Proceedings of the IEEE International Symposium on Systems Engineering (ISSE), Rome, 28-30 September 2015
DOI:10.1109/SysEng.2015.7302795

A Problem Solving Method Using Context Types

Tim Mackley
School of Aerospace, Transport & Manufacture
Cranfield University
Cranfield, United Kingdom
t.c.mackley@cranfield.ac.uk

***Abstract*— Many shortfalls in problem solving can with hindsight be attributed to applying the wrong approach for the specific problem and its situation or context. Having identified a problem it can then be both a challenge to determine strategies that will succeed in its solution and also to communicate the value of what is proposed to gain acceptance of the way forward. We encourage a "systems approach", but how do we determine the particular approach to take for problems as diverse as the next airplane concept compared with improving the UK National Health Service?**

The challenge addressed in this paper is to select an approach based upon both an understanding of the problem context and an identification of the severity of the problem in terms of the risk it poses to the problem solver. This paper proposes a method based on "Context types" characterized by four-quadrant matrices. It allows the assignment of qualitative risk to a problem which in turn allows the user to tailor a problem solving approach accordingly. Some Context types are derived from existing concepts of systems thinking, but others are devised in order to provide a more comprehensive analysis of complex problems.

Keywords— systems thinking; systems engineering and theory; context; complexity theory; risk analysis; engineering management..

I. INTRODUCTION

Many shortfalls in problem solving can with hindsight be attributed to applying the wrong approach for the specific problem and its situation or context. Having identified a problem it can then be both a challenge to determine strategies that will succeed in its solution and also to communicate the value of what is proposed to gain acceptance of the way forward. The challenge addressed in this paper is to select an approach based upon both an understanding of the problem context and an identification of the severity of the problem in terms of the risk.

The method described proposes *Context Types* to help analyze a problem context. The analysis of each type is reduced to a four-quadrant matrix, where a particular quadrant can be used to define the appropriate system thinking or systems engineering approach. Each quadrant is also related to the likely level of risk or difficulty in addressing the problem. The resulting level of risk is then expressed graphically as a Kiviat diagram in order to present it in a way that can facilitate communication and understanding by a wider audience.

In section II of this paper there is a review of the current theory that can be used to develop an understanding of the types of context and lead to suggested approaches. These are examined and, where appropriate are expanded to generate a set of generic Context types described as four-quadrant matrices. Using four-quadrant matrices and everyday terms, the concept of Context types is intended to be easy to use and communicate. New and complementary Context types are proposed in Section III with the aim of providing a more complete analysis of the problem context. Section IV outlines how the Context types can be used to suggest problem solving strategies and indicate the level of complexity and risk involved.

The intention is to present problem solving through Context types as a method of guiding a problem solving strategy that is easy to understand and implement. By describing how to address problem contexts of different types, the method presents a unification of existing systems thinking approaches to provide a problem solving approach that can be tailored to specific circumstances.

II. EXISTING CONCEPTS

There are a number of existing concepts that allow a distinction to be made between different types of context and these are outlined below.

A. Problem types

Problem types characterize a problem in terms of uncertainty in requirement and in solution [1].

Painting-by-numbers (PBN) – clear objective and clear solution

Foggy – uncertain objective, uncertain solution

Movie – uncertain objective, clear solution

Quest – clear objective, uncertain solution

Obeng defines a *Painting-by-numbers* problem as one where "you and most stakeholders are sure of both what to do and how it is done" based on similar experience. The fact that the problem is well defined and there is a clearly defined

© 2015 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

solution, means that technical, cost and timescale risk can be well identified; the challenge is perhaps to do it better.

A *Foggy* type of problem is very different in that "you and most of your stakeholders are unsure of what is to be done and unsure of how it is to be achieved". The secret of success here, according to Obeng is to "proceed very carefully, to proceed one step at a time".

In the *Movie* type "you and most of your stakeholders are very sure about how the project should be conducted but not what is to be done". Typical expertise and facilities are in place, either looking or waiting for the problem to be tackled. In a Movie, Obeng says that concentration should be on "finding yourself a good script and the movie will write itself".

For a *Quest*, "you and most of your stakeholders are sure of what should be done…however, you are unsure of how to achieve this". The secret here, Obeng says, is to "get your knights fired up and send them off to seek [a solution] in parallel".

Obeng's aim is to identify that not all problems are of the same type and used characteristics of uncertainty in both objective and solution to categorize them. In doing so he emphasizes that a single approach was not appropriate for all. His four types are already arranged as a four quadrant matrix as shown in Fig. 1.

B. *Management type*

In the early 1960s Jay Forrester applied the concept of System Dynamics to the industrial organization [2]. This provided the basis for further work [3] using System Archetypes to analyze a business as a system. The basis of his contention was that dynamic situations could be described in terms of reinforcing and balancing loops of cause and effect and that simulation using archetypes is able to replicate the behavior of ailing organizations thus providing a diagnosis of the reasons for their malaise. The original set of around ten systems archetypes can be expressed as a reduced set of four; Underachievement, Relative Underachievement, Out-of-control and Relative Control archetypes [4]. These represent situations where there is either a problem in terms of availability of resource or in terms of an inappropriate control action being applied [5]. In this context type an assumption is made that inappropriate control is applied unintentionally and therefore as a result of a lack of situational awareness.

Using axes of "lack of situational awareness" and "inadequacy of resource" the four quadrant matrix of Fig. 2 can be identified.

C. *Values type*

The concept of Divergence of values [6] consists of unitary, pluralist and conflicting/coercive situations:

Unitary - in that they all have a common goal and view of what is to be achieved and ultimately how.

Pluralist - in that stakeholders cannot agree on goals and tend to pursue their own objectives, but that there is mutual benefit in the collaboration.

Conflicting/Coercive - in that goals and objectives diverge, but that some group or groups get their way at the expense of others.

These situations are interpreted as distinguishing between the number of different viewpoints, and the degree of conflict that exists between stakeholders. In a collaborative environment an increasing number of viewpoints change a situation from *unitary* to *pluralist*. However, where there are conflicting priorities increasing the number of viewpoints will turn a situation from a *coercive* or simple conflict into *anarchy*. The resulting Context type is shown in Fig. 3.

D. *Complexity type*

This concept makes the distinction between Detailed and Dynamic Complexity [7], which is a reminder that a difficult problem can be complicated due to the number of parameters to be considered, whereas it can be complex due to the nature of its interactions or interconnections.

This type characterizes the level of complexity exhibited by a system solution:

Detailed Complexity – where complexity is due to the number of variables (but cause and effect is clear).

Dynamic Complexity – complexity where the relationship between cause and effect is not clear or deterministic.

Senge's complexity types describe different types of complexity and as such need to be counterbalanced by less complex situations; as not all problems are complex or complicated. *Detailed complexity* and *dynamic complexity* are cases where there are many variables, but with differing levels of interaction. Increasing levels of interaction with a small number of variables might be termed as ranging from *back-of-the-envelope* to s*imple dynamics* respectively (Fig. 4).

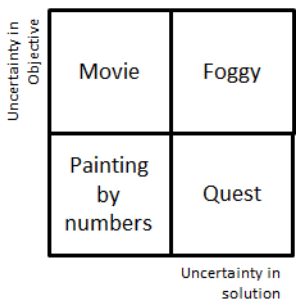

Fig. 1: Problem type

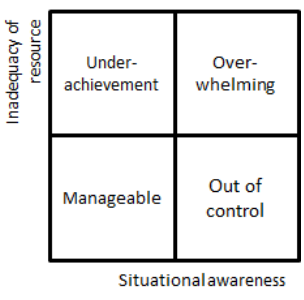

Fig. 2: Management type

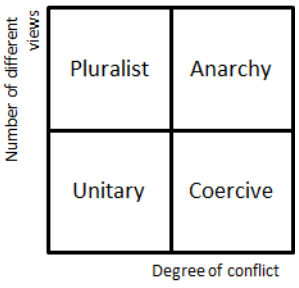

Fig. 3: Values type

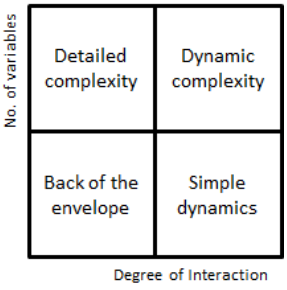

Fig. 4: Complexity type

E. Co-ordination type

Finally Meier [8] distinguishes between types of organization of a system from a unitary system to a system of systems, on the basis of operational and development independence of its components. His definition for a system-of-systems is:

"an assemblage of components which individually may be regarded as systems, and which possesses two additional properties:

Operational Independence of the components: if the system-of-systems is disassembled into its component systems the component systems must be able to usefully operate independently. That is, the components fulfill customer-operator purposes on their own.

Managerial Independence of the components: the component systems not only can operate independently, they do operate independently. The component systems are separately acquired and integrated, but maintain a continuing operational existence independent of the system-of-systems."

Maier's concept of *system-of-systems* contrasts with a unitary or *centralized* system; a system-of-systems displays both development and operational independence whereas the *centralized* system has neither of these. Considering solely development independence will lead to an *off the shelf* solution (i.e. assembled from separately developed components), whereas solely operational independence implies an *asset management* case (see Fig. 5).

III. Further Context Types

This section, additional context types have been developed to complement the five generated from current theory. To cover the variety of problem situations a total of eleven Context types are described. In each case it is useful to keep in mind the question "how critical could this Context type be to influencing the required approach of the problem solver?"

A. Evolution types

Obeng's Problem Type concept is an established way of addressing a particular problem at a given time, but often the challenge comes from how the problem changes over time. Important considerations are: how much has the requirement changed; what is the uncertainty of the requirement or in the solution as a result; when and how often does the problem need to be addressed to ensure continuous capability provision?

The rate of change of requirement is important as this will tend to erode any spare capacity built into the system or may expose areas where the system currently has no inherent capability. This will determine how long it will be before the system is in capability deficit and will drive the time at which modification is required as well as the duration of modification activity that can be tolerated. For instance, in a rapidly changing environment, capability may need to be updated on a regular basis and the time taken to perform the update must be consistent with those challenging timescales in order to converge upon a solution before further updates are required. Equally the uncertainty in requirement is important as this will drive the type of approach needed to address the capability update and indicate the time that the activity is likely to take. Effectively this is predicting the Problem type [1] that is likely to be encountered at the time in the future when the modification will be required [9].

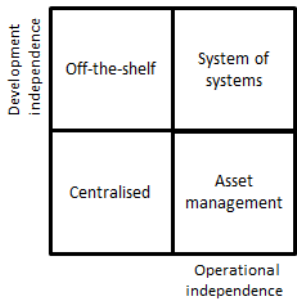

Fig. 5: Co-ordination type

For this type the axes of the four quadrant matrix are uncertainty in future objectives and uncertainty in future solution. If the future objectives and solution are clear, then the situation will be one of routine *obsolescence management*. This could be the situation for road vehicle rental firms; vehicle design has remained fairly invariant over many years and the users expectations are very much in line with what a current road vehicle can provide. However what if the future objectives or possible solutions were not known? Imagine that current vehicle solutions based on oil based fuels were becoming less economic and vehicles using alternative energy become more attractive – broadening the business to consider these would be seen as *opportunity development*. Conversely, if we imagined that the technologies of cars in the future are to become expensive and cars or their components become leased then this is more an area of *service development* (such as leasing of batteries for electric cars). A rapidly changing environment with novel and emerging solutions could be termed as represents *capability development*, resembling the approach often taken in military development, but would arguably fit well with mobile computing and communication solutions. Evolution types can thus be identified as in Fig. 6.

B. Response type

The focus for this type is the urgency of the need. Depending on the complexity of the problem, a more or less urgent need will have a bearing on the approach taken. To characterize urgency a distinction is made between developing a solution under normal commercial conditions i.e. working in a viable and competitive situation, and an emergency situation where corners are allowed to be cut or significant extra resource is justified. Urgent but non-complex situations can be addressed by cutting corners as the consequences of this can be evaluated. If a situation is both urgent and complex then simple measures are often not appropriate as they may have consequences that in themselves can have serious implications. In the matrix below the distinction is made between the former, similar to the Urgent operational requirement process employed by military organizations and the latter being a *systemic emergency*. An example of a systemic emergency might be an outbreak of a highly virulence strain of flu and its effect on a countries health service and economy. *Routine* and *systemic development* make up the four quadrants of Fig 7.

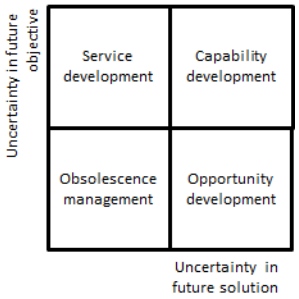

Fig. 6: Evolution type

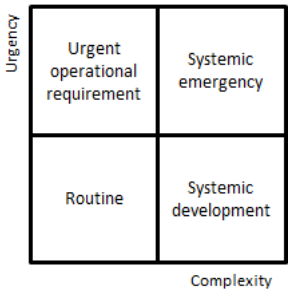

Fig. 7: Response type

C. *Situation type*

It is clear that the starting point will have a significant bearing on the solution and so this aspect will be key in determining the approach required. The following situations might be encountered based on differences in uncertainty of design baseline and the degree of change required.

The Situation Type involves consideration of what the starting point of the activity is. For instance this may be:

- Design starting from a *clean sheet*, with little or no previously defined concept of design or legacy constraint (e.g. new capability acquisition). The truly clean sheet is not a common situation for the system designer, although it is perhaps more prevalent in some domains than others (e.g. defense).
- An *upgrade* of capability, where the starting point is going to have a considerable bearing on the solution that might be chosen (e.g. mid-life update). In this situation it will be normal to identify the "capability gap" that needs to be met.
- A need for *system review*, to identify changes required to the system baseline to be fit for the existing purpose, rather than from the definition from stakeholders of a required change in capability.
- Simply a *reconfiguration* of what is already in place, but used in a different way to solve the problem. In isolation this is a relatively simple case, but it can also describe a system-of-systems which provides challenges of its own (see Coordination type).

The four quadrant matrix for Situation type is given in Fig.8.

D. *Risk type*

Risk and maturity are key elements of a system development that should be considered together. With an immature system, achieving the desired system outcome without clear knowledge of probability of success or related consequences represents a risk. Equally, a relatively mature system can be a risk if there are severe consequences should it fail. Engineers work at trying to find a suitable balance between risk and maturity of a system design. The preference is a mature/low risk combination or a *no brainer*, but for higher risk situations a project may choose a mature solution to *play it safe*. If there is solution immaturity then low risk solutions represent *calculated risks*, with a high risk/immature solution being a *gamble*.

The four quadrant matrix for Risk type could be drawn as in Fig. 9.

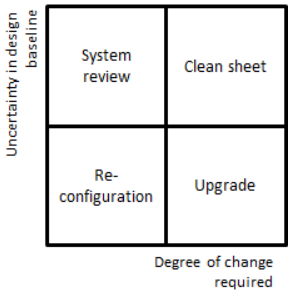

Fig. 8: Situation type

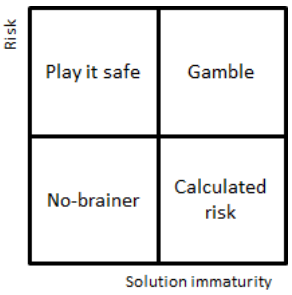

Fig. 9: Risk type

E. *Target type*

Enterprises will often find themselves facing different types of target. Some enterprises are required to deliver to strict timescales and others might have a reputation based on the quality of their product or service. As shown in Fig. 10, these represent orthogonal axes, where a high quality challenging target situation can be seen as an O*lympic sprint* compared with a relaxed timescale at a familiar and achievable quality being the *stock in trade*. *Critical path* and *gold standard* provide the remaining quadrants.

F. *Business area type*

A particular challenge for a business is to ensure it has the capability to deal with a problem, and in particular that it has a properly trained and prepared workforce. A distinction can be made between the requirements that a given context places on expertise that is gained with professional qualifications on the one hand, compared with experience on the other. Whereas expertise might be acquired quickly, experience has to be accumulated over time: in some areas, expertise is in short supply and that introduces challenges of its own. Types of work are often referred to as "collar workers", but the different "collars" do not always reflect the distinction of education and experience, so categories of *low skill*, *professional*, *trades*, *gold collar* have been chosen as in Fig. 11.

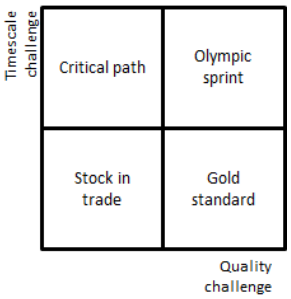

Fig. 10: Target type

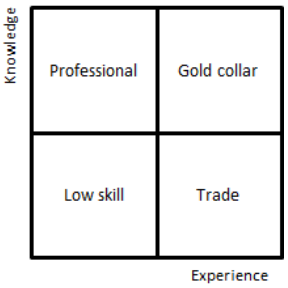

Fig. 11: Business area type

G. Combinationed types

An analysis of the identified types shows that there are common axes. For instance, risk type compares risk against solution immaturity whereas Obeng's problem types compare solution immaturity to objective uncertainty. This allows the combination to be described as a 3D matrix introducing types of; *play it safe PBN*, *surefire success movie*, *critical quest,* and *freezing fog*. This combination can be described as "Problem risk type".

Also the types of response and complexity share an axis of degree of interaction, which leads to urgency being compared with both number of parameters to be considered and degree of interaction. This introduces types of *urgent operational requirement*, *balanced scorecard*, *tiger team*, *systemic development* and *systemic emergency*. This is described as "Urgent complexity types".

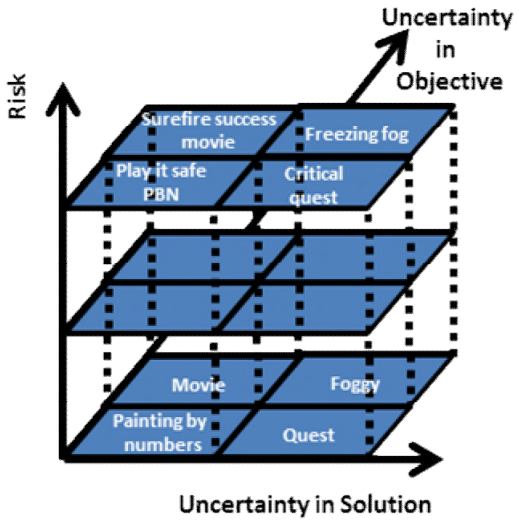

Fig. 12: Problem risk

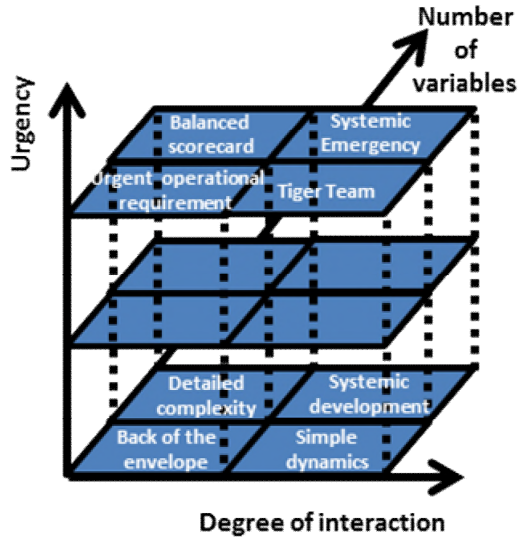

Fig. 13: Urgent complexity

IV. Problem Solving Approach and Risk Evaluation

The use of the four quadrant matrix for describing each Context type, allows a spectrum of context to be identified. Each matrix is structured in such a way that risk increases as the value of any single axis increases. Fig. 14 is numbered to provide a reference and, in coarse terms we might conclude that quadrant 1 represents low risk, quadrant 4 represents high risk and quadrant 2 and 3 indicating a medium risk. Thus an overall context risk might be evaluated by identifying where a given context falls for each of the types.

For identifying risk it is important to ensure that all potential contributors are considered. There is perhaps no guaranteed way of determining that the list of Context types addresses all elements of potential risk in a system solution to a problem and this is an area which deserves further analysis against more traditional risk indices. However, it is possible to identify key domains of a system's problem and solution space that should be considered. Key domains of a system have been described as: product and producing domains [10]; product, process and organization [11]; customer, functional, physical and process [12]. These can be combined to give domains of *requirement*, *solution*, *process* and *organization*. Mapping the eleven Context types to these four domains there is coverage in each domain with either two or three types each.

The division is shown in Table I. Table I also shows a simple illustration of the approach for two example problems. Imagine being asked to address problems facing the UK National Health Service, or being asked to work out a strategy for developing a new concept of airplane based on a new distributed propulsion concept. The table shows an analysis of both problems; the crosses represent the problem of developing the new aircraft and the ticks represent the problem of addressing the challenges of the UK National Health Service (NHS). The risk profile for the distributed propulsion problem is analyzed as 4,4,3 and so seems to represent a medium risk, with a fair balance between areas that are risky and manageable: the situation for the problem of the NHS shows a risk "profile" of 0,5,6 which indicates no easy areas, with risk in almost half the areas being high. For distributed propulsion the risk is reasonably well distributed across the system domains and therefore requires a balanced approach: for the NHS there are significant organizational risks to overcome and these stand-out compared to risks of process, requirement and solution.

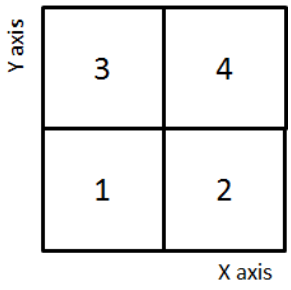

Fig. 14: Reference matrix

TABLE I. CHARACTERIZING RISK: EXAMPLES

Type	Quadrant 1	Quadrant 2,3	Quadrant 4
Process			
Problem		X √	
Evolution		X √	
Response		X	√
Requirement			
Situation		√	X
Divergence of values	X	√	
Management	X		√
Solution			
Risk		X √	
Complexity			X √
Organization			
Coordination	X		√
Target	X		√
Business area			X √
Summary risk	**X (4) √(0)**	X (4) √ **(5)**	X (3) √ **(6)**

This can be effectively visualized using the Kiviat diagram (Fig. 15), which gives an immediate pictorial view of what areas represent the greatest risk (with 1, 2 and 3 being low, medium and high risk respectively).

The four quadrant matrices can be used to gain an idea of overall difficulty by considering each type individually and assessing the combination of the outcomes. However, this four quadrant notation has the risk of dividing up the problem without considering the interactions. As this is is a qualitative tool to inform a strategic approach, these overlaps are considered small and are expected to be addressed in ensuring a coherent strategy for the whole problem. Some overlap in the Context types can readily be identified by the Combined types, which reflect combinations of issues that should be addressed to identify their impact on the approach taken. In the examples given the *Problem risk type* results in a *critical quest* for both the UK NHS and new aircraft concept, whereas the *Urgent complexity type* emphasizes a *systemic emergency* for the NHS rather than a *systemic development* for the aircraft.

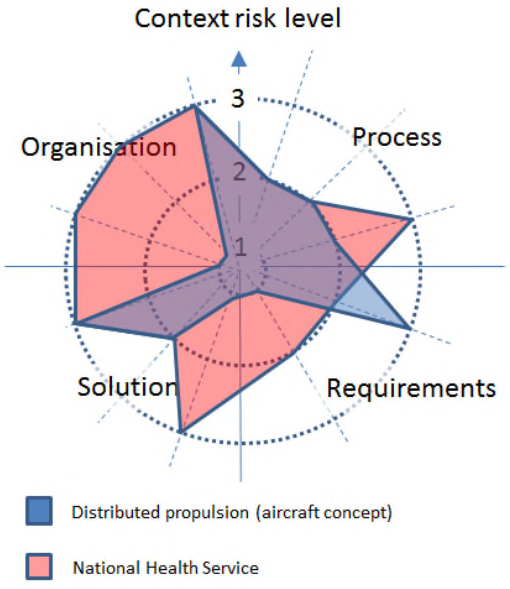

Fig 15: Kiviat diagram presenting risk profile

Consideration of context type can have a bearing on the approach used; some examples are identified in Table II. Further work is ongoing to elaborate this method as part of a comprehensive system design method, with Context types enabling a guide to the system design strategies to be employed.

BIBLIOGRAPHY

[1] Obeng, E. (1995). *All change! The project leader's secret handbook.* Pearson Publishing.

[2] Forrester. (1961). *Industrial dynamics.* Cambridge MA: Productivity Press.

[3] Senge, P., Kleiner, A., & Roberts, C. (1994). *The fifth discipline fieldbook.* London: Nicholas Brearley Publishing.

[4] Wolstenholme, E. (2003). Towards the definition and use of a core set of archetypal structures in system dynamics. *System Dynamics Review, 19*, 7-26.

[5] Wolstenholme, E. (2004). Using generic system archetypes to support system thinking and modelling. *System Dynamics Review, 20*, 341-356.

[6] Jackson, M. (1994). Critical systems thinking: beyond the fragments. *System Dynamics review, 10*, 213-229.

[7] Senge, P. (1990). *The fifth discipline: the art and practice of the learning organization.* New York: Doubleday Currency.

[8] Maier, M. (1998). Architecting principles for systems-of-systems engineering. *Systems Engineering, 1*(4), 267-284.

[9] Mackley, T., Deane, J., & John, P. (2010). Addressing the time dimension and agility in the provision of capability. *System of Systems Engineering (SoSE), 2010 5th International Conference*, pp. 1-8. Loughborough.

[10] Mackley, T. (2008). The role of lifecycle systems in the through-life engineering of system solutions. *INCOSE Internation Symposium*, *18*, pp. 691-705. Utrect.

[11] Eppinger, S. (2002). Design reserach theories, methodologies and product modelling. *International Conference on Engineering Design ICED 01*, (pp. 283-290).

[12] Suh, N. (1990). *Principles of design.* New York: Oxford University Press.

[13] Hitchins, D. (2007). *Systems Engineering. A 21st Century Systems Methodology.* Chichester: Wiley.

[14] Checkland, P., & Poulter, J. (2006). *A short definitive account of Soft Systems Methodology and its use for practitioners, teachers and students.* Chichester: Wiley.

[15] Ulrich, W. (1987). Critical Heuristics of social systems design. *European Journal of Operational Research, 31*, 276-283.

ACKNOWLEDGMENT

Thanks go to Prof Philip John for his collaboration and support in contributing ideas to this paper.

TABLE II. EXAMPLES STRATEGIES TO EMPLOY FOR EACH CONTEXT TYPE

Type	Sub-type	Approach
Situation type	Reconfiguration	Design is largely unchanged, but requalification is required for any new operational requirements. This requires examination of the use of the systems and the conditions involved to determine if there has been an extension to the performance envelope that will need to be re-qualified.
	System review	In cases where there is no scheduled system upgrade, but there are clear symptoms of the system performing beneath the desired performance levels the first step will be to diagnose an agree the appropriate way forward. In order to identify, analysis and diagnose the root causes of underperformance, it is appropriate to employ an issue or soft systems analysis such as the Rigorous Soft Method [13] or Soft Systems Methodology [14]
	Upgrade	Upgrades will usually reflect a need to modify the system as elements have become obsolete, or because an insertion of new technology is desired. A modular design, with standardized interfaces will enable replacement of affected modules and result in a minimal requalification for the upgraded build standard. Upgrade is facilitated if it part of a pre-envisaged Incremental Development Lifecycle model.
	Clean sheet	Design from new using an exploratory, capability based approach. Suited to an initial approach of implementing a spiral based lifecycle and verified for completeness against methods such as complete systems methodologies such as Hitchins' Generic Reference Model [13].
Values type	Unitary	Approach can be based on Consensus, with clear definition of boundary and architecture. It is perhaps the simplest case for systems engineering, where there is a clear overriding client objective and other stakeholder requirements are defined purely as constraints on the design. Trade-offs will generally be at the design level.
	Pluralist	In contrast to the Unitary case, there will be different driving perspectives on the objectives and priorities will differ. The approach will be subject to agreement based on compromise, perhaps resulting in hybrid architectures. Discussions will need to be informed trade-off studies at the requirements level. Soft systems methodologies can be used to establish suitable compromises.
	Conflict/Coercive	Stakeholder views may appear unitary, but mask coercion. Ulrich`s Critical Heuristics [15] can be used to establish where the system boundaries ought to be. Regulation may subsequently be required to enforce appropriate architecture.
	Anarchy	There is no sense of centralized objectives and responsibility, and therefore no coordinated strategies for achieving outcomes. No meaningful structure exists. Architectural rules and structure need to be established and enforced.

TABLE II. EXAMPLES STRATEGIES TO EMPLOY FOR EACH CONTEXT TYPE (CONTINUED)

Risk type	No-brainer	Solutions to problem have relatively little risk exposure due to experience. This requires following the established process gained by experience. In cases of an established process, then "Lean" techniques can be considered to improve time, cost or quality.
	Play it safe	The consequence of risks involved requires an established, tried and tested approach. This tends to be highly procedural based on previous experience, with changes to established architectures and design being resisted due to the effort and cost of the involved requalification.
	Calculated risk	Despite lack of maturity, the relatively low level of risk means that a trial based approach is both acceptable and desirable as it can provide validated outcomes to converge on the solution.
	Gamble	This applies for situations where there is tangible risk to the system or its context and would be a safety or security issue or other situations with significant implications on an enterprise; typically this would be a situation where options are limited. With the lack of confidence, fail safe measures should be incorporated to limit the damage in case of unforeseen or consequential effects.

https://dspace.org/

A problem solving method using context types

Mackley, Tim

2015-10-26
IEEE
https://doi.org/10.1109/SysEng.2015.7302795
Mackley T. A problem solving method using context types. In: IEEE International Symposium on Systems Engineering (ISSE), 28-30 September 2015, Rome, Italy

Downloaded from a DSpace repository. https://dspace.org/