint of a computer described according to system form and function}
- 6. the arrangement of the various devices in a complete computer system or network
 {the information technology viewpoint of a computer network as a system}' [Collins, 1991].

Meaning 1 confirms architecture to be a body of practice. It is applied to the design and supervision actions of particular classes of structure, such as vessels, buildings, cities.

Meaning 2 conveys that architecture can manifest itself as patterns of significance and value. Meanings 3 and 4 convey architecture to be a collective attribute of systems.

In contrast, Meanings 5 and 6 present a contemporary information technology and software use of the term for computers (plus the software representations of data, processes and control that they host) when considered in system terms.

For systems engineering and, as the definitions above suggest, generally for the systems reasoning mind, it is axiomatic that architecture is an attribute of system that characterises a system's order. In the IPTL survey 67% spontaneously identified architecture with structure, with 50% referring to product structure and 17% translating this directly into consequent project structure.

Architecture is thus commonly understood as a description of the composition and structuring of a man-made system; of order that arises from intent and directed design. This is in conformity with the IEEE definition of 1990: 'the organizational structure of a system or component'. That is, a factual listing of parts and their organisation or relationship [IEEE, 1990].

A decade later, however, the influential standard IEEE STD 1471 had evolved this definition into 'the fundamental organization of a system embodied in its components, their relationships to each other, and to the environment and the principles guiding its design and evolution' [IEEE, 2000]. This definition had moved beyond an objective description of a system-of-interest, extending it to include the setting, if not behaviour, in an environment of operation. It also introduced the notion of the decision (or design) rationale behind these descriptions. In doing so, it began to equate architecture to design actions and the discipline that governs them.

Testing the lexical purity of 'architecture' in IT/software texts reveals that the word's removal by synonymising it with 'design' (noun and verb) or 'system' or 'structure' or 'organisation' (as in organised structure) invariably leads to no discernable distortion or loss in semantic intent – indeed it can often be deleted without any loss of meaning (a hallmark of a term in fashion).

In a related engineering domain, Chang's analysis of the principles of mechanical systems design never uses the term architecture in a paper on form, function and behaviour (the latter meaning effect as used here) [Chang, 2000]. He relies on 'system' and 'design' to completely convey the concepts of structure and model. Similarly other engineering disciplines faced with complex design and solution descriptions have paid the architecture term little or no attention.

An additional further lexical influence from IT is seen in the prevalence of an architecture approach to the design of information systems for 'enterprises', i.e. for business organisations. This has given rise to the term 'enterprise architecting'. Since almost all organisations depend to an extent on the design of IT systems, enterprise architecting has a rising influence on engineering systems design.

As yet, no dictionary, not even an American dictionary that might have sympathy towards 'verbizing' nouns, recognises 'architecting' as an action or activity. As Rehtin observes,

‘“architecting” refers only to the process. Architecting is an invented word to describe how architectures are created’ [Rechtin, 1997, Preface].

Architecting’s positive discrimination of its terms from its older systems engineering cousin has been fuelled in some measure by practice distinction and vocational exclusivity. This promotes practice, proponent and practitioner status alike. There is a suggestion of disciplinary competition, leading to partisan positioning by ‘architecting’ with respect to ‘systems engineering’ in an effort to gain (in Kuhn’s terminology) jurisdiction. A systems engineering and architecting distinction would thus appear to arise partially from values, beliefs and ideas, and hence to be culturally rooted.

As observed in §3.3.1, etymologically architect derives from the Greek for ‘the director of works’. Hence architect is associated with technical leadership and connotes seniority as much as skill. Typically it is used in the singular form and is less prominently associated with a team activity.

Architecting practitioners have thus elevated the most strategic-thinking, high-level design to be architecting, relegating all else to be termed ‘design’, which is then a subordinate/subsequent action to that of architecting. It is an exaltation that is taking root. In the IPTL interviews 50% equated architecture to a high level abstraction of system information that conveys a holistic view, such as NEC, and its decision-making pathway to lower level detail.

According to this model of architecting practice, systems engineering is concerned with the conduct of implementation-related design, and architecting with the strategic decision making across all engineering contributions. Architecting then becomes the hub of design. It is a model with seductive promise to those mired in IT complexity, but is in essence barely more than a re-titling or re-stratification of the recursive transformations described by systems engineering

Systems engineering and architecting paradoxically govern design of the range of system structure that respectively lies above implementation and systems engineering – in both cases, from the system-of-interest down to the point at which responsibility is transferred to the engineering skills and technology of implementation media.

In it, the lexical map of IT subordinates the role of systems engineering to that of implementing the system elements. Architecting takes precedence (temporally and in ranking) with respect to systems engineering. It establishes ‘architecting to be “the front end of systems engineering” ... setting up the necessary conditions for systems engineering and certifying its results’ [Rechtin, 1997, p.213]. To the systems engineering mind, this would appear to usurp, or less emotively to encapsulate, the essence of systems engineering within an architecting discipline.

Rechtin’s advocacy for an architecting discipline proffers that ‘generally speaking, engineering deals with measurables ... and hard sciences ... a deductive process. Architecting deals largely with ... nonquantitative tools and guidelines ... [and] is an inductive process’ [*ibid*, Preface]. He presents ‘architecting’ to be the role having decision-making precedence over systems engineering, with decisions governed by heuristics allied with esoteric gifts. This mode of thinking has been persuasive and pervasive.

Further, Rechtin sees architecting to be ‘distinguished from systems engineering in its greater use of heuristic reasoning, lesser use of analytics, closer ties to the client, and particular concern with certification of readiness for use’ [*ibid*, p.21]. To Rechtin and to a community of advocates and practitioners, architecting appears as a super-refined layer of systems engineering lying on the top of Figure 2.5, acting as the ultimate layer of client contact and ‘certification’ of the built system.

This is a position that has been assumed by leading architecting practitioners. In his influential ‘4 + 1’ model that influenced software methodology in the mid ‘90s, Kruchten sees systems engineering serving only to address the physical (form) view of a system and thus to be concerned with implementation issues [Kruchten, 1995, Figure 1].

Levis, in his award-winning trio of INCOSE papers, also advances that ‘system design and implementation will involve the specialization of the architecture [sic.] and the addition of all of

the details of the design so that the system can be manufactured. This specialization process is the task of the general contractor or system engineer' [Levis, 2000, p.229].

For many in the IT/software design community, rather than being a system attribute, architecture has now come to mean the system itself. As DoD puts it, 'the term "architecture" is generally used both to refer to an architecture description and an architecture implementation' [DoD, 2003], i.e. the implemented, real-world system. This does not sit comfortably with the semantics of systems engineering. Confusion between system and architecture (and arguably between systems engineering and 'architecting') urgently requires resolution.

Systems engineering and architecting govern design of that range of system structure that respectively lies above implementation and, paradoxically, systems engineering – in each case from the system-of-interest down to the point at which responsibility is transferred to the skills and technology of implementation media.

As a contribution to the evolving "architecture question" and to a "conceptual cleansing", the premise advanced in this investigation is that architecture should be seen as the descriptive essence of systems; in no sense is it the system itself. Architecture is the totality of every possible communicable view of an actual or conjectured real-world system: the summation of all possible transmissible models that inform of the existence of an object. It is therefore an abstract notion; a set of descriptions of the nature, arrangement, workings, holistic interaction opportunities and, additionally as preferred, the rationale for the existence of this order.

The ontological argument used to support this premise (and to add to the many presently being debated) is as follows.

- In conformance with §3.4.3:
 - It is proposed that, in the continuum of the manifold of reality, regions exist whose limits form enclosed surfaces. Such a multi-dimensional surface delineates an object.
 - A corollary of this proposition is that any such region can be completely comprised of subordinate non-overlapping regions or objects.
 - These subordinate objects mutually interact according to a universal order which is detectable as properties or characteristics at the holistic object boundary.
 - The greater the number of interactions involved, the greater the opportunity for novel outcome; and this leads to properties at the holistic boundary that are not detectable at subordinate boundaries.
 - This is the necessary and sufficient requirement for a system to exist in reality.

A system therefore exists in the manifold of reality outside of human existence and not as a result of cognition.

- In conformance with §3.5.2:
 - Human perception of real-world systems is structured according to stances (and other related cognitive primitives). They are hugely simplified but workable approximations of objects in reality, i.e. models. (These models physically actually exist as biochemical orderings that, according to evolved rules, mimic the complexity of reality).
 - These models are discernable by human consciousness and can be manipulated according to a wide range of other cognitive primitives.
 - One set of primitives handles the faculties of communication; a specialised and important set that deals with intra-cognitive model transfers between individuals, and that extracts/inputs information out of/into these models according to percept.
 - The information in the models is structured as a concordance of percept and stances.
 - The totality of this information creates the awareness of system architecture.
 - This information is coded, decoded and navigated naturally by the individual (by pre-wired mental processing), see Table 3.8 and 3.10.

Architecture is thus the outcome of individual human awareness of a system and is abstract.

- In order to communicate and share awareness of architecture:
 - The models are tuned to accommodate the limitations of human senses, the intentional reasoning of stances, and the mental processing and storage capacity.
 - Communal, standardised conventions are employed in order to materialise the cognitive models; this communication may benefit from intermediate storage, e.g., documents, electronic storage, etc., as necessary.
 - The information in these models is coded, decoded and navigated communally according to shared rules or standards. These range from natural language rules to mathematical rules, see Figure 3.9.

The semiotic, semantic, syntactic and pragmatics rules for shared awareness of the architecture of information-based systems are now commonly termed architecture frameworks in IT and software engineering disciplines, and increasingly so in systems engineering. How these rules are defined is the subject of §10.4.3.

As advanced in §3, information content is bounded in practice by the span of interest (from the system-of-interest down to some level of detail), and by the direction and profile of concern, i.e. viewpoint. These are substantially independent degrees of modelling freedom, and when applied in combination they radically reduce the information handling challenge.

To take description resolution first; an architecture has to be defined at some level of system detail. This generally relates to responsibility down to some scale or granularity of interest: to some ranking in a hierarchy of design actions. However, it is the recursive nature in any system description that also means architecture applies at any and at all levels of detail.

Rechtin's view that 'architecting ... can be recursive ... [and] may reoccur on subsystems' [*Rechtin, 1997, p142*] is misleading – it is absolutely recursive and it does occur 'on subsystems', or for that matter at any degree of granularity. The equivocation in this statement is a concession to the hierarchical supremacy of architecting that characterises much of his argument. Subsequently he admits that 'the architecting process ... continuously progresses from the abstract to the concrete in a steady reduction of abstraction ... until the lathe stops turning or the final line of code typed in'. His elegant, in-passing capitulation confirms that architecture may be described or modelled at any chosen level of detail and eventually, in a summation of contiguous bands, must describe every level.

The resolution or scale of consideration of architecture thus relates to responsibility of the observer and, in that regard at least, architecting does have a localised correspondence to practitioner status or more specifically status in a particular organisational hierarchy.

In contrast to level of interest, it is observer concerns that direct and filter a distinct consideration – a viewpoint – of a system. That viewpoint is described by appropriately selected system models.

Each system model contributes to building a multi-dimensional picture of a system's presence in reality. The degree of fidelity, detail and model dimensions contained in a system description will always fall short of absolute rigour, but a workable approximation is struck that balances the information value, information management and required effort against the individuals' cognitive abilities and the effectiveness of a communal database.

For each individual system and their given situation, some modelling dimensions are highly significant and influential contributors to meeting their concerns; others may be valid but have minimal or no relevance to the observer. In this way, the selection of useful and practicable models depends on the nature of the system, on its environment, on the concerns of the system's observers or stakeholders, and on the effectiveness of the modelling notations that can inform those concerns.

Thus, despite being faced with the impossibility and the resource impracticability of building a 'fully' populated multi-dimensional model space (§3.6) that has total coverage in its representation of reality, a framework of sparse but sufficient system models is the system engineers' compromise. Though each model may be far from ideal in isolation, each selectively contributes to understanding, and at the chosen level of detail, convey system's form and function, and thereby its chosen design.

In the multi-dimensional panorama of system model space, specific observer positions, or regions of affiliated observer positions, have through experience been become associated with practicable and effective sets of system models. Because of this, viewpoints may be defined independent of an individual system-of-interest, since they relate to an external framework of common concerns. In practice this is limited by the modelling conventions that can be applied to the system, but this is a matter generally influenced by system class (as in Figure 3.2), application domain or predominant implementation technology. A viewpoint may thus apply to many systems, e.g. IT systems, human work group systems, transport systems, financial systems.

From a defined viewpoint and for a specific system-of-interest, the observer sees a unique view or description of the whole system. Each view is necessarily a partial representation of the complete system. It abstracts essential detail in the form of a model, or more typically a related set of models, and is a representation of the holistic system (down to some level of detail) seen from the perspective of a set of related issues or concern of an observer.

Viewpoints thus filter information, restricting to that needed by observers to reason about and resolve their concerns. They specify the conventions for the composition and construction of corresponding views by defining the models and conventions to be used. They act as templates or profiles that filter communicated information, as described in §3.5.1, to accommodate the limitations of the engine of systems reasoning. They are a revealing manifestation of primitive cognitive mechanisms.

Quite inappropriately, in ISO/IEC 15288 architecture is explicitly associated with one process: architecture design. It is a legacy of the weakness of late 1980s system life cycle modelling concepts that were inherited from the software thinking behind ISO/IEC 12207 [ISO, 1995].

In principle there are as many viewpoints as individual system stakeholders/concerned parties. In practice however, as advocated in 15288 Clause 5.5.2, stakeholders are generally identified in classes according to their loci in a frame of observation. This draws together related viewpoints into descriptions of common.

This aggregation into viewpoints brings a life cycle management benefit. The models that form the views are generally populated and exploited by different parties at different periods in the life cycle. Viewpoint definition thus brings greater consistency and discipline to the modelling of systems, and this is soon becomes evident. In the IPTL interviews, 45% advanced that MODAF provides formal rules and techniques that can link the interests of different parties, 35% that architecture relates and communicates systems views and concerns as seen by different parties.

Albeit presently with a strong IT/C4 bias, architecture frameworks are a basis for defining, populating and employing a matrix of system models with workable level of representational rigour and consistency regarding all aspects of a system, all stakeholders, and all points in the system's life cycle. In passing, this emphasises the tautology in the term 'model-driven systems engineering' – all systems engineering, indeed all design by humans, is based on models; it could be no other way.

As a postscript, the significance of a natural system's perceived order (and in the case of animate systems, its fitness) may be defined through observation and deduction according to the laws of physics. However, whether natural systems can be associated with a notion of an architecture that includes design rationale must rest on one's epistemological foundation.

It is important to recognise that the foregoing critique on the nature and description of architecture is ultimately founded on philosophical and psychological propositions. These may be interpreted as realism or simply as expedients of instrumentalism. Whichever; if this approach is to advance the practice of systems engineering, the salient issue is that architecture managed in accordance with these conventions must offer useful predictions. This is considered next – firstly, with regard to how it accords with the principles and practices of design, and then in §10.4.3 with how it explains what has been (and continues to be) largely empirical development of standardised matrices of system models, i.e. architecture frameworks.

10.4.2 The Pattern in Design

In common use as a verb, 'design' is defined as 'to work out the structure or form of (something), to form or conceive in the mind; invent, to intend, as for a specific purpose'. As a noun it applies to the 'work products' or artefacts that result from these actions; that is, 'the arrangement or pattern of elements or features, an end aimed at or planned for; intention; purpose' [Collins, 1991, design, meanings 1, 3, 4, 7, 11]. Purpose, intention, mind, conceiving, working out, arrangement, pattern, order, structure and elements all illustrate why architecture and design are such closely associated, often-interchanged terms.

Created order, structure and patterns are outcomes of conducting design. Crucially it is the order, structure and patterns in design actions that are the genesis of these attributes. Design, even in nature, is not an outcome of anything other than a highly structured, causative sequence of actions (intentional or not); as shown in Figure 3.3.

This section therefore considers this causative structure in man-made system design – the Formal Cause – and how it is employed in systems engineering in order to create purposeful outcomes. By understanding how the principles in §3, §4 and §6 are transformed into recognisable strategies of design practice it becomes easier to fathom what is 'going on' during design, and how individuals and teams might discipline themselves better to practice it. It is how system models are defined and resolved that shapes the patterns of design action, and that builds layers of architecture detail.

§10.3.1 emphasises, as does Hall [Hall, 1989] and so does Hitchins, that engineering design is a problem-solving paradigm. According to Finklestein, 'problem solving may be defined as that form of activity in which an organism is faced with a goal to be reached, a gap in the 'route' to the goal and a set of alternative means none of which are immediately and obviously suitable' [Finklestein, 1983, p.216]. It is a path of resolution followed in conformity with the guiding criteria of a goal (the stakeholder requirements/user needs/intervention intention) subject to the constraints of viability (the opportunities and reality of implementation technologies).

Descarte saw design (and much else about cognition) to be rule-based. In his *Rules for the Direction of the Mind*, he points to design as an undertaking requiring the mind to 'examine each and every item which pertains to our design in a continuous and uninterrupted process of thought and to include all of these in an adequate and orderly enumeration' [Descarte, 1625, Rule VII]. His call for orderly (not reductionist, nor sequential) process is qualified by his 'causal adequacy': that the reality of complex phenomena, whilst not appearing as logically consequent from 'simple' phenomena, can be sufficiently understood by 'orderly enumeration'.

Rechlin's architecting presents matters in a somewhat less orderly manner: as 'an eclectic mix'; an amalgam of 'creative and heuristic approaches that combine art and science'. For him, design is to approach 'qualitative problems using guidelines, abstractions, and pragmatics generated by lessons learned from experience, that is, heuristics' [Rechlin, 1999, P.25]. Many of his heuristics are barely more than adages. His interpretation imputes long-held beliefs that design relies solely on transcendental effects derived from the gifts of natural aptitude and intuition, and sharpened by the conditioning of experience. In contrast, even Norris [§10.1] offers a clear pattern of 'guess a design, assess it, modify it, reassess it and repeat this until you are satisfied'; though his 'guess' does intimate a neatly packaged transcendentalism.

Undoubtedly design does benefit from enigmatic creative flashes – the remarkable leaps of intuition and creativity that a favoured few disproportionately possess: the gifted 'right stuff' of §6.4.2. – but this does not mean they are inexplicable. The basic mechanisms of holistic thinking, boundary building, model visualisation, abstraction handling, information hiding, association, elision, dialectics, scaling, pattern recognition, extrapolation, interpolation, pattern matching, and other techniques in the armoury of systems reasoning are for the most part identifiable, relatable and (especially when performed in the context of a team undertaking) manageable as engineering practice and discipline. They underpin systems engineering. They are part and parcel of design.

System design (verb) is about the formulation and concordant resolution of as many system models as is necessary in order to describe a real-world system's composition and ordering

such that it may then be realised and intervene in some future reality according to its originating intentions.

The processes of design thus build models of that realisation and future reality – as descriptions of intended intervention, as descriptions of the function of some intervening agent, and as descriptions of the physical composition and ordering of that agent. These models are progressively resolved, each with the other according to ever greater detail, until the risks of achieving a viable and valid outcome diminish to an inconsequential level. This is the profound message in the design triad of Figure 4.9 and in its recursive resolution shown in Figure 4.8. These two figures convey that the actions of design successively dissect down to a foundation of fabrication processes: to Rechten's 'turning of the lathe and writing of the code'.

In addition, they also bring about the associated pyramid of system build and test [Figure 4.10] that explains how the whole is to be constructed from the minutiae of elemental parts, i.e., how real-world structure is synthesised from technologically diverse 'raw materials'.

System design (noun) is, as a minimum, the accumulated set of concordant descriptions or models of the system. Optionally it may include the rationale behind the model building and the path of decision making followed in order to reach a consistent, relevant set of models at a particular level of detail. This is barely indistinguishable from the architecture definition of IEEE 1471. It underscores architecting to be an intimate constituent of design (verb), and architecture an essential quality of a design (noun).

This concordant resolution of models follows a sequential decision-resolution path; one that is continually revisiting different levels of modelling detail, past decisions and preceding lines of decision rationale. It is shown as a path in three planes of freedom in Figure 10.8. In it, the front face equates to Figure 4.14, the left face to Figure 4.12 and the base to the documented rationale of design.

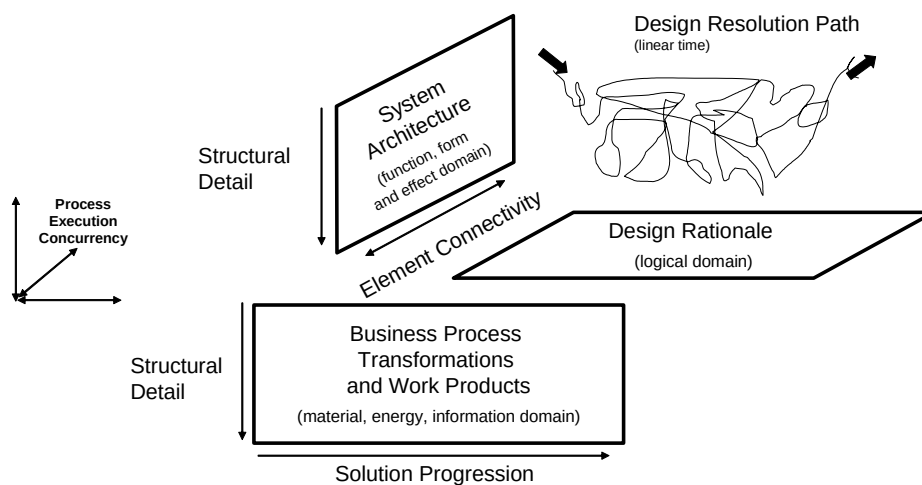


Figure 10.8 Problem to Solution Resolution

The step-by-step decision-making flow of the design of man-made systems – Dawkins' 'observed fact' of 'progressive improvement' – is well recognised. Finkelstein and Finkelstein offer a review of the design process that describes strategies and methods drawn from multiple texts [Finkelstein, 1983]. In doing so, they make repeated reference to systems engineering (which they equate to a set of methods). They observe that 'design consists of a sequence of stages starting from the perception of need and terminating in a final firm description of a particular design configuration. Each stage is in itself a design process and is an iterative sequence of steps' [ibid, p.216] – a life cycle management message of §5.3.2.

Engineering design is therefore much more than a resolution arising out of a sequence or iteration of process transformations or work product outputs. As conveyed in the process relationships of §4, it is also a matter of recursion that resolves transformation and work

products definition at different scales. 'The design process is ... repeated sequentially in a number of stages, proceeding from a global view of the system ... to progressively more localised considerations, and from an abstract and fluid description to a concrete and firm one' [*ibid*, p.217]. Abstraction favours functional descriptions; firm detail favours implementation descriptions.

Despite his Rule VII, it is widely held that Descartes does not give due account to system design's dependence on holism-based synthesis. Unfairly, just as against Descartes, this accusation is often levelled at systems engineering. Certainly, design analysis is concerned with decomposition and reduction, but crucially it is *equally* concerned with design synthesis with composition and holism. Finally, when the design triad of Figure 4.9 has been navigated to a satisfactory resolution, then commensurate contributions of effort and creativity will have been expended from both analysis and synthesis.

The essence of the design of architecture is thus:

- 1) Starting from some initiating information (often, but by no means exclusively, an understanding of intended effect), the mutual resolution of the three models (or composites of models) in Figure 4.7 at a consistent level of structural detail;
- 2) For each part of the resulting form model, if the risk/benefit decision to conferring responsibility to other parties is not satisfied, then a fresh echelon of effect, function and form detail [Figure 4.7] is explored and resolved;
- 3) Repeat 2 'until you are satisfied', i.e. the risk/benefit eventually favours conferring responsibility to another party through an agreement based on the models defined (now seen as requirements) – in this manner all system elements are determined and appear as another party's system-of-interest;
- 4) Repeat 2 and 3 until descriptions are reduced to being a description of a technology fabrication process.

This sequence amounts to a hierarchy of correlated transformations of systems descriptions over multiple levels of structural resolution (scale), as conveyed in Figure 4.14. From this the complete description of a solution evolves. The strategies used to achieve this and the factors that influence their choice, and the tactical shifts that come about as a result of prevailing and changing factors, are what distinguishes the interweaving of different patterns of design for different systems.

The resolution of each design triad is based on conjectures that derive from:

- precedence (what has and has not worked before; styles; stock-in-trade gambits);
- patterns (recognisable functional or material structure seen to work in different situations and having an equivalent architectural form in a different circumstance);
- equivalence (known, technologically realisable characteristics and interactions that are aspects of the outcome sought);
- incremental variation (empirical deviations that explore successful solution directions).

Three common strategies for tackling each subordinate level of design triad are detailed in Figure 10.8. It sets out patterns of design with regard to system element creation. These are, left to right:

- Accumulative design, suited to well-precedented architectures or open architectures. In this, a strong and stable framework defines models of function. It favours standard units of composition influenced by COTS, re-use and plug-and-play opportunity. It works well with standard interfaces and protocols or where previously successful approach can be repeated;
- Contiguous design, which is influenced by precedence in design, known architecture and, if required, an incremental delivery strategy. Contiguous design schemes starts with the element(s) that are best understood, most precedented, most strategic and most stable. It builds from this secure foundation, progressively assimilating constraints as composition is defined. It follows an architecture pattern already

understood, which may be re-used or be a standard, and enables novel technology options to meet standardised element requirements.

- Accretive design characterises situations, where the architecture is being explored in the light of component opportunities. Architectural forms and concurrent increments of solutions are explored and designed across the whole solution space according to the evolutionary nature of an accretive design scheme. Unprecedented solutions can be explored, with trade-off iterating between architecture and technology options.

The model building and reconciliation strategies employed to resolve the design triad metamorphose through the life cycle as the purpose and nature of the models employed changes. Early on, higher-level abstraction addresses feasibility. The further through the system life cycle, or the higher the implementation risks, the deeper is the focus on modelling detail to explore technology.

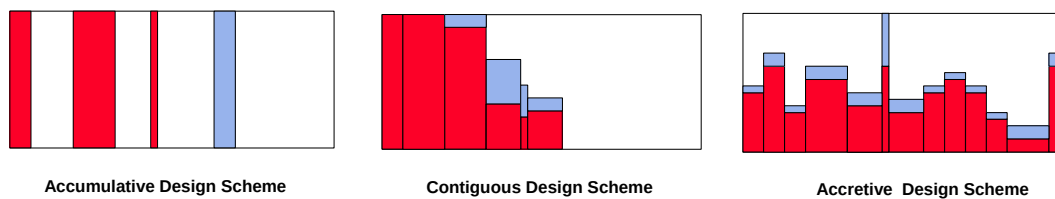


Figure 10.8 Design Triad Resolution Schemes

Rechtin rightly questions the generality of Brook's statement that "the best architectures are the product of a single mind" [Rechtin, 1997, p.45]. He transforms this statement into 'the team of a single mind' or 'a single vision', though makes little in the way of objective explanation of how this may come about. This team design mind is an enigmatic concept. It is akin to distilling multiple contributions to become commonly held team knowledge, assembled from models according to diverse viewpoints.

The trade-off and inconsistency resolution between models occurs at every level of system detail. It leads to creative tension between divergent exploration to open up design opportunities, fresh ideas and preconceptions, and the convergent constraints of a solution that accommodates perceived limitations and past technological dependencies. It is a balance played out at every level of compositional detail. At every layer of structural resolution, composition is evolved and resolved. Evolved, in that heterogeneity of diverse function and/or form is analysed in the detail beneath; resolved, in that a homogeneous composite is synthesised, when viewed from a level of detail above. Resolution of this analysis and synthesis is a concurrent multiple scale issue.

There is thus a paradox in the recursive nature of transformations and work products. Every level of structural detail can only be understood in the context of every other level. Optimisation at any one level of detail can compromise others. Design thus seeks to achieve coherence and reconciliation between levels of architecture by pursuing a discipline of compromise.

Whilst human intellect appears able to maintain multiple scales of visualisation to some extent, there seems to be no way to 'average' or otherwise algorithmically combine in a linear theoretical fashion the non-linear contributions of hierarchical structure. Drawing on the systems reasoning of §3.5.1, a necessary design strategy to cope with the challenges of information capacity and processing is:

- to move recursively up and down levels of structural granularity, continually changing resolution and at the same time field of view in order to resolve patterns of composition and connectivity;
- to encapsulate meaningful and mentally manageable regions of functional and physical solution coherence.

In this way, order, structure and coherence in models is an outcome of repeated rescaling of downward analysis or upward synthesis. As the DIS puts it, 'one rule of thumb used in

industry is that “a good systems engineer understands the details that are being address two levels below him and one level above him” ’ [DIS, 2005, p.61]. To this it might be added that ‘this rule applies at any level of structural detail concern’.

Different patterns generally emerge when the system is viewed at different levels of abstraction. Deduction according to these diverse patterns helps counter any dependency on linear analysis/synthesis assumptions that might suggest proportionality, superposition and reversibility relationships. Necessarily, complexity and novel properties are a product of non-linearity. The goal of complex system design is to create evident homogeneity from innate heterogeneity.

In their survey on the five leading software architecture design approaches, Hofmeister et al. acknowledge that the ‘design methods were developed independently, their descriptions use different vocabulary and appear quite different from each other... [they] were developed in different domains [and] naturally exhibit domain characteristics and emphasize different goals’; the ‘domains’ being software quality, industrial systems, incremental delivery, software-intensive systems and product families [*Hofmeister, 2007*].

The authoritative team of researchers who contributed to this survey endeavour to designate the basis of software system design. In their analysis of design methods for software architecture, they offer a generic approach that is derived from four leading software design methods. In the composite that they propose two of the three stances are distinguishably addressed. Their ‘architecture analysis’ models facilitate the function stance model and the ‘architecture synthesis’ model facilitates the physical stance model. These model names evoke equivalent work products from ISO/IEC 15288’s processes. However In contrast, the intentional stance is vaguely introduced and only retrospectively accommodated by an activity of ‘architecture evaluation’.

It must be acknowledged that the ISO systems engineering standard’s process titles of Requirements Analysis and Architecture Design seriously fail to convey the true nature and symmetric relationship of the processes they define. Hofmeister’s ‘architecture analysis’ and ‘architecture synthesis’ are far better candidates and could with little collateral disruption be employed to shed the baggage of a ISO/IEC 12207 software engineering titling legacy.

If this were adopted, design in ISO/IEC 15288 would apply title-wise to two processes, which would then formulate architecture to a level of detail that securely transfers matters to implementation detail. The Formal Cause layer of Figure 4.7 would thus become the ‘architecting layer’ and the misnomer or Architectural Design, which conveys the notion that design’s only creative subject is system form, would be corrected. In all other respects, the activity detail of ISO/IEC 15288 Requirements Analysis and Architectural Design could mostly live on unchanged.

The C4I/NEC region of defence system design is increasingly subject to the dual influences of a model/work product-dominated approach, as specified by MODAF (which says nothing about transformations), and by a process transformation-dominated approach, as specified in systems engineering standards (with only high-level definitions of work products) and as advocated in DIS. Neither of these representations can exist in isolation of the other. They are, as emphasised in §4.3.2, totally interdependent representations of process.

Neither the data/material/artefact transformation view of ISO/IEC 15288:2002 nor the work product view of MODAF are complete descriptions of system design, and they should not be defined in isolation of each other, as discussed with regard to process definition in §4.4.3. Proposals to resolve this disconnect have been published [*Levis, 2000, pp.225-247*], [*Arnold, 2002b, p.9*], [*Dickerson, 2004, p.12*]. However, none so far have adequately reconciled the process/activity detail of any of the standards in Figure 4.4 with the model detail of DODAF/MODAF.

Some degree of reconciliation is offered in the next section. In it the systems reasoning of §3, rationalised according to the systems engineering principles of §4, is shown to explain the practicalities of architecture frameworks.

10.4.3 System Model Frameworks

As suggested in Figure 3.9, language (in the sense of a 'systematic means of communicating' [Collins, language, meaning 4]) is the generic mechanism for achieving shared interpretations and projections of reality between minds, and thereby for building communal representations that can be acted on in concert by project teams. Fundamentally, language is about the sharing of models between individual minds; about standardised information exchanges – linguistic, symbolic and mathematical – and how these are coded, transmitted, decoded and recorded.

This last section on the design of architecture therefore looks at the norms that have been developing to achieve a sharing of prescribed system descriptions. The importance of these prescriptive descriptions for the coordination of team and inter-organisational communication is rapidly gaining the attention of system designers and acquirers. For example, almost half the IPTs interviewed were concerned to some extent with the formalisms of MODAF as a mechanism of shared representation across IPTs and their trading boundaries. 35% of IPTs advanced the point that architecture relates and communicates system views and concerns as seen by different parties, and 45% asserted that MODAF is a means of providing rules and techniques that describe the diverse aspects of a system's design to its many concerned parties.

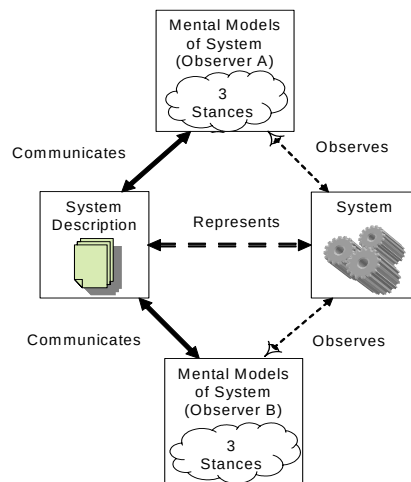


Figure 10.9 Design Team Communication

According to the propositions of Dennett, an object's design depends on the individual mind's perceived dimensions of an existent or a yet-to-be-realised system that is cognitively modelled according to three modelling modes. In order to extend the formulation and resolution of models in an individual's mind into an integrated team undertaking these models must be unambiguously communicated and rebuilt from one mind to another. This notion is conveyed in Figure 10.9, with Observers A and B exchanging information held in their stances by means of some codified, mutually recognisable communication that typically is recorded as shareable system models, i.e. documented system descriptions.

In the business information domain, and in a corresponding way in military information systems, these models have been organised according to the needs of the enterprise level of system classification in Figure 3.2 so as to form standard frameworks that regulate observer-to-observer communication. Since the late '80s these frameworks, such as DODAF/MODAF, have been addressing some important regions of systems engineering principle and practice. However, one outcome of this has been that enterprise architecture frameworks have largely been evolving along somewhat isolated and distinct lines of codification from systems engineering.

As a consequence of this, IPTs are beginning to find themselves centre stage in a divergence of technical emphasis, methods and tools. On the one hand IT architects, and on the other systems engineering practitioners, are espousing their own interpretations of how systems

reasoning manifests itself in engineering and management practice. Separately, their thinking, nomenclature and application of common systems modelling ground have diverged to become two viewpoints of one whole, and without check this will result in a branching of practices.

The specification of architecture frameworks that is being developed to suit the general structure of military enterprise models has evolved predominantly from the C4I community of interest, with its clear IT implementation basis. Starting from a DoD C4ISR Framework root [DoD, 1996a], their development continues to favour IT orientation despite the evolving frameworks being promoted as specifications that apply to military systems in general. Even after several, separately inspired framework refinement by DoD, MOD and NATO, the UK's approach is still described by the MODAF practitioner community as MoD's chosen implementation of an Enterprise Architecture Framework, confirming business IT as the continuing thrust of evolution. Not surprisingly, despite the expansionist dropping of the early, fully justified C4ISR title of DoD's modelling framework [*ibid*], and its subsequent re-branding as DoD's Architecture Framework [DoD, 2000], the underlying IT limitations persist.

Well before the '80s, systems engineering processes were defining an interrelated set of process work products that modelled future or existing real world systems in textual terms. Most of these descriptions defined views from the cardinal technical or management viewpoints and they were designated requirement or design documents. Each description provided a partial view of the product and/or service attributes of a system, and contributed to the overall impression of a complete description of a system's design (architecture); each was a facet of some consummate holism.

It requires many such work products to build rigour and adequacy into the description of that holism. Each of the very many possible partial representations serves some purpose or meets some concern regarding the creation or utilising of the system. The value of any particular view depends on the nature of system, the point in the system life cycle, the methods and tools being employed and many other factors; ultimately model value is judged by faithful satisfaction of observer concerns.

Out of the diverse possibilities of viewpoint many individual models of a system can be created. Each viewpoint defines a view of a specific system expressed according to language constraints that regulate the formalism of description. For the system as a whole, each system description/model relates consistently to every other description/model. Achieving consistency between the many possible views of a system is then a key engineering challenge.

Despite multiple interpretations of the term architecture, this foundational concept of concordant, semi-orthogonal models is pervasive. Each model in its way contributes to a unified, coherent architecture description that entails a multi-faceted or multi-dimensional complexity, and requires different viewpoints to marshal many representations and so satisfy different concerns and perspectives of diverse interested parties.

The portmanteau term 'architecture framework' is now commonly, though not exclusively, applied to schema that bring order, identity, discipline and repeatability to these system descriptions so that they can be assembled from a set of mutually consistent, standardised views. Assembled over time from existing practices of major stakeholder classes, these frameworks have been devised to be conventional references that bind different concerns to various information artefacts of design analysis and design synthesis.

In the context of military C4 systems in particular, the definition of standardised views has been seen as an important contributor to understand why systems with diverse development histories have, when integrated in new operational situations and novel configurations, proved to have indifferent levels (and sometimes complete failures) of interoperability. Integration shortcomings and interoperability weaknesses are well-recognised, symptomatic outcomes of independently managed system developments; a situation that has been compounded by growing individual system complexity and joint-Service/coalition deployments. This has highlighted the value, indeed the business imperative, of technically bridging these managerial, organisational and national boundaries by employing engineering practices that adhere to standardised system descriptions and to complementary design techniques.

The existence of standard description frameworks offers a route to easier identification of system capability duplications and of gaps across existing or proposed sets of assets, and it points to a more disciplined approach to military capability planning. Further, acquirer imposition of frameworks on competing suppliers means that equivalent system descriptions ease the acquirer's task of calibrating the relative merits of competing, proposed solutions.

Moreover, the pace of problem and solution changes means that stable, cut-and-dried operational environments are a boon that neither the designer nor the acquirer can any longer depend on. Concomitant, managerially disjointed development of military equipment has been the norm; a consequence rather than intent. It is the ultimate distinguishing feature of the so-called 'system-of-systems' challenge: the fallacious, yet well understood term that has been tellingly coined by the defence systems acquisition sector to describe the predicament of ill-coordinated, concurrent design of systems that are destined to have to interoperate.

In the face of this composite of concurrently evolving complexity and its associated intricacy of interface interactions, the definitions of standardised process transformations and text-based descriptions of work products seen in Figure 4.4 and Figure 6.2 are not longer proving sufficient. The path of 'traditional' systems engineering standardisation is failing to provide neither the detail nor rigour necessary to adequately undertake retrospective analysis, or current or future design. An echelon of more detailed work product codification is necessary in order to tackle the intricacies and particulars of discord encountered in situations of novel interoperation. By and large architecture frameworks have stepped up to this challenge – albeit with an information systems bias.

Thus, from analysing yesterday's problems, architecture frameworks are rising above the disparaging label of 'systems engineering archaeology' to become a basis for specifying and enabling many of tomorrow's more challenging solutions. It is an approach that is at one with the systems reasoning that goes to make up the systems engineering discipline.

So how are architecture frameworks and systems reasoning intertwined? The answer lies in Table 10.1. It conveys the cardinal modelling of objects and is derived from the systems reasoning arguments of §3. Specifically, it presents a construct based on the ontological and teleological schema in Table 3.8, and on the consequential model qualities detailed in Table 3.10. At a high level of abstraction it offers a taxonomy or frame of reference for all possible system models according to three primary cognitive dimensions of viewpoint, percept and stance. It is a generic framework of "the descriptive essence of systems", i.e., a frame of reference for system models that can be employed to variously build progressively richer pictures of architecture in a strongly coordinated manner.

			Mental system representations for reasoning about system intent		
			Stance (Psychological / Engineering)		
			Intentional / Effect	Design / Function	Physical / Form
Tangible system representations for communication between concerns	Viewpoint X (Concern/ Role)	e.g. X = Specification/Acquirer; Architecture/Designer; Fabrication/Technologist; Construction/Builder; Sustainment/Maintainer; Operation/User; Achievement/Project Mgr; Enterprise/Business Mgr; Wellbeing/Society.	Intervening Entity <i>(What/Who)</i>	Logical Object <i>(What/Who)</i>	Material Constitution <i>(What/Who)</i>
			Absolute Location <i>(Where)</i>	Object Association <i>(Where)</i>	Structural Connectivity <i>(Where)</i>
			Absolute Time <i>(When)</i>	Event Relationship <i>(When)</i>	Relative Time <i>(When)</i>
			Situational Transformation <i>(Why + How)</i>	State Transition <i>(How)</i>	Material Transformation <i>(Why)</i>

Table 10.1 System Model Qualities According to Viewpoint and Stance

The table shows the correlation between different individuals' percept of an object, filtered according to their respective concerns, and mapped to the stances they use to reason about its effect, function and form. It conveys the fundamental model qualities required to build viewpoint (vertically) and stance (horizontally). Whatever the viewpoint, the matrix of cells lays out the fundamental structure of information about an object, providing individual observers with a map to navigate modelling opportunities and options.

The cells build rows that correspond to the substantive, spatial, temporal and causative faculties of percept. Respectively, examples of these are:

- data, equipment, people, enterprise functional units, process artefacts, action;
- relationship, threads, configuration, connectivity;
- timing, sequence, events, state changes;
- purpose, rationale, effects, actions, outcome.

Stances are the mental representation for reasoning about an object's/system's intent. The cells in Figure 10.1 satisfy this by building columns that correspond to stance models of effect, function and form. Taken in this order of intentional criticality, examples of these are:

- military intervention, mission, use case, task;
- activities, controls, events, states, logic, protocol, functional objects;
- data entities, aircraft, radios, guns, military units, computers, communication channels, humans, command posts.

To give greater relevance and meaning to each cell, the corresponding lines of cognitive enquiry into an object are indicated in parentheses according to the primary interrogatives.

To the left of Table 10.1 the viewpoints govern how, out of the possibilities of all object representations, cell information is filtered and combined to build object awareness according to the various concerns of each observer or category of observers. They are categorised here according to the roles of acquirer, designer, technologist, builder, maintainer, user, project manager, business manager and society. Each of these particular sets of concern leads to its own profile of system description that, with respect to the observer categories exemplified in Table 10.1, address the system's specification, its architecture, its fabrication detail, its construction information, its sustainment policy, its service operation, its effective creation, its value to supplier/acquirer management and the wellbeing from its social impact.

This disciplined segregation of observer viewpoints assists different parties to generate and employ distinctly different representations of the same object. Different aspects of the same design can be worked on separately when and where responsibility and creative independence exists in the system life cycle. Frequently this aligns with a degree of temporal and even geographical separation of different concerned parties. Whatever the degree of separation, they remain guided by a common framework that directs them to resolve in an orderly manner where different aspects of the design constrain one another.

To achieve this unified independence, each system viewpoint defines a set of models of exactly the same holistic system; what varies is the particular direction and profile of concern toward that holism. Just as each of the models that comprises a view have a correspondence with differing degrees interdependence, so each of the viewpoints in a framework is in some way related to all others. In practice the viewpoints are chosen to be sufficiently independent yet sufficiently applicable to simplify and leverage different, dedicated lines of reasoning about the complete design.

Although Table 10.1 implies cells to be individual or discrete groups of models, given the many degrees of freedom that characterise architecture and the limitations of individual modelling languages/conventions, it is unlikely in practice to reduce to such compartmentalised simplicity. It is to be expected that multiple modelling conventions are required to even partially describe each cell, and that these models will spread across the system description space with no regard to the boundaries of its twelve-fold modelling

categorisation. The cells are therefore delineated in Table 10.1 by dotted lines to convey this concept.

In order to achieve a joint/communal understanding as conveyed in Figure 10.9, the information in the cells has to be capable of being expressed in conformance with tangible, codified representations of an object, according to conventions that derive from the faculties that build percept. With this link established, equivalent mental models can then be shared between concerned parties. When that communication is capable of being expressed in a permanent, tangible way, it becomes an integrated team/communal representation that can be shared across location and time. With sufficient modelling richness contributed from different viewpoints, and with model correspondence rules within and between cells established, concordance across all cells can be refined, and even new viewpoints derived.

From this it can be inferred that the term architecture framework is somewhat of a misnomer – system model framework would be a more apposite title. Even the interpretation and use of the term architecture framework by architecting practitioners has developed diversity and been driven by different persuasions. Zachman spotted the gathering clouds of discord early on. Speaking generally on the concepts and terminology associated with architecture he observed in 1992 that “there is little wonder we are having difficulty communicating with one another about information system architecture. And you can see how we could all be using the same words meaning entirely different things and having emotional difficulties over the whole problem!” [Zachman, 1992]. A decade and a half later this problem remains.

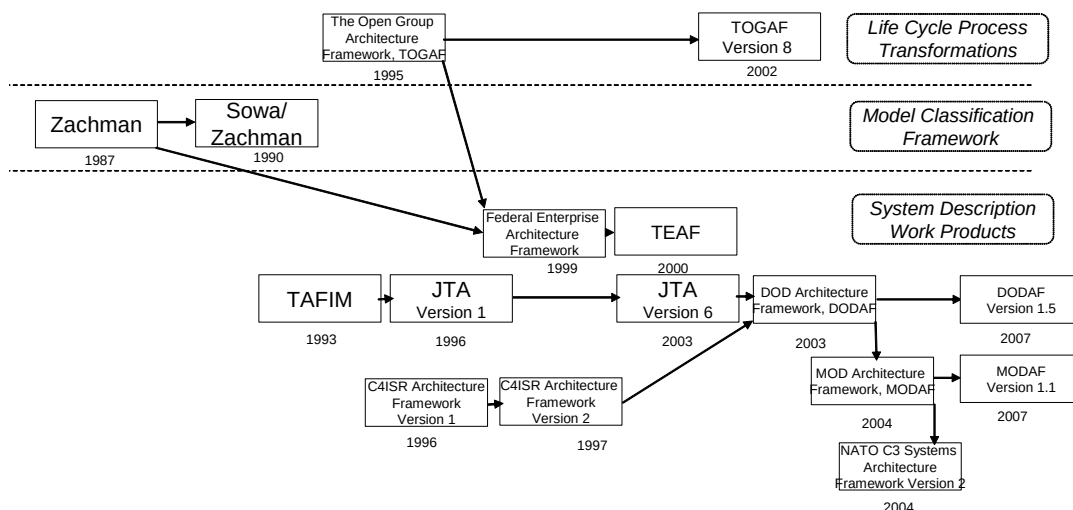


Figure 10.10 Architecture Frameworks Evolution and Influences

Figure 10.10 bring home this point. It shows a timeline of how over two decades architecture frameworks have evolved, and the strands of development and influence that this has entailed. Over the period of their evolution they have spread, one into the domain of the other.

If it were not for TOGAF, The Open Group Architecture Framework, the distinction between the systems engineering standards in Figure 4.4 and the ‘architecting’ standards in Figure 10.10 would be clear-cut. In essence the difference would be divided by approach: systems engineering standards would be process transformation dominated, ‘architecting’ standards would be process output work product dominated. In contraventions to this and despite being titled from its inception as an architecture framework, TOGAF stands apart in Figure 10.10 as an essentially IT/software engineering restatement of the transformation dominated contents of the systems engineering standards in Figure 4.4.

TOGAF is presented as ‘a framework – a detailed method and set of supporting tools – for developing an enterprise architecture’ [TOGAF, 2007]. It applies to ‘any organization undertaking, or planning to undertake, the design and implementation of an enterprise [i.e.

business] architecture for the support of mission-critical business applications, using open systems [i.e. standardised information technology network] building blocks', with the clarifying rider that 'an architecture description is a formal description of an information system' [*ibid*].

TOGAF sees that 'an architecture framework is a tool which can be used for developing a broad range of different architectures. It should describe a method for designing an information system in terms of a set of building blocks' [*ibid*]. 'Tool' and 'building block' are obscurely used terms. However a mapping of the principal graphical representation of TOGAF reveals it to be an architecture framework 'tool' that is essentially a quasi-linear sequence of system life cycle transformations set on a foundation of a Requirements Management process, as the mapping of TOGAF's top-level graphic [TOGAF, 2007] conveys in Figure 10.11. The 'building block' transformations approximate to established system life cycle processes, and the equivalences are indicated in this figure according to ISO/IEC 15288 process titles.

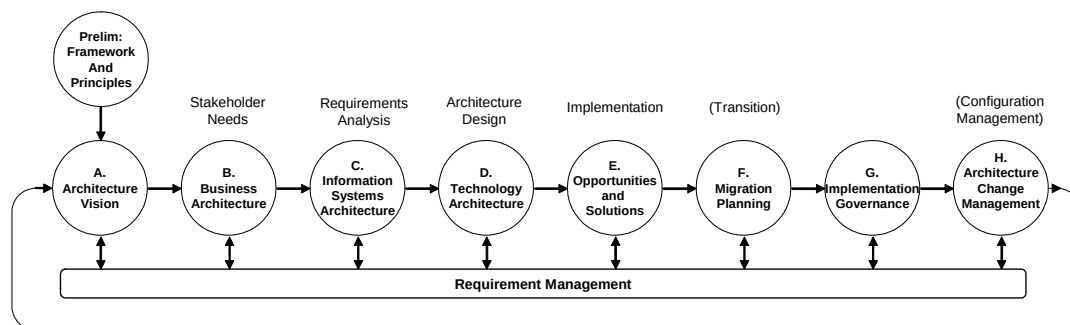


Figure 10.11 The TOGAF Architecture Development Cycle [mapped from TOGAF, 2007]

TOGAF indecisively ascribes two meanings to architecture depending on its contextual usage: a traditional view of 'a formal description of a system, or a detailed plan of the system at component level to guide its implementation' and also a restatement of ISO/IEC 42010/IEEE 1471 that includes the 'principles and guidelines governing their design and evolution over time'.

In the face of IT's growing categorisation or 'checklists' of system models according to different system concerns, TOGAF subsequently developed 17 views according to different 'architectures', i.e., viewpoints, in its Version 2.0:

- a Business (or Business Process) Architecture defines the business strategy, governance, organisation, and key business processes;
- a Data Architecture describes the structure of an organization's logical and physical data assets and data management resources;
- an Applications Architecture provides a blueprint for the individual 'application systems' interactions, deployment and relationships to the core business processes of the organization;
- a Technology Architecture describes the software and hardware capabilities required to support the business, data, and application services.

From its IT perspective, TOGAF relegates systems engineering to contributing to just one of its 17 views (that of the system's functional model) 'concerned with assembling software and hardware components into a working [IT computer/network] system'.

Stepping back in the time scale of Figure 10.10 to 1987 one finds the genesis of present-day architecture framework thinking. This innovative work by Zachman can be seen with hindsight as a system model classification scheme that structures the descriptions of IT systems. Zachman's original framework used the term 'model of the information system', as opposed to the later extended version of the framework that that less informatively and more

expansively described itself as a 'system model' [Sowa, 1992]. The original 'framework for information systems architecture' was a 5 x 3 matrix in which he asked the "what", "how" and "where" representations of a system on behalf of the owner, designer, builder, plus 'out-of-context' and 'machine language' concerns. This work provides a foundation for FEAf, the US Federal Enterprise Architecture Framework (FEAF), which styled itself as 'a mechanism for linking Agency Federal Architecture' [FEAF, 1999].

FEAF conveys the purest practical implementation of the original Zachman framework and is shown in Figure 10.11. It applies to any business organisation, considered according to its IT system. It is basically a framework that classifies models of an IT system, defined from an enterprise's needs downwards to the supplied IT components from which the system is comprised. Zachman's original three-fold interrogative model is best represented in the later FEAf representation shown in Figure 10.11 [*ibid.*], in which the rows (in contrast to Table 10.1) contribute to the concerns of the parties to their left.

	Data Architecture	Application Architecture	Technology Architecture
Planner Perspective	List of Business Objects	List of Business Processes	List of Business Locations
Owner Perspective	Semantic Model	Business Process Model	Business Logistics System
Designer Perspective	Logical Data Model	Application Architecture	System Geographic Deployment Architecture
Builder Perspective	Physical Data Model	Systems Design	Technology Architecture
Subcontractor Perspective	Data Dictionary	Programs	Network Architecture

Figure 10.11 FEAf Taxonomy of System Models for Information Systems [FEAF, 1999]

The three primary interrogatives of the original Zachman scheme were subsequently extended to include all of Kipling's 'six honest serving men' [Kipling, 1902] and this is shown in Figure 10.12 according to one of Zachman's own graphics [Zachman, 1992]. In it, motivation broadly equates to effect, activities and timing largely to function, with data entities, people and location comprising the form.

From the influences of systems reasoning shown in Table 10.1, a basis of theory to underpin this well-known classification becomes apparent. By transposing Table 10.1 into Figure 10.11 and Figure 10.12, the natural attraction of what for Zachman was an empirically-based structuring becomes clear. It is its foundation of systems reasoning principles that is the reason why the taxonomy brings discipline and benefit to IT practice. However, from this understanding the weaknesses, limitations and potential misguidance that the framework contains also become equally clear.

In fairness, Zachman never claimed rigour or deep theory in his table of perspectives and abstractions. As he put it, 'the Framework is a logical structure for descriptive representations, i.e. models or design artefacts, of any complex entity and is neutral with regard to the processes or tools used for producing the descriptions, and is not prescriptive about description rules. It is therefore a system description classification scheme that offers a checklist of candidate representations and acts as a reminder that any one representation is part of a larger set, not to be considered in isolation' [Zachman, 1987].

Zachman's framework is, at face value, a taxonomy of models drawn empirically from the similarities seen in different industries. Ungraciously, it is seen by some to be 'popular within IT architecture departments but has little hold of either the developer or user communities ... [and] can be difficult to digest and sometimes of questionable utility' [Wikipedia, 2007]. Its popularity appears to rest on its revelation to the IT community that a systematic approach to system modelling – according to an empirically derived classification structure, with a reasonably comprehensive checklist 'justified' by interrogatives – is a disciplinary advance. Thus, whereas it may have limited methodological value, its merit should be seen as cultural

and as an early conceptual reference point for later architecture formalisms. Notwithstanding, with its systems reasoning foundation and its theory-based rationality, Table 10.1 offers the potential of insight beyond that offered by the Zachman Framework.

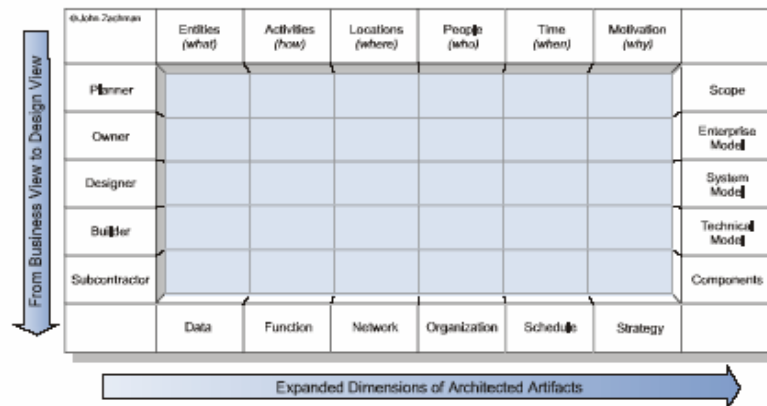


Figure 10.12 A Taxonomy of System Models for Information Systems [Zachman, 1992]

The part of Figure 10.10 most closely identifiable with architecture frameworks is the lower region. It presents the evolution of system description work products according to frameworks that structure and define system models. They arose as an outcome of a long string of international failures to create effective, even working information technology systems. It was this that eventually led to US Federal Government regulation. The formalism that was inherent in the already emerging frameworks of 'architecture' was given impetus when it was seized as a white knight to rescue a troubled sector of industry and commerce, and to act as a fresh initiative in response to the Clinger-Cohen Act.

Its clearest early definition is to be seen in the Technical Architectural Framework for Information Management (TAFIM), which provided a set of services, standards, design components and configurations for use in the design, implementation, and enhancement of information management system architectures.

The TAFIM reference model was developed by the Defense Information Systems Agency and helped to set a direction of thinking that was substantially codified in the DoD Joint Technical Architecture (JTA). Interoperation or large-scale integration was a strong motivator, and the JTA is intended to help achieve weapon systems interoperability and an open systems approach to weapons-system design. It focuses on 'systems that produce, use or exchange information' and it thus takes strongly IT-oriented viewpoints.

Many of the same IT issues are present within and between any US Federal agencies. Recognising this, the US defined a Federal Enterprise Architecture Technical Reference Model (FEA TRM) to 'facilitate cross-agency analysis and the identification of duplicate investments, gaps, and opportunities for collaboration within and across Federal agencies' [DoD, 2005]. The DoD's instantiation of the FEA TRM is the DoD TRM, and this defines strategic patterns for designing and implementing information-intensive systems. It takes the form of a standard to be followed for information systems services across DoD's organisational structure.

A complementary action was directed at the design and implementation of operationally deployed, information-intensive military systems. It was concerned with command and control, and the gathering and communication of the information that this depends on. It was directed at the wide range of network-based systems, under the umbrella title of C4ISR, that were being individually created by different military authorities with limited regard to their interoperability or relative contribution to overall capability. The resulting standardisation – the C4ISR Framework – marshalled the models used to engineer network-enabled systems.

The C4ISR Framework evolved as shown in Figure 10.10 to become the DoD Architecture Framework (DODAF). The 'DoD Architecture Framework ... provide[s] guidance for describing architectures. The Framework provides the rules, guidance, and [work] product

descriptions for developing and presenting architecture descriptions that ensure a common denominator for understanding, [and] comparing ... architectures' [DoD, 2003d, p1-1]. 'An architecture description is a representation of a defined domain, as of a current or future point in time, in terms of its component parts, what those parts do, how the parts relate to each other, and the rules and constraints under which the parts function' [ibid p.1-3].

Levis observes that 'the [DoD] Architecture Framework provides definitions a set of work products (models) that comprise three views of a system. The Framework provides little guidance on how to produce those products and is not supported directly by existing systems engineering formalisms and tools ... A well-thought-out process is necessary for mutually consistency of the work products' [Levis, 2000].

In Figure 10.13, a precise rearrangement of one of DODAF's principle figures [DoD, 2003d, Figure 3.2] conveys how its viewpoints loosely approximate to the hierarchy of concerns seen in '3 Causes' structure in Figure 4.7. The corresponding views, comprising sets of models with defined purpose and notations, are annotated to the right of the figure.

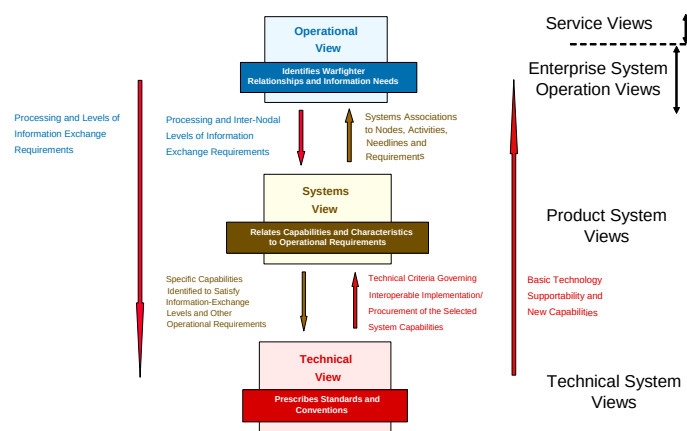


Figure 10.13 DODAF View Relationship [Rearranged from DoD, 2003d, Fig 3.2]

Uppermost in Figure 10.13, the Operational View (passably) addresses the concerns associated with the capability to deliver service (interventions/effects/missions) in an operational environment. It is concerned with the business operations expressed in terms of the enterprise's relevant (and possibly reconfigured) elements or nodes, to the tasks and activities these are required to perform, and to the information exchanges (type, nature, frequency, task relationship) they depend on.

MOD has repeated much of DODAF in its equivalent architecture framework, termed MODAF. In this the Operational Views (OVs) define the logical aspects of the system. They are a description of the key functional and information aspects. They are seen by MOD as contributing to the development of User Requirements, to the capturing of future concepts and to supporting operational planning.

The OVs comprise models that predominantly apply to the enterprise level in Figure 3.2; that is to systems that are 'self-determining ... controlled real-time by animate/human intelligence residing in individuals or in multifunctional teams according to ... a competitive environment ... [and] are goal-seeking, self-directing, self-organising and ... comprised of deployed assets and resources, controlled according to some authority structure' [§3.4.3]. As a consequence much of the operational view is associated with the goals, and the self-direction and self-organisation of the enterprise's assets and resources. The OV models relate to the functioning of a physical structure that is heavily constrained by existing military asset/control legacies.

In the middle of Figure 10.13 is the Systems View, the SVs. It is concerned with the system as a product, with its interconnections (including physical locations) and interoperation, and its performance parameters. This may describe the internal construction (key component

hardware and software; data stores, circuits, and networks) and the operations of particular systems that provide or support the service (mission) and business functions.

The SVs describe the real-world assets that comprise a military capability. They define resources, their interface and interactions, and are used to specify solutions to operating requirements specified in the OV. They therefore are a basis for SRDs and ADDs. They are substantially developed by and exploited by MOD's acquisition community, with support from its industrial partners.

The SVs shift the focus to the product/service system classification in the middle of Figure 3.2; that is to systems that are 'integrations of heterogeneous constructive structures ... characterised by their self-standing nature ... [and] are designed and built ... according to predefined scenarios and defined service delivery rules or operating procedures' [§3.4.3]. They depend on definitions of the functional and the inherited physical constraints of the enterprise assets. Some models are little more than mappings between the two viewpoints. Nevertheless, the SV models drive down into deeper descriptive detail and thus towards implementation detail. This means that they become increasingly subject to the implementation conventions/ standards/guidelines defined in the Technical Standards View.

These Technical Standards View underpin OVs and SVs are. They address the concerns about implementation conventions, standards and guidelines that govern engineering decisions on the arrangement, interaction and interdependence of system parts.

Framing all of this, an All Views is provided as the ontological and lexical map of the domain of discourse – the overview that sets the context and ties together the other views. It establishes the knowledge to interpret and relate models. This ontological map conventionally employs a pictorial representation of object classes/objects and their relationships, relying on iconic or symbolic conventions and natural language definitions. Information is key to the nature of objects and their relationships. Typically the objects are pre-existent classes, e.g. nodes of military command and control structure, or military surveillance, or interdiction or aegis, and can be represented by class icons, e.g. elements of a business organisation IT system, elements of a region of NEC.

Since the models in an architecture framework are descriptions of conjectured or observable objects, i.e. a system, each can model be mapped to the categorisation framework in Table 10.1. Accordingly, Figure 10.14 is laid out to convey each of the DODAF models as they individually describe facets of a C4ISR military system. Rendered according to DoD-defined viewpoints, they build up an adequately comprehensive representation within the limitations of the practicable modelling conventions in common use.

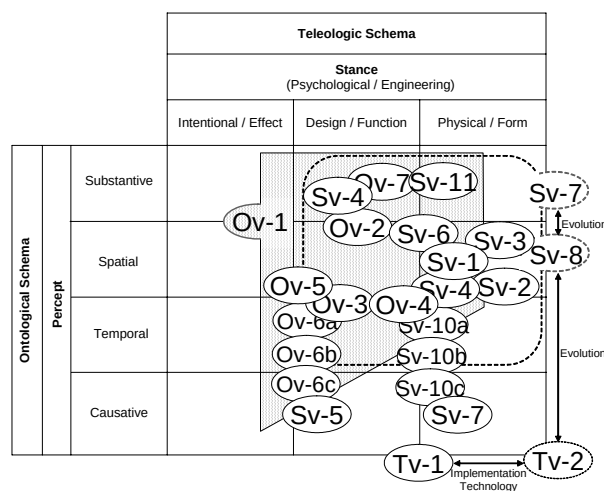


Figure 10.14 A Framework of Models of a C4ISR Military System Defined by Views Rendered According to DoD-defined Viewpoints

The oval regions represent the centres of concern and thus the foci of validity for each model, though (other than for Ov-1, Sv-7, Sv-8) in this figure they do not convey (for presentation clarity) the differing extents of effective coverage. Ideally, the spreads of each model would not only provide total coverage of the model space but would also provide an adequate degree of overlap so that reliable concordance could be established across all models. This concept is notionally portrayed in Figure 10.15 in terms of model correspondence over a region of 2-dimensional DODAF model space.

Given this correspondence feature of models, views cannot be orthogonal, nor should they be. Without elements in one view connecting to elements in other views through specific modelling rules, conventions or heuristics, it would not be possible to build a comprehensive, self-consistent set of system descriptions. Some degree of non-orthogonality is necessary; it is an attribute of a framework of system models that relies on concordance and on pair- or set-wise correspondence of models.

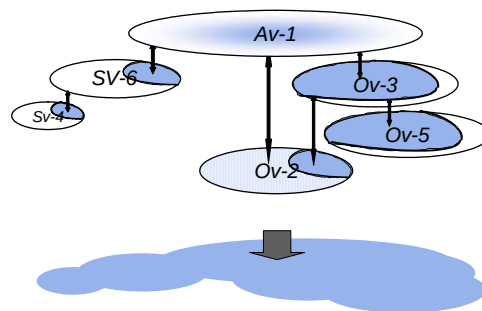


Figure 10.15 System Model Coverage and Concordance Portrayed 2-dimensionally

The regions of apparent sparsity and over population in Figure 10.14 arise partly from the way DODAF and its derived equivalents have been empirically assembled according to (local) custom and practice of diverse stakeholders concerns and responsibilities. It also and partly arises from the limitations of the effective, practicable modelling conventions available. The overlaps, gaps and inconsistencies that are evident might yet benefit from a reassessment based on a stronger systems reasoning foundation.

A region of weaker modelling is hinted at in Figure 10.14 regarding DODAF's description of system effect, with service or the operational capability envelope not well covered. However, for models to provide a satisfactory service viewpoint, as a minimum the viewpoint would need to described the immediate operational environment with somewhat similar detail and fidelity to the viewpoints of a system-of-interest considered in isolation. This would require an extension of the modelling framework to include at least the architecture of the proximate operational environment, sufficient to describe the product's intervention envelope. It would have to model capability-based planning information and details of delivered service, i.e. models that would comprise a 'strategic' view of the capability.

A Strategic View of this nature is a step that MODAF has partially taken with its extension of DODAF into systems engineering work products that attempt to define the system-of-interest environment and the resulting service interventions in it. From a systems engineering standpoint this means that 'architecture' work products move into a wider compass of engineering work products (such as in Appendix A). It would result in management work products (project management and enterprise work products) being the remaining distinguishing factor from the gamut of systems engineering work products.

However, these management work products are also the subject of MODAF models, and this transgresses the notion of what architecture is yet further. They result in a new viewpoint, the Acquisition Viewpoint, that describes 'programmatic details, including dependencies between projects and capability integration across the all the DLoDs'. The view created is concerned with the 'project system' as much as with the system-of-interest views, and appears to violate the consistent holistic system boundary principle laid down by ISO/IEC 42010, and referenced by most architecture frameworks including MODAF. Their addition takes MODAF a step towards being a repository of system life cycle work products rather than a consistent set of

system-of-interest technical models. The expansion of architecting to include management aspects of systems engineering reduces the distinction between them still further.

Inexorably, and probably inevitably, architecting is expanding into the same intellectual and practice territory as systems engineering. But it does so bringing fresh insights and some different techniques to what some term 'model-based' systems engineering.

Thus, despite a divergence of approach (certainly terminologically) the two communities who respectively identify with architecting and systems engineering are seen to employ many principles and much practice in common. Both derive from systems reasoning. It is important that they now exchange and harmonise approaches to mutual benefit. It is important they do so to provide IPT's with a harmonised way of crafting and managing the architecture of their systems, particularly C4I/NEC systems at the enterprise level in Figure 3.2.

In this region enterprise architectures readily expose the nodes of command and control that are a cardinal feature of military systems' architecture are not simply comprised of inanimate information technology resources. The ultimate decision makers in any enterprise system are humans. Ackoff reminds us 'that a systemic transformation from an animated system to a social system (where both the whole and the parts are purposeful) is required if an enterprise is to be successful in the current environment, which is characterized by an increasing rate of change, interdependence, complexity, production, and dependence on knowledge and information' [Ackoff, 1999]. Information technology-dependent assets, NEC, and indeed many complexities of present-day military capabilities, ultimately rest on the effect, function and even form of the most complex of systems element: humans.

The discipline of 'engineering' the human elements of a system – human factors – is a particularly challenging example of the general case of how systems engineering must seamlessly interact with the implementation technology-based engineering specialisms that address the Material Cause. This next section exemplifies this generality by considering how humans are factored into the systems engineering picture and what this relates to IPTs.

10.5 Factoring in the Humans

10.5.1 The Nature of Human Factors Engineering

This last subject of enquiry in this investigation is concerned with what was not evident in IPTL responses. Given the purpose and style of the interviews this could mean that all is well and the topic requires no attention. However, other indicators suggest that it is a matter that will benefit from greater systems engineering attention.

Just 15% of the IPTs interviewed made any reference to the human factor in the systems they were responsible for. A low level of concern for human factors inside, or for that matter outside systems is consistent with the survey by the Human Factors Integration Defence Technology Centre published in March 2007 which called for greater integration of human factors with the systems engineering discipline in IPTs [DTC, 2007].

Any lack of inclination towards human factors set in a systems engineering context should not be seen as exceptional. As Ramo put it in his consideration of systems engineering team characteristics, 'it is quite often assumed that narrowly conditioned engineers skilful in the details of technology, but with no knowledge of people and the workings of our social systems ... [follow] some "technology-pure" approach that disregards the human elements. That concept of the systems approach is erroneous ... [It] implies much more than technology... [to] create a unifying and integrating [project] team' [Ramo, 1998, p.21]

The conduct of engineering leads to systems that are created by humans, for the benefit of humans. Typically they draw on the capabilities of humans, and they will certainly need to respond to the needs and constraints of humans. Without an appreciation of human factors, systems engineering endeavours may well fall short of the mark, however good the technical insight and creativity that lies behind them.

Many engineers cloak themselves in a misguided security that their world is confined to the deterministic, to functions that can be objectively enshrined in equations, to laws of physics

that describe with rigour the entities they deal with. People, teams, social systems appear a world apart, scarcely an engineering-related discipline.

Thus it is that most engineers still view humans as an adjunct to equipment and not part of engineering. They are blind even to the saving compensation that the human component can bring to residual design shortcomings during system operation.

As a systems engineering point of view for analysing or synthesising solutions, an equipment-centric view can have validity – but only once humans have been rationally eliminated as contributing elements within some defined boundary of creativity. Otherwise it is a view that pre-empt a trade-off in which human characteristics could have contributed to a more effective solution. Understanding how to trade the animate system element for the inanimate alternative remains something of a blind spot for many systems engineers.

Singleton's 1970's view of the 'human operator, a natural system but a necessary component of almost all man-made systems' [Singleton, 1974], may appear under challenge. Increasingly, automated and autonomous systems are being introduced to improve performance, reduce operating costs, and operate in hazardous and remote locations. However, even with the present-day drive towards automated systems the human cannot sensibly be 'bounded' out of the solution. Behind every autonomous system the guiding hand and mind of the human can eventually be found.

The trade-off that the systems engineer is faced with is not of simple alternatives, functional like for functional like. The respective complexities, characteristics and behaviours of the animate and inanimate do not offer such symmetry. It requires a crafted, intimate blending of humans with engineered artefacts – human-technology co-operation rather than human-technology interaction; a proactive and enriching fusing rather than a reactive accommodation of the dissimilar.

Humans are seen to exhibit a distinctive and perplexing range of 'implementation constraints'. In the public mind, and with justification, system failure is frequently synonymous with human failure – with driver or pilot error, with disregard for procedure, even with corporate mendacity. Where truly does the accountability for such failure lie: in the fallible operator; in the design misjudgements in the allocation of functional responsibility in equipment-centred designs; in the failed analysis of emergent complexity in human-equipment interaction? Rightly, maturing public awareness and legal enquiry focuses evermore on these last two causes. At their heart lies a crucial relationship – a mutual understanding – between systems engineering practitioners and human factors specialists.

Both of these groups of professionals must therefore be open to, and capable of performing, trade-off between the functionality and behaviour of multiple, candidate engineering implementations and humans.

Despite the distinct complexities and gross non-linearities faced in human factors, it is not a discipline of heuristics. It is a science of structured and disciplined analysis of humans – as individuals, as teams, and as a society. It is true that the organic complexity of humans encourages a more experimental stance in order to gain understanding, and that system life cycle models need accordingly to accommodate a greater degree of empiricism. However, as engineers strive to fathom the intricacies of networked architectures, emergent risks of novel technology and unprecedented level of complexity, the inanimate is also frequently demanding greater empiricism, as seen in MOD's NITEWorks initiative.

The notion of considering humans to be parts of systems is longstanding. With engineering a thing of the future, Hobbes clarifies in *Leviathan* that 'By Systemes; I understand any numbers of men joined in one interest, or one Business' [Hobbes, 1651, ii. xxii 115]. This point of view is today more commonplace, but techniques by which systems reasoning, engineering discipline and human factors are effectively integrated are not widespread.

Though it has not traditionally been viewed so, human factors issues fit into a model of systems engineering just as with engineering based on any other implementation technology-based engineering discipline, and this is implied in Figure 2.6. Whilst the material structure of the human element is patently 'off-the-shelf technology', its 'design', 'fabrication' and 'verification' consists of activity descriptions, operator selection and conditioning actions fully in accordance with specific implementation technologies, and thereby with the Implementation

Process in the ISO/IEC 15288 model. The term Human Factors Engineering has good grounds to be used as a descriptor that can help draw human factors and systems engineering more closely together.

In an exploration of current barriers to Human Factors Integration (HFI) and in order to advance effective HFI in IPTs, the Human Factors Integration Defence Technology Centre conducted a series of interviews with IPT members (and also with industrial comparators) to draw out HFI issues, difficulties and successful strategies [DTC, 2007a]. This was almost concomitant with this investigation's IPTL interviews. It was concluded that 'the HFI process should accommodate synergies with other groups/disciplines'. In particular 'integration of human factors with the systems engineering discipline' was highlighted from the interviews as being important.

The HF DTC's primary conclusion of insufficient, timely HF engagement in projects is of profound importance. Repeatedly, failings in operational systems are traceable to poor integration of human factors into the fundamental steps of design. As the HF DTC put it, there needs to be an 'increased integration of HFI at a system level' and that this should be the result of an 'overall approach [that] needs to be *trans-disciplinary*' [DTC, 2007b].

Two issues highlighted in the report were the 'problems of integrating HFI with Systems Engineering' and the 'lack of standardisation in the HFI process' [ibid]. Both of these issues are addressed in the next section.

For the IPTs interviewed by the HF DTC there were no explicit human factors requirements in their URDs. The report concluded there is a need to promote an understanding that many requirements have a human dimension. When 'HF requirements' were expressed in specification/requirement documents, they were nebulous functional and performance requirements that lacked testability. Also, the HFI process should cover the generation and/or validation of user, system and sub-system requirements using disciplined methods and tools, such as for task analysis and prototyping.

A top-level message was of the need for a more disciplined and explicit IPT approach to human factors; one that is clearly integrated into IPT engineering and project management practice.

Over much the same period as the HF DTC survey and report, the US National Research Council of the National Academies commissioned a report on human-system design [NRC, 2007] which amongst its primary recommendations proposed that 'the U.S. Department of Defense and other government and private organizations:

- should refine and coordinate the full integration of human-related design considerations with systems engineering in organizational policies and process standards, such as the DoD 5000 series and the ISO systems engineering standard.
- The U.S. Department of Defense and other government and private organizations should revise current system acquisition policies and standards to enable incremental, evolutionary, capabilities based system acquisition that includes HSI requirements and uses risk driven levels of requirements detail, particularly for complex systems of systems and for collaboration-intensive systems.
- The U.S. Department of Defense and other government and private organizations should put the operational requirements of human-system integration on a par with traditional engineering requirements at the beginning of the initial requirements analyses to determine which requirements have priority and provide an opportunity for negotiation.'

This human factor/systems engineering synergy is given emphasis in DoD, with the term Human Factors Engineering (HFE) in more frequent use, as evidenced by Chapter 6.3 on Human Factors Engineering in the DoD *Defense Acquisition Guide* [DoD, 2002].

As with any implementation technology-oriented practice, the point at which the downward drive of systems engineering and upward drive of technology offerings meet is indeterminate in a model of hierarchical structure resolution, and this is conveyed in Figure 10.16. There is a steady downwards introduction of design decisions and implementation constraints until some point in the structural decomposition of form at which an implementation technologies-

dependence begins to dominate the further resolution of a solution. It is the point where the lintel of systems engineering in Fig 2.8 gives way to the responsibility to the pillars of implementation technology. It is when terminology, models, fabrication processes, raw materials, methods, tools all become distinguishably different.

The intermeshing of systems engineering and human factors commonly occurs at the top levels of Figure 10.6 though in many cases the decision to design-in human operators is a high level architecture decision, even a stakeholder requirement, and thus an intimate part of systems engineering decision making. The symbiotic dependence of systems engineering and human factors in this regard, plus the risks of ill-coordinated decision making between the two disciplines, make the relationship between the two areas of discipline critical to integrated team effectiveness.

The integrated team's systems engineering task is to conduct the trade off of interdependence among candidate elements in an 'ecosystem' in which each element responds to its context through some accommodation of all other elements. Their juxtaposition gives rise to transitive interactions out of which emerges the novel properties. In the case of inanimate elements, generally, the nature of interactions is substantially determinate, with reliable models that prediction of emergence can draw on. However, in the case of the human element, and especially the case of multiple human elements, trusted models, equivalences and precedence are a much greater challenge.

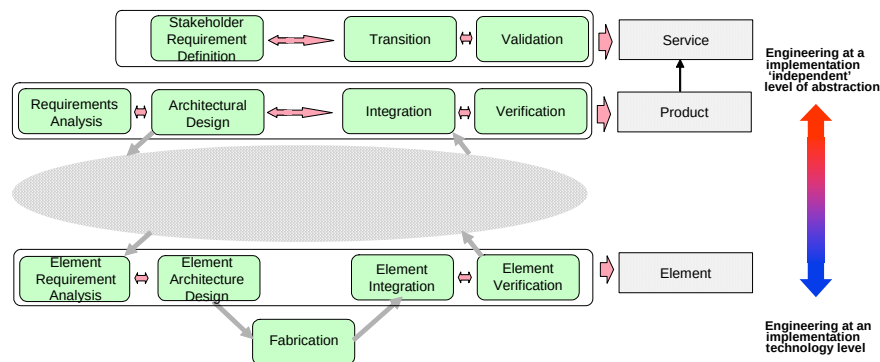


Figure 10.16 The Progression from a System View to an Implementation Technology View

Thus, as systems engineering gives way in some mid-region of Figure 10.16 to human factors, mutual understanding and recognition really count. At this changeover, technology, terminology, techniques and tools can lead to many disconnects. Moreover, it is a change from a deliberately broad-ranging, holistic point of view to a system element-focussed point of view. It is here that different cultures can throw up testing barriers.

There is evidence in the principles behind the ISO/IEC 15288 model, and the practice understanding being garnered through its use, that a recursive model of the form in 4.14 considerably assists in bridging the literal and metaphorical grey area in Figure 10.16. Equivalence may not come naturally for many, but by commuting:

- human 'element requirements analysis' into work, scheduling, behaviour, and character and cultural profile requirements;
- human 'element architecture design' into physical ergonomic characteristics, anthropometric definitions, teaming profiles, and the aptitude, knowledge, skill and experience levels of competence definitions;
- human 'fabrication' into operator selection/recruitment, physical and cognitive training and development, and interpersonal and social conditioning;
- human 'element integration' into team building and working, procedure and knowledge provision; context familiarisation;

- human ‘element verification’ into examination, monitoring, certification or licensing;

it does appear possible to use a common systems engineering ontology to reconcile the practices and traditions of markedly different and long-decoupled communities.

Effective ontological equivalence is nonetheless not the only challenge. The modelling techniques required to convey the opportunities for human function and form characteristics contributing to overall system characteristics can look unfamiliar to the engineering mind:

- alertness models;
- use cases;
- physiological models;
- workload assessments;
- task flow descriptions;
- goal representations;
- concept maps,

contribute to the communication and modelling incongruence.

As Hitchins characteristically puts it ‘the human factors specialist finds it difficult to be precise in engineering terms about matters of engineering concern, while the design engineer might like nothing better than a transfer function describing a human that he could plug into his calculations’ [*Hitchins, 1992, p.48*].

However, common ground is developing. The human factors techniques and tools associated with:

- scenarios;
- organisation node charts;
- event models,

are not only recognisable to engineers but are beginning to share the same notations and methods for both communities.

Given the potential for synergy, and the evident benefits it offers, do systems engineering concepts embodied in a model such as ISO/IEC 15288 meaningfully and effectively apply to human factors? If so, then this reference point for systems engineering should also be a touchstone for any implementation technology-based discipline. These matters are considered in the following section.

10.5.2 Humans Inside and Outside Systems

Considerable emphasis is given in DoD’s Defense Acquisition Guidebook to integrating human factors into the actions of the defence acquisition community. Its section on human factors engineering (a requirement of DoDI 5000.2) is helpfully presented in the context of systems engineering. Titled ‘Human Systems Integration’, it ‘addresses the human systems elements of the systems engineering process’. The DoD Defense Acquisition Guidebook assertion that it is a responsibility of the (US DoD) IPT manager ‘to ensure human factors engineering/cognitive engineering is employed during systems engineering over the life of the program’ [*DAG, 2004*].

This task is supported by DODAF, which asserts that ‘architectures provide a construct for describing human activities and the flow of information needed by humans to accomplish or support military operations. For most systems, humans play a significant role in how systems perform and are operated, and human factors should play a significant role in how systems are designed’ [*DOD, 2003d, p2-5*].

In a comparable way, MODAF System Views ‘address the involvement of humans in both the operation of systems and in carrying out functions in their own right’ [*MODAF, 2007b*].

Extensions to address this were introduced in Version 1.1 though are limited to the identification of command and control posts and roles explicitly in SV-1, and to the equivalent functions in SV-4.

When considering humans and systems, many approaches have only modelled people as being outside the system in question. The concerns lay with the design of the human/'system' interface; that is, the system is considered to be entirely composed of inanimate elements. The human has been considered to be external to the system boundary and thus to be viewed purely as a user of a wholly inanimate system. Unfortunately, this tradition, plus the influence of IT equipment-centric models, means that 'in MODAF the term System specifically excludes human factors' [MODAF, 2007c]. The lauded 'increasing integration of HFI at a system level' has still some way to go.

The following approach [Arnold, 2002] to modelling humans as both part of and external to a system-of-interest counters this propensity and harmonises with the provisions of ISO/IEC 15288.

As elements within a system-of-interest boundary of creativity, human operators bring some powerful properties and performance attributes. They bring intelligence to systems; the potential for real-time response to specific circumstance; adaptation to changing or unforeseen need; a natural optimisation of services delivered; and, when society requires people to integrally exercise control of a system, they legitimise system functionality. In the fields of transportation, medicine, defence and finance the evidence of these benefits abounds.

Looking outward from a system-of-interest there lays an operational environment, composed notionally of an aggregation of systems, each potentially comprising humans with a stake in the services delivered by the system-of-interest. As external system actors, their distinctive and complex characteristics compound and progressively emerge to form group and social phenomena. Amongst their ranks one finds, individually and communally, the system beneficiaries. Their needs dominate the definition of required system services and ultimately they are the arbiters of solution acceptability.

Cooperating engineers and human factors professionals must combine to address two human factors-related concerns. One is an equitable synthesis of system-of-interest properties from palpably dissimilar candidate elements, with necessary concerns for micro-scale usability in the interfaces between operator and inanimate equipment: the man-machine interface. The other addresses the macro-scale operation of this combination of system elements to form an optimised socio-technical work system that delivers agreed services into its environment of use. This product and service view of humans and equipment is the 'dilemma' discussed by Johnson [Johnson, 1987].

From a systems engineering point 'the human within the system' is of view not a stakeholder, or in any way a system user, but is an operator (using the term as in a skilled operative who is part of a harmonised technical capability). This is not to say that an effective man-machine interface is not a requirement, however, the operator does not conform to the role of a stakeholder, but to that of a functional/physical element integrated with all other system elements. Role is distinct from entity, i.e. an individual person, an integrated team or a crew.

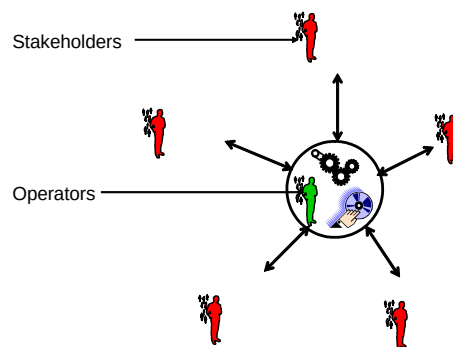


Figure 10.17 Operators are not Stakeholders

A particular instance of a human may, however, be considered both a user of the system-of-interest, i.e. a stakeholder, and as elements of a system-of-interest, i.e. an operator. In the first case the human user is a beneficiary of the operation of the system. In the second case the human is an operator carrying out specified system functions. An individual can thus be, simultaneously or sequentially, a user and a system element. This is conveyed in Figure 10.17. In it the system-of-interest is shown as comprising the three exemplar implementation media classes of hardware, software and humans that are employed in ISO/IEC 15288.

The stakeholders do not exist in isolation in the system-of-interest's environment. They are themselves parts of regions that have been designed to operate rationally and with intention: the notional systems which the environment is comprised of. They were 'designed' and 'implemented' to still act as, operators in their own system – one of many systems immediately interacting with the system-of-interest and forming its proximate context. Conveyed in Figure 10.18, this multitude of systems comprises the operational environment; it is represented by the pentagon of concurrently existing, mutually interacting systems.

Outside of the system-of-interest other systems interact throughout the life cycle. These are termed the enabling systems and they too are highly likely to comprise humans who, with some degree of weighting, are also stakeholders in the system-of-interest. This is illustrated in Figure 10.18 by a production and maintenance enabling systems.

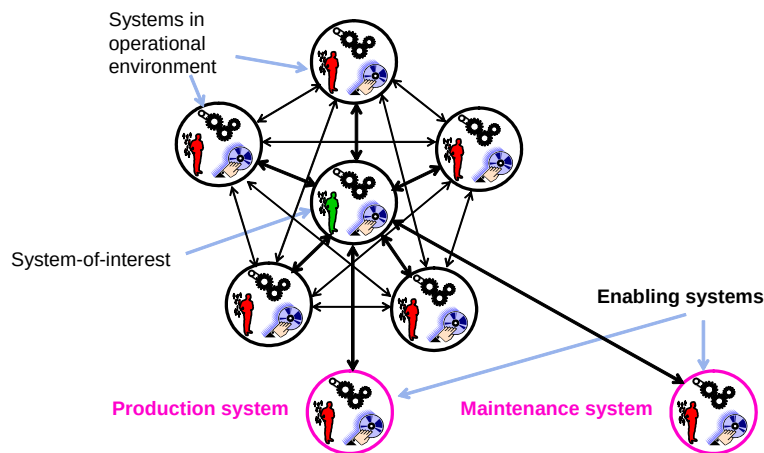


Figure 10.18 Interactions with Humans throughout the System's Life Cycle

The models employed in ISO/IEC 15288 accommodate this concept, that humans are candidate elements within a system product as well as being users external to a system-of-interest with a stake, i.e. stakeholders, in the nature and effectiveness of system services in the operational environment or in its interaction with other lifetime environments.

To distinguish the human within the system-of-interest from the stakeholder outside it, the term operator is used to designate the former. There may be multiple operators, e.g., a team or crew, within a system's boundary. The terms operator and stakeholder/user thus refer to roles and not to individuals or cooperating groups of people, see Figure 10.17. For example, a vehicle driver commonly has a stake in the transportation service that conveys them from one location to another, yet is clearly contributing to the system functionality that provides that service.

Moving from the proximate context, there is a progressive decoupling of humans from immediate interaction with the system-of-interest. Ultimately they aggregate to exert a collective influence on the intervening system. Cumulatively this takes the form of civil well-being, public interest or social values. These characteristics emerge as phenomena in user groups, organisations and society at large. This is in keeping with Parson's model of social systems [§3.4.1] and with Figure 3.2.

These phenomena introduce progressive layers of contrasting constraints on an intervening system-of-interest solution; an accumulation of needs that communally humans annex to the requirements for a system's services.

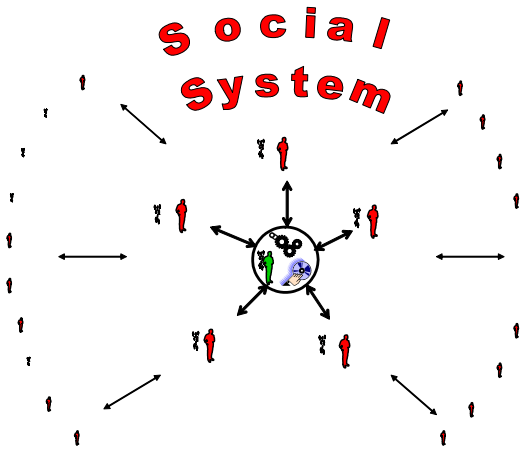


Figure 10.19 The Social System Setting

These needs and requirements instituted by people en masse are juxtaposed with their functional contribution to solutions as individuals and groups.

As an element within a system:

- people bring intelligence to systems and the potential for real-time response to the unforeseen;
- people make a system adapt to need and naturally tend to optimise the services delivered;
- society often requires people to be in control of systems and thus people can make systems legitimate according to social conventions.

As an external agent in a system's environment:

- people utilise and create systems and thus are the agents of need and the source of solutions;
- people apply and maintain systems, and thus can ensure sustainable services;
- people accept (or reject) a system and act as the ultimate arbiter of system fitness and success.

Not all the contributions are beneficial, for humans are the most complex 'off-the-shelf' system element available to the system designer. The implementation technology is already preordained, and the functional logic and parametric characteristics are already substantially fixed. In consequence they exhibit a very considerable range of 'implementation constraints' that designers must take account of when seeking to benefit from their impressive and unique list of positive attributes.

The constraints of human operators often seem to set a ceiling on system performance:

- physical limits, e.g., reach, climatic tolerance;
- mental limits, e.g., data rates, concentration, fatigue;
- non-repeatability, e.g. skill limits, attention span;
- carelessness, e.g., neglect, boredom;
- self-will, e.g., interest, personal desires, perverseness.

Aggregated outside of the system-of-interest, these characteristics of humans are equally, if not more, challenging as they compound and progressively emerge as highly complex, often unpredictable phenomena in user groups, organisations and society at large.

The task then in human-centred design is to employ techniques that leverage the positive features and minimise the negative characteristics of humans as they form a part of, or externally interact with, a system. Figure 10.20 illustrates progressive layers of constraint on product solution that humans can confer as they switch from being system elements to environmental influences. It also illustrates the accumulation of need that, individually and communally, stakeholders can demand of the requirements for a system, and these can arise from every stage in the system life cycle.

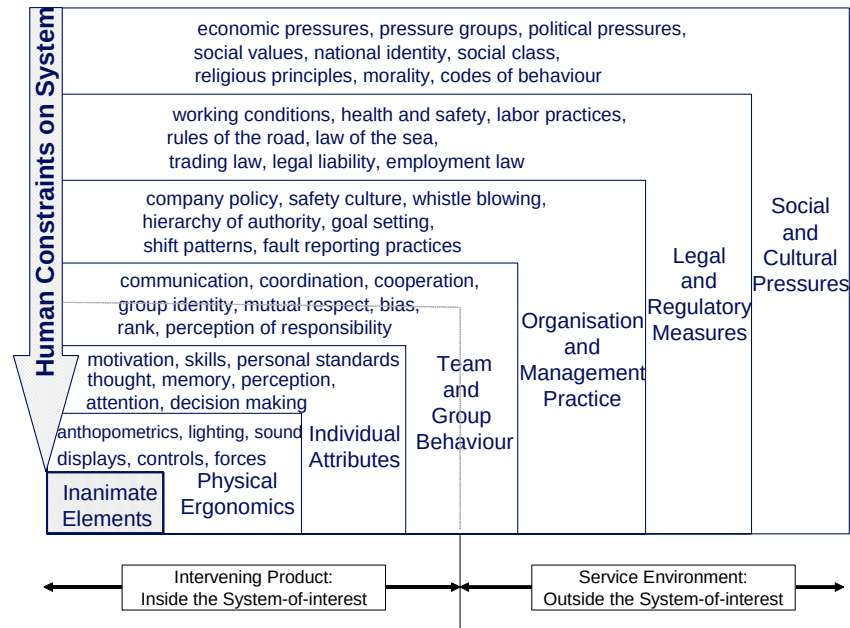


Figure 10.20 A Hierarchical Model of Human Influences and Interactions with Inanimate Elements

Figure 10.20 is adapted from Moray's 'generic hierarchical systems oriented approach to analysis and design' [Moray, 1994, p70] and offers a human factors view of hierarchies in an analogous way to Table 3.2.

The implementation technology base of ISO/IEC 15288 accommodates the primary branches of engineering and, to a large extent, applies to the science (or maybe engineering) of humans when they are considered either as elements of the bounded product, or as part of the environment in which its intervening service are delivered.

The ISO process assessment model for human-system issues [ISO, 2003a] employs models of human-equipment relationship that are compatible with, and link to, specific human-centred descriptions and techniques employed in ISO/IEC 15288. It provides a broadly based human-system (or more correctly, human-equipment) process model, detailing the criteria of effectively instantiated processes that account of people and their interaction throughout the system life cycle. The model is structured as an 'overlay' to ISO/IEC 15288 and is intended to augment it for systems that have a significant reliance on the human element. It is presented in the form of a process assessment standard that complies with the provisions of the ISO process assessment model [ISO, 2003b]. The processes are therefore described in terms of their purpose, outcomes, practices and work products. This permits them to be readily used for capability evaluation.

The human-system model employed has four views of system and enterprise that complement ISO/IEC 15288. They are described in terms of the activities that constitute four areas of processes, see Figure 10.21. These processes address:

- Human Factors in the Life Cycle – issues related to people engaged in the stages and enabling systems across the life cycle. These are related to anticipation and tracking the particular needs of each stage in the life cycle. The intent is to build an effective path of life cycle contributions according to human-system issues.
- Human Factors Integration – issues related to effectively integrating people with the system-of-interest. The processes here are concerned with strategy, procurement, planning. This ensures that human-system issues are addressed by the appropriate stakeholders, at the optimum points in the design cycle. It reduces life cycle costs by ensuring that ‘design for people’ is used in the organisation.
- Human-centred design – human factors engineering to the engineering mind. The processes here are technical, concerned with requirements, design and evaluation. This enables user-centred technical activity to deliver Quality In Use.
- Human Resources process – the provision of competent staff. This is concerned with selection and training, and with responding to technological change. This area resolves issues concerned with the ‘implementation’ of the human part of the system-of-interest. It ensures continued, timely delivery of the correct number of competent people required to deliver effective service.

Figure 10.21 illustrates how these four well-established areas of human centred concern relate to the basic structure of ISO/IEC 15288. It indicates the affinities between the human-centred processes in the model and the technical and management processes in the systems engineering model. It serves to demonstrate that, in accommodating the nature of the least engineering-like implementation medium, it should be possible to harmonise systems engineering with any implementation technology-based branch of engineering.

As a complementary line of reasoning, Figure 10.21 illustrates that systems engineering can be metamorphosed into the practice and custom constraints of other disciplines. In other words, it systems engineering offers an engineering meta-model that unites diverse implementation technology-based engineering disciplines.

A side issue of Figure 10.21 is that it illustrates the validity of joint capability assessment models comprised of ‘horizontal’ systems engineering and ‘vertical’ engineering disciplines, with CMMI exemplifying one such model for systems engineering and software engineering.

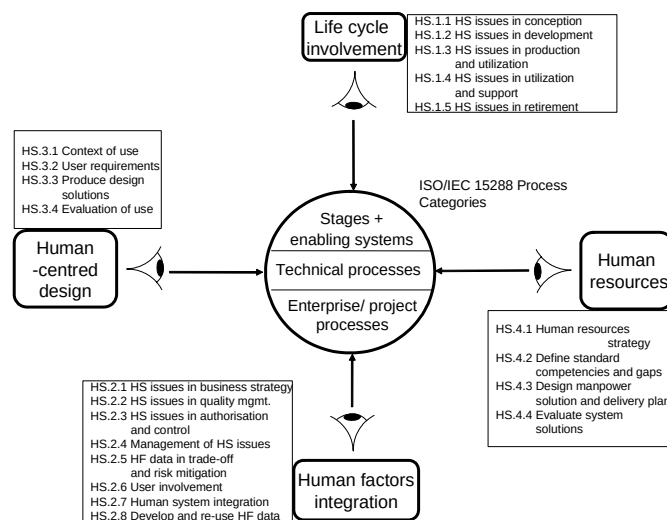


Figure 10.21 The ISO Systems Engineering Model Structure Applied in ISO/PAS 18152:2003, [Arnold, 2002b]

This section has demonstrated that humans, as a system implementation medium and as a given of the operational environment, may be treated according to the principles and practices of systems engineering. This is a sound basis for achieving the federation of systems engineering and human factors sought in the human factors reports cited in §10.5.1.

10.5.3 Human-Intensive Systems

§10.5.2 saw that the characteristics of humans, considered as system elements, build complexity outside the system through layers of work teams, enterprise and social interactions to emerge as the social and cultural pressures that set the ultimate context for man-made system creation and use.

Writing in 1998, Ramo considers that, ‘as compared with a few decades ago, a substantially greater number of professionals exist today who are well seasoned in interdisciplinary problems. They know how to relate the many facets of one technology to another and to relate these in turn to all the non-technological factors that characterize practical problems. They are “systems engineers.” This is appropriate if the word “engineer” is used in what probably should be its true meaning ... To be professional in the overall application of science to society, some of the group must know society well. Such qualifications are essential for the competent systems-engineering team’ [Ramo, 1998, p.20].

Ramo was looking towards a ‘properly carried-through systems design for a more complex social-engineering matter, if handled by a team that includes the social specialists’. Implicitly or explicitly, the IPT must therefore draw on such specialist skills if they are to deliver effective solutions into a complex, publicly transparent and evermore legalistic military environment. Further, this is subject to greater-than-ever social checks and balances regarding the way government, industry and the tax payer relate. Thus, whereas social systems may appear to stray outside the normal confines of systems engineering, they frame the environment in which enterprise systems are created and in which they operate. They are at the very least a concern of systems engineering and its practicing teams – the rule of thumb of “one level up”, or even two levels of Figure 3.2, applies.

This section therefore closes this investigation with a consideration of how far into the social systems reaches of Figure 3.2 the discipline of systems engineering might reasonably apply.

Ramo, who had turned his attentions to applying the systems engineering of the US missile programmes to specific social matters with indifferent results [§2.3.1], however does see the necessity for a warning. ‘Substantial disagreement exists among the professionals as to how useful [systems engineering] is for the bigger problems of society, or for the smaller ones when they are more “social” ... Some experienced systems engineers ... [are] certain the discipline is inappropriate for “people” problems. In this viewpoint, they are sometimes joined by experts schooled in the more unpredictable behavior of man. Some of these more socially trained individuals are concerned that the systems approach’s disciplines cannot be applied successfully to the real-life problems of the human aspects of our civilization’. [Ramo, 1998, p3].

Certainly in its infancy, systems engineering had mostly focussed on inanimate systems; it had been a tide of action in operations research that had addressed many of the human intensive/enterprise system issues towards the upper levels of Figure 3.2. One of Operations Research’s pioneers, Ackoff sensed a growing weakness with the methods employed. By applying systems thinking to human behaviour he considered purposeful, intentional human system according to their psychological, cultural and social drivers. The resulting complexity he saw to require a participative approach in which ‘ends planning’ comprises interactive stakeholder participation in designing an ‘idealized future’.

Resonating with this participative idea, Checkland was at the same time speaking of collective human activity systems. A natural scientist and manager by training, he attempted to apply contemporary, ‘traditional’ systems engineering approaches that had solved objective problems and formulated clear client recognition of an essentially structured problem. This class of problem he termed ‘hard’. However, the problems he was addressing were those of business organisations and human activity systems, centred on the enterprise systems of Figure 3.2 and strongly influenced by the social system level.

The ill-structured, ill-defined and subjective problems they present are difficult to define, more akin to social improvement situation. He approached ‘the development of principles concerning the use of system ideas in problem solving in real-world situations’ [Checkland, 1999, p161] and sought a meta methodology to handle such situations.

Such situations could be coped with according to a process of inquiry, methodologically organised as a learning system of enquiry. A key part of this learning process was a discovery of the nature of the problem or 'situation' in which the so-called 'hard' systems thinking of systems engineering of the day proved inappropriate. The 'soft' systems thinking approach he describes is more appropriate in fuzzy ill-defined situations involving human beings and cultural considerations.

The soft systems methodology assumption is that the world is very complex, 'problematical' and 'mysterious'. In developing a methodology, Checkland (an elegant writer) uses an imprecise, shifting lexicon that makes pinning down the ontological map of 'soft' systems thinking and practice difficult. The linguistic modelling he employs readily uses terms encapsulated in inverted commas, and his graphical models are conveyed as 'fried-egg shapes and curved lines' [*ibid*, p.A13] to show his fluidity of the semantics.

In his enquiries, Checkland makes a journey of discovery about a 'soft' approach only to arrive at a 'learning' pattern of discourse, debate and pictorial representation that ultimately equates to a requirements capture-biased view of the iconic triangle of Figure 4.9.

It was an empirical journey in which 'changes ... were never... academic ...[but] always made as a result of using the approach in a complex world' [*ibid*, p.A16]. The staging points of his route of enquiry are presented in Figure 10.22

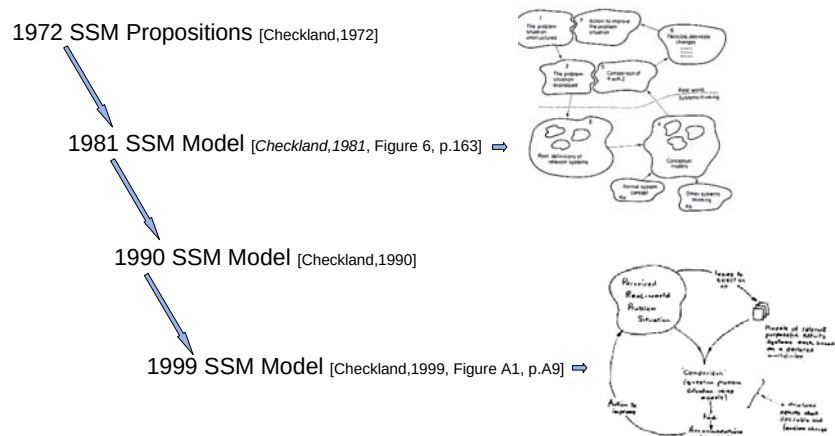


Figure 10.22 Checkland's Development of Soft Systems Methodology

Starting from an engineering-like sequence [Checkland, 1972] he sees classic systems engineering to 'convey the reductionism of natural science' [Checkland, 1981] when applied to an explicit definition of a required objective taken to have an engineered end. For him the word 'system' is a label for something existing in the world outside ourselves and he assumes that world to be a set of interacting systems, 'some of which do not work very well'. From his organisational management perspective, Checkland sees his systems-of-interest as being human-intensive enterprises set in social systems, though he acknowledges that information technology systems are often core [Checkland, 1998].

A decade later he had moved from solving a problem to exploring a 'situation in question ... regarded as problematical'. This he now saw to have 'moved away from classic systems engineering'.

He saw human activity systems in which people were attempting to take purposeful action. This action was meaningful for them – for their world view – and could lead to many interpretations or insights of 'purpose'. Models based on lax notations communicated the perspectives or viewpoints of different participative parties. Each was a set of linked activities which could exhibit the emergent property of purposefulness – essentially they were candidate functional solutions. These viewpoints were resolved as an accommodation amongst people that certain action was desirable feasible.

The steps of this 'soft' approach took the form of a seven-stage model, shown upper right of Figure 10.22 [Checkland, 1981, p.163]. Fundamentally it was a learning cycle with a high

degree of empirical exploration based on the enquiry of participants. As Checkland styles it, a debate structured by questioning perceptions of the real situation reachable by the people concerned.

The seven-stage process of inquiry allows exploration of how people in their individual situations create for themselves the meaning of their world and would then act intentionally as operators or users of the system-of-interest. The seven stages comprise a cycle and give the impression of a well-structured, prescriptive process to be followed systematically; though it is freely associated with neologisms such as 'climate', 'root definition', 'expression', 'obvious', 'relevant', 'specified-task-carrying-out' – Checkland's own inverted commas – that stimulate but do not always inform the reader.

Even the definition of 'soft' in his methodology is acknowledged as a 'slippery concept', with 'hard' applied to a problem type and 'soft' not to the 'fuzzy ill-defined situations involving human beings and cultural considerations' but to the empiricism and learning attributes of the problem solving method.

By 1999, Checkland resolved the method further and reduced it to an unremarkable four-stage model:

1. Finding out about a problem situation;
2. Formulating some relevant purposeful activity models according to scenarios of the perceived problem;
3. Debating the situation, using the models, seeking:
 - (a) changes which would improve the situation,
 - (b) accommodations to enable action-to-improve to be taken,
 and stakeholder consultation for validation;
4. Take action in the situation to bring about improvement.

Its echoes of: service requirement; product functional design; product physical design; and implementation, sound clearly in this model of action.

His figure that describes this model is shown lower right in Figure 10.22. It is the cardinal Figure A1, p.A9 of his 30-year retrospective [*Checkland, 1999*]. Its text is mapped directly to, recounted verbatim, in Figure 10.24. Fundamentally, matters reduce to the design triad of Figure 4.9.

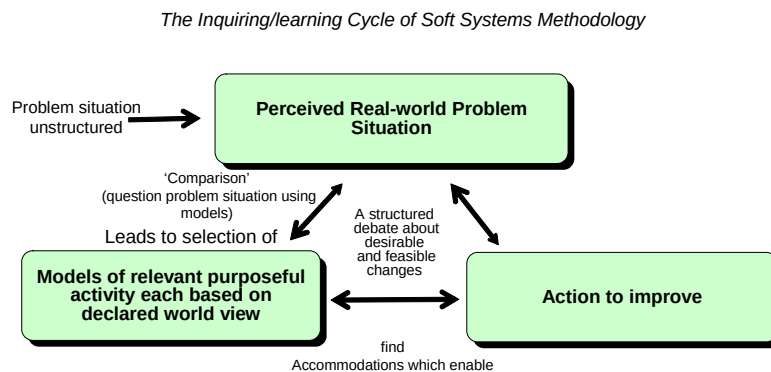


Figure 10.24 Verbatim Mapping of SSM Figure A1 [*Checkland, 1999*]

Given the systems reasoning foundation of Figure 4.9, this is not to say that the techniques employed in Checkland's requirement capture and communication, and his modelling notations, do not make a major contributions to understanding. Nor that systems engineering has evolved to adequately deal with his 'fuzzy ill-defined situations involving human beings and cultural considerations'. Indeed it illustrates that different minds (and different aptitudes), different language and different cultures are naturally tuned to work at different levels in Figure 3.2 according to different degrees of real world abstraction.

In the end, systems engineering is limited to being a model of how human activity systems should act if they are to effect intentional, technology-based interventions in the course of actuality. Such activity systems, e.g. IPTs, founded on the characteristics of their individual human elements, turn on systems reasoning: a massively complex primitive of individual and team cognition, and a cardinal constituent of systems engineering. Nonetheless, any assumption that the ubiquity of systems reasoning opens a systems engineering door to solving a profusion of the world's socially-derived problems is naïve. Ramo and others in the '60s found themselves being unsuccessfully drawn into this.

Before overreaching into the backyards of others, there is still much to be resolved in systems engineering's founding territory – in the engineering and management principles that underlie defence systems, and in the IPT practices that enable these principles to be effectively employed in support of society.

10.6 Outcomes

This chapter has considered how results of this investigation's re-assessment of systems engineering principles could be applied to advance IPT technical practice. Three examples of this have been examined. Each is concerned with a key region of IPT responsibility: the definition of requirements; the design of system architecture; and the harnessing of implementation technologies, exemplified here by human factors.

The first of these three relates to a cardinal goal of Smart Acquisition; that is, improved requirements management. There are indications that MOD is still encountering difficulties in developing effective and robust requirements. As the techniques and staff competence in requirements management advance, problem complexity seems always to outpace them. It is suggested that persistent sources of difficulty have not yet been adequately recognised or dealt with.

One of these difficulties arises from the range of IPT interpretations and project conventions regarding exactly which information on a future system is found in what document, known by what title, and how the information is structured. A concise mapping from the logical representations of the design triad in Figure 4.9, and from its complementary build and test triad in Figure 4.10, to those documents or work products that are termed requirements is necessary to resolve this.

This substantially technical issue is compounded by an influence of trade. The formalism needed at agreement or contracting interfaces means that different requirement definitions must be brought together so as to form concise, effective specifications that can enable contracts, commitments and obligations. This also influences the form and content of requirements. It is the recognition and explicit resolution of these inconsistencies, more than the specific convention adopted, which is seen as the issue.

It is held that the manner in which requirements and hence specifications are employed has a strong influence on the effectiveness of how supply chains work and is intimately bound up with trading culture. Calls for changes, such as in DIS, to the culture that governs MOD/defence industry cooperation and risk management fall at the door of IPT's, and this is a strong determinant of the nature and style of requirements and their management.

The second technical area of IPT responsibility examined was system design. Design is a creative, decision-making activity and, in its detail, invariably an exploration of an unprecedented problem. Nonetheless, this creativity is a product of actual, real world sequences of practitioner action – the tasks and activities that comprise systems engineering – conducted on each occasion according to unique circumstances. Building from this investigation's advancement of systems reasoning, it has been possible to hint at how some of the instinctive, deeper imponderables of engineering design come about, and to give them a sounder, more rational basis. It is seen that cognition, models of conjecture and reality, language conventions in their most general sense, frameworks that provide structure and enable navigation, and patterns that orchestrate the resolution of model concordance all play a part in design. Turning this instinctive melange into a logical prescription for collaborative

endeavour is what systems engineering has struggled with for a half century. Resulting procedure has not served the creative dimension of design well.

However, latter-day advances in the area of language formalisms, modelling conventions, communication protocols and enabling tool support have all received a massive boost from IT and software engineering practitioners, who have contributed major steps of thinking about architecture descriptions and how these relate in particular to the characteristics of enterprise systems, e.g. business enterprises, military NEC. A harmonious accommodation of these advances by the practice of systems engineering, and a complementary reconciliation of evolving architecture frameworks with systems engineering's foundation of systems reasoning is, in the light of the foregoing analysis, both tenable and technically reasonable. For IPTs it would appear massively advantageous, since MOD's proactive position on systems engineering and on architecture frameworks currently presents IPTs with the challenge of observing two doctrines that exhibit a degree of contention. A major alignment, if not a fusing, of systems engineering design and enterprise architecting is an exigency that might be akin to a far-reaching Kuhn paradigm shift [§2.4.2].

Earlier chapters consider humans as the enablers of man-made systems: the manifestation of the 'Fourth' of Aristotle's Cause. As stated, 'systems ... are created by humans, for the benefit of humans ... draw[ing] on the capabilities of humans, and ... respond[ing] to the needs and constraints of humans'. This notion is extended in the third example of technical practice advancement by an analysis of the relationship between systems engineering and the human factors discipline, using this to exemplify a general relationship of systems engineering with implementation-focused disciplines. It illustrates how the symmetry and correspondence of processes from one echelon of system detail to another is the key to systems engineering's seamless collaboration with each of the major engineering disciplines.

This also demonstrates the equivalence of the inanimate and animate in the composition of systems and their environments. It points to systems engineering being dependent on an understanding of the capabilities and behaviour of humans often within and always outside of a system-of-interest. An analysis of the human as an 'implementation medium' saw that aggregations of their individual physical and psychological characteristics disappear into an amorphous substructure, as strata of collectives are built in the manner described by Parsons [§3.4.1] to emerge as the complex of society.

From the early days of systems engineering, it has been adopted as a starting point to devise methods to tackle the higher reaches of this social order. From it models of action have been empirically devised to understand, modify and intervene in 'human-intensive' systems. A critique of a leading approach to this challenge shows that evolved patterns and methods of 'soft' approaches nevertheless rest on the fundamentals of systems reasoning and on the iconic design triad of systems engineering. There is an argument that the objectivity, formalism and discipline of systems engineering are still worth employing to tackle the 'people problem', and that evolved forms of systems engineering await exploration.

Thus it is that this chapter has no neat endings; no comfortable conclusion. Its three examples are pointers to ongoing change. It should be viewed as the work-in-progress part of this investigation – and of systems engineering in general.

11 CONCLUSIONS

11.1 Summary

11.1.1 A Recapitulation

This investigation may appear to have provocatively rattled the bars of many well-established and regarded cages of theory and practice: the ventures of systems engineering into the social sphere, the meaning of capability, Checkland's Soft Systems Methodology, requirements management in MOD, Kant's recourse to transcendental logic, the conceptual distortion of architecture in IT circles, the regarded regime of Systems Thinking or a Systems Approach, aspects of ISO/IEC 15288 model, MOD's CADMID, Zachman's architecture framework, Hitchins' system classification and, not least, 'traditional' systems engineering. To see it in an iconoclastic light would be to miss its intent.

These, and other issues touched on, all serve important purposes in systems engineering. They are all models with value; some of their time, others to satisfy prevailing viewpoints and associated concerns. Yet they, and the principles and practices investigated in this report, are in the end all constrained to being impoverished cognitive representations of an overwhelming complexity of reality and, to a much lesser extent, human consciousness.

The intent of this investigation was to emphasise the essential connectivity of all these issues – selecting them as illustrative of a whole raft having similar systems engineering relevance and importance – and to show how, by re-adjustments in the way they are viewed and analysed, and then re-assembled and synthesised into practical systems engineering discipline, systems engineering can itself be moved along its journey of growing maturity and value creation.

This period of research has investigated systems engineering's "relevance, as a measure of pertinence to challenges; correctness, in the sense of 'to adjust or make conform, especially to a standard'; and value, being the delivered benefit to individuals and society" [p.1-1].

- To bring relevance, principles have been interpreted in a practical context of MOD IPTs. From a UK standpoint this is a setting that has considerable technical and industrial influence, and is capable of giving a systems engineering lead for other business sectors;
- To judge correctness and conformance, especially to a standard, the systems engineering International Standard, ISO/IEC 15288, has been repeatedly used as a point of reference. It gives a focus to thinking and it provides 'an emblem to rally around' for systems engineering as its recognition in international trade and commerce steadily grows;
- To demonstrate value, it has been shown that systems engineering advancements could bring yet more benefit to the defence acquisition market: one of the largest areas of UK public expenditure. Choosing the DE&S's domain of application is to seek improvements in the investment of an annual budget of up to £16 billion (43% of the UK Defence budget).

The six years of part-time research, one year of analysis and one year of write-up represent a significant proportion of the lifetime of systems engineering as an identifiable practice. Over this period, the subject has evolved, gained greater recognition and been more widely applied.

Some concepts and advances proposed by this research have already contributed to this changing scene, particularly to systems engineering standards and their use in business. At the completion of this investigation their acceptance and use by others has diminished their novelty. The technical and managerial interpretations and proposals made during this investigation are thus summarised. Where relevant, the circumstances of influence or application are indicated in italics.

11.1.2 Principles and Engineering Knowledge

A number of engineering insights and advancements of understanding resulted from the period of research. These are concerned with:

- A comparison of the principles behind the 'creation' of biological systems and man-made systems, and how this contributes to an understanding of systems engineering;
 - This led to the identification of 10 fundamental operations that govern the life of animate and man-made systems. These were seen to depend on four processes that are essential in order to effect the management of any system life cycle. Coupled with the archetypal 'plan', 'check', 'act' of the Shewhart Cycle control cycle, they form a minimal and sufficient set of system life cycle management processes and justify the set employed in the systems engineering International Standard.
- A reformulation and rebalancing of the primary disciplines that contribute to the current perception of systems engineering's business role, process scope and organisational influence profile;
 - From a recounting of the history and evolution of systems engineering as a recognised discipline, the cardinal factors that led to its emergence are analysed. This leads to the conclusion that three regions of business and academic discipline are essential to its formulation and application. Systems reasoning, engineering and management are set in relation to one another in a manner that shows how systems engineering fits into the wider picture of professional disciplines.
- A positioning of systems engineering in a continuum of business process that help to clarify its relationship with existing management and technical roles, in particular with project management;
 - A model of business processes conveys the accountability and influence profiles of the major management and technical functions within most organisations. It demonstrates the value of systems engineering awareness throughout any organisation, the benefit of competence in at least one implementation technology-based engineering discipline, and the natural synergy that exists between systems engineering and project management.
- A complementary set of viewpoints of a system, seeing it as a product, capability or service, that help to modify the traditional product focus taken in systems engineering;
 - Derived from a through-life consideration, the separation of product and service viewpoints of any system is shown to be profoundly influential. It rationalises the need for system life cycle processes that separately accommodate both viewpoints. These are shown to be integral, though generally not well-discriminated in frameworks of system models that describe architecture. It is shown that a much used capability term in defence systems, has a clear technical meaning that needs to be better reflected in systems engineering's codification.
- The reiteration, justification and partial re-formulation of a minimal, sufficient set of systems engineering processes from which life cycles models can be built;
 - The logic of why systems engineering is composed of a now widely-accepted set of processes is built up from the intellectual and evolutionary foundation of systems reasoning. This draws on the philosophical, scientific and psychological precepts that govern human cognition. The idealised model of transformations that emerges is shown to be the quintessential representation that morphologically can be transposed to the principal life cycle patterns.
- The presentation of enabling systems as a system class that simplifies the definition of a system's boundary and its descriptions;

- A concept that was graphically contributed from this investigation to the systems engineering International System, enabling systems are shown to be an effective mechanism for reducing the size and complexity of models of systems engineering. Associated with separate stages of a life cycle and thus with a sequence of different environments, one party's enabling systems are another's system-of-interest.
- A taxonomy of system life cycle forms that relate to a hierarchy of progressive complexity and circumstances of use;
 - The commonly seen patterns in most man-made system life cycles are presented in a hierarchical classification structure. Successive members of this classification provide progressively richer models, and are able to encapsulate ever greater structural and logical complexity according to this investigation's commensurate classification of systems complexity.
- An analysis of the business model behind the International Standard on systems engineering, illustrating its strengths and weaknesses;
 - The close correspondence between the presentation of principles in this investigation and the provisions of the systems engineering International Standard is used to justify the validity and value of this major step in systems engineering's codification. Its blending of logical and physical descriptions, according to functional and object-orient modelling techniques, is shown to provide commanding advances at the expense of entirely tolerable compromises, omissions, weaknesses and inconsistencies.
- The definition of a set of artefacts and information items, based on a proposed work product classification, that may be used as indicators of an organisation's systems engineering capability;
 - To address ISO/IEC 15288's failing to present even an indicative set of work products associated with its process transformations, and thus a resultant inability to correlate and therefore validate the two supreme viewpoints of process definition, this investigation has analysed the text of the standard and formulated a complete set of rigorously derived work products that should have been presented as part of the model. To structure this, a refined set of work product classes had been specified. This classification and the member sets of work products have been accepted for publication during 2008 by ISO in its systems engineering life cycle process assessment reference model, to be identified as *ISO/IEC 15504 Part 6*.
- A description of requirements that conveys how they contribute to and may confuse technical communications and agreements in inter-organisational trading;
 - Some of the technical reasons for persistent difficulties in developing effective and robust requirements for systems, and defence systems in particular, are analysed. Contrasting instantiations of requirements definition are seen to derive from different interpretations of the complementarity that divides and unites functional and physical representations. Compounded by a need to meet communication needs in trading transactions, the challenges to achieve intra- and inter-organisational uniformity in requirements definition, yet accommodate local needs and conventions, are seen as a yet-to-be resolved factor in defence systems acquisition. Discrimination in favour of either a concurrent logical or a sequential physical modelling representation is seen to be one route to ameliorating this situation.
- A review of the fundamentals of design derived from systems reasoning that leads towards a reconciliation and better harmonisation of systems engineering and enterprise/software architecting;
 - Building from this investigation's advancement of systems reasoning, some of the presumed reconditeness of engineering design is placed on a better and more logical footing. This rational basis for creativity is translated in the pragmatism of standardised frameworks of system models, such as MODAF.

The diversity across frameworks are categorised and explained by way of reference to their common origin in the cognitive models of systems reasoning. The patterns in a system's design and the patterns of system life cycle process application are seen to be 'two sides of the same coin' – the one emanates from the other – and this points to the formulation of methods that could bring better team-based discipline to what has hitherto been seen as the preserve of individual talent and mastery.

- An analysis of the role of humans within and outside systems-of-interest, and how this relates the disciplines of systems engineering and human factors and how it illustrates the general relationship with implementation technologies.
 - The advancement of human factors as an engineering-related discipline is a theme that runs through this investigation. By considering the symmetry of humans within and outside a system-of-interest boundary, and their attributes, traits, functions and roles associated with humans, it is easier to accommodate and trade their merits and hindrances to better effect, whether they are considered as stakeholders in the system environment or as operators within the system. This investigation's graphical representation of these concepts has been incorporated in the *INCOSE Systems Engineering Handbook V 3*. This mode of thinking is extended to a consideration of people as they form social systems, and it is seen that whilst the principles remain largely in tact, the practicalities mitigate against extending the methodologies of systems engineering beyond reasonable bounds.

11.1.3 Practice and Management Understanding

In considering systems engineering practice at the organisational and the individual level, this research has thrown light on some aspects of management and on trading of complex systems. These clarifications relate to:

- An examination of disciplines and their maturity, and an assessment of where systems engineering sits in such ranking;
 - In an assessment of the dependability of systems engineering as foundation for organisational advancement and improvement, particularly in domains dependent on complex system creation and use, an analysis of the characteristics of disciplines is used to gauge the current and likely future status and impact of systems engineering as a discipline. As a discipline, but not a profession, systems engineering is judged to be a stable, well-founded and instrumental contributor to business organisation, well beyond its current high-technology constituency of practitioners and advocates. This is seen to be a justification for its academic, technical and commercial recognition.
- A harmonising of systems-oriented awareness, knowledge and consciousness is advanced to provide a novel, coherent balance that rationalises the disparity between reality and cognition;
 - Titled systems reasoning to distinguish it from other physical and metaphysical conglomerate assembled by others, this amalgam draws on three disciplinary threads to postulate how mankind handles the complexity of reality, without reliance on transcendentalism or mysticism and within the limitations of evolved neural capabilities. It is advanced that the essence of human awareness of systems can be accounted for by a particular amalgam of regions of philosophy, the sciences and psychology. It is offered as the quintessence of how humans perceive and abstract information-compressing order, and how they understanding from an otherwise overwhelming immensity of real-world information. The mental mechanisms that have evolved to enable this are shown to readily map to much that is familiar in the 'common sense' of systems engineering.
- A description of systems engineering in business and inter-organisational trading, asserting its widespread and necessary contribution to business undertakings;

- This investigation has positioned the science, technology and engineering on which systems engineering depends within a framework of business management and trade. A consideration of systems engineering in isolation of these factors has technical merit but limited social value. The selection of MOD IPTs has provided a business focus to many aspects of this investigation. The IPTs have been shown to be pivotal in MOD's introduction of enhanced acquisition practice and proposed advancements to this do not stop at the boundaries of the DE&S, or indeed MOD.
- An inter-related set of models of systems engineering expressed in terms of business process, organisational capability and professional competence;
 - It is advocated that a triptych of views of systems engineering is a requisite for communicating and advancing the discipline. Based on a foundation of business process, three interconnected models are advanced. The business process model adopted is that of ISO/IEC 15288. The transformation of information from this into models of organisational capability and professional competence provide a coherence, rigour and concordance of viewpoints of the discipline not yet achieved in other representation.
- A model of systems engineering competence expressed in terms of business process and organisation roles that may be used to describe professional attainment and career development.
 - This investigation proffers that at the limitations of useful codification, competence must have taken over to adapt and fashion in order to meet the specifics of individual projects and enterprise. Accordingly, it maps from that codification to a compatible structure of individual and team competence. In doing this it establishes the basic dimensions of a competence framework and how they contribute to a definition of the whole. This is seen to be compatible with SFIA, a well-founded UK model of IT and software engineering competence and roles. A limited demonstration of how such a systems engineering competence framework might look is presented.
- A survey of MOD IPT's experiences with systems engineering principles and practice, with an assessment of the levels of use and effectiveness as perceived by IPT Leaders;
 - In the absence of credible, document information on how systems engineering is authoritatively perceived across DE&S and within individual IPTs, a survey of IPTL opinions was designed and conducted. This confirmed some well-expected results along with insights and themes that threw new light on challenges and opportunities to advance IPT practice by employing systems engineering principles. The survey detected an IPTL empathy with systems engineering's purpose and potential. This both demonstrated a natural synergy with project management, and substantiated that IPTL commitment to systems engineering went beyond doctrinaire observance to informed, candid endorsement. It is concluded that perhaps the most influential faction with regard to systems engineering's long term health in MOD are engaged proponents.
- A comparison of systems engineering policy and its implementation in the UK MOD acquisition teams with an overseas equivalent: the Swedish Defence Materiel Administration, FMV;
 - Comparative studies provide points of reference and, whether intended or not, audit information. The study of how FMV and DE&S have approached the incorporation of systems engineering into their business suggested that history, culture and economic size all have a part to play in the differences detected. Technical environments and government/industry relationships are sufficiently similar to be discounted. The stronger FMV alignment with international standards is seen to be the most significant interesting distinction and this remains an issue worthy of future analysis.

- A critique of the acquisition life cycle employed by MOD and a prognostication regarding the nature and possible direction of changes to it;
 - The doctrine of Smart Acquisition is almost indivisible from its emblematic CADMID. Powerful as CADMID has been, like its predecessor life cycle models, it is not immutable. It is demonstrated that it contains weaknesses that do not favour the trend towards NEC and the sharp reduction in major, new-build equipment programmes. Extending basic life cycle patterns to better balance product creation and service delivery, it has been shown that new models can be evolved to handle anticipated changes in acquisition regime and the through life nature of DE&S IPTs. An example of a such a model is presented. Branded the *Infinity* model it is based on sufficiently different fundamentals to influence life cycle management culture, but not so radical that it introduce a discontinuity with respect to existing programmes governed by CADMID.
- An analysis of how systems engineering practice could be better planned and managed in MOD IPTs without disrupting existing management methods and artefacts.
 - Shortcomings in MOD's committed presentation, effective guidance and explicit support for systems engineering is seen as a distinct failure of Smart Acquisition. Corporate advocacy for its role in enhancing MOD acquisition has not been matched by promised institutionalisation or sustained engagement. To fill this gap, and to establish a foundation of systems engineering in each IPT, it is suggested that the obligatory TLMPs be extended into an instrument of project systems engineering, bringing greater day-to-day relevance to TLMPs, and more closely aligning existing practice and project procedure to systems engineering discipline.

11.2 Future Lines of Enquiry

This research has encountered questions that it has not been possible to address in sufficient measure due to effort limitations or their appearing peripheral to the main scope of enquiry. The benefits of pursuing them as future lines of enquiry became apparent during the course of this investigation. They are as follows:

- The challenges in meshing the discipline of systems engineering with conflicting or related disciplines. Whilst a balanced co-existence with project management appears secure, discontinuities remain with software design of architecture and with the architecture practices for the IT intensive systems that enable enterprise to operate and be controlled. More assessment of the inconsistencies and the opportunities seen in this situation could benefit jointly the advancement systems engineering, software engineering and IT practices.
- A notable weakness arising from systems engineering's failure has accommodate service provision and service management issues is deserving of a more effective investigation. The role of systems engineering in aspects of service delivery and sustainment needs to be strengthened to give proper balance to life cycle management.
- Questions raised about order, entropy and the physics of system have only been superficially considered. They merit a more structured and rigorous analysis in order to bring better quantification, or at least categorisation, into the consequences of complexity. It is unclear whether growing man-made system complexity and the pace of business demands can be handled by either scaling or extrapolation, or whether there are step changes required in discipline methods and/or solution structure to counter their adverse effects.
- The competence framework developed in this investigation was intended as a demonstration of feasibility rather than a completed model. There is benefit in proceeding to a finished framework having explicit traceability to the systems engineering

International Standard and to the Skills Framework for the Information Age (particularly in the face of strong 'architecting' influences from the software and information technology domains).

- The insights offered by the particular composition and balance that was used in order to build a body of systems reasoning suggests that its further development could throw light on existing and perhaps new interpretations of systems engineering.

11.3 Reflections

The duration of this research has spanned roughly one eighth of the period since the emergence of a discipline that is now recognised as being a highly effective approach to creating, evolving and applying large-scale, technically complex defence systems. For better or worse, it became known as systems engineering. With use, it is becoming recognised as relevant to many forms of man-made systems, including those comprising humans.

Systems engineering was originally constituted from existing regions of engineering and management practice. To this was added a system-oriented approach that, when viewed explicitly, introduced a novel and highly promising differentiator with evident widespread relevance. Unlike other branches of engineering, systems engineering was therefore not based on any radical advancement in technology; it was more a reaction to an accumulating level of technological diversity and opportunity that had crossed a threshold in an industrial society struggling to securely harness and control technology.

Systems engineering may thus be viewed as a response to a condition of society as much as to a pursuit of technological and engineering excellence.

Its basis of systems reasoning had the effect of simultaneously ordering and disciplining not just technical understanding about complex, engineering interventions in the real world. It also contributed to coordination and management of communal endeavours that caused these to come about.

Faced with this rich complexity of systems engineering, an investigation of this type must be something of a personal voyage. This has certainly been the case here. It will find favour with some, less so with others; for, as with viewpoints of any complex entity, it is all a matter of observer concerns coloured by the conditioning of past experiences and past ways of seeing things, and the nature of anticipated challenges ahead. There is, therefore, no definitive view of systems engineering. If there were then systems engineering would lack an essential property: the potential of exhibiting great complexity in order to create and control great complexity.

The complexity gap between society's demand for man-made systems to intervene in the real-world, and our ability to create and control such complexity, has probably been substantially invariant over history; from flint axe, through ziggurats, to Internet. The most powerful 'tool', using Ramo's expression, that impacts the wave front of absolute man-made complexity is what we presently term and package as systems engineering. Our ability to discipline ourselves to conduct and master this tool is a major determinant of where the wave front of controllable complexity lies at any particular time.

A discipline with the potential for such influence on society would thus seem to rest on some fundamentals of mankind. This proposition is supported by some of the propositions advanced in the course of this investigation. It would appear that a growing body of engineering and technological knowledge, a refined understanding of managing enterprise resources and structures, and a developed sense of how to communicate the real-world interventions required by society does indeed depend on some cardinal dimensions of the human mind.

From the early Greek philosophical interpretations of some of these dimensions, through the structured discipline established by scientific method, to recent psychological enquiry into the machinery of human perception, comprehension and intention, systems engineering would appear to emanate from some profound cognitive determinants. Even in a fast moving and occasionally turbulent animate and inanimate world, these determinants will not change

significantly. What will change, and perhaps radically so, is our ability to better leverage our innate cognitive gifts by developing communally-optimised systems engineering approaches that effectively enable the potency of evermore complex, man-made machines.

Thus far, a persistent challenge for systems engineering has been to abstract a scope, definition, description and modes of application that are able to preserve its potential for vast application complexity, whilst at the same time make it appear comprehensible, uniform, teachable, relevant and readily valued. Whatever this level of abstraction is, it must of needs be complemented by individual and communal competence, and be supported by a compatible environment in which to practice it. This is a massively delicate and under-appreciated balance to strike, and, importantly, to maintain it in the changing circumstances of a maturing discipline. Struck incorrectly one way or the other, systems engineering will suffer; indeed, has already suffered.

This balance rests on how far the codification of sound principles should be taken and at what point a culture of effective practice takes over. Overlap is inevitable and desirable. Whilst this challenge may be concisely stated, its resolution is subtle, dynamic and only empirically verifiable. Huge, business-driven factors affect it: the inertia of extant approaches and past investments; established cultures in particular application domains and different nations; volatile social imperatives and expectations that push the boundaries of technical and collective achievement to and beyond the limit; the disruption of radical, generally unforeseen technology advances that completely change the science and technology scene.

Because of this dynamic situation, there is no final resolution. World order, our science and technology base, systems engineering knowledge, skills and experience, and the context of related disciplines are all on the move.

This investigation has looked at these matters in the opening years of the C21st. It would have looked rather different 50 years ago closer to systems engineering's inception. It will look different in 50 years time, but maybe not so different – other, hopefully, than systems engineering's wider recognition and adoption.

It is of course possible that the system in systems engineering will simply fade away by name and identity through this century, leaving engineering profoundly transformed to better meet late C21st technical and business complexity. It may with hindsight have been an essential transformational expedient that enabled essential evolutionary advances in engineering principles, practice and culture.

However, for the present, systems engineering appears to be a sound, dependable basis for many of MOD's stated acquisition enhancement actions, and for its IPTs to become more effective units in a key UK government and industry relationship.

Beyond this, systems engineering should now be seen as a dependable basis for advancements in a wide range of businesses that are not yet following its precepts.

Behold, the people is one, and they have all one language... and now nothing will be restrained from them, which they have imagined to do.

[*The First Book of Moses, called Genesis*, Chapter 11, Verse 6]



The Tower of Babel

Pieter Bruegel the Elder, 1563
Kunsthistorisches Museum, Vienna

REFERENCES

- Abbott, 1988 *The System of Professions: An Essay on the Division of Expert Labour*, Abbott,A., The University of Chicago Press, Chicago, IL, 1988
- ACH, 2007 *Capability Management Handbook (Interim Edition)*, TLMC Design Team, UK Ministry of Defence, 26 February 2007
- Ackoff, 1979 *The Future of Operational Research Is Past*, Ackoff, R.L., Journal of the Operational Research Society, Vol. 30 No.2, pp.93-104, 1979,
- Ackoff, 1999 *Transformational Leadership*, Ackoff, R.L., Strategy and Leadership, 27(1), pp.20-25, 1999
- ADFP 1994 *Australian Defence Force Publication (ADFP) 101 – Glossary*, May 1994.
- Aesop attributed to *The Four Oxen and the Lion*
- AFSC, 1966 Manual 375-5, *Systems Engineering Management Procedures*, Air Force Systems Command, AFB, Washington DC, 18 Jul 1966
- Alchian, 1950 *Uncertainty, Evolution and Economic Theory*, Journal of Political Economy, 58, June 1950
- AMS, 2000 *Acquisition Management System v2.7*, SPRINT Implementation Team, MOD, Bath, March 2000
- AMS, 2004 *Acquisition Management Systems*, DDefAcqAMSTL, MOD, <http://www.ams.mod.uk/> Accessed Aug 2004
- AMS, 2004 *Acquisition Management Systems*, Version 8.1, January 2004
- AMS, 2007 *Requirement and Acceptance, Guidance, Policy and Acceptance for the UK Acquisition Community, Version 1*, April 2007, AMS accessed Oct 2007.
- AMS, 2007a <http://www.ams.mod.uk/content/randr/active/pg000067.htm>, accessed May 2007
- AMS, 2007b <http://www.ams.mod.uk/content/docs/tlmguide/matmodv3-1.doc>, accessed August 2007
- Ansoff, 1965 *Corporate Strategy*, Ansoff, H.I., McGraw-Hill, New York, ix. p166, 1965
- AOF, 2007 <http://www.aof.mod.uk/amscontent/docs/sysengin.doc>, Accessed October 2007
- AOF, 2007a http://www.ams.mod.uk/content/docs/reqweb/content/ur_defne.htm?zoom_highlight=15288, Accessed October 2007
- AOF, 2007b http://www.ams.mod.uk/content/docs/reqweb/content/sr_defne.htm?zoom_highlight=15288, Accessed October 2007
- AOF, 2007c <http://www.ams.mod.uk/content/docs/reqweb/content/ecs.htm>, Accessed October 2007
- AOF, 2007d http://www.aof.mod.uk/aofcontent/tactical/randa/content/evolutionstage7.htm?zoom_highlight=ECS, accessed October 2007
- AOF, 2008 *Acquisition Operating Framework*, <http://www.ams.mod.uk/aofcontent/aofintro.htm>
- Aristotle, Anim. *De Partibus Animalium*, Aristotle, Oxford University Press, 10 0198751281, 1992
- Aristotle, Met. *Metaphysics*, Aristotle, Oxford University Press, ISBN 9780198751069
- Aristotle, Phys. *Physics*, Aristotle, Rutgers University Press, ISBN 1 888009 03 9, 2002
- Aristotle, 350 BC *On the History of Animals*, Aristotle, 350 BC
- Arnold, 1997 *Guide to the DERA Systems Engineering Practices Reference Model*, DRA/LS(SEC)/ATS/PROJ/018/G01, Defence Evaluation and Research Agency, Farnborough, Hants, 5 April 1997
- Arnold, 2000 *From Process Towards Profession*, Arnold, S., INCOSE Annual Symposium, Minneapolis, MN, 2000

REFERENCES

- Arnold, 2001 *Addressing the People Problem – ISO/IEC 15288 and the Human-system Life Cycle*, Arnold, S., Sherwood-Jones, B., Scottish Systems Engineering Convention, Glasgow, 29-29 Nov 2001
- Arnold, 2001a *System of Systems Thinking – Defence Capability Practice*, Arnold, S., Defence Systems and Equipment Conference, London 11-14 September 2001
- Arnold, 2001b *Humans and the System Life Cycle*, Arnold, S., Sherwood-Jones, B., Scottish Systems Engineering Convention, Glasgow, 29 Nov 2001
- Arnold, 2002 *The Human Dimension*, Arnold, S., Earthy, J.V., Sherwood-Jones, B., 5th Annual Systems Engineering for Defence Conference 21-22 May 2002, RMCS Shrivenham
- Arnold, 2002a *Developing a Coalition Systems Engineering Process*, Arnold, S., Cook, S.J., Arnold, Insight, Vol. 5. Issue 3, International Council on Systems Engineering, October 2002
- Arnold, 2002b *ISO/IEC 15288 and the Human-System Life Cycle*, Arnold, S., Earthy, J.V., Sherwood-Jones, B, INCOSE International Symposium, Las Vegas, 2002
- Arnold, 2003 *Systems Engineering Coming of Age*, Arnold, S., IEE Journal, IEE Publications, March 2003
- Arnold, 2003a *Software and System Convergence – Life Cycle Management*, Arnold, S., Software Process Improvement Forum, Montreal, May 2003
- Arnold, 2003b *Systems Engineering*, Arnold, S., IEE Journal on Engineering Management, July, IEE Publications, 2003
- Arnold, 2004 *Viewing Systems from a Business Management Perspective: The ISO/IEC 15288 Standard*, Arnold, S., Lawson, H., Systems Engineering, The Journal of the International Council of System Engineering (Volume 7 Number 3), Wiley 2004.
- Arnold, 2005 *System Life Cycle Management at the Swedish Defence Materiel Administration and the United Kingdom Defence Procurement Agency*, Arnold, S., Lawson, H., QinetiQ/KI/CI/TR050938, March 2005
- Arnold, 2005a *Placing Defense Capability in the Systems Engineering Lexicon*, Arnold, S., Insight, Vol. 8. Issue 1, International Council on Systems Engineering, October 2005
- Arnold, 2005b *Requirements – The Good, the Bad and the Ugly*, Arnold, S., Martin. J.N., INCOSE Annual Symposium, Rochester, NY, June 2005
- Arnold, 2005c *Coherent Systems Require Coherent Practices*, Arnold, S., Keynote Address, IEE/HF DTC Symposium on People and Systems, November 2005
- Ashby, 1956 *Introduction to Cybernetics*, Ashby, W.R. Chapman & Hall, ISBN 0-416-68300-2, 1956
- Bar-Yam, 2003 *When Systems Engineering Fails --- Toward Complex Systems Engineering*, Bar-Yam, Y., IEEE International Conference on Systems, Man and Cybernetics, 2003, Vol 2, pp. 2021- 2028, ISBN: 0-7803-7952-7, 2003
- Bawden, 2007 *Organised complexity, meaning and understanding: An approach to a unified view of information for information science*, Bawden, D., Aslib Proceedings, Vol. 59 pp.307 – 327, Emerald Group Publishing Limited, 2007
- v. Bertalanffy, 1950 *The Theory Of Open Systems in Physics and Biology*. von Bertalanffy, L., Science, 3
- v. Bertalanffy, 1952 *Problems of Life: An Evaluation of Modern Biological and Scientific Thought*, New York: Harper, 1952.
- v. Bertalanffy, 1968 *General Systems Theory: Foundations, Development, Applications*, von Bertalanffy, L., New York: Braziller 1968
- v. Bertalanffy, 1975 *Perspectives on General Systems Theory*, von Bertalanffy, L., New York, Braziller, 1975]
- Bloom, 1956 *Taxonomy of Educational Objectives, the classification of educational goals – Handbook I: Cognitive Domain*, Bloom, B. S. (ed.), New York, McKay, 1956

REFERENCES

- Boehm, 1988 *A spiral model of software development and enhancement*, Boehm, B., Computer, May, 61-72, 1988
- Boehm, 2003 *COSYSMO: Constructive Systems Engineering Cost Model Coming of Age*, Boehm, B.W., Reifer, D.W., Valeri, R.A, INCOSE International Symposium, Washington, DC, 2003
- Boltzmann, 1877 *On the Relation of a General Mechanical Theorem to the Second Law of Thermodynamics*, trans. Brush, Kinetic Theory. Volume 2: Irreversible Processes, Oxford, Pergamon Press, 1966
- Booch, 1987 *Software Components with Ada*, Booch, G., Benjamin/Cummins Publishing Company, Inc, ISBN 0-8053-0610-2
- Booch, 1991 *Object-Oriented Design With Applications*, Benjamin Cummings, Booch, Menlo Park, CA, 1991.
- Borges, 2000 *The Celestial Emporium of Benevolent Knowledge*, Borges, J.L., Levine and Weinberger (2000) Penguin,]
- Boulding, 1953 *The Organization Revolution: A Study in The Ethics of Economic Organization*, Boulding, K. E., Harper, New York, 1953.
- Boulding, 1956a *General Systems Theory – The Skeleton of Science*, General Systems, Volume I, Boulding, K.E., pp. 11–17, 1956
- Boulding 1956b *Toward a General Theory of Growth*, General Systems, Vol 1, I Boulding, K. E. 1956 pp. 66–75
- Boulding, 1981 *Evolutionary Economics*, K. Boulding, Sage Publications, Beverly Hills 1981
- Butterfield, 1957 *The Origins of Modern Science*, Butterfield, H., 1300-1800, New York, The Free Press
- Brentano, 1874 *Psychology from an Empirical Standpoint*, Franz Brentano, edited by McAlister, L.L., London, Routledge, 1995
- Brill, 1998 *Systems Engineering – A Retrospective View*, Brill J.H., Systems Engineering Vol 1 No 4, Wiley InterScience, 1998
- de Broglie, 1939 *Matter and Light the New Physics*, de Broglie L., 1939, Norton & Co, Inc, New York, pp Studies I
- Bronowski, 1973 *The Ascent of Man*, Bronowski, J., British Broadcasting Corporation, London, ISBN 0 563 104988, 1973
- Brook, 2002 *Dennett's Contribution to Science*, Brook, A., Ross, D. eds., Cambridge University Press,. ISBN 0521008646, 2002
- Cannon, 1927 *The Wisdom of the Body*, Cannon, w.B., 1927
- Chang, 2000 *The Need for a Form, Function, and Behaviour-based Representation System*, Chang, E., Li, X., Schmidt, L., published at <http://www.enme.umd.edu/DATLab>
- Checkland, 1999 *Systems Thinking, System Practice*, Checkland, P.B., Wiley and Sons Ltd, Chichester, ISBN 978 0 471 98606 5
- Checkland, 1972 *Towards a system-based methodology for real-world problems*, Checkland P.B., Journal of Systems Engineering, Vol 3, Part 2, 1972
- Checkland, 1981 *Systems Thinking, Systems Practice*, Checkland, P.B., Wiley, Chichester, ISBN 0 471 279110, 1981
- Checkland, 1998 *Information, Systems and Information Systems*, Checkland, P.B, Howell, S, Chichester: Wiley,. ISBN 0471958204, 1998
- Checkland, 1999 *Systems Thinking, Systems Practice – A 30 year retrospective*, Checkland, P.B., Wiley, Chichester ISBN 978-0-471-98606-5 1999
- Clausius 1864 *Abhandlungen, iiber die*, Clausius J E. R. 1864, trans. 1867. Mechanical Theory of Heat 1st. mechanische Wametheorie. (Braunschweig: Vieweg)
- CMH, 2007 *Capability Management Handbook (Interim Edition)*, MOD, 26 Feb 07
- COA, 202 *The Strategic Plan for the Defence Estate*, Department of Defence, Commonwealth of Australia ISBN 0 642 80646 2 2002

REFERENCES

- Cockburn, 1993 *The impact of object orientation on application development*, Cockburn, A., IBM Systems Journal, August, 420-444, 1993
- Cockburn, 1994 *In search of methodology*, Object Magazine, 4(4):52-56,1994.
- Collins, 1991 *Dictionary of the English Language*, Harper Collins 1991, London, ISBN 0 00 433134-6
- Cook, 2003 Elements of a Framework for the Engineering of Complex Systems, Cook, S.C., Kasser, J.E., Ferris, T.L.J. Proc of ANZSYS 03, Melbourne, November, 2003
- Confucius, Har. *Harmony and Consensus*, Confucius
- DAG, 2004 *Defense Acquisition Guidebook*, Chapter, 6.3.3.4 Support Strategy and Acquisition Strategy, Last modified on 16 Dec 2004
- DAU, 2004 http://acc.dau.mil/simplify/ev.php?ID=6174_201&ID2=DO_TOPIC, accessed July, 2004
- DAU, 2006 http://akss.dau.mil/DAG/Guidebook/IG_c4.2.2.asp. Last modified 7/10/2006DoD 1998 Memorandum May 10, 1995; Defense Acquisition Reform: Status and Current Issues, US Congressional Research Service, January 1998]
- Dawkins, 1976 *The Selfish Gene*, Dawkins, R., Oxford University Press, ISBN 0 19 929115 2
- Dawkins, 1986 *Blind Watchmaker*, Longman 1986
- Dawkins, 1998 *Unweaving the Rainbow*, Dawkins, R., Allen Lane, Penguin Press, ISBN 0 713 99214-x, 1998
- Dawkins, 2001 *Eulogy for Douglas Adams*, Dawkins, R., Church of Saint Martin in the Fields, London, 17th September 2001
- Dawkins, 2005 *The Ancestor's Tale*, Dawkins, R., Weidenfeld & Nicholson, London, ISBN 0 297 82503 8, 2005
- Dawkins, 2006 *The God Delusion*, Dawkins, R., Bantam Press, ISBN 0, 593 05548 9, 2006
- Dennett, 1971 *Intentional Systems*, Dennett, D.C., Journal of Philosophy LXVIII, 1971
- Dennett, 1978 *Intentional Systems*, John Haugeland (red.), Mind design, Bradford Books, Montgomery, Vermont. 23sidor., 1981, original 1978
- Dennett, 1987 *The Intentional Stance*, Dennett, D.C., MIT Press, Cambridge MA, ISBN 0-262-54053-3, 1987
- Dennett, 1995 *Darwin's Dangerous Idea: Evolution and the Meanings of Life*, Dennett, D.C., Simon and Schuster, 1995
- Dennett 1996 *True Believers*, Dennett,D.C. in Mind and Cognition, Ed W Lycan, Blackwell Publishers, ISBN 0631205454, 1996
- DERA, 1997 *Systems Engineering Practices Reference Model*, Arnold, S., DERA/LS(SECFH)/PROJ/018/G01, Farnborough, May 1997
- Descarte, 1625 *Rules for the Direction of the Mind*, Descarte, R., Bobbs-Merrill Co, ISBN10 0672603349, 2000
- Dickerson, 2004 *Using Architectures for Research, Development and Acquisition*, Dickerson, C.E., Soules, S.M., Sabins, M.R., Charles, P.H., Department of Assistant Secretary of Defence, DoD, Washington, 2004, ISBN 1 932433-08-2
- DIS, 2005 *Defence Industrial Strategy, Defence White Paper*, The Stationery Office, ISBN 0101669720, December 2005
- DND, 2002a *Defence Plan Online – Lexicon*, Department of National Defence, available at http://www.vcds.forces.gc.ca/DPOOnline/Lexicon/Intro_e.asp?sltr=C; accessed 2002
- DND, 2002b *Capability Based Planning for the Dept of National Defense*, Canadian Department of National Defense, 27 May 2002
- Dobzhansky, 1974 *Studies in the Philosophy of Biology: Reduction and Related Problems*, Dobzhansky, T., Ayala F.J., eds. University of California Press, ISBN-10- 0 520 026497, 1974
- DoD, 1974 Mil-Std-499A (USAF), *Engineering Management*, US Department of Defense, Andrews AFB, Washington 1 May 1974

REFERENCES

- DoD, 1992 *Mil-Std-499B, Systems Engineering*. A coordination draft never formally issued, DoD, May 1992; a copy can be found on <http://www.incose.org>
- DoD, 1995 *Mil-Std-961D DoD Standard Practice for Defense Specification*, DoD, Washington, 22 March 1995
- DoD, 1996 *DoD Guide to Integrated Product and Process Development (Version 1.0) DUSD(A&T), February 5, 1996*
- DoD, 1996a *Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR) Architecture Framework, Version 1.0*, Integrated Architectures Panel of the C4ISR Integration Task Force, DoD, Washington, 7 June 1996.
- DoD, 2000 *DoD Architecture Framework 2.1*, DoD Architecture Framework Working Group, DoD, Washington, October 2000
- DoD, 2002 *Mandatory Procedures for Major Defense Acquisition Programs (MDPAPS) and Major Automated Information System (MAIS) Acquisition Programs*, DoD 5000.2-R, DoD, Washington, April 2002
- DoD, 2002a *US DoD Joint Chiefs Manual CJCSI 3180.01, Joint Requirements Oversight Council (JROC) Programmatic Processes for Joint Experimentation and Joint Resource Change Recommendations*, 31 October 2002
- DoD, 2003 *DoD Defense Acquisition System Directive*, 5000.1, 12 May 2003
- DoD, 2003a *US DoD Joint Chiefs Manual CJCSI 3170.01, Joint Capabilities Integration and Development System*, 24 June 2003
- DoD, 2003b *US DoD Instruction 5000.2, Operation of the Defense Acquisition System*, 12 May 2003
- DoD 2003c *CJCSI 3170.01C, Joint Capabilities Integration And Development System*, DoD, Washington, DC, May 2003
- DoD, 2003d *DoD Architecture Framework Version 1.0, Vol 1 Definition and Guides*, DoD Architecture Framework Working Group, 15 Jan 2003
- DoD, 2003e [MIL-STD-961E Defense And Program-Unique Specifications Format And Content, US Department of Defense, August 2003]
- DoD, 2003f Project SESS-00401, USD (AT&L), 1 May 2003]
- DoD, 2004 *Policy for Systems Engineering in DoD*, USD, AT&L Feb 20, 2004
- DoD, 2005 *DoD Enterprise Architecture Technical Reference Model v0.04*, DoD EA Congruence Community of Practice, Aug 2005, can be seen at http://www.defenselink.mil/cio-nii/docs/DoD_TRM_V0.4_10Aug.pdf. Accessed August 2007
- DoD, 2007a *DoD Architecture Framework 1.5, Volume 1, Definitions and Guidelines*, US Department of Defense, April 2007
- DoD, 2007b *Systems Engineering Plan Preparation Guidance, Version 2.0*, OUSD(AT&L), 2090 Defense Pentagon, Washington, DC, October 18, 2007
- Donne, 1621 *The Anatomy of the World*, John Donne, 1621
- Douglas 1985 *Creative Interviewing*, Douglas, J.D., Sage Library of Social Research, Sage Publications, ISBN 0-8039-2160-8, 1985
- DPCA 1971 *Government Organisation for Defence Procurement and Civil Aerospace*, Rayner, D., Cmnd. 4506, HMSO, April 1971
- Dreyfus, 1986 *Mind Over Machine. The power of human intuition and expertise in the era of the computer*, Dreyfus, H.L., Dreyfus, S.E., Free Press, New York, ISBN: 0029080606, 1986
- DTC, 2007 *Review of Human Factors Integration from within and outside of the MOD – Interim Report*, Report 338133, Human Factors Integration Defence Technology Centre, Work Package 3.1.2, April 2007
- DTC, 2007a *Review of Human Factors Integration from within and outside of the MOD – Interim Report*, Report 338133, Human Factors Integration Defence Technology Centre, Work Package 3.1.2, April 2007

REFERENCES

- DTC, 2007b *Review of Human Factors Within and Outside MOD*, http://www.hfidtc.org/WP3/wp3_1_2.htm
- DUSD, 1994 *Reengineering the Acquisition Oversight and Review Process. Acquisition Reform Process*, DUSD(AR) position to report to USD(A&T), Action Team. Vol 1, 1994
- EAC, 2006 *Enabling Acquisition Change*, McKane, T., MOD Report , June 2006
- EB, 1797 *Encyclopaedia Britannica*, Bell and MacFarquhar, Edinburgh, 1797.
- EIA, 1999 ANSI/EIA-632-1998, *Processes for Engineering a System*, Electronic Industries Alliance, Arlington, VA, Jan 1999
- Elliott, 1998 *The Sociology of the Professions*, Elliott, P., Macmillan, London, 1972
- Elliott, 2005 *Systems and the Law*, Elliott, C. System Safety Conference , IEE, 2005
- Emes, 2005 *Confronting an Identity Crisis – How to “Brand” Systems Engineering*, Systems Engineering Vol 8, 2, pp164-185, Wiley InterScience, 2005
- FEAF, 1999 *Federal Enterprise Architecture Framework (FEAF), Version 1.1*, US Federal CIO Council, 1999
- Feynman, 1964 *The Feynman Lectures on Physics*, 1964-6, Vol. 1 (Addison Wesley 1970, ISBN 0-201-02115-3
- Fidler, 2007 *Chairman’s Introduction, Registration Standards*, Fidler, K. , Engineering Council, 10 Maltravers St, London WC2R 3ER, 2007
- Finklestein, 1983 *Review of Design Methodology*, Finklestein L., Finklestein, A.C.W., Vol. 130, Pt A, No 4, p217, 1983
- Flaccus, 56BC Writings of Quintus Horatius Flaccus, 56 BC
- Flood, 1991 *Critical Systems Thinking*, Flood, R.L. Jackson M.C., Wiley & Sons, Chichester, ISBN 0-471-93098-91991,
- Flood, 1993 *Dealing with Complexity: An Introduction to the Theory and Application of Systems Science*, Flood,R.L. Carson, E.R., Springer, ISBN 030644299X, 1993
- Forrester, 1961 *Industrial Dynamics*, Forrester, J. W., MIT Press, Cambridge, Mass., 1961
- Forsberg, 1991 *The Relationship of System Engineering to the Project Cycle*, Forsberg, K., Mooz, H., Joint Conference sponsored by the National Council of System Engineers and the American Society for Engineering Management, Chatanooga, TN, October 1991.
- Forsberg, 1995 *Application of the “Vee” to Incremental and Evolutionary Development*, Forsberg, K., Mooz, H., Proceedings of the Symposium of the NCOSE, 1995
- Forsberg, 1996 *Visualizing Project Management*, Forsberg, K., Mooz, H. , Wiley and Son, ISBN 0 471 57779 0, 1996
- Foucault, 1966 *The Order of Things: An Archaeology of the Human Sciences*, Foucault,M.,1966, trans. Pantheon Books 1970
- Freidson, 1986 *Professional Powers: A Study of the Institutionalization of Formal Knowledge*, Freidson, E., Chicago, IL: University of Chicago Press Chicago, 1986
- Freidson, 1994 *Professionalism Reborn: Theory, Prophecy, and Policy*, Freidson, E., Chicago, IL: University of Chicago Press ISBN 978-0-226-26221-5, 1994
- Fulbright, 1966 Sen.J.W.Fulbright, US Foreign Relations Committee Hearings, 1966
- Gantt, 1917 *Gantt Chart*, Gantt, H.L., 1917
- Georgescu-Roegen, 1971 *The Entropy Law and the Economic Process*, Georgescu-Roegen, Harvard University Press, Cambridge, Mass, 1971
- Gordon, 1978 *Structures: or, Why Things Don't Fall Down*, Gordon, J.E., Penguin Books, 1978
- Gould, 1972 *Punctuated Equilibria*, Gould,S.J., Eldredge,N. pp. 82-115, in Models in Paleobiology. ed. Schopf, T.J.M.,San Francisco, Freeman Cooper. 1972
- Griffin, 2007 *Boeing Lecture 2007*, Griffin M.D., Purdue University, March 2007
- Gruhl, 1992 *Lessons Learned, Cost/Schedule Guide*, Gruhl,N. NASA Comptroller's Office, 1992

REFERENCES

- Hall, 1962 *A Methodology for Systems Engineering*, Hall, A.D., Van Nostrand, Princeton, NJ, 1962
- Hall, 1989 *Metasystems Methodology*, Hall A. D., Pergamon Press, Oxford, England, 1989
- Halsey, 2004 *A History of Sociology in Britain: Science, Literature, and Society*, Halsey, A. H., University of Oxford Press, ISBN 0199266603, 2004
- Hitchins, 1992 *Putting Systems to Work*, Hitchins, D.K., John Wiley & Sons, 1992
- Hitchins, 2003 *World Class Systems Engineering – the five layer model*, Hitchins, D.K., <http://www.hitchins.co.uk>, accessed August 2003
- Hitchins, 2003 *Advanced Systems Thinking, Engineering and Management*, Hitchins D., Artec House, ISBN 1-58053-619-0, 2003
- HOC, 1998 *House of Commons Defence Committee, Seventh Report, 1997-98*, Defence Committee Publications, ISBN 0 10 554327 6
- HOC, 1999 *House of Commons Defence Committee, Session 1997-9*, Defence Committee Publications, ISBN 0 10 555025 6
- Hobbes, 1651 *Leviathan*, Hobbes, T., 1651, etymologically referenced in OED, 2004
- Hodgson, 1991 *The X-Model: A process model for object-oriented software development*, Hodgson, R., Proceedings International Conference on Software Engineering and its Application, Toulouse, pp. 713-728, 1991
- Hofmeister, 2007 *A General Model of Software Architecture Design Derived from Five Industrial Approaches*, Hofmeister, et al. Journal of Systems and Software 80 2007, p106-126
- Honour, 2004 *Value of Systems Engineering*, Honour,E.C., Axelband,E., Rhodes,D.H., Lean Aerospace Consortium Technical Report, Sep 2004
- Hughes, 1998 *Rescuing Prometheus: Four Monumental Projects That Changed Our World*, Hughes, T.P., Pantheon Books, New York, NY, ISBN: 9780679739388, 1998
- Hume, 1739 *A Treatise of Human Nature, I*, Hume, D., 1739, Oxford University Press, ISBN 10 019926385X, 2007
- Hume, 1752 *On the Balance of Trade*, Hume, D., 1752
- iCMM, 2001 *integrated Capability Maturity Model Version 2.0 (iCMM)*, Federal Aviation Administration, Washington, D.C., 2001
- IEC, 2005 *61508 Functional Safety - Safety related systems*, International Electrotechnical Commission, 1211 Geneva 20 Switzerland, 2005
- IEEE, 1990 *IEEE Std 610, Standard Glossary of Software Engineering Terminology*, Institute of Electrical and Electronic Engineering, New York, NY10017, 2000 1990
- IEEE, 1998 *IEEE 1220-1998, Standard for Application and Management of the Systems Engineering Process*, 345 East 47th Street, New York, NY 10017, USA
- IEEE, 1998a *IEEE/EIA 12207.0, Standard for Information Technology - Software Life Cycle Processes*, Institute of Electrical and Electronic Engineering, New York, NY10017, ISBN 0 7381 2518 0, 2000
- IEEE, 2000 *IEEE Std 1471-2000, IEEE Recommended Practice for Architectural Description of Software Intensive Systems*, Institute of Electrical and Electronic Engineering, New York, NY10017, ISBN 0 7381 2518 0, 2000
- IET, 2006 *Capability Engineering: at Home and Abroad*, IET Forum, 7 November 2007, see http://www.iee.org/oncomms/pn/systemseng/CF_review.pdf and IET.TV
- INCOSE, 2003 *G2SEBoK, Guide to the Systems Engineering Body of Knowledge*, INCOSE, 2003, at <http://g2sebok.incose.org/app/mss/menu/index.cfm>. Accessed November 2007
- INCOSE, 2005 *Systems Engineering Core Competencies Framework*, The UK Advisory Board of the INCOSE UK Chapter, 2005
http://www.incose.org.uk/Downloads/Systems%20Engineering%20Core%20Competencies%20Issue%201b%20_2_.pdf
- INCOSE, 2007 *INCOSE Certification Program*
<http://www.incose.org/educationcareers/certification/index.aspx>

REFERENCES

- INCOSE, 2007a *Systems Engineering Vision 2020*, International Council on Systems Engineering, 2007
- INCOSE, 2007a <http://www.incose.org/practice/whatisssystemseeng.aspx>, accessed August 2007
- INCOSE, 2007b *INCOSE Handbook Version*, 3.1 Aug 2007
- ISO, 1995 ISO/IEC 12207, *Information technology — Software life cycle processes*, International Standardisation Organisation/International Electrotechnical Commission, 1211 Geneva 20 Switzerland, 1995
- ISO, 1998 *Information Technology - Software Process Assessment: Part 2 : A reference model for processes and process capability (normative)*, International Standardization Organization/International Electrotechnical Commission, CH-1211 Geneva 20, Switzerland, 1998
- ISO, 2000 ISO 9001:2000, *Quality management systems — Requirements*, International Standardisation Organisation/International Electrotechnical Commission, 1211 Geneva 20 Switzerland, 2000
- ISO, 2002 ISO/IEC 15288:2002, *Systems engineering – System life cycle processes*, International Standardization Organization/International Electrotechnical Commission, CH-1211 Geneva 20, Switzerland, 2002.
- ISO, 2002a ISO/IEC 15504, *Information technology – Process Assessment, Parts 1 – 6*, International Standardisation Organisation/International Electrotechnical Commission, 1, rue de Varembe, CH-1211 Geneve 20, Switzerland, 2002
- ISO, 2003 ISO/IEC TR 19760, *Systems engineering – A guide for the application of ISO/IEC 15288 (System life cycle processes)*, International Standardization Organization/International Electrotechnical Commission, CH-1211 Geneva 20, Switzerland, 2002.
- ISO, 2003a ISO/PAS 18152:2003, *Ergonomics of human-system interaction -- Specification for the process assessment of human-system issues*, International Standardisation Organisation, 1211 Geneva 20 Switzerland, 2003
- ISO, 2003b ISO/IEC 15504-2:2003, *Information technology -- Process assessment -- Part 2: Performing an assessment*, International Standardisation Organisation/International Electrotechnical Commission, 1211 Geneva 20 Switzerland, 2003
- ISO, 2004 ISO/IEC 15504, *Information technology – Process Assessment, Parts 1 – 5*, International Standardisation Organisation/International Electrotechnical Commission, 1, rue de Varembe, CH-1211 Geneve 20, Switzerland, 2004
- ISO, 2006 ISO/IEC 15504-5:2006, *Information technology – Process Assessment – Part 5: An exemplar Process Assessment Model*, International Standardization Organization/International Electrotechnical Commission, CH-1211 Geneva 20, Switzerland, 2002.
- ISO, 2006a ISO/IEC 15289:2006 *Systems and software engineering -- Content of systems and software life cycle process information products (Documentation)*, International Standardisation Organisation/International Electrotechnical Commission, 1, rue de Varembe, CH-1211 Geneve 20, Switzerland, 2006
- ISO, 2007 ISO/IEC FCD 24765, *Systems and software engineering vocabulary Edition: 1*, Stage: 40.20, International Standardisation Organisation, 1211 Geneva 20 Switzerland, 2007
- Jackson, 1984 *Towards a System of Systems Methodologies*, Jackson, Journal of Operational Research, Vol 35, Operational Research Society, 1984
- Johnson, 1973 *The Theory and Management of Systems*, Johnson, R., Kast, F., and Rosenzweig, J. 3d ed. New York, McGraw-Hill, 1973
- Johnson, 1987 *The Role of Man in the Systems Design Process: the Unresolved Dilemma*, Johnson, E. M., Chapter 12 in *System Design: Behavioral Perspectives on Designers, Tools and Organization*, Rouse, W., Boff, K. (eds.) North-Holland, New York, 1987 ISBN: 0-444-01230-3
- Jones 1999 Karl Marx, quoted in *Almost Like a Whale*, Jones S., Doubleday, 1999
- Jordan, 1968 *Themes in Speculative Psychology*, Jordan, N. London, Tavistock, 1968

REFERENCES

- Kant, 1781 *Critique of Pure Reason*, Kant, I., 1781, trans. Smith, N.K., St. Martins, New York, 1965
- Kennedy, 1961 *Inaugural Address*, Kennedy, J.F., Washington, 20 January 1961
- Kerzner, 2003 *Project Management: A Systems Approach to Planning, Scheduling, and Controlling*, Kerzner, H., 8th Ed., Wiley, ISBN 0-471-22577-0, 2003
- Kipling, 1902 *The Elephant's Child, Just So Stories*, Kipling, R., 1902
- Kline, 1995 *The Conceptual Foundations for Multi-Disciplinary Thinking*, Kline, S.J., Stanford University, Stanford CA, 1995
- Koehler, 1938 *The Place of Values in the World of Fact*, Koehler, W., Liveright, 1938
- Kruchten, 1995 *Architectural Blueprints – the “4+1” View Model of Software Architecture*, Kruchten, P., IEEE Software 12(6), pp.42-50, 1995
- Kuhn, 1962 *The Structure of Scientific Revolutions*, Kuhn, T.S., University of Chicago Press, Chicago, IL, ISBN 0-226-45808-3, 1962
- Landauer, 1991 *Information is Physical*, Physics Today, Landauer, R., 44, 23-29, 1991
- Le Lann, 1997 *An Analysis of the Ariane 5 Flight 501 Failure - a System Engineering Perspective*, Le Lann, G., Engineering of Computer-Based Systems Proceedings., pp.339 – 346, Mar 1997
- Levis, 2000 *C4ISR Architectures: I, Developing a Process for C4ISR Architecture Design*, Systems Engineering Journal, Wiley pp.225-247
- Lucretius, *Nat.* *De Rerum Naturā*, Lucretius, 94BC, Oxford University Press 1947
- Machiavelli, 1515 *The Prince*, Machiavelli, N., 1515, Bantam Classics ISBN 10 0553212788, 1984
- Magee, 2004 *Complex System Classification*, Magee, C.L., de Weck, O.L., Fourteenth Annual International Symposium of the International Council On Systems Engineering (INCOSE) 20 June – 24 June 2004
- Malthus, 1798* *Essay on the Principle of Population*, Malthus, R.T. Oxford University Press, ISBN 01928374781999
- Martin, 2007 *Systems Are Imaginary – Systems Are Not Real, Some Thoughts on the Nature of Systems Thinking*, Martin, J. INCOSE International Symposium, San Diego, 2007
- Mazza, 1994 *Software Engineering Standards*, Mazza, C., Fairclough, J., Melton, B., de Pablo, D., Scheffer, A., Stevens, R., Prentice Hall International (UK) Ltd, ISBN 0 13 10658 9, 1994
- Mede, 1641 Mede, J., The Apostasy of the Latter Times, 64, 1641, cited OED, 2002, system, defn I.,1
- Midwinter, 2000 *Something old, something new and something just in time*, J.E.Midwinter, Engineering Science and Education Journal, Vol 9 Issue 4, Oct 2000
- Miller, 1978 *Living Systems*, Miller, J.G., McGraw-Hill, New York, NY, ISBN 0-87081-363-3, 1978
- Miller, 1992 *Greater than the sum of its parts: Subsystems which process both matter-energy and information*, Miller, J.L., Miller, J.G., Behavioral Science, 37, 1992.
- Mindpapers, 2007 *A Bibliography of the Philosophy of the Mind and Science of Consciousness*, <http://consc.net/mindpapers/2.1b>, accessed October 2007
- MOD, 1999 *Smart Procurement Initiative Implementation, Stage 2 (Final) Report*, 29 Feb 1999
- MOD, 1999a *Acquisition Handbook*, Defence Smart Procurement Team, UK Ministry of Defence, April 1999
- MOD, 2003 *Smart Acquisition Handbook*, UK Ministry of Defence, January 2003
- MOD, 2005 *The Acquisition Handbook, Edition 6, MOD, October 2005*
- MOD, 2005a *MOD Architectural Framework, Executive Summary v1.0*, MODAF Partners, August 2005
- MOD, 2004 *Smart Acquisition Handbook*, UK Ministry of Defence, January 2004

REFERENCES

- MOD 2006 <http://www.mod.uk/DefenceInternet/MicroSite/DPA/>, accessed Aug 2006
- MOD, 2006b *Enterprise Architecture, EA Policy – U_ver 2*, DGInfo-6-5_IX, 23 Nov 2006, can found on <http://www.modaf.com>
- MOD, 2007 *Capability Management Handbook (Interim Edition)*, TLMC Workstrand, MOD 26 Feb 07, accessible at <http://ams.mod.uk/content/docs/ecc/capmanha.doc>. accessed October 2007
- MOD, 2007a <http://www.mod.uk/defenceinternet/microsite/des/ourteams/integratedprojectteamsipts.htm>, accessed August 2007
- MOD, 2007a http://www.dr.mod.uk/glossary_2nd.htm#CALSM, accessed October 2007
- MOD, 2007b <http://www.mod.uk/DefenceInternet/AboutDefence/Organisation/KeyFactsAboutDefence/SmartAcquisition.htm>, accessed May 2007
- MOD, 2007c http://ams.mod.uk/content/docs/ils/ils_web/plcs/whyinmod.htm, accessed October 2007
- MOD, 2007c <http://www.ams.mod.uk/content/docs/ecc/ecchandb.pdf> accessed May 2007
- MOD, 2007d <http://www.mod.uk/DefenceInternet/AboutDefence/Organisation/KeyFactsAboutDefence/SmartAcquisition.htm>. accessed May 2007
- MOD, 2007e http://www.ams.mod.uk/content/docs/peopacq/alts/handbook/handbook_section_05_acf_introduction.doc, accessed Aug 2007.
- MOD, 2007f http://www.ams.mod.uk/content/docs/reqweb/content/req_plan.htm. accessed April 2007
- MODAF, 2007 <http://www.modaf.org.uk/>, accessed August 2007
- MODAF, 2007 <http://64.233.183.104/search?q=cache:RZ5DxQ8EXWwJ:www.modaf.com/FAQ/+MODAF+Enterprise+Architecture+Framework&hl=en&ct=clnk&cd=1&gl=uk>, accessed October 2007
- MODAF, 2007b <http://www.modaf.org.uk/3Modelling/65/what-are-the-modaf-viepoints>, Accessed October 2007
- MODAF, 2007c MODAF Version 1.1. SV-1 Resource Interaction Specification, <http://www.modaf.org.uk>, accessed Oct 2007
- Moray, 1994 *Error Reduction as a Systems Problem*, Moray, N, pp 67,91, Ch 5 in *Human Error in Medicine*, Bognor, M.S., (Ed.), Lawrence Erlbaum Associates Publishers, 1994
- MOS, 1958 Ministry of Supply study in 1958
- MOT, 1968 *Report of the Steering Group on Development Cost Estimating*, Ministry of Technology, Downey, 1968
- Nagel, 1986 *The View From Nowhere*, Nagel, T. Oxford University Press, ISBN is 0195056442, 1986
- NAO, 2002 *Ministry of Defence: Implementation of Integrated Project Teams* Report by the Comptroller and Auditor General, HC 671, NAO, ISBN: 010291445, 2002
- NAO, 2002a *Report by the Comptroller and Auditor General*, HC 671 2001-2002, 14 March 2002, ISBN: 0102914451
- NAO, 2003 Ministry of Defence, *Through-Life Management*, Report by the Comptroller and Auditor General Session 2002-20, NAO Report, ISBN 0102921539, 21 May 2003
- NAO, 2004 *Ministry of Defence: Major Projects Report 2003*, Comptroller and Auditor General, The Stationary Office Report, ISBN: 0102926581 Jan 2004
- NAO, 2005 *Driving The Successful Delivery Of Major Defence Projects: Effective Project Control Is A Key Factor In Successful Projects*, HC 20, NAO, 2005
- NASA, 1972 *Managing NASA in the Apollo Era*, SP-4102, Ch 7, History Division, NASA, 1972
- NATO, 2004 *Proposed NATO Policy for System Life Cycle Management*, Conference of National Armament Directors, NATO/PFP Unclassified Document PFP(CNAD)D(2004)0011 + PFP(AC/327), 27 September 2004
- Newton, 1687 *Philosophiae Naturalis Principia Mathematica*, Newton, I., 1687

REFERENCES

- Norris, 1983 *Editorial*, Norris, W.T., IEE Proc Vol 130, Pt. A, No.4 June 1983
- NRC, 2007 *Human-System Integration in the System Development Process*, R Pew, A Mavor, editors, National Research Council of the National Academies, The National Academic Press, ISBN 978-0-309-10720-4, 2007
- OED, 2004 *Oxford English Dictionary* , Oxford University Press, 2004
- Oliga, 1986 *Methodology in Systems Research: The Need for a Self-reslective Commitment*, Mental Images, Values, and Reality, Society for General Systems Research, Louisville, KY, 1986
- O'Reilly, 2004 Presidential Address, O'Reilly, J, IEE Journal, 2004
- OUSD, 1995 *Acquisition & Technology)/Acquisition Program Integration. Rules of the Road — A Guide for Leading Successful Integrated Product Teams*, Preview, DPA, MOD, May 2006,
- Paley, 1819 *Natural Theology*, Paley, W., XXV 1819, (OED, 'system' definition I.1.a)
- Parsons, 1961 *Theories of Society: foundations of modern sociological theory*, Parsons, T.E.F., Free Press, New York, 1961
- PE, 1988 *A Report on the Arrangements for Managing Major Projects in The Procurement Executive*, Jordan, Lee and Cawsey, PE, MOD, 1988
- Penrose, 2004 *The Road to Reality*, Penrose, R., Random House, 2004
- Perry, 1994 *Specification & Standards – A New Way of Doing Business*, Perry, W.J. Defense Secretary, DoD, 29 June 1994
- Petroski, 1994 *Design Paradigms : Case Histories of Error and Judgment in Engineering by Henry Petroski (Paperback - May 27 1994]*
- Plato, Crat. *Cratylus*, Plato, Hackett Publishing, ISBN 0872204162, 1998
- Plato, 375BC *The Republic*, Plato, Penguin Books Ltd, London, ISBN 0 14 044048 8, 1987
- Plato, Tim. *Timaeus* , Plato, Hackett Publishing, ISBN 0872204464, 2000
- PMBOK, 2004 *A Guide to the Project Management Body of Knowledge, Third Edition*, Project Management Institute, Pennsylvania, 19073, 1-936099-45-X
- PMI, 1998 *The Project Management Institute Project Management Handbook*, PMI, Jossey Bass. ISBN-10: 0787940135, 1998
- Polanyi, 1968 *Life's Irreducible Structures*, Polanyi, M., Science Vol. 160, 1968
- Preview, 2007 *Preview*, Issue 157, DPA, MOD, March 2007
- Quine, 1960 *Word and Object*, Quine, W.V., Wiley, New York, ISBN 0-262-67001-1, 1960
- Ramo, 1998 *The Systems Approach*, Ramo, S and St Clair, R., KNI Inc, Anaheim, CA, 1998
- Ratcliffe, 2002 *Scenario planning: strategic interviews and conversations*, Ratcliffe J., Foresight - The Journal of Future Studies, Strategic Thinking and Policy, January 2002
- Rechtin, 1991 *Systems Architecting: Creating and. Building Complex Systems*, Rechtin, E., Prentice Hall, ISBN 0-13-880345-5, 1991
- RH, 1966 *Random House Unabridged Dictionary*, First edition, Merriam-Webster, 1966
- Rhodes, 2004 *The Case for Evolving Systems Engineering as a Field within Engineering Systems*, Rhodes D., Hastings D. *Engineering Systems Symposium*, MIT, March 2004]
- Rosenbleuth, 1943 *Behavior, Purpose and Teleology*, Rosenblueth, A., Wiener, N., Julian Bigelow, J, Philosophy of Science, 10, S. 18–24, 1943
- Ross, 2000 *Dennett's Philosophy*, Ross, D., Brook, A., Thompson, D., ISBN 0262182009, 2000
- Royce, 1970 *Managing the Development of Large Software Systems: Concepts and Techniques*, Royce, W.W, WESCON Technical Papers 14 1970
- Sandom, 2004 *Human Factors for Engineers*, Sandom,C., Harvey,R., IEE, ISBN 0 86341 329 3, 2004

REFERENCES

- Satre, 1943 *Being and Nothingness: An Essay on Phenomenological Ontology*, Sartre, J.P., Routledge Classics, ISBN 0415278481, 1943
- Schaffer, 2003 *The Myth of the Gruhl Curve*, Schaffer, M., unpublished Cost Analysis Division document, NASA., 2003
- SDR, 1998 *Strategic Defence Review Study 2F3/7 Smart Procurement Final Report Issue 1*, PPB/P(98)2, 19 January 1998, p5
- SEI, 2002 *Capability Maturity Model Capability Maturity Model® Integration (CMMISM)*, Version 1.1 Staged Representation, CMU/SEI-2002-TR-012, Carnegie Mellon University, Pittsburgh, PA 15213-3890 2002
- Senge, 1990 *The Fifth Discipline: The Art & Practice of The Learning Organization*, Senge, P.M., Doubleday Currency, ISBN 0-385-26095-4, 1990
- SFIA, 2003 *SFIA 2.0, Skills Framework for the Information Age*, SFIA Foundation, London EC4 1RS, 2003
- Shaeffer, 2005 *Technical Excellence through Systems Engineering*, Schaeffer, M., Boeing Technical Excellence Conference, July 26, 2005].
- Shakespeare, 1597 *Romeo and Juliet*, Shakespeare W, Act II, ii, 1-2 1597
- Shakespeare, 1623 *As You Like It*, Shakespeare, W., II.vii.139-166, 1623
- Shannon, 1948 *A Mathematical Theory of Communication*, C. E. Shannon, Vol 27, The Bell System Technical Journal, July, October, 1948
- Shaw, 1984 *Abstraction Techniques in Modern Programming Languages*, Shaw, M., IEEE Software, Vol 1(4), Oct 1984
- Sheard, 1997 *The Frameworks Quagmire, A Brief Look*, Sheard, S.A. Proceedings of INCOSE Symposium, Los Angeles, August 1997
- Sheard, 2000 *The Shangri-La of ROI*, Sheard, S., INCOSE International Symposium, Minneapolis, MN, 2000
- Shewhart, 1931 *Economic Control of Quality of Manufactured Product*, Shewhart, W.A., Van Nostrand, ISBN 0-87389-076-0, 1931
- Singleton, 1974 *Man-Machine Systems*, Singleton, T., Penguin Education ISBN 0 14 08.0568 0
- Sowa, 1992 *Extending and formalizing the framework for information systems architecture*, Sowa, J.F. Zachman, J.A., IBM Systems Journal, Vol 31, No 3, 1992
- Stevens, 1998 *Systems Engineering, Coping with Complexity*, Stevens, R., Brook, p., Jackson, K., Arnold, S., Prentice Hall Europe, ISBN 0 13 095085 8, 1998
- Strogatz, 2003 *Sync: Rhythms of Nature, Rhythms of Ourselves*, Strogatz, S.H., Allen Lane. 2003
- TAFIM, 1994 *Technical Architecture Framework For Information Management (TAFIM) Volumes 1-8, Version 2.0*, DISA Center for Architecture, U.S. Department Of Defense, Reston, VA, 1994
- Taylor, 1911 *The Principles of Scientific Management*, Taylor, F.W., New York Harper & Brothers 1911
- TEAF, 2000 *Treasury Enterprise Architecture Frameworkn Version 1*, US Department of the Treasury July 2000 .
- Tebbit, 2005 *Acquisition reform has put MoD on a robust pathway to the future*, Tebbit, Preview, DPA, MOD, Oct 2005
- TLMP, 2007a <http://ams.mod.uk/content/docs/tlmguide/tlmstrat.doc>, Accessed Oct 2007
- TLMP, 2007b <http://www.aof.mod.uk/content/docs/tlmguide/tlmplnpr.doc>, Accessed Oct 2007
- TLMP, 2007c <http://www.ams.mod.uk/content/docs/tlmguide/matmodv3-1.doc>, Access Oct 07
- TOGAF, 2007 <http://www.opengroup.org/architecture/togaf7-doc/arch/> Accessed August 2007
- Tran Van Doan, 1997 *Habermas on Politics*, Tran Van Doan, J.B., National Taiwan University, Taiwan, 1997
- Tribus, 1971 *Energy and Information*, Tribus M., McIrvine E.C., v 225, Scientific American, September, 1971

REFERENCES

- Tully, 1993 *System Development Activity*, Tully, C., Chapter 3 in *Systems Engineering*, ed. B.Thome, Wiley, ISBN 0 471 93552 2, 1993
- UK-SPEC, 2004 United Kingdom Standards of Professionalism, Engineering Council, May 2004
http://www.ukspec.org.uk/files/CE_IE.pdf
- USAF, 1966 *AFSCM 375 -5 - System Engineering Management Procedures*, Headquarters, Air Force Systems Command, Washington, DC,1966
- USAF, 1974 *Mil-Std-499A (USAF)*, 1974, Engineering Management, US Department of Defense, Washington, DC, 1974
- USAF, 2005 *Official USAF Biography, Schriever, B.A.*,
 (<http://www.au.af.mil/au/aul/school/ots/schriever.htm>, accessed January 2007
- Webster, 1995 *Webster's Second New International Unabridged Dictionary*, Merriam-Webster , ISBN: 9780877794684, 1995
- Weiner, 1948 *Cybernetics: or, Control and Communication in the Animal and the Machine*, Weiner, N., The Technology Press of M.I.T. and John Wiley and Sons, New York, 1948
- Vilkomir, 2004 Combining Agent-Oriented Conceptual Modelling with Formal Methods, Vikomir,S., Ghose, S., Krishna., A., Australian Software Engineering Conference 2004
- Wikipedia, 2007 http://en.wikipedia.org/wiki/Zachman_framework, Accessed August 2007
- Yates, 2000 *Origins of Project Management*, Yates,J., Knowledge Management Magazine, Sloan School of Management, MIT, August 2000,
- Zachman, 1987 *A Framework for Information Systems Architecture*, Zachman, J.A., IBM Systems Journal, vol. 26, no. 3, IBM Publication G321-5298], 1987
- Zachman, 1992 Unpublished lecture given by John Zachman at *Data/Client Server World*, 7 December 1992, Chicago - Verbatim transcript by Richard.A.Martin (personal communication)

APPENDIX A SYSTEMS ENGINEERING CAPABILITY INDICATORS

The identification of system life cycle process work product, as defined in Figure 6.3 was followed until a stable set of work product classes and descriptions was achieved and all the objects implied in the text of ISO/IEC 15288:2002 were associated with a work product. Minimal change to the implied titles of the work products was undertaken. Likewise, the characteristics of each product are essentially verbatim from the text of this system life cycle process reference model.

The classification structure that resulted from repeated cycles around the procedure described in Figure 6.3 is listed in the following Table. Each work product class member thus inherits the generic typical characteristics of the class to which it was assigned. To these are added the individual characteristics that are listed in the column to the right of the following table.

Specific Work Product types used in individual organisations are created by their process owners and applied by the process users in order to satisfy an outcome of a particular process purpose.

The list below sets out for corporate and individual project specifiers of business procedure and processes a frame of reference and check list to guide them as they define the work products mandated and expected in the course of conducting a system life cycle. It also guides those identifying that appropriate work product exist as evidence of the capable execution of processes.

ISO/IEC 15288 WORK PRODUCTS

NOTE: Generic Work Product types are included in the list for completeness.

WP ID	Work Product Name	Work Product Typical Characteristics
1.00	object	refer to B.1 for Generic Work Product typical characteristics
1.01	supplied system	<ul style="list-style-type: none"> – agreement – stakeholder requirements – delivery conditions – acceptance criteria – compliance deviations – handling and storage – safety and security conditions – transfer of ownership
1.02	enterprise resource assignment	<ul style="list-style-type: none"> – project personnel profile – working environment – projects materials – projects services – resource capacity – resource needs profile – multi-project resource conflicts
1.03	competent personnel	<ul style="list-style-type: none"> – staff experience, skills and knowledge – competence to perform life cycle processes – team working ability – staff assessments – training and education needs – retraining, reassignment or reallocation plans – recruitment criteria
1.04	information item	<ul style="list-style-type: none"> – storage medium – security classification – privacy requirements – access rights – storage environment

		<ul style="list-style-type: none"> – information legislation – currency and validity
1.05	system element	<ul style="list-style-type: none"> – delivery schedule – acquisition agreement – architecture design – interface control descriptions – implementation technology – acceptance criteria – compliance deviation – useful life – handling instructions – storage and release conditions
1.06	qualified operators	<ul style="list-style-type: none"> – operator competence definition – operator selection criteria – training criteria – qualifications – operating instructions – failure detection instructions – authorization to operate – training resources – system training mode – service availability
1.07	integrated system	<ul style="list-style-type: none"> – architectural design – assembly sequence – integration non-conformances – configuration status – verification readiness
1.08	verified system	<ul style="list-style-type: none"> – conformance evidence – configuration status – deviations – corrective actions
1.09	installed system	<ul style="list-style-type: none"> – delivery schedule – intermediate storage – installation location – operational location – operational readiness – operational data
1.10	validated system	<ul style="list-style-type: none"> – service requirements – operational site – qualified operators – service non-conformances – operational status

Appendix A SYSTEMS ENGINEERING CAPABILITY INDICATORS

		<ul style="list-style-type: none"> – acceptance condition
1.11	operational system	<ul style="list-style-type: none"> – service requirements – service availability – service life – service non-conformances – maintenance conditions – operators
1.12	waste products	<ul style="list-style-type: none"> – waste disposal conditions – waste disposal equipment – system destruction procedure – treaded waste – waste handling procedure – melted, crushed, incinerated or demolished system elements – knowledge safeguards – operator skills safeguards
1.13	disposed system	<ul style="list-style-type: none"> – current condition – disassembled elements – useful remaining life – reuse conditions – operator redeployed/retirement – reconditioning actions – value to other systems – storage conditions
1.14	supplier payment	<ul style="list-style-type: none"> – contract conditions – work offset – acceptance confirmation
2.00	description	refer to B.1 for Generic Work Product typical characteristics
2.01	tailored system life cycle stage model	<ul style="list-style-type: none"> – stage purpose – stage outcomes – enabling system services – succeeding stage criteria – stage exit criteria – approval to proceed
2.02	tailored system life cycle process model	<ul style="list-style-type: none"> – process purpose – process outcome – activity identity and detail – output work products – input work products

2.03	system life cycle stage model	<ul style="list-style-type: none"> – business strategy – business area strategy – stages – gates – gate review achievement criteria – gate review authorities – system life cycle management policy and procedures – system life cycle performance measures – risk management policies and procedures
2.04	system life cycle process model	<ul style="list-style-type: none"> – policy to adapt system life cycle processes – life cycle process performance/trends – project requirements and needs – quality management policies and procedures
2.05	configuration item list	<ul style="list-style-type: none"> – identifiers or markings – relevant standards – product sector conventions – specification or description identity – configuration baseline descriptions – released item configuration – baseline timing and conditions – authorizations
2.06	information item list	<ul style="list-style-type: none"> – identifier – security classification – privacy requirements – access rights – storage environment – distribution schedules
2.07	stakeholders profile	<ul style="list-style-type: none"> – stakeholder classes – stakeholder identity – acquirer organizations – supplier organizations – regulatory bodies – members of society – stakeholder representatives
2.08	system functional model	<ul style="list-style-type: none"> – product function – performance parameters – data flows – quality in use measures – functional views and viewpoints – modelling notation

Appendix A SYSTEMS ENGINEERING CAPABILITY INDICATORS

2.09	architectural design description	<ul style="list-style-type: none"> – system elements – structure views and viewpoints – non-functional views and viewpoints – design rationale – interface requirements – implementation technology – planned evolution – technology upgrades
3.00	plan	refer to B.1 for Generic Work Product typical characteristics
3.01	acquisition strategy	<ul style="list-style-type: none"> – acquisition plan – life cycle model – make/buy/reuse policy – supply chain relationships – market influences – technology investment strategy
3.02	supply strategy	<ul style="list-style-type: none"> – enterprise objectives – business area objectives – asset base – resource availability – service offerings – product families
3.03	business strategy	<ul style="list-style-type: none"> – new business opportunities – business areas – opportunity viability – risks to organization – projects prioritisation – current product/service portfolio – current assets – resource availability – investment strategy
3.04	system life cycle management policy	<ul style="list-style-type: none"> – business achievement criteria – life cycle reference models – stage gate/milestones approval criteria – system life cycle processes improvement policy
3.05	resource allocation plan	<ul style="list-style-type: none"> – project resource needs – multi-project interfaces – resource capacities – common enabling systems – common system elements

3.06	system life cycle process policy	<ul style="list-style-type: none"> – system life cycle process definitions – life cycle reference models – process application policy – tailoring/adaptation policy
3.07	training strategy	<ul style="list-style-type: none"> – competence profiles – knowledge – skills – experience – aptitude – recruitment – re-assignment
3.08	quality management policy	<ul style="list-style-type: none"> – business strategy – organization quality goals and objectives – quality management standards
3.09	project management plan	<ul style="list-style-type: none"> – work breakdown structure – task relationships/external dependencies – achievement milestones – organizational infrastructure – procured items and services – project review times/events – reserves for risk management
3.10	project acquisition plan	<ul style="list-style-type: none"> – staff recruitment – acquired materials and goods – acquired services – solicitation plans – supplier selection – contract monitoring schedule
3.11	technical management plan	<ul style="list-style-type: none"> – technical achievement criteria – technical roles and responsibilities – technical resources – selected methods and tools – supporting technical services – technology readiness – technical risks – technical review schedule
3.12	service management plan	<ul style="list-style-type: none"> – agreement – service levels – service conditions – operational location – service non-conformances – maintenance – operator training

Appendix A SYSTEMS ENGINEERING CAPABILITY INDICATORS

3.13	project quality plan	<ul style="list-style-type: none"> – project quality objectives – enterprise quality goals and objectives – enterprise quality management policies and procedures – quality standards – infrastructure assets and capability – enabling systems – infrastructure capacity – service availability – allocation to tasks – project performance criteria – stakeholder satisfaction criteria – project status reporting schedule – enterprise reporting schedule
3.14	decision-making strategy	<ul style="list-style-type: none"> – decision categories – prioritisation scheme – effectiveness assessment – project initiation or progression approvals – decision-making parties
3.15	risk treatment strategy	<ul style="list-style-type: none"> – risk avoidance actions and rationale – risk mitigation actions and rationale – risk transfer actions and rationale – risk retention actions rationale
3.16	configuration management strategy	<ul style="list-style-type: none"> – levels of system integrity, security, safety – configuration baseline criteria, events and timing – release authorities – conditions of release and storage – audit conditions
3.17	information management strategy	<ul style="list-style-type: none"> – business policy – knowledge base – security requirements – privacy legislation – information audit
3.18	implementation strategy	<ul style="list-style-type: none"> – implementation procedures – fabrication processes – enabling tools and equipment – fabrication constraints – verification conditions
3.19	integration strategy	<ul style="list-style-type: none"> – assembly sequence and configurations – verification information – fault isolation and diagnosis constraints

		<ul style="list-style-type: none"> – operator integration
3.20	verification strategy	<ul style="list-style-type: none"> – product requirements – verification methods and techniques – configuration sequences – disassembly strategies – fault diagnosis steps – inspections and comparisons – static and dynamic tests – demonstrations and acceptance criteria – regression criteria – verification enabling systems requirements – non-conformance handling – reviews and audits
3.21	transition strategy	<ul style="list-style-type: none"> – installation constraints – operating environment dependencies – commissioning instructions – operator incorporation – acceptance conditions
3.22	validation strategy	<ul style="list-style-type: none"> – stakeholder requirements/acquirer agreements – safety, technical and commercial criticality constraints – validation enabling system requirements – non-conformance handling – validation limitations and deferred actions – service conformance recording – validation steps, – operational states, scenarios and missions – conformance confidence, diagnosis, discrepancies – validation methods and techniques – purpose, conditions and conformance criteria
3.23	operation strategy	<ul style="list-style-type: none"> – service availability schedule – introduction to service conditions – operator capacity and renewal – system modification schedules – release and re-acceptance criteria – service withdrawal circumstances – service migration and concurrent services – customer satisfaction criteria

Appendix A SYSTEMS ENGINEERING CAPABILITY INDICATORS

3.24	maintenance strategy	<ul style="list-style-type: none"> – operational availability requirements – corrective and preventative maintenance policy – maintenance capabilities and locations – maintenance enabling system requirements – maintenance staff competence – maintenance schedules – service restrictions and suspensions – replacement system element logistics – health, safety, security and environment legislation
3.25	disposal strategy	<ul style="list-style-type: none"> – waste product handling – system element recycling options – disposal enabling system requirements – disposal constraints, regulations and directives – disposed material handling – audits and inspections – health, safety, security and environment legislation
4.00	procedure	refer to B.1 for Generic Work Product typical characteristics
4.01	supplier selection procedure	<ul style="list-style-type: none"> – business policy – supply chain circumstances – technical and commercial issues – selection rating criteria – negotiation schedules – decision and rationale feedback
4.02	system life cycles management procedure	<ul style="list-style-type: none"> – enterprise strategic plans – business area plans – risk management policy – quality management policies – stage entry and exit criteria – project tailoring policy – key milestones definition – reporting events
4.03	quality management system	<ul style="list-style-type: none"> – quality policy – access mechanism – information medium – standards – customer satisfaction

		<ul style="list-style-type: none"> – project quality plans – quality assessments – quality improvements
4.04	implementation procedure	<ul style="list-style-type: none"> – selected implementation technology – implementation enabling systems – fabrication safety, security and environmental standards and legislation – physical fabrication and conditioning techniques – software generation and coding – operator task training – system element verification actions – conformance and quality reporting
4.05	integration procedure	<ul style="list-style-type: none"> – system elements – interface control descriptions – configuration baselines – diagnostic and regression tests – integration enabling systems
4.06	verification procedure	<ul style="list-style-type: none"> – verification methods – verification enabling system – assembly sequence – fault diagnosis sequence – non-conformance handling
4.07	transition procedure	<ul style="list-style-type: none"> – installation location preparation – transition enabling system – operator induction – activation conditions – transition non-conformances – operational readiness criteria – acceptance requirements
4.08	validation procedure	<ul style="list-style-type: none"> – agreement – acceptance conditions – service requirements – operational states, scenarios and missions – validation enabling system
4.09	operation procedure	<ul style="list-style-type: none"> – service availability requirements – service non-conformance reports – corrective action request – maintenance requirements
4.10	maintenance procedure	<ul style="list-style-type: none"> – servicing skills – preventative maintenance schedules – system element failure rates

Appendix A SYSTEMS ENGINEERING CAPABILITY INDICATORS

		<ul style="list-style-type: none"> – spares holdings – maintenance enabling systems – random faults detection – systematic fault detection – replacement schedules – level of system element replacement – useful life of degradable system elements – corrective design/implementation actions
4.11	disposal procedure	<ul style="list-style-type: none"> – service withdrawal/transfer sequences – power/fuel termination – systems interfaces – disassembly sequences – intermediate/long term storage/archive conditions – operating knowledge recording – health, safety, security and environmental legislation
5.00	record	refer to B.1 for Generic Work Product typical characteristics
5.01	supplier justification record	<ul style="list-style-type: none"> – supply chain – technical and commercial issues – competitive solicitations – selection criteria – offerings – rating
5.02	decision register	<ul style="list-style-type: none"> – problem/opportunity circumstances – decision-making authority – decision-maker identity – concerned parties – subject matter experience – subject matter knowledge – courses of action open – actions taken – problem/opportunity resolution outcome – audits
5.03	decision history record	<ul style="list-style-type: none"> – problem/opportunity circumstances – actions taken – problem/opportunity resolution outcome – learning from experience

5.04	risk register	<ul style="list-style-type: none"> – risk categories and responsibilities – risk sources and relationships – initiating events and occurrence probability – risk consequences – threshold of acceptability – risk ranking
5.05	risk history record	<ul style="list-style-type: none"> – history of risks – risk treatment actions – risk budget – risk treatment outcomes
5.06	configuration baseline	<ul style="list-style-type: none"> – composition – configuration – levels of system integrity, security, safety – release events and timing – release authorities – conditions of release and storage
5.07	configuration history record	<ul style="list-style-type: none"> – baselines – releases – authorisations – change rationale
5.08	information history record	<ul style="list-style-type: none"> – agreement constraints – organizational information policy or legislation – information currency and validity – medium longevity – data security and privacy legislation – intellectual property – information audits
5.09	stakeholder requirements traceability	<ul style="list-style-type: none"> – agreement – stakeholder class – service requirement – validation – acceptance conditions
5.10	stakeholder requirements record	<ul style="list-style-type: none"> – stakeholder requirements baselines – changes of need and origin – persistent needs – agreements

Appendix A SYSTEMS ENGINEERING CAPABILITY INDICATORS

5.11	systems requirements traceability	<ul style="list-style-type: none"> – stakeholder requirement – system functional model – architectural design – agreement
5.12	system requirements record	<ul style="list-style-type: none"> – system requirements baselines – design rationale – modelling conventions
5.13	system architectural design traceability	<ul style="list-style-type: none"> – system product requirements – architectural design – agreements – regulatory requirements – product sector requirements
5.14	system architectural design record	<ul style="list-style-type: none"> – architectural design baseline – structural and functional partitioning – interface and control definitions – design decisions – traceability to requirements
5.15	implementation record	<ul style="list-style-type: none"> – architectural design specification – system element specification – supplier agreements – legislation – organizational policy
5.16	integration record	<ul style="list-style-type: none"> – integration baselines – integration strategy – integration enabling systems – non-conformances and errors – corrective or improvement outcomes
5.17	verification record	<ul style="list-style-type: none"> – compliance deviations – verification data – authentications – corrective actions

5.18	transition record	<ul style="list-style-type: none"> – installation baseline – installation test – operator training/certification – acceptance authorisation – user training – corrective actions
5.19	validation record	<ul style="list-style-type: none"> – stakeholder requirements – legal and regulatory requirements – operator qualification – service levels – non-conformances and deviations – corrective actions
5.20	operation record	<ul style="list-style-type: none"> – operational configuration – modification information – operation non-conformances – service levels – service period
5.21	maintenance record	<ul style="list-style-type: none"> – operational information – maintenance information – product/service non-conformance – trends and failure patterns – detected design errors – services threats and failures
5.22	disposal record	<ul style="list-style-type: none"> – disposed material – storage locations and conditions – anticipated storage effects – recycled material – re-used system elements
5.23	payment record	<ul style="list-style-type: none"> – contract conditions – acquirer information – quality measures
6.00	report	refer to B.1 Generic Work Product typical characteristics
6.01	supplier assessment report	<ul style="list-style-type: none"> – supplier offerings – proposal rating justification – supplier preference – selected supplier

Appendix A SYSTEMS ENGINEERING CAPABILITY INDICATORS

6.02	delivery acceptance report	<ul style="list-style-type: none"> – delivered product/service – product/service requirements – agreement – compliance confirmation method – compliance tools
6.03	supply performance report	<ul style="list-style-type: none"> – cost – performance – schedule – risks – execution of agreement assessment – undesirable outcome evaluation – corrective action recommendations
6.04	system life cycle model review	<ul style="list-style-type: none"> – life cycle models – stages – life cycle processes – achievement criteria – life cycle model improvements
6.05	resource commitment report	<ul style="list-style-type: none"> – resource allocation – project interdependencies – project plans – project reviews – project authorisation – recruitment – training
6.06	investment decision report	<ul style="list-style-type: none"> – business strategy – new business opportunities – cost and delivery projections – viability assessment – risks to organization – projects prioritisation – current product/service portfolio – current assets
6.07	system life cycle process review	<ul style="list-style-type: none"> – enterprise goals and policies – quality audits – method and tool support – project review – tailoring policy
6.08	system life cycle improvement report	<ul style="list-style-type: none"> – system life cycle performance measures – process execution performance/trends – process improvement opportunities – processes improvement actions

6.09	training report	<ul style="list-style-type: none"> – competence profile – knowledge – skill – recruitment – re-assignment
6.10	customer satisfaction assessment	<ul style="list-style-type: none"> – customer satisfaction definition – information sources and schedule – customer satisfaction status
6.11	quality management report	<ul style="list-style-type: none"> – quality objectives – non-conformance information – project performance trends – product/service quality improvements
6.12	project status report	<ul style="list-style-type: none"> – planned cost, resources, schedule – actual cost against project plan – actual or estimated labour costs – actual or estimated material costs – actual or estimated service costs – actual measured achievement – milestone completion
6.13	project resources and services report	<ul style="list-style-type: none"> – project structure and responsibilities – team competencies – adequacy and availability supporting infrastructure – readiness of enabling systems – technology readiness – technology insertion plans
6.14	project quality report	<ul style="list-style-type: none"> – performance measures – business audit results – technical results – project self-assessment results – quality deviations or variations – corrective actions
6.15	project progress report	<ul style="list-style-type: none"> – project status – risk profile – stage progression criteria – milestone achievements – projected performance against plans – business and technical risks – replanning needs – resource authorisation – project readiness to proceed

Appendix A SYSTEMS ENGINEERING CAPABILITY INDICATORS

6.16	corrective action report	<ul style="list-style-type: none"> – non-compliance – agreement – stakeholder requirements – systems requirements – service requirements – diagnosed faults – design change
6.17	decision report	<ul style="list-style-type: none"> – problem or opportunity – assumptions – resolution status – outcomes – trends
6.18	risk management report	<ul style="list-style-type: none"> – identified risk – risk treatment actions – impact on plans – risk status – project risk reserves
6.19	configuration management report	<ul style="list-style-type: none"> – configuration item audit – baselines – release time – approvals – change history
6.20	Information management report	<ul style="list-style-type: none"> – information item audit – integrity, validity and availability status – replication requirements – transformation requirements – technology infrastructure retention
6.21	stakeholder requirements report	<ul style="list-style-type: none"> – integrity analysis – conflicting requirements – missing requirements – incomplete requirements – ambiguous requirements – unverifiable requirements – requirements integrity resolution strategy
6.22	system requirement report	<ul style="list-style-type: none"> – requirements uniqueness – requirements completeness – requirements non-ambiguity – requirements consistency – requirements implementability – requirements verifiability – requirements integrity resolution strategy – traceability

		<ul style="list-style-type: none"> – derived requirements
6.23	system architectural design report	<ul style="list-style-type: none"> – feasibility – effectiveness – design stability – enabling system constraints
6.24	implementation report	<ul style="list-style-type: none"> – suppliers – implementation technology – verification data – acceptance – handling and storage
6.25	integration report	<ul style="list-style-type: none"> – integration data – system element delivery – assembly sequence – integration faults – maintenance
6.26	verification report	<ul style="list-style-type: none"> – verification data – verification non-conformances – corrective action information – diagnosed faults – trends and patterns of failure – evidence of design or implementation errors
6.27	transition report	<ul style="list-style-type: none"> – capability readiness – installed system – operational location and environment – operator readiness – acceptance criteria – support enabling system readiness – corrective action reports
6.28	validation report	<ul style="list-style-type: none"> – validation data – user communication – stakeholder satisfaction – service restoration – service amendment actions
6.29	operation report	<ul style="list-style-type: none"> – delivered services – service levels – service satisfaction – performance parameters – acceptable/revised service parameters – operational situation – service non-compliances – operating personnel performance – operations plans

Appendix A SYSTEMS ENGINEERING CAPABILITY INDICATORS

		<ul style="list-style-type: none"> – system activation actions – operator training and accreditation – usability review – occupational safety – environmental protection
6.30	maintenance report	<ul style="list-style-type: none"> – logistics analysis – replenishment levels – replacement schedules – repair rates – spares availability, spares integrity – failure data – events and histories – lifetime data – preventive and corrective maintenance – adaptive and perfective maintenance – operator replenishment – maintenance personnel performance
6.31	disposal report	<ul style="list-style-type: none"> – disposal confirmation – health factors – safety factors – security factors – environmental factors
7.00	request	refer to B.1 Generic Work Product typical characteristics
7.01	acquisition request	<ul style="list-style-type: none"> – supply solicitation – requirements – tasking statement – acquirer processes
7.02	acquisition agreement change request	<ul style="list-style-type: none"> – needs – stakeholder requirements – agreement terms and conditions – delivery schedule – acceptance criteria
7.03	supply proposal	<ul style="list-style-type: none"> – supply request response – supplier's project plans – proposed cost, schedule and system performance – anticipated risks – supply terms and conditions
7.04	supply agreement change request	<ul style="list-style-type: none"> – acquirer requests – supplier requests

7.05	project control directive	<ul style="list-style-type: none"> – project objectives – project plans – risk response directives – personnel deployment and re-assignments – infrastructure deployment and re-assignments – corrective actions – preventive actions
7.06	supplier directive	<ul style="list-style-type: none"> – acceptance criteria – product/service defects – instructions to rectify – supplier monitoring – modified terms and conditions
7.07	supply change directive	<ul style="list-style-type: none"> – change actions – stakeholder requirements – operational constraints – delivery requirements – project plan changes
7.08	authorization to proceed request	<ul style="list-style-type: none"> – stage gate review – project review – project report
7.09	configuration baseline change request	<ul style="list-style-type: none"> – design release – corrective maintenance – technology upgrade – obsolescence – in-service modification
7.10	operation request	<ul style="list-style-type: none"> – change of need – context of use – service level – service non-compliance – maintenance schedule – modifications and upgrades – operator assignment – consumable materials
7.11	maintenance request	<ul style="list-style-type: none"> – acquired maintenance services – consumable materials – energy sources – operator provisions

Appendix A SYSTEMS ENGINEERING CAPABILITY INDICATORS

8.00	specification	refer to B.1 for Generic Work Product typical characteristics
8.01	acquisition agreement	<ul style="list-style-type: none"> – supplier – product or service need – achievement milestones – reporting milestones – completion conditions – exception handling procedures – change control procedures – acceptance criteria – delivery conditions – payment schedules – intellectual property rights – acquirer terms and conditions
8.02	delivery acceptance criteria	<ul style="list-style-type: none"> – agreement criteria – product/service compliance criteria – delivery procedures and conditions – verification procedure – validation procedure
8.03	supply agreement	<ul style="list-style-type: none"> – acquirer – product or service offered – supplier terms and conditions – acceptance information/demonstration – deviation handling – delivery procedures – intellectual property rights – responsibility transfer definition – completion conditions – liabilities
8.04	project requirements	<ul style="list-style-type: none"> – enterprise objectives – requirements/review milestones – acquirer/supplier agreement – project objectives – projects accountabilities and authorities – project achievement criteria – project life cycle stage responsibility – project completion criteria – project constraints – scope of technical activities – life cycle process tailoring – project assessment and control plans

8.05	project authorization	<ul style="list-style-type: none"> – project initiation directives – project re-direction directive – reporting to enterprise management – stage gate approval criteria – project termination criteria
8.06	system life cycle process measures	<ul style="list-style-type: none"> – business strategy – quality policy – customer satisfaction – project reviews – non-conformances
8.07	enterprise resource requirement	<ul style="list-style-type: none"> – project plans – resource allocation – project interdependencies – recruitment – training – enabling systems
8.08	quality management goals	<ul style="list-style-type: none"> – business strategy – quality policy – customer satisfaction
8.09	quality measures	<ul style="list-style-type: none"> – customer satisfaction – non-compliances – project reviews
8.10	project infrastructure and services requirements	<ul style="list-style-type: none"> – project organization – project roles and responsibilities – definition of role competence – staff appointments – designation of legally responsible roles and individuals – methods of team working
8.11	project performance measures specification	<ul style="list-style-type: none"> – defined project measures of achievement – enterprise quality metrics – schedule of measures collection – measures analysis procedure – project performance deviation indicators
8.12	stakeholder requirements	<ul style="list-style-type: none"> – system purpose – stakeholder concerns – operational scenarios – context of use – user culture and social profile – measures of effectiveness – service validation conditions – service requirements to stakeholder needs

Appendix A SYSTEMS ENGINEERING CAPABILITY INDICATORS

		traceability
8.13	stakeholder requirements constraints on solution	<ul style="list-style-type: none"> – stakeholders' assumptions and rationale – stakeholder perceived constraints – previous implementation decisions – operator staff capabilities and competence – agreements – acquiring organization constraints – health and safety – security standards – reliability and availability – supportability – environmental protection
8.14	system requirements	<ul style="list-style-type: none"> – functional boundary definition – critical performance parameters – behaviour and properties required – system stimuli and responses – interface constraints – user and environment behaviour – quality in use measures – implementation constraints
8.15	system requirements constraints on solution	<ul style="list-style-type: none"> – performance requirements – product sector standards – interface requirements – verification enabling system agreement
8.16	system verification performance measures	<ul style="list-style-type: none"> – performance parameters – critical performance measures – quality in use – non-compliance criteria – verification enabling system
8.17	system interface requirements	<ul style="list-style-type: none"> – system boundary definition – external systems – external interface specifications – human-system interfaces – product sector interface standards
8.18	human-equipment interface requirements	<ul style="list-style-type: none"> – operator competence and capabilities – usability requirements and standards – operator and user recruitment – operator training needs – operator certification requirements
8.19	system element requirements	<ul style="list-style-type: none"> – system architecture definition – system element design constraints

8.20	implementation constraints on solution	<ul style="list-style-type: none"> – implementation technology selected – implementation technology constraints and limitations – acquirer furnished materials – system elements for adaptation – implementation enabling systems limitations
8.21	implementation enabling system requirements	<ul style="list-style-type: none"> – agreement – verification results – non-conformance handling procedures – packaging and storage conditions
8.22	integration constraints on solution	<ul style="list-style-type: none"> – architectural design – interface requirements – diagnosis/rectification accessibility – integration enabling systems
8.23	integration enabling system requirements	<ul style="list-style-type: none"> – integration enabling system facilities – integration procedures – intermediate assembly configurations – assembly equipment and jigs – integration constraints and limitations
8.24	verification constraints on solution	<ul style="list-style-type: none"> – measurement methods – verification enabling systems – availability – interconnection – accuracy, uncertainty, repeatability
8.25	verification enabling system requirement	<ul style="list-style-type: none"> – verification facilities – system requirements – integration strategy – non-conformance diagnosis – acceptance criteria – interfaces – availability
8.26	transition constraints on solution	<ul style="list-style-type: none"> – operational site conditions – operating environment dependencies – commissioning instructions

Appendix A SYSTEMS ENGINEERING CAPABILITY INDICATORS

		<ul style="list-style-type: none"> – acceptance conditions – operator induction – activation conditions – acceptance requirements
8.27	transition enabling system requirements	<ul style="list-style-type: none"> – operational location – installation skills – fault identification
8.28	validation constraints of solution	<ul style="list-style-type: none"> – operational location access – operator availability – renewable materials – acceptance criteria – validation enabling system
8.29	validation enabling system requirements	<ul style="list-style-type: none"> – verification facilities – system requirements – installation strategy – non-conformance diagnosis – acceptance criteria – operators competence – interfaces
8.30	maintenance constraints on solution	<ul style="list-style-type: none"> – maintenance enabling systems – re-use strategy – logistics strategy – system element holdings – re-supply limitations – maintenance scenarios – maintenance locations – maintenance environments – maintenance staff competence – maintenance enabling systems
8.31	maintenance enabling system requirements	<ul style="list-style-type: none"> – service level agreement – operational situation – activation procedure – required instances/continuous service – service conformance audits – transfer/changeover of services
8.32	disposal constraints on solution	<ul style="list-style-type: none"> – disassembly location and access – disposal enabling systems – availability of storage locations – disposal skill levels
8.33	disposal enabling systems requirement	<ul style="list-style-type: none"> – disposal support services – disposal location – environmental legislation

APPENDIX B SYSTEMS ENGINEERING COMPETENCE FRAMEWORK

B.1 SFIA 2 Skill Components Mapped to 5 Levels

The SFIA Model Level is structured according to four attributes of responsibility and accountability as follows:

Autonomy

- Leadership
- Accountability
- Problem solving
- Originality

Complexity

- Scope
- Context
- Understanding
- Novelty

Influence

- Influence level
- Decision-making
- Relationship
- Responsibility

Business Skills

- Processes/procedures
- Standards/methods
- Tools
- Application
- Team working
- 'Outside' Communication

Originally defined according to a 9-level scale to cover all ranks of job seniority in an organisation this was then compressed by the British Computer Society for the purposes of their career development. Using the SFIA text, these are again mapped in Table B-1 to describe a five-level professional attainment scale. This provides the vertical dimension of the competence framework, as described in 6.4.3.

<p>DERA/QinetiQ Level 5</p> <p>SFIA Level 6/7</p> <p>Set Strategy, Initiate, Influence</p>	<p>Autonomy</p> <p>Has defined authority and responsibility for policy formation and the technical, financial and quality aspects of its application. Establishes organisational objectives and delegates assignments. Accountable for decisions made and actions taken by self and subordinates.</p> <p>Influence</p> <p>Influences policy on systems engineering contribution to business objectives. Influences industry and highest level of own organisation. Decisions crucial to organisational success. Advances knowledge and exploitation of systems engineering. Develops strategic relationships with customers and industry leaders.</p> <p>Complexity</p> <p>Highly complex work activities covering technical, financial and quality aspects and formulates strategy. Work involves creative application of wide range of technical and/or management principles. Has deep understanding of systems industry and emerging technologies, and implications for the wider business environment.</p> <p>Business skills</p> <p>Strategic management and leadership skills. Understands, explains and presents complex technical ideas to both technical and non-technical audiences. Has a broad understanding of all aspects of systems engineering and deep understanding of area(s) of specialisation. Understands and communicates the role and impact of systems engineering in the employing organisation. Is able to understand and communicate the potential impact of emerging technologies on organisations and individuals and can analyse the risks of using or not using such technologies. Takes initiative to keep both own and subordinates skills up to date and to maintain awareness of developments in, and application of, systems engineering.</p>
<p>DERA/QinetiQ Level 4</p> <p>SFIA Level 5</p> <p>Ensure, Advise</p>	<p>Autonomy</p> <p>Works under broad direction. Full accountability for own technical work or project/supervisory responsibilities. Receives assignments in the form of objectives. Establishes own milestones, team objectives and delegates assignments. Work is often self-initiated.</p> <p>Influence</p> <p>Influences organisation, customers, suppliers and peers within industry on contribution of specialisation. Significant responsibility for the work of others and for the allocation of resources. Decisions impact on success of assigned projects i.e. results, deadlines and budget. Develops business relationships with customers.</p> <p>Complexity</p> <p>Challenging range – variety of complex technical or professional work activities. Work requires application of fundamental principles in a wide and often unpredictable range of contexts. Understands relationship between specialism and wider customer/ organisational requirements.</p> <p>Business skills</p>

Appendix B Systems Engineering Competence Framework

	<p>Advises on the available standards, methods, tools and applications in own area of specialisation and can make correct choices from alternatives. Can analyse, diagnose, design, plan, execute and evaluate work to time, cost and quality targets. Communicates effectively, formally and informally, with colleagues, subordinates and customers. Demonstrates leadership. Clear understanding of the relationship between own area of responsibility /specialisation to the employing organisation and takes customer requirements into account when making proposals. Takes initiative to keep skills up to date. Maintains awareness of developments in the industry. Can analyse user requirements and advise users on scope and options for operational improvement. Demonstrates creativity and innovation in applying IS solutions for the benefit of the user.</p>
<p>DERA/QinetiQ Level 3</p> <p>SFIA Level 4</p> <p>Enable</p>	<p>Autonomy</p> <p>Works under general direction within a clear framework of accountability. Substantial personal responsibility and autonomy. Plans own work, to meet given objectives and processes.</p> <p>Influence</p> <p>Influences team, and specialist peers internally. Influences customers at account level and suppliers. Some responsibility for work of others and allocation of resources. Participates in external activities related to specialisation. Decisions influence success of projects and team objectives.</p> <p>Complexity</p> <p>Broad range of complex technical or professional work activities, in a variety of contexts.</p> <p>Business skills</p> <p>Selects appropriately from applicable standards, methods, tools and applications and use. Demonstrates analytical and systematic approach to problem solving. Communicates fluently orally and in writing and can present complex technical information to both technical and non-technical audiences. Is able to plan, schedule and monitor work activities in order to meet time and quality targets and in accordance with health and safety procedures. Is able to absorb rapidly new technical information and apply it effectively. Good appreciation of wider field of IS, how IS is used in relevant employment areas and how IS relates to the business activities of the employer or client. Maintains awareness of developing technologies and their application and takes some responsibility for personal development.</p>
<p>DERA/QinetiQ Level 2</p> <p>SFIA Level 3</p> <p>Apply</p>	<p>Autonomy</p> <p>Works under general supervision. Uses discretion in identifying and resolving complex problems and assignments. Specific instruction is usually given and work is reviewed at frequent milestones. Determines when problems should be escalated to a higher level.</p> <p>Influence</p> <p>Interacts with and influences department/project team members. Frequent external contact with customers and suppliers. In predictable and structured areas may supervise others. Decisions may impact work assigned to individual/phases of project.</p> <p>Complexity</p> <p>Broad range of work, sometimes complex and non routine, in variety of environments.</p> <p>Business skills</p> <p>Understands and uses appropriate methods tools and applications. Demonstrates analytical and systematic approach to problem solving. Takes initiative in identifying and negotiating appropriate development opportunities. Demonstrates effective communication skills. Contributes fully to the work of teams. Can plan, schedule and monitor own work (and that of others where applicable) competently within limited time horizons and according to health and safety procedures. Is able to absorb and apply new technical information. Is able to work to required standards and to understand and use the appropriate methods, tools and applications. Appreciates wider field of IS, how own role relates to other IS roles and</p>
<p>DERA/QinetiQ Level 1</p> <p>SFIA Level 2</p> <p>Assist</p>	<p>Autonomy</p> <p>Works under routine supervision. Uses minor discretion in resolving problems or enquiries</p> <p>Works without frequent reference to others.</p> <p>Influence</p> <p>Interacts with and may influence department. May have some external contact with customers and suppliers. May have more influence in own domain.</p> <p>Complexity</p> <p>Performs range of varied work activities in variety of structured environments.</p> <p>Business skills</p> <p>Understands and uses appropriate methods tools and applications. Demonstrates a rational and organised approach to work. Awareness of health and safety issues. Identifies and negotiates own development opportunities. Sufficient communication skills for effective dialogue with colleagues. Able to work in a team. Able to plan, schedule and monitor own work within short time horizons. Can absorb technical information when it is presented systematically and apply it effectively.</p>

Table B-1 SFIA Responsibility and Accountability Mapped to 5 Levels

Appendix B Systems Engineering Competence Framework

DERA/QinetiQ (Extended) Competence Level Descriptions According to the SFIA Skill Level Classes					
Level	Business Skills		Complexity		Autonomy and Influence
	Grasp	Scope	Profundity	Novelty	
5	Recognised, substantial body of knowledge and range of understanding	of principles, underlying systems and know-how.	Significant range of fundamental principles to	unpredictable variety of contexts, complex and without precedent.	Substantial personal autonomy, significant contributions to specialism development, strategic corporate advice.
4	Deep or wide knowledge	of basic principles, theory and practices.	Range of practices, procedures and specialist techniques, lacking formal guidance to	some unpredictable contexts.	Personal autonomy, creativity and mentorship, technical and professional advice at corporate level.
3	Good understanding	of principles, processes, techniques, equipment and applications.	Established practices, principles and specialist techniques or skills, within an accepted framework to	broad range of complex, non-routine, varied activities and contexts.	Guidance from more experienced staff, provides specialist/technical advice to colleagues.
2	Working knowledge	of concepts, processes, techniques, equipment and applications.	Working knowledge or skills, within a clear operating framework to	range of non-routine, more complex, varied contexts.	May be supervised.
1	Basic knowledge/ <i>Familiarity</i>	of technical terms, processes, techniques, equipment and applications/ <i>basic concepts, terminology and relevance.</i>	Operational knowledge or skills/ <i>Appreciation of influence</i> to	range of routine or predictable activities/ <i>range of routine or predictable activities.</i>	<i>Supervised.</i> <i>Associated</i>

Table B-2 DERA/QinetiQ (Extended) Competence Level Descriptions According to the SFIA Skill Level Classes

The text of the competence levels used by DERA and subsequently by QinetiQ is transposed verbatim in Table B-2 and structured to illustrate the practicability of instantiating the generality of the competence according to corporate frameworks used by individual organisations for their job descriptions, grading, career development and performance review. The italic text indicates clarifying additions.

Appendix B Systems Engineering Competence Framework

ISO/IEC 15288 Processes	5.1 Agreement Processes These processes define the activities necessary to establish an agreement between two organizations. If the Acquisition Process is invoked, it provides the means for conducting business with a supplier of products that are supplied for use as an operational system, of services in support of an operational system, or of elements of a system being developed by a project. If the Supply Process is invoked, it provides the means for conducting a project in which the result is a product or service that is delivered to the acquirer.					
	5.1.2 Acquisition Process The purpose of the Acquisition Process is to obtain a product or service in accordance with the acquirer's requirements.			5.1.3 Supply Process The purpose of the Supply Process is to provide an acquirer with a product or service that meets agreed requirements.		
SFIA Skills	Procurement Contract management		Account management	Marketing	Selling	Sales support
SFIA Example Tasks	The overall management and execution of the acquisition of system products and services from third-party suppliers to the business.			Achieve satisfaction for the customer and an acceptable business return for the organisation. Analyse, coordinate and stimulate prospective or existing markets for products and services. Identify sales prospects, develop customer interest, and prepare, execute and monitor supply agreements, including technical advice and assistance, to customer organisations. Monitor sales and delivery obligations, and assist the customer to ensure that it gains maximum benefit from the products and services supplied and available.		
SFIA Level 6 (7) Set Strategy, Initiate, Influence	Establishes acquisition strategy, policy, standards, methods and processes to bring added value to the organisation. Influences policy on the approval of suppliers, conduct of tendering process and purchasing procedures. Manages contracts with suppliers including identifying opportunities for business improvement. Ensures implementation of procurement strategies and evaluation criteria in line with procurement legislation. Leads the acquisition of system products and services, from clarifying requirements to confirming acceptance of delivered system. Agrees indicators and targets, and meets budgets for the acquisition. Develops mutual understanding and relationships with suppliers, acting as the senior arbiter to ensure problems are resolved. Leads major meetings with suppliers, enforcing or renegotiating contracts as required.			Oversees the organisation's customer facing activities to ensure they are aligned with corporate objectives and policy. Manages dealings with customer; initiates procedures to improve service to and relationships with customers. Builds relationships with key senior staff in the customer organisation in order to increase business opportunities. Advises on the selection of systems and technology to meet customer business objectives. Oversees the strategic decisions management and planning of marketing and business opportunities to meet its business objectives, including strategies market segmentation customer retention. Approves proposals and initiates the implementation of development activity in customer services and systems.		
SFIA Level 5 Ensure, Advise	Implements, maintains and disseminates procurement strategy, policy, standards, methods and processes. Investigates the technical and commercial options for fulfilling the acquisition needs, including possible sources of supply. Ensures that suppliers are approved in accordance with company procedures. Manages tender, agrees the preferred options and potential suppliers with the business, negotiates with preferred suppliers. Develops and manages agreements for product and services with suppliers to meet key performance indicators and agreed targets. Develops, establishes and places contracts, including technical schedules and acceptance procedures and criteria. Maintains mutual understanding between the organisation and its suppliers. Ensures contracts are monitored, evaluates the performance of suppliers, reviews and acts to correct the performance of suppliers.			Develops, implements and oversees the organisation's marketing and sales activities to ensure they align with corporate objectives and meet specified objectives. Manages and monitors market research, analysis and planning. Initiates the implementation of development/change in products and services. Approves sales plans targets and proposals, and controls and develops sales activities. Ensures that reliable cost, effort and risk estimates and project plans are produced and takes full responsibility for the technical content of bids and sales proposals. Negotiates with customer representatives at the most senior level on prospects, technical and commercial issues, taking responsibility for ensuring customer assistance and advice. Maintains customer contact during the supply process and after the conclusion of agreements, to ensure customer satisfaction. Establishes metrics to provide data on performance and help with the continuous improvement of customer service and sales support activities.		
SFIA Level 4 Enable	Monitors contracts to ensure acquired products and services requirements are met and reports on the performance of supplier. Investigates and acts to correct problems in contract delivery. Liaises between supplier and users and arbitrates as necessary.			Plans and conducts market research, investigates and analyses customer dynamics to inform marketing plans, including market segmentation and customer loyalty. Collects and uses market information in order to meet the organisation's sales objectives. Responds to existing sales opportunities and develops new opportunities. Helps customers clarify their needs and requirements, devises solutions and assesses their feasibility and practicality. Produces estimates of cost and risk and initial project plans to inform sales proposals. Maintains effective customer relationships, both during and after the conclusion of sales agreements, to ensure customer satisfaction. Monitors and reports on performance, customer satisfaction, market intelligence and competitors.		
SFIA Level 3 Apply	Liaises day-to-day with supplier. Collects information about contract performance and reports to management.			Organises and conducts market research and events. Investigates and analyses customer dynamics, including customer loyalty, and uses research to inform marketing plans. marketing and drafts marketing support materials Provides customer service, including technical advice and guidance, such as features, operational requirements, integration, portability of systems and information. Helps customers clarify their requirements and documents the conclusions reached. Progresses delivery of products and services. Demonstrates, installs, commissions and upgrades. Provided in-depth support and diagnosis of complex products and services, both to customers and to those delivering solutions.		
SFIA Level 2 Assist				Works with technical and non-technical customer representatives to identify needs and sales opportunities. Selects from and uses marketing tools appropriate to a project. Conducts market research and maintains marketing information. Contributes to marketing plans. Communicates effectively with customers. Assists in devising solutions to customer requirements and solves straight-forward problems. Assists in the provision of customer service, including technical advice and guidance, such as features, operational requirements, integration, portability and updates. Demonstrates, installs and upgrades products and services.		

5.2 Enterprise Processes

The Enterprise Processes manage the organization's capability to acquire and supply products or services through the initiation, support and control of projects. They provide resources and infrastructure necessary to support projects and ensure the satisfaction of organizational objectives and established agreements. They are not intended to be a comprehensive set of business processes that enable strategic management of the organization's business.

Enterprise Environment Management Process	5.2.3 Investment Management Process	6 System Life Cycle Stages
The purpose of the Enterprise Environment Management Process is to define and maintain the policies and procedures needed for the organization's business with respect to the scope of this International Standard.	The purpose of the Investment Management Process is to initiate and sustain sufficient and suitable projects in order to meet the objectives of the organization. This process commits the investment of adequate organization funding and resources, and sanctions the authorities needed to establish selected projects. It performs continued qualification of projects to confirm they justify, or can be redirected to justify, continued investment.	Establish a life cycle model that is comprised of life cycle stages. Define the purpose and outcomes for each stage. This provides a framework for the detailed modelling of system life cycles using the system life cycle processes.
IS strategy & planning	Programme management	Emerging technology monitoring
The development or review of a system-based product and service strategy to support an organisation's business goals, and the development of plans to drive forward and manage that strategy. Working with others to embed the strategic management of a system strategy as part of the management of the organisation	The selection and planning of a programme of projects and related activities to achieve a set of business objectives; the management of the programme within a controlled environment such that it maximises the associated business benefits. The identification of new and emerging system and implementation technologies, methods and techniques and the on-going assessment of their relevance and potential value to the organisation. The maintenance of emerging technology awareness among staff and business management.	
Leads the creation or review of a system-based strategy which meets the requirements of the business. Develops plans to drive forward the systems strategy. Develops processes which ensure that the strategic management of systems is embedded in the management of the organisation.	Ensures that a programme of system projects is managed to maximise business benefits. Sets the business objectives and authorises the selection and planning of all systems projects and activities. Plans, directs, and co-ordinates activities to manage and implement complex interrelated projects from contract/proposal initiation to operational stage. Leads the programme in determining customer requirements and translating these and organisational requirements into operational plans. Determines, monitors, and reviews all programme economics to include programme costs, operational budgets, staffing requirements, programme resources, and programme risk.	
	Monitors market to gain knowledge and understanding of currently emerging systems. Evaluates new and emerging technical developments, technologies, techniques and methods based on own area of expertise. Assess their likely relevance and potential value to the organisation and ensures programme management and staff are informed of this. Monitors and evaluates changes to programme management practices and initiates improvement to organisation practices.	

System Life Cycle Processes Management Process			
5.2.5 Resource Management Process			
5.2.6 Quality Management Process			
The purpose of the System Life Cycle Processes Management Process is to assure that effective system life cycle processes are available for use by the organization. This process provides system life cycle processes that are consistent with the organization's goals and policies, that are defined, adapted and maintained in a consistent way in order to meet the nature of individual projects, and that are capable of being applied using effective, proven methods and tools.			
The purpose of the Resource Management Process is to provide resources to projects. This process provides resources, materials and services to projects to support organization and project objectives throughout the life cycle. This includes a supply of educated, skilled and experienced personnel qualified to perform life cycle processes. This process assures that there is effective co-ordination and sharing of resources, information and technologies.			
The purpose of the Quality Management Process is to assure that products, services and implementations of life cycle processes meet enterprise quality goals and achieve customer satisfaction.			
Business Process Improvement	Business continuity planning	Quality management	Quality assurance
The identification of new and alternative approaches to performing business activities that are made possible by the availability of systems-based technology. The analysis of business processes, appreciating the potential application of the system life cycle to the processes, assessment of the costs and potential benefits of the new approaches considered and, where appropriate, management of change, and assistance with implementation.			
Overall management of the organisation's resources and assets, including intellectual asset, capitalising on potential opportunities and management of change. Matching of overall capability and capacity to meet current and predicted needs for the organisation's services, aiming to minimise operating costs, improve investment decisions and optimise the total cost of ownership. Identification, assessment and management of the availability, integrity and confidentiality risks of resources which support critical business services.			
Management of, or provision of advice on, the application of appropriate quality and/or environmental management and process improvement techniques to any organisational function. Ensure that the agreed quality standards within an organisation are adhered to and that best practice is promulgated throughout the organisation. Achievement and maintenance of compliance against, national and international standards, if appropriate. Independent assessment of the quality of projects, organisational functions, process, deliverable, product or service against agreed standards, best practice, or specified requirements.			
Analyses business processes; identifies alternative solutions, assesses feasibility, and recommends new approaches, where the changes include major system life cycle factors. Evaluates the financial, cultural, technological, organisational and environmental factors which must be addressed in the change programme. Establishes requirements for the implementation of significant changes in organisational mission, business functions and process, organisational roles and responsibilities, and scope or nature of product and service delivery.			
Establish organisational objectives for the coordination of corporate assets and services in accordance with organisational strategy. Provides the required performance and adequate capacity of corporate resources. Develop common approaches and organisational standards governing the acquisition, use and maintenance of resources in line with corporate objectives, and agree priority areas for investment. Plan business continuity and contingencies to address exposures and maintain agreed levels of continuity. Set criteria for the audits of resources and evaluate their effectiveness. Inspire partnerships and collaborative arrangements with other organisations in order to maximise exploitation of resources. Set the direction and lead the introduction, promotion and appropriate use of techniques, methodologies and tools throughout the organisation. Provide advice, assistance, and leadership in area of corporate services and assets. Manage the provision of specialist knowledge across the business, providing knowledge and advice in individual areas of expertise. Provide staff development and training services in line with strategic, business a			
Sets quality strategy and ensures that adequate technology, procedures and resources are in place to support it. Measures and reviews achievement of the quality policy in terms of meeting the organisation's needs and objectives. Develops organisational commitment to ongoing quality and environmental improvement. Plans, resources (either directly or indirectly) and monitors the internal quality audit schedule. Prioritises and initiates areas for improvement by changing approaches and working practices. Defines and reviews quality and environmental systems, typically using recognised models. Achieves and maintains compliance against national and international standards if appropriate			
Analyses system life cycle processes; identifies alternative solutions, assesses feasibility, and recommends new approaches. Contributes to evaluating the factors which must be addressed in the change programme. Helps establish requirements for the implementation of changes in the business process.			
Manages service delivery function, in order to achieve business goals, performance targets and agreed service levels. Examines issues pertaining to service capacity and plans, and recommends changes to ensure adequate. Manages assets held by the organisation to maximise the availability and reuse whilst minimising costs. Owns the business continuity planning process and leads the implementation of resulting plans. Coordinates systems which support business, safety and security critical processes, and assesses exposure and risks; identifies priority areas for improvement. Provides expertise and support on use of methods and tools, and knowledge and advice in own areas of expertise. Determine training and development needs in line with business requirements and develops and provides training service for the organisation. Ensures that professional standards are maintained.			
Advises on the application of appropriate quality and/or environmental management techniques. Ensures that independent appraisals follow agreed procedure and advises on audit process. Evaluates and independently appraises the internal control of processes based on investigation evidence and assessments undertaken by self or team. Facilitates improvements to processes by changing approaches and working practices, typically using recognised models. Undertakes communications and training activities to update colleagues on implication of revisions to quality standards.			
Monitors demand of organisation assets, maintains capacity to ensure adequate service level. Controls and monitors access to assets, and records proper usage. Drafts and maintains procedures and checklists for use of organisation assets. Manages related training service, specifying content and structure of training to deliver agreed outcomes. Provides input to business continuity planning process and implements resulting plans.			
Uses standards to review past performance and plan future activities, identifies opportunities for maintaining and updating quality standards in the light of emerging best practice, monitors and reports on the outputs from the quality assurance and audit processes, include safety assessments of the design, testing and validation and verification methods used. Advises on the development, maintenance, control and distribution of quality and environmental standards and ensures that this supports organisational objectives.			
Monitors access to organisational assets and records proper usage. Delivers existing training materials on the use of resources and assets			
Investigates and documents the internal control of specified aspects of mechanism and process, and assesses how this compares to the relevant standard. Takes responsibility for the control, distribution and filing of quality standards. Creates procedures and standards, according to documented procedures. Plans a programme to audit processes, products or services, and analyses and formally reports audit evidence. Identifies changes required to quality and/or environmental standards as a result of an audit or changes to current practice and takes responsibility for ensuring that they are made, either directly or indirectly.			
Uses appropriate methods tools and applications in the development, maintenance, control and distribution of quality and environmental standards. Distributes new and revised standards according to documented procedures. Makes technical changes to quality and environmental standards, according to documented procedures. Collects and collates evidence as part of a formally conducted and planned audits of processes, products or services. Examines records as part of specified testing strategies for evidence of compliance with management directives, or the identification of abnormal occurrences.			

5.3 Project Processes			
<p>The Project Processes are used to establish and evolve project plans, to assess actual achievement and progress against the plans and to control execution of the project through to fulfilment. Individual Project Processes may be invoked at any time in the life cycle and at any level in a hierarchy of projects, as required by project plans or unforeseen events. The Project Processes are applied with a level of rigour and formality that depends on the risk and complexity of the project.</p>			
5.3.2 Project Planning Process		5.3.3 Project Assessment Process	5.3.4 Project Control Process
<p>The purpose of the Project Planning Process is to produce and communicate effective and workable project plans. This process determines the scope of the project management and technical activities, identifies process outputs, project tasks and deliverables, establishes schedules for project task conduct, including achievement criteria, and required resources to accomplish project tasks.</p>		<p>The purpose of the Project Assessment Process is to determine the status of the project. This process evaluates, periodically and at major events, the progress and achievements against requirements, plans and overall business objectives. Information is communicated for management action when significant variances are detected.</p>	<p>The purpose of the Project Control Process is to direct project plan execution and ensure that the project performs according to plans and schedules, within projected budgets and it satisfies technical objectives. This process includes redirecting the project activities, as appropriate, to correct identified deviations and variations from other project management or technical processes. Redirection may include replanning as appropriate.</p>
Project management	Project office	Systems development management	Technical authority
<p>Plan projects to meet identified business needs. Estimate and plan resources in order to carry out programmes of time, budget and quality targets, and in accordance with appropriate standards. Develop, produce and maintain time, resource, cost and exception plans. Acquire the necessary resources and skills, within agreed parameters of cost, timescales, and quality. Provide of project management processes, procedures, tools and techniques to programme and project managers and their teams, and establish their use.</p>		<p>Assess projects to meet identified business needs. Track and report progress and performance of projects (including those performed by third parties under contract). The analysis of performance and the maintenance of metric data and estimating models. Assess the utilisation of the necessary resources and skills, within agreed parameters of cost, timescales, and quality. project assurance teams and quality review meetings. Maintain programme and/or project documentation.</p>	<p>Control projects (including those performed by third parties under contract) to meet identified business needs. Service project control boards. Administer project change control, including use of configuration management systems. Provide direction and guidance on all technical aspects of the development of, and modifications to, systems to ensure that they take account of relevant technical strategies, policies, standards and practices and that they are compatible with existing and planned assets and infrastructure. Provide support and guidance on project management processes, procedures, tools and techniques to programme and project managers and their teams.</p>
<p>Sets organisational strategy governing the direction and conduct of project management. Authorises the management of large scale projects. Influences the preparation and maintenance of realistic project, quality and risk plans. Leads project planning and scheduling activities for strategic and corporate projects. Develops, recommends and promotes the application of sound project management technologies. Authorises the management of small-scale projects. Sets strategy for resource management, authorise allocation of resources for projects.</p>		<p>Sets organisational strategy governing the direction and conduct of project management. Authorises the management of large scale projects. Leads project reporting activities for strategic, high impact, high risk projects. Leads project monitoring, and reporting activities for corporate projects. Maintains overview of contribution of programme to organisational success.</p>	<p>Sets organisational strategy governing the direction and conduct of project management. Authorises the management of large scale projects. Leads project controlling activities for strategic, high impact, high risk projects. Manages risk and sees that solutions to problems are implemented in line with change control processes. Maintain overview of contribution of programme to organisational success. Manage resources necessary for all stages (planning, estimation, execution) of individual projects to ensure technical financial and quality targets are met. Responsible for specifying and implementing technical standards in tools, methods and processes in projects.</p>
<p>Identifies resources necessary for all stages (planning, estimation, execution) of individual projects to ensure technical financial and quality targets are met. Responsible for advising, specifying and implementing standards, processes, procedures, methods, tools and techniques. Contributes to project and programme management to ensure conformance to standards.</p>		<p>Contributes to reviews and audits of project and programme management to ensure conformance to standards.</p>	<p>Identifies all stages (planning, estimation, execution) of individual projects to ensure technical financial and quality targets are met. Evaluates project performance and recommends changes where necessary</p>
<p>Uses and recommends project control solutions for planning and scheduling. Sets up and provides detailed guidance on project management procedures, processes, tools and techniques. Provides basic guidance on individual project proposals.</p>		<p>Uses and recommends project control solutions for tracking projects. Attends project assurance teams and quality review meetings.</p>	<p>Uses and recommends project control solutions. Attends project control boards. Provides detailed guidance on project management procedures, processes, tools and techniques.</p>
<p>Uses recommended project control solutions for planning and scheduling. Sets up project documentation. Provides guidance on project management procedures, processes, tools and techniques.</p>		<p>Uses recommended project control solutions for tracking projects. Compiles and distributes reports. Provides administrative services to project assurance teams and quality review meetings.</p>	<p>Uses recommended project control solutions for projects. Provides administrative services to project control boards.</p>
<p>Assists with the compilation of project management reports. maintains programme and project files from supplied actual and forecast data.</p>		<p>Assists with the compilation of project management reports. maintains programme and project files from supplied actual and forecast data.</p>	

5.3.5 Decision-making Process The purpose of the Decision-making Process is to select the most beneficial course of project action where alternatives exist. This process responds to a request for a decision encountered during the system life cycle, whatever its nature or source, in order to reach specified, desirable or optimized outcomes. Alternative actions are analyzed and a course of action selected and directed. Decisions and their rationale are recorded to support future decision-making.	5.3.6 Risk Management Process The purpose of the Risk Management Process is to reduce the effects of uncertain events that may result in changes to quality, cost, schedule or technical characteristics. This process identifies, assesses, treats and monitors risks during the entire life cycle, responding to each risk in terms of appropriate treatment or acceptance.
	Business risk management
	The planning and implementation of organisation-wide processes and procedures for the management of risk in all system-based activities and projects.
	Plans and manages the implementation of organisation-wide processes and procedures, tools and techniques for the management of risk in all system-based activities and projects.
	Implements and operates organisation-wide processes and procedures, tools and techniques for the management of risk in all system-based activities and projects.

5.3.7 Configuration Management Process	5.3.8 Information Management Process
The purpose of the Configuration Management Process is to establish and maintain the integrity of all identified outputs of a project or process and make them available to concerned parties.	The purpose of the Information Management Process is to provide relevant, timely, complete, valid and, if required, confidential information to designated parties during and, as appropriate, after the system life cycle. This process generates, collects, transforms, retains, retrieves, disseminates and disposes of information. It manages designated information, including technical, project, enterprise, agreement and user information.
Configuration management	Information Resource Management
The systematic management of documentation, system products, services and assets in terms of their identification as configuration items (CIs), with the definition of their structures and relationships. Storage, access, problem reporting and change control of CIs and the application of status accounting and auditing, often in line with acknowledged external criteria such as ISO 9000, throughout all stages of the system life cycle.	The overall management and exploitation of information resources including the development and taking forward of strategy and policies for information resources management. Ensuring that the information needs of the business will be met.
Investigates, recommends and is responsible for tools, techniques and processes for ensuring appropriate versions of a product's or service's or organisation's assets or components are available and managed, as well as managing those assets.	Establishes and manages information resource management strategy and policies for the business. Plans and implements processes to take forward the strategy and policies.
Maintains tools and techniques for ensuring appropriate versions of a product's or service's or organisation's assets or components are available and managed as well as managing those assets.	Contributes to the development of information resource management strategy and policies for the business. Plans and implements processes to take forward the strategy and policies. Co-ordinates strategy for information resource management with strategies for data storage and management.
Ensure appropriate versions of a product's or service's or organisation's assets or components are available and managed while protecting those assets and components from unauthorised change, diversion, and inappropriate use.	
Uses configuration management tools to identify, track, and log system elements and their changes to maintain a record of the status of hardware and changes to the system.	

5.4 Technical Processes			
<p>The Technical Processes are used to define the requirements for a system, to transform the requirements into an effective product, to permit consistent reproduction of the product where necessary to use the product to provide the required services, to sustain the provision of those services and to dispose of the product when it is retired from service. The Technical Processes define the activities that enable enterprise and project functions to optimize the benefits and reduce the risks that arise from technical decisions and actions. These activities enable products and services to possess the timeliness and availability, the cost effectiveness, and the functionality, reliability, maintainability, producibility, usability and other qualities required by acquiring and supplying organizations. They also enable products and services to conform to the expectations or legislated requirements of society, including health, safety, security and environmental factors.</p>			
Stakeholder Process	Requirements Definition	5.4.3 Requirements Analysis Process	5.4.4 Architectural Design Process
The purpose of the Stakeholder Requirements Definition Process is to define the requirements for a system that can provide the services needed by users and other stakeholders in a defined environment. It identifies stakeholders, or stakeholder classes, involved with the system throughout its life cycle, and their needs and desires. It analyzes and transforms these into a common set of stakeholder requirements that express the intended interaction the system will have with its operational environment and that are the reference against which each resulting operational service is		The purpose of the Requirements Analysis Process is to transform the stakeholder, requirement-driven view of desired services into a technical view of a required product that could deliver those services. This process builds a representation of a future system that will meet stakeholder requirements and that, as far as constraints permit, does not imply any specific implementation. It results in measurable system requirements that specify, from the developer's perspective, what characteristics it is to possess and with what magnitude in order to satisfy stakeholder requirements.	The purpose of the Architectural Design Process is to synthesize a solution that satisfies system requirements. This process encapsulates and defines areas of solution expressed as a set of separate problems of manageable, conceptual and, ultimately, realizable proportions. It identifies and explores one or more implementation strategies at a level of detail consistent with the system's technical and commercial requirements and risks. From this, an architectural design solution is defined in terms of the requirements for the set of system elements from which the system is configured. The specified requirements
Business analysis		Systems design	Systems architecture
The methodical investigation, analysis, review and documentation of all or part of the stakeholder needs for systems in terms of functions and services, and the resources they use. The definition of requirements for improving any aspect of existing, defined, long-lived needs. The creation of viable specifications in preparation for the construction of systems to meet stakeholder needs.		Design and specification of solutions and systems architecture to meet defined needs. Provide specialist expertise and practical assistance in the investigation, evaluation and interpretation of system properties in order to ensure its coherence, availability and integrity to meet systems requirements.	The specification of systems architecture, identifying the system elements and their interrelationships needed to meet the present and future needs of all stakeholders.
Initiates and influences needs analysis. Proposes and champions potential system solutions to emerging opportunities.		Controls system design within an enterprise or industry architecture. Influences industry-based models for the development of new systems. Develops effective implementation strategies and procurement. Overall responsibility for managing and coordinating the architecture of systems.	The specification of systems architecture, identifying the system elements and their interrelationships needed to meet the present and future needs of all stakeholders.
Acts as consultant to team and customers on stakeholder needs and potential system solutions		Ensures that the system architecture balances functional, service quality and operating requirements. Establishes policy for selection of architecture components. Reviews others' system design to ensure selection of appropriate technology, efficient use of resources, and integration of multiple systems and technology. Sets standards for analysis tools and techniques, advises on their application, and ensures compliance.	
Creates stakeholder requirements specification and the rationale for development of system solution by investigating stakeholder needs.		Recommends/designs structures and elements for systems which meet business needs. Delivers 'technical visualisation' of proposed system for approval by customer and execution by system developers. Maps work to user specification and removes errors and deviations from specification to achieve effective systems. Works with customer and/or system design team to refine systems requirements.	
Find out whether and how an existing system supports current needs and recommend incremental improvements to the system.		Specifies requirements for and designs systems or subordinate systems to meet defined objectives following established process. Establishes the requirements for systems which meet defined service requirements.	
		Undertakes complete design of simple systems using architectural precedence and existing components. Assists as part of a team on design of components of larger systems. Assists in establishing the requirements for subordinate systems which meet defined system requirements.	

ary,
tivities that

5.4.5 Implementation Process					5.4.6 Integration Process	5.4.7 Verification Process
The purpose of the Implementation Process is to produce a specified system element. This process transforms specified behaviour, interfaces and implementation constraints into fabrication actions that create a system element according to the practices of the selected implementation technology. The system element is constructed or adapted by processing the materials and/or information appropriate to the selected implementation technology and by employing appropriate technical specialisms or disciplines. This process results in a system element that satisfies architectural design requirements through verification and stakeholder requirements through validation.					The purpose of the Integration Process is to assemble a system that is consistent with the architectural design. This process combines system elements to form complete or partial system configurations in order to create a product specified in the system requirements.	The purpose of the Integration Process is to assemble a system that is consistent with the architectural design. This process combines system elements to form complete or partial system configurations in order to create a product specified in the system requirements.
Technical	Database	Programming/s	Systems	Media	Systems integration	Systems testing
specialism	design	development	ergonomics	creation		
The design, creation, testing and documenting of new and amended system elements from supplied specifications in accordance with agreed standards. The specification, design and maintenance of implementation technology to support solutions to needs. The planning, design and creation of technology to delivered system services, including managing the quality assurance. The application of evaluation skills to the assessment of the ergonomics, usability and quality in use of products or services. Synthesis of test tasks to be performed (from statement of user needs and interface specifications); the design, evaluation and analysis of performance, and inputting results to the development team. The management of, and provision of expert advice on a specific technical specialisms.					Incremental and logical integration and testing of system element and their interfaces in order to form complete systems.	The planning, design, management, execution and reporting of tests, using appropriate testing tools and techniques and conforming to agreed standards, to ensure that new and amended systems, together with any interfaces, perform as specified.
Provides organisational leadership and guidelines for provision of high level technical knowledge. Initiates multi-system or multi-enterprise system element architecture, creating and enhancing industry solutions. Maintains an in-depth knowledge of specific technology specialisms, including techniques, methods, products or application areas. Responsible for organisational commitment to high standards in human factors. Specifies ergonomics standards and methods to meet organisational objectives. Can supervise specialist technical consultancy. Assists management and exploitation of the organisation's technology assets. Initiates the research and development on implementation technology. Delivers effective information to the organisation or customer.						Sets objectives and standards for systems testing function in the enterprise. Manages application and ensures function focuses on delivering business advantage.
Takes technical responsibility for all stages in the implementation process. Manages or advises systems development teams, including human factors and information. Reviews designs to ensure quality of the implementation of system elements. Maintains knowledge and provides detailed advice of specific technical specialisms, implementation technologies, techniques, methods, products or application areas. Sets standards for implementation techniques and tools, advises on their application and ensures compliance.					Designs and builds intermediate build components and interfaces. May contribute to the overall design of the solution and define the technical criteria for product/ component selection. Leads practical implementation work under the technical direction of the system designer/architect. Contributes to decisions about tools, methods and approaches to be used in the project.	Sets standards and techniques for test environment, advises on their application and ensures compliance.
Designs, implements, tests, and documents system elements. Takes technical responsibility for several activities in the implementation process. Transforms design models to appropriate physical implementations. May understand and use more than one implementation technology. Enables implementation by analysing system interactions and recommending improvements to system element interfaces, including human interfaces. Controls the creation of system/user information to meet customer requirements and organisational standards. Works with systems designers to create and maintain standards for implementation technologies and assists in setting design standards.					Controls the integration of system elements in new systems, upgrades, enhancements and conversions.	Scopes and creates test plans and test data, mapping back to requirements. Maintains the integrity of the test environment.
Transforms systems element models to appropriate physical designs. Assists in design. Fabricates, tests, and documents simple system elements to a clear design specification, including software and user behaviour. and provides input into project plans. Creates and maintains system/user documentation to meet customer requirements and organisational standards.					Integrates system elements in new systems, upgrades, enhancements and conversions.	Creates test plans and test data, mapping back to requirements. Executes test plans, recording and reporting outcomes.
Assists in the implementation of system elements. Assists in detailed design, fabrication, testing, fault resolution and documentation of simple system elements by following guidelines.					Assists in integrating system elements in new systems, upgrades, enhancements and conversions.	Assists in the execution of test plans, recording and reporting outcomes.

5.4.8 Transition Process	5.4.9 Validation Process	5.4.10 Operation Process			
The purpose of the Transition Process is to establish a capability to provide services specified by stakeholder requirements in the operational environment. This process installs a verified system, together with relevant enabling systems, e.g. operating system, support system, operator training system, user training system, as defined in agreements.	The purpose of the Validation Process is to provide objective evidence that the services provided by a system when in use comply with stakeholders' requirements. This process performs a comparative assessment and confirms that the stakeholders' requirements are correctly defined. Where variances are identified, these are recorded and guide corrective actions. System validation is ratified by stakeholders.	The purpose of the Operation Process is to use the system in order to deliver its services. This process assigns personnel to operate the system, and monitors the services and operator-system performance. In order to sustain services it identifies and analyzes operational problems in relation to agreements, stakeholder requirements and organizational constraints.			
Systems installation/ decommissioning	Service delivery management	Network control	ICT operations	Service level control	Application and system support
The installation of systems and system elements, following plans and instructions and in accordance with agreed standards. Resolving malfunctions found and recording the results. The reporting of details of system elements installed so that the organisation's configuration management records can be updated.	The planning and allocation of resources required to enable effective delivery of service to users/customers/dients, while responding to changing business and user requirements. Will typically include responsibility for ensuring that effective relationships exist and are monitored, measured and reported upon with both internal and external users/customers/clients and vendors/suppliers and that appropriate resource control is exercised.	The administration and operation of the system to deliver an agreed level of service. Negotiate and execute service level agreements, to provide the agreed levels of service. Conduct day-to-day operation and control of systems, including routine integrity and sustainment actions. Plan, schedule and measure operational facilities, monitor availability and performance, handle incidents, and produce performance statistics, diagnostic information, and operating environment reports. Use tools, as required, to capture, analyse, store and report accurate service quality details. Maintain operating plans and schedules. Investigate and resolve service problems recommending advice or training about system application, devising work-arounds, fault correction, modifications, enhancements and documentation updates. Defines maintenance and support			
	Maintain overview of contribution of service delivery to customer success and manage user expectations. Authorise allocation of resources for service delivery monitoring arrangements. Set strategy which inspires a commitment to excellence in service delivery. Develop relationships with customers at the highest level to identify potential areas of mutual commercial interest for future development. Provide leadership on the identification of future trends (e.g. technical, market, industrial, socioeconomic, legislative, etc.).	Initiate the development of resources to monitor, measure and report on service delivery arrangements. Identify any changes to business and user service requirements through regular contact with customers, share information on emerging trends (e.g. new technologies, changes in the structure of the market etc.) with customers and identify areas for improvement. Plans upgrades and improvements to system services.			
		Supports and optimises capability. Ensures management of operations and operational resources. Plans operational and support schedule to meet business demands. Ensures operational performance is maintained at agreed service levels over time. Owns, develops and manages monitoring systems which meet real customer and organisational needs. Provides operational input on planning of installation and upgrade work. Manages system enhancements to improve business performance. Ensures all requests for support and/or system changes are dealt with according to set standards and procedures.			
Plans and controls the installation of a system.		Provides technical expertise to operations management and staff. Contributes to the planning of operational and support schedules. Enables the deployment of operational resources in order to meet service levels and evaluates results. Contributes to the planning of installation and upgrade work. Analyses service level information to help plan future requirements. Identify operational resources to meet service levels over time. Enhances systems to improve business performance. Maintains support process and checks all requests for support are dealt with according to set standards and procedures.			
Installs system elements		Executes operational tasks to meet schedules and targets. Provides support to the team. Monitors, gathers and reports service level information to ensure compliance with service level agreements. Identifies and resolves problems with systems to maintain underlying business processes and/or continuity of service.			
Assists in installation of system elements.		Carries out routine operation of systems. Monitors the performance of systems. Assists in investigation and helps resolve problems relating to systems. Assists with specified support procedures.			

5.4.11 Maintenance Process				
The purpose of the Maintenance Process is to sustain the capability of the system to provide a service. This process monitors the system's capability to deliver services, records problems for analysis, takes corrective, adaptive, perfective and preventive actions and confirms restored capability.				
Network planning	Change control	Network administration & support	Database administration	User support
Provide of support to enable users to make effective use of systems. Provide day-to-day support, including resolution of user problems. Receive and resolve problems involving systems. Keep records of systems supported, users, problems and resolutions. Manage all changes from request for change through to implementation and review, to support continued availability, effectiveness and safety. Create and maintain system configuration plans. Install, configure, monitor and maintain upgrades. Participate in the creation of service level agreements, and plan infrastructure necessary to ensure provision of services to meet agreements.				
Creates and maintains overall support plans according to the organisation's business strategy. Sets the direction for the assessment, analysis, development, documentation and implementation of changes based on Requests For Change. Contributes to service level agreements with customers and plans all aspects of the infrastructure necessary to ensure provision of services to meet such agreements.				
Develops implementation plans for dealing with complex requests for change, evaluates risks to integrity inherent in implementing proposed changes, seeks authority for those activities, undertakes review of effectiveness of change implementation, suggests improvement to organisational procedures governing change management. Leads the assessment, analysis, development, documentation and implementation of changes based on Requests For Change. Creates and maintains support plans for own area of responsibility, contributes to setting service level agreements, and plans the infrastructure necessary to provide the services to meet such agreements.				
Assesses, analyses, develops, documents and implements changes based on Requests For Change. Configures and administers system configuration. Resolves problems to enabling agreed levels of support to be met. Evaluates and advises on operating techniques and tools.				
Receive escalated problems, gather further information and resolve or channel to appropriate support function. Resolves complex problems affecting use of systems to maintain underlying business targets. Develops, documents and implements changes based on Requests For Change. Applies change control procedures. Day-to-day administration and support of configuration.				
Receive and record problems from users, gather prescribed information and resolve or escalate according to given procedures. Investigate and help resolve problems relating to routine use of systems and related equipment. Assist in management of system support activities. Investigate system problems and assist with identification and implementation of remedial solutions.				
5.4.12 Disposal Process				
The purpose of the Disposal Process is to end the existence of a system entity. This process deactivates, disassembles and removes the system and any waste products, consigning them to a final condition and returning the environment to its original or an acceptable condition. This process destroys, stores or reclaims system entities and waste products in an environmentally sound manner, in accordance with legislation, agreements, organizational constraints and stakeholder requirements. Where required, it maintains records in order that the health of operators and users, and the safety of the				
As per Systems installation/decommissioning				
The decommissioning of systems, following plans and instructions and in accordance with agreed standards. The recording and reporting of system element details so that the configuration management records can be updated.				
Plans and controls the decommissioning of a system.				
Removes system elements.				
Assists in removal of system elements.				