

CRANFIELD UNIVERSITY

H AZIZ

A METHODOLOGY FOR THE DISTRIBUTED AND COLLABORATIVE
MANAGEMENT OF ENGINEERING KNOWLEDGE

SCHOOL OF APPLIED SCIENCES

PhD THESIS

ProQuest Number: 10832140

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



ProQuest 10832140

Published by ProQuest LLC (2018). Copyright of the Dissertation is held by Cranfield University.

All rights reserved.

This work is protected against unauthorized copying under Title 17, United States Code
Microform Edition © ProQuest LLC.

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346

CRANFIELD UNIVERSITY

SCHOOL OF APPLIED SCIENCES

PhD THESIS

ACADEMIC YEAR 2005-2006

H AZIZ

A METHODOLOGY FOR THE DISTRIBUTED AND COLLABORATIVE
MANAGEMENT OF ENGINEERING KNOWLEDGE

Supervisor: J Gao

September 2007

This thesis is submitted in partial fulfilment of the requirements for the degree of
Doctorate of Philosophy

© Cranfield University 2007. All rights reserved. No part of this publication may be
reproduced without the written permission of the copyright owner.

Abstract

The problems of collaborative engineering design and management at the conceptual stage in a large network of dissimilar enterprises was investigated. This issue in engineering design is a result of the supply chain and virtual enterprise (VE) oriented industry that demands faster time to market and accurate cost/manufacturing analysis from conception. Current tools and techniques do not completely fulfil this requirement due to a lack of coherent inter-enterprise collaboration and a dearth of manufacturing knowledge available at the concept stage. Client-server and peer to peer systems were tested for communication, as well as various techniques for knowledge management and propagation including Product Lifecycle Management (PLM) and expert systems. As a result of system testing, and extensive literature review, several novel techniques were proposed and tested to improve the coherent management of knowledge and enable inter-enterprise collaboration. The techniques were trialled on two engineering project examples.

An automotive Tier-1 supplier which designs products whose components are sub-contracted to a large supply chain and assembled for an Original Equipment Manufacturer (OEM) was used as a test scenario. The utility of the systems for integrating large VEs into a coherent project with unified specifications were demonstrated in a simple example, and problems associated with engineering document management overcome via re-usable, configurable, object oriented ontologies propagated throughout the VE imposing a coherent nomenclature and engineering product definition. All knowledge within the system maintains links from specification - concept - design - testing through to manufacturing stages, aiding the participating enterprises in maintaining their knowledge and experience for future projects. This potentially speeds the process of innovation by enabling companies to concentrate on value-added aspects of designs whilst 'bread-and-butter' expertise is reused. The second example, a manufacturer of rapid-construction steel bridges, demonstrated the manufacturing dimension of the methodology, where the early stage of design, and the generation of new concepts by reusing existing manufacturing knowledge bases was demonstrated.

The solution consisted of a de-centralised super-peer net architecture to establish and maintain communications between enterprises in a VE. The enterprises are able to share knowledge in a common format and nomenclature via the building-block shareable super-ontology that can be tailored on a project by project basis, whilst maintaining the common nomenclature of the 'super-ontology' eliminating knowledge interpretation issues. The two-tier architecture developed as part of the solution glues together the peer-peer and super-ontologies to form a coherent system for internal knowledge management and product development as well as external virtual enterprise product development and knowledge management.

In conclusion, the methodology developed for collaboration and knowledge management was shown to be more appropriate for use by smaller enterprises collaborating in a large Virtual Enterprise than PLM technology in terms of: usability, configurability, cost of system and individual control over intellectual property rights.

Acknowledgments

I would like to thank the Engineering and Physical Sciences Research Council (EPSRC) for their funding of the research project.

My professor Dr Gao who accepted me to participate in his team's research that made this thesis possible and who patiently aided me through the many years of the research and thesis write up, I owe my deepest gratitude to.

I would like to thank Denny (Wai M Cheung) who worked with me on the EPSRC side of the research and who made the journey in academia very enjoyable. Also Professor Maropoulos who's insight and direction helped me massively.

I'd like to thank all the staff at Cranfield university, and the colleagues at the Technical Research Centre in Finland for their support through my research years.

On a personal level I'd like to thank all the friends and fellow students Cranfield who helped the days while away in the "goldfish bowl" in building 50.

Finally but not least I'd like to thank my mum (Razika), dad (Ali), brother (Ayser), wife (Layla) and newborn daughter (Yasmine) in helping me get to the gritty end of the thesis despite the odds.

Contents

1	INTRODUCTION.....	16
1.1	Industrial Problems and Requirements.....	16
1.2	Aims and objectives	21
1.3	Scope of Project.....	23
1.4	Potential benefits to industry	24
1.5	Research Methodologies.....	25
1.5.1	Creating Functional specifications	26
1.5.2	Method for compiling literature survey, and analysing existing work.	26
1.5.3	eXtreme Programming methodology for application development.....	29
1.5.4	Methodology for configuring systems with company data	29
1.5.5	Methodology for user based functional testing	30
1.6	Structure of the thesis.....	30
2	ACADEMIC LITERATURE SURVEY.....	32
2.1	Virtual Enterprises	32
2.2	Inter-Enterprise collaboration.....	37
2.3	Peer to Peer networks.....	40
2.4	Engineering ontologies.....	45
2.5	Summary of academic literature and research gap	47
3	TECHNOLOGY REVIEW.....	49
3.1	PDM/PLM related issues.....	49
3.2	Conceptual Design Tools	52
3.3	Design for manufacture.....	57
3.4	Web based design.....	59
3.5	Knowledge based CAD.....	63
3.6	Manufacturing KBS.....	64
3.7	Feature Based Design	67
3.8	Application of STEP based tools.....	70
3.9	Summary of technology review.....	74

4	PROPOSED METHODOLOGY FOR DISTRIBUTED COLLABORATION	77
4.1	Communication as a foundation of collaboration	77
4.2	Format of data model and its structure for collaboration	79
4.2.1	Demonstration of communication architectures	82
4.2.2	Proposed 2 tier architecture for heterogeneous inter-enterprise collaboration	86
4.3	Medium for de-centralised collaboration	88
4.3.1	PLM systems	89
4.3.2	Open source client server applications	94
4.3.2.1	Open source databases	95
4.3.2.2	Open source Web-servers	96
4.3.2.3	Solid modeller	97
4.3.2.4	Runtime environments	97
4.3.2.5	Application Systems	98
4.3.2.6	Systems construction	100
4.3.3	Application Service Providers	103
4.3.4	Peer to Peer Systems	104
4.3.5	Proposed “medium” for VE	107
4.3.6	Summary of communication architectures	111
4.4	Summary of proposed methodologies	115
5	UNIVERSAL ONTOLOGY FOR ENGINEERING KNOWLEDGE MANAGEMENT	117
5.1	Ontologies	117
5.1.1	Knowledge Representation Methodology	121
5.2	Design knowledge translation	123
5.2.1	Interactive versus automated design generation	123
5.2.2	Design Knowledge Base	125
5.2.3	Interactive translation methods	125
5.2.4	Application within collaborative framework	127
5.2.5	Summary	128
5.3	Generic super-ontology description	129
5.3.1	Knowledge base hierarchy	129
5.3.2	Specifications	130
5.3.3	Workflows	132
5.3.3.1	Workflow of ArvinMeritor processes	133
5.3.3.2	Workflow implementation in ontology	135
5.3.4	Bill of Material	136
5.3.5	Personnel	137
5.3.6	Customer Supplier Management	138
5.3.7	Technical Details	141
5.3.8	Requirements	142
5.3.9	Summary	143
5.4	STEP models	144
5.4.1	STEP geometry and assembly generation	146
5.4.1.1	Proprietary CAD method	147
5.4.1.2	Open-standard method	149
5.4.1.3	Selected Solution	153

5.5	Summary	154
6	IMPLEMENTATION OF INDUSTRIAL CASE STUDY	156
6.1	Format implementation (tier-2).....	156
6.2	Medium Implementation (tier-1).....	159
6.3	Super-ontology KBS Implementation	161
6.3.1	Product design ontology	161
6.3.1.1	ArvinMeritor example	162
6.3.1.2	Customer requirements sub-ontology	167
6.3.1.3	Mabey&Johnson example.....	168
6.3.2	Programming methods for knowledge base	172
6.3.2.1	Objects in knowledge base.....	173
6.3.2.2	Classes for STEP ontology	175
6.3.2.3	JESS rules within the ontology	177
6.3.2.4	STEP Ontology	179
6.3.2.5	Generic collaborative ontology	182
6.3.3	User Interface	183
6.4	Summary	191
7	DISCUSSION, CONCLUSIONS & FURTHER WORK.....	194
7.1	Discussion	194
7.2	Conclusion	203
7.3	Suggestions for further research and follow on research by third parties.....	206
8	REFERENCES.....	207
❖	APPENDIX (A) SYSTEM DEPLOYMENT	215
▪	Software installation and setup	215
▪	Customer requirement entry	217
▪	Concept Design generation	219
▪	Collaboration.....	221
▪	Downstream processes	222
▪	Summary	222
❖	APPENDIX (B) TEST WORKFLOWS (ARVINMERITOR DATA).....	223
❖	APPENDIX (C) STEP ONTOLOGY IN KNOWLEDGE INTERCHANGE FORMAT	232
❖	APPENDIX (D) GENERIC COLLABORATIVE ONTOLOGY, ARVINMERITOR TEST EXAMPLE	258
❖	APPENDIX (E) GENERIC COLLABORATIVE ONTOLOGY, TEST DATA 282	

List of figures

Figure 1 Format, Medium, Ontology and workflow	78
Figure 2 Depth and breadth of traditional B2B connections	85
Figure 3 New B2B model with separate layer for authentication and establishing access-control	86
Figure 4 two tier communication architecture.....	87
Figure 5 High level architectural overview of open-source client server system and user interface	101
Figure 6 Super peer architecture showing the interaction within internal peer nets and interaction with external peers.....	108
Figure 7 JXTA configuration window and user interface	109
Figure 8 Query speeds over different types of topologies.....	114
Figure 9 RDF triple	119
Figure 10 Example RDF-graph triple of ArvinMeritor knowledge	120
Figure 11 Data persistence in document oriented systems.....	121
Figure 12 Pre defined queries match requirements to specifications	126
Figure 13 Interactive process for generating specification from requirements	127
Figure 14 top level classes in knowledge base	129
Figure 15 Knowledge base - specifications definition	130
Figure 16 specifying the reliability of latch opening.....	131
Figure 17 location of knowledge item for temperature range of the reliability of opening of latch	132
Figure 18 ArvinMeritor lifecycle	134
Figure 19 ArvinMeritor Phase1	135
Figure 20 ArvinMeritor Phase 1 project Go/No go	135
Figure 21 Workflow definition within knowledge base.....	135
Figure 22 Bill of materials component configuration	136
Figure 23 bill of material sub ontology structure	137
Figure 24 Personnel project time management	138
Figure 25 VE inter-enterprise relationship definition.....	139
Figure 26 claw and pawl rivet relationships	140
Figure 27 Technical details sub-ontology	141
Figure 28 Requirements specification	142
Figure 29 Sequence diagram of STEP file generation and storage	149
Figure 30 Translating EXPRESS schema to knowledge base.....	152
Figure 31 STEP P21 import sequence.....	153
Figure 32 ArvinMeritor Latch assembly model	163
Figure 33 arvinMeritor latch assembly items	164
Figure 34 concept development sequence real-life example.....	165
Figure 35 Mabey bridge joints.....	170
Figure 36 Assembly of Mabey bridge components.....	171
Figure 37 : RDFgraph of Mabey's knowledge structure, illustrating the hierarchical KBS ideally defining the company's data model.....	172
Figure 38 (above) superclasses containing the ontology under types, AP-224 entities and user interface elements.....	184

Figure 39 (left) AP 224 class structure	184
Figure 40 Material selection window	186
Figure 41 user menu for data entry functional specifications	187
Figure 42 Assembly management and creation interface.....	188
Figure 43 Configuration of assembly component	189
Figure 44 Ready-made queries to actively assist in configuring new designs using existing knowledge	190
Figure 45 Exploded view of knowledge base hierarchy, enabling the graphical navigation of the system.	191
Figure 46 Requirements entry	217
Figure 47 requirements entry form.....	218
Figure 48 user_interface for structured AP224 creation	219
Figure 49 Action entity form.....	220
Figure 50 selecting exiting Action_Method to be inserted into action object.....	221
Figure 51 logging into remote knowledge base.....	222

List of Tables

Table 1: Disadvantages of conceptual design sessions	19
Table 2 Query terms for literature review	27
Table 3 A breakdown of references from different journal Sources	29
Table 4 Data format comparison	84
Table 5 Windchill configuration.....	91
Table 6 Implementation of KM on PLM.....	92
Table 7 Comparison of P2P protocols.....	106
Table 8 Comparison of key features of tested system.....	112
Table 9 example domain data for an ontology	120

Glossary

2D – Two Dimensional (model representation)

3D – Three Dimensional (model representation)

3GL – Third Generation language (programming)

ABAP – Programming language for SAP ERP system

ACIS – Geometric kernel for 3D solid models

AFR – Automatic Feature Recognition

AI – Artificial Intelligence

AP – STEP Application Protocol e.g. AP214, AP224

API – Application Programming Interface

ASP – Application Service Provider

B2B – Business to Business

BoM – Bill of Material
B-REP – Boundary Representation Model
C++ - An object oriented programming language
CAD – Computer Aided Design
CAM – Computer Aided Manufacture
CALS – Continuous Acquisition and Lifecycle Support
CAPP – Computer Aided Process Planning
CE – Concurrent Engineering
CFD – Computational Fluid Dynamics
CLIPS - C-Lisp language to define knowledge bases and expert system rules
COM - Component Object Model
CORBA – Common Object Request Broker Architecture
CPC – Collaborative Product Commerce (an extension of PDM for inter-company data sharing)
CPD – Collaborative Product Development (an extension of CPC for product development)
CRM – Customer Relationship Management system
CSG – Constructive Solid Geometry
DAML-OIL – Predecessor to OWL
DFA – Design for Assembly
DFM – Design for Manufacture
DFMA - Design for Manufacture and Assembly
DfX – Design for X (can be assembly, manufacture, reliability, maintainability)
DTD – Data Type Definition (define XML tags)
EJB – Enterprise Java Beans (from J2EE standard)
EDI – Electronic Data Interchange format
ERP – Enterprise Resource Planning
ES – Expert System
EXPRESS –programming language used to define STEP complaint files.
FBD – Feature Based Design
FEA – Finite Element Analysis
GB – GigaByte

GT – Group Technology
GUI – Graphical User Interface
HTML – HyperText Mark-up Language
HTTP – HyperText Transfer Protocol
IDL – Interface Definition language of CORBA
IPR – Intellectual Property Rights
IT – Information Technology
ICT – Information and Communications technology
IDE – Integrated Development Environment
IEEE - Institute of Electrical and Electronic Engineers (US)
IGES - Initial Graphics Exchange Specification
ISO – International Standards Organisation
J2EE – Java 2 Enterprise Edition
J2SE – Java 2 Standard Edition
JAR – Java Archive
JDBC – Java Database Connectivity
JDK – Java Development Kit
JESS – Java Expert Systems Shell
JNI – Java Native Interface
JSP – Java Server Pages
JXTA – Juxtapose, P2P protocol
KBS – Knowledge Based System
KIF – Knowledge Interchange Format
KM – Knowledge Management
KQML – Knowledge Query Manipulation Language
LAN – Local Area Network
LDAP – Lightweight Directory Access Protocol
MCAD – Mechanical CAD
MRP – Materials Resource Planning
MRPII – Manufacturing Resource Planning
MSBO - Manifold Solid Boundary-Representation B-REP Object Entity
NC – Numerical Control (of machine tools)

NIIP – National Industrial Information Infrastructure Protocols
ODBC – Open Database Connectivity
OEM - Original Equipment Manufacturer
OLE – Object Linking and Embedding
OMG – Object Management Group
OO – Object Oriented (used in the context of programming and data modelling)
OSS – Open Source Software (GNU licence compliant)
OWL – Web Ontology Language (supersedes RDF)
P2P – Peer to Peer
PAL – Protégé Axiom Language
PDM – Product Data Management
PHP – Perl HyperTex Processor Language
PLM - Product Lifecycle Management
PSL – Process Specification Language
QS9000 – Quality Standard (used in automotive industry)
RAD – Rapid Application Development
RAM – Random Access Memory
RDF - Resource Description Framework
RFQ – Request for Quotation
RMI – Remote Method Invocation (Java Distributed Objects)
RPM – Revolutions Per Minute (hard disk speed)
SDAI – Step Data Access Interface
SME – Small to medium Enterprise
SQL – Structured Query Language
SSL – Secure Sockets Layer (e.g. implementation OpenSSL)
STEP - Standard for the Exchange of Product Model Data
TCP/IP – Transmission Control Protocol/Internet Protocol
UML – Unified Modelling Language (primarily used to model OO applications)
UNIX – A family of operating systems
URI – Universal Resource Indicator
VE – Virtual Enterprise
VR – Virtual Reality

WAN – Wide Area Network

WAR – Web Archive (J2EE application)

WfMC - Workflow Management Coalition (www.wfmc.org)

XHTML – XML Hypertext Markup Language hybrid

XML – eXtensible Markup Language

XMLRPC- XML Remote Procedure Call

XP – eXtreme Programming

XPDL – XML Process Definition Language

XSL – eXtensible Stylesheet Language

Related publications

Journal papers

H. Aziz, Dr J.X Gao, Prof P. Maropoulos, W.M Cheung, Application of Product Data Management Technologies for Enterprise Integration. International Journal of Computer Integrated Manufacturing, Volume 16 Number 7/8, (2003), 491-500.

H. Aziz, Dr J.X Gao, Prof P. Maropoulos, W.M Cheung, Open Standard, Open Source and Peer to Peer Tools and Methods for Collaborative Product Development. Journal of Computers in Industry, (2005), Vol. 56, 260-271.

Conferences

H. Aziz, Dr J.X Gao, Prof P. Maropoulos, W.M Cheung, "Advanced Tools and Technologies for Collaborative Product Development and Knowledge Management", 18th National Conference on Manufacturing Research (NCMR), Leeds, UK, September, (2002), 115-120.

H. Aziz, Dr J.X Gao, Prof P. Maropoulos, W.M Cheung, "Application of Product Data Management Technologies for Enterprise Integration ", 1st CIRP International Seminar in Digital Enterprise Technology (DET02), Durham, UK, 16-17th September, (2002), 273-280.

H. Aziz, Dr J.X Gao, Prof P. Maropoulos, W.M Cheung, "A Design Environment for Product Knowledge Management and Data Exchange", *2003 CIRP Design Seminar*, Grenoble, France, 11th-14th May (2003), 43-54.

H. Aziz, Dr J.X Gao, Prof P. Maropoulos, W.M Cheung, "New Technologies and Standards for Design Knowledge Management in Collaborative Conceptual Design", *9th International Conference of Concurrent Enterprising (ICE2003)*, Dipoli Congress Center, Espoo, Finland, 16-18 June (2003).

H. Aziz, Dr J.X Gao, Prof P. Maropoulos, W.M Cheung, "Tools and techniques for distributed collaborative design and knowledge management", *1st International Conference on Manufacturing Research*, Glasgow, UK, 9th-11th September (2003), 339-344.

H. Aziz, Dr J.X Gao, Prof P. Maropoulos, W.M Cheung, "Open Standard, Open Source and Peer to Peer tools and methods for collaborative product development", *12th Symposium of Product Data Technology EUROPE*, Manchester Conference Centre, UMIST, November (2003), 151-158.

Conference Proceedings

H. Aziz, Dr J.X Gao, Prof P. Maropoulos, W.M Cheung, "Case study of collaborative knowledge management in the early phases of product development", *Perspectives from Europe and Asia on Engineering Design and Manufacture, A Comparison of Engineering Design and Manufacture in Europe and Asia*, (ed.) Xiu-Tian Yan, Cheng-Yu Jiang and Neal Juster, Kluwer Academic Publishers, (2004), ISBN 1-4020-2211-5, 283-295.

H. Aziz, Dr J.X Gao, Prof P. Maropoulos, W.M Cheung, "A design environment for product knowledge management and data exchange", *Methods and Tools for Co-operative and Integrated Design*, (ed.) S. Tichkiewich and D. Brissaud, Kluwer Academic Publishers, (2004), ISBN 1-4020-1889-4, 257-267.

1 Introduction

1.1 Industrial Problems and Requirements

Collaborative engineering design and management in a large network of dis-similar enterprises has become a necessity in the new world of globalisation and outsourcing within the engineering industries. Formulating techniques and technologies in this area have been evaluated in depth by (Wang et al 2002) in their review of collaborative conceptual design tools and techniques. Highlighting the gap that exists between the need for collaboration and the available tools which enable simple 'sharing' of design knowledge, the authors concluded that there is a major gap in the techniques to enable real time collaboration in a heterogeneous environment. Previous research on the topic had studied various aspects of the problem including the Palo Alto Collaborative Testbed (PACT) by the Advanced Research Projects Agency (ARPA) where existing engineering tools were integrated using an encapsulating agent (Cutkosky 1993) upon which a number of developments were developed by research teams including JATLite, a Java based agent infrastructure (Jeon et al 2000) and METAMORPH agent based architecture (Shen 2000) all of which shared the same principle of creating agents integrated with the individual applications to handle the collaboration between tools. The collaboration aspects of Virtual Enterprises (VE) in manufacturing were coined (Byrne 1993) as the 'virtual corporation'. This was further evolved into the ideas presented by (Barnett et al 1994) in 'an architecture for the virtual enterprise' which laid out the logical sequences for the establishment, operation and dissolution of a VE. Further elaboration on by (Hardwick 1996) where the basic features of a VE were laid out, and techniques utilising the Standard for the Exchange of Product models (STEP) standard and client-server web based internet for enabling a virtual collaboration. This has so far been implemented in real life in the current crop of Product Lifecycle Management systems in use today such as Enovia (www.ibm.com), Windchill (www.ptc.com) and SAP (www.sap.com).

However Hardwick's idea of a VE was challenged by (Zhang 2000) who proposed an online service for VEs to translate their shared knowledge into STEP files using an

online client-server application. Though such services are in use today, they do not aid VE in real-time collaboration or knowledge sharing because they are limited to sharing 'model' information only in a coherent manner and do not enable the sharing and editing of information outside the scope of the STEP model. Other perspectives on VEs include (Camarinha Matos et al 1999), on the other hand (Reid 1999) realised the inherently de-centralised nature of a VE and formulated processes for creating and managing the VEs without actually suggesting a technical implementation for VEs. Reid's formulation for the inherent de-centralised nature of a virtual enterprise forms the core pillar of this research's network topology for collaboration.

From an engineering aspect, there is still a lack of intuitive de-centralised collaboration at the conceptual stage of product development processes due to the absence of effective tools and techniques to translate early design ideas and manufacturing knowledge into analysable forms, and make use of existing knowledge and expertise to aid the development process. The meaning of collaboration within the context of engineering design is the sharing of existing engineering design expertise between individuals from within a VE that is composed of entities that are collaborating in a one-off venture and have not collaborated previously. The entities are presumed to be dissimilar, meaning that they have a different internal company operation, product development process and tools to the other participants in the VE.

This impedes decision making, especially in areas of product configuration, lifecycle costing, responsiveness to changing markets and meeting customer demands. This is relevant for many discrete-component, configuration controlled product engineering companies such as aerospace and automotive supply.

Engineering companies with a variety of Original Equipment Manufacturers (OEMs) customers face a huge problem with meeting the individual requirements of their clients. Interaction with client and product definition primarily takes place during the early concept stage of product development. Thus it is imperative that participants in the VE network (clients, engineers, and VE entities) can interact effectively and produce concept designs for the product that are practical to manufacture, cost effective and do

not repeat the same mistakes made in earlier projects (this of course does not preclude the addition of new design and thought flaws). Mistakes made at this stage produce an exponential rise in problems/cost/time issues at the engineering design, testing and manufacturing stages, thus companies aim for zero changes in design specifications once the concept is approved. However, the partition of conceptual design engineering from the manufacturing and assembly knowledge creates particular problems in the Design for Manufacture and Assembly (DFMA) synthesis of the concept.

Conceptual design is an essential part of the product lifecycle, which has a disproportionate effect on the costs, and eventual success, of the artefact. Traditionally this phase of the project is carried out in face to face meetings and brainstorming sessions with many delegates, from the different disciplines in the enterprise, with external customers and suppliers also taking part. According to the author, these conceptual-design-sessions have a number of disadvantages as summarised in Table 1.

<i>Disadvantage</i>	<i>Reasoning</i>
Gathering geographically disparate people regularly	expensive and time consuming
Scheduling all members at the same time	Conflicts with individual schedules of members, especially when a large number of participants are involved
Evaluating generated ideas	have to be evaluated separately and then the group reconvened to discuss results
Transformation of ideas into concepts	Ideas generated have to be recorded several times over, and provide no “template” or starting point for the detailed design in a digital form
Storing gathered generated knowledge	/ Not automatically classified and stored within an intelligent knowledge base. Memos and minutes of the meeting cannot efficiently describe “context” “rules” of the knowledge effectively or maintain links between different chunks of knowledge
Reuse of existing knowledge	depends on the experience and memory of the members in the meeting, and is open to bias

Table 1: Disadvantages of conceptual design sessions

The compound effects of mistakes in the conceptual design phase can cause massive disruption at later stages of the lifecycle. These include design for assembly, design for manufacture, eco-friendly design, and ease of maintenance during the product life and disposal.

In an ideal world there would be a tool or method to give a holistic view of the product lifecycle at all stages and ease the resolution of the design to meet required specifications from the conceptual stage accessible via a simple interface to ensure ease of use, and thus adoption by novice participants. A number of research papers tackling this issue are reviewed in the next chapter; however, there are a number of facets that the author considers to be basic flaws in current thinking on the subject.

Large enterprises have invariably been the target of software vendors and researchers for such tools, resulting in centralised applications inherently advantageous for larger enterprises. These are beyond the means of small to medium enterprises (SMEs) both in terms of capital expenditure on the system and in terms of the associated complexity and difficult interoperability with the large number of associated companies and clients utilising diverse Information and Communication Technology (ICT) systems (in fact, the lower the company is down the supply chain, the more OEMs it works with and thus is exposed to working with more diverse business systems' data standards and nomenclature).

In spite of research and development being focused on large enterprises, they still face numerous difficulties with systems such as Product Lifecycle Management (PLM) because enterprises evolve. An evolving enterprise needs an evolving data management system. However large PLM and Knowledge Management (KM) applications, such configuration changes have to be made centrally by IT staff, not the participants (such as engineers, designers, manufacturing personnel) who understand and use the systems' functions and the data models' intricacies.

The third problem arises when enterprises wish to collaborate with a large number of suppliers and original equipment manufacturer (OEM) customers (by large number, this research assumes a many-many relationship). Current applications enable collaboration using business-to-business (B2B) protocols. However, these do not take into account that disparate enterprises do not have unified data models or workflows. This lack of standardisation results in a lot of time and money spent mapping B2B links between different vendor's systems as well as the uniquely customised data models and workflows implemented on top of vendor-specific applications. Within the setting of a Virtual Enterprise (VE), where alliances can include dozens of enterprises and often last for only a single project, these costs would be prohibitive (because it would in effect entail repeating the task of inter-enterprise integration between each enterprise collaborating in the VE). This is a strong factor in reducing the ability of large enterprises to participate in one-off VE partnerships, reducing agility and potentially losing business.

Some solutions have included the use of application service providers (ASP) or enabling a large enterprise's data system to function as an ASP for their collaborators' benefit in a VE setting. However the problem with this type of system is the management of intellectual property of the individual enterprises and whether enterprises can give up control of their information to a partner, who may, especially in the case of small and specialised enterprises, also be collaborating with their competitors simultaneously. Such fears can potentially hinder the formation of VEs between enterprises of any size. Though naturally the larger, richer enterprises can make use of a third-party Application Service Providers (ASP) that smaller enterprises cannot afford.

The enterprises' prime asset, its knowledge, whilst stored in an electronic format, is not managed coherently thus perpetuating an 'invisible' limit on the company's knowledge, based on the impulse of knowledge workers' recollections of previous experience. Researchers and vendors have invariably attempted to solve the problem of coherency through centralisation; however this is not applicable in a collaborative and distributed development environment which is inherently de-centralised.

The attempts at inter-enterprise collaboration are further complicated when enterprises with different depth of detail in their data models and different taxonomies (in this research, taxonomy is a term used to denote the data model, or models, used internally by an enterprise to manage its product and project data) collaborate together. Creating a situation in which detail is lost in the transfer from high to low detail enterprise data systems.

Finally, the prime reason for this investigation is that SMEs have up to now been left out of developments in PLM, knowledge management and VE developments in spite of forming the majority of the world's engineering community. This negatively affects the ability of SMEs to manage their knowledge, reuse existing expertise, collaborate with other SMEs in the virtual enterprise settings and collaborate with the larger enterprises that form their customer base. This weakness affects not only the SMEs but also the large enterprises that utilise SMEs within their supply chain or VE network.

Smaller engineering enterprises lack the infrastructure and manpower for complex solutions. Additionally, it was found that employees simply do not have the time or the budgetary resources to learn the necessary application functionality and new concepts associated with their operations. It has been observed that rigid procedures and centralisation do not produce the desired results. In fact adherence to procedures and workflows is widely considered to be a nuisance as the knowledge workers who follow those procedures are detached from the process of drawing up these procedures and from the running and "control" of these processes and tools.

1.2 Aims and objectives

The first aim of the project is to investigate and formulate potential solutions to the problems detailed in the previous pages, namely: Difficulty in integrating heterogeneous enterprise data models and processes between different enterprises rapidly and consistently whilst taking into account different levels of detail of the data models, impossibility of arranging a VE scheme where each participating enterprise keeps full control of its own Intellectual Properties whilst participating and sharing knowledge fully. A lack of knowledge flow and reuse between the early and downstream stages of

the design-manufacture process. And finally, there are no usable systems available that give small enterprises a realistic option of applying advanced PLM and Knowledge management techniques in a VE setting (thus becoming, effectively, the weakest links in the integrated collaborative project).

The second aim of the project is to illustrate the problems, and demonstrate a functional solution. Two engineering companies were tested. These were automotive and discrete machining companies and the software and working methodology developed was tested in integrating their product realisation cycles. The integrations were with their supply chains and customers which constitute the VE in these instances.

Main problems to be tackled by this research are the following:

- Difficulty in real-time collaboration between small enterprises in a one-off VE.
- Problems with reusing knowledge from existing projects, specifically for matching requirements to specifications and design (when re-use of existing processes and components is possible);
- Problems with sharing a common knowledge base definition for VEs.
- The ambiguity regarding intellectual property rights (IPR) management in Virtual Enterprises.

The project objectives are:

- ❧ Review of existing research on the topics of inter-enterprise collaboration, Virtual Enterprises and Knowledge Management and sharing in engineering;
- ❧ Determining the gap in current knowledge. Testing of existing technologies and the extent to which they meet aims and objectives of creating a heterogeneous VE knowledge management tool;
- ❧ Development of a two-tier architecture system and Peer to Peer network to enable real-time collaboration between small enterprises in one-off VEs whilst enabling the local management of each enterprises' IPR;
- ❧ Development of a common ontology for engineering VEs.
- ❧ Evaluation of the validity of the developed methodologies for meeting the aim and objective requirements;

The project will prove the following three methodologies:

- A generic super-ontology containing sub-ontologies that define the core engineering and project knowledge used and generated within an engineering enterprise.
- A two-tier architecture for enabling heterogeneous inter-enterprise collaboration
- A super-peer net architecture for engineering collaboration and knowledge management in a Virtual Enterprise setting.

This thesis will detail the construction of a format and medium for inter enterprise collaboration as well as taxonomy and format for the shared ontology and knowledge base.

1.3 Scope of Project

The Project scope is limited to the following. Review of existing literature and projects in academia, commercial development and open-source developments in the fields of inter-enterprise collaboration, knowledge management and standardised-taxonomies for engineering. Following on from the literature review, small pilot programs were implemented and tested with a subset of project data from two engineering companies using real-world data. Implementation of the software and methodology for inter-enterprise collaboration and knowledge management. Running tests on one of the engineering companies' data using a small subset of users (only three nodes to prove the validity of the research). The limitations of the implementation are:

- No full implementation of a holistic engineering taxonomy (this is beyond the scope of the research, and the subset of the ontology implemented suffices to demonstrate the basic viability of using a superset ontology based engineering knowledge management system).
- No full lifecycle testing of an engineering project (as these last beyond the period of the research project).

- Benchmarking of system against only one commercial PLM system as no other systems were available for testing. However, this will suffice as other systems are comparable in performance and features.
- No direct CAD-knowledge management integration, as this is beyond the scope of the project. The STEP standard taxonomy integration suffices to demonstrate that product data can be shared and manipulated on a heterogeneous peer-peer network regardless of the actual solid-modeller used by each participant.

1.4 Potential benefits to industry

The two companies were: ArvinMeritor a Tier-1 automotive supplier, with its customers being five large OEMs that dictate terms of work and standards to the Tier-1s (and each of the working standards of the OEMs are inevitably different to the other, to confuse matters for the Tier-1 further). In addition, the Tier-1 company uses 40 Tier-2 and Tier-3 companies to actually manufacture the components that go into their sub-system. The Tier-1 company's task is to understand the OEMs specifications, and translate that into a functional specification which is agreed upon with the OEM and then developed jointly with the 40 external enterprises with the oversight of the OEM. The company would run seven to eight such projects in parallel, many of which will 'share' key techniques and knowledge in order to reduce the repeating work.

The second company's task is to design and construct bridges of different spans using the same design elements. Here the ability to reuse existing design knowledge and to share it with contractors who are building the bridges and the customer who creates the specification would reduce the design-test-production cycle as well as ensure that the design meets the criteria that the customer specifies.

The benefits for the two specific companies, and industrial companies working with large supply chains or in a VE are:

- rapid appraisal of design ideas, their potential manufacturability by combining the knowledge bases of all participants in the VE from the earliest concept stage.

- cost-estimation for their customers of different configurations, from simple concept designs.
- Ability to share the companies' knowledge bases within a Virtual Enterprise setting, whilst maintaining the company's intellectual property rights and control over internal knowledge.
- Standardise the Inter-enterprise data and workflow models at three levels to enable truly seamless collaboration between disparate enterprises including the Data model taxonomy, Programming language level, Interchange format and communications.
- The system has to be able to reuse the company's existing base of knowledge and to push the manufacturing knowledge higher up into the design chain.

1.5 Research Methodologies

The subject of the research has a strong focus on Information Technology and, due to the rapid development of technologies and methodologies in this area, a pro-active and ongoing practice was adopted for the literature survey. Another factor that affects the research methodology within this sector is that a much wider sphere of researchers are participating in similar fields, and very often with very little awareness of applications, practices and methods in other disciplines. This is not in reference to academic researchers' knowledge of other academic works, but the wider gulf that exists between software development teams and academic researchers.

Research in different application focuses like the electronics industry, shipbuilding or the construction sector all have research in the field of collaborative product development and engineering knowledge management and coherent collaboration. However there is very little in the way of diffusion of ideas and knowledge between these different disciplines.

The experimental side of the research consisted of:

1. Functional specification for a tool to fulfil the requirements stated in the introduction ;
2. Literature survey of existing tools and techniques (see below for breakdown);

3. eXtreme Programming (XP) methodology to program the software application (www.rational.com);
4. Configuration of the application with data from a company for evaluation purposes; and
5. User based functional testing and analysis (quantitative or qualitative as obligated by the test in question).

1.5.1 Creating Functional specifications

Selection of the right tools for the experiment was key to the successful deployment of the systems. In order to test the theories, best-practice tools were selected based on their functionality and applicability for solving the problem at hand. The literature survey of commercial, research and open source applications and techniques enabled the researcher to narrow down the choices to the most appropriate ones, before creating the application, generating the product data, populating it with the project data and testing them against each other using identical hardware, with identical data and scenarios.

The choice of systems reflected a necessity to test different techniques against each within a constrained time-frame and a need for rapidly deployable systems as only one researcher had to undertake the entire process consisting of:

survey -> evaluation -> benchmarking -> selection -> programming -> testing -> data acquisition -> data population -> benchmarking of tested systems

Data acquisition used real-life test data from live engineering projects, and the data input was the same across all applications.

1.5.2 Method for compiling literature survey, and analysing existing work.

The wide subjects within Information Technology covered by the research include areas in:

- Knowledge management;
- Business to business connectivity;
- Query methods;
- Data-persistence methods ;

- Methodologies for inter-enterprise collaboration;
- different standards for product data models; and
- Methods and technologies for processing and transforming data into usable information.

Queries for academic journals and conference proceeding were using keywords, and also reverse look-up of references from journal and conference papers. The online services of Cranfield Library were used to extract the reference information and full papers acquired in a number of methods, including electronic journals, conference proceeding texts and Inter-Library Loans (see Table 2).

Search fields	Key search terms	Primitive query terms
Title	Product Data Management	Cad
Abstract	Collaborative Design	PDM
keywords	Concurrent Design	PLM
	Concept Design	Knowledge
	Knowledge management	KM
	Ontology	KBS
	Assembly Planning	CAPP
	Engineering Design	VE
	Design Knowledge	
	Virtual Enterprise	

Table 2 Query terms for literature review

The primitive queries naturally yielded thousands of results, but are required for the retrieval of articles that were not found by the more specific query terms or were not referenced in other articles.

Commercial vendors of software in this area also create “research” and beta- alpha- projects to research into new areas. These developments can be found by passing queries to internet search engines and looking up references from academic journals. Standards websites such as The World Wide Web Consortium www.w3c.org and

www.semanticweb.org have links to the vendors and open-source developers researching in this field.

Open source development can be found through links in the Open Source Developer Network, which acts as a portal for the software developments under the General Public Licence. In addition free text queries in search engines are used to retrieve any open source projects not listed on the semi-official portals.

Management of reference and bibliographical resources is handled by Biblioscape Reference Information Manager. There were 520 references used within the project (NOTE: not all references were quoted in this thesis). A breakdown of the reference sources is in Table 3, including the number of articles from each source.

Journal Sources	
Computers In Industry	21
Computer Aided Design	21
Design Studies	16
Journal of Materials Processing Technology	12
Robotics and Computer Integrated Manufacturing	10
Journal of Engineering Manufacture	8
Computers and Industrial Engineering	8
Automation in Construction	6
Engineering Applications of Artificial Intelligence	6
Expert Systems with Applications	5
AI-EDAM	4
Applied Intelligence	4
International Journal of Computer Integrated Manufacture	4
Computer Integrated Manufacturing Systems	4
Computers & Structures	3
Artificial Intelligence	2
Concurrent Engineering Research and Applications	2
Data & knowledge Engineering	2

Decision Support Systems	2
International Journal of Advanced Manufacturing Technology	2
International Journal of Production Research	2
Journal of Engineering and Technology Management	2
Other Journal papers	32

Table 3 A breakdown of references from different journal Sources

Total 178 Journal Articles reviewed in addition to 40 conference papers. This from a total of over 500 articles surveyed. Of these 100 were referenced in this thesis.

1.5.3 eXtreme Programming methodology for application development

eXtreme Programming (XP) is a methodology devised in the 1990s to aid in the rapid development of computer software. It is a widely used methodology for rapidly developing usable applications using standard 3GL (3rd generation programming languages). Some of the advantages of XP (2 brains on a screen) could not be realised due to the author working alone on the project, however the Rapid Application Development (RAD) oriented development and the principle of making small changes to the system often, instead of making major changes to the system at long intervals was used to good effect to develop the system and ontologies described in this report with no operational bugs, and with only one developer in a constrained period of time. More information can be obtained from www.rational.com

1.5.4 Methodology for configuring systems with company data

This component of the research also fell under the XP methodology element, and company data was fed in small chunks and verified before continuing to load the data incrementally. Using this methodology the entire ArvinMeritor project data was fed into three separate systems for evaluation by one developer in a constrained period of time.

In order to aid the development of the system and deployment using XP methodology, systems that can effectively utilise the XP methodology were chosen for the development of the system. The basic requirements are:

- Ability to modify small parts of code and re-evaluate result rapidly;
- Test system with incomplete data;

- Modify and retest code in a live environment; and
- Multi-user collaborative capability (although this was not a requirement in this instance as there was only one developer).

1.5.5 Methodology for user based functional testing

The results for analysis are primarily qualitative in nature. The reason for having qualitative rather than quantitative results is due to the fact that the calculation of efficiency gains made through the use of the proposed methods are a combination of quantifiable variables like 'time to set up virtual enterprise' and other issues like the overall usability of the system which can be qualitatively analysed through user feedback.

Functional differences can be illustrated using matrices; however this project refrained from adding arbitrary weighing to the different functions in order to not create a specific, thus biased, quasi-quantitative analysis. Instead the possibility was left open for potential benefactors of this research to weigh the features and advantages / disadvantages of the various tools according to their unique requirements.

The benchmarks used for testing off the systems against each other were focused on VE deployment in a de-centralised manner with a heterogeneous ontology and knowledge infrastructure. Due to this, factors such as cost, speed of deployment, and ease of adding third-parties to a project were deemed to be more important factors than whatever commercial, but irrelevant, value added functions were included in a particular application. Only if a value added function within a particular system had a direct consequence on its usability within a collaborative engineering framework was it included. Therefore tight integration with bespoke CAD CAM systems was not a favourable factor for any system (unless integration was to an open-standard based model). More details about the actual testing can be found in the relevant sections of the thesis.

1.6 Structure of the thesis

The first chapter of the text introduces the problems to be researched, aims and objectives of the work, scope of the tasks to be completed and tested, and research

methodologies that were used for carrying out the literature survey, software evaluations, programming and analysis of the results. This is followed by an extensive literature review of existing research carried out on the topics researched, the literature review concludes with a statement of the gaps in current research identified from the existing research carried out (Chapter 2). Chapter 3 is a review of implementations, both academic and commercial of VE, PLM and KM systems of relevance. The main body of the research work, development and testing of the new methodology and its associated technologies are detailed in Chapters 3 and 4. Chapters 5 and 6 detail the implementation and testing of the applications developed (both the Knowledge Management and Inter-enterprise collaboration architecture). Chapter 7 compares the actual results of the work with the stated aims and objectives of the research and summarises the contributions to knowledge and understanding achieved by the research work and lays out a path for further work on the topic. This is followed by the list of references. Finally the appendices, includes the full workflows, and source code of the tools developed for demonstrating the functions laid out by the research aims.

2 Academic literature survey

This literature survey examined academic journals and conference publications. The literature survey scans through the research in the main spheres that concern this research including Virtual Enterprises, inter-enterprise collaboration, peer-to-peer networks, engineering ontologies and engineering knowledge reuse.

2.1 Virtual Enterprises

A Virtual Enterprise (VE) is the idea of multiple independent enterprises forming a temporary alliance for the duration of one project. VEs have existed in practice (if not in name) since the dawn of civilisation, and that form of cooperation is now being explicitly defined and its scope and functional basis explained in the articles below. The VE is used as the core inter-enterprise collaboration mode in this research, and the aim of the research is to find ways of actually implementing an electronic system that enables enterprises to function as part of one or many VEs in a heterogeneous system.

Decreasing innovation cycles, changing market situations as well as growing specialisation in individual market segments demand new ways of economic thinking, increasingly forcing enterprises into co-operations, sometimes even with direct competitors. These co-operations enable enterprises to share skills, costs, access to one another's markets and resources and, at the same time, decrease the risk of investments. Presented, discussed and designated as the corporate and cooperation model of the future by (Byrne 1993) as the 'virtual corporation'. Of course at this juncture the 'Virtual Enterprise' business model was not yet fully evolved.

Byrne's initial ideas were further developed into the ideas presented by (Barnett et al 1994) in 'an architecture for the virtual enterprise' which laid out the logical sequences for the establishment, operation and dissolution of a VE. In this paper, the authors state that the VE is a set of separate organisations connected for a brief period of time by a set of Objectives, policies and procedures that define the business processes for interaction in the project and these processes are owned by the 'collective' VE and not any one enterprise. In addition the design, marketing, manufacturing and distribution

side of the business processes are owned by individual enterprises within the VE but utilised in a collective manner within the scope of the VE.

Further elaboration by (Hardwick 1996) in the National Industrial Information Infrastructure Protocols (NIIP) consortium where the authoritative definition for a VE was formalised. Here the VE was defined as 'A temporary consortium of independent member companies that come together to quickly exploit fast-changing worldwide product manufacturing opportunities'. Virtual enterprises assemble themselves based on cost effectiveness and product uniqueness regardless of organization size, geographic location, computing environments, technologies deployed, or processes implemented. Virtual enterprise companies share costs, skills, and core competencies, which collectively enable them to access global markets with world-class solutions that cannot be provided individually (NIIP 1998, <http://www.niip.org/>). The (NIIP) Consortium was formed under a cooperative development agreement with the U.S. government. Layering on the International Standard for the Exchange of Product Data (STEP), the NIIP focuses on enabling the effective interoperation of manufacturers and suppliers who utilize a wide range of computer systems, operating environments, and business processes. In NIIP specifications, the information protocols are explicitly divided into three layers to meet the requirement of an information system for virtual enterprises (Hardwick et al. 1997). The bottom layer consists of communication protocols that dealt with the communication and data stream between the clients and servers. The second layer, organization protocols, lies upon the communication protocols.

The organization protocols organize the data, which are processed by communication protocols, into models and then define the relationships of the data. The top level of the NIIP data protocols is the management protocols, which present the operations that can be applied to the product data. Based on the protocols, the NIIP Consortium built the NIIP reference model (NIIP 1998). This model has been tested under many scenarios by member companies. The NIIP addresses these challenges of STEP integration (fine-granularity of STEP data models and the integration of STEP information) by defining STEP data objects. These data objects are analogous to the application objects identified in the application reference model (ARM) of each STEP model.

While the NIIP details an extensive work and knowledge management view of the reference architecture (NIIP, 1998), the main difficulty is in the definition and management of this knowledge. In addition, agile manufacturing enterprises might be very short-lived, and there is a need for expedient methods for re-creating the access-control and other administrative level functions for future VEs. This project was the first comprehensive implementation of a VE in terms of an actual application. However, this implementation has one major flaw, the reliance on a centralised server for holding and managing the shared knowledge base.

Building on the NIIP research, (Zhou and Nagi 2002) stated that the shortfall of the, then, current state of the art in VE implementation was the lack of easy to use applications that enable VEs to be created. He states that the essential components of the VE implementation architecture are the integration of the various components of VE systems including the 'data-model' and workflows that link the business processes of the VE. Zhou's proposed solution involved the extension of the STEP standard using the EXPRESS modelling language to codify the ontology needed for inter-enterprise integration and for the knowledge not already defined in the STEP Application Protocols. The authors suggested the use of Common Object Request Broker Architecture (CORBA) and its interface Definition Language (IDL) to achieve the data-level inter-enterprise integration in the VE. This proposal is similar in nature to the large number of 'intelligent agent' type systems that rely on CORBA to bridge the gap between each enterprise. The authors already see clearly the shortcomings of implementing VEs (lack of easy to use tools, standardised ontologies for VEs and difficulty in integrating multiple systems from heterogeneous enterprises into a single 'VE' using normal B2B integration techniques). However the solution proposed to this problem does not fulfil the basic requirements of de-centralisation as CORBA still requires individual inter-enterprise integrations on a one-one basis. However the authors have correctly identified the basic problems of VE implementations.

(Wang Lihui et al 2003) wrote an extensive survey of collaborative design systems, and highlighted eight areas as having scope for development including: System architecture

for web-based collaborative design, Collaborative Conceptual Design Modelling and Data Sharing, Conceptual Design Selection, Knowledge Management in Collaborative Environments. The survey reviewed over 80 research projects around the world and offers a comprehensive glimpse into the current state of the art in research. The authors' conclusions on the topic shows the limitation of thinking within a 'web centric' view of collaboration i.e. intrinsically a client-server centralised model of collaboration, however the authors also, rightly, highlight the problems of shareable ontologies and useful systems for knowledge management that are not too narrow in scope to be useful within a generalised engineering environment.

(Beckett 2003) discussed the topic of communication and understanding in VEs between unfamiliar participants. He discussed the best tools and standards to apply in VE settings for collaboration and knowledge management and has an overview of various methodologies and applications. He correctly notes that a true VE should be considered practicable for one-off collaboration projects (and thus implicitly a longer term relationship between enterprises can be termed as a supply chain or extended enterprise). The research also shows that a common ontology that normalises the underlying nomenclature into a commonly agreed upon standard across the industry to enable ad-hoc relationships to be rapidly established. The research however does not provide a practical means for realising this.

(Camarinha-Matos 2003) on a similar note, reviewed current trends in VE developments, and conclude that there is a need to develop a generalised framework for VEs, to enable harmonisation, international collaboration and rapid deployment, once again without offering practical means or architecture for implementing that requirement. The research does however show that there is a general trend among disparate research groups of the basic shortcomings underlying the lack of implementable true VEs in a de-centralised engineering environment.

Towards a practical implementation of a VE system (Kim 2001) developed a 'Distributed open-intelligent PDM' system, which adopts ISO standard STEP, whilst offering standard PDM functions. A dynamic and flexible workflow model is

implemented. This could greatly enhance the flexibility of the system. However the system still relies on a single central node or controller, and does not implement a data-model usable by all enterprises, thereby maintaining all of the shortcomings of traditional PLM and other centralised systems.

(Zha and Dn 2001) proposed a STEP based application to manage the entire product lifecycle. The information that is not already defined in STEP is modelled in EXPRESS. The system is focused primarily on assembly mating features and does not consider the machining requirements of each component. In addition the systems' implementation on a centralised system with no illustrated method for inter-enterprise integration rules it out for a pure VE use. It is interesting to note however the authors' use of a standard ontology and building up the additional requirements of a collaborating enterprise by utilising an open modelling standard.

On the important question of the inherent de-centralisation of a VE, (Vasara et al 2003) proposes the ARACHNE, the adaptive network strategy to enable integration between 87 enterprises; the authors highlight the benefits of peer to peer networks to achieve synergy between collaborators. Their developed methodology called RosettaStone enabled many-to-many integration between enterprises using a three-tier architecture.

In summary, the above articles, and the studies by (Barnett 1994) and (Byrne 1993) that laid the foundations for the 'Virtual Enterprise', highlighted some of the possible schemas for implementing VEs in an engineering setting. The relationship between enterprises as defined by a VE is not actually implemented in the systems described above. It is interesting to note that individual components of a true VE system are researched in the above examples (some focusing on the inter-enterprise collaboration, some on de-centralisation, some on creating a common ontology) however none of the research has currently proposed a holistic overview of the VE problem (i.e. tackling the de-centralisation, collaboration and ontology aspects as one whole problem), in addition, none of the projects provide a clear and usable framework or implementation for a true de-centralised VE. This leaves a clear gap open for a true VE implementation that includes provision for separate enterprises to work alone, collaborate on an 'equal'

basis in a de-centralised manner without the expense and time for implementing B2B integration and have a common nomenclature ontology to enable seamless integration between enterprises' data models and enable the reuse of legacy 'knowledge' in the system through a unified query mechanism and standardised structuring of the ontology.

2.2 Inter-Enterprise collaboration

Following on from the review of holistic Virtual Enterprise, some of the research conducted on components that constitute the VE are investigated. Inter-enterprise integration is not a field with clearly defined boundaries for its scope and includes such topics as Business to Business (B2B) integration, Virtual Reality Markup Language and other 'virtual world' technologies. At a basic level it is a tool to enable people, machines and processes from different enterprises to collaborate together in a coherent and understandable manner.

The inter-enterprise collaboration of relevance to this research, which is the technology and techniques to enable the creation, running and management of the authentication and data-model integration of rapidly constructed 'virtual enterprises' is reviewed. A number of related techniques and technologies including B2B and web services implemented for enabling inter-enterprise collaboration are reviewed and the gap in present research identified for enabling such inter enterprise integration in a disparate, de-centralised VE are highlighted.

Inter-Enterprise integration, by its very nature excludes Business-to-Customer integration, and does not differentiate between one-off partnerships and longer term 'extended enterprise' collaboration between enterprises, this is an important distinction as many so-called B2B integration schemas are only cost and time efficient when deployed within the context of a longer term supply chain relationship as opposed to a VE. From the paper below it can be seen that much of the focus of research in inter-enterprise collaboration has been implicitly within the context of a large OEM 'controlling' smaller suppliers in an extended-enterprise or supply chain scenario.

(Trienekens 2001) in the paper on 'inter-enterprise relationships' laid out the various components of the high-level relationships between business process integration (between enterprises), organisational collaboration and the strategy of business partnerships. The paper focused on the business process integration, and the distinction between in-company processes and the business processes that are conducted on an intra-enterprise basis. The inter-enterprise relationships and business process boundaries developed by Trienekens are applicable within the scope of industries with well defined business processes that are similar from enterprise to enterprise. The authors do not describe a particular technology or methodology for creating these intra-enterprise business processes or any means of creating sharable knowledge bases within the collaborative scheme. The ideas proposed in this paper do in theory offer a solution to the problem of inter-enterprise relationships and business processes with separate processes for internal business processes. However, due to the complexity of the resulting system implementing such a solution, it would only be practical within the context of a large enterprise having the scope to deploy separate business processes and procedures to manage its relationship with lower tier suppliers, as opposed to a situation where a large number of small enterprises wish to create a simple environment for inter-enterprise relationships within the scope of a VE.

A number of approaches for the technical implementation of inter-enterprise collaboration have been the focus of researcher's activities for the past 10 years. Initially agent based systems were developed. These were later supplanted by workflow based tools to achieve inter-enterprise collaboration and inter-enterprise business processes. The workflow developments were themselves then supplanted by the development of B2B protocols and applications which were designed from the outset for bridging the process and knowledge gap between enterprises as opposed to the generic workflows which are more suitable for the definition of complex and relatively static internal company processes. Web Services created another standardised dimension to enable the rapid construction of portals from multiple business applications for inter-enterprise collaboration. The research work shown below illustrates a good cross-section of research in the areas mentioned above.

(Ulueru et al 2000) proposed a multi-agent based system called Multi-resolution-collaborative-architecture for inter-enterprise collaboration. The system proposed is essentially a three-tier client server application that uses agents to discover and intelligently ascertain the correct interfaces between business applications (i.e. a form of automated intelligent B2B mapping) in the form of 'middleware'. The problem with this type of system is that its scope is limited by the intelligence of the agents within its 'vertical hierarchy' and due to the architectures' inherent centralised nature is of limited value in a de-centralised VE where the usability of automated agents to create fast B2B links may be of value. This research illustrates clearly the limitations of agent based systems within the limitations of the current state of AI agents.

In 'Anatomy of enterprise collaborations' (Jagdev 2001) outlines the necessity for clearly defining the different types of possible inter-enterprise collaborations, or bilateral enterprise collaborations, that occur between enterprises including supply chains, extended enterprises and VEs. The paper highlights the essential differences between the various types of inter-enterprise collaborations, adding no new definitions per se, but showing the essentially short term and egalitarian nature of a VE vis-à-vis the centrally controlled and hierarchical nature of extended enterprises and supply chain relationships.

In summary the different researchers within inter-enterprise collaboration highlight a common issue within inter enterprise collaboration, that is, the lack of clear boundaries between the different types of relationships that enterprises establish between each other. Another common theme running within the scope of current research are the clarification of rules and protocols for sharing information between enterprises. The research also identifies the de-centralised VE as the ideal inter-enterprise relationship for companies to collaborate in projects where there is no one 'controlling' enterprise. Some of the work tackled the problem of establishing the data-bridges between enterprises, and the usual middleware based solutions were proposed, notwithstanding the problems of such client-server solutions in establishing B2B links without a process involving programmers, and a considerable expense in terms of time and money that

makes it more suitable for medium-long term inter-enterprise relationships rather than VEs that cannot justify such an investment due to their short-term nature.

2.3 Peer to Peer networks

The implementation of Virtual Enterprise requires the creation of an information technology infrastructure that enables disparate enterprises to share the knowledge stored in their respective applications. A key aspect of this relationship is the topology of the data network, and there are several topologies used in modern inter-enterprise integrations, including client server topologies, one-to-one and one-to-many business to business integrations of client server systems as well as the application server and trusted third party application server models of inter-enterprise collaboration. Peer to peer (P2P) networks are a newer topology created for intrinsically de-centralised network topologies where no one node in the network is a central node. All nodes being equal and connected to all other nodes in a heterogeneous topology, makes this topology the closest analogue to the operational layout, hierarchy and distribution of a typical VE.

In 'harnessing the power of disruptive technologies' (Oram 2001) offers an insight into the potential applications of peer-to-peer systems as was seen in 2000. The authors rely on the views of pioneers of the peer-to-peer era, and among the applications of peer to peer techniques include a discussion of the potential for their application in de-centralised business collaboration networks. The paper makes a multitude of claims with regards to the advantages of peer-peer nets over client-server systems including empowerment of all the 'users' within the system, ability to share information without losing control over the original information, and the ability to manage collaborative sessions without needing a central node or controller. Some of the areas highlight the potential applications for a VE including the ability to form closed networks of peers (which is functionally analogous to what a VE requires) and the ability to mark-up and amend information from multiple sources in a transparent manner. The paper, by itself does not focus on VE's per se, but gives an introduction to the functions and general advantages of P2P systems.

The above paper highlights one of the potential problems of de-centralised systems vis-à-vis centralised applications, and that is the issue of search and query across a heterogeneous network with no central mechanism for indexing or searching the available resources. (Lv et al 2002) discuss this particular issue in 'Search and replication in unstructured peer-to-peer networks' the algorithm used by some traditional de-centralised applications like GNUTELLA www.gnutella.org to overcome the lack of a centralised server to store information on the available users, nodes and data on the peer-to-peer network. The authors illustrate the inefficiency of the flood based query mechanism when large numbers of users process queries simultaneously. This query process blindly sends out queries to all possible nodes in a network (whether internal or the wider internet), and naturally when many nodes are sending out queries it overwhelms each node in the peer-to-peer net with queries from all other peers using the same protocol (which can be in the millions). The authors correctly highlight the problems with pure P2P networks' query mechanism, however the solution they propose of 'multiple random walks' based queries, whilst more efficient than traditional flood queries, still cannot compete on speed and efficiency with indexed client-server systems.

Other solutions proposed for the problem of queries in P2P systems included (Harren et al 2002) who proposed using a distributed hash table to speed up queries and reduce the network overheads generated by each query. This method however has the dual problem of being able to match exact terms only and requires the construction of very complex query mechanism to create a usable system. However even using these complex shared hash tables and queries (which are by default impractical for use in a 'simple' application), the performance of the queries, and the ease of adding additional terms, in what is a highly dynamic system, does not meet the requirements of serious VEs with very large volumes of knowledge in their knowledge bases.

As can be seen from the focus of the research above, the primary issue with uptake of peer to peer systems is the efficiency of the query mechanism available in a de-centralised system, further work by (Yang 2002) proposes yet another potential solution for solving this problem. The authors implement their new query mechanism (similar to

Lv' solution) over the GNUTELLA protocol and claim to maintain the same query speed whilst using 20% of the resources. The problem with all these proposals is that they are benchmarked against the base performance of P2P systems, whereas within the sphere of a VE, the performance comparison is with client-server systems with their cached and indexed knowledge bases.

(Cohen 2002) proposed replicating all the data in the peer network (using a number of replication strategies) in order to speed up the speed of the query mechanism. This in effect created a mirror of all the resources on the peer network on a centralised node, and the queries run against this node result in a performance similar to a client-server application. This is a novel approach, and at least acknowledges the fact that P2P query performance must compete on speed and resource use with the efficiency of client-server systems.

Other issues of peer-to-peer systems include the problem of managing access control in an environment where, in theory, anyone can pretend to be anyone and there is no centralised mechanism for controlling and managing access control. (Damiani et al 2002) discussed this issue at depth in 'a reputation based approach for choosing reliable resources in peer-peer networks'. The authors highlight the historical reality that P2P systems were initially created for sharing files of dubious legality between private individuals, and thus initial implementations of P2P systems were designed from the outset to maintain the anonymity of its users. However within the setting of business use-cases anonymity of enterprises on a peer network is not desirable in the least. For this end, the authors proposed a reliability assessment scheme for each node in a peer-network where reliability points are awarded for resources by other peers. Of course this type of system is only of use for file sharing type portals, but the authors do realise the need for reliability assessment in peer networks as well as proposing the use of peer-reviews to assess the reliability of other peers.

Scalability of peer networks (not simply the query of resources and peers) as well as the dynamic nature of a peer-networks' topology was assessed by (Jovanovic et al 2001). The researchers constructed a crawler to assess the real-world use of pure peer networks

and observed that in practice peer-to-peer nets created individual “small-worlds” of peer networks limiting the overall scope of a, theoretically, global peer network by the limited hops performed by most peer-to-peer queries. According to the authors this limits the scope for truly global level queries by peers in a network, as in practice network timeouts limit their search space to the networks of geographical proximity. This is potentially a problem for some types of widely distributed VEs, and is a feature of pure peer-to-peer nets that cannot (as of today) be overcome within the limitations of working in a pure P2P environment.

Further to the problems highlighted by (Jovanovic et al 2001), (Lu and Callan 2003) faced a problem with the search and distribution of electronic documents in a global peer-network, where far-removed peers were not able to find peers and queries could not be processed due to the geographical distance between peers, creating dis-connected peer ‘islands’ on the internet, limited, usually by the interconnecting networks between countries and regions of the world. To overcome the lack of ‘global’ usability of pure P2P systems, the authors proposed a hybrid peer-peer network using so-called ‘regional-directories’ on ‘leaf nodes’ in the local peer net to hold information on the peer-nodes in their region as well as indexed search terms for the electronic documents contained in the peer-nets of the regions’ peer nodes. The authors proposed the use of content-based-retrieval mechanisms on the ‘leaf nodes’ (note that a leaf node here is a functional equivalent to a super-peer) to decide what documents the leaf node should index in its directory. The authors, here have created a novel solution to the scalability problem of peer-to-peer networks, without actually fully detailing the partially ‘super-peer’ net described in their paper.

Building on to the above research, (Yang and Garcia-Molina 2003) described the ‘design of a super-peer net’ fully clarifying the working methods, topology and advantages of super-peer networks vis-à-vis both client-server systems and pure P2P networks.

The research above has illustrated the various topologies and mechanisms for establishing and running P2P nets. However, the above systems are utilised in the test examples for sharing flat-files between peers. In the case of a complex inter-enterprise

collaboration, simple flat files (i.e. excel sheets, text files, word documents) need to be augmented by structured databases. Towards this end (Gribble et al 2001) discussed the potential for creating databases distributed across peers in “what can databases do for Peer-to-Peer”. The authors highlight problems associated with operating a database with its rows spread, effectively over a number of disconnected nodes (peers). The problem is that effectively indexing and managing the persistence at a performance comparable to a large single dataset becomes problematic due to the dynamic nature of the data held on each database. Effectively ascertaining the ‘correct’ version of a row of data becomes almost impossible without real time connectivity and very high speed processing. Current P2P systems focus strictly on handling semantics-free, large-granularity requests for objects by identifier (typically a name), which both limits their utility and restricts the techniques that might be employed to distribute the data. These current content sharing systems are largely limited to applications in which objects are large, opaque, and atomic, and whose content is well-described by their name; The authors outline the differences between the problems of queries in a traditional distributed database and a P2P based database, these include the issues of caching, indexing and synchronization that can be managed in a homogenous distributed database, but not in fundamentally ‘independent’ peers of a P2P network. The authors then discuss the problems unique to a P2P based database including the following factors: scope of decision making (the scope of the query executed by one peer in the P2P net, should it search through all databases on all peers, or restrict the search to a certain set of peers); extent of knowledge sharing (who can see which datasets); heterogeneity of information sources (the ontology or schema of the databases that the query is executed against can vary if the P2P system’s ontology is not standardized); dynamicity of participants (whether all relevant nodes are online at the time of the query), and if datasets have changed whilst a node is offline); data granularity (i.e. the hierarchy of the data accessible by peers); and finally freshness and update consistency (the methods in which changes to dynamic data are propagated throughout the P2P net). The authors correctly identify many of the issues related to having distributed knowledge bases and databases over a P2P net, with some caveats for the fact that super-peer nets and harmonized data models and access control mechanisms were not suggested.

In summary, this section highlighted some of the relevant research carried out in the areas of P2P networks including the topologies, access control mechanisms, peer net performance and issues relating to the creation and sharing of complex knowledge bases over peer nets. Fundamentally, much of the research is overlapping, with different strands of work effectively providing solutions to problems highlighted by other researchers, such as solutions for the scalability and persistence of peer-nets using super-peers, and other groups highlighting potential uses of databases for complex distributed applications but without highlighting the possibility of using super-peers to overcome the stated problems with indexing and caching of data on peer-nets with intermittent connections.

2.4 Engineering ontologies

Ontologies, in the context of this research, are the definition of the terminology and relations between terms of the scope of the knowledge contained within the enterprise engaged in an engineering activity.

(Gruber 1995) and (Cutkosky et al 1993) introduced the basic principles for the construction of ontologies “as a way of specifying content specific agreements for the sharing and reuse of knowledge among software entities”. In other words, the use of the word ontology to describe the schema or complex data-model that acts as the skeleton for knowledge bases. Throughout this thesis “ontology” will be referred to in this context.

The topic of data models vs ontologies was discussed at length by (Spyns et al 2002). Here it is stated that one of the fundamental differences between a data model and ontology are the system independent nature of ontologies in comparison with the inherently system (and in many instances) database specificity of a data-model. This may seem like a trivial difference in light of the fact that both an ontology and a data model are utilised using computer software. However the inherently heterogeneous ontology opens up the possibility for ease of implementation in heterogeneous environments as may be found in a typical VE, as opposed to system and database

specific data models. Other differences include the fact that an ontology contains the semantically complete definition of 'knowledge base' and is inherently object-oriented as opposed to relational in nature allowing it to describe knowledge without the complexity of defining multiple tables for one-many and one-one relationships as would be necessary in a typical relational data model.

Building on a discussion on the usefulness of ontologies for knowledge sharing and reuse (Borst et al 1997) highlights (without the implementation of an actual example) the possibility of using ontologies to define the core of a complex knowledge base structure as used in industrial enterprises. The example shown is constructed with numerical simulation of physical systems such as the mechatronics and thermodynamics aspects of engineering design. The most interesting result, from the perspective of this thesis, is the illustration of the possibility for constructing loosely linked generic ontologies that can be used as "generic building blocks in ontology construction".

Further to the issues of interoperability between ontological applications, (Pouchard et al 2000) describe in detail the construction of an ontology "Process Specification Language" that aims to bridge one of the problems that collaborative engineering enterprises face in integrating their individual processes within a VE. However the PSL is an abstract language, and thus faces the perennial problem of not being able to map automatically in all scenarios between the manufacturing processes defined in different enterprises. The approach (using KIF as its language), goes some way to solving the problems of interoperability, but due to the need to maintain a certain level of abstraction above the individual manufacturing processes means that actual interoperability between enterprises will be problematic without manual intervention.

Developing the topic of ontology sharing in heterogeneous environments, the development of the "semantic web" needs new methods of ontology construction both in terms of software tools and ontology construction methods. (Breitman et al 2003) proposed the idea that ontology construction be in a machine processable format such as (in this instance DAML-OIL) www.w3c.org with the principle that only by formalising an ontology in such a format that can be tested and benchmarked can actual working

solutions for ontologies be constructed. Of course, the DAML-OIL used in the above work has now been superseded by the Web Ontology Language (OWL) www.w3c.org.

In summary, the ontology construction methods and ontologies described above illustrate the fact that often researchers have concentrated on formulating the methods of constructing ontologies, and also the research highlights the dilemma that researchers constructing application specific ontologies face in deciding the level of detail or abstraction that an ontology should have, and the effect that this decision has on the usability of the ontology (for more abstract ontology definitions) and interoperability (for complete subject ontology specifications). Some of the more useful notes that can be used in this research are the use of executable applications (programs that can run in real time whilst modelling the ontology) for modelling the ontology, and the possibility for creating large feature complete ontologies and linking them together like building blocks (in a graphical user interface environment) depending on the application needed at hand.

2.5 Summary of academic literature and research gap

The topics of Virtual Enterprises, Inter-Enterprise Collaboration, Peer to Peer networks and Engineering ontologies as summarised in the selected papers reviewed above are a set of inter-related topics of research and application aimed directly at helping people, organisations and software agents to communicate in a coherent manner. That is the core idea behind all the research, and the many approaches and methods at achieving this aim, as outlined above, illustrate the dilemma of researchers faced with an almost limitless set of possible solutions for their research problems, but not being able to achieve their aims in the real world due to the lack of ubiquity of any one of the methods or solutions that have been made to date.

That point illustrates the 'gap' in current thinking on the above subjects. All of the researchers provide individual clues and solutions to aspects of the problem, but none of the researchers set out to solve the problem of actually making the solution for a workable, shared, interoperable knowledge base running a distributed virtual enterprise work. This links in with the aims and objectives set out in (1.2) including the creation of

a technical solution to the VE application topology issue for the de-centralised VEs. Solving the issues raised with knowledge sharing between enterprises using a common ontology for engineering VEs.

The next chapter, illustrates some of the research applications constructed towards fulfilling the needs of VEs by researchers, with comments as to their usability within the scenario of a heterogeneous VEs.

3 Technology review

Traditionally the academic researchers have been ahead of vendors in terms of developing new methodologies and technologies, and these tended to trickle down to the commercial users via the software vendors.

But the rapid advancement of software technology and communications tools in the boom years of the 90s meant a massive injection of research funding within commercial enterprises which operate in a fiercely competitive environment and rely on the use of large teams to develop their tools. In spite of the added workload of these commercial vendors which are application oriented in nature, they still have at their disposal a greater capacity of brain hours for solving problems and coming up with research ideas both from their Research and Development staff and as feedback from their large testing teams as well as customers. Whilst these commercial R&D houses ruled the roost until the late 90s, the bursting of the Internet bubble in 2000 and the subsequent downsizing within these enterprises enabled the Open-Source software community to make strong inroads on research and development activities (often through the open-sourcing of many closed research projects by the likes of IBM and other organisations with a large research element).

This chapter reviews the enabling technologies and some existing solution implementations for inter-enterprise collaboration, knowledge management and engineering design. The importance of this chapter is in supporting the clarification of the 'gap' in current developments and also support the construction of the solution outlined in the aims and objectives.

3.1 PDM/PLM related issues

Product Data Management and Product Lifecycle Management were developed as a result of the initial experience of Computer Aided Design and Manufacturing (CAD/CAM) systems in the 80s and 90s. The engineering companies that used CAD/CAM needed to find a way of implementing Continuous Acquisition and Lifecycle Support (CALS) methodology in their enterprises, and of integrating their

design and manufacturing system across a geographically dispersed enterprise. PDM and PLM were born of this requirement, and function effectively as shared storage for CAD data and associated documents. They implement workflows and engineering change management and version control in the newer PLM systems.

(Kim et al. 2001) developed a “Distributed open-intelligent PDM”. The philosophy behind this particular PDM implementation is the use of open-standard only tools and file formats. The PDM system contains all the software tools contained in other PDM applications such as Parametric Technology Corporation’s (PTC) Windchill, however there are a number of features currently unique to this implementation (at least it does not come as standard on other PDM applications). The use of intelligent agents to perform many of the PDM tasks and Knowledge Query Manipulation Language (KQML) for queries sets it apart from other PDM implementations. The PDM software implements a dynamic and flexible workflow model, as opposed to the rigid workflows seen in many commercial PDM applications; this greatly enhances the flexibility and therefore usability of the system in a real-world situation, where rigid workflows are considered to be a nuisance. In addition, the program is designed from the outset for the use of Standard for The Exchange of Product model data (STEP) based files to store and distribute product information (www.iso.ch) STEP Data Access Interface (SDAI) is used for content searching of Product data. However, no mention is made about which Application Protocol (AP) of STEP they are intending to use.

(Parametric Technology Corporation 2001) Dynamic Design Link is an add-on factor (module) for Windchill and contains several useful features for conceptual collaborative design. It is a tool to enable customer-focused mass customisation of engineering products and consists of an online non-geometric configuration portal to allow customers to specify their products and the way it functions. The use of intelligent parts from PRO/Engineer CAD/CAM system with fully parameterised models enables the construction of “virtual” products by the customer. The system also has a Knowledge Based System (KBS) for the entry and re-use of customer requirement data and conceptual design data. However it can be used only for configuration controlled products (where the working components can be taken from a resource of available

components and sub-systems) such as cars, trucks, computers, pre-fabricated buildings. It does not enable the concurrent development of one-of-a-kind engineering products. It can be a useful portal for the collection of customer requirements and the control of conceptual design knowledge and rules/constraints. Combined with a conceptual design tool it is a very useful tool for enabling the engineering company working in design-to-order environments to have the same sort of customer focus and perceived flexibility as configuration controlled designs and assemble to order products.

(Youchon, et al 2001) created a Unified Modelling Language (UML) mapping schema for STEP PDM (www.iso.ch). The aim of the tool is to integrate the different STEP APs (however not AP 224) within PDM environments. In order to ease the development of PDM systems a methodology using UML instead of EXPRESS schemas (EXPRESS being the language that ISO-10303 STEP is defined in) is being proposed. The UML mapping schema is similar to a UML class-diagram and is intended as the middle-ware to translate STEP model data directly into PDM structured data, negating the need for re-typing, and improving consistency between the STEP data files and the PDM system (i.e. easier synchronization). The example here is implemented using I-DEAS CAD software and the Metaphase PDM system. However the methodology is generic and can be applied to any Object Oriented (OO) based PDM system (such as the Java based Windchill) in alliance with CAD systems that can generate valid STEP files. This development greatly eases the process of integrating different CAD systems with different PDM systems and allows the sharing of data and generation of PDM product structures and projects without re-entering data. The only problem is the lack of integration of manufacturing data (AP224) as part of the schema. This oversight has not been delved into in this thesis, however and no mention of future developments in this area are cited. Also the author does not explain if the process of sharing the STEP data is uni-directional or bi-directional (i.e. CAD to PDM and PDM to CAD). If the latter ability exists then it would be a great tool (however the STEP editors would have to be tailor made for each PDM environment.) and allow different users to append data to STEP/PDM/project data without the need for a CAD package on their PC.

(CoCreate 2002) is an online collaboration tool with a difference, it allows the editing of solid models, in addition to all the other collaboration and mark-up features offered by mainstream tools. The tool has similarities to Alibre design (alibre 2002). However, unlike Alibre it does not aim to be a fully functional design environment. The program is very promising and with further development would certainly make the “next-generation” of design tools where the entire process of designing a product will be made using network enabled and network-centric tools, instead of standalone-workstations.

In summary, PDM and its successor PLM are the prime choice of larger enterprises in their aim of streamlining their development and production cycles. However the systems are inherently centralised and the PDM/PLM vendors have not agreed on a set standard to use for Data Interchange between different vendors’ systems (essential for inter-enterprise use). Also the prime problem of the systems’ complexity and lack of ‘object level’ ontologies limits its use to a document oriented system.

3.2 Conceptual Design Tools

Conceptual design, is a stage in product development where a rough requirement and functional specification is available, and from which multiple rough design ideas need to be generated, to aid the decision making process in selecting a particular design or configuration for further development. This is an essential stage of the design process and a bad decision at this stage has massive cost and time consequences upon the following processes (engineering design, prototyping, testing and manufacture).

(Rodgers et al 1998) described a new software module for conceptual design called CADET (Computer Aided Design Evaluation Tool). It is a knowledge-based system that uses natural text to define the product concept. The aim of this tool is not to perform the design functions for the engineers, instead it aims to document and extract as much knowledge as possible out of the design ideas generated by engineers and other disciplines in the project. In effect, it acts as an extension of human memory, instead of other KBS systems that aim to augment the designer's "intelligence". The use of “natural text” enabled the system to incorporate multi-disciplinary perspectives of the design ideas into the model (i.e. not only the perspective of the draughting staff). The

stated advantage of using natural language for defining the knowledge is that it is not possible to define an algorithm for creativity. This tool is therefore aimed mainly at "new" or "creative" design as opposed to "routine" or "variant" design. The software has been implemented in web-based collaborative mode, to allow for concurrent design approach in development, as well as the dissemination of existing knowledge. The approach taken by this software system is very simple and can very easily be used in the development of conceptual designs for different industries. However, it does not address the embodiment and design geometry/features/manufacturing properties that are an essential part of conceptualising complex engineering solutions.

(Regli et al 2000) created a prototype knowledge based Solid Model repository for engineers that can be accessed via HyperText Transfer Protocol (HTTP). In many ways the functionality of this application mirrors that of the Windchill PLM application (with the Visualisation module). However there are a number of features, which set this development apart from the rest of the PDM/PLM storage models. First of all, the use of intelligent Java based interface called Conceptual Understanding and Prototyping (CUP) allows the designer to see legacy product model data which may be related to the conceptual design at hand, this uses graph based (Model Dependency Graph) which is a Constructive Solid Geometry (CSG) tree that is exploited for similarity assessment using the researcher's algorithms. The CUP interface allows the designer to make basic geometries using primitives (for conceptual design purposes), it also allows the definition of functional, structural and behavioural properties of the product concepts. The concept ideas entered will allow the knowledge base to retrieve related works on a "case-based design" philosophy (i.e. re-use of existing components within a new design). The software is integrated with I-DEAS and Metaphase. The shortcomings of this software are the lack of any ability to create "new" designs; since it is aimed at the routine or case-based design paradigms. Also these shortcomings mean that the basic concept geometry generated cannot be analysed for manufacturability. The author states that Virtual Reality Markup Language (VRML) is used for the basic models, which are adequate from a surface modelling perspective but hold insufficient data for engineering applications.

(Impactxoft 2002) is a Web-based functional design tool that has feature sets which overlap between (alibre 2002) and (technosoft 2003). It consists of 3 primary modules including Functional Modelling and Functional Object Representation (an AI representation of each model/feature) which allows easy transfer of the behavioural properties and logic of each element instead of the logic. The design tool allows synchronous product design over the network using functional representations of the models. The database acts as a directory service instead of a central repository allowing increased flexibility of use, but also increased problems of backup. The models can be exported to STEP geometry files, although it is not stated if the program supports entry of further meta-data on top of the geometry, or how much of the design and behavioural logic is lost in the transfer.

(Alibre 2002) is a new way of developing products. All the design, planning and data entry is done via a web portal server. It uses STEP as the data storage and exchange medium and aims to fully integrate the CAD and PDM processes into a single seamless environment. It also accepts legacy CAD models from previous projects and clients/sub-contractors. It does use a client side application that has to be downloaded on first use, which is very easy to install and work with compared to traditional CAD and PDM environments. It has many similarities to (CoCreate 2002) OneSpace, which however does not provide the full-design functionality of Alibre.

(Fuyama et al. 1997) created a conceptual design tool for steel bridges called Building Engineering Reasoning Tool (BERT). The tool includes material fabrication and cost estimation functionality and was implemented in an Object Oriented (OO) programming environment (Macintosh Common Lisp). The different components of the steel structure are implemented as classes within the application and this application structure enables the program's functionality to be expanded more easily. However the use of Lisp is a disadvantage compared to more modern programming languages such as Java. The software can generate efficient designs and optimum member sizes as well as cost estimates based on material and fabrication details entered into the database. The software is relevant to our work primarily because of the involvement of Mabey&Johnson whose primary product is make-to-order steel bridges. Incorporation

of similar functionality would certainly be advantageous to Mabey&Johnson, especially since it allows their customers to create a concept design bridge on the company's PLM site and evaluate its potential benefits before speaking to a "real" person at the company. This however cannot jeopardise the generic nature of the project's proposed engineering solution.

(Hsu et al. 1998) introduce a tool to aid the concept design of mechanical assemblies using PRO/ENGINEER as the CAD design platform. The example uses chair designs, but can be applied, with sufficient, relevant domain knowledge to other problems. The inputs to the system are a description of the functional requirements (as state transition diagrams). The tool has five main sections: A library of assembly oriented components, Results of the Design for Assembly (DFA) analysis for each of the concepts, Graphical User Interface (GUI) for the user interaction, a search algorithm to test all feasible concepts to satisfy the functional requirements and an assembly "integration" module. This application uses a case-based design system, and the authors make mention of the CADET tool developed by Cambridge University, (Rodgers and Huxor 1998). The tool is designed to work on a standalone workstation, and due to its development within PRO/DEVELOP cannot work in a distributed collaborative approach. The same author also mentioned a software solution similar to AML/TIE to decompose concept designs into feature objects and automatically generate AP-224 or AP-203 based files/geometry for conceptual design (Song et al 2000).

(Dahmus, 2001) described a methodology for modular product architecture. Taking as a starting point techniques developed by VW group to standardise its products across different models and brand names by constructing a "modularity matrix" based on functional structures developed from the concepts of the products. Using the modularity matrix different products can be built up simply by using a diagrammatic (block diagrams) approach to the functional design. The researchers have not produced a computerised version of their matrix, however it is an interesting principle to use on a computerised concept design model and would be a novel method of reusing existing engineering models (i.e. to add functional data to each sub-system or component and assemble them with other components using a linked block diagram approach).

(Sharpe, 1995) at the Engineering Design Centre at Lancaster University developed a diagrammatic conceptual design application called SchemeBuilder, which can be adapted for different industries (primarily mechatronics). The Application itself is produced using the Visio and Visual Basic development environments and runs natively only on standalone Windows PCs. The tool has a number of shortcomings, which include the inability to map design functions and constraints onto conceptual geometry, and not having any potential for application to assembly or process planning.

A similar tool available commercially is (netViz 2005) which is a three-dimensional (3D) diagrammatic tool for conceptual design, in many ways similar to Visio from www.microsoft.com and the program made by (Sharpe 1995). However this is a much more complex application that can store different types of meta-data within its objects, link to many different databases for knowledge and has algorithms to control relationships between objects. The program runs in client-server mode over Transport Control Protocol/Internet Protocol (TCP/IP) and also allows the execution of links and external programs and the creation of workflows (functionality very similar to that offered by the workflow with commercial PLM). It can automatically generate 3D views of the two-dimensional (2D) diagram for presentation purposes, and these 3D views are also fully functional. It can link together individual components of an assembly, however it does not contain the assembly relationships and geometric tolerances needed for engineering. It can be a useful early conceptual tool however.

In summary, the tools highlighted above indicate that concept design tools are still aimed squarely at the design engineer to quickly and easily generate concept geometries and configurations. The tools are not a component of a holistic product lifecycle where a much wider perspective is taken into account at the concept stage, namely customer requirement, engineering limitations, manufacturing constraints and supply chain (or VE) capabilities.

3.3 Design for manufacture

The methodology and tools developed to overcome the problem of unrealisable designs is referred to as 'Design for Manufacture (DfM). DfM aims to overcome the problems by providing constraints mechanism at the concept and detail design stages to increase awareness at all levels of the product lifecycle of the abilities and limitations of the manufacturing processes available.

(Mukherjee and Liu 1997) presented a new paradigm for conceptual design for manufacture. Instead of using complete geometry (which is often unavailable at the concept stage) or the functional reasoning methods using natural language, as per (Rodgers and Huxor 1998), the developers set about making a concept design tool for manufacturing using "sketching abstraction". This paradigm uses well-defined geometric features for the functionally essential parts, and abstracting the rest of the geometry using a linkage system. Essentially this is very similar to Group Technology (GT) coding and feature classification. The authors claim that the sketching abstractions provide sufficient information to generate some rough process plans, the sketches can also be converted to solid models and used as the skeleton for the detailed solid modelling process (even though the scope of this particular exercise was for 2D sketches and metal stamping applications). The authors' claim that the sketching paradigm can be easily translated for 3D components and assemblies is unverifiable, and unfortunately no methodology for realising this potential is hinted at.

(Chui and Wright 1999) developed an interactive Java applet (Cybercut) that works over Transport Control Protocol/Internet Protocol (TCP/IP) connections. This was initially intended as a tool for students to learn about the interaction between the CAD/CAM and Computer Aided Process Planning (CAPP). However the functionality of the program was above expectations and was extended to become a simple distributed conceptual design and process-planning tool. It uses a similar principle to other tools described in this report in that there is a design-by-features tool for geometry (here it is described as a Destructive-solid-Geometry) where features are defined on a stock item. From this the machining rules interrogate the features to generate the toolpath. The program is intended as a rapid prototyping tool but the methods used are

clearly very similar to the principles used for conceptual design by manufacturing features. The designs generated are not intended to be stored permanently and the author does not mention file standards such as STEP or Virtual Reality Modelling Language (VRML). Furthermore the tool is not integrated as part of an overall product development system.

(Chen et al. 1998) have developed an Integrated Concurrent Engineering for Mechanical Parts (ICEDMP). The aim is to design the mechanical components whilst using Design for Manufacture (DfM) and Design for Assembly (DfA) criteria for the designs. The system contains a part library, a design guidelines checklist a part modeller linked to PRO/ENGINEER and a feature translator to extract manufacturing features. The decision making tool (rule based) and design critique system was developed using the CLIPS language for the definition of expert system rules. The whole system has a Graphical User Interface (GUI) with a number of buttons to access the different functions (and these are simply links to execute external programs, such as PRO/Engineer and CLIPS). The system is limited to rotational parts and requires the use of a fully featured modeller to generate the geometry for design, therefore it cannot be considered as a concept design tool. There is no attempt to make the software multi-user, or collaborative.

(Shehab and Abdalla 2001) have created a manufacturing cost estimation tool for the conceptual phase of the product design process. The main interest for our research is that the software tool they developed also includes manufacturing feature representation and process planning application as part of the program. The package uses a fuzzy logic knowledge representation for the cost analysis and CAPP functionality as well as a CAD tool for creating the geometry. There is a single "user-interface" which allows the designer access to all the functionality of the system. The full functionality of the program can: recommend assembly techniques (not a DFA system), select materials using the Cambridge Material Selector (Ashby 2004), manufacturing processes and estimate total costs. The geometry creation is in AutoCAD. The system cannot be used in a collaborative manner and, although the author claims to generate geometric models

with explicitly defined manufacturing features no mention is made of any AP-224 capability or any method of collaboration between partners in a VE setting.

(Delmia 2002) Produced commercial software for preliminary planning, target costing, process engineering enables an efficient product design and a graphically powerful visualization of parts, subassemblies, and products. On the basis of preliminary default values for variants, special equipment codes, and piece numbers, the technology requirements for later production will be determined. There are many other modules such as Envision/Assembly which is a DFA tool generating assembly sequences and presenting them in GANTT charts. DPM assembly planning and verification tool which uses CAD geometry to generate efficient plans, and includes a manufacturing database similar in concept to the LOCAM production planning system (Sharma et al 2004). This program offers features that are more advanced than FeatureCAM; however there are add-on packages to give similar functionality to FeatureCAM.

In summary, the tools reviewed implemented DfM as a purely engineering tool disjoined from the overall enterprise system. None of the tools attempt to integrate multiple heterogeneous enterprises together, and thus the system is limited to products that are developed totally in-house by a single company.

3.4 Web based design

This section gives an overview of research and development in the area of TCP/IP (networking over the internet) based tools to aid the process of product design. These tools are client-server applications, where data and logic is stored centrally, with the functions and information shared by all users. The system is intended to overcome the issue of collaborative product development in a distributed environment.

(Mishra et al 1997) developed a 3-tier client server computer aided design tool called CollabCAD. The program consists of two components, the client and the server. It uses Java Remote Method Invocation (RMI) for distributed objects and makes use of a number of open-source technologies for its implementation. These are: Java, Java3D (for the visualisation of the models), OpenCascade (the geometry kernel)

<http://www.opencascade.com>, OZONE (an object oriented database to hold the model data) <http://ozone-db.org> a freeware JAVA speech Application Programming Interface (API) for voice commands and recognition <http://www.jpetererson.com>, and a public-domain collaborative virtual workspace application. The application provides the ability for multiple clients to view and manipulate the same geometry over the internet simultaneously and also to generate Numerical Control Codes (NC) codes from the manufacturing module (although this is not automated). The software can save geometric files in STEP, although as of yet these are not stored as manufacturing features. The next revision of the program will have feature based design and mesh generation for Finite Element Analysis (FEA). The program can be integrated within other tools using the JPython client side scripting language. This tool is very promising; though it does not have the full feature set needed for conceptual design that tools such as PRO/DESKTOP offer, however it has the collaborative and interoperability advantages over PRO/DESKTOP. In order for PRO/DESKTOP to achieve similar results and platform independence it would have to be completely rewritten in Java.

(Shyamsundar and Rajit 2001) developed the collaborative Product Assembly Design System (cPAD) tool. It is an internet based tool for the collaborative assembly design of mechanical assemblies and constraints. It utilises the Assembly Representation (AREP) developed by the authors consisting of assembly hierarchy graphs (for assembly representation), relational graphs (between components), assembly level information and component/sub-assembly information. The system uses a three tier architecture for the information model with an Application Service Provider (ASP) and middle-tier Intelligent server written in Java to control the communication between client and ASPs. The software has a web server (Apache), Parasolid based kernel for solid models (in C++), an Microsoft (MS) access database and catalogue servers with Java Database Connectivity (JDBC) interface for user data and as a repository for existing (read : reusable) data. Finally there is a visualisation server which converts the parasolid model into a compact geometry model. The client interface is a Java applet utilising SWING and Java3D. The tool allows the creation and editing of basic primitives and embodiment cases for assemblies, and also entering the entire model data needed for assembly operations (constraints, assembly hierarchy, integration with parts catalogue,

and annotations). The tool enables Internet collaboration because it compresses CAD files by removing unnecessary feature geometry and sending only the external surface geometry needed for collision detection and assembly/geometrical constraints for assemblies. This results in a very effective tool for collaboration between OEMs responsible for overall product manufacture/assembly and the various supply chain companies, and reduces the possibility of components and sub assemblies not fitting together. The tool allows the generation of geometry as well, and the classification of assemblies and parts into libraries for re-use. However there is no provision to enable collaborative working in real-time, or to generate STEP compatible outputs for later processing (however both these functions are within the potential capabilities of this application). The lack of version control and integration with PDM application is another shortcoming that can however be overcome. The system has great potential, however the inclusion of machining feature abilities (and mapping those features onto STEP AP224 files), as well as mapping the assembly information onto AP203 or AP214 assembly data would be a good course of development.

(Huang and Mak 1999) developed an Internet based tool to implement the Lucas Design For Assembly Guidelines on the Internet using ActiveX technology for the interactive elements (such as forms). The tool primarily uses forms to elicit knowledge about the assemblies, following the Lucas DFA guidelines and generate results server side for display to the user. The team envisage use of the ActiveX components as a generic Design for X (DFX) tool. Many users can use the program, however there is no facility to enable concurrent “viewing” of the work in progress. It is a simple implementation and does not make use of or create explicit geometry that can be use in other applications such as CAD/CAPP, which limits its utility.

(Roy and Kodkani 2000) presented a software tool to allow engineers to exchange and collaborate on conceptual mechanical designs over the web and to evaluate existing concepts and select the best concepts based on different criteria. The method adopted for concept selection is the “gallery method” where different designers rank each other’s concepts according to the criteria (see also Case Based Reasoning Approach). A shared whiteboard is then used over the web to display all possible concepts and

discussion using an online chat tool is also possible to make the final selections collaboratively. The system is constructed using MS Access as the database and PERL language programming scripts/ Hypertext Markup Language (HTML) for processing the logic and displaying results. This tool does not yet include any ability to generate/view or markup any real CAD data and the authors are developing the system further in that direction. The system is not integrated with any PDM/PLM or CAD tool and lacks much of the functionality included therein. Due to this fact it cannot be considered to be a fully-fledged conceptualisation tool, however the concept ranking principle with different criteria would be a useful additional “tag” to models created within a PLM environment, and useful as part of a methodology for engineering concept design re-use and classification.

In summary, the web based design systems were designed as classic client-server applications to solve specific subsets of a product design. The systems do not use a systematic nomenclature that can be extended for other components of the overall product lifecycle, nor do they give any provision for the independent control of Intellectual Property Rights (IPR) of collaborating partners. Furthermore the architecture of the tools rely on overtly complex Java2 Enterprise Edition (J2EE) and eXtensible Markup Language (XML) messaging technologies that cannot be developed by the application users (the layman), making practical use of the tools in an dynamic and evolving enterprise laborious and impractical. Similar to the internet based design systems, the design information portals suffer from the inherent shortfalls of traditional client-server systems. Centralised authority limits its application in de-centralised VEs, the need to use high level programming languages to modify the ontologies precludes widespread adoption and use in working environments that are fluid and dynamic (therefore not having the time nor resources to constantly employ specialised personnel to modify the data models and logic). Furthermore centralised control, implies control by one of the participants in the VE, which causes a conflict of interest for specialist enterprises that work with competing companies simultaneously.

3.5 Knowledge based CAD

Computer aided design systems relying, to a greater or lesser, extent on Expert System (ES) rules to define the final design representation are referred to in this text generically as knowledge based CAD. They can range from simple parameterisation of 3D solid models (as implemented by all major CAD vendors today), to hybrid expert system / solid model based systems (CATIA), and finally the purely knowledge based CAD system where all geometry is defined in rules and variables such as Knowledge Technology International's (KTI) ICAD.

(KTI 2002) are the authors of ICAD the first, and most functional, knowledge based engineering design tool. All the designs are constructed using rules and parametric equations, however the program is not user friendly and the effort needed to build a single design model from scratch is very laborious for the design engineer. The program does have its strengths in intelligent configuration controlled models. The other tool available commercially is CATIA v5 which integrates an expert system with a fully featured solid modelling and surfacing tool. CATIA is a hybrid of geometric and non-geometric model definition programs www.ibm.com.

(Technosoft 2002) produced a commercial knowledge based CAD tool which is similar in functionality to ICAD (*see above*). The tools consist of two components (AML and TIE). AML has been optimised for web based collaborative development, as opposed to ICAD and also has TIE which is used to assess conceptual designs. The multi-discipline aspects of the program are also apparent, and this tool is more suited to generating and assessing designs at the conceptual level than ICAD. At the same time the model has to be fully programmed in to the KBS giving the same usability issues as ICAD.

(Kavakli 2001) presented an interactive KBS called NoDes which is integrated within a CAD system to allow the designer to specify designs using functional requirements. The generative "intelligent" nature of this KBS means it is intended for routine or "parameterised" design tasks. The system is optimised to generate solutions for plastic and metal window frame designs, and naturally the base "shell" can be used for any other purpose within Mechanical CAD (MCAD). It is written in LISP and is intended

solely for single user interaction. However it has potential to be used as a customer oriented “shop window” to define their requirements and view results graphically on-screen.

(Varun and Yollanda 2002) produced a comparison table for creating and storing knowledge bases in eXtensible Markup Language (XML), Resource Description Framework (RDF) or Darpa Advanced Markup Language (DAML). This table has a breakdown of the specific capabilities of XML and RDF and DAML to define data tags for knowledge based systems (i.e. correct definition of ontologies). The authors indicate that RDF and DAML are more suited for the storage and dissemination of knowledge than pure XML 1.0. In fact RDF is based on the XML framework and is as easy to implement as pure XML. The Protégé-2000 ES tool can use RDF, DAML (and its successor OWL) and pure XML for the storage/dissemination of knowledge. More information about RDF can be found on www.semanticweb.org . Since this article was published, the RDF and DAML standard have been superseded by the Web Ontology Language (OWL) which combines the format of RDF with advanced ontology concepts from DAML. The review highlights the choice of standards to encode the knowledge base for the collaborative design platform.

In summary, the systems reviewed highlighted the viability of using expert systems to save and generate engineering design knowledge and rules, as well as validating the principle that rule and knowledge based systems can effectively complement, but not replace, draughting based systems. This idea was further developed in this project, to allow the mixing of geometry with rules and a product data ontology in a shared and distributed environment to enable the coherent management and generation of designs in a heterogeneous VE.

3.6 Manufacturing KBS

Knowledge bases that persist the expertise of manufacturing processes and sequences are referred to, in this thesis, as Manufacturing KBS. The main role of manufacturing KBS is to aid the implementation of DfM methodologies for design engineers, and aid production engineers in the generation of process plans.

(Sanchez et al 1997) proposed a KBS for the integration of manufacturability issues into the conceptual design phase of a project (to work to DFM guidelines). The application is designed for prismatic components and 8 basic manufacturing features have been defined. The program takes geometry created in PRO/Engineer and using a feature recognition tool breaks the model down into its manufacturable features and passes the data to a manufacturability analysis module for generating process plans using the expert knowledge within the KBS. In effect the researchers tried to create a seamless integration between concept design and manufacturing using an expert system based CAPP and expert system based feature recognition. The authors make use of the Step Data Access Interface (SDAI) from STEP Part-22 (www.iso.ch) for storing processes, manufacturing resources and features. However, the software does not address the issues of collaborative design (with many concurrent designers), or geographically remote interoperability. This shortcoming is a serious omission since in order to work using concurrent engineering principles engineers and other disciplines need to share ideas and work together whilst being geographically isolated from each other.

Park (1999) described a software solution similar to (Sanchez et al 1997). The software system they developed is a multiple-agent-based tool with 3 main agents (design, process planning and manufacturing systems). The design agent uses a combination of design-by-features and feature-recognition in order to map out the manufacturing features. The software uses the ACIS solid modelling kernel for geometry, and the process plans for manufacturing are generated using GT code and part families, each feature being classified into one of many families of parts. The software tools use HTTP for communication and Knowledge Query and Manipulation Language (KQML) agents for queries. The author concedes the lack of true Concurrent Engineering (CE) support within the application because of the lack of versioning and data management functionality within the application. There is also no stated support for downstream applications by the use of STEP file exchange.

(Maropoulos 2000 & 1998) described an aggregate Process Planning application called Concurrent Assembly and Process Assessments BLocks for Engineering manufacture

(CAPABLE). The tool can be used to generate assembly using Artificial Intelligence (AI) and feature based methodologies to extract manufacturing process data from STEP AP224 files. The latest configuration of the application is called CAPABLE/Welding and used for the planning of metal fabrication processes.

(Chin and Wong 1999) have developed an integrated product concept development and evaluation system for plastic injection moulding parts (EIMPLAN). The tool is intended as a mechanism to gather information on many different concepts, allowing the designers to evaluate all concepts from manufacturability point-of-view as well as taking customer requirements on-board. The Expert System (ES) was constructed using KAPPA-PC Expert System Shell and the algorithms use GT codes for generating process plans for the mould. The program however does not demonstrate any multi-user or distributed functionality, which is similar to the shortfall that other conceptual design tools, evaluated here, demonstrate. The novelty of the tool however is the integration of manufacturing, process-planning, customer requirements and cost factors into a single interface. This “single-interface” approach demonstrates the utility of making applications more easily accessible to designers and customers.

(Fenves et al. 2000) have developed a tool for construction engineering but which has interesting features equally applicable for mechanical machining and assembly operations. The tool (SEED-config) is a conceptual design tool with four components: Design information repository – holding all information in a hierarchical order and allows the editing retrieval and storage of design cases. A knowledge manager creates the technology nodes. A classification reference manager is used for querying and manipulating classifications of components and sub assemblies (in this case buildings, but can be applied to others). Finally, a very basic geometry modeller creates the topology and geometry for the concept designs being developed. The software allows graphical “tree” and hierarchical views of the design knowledge within the system. This interface also extends to the other functionalities within the application. However the classification reference manager is probably not applicable to our particular engineering problems. This tool is only a proof-of-concept program and the geometric and real-time

collaborative aspects of the tool have not yet been implemented. Nevertheless the graphical knowledge and information management tools are very promising.

In summary, knowledge based systems as implemented by the above mentioned researchers aid the development cycle by pushing manufacturing knowledge higher up in the product cycle. However all the tools above are designed and constructed as standalone tools that work in dis-jointed parallel with CAD and PLM software. The main positive element in the above researchers' work is the application of STEP as the standard for defining and manipulating the product. The tools however are, of little value to modern VE oriented development projects which need de-centralised access to knowledge over a heterogeneous network via a commonly understandable medium which can be extended to suit the particular project's function.

3.7 Feature Based Design

Feature Based Design (FBD) is a mature technology developed in the 1980s that however merits mention as it is the main methodology for design draughting in 3D Computer Aided Manufacture (CAM) today. FBD allows engineering designers to generate their product geometry using either product or manufacturing features as opposed to abstract geometry. It is generally applied using 3D solid models using both Boundary Representation models (B-REP) and Constructive Solid Geometry (CSG) and is, in theory, more intuitive for the designer as it uses design concepts as opposed to xyz geometry to codify the design idea. Furthermore, because the features used are not mere xyz coordinates, they also implicitly contain information about the function and manufacturing processes used to realise the product that are of course not included in simple solid models. The caveat for feature based design, is however the fact that the development team are limited in their designs by the available features and complex features (a combination of multiple features) require a leap of imagination among the designers and can produce designs which are invalid.

(Brunetti and Golob 2000) From the Frauhonhofer Institute of computer Graphics described a prototype system (SINFONIA). This is a design-by-features tool intended for use as at the conceptual stage of development. The feature definition adopted within

this application has been that recommended by the FEMEX (Feature Modelling Experts) working group. This definition structures data into four levels: Assembly model, Part model 1, Feature model 1, Generic model (geometry). The software captures conceptual design intent (requirements), functions, product structure, assemblies, and individual components with explicitly declared features. In addition the program also stores the “principle solution” which is the “effect” of the component such as fluid flow, load bearing structure, electrical resistor etc... and at a higher level the “working principle” which is a structure of principle solutions needed for the particular problem. Finally there is the “embodiment” which contains the overall layout (assembly or packaging) for the product. The researchers aim to create a fully feature based conceptual design tool which can supply data to downstream applications (CAD tools, CAPP, CAM). This includes the storage and reuse of the design knowledge and intent gathered in the early stages of the concept. However, the group does not state how the data is to be shared across different applications in the downstream processes and whether the software tool works in a real-time collaborative and distributed environment.

(Case and Hounsell 2000) presented another method of extracting features from the conceptual design. The application FRIEND (Feature-based validation Reasoning for Intent-driven ENgineering Design). In this program the designer’s intent is entered consciously during the modelling phase (which is based on standard 3D solid modelling instead of a “design-by-feature” application). The software is “design intent” driven and is therefore not limited to “building block” type of feature-based design. It provides the necessary data to create the process plans however it can only recognize a limited number of features and is capable of working with single components only (no assemblies). The application is not specifically designed for conceptual design or for passing data on to process planning applications, and there are no claims made about any collaborative or networking capabilities to the application. Lastly, the software does not mention any use of AP224 for the storage and dissemination of the explicitly defined manufacturing features.

(Nam and Wright 2001) describe a real-time collaborative 3D CAD tool Syco3D. The program was developed to enable concurrent CAD users to share geometry over a network in real-time. The system is implemented in SGI Unix (IRIX) and written in Tcl/tk script. The system has been tested with only 2 concurrent users and over a LAN rather than an extranet IP connection. The software tool is essentially a normal 3D CAD package but with a second window containing the “shared” view. The 2 users run a native CAD session on their workstations and see each other’s models in the shared window. This can be updated manually after modifications have been made to the geometry, and the CAD system stores a constructive tree view of the modifications committed to the models. The authors have noted the shortcomings of this prototype system as the inability to work across extranet/internet, no version control or security features. However there are some other disadvantages relevant to real-world applications such as the inability to share data with other CAD users and no stated methods to integrate within PDM or other knowledge management systems. However the shared workgroup environment is a rational and workable scheme to view and share models in a collaborative session and may be applicable to our problem.

(Inouye 2001) has created an open source automatic manufacturing feature recognition tool for configuration-controlled designs. The EXPRESS schema is in the public domain and can be obtained freely from the author. This schema may be vitally useful if the PRO/Desktop application is utilised for generating geometry. This schema will expedite the generation of process plans from the conceptual design. The complete functionality of this EXPRESS schema is limited to recognition of prismatic manufacturing features, which renders it useless for most engineering design tasks as rotated components far outnumber prismatic machined components in a design.

In summary, the above research indicates that the use of FBDs continues to be considered a prime technique to aid the process of improving the coherence and integration of the design lifecycle. The natural attraction of FBDs is of course the intuitive interface for the end users working with the system, and the adoption of STEP as the main standard for persisting the features opens up the possibility for interoperability with heterogeneous PLM and CAM/CAPP systems. Inouye’s schema

for extracting features from standard STEP files expedites the process of integration. However, the researchers have primarily based their work around rigid, closed working concepts which are not relevant in an open de-centralised and shared working environment of a modern VE based development team.

3.8 Application of STEP based tools

The following reviews look at tools which make use of the International Standard ISO 10303, popularly known as STEP to aid the process of product development, design sharing and design for manufacture. STEP (www.iso.ch) is a standard developed to replace the obsolete Initial Graphics Exchange Standard (IGES) (www.uspro.org) standard which has been used to exchange 3D and 2D geometry between different vendors' CAD systems. The aim of STEP is to encompass the needs of all branches of engineering using separate Application Protocols (AP) that define the taxonomies for each branch of engineering. The taxonomy is defined in a language called EXPRESS and the STEP file can be persisted (as standardised) in a number of formats including normal text (Part 21), XML and as a database schema.

RAMP (Rapid Acquisition of Manufactured Parts) is the first tool that can utilise STEP AP224 standard for product definition using machining features (LSC 1999). It was created by the Royal Navy in order to reduce its inventory of spare parts and allow the rapid machining, using Just In Time (JIT) philosophy, of parts on a need basis. In order to achieve that aim the electronic model of the component needs to store the information for the complete lifecycle of the component. This can be achieved through the use of the STEP standard. The other advantages of STEP are the platform and vendor independence, and also being an ISO standard means that Application Protocol information is fully in the public domain. AP224 contains the full product information for the domain of product geometric model, machining features, NC generation (tool path), manufacturing process (for automated CAPP) and inspection schedules. The RPTS MP (RAMP Product Data Translation System for Mechanical Parts) software was tested for the creation of prismatic and axi-symmetric single piece machined components. The software itself is based on PRO/ENGINEER, and uses a design-by-features working method to insert STEP manufacturing features into a "stock" raw

material block. It also enables the inclusion of the other lifecycle information defined in the AP224 standard. After the model is generated the data can be verified using a Quality assurance tool called RSVP (SCRA RAMP STEP Validation Process) which is a viewer and parser for AP224 files. The STEP 224 files are sent for the generation of process plans in the LOCAM software tool. This has been written to parse AP224 files and generate a process plan according to the complex manufacturing rules contained within its graphically-accessible knowledge management tool. The major shortcoming with the approach taken by RAMP is the inability to generate feature geometry (AP224 machining features), interactively using a network based applet which allows people without the specialised CAD tool to make concept designs and test them over a network. Also that facility would enable collaborative working practises and enable concurrent design between dispersed collaborators.

The application protocol definition for STEP 224 contains the full specification for the Application Protocol for the representation of information needed to produce a mechanical part definition for process planning of a single piece or an assembly of piece parts for machining operations, and specifies the integrated resources necessary to satisfy these requirements. Much of the required functionality for process planning and manufacturing applications (as needed for DfM tools) cannot be appended to the product model. Also representation of assemblies would be a required feature of the manufacturing processes that is missing from the standard; this would induce difficulties in conceptual assembly design or DfA working practices. These missing features are implemented in another STEP Application Protocol, AP214 has within its scope all the necessary information missing from AP224 including assembly information, process plan information, configuration data for parts and composite parts (also defined in AP203). The other shortcomings are the lack of tools to actually create parts that comply with AP224. With the only exception being (LSC 1999).

(Bhandrakar and Nagi 2001) in association with STEPTOOLS Inc, described an Initial Graphics Exchange Specification (IGES) to STEP AP224 translator that is written using ST-DEVELOPER (by STEPTOOLS) and holds the model data in an OO database (ROSE). The translation process is as follows:

- The IGES reader parses through the IGES file and reads in all the entities into memory;
- The code then looks for a Manifold Solid Boundary-Representation (B-REP) Object Entity (MSBO) entity. The MSBO is the IGES entity used for representing topological information in the B-REP format;
- If there are no MSBO's then form-feature information cannot be found from the available IGES data;
- If MSBO entity is found, the code parses through the MSBO and identifies any voids if present and creates corresponding AP224 object and populates them;
- If no voids are found, the surface and edge data is examined to identify other form features like steps, protrusions, cut-outs; and
- After all the form features are identified, the STEP data is stored in the 'ROSE' format, which can be converted into a STEP Part 21 file format using the format utility.

The above sounds like a very promising manner to create AP224 files, using feature recognition. However, the researchers revealed that the system has not been implemented. Nevertheless, the author has presented a functional specification of how to implement such a translation tool. However, it has to be noted that the author does not mention any ability to manually enter information that is not contained within the IGES geometry.

(Sharma and Gao 2004) proposed a feature based conceptual design system (FBCDS) that's integrated within client-server applications such as PDM. The tool would be used to generate AP224 files, and be based on an ACIS solid modeller engine. DfM rules are used to check the validity of manufacturing feature attributes and developed using CLIPS. LOCAM is integrated with CLIPS and performs the automated process planning functionality on the generated AP224 file. The user interface for the complete design tool is an ActiveX control and can thus be deployed across the Internet and accessed via Internet Explorer. There is a graphical "hierarchical" feature editor for the CSG tree, which allows drag-and-drop creation and editing of product model features. The integration with the, now-defunct, PDM application (MOTIVA) is very strong and

the design generated in the FBCDS automatically checked into the database. Also the program can generate PDM product structures. The integration lowers the amount of repetitive work that has to be done by the engineer. The above application seems to be a good basis to continue upon for further work.

There are currently around 4-5 commercial development tools for STEP based applications and for STEP schemas using the EXPRESS definition language. (Haenisch 2001) are the producers of the ExpressDataManager, which is both an express schema development environment (both EXPRESS C and G), as well as producing commercial Step Data Access Interfaces (SDAI) that enable STEP data to be stored and manipulated from databases via bindings to XML, C++ and Java. EPM is used in PTC Windchill to automatically handle translation of CAD models from native CAD formats such as PRO/E to vendor neutral STEP. The standard supported is AP203 (www.iso.ch).

(LKsoft 2002) created the Java SDAI implementation to generate Java classes accessible to other Common Object Request Broker Architecture (CORBA) programs that enables STEP read/write functionality for applications. It implements SDAI for Java only. Steptools, www.steptools.com (2002) ST-Developer is the most mature product and implements all Part-21 and Part-22 STEP bindings. It also produces a ST-Repository for storage of OO database of STEP objects (whatever the binding, Java, C++).

In summary, the above tools illustrate the viability of STEP native Computer Aided Process Planning (CAPP) and CAD systems. However the above tools are limited by the fact that they do not provide an easy method of sharing the knowledge persisting in the STEP taxonomy due to a lack of widespread adoption of EXPRESS parsers. The use of XML to persist (store in the database or file system) STEP files was also found to be cumbersome and creates overtly large file sizes compared to a Part 21 file. The step tools in the final review illustrate the fact that the automated CAD translation systems to and from STEP work with AP203 files only (holding the basic geometry data only) thereby losing the additional meta-data associated with features, materials and manufacturing processes for each item and assembly. However the availability of a STEP native CAPP application provides an important downstream processing

application that would ideally be integrated within a holistic product lifecycle system to provide early manufacturing evaluation.

3.9 Summary of technology review

The survey looked at a broad stream of research in the topics of concurrent engineering and early design engineering through the inclusion of an overview of some commercial, and more importantly, open-source research in the area.

In the area of PDM/PLM, it was observed that the competitive needs of each vendor have outweighed the demands of customers, creating systems that are difficult to integrate from vendor to vendor. In addition the PDM/PLM systems do not implement taxonomies which are extensible and configurable at the user level, creating systems with disconnect between the dynamic needs of end-users and the available functions of the tools.

State of the art in Conceptual Design demonstrated an impressive application in Alibre which combines a distributed CAD system (STEP native) with PLM functionality in a multi-user environment. What lets down the system, is a lack of end-to-end integration from concept to manufacture and the reliance on a central database to manage the system.

Design for Manufacture tools and methods were overwhelmingly academic in nature, illustrating the lack of application of this important concept. In practice the systems reviewed were closed box knowledge based systems disconnected from the product lifecycle and not applicable as part of a VE project due to lack of share-ability and integration with the wider knowledge bases.

Knowledge Based CAD systems and Manufacturing KBS, whilst existing in the commercial domain in the form of ICAD and as a hybrid form in CATIA, suffer from the lack of shared and distributed functionality. The commercial implementations also, for commercial reasons, integrate only with PLM systems from their respective vendors. Data interchange in a heterogeneous VE becomes impossible via the lossy (reduced

features exported permanently losing model information) model export functions (CATIA-STEP) and a lack of de-centralised sharing of models with systems from other vendors (or open source systems).

Design portals and web based design systems, try to overcome some of the problems of 'thick client' systems by holding all functions and processes on a centralised server. As was demonstrated, this method of working is sufficient for enterprises working in a multi-tier supply chain with a large OEM. However such systems are not implementable for non-hierarchical VEs.

Feature Based Design systems demonstrate the inherently superior usability of graphical development systems based around clearly defined features assembled together into products. They, however are also the most inflexible when it comes to the creation of new designs not based on existing features, and revert to a standard geometric system. The tools also were not integrated into the entire product lifecycle but were standalone products good for early concept evaluation purposes at most. The systems were not constructed with a distributed architecture (probably due to the large size of the models and features making them impractical to use over intermittent Wide Area Networks).

The STEP tools and methods reviewed illustrate the viability of utilising the STEP standard for defining, sharing and processing engineering product data. However the tools utilised the standard SDAI and STEP P21 formats (www.iso.org) for sharing the models limiting them effectively to the use of systems with EXPRESS parsers and a centralised database with an SDAI connector.

The basic conclusion is that there are a lack of usable solutions to bridge the gaps between specifications, concept and manufacturable product configurations within a single enterprise and even more so in the extended, or virtual, enterprise where the homogeneous information systems currently being researched and implemented do not offer realistic solutions to the inter-enterprise knowledge sharing issue. Thus the aim of this work is to articulate a methodology and techniques to create workable

heterogeneous product development system for Virtual Enterprises of inter-dependent small to medium enterprises.

4 Proposed Methodology for Distributed Collaboration

This chapter details the problems with current inter-enterprise communications for VEs and details a solution that was developed to overcome the shortfalls in establishing rapid, VEs between dissimilar enterprises. This is considered to be the 'Tier-1' within the overall two-tier architecture for inter-enterprise collaboration and is responsible for connecting the enterprises to each other in a mutually understandable format and medium.

4.1 Communication as a foundation of collaboration

One of the foundations of effective collaboration is communication. Communication, within the context of this research is thought of as composed of a medium and a format. Once mediums and formats are formally established, comprehensible communication between two entities may take place. This is applicable to any type of communication. For example a conversation between two persons would be:

Medium: Oral (face-face).

Format: Language (known to both).

Or another example would be written communication between groups of people would be:

Medium: paper and ink.

Format: alphabetical characters in common language.

Naturally both these elements must match at all collaborating parties for mutually understandable communications to take place. This component of the research aimed to solve the key problem of establishing communication links between geographically separate and organisationally unrelated entities, rapidly and at low cost, with the aim of the communication being the collaborative development of engineering products. Whilst this may sound like a simple problem in today's world of standardised data-telecoms, the lack of end-to-end standardisation of taxonomies, authentication / access control systems and formats between enterprises has been a glaring shortcoming of modern Business to Business applications (in an analogue to the shortcomings of

battery technology in the rapidly developing mobile electronics industry), in effect this has been one of the major drawbacks that has stunted the growth and spread of PLM and KM technologies to smaller enterprises today, as shown in the literature and technology reviews.

Figure 1 illustrates the holistic operation of the system including the heterogeneous ontologies, the medium to transport the data between the ontologies and the format for understanding the communication. Ontology for enterprise A contains the business and process knowledge for company “A” within the virtual enterprise. Similarly Ontology for enterprise “B” contains the domain ontology (which is in this instance DIFFERENT from the ontology in enterprise “A”). The “medium” in this figure is simply the means by which communications takes place (in this instance via TCP/IP i.e. over the internet) The red line denoting the format of the taxonomy denotes the file formats used to encode the ontologies of enterprises “A” and “B”. Finally the workflow processes aim to glue the disparate business processes from the unrelated enterprises to enable the automation of the business processes in the VE.

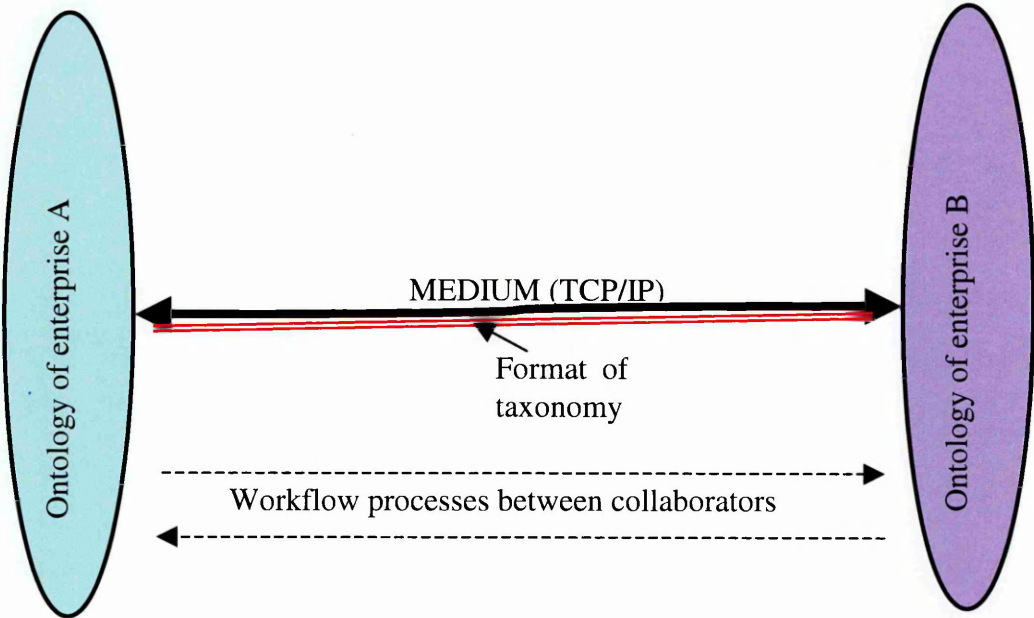


Figure 1 Format, Medium, Ontology and workflow

4.2 Format of data model and its structure for collaboration

The 'format' aspect of collaboration between enterprises is independent of the medium used. However the key problem, within the context of this research, of the format is creating a universal language and taxonomy (a taxonomy in this context is the nomenclature used in a collaborative engineering setting) that will meet the following challenging criteria, which are a subset of the overall aim of the research:

- Providing a superset of all possible taxonomies in use today or in the future for inter-enterprise collaboration between engineering oriented entities.
- Providing a format that can be used commonly by all and at no cost to enable the rapid dissemination and implementation by all parties in a collaborative venture.
- Enabling the rapid authentication and access management of all participants.
- Providing a method to rapidly resolve syntactic differences between different enterprises' data structures.

The above mentioned requirements have proven to be a tough challenge for researchers in the past as shown by the disparate technologies which only look at partially solving the problems. The solutions in existence today include well established International Standards (OSI IEEE 802) for network links and data communications protocols such as Transport Control Protocol/Internet Protocol (TCP/IP) for top-level data communications between Institute of Electrical and Electronic Engineers (IEEE) (www.ieee.org) 802 connected devices which are most computers and Local Area Network (LAN) enabled devices.

However even though there are clearly established guidelines and even standardised formats (such as XML and RDF standards of the www.W3C.org) there remains a practical disconnect in the sphere of standardised nomenclature (or taxonomies) for collaboration between disparate enterprises (as demonstrated by the continuing research on the problems of B2B integration in a large number of industrial sectors shown in the literature review).

Since no two enterprises are the same, the idea of using XML based messaging for inter-enterprise collaboration is not easy as the collaborating companies' XML schemas have to be mapped to each other.

In order to achieve speedy interoperability, a standard has to be set for basic messaging. There are a plethora of standards in development by various groups including RosettaNet (www.rosettanet.org), an endless list of XML schemas and data type definitions. These standards implement different levels of detail in the ontology. Some like STEP PDM attempt to implement the complete information structure of the engineering enterprise. Others like JuxtaPose (JXTA) by Sun Microsystems (2003) implement only the "messaging" components.

The choice of standards to use depends on the level of "standardisation" that all enterprises can adhere to. However, with literally millions of enterprises, large and small, each with its own company-specific data structures and peculiarities, it can be safely assumed that radically altering the internal workings of all enterprises, to adhere to a single standardised taxonomy and data transfer medium is an improbable proposition on the grounds of :

- Cost (data-remodelling, loss of productivity, and lack of short term Return On Investment);
- Unsuitability of a standardised taxonomy for all companies' working needs; and
- Lack of legislation forcing companies to adhere to any standardised medium.

To have a holistic overview of the problem, details of various company taxonomies and standards designed for inter-enterprise collaboration were collated. These are:

- ArvinMeritor taxonomy and data models (Tier-1 automotive supplier);
- Mabey & Johnson data models (design, manufacture, installation and maintenance of rapid construction steel bridges);
- ISO 9001:2000 management standards guidelines (simple management standard); (www.iso.org)
- ISO STEP-PDM (www.iso.org), STEP standard taxonomy defining an internationally agreed upon taxonomy for engineering Product Data

management, this standard complements the ISO-10303 STEP and Industry Foundation Classes (IFC www.iai-international.org/) set of standard which defines the product models for different industries, modelled in EXPRESS and stored in plain text format, XML binding is also available;

- ISO 10303 STEP, basic engineering product data, different Application Protocols define the data models for different industries. (see www.iso.org for full details), generally modelled in EXPRESS, and stored in plain text format;
- RosettaNet – XML based generic B2B standardisation model;
- JXTA, (www.jxta.org) open-source generic low-level authentication and data management nomenclature for all types of applications (based on XML format);
- Proprietary B2B integration tools (PTC Windchill info*engine, Biztalk) www.ptc.com and
- Proprietary data models for enterprises (SAP's ERP data model, COMPIERE ERP data model) www.sap.com www.compiere.org .

Taxonomies above provided details of different characteristics of an enterprise's data model, and none of them can be considered to be a 'complete' representation of a company's data model (none of them claim to be, with the exception of SAP's massive taxonomies). The only example of such a complete taxonomy in existence is SAP's ABAP (SAP's proprietary programming language) based taxonomies that attempt to define all areas of any potential enterprise's data needs from accounting to project management to PDM/PLM and all in between. SAP's 'kitchen sink' approach using an extremely large and complex proprietary data model (closed source and rather pricey), allied to a proprietary data format and language (ABAP) excludes it from being a potential solution for solving the inter-enterprise collaboration problem as its overt complexity, extremely high cost, and uncompromising approach to enterprise's existing data models makes it wholly impractical for widespread adoption by the SME's that comprise the majority of today's collaborating engineering industries.

The rest of the taxonomies however each provided a subset of overlapping functions that can be normalised to provide a 'standards based' enterprise data model usable by all. Excellent examples of taxonomies are the Application Protocol 2xx standards of the

ISO-10303 STEP standard. A more in depth-look at the actual taxonomy created can be found in Chapter 5.

For the actual formats of the data, the following standards were looked at:

- XML – eXtensible mark-up language (www.w3c.org), a plain text format mark-up language used to define industry specific mark-up languages (some of which are defined above). BizTalk, MathML and RosettaNet are examples of taxonomies implemented with XML markup;
- RDF, Resource Description Framework, (www.w3c.org) is a development of the XML formatted language for ontologies, incorporating a ‘node-triple’ to describe the context and Universal Resource Indicator of the item of knowledge. RDF by its very nature means that three entities are needed to represent the same data as represented in a standard XML formatted document;
- OWL, Web Ontology Language (www.w3c.org) is the definitive XML/RDF based ontology modelling and manipulation format incorporating Description Logics and constraints. It uses RDF node triple on an XML format;
- EXPRESS is an ISO Standard (ISO 10303 Part 11) (www.iso.org) modelling and plain-text data-representation language for data models. (currently STEP, and STEP-PDM are defined using the express language and text-format);
- SQL Structured Query Language, plain-text format relational database models and data definition language; and
- Proprietary CAD models with additional meta-data for non-geometric information, for example the CATIA and Pro/Engineer CAD formats.

4.2.1 Demonstration of communication architectures

The various technologies were tested using the same test computers as in all experiments, and the same data set was loaded to maintain the integrity of the test. Each application was tested in series, first internally in the lab, and then over the Wide Area Network between Cranfield and Durham universities. Test criteria used were:

1. Functionality for defining enterprise data models for the following data:

Object oriented data model, text information storage, material, manufacturing and other physical criteria, storage and storage of geometry of engineering product.

2. Data Management ability:

Revision control, lifecycle data, user data, access management and audit trail.

3. Finding and interpreting data:

Query ability, query standards, query speed, intelligent search (agents) and explicit taxonomy definitions.

4. Collaboration:

Usability over network, Schema mapping ability (1-1 and 1-many).

Testing method was as follows:

XML, RDF and OWL were tested using the Protégé system (and this ensured optimal comparison of the systems). Further data was interpreted from the standards documentation for the three overlapping standards from the World Wide Web Coalition's test data www.w3c.org.

EXPRESS was tested using the Express Data Manager (EDM) from JOTNE www.jotne.no. Testing of STEP CAD models was conducted using the open source modelling engine and the Granite1 solid modeller incorporated in the Pro/Desktop environment. In the test environment, test-examples of company specific engineering designs were modelled in STEP and also the Pro/Desktop

SQL was tested using the MySQL database and PHPmyAdmin www.mysql.com tool. In the test scenario a subset of a STEP AP was modelled into relational form and loaded as an SQL script and populated with data. This was the same data model as used for the XML/RDF/OWL/EXPRESS test cases. Results are summarised in Table 4.

Form at	Function defining model for data	Data managemen t ability	Finding and interpreting data	Collaboration
XML	all types of data can be defined including most taxonomies and schemas.	Nothing built in, but can be added as entity to data model	All XML and plain text queries. No specific rules engine for agent based searches and inferencing	Bulky data model slow over network, 1-1 mapping simple, 1-many extremely difficult!
RDF	Since its XML based it inherits the flexibility of XML, with the addition of inference. First truly OO implementation	Nothing built in, but can be added as entity to data model	First implementation of inferencing using an XML standard, RDF triples used for explicit data-model definition and defining entity context	Same as XML, however 1-many mapping is simplified by having explicitly defined context for the entities and a URI for

				that definition.
OWL	Builds upon RDF/XML and much more data-complete than plain XML. It has better constraints mechanisms. True OO implementation	Nothing built in, but can be added as entity to data model	Full ontology definition standard building on previous RDF/XML with full inferencing, explicit/implicit definition of ontology enabling autonomous intelligent agents to manipulate /w query data from any source OWL/DL.	Bulky data over network, but 1-1 and 1-many data mapping is simple for well defined ontologies using OWL's built in coherency for ontologies.
EXPRESS	EXPRESS-G has been used to model all STEP standards including entities, types, rules, constraints. OO implementation	Nothing built in, but can be added as entity to data model	EXPRESS data models are standardised data models with strongly defined rules and constraints. Queries are very fast due to this standardisation, however the data models don't have implicit relations built in, and few EXPRESS parsers are in use.	Part-21 files (plain text) efficient over network. 1-1 and 1-many data mapping simple if both businesses are already using Express bases systems.
SQL	Simple relational model, not OO, preset entity types, limited rules	Nothing built in, but can be added as entity to data model	SQL is queried using SQL. Not suitable for free text searching, or for agent based systems	Very fast over network using RDBMS, data modelling easy for 1-1, but complex for 1-many.
Proprietary CAD standard	Pre-defined CAD models with extensible 'meta-data' for adding enterprise specific data model.	Built in lifecycle and revision control.	Requires proprietary system for finding meta-data. Data is held in binary format so cannot be searched by third-party agent based systems or search tools. System has no inherent	Slow over networks. Data modelling very laborious and costly, if both enterprises not using same proprietary standard.

Table 4 Data format comparison

In summary, Table 4 illustrates the range of possible data modelling languages to use for sharing and interpreting data between enterprises using complex heterogeneous taxonomies. Whilst traditional standards like SQL with relational databases perform well on the speed aspect of data access, they are let down by a lack of semantic interoperability that hinders the ability of intelligent agents to reuse data from their repository without explicit knowledge of the data-structure and its implied meanings. The EXPRESS language is in an unenviable position of being a well thought out modelling language for taxonomies that is an ISO standard and used to create the largest most comprehensive standardised taxonomies, STEP and Industry Foundation Classes (IFC). However it suffers from the lack of URI and description logics, as well as from a lack of ubiquity which limits its usefulness in a practical business to business settings,

as demonstrated by the small number of practical implementations using EXPRESS for modelling ontologies (beyond STEP) a decade after its release, and the difficulty of interoperability with the practical tools that are geared towards the W3C and OMG (UML) based standards. XML, whilst a useful markup language, does not provide the semantic meanings necessary for an intelligent taxonomy. This functionality is remedied in RDF (N-triples) and OWL which add description logics and constraints. Of all the languages OWL was the standard that most closely met the need for the data format of the collaborative knowledge base.

Figure 2 illustrates the problem graphically. The figure shows the data-model and data volume situation at present, where two fundamentally dissimilar data models from different enterprises (A and B) whose data models and depth of knowledge evolved over many years, attempt to establish a coherent Business to Business collaborative system between them. They are meant to illustrate the level of abstraction of the ontologies used within an enterprise vis-à-vis the ontologies shared between enterprises.

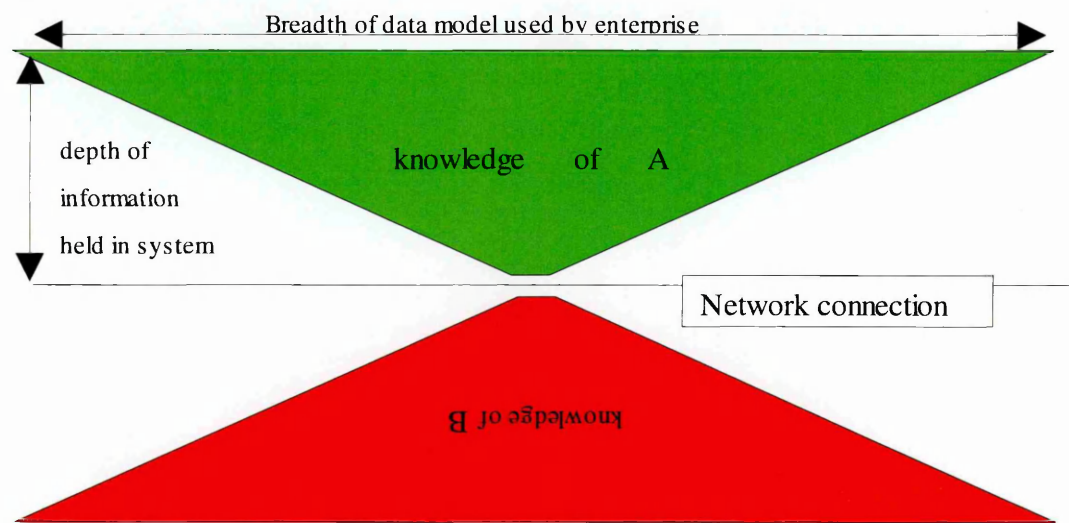


Figure 2 Depth and breadth of traditional B2B connections

The layout of the diagram is designed to illustrate the complexity of the ontology (breadth), and the amount of knowledge that is actually stored in the knowledge base (depth).The 'area' bounded by the breadth of the ontology and depth of the knowledge constitutes the 'Knowledge to be shared' within the collaborative environment.

It is practically impossible to match these data models on a one to one basis from one enterprise (e.g. A) to the other (e.g. B) rapidly, let alone on a one-to-many or many-many basis with multiple collaborating parties.

4.2.2 Proposed 2 tier architecture for heterogeneous inter-enterprise collaboration

With the above details in mind, in the view of the author only a very low level subset of all enterprises can be “standardised”. This small subset should exist only as a medium to enable communication, identification and access control management. All other aspects of the collaborative data environment are enterprise specific (although constructed from a subset of ontological components).

This project created the two-tier solution as shown in Figure 3, which illustrates the overall architecture, and in Figure 4 which shows an exploded view of the light blue and yellow tiers of the architecture. This architecture enables real-time connections between enterprises by eliminating the need for complex data mapping between heterogeneous taxonomies and data standards.

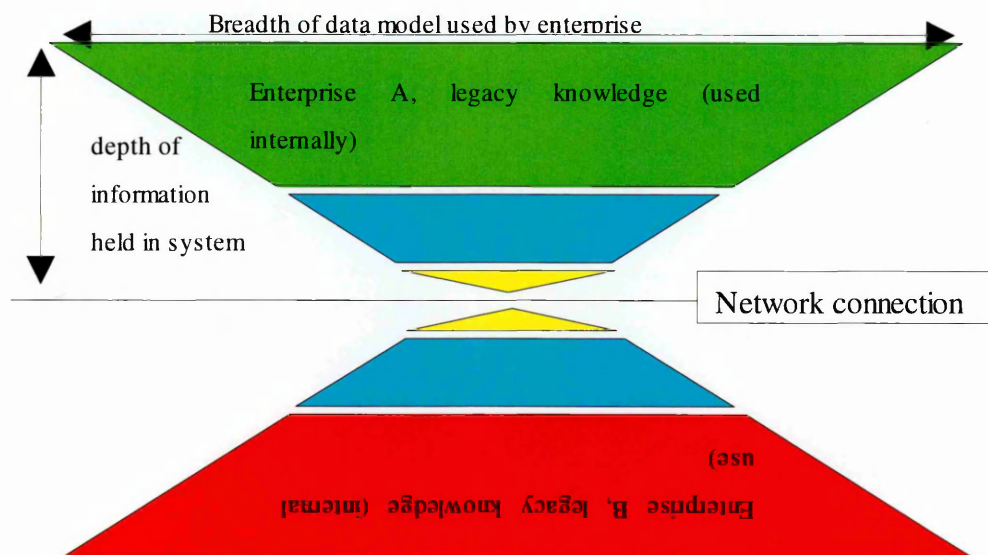


Figure 3 New B2B model with separate layer for authentication and establishing access-control

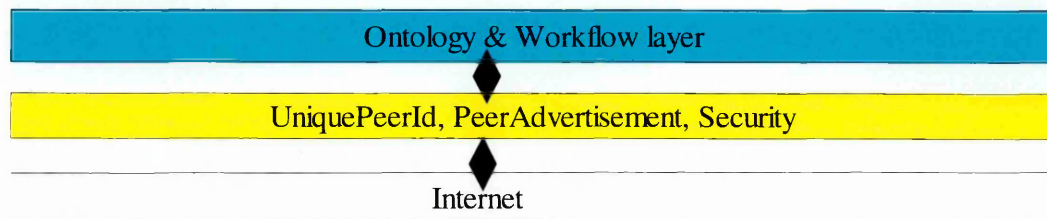


Figure 4 two tier communication architecture

The two-tier architecture is a unique idea which in effect solves one of the great problems of inter-enterprise collaboration, through a very simple yet effective solution. Each enterprise can customise and run its internal ontologies (Enterprise A in green and Enterprise B in red) and workflows within their own company's internal requirements, and when the company collaborates with another company, they can visually bridge the differences in taxonomies without technical complexities or programming. The communications and security aspects (authentication tier), see Figure 4 for a detailed breakdown, are handled automatically by the JXTA open protocol (the yellow tier). Speeding up of the integration process is further enhanced through the use of open standard semantic web enabled ontology formats (red and blue tiers) for storing the information, plus the use of open-standards in addition to company ontologies to define those ontology components. The blue tier bridges the gap between the internal and global ontology through the use of international standard nomenclature, the idea being that through the continued use of the blue tier, the depth of the red tier can be reduced (the depth illustrating the volume of knowledge held). The breadth of the red tier cannot however be eliminated as individual enterprises will always remain unique to an extent. However this system enables companies to persist as much of their reusable knowledge as is practicable in an open standard manner that enables rapid collaboration.

The aim of the system is to start off with a subset of the company's data models which are standardised for the requirements of a one off project, and the shared standardised data model to gradually permeate upstream into the company's existing base of knowledge and eventually supersede the internal data model after a number of collaborative projects actively increase the scope and depth of the new data model and thus rendering it an essential part of the company's daily practice.

The reason the two-tier architecture can achieve such a critical mass of users are:

- ⌘ The ontology and workflow layer implements a full subset of internationally agreed standards for product data modelling and business process automation;
- ⌘ It comes in a free package which is easy to install and use;
- ⌘ Because it's a free easy to use package implementing acknowledged nomenclature standards in an open and extensible format, it can be used out of the box for collaboration by a large number of enterprises without hassle; and
- ⌘ The flexibility and breadth of the data model is sufficient for all enterprises' production needs, and can thus supplant and eventually replace legacy systems as its advantages permeate through the enterprise.

From the above it can be seen that the critical aspect of the system is maintaining the ease of use and ease of extension that enables the collaborating partners and internal users to broaden the data model for their requirement within the scope of the ISO standards based on functional requirements and without the need for high level expertise. This is particularly critical for uptake in the small to medium sized sectors. These drivers were key to the idea of creating this flexible solution that accommodates the needs of established companies with internal knowledge bases, as well as the smaller collaborators with more limited requirements. The combination of the format and medium constitutes the first tier of the two tier architecture with the generic super-ontology as detailed in chapter 5 constituting the second tier.

4.3 Medium for de-centralised collaboration

The second pillar of collaboration is the communications architecture between the individual participants in a collaborative team i.e. the medium. In the sphere of inter-enterprise collaboration, electronic data communication is the prime medium for enabling the exchange of product information and related data between physically separate collaborators. However, electronic data communication can be structured in different ways, and each method has its own strengths and weaknesses vis-à-vis practical collaboration and knowledge sharing. These are:

1. Offline links such as physical disks which are shared, or e-mails with attachments;

2. Client-server systems controlled by the largest member of the collaborating team;
3. Client-server system controlled by a third-party (Application Service Provider); and
4. De-centralised systems such as Peer-Peer architecture.

Four applications were constructed and tested. Some are centralised, controlled by a third party or de-centralised. The technologies and techniques investigated are:

- Commercial Client Server application (centralised -commercial) ;
- Open-Source Client Server application (centralised-open source);
- Application Service Providers (independently controlled centralised); and
- Peer-Peer net based services (de-centralised).

The following sections will provide details of the terms.

4.3.1 PLM systems

As a benchmark for demonstrating the functionality of the research, a commercial PLM system of the type widely deployed to fulfil the task of collaborative design and knowledge management in today's leading projects was used. In this instance PTC's Windchill was made available for the researcher's use, and this best-of-breed PLM system acted as a benchmark for the collaborative design system. Some of the unique (brand-specific value added functionality) advantages of the system were eliminated from the test in order to make it a level-playing field for comparing the key functions required for effective de-centralised collaboration as specified in the introduction. The functions eliminated from the tests include:

- ❏ Pro/Engineer connectivity (as the intention is for generic CAD connectivity not the vendor's specific application);
- ❏ 3rd party visualisation tool that is included in Windchill (this is a cost based value added tool, which can be purchased by any enterprise wishing to use its functions); and
- ❏ Any additional packages that are not part of Windchill basic PLM (i.e. Rapid prototyping add-on, manufacturing evaluation add-on etc..). These were eliminated because they are functions relevant for very specific industries that

can still purchase these tools as 3rd party add-ons with any of the competing solutions.

Windchill is a large, out-of-the-box PLM system. The system is widely used in the aerospace and automotive sectors, especially in companies that also utilise the Pro/Engineer solid modeller as the system is provided by the same vendor (PTC) and thus is closely integrated into a family of application suites. It uses open source technology for some of its key components. Windchill is a traditional document management tool, that stores all manner of electronic data in an object-relational store with a large number of attributes and revision control functions, data flow is managed by a highly customisable workflow engine. It does not however have an intelligent method of containing and persisting information in an object oriented format. To alleviate this, the functionality of the system has been extended to include the management of knowledge in ontology. The integration with Protégé was made using Windchill's bespoke but highly configurable workflow engine. More information on Windchill's specific functionality can be found in the Windchill manual and application developer guide.

PTC Windchill was installed using the following configuration on the Dell Xeon server see Table 5:

Technology	Use
Oracle 8i Database server, with medium sized database and UTF16 character set.	Storage of schema and data is on an Oracle database. The data is accessed via a Java Database Connectivity (JDBC) interface.
Apache 1.3 web server (port 80)	Interface to enable the web based access of the Windchill system
Java 2 SDK (1.3.1)	The Windchill system was written and compiled in Java2, and thus a Java runtime environment is necessary to execute the application on the host operating system.
LDAP (Lightweight Directory Access Protocol) server	LDAP is a protocol to centralise the management of users and access control. It is an open standard with both commercial and open-source implementations. In this instance a commercial LDAP server was utilised by Windchill.
LDAP editor java component	This is a Java2 based cross-platform utility to edit LDAP databases.
Add administrator user to the	This is the first step to complete before the server

new LDAP directory	is installed as it enables an administrative user to configure the system for general use.
Windchill 6.2.2 foundation server (with patches for 6.2.6)	This is the base PLM system that is installed once the pre-requisite items are installed.
J2EE application server	The Windchill application server is a J2EE application, and thus requires the availability of a J2EE compliant server to run.
Tomcat Servlet engine	The system uses servlet technology to distribute the Java based business functions of the Windchill PLM system. Thus the Tomcat servlet engine is required to provide some of the business functions of the server side Windchill application to HTML based clients.
SQL scripts for populating the database and generating the data models	The database schema of the Windchill system is distributed in a large SQL file separate to the database. Once the database server is installed, the SQL script can be run to populate the database with the schema and test data.
Populate LDAP server with test user entries	Once the database and Windchill servers are installed the final step to running the system is the creation of normal (non-administrative) users to be able to login to the system and establish projects and populate the PLM system.
Windchill Info*Engine B2B connectivity tool	The info*engine system was created by PTC to enable business to business connectivity. However the system was not utilised in this project, it was installed purely because it is a pre-requisite for running the PLM.
PTC's CAD visualisation package and local runtime.	The visualisation tools were installed to aid the sharing of geometric data from the Pro/Desktop system. It is a post-installation add-on tool.

Table 5 Windchill configuration

The setup was complemented with the installation of the protégé system for the KM component. Full installation procedures are included in PTC's documentation available on the web server <http://windy1.cranfield.ac.uk/> . Table 6 summarises the modification made to the system

Implementation	Description
OWL data type added to Windchill system.	The data type was added via the Windchill web based control panel. Once added ,t eh invocation of an OWL document automatically runs the Protégé ontology editor from within the Windchill system.
Document Object modified to	The serialisable attribute ensures that each

include OWL repository with the following attributes Serialisable, Revision controlled and Manageable (via workflow).	new document created in the knowledge base has a new item number, the revision controlled attribute enables the creation of multiple versions and revisions of the same document (creating an auditable history). The manageable attribute enables the ontologies to be integrated within larger workflows as individual processes.
Data model updated via Rational Rose and Windchill java classes recompiled and overwriting the older JAR (Java Archive) files.	
Windchill explorer added a menu item for Protégé to run the knowledge base editor when the knowledge base is selected in the project.	In case a user wanted to invoke the ontology editor without creating a new ontology or opening an existing ontology, a menu shortcut was added to the Windchill system.
The workflow editor was used to model some of the test enterprises' data flows.	the implemented workflows can be seen in Appendix (b)
The workflow engine's 'method engine' function was used to invoke the Protégé editor whenever an object in the knowledge base was about to be edited.	This was not a requirement as the inclusion of the OWL datatype, in theory, automatically invokes the protégé editor. However for the sake of completeness this function was added to the workflow engine editor.
The Bill of Material and assembly functions of the Windchill system were disabled,	To enable the use of the OWL knowledge base functions. The implemented BoM system is described in the next chapter and the source code can be found in the Appendices.

Table 6 Implementation of KM on PLM

The data model is accessible using the Windchill explorer application, and the serialisable and manageable attributes for the ontology components allow multiple users to access and update the data in real time remotely. The data management components of the Windchill systems automatically take care of the storage and revision control of the objects. The ontologies were also usable from within workflows as individual processes in a chain.

In the test evaluation, the project wizard created the Windchill project, and the ontology was associated with the Protégé application (knowledge base editor) for editing the ontologies.

Pro-desktop models were uploaded to the database as individual components, and assemblies were created using Windchill's built in assembly management system.

The knowledge management system was installed on each client using an online installer, and alternatively a solution packaging the knowledge editor into a java package to be loaded online was also tested (requiring no local installation).

The ontologies themselves functioned in exactly the same manner as on the other application methods, with the added overhead of being loaded on top of the Windchill system.

The actual data models created are detailed in the next chapter in addition to the workflows which were modelled. This implementation used as much of the Windchill systems' built in functionality as possible. The complex nature of Windchill's info*engine for inter-enterprise collaboration meant that it was not possible to modify it for working over the JXTA protocol which would have made it compatible with the the 2-tier architecture that was developed as part of the research to solve the problem of inter-enterprise collaboration. The inherent incompatibility of Windchill with the flexible and feature-complete 2-tier architecture was due to the following factors:

- ❧ info*engine relies on a complex persistence engine based on transactions rather than the simple yet flexible 'object based' update and modification rights in the two-tier architecture;
- ❧ info*engine's access control management is via Windchill's LDAP server, whereas the two-tier architecture uses a simple XML file based access control system;
- ❧ The two-tier architecture has a very low subset of functions for communication and authentication, whereas info*engine, which was designed to integrate with ERP and CRM systems uses a complex one-one integration and a complex transaction registration system, that is a necessity when multiple transactional systems collaborate;

- ❏ The two-tier architecture supports the use of feature rich knowledge models which are updated and modified in real time (with immediate persistence across the collaborative partners). ON the other hand, info*engine breaks down immediately if a data-model is changed at any node of the collaborative enterprise and thus requires constant maintenance of the fragile 'one-one links' between systems; and
- ❏ info*engine works on the principle of batch-transaction processing, whereas the simple 2-tier architecture works in real-time.

The above details illustrate the inadequacy of rigid centralised transactional systems when it comes to distributed and collaborative enterprise integration, compared to the supremely simple and flexible two-tier architecture. The above issues are not a specific issue with PTC's Windchill, but symptomatic of all complex transactional client-server systems that have to maintain consistency in an operating environment of large records of transactions.

4.3.2 Open source client server applications

The first system for comparison with the commercial PLM was an open-source alternative, working on the same general principles as Windchill. The key points of difference between the two, of course is the zero capital expenditure required for the open-source system, which thus gives it a hypothetical advantage in distributed collaborative environments over the commercial system. The main advantages of the open source system can be said to be:

- Free implementation at all nodes of the Virtual Enterprise, for effective data integration; and
- Rapid acceptance by smaller Tier-2 and Tier-3 companies that cannot afford the expense of PLM systems, thus creating a groundswell of users for inter-enterprise collaboration

To what extent these potential advantages of open source systems over commercial systems overcome their relative weakness in vertical integrated applications and lack of corporate service and support are summarised below.

The applications used in this project include the Protégé ontology editor and Sun's JXTA peer to peer protocol. Both of these tools offer the best usability, portability and cost/performance capability within their niches, and when allied to a scalable open source object/relational database, offer enterprise level performance for zero capital expenditure, and low customisation cost. Moreover they free the customer from the clutches of software vendors and the vagaries of obsolescence. Other open-source tools used in this work include a solid modeller that has to be able to output STEP files and an application server tool that forms the foundation of the open-source PLM solution. More information on each application and comparison of the choice of tools and rationale for selection are below:

4.3.2.1 Open source databases

A Number of databases were considered for the Open Source implementation, these include:

MySQL: was utilised due to its stability, ease of use and close integration with the PHP scripting language that the open source PLM was written in. MySQL is the fastest and smallest of the open source databases that are still viable for large and complex data structures.

SAP-DB: This object-relational database developed by SAP and released into the open-source domain as most of their clients bought into ORACLE. The database has performance issues and is prone to crash, also there is little support for it in the OSS world, meaning that updates and bug fixes are rare.

PostgreSQL: is a large object-relational database management system that includes stored procedures and functionality at the same level as enterprise DBMS like Oracle and DB2. However the system has a large overhead on computing power (memory and processor), and is slower than MySQL. It was not utilised because it is optimized for the central management of very large databases. Whereas in the distributed database model implemented in this research, the speed of database on low end computers (not large server farms) is critical, and in the distributed model, each node only has a subset of the database and thus not requiring the ability for holding terabytes of data.

In summary a wide range of databases are available in the open source domain. Within the context of this research, a lightweight easy to use database is most appropriate to create easily distributed applications among VE participants.

4.3.2.2 Open source Web-servers

AOL: Is a high performance web server initially developed as a commercial tool. It was not utilised because integration with databases and scripting languages was not as good as apache below, and it has a smaller number of active developers making it not very future proof (by OSS definition). AOL is also not implemented on non-UNIX systems.

Apache: Is the most widely used web server in the world and open-source too. It is closely integrated with all the necessary components for an enterprise system and is implemented on windows and UNIX platforms.

Tomcat: Is also developed by the apache foundation that implements the Apache webserver (above). However Tomcat is the model implementation for a Java Servlet engine and EJB server, and was thus used in this project to serve the servlets alongside the apache webserver. Note, tomcat can also serve standard apache webpages, but for the sake of simplicity the PHP based applications were served on Apache and the JSP based applications were served on Tomcat.

JBOSS: is a complete implementation of a J2EE application server and can serve JSPs <http://java.sun.com/javaee/> . It provides web services, component models, component management and a wider set of communications Application Programming Interfaces (APIs) than Java2 Standard Edition (J2SE). However its overt complexity and our limited needs mean that tomcat (servlet engine) was sufficient for our requirements that do not need a full J2EE server.

The test open source PLM makes use of the simple and widely used apache and tomcat servlet engines. They are sufficient in scope and functionality for both the small test application and for wider general use in client-server systems.

4.3.2.3 Solid modeller

To generate the models that will be persisted on the PLM systems, a number of free or open source modellers are used. The reason for this choice is to enable a unified platform for model generation among collaborators (and thus having a platform that is both easy to use and cost free for each enterprise to use). The models are stored in Industry Standard STEP.

Pro/desktop: is a free solid modeller developed by PTC in the USA for use as part of the school curriculum. The solid modeller itself is very capable and based on the GRANITE1 modelling engine developed by PTC. It uses STEP as the native file format, and has a very low overhead for use on low-end desktops easily, making it ideal for use as a component in a enterprise collaboration portal. It can also visualise and edit many CAD formats, making expensive 3rd party visualisation tools superfluous.

OPENCASCADE: a powerful solid modelling engine that has been developed as an open-source tool from scratch, it uses STEP as its native format and is thus ideal for implementation in a open-source application. However, as of writing, no useful implementations have been made relevant to this research project, thus it was not utilised in this work. However the OPENCASCADE engine has a strong potential in the future due to its advanced architecture and OSS nature www.opencascade.org.

In this instance Pro/Desktop with its intuitive interface and easy to program feature based design system was sufficient for generating the feature geometries for the collaborative design system.

4.3.2.4 Runtime environments

The Open Source application server runs a set of business rules and functions on top of the database. These need a platform or language to interpret and operate the business rules update information on the database and manage users and persistence. The leading technologies for client-server applications were implemented to test the suitability of each technology for the task of inter-enterprise collaboration.

PHP runtime environment: Perl Hypertext Processing is a very widely used and simple scripting language that combines HTML markup for visualisation with perl code for functions and methods. The PHP runtime engine is closely integrated to MySQL and apache making it an ideal and easy to implement/use collaboration portal for web-based applications.

Java servlet runtime environment: As an answer to PHP and Microsoft's ASP, Sun Microsystems created JSP, which basically implements Java applications via webpages, where JSP code is combined with HTML for creating simple web-based applications. JSPs are served through Tomcat and similar servlet engines and are packaged in a zip format for ease of distribution.

J2EE engine: Not used in this research due to its complexity of implementation, J2EE allows the creation of large centralised applications through the use of extensible and easy to maintain Enterprise Java Beans (EJBs). J2EE engines include the open source JBOSS and commercial IBM websphere.

ZOPE engine: Developed in the US for the US military, Zope is a python based application server combining a webserver, object-oriented database engine and a zope runtime environment for running the applications. The system is very highly scalable, and is widely used due to its integrated nature, included management tools and excellent reliability. In this research a zope based application server was tested (see PLONE below).

The runtime environment used in the system is wholly dependent on the application systems (see below) used. The pros and cons of each runtime environment is beyond the scope of this work as the use within this thesis is limited to the OSS PLM system test.

4.3.2.5 Application Systems

Applying the above runtime environments, a number of open source document Management systems were modified and used in the PLM role to verify the usability of free open source systems in the role of traditional PLM systems.

TUTOS system (PHP) (www.sourceforge.net) : Out of the many base PDM systems evaluated this was chosen for offering a complete PDM and project management solution that is modifiable very easily through configuration files. The system implements version control, access control, auditing, projects, GANTT charts, tasks and shared calendars for project employees. System lifecycles via workflows that can be configured simply. The system is very robust and was tested with a large base set of project data (10 Gb) without problems. The system was originally developed to manage software projects, but was modified for this research project for engineering product development through the substitution of software development nomenclature to engineering specific terminologies.

PLONE base system (ZOPE) (www.plone.org): This system is an implementation of a document management system on top of the zope engine, and was modified into providing PLM functions including version and revision control, workflows for the engineering processes, user based access control, shared tasks and project calendars. The system however lacked some key features found on TUTOS and Windchill and that is the effective management of Engineering Changes and issues.

Egroupware (PHP) (www.sourceforge.net): This is similar in principle to TUTOS, but lacks the engineering specific functions.

PHPCollab (PHP) (www.sourceforge.net): Once again, one of a large number of groupware systems on the market, but lacking the features necessary to be modified into a PLM type system.

Track+ (servlet) (www.sourceforge.net): is a complete project management tool that lacks some key engineering specific functions including version control of engineering documents and engineering change management functions.

ProjectOpen (J2EE) (www.sourceforge.net): is the definitive open-source project management tool, and comes suitably written for J2EE implementation. However, what the system provides in terms of scalability and robustness, it fails to provide in terms of

ease of modification and inclusion of extra functionality. It was thus not utilised as it would have required a team of software engineers to implement a simple add-on for engineering change management or configurable workflow.

DotProject (PHP) (www.sourceforge.net): is an up-and-coming project management / PLM system, but lacks the depth of PLM functions found in TUTOS to be a real contender at the time of writin. However the system is worth mentioning because, like opencascade, it has a strong potential for the future.

TUTOS was the most mature system usable for product development purposes when this research was carried out and was thus used within the test PLM system implementation. It was the only system with full support for projects and workflows.

4.3.2.6 Systems construction

With the building block choices being made, the PDM specific application can be constructed as seen on Figure 5. The bottom layer consists of the database. On top of this layer the two open source gateways, one is the Apache web server for serving static and PHP based web pages and the middle tier is the Tomcat servlet (in lieu of the JBOSS J2EE server) engine which serves the Java servlets (JSP). These three layers (database, web server and application server) form the server side of the system.

On the client side, the user has three main applications i.e. (i) the web browser through which interactions with the PLM system are carried out, (ii) the Protégé Java applet that allows the user to query and manage the knowledge base and (iii) a CAD system to enable the user to create and manipulate the STEP based models held in the PLM system.

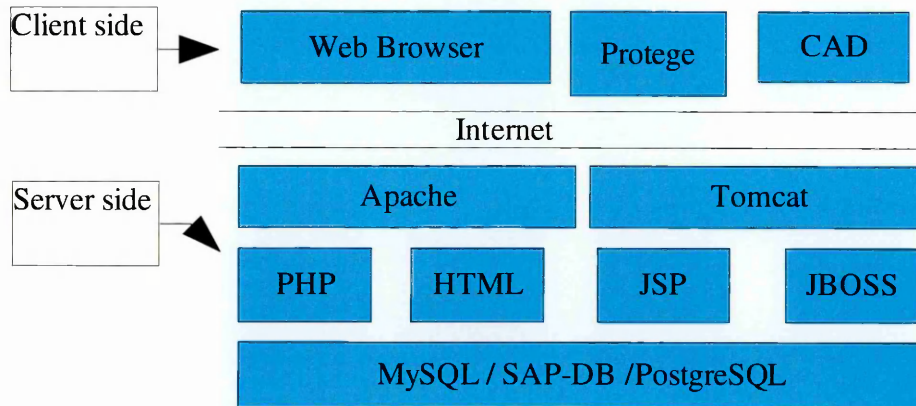


Figure 5 High level architectural overview of open-source client server system and user interface

This user interface ensures ease of use for the user because the web-browser and its own cad system are already familiar to the user with the only 'new' tool for interaction being protégé. This extension also includes the management of STEP entities within the PLM system in an intelligent and object oriented manner. Knowledge is stored as objects within the modified PLM system. This was integrated with the lifecycle management and workflow functions offered by the PLM system.

The individual third-party components of the system are:

MySQL database engine <http://www.mysql.com>

Perl Hypertext Processor (PHP) runtime engine <http://www.php.org/>

TUTOS base application <http://www.tutos.org/>

Apache and tomcat webserver <http://www.apache.org/>

Solid modeller (Pro Desktop) <http://www.ptc.com/>

Installation of the basic system was as follows:

1. Installation of the Apache webserver: Setting the open port to port 80 for HTTP requests. Setting the document root to WEBSERVER for managing the applications easily from a single source folder
2. Installation of the tomcat servlet engine to manage the servlets: Setting the port to 8080 for web access. Setting the document root to WEBSERVER in line with apache's setting for a single integrated management of the PLM application

3. Installation of the MySQL database server: Setting the management port to 3303. Installing the management software under the WEBSERVER folder to be accessible via a web browser. Creating a database 'PLM_SYSTEM'
4. Installing the TUTOS server under WEBSERVER: Setting the sever to use database PLM_SYSTEM accessible through port 3303 (as defined above). Installing the PLM modified PHP files for the CAD management and ontology management components
5. Copying the Protégé system's Java runtime files (Web Archive) WAR under WEBSERVER and registering it with the Tomcat servlet engine for use with the PLM system

Technically, the base application of the system already defined the functions for creating projects, document management system, version control and task management for groups and individuals. The additional required functions to create the ontological knowledge management system were:

1. Integration with the Ontology management system a Protégé Web Archive distributable using servlets (WAR);
2. Native storage of knowledge base objects on the system (managed by Protégé);
and
3. Creation of workflows for tasks and knowledge management

The above functions were very simply implemented. The protégé system was loaded as a Java archive onto the webserver making it accessible in real-time from any computer that connects to the server via HTTP. Similar to the Windchill implementation, a project was created with tasks assigned to the project. A project repository defined in the system contained the base ontology which was uploaded to the server by the administrator. The ontology was made accessible to the project users and defined as revision controlled and as having a lifecycle 'beginning-life-end'. The workflows were simply defined as system tasks, with each task being assigned to a user, and having a set of tasks that have to be completed before further work can commence. End user operation was identical to the Windchill system, being via a web-based user interface and using the protégé editor for adding data to the knowledge base.

Inter-enterprise integration is possible using the proprietary built in replication service in MySQL database. However this is a complex and cumbersome system compared to the implementation detailed below for the P2P system.

4.3.3 Application Service Providers

Application Service Providers (ASPs) can be set up in two ways, either by a large 'controlling' enterprise, or through independent third party hosting. These services intend to provide the same utilities as enterprise level systems but in a non-enterprise specific service. ASPs offer project and product data management vaults where the VE administrator can customise the third-party portal for their own use. It also offers opportunities to contact companies already listed on the ASP who might join the VE if they had required services to offer. The ASPs generally use software very similar to the PLM solutions like Enovia and Windchill. However, they offer them as low cost solutions to some customers with specific needs, which cannot be met by purchasing their own server software. These include:

- ⌘ Reduced cost for the enterprise as maintenance and backup is delegated;
- ⌘ Increased opportunities if customers and VE initiating enterprises seek out partners through the portal;
- ⌘ This is an "egalitarian" system where no one enterprise controls the VE server and data; and
- ⌘ It sets down de-facto standards for data exchange, to which other enterprises in the same domain will adhere to in order to join the network of enterprises.

There are of course some fundamental disadvantages to the use of ASPs, and other centralised systems, for product development, and these include:

- ⌘ The bandwidth and server bottleneck problem associated with centralised services. The security fears of intellectual property rights being compromised;
- ⌘ The potential risks of downtime and data losses in an "uncontrollable" environment and the liability issues associated with it;
- ⌘ The difficulty of creating direct interfaces from the enterprise system to the ASPs portal; and

2. The exact functionality required for the VE may not be available from the “generic” ASP.

There are already some ASPs operating in the automotive and aeronautical sector enabling supply companies to interact and bid openly for contracts with OEMs and then manage the project/product information on the portal. However due to the disadvantages highlighted above, the authors sought to find a third way. Whilst traditional client-server systems can operate in a collaborative manner, for example over a LAN or Internet, they are not truly distributed systems, as architecturally they are still centralised with the associated issues of centralised systems.

4.3.4 Peer to Peer Systems

Business to Business (B2B) integration is the traditional method for companies to collaborate and create VE networks. This area includes Electronic Data Interchange, XML based messaging and portal solutions. Integrating 15 companies together on a one-to-many basis where a single repository manages the project knowledge and workflows would need 14 separate mappings to achieve such integration. If we go further and try and make one-one schema mappings between 15 collaborating enterprises then 104 separate one-one mappings must be made! This clearly illustrates the impracticality of using centralised systems for de-centralised collaboration between any more than 3-4 participants.

Peer to Peer (P2P) applications address the needs of de-centralised organisations to collaborate and share knowledge regardless of geographical location. The principle of P2P has been around for a long time, and is today implemented in a number of applications such as instant messaging and file-sharing (www.GNUtela.com). There are already a number of P2P PLM in existence. Primarily aimed at the lower end of the market. The two commercial applications are AutoManager workflow from Cyco (www.cyco.com) and Columbus from Oasys Software (<http://www.oasys-software.com/>). The latter is available for free, and aimed at AutoCAD users within the construction sector. However they are crude solutions relying on the underlying file system and adding some "meta tags" to files for version control. There is no workflow or process management implemented and access control is via the standard operating

system access control functions. As an example of what can be achieved, Alibre is a P2P CAD/PLM and collaboration tool in one. It uses the STEP standard and combines low cost with rapid configuration please see www.alibre.com for further details.

The advantages offered by P2P applications over the centralised systems introduced above are:

- No single point of failure, the network is alive as long as one peer is on-line;
- Distributed sharing of bandwidth, storage and processing power so the system becomes more powerful as more collaborators participate;
- Lower running cost due to the lack of servers or high bandwidth central nodes; and
- Maintaining individual control over proprietary parts of the collaborative knowledge base.

P2P groups can be used to create profiles of the peer, and also more importantly of the peer's list of contacts within different domains. These profiles can be used within the network to search for and assess people's competences, interests, and memberships of trusted groups, and can aid in the construction of new relationships based on commonalities and third party assessments. The above methods facilitate the creation of ad-hoc collaborative groups based on mutual trust after enterprises have gained some experience of using the collaborative system (and hence good ratings by peers). Such a portal can achieve a dynamism of its own, and such trust based peer reviewed portals have already proven themselves in e-commerce sites like www.ebay.com.

There have been a number of issues that reduce the performance of the system using pure P2P architecture. The lack of indexing and routing services in P2P degrades the peer discovery and query functions www.jxta.org. To overcome this, the advantages of client/server systems in indexing were combined with the independence and interoperability of P2P systems into a hybrid system where “super peers” act as peers to the extended P2P network and as a server to the enterprise's internal peer network. In addition rendezvous peers can be assigned to manage some of the peer information assigned to particular peer nets or projects. This hybrid has been shown to have the best

potential for high-performance de-centralised services. Full details of the systems are below under 'implementation'.

Finally, there are, as always a number of competing peer-peer messaging standards available on the market (see Table 7), and in order to ascertain which standard and implementation would be most suitable for the problem at hand a number of different tools and standards were evaluated in depth as shown in Table 7.

GNUtella	is an open-source framework for peer-peer communications that is programming language independent. The system is suitable for instant messaging and conferencing, however it does not implement advanced security and shared-profiles that aid the construction of advanced virtual enterprises where sensitive commercial information is shared.
JXTA	This system was the chosen standard, and has been open-sourced by its creator Sun Microsystems. It offers all required features for inter-enterprise collaboration.
Microsoft messaging systems	These systems are commercial, proprietary, and are controlled by a central server. The system is usable, however its lack of openness makes it a bad choice for enterprises to achieve vendor independence and low cost.
Jabber	is another open-source messaging standard that is being widely deployed. Once again it relies on a series of central servers and, like GNUtella, lacks the required features for enterprise collaboration.
Kazaa type systems	which are centrally indexed file sharing systems. Once again it offers good performance and can offer the necessary authentication needs, however it is limited by the fact that central control of the server means that one of the parties or a mutually trusted party must be in control over the system.
Torrents	The torrent phenomenon is relatively new to the world of networking and uses IP tunnelling to share files between peers via a centralised indexed service. However torrents are technically very limited and have no dynamic updating of the indexes or access management. They are useful only as low overhead downloading servers.

Table 7 Comparison of P2P protocols

4.3.5 Proposed “medium” for VE

The implementation was kept as simple as possible. The simplicity of the architecture is itself a bonus and enabled the author to concentrate on the value adding aspects of the project. As mentioned previously, the flexible Protégé ontology development and user interface environment is used. The back end consists of the open source MySQL database with the Java database connectivity (JDBC) connector to Protégé. The knowledge base ontology is defined in a Clips file and the instances are stored in the database. Connectivity is achieved using an open source implementation of JXTA open standard P2P network protocol (www.jxta.org). The choice was made because JXTA implements a unique but anonymous identification mechanism for peers and for rendezvous peers. As well as “advertisement” implemented for all peers that give information on the one peer to other peers. Rendezvous peers can act as managers for peer groups and store the peer advertisements for the group for distribution to other P2P networks. Implementation of JXTA is in Java 2 standard edition, an extension to enable RDF queries and ontologies to be shared over P2P is used to share the Protégé knowledge base.

Figure 6 illustrates the proposed hybrid super-peer net architecture for inter-enterprise collaboration in egalitarian Virtual Enterprises. The yellow bubble shows an exploded view of the internal peer-net of the enterprise. Inside this internal-peer net there are a number of functional sub-systems as well as internal users. Each function within the super-peer net is a standard JXTA node (including the super-peer itself). The actual difference in the function of each node is down to the configuration selected for that node. This allows absolute flexibility for dynamically changing modifying adding and removing nodes within the internal peer net. It is also possible to make any one or many of the internal peer net nodes an external peer-net member via a simple configuration change. The functions used in this instance were the protégé knowledge base (this is physically co-located on the same machine in this example, but for the sake of topological accuracy has been included as a separate node), Each “F” peer is an end user accessing the internal peer net (and the external super-peer net via the super peer). The PL node has the internal workflow functions of the peer net that are accessible by the wider super-peer network via the super-peer.

Outside the peer-net is the external peer-to-peer environment which interlinks all the super-peers from every collaborator together. Note that each super-peer is connected to every other super-peer making a multiple redundant connection. In addition one, or many of the super peers within the network act as cache and rendezvous peers to enable peers or super-peers with intermittent connections to effectively participate in the peer network by having their internal shared knowledge bases stored on cache in case they are not available to serve other peers that need to access their knowledge. This enables peers to connect to each other and query for and manipulate knowledge on different peers on the internal and wider peer-net transparently.

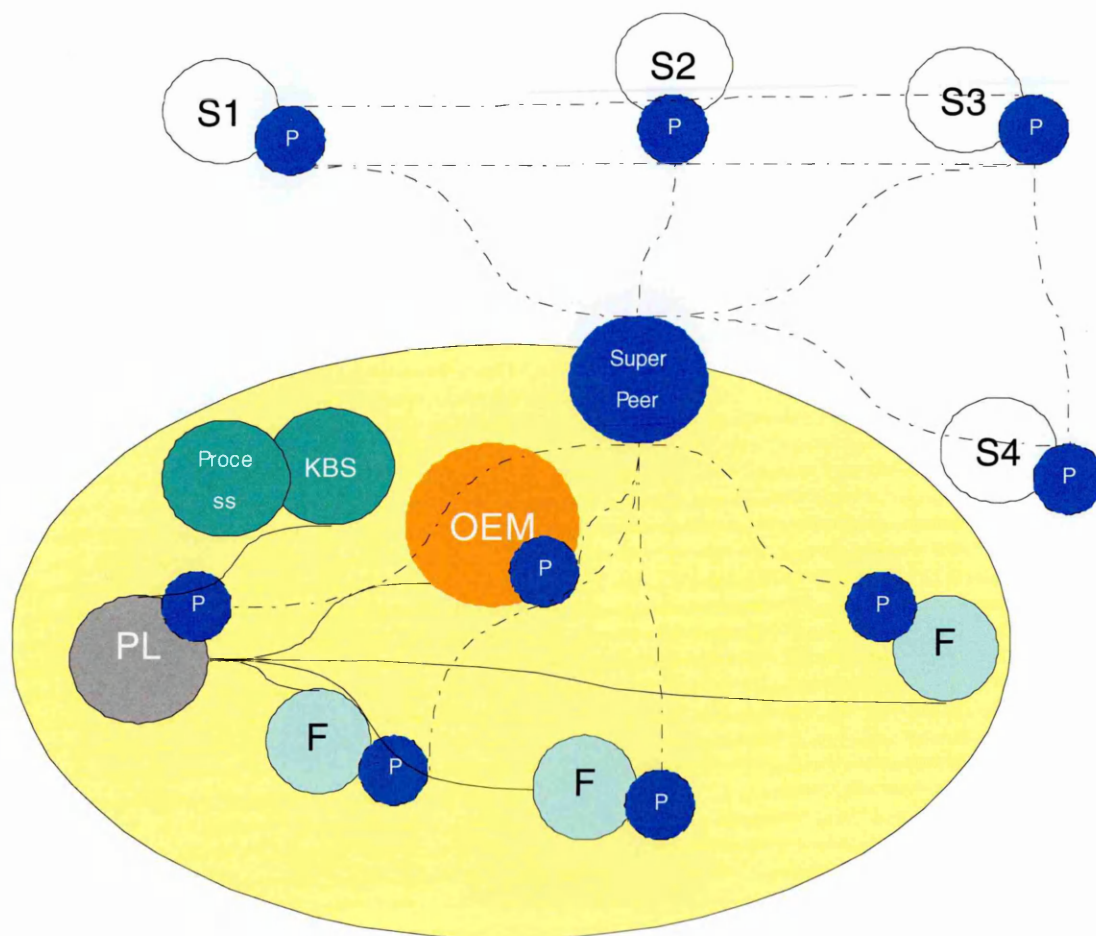


Figure 6 Super peer architecture showing the interaction within internal peer nets and interaction with external peers

The architecture was set-up with a super-peer net in order to create a number of virtual servers at each of the project participant's sites. These super peers are visible as peers to

both the other super-peers and the internal peer net which they serve. They also act as the default rendezvous peer, and peer information is shared between the super-peers in order to improve the redundancy and query speed of the system. Figure 6 illustrates the topology of the system.

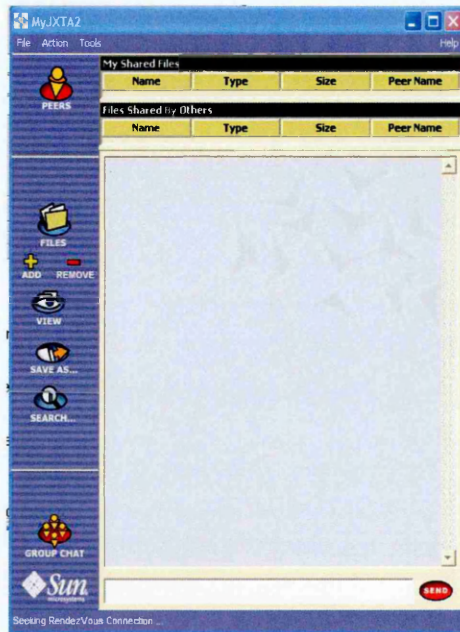


Figure 7 JXTA configuration window and user interface

Figure 7 illustrates the JXTA user interface. The window shows the user interface from where queries and project management of collaborative groups are readily implemented by the JXTA protocol. The systems' settings enable enterprises and users without static addresses to collaborate using dynamic addressing, and this flexibility, as well as the users' ability to work offline (that cannot be done with web based systems) empowers users in all possible scenarios in a VE.

System components are:

- ✧ JXTA2 Java binding;
- ✧ MyJXTA2 Java SWING implementation of the user interface for file/database sharing and messaging; (Figure 7)
- ✧ XML file containing peer lists and access control rules;
- ✧ Protege knowledge management system;
- ✧ MySQL database containing the knowledge base;

- ✎ JXTA RDF parser, for querying the knowledge base via the standard JXTA interface;
- ✎ Rendezvous peer information (XML file).

System installation

1. Install the Java2 SDK (latest version);
2. Install Protege knowledge base and MySQL database+JDBC driver;
3. install MyJXTA2 with Java SWING user interface;
4. Install add on for RDF processing;
5. Copy access control XML file to rendezvous peer folder;
6. Make the MySQL database shared with the JXTA system;
7. Store all flat-files used in collaboration, under the JXTA shared folder; and
8. Install OpenSSL add-on for encrypting the local folders where data is held (this works best under a Linux system such as SuSE, however it is possible to make it work on windows).

System customisation

The system functionality was customised relatively easily compared to the client-server systems listed above. The basic steps for full peer-peer functionality for the collaborative portal were as follows.

1. Create knowledge base to JXTA search engine link via RDF. JDBC link from the knowledge base to the system (MySQL's JDBC bridge). Run the database and system, then the JXTA server will be able to run queries on server both locally and remotely. See www.mysql.org for details on installing the JDBC connectivity.
2. Enable SSL security for the system. Encrypt local folders that contain the shared knowledge base, whilst defining a separate key store for managing access control (please see www.openssl.org for detailed installation instructions).
3. Add the knowledge base to the list of shared files on the system.
4. Create the list of users in XML with their respective access control levels. See www.jxta.org for full documentation of the access control levels.
5. Create and distribute the shared keys via any medium of secure communication.

The above steps (including source code and references detailed instruction where listed) enable the basic function of communication and authentication between the participants of the collaboration. Its simplicity of theory as well as ease of implementation was arrived at after working for extensive periods of time on traditional systems for inter-enterprise collaboration (namely the big client-server systems). It is therefore worth mentioning the extreme simplicity and usability of the components above vis-à-vis the client-server systems that were developed or utilised at the beginning of the investigation.

As can be seen from the user interface above, its operation is as simple as the popular instant messaging software, but allied to powerful access control and knowledge management functions, the system becomes an extremely useful and powerful tool for engineering collaborative enterprises.

4.3.6 Summary of communication architectures

The results of the systems comparison were a combination of empirical performance tests, with a comparison of system features vis-à-vis the project requirements, see Table 8. For the test, the server was located in Southern England, and users concurrently created the new project, generated new concepts and optimised the design gradually from two remote locations in addition to users at the main site.

Application	ontology	VE setup	Deployment time	Lifecycle management	Total cost of ownership
Windchill	RDFs ontology in document container, access via windchill exp	B2B integration via Info*engine (one-to-many integration), manual	1 month for small project.	Graphical workflow	Server and client licence, implementation and maintenance
Open PDM	RDFs ontology accessed via Protégé applet	B2B integration via XMLRPC (one-to-many integration), manual	1 week	PHP workflow	Implementation and maintenance

Pure P2P	RDFs ontology in java application	Peer to peer, automatic discovery	1 day	Version control and rule based system	Training on P2P and knowledge app
SuperPeer net	RDFs ontology in java application	SuperPeer automatic discovery	1 day	Version control and rule based system	Training on P2P and knowledge app

Table 8 Comparison of key features of tested system

Table 8 compared the key features requisite for a collaborative project environment, these were:

- **Ontology:** how easy was it to enter, query, and reuse both the ontology and domain knowledge within the collaborative group? Since this is common to all the tools, the only difference was the way in which the ontology was stored within the different systems.
- **VE setup:** how rapidly could the VE be set up and start to operate on all the remote sites?
- **Deployment time:** The time to customise and deploy the entire project including the project management framework (access rules, task lists, workflows), project ontology (sub-ontology selection from the generic ontology), and communications setup.
- **Lifecycle Management:** The ability to control the state of a document, manage versioning and history of the data.
- **Total Cost of Ownership:** The costs of licences, implementation, system integration, training and maintenance.

All ontologies deployed shared a common back end. The applications were tested on identical hardware with identical configurations and internet connections. The apparatus were Dell Xeon workstations with 1GB RAM, 15K rpm disks and 1mb internet connection, the operating system was Windows 2000 on all machines.

The test suffices for a small scale pilot. The test, see Figure 8, simulated a small scale VE setup through from initiating the VE to running the system within the collaborative network. This setup has a typical VE with a large number of small servers distributed within a common network; a small number of transactions are processed at each node.

The results obtained are verified by the test results on JXTA obtained by (Antoniou et al 2005), who tested the data throughput of the Java 2SE implementation of the JXTA standard.

Figure 8a. Time to update instance

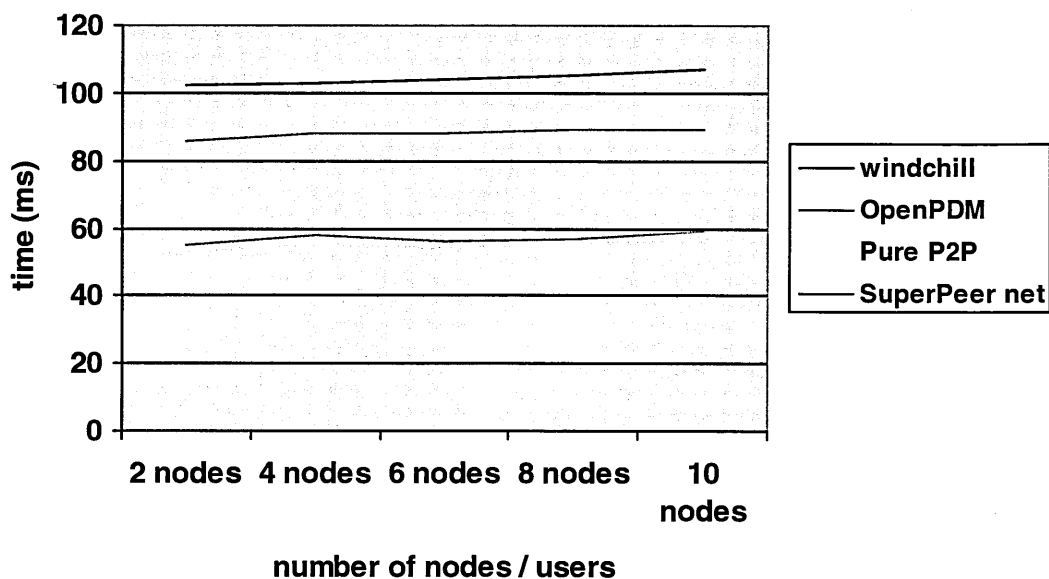


Figure 8b. Simple query (single term search over the entire KB)

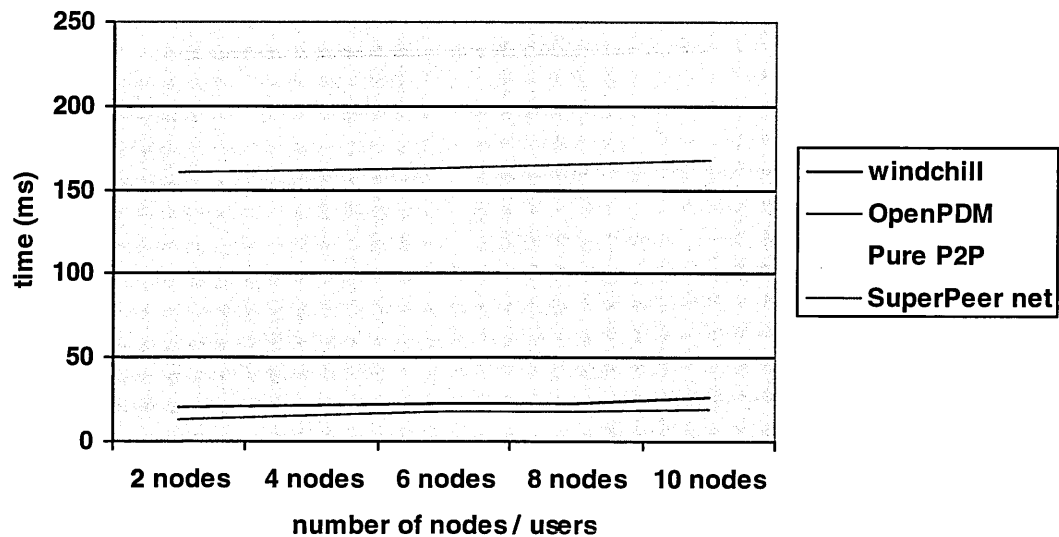


Figure 8c. Complex Query (generating an inferred hierarchy of the KB before processing a result tree)

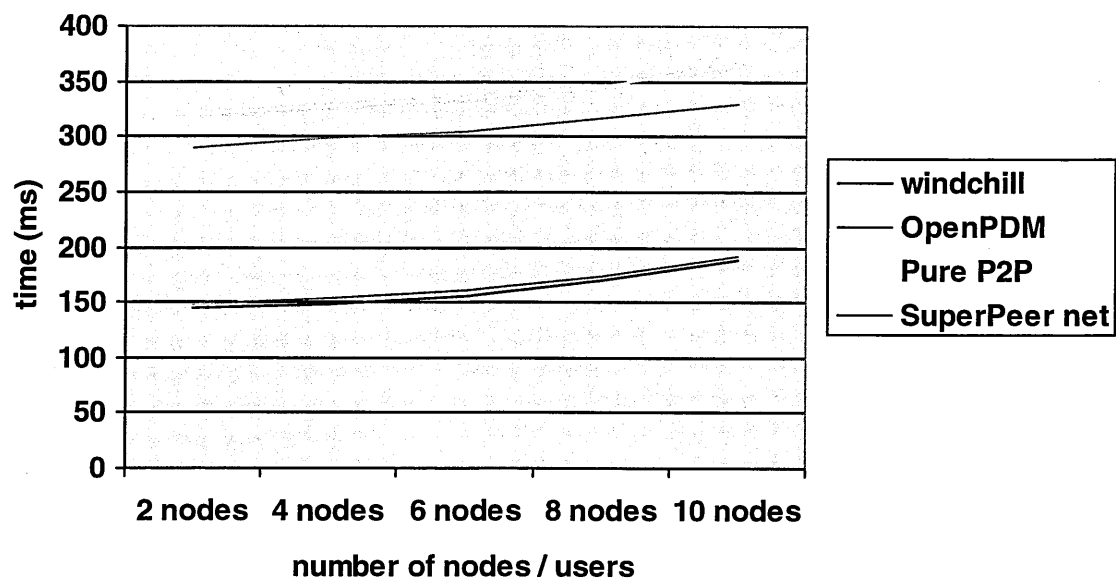


Figure 8 Query speeds over different types of topologies

As stated earlier the number of users was not sufficient to fully test the scalability of the applications; however the distributed nature of the applications means that the number of concurrent transactions are much lower in P2P type applications than centralised programmes. The Peer based applications showed a base performance lag when compared to the client-server applications due to routing of peer based systems without the help of a centralised indexing service. Otherwise the performance of the systems in practical use showed no noticeable differences. In theory peer-based systems will scale to larger systems better than centralised systems because it inherently shares the load between hundreds (if not thousands) of independent 'servers' as opposed to a single central server. However this aspect could not be verified in this project due to the lack of a large enough user base to fully test performance under those conditions.

4.4 Summary of proposed methodologies

There are a plethora of modern techniques available for implementing data management systems for distributed environments. However there are notable operational differences between the traditional client-server and the new generation of peer-peer systems. The development of technology over the past 20 years has created a level of processing power, storage capacity and network bandwidth at the lower tier of the computer hierarchy (the humble personal computer, or small company server), that enables it to effectively act as not only a client but a server, or a small server of a constellation of servers operating in an egalitarian computing schema between multiple enterprises that are operating in an egalitarian Virtual Enterprise.

Overall the Peer to Peer architecture using the super-peer net configuration (Figure 6) fulfils the criteria for de-centralised inter-enterprise collaboration in a heterogeneous network of egalitarian enterprises more completely than the traditional PLM based solution, or the Open source based client-server system. The peer to peer network enabled enterprises to download install and configure the JXTA system in a matter of minutes. Once the configuration information was distributed between the peer networks defining the borders of the virtual peer network, even enterprises with intermittent connections over a modem shared the same ability to serve their knowledge base with their collaborating network as a large Tier-1 company with dedicated lines and a

registered Domain Name. Furthermore the simplicity of the knowledge management system and its uniform user interface across all computing platforms (Windows, UNIX, MacOS, Linux) ensures common understanding between peers using totally different computing systems. The ability to have participation in multiple VEs using the single client and knowledge base further leverages the ability of smaller enterprises to work with multiple OEMs and reuse their internal knowledge in multiple dis-jointed projects.

The commercial PLM system, although highly scalable and offering excellent workflow configuration utilities, was very complex to set up and modify. It is also prohibitively expensive and its business to business connections were impossible to effectively use in many-to-many inter-enterprise mappings. The cost of the system also places it beyond the abilities of smaller enterprises that form the backbone of all engineering projects (whether managed by a large OEM or developed jointly by a large number of small enterprises). The knowledge management system functioned well within the PLM system, however functionally the same was true of the Open Source implementation that did not incur the massive complexity and up-front cost of the commercial PLM. This chapter laid out the infrastructure of the system. The system described the choices of format and medium for collaboration for effective virtual enterprise collaboration. Now the light blue shaded areas of Figure 3, the actual collaborative ontology for classifying and storing the project and product knowledge will be defined in the next chapter.

5 Universal Ontology for Engineering Knowledge Management

Once the mediums and formats used to establish communications between the collaborating partners was finished (see chapter 4), there was thence a need to do something with this collaboration. The aim of the entire research was, to recap, help collaborative enterprises working in distributed and incoherent environments to work together rapidly and efficiently from the smallest participants to the large OEMs. The second major component of the research was to establish formal working methods and systems for persisting, managing and using the huge amounts of data and information generated and inferred in a typical product lifecycle (stored in the light blue shaded tier in Figure 3), and present and share it in a way that enables the collaborators to infer the correct interpretation and hence enhance the knowledge base of the collaborative project as well as the individual participants (Figure 4).

As mentioned earlier, in the author's opinion, the evolutionary, efficient storage of knowledge in easy to retrieve and easy-to-reuse formats reduces the amount of time re-learning and searching for existing knowledge and can be hoped to aid in the efficient and steady evolution of the wealth of knowledge persisted within the internal enterprise as well as the wider collaborative Virtual Enterprise.

With this in mind, this chapter will go through a number of developments which were made to aid individual as well as collaborative enterprises in their internal and collaborative knowledge management (in this instance ArvinMeritor & Mabey&Jonhson as test cases).

5.1 Ontologies

The development of the common product development ontology is interrelated to the problem of collaboration and problems of collaborative design between enterprises of different sizes and complexity, that is, mapping between high and low content data models which results in irretrievable loss of information from the high content data model. This problem cannot be overcome traditionally by creating a mapping from one

data model to another. Instead the author has sought to create a project oriented ontology that can be created, shared and used by all parties collaborating in an enterprise in real-time (in this instance an ontology is a fully defined taxonomy of a domain of knowledge). This eliminates the problems faced when low-end suppliers collaborating with advanced enterprises face integration issues.

The data gathered from the companies was organised hierarchically to obtain a basic structure of classes, sub-classes and attributes, and then further re-organised into classes and slots (whilst eliminating duplicates). The company-specific sub-ontologies for Mabey and ArvinMeritor were constructed (see chapter 6.3.1 for details of each sub-ontology). Normalisation of the data to eliminate duplication of information was followed by generalisation to standardise the nomenclature of information within the ontology.

A knowledge base has been defined for geometry using the Protégé system as the User Interface and Java Expert System Shell (JESS) as the programming language. The complete design knowledge required for analysis is stored in the knowledge base in STEP Application Protocol (AP-224) for manufacturing features. The relationships and rules for AP-224 were originally defined in the EXPRESS language, some of the rules being re-written using the Knowledge Interchange Format (KIF) format with a forward-chaining algorithms in JESS for faster model verification (please see Section 4.3). For some of the model constraints the Protégé Axiom Language (PAL) was used, this is being used to construct queries for users.

Details of two engineering enterprises and the efforts to improve use of knowledge and information systems have been given. The area where there has been much academic development, but relatively little application by enterprises, has been knowledge management (semantic knowledge management). There are various reasons for the lack of general application of KBS/ES, but difficulty in using the systems and the lack of user knowledge of what constitutes a Knowledge management system are two major factors.

Traditionally knowledge management tools have been arcane from a user perspective and difficult to implement/manage and integrate with other enterprise systems from the administrator's perspective.

In this case the new standard Resource Description Framework (RDF) created by the World Wide Web Consortium W3C (www.w3c.org), has been utilised as the medium to structure and persist the knowledge base. An RDF knowledge base was created to hold the non-geometric data of ArvinMeritor, and compared with a small example knowledge base for Mabey&Johnson created in CLIPS. The CLIPS file was smaller (marginally) for the same number of instances of knowledge. However RDF has the advantage of being an open-standard and can be parsed, manipulated and queried by Web browsers and other widely available tools (including middleware and intelligent search engines). RDF is based on a graph-node triple (see Figure 9) to encode its subject; metadata and relationship using Uniform Resource Identifier URIs (note that the URI is global, never local). RDF is also integrated into the communication system, enabling end-to-end queries to be run over a wide distributed knowledge base (albeit, slowly on the test system).

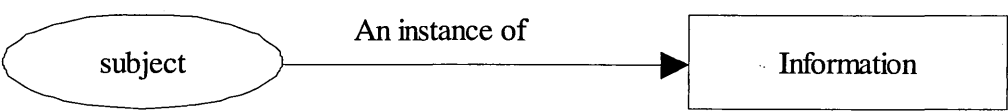


Figure 9 RDF triple

Using this notation the exact meaning conveyed by the author can be stored and processed by computer. As an example, the word “vehicle” in 2 separate RDF documents can be seen in Table 9.

Subject:	An instance of:	Information:
ford motor company	family car	Mondeo
Boeing	Airliner	777

Table 9 example domain data for an ontology

These can be held in an XML formatted document with the same name and the parser can distinguish the exact meanings associated with the 2 vehicles using the URI links, one to Ford’s schema and one to Boeing’s.

Using the above schema, it can be seen that integration of RDF based company/industry specific ontologies would be ideal both for the company’s internal storage of its meta-data (knowledge), and also as part of a shared environment where federated Enterprise Level software can make use of information from other enterprise sources and deduce the correct syntactic / ontological meaning of each instance of knowledge extracted and present the user the exact data requested.

Figure 10 illustrates a graph-node triple from ArvinMeritor, the subject URI is in an oval and the information is in the rectangular boxes. As it can be seen the subject URI can also be part of a “nested” triple, which means that the information within the nested triple has a set of contextual and descriptive information associated with it by default.

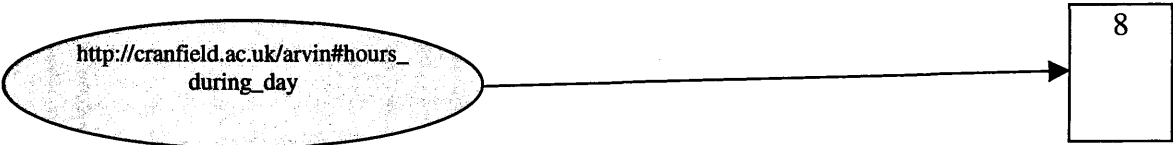


Figure 10 Example RDF-graph triple of ArvinMeritor knowledge

For example. A pilot making a query for a vehicle would receive replies about boats, aircraft, and cars. If that same pilot made a query for airliner then replies with Boeing would be returned, however. In a simple free text query replies with syntactically similar but meaningfully different terms would be received. Also the “Boeing” reply can include extra information that the RDF processor acquires from the graph-node triple which can be used as meta-data or to refine the query. Further enhancement resulted in a Web Ontology Language (OWL) translation of the knowledge base. This translation was automatic, and was carried out using the latest version of Protégé version 3.

The knowledge bases implemented for the two test cases (ArvinMeritor and Mabey&Johnson) have been integrated with STEP AP-224 Features within the Protégé

System, reducing repetitive tasks of re-entering information already stored in STEP models and also making the meta-data and Feature information available for users to query and reuse within the framework of a unified Design Data repository.

The integration of the two companies' sub-ontologies (ArvinMeritor and Mabey) has been simplified by the use of Protégé and Knowledge Interchange Format (KIF) compliant rules and functions for both the company ontologies and the AP-224 definitions.

5.1.1 Knowledge Representation Methodology

Figure 11 illustrates the data persistence problem in document oriented PLM. Documents and CAD models contain the company's knowledge asset with meta-tags to identify the basic purpose of the file. In the figure 'Kn' denotes an item of knowledge. Both documents A and B within the system contain unique knowledge. However, Kn1 is stored in both documents. In this example, a user viewing or modifying document A or B will need to be instinctively aware of such data duplication and update both documents Kn1 values. In most instances such 'simple' issues cause major problems for larger teams where the volume of documents contained in the vault is too large for individuals to be instinctively aware of it. Additionally the problem of finding knowledge, as opposed to documents, is intrinsic.

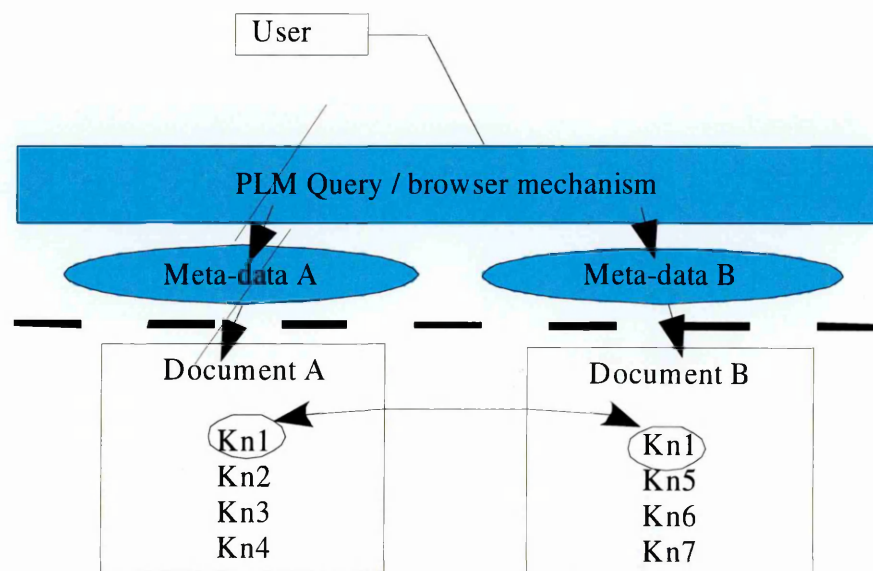


Figure 11 Data persistence in document oriented systems

The above problem can be solved by Knowledge representation, in this methodology, it is split into two identifiable components. The first one is the encoding and nomenclature component with the second component defining the terminology of the knowledge to be represented. This distinction between the encoding and terminology enables the deployment of flexible knowledge bases using different industry schemas whilst retaining the integration of the knowledge base.

This has meant that standards such as STEP have to have their schemas translated into the encoding specifications created, as opposed to the original Express schemas. In this project knowledge has been defined as:

The semantically complete definition of a domain's information that is both machine readable and interpretable.

Thus a knowledge base contains an ontology which defines the classes and their relationships, instances or objects that form the domain 'data', meta-data that constrains the data within a particular domain (transforming it into information), and Universal Resource Identifiers (URI) that allow the global identification and contextual interpretation of the information. These schemas are shared within the collaborative environment of a project involving many parties.

The resource description framework www.w3c.org (RDF) and the superseding Web Ontology Language (OWL) has been used in this application as the format for encoding the knowledge base, as opposed to eXtensible mark-up Language (XML). The reasons for this choice are the extra flexibility and 'machine-understandable' format of the RDF graph triple model as opposed to the simple 'machine-readable' XML based mark-up vocabularies. In effect, any RDF-parser can derive the semantics and context from the URI and metadata attached to every instance. OWL in addition adds the ability to automatically infer relationships and facts from the knowledge base using Description Logics (OWL-DL) in the encoding and a suitable inference engine (in this instance RACER).

5.2 Design knowledge translation

Companies involved in advanced and complex engineering work like automotive Tier-1 suppliers often receive Requests-For-Quotation from OEMs from which they must generate a quotation that involves predicting the cost and time to manufacture and assemble a product that has not been designed and manufactured yet. This is an extremely tricky and time consuming task that often results in companies making loss-making bids as not all factors were considered when making the bids. In order to alleviate this issue, enterprises sought a tool to enable them to efficiently predict the potential costs and time to manufacture new products for their OEMs. In order to do this they need a tool to enable them to persist the inferred knowledge used to transform a specification to a design. For companies making stable designs with evolutionary progress rather than revolutionary new design a simple methodology used to translate design models from the knowledge bases was developed to enable a semblance of manufacturing evaluation.

Primarily this relies on the integration of modules constructed in other phases of the project in software as well as executing the necessary procedures (or workflows). The environment itself is simply a component of the knowledge base with the addition of queries and rules specifically constructed for the display of relevant information to the user. This enabled the user to use the system as a building-block interactive tool for the construction of a constrained product model based upon earlier design iterations as well as new specification information.

The example used for the project was ArvinMeritor's product development plan, knowledge base for the request – specification – concept parts of the project (phases 0 and 1 in the lifecycle) as shown in Figure 18. The AP-224 models are defined interactively within the system.

5.2.1 Interactive versus automated design generation

There are many methods to transform a design specification into a full product model for testing and manufacturability analysis. The method most commonly used is the iterative reuse of tacit knowledge within the minds of experienced development and

manufacturing engineers. This method is the most time consuming and causes the largest leak of development and manufacturing expertise due to the undocumented nature of its application. In addition this method is rarely fully comprehensive and results depend greatly on the abilities of individuals to make the correct design decisions. This method is also not scalable as the 'domain range' for a query is limited to the experience of one individual and is therefore efficient only for smaller enterprises and one-man-shops.

The second method is the use of fully automated design generation tools. These tools are functionally analogous to the configuration generation tools used in a number of industries including telecommunications, software and micro electronics; they are also in use in pharmaceuticals and biomedical applications. These tools are most useful when the product has a predictable design and can be modelled 100% using mathematical functions, parameterised design rules and expert systems. Unfortunately due to the complex nature of most products, any fully automated system to be really useful would have to be extremely large and replicate all the intrinsic design decisions as well as all possible variables and factors in the design making for an extremely large and complex rule base. The problems of such complexity mean that in real-life fully automated systems have been seldom applied to solve complex design problems, especially when there are multiple parties involved in the design generation process and not all of them having the technical capacity to construct and maintain the fully automated expert system.

Finally, a hybrid method that uses the knowledge persistence aspects of the automated design generation tools whilst enabling the user full control of the actual design, so as not constrain the design within a narrow range of scenarios was evaluated. This semi-automated method, also called interactive-method keeps the knowledge base and rules but puts a human in the loop at every step and allows the easy modification of the knowledge base by the user in order to expand the capabilities of the system in an evolutionary way, and making the new knowledge available to be reused within the other areas of the enterprise so as to synchronise the up to date capabilities from different functions of the enterprise and thus generating design examples that reflect the

current state of capabilities and capacity. Other advantages of this method include the fact that a system can be constructed relatively rapidly without the need to have a solution that generates 100% of the design.

5.2.2 Design Knowledge Base

The knowledge bases for the product knowledge and the knowledge base to define manufacturing models in STEP-AP224 are combined into a single project. Here multiple users can work on different aspects of the concept design. The combined knowledge base separates the two main knowledge categories under separate superclasses. AP-224_model is the superclass for the AP-224 model definitions and product_knowledge_base being the superclass for the generic product knowledge base see Figure 38. The hierarchical object-oriented structure of STEP was recreated using the OWL ontology. The element translation is shown in Figure 30, please refer to <http://protege.stanford.edu/> for a full definition of classes, superclasses and slots. Please refer to <http://www.iso.ch> for ISO standard 10303, Part 224 for a full reference of the standard. Appendix C contain the full ontology constructed for the AP224 standard.

5.2.3 Interactive translation methods

The functions needed to help the user generate new designs for manufacturing evaluations from the data entered are many. These functions are:

- ✎ Mapping customer request information to a product specification template;
- ✎ Mapping product specification template to an existing specification example;
- ✎ Mapping the resulting product specification to existing components within the BoM; and
- ✎ Mapping the BoM components to geometry and manufacturing features within the AP-224 knowledge base.

The customer request information was created from a Request For Quotation (RFQ) form made by Nissan Motor Company for their Tier-1 supply chain (in this instance given to ArvinMeritor). The product specification template used was from a Ford Motor Company internal specification book (once again used as a base standard for their supply chain to adhere to). This mix of vendors for RFQ to Specification allows for the

demonstration of the simplicity and versatility of the mapping tool for matching an RFQ to a specification from different OEMs for the same product (a car latch) that can be applied generically.

Requirements are split into requirement types, there are 79 different requirements which are categorised with multiple attributes in order to enable the query and rules mechanism to accurately map them to the desired specification.

The application example in Figure 12 illustrates the iterative knowledge construction process through the use of the well defined knowledge base and query mechanisms

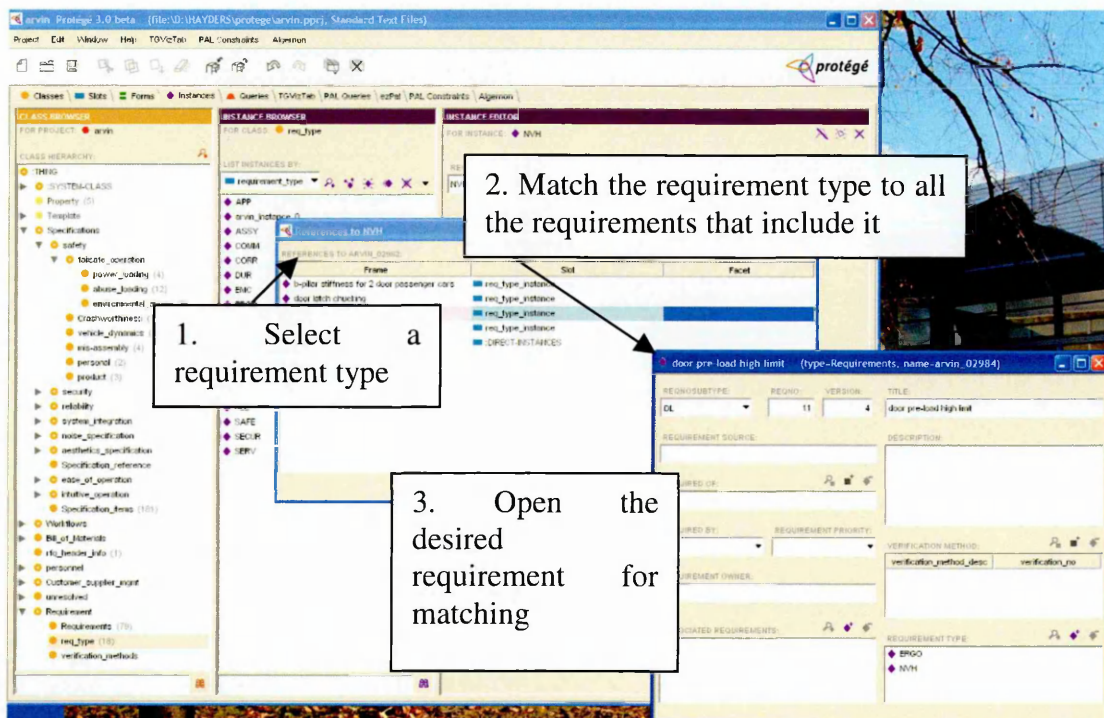


Figure 12 Pre defined queries match requirements to specifications

In step 1, the user selects the category or form of requirement that he needs to specify from the menu. The window for creating the requirement has a link to see related requirements in the current project (and if necessary define them). This aids the user in having a holistic view of the inter-related nature of the particular requirement and ensuring that the full knowledge is entered to generate a useful specification for the product. The second window has a link to query existing items of knowledge from older

projects that gives the user the ability to reuse components born from similar requirements in historical projects all from one view. The full sequence is shown in Figure 13

5.2.4 Application within collaborative framework

The application is an integral function of the Protégé environment and in an inter-enterprise framework it is distributed among the collaborators using the Peer to Peer application method described in 4.3.4. There are also specific workflows defined for the product development process in terms of the acquisition of the request for quotation information, generation of functional specifications, matching the specifications against existing specifications for designs and generating a basic design based upon components from earlier design iterations. These workflows are shown in the workflow section and appendix (b). The workflows are executed either using the Windchill Workflow engine or the Compiere workflow engine www.compiere.org and both of these are Java2 Enterprise Edition (J2EE) applications.

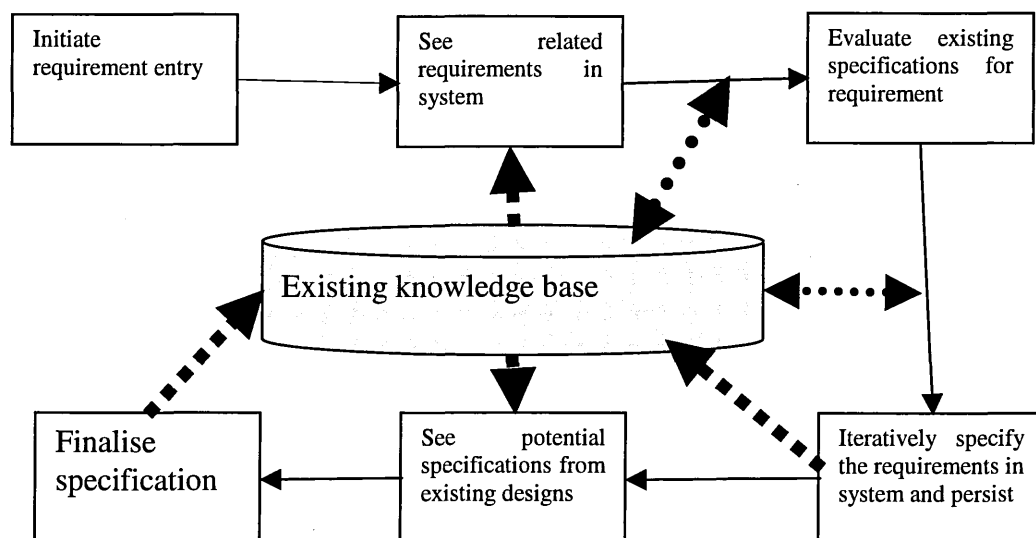


Figure 13 Interactive process for generating specification from requirements

Manufacturability analysis is carried out using the aggregate process planning tool CAPABLE, (Maropolous et al 2001). The output of this process plan is fed into the

Enterprise Resource Planning (ERP) system which then generates the capacity plan, Bill of Materials and, if necessary, pro-forma invoices for the RFQ. The ERP system used was simply a processing engine, and was not directly integrated within the inter-enterprise collaborative framework, leaving the potential users the freedom to specify and make use of their own proprietary tools.

5.2.5 Summary

The methodology and example application above was a simple application for design intent translation utilising workflows and pre-constructed queries. With the potential future addition of capacity planning and enterprise resource planning functionality, the rapid generation of design examples for manufacturability analysis is enabled albeit the system is not a fully automated design translation tool for the reasons elaborated on earlier.

5.3 Generic super-ontology description

Detailed below is the generic super-ontology component that was constructed using the ArvinMeritor and Mabey&Johnson project data and incorporating third party information such as the format and content of complex RFQ (Nissan automotive).

5.3.1 Knowledge base hierarchy

On a top level the knowledge base was split into the following separate top-level classes:

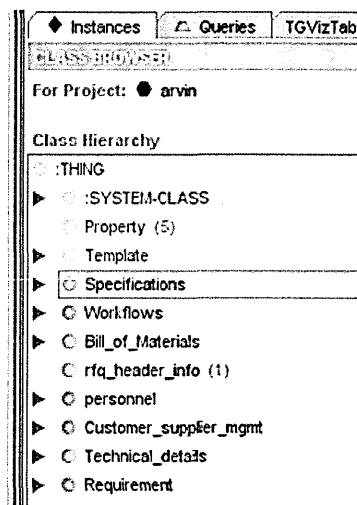


Figure 14 top level classes in knowledge base

These top level classes (Figure 14) are abstract classes and are simply place-holders for the detailed knowledge contained in their subclasses. It is important to note at this point that the nature of the ontology that defines such a wide scope of a company's activities is by its very nature inter-linked between different classes under different top-level classifications. It is in fact possible to reach knowledge contained under 'requirement' from the 'bill of material' and to reach the customer_supplier_management from specifications. In further depth:

5.3.2 Specifications

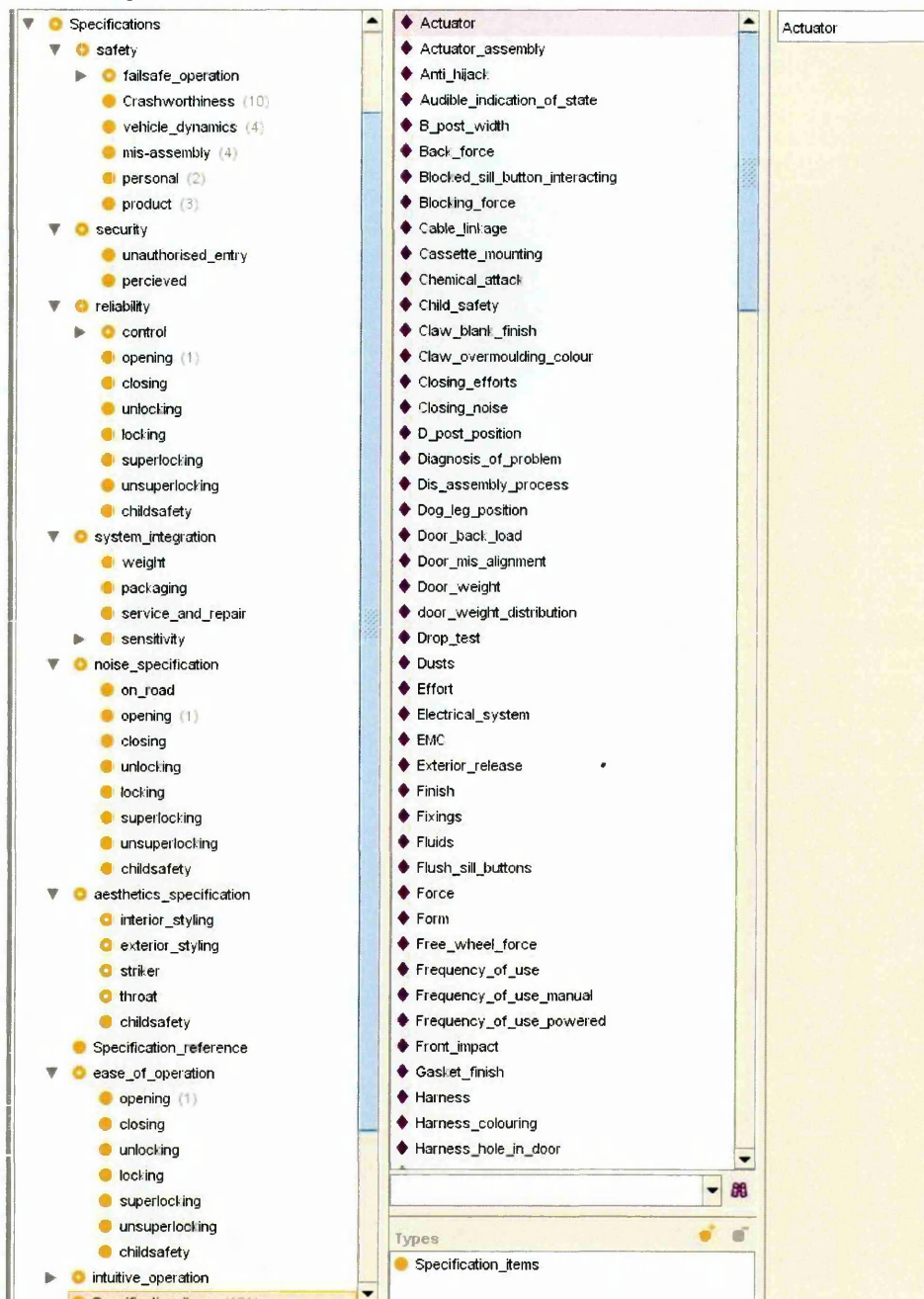


Figure 15 Knowledge base - specifications definition

Figure 15 shows the interface where the specifications for the engineering design product are produced / derived from the Requirements information (which is itself based on the RFQ). The final class under specifications is 'Specification_items' which contains the variables that are used in the specification creation and management section (181 possible objects). The sub-sections of specifications deal with (in this

example) latches for cars, and the nomenclature and structure reflects the specific requirement. The specifications themselves are split into multiple sub-specifications for safety, security, reliability, system integration, noise specifications, aesthetics specification, ease of operation and intuitive operation of the automotive latch.

The interface is split into three panes the left pane shows the top-level specification and sub-specification classes, the middle pane shows the “specification items” variables used in the specifications definitions and the right pane shows a description of the highlighted object from the middle pane.

To illustrate the operation one example is shown below.

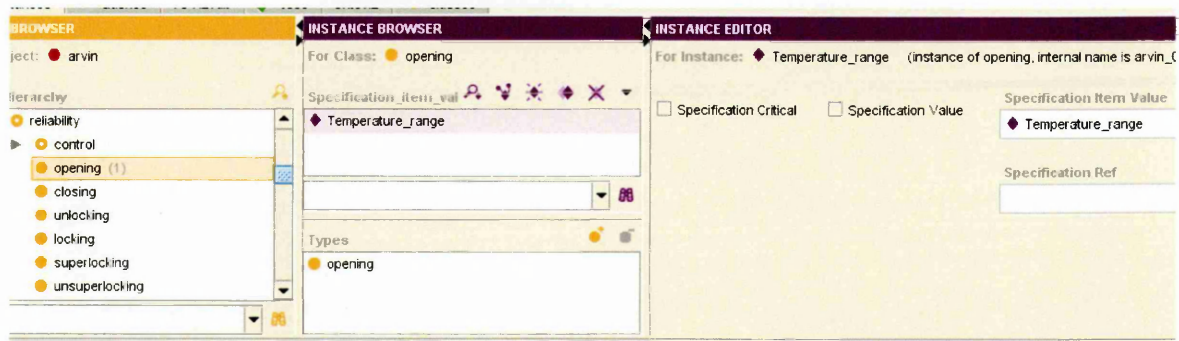


Figure 16 specifying the reliability of latch opening

Here the reliability required of the latch is specified (Figure 16). This form specifies the stated reliability of the latch being designed. The user selects the ‘opening’ sub-section of the reliability section. In the centre pane the user can specify the opening reliability specification object (in this instance it is the temperature_range) which is a predefined standard object within specifications (obtained from the specification_items class, which defines the possible variables for specifications). The right pane is the data entry form where additional variables can be added and the value and level of importance for the ‘temperature_range’ variable within the opening reliability specification can be amended (there are default values from previous projects which are automatically actuated when first instantiating the variable in the specification construction to aid inexperienced users on the specification construction stage, which can be, of course changed from this form).

These specifications, once constructed form the foundation of what the engineering design works towards. The real strength of course is that all the knowledge base specifications are defined in the same manner, enabling the design engineer to immediately bring up older projects with similar variables and reuse it in the new design.

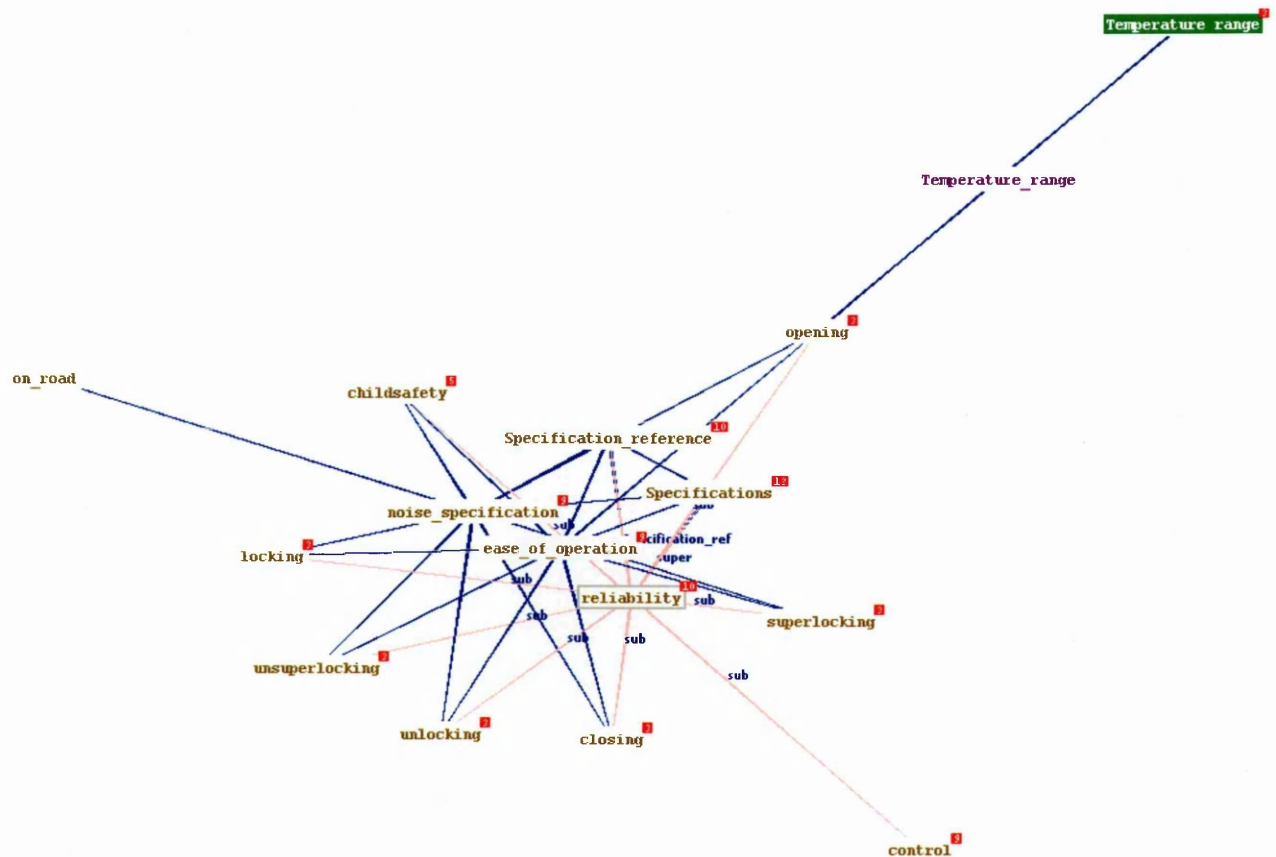


Figure 17 location of knowledge item for temperature range of the reliability of opening of latch

The figure above (Figure 17) illustrates the actual location of the temperature range instance for the opening specifications of the reliability sub-class of the 'specification' definitions of the knowledge base. This demonstrates the level of depth of the ontology constructed and also serves to show the relative simplicity of the end user interface when taking into consideration the complexity of the actual ontology structure.

5.3.3 Workflows

This sub-section details the research into and construction of workflows to function as a quality management tool within an overall concept and product development process. Some of the requirements for quality management are detailed in ISO 9001:2000, these

requirements are implemented in the system as workflows. A set of generic workflow elements were constructed based upon the company specific workflows below.

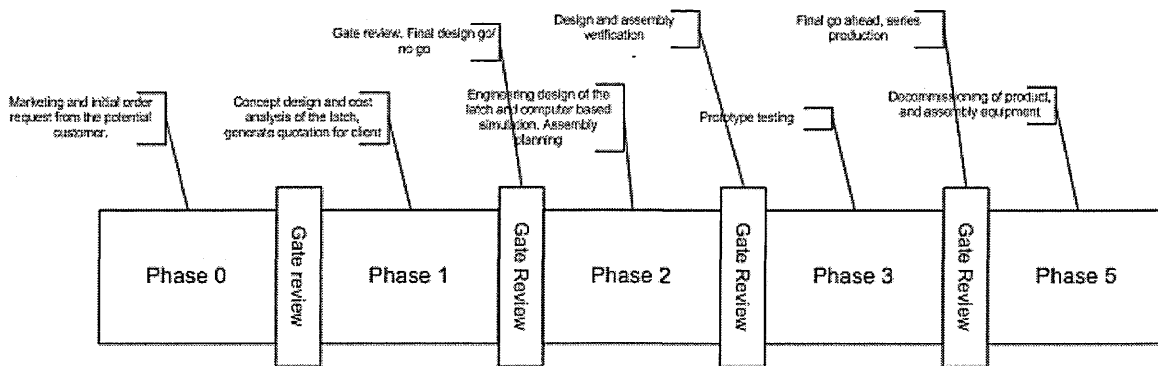
The actual engines for firing and managing the workflows were restricted by the overall architecture of the system. Whilst PTC Windchill's workflow system is undoubtedly superior in terms of usability, it could not, in this instance be used operationally in the final implementation since the overall architecture of the centralised PLM system did not meet the requirements of the research.

These were implemented using the XPDL Open Workflow specification, see (www.wfmc.org), in order to keep the workflow implementation portable across many different enterprise applications. The actual workflows were simply a glue to bind together the various tools and functions within the product development process and the workflows illustrated below demonstrate the closed-loop cycles usable for engineering change and other closed loop control mechanisms.

5.3.3.1 Workflow of ArvinMeritor processes

The company's development workflow follows the Concept2Customer guidelines contained in the Gate check-list document developed internally by the company, but reflecting best-practice solutions see Figure 18, Figure 19 and Figure 20,. The full set of tasks for the project phase have to be completed before the gate review is approved and the next phase of the project may commence. The phase 1 work processes (Figure 19) are split into 10 sub-sections (rectangular boxes) with sub-reviews (diamond shaped) within the phase to keep the project on track. A full diagram of the tasks and sub-section of the workflow is in Appendix (B). The workflow diagram was used to construct an example workflow in Windchill to manage the project cycles. see Figure 20 for an example workflow for the phase 1 process 'determine project go/no go recommendation' which three separate tasks (project scope, marketing strategy and costs and assumptions) are completed, and the result of which are evaluated to determine whether the project is feasible for further development.

ArvinMeritor: Lifecycle



Notes:

Phase 4 of the lifecycle is simply the series production of latches for the customer, in line with agreed protocol for delivery schedules and quality control.

The workflow diagrams for Phase 0 are not included as this phase consists primarily of ad-hoc activities.

The lifecycle and workflow diagrams included here adhere to the standards set down by the Workflow Management Coalition. <http://www.wfmc.org/>

The diagrams are structured in a three layer format:

- Lifecycle
- Intermediate review
- Individual tasks

Figure 18 ArvinMeritor lifecycle

ArvinMeritor: Phase 1

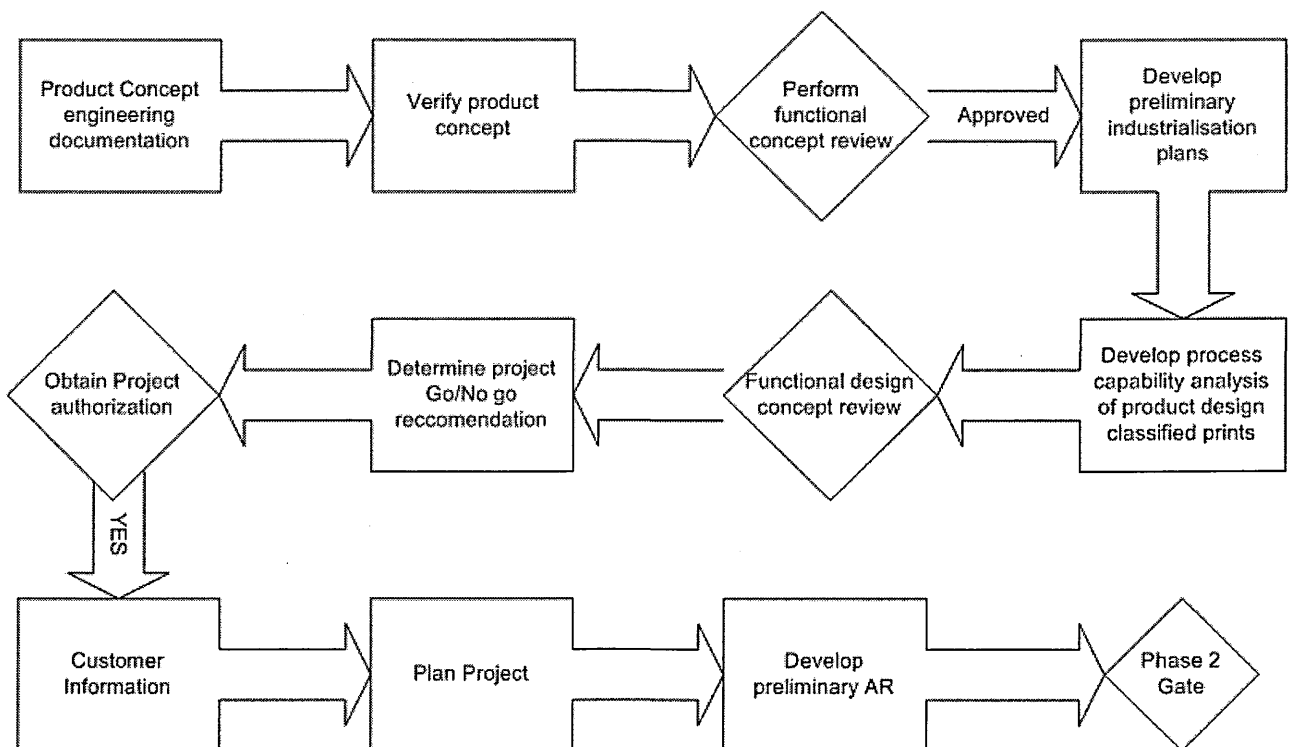


Figure 19 ArvinMeritor Phase1

ArvinMeritor: Phase 1, Project Go/No go recommendation

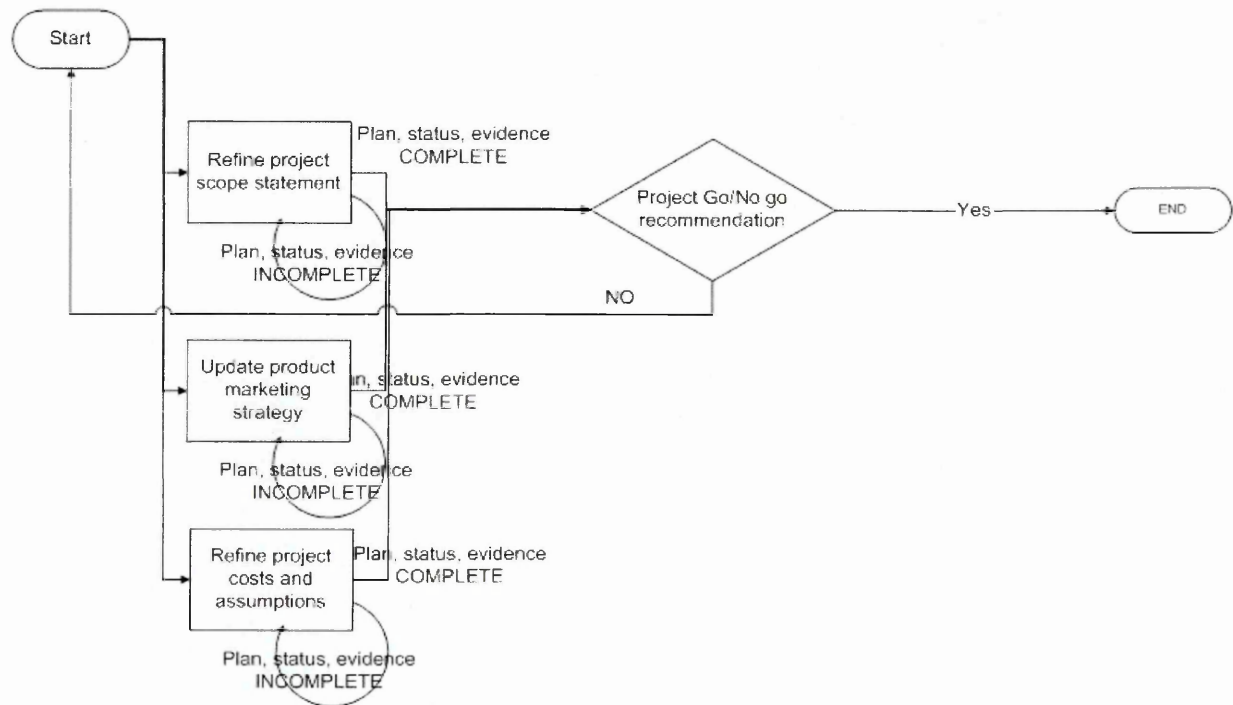


Figure 20 ArvinMeritor Phase 1 project Go/No go

5.3.3.2 Workflow implementation in ontology

The screenshot displays a software interface for workflow implementation in ontology. On the left, a 'Hierarchy' pane shows a tree structure of workflows. Under 'Workflows', 'Phase_1' is expanded, listing tasks such as '10_Plan_project (1)', '14.3_Develop_preliminary_AR (1)', '2_Product_concept_engineering_documentation', '3.6_Perform_functional_concept_review', '3_Verify_product_concept', '4.04_Develop_process_capability_analysis_of_product_design_classification', '4_Develop_preliminary_industrialisation_plan', '5.07_Functional_design_concept_review', '5.09_Determine_project_go_no_go_recommendation', '6_Obtain_project_authorisation', and '7_Customer_information (1)'. Below this, 'Phase_2', 'Phase_3', 'Phase_4', and 'Phase_5' are listed. The main area shows a form for 'arvin_00183'. It includes a 'Phase Evidence' dropdown set to 'None', a 'Phase Plan' dropdown set to 'Being Planned' (with a list showing 'In_plan', 'Being Planned', and 'Not planned'), and a 'Phase Status' dropdown set to 'Risk'. A 'Phase Item' field contains 'Confirm_government_regulations'.

Figure 21 Workflow definition within knowledge base

The screen dump in Figure 21 illustrates the workflow hierarchy and definition. In this instance phase 1 workflows are shown and the 'customer information' sub workflow is shown in the right pane with the drop down forms that set the stage and status of the workflow process (the workflows are fully specified in diagrammatic form in Appendix (B) Test workflows (ArvinMeritor Data). The data and layout are straightforward for the end user. A hierarchical left hand menu breaks down each lifecycle stage into separate workflow stages which are broken down into individual running processes as defined in the 'instances' shown in the middle column and with their lifecycle states and status shown in the right column.

5.3.4 Bill of Material

One of the more complex components of the ontology is the Bill of Materials section. The figure below (Figure 22) illustrates the definition of the individual components of the bill of materials assemblies of all possible configurations within the knowledge base (note that the geometry and feature definitions are in the STEP ontology, this interface shows only the BoM configuration details).

The screenshot displays the 'Bill of Materials component configuration' interface. It features a hierarchical tree on the left, a central list of components for a selected class, and a detailed form on the right for a specific instance. The form includes fields for part numbers, suppliers, issue levels, product descriptions, quantities, and material specifications.

Figure 22 Bill of materials component configuration

The actual structure of the Bill of Material component is a complex relationship between all the components of the system. Since the bill of material is the 'core' of the

system it brings together the various VE participants (who make components of the final product') as well as the OEM who manages the specifications according to the

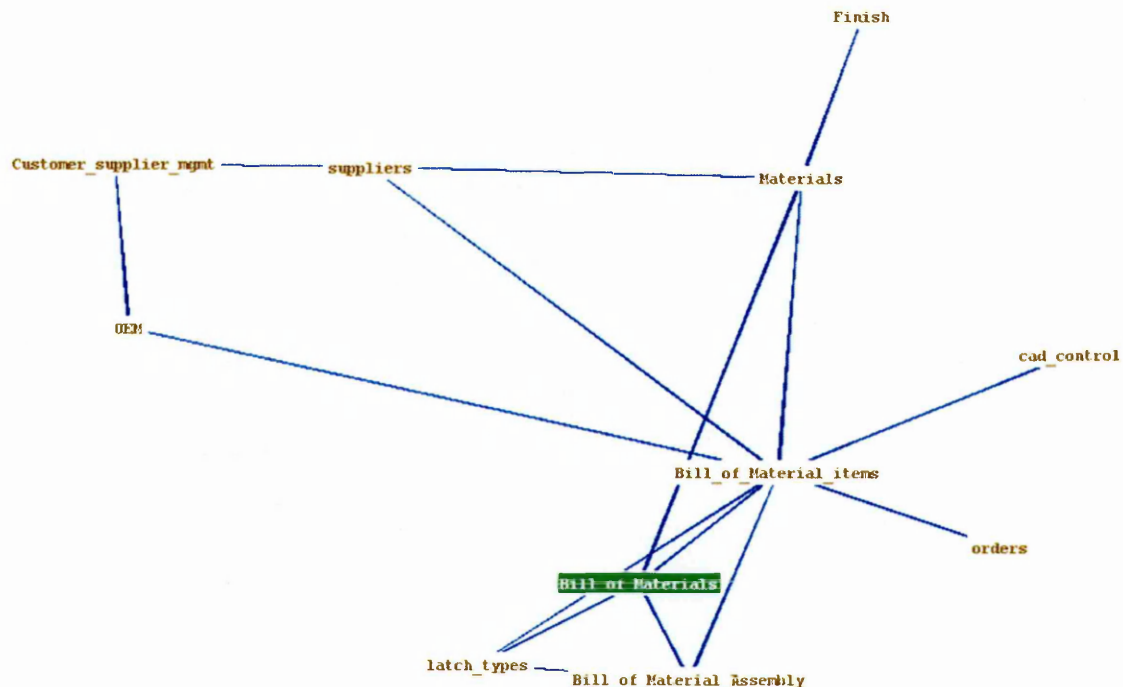


Figure 23 bill of material sub ontology structure

requirements. An illustration of the internal structure is Figure 23. The final product is an instance of 'bill of material assembly and is composed of bill of material items. Each of the items has materials, suppliers, finish quality, configuration (latch type), orders and versions (cad_control) associated with it. A user adding a new component to the engineering design will fill in the component form in the bill of material items form that will automatically populate the rest of the knowledge base with new materials, new manufacturing processes and new latch type configurations (if applicable), this would then be available generally when creating new bills of materials for design configurations.

5.3.5 Personnel

This module within the knowledge base provides the role of access control management, project time management and role management that is essential within a de-centralised VE. The roles defined for users here as well as the time sheet information for each participant provide the coherent and simple project and time management necessary within engineering projects, whilst maintaining a de-centralised user

environment. Each individual ‘person’ within the system and each task/role and timesheet is associated with a project and a date and is used to audit the tasks performed by person as is necessary in all OEM and tier-1 level VE projects.

The screenshot displays a software interface for managing personnel project time. The top section is divided into three main areas:

- Project Hierarchy (Left):** A tree view showing categories like 'personnel', 'advanced_engineering (1)', 'client (1)', 'engineering', 'manufacturing', 'marketing', 'subcontractor', 'testing_legislation', 'timesheet_template (5)', and 'Customer_supplier_mgmt'.
- For Class: timesheet_template (Center):** A list of instances including '23', '34', 'arvin_00157', 'arvin_Instance_0', and 'arvin_Instance_1'.
- For Instance: 23 (instance of timesheet_template, internal name is arvin_00107) (Right):** A form with fields for 'Contact Address', 'Contact Number', 'Hours During Day' (set to 1), 'Contact Name', 'Week Number' (set to 23), 'Year' (set to 2002), 'Role', and 'Day Of Week' (set to thursday). A 'Create Instance' button is visible.

The bottom section shows a zoomed-in view of the form for 'arvin_Instance_0 (instance of timesheet_template)'. Red boxes highlight the 'Week Number' and 'Year' fields in the 'Contact Name' section, and the 'Day Of Week' field in the 'Role' section.

Figure 24 Personnel project time management

5.3.6 Customer Supplier Management

This component of the super-ontology exists primarily to glue together the complex chain of enterprises within the VE in terms of their competencies, existing supply agreements and inter-enterprise relationships.

The information is automatically embedded and generated when a design iteration is created within the BoM. Figure 25 illustrates the complex inter-enterprise relationships in a simple diagram, from left to right a list of 'suppliers' or component manufacturers is shown. One of these 'Barton' is expanded to illustrate the relationship between 'supplier' a component (a claw_and_pawl_rivet for a latch), the associated material (BS-3111_type_9/2_con_B), and via the customer_supplier_management module the relationship is explicitly defined with the 'OEM'.

Looking from right to left, the relationship is seen from the 'OEM' perspective. Here a top level OEM 'Volvo' is used as an example to illustrate the links in relationship between the enterprises in the VE. The backplate subassembly for the front right latch is used as the example. This is produced by the 'gammastamp' enterprise from the 'supplier' class of enterprises. Notably, the system also indicates that the company gammastamp is also the production supplier for another item 'claw and pawl rivet' that was used as an example of the item produced by 'Barton'. This illustrates a common situation within VEs where multiple enterprises are capable of producing the same components needed within the VE project. The system actively pushes such information to the VE participants enabling the pursuit of previously unknown potential producers based on the company's inherent internal ability + a project relationship with an OEM for a particular component.

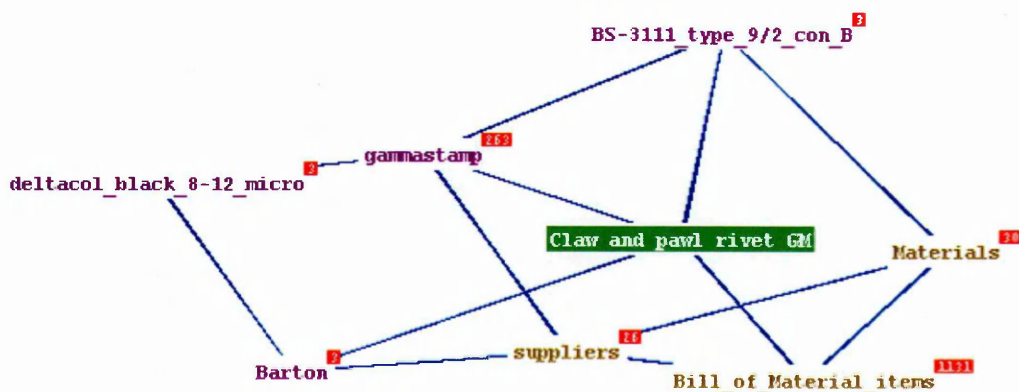


Figure 26 claw and pawl rivet relationships

The number of potential relationship webs is limited only by the number of enterprises in the chain and by the variety of products and materials they use. Here's another clearer view of the 'gammastamp' and 'Barton' production of the same item 'claw and pawl rivet' see Figure 26.

5.3.7 Technical Details

This module of the super-ontology was based on the Mabey&Johnson and ArvinMeritor internal documents and work practices. It is a large ontology module with many optional components that can be chosen or disabled within the VE.

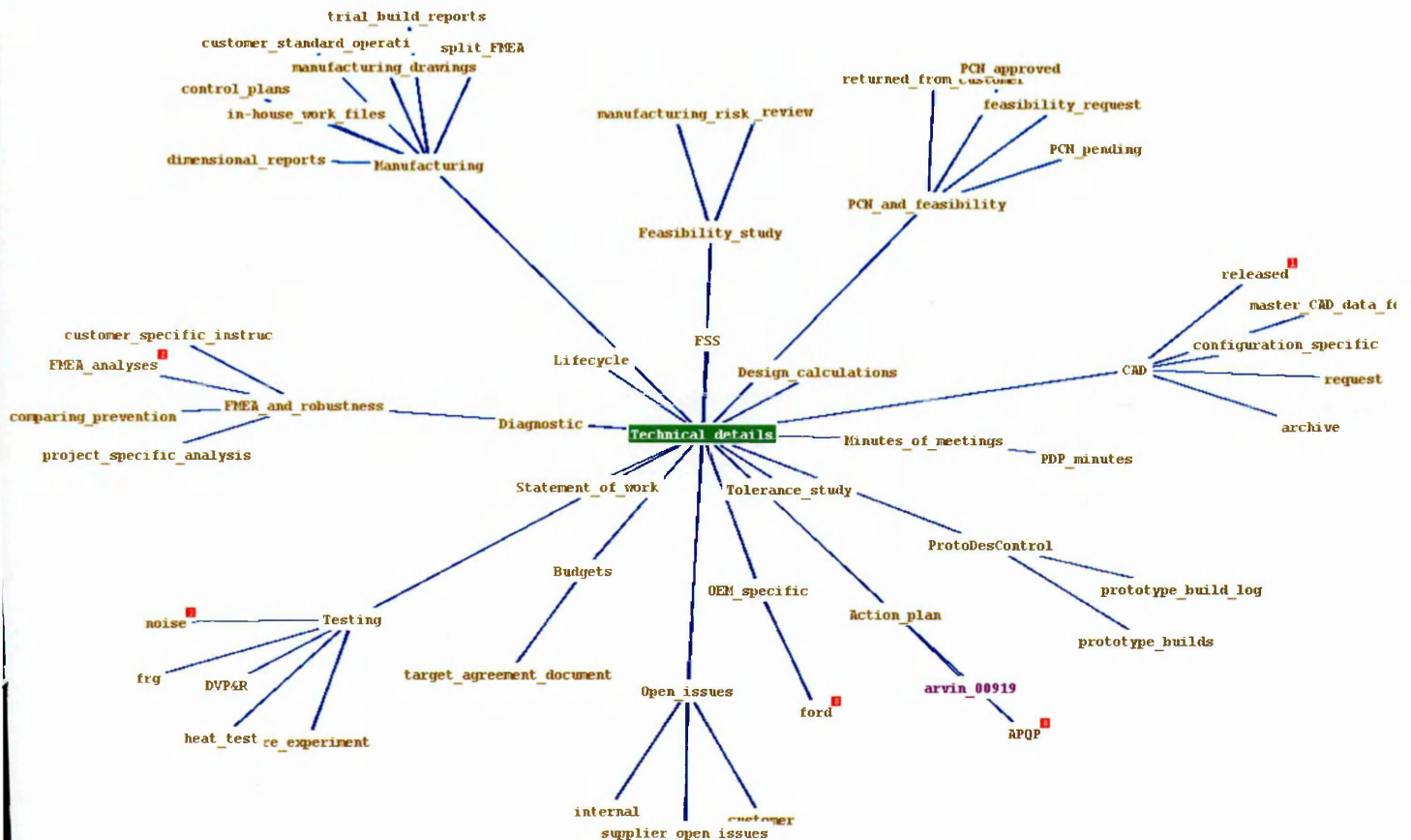


Figure 27 Technical details sub-ontology

The sub-ontology has a number of components including modules for testing, prototyping, budgeting, tolerance studies, manufacturing, Failure Modes & Effects Analysis (FMEA), CAD models management, feasibility study and an issue management module.

This module is designed to give the VE flexibility in defining their own ‘module’ within the super ontology whilst maintaining the structure of the super ontology as used within the entire VE for multiple projects. The one method of overcoming the difficulty of standardising on such a system is to standardise the structure of the sub-ontology and

leave the actual structure of each instance data open to whatever each individual VE chooses.

5.3.8 Requirements

The specifications are built up using the knowledge gained from the requirements specified by the top level customer of the VE (the OEM).

Figure 28 Requirements specification

The module for requirements automatically populates a knowledge base with a base set of requirements as shown in Figure 28. The list on the left shows the list of ‘requirements’ as created by the customer of the VE. The form on the right shows the details for the highlighted requirement. Each product requirement in this instance has around 80 separate objects with specifications. Each of these objects has a number of fields associated with it. The fields are both descriptive of the nature of the requirement

as well as intrinsically creating links with the 'specifications' module of the super-ontology creating the bridge between requirements and specifications.

This example shows the requirements for corrosion resistance. A latch example is used (although corrosion resistance is applicable to many fields). A number of options are selectable including specifying whether this particular requirement is specified by regulatory authorities (i.e. an independent standard that has to be met), by the customer or by the VE collaborators (corporation). Each requirement can be chained in an association with other requirements, creating concrete relationships between requirements and constraining the potential solutions into taking into account related requirements. Each requirement also has a requirement type that is selected (it can be multiple selections), in this instance the requirement is for DUR (durability) and CORR (corrosion resistance). Finally the requirements have a set of processes used for verification of the design against the requirement, these verification methods are also stored within the system and can be reused by choosing them from the 'Verification method' selection list. Finally, the requirement also has a description of its role associated with it that is searched using free text search; this is there to instantiate any possible requirement variables not already set in a field.

5.3.9 Summary

The above brief introduction illustrated the generic super-ontology created in this research to act as a base ontology for inter-enterprise collaboration. The user interface elements were demonstrated as were the individual modules within the super-ontology. Also the relationships created between specifications and requirements and workflows within the applications demonstrate the method used to manage the flow of the design process seamlessly and the instantiation of existing objects within a new design illustrate the reuse of knowledge from previous projects.

Within the hierarchy of the two-tier architecture for inter-enterprise collaboration, this super-ontology forms 'tier-2' (format). The first tier (medium) being simply the JXTA authentication classes over a super-peer net topology that enable participants in a VE to see each other.

5.4 STEP models

The STEP AP-224 feature models (a component of the 'format', tier-2 of the two tier architecture) are defined in Protégé and use the Java expert system shell (JESS) to define the rules and functions of the standard. In addition some of the rules have been created using the Protégé Axiom Language (PAL) and the constraints these impose enable error-checking for the user during the definition of features.

However, these constraints can only be applied to individual entities within the STEP model and cannot enforce any constraints between entities/features. Due to the size of the standard, only a subset of AP-224 has been translated from the original Express schema. The definition of the Express (STEP) data types is also contained in the Clips interface. This is very flexible as it allows for the inclusion of STEP data types to other components within the ontology on a need basis without having to have any expertise in Express or any other programming language. The above enables the mixing of feature and meta-data information in the knowledge base, meaning that users can access the information stored in STEP models using queries and RDF parsers. This integration at low level between the geometric, feature and 'meta-data' within a single environment is intended to reduce repetition and errors, and also enables the reuse of all the data created during the conceptual design process. A STEP AP-224 based automatic process planner by (Sharma et al 2004) generates plans from concept designs. The aggregate process planning tool by (Cheung et al 2003) generates assembly plans from incomplete models to enable the optimization of the design for assembly and manufacturing from the early stages.

In Chapters 2 and 3 an overview of current trends and research developments in the fields of distributed and collaborative design was presented. As can be seen the current trend is focusing on a number of highly specialised areas. Most notably: Researchers value the importance of Design for Manufacture and Design for Assembly, and realise the lack of current computerised support for such activities in a collaborative and distributed environment.

In order to overcome this problem conceptual designs rather than fully featured designs will have to be assessed for their manufacturability and ease of assembly. The product models must be created with the full definition machining features from the earliest stages of conceptual design. Process planning and assembly planning must be carried out on the conceptual design iterations so that only the most suitable candidates are chosen for further development. Manufacturing and assembly capabilities are dynamic, especially when supply chains are involved. Therefore the updating of manufacturing rules has to be as effortless as possible. One idea would be to create an XML DTD to define manufacturing rule-logic. Although this would have to be adopted by machine tool manufacturers, who would supply their customers with the manufacturing rules already with the machines, it would reduce the amount of knowledge engineering needed.

The problems of sharing product data within the supply chain has to be overcome with the use of international standard definitions for all aspects of the product data lifecycle.

- ⌘ STEP AP224 contains the machining feature information necessary for a single machined item to be produced; www.iso.org
- ⌘ STEP AP214 contains the necessary information to make configuration based designs (from AP203), and to make assemblies, generate BoM and store process plans for the components with the geometric data; and
- ⌘ STEP Data Access Interface (SDAI) allows the exchange of STEP model data across networks and databases.

The above Application Protocols have the necessary information to encode the models and features of the engineering designs for the purposes of manufacturing process planning and geometric design.

A number of automatic feature recognition tools have been developed to translate legacy data into manufacturing features. However the only program with sufficient feature definitions commercially available is produced by Geometric Software (www.geometricsoftware.com) , and is a very costly piece of software. The feature recognition capabilities emerging in new CAM tools such as FeatureCAM (Engineering Geometry Systems 2001) are generally licensed from Geometric Software. Also it may

be possible to utilise the express schema for automatic feature recognition (Inouye, 2001).

The ability to generate and analyse concept designs in a collaborative environment, yet geographically separated is necessary, and much of the research effort over the last 2-3 years has focused on web-based applets that can generate designs whilst being viewed simultaneously by different clients. These tools must combine several features in order to be successful:

- ❧ The ability to store and query existing conceptual design knowledge, as well as capture new design knowledge;
- ❧ To generate sufficient geometry to allow rough process planning and/or assembly planning to be carried out on the component on the fly;
- ❧ To store the geometry in such a way as to make it reusable both for downstream applications (solid modellers/FEA tools/toolpath generation), as well as to be stored for re-use as concept ideas in the future (perhaps by utilising GT codes for features or through the use of Case Based reasoning approach and give each component a rating during the concept development stage);
- ❧ The software should be integrated within a collaborative knowledge management application so the process of storing the data and generating model revisions, and product tree structures would be seamless within the collaborative knowledge management application. Also this would allow the conceptual design process to be integrated within the development workflows for the project; and
- ❧ The tools should be provided with a single “window” interface to undertake all tasks and generate results. Many of the current researchers are attempting to simplify the conceptual design and manufacturing/assembly evaluation using simplified GUIs often implemented in JAVA (Sun Microsystems) and HTML for browser access.

5.4.1 STEP geometry and assembly generation

The requirement to integrate STEP geometry and assemblies, generate AP-224 feature models for process planning and integrate geometric and feature models with other

design data and project information has been tackled using two separate methodologies and their applications, the proprietary CAD system method and the open-standard method. The reasons for attempting two approaches was to verify the viability and select the methodology which adheres to the appropriate, but clear and simple, criteria for collaborative product development:

- Be compliant with open-standard and easy to deploy and use by companies, suppliers and clients;
- Reduce network traffic problems by minimising downloads of applications for design and knowledge management functionality;
- Be usable without any formal training; and
- Generate Industry standard STEP data and include a subset of AP-224 feature information from the concept design stage for easy design analysis.

5.4.1.1 Proprietary CAD method

In this method, STEP AP-224 output was added to an existing solid modeller (Granite 1) using the Pro Desktop API. A diagram illustrating the sequence of actions of the system is shown in Figure 29.

The Pro desktop API was found to be not ideal for this kind of customisation for a number of reasons:

- The programming is done in visual basic (VB) and C++
- The native file format is proprietary and the STEP export method in Visual Basic for Applications (VBA)
- The STEP output is AP 203 and does not include the feature information (that is available already in the Pro/Desktop native format)
- Connection with other sources of model information (non-geometric) not possible from within the Pro/Desktop environment.
- Connection to data services such as PLM are only possible to PTCs own proprietary systems and only using the Pro/Desktop full version application (not the freeware ProDesktop Express).

In order to overcome these shortcomings, the following software engineering tasks were made:

1. Enable the interaction of XML and RDF (OWL) based meta-data with the solid modeller.
2. This was completed using the Protégé expert system shell that has already been utilised for the structured and semantic storage of the companies' knowledge bases within the Enterprise system.
3. Create an ontology for storing the STEP AP-224 meta-data in Protégé
4. This ontology stores the EXPRESS entities in Protégé
5. Can be edited within Windchill, Protégé or text editor
6. Perhaps direct integration into Pro/Desktop (not implemented in this demo)
7. Persist the meta-data in XML on the server side (Windchill)
8. Can be done using Windchill methods for checkIn() and checkOut() methods to standard Windchill document (WCDocument)
9. Also using the JDBC interface in Protégé directly to the Oracle database.
10. The interface can use the Loadfile() and SaveFile() methods from Pro/Desktop to connect to an application. A link between the file system and web server have to be created. This can be accomplished using the Java Native Interface connection to Windchill methods.
11. Create an interface between Pro/Desktop and Protégé using the Protégé plug-in development tools in Java create a plug-in that connects Protégé with Pro/desktop (Using the COM function in Pro/Desktop proDesktopLib.IpartTranslator ExportSTEP) and to use the meta-data in Protégé-2000 to append to the Part-21 file generated and produce a valid AP-224 file.

The STEPTrans java package, see Figure 29, contains a number of classes, with the following roles. The STEPmain.class contains the main() method which initialises, runs and terminates the application. The program in this instance is event-driven and accessed through a SWING form Frame1.class. The Frame1.class contains the basic functions to open, close and save files. The main controls available are “select

ProDesktop file” file menu selection and select Protégé library (which brings up an XML selection tool).

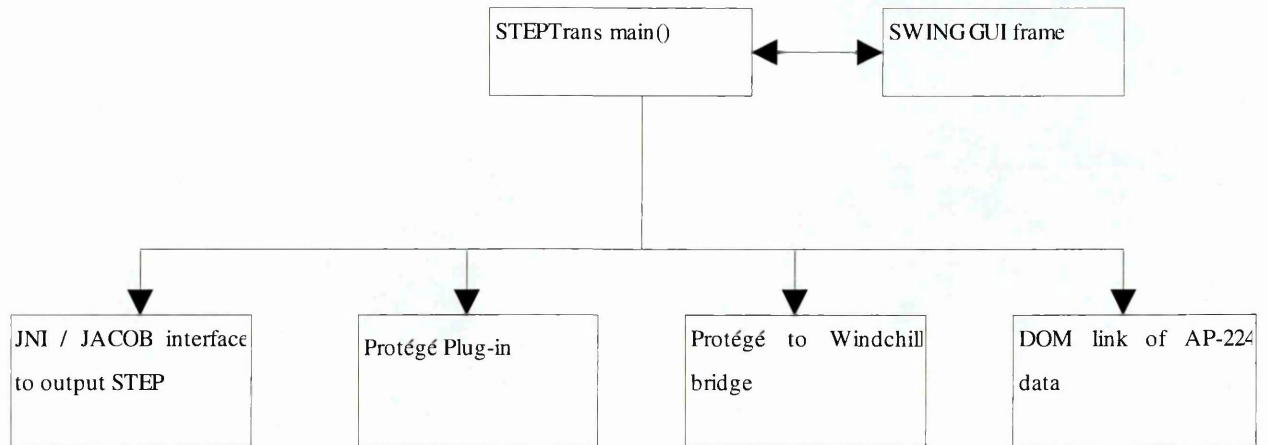
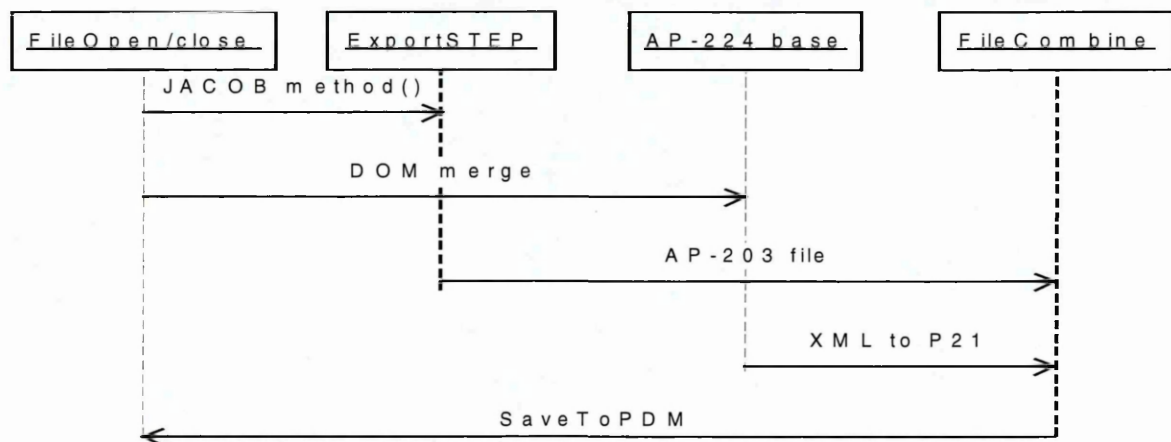


Figure 29 Sequence diagram of STEP file generation and storage

Then the merge button takes the Pro/Desktop file, passes it to the ExportSTEP class and takes the result and appends the stripped text from the Protégé file selected. The resulting output is saved as a . STEP (.stp) file with the same name as the original ProDesktop file. Connection to Windchill is via the Protégé to Windchill link.

5.4.1.2 Open-standard method

In the second methodology, the use and modification of an AP-224 model was split into a two stage process. The first stage is the generation of AP-203 geometry using any available modeller (in our instance models were generated using Pro/Desktop, and also

the open-source modeller OPENCASCADE/ExoTK) and the use of open source modellers such as Open-Cascade is encouraged for this purpose.

In the second stage of the method, the geometry is imported into the knowledge base editor and combined with the other elements of information from the design process into a single repository.

The implementation of this application was made using the Protégé ontological editor to create the AP-224 standard entities, rules and functions. The original standard was written using EXPRESS, and in this project the EXPRESS based STEP standard has been translated into a JESS/RDF based knowledge base see Figure 30. The direct translation from EXPRESS to a JESS based system was not straightforward. There were a number of software issues which were tackled: EXPRESS allows entities to have multiple superclasses and TYPES and entities can have the same name within one schema. This causes a conflict in the JAVA/JESS/CLIPS based system, because a class can have only one superclass and a name can be used only once (whether for a class or a “slot”). The problem was overcome by creating Metaclasses (template classes) for the classes with multiple inheritances, and preceding all Express entity definitions by a _ in the JESS based system.

Many of the rules defined in EXPRESS were simple constraints and there was no need to implement these constraints in JESS. The constraints were implemented using Protégé Axiom Language and Protégé’s standard “facet” definitions (see Appendix (c) for definitions and source code). Importing the Part 21 data was not straightforward (there is no STEP import mechanism in Protégé). (See Figure 31)

- ❧ An import filter was written in Java for the STEP classes using a text parser. Integration with protégé was made using the java calls available in the JESSprotege Tab.
- ❧ Use the (bind ?new_jess_function (new java.text.MessageFormat.parse(String “stepfile.stp”, ParsePosition pos))) to make the new step file.
- ❧ Use (set-member) and (get-member) to access the functions variables in java.

- ❧ The text data (P21) populates the knowledge base and becomes accessible to the wider development community and non-CAD specialists.
- ❧ Once the Part 21 data is incorporated into the knowledge base, features have to be created to represent the Boundary Representation (B-REP) geometry as AP-224 machining features. The feature creation is made using a SWING form in protégé, and it is interactive (not Automatic Feature Recognition).
- ❧ If the user has access to Automatic Feature Recognition (AFR) application then the features can be imported into Protégé without the need for interactive feature generation.

The resulting feature/geometry model is stored in RDF/OWL defined knowledge base. However at the same time, the company's ontology (that had been defined earlier for both our test cases ArvinMeritor and Mabey&Jonhnson) is integrated with the feature/geometry model to create a common repository with the complete information for the particular project.

Queries, constraints and rules are contained within this data repository and allow for the rapid generation of ideas. The rules and constraints defined allow the developers to experiment with different configurations and see get feedback about the viability immediately. More importantly, in the next phase of the programme the planned integration with LOCAM (LSC 1999) will allow the manufacturing engineers to generate preliminary process plans for manufacturability analysis.

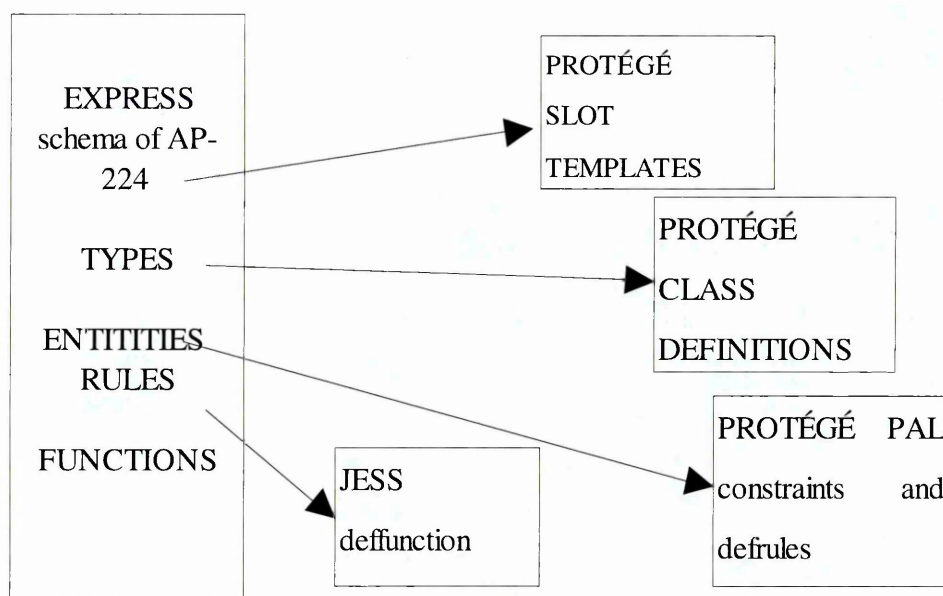


Figure 30 Translating EXPRESS schema to knowledge base

The information contained in the mixed-mode knowledge base can be queried using Protégé Axiom Language Queries, Protégé Queries, CLIPS/JESS defquery, OWL inference engine and RDF parser. The wider uptake of the semantic web in the near future will ensure the wide propagation of information from this knowledge base through machine understandable encoding of the data RDF parsers, OWL inferencing and KQML queries.

The use of the familiar Protégé user interface ensures that clients can easily query and manipulate information within the knowledge base. The application is written in Java and the GUI elements in SWING making the application platform independent and sharing a common look and feel across platforms.

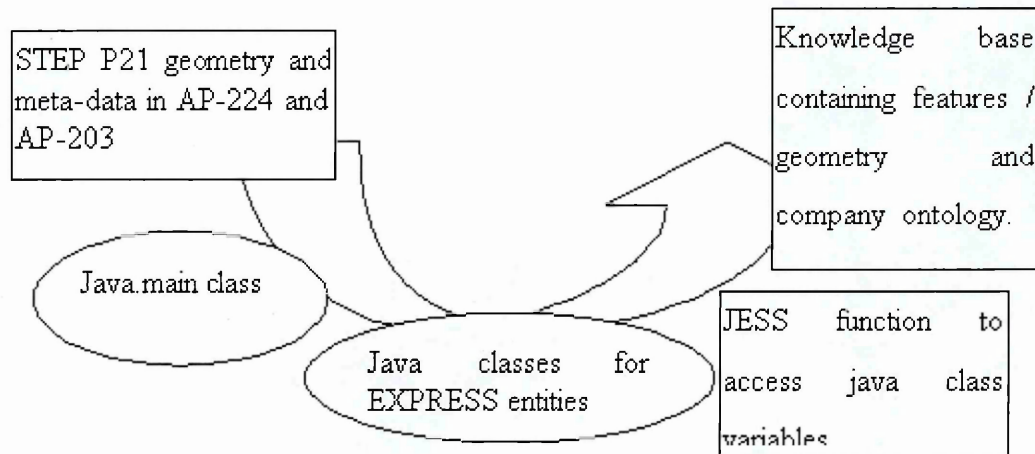


Figure 31 STEP P21 import sequence

5.4.1.3 Selected Solution

The method of integrating ProDesktop with a subset of protégé for AP-224 generation was discontinued due to a number of issues that arose:

- ❌ The integration between Java application and the COM interface in ProDesktop is slow and unreliable and also requires the use of a Windows machine due to the use of COM components and the JACOB Java Native Interface (JNI) based bridge for Java;
- ❌ The lack of platform independence hinders use of the application by a wider user base;
- ❌ Customisation of a certain vendors' CAD product to generate the geometry is inflexible. It ties users in to using a certain product;
- ❌ The alternative method of integrating the P21 AP-203 file directly with the knowledge base ensures that users can use any CAD modeller to create their geometry and still make use of the conceptual design knowledge base and AP-224 feature editor; and
- ❌ The rules defined using JESS are many times faster than Express validation or any procedural application due to the use of forward chaining RETE algorithm (for dynamic data this is much faster than PROLOG based expert systems). This type of algorithm saves processing capacity by only firing the rules and functions if any of the variables is altered or reset to its global state. However

this is made at the expense of increased use of storage capacity for results of rules and functions.

5.5 Summary

This chapter described the development of a generic ontology system, which constitutes the format, or second tier of the overall two-tier architecture described in Chapter 4. In this instance, the application that the end user manipulates the company ontology with to provide a holistic knowledge management service and nomenclature for engineering enterprises in a Virtual Enterprise setting was shown. Section 5.1 gave an overview of the ontology and the reasoning for the choice of Web Ontology Language ontology on an RDF triple format for the shared ontology. The use of semantic web technologies enable intelligent agents to find and infer knowledge from the ontology with its correct context through the use of URI to ascertain the context and description logics to explicitly define the constraints and hierarchy of the knowledge nuggets. Section 5.2 detailed the system's functional architecture for capturing design intent (functional requirements) and inferring possible design specifications and concepts based on existing designs. Though the use of a generic sub-ontology for the customer requirements, and the explicit links with the subsequent development cycle of the design on an object by object level enables even simple queries in Protégé to infer possible design configurations (as long as there is existing knowledge in the knowledge base, naturally). This section detailed briefly some of the lifecycle and workflow processes that were captured and incorporated as part of the project. The importance of the workflow component within the ontology is due to its behaviour as the glue that binds individual processes of the development cycle together. Thankfully an international standard for workflow processes has been agreed upon and this standard XML Process Description Language (XPDL) (www.wfmc.org) was translated and incorporated into the ontology. Section 5.4 looked at the possible methods of incorporating the ISO-10303 STEP models into the ontology and possible means of capturing, manipulating and reusing the STEP data

Overall, the skeletal ontology does not hold every single item of knowledge used internally in a system, however it does hold all necessary product related knowledge in

a format that is shareable, reusable and extensible across a distributed and heterogeneous environment and holds the necessary information required for multiple parties to collaborate within the bounds of this application and the second tier of the two-tier inter-enterprise integration architecture. It is important to note at this point that the 'format' tier described in this chapter is shallow in terms of its knowledge content during the early adoption and use stage within an enterprise. The 'red tier' as seen in Figure 3 and Figure 4 constitutes the domain knowledge of the participating enterprises that is located outside of the system. At an early stage of the use of the two-tier system, the red band would constitute the vast majority of the knowledge, and would slowly seep into the 'format' tier of the VE system as collaborative projects are initiated and knowledge starts to enter the system.

6 Implementation of industrial case study

The test example was implemented using real-data from live projects of the co-operating enterprises in the research. The data was used in conjunction with a number of implementations for the format and medium mechanisms of the communications architecture. The actual ontology was implemented using the OWL ontology format on the Protégé ontology editor and this was integrated to the three communications architectures implemented.

6.1 Format implementation (tier-2)

The chosen format for the system was a Web Ontology Language OWL based ontology defining the normalised data model for a superset of international standards as well as best-practice examples from commercial collaborating enterprises. This data model was implemented using the Protégé ontology creation tool described in depth in chapter 4.

The authentication tier of the two-tier solution was implemented using the JXTA standard. Since this is standard already comes with existing implementations in XML (for the data model) and binding in Java and C a simple compilation of the tool in those languages was sufficient for making it usable with the tools below.

As can be seen there were a number of different systems implemented for the medium aspect of the communications infrastructure, and separate implementations for each of the mediums was made.

The implementation of the two tier architecture in full was possible only through the use of the Peer to Peer communications architecture. The implementations using Windchill and the Open Source PLM used proprietary standards for the communications component. Reasons for not implementing the JXTA protocol with Windchill are detailed under 'Windchill implementation'.

The following components were used or created for the 'format':

Web Ontology language (OWL) is the standard that supersedes the Resource Description Framework (RDF) as the format for persisting the Semantic Web initiative ontologies and knowledge bases. It is meant to standardise the persistence and sharing of ontologies and knowledge bases and aid the realisation of a common international 'Semantic web' as an intelligent evolution of the current HTML/XHTML based internet. The Standard itself is implemented using RDF notation on an XML schema, and there are a number of complementary versions of OWL to fulfil different requirements. These standards are Description Logics (OWL-DL), OWL-Full and OWL-lite. Further information on OWL may be obtained from www.w3c.org and www.semanticweb.org.

The RACER OWL inference engine which can read and infer OWL Description Logics knowledge bases was used in conjunction with Protégé and operates transparently to the end user.

Java2 Standard Development Kit (J2SE 1.4.2); This language and runtime environment was used for running the protégé system. It is a heterogeneous system that runs on any computing platform and has a system SWING to implement common user interfaces across different computing platforms. All the ontology editors and inference engines are already implemented in Java giving it a critical mass of tools to simplify the construction and rapid distribution of usable knowledge based inter-enterprise collaboration systems.

The Java/SWING implementation of the Protégé with OWL, a knowledge base editor and query tool; This integrated tool forms the user interface element of the system and works on all computing platforms with a Java Runtime Environment.

The OWL ontology implementation that will be described in detail in 6.3;

OWL implementation of the XPDL workflow engine. XPDL is an open format for the creation and persistence of business process workflows created by the Workflow Management Coalition (WfMC). The standard itself is a simple XML Data Type

Definition (DTD) and in this project the DTD was translated into OWL for integration into the overall OWL based ontology.

MySQL 4 database with JDBC driver to save the OWL knowledge base and workflow engines; Even though it is possible to save the knowledge bases in flat-file OWL format, the use of a Relational Database engine increases the performance of the ontology for queries by 90% for large systems. Thus all knowledge bases were stored locally on a MySQL database. Further details from <http://protege.stanford.edu/>

Java binding of the JXTA open-source peer-to-peer messaging standard. This binding enables the calling of JXTA functions in Java applications, and also comes with an implementation of the JXTA standard in Java with a SWING User Interface. Please see www.jxta.org for further details.

XML file defining the Virtual Enterprise members and access control through the JXTA protocol; This file is a component of the JXTA standard and includes a list of access users, their unique identifiers (UID) and a list of rendezvous peers (ad-hoc servers on a peer net). This list defines the level of access that individual members within a Virtual Enterprise enjoy.

OpenSSL Secure Sockets Layer for securing and encrypting the data flow through the peer-to-peer system. This security mechanism is implemented in the JXTA standard, and was enabled to ensure the security of sensitive product data information that flows between the nodes of the Virtual Enterprise.

Since the format implementation is essentially a core component of the 'peer to peer' implementation, thus a full description may be found below under 'peer to peer implementation'. The authentication aspect is also detailed below under 'peer to peer implementation'.

6.2 Medium Implementation (tier-1)

The implementation was kept as simple as possible. The simplicity of the architecture is itself a bonus and enabled the author to concentrate on the value adding aspects of the project. As mentioned previously, the flexible Protégé ontology development and user interface environment is used. The back end consists of the open source MySQL database with the Java database connectivity (JDBC) connector to Protégé. The knowledge base ontology is defined in a Clips file and the instances are stored in the database. Connectivity is achieved using an open source implementation of JXTA open standard P2P network protocol (www.jxta.org). The choice was made because JXTA implements a unique but anonymous identification mechanism for peers and for rendezvous peers. As well as “advertisement” implemented for all peers that give information on the one peer to other peers, rendezvous peers can act as managers for peer groups and store the peer advertisements for the group for distribution to other P2P networks. Implementation of JXTA is in Java 2 standard edition, an extension to enable RDF queries and ontologies to be shared over P2P is used to share the Protégé knowledge base.

Figure 6 illustrates the developed hybrid super-peer net architecture for inter-enterprise collaboration in egalitarian Virtual Enterprises. The yellow bubble shows an exploded view of the internal peer-net of the enterprise. Inside this internal-peer net there are a number of functional sub-systems as well as internal users. Each function within the super-peer net is a standard JXTA node (including the super-peer itself). The actual difference in the function of each node is down to the configuration selected for that node. This allows absolute flexibility for dynamically changing, modifying, adding and removing nodes within the internal peer net. It is also possible to make any one or many of the internal peer net nodes an external peer-net member via a simple configuration change. The functions used in this instance were the protégé knowledge base (this is physically co-located on the same machine in this example, but for the sake of topological accuracy has been included as a separate node). Each “F” peer is an end user accessing the internal peer net (and the external super-peer net via the super peer). The PL node has the internal workflow functions of the peer net that are accessible by the wider super-peer network via the super-peer.

Outside the peer-net is the external peer-to-peer environment which interlinks all the super-peers from every collaborator together. Note that each super-peer is connected to every other super-peer making a multiple redundant connection. In addition one, or many of the super peers within the network act as cache and rendezvous peers to enable peers or super-peers with intermittent connections to effectively participate in the peer network by having their internal shared knowledge bases stored on cache in case they are not available to serve other peers that need to access their knowledge.

This enables peers to connect to each other and query for and manipulate knowledge on different peers on the internal and wider peer-net transparently.

The architecture was set-up with a super-peer net in order to create a number of virtual servers at each of the project participant's sites. These super peers are visible as peers to both the other super-peers and the internal peer net which they serve. They also act as the default rendezvous peer, and peer information is shared between the super-peers in order to improve the redundancy and query speed of the system. Figure 6 illustrated the topology of the system.

As stated earlier the number of users was not sufficient to fully test the scalability of the applications; however the distributed nature of the applications means that the number of concurrent transactions are much lower in P2P type applications than centralised programmes. The Peer based applications showed a base performance lag when compared to the client-server applications due to routing of peer based systems without the help of a centralised indexing service. Otherwise the performance of the systems in practical use showed no noticeable differences. In theory, peer-based systems will scale to larger systems better than centralised systems because it inherently shares the load between hundreds (if not thousands) of independent 'servers' as opposed to a single central server. However this aspect could not be verified in this project due to the lack of a large enough user base to fully test performance under those conditions.

Overall the Peer to Peer architecture using the super-peer net configuration (Figure 6) fulfils the criteria for de-centralised inter-enterprise collaboration in a heterogeneous

network of egalitarian enterprises more completely than the traditional PLM based solution, or the Open source based client-server system. The peer to peer network enabled enterprises to download install and configure the JXTA system in a matter of minutes. Once the configuration information was distributed between the peer networks defining the borders of the virtual peer network, even enterprises with intermittent connections over a modem shared the same ability to serve their knowledge base with their collaborating network as a large Tier-1 company with dedicated lines and a registered Domain Name. Furthermore the simplicity of the knowledge management system and its uniform user interface across all computing platforms (Windows, UNIX, MacOS, Linux) ensures common understanding between peers using totally different computing systems. The ability to have participated in multiple VEs using the single client and knowledge base further leverages the ability of smaller enterprises to work with multiple OEMs and reuse their internal knowledge in multiple disjointed projects.

The commercial PLM system, although highly scalable and offering excellent workflow configuration utilities, was very complex to set up and modify. It is also prohibitively expensive and its business to business connections were impossible to effectively use in many-to-many inter-enterprise mappings. The cost of the system also places it beyond the abilities of smaller enterprises that form the backbone of all engineering projects (whether managed by a large OEM or developed jointly by a large number of small enterprises). The knowledge management system functioned well within the PLM system, however functionally the same was true of the Open Source implementation that did not incur the massive complexity and up-front costs of the commercial PLM.

6.3 Super-ontology KBS Implementation

The product information components (ISO STEP, WfMC, and industrial ontologies) were integrated together using the Protégé ontology editor. The system architecture for the individual components and the sample data that was used is detailed in this section.

6.3.1 Product design ontology

In order to build up a knowledge base of concept definition, customer requirement definition and engineering product / project knowledge, an example company's processes and information have to be gathered and mapped onto a specific Ontology.

There is a very flexible interpretation of what an ontology may contain based on the particular subject matter. So it can be used to store simple text, numeric information or anything else. The most important thing about the ontology is the correct decomposition and structuring of the knowledge.

The examples used in this case were, ArvinMeritor Product Development Project (PDP). And Mabey&Johnson processes data and procedures. The project data is captured, and classified, then transformed into a more “generic” terminology.

6.3.1.1 ArvinMeritor example

ArvinMeritor is a large Tier-1 automotive supplier. The investigation of concept design and development processes is being carried out with the Access Control Systems division based in Birmingham. The products of this division are security latches and powered latches for commercial and vehicles and cars. The company designs, produces prototypes and assembles the latches. The actual component production is carried out by sub-contractors. The tooling for the latch assembly is supplied by a subcontractor; however the “jig” is designed in-house.

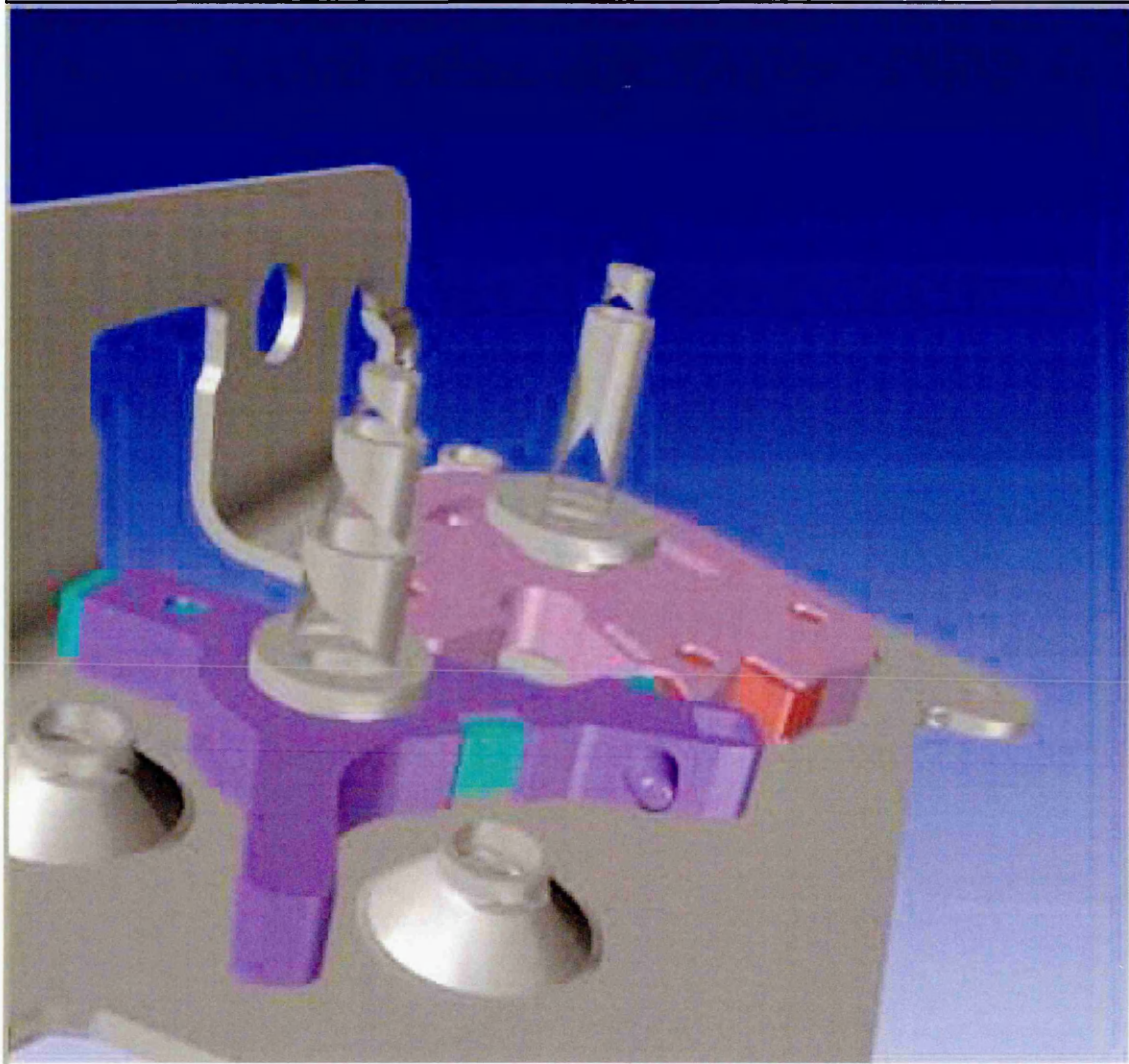


Figure 32 ArvinMeritor Latch assembly model

Figure 32 shows the main latch assembly overview in the Pro/Desktop CAD viewer. The model is defined in a STEP assembly using AP-203. The screenshot illustrates both stamped and machined components and their assemblies. Figure 33 furthermore shows the individual components that were modelled in this example implementation in an exploded assembly view. All views are shown in 3D.

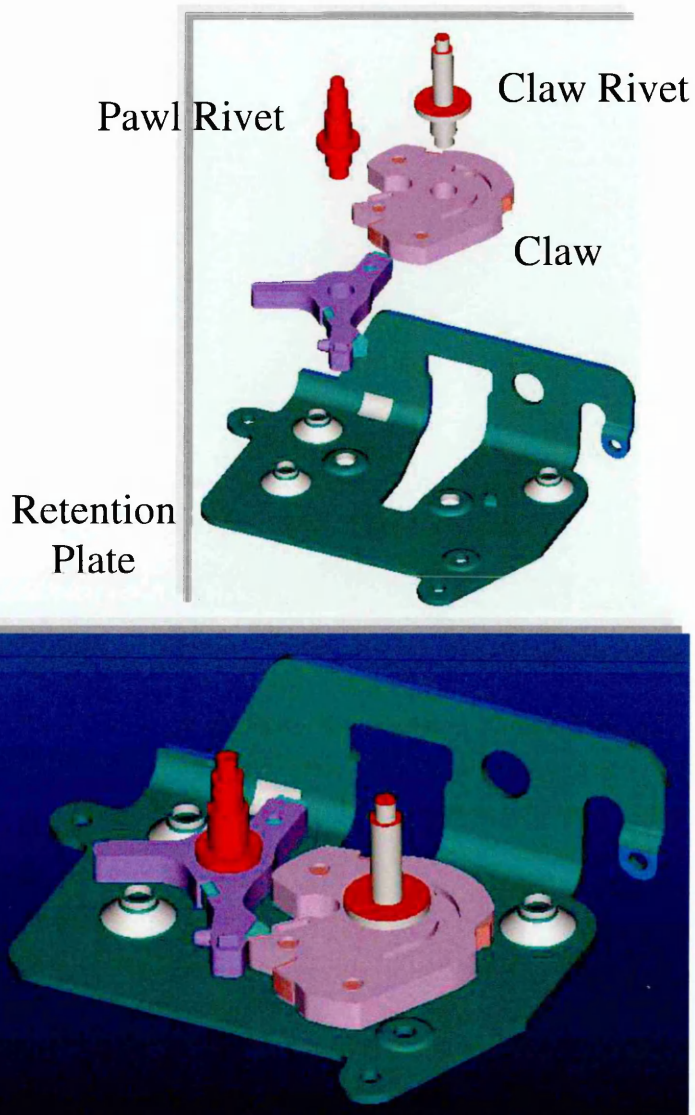


Figure 33 arvinMeritor latch assembly items

ArvinMeritor development process

There are many ways in which customer's requirements are met. Increasingly Arvin proposes technologies to the customers without their explicit requirement. Making for a more push rather than pull oriented relationship between the OEM and tier-1 supplier. Although there are still some customers who define their product specifications exactly.

Concepts are thoroughly proven before being sent to design engineering. Some of the terminology that ArvinMeritor defines are: Concept Evaluation: Phases 0 and 1 of the project “activities to identify a feasible and viable concept, to allow initial costing and assessment” see Figure 18 and Figure 19.

Tools used in this phase are initial BoM, identifying closest proven engineering elements, surrogate/synthetic costs, warranty, field and Internal PPM (pre-production) level. One-off physical and mathematical models to simulate kinematics and package size constraints.

The Concept to Customer C2C is the product development method developed and used by ArvinMeritor. It is separated into Phases 0, 1, 2, 3, 4 and 5. Phase 0 is customer/sales/marketing related. Phase 1 is concept design, Phase 2 is engineering, Phase 3 is testing and 4 is manufacture. The last phase is closure of the project / decommissioning tools. Appendix (b) has full details of the modelled lifecycle, phases and Phase 1 task breakdown.

This project is interested in Phase 1 primarily, but also the interaction with the preceding and following phases. The concept design process is irregular and made on an ad-hoc basis. The engineers believe that any rigid workflow at this stage is unworkable and liable to simply increase workload without adding benefits.

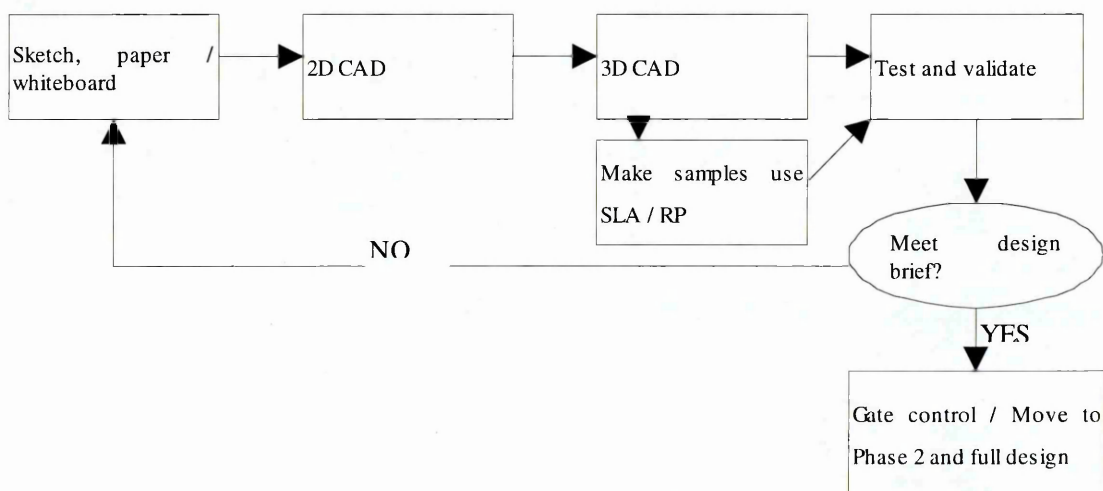


Figure 34 concept development sequence real-life example

Figure 34 shows roughly the basic steps used at ArvinMeritor for their concept developments. It starts with a sketch and whiteboard design that is then saved in a simple 2D cad file. Once the basic schematic is worked out, a 3D solid model is generated. If the model is valid (according to the judgements of the advanced engineering staff), then a stereolithography (SLA) or rapid prototype (RP) model is created, which along with the design is tested and validated. In the gate review meeting, a decision is taken on whether the design meets the design brief. If not the design goes back to the sketch board. If yes, the design is approved to be sent to the gate control and phase 2.

Currently there are major problems with unworkable designs passed onto engineering which have to be rejected, so the Test/validate gate review is not working perfectly. Sometimes a full design brief is issued to the concept engineers in order to start a project. However such design briefs are only given for so-called high-level projects with large OEMs. Note that the design brief is a presentation of concept to customer and NOT the written specification FROM the customer.

There are no specific brain-storming sessions at the company, and the only way of making new concepts and recording them, are via minutes of meetings, which are stored in the project specific folder.

The concept design engineers concede that a large amount of useful concept and engineering ideas and historical information is lost because there's no specific way of entering the information. There are however some attempts to remedy this using low-tech solutions such as creating word documents titled "Lessons learned". There is also a Variety Reduction Program to tackle the issue of knowledge management and consolidation of existing expertise and re-use of that expertise.

There are benchmarking processes made during the concept design process and this is usually a feature matrix type using an excel spreadsheet. Cost being the most important factor in the benchmarking process.

The design concept generated tends to be a “global” family of products type design rather than a specific configuration for a particular vehicle model. E.g. same global design for Nissan made at concept, with many sub-types for specific models and vehicle configurations and latch feature sets.

There are gate reviews at the end of the process and all branches are included in this review including manufacturing. All have to approve a design before it can move to the next stage. The company uses QS9000 to lay out and standardise its documentation. However QS9000 is not used in Advanced Engineering.

The specific project used as an example for this exercise was Product Development Project PDP684 (Ford and subsidiary brands). Here the OEM provided a matrix of functional requirements and some engineering specification compliance.

Within the PDP684 the company is using Ford’s APQP – Advanced Product Quality Planning which is a structured method for defining and evaluating the actions necessary to ensure a product that satisfies the customers and supplier led and is required for all system sub-system and component manufacturing locations.

The files for the project are held on a network the structure and content of the directories were mapped onto a set of tags that can be used as a skeleton for constructing a KBS and PLM vault/configuration. The layout of the files is inconsistent, there are many redundant files and duplicates in ArvinMeritor’s file structure. These were eliminated and replaced with abstract classes in the Ontology.

6.3.1.2 Customer requirements sub-ontology

To obtain customer request data and translate that into a useful design brief (i.e. creating a functional specification for the design, from a set of requirements. The requirements must be entered in a structured manner), the steps and knowledge contained are obtained in the following manner:

The customer sends a Request for Quotation. This document is split into 2 parts:

1. Product specification: What product, its performance criteria and how to test it.

2. Statement of work: Commercial info; Quality; Teams selection; Location; CAD system / standard; Delivery / logistics details; Warranty; and Payment.

Different customers approach the enterprise with different requirements. Some customers present a completed RFQ with all performance and dimensional constraints. Others come with requests for the company to develop a new specification / product standard. Some companies come clueless and want to maybe buy a “budget” component using configuration controlled parts from existing products. The main phases for getting and using customer requirements are during Phases 0 and 1.

Phase 0 - turns an enquiry into a profitable order, there are many reasons to reject enquiries and bad business should be eliminated.

Phase 1 - concept design part (This will be elaborated on below).

As an example. A copy of a Nissan Request for Quotation (RFQ) for automotive supply as sent to ArvinMeritor was used to map the customer knowledge into the expert system, see (Appendix (D) Generic Collaborative ontology, ArvinMeritor test example, Appendix (E) Generic Collaborative ontology, test data). The terminology used was decomposed to more generic forms. This maintained the neutrality of the customer requirements knowledge. From experience customers may try to “confuse” the supplier by using different formats for sending the quotation, however the “basic” knowledge contained therein is the same for most OEMs. This type of “template” approach is similar to the RFQ process definition available online through www.covisint.com

This example has not dealt with the full production/design details and capacity plans that customers specify (usually after the company is selected), but with the early, initial interaction with the customer.

6.3.1.3 Mabey&Johnson example

Mabey&Johnson’s Lydney works was used as a second test example for applying intelligent collaborative product development processes and systems. The company manufactures steel panel bridges for rapid assembly on-site. The bridge system relies on the use of standardised components and joined together in the desired configurations.

The Lydney site is tasked with interpretation of designs into 3D solid models, manufacturing and assembly planning, and machining/testing. The unit is linked via a Wide Area Network (WAN) to its parent company, which provides the designs and configurations as 2D AutoCAD models and via a proprietary Enterprise Resource Planning system called PICS.

The company uses ISO-9001 Quality Assurance and manual workflows for all tasks at the Lydney site. The problems faced by the personnel at the site include a manual and laborious link with head office which slows down communications. Company processes and procedures are all manually created and stepped through. The adherence to the Quality Standards is not assured due to the potential for personnel to forget tasks or not follow the correct sequences for completing tasks. The links between the design and manufacturing offices are very weak. Designers are primarily structural engineers who are not literate in 3D CAD and do not have access to manufacturing knowledge in order to improve the ease of manufacture of their designs.

Once designs and specifications are sent to Lydney, many changes have to be made due to the lack of Design for Manufacture and Design for Assembly (DFM/DFA) at the design stage; these engineering changes are processed manually and slowly, and involve interlinked stages between different company sites, slowing the process even more. Once changes are made to designs, and manufacturing plans, it is difficult to save the knowledge gained in a digital format that is accessible to the personnel making design and manufacturing decisions. So the only progress which can be achieved is when personnel gain very broad experience (and of course even here this is reliant on the person's memory).

The company's current IT infrastructure consists of Local Area Network with shared drives for storing CAD model data in project oriented directories. Any other documentation is stored in Microsoft Word and Excel files stored on network drives. There are some separate databases to manage process plans (used exclusively within the Lydney site and not company wide). PICS ERP system which is used to push orders and BoM requirements from headquarters to Lydney. Internet links to download 2D CAD

files from head office, usually sent as attachments to e-mails. Revision control is done manually using the QA processes, and managed by the CAD manager at Lydney. Notification, engineering changes and releases are made manually.

To alleviate this situation the company has created some applications to link the design and manufacturing processes of the company within the ERP system. However many shortfalls remain with the methods applied, briefly these include a lack of automated workflows to automate the process of managing documents, project sequences and engineering change / revision control and a lack of interaction between different departments, and links between the marketing, engineering and manufacturing sectors of the business are through a one-way push system (PICS).

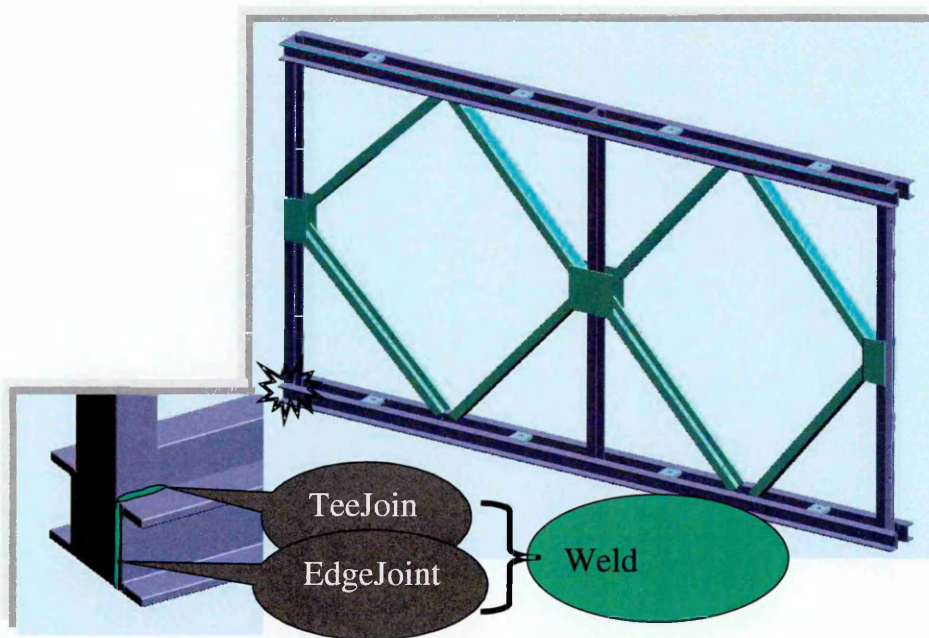


Figure 35 Mabey bridge joints

Figure 35 shows the manufacturing processes involved in constructing the bridge assembly. The items are modelled in the ontology as STEP elements.

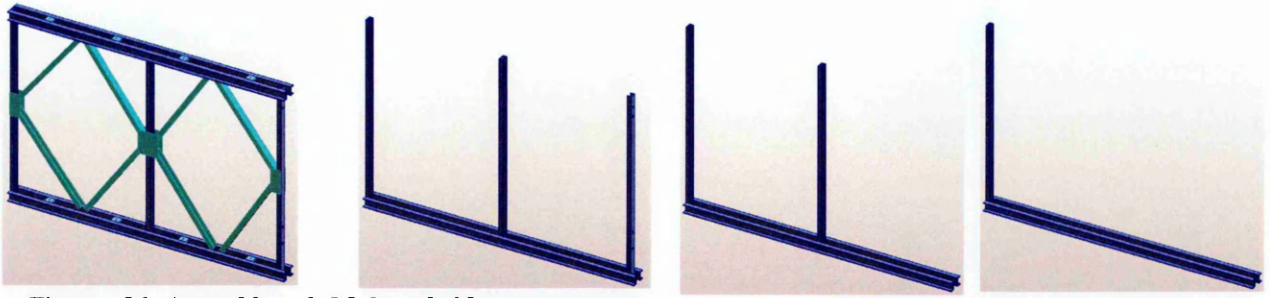


Figure 36 Assembly of Mabey bridge components

Figure 36 shows the step-by-step assembly processes of the bridge components in the system defined in the ontology as simple STEP AP-214 assemblies on top of AP-203 geometric components.

There is no framework or technology to store and reuse knowledge associated with the company's products, activities and user experiences. There are different technologies and techniques that can be used to overcome these shortcomings. Mabey&Johnson already has in-place the manual workflows and procedures to manage the project data. This can be updated by linking the separated workflows from the engineering and manufacturing together (in a similar fashion to the concept2customer framework of ArvinMeritor). Figure 26 illustrates a small subsection of the ontology structure showing some of the manufacturing processes and the information contained in the Bill of Materials.

The linked workflows enable the development to proceed in a concurrent fashion and using the approval-by-consensus manner, reduction in rework from different departments can be achieved.

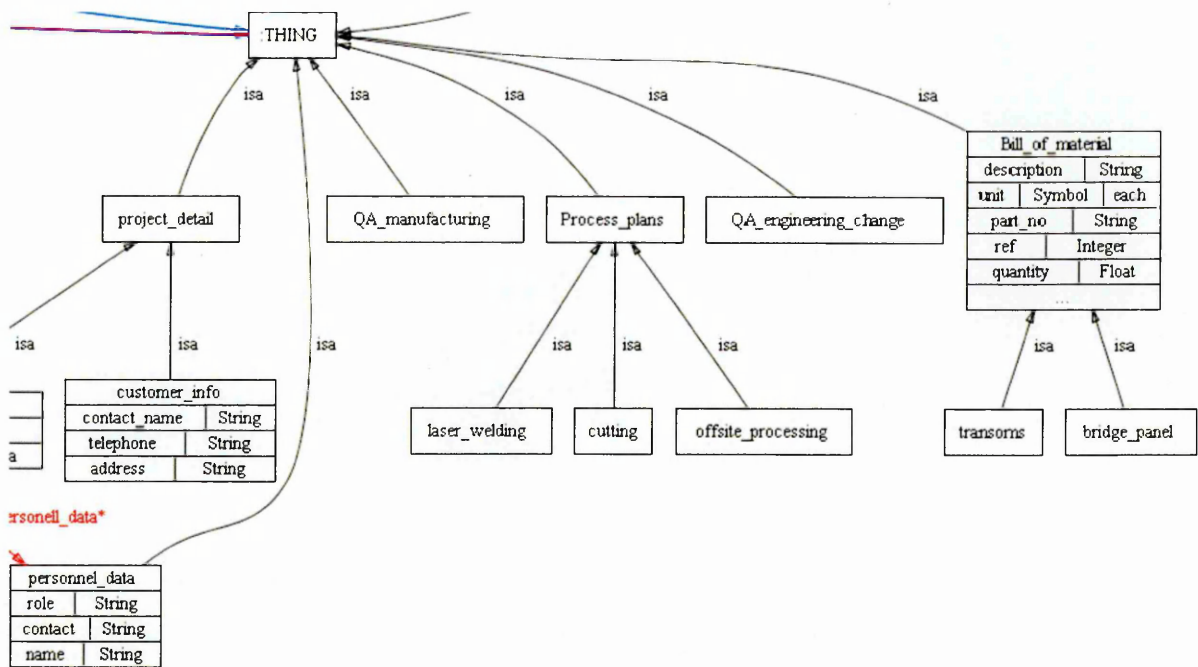


Figure 37 : RDFgraph of Mabey's knowledge structure, illustrating the hierarchical KBS ideally defining the company's data model

6.3.2 Programming methods for knowledge base

The items below are the test objects instantiated from the knowledge base. In this example ArvinMeritor company's Ford Motor company project was used as a test example. The associated source code detailed in Appendix (E) are simply a definition of all the test objects used for the example described in the main text. A number of Bills of materials and a series of existing components are defined within the system, linked together with the full lifecycle information and STEP defined entities. There are two types of objects within the collaborative knowledge base 'knowledge' and 'rules'. The knowledge objects are defined below.

To better understand the code, here is a simple breakdown of the syntax. The [Arvin_XXXXX] defines the unique object number (allowing versioning of the objects and maintaining easy synchronisation for the knowledge base. Following this 'of' defines what class the object is instantiated from e.g. in example below Bill_of_Material_item is the name of the class which object Arvin_01961 is instantiated from. Following this, every item is contained in a () separated by a carriage return. Each () contains a slot and its value or values (for slots which allow multiple values per

object). The structure is simple (SLOT value), e.g. in the example below (bom_part_no “FA11B00R”) bom_part_no is the slot name and its value is FA11B00R. Each object is separated from the other by two carriage returns.

6.3.2.1 Objects in knowledge base

The object below is an entity in the knowledge base, object number ‘23’. It is located in advanced engineering class and the data in the instance has the contact details and timesheet information for user “Gurbinder Kalsi”.

```
(([arvin_00023] of advanced_engineering
  (contact_name "Gurbinder Kalsi")
  (contact_address "birmingham")
  (role "concept design engineer")
  (contact_number 1234111104)
  (timesheet_instance [arvin_00107])))
```

The object below is a Protégé Axiom Language constraint on the knowledge base. This particular constraint limits the ability of users to select only weekdays for timesheet data.

```
(([arvin_00102] of %3APAL-CONSTRAINT
  (%3APAL-DESCRIPTION "constrains the selection of a single weekday for a particular timesheet")
  (%3APAL-NAME "week_day_constraint")
  (%3APAL-STATEMENT "(Predicate)")
  (%3APAL-RANGE "monday\ntuesday\nwednesday\nthursday\nfriday")))
```

This object is a timesheet template for a user defining the date and the number of hours worked during that day.

```
(([arvin_00107] of timesheet_template
  (year 2002)
  (hours_during_day 1)
  (day_of_week thursday)
  (week_number 23)))
```

This object is a reference object linking a user’s timesheet item to a client’s project.

```
(([arvin_00156] of client
  (timesheet_instance [arvin_00157])))
```

This object is from sub-project ten of the product development process, and defines the lifecycle status of this phase of the project. This is ‘in_Plan’ i.e. still in planning and the phase is behind schedule ‘Phase Status Late’.

```

([arvin_00181] of 10_Plan_project
  (Phase_Evidence Fully_Verified)
  (Phase_item Develop_project_team_meeting_requirements_and_expectations)
  (Phase_plan In_plan)
  (Phase_Status Late))

```

Develop Preliminary Advanced Research, is also an initial phase of a project lifecycle, and this object defines its current status 'not planned' and the project phase is still 'on_target' within its schedule.

```

([arvin_00182] of 14.3_Develop_preliminary_AR
  (Phase_Evidence Incomplete)
  (Phase_item Establish_project_budget)
  (Phase_plan Not_planned)
  (Phase_Status On_target))

```

The customer information object here is an item concerning compliance with government regulations and this item is still in the planning state. The item is marked as a potential Risk to the project.

```

([arvin_00183] of 7_Customer_information
  (Phase_Evidence None)
  (Phase_item Confirm_government_regulations)
  (Phase_plan Being_Planned)
  (Phase_Status Risk))

```

This object of the Materials class defines the name of a material used in the product and the companies that can potentially supply the material.

```

([arvin_01493] of Materials
  (material_type "CR1_Aluminium_killed_FeP13")
  (potential_supplier [arvin_01494]))

```

This item shows that multiple companies may be associated with a material aiding the developers in having a more holistic overview of their supply chain's abilities.

```

([arvin_01513] of Materials
  (material_type "Stainless_steel_BS5770_part4_301s21")
  (potential_supplier
    [arvin_01512]
    [arvin_01514]))

```

The Supplier Class instantiates a separate object defining each of the companies that can potentially collaborate on a project, with their basic details. These can then be

referenced from individual objects in other classes where the company has expertise or can supply a particular material.

```
((arvin_01494) of suppliers
  (contact_name "Mr Smith")
  (company_name "gammastamp")
  (contact_address "birmingham"))
```

6.3.2.2 Classes for STEP ontology

The source code below shows the class and variable 'slot' definitions of the ontology defining the STEP application protocols (AP-224). Defclass is a command declaring that a class is about to be defined. The TOP_LEVEL-SLOT_CLASS is actually a definition of the reusable variables or slots which are available for use within the ontology by the different classes. These are 'top-level' slots, meaning that any class in the ontology may utilise them.

The role of the class can be either as an abstract or concrete class (Abstract cannot be instantiated), IS-A defines the inherited properties from a super class.

For every slot definition, single_slot defines that each object may have one instance of that variable, or multiple instances of the variable can be defined with multiple_slot and this can either be limited to a certain number of instances or unlimited.

Type defines the data type of the slot, which can be integer, string, floating-point, instance of a class (or multiple instances of a class), or a symbol (pre-defined multiple-choice list).

Create-accessor defines the access rights and privileges for that slot within the overall ontology.

The actual items defined as top level slots are the same as those defined in EXPRESS for the STEP standard. These include mundane items like axis2_placement which is a symbolic variable with a choice of potential values being 2D or 3D (i.e. defining whether the second axis is within a 2D or 3D path).

Other definitions include `attribute_select` which is also a symbolic variable with multiple choice values including `property_definition_representation` `external_source_representation` `date_role` `person_and_organization` `approval_role` `organization_role` `action_request_solution` `person_and_organization_role` `application_context`. This slot is used in classes to define the role of the instantiated object with one of the above attributes.

There are almost 100 variable 'slot' types defined for the STEP ontology alone, a slot is analogous to a field. Some of these are included below.

```
(defclass %3ACLIPS_TOP_LEVEL_SLOT_CLASS
  (is-a USER)
  (role concrete)
```

A dimension count detailing whether the object is a 2D or 3D object. The range of the slot is an integer and limited from 1 to 3 dimensions. Cardinality is set at 0 1, meaning that the field may be left empty or else only one value may be entered per object. Create-accessor defines the access control for the object which is read-write ability to anyone allowed to access the knowledge base.

```
(single-slot dimension_count
  (type INTEGER)
  (range 1 %3FVARIABLE)
  (cardinality 0 1)
  (create-accessor read-write))
```

This slot defines the `postal_code` variable that can be used in classes in the systems' address book related functions. It is set as a string thus not restricting the input of different countries' unique forms of postal codes.

```
(single-slot postal_code
  (type STRING)
  (cardinality 0 1)
  (create-accessor read-write))
```

This multiple choice value defines the role of the STEP object, whether it's a product model, a product definition formation (in other words a concept design) or a product definition (functional specification).

```
(single-slot product_or_formation_or_definition
  (type SYMBOL)
  (allowed-values product product_definition_formation product_definition)
  (cardinality 0 1)
  (create-accessor read-write))
```

This slot defines the placement of the second axis either on a 2D or a 3D geometry.

```
(single-slot axis2_placement
  (type SYMBOL)
  (allowed-values axis2_placement_2d axis2_placement_3d)
  (cardinality 0 1)
  (create-accessor read-write))
```

This multiple-choice slot defines the attribute of the model that is to be defined.

```
(single-slot description_attribute_select
  (type SYMBOL)
  (allowed-values property_definition_representation external_source representation
date_role person_and_organization approval_role organization_role action_request_solution
person_and_organization_role application_context)
;+ (cardinality 0 1)
  (create-accessor read-write))
```

6.3.2.3 JESS rules within the ontology

The function classes below are simply abstract definitions for the system classes which instantiate functions and rules within the Protégé system. These are core to the Protégé system and the classes which define the 'rule' entities of the STEP standard and the inter-enterprise knowledge base are instances of the subclasses of class JESS-Definition below.

```
(defclass %3AFUNCTION
  (is-a %3ASYSTEM-CLASS)
  (role abstract)
  (single-slot %3ADEFINITION-NAME
    (type STRING)
    (cardinality 0 1)
    (create-accessor read-write))
;+
```

```

        (single-slot %3ADEFINITION
          (type STRING)
          (cardinality 0 1)
          (create-accessor read-write)))
;+

```

This abstract class is the root class for the main rule engine classes. It declares the global variables which are included in the concrete subclasses. The declared variables simply include the name of the 'object' a string containing the source code as a string and whether the object is saved in the knowledge base or flushed.

```

(defclass %3AJESS-DEFINITION
  (is-a %3ASYSTEM-CLASS)
  (role abstract)
  (single-slot %3ADEFINITION-NAME
    (type STRING)
    (cardinality 0 1)
    (create-accessor read-write))
  (single-slot %3ADEFINITION
    (type STRING)
    (cardinality 0 1)
    (create-accessor read-write))
  (single-slot %3AKB-SAVE
    (type SYMBOL)
    (allowed-values FALSE TRUE)
    (cardinality 0 1)
    (create-accessor read-write)))
;+

```

Deffacts are subclasses of JESS_definitions that define concrete 'facts' within the system (constants). It is a concrete subclass of jess-definition and the objects instantiated are the constants within the knowledge base.

```

(defclass %3AJESS-DEFFACTS
  (is-a %3AJESS-DEFINITION)
  (role concrete))

```

The defglobal function within JESS defines static and global constants within the knowledge base. It's a concrete class and the instantiated objects are stored in the knowledge base.

```

(defclass %3AJESS-DEFGLOBAL
  (is-a %3AJESS-DEFINITION)
  (role concrete))

```

Defmethods are routines which can run on knowledge base (defined in KIF). The routines are instantiated as objects of the class, and also inherit the base properties from jess-definition.

```
(defclass %3AJESS-DEFMETHOD
  (is-a %3AJESS-DEFINITION)
  (role concrete))
```

The classes below are used by the JESS engine, and included for reference purposes.

```
(defclass %3AJESS-DEFTEMPLATE
  (is-a %3AJESS-DEFINITION)
  (role concrete))
```

The JESS-RULE class below is a concrete class which is a subclass of JESS-Definition. This is the class which is used for instantiating rules in the JESS expert system engine. Each rule is an object of the class below and inherits the variables found above. The instances are for both the STEP standard rules and the generic knowledge base workflow and rule engine.

```
(defclass %3AJESS-RULE
  (is-a %3AJESS-DEFINITION %3ARULE)
  (role concrete))
```

6.3.2.4 STEP Ontology

The following concrete classes below define the Express schema for the STEP standard AP-224 translated into KIF. The class names and types are directly equivalent to the STEP standard entities, and have been translated to a KIF knowledge base as a component of the integrated knowledge base. The STEP documentation (www.iso.org) fully documents each class (entity) in the schema, however a number here are also documented for the sake of brevity and to clearly understand how the STEP schema can be translated from Express to KIF.

```
(defclass _TYPE
  (is-a USER)
  (role abstract))
```

This class defines the structure of the object identifiers in the STEP models.

```
(defclass _T_identifier
  (is-a _TYPE)
  (role concrete))
```



```

        (single-slot identifier
          (type STRING)
          (cardinality 0 1)
          (create-accessor read-write)))
;+

```

This class defines the items in a feature based process plan item

```

(defclass _T_feature_based_pp_ordered_item
  (is-a _TYPE)
  (role concrete)
  (single-slot feature_based_pp_ordered_item
    (type STRING)
    (default "product_definition_formation")
    (cardinality 0 1)
    (create-accessor read-write)))
;+

```

This class defines whether the topology of the geometry is set to reversible or not.

Boolean answer.

```

(defclass _T_set_of_reversible_topology_item
  (is-a _TYPE)
  (role concrete)
  (single-slot set_of_reversible_topology_item
    (type SYMBOL)
    (allowed-values FALSE TRUE)
    (cardinality 0 1)
    (create-accessor read-write)))
;+

```

This class defines the STEP model's value qualifier to one of three possible values 'precision' 'type' or 'uncertainty'.

```

(defclass _T_value_qualifier
  (is-a _TYPE)
  (role concrete)
  (single-slot value_qualifier
    (type SYMBOL)
    (allowed-values precision_qualifier type_qualifier uncertainty_qualifier)
    (cardinality 0 1)
    (create-accessor read-write)))
;+

```

Instantiations of this class are used to define the unit of measurement used for different objects.

```

(defclass _T_si_prefix
  (is-a _TYPE)
  (role concrete)
  (single-slot si_prefix

```

```

                (type SYMBOL)
                (allowed-values exa peta tera giga mega kilo hecto deca deci centi milli micro nano pico femto atto)
;+                (cardinality 0 1)
                (create-accessor read-write)))

```

Ratio definitions in the STEP models are instantiated from this class as floating point numbers.

```

(defclass _T_ratio_measure
  (is-a _TYPE)
  (role concrete)
  (single-slot ratio_measure
    (type FLOAT)
;+    (cardinality 0 1)
    (create-accessor read-write)))

```

Objects of this class define the shape of the model whether it's a complete 'product definition shape' a 'shape aspect' or an 'aspect relationship' between objects.

```

(defclass _T_shape_definition
  (is-a _TYPE)
  (role concrete)
  (single-slot shape_definition
    (type SYMBOL)
    (allowed-values product_definition_shape shape_aspect shape_aspect_relationship)
;+    (cardinality 0 1)
    (create-accessor read-write)))

```

This class defines the properties for the characterised material property attribute of the STEP model.

```

(defclass _T_characterised_material_property
  (is-a _TYPE)
  (role concrete)
  (single-slot characterized_material_property
    (type STRING)
;+    (value "material_property_representation")
;+    (cardinality 0 1)
    (create-accessor read-write)))

```

This class allows the definition of the document reference item from a list of possibilities for the STEP model. The potential variables are directed action, property definition, action method, externally defined feature definition or dimensional characteristic definition.

```

(defclass _T_document_reference_item

```

```

(is-a _TYPE)
(role concrete)
(single-slot document_reference_item
  (type SYMBOL)
  (allowed-values directed_action property_definition action_method
externally_defined_feature_definition dimensional_characteristic_representation)
;+ (cardinality 0 1)
(create-accessor read-write)))

```

The objects instantiated from this class of the STEP model defines the limit condition of the model and have three possible values detailed under 'allowed values' below.

```

(defclass _T_limit_condition
  (is-a _TYPE)
  (role concrete)
  (single-slot limit_condition
    (type SYMBOL)
    (allowed-values regardless_of_feature_size least_material_condition maximum_material_condition)
;+ (cardinality 0 1)
(create-accessor read-write)))

```

6.3.2.5 Generic collaborative ontology

As per the STEP ontology above, the generic collaborative ontology is also defined using the KIF like language in Protégé which can be interpreted by agent based systems, RDF based data-mining systems and expert systems. Explanations of the source code syntax are detailed in the sub sections above. Listed below are brief extracts illustrating the ontology. The complete code can be found in Appendix (d)

First the Slots are defined (the 'object types'):

```

(defclass %3ACLIPS_TOP_LEVEL_SLOT_CLASS
  (is-a USER)
  (role concrete)
  (single-slot quantity
;+ (comment "quantity of items in BoM")
  (type INTEGER)
  (default 1)
;+ (cardinality 1 1)
(create-accessor read-write))

```

The data structure 'classes', each of which includes one or more of the slots above:

```

(defclass reliability
  (is-a Specifications)

```

```

(role abstract)
(single-slot specification_value
  (type SYMBOL)
  (allowed-values FALSE TRUE)
;+ (cardinality 0 1)
  (create-accessor read-write))

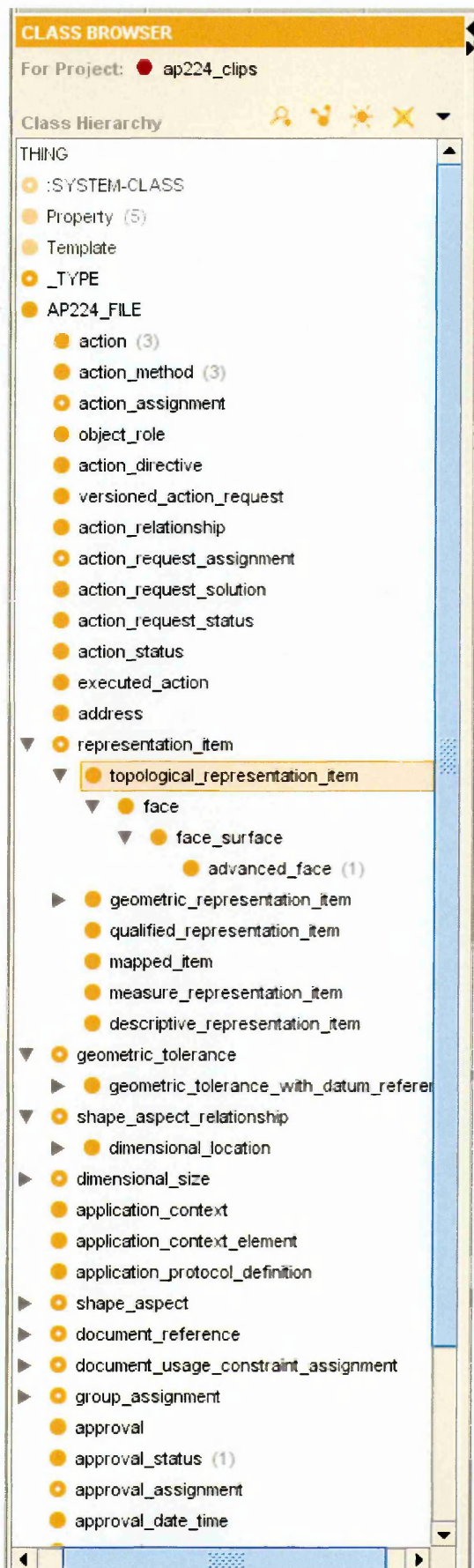
(defclass ajar_switch
  (is-a control)
  (role concrete))

(defclass system_integration
  (is-a Specifications)
  (role abstract)
  (single-slot specification_value
    (type SYMBOL)
    (allowed-values FALSE TRUE)
;+ (cardinality 0 1)
    (create-accessor read-write))
  (single-slot specification_critical
    (type SYMBOL)
    (allowed-values FALSE TRUE)
;+ (cardinality 0 1)
    (create-accessor read-write))
  (single-slot Specification_item_value
    (type INSTANCE)
;+ (allowed-classes Specification_items)
;+ (cardinality 0 1)
    (create-accessor read-write))
  (single-slot Specification_ref
    (type INSTANCE)
;+ (allowed-classes Specification_reference)
;+ (cardinality 0 1)
    (create-accessor read-write)))

```

6.3.3 User Interface

The screenshots give a general overview of the user interface and class structure of the system. Overall there are over 7000 frames (objects and classes) in total in the test system that was implemented. Full source code is in Appendix (c) for the step ontology code, Appendix (d) for the generic collaborative ontology code and Appendix (e) for the test data).



- ▶ ● _TYPE
- ▶ ● AP224_FILE
- user_interface

Figure 38 (above) superclasses containing the ontology under types, AP-224 entities and user interface elements

The geometric and manufacturing features as defined in STEP AP-224 are accessible via the user interface shown in Figure 38. The _TYPE superclass contains the elements containing all the types in the STEP standard. The AP224_file, contains a superclass and subclasses of all the classes defined in ISO-STEP224 (the AP224 classes utilising the element types defined under superclass _TYPE. The actual usable elements are contained under 'user_interface' which presents a logical easy to use view of the AP224_file ontology for the collaborating professionals to enter and query knowledge and verify the input within the declared and implied constraints of the knowledge base.

Figure 39 (left) AP 224 class structure

Figure 39 Illustrates the subclasses and structure of the AP224 sub-ontology. As can be seen, invoking the AP224 superclass enables the user to directly manipulate the components of the AP224 file that are not purely geometric such as action, object_role,

approval and thousands of other elements defined in the STEP standard but not supported by modellers.

As can be seen in the example, classes with associated objects have a grey bracket to their left with the number of objects between the brackets.

As far as was possible, within the confines of the research project, the AP-224 standard was implemented in this ontology to the following extent:

100% of the types

35% of the classes

10% of the rules

The entire implementation can be found in the Appendix (d).

Even with such a seemingly small percentage implemented, the system came to over four thousand classes (excluding the generic requirement/specification sub-ontology and workflow sub-ontology).

Figure 40 illustrates the material selection window that lists all available material types in alphabetical order. Each of these are instantiations of the material class. Each material is then linked to a number of components that are manufactured using that material and in turn to the companies that have the capacity to manufacture those components.

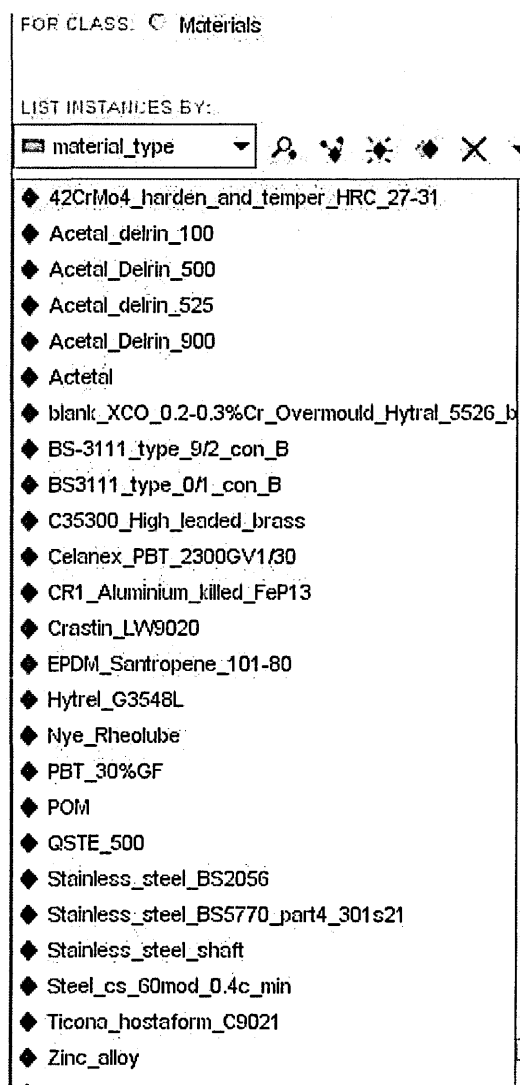


Figure 40 Material selection window

The ArvinMeritor and Mabey&Johnson data were integrated into a single ontology and redundant elements with differing nomenclature were eliminated. The structure of the ArvinMeritor ontology was much more feature complete than Meritor as they have a very complex product and product development process.

Figure 41 is split into two separate screen dumps of the class hierarchy within the user interface element of the knowledge base. This is the submenu for defining the functional specifications of the project (this data is reusable because each specification items maintains a link to the requirement from which it was derived). Each of the functional specification objects are kept in the same repository throughout the lifecycle, and can be reused and used for evaluating new concept designs in future projects. Each

enterprise in the collaborating Virtual enterprise has a copy of the blank ontology, but only their internally developed data is filled in. External data is visible but not editable, this is controlled via the access control mechanism built into JXTA.

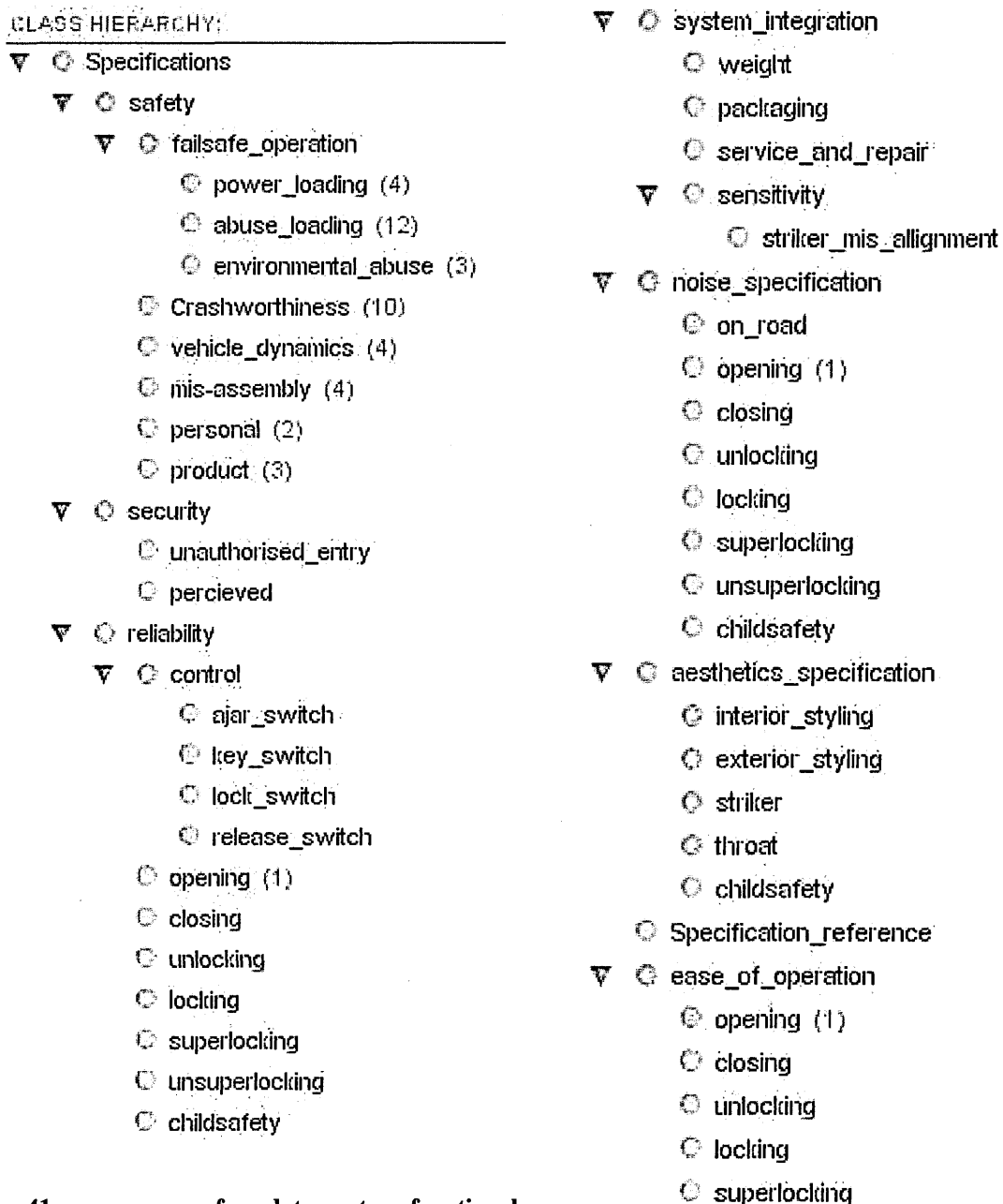


Figure 41 user menu for data entry functional specifications

In the above examples it can be seen that the specifications are laid out in a hierarchical menu under the specifications superclass. Each of the sub-topics such as safety, security, reliability, noise_specifications have a series of sub-classes that define the individual properties of the design. Double clicking on any of the concrete (instantiable) classes brings up the data entry window where the object can be created.

Existing objects in the class are indicated by the brackets next to the class name, as well as the number of objects defined in that class.

latch_type	issue_in	BOM_Ma	quantity	Customer	product	bom_pis	BOM_w	prototyp	product	Latch_L	bom_se	bom_ref
Rod	3	CR1_Alu	1			FA10B03T 60.0			Boat_ji...	RIGHT	003M5...	130
Powered	2	1		SMR		FA16P0...		Bentley	Actuator...	RIGHT	003M5...	258

Figure 42 Assembly management and creation interface

Figure 42 shows the assembly management and creation interface. Here individual items or subassemblies can be added to the assembly, which at the same time has configuration information defined in the left pane. This also restricts the selection of components based on the component configuration. Furthermore components from existing designs can be reused for concept construction and new components developed from them. The right pane displays all relevant information about each of the components. Furthermore it is possible to click on each component and change its configuration.

The screenshot shows a software interface for configuring an assembly component. The window title is "Sill-lever_independent_lock_LH (type=Bill_of_Material_items, name=arvin...)". The interface is divided into several sections with input fields and dropdown menus. The fields are as follows:

- BOM OEM PART NO.:** GO3N6A-R21813-PIA004
- PRODUCTION SUPPLIER:** Barkley
- BOM PART NO.:** FA11J02R
- PROTOTYPE SUPPLIER:** SMPI
- ISSUE LEVEL:** 1
- CUSTOMER:** (empty field)
- PRODUCT DESC.:** Sill-lever_independent_lock_LH
- BOM REFERENCE NO.:** 201
- QUANTITY:** 1
- LATCH TYPE:** Independent
- BOM WEIGHT:** (empty field)
- LATCH LOCATION:** LEFT (dropdown menu)
- BOM MATERIAL:** (empty field)

Figure 43 Configuration of assembly component

The assembly configuration is on the left (defining the type and location of the assembly), and the right pane shows the individual components within the assembly that can be dropped in. The user interface supports both drop down selection of components and drag-and-drop of new components from a pane.

The assembly is automatically updated if any of the components are modified from anywhere. The geometric data is associated for each component separately under the AP-224 classes which also defines, if applicable, the machining features for the components.

Only after defining the components may they be added to an actual product assembly.

There are various interfaces available for the user to extract information out of the knowledge base. These include a library of pre-defined queries (Figure 44) that users may activate manually, or via rules that trigger automatically when an object is created or modified.

The queries include:

- ☒ Selection of components based on location of latch;
- ☒ Selection of components based on the type of latch;
- ☒ Automatic retrieval of components that have been modified within a certain (user-defined) time-frame; and
- ☒ Comparison of existing product specification with new product RFQ.

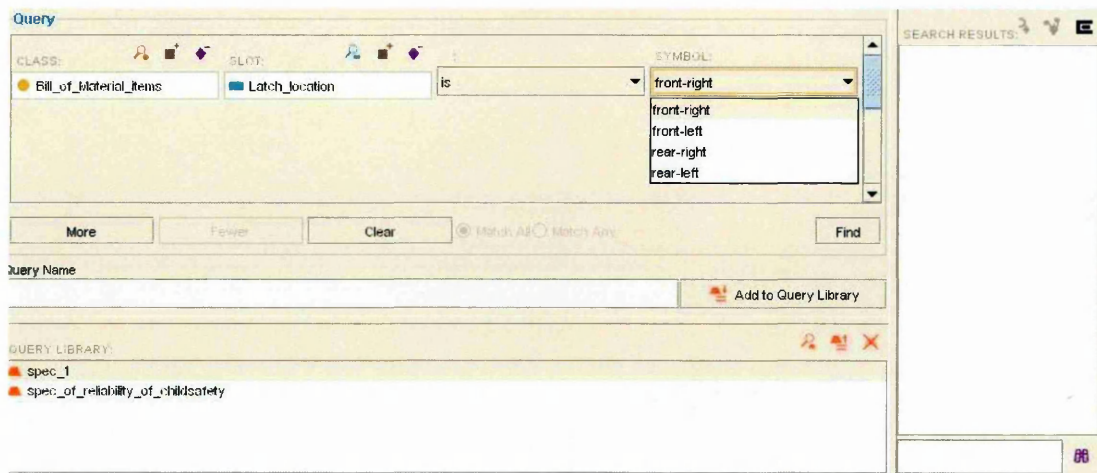


Figure 44 Ready-made queries to actively assist in configuring new designs using existing knowledge

These queries allow the end-users to make full use of existing domain knowledge when designing new products or configurations. The easy-to-use user interface allied with the dynamic updating of content enables multiple users to work on the product design concurrently.

The screenshot in Figure 44 shows an overview of the query interface of the system. The bottom left pane shows the list of pre-defined queries, the top left shows the currently selected query into which the user can type variable information (if needed). The right pane shows the output of the selected query. All items in the output are objects.

The knowledge base can also be traversed graphically, by following the relations between objects using the TouchGraph visualisation library (Figure 45). This interface has the advantage of allowing the user the freedom to explore new ideas by presenting him with all related objects within a user defined radius. An interesting aspect that can be seen in the figure is that explicit relationships between classes and instances of knowledge can be graphically seen (over a radius of one or many hops) giving the engineers a very high level oversight of the project and its dependencies and interfaces without having to make any extra effort to define those relationships. The interface is however impossible to use, in practice, if there are more than 40 objects within radius

due to the limitation on the number of objects that can appear on screen at any one time (dark yellow items are classes, brown items are instances in knowledge base).

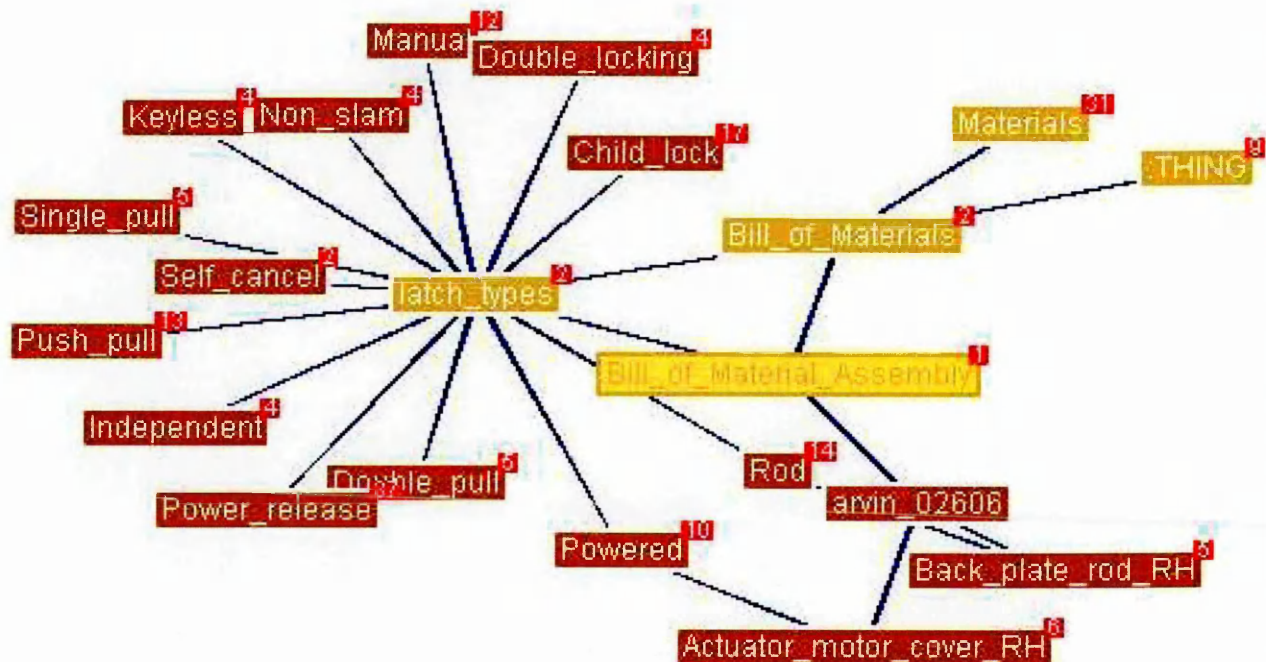


Figure 45 Exploded view of knowledge base hierarchy, enabling the graphical navigation of the system.

6.4 Summary

The test implementation of the system was the main component of the research work undertaken. It included an extensive programming phase where a number of different software configurations were developed, setup and functionally demonstrated using the limited knowledge base examples above, namely to aid the inter-enterprise collaboration of heterogeneous small to medium sized enterprises in complex engineering lifecycles and aid them in the rapid creation of concepts from specifications and collaboratively developing engineering products manufacturable at the distributed heterogeneous manufacturing facilities of the collaborating enterprises.

The industrial example, due to its size and complexity was split into several complementary subsections dealing in turn with the core aspects of the system. The first one dealt with the format and medium of collaboration. Here a new two-tier architecture

was innovated to overcome the perennial problem with business-business enterprise system mapping.

Analysis of enterprise data models and functional specifications of high-end engineering projects in conjunction with an implementation of ISO STEP product model standards were implemented in a unified semantic-web compliant ontology that overcomes the lack of widespread support of the EXPRESS language in the world wide web, as well as creating a superset ontology for general purpose use by engineering enterprises for both in-enterprise as well as for collaborative purposes.

This ontology was integrated with the three implemented systems (PLM, open source application server and P2P framework). The three systems were a commercial transactional PLM system, an open-source client-server document management system modified into a hybrid PLM/KM system and a new peer to peer system using an open standard JXTA platform for inter-enterprise collaboration and fully implementing the new two-tier inter-enterprise collaboration architecture.

The performance of the three systems did not show major variations, it is to be noted however that the topology of the peer-peer system would show performance advantages over the two client-server systems because it does not have a single bottleneck for bandwidth and processing. This level testing would only be possible with several hundred concurrent users and this was not available within the confines of this research.

The ontology and industrial example of the knowledge base functioned flawlessly and was usable from all three systems without problem. The user interface was intuitive for a small set of unaccustomed users selected from among Cranfield and Durham university students. It has to be noted however that all students were postgraduate engineers (this however reflects the average level of technical ability for potential users of such a system). It is to be noted that no large scale usability testing was carried out on the application and the basic use-case scenario with a number of students is sufficient evidence to verify the usability of the application for basic research, if not large scale operational purposes.

Since all three communication architectures implemented the same ontology and user interface, the difference between them came down to compliance with the ideals of an egalitarian and easy to integrate inter-enterprise system. In this regards only the peer-peer system fulfilled the requirements for a virtual enterprise system.

For details on deploying and running the system please see Appendix (a).

7 Discussion, Conclusions & further work

7.1 Discussion

Overall results of the research have shown that large enterprise PLM systems like Windchill are unsuitable for SME enterprises working in partnership with a large number of other small enterprises in a VE due to inherent centralisation, overly complex set up requiring many months for customisation and additionally the enterprise-specific customisation carried out cannot be reused on a system from another vendor.

High end PLM systems such as Windchill have a sophisticated application layer for inter-enterprise communications, and the inherent weakness of this is the client/server paradigm whereby every enterprise needs to have custom integration between it and its extended enterprises' systems to function. This has been elaborated on in the previous sections. On another note, the RosettaNet standard, implemented in Windchill, can decrease the amount of customisation needed, but not eliminate it as RosettaNet only provides a subset of the messaging and data models of the inter-enterprise link. Finally, unlike the ontological 2-tier system, Windchill's data models and workflows are neither portable nor standardised.

Other issues encountered were the difficulty in modifying the data models. Windchill has a rather rigid data modelling interface based on the Unified Modelling Language (UML) to model the Java data model classes, and generate SQL scripts for the database. It can be modified by a professional who has to model, program, compile, update the database and integrate the code into Windchill before it can be used. This does not provide for the flexibility and ease of use for modelling ontologies that the project manager needs.

The costs of implementation, licences and maintenance are very large by themselves and only practical when the enterprises involved can share the cost between each other over a long period of co-operation (as in an extended enterprise). However, the case

with VEs, in which a partnership is valid for the relatively short period of a single project, this would be unacceptable.

The Windchill and open source systems are the evolution of product development management systems from document centric PLM systems to knowledge centric, intelligent systems of the future. However, it had been discovered during the course of the project that centralisation, by its very nature, is an inhibiting factor for one-off inter-enterprise collaboration as the only methods available for collaboration in such an environment are B2B schema mapping integration, centralised web-enabled system or use of an Application Service Provider.

The requirements set out in section 1.2 are fulfilled by the new methodology: Requirement 1 is satisfied by the use of domain specific ontologies distributed to the clients (users) via the P2P network and open standards: Elimination of the interoperability issues for product and project knowledge, enabling small enterprises to implement application standards that were beyond their means, such as STEP-PDM, and easier set up for inter-enterprise collaboration.

Requirement 2, is fulfilled through the use of RDF and OWL ontologies that can be queried by agent based systems and transformed by KIF based rules. The information contains the URI and meta-data eliminating the ambiguous context of the knowledge. Efficient query and retrieval mechanisms, intelligent agents to function on context aware information and provides an object oriented representation of an enterprise's data model.

The third requirement is met by de-centralisation: Elimination of centralised bottlenecks in bandwidth and resources, empowerment of collaborators within networks to “control” the knowledge they create, solving the management of intellectual property rights within VEs, elimination of centralised administration that results in reduction in cost and complexity and enable domain professionals to tailor the system. This, however, has a cost in terms of ease of maintenance and centralised control, as well as some query speed problems encountered due to the lack of a complete indexing service.

In addition the methodology described can be:

- (i) Open source: Elimination of software licence costs, a solution to the problem of vendor lock-in in the long term, elimination of unnecessary complexity and freedom to modify the application; and
- (ii) Platform and application independence: Enable the enterprise to concentrate on its work and not be tied in to any vendor, the rapid and complete migration from the proposed system to future applications, empower the enterprise to leverage its existing investments.

Comparison with other research in the field

It is impossible to make a direct correlation with the work of this research and that of other research in the field as each research focused on slightly different aspects of inter-enterprise collaboration and knowledge management and with different presumptions.

(Byrne 1993), (Barnett et al 1994) and (Hardwick et al 1996) laid out the foundations for defining the Virtual Enterprise in the earliest works on the topic. At that time, the telecom infrastructure available did not support the technical running of true VEs due to the high cost and low availability of bandwidth that would be needed to maintain Peer to Peer connections between multiple participants in a VE. This research is thus an implementation of some of the VE topology that the above mentioned research developed. Much of the detail created such as the super peer net architecture, and integration with ontology based KM systems were not discussed in the earlier research, and in that aspect this research explores the area of inter-enterprise interoperability in greater depth. (Zhou and Nagi 2002) and (Zha and Dn 2001) implemented a solution for VEs using STEP and extending the data model using express to contain the enterprise level knowledge not defined in the standard STEP APs. This is a slightly different approach to the one taken by this research, and has the strength to direct interoperability with CAD systems, and the weakness of its use as a shared knowledge base that can be sliced and diced in an easy manner and crawled by RDF/OWL agents. Zhou's work is also client-server based, which, once again brings the system back under the 'control' of a single party within the VE. Building on the work of (Beckett 2003) the project took

into account his observation that a true VE should function within the scope of a one-off project. Towards this end, this research constructed a 'super-ontology' usable between partners in real time and modifiable using the building block elements to eliminate entities not relevant to a particular project. The above mentioned issue was also raised as a core problem with the implementation of VEs by (Camarinha-Matos and Afarmanesh 2003).

Comparing this work with some of the research works published on the topic of inter-enterprise integration also shows the overlap between this implementation and that of other researchers and also illustrates that different approaches to solving the problem of inter-enterprise integration are possible, depending largely on the types of enterprises collaborating, the nature of their collaboration and the hierarchy of the inter-enterprise relationship. (Ulieru et al 2000) for example continues with the general trend of deploying a centralised system to manage inter-enterprise relationships (which is fine in a hierarchical relationship between the major enterprise and its 'collaborators'). Of course this solution is particular to situations where there is a large controlling enterprise, and that is not the case in the scenario that this research is aimed at, the different scenarios of inter-enterprise collaboration being clearly illustrated by (Jagdev 2001).

Comparing this work to other researchers implementing peer-to-peer topologies for business collaboration also shows that the often narrow focus of different research teams differentiates their work from the 'enveloping' system created within this work. Peer to Peer research work such as (Oram 2001) are focused on the creation of and running of de-centralised services at a higher level of abstraction than the enterprise-oriented implementation. Much of the peer-to-peer level research focuses on improving the performance of peer discovery and queries over the peer nets (Jovanovic et al 2001), (Lu and Callan 2003) and (Lv et al 2002). Others have concentrated on the problems of 'reliability assessment' of peers on a p2p network (Damiani et al 2002). The only point of comparison is then the performance of queries / peer discovery and how it correlates to the findings of these researchers. The implementation of different topologies in this research has shown quite clearly that pure p2p networks are unworkable in practical

terms due to the time it takes to query every possible node on the internet, it was seen that due to the practicality of timing out a query after a set amount of time, only peers within a close geographic proximity (from a network topology perspective and not necessarily correlating 100% to geographic proximity) were found in queries. Implementation of centralised servers to manage peer information, of course works, however that defeats the aim of creating a truly de-centralised VE. Towards that end the work of (Yang and Garcia-Molina 2003) correlates closely to the results obtained in this research, and super-peer nets offer the best hybrid solution between the speed of centralised services and the decentralised nature of a VE.

The research project essentially consisted of two complementary components. One was the method of achieving VE connectivity on a purely de-centralised topology, and the second was achieving data-model interoperability between the dis-similar enterprises. (Spyns et al 2002) of course stated the case for ontologies vs data-models due, as one factor, to the data-system independent nature of an ontology. The closest possible practical ontology construction method was mentioned by (Borst et al 1997) who first brought up the idea of creating 'reusable ontology blocks' that can be constructed at will on a project by project level (please note that ISO-STEP APs are constructed and approved in a centrally managed committee and not loosely by users according to their specific requirements.). Comparisons with the work of (Pouchard et al 2000) show that his work was on an abstraction level too high to be of immediate use to engineering enterprises, and the PSL language defined therein is limited to defining only the VEs business processes but not project specific information as is detailed in STEP APs.

Comparing this research with some research based around PDM technologies we find a basic disconnect between the ideals of de-centralisation and the inherently centralised nature of the PDM/PLM type solutions. (Kim et al 2001) for example implemented an open-standard data management architecture (based around STEP). However the implementation was carried out over a centralised stack, therefore limiting its use to internal company use or for limited inter-enterprise collaboration where the 'controlling' enterprise has the PLM system and its subordinate partners standardise around STEP. Comparing approaches for modelling the ontology of the system,

(Youchon et al 2001) used a UML schema on top of the highly detailed STEP PDM ontology. This research, of course, included the same STEP entities but without the problem of relying on different systems having an exotic SDAI or Express parser implementation which widens the utility of the OWL 'building block ontology' to a user base much wider than that of pure STEP based enterprises helping achieve ubiquity through implementing a more widely used ontology language for encoding the STEP schema.

In terms of aiding the engineers and project participants to reuse the engineering knowledge contained within the VE, other researchers such as (Rodgers and Huxor 1998) developed natural text-based knowledge-based systems. The designers in this instance took a completely different approach to KM by removing all notions of a rigid ontology to structure and define the structure and semantics of the knowledge and used some 'natural language processing' ostensibly suitable for creative projects. This natural language knowledge base of course has its advantages, but it would render any attempt at inter-enterprise collaboration and integration a near impossibility due to the absolutely unstructured form of its knowledge base, and the large amount of processing required to perform the queries necessary to make the system work effectively in a decentralised environment with people running hundreds of possible queries concurrently. (Regli et al 2000) proposed a hybrid solid based knowledge management / Constructive Solid Geometry based system to reuse existing models in new concepts with similar configurations. The approach taken by Regli is useful only for the conceptual stage of the design process and has the shortcoming of not translating directly to later stages of the design and manufacturing process. However the ontology developed in this instance gives engineers the flexibility of defining an incomplete design at the concept stage and then fleshing out the details at later stages. In addition the advantage of the approach of using a unified ontology is the ability to look at previous design concepts based on the functional specifications of previous projects. It can help eliminate 'reinventing the wheel' whilst giving users a simple interface to find dis-similar projects with similar 'functional requirements' as specified in the ontology. Comparing with the work of (Hsu et al 1998), it can be seen that whilst the combined geometry and case based approach to conceptual design creation and reuse has some similarities to the geometry

+ ontology approach used in this research. The main difference between the two approaches was the fact that the work by (Hsu et al 1998) was for a standalone concept development environment that does not work within the topology of a distributed VE, and does not show any method of 'sharing' knowledge of concept designs between members of a VE. Once again this illustrates the fact that often research work and prototype systems are constructed with 'single functionality' concentrated on the core research question of that team within a narrow scope, but this has limited the function of the work to single-use scenarios that are rare in the modern distributed economy and VE.

The various 'diagrammatic' tools for functional concept design including the works of (Sharpe 1995) and the commercial (Netviz 2005) suffer from the problem of not having a complete lifecycle solution to the engineering concept – design – prototyping – manufacturing – maintenance – decommissioning. This of course has one of the side effects of limiting the ability of users to 'drill down' from an old concept design down to see any manufacturing / testing / decommissioning issues with that concept and be able to take those issues into consideration at an earlier stage of the concept development process for their new concept. This is a key advantage of the new 'super ontology' approach that integrates all the knowledge within ontology from its functional specification down to its geometry and manufacturing information (AP-224).

Comparing this research with researchers in the field of Design for Manufacture such as (Mukherjee and Liu 1997), (Sanchez et al 1997), and (Chen et al 1998) shows that, for the purposes of conceptual design for manufacture, the goals, i.e. constraining conceptual designs within the capacity of manufacturing processes are the same. In theory this provides the VE with conceptual designs that are usable lower down the lifecycle for manufacturability analysis, which is a similar function to that provided by the super-ontology integrated from concept – manufacture (i.e. a concept can be created, and based on the functions within the concept, previous designs can be searched to find similar functionality, but with a complete design / manufacturing analysis already completed. Similarly works by (Chui and Wright 1999) help in the creation and processing of individual concept designs and their associated plans for manufacture, but

provide no mechanism for reusing existing designs and rather rely on creating designs from scratch every time. (Huang and Mak 1999) on the other hand produced a set of tools that aid the design process by simply constraining designs within a set of design for assembly / design for manufacturing guidelines as opposed to allowing the comparison of different concepts and the reuse of old concepts for new designs.

(Shyamsundar et al 2001) created a system that aims for a similar comprehensive overview of solving the problems of collaborative design in a distributed VE. The approach taken has a more comprehensive integrated geometry creation/edit tool, however it is a three-tier centralised application, that once again limits its utility within a pure VE setting. (Roy and Kodkani 2000) created a concept ranking system using a so-called gallery of different concepts and their usability for different functional requirements. This research does not have the same feature set up as a 'gallery' but it does functionally the same step through the concept browsing of different concept design projects in the query window. (Roy and Kodkani 2000) work on the other hand is a simple standalone concept selection tool with no downstream processes or knowledge management functions included.

Another set of researchers looked at the problem of mixing 'knowledge' entities with design geometry by integrating a knowledge based system or ontology into an existing modeller (as opposed to the approach of this research which integrated STEP models into the ontology) and remained CAD system neutral). The work of (Kavakli 2001) is a simple example that integrates a LISP based knowledge base into a CAD system with no multi-user or VE functionality. (Park 1999) integrated an agent based system for DfM guidelines into an ACIS modeller, with the limitation being the modification of this dynamic knowledge base by end users on an ongoing basis and an inability to distribute and share this knowledge base within a VE. (Brunetti et al 2000) approached the issue of knowledge-geometry integration by creating manufacturing features from which concepts are created. This is a very easy to use concept and allows even laymen to create new concepts and analyse their usability. However the system is limited by the ability to create feature primitives and by the lack of sharable knowledge for a VE. (Nam and Wright 2001) and (Case and Hounsell 2000) introduced solutions similar to

(Brunetti et al 2000) using features to define conceptual designs that are usable for manufacturing / assembly. In both instances the systems display the weakness of feature based designs in their limited set of 'features' that limit the potential for the creation of new products and designs without resorting to the creation of rather complex 'features' using a programming interface.

There are a number of new research projects that have built on the research carried out and published as part of this thesis. The following works directly reference the work of the author:

(Brandt et al 2005) at the University of Aachen in Germany, published works titled 'Ontology based Information Management in Design Processes'. The issues raised by this paper are the difficulty in managing the data in heterogeneous design environments using traditional tools such as PDM systems. The researchers quote, and accept the following assertions made in this research:

- ❧ PDM/PLM systems lack essential knowledge management capabilities.
- ❧ PDM/PLM systems are less suited to the conceptual design stage.

The researchers propose creating a process data warehouse using an ontology schema to define the potential ad-hoc and heterogeneous processes and knowledge that constitutes knowledge at the conceptual stage of the engineering design. The researchers have used a centralised model for the repository, and made use of OWL and Karlsruhe Ontology (KAON) (<http://kaon.semanticweb.org>) technologies for the repository.

The SEEDS project (Systems Engineering Estimation and Decision Support) of the AMRC (Aerospace Manufacturing Research Centre) (2005) <http://www.cems.uwe.ac.uk/amrc/seeds/semanticwebmodelling.htm> at the University of West of England works on providing decision support for engineering design and manufacture (primarily in the aerospace sector, as denoted by the centre's title). The researchers, from their own experience, have come to the conclusion that open-standard methods are the optimal means of creating the software and knowledge management infrastructure for engineering systems (due to easy of interoperability and future

development). The SEEDS researchers use these research findings on the topic of inter-enterprise collaboration to reinforce the argument for pursuing the creation of a semantic web ontology for collaboration (however, in the SEEDS project, the collaborating partners work via a web based centralised node rather than a decentralised peer-to-peer model). The SEEDS researchers also contend that the creation and sharing of open-source software and tools enhances the ability of enterprises to collaborate.

STEP

The subset of the STEP standard AP-224 has functioned well on the frame-based system Protégé. The method used to re-create the Step schema on the frame based system can be applied to the rest of the STEP standard to make it easier to access and manipulate with RDF parsers and intelligent agents over the web. The system enabled the creation and manipulation of the STEP models using the graphical interface, and the system was successfully integrated with the collaborative development ontology as described in chapter 5.

Ontology

This report gave an overview of the ontology developed for the initial stage of the product development process relying primarily, as a source on the experience and procedures of an automotive supply company and using real-life project information from two separate OEMs (Ford and Nissan). With these sources of information, a highly detailed ontology was constructed and successfully tested.

7.2 Conclusion

The objective of the research has been to find ways of improving knowledge management and collaboration between enterprises in engineering projects.

Creation of a novel 2-tier architecture for collaborative product development between dissimilar enterprises using a highly scalable hybrid super-peer network.

The three main contributions to knowledge made by the project were as follows:

1. A generic shareable super-ontology (as the top tier of the main architecture).
This generic super-ontology not only saved all the engineering knowledge generated in a project, but also maintained the explicit links between the evolutionary phases of the project down to a fine grained level (individual entities) that enabled the future reuse of design and production knowledge through the interactive generation of customer requirements and specifications. In addition, the ontology enabled the consolidation of manufacturing features, geometry and the product development knowledge at the conceptual stage to enable early design manufacturability analysis (however with the addition of external functions for manufacturability analysis). The principle of having building block shared super-ontology for the product knowledge that is created on a project-by-project basis (taking into account the uniqueness of each project) is quite new. Previous attempts on the subject have concentrated on creating a 'one size fits all' solution or else simply a data exchange standard so abstract that it was of no practical use for rapidly establishing collaborative projects between partners. The method laid out in this thesis is unique in that it suggests the creation of standardised classes and attributes for the knowledge base which cover a domain holistically by incorporating multiple international standards in use by industry (even from competing rather than complementary standards bodies). This aims to give enterprises in the engineering sector maximum flexibility in operating in multiple heterogeneous projects with different rules and standards (via a single interface and communication medium). This was demonstrated in Chapters 5 & 6 using an ArvinMeritor project with the Ford Motor Company and a number of VE participants (subcontractors), Gammastamp, Barton etc. The system was functional and allowed the capture of requirements in a standardised way, generation of specifications and from there the creation of new concept designs (see chapter 5). The test data was loaded into the system and tested again in a network between Cranfield University and Durham University. It was however not used operationally by the participating companies directly.
2. Creation of a two tier architecture with formalisation of format and medium as separate components to enable inter-enterprise collaboration in a heterogeneous

Virtual Enterprise. The bottom tier is simply an authentication and trust management layer, with the all important second tier holding the super-ontology described above. The combination of a common communication medium, as well as a common format for understanding and a common pool of reusable ontological entities for describing and entering knowledge, enables the rapid establishment of many-many inter-enterprise B2B links in real-time. A clearer understanding of the two-tier architecture would be to think of it as simply the formal combination of the de-centralised super-peer net and super ontology into a systematic view. The combination is more than the sum of its components as a de-centralised super peer net and the generic super ontology only make a VE possible when combined into one system. The 'format' within the system of course was the OWL language and JXTA protocols for the tier 2 and 1 respectively, whilst the "medium" for inter-enterprise collaboration was the topology and standard for the data connection between the participants in a VE (client-server / ASP and P2P were all tested for viability).

3. The creation of a de-centralised super-peer net for Virtual Enterprise collaboration that enabled both the egalitarian participation and as the mechanism to distribute the ontology, collaborate and share the knowledge base is a unique employment of an existing technique within the domain of engineering knowledge management and virtual enterprise set up and management. This was optimised for the two industrial examples (ArvinMeritor and Mabey&Jonhson) but was demonstrated only in a network between Cranfield University (with multiple separate instances) and Durham University.

Associated tasks and present status

- ⌘ The system as it is, is functional on a limited basis;
- ⌘ The STEP AP-224 standard has not been fully modelled due to the size of the standard; and
- ⌘ Benchmarking of the system using a truly distributed development team would be necessary to draw full conclusions as to the methodology's applicability to very large scale projects.

7.3 Suggestions for further research and follow on research by third parties.

A full project level benchmarking of the methodology is required to fully quantify the benefits suggested by the small case studies and applications developed. In addition integration of the system with a widely used solid modeller would enhance the usability of the tool for average engineers over their design lifecycle.

In addition, taking into account the speed of technical progress. The W3C has released the final Web Ontology Language (OWL) that supersedes the RDF format used in this project. Among its potential benefits would be:

- ❧ International acceptance of the new standard.
- ❧ Better support for inferences and a richer set of constraints.
- ❧ Maintaining the RDF format for easy migration.

For the medium to long-term, further development of the 2-tier inter-enterprise collaboration architecture would be beneficial for the overall development of the Virtual Enterprise research field.

8 References

alibre "alibre design online colaborative CAD tool." (2002): www.alibre.com

Androutsellis-Theotokis S, "A Survey of Peer-to-Peer File Sharing Technologies" Athens University of Economics and Business, (2002).

Antoniou. G, Hatcher P, Jan M and Noblet, D.A, "Performance Evaluation of JXTA Communication Layers (extended version)", 5th International Workshop on Global and Peer-to-Peer Computing, (2005).

Ashby. MF, Brchet. YJM, Cebon. D, and Salvo. L, "Selection strategies for materials and processes", Materials and Design, vol 25,1, p51-67, (2004).

Barnett W, Presley A, Johnson M, and Liles D, "An Architecture for the Virtual Enterprise". In the proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, San Antonio, TX, (1994).

Beckett, R, "Determining the anatomy of business systems for a virtual Enterprise", Computers in Industry, 51, 127-138, (2003).

Bhandarkar Mangesh P., Nagi Rakesh, "STEP-based feature extraction from STEP geometry for agile manufacturing", Computers in Industry, vol 41, 1 p3-24 (2000).

Borst P, Akkermans H. and Top J, "Engineering ontologies", International Journal of Human-Computer Studies, 46 , 2-3 , p365—406, Academic Press, (1997).

Brandt S.C, Jan M, Miatidis M, Theissen M, Jarke M and Marquardt W, "Ontology-based Information Management in Design Processes", (2005).

Breitman KK and do Prado Leite JCS, "Ontology as a requirements engineering product", Requirements Engineering Conference, 2003. Proceedings. 11th IEEE International , 309--319, (2003).

Browne J. and Zhang J, "Extended and virtual enterprises--similarities and differences", International Journal of Agile Management Systems, vol 1, 1, p30-36, Emerald Group Publishing Limited, (1999).

Brunetti G. and Golob B."- A feature-based approach towards an integrated product model including conceptual design information." Computer-Aided Design 32, p877-887, (2000).

Byrne, J. A., "The Virtual Corporation". Business Week. p98-103, (1993).

Cai, M. and Frank, M., "RDFPeers: a scalable distributed RDF repository based on a structured peer-to-peer network", Proceedings of the 13th conference on World Wide Web, p650--657, ACM Press New York, NY, USA, (2004).

Camarinha-Matos, L; Afsarmanesh, H, "Elements of a base VE infrastructure", Computers in Industry, 51, p139-163, (2003).

Camarinha-Matos, L. and Afsarmanesh, H. , The Virtual Enterprise Concept , Proceedings of the IFIP TC5 WG5. 3/PRODNET Working Conference on Infrastructures for Virtual Enterprises: Networking Industrial Enterprises, p3-14, Kluwer, BV Deventer, (1999).

Camarinha-Matos, L. and Afsarmanesh, H, Garita C and Lima C, "Towards an architecture for virtual enterprises", Journal of Intelligent Manufacturing, vol 9, 2, 189-199, Springer, (1998).

Case K and Hounsell M, "Feature modelling: a validation methodology and its evaluation." Journal of Materials Processing Technology, vol 107, p15-23, (2000).

Chen K.H, Chen S.J, Lin L and Changchien S.W, "An integrated graphical user interface (GUI) for concurrent engineering design of mechanical parts." Computer Integrated Manufacturing Systems, vol 11, p91-112, (1998).

Cheung W, Maropoulos P, Gao J. and Aziz H, "Knowledge-enriched product data management system to support aggregate process planning", Proceedings of the First International Conference on Manufacturing Research, Advances in Manufacturing Technology, vol 17, p253-258, (2003).

Chin KS and Wong TN, "Integrated product concepts development and evaluation." International Journal of Computer Integrated Manufacturing, vol 12, p179-190, (1999).

Chui WH and Wright K, "A WWW computer integrated manufacturing environment for rapid prototyping and education", International Journal of Computer Integrated Manufacturing, (1999).

CoCreate "CoCreate onespace collaborative product design.", (2002).

Cohen E, "Replication strategies in unstructured peer-to-peer networks", Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications, p177-190, ACM Press New York, NY, USA, (2002).

www.compiere.org Compiere "Open Source Enterprise Resource Planning", (2003).

www.covisint.com Covisint "Portal for automotive supply chains", (2003).

Cutkosky M.R, Engelmores R.S, Fikes R.E, Genesereth M.R, Gruber T.R, Mark W.S, Tenenbaum J.M. and Weber J.C, "PACT: An Experiment in Integrating Concurrent Engineering Systems" IEEE Computer, vol 26, 1, p28--37, (1993).

www.cyco.com Automanager workflow software tools

Dahmus, Gonzalez-Zugasti and Otto "- Modular product architecture." Design Studies 22, p409-424, (2001).

Damiani E, Paraboschi S, Samarati P and Violante F, "A reputation-based approach for choosing reliable resources in peer-to-peer networks", Proceedings of the 9th ACM conference on Computer and communications security, p207--216, ACM Press New York, NY, USA, (2002).

Daswani N, Garcia-Molina H and Yang B, "Open problems in data-sharing peer-to-peer systems", Proceedings of the 9th International Conference on Database Theory, p1-15, Springer, (2003).

Davies N.J, Fensel D and Van Harmelen F, "Towards the Semantic Web: Ontology-Driven Knowledge Management", John Wiley and Sons, (2003).

www.delmia.com delmia "delmia manufacturing/assembly software solutions.", (2002).

Engineering Geometry Systems "FeatureCAM feature based design with feature recognition and CAPP", (2001).

Fenves, Rivard and Gomez "SEED-Config: a tool for conceptual structural design in a collaborative building design environment." Artificial Intelligence in Engineering, 14, p233-247, (2000).

Fuyama, Law and Krawinkler "An interactive computer assisted system for conceptual structural design of steel buildings." Computers & Structures, 63, p647-662, (1997).

www.geometricsoftware.com Geometric Software Solutions.

Gong L, "Peer-to-Peer Networks in Action", Internet Computing, IEEE , vol 6, 1, p37—39, (2002).

Gribble S, Halevy A, Ives Z, Rodrig M and Suciu D, "What can databases do for peer-to-peer", WebDB Workshop on Databases and the Web, (2001).

Gruber T.R, "Toward principles for the design of ontologies used for knowledge sharing?", International Journal of Human Computer Studies, vol 43, 5-6, p907--928, (1995)

Hardwick M, Spooner D.L, Rando T and Morris KC, "Sharing manufacturing information in virtual enterprises", Communications of the ACM, vol 39, 2, p46--54, ACM Press New York, NY, USA, (1996).

Harren M, Hellerstein J.M, Huebsch R, Loo B.T, Shenker S and Stoica I, "Complex Queries in DHT-based Peer-to-Peer Networks", 1st International Workshop on Peer-to-Peer Systems (IPTPS'02), Springer, (2002).

Hsu and Liu, "Conceptual design: issues and challenges.", Computer-Aided Design 32, p849-850, (2000).

Hsu, Fuh and Zhang, "Synthesis of design concepts from a design for assembly perspective.", Computer Integrated Manufacturing Systems 11, p1-13, (1998).

Huang and Mak, "Design for manufacture and assembly on the Internet.", Computers in Industry 38, p17-30, (1999).

Haenisch J, "Express data manager for intelligent design.", www.epm.jotne.com, (2001).

www.iai-international.org Industry Foundation Classes

www.ieee.org Institute of Electrical and Electronics Engineers.

Impactxoft, "impactxoft IX SPeeD.", <http://www.impactxoft.com/>, (2002).

Inouye T, "Open source automatic feature recognition tool for configuration controlled designs", Fujitsu Kyushu System Engineering Ltd, <http://www.fqs.co.jp>, (2001).

www.iso.org, "STEP Application Protocols ISO 10303" International Standardisation Organisation.

Jagdev H. S, Thoben K-D, "Anatomy of enterprise collaborations, Production Planning & Control", Volume 12, Number 5 / July 1, p437-451, (2001).

JBOSS, "Java 2 Enterprise Edition Application Server", www.jboss.org, (2003).

Jeon H, Petrie C and Cutkosky MR, "JATLite: a Java agent infrastructure with message routing", Internet Computing, IEEE, vol 4, 2, p87-96, (2000).

Petersen. J, "A public domain collaborative workspace application", www.jpetereson.com (2001).

Jovanovic, M.A., Annexstein, F.S. and Berman, K.A. , "Scalability Issues in Large Peer-to-Peer Networks-A Case Study of Gnutella", University of Cincinnati, (2001).

www.jxta.org "Juxtapose Open Peer to Peer standard", Sun Microsystems.

<http://kaon.semanticweb.org> Karlsruhe Ontology.

Kavakli M, " NoDes: kNOwledge-based modeling for detailed DESign process -- from analysis to implementation.", Automation in Construction 10, p399-416, (2001).

Kim Y, Kang S-H, Lee S-H, Yoo S-B, "A distributed, open, intelligent product data management system.", International Journal of Computer Integrated Manufacturing 14, p224-235, (2001).

Krishnan R, Smith M.D and Telang R, "The economics of peer-to-peer networks", Journal of Information Technology Theory and Applications, vol 5, 3, p31-44, (2003).

KTI International, Knowledge Technologies "kti ICAD knowledge based design tool.", (2002).

LDAP, "Lightweight Directory Access Protocol"
<http://www.gracion.com/server/whatldap.html> (2001).

Lihui Wang, "Collaborative conceptual design- State of the art and future trends", Computer Aided Design, p981-996, (2002).

LKsoft "Java based SDAI and step schema compiler.", (2002).

LSC Group "White Paper - The UK Navy Mechanical RAMP Project.", (1999).

Lu, J. and Callan, J, "Content-based retrieval in hybrid peer-to-peer networks", Proceedings of the twelfth international conference on Information and knowledge management, p199--206, ACM Press New York, NY, USA, (2003).

Lv Q, Cao P, Cohen E, Li K and Shenker S, "Search and replication in unstructured peer-to-peer networks", Proceedings of the 16th international conference on Supercomputing, p84-95, ACM Press New York, NY, USA, (2002).

Maropoulos P, Yao, Bradley and Paramor "- An integrated design and planning environment for welding; Part 1: Product modelling." Journal of Materials Processing Technology 107, p3-8, (2000).

Maropoulos P, Yao, Bradley and Paramor "- An integrated design and planning environment for welding; Part 2: Process planning." Journal of Materials Processing Technology 107, p9-14, (2000).

Maropoulos P, Bradley and Yao "- CAPABLE: an aggregate process planning system for integrated product development." Journal of Materials Processing Technology 76, p16-22, (1998).

Mishra P, A Varshney and A Kaufman, "collabcad java client server collaborative cad/cam tool." Proceedings TeamCAD: GVU/NIST Workshop on Collaborative CAD, (1997).

Mukherjee A and Liu CR, "Conceptual design, manufacturability evaluation and preliminary process planning using function-form relationships in stamped metal parts.", Robotics and Computer-Integrated Manufacturing 13, p253-270, (1997).

Nam and Wright "The development and evaluation of Syco3D: a real-time collaborative 3D CAD system.", Design Studies 22, p557-582, (2001).

Netviz, "Data Driven Visualisation system", www.netviz.com, (2005)

www.oasys-software.com, Oasys Systems.

OPENCASCADE "Open Source 3D solid modeller Engine" www.opencascade.org, (2003).

Oram, A, "Peer-to-Peer: Harnessing the Power of Disruptive Technologies", O'Reilly & Associates, Inc, (2001).

www.openssl.org, "Open source implementation of the Secure Sockets Layer", The Open SSL Project, (2004).

Parametric Technology Corporation, "dynamic design link for customer focused collaborative design over Windchill." www.ptc.com, (2001).

Park H G, "Enhancing manufacturing product development through learning agent system over internet.", Computers & Industrial Engineering 37, p117-120, (1999).

PD Tec, "ECCO express based IDE.", (2002).

www.php.org "Perl Hypertext Processor" (PHP).

Pouchard L, Ivezic N and Schlenoff C, "Ontology Engineering for Distributed Collaboration in Manufacturing", Proceedings of the AIS2000 Conference, March, (2000).

Protégé ontology Editor, Stanford University, <http://protege.stanford.edu>, (2005).

Regli W.C, and V.A.Cicirello, "Managing digital libraries for computer-aided design", International Journal of Computer-Aided Design, 32, p119-132, (2000).

Reid R.L, Rogers KJ, Johnson M.E, and Liles D.H, "Engineering the Virtual Enterprise", Automation & Robotics Research Institute, (1997).

Rodgers and Huxor, "The role of artificial intelligence as 'text' within design CADET.", Design Studies 19, p143-160, (1998).

Roy U and Kodkani SS, "Collaborative product conceptualization tool using web technology.", Computers in Industry 41, p195-209, (2000).

Sanchez JM, Priest JW and Soto R, "Intelligent Reasoning Assistant for Incorporating Manufacturability Issues into the Design Process.", Expert Systems with Applications 12, p81-88, (1997).

www.sap.com, SAP AG, Germany.

SEEDS, "Systems Engineering Estimation and Decision Support", Part of the AMRC (Aerospace Manufacturing Research Centre), University of West of England, (2005).

Sharma R, and Gao J.X, "A Progressive design and manufacturing evaluation system incorporating STEP AP224.", Computers in Industry, (2004).

Sharpe, John E.E, "Computer tools for integrated conceptual design.", Design Studies 16, p471-488, (1995).

Shehab E.M and Abdalla H.S, "Manufacturing cost modelling for concurrent product development.", Robotics and Computer-Integrated Manufacturing 17, p341-353, (2001).

Shen W, Maturana F, and Norrie D.H, "MetaMorph II: an agent-based architecture for distributed intelligent design and manufacturing", Journal of Intelligent Manufacturing , vol 11, 3, p237-251, Springer, (2000).

Shyamsundar N and Gadh R, "Internet-based collaborative product design with assembly features and virtual design spaces.", Computer-Aided Design 33, p637-651, (2001).

Song E.Y and Feng S.C "Information Modelling of Conceptual Design Integrated with Process Planning.", International Mechanical Engineering Congress and Exposition Symposia on Design For Manufacturability, (2000).

www.sourceforge.net Sourceforge Open source software development repository.

Spyns P, Meersman R, and Jarrar M, "Data modelling versus ontology engineering", ACM SIGMOD Record, vol 31, 4, p12-17, ACM Press New York, NY, USA, (2002).

Staab S and Maedche A, "Ontology engineering beyond the modeling of concepts and relations", Proceedings of the ECAI'2000 Workshop on Application of Ontologies and Problem-Solving Methods, (2000).

Sun Microsystems, "JuxtaPose JXTA peer to peer framework", www.jxta.org, (2003).

Sun Microsystems, "Java2 Standard Edition", <http://java.sun.com>.

Technosoft, "AML and TIE for collaborative conceptual design" www.technosoft.com (2003).

Tomcat servlet engine, Apache Foundation, www.apache.org, (2005).

The Ultimate Team Organisation Software, "TUTOS", www.tutos.org (2003).

Trienekens J.H, "Views on inter-enterprise relationships", Production Planning & Control, vol 12, 5, p466-477, Taylor & Francis, (2001).

Ulieru M, Norrie D.E, Kremer R and Shen W, "A multi-resolution collaborative architecture for web-centric global manufacturing", Information Sciences, Volume 127, Issues 1-2 , August, Pages 3-21, (2000).

www.wfmc.org Workflow Management Coalition.

Yang B and Garcia-Molina H, "Improving search in peer-to-peer networks", Proceedings. 22nd International Conference on Distributed Computing Systems, p5-14, (2002).

Yang B and Garcia-Molina H, "Designing a Super-Peer Network", Proceedings of the 19th International Conference on Data Engineering (ICDE'03), vol 10, 63, (2003).

Youchon, Soon-hung and Hyowon, "Mapping product structures between CAD and PDM systems using UML.", COMPUTER AIDED DESIGN, (2001).

Varun, Ratnakar and Yolanda G, "A comparison of semantic markup languages", Proc. of the 15th Int. FLAIRS Conference, (2002).

Vasara P, Krebs V, Peuhkuri L and Eloranta E, "Arachne--adaptive network strategy in a business environment", Computers in Industry, 50, p127-140, (2003).

www.w3c.org World Wide Web Coalition.

Wang L, Shen W, Xie H, Neelamkavil J and Pardasani A, "Collaborative conceptual design -state of the art and future trends", Computer-Aided Design, vol 34, 13, p981-996, Elsevier, (2002).

Zha X.F and Du H, "A PDES/STEP-based model and system for concurrent integrated design and assembly planning", Computer-Aided Design, (2001)

Zhang Y, Zhang C and Wang H.P, "An Internet based STEP data exchange framework for virtual enterprises", Computers in Industry, vol 41, 1, p51--63, Elsevier Science Publishers BV Amsterdam, The Netherlands, (2000).

Zhou L and Nagi R, "Design of distributed information systems for agile manufacturing virtual enterprises using CORBA and STEP standards", Journal of manufacturing systems, vol 21, 1, p14-31, Elsevier, (2002).

❖ Appendix (A) System deployment

This chapter details and illustrates the deployment of a test run application from installation to data entry and collaboration. It is a simple use-case scenario based on hypothetical data.

Section one deals with the process of software installation and configuration, section two details the entry of customer requirements data into the system, section three shows how legacy knowledge aids the process of generating a design specification and concept designs from the requirements list, section four details the collaboration between parties and section five gives a brief overview of the possible generation of schedules and process plans.

▪ Software installation and setup

The software needed by the user, and the sequences for installation are as follows.

Java 1.5 runtime environment or developer kit that can be downloaded for free from <http://java.sun.com> . The software installs automatically by double clicking on the downloaded install.exe file.

Next download the MySQL database and JDBC (ConnectorJ) driver from <http://www.mysql.com/> MySQL 4.1 should suffice. Once again this will install itself automatically. Please keep the port setting at 3306, the hostname localhost and whatever username and password you defined on startup.

Use the MySQL control panel to create one blank database called protégé and create a user protégé with a password of your choice to manage the database.

Load the JDBC driver JAR file under the java path (under lib).

Next download the latest protégé ontology editor from <http://protege.stanford.edu> run the installation and choose to install all into the default installation directory.

Download the GENERICONTOLOGY zip file. Extract the contents into a convenient folder.

Run the Protégé system from the desktop icon or start menu.

In the splash screen, choose to create new project. Select protégé database and tick on create from existing sources. Click next. Fill in the following details in the form:

JDBC Driver class Name: com.mysql.jdbc.Driver

JDBC Driver URL: jdbc:mysql://localhost:3306/protege

Table: protege

Username: protege

Password: yourpassword

Now select the ontology that was extracted from the generic ontology.zip file. Once this is completed you have a knowledge base with a database backend running.

Next install the peer to peer client and connect it to the knowledge base via the JDBC connector with the same settings as for the protégé system. Now the backend knowledge base is connected to the peer to peer network.

Open the configuration window on starting up the peer to peer system. Enter the settings for any rendezvous peers (none for this test), relays or proxies (if you use a network with proxy then enter this setting). Select whether to use TCP or http transport. Open the profile.xml file in the peer net folder and edit your user settings and any trusted members.

The final step is to install a geometric modeller to view and create the STEP models. In this case the Pro/desktop modeller can be downloaded from <http://www.ptc.com/> the installation defaults can be used.

Now the system is completed with a backend database, ontology, knowledge management user interface, peer to peer connection and a geometric modeller. The next

step is to begin using the system. But before continuing, its best to first restart the computer and make sure that all the services are running correctly (MySQL service).

▪ Customer requirement entry

Open the protégé knowledge base. Select the first project on the splash screen (it was the last entry loaded from the installation phase, and so will be the first item in the list). Go to the instances tab, and on the left pane select the requirements superclass. There are three subclasses under it, requirements, req_type and verification_methods. The requirements subclass is where the requirements are stored, the other two classes contain related objects on classifying the requirements (req_types) and the verification methods used to verify each requirement (verification_methods), Figure 46 illustrates this clearly.

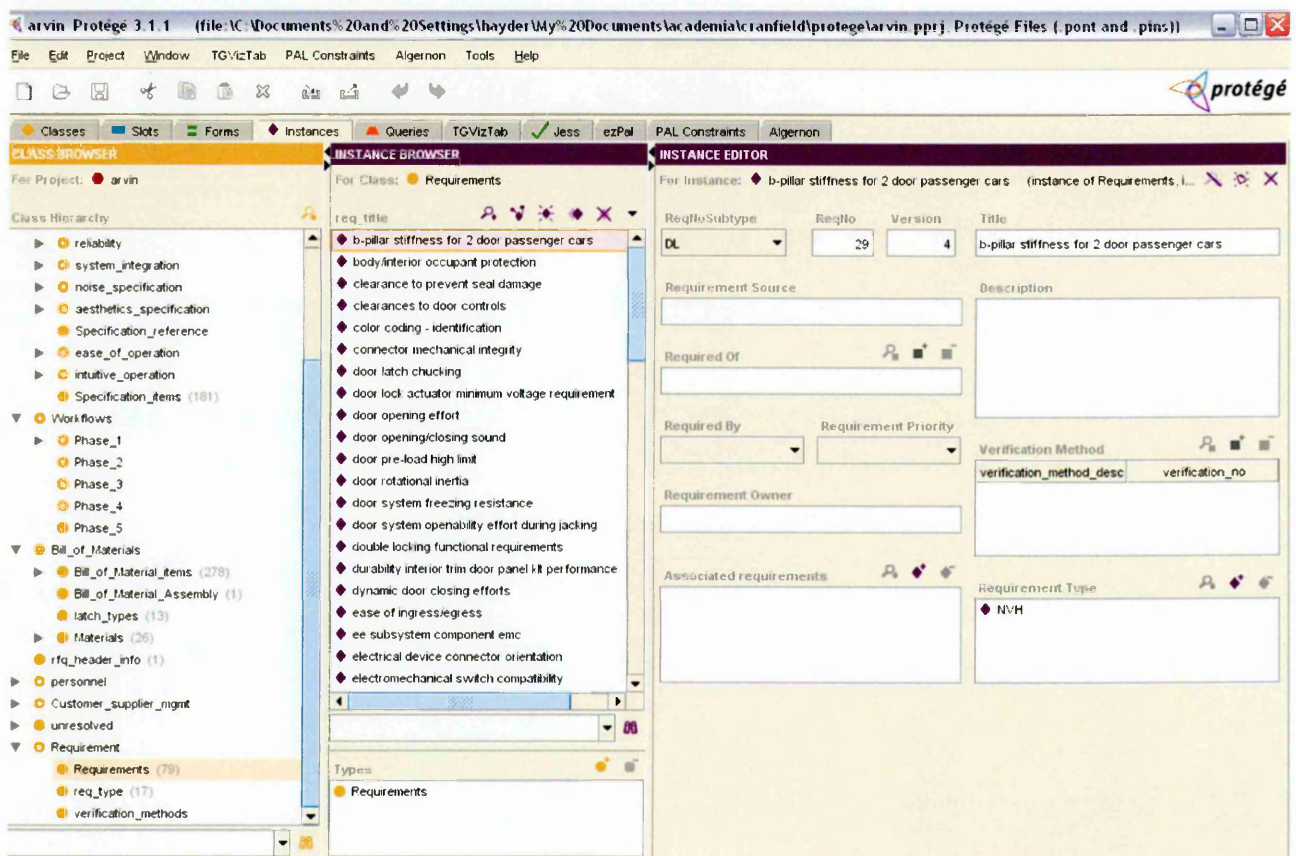


Figure 46 Requirements entry

The user interface is split into three panes. The left pane shows the program's structure (bill of material, Human Resources, Customer and Supplier Management, workflow details and specification classes can be seen in Figure 46). The middle pane shows a list



of existing objects for that function, and the right pane shows the form for viewing and modifying the data. A new object will be added by pressing on the icon in the middle pane. This brings up the form Figure 47.

Figure 47 requirements entry form

The following items must be filled in, firstly the requirement type (this is a code to classify the requirement into categories), and is selected from instances of the req_type class. Manually enter a requirement number, version for the object and a title. The source of requirement (who made the requirement) and the required_of statement where the requirement is assigned to a person (or a role) for responsibility. The required_by list allows the classification of whether the requirement fulfils customer, corporation, regulation or design needs. Requirement priority enables the correct prioritisation of the requirement whether it's a specification or standard requirement. The requirement owner in this instance is the person who manages the requirement process, and this item only needs to be entered if the person entering the knowledge is logged in under a different name. The description is where the actual requirement is stated. It is a free text area thus not restricting the requirement entry by limited vocabulary. Finally any dependent or associated requirements can be selected here (creating a chain of dependencies) and the requirement type is classified for development purposes into manufacturing, assembly, durability and a range of other classifications.

As can be noted, requirements are always new, it is not possible to base a new requirement on a requirement existing in the knowledge base as this will skew the specifications generated and concept design ideas in the system.

▪ Concept Design generation

Login to the protégé system (if not logged in yet). Under the classes and instances tab, on the left tree menu select 'AP224' for a full feature set or using the user_interface superclass to access the structured forms, see Figure 48.

The figure displays two side-by-side screenshots of the AP224 Clips user interface. The left window, titled "erererere (instance of user_interface, internal name is ap224_clips_00381)", contains a form with the following sections: "Summary Information" (This slot currently has 1 value.), "Description" (dsffewfrew), "Name" (ererere), "Chosen Method" (a table with columns consequence, name, description, purpose), and "Id" (dfgdsg). The right window, titled "Ap224 Clips 00355", contains a form with the following sections: "Summary Information" (This slot currently has 1 value.), "Consequence", "Description", "Name" (rewtgretgre), and "Purpose".

consequence	name	description	purpose
efefqrwer	erew	rerwerwere...erew	

Figure 48 user_interface for structured AP224 creation

The forms under user_interface simply collate the AP-224 classes into combined forms to ease of access. The actual knowledge entered is stored under the AP224 superclass.

Alternatively, the standard AP224 classes allow the entry of the data directly by its class name. In Figure 49 the action entity form is invoked. As can be seen, the system has dependencies, and in this instance a method must be chosen for the action object to be complete. The system gives the user a nested view to enable him to either choose from existing methods or create a new method for the action.

The screenshot shows a window titled 'ap224_clips_00380 (instance of act...'. The form is divided into several sections:

- Description:** A large text input field.
- Name:** A text input field.
- Chosen Method:** A section with a table and four icons (person, star, square, square) to its right.

consequence	name	description	purpose
- Id:** A section with a label 'Id' and four icons to its right. Below it is a table with one header 'identifier' and one empty row.

identifier

Figure 49 Action entity form

In this instance, we decide to click on the small rectangle with a + beside it to select an existing method. A new form pops up (Figure 50) with a list of action_method object that have already been created. If the list of objects is long, it is also possible to search for the required action_method using the search bar at the bottom.

This small example illustrated the explicitly interlinked nature of the system that minimises the chances of incorrect or incomplete data to be entered into the knowledge base. It also illustrates the ability to actively reuse existing knowledge in the system in different projects and thus constantly building up a network of inter-related objects with multiple relationships and dependencies on multiple objects.

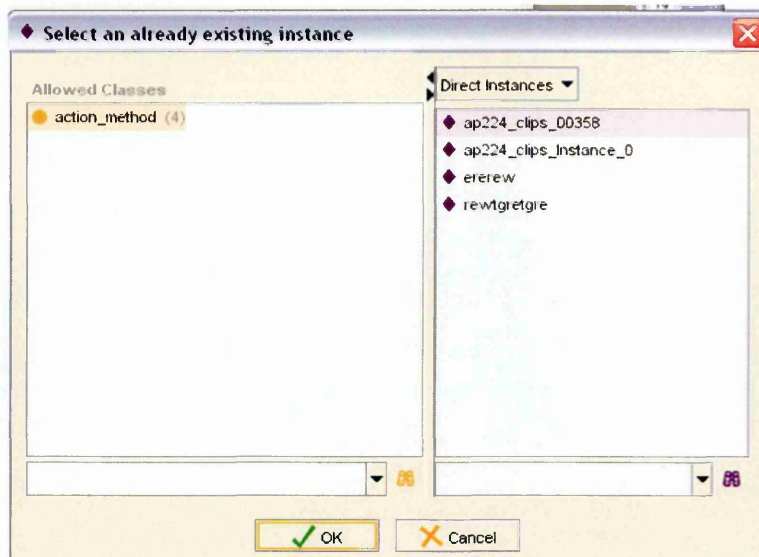


Figure 50 selecting exiting Action_Method to be inserted into action object

This is the building block of the knowledge base. It enables the knowledge worker to implicitly construct and improve the knowledge base (and structure it) without having to explicitly think about or create relations between objects.

▪ Collaboration

The basic settings before collaboration is possible are simple common sense prerequisites: Make sure that the computer is connected to the network, there is at least one other collaborator on the internal or external network with a correctly configured p2p system and who is also logged in and online. Whilst logged in to the protégé system, start the p2p application.

At the prompt type in the login name and password assigned from the installation stage, and list of connections will be visible on the left tab. Add a contact to the list of contacts on both peers (by typing in the peer-name or the unique identifier UID of the other peer). This establishes the connection between the peers. Now it becomes possible for each peer to interact together and use each others knowledge bases collaboratively, see Figure 7. When logging into the protégé system, click on open project, server, and type in the name of the machine to log into, see Figure 51.

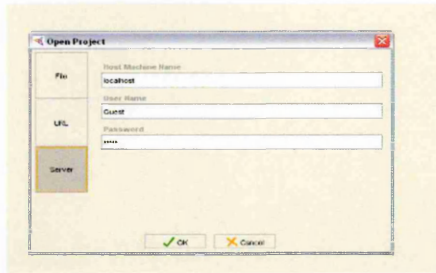


Figure 51 logging into remote knowledge base

The interaction with the knowledge base then happens directly via the protégé system as shown in the examples above. Please note that it is possible to connect to several instances of remote knowledge base concurrently and also assign a super peer to manage a knowledge base centrally. The instructions are identical regardless of the topology used in a particular VE.

▪ **Downstream processes**

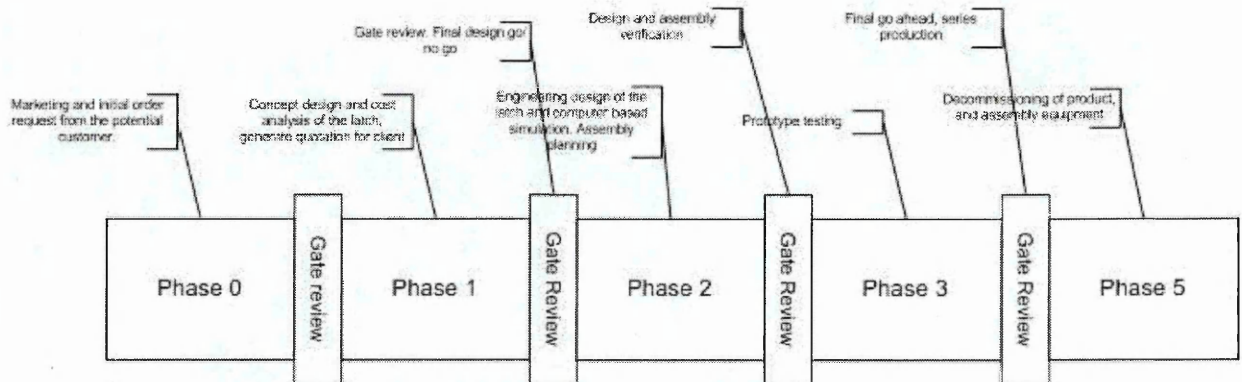
Once a concept has been agreed upon, then there are two ways of proceeding onto phase 2 of the project. The first method is to attempt to reuse existing components and subassemblies from existing designs that have similar design specifications, and the second method is to generate new assemblies and components using a STEP modeller. Any STEP modeller can be used, and the assembly and components checked into the protégé system either in binary format (as blobs) or maintaining a link to the file system location of the STEP models (this process is faster).

▪ **Summary**

This chapter gave a brief overview of the use-case scenario for deploying and using the collaborative knowledge management framework. Although the system is not as functional or tightly integrated as a full blown commercial solution, it illustrated the potential for using similar systems in real-life collaboration between enterprises.

❖ Appendix (B) Test workflows (ArvinMeritor Data)

ArvinMeritor: Lifecycle



Notes:

Phase 4 of the lifecycle is simply the series production of latches for the customer, in line with agreed protocol for delivery schedules and quality control.

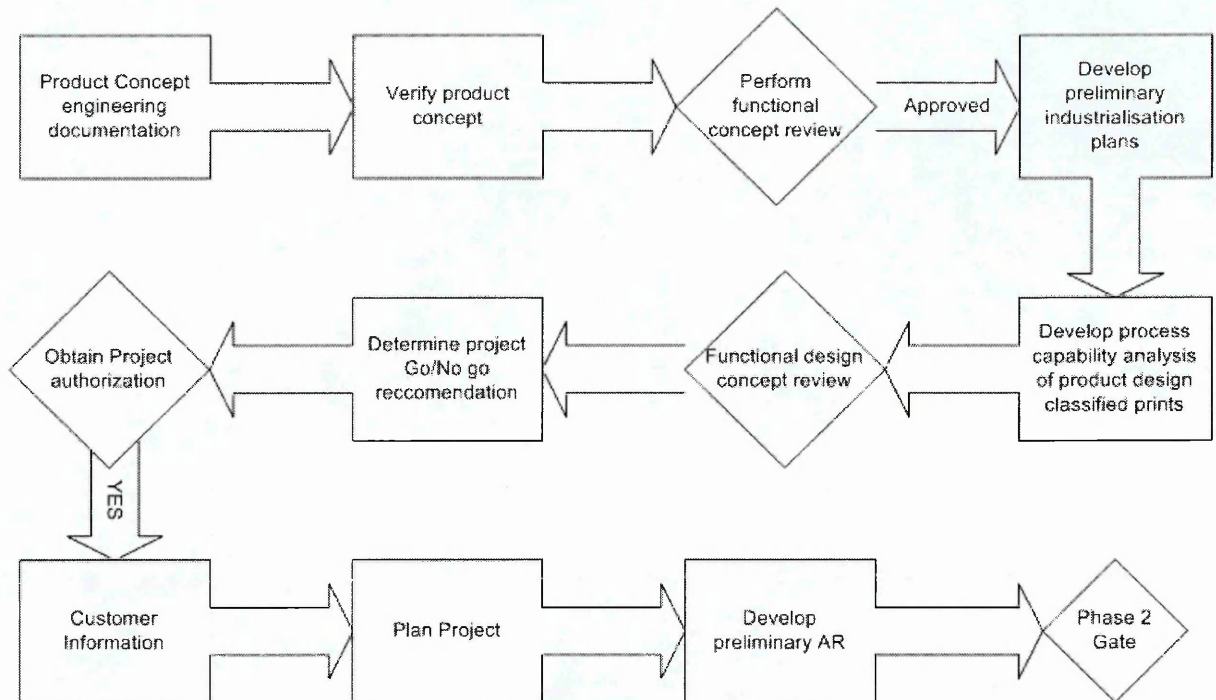
The workflow diagrams for Phase 0 are not included as this phase consists primarily of ad-hoc activities.

The lifecycle and workflow diagrams included here adhere to the standards set down by the Workflow Management Coalition. <http://www.wfmc.org/>

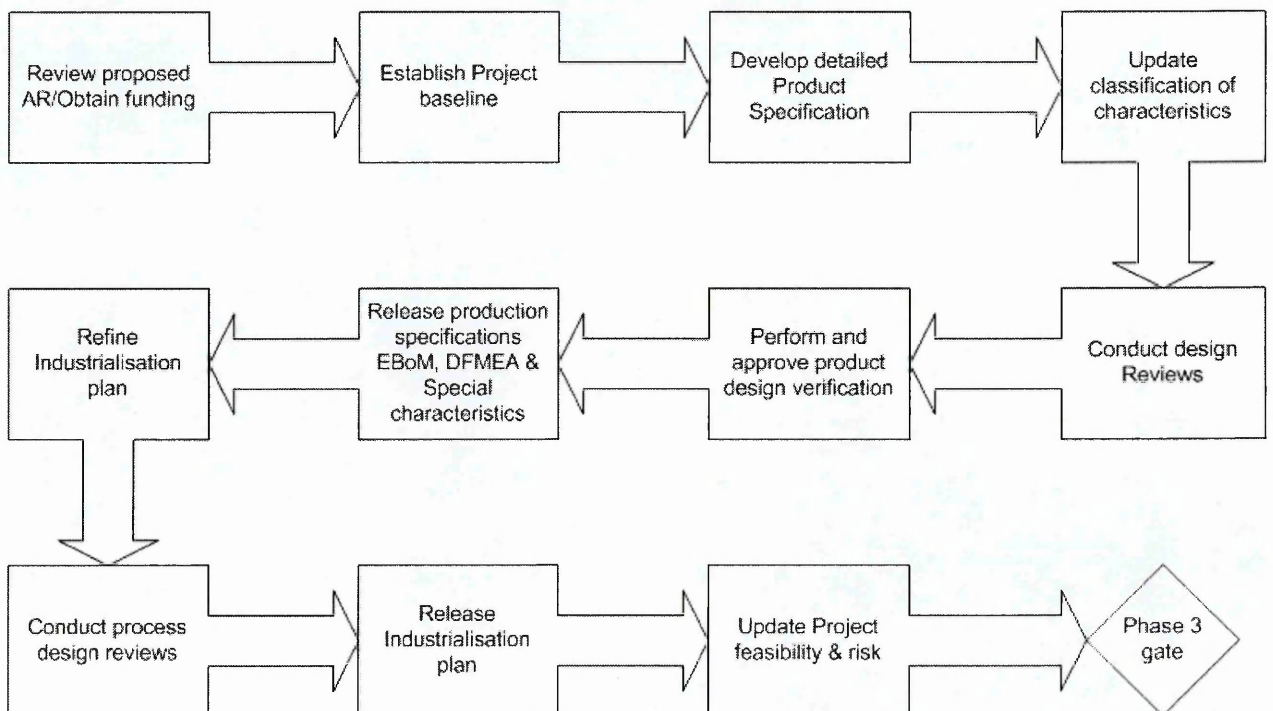
The diagrams are structured in a three layer format:

- Lifecycle
 - Intermediate review
 - individual tasks

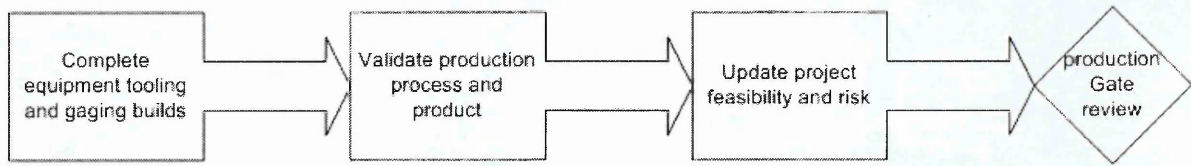
ArvinMeritor: Phase 1



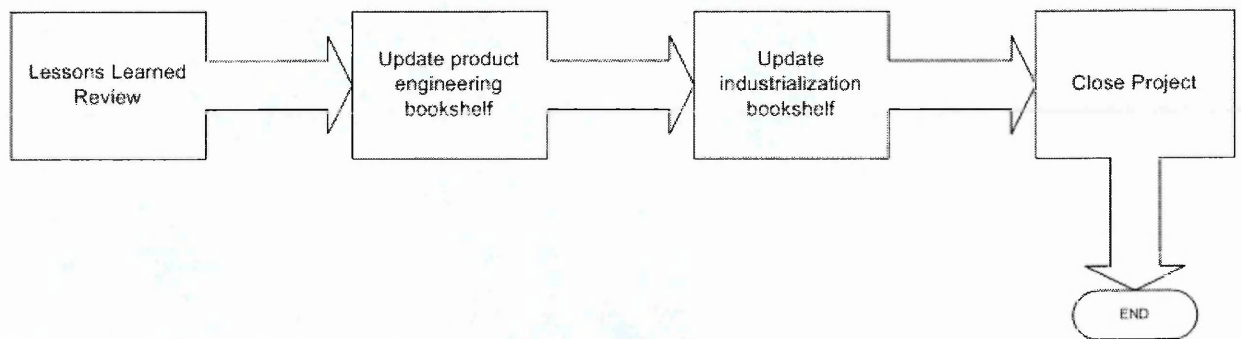
ArvinMeritor: Phase 2



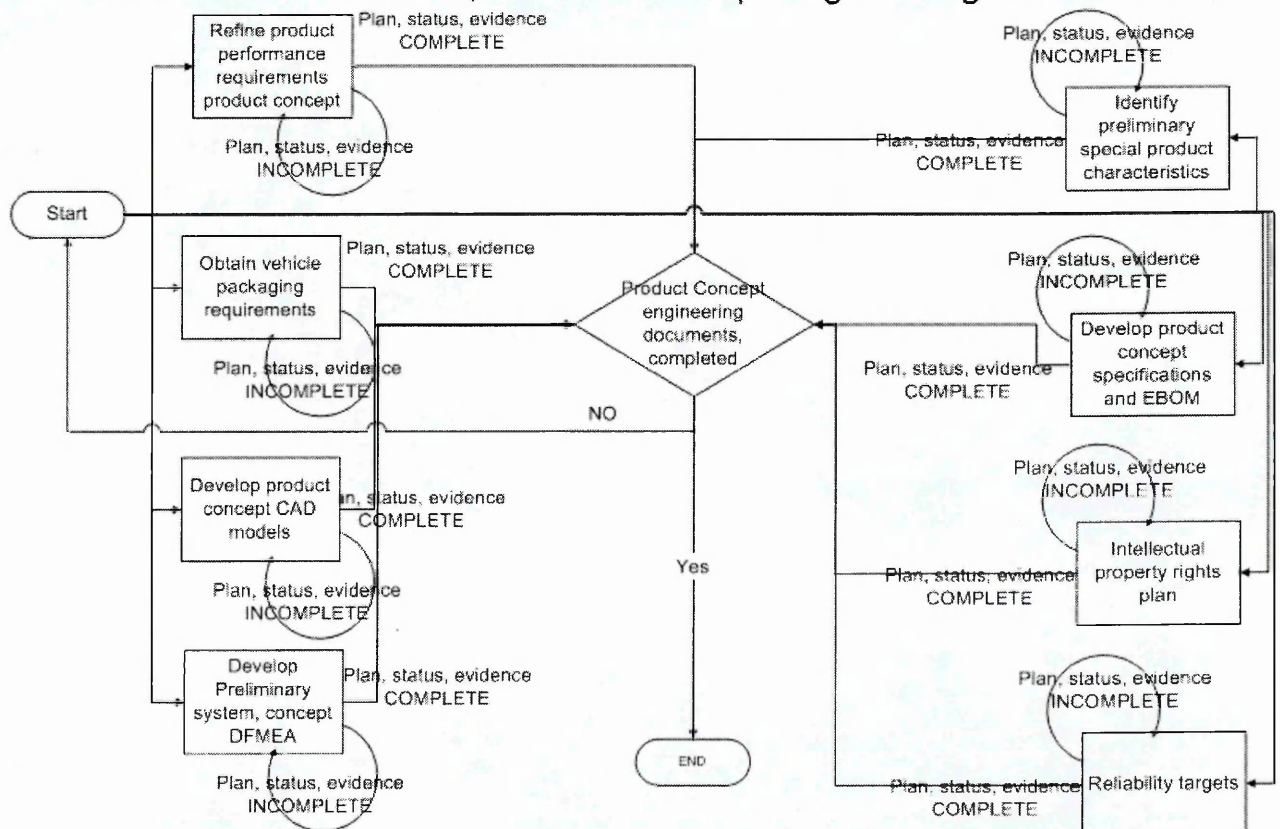
ArvinMeritor: Phase 3



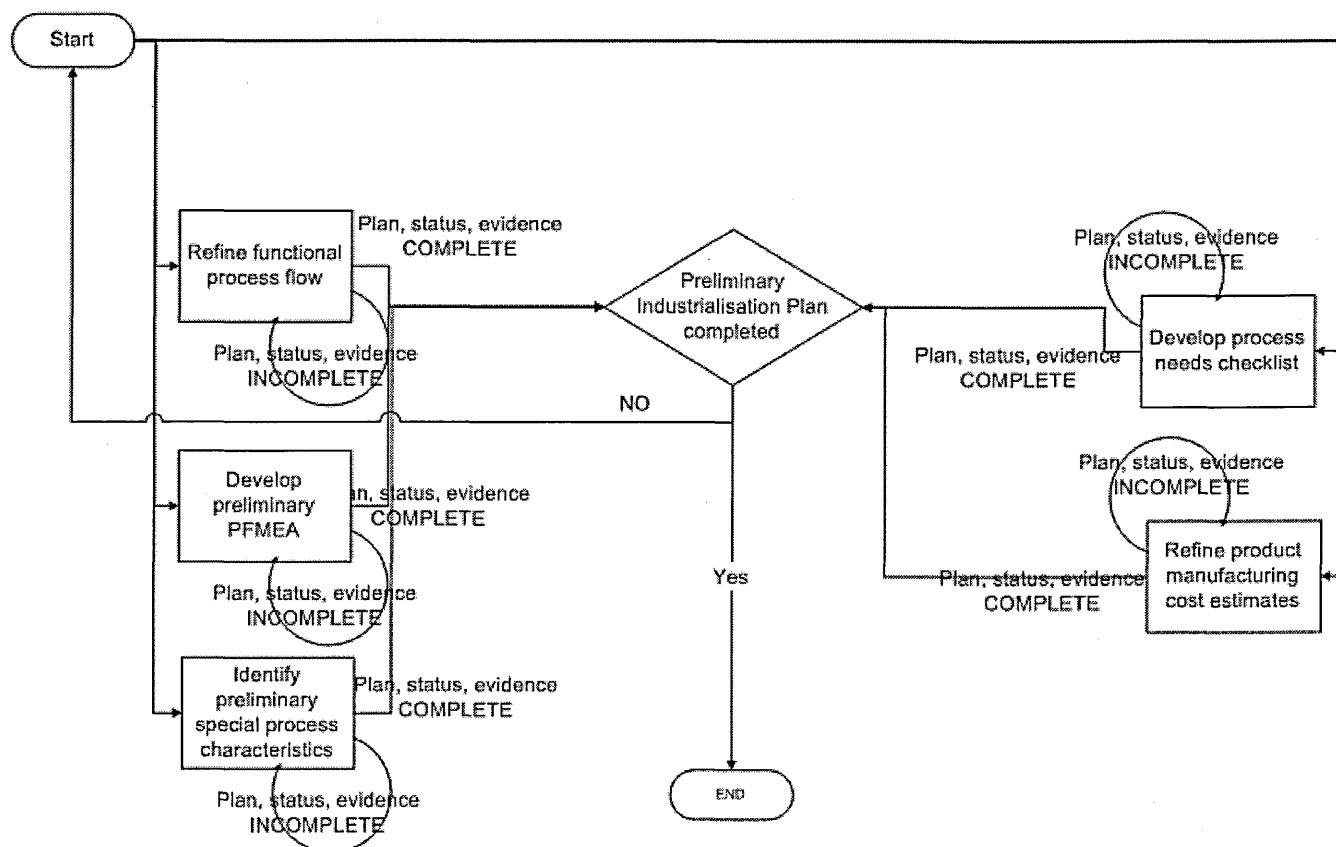
ArvinMeritor: Phase 5



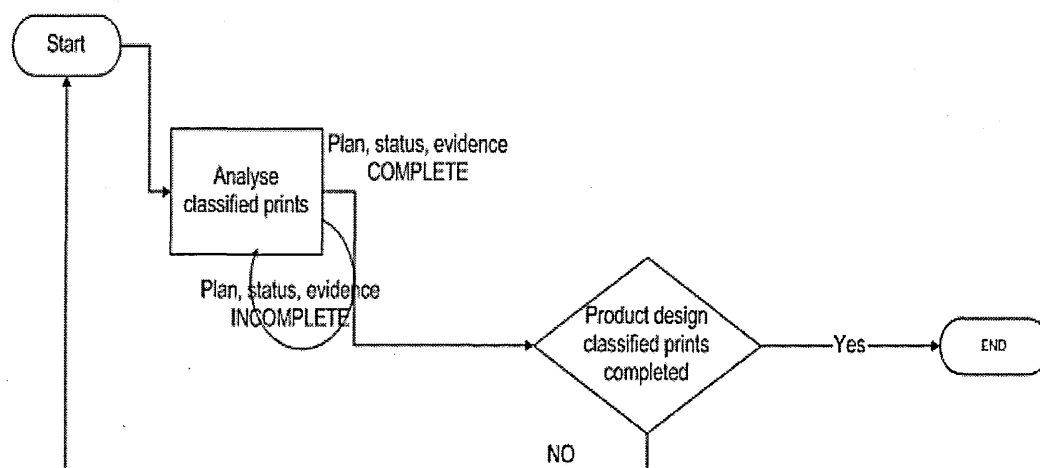
ArvinMeritor: Phase 1, Product concept engineering documentation



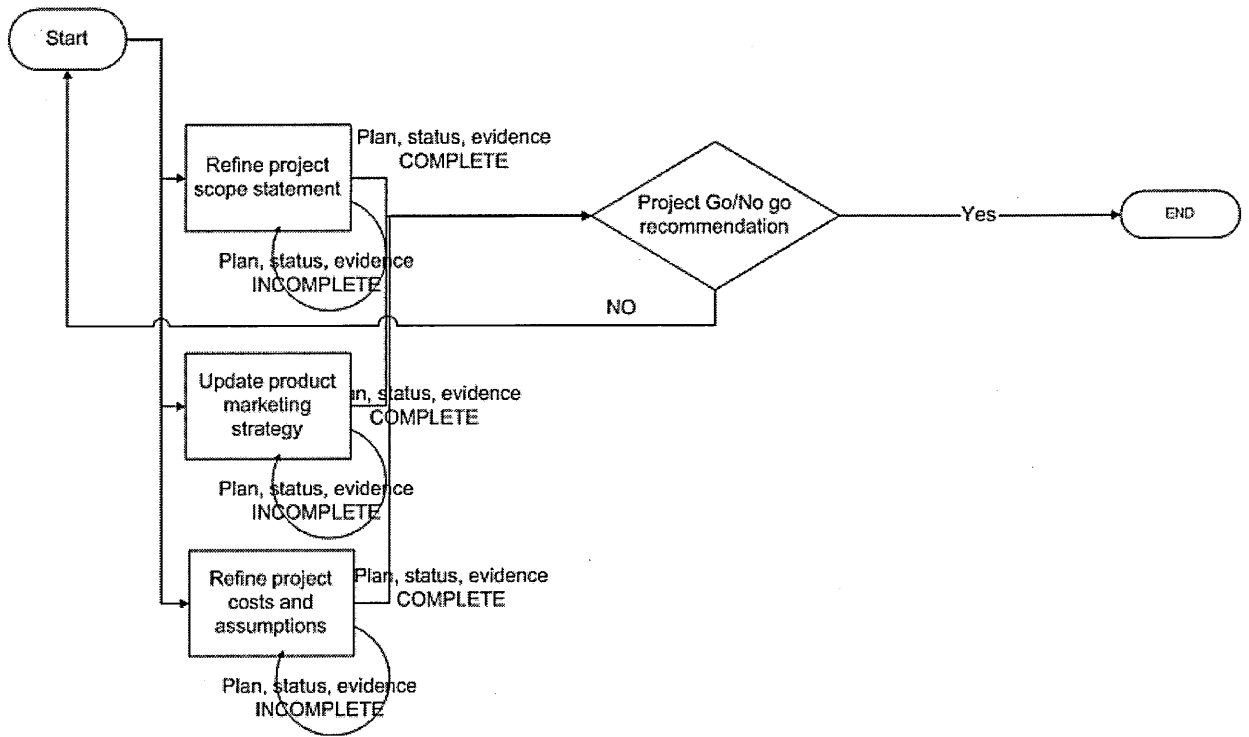
ArvinMeritor: Phase 1, Preliminary Industrialisation Plan



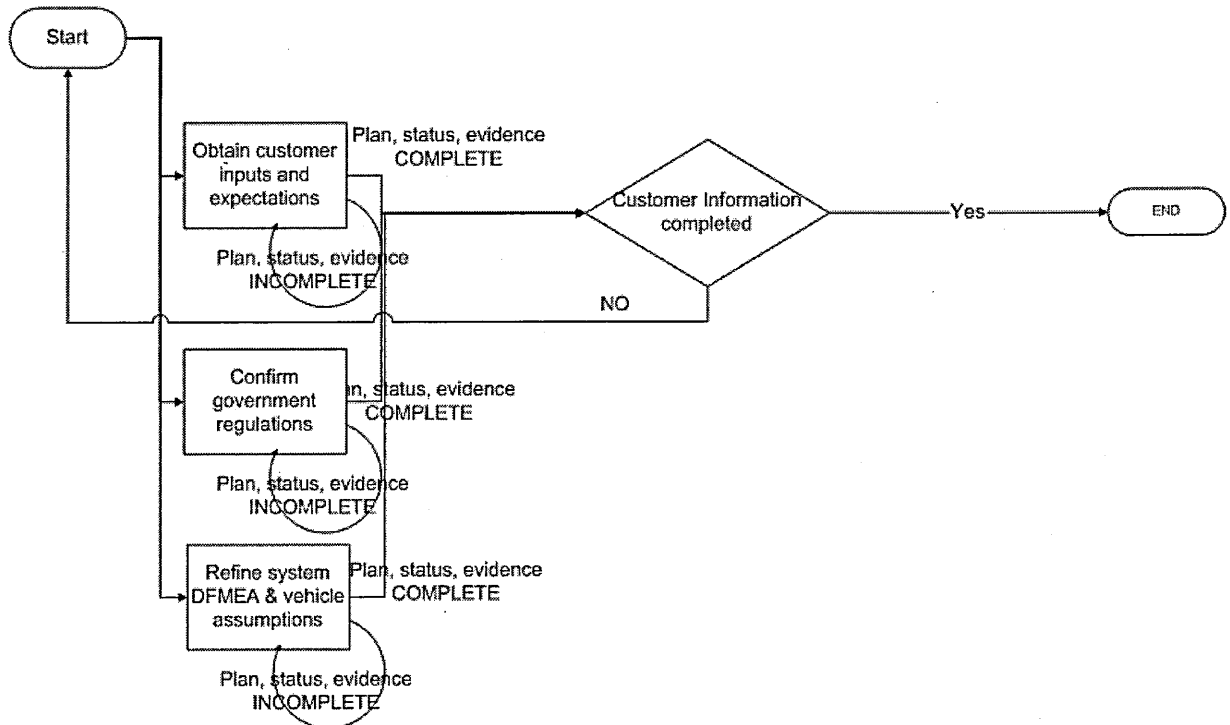
ArvinMeritor: Phase 1, Process capability analysis of product design classified prints



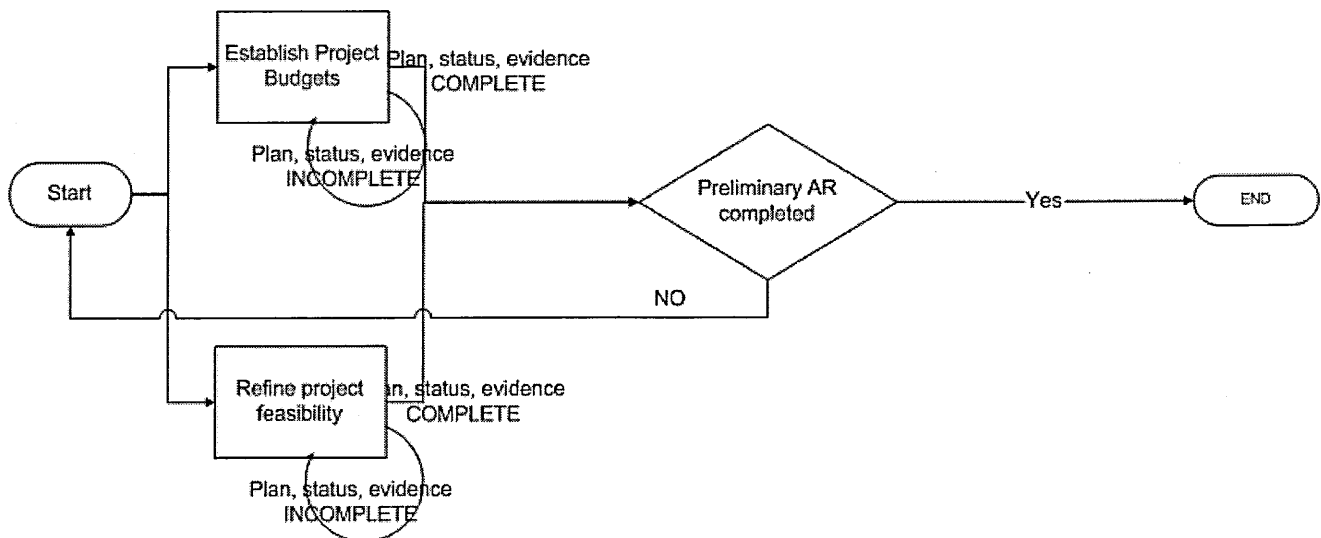
ArvinMeritor: Phase 1, Project Go/No go recommendation



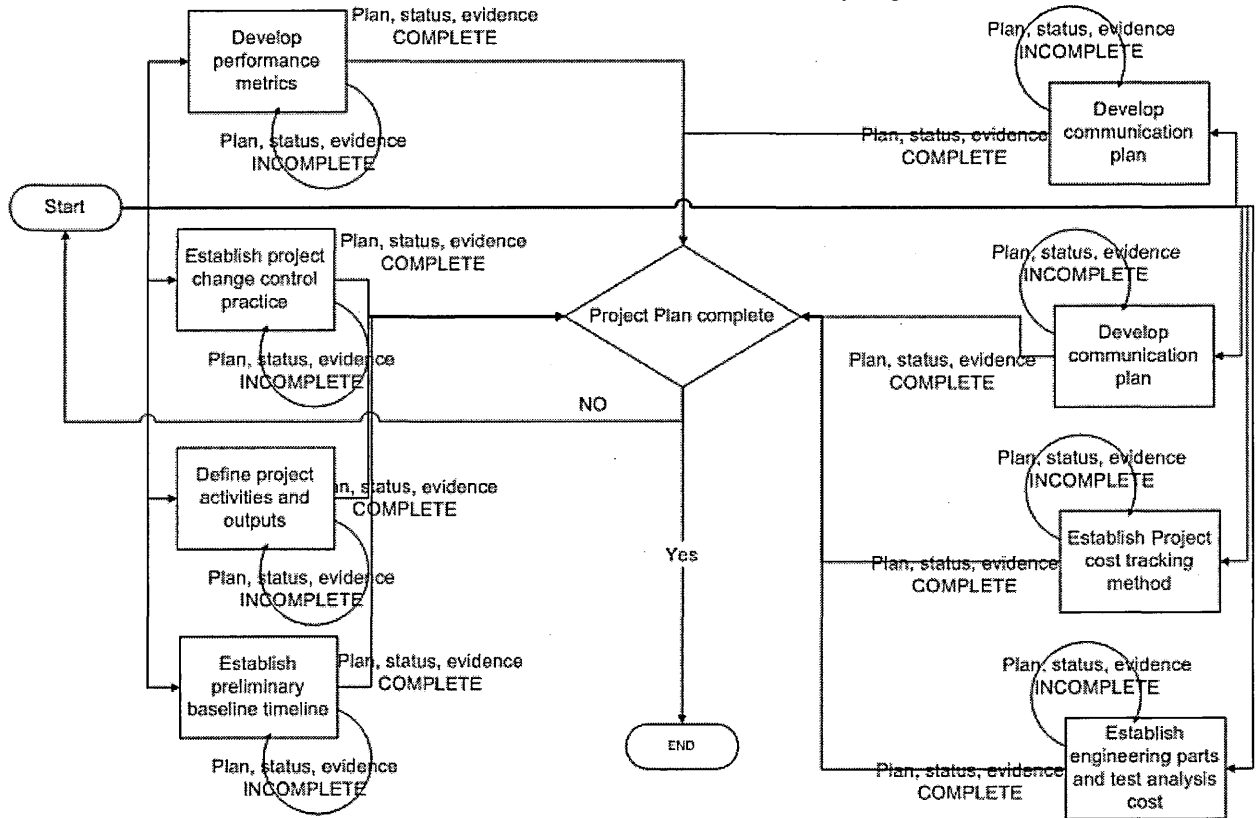
ArvinMeritor: Phase 1, Customer Information



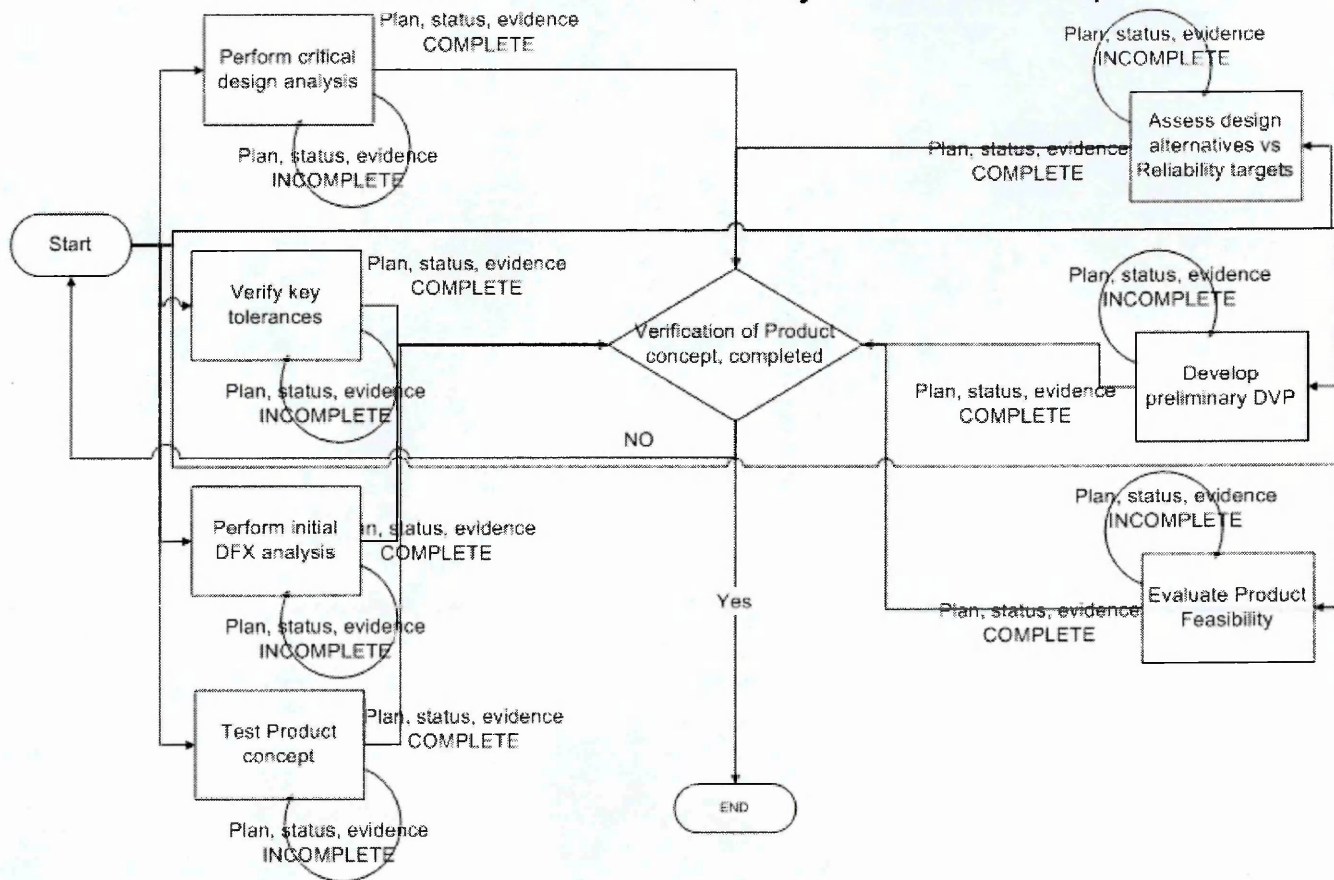
ArvinMeritor: Phase 1, Develop Preliminary AR



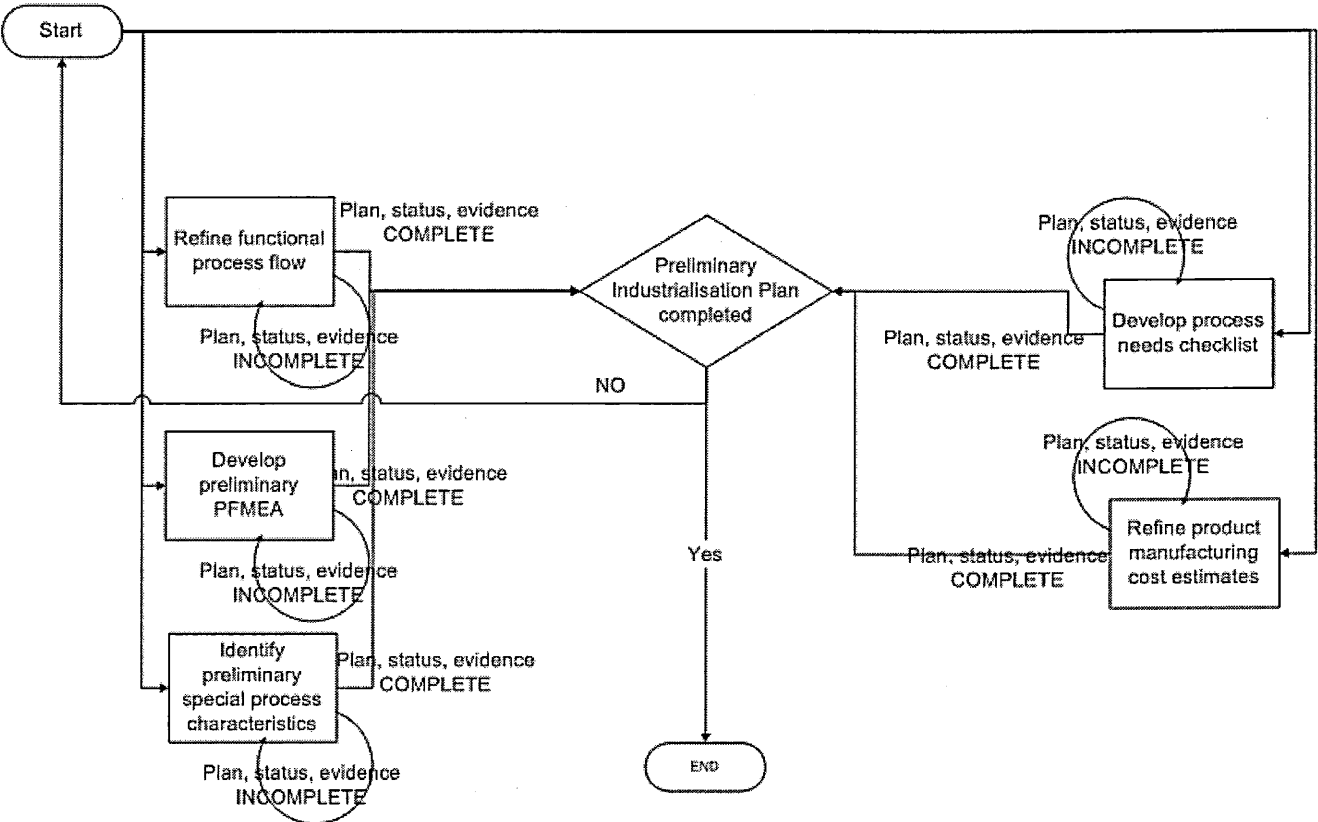
ArvinMeritor: Phase 1, Plan project



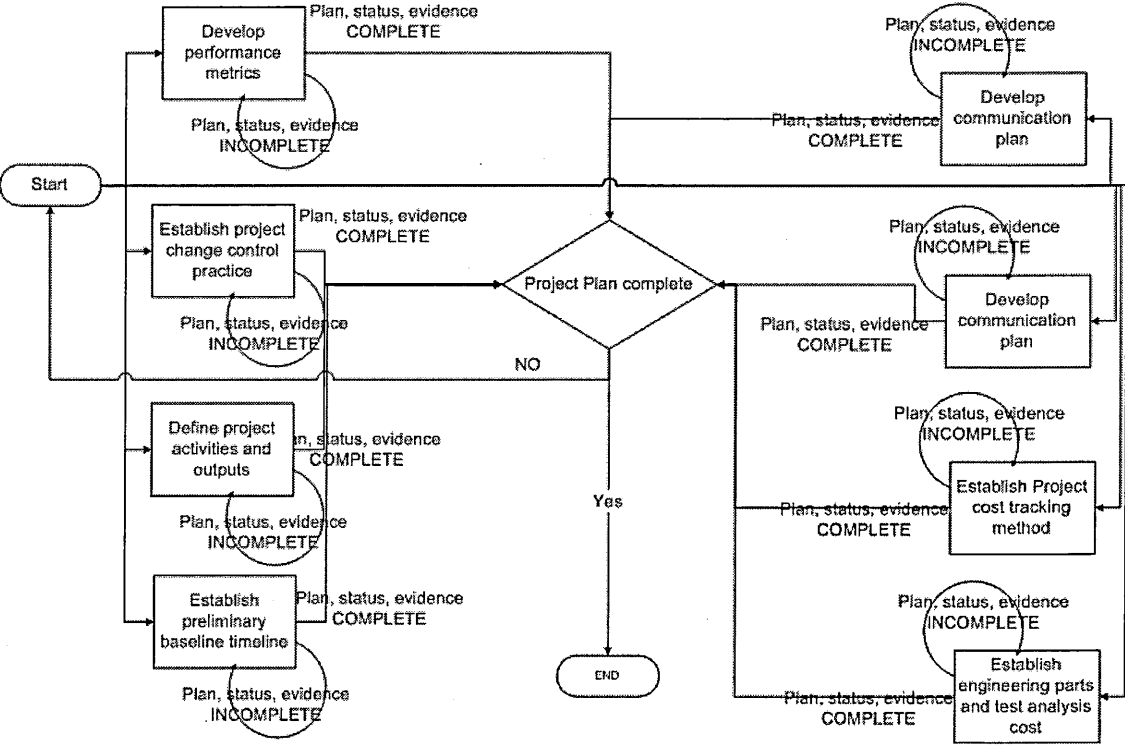
ArvinMeritor: Phase 1, Verify Product concept



ArvinMeritor: Phase 1, Preliminary Industrialisation Plan



ArvinMeritor: Phase 1, Plan project



❖ Appendix (C) STEP ontology in Knowledge Interchange Format

The source code below shows the class and variable 'slot' definitions of the ontology defining the STEP application protocols (AP-224).

Defclass is a command declaring that a class is about to be defined. The TOP_LEVEL-SLOT_CLASS is actually a definition of the reusable variables or slots which are available for use within the ontology by the different classes. These are 'top-level' slots, meaning that any class in the ontology may utilise them.

Role of the class can be either as an abstract or concrete class (Abstract cannot be instantiated), IS-A defines the inherited properties from a super class.

For every slot definition, single_slot defines that each object may have one instance of that variable, or multiple instances of the variable can be defined with multiple_slot and this can either be limited to a certain number of instances or unlimited.

Type defines the data type of the slot, which can be integer, string, floating-point, instance of a class (or multiple instances of a class), or a symbol (pre-defined multiple-choice list).

Create-accessor defines the access rights and privileges for that slot within the overall ontology.

The actual items defined as top level slots are the same as those defined in EXPRESS for the STEP standard. These include mundane items like axis2_placement which is a symbolic variable with a choice of potential values being 2d or 3d (i.e. defining whether the second axis is within a 2d or 3d path).

Other definitions include `attribute_select` which is also a symbolic variable with multiple choice values including `property_definition_representation` `external_source` `representation` `date_role` `person_and_organization` `approval_role` `organization_role` `action_request_solution` `person_and_organization_role` `application_context`. This slot is used in classes to define the role of the instantiated object with one of the above attributes.

There are almost 100 variable ‘slot’ types defined for the STEP ontology alone. These are included below.

```
(defclass %3ACLIPS_TOP_LEVEL_SLOT_CLASS
  (is-a USER)
  (role concrete)
  (single-slot dimension_count
    (type INTEGER)
    (range 1 %3FVARIABLE)
    (cardinality 0 1)
    (create-accessor read-write))
  (single-slot postal_code
    (type STRING)
    (cardinality 0 1)
    (create-accessor read-write))
  (single-slot product_or_formation_or_definition
    (type SYMBOL)
    (allowed-values product product_definition_formation product_definition)
    (cardinality 0 1)
    (create-accessor read-write))
  (single-slot frame_of_reference
    (type INSTANCE)
    (allowed-classes application_context)
    (cardinality 0 1)
    (create-accessor read-write))
  (single-slot id_attribute_select
    (type SYMBOL)
    (allowed-values representation action application_context)
    (cardinality 0 1)
    (create-accessor read-write))
  (single-slot street
    (type STRING)
    (cardinality 0 1)
    (create-accessor read-write))
  (single-slot founded_item_select
    (type SYMBOL)
    (allowed-values representation_item founded_item)
    (cardinality 0 1)
    (create-accessor read-write))
  (single-slot axis2_placement
    (type SYMBOL)
    (allowed-values axis2_placement_2d axis2_placement_3d)
    (cardinality 0 1)
    (create-accessor read-write))
  (single-slot upper_index_on_control_points
    (type INTEGER)
    (cardinality 0 1)
    (create-accessor read-write))
  (multislot control_points_list
    (type INSTANCE)
    (allowed-classes cartesian_point))
```

```

        (cardinality 0 2)
        (create-accessor read-write))
    (single-slot description_attribute_select
        (type SYMBOL)
        (allowed-values property_definition_representation external_source_representation date_role
person_and_organization_approval_role organization_role action_request_solution person_and_organization_role
application_context)
;+        (cardinality 0 1)
        (create-accessor read-write))
    (single-slot country
        (type STRING)
;+        (cardinality 0 1)
        (create-accessor read-write))
    (single-slot degree
        (type INTEGER)
;+        (cardinality 0 1)
        (create-accessor read-write))
    (single-slot id
        (type INSTANCE)
;+        (allowed-classes _T_identifier)
;+        (cardinality 0 1)
        (create-accessor read-write))
    (single-slot characterized_definition
        (type SYMBOL)
        (allowed-values characterized_product_definition characterized_object_shape_definition)
;+        (cardinality 0 1)
        (create-accessor read-write))
    (single-slot purpose
;+        (comment "text")
        (type STRING)
;+        (cardinality 0 1)
        (create-accessor read-write))
    (single-slot role_select
        (type SYMBOL)
        (allowed-values action_assignment action_request_assignment approval_assignment
approval_date_time document_reference group_assignment security_classification_assignment)
;+        (cardinality 0 1)
        (create-accessor read-write))
    (single-slot day_in_week_number
        (type INTEGER)
        (range 1 7)
;+        (cardinality 0 1)
        (create-accessor read-write))
    (single-slot closed_curve
        (type SYMBOL)
        (allowed-values FALSE TRUE)
;+        (cardinality 0 1)
        (create-accessor read-write))
    (multislot v_multiplicities
        (type INTEGER)
        (cardinality 0 2)
        (create-accessor read-write))
    (single-slot ap224_clips_00354
        (type INSTANCE)
;+        (allowed-classes action)
;+        (cardinality 0 1)
        (create-accessor read-write))
    (single-slot transformation
;+        (comment "functionally_defined_transformation")
        (type STRING)
;+        (cardinality 0 1)
        (create-accessor read-write))
    (single-slot feature_based_pp_person_and_organization_item
        (type SYMBOL)

```

```

                (allowed-values action_directive product_definition_formation)
;+                (cardinality 0 1)
                (create-accessor read-write))
(single-slot axis
  (type INSTANCE)
;+  (allowed-classes direction)
;+  (cardinality 0 1)
  (create-accessor read-write))
(multislot u_knots
  (type FLOAT)
  (cardinality 0 2)
  (create-accessor read-write))
(single-slot knot_spec
  (type INSTANCE)
;+  (allowed-classes _T_knot_type)
;+  (cardinality 0 1)
  (create-accessor read-write))
(single-slot tolerance_select
  (type SYMBOL)
  (allowed-values geometric_tolerance plus_minus_tolerance)
;+  (cardinality 0 1)
  (create-accessor read-write))
(single-slot application_protocol_year
  (type INSTANCE)
;+  (allowed-classes _T_year_number)
;+  (cardinality 0 1)
  (create-accessor read-write))
(single-slot relating_action
  (type INSTANCE)
;+  (allowed-classes action)
;+  (cardinality 0 1)
  (create-accessor read-write))
(single-slot angle_selection
  (type INSTANCE)
;+  (allowed-classes _T_angle_relator)
;+  (cardinality 0 1)
  (create-accessor read-write))
(single-slot comment
;+  (comment "text")
  (type STRING)
;+  (cardinality 0 1)
  (create-accessor read-write))
(single-slot v_upper
  (type INTEGER)
;+  (cardinality 0 1)
  (create-accessor read-write))
(single-slot internal_location
  (type STRING)
;+  (cardinality 0 1)
  (create-accessor read-write))
(single-slot town
;+  (type STRING)
  (cardinality 0 1)
  (create-accessor read-write))
(single-slot dimensional_characteristic
  (type SYMBOL)
  (allowed-values dimensional_size dimensional_location)
;+  (cardinality 0 1)
  (create-accessor read-write))
(single-slot _representation_item
  (type SYMBOL)
  (allowed-values geometric_representation_item topological_representation_item)
;+  (cardinality 0 1)
  (create-accessor read-write))

```



```

        (single-slot v_degree
          (type INTEGER)
;+      (cardinality 0 1)
          (create-accessor read-write))
        (single-slot level
          (type STRING)
;+      (cardinality 0 1)
          (create-accessor read-write))
        (single-slot feature_based_pp_dated_item
          (type SYMBOL)
;+      (allowed-values versioned_action_request directed_action document)
          (cardinality 0 1)
          (create-accessor read-write))
        (single-slot supported_item
          (type SYMBOL)
;+      (allowed-values action_directive action action_method)
          (cardinality 0 1)
          (create-accessor read-write))
        (single-slot p
          (type FLOAT)
;+      (cardinality 0 1)
          (create-accessor read-write))
        (single-slot %3ASTARTUP-EXPRESSION
          (type STRING)
;+      (cardinality 0 1)
          (create-accessor read-write))
        (single-slot year_number
          (type INTEGER)
;+      (cardinality 0 1)
          (create-accessor read-write))
        (single-slot reference_designator
          (type STRING)
;+      (cardinality 0 1)
          (create-accessor read-write))
        (single-slot u_degree
          (type INTEGER)
;+      (cardinality 0 1)
          (create-accessor read-write))
        (single-slot %3AKB-SAVE
          (type SYMBOL)
;+      (allowed-values FALSE TRUE)
          (cardinality 0 1)
          (create-accessor read-write))
        (single-slot street_number
          (type STRING)
;+      (cardinality 0 1)
          (create-accessor read-write))
        (single-slot assigned_action
          (type INSTANCE)
;+      (allowed-classes executed_action)
;+      (cardinality 0 1)
          (create-accessor read-write))
        (single-slot reversible_topology
          (type SYMBOL)
          (allowed-values      reversible_topology_item      list_of_reversible_topology_item
set_of_reversible_topology_item)
;+      (cardinality 0 1)
          (create-accessor read-write))
        (single-slot represented_definition
          (type SYMBOL)
          (allowed-values      property_definition      property_definition_relationship      shape_aspect
shape_aspect_relationship)
;+      (cardinality 0 1)
          (create-accessor read-write))

```

```

(single-slot geometric_set_select
  (type SYMBOL)
  (allowed-values curve surface point)
;+
  (cardinality 0 1)
  (create-accessor read-write))
(single-slot telex_number
  (type STRING)
;+
  (cardinality 0 1)
  (create-accessor read-write))
(single-slot si_unit_name
  (type SYMBOL)
  (allowed-values metre gram second ampere kelvin mole candela radian steradian hertz newton
pascal joule watt coulomb volt farad ohm siemens weber tesla henry degree_celsius lumen lux becquerel gray sievert)
;+
  (cardinality 0 1)
  (create-accessor read-write))
(single-slot transition_code
  (type SYMBOL)
  (allowed-values discontinuous continuous cont_same_gradient
cont_same_gradient_same_curvature)
;+
  (cardinality 0 1)
  (create-accessor read-write))
(single-slot feature_based_pp_approved_item
  (type SYMBOL)
  (allowed-values directed_action product_definition_formation)
;+
  (cardinality 0 1)
  (create-accessor read-write))
(single-slot day_in_month_number
  (type INTEGER)
  (range 1 31)
;+
  (cardinality 0 1)
  (create-accessor read-write))
(single-slot region
  (type STRING)
;+
  (cardinality 0 1)
  (create-accessor read-write))
(single-slot feature_based_pp_classified_item
  (type STRING)
  (default "product_definition_formation")
;+
  (cardinality 0 1)
  (create-accessor read-write))
(single-slot self_intersect
  (type SYMBOL)
  (allowed-values FALSE TRUE)
;+
  (cardinality 0 1)
  (create-accessor read-write))
(single-slot parameter_value
  (type FLOAT)
;+
  (cardinality 0 1)
  (create-accessor read-write))
(single-slot person_organisation
  (type INSTANCE)
;+
  (allowed-classes _T_person_organisation_select)
;+
  (cardinality 0 1)
  (create-accessor read-write))
(single-slot angle_relator
  (type SYMBOL)
  (allowed-values large small equal)
;+
  (cardinality 0 1)
  (create-accessor read-write))
(single-slot dated_approval
  (type INSTANCE)
;+
  (allowed-classes approval)
;+
  (cardinality 0 1)
  (create-accessor read-write))

```

```

        (single-slot status
;+          (comment "label")
            (type STRING)
;+          (cardinality 0 1)
            (create-accessor read-write))
        (single-slot name_attribute_select
            (type SYMBOL)
            (allowed-values action_request_solution derived_unit person_and_organization
product_definition property_definition_representation)
;+          (cardinality 0 1)
            (create-accessor read-write))
        (single-slot feature_based_pp_action_item
            (type STRING)
;+          (value "product_definition_formation")
;+          (cardinality 0 1)
            (create-accessor read-write))
        (single-slot day_in_year_number
            (type INTEGER)
            (range 1 366)
;+          (cardinality 0 1)
            (create-accessor read-write))
        (single-slot person_organization_select
            (type SYMBOL)
            (allowed-values person organization person_and_organization)
;+          (cardinality 0 1)
            (create-accessor read-write))
        (single-slot measure_value
            (type SYMBOL)
            (allowed-values length_measure plane_angle_measure ratio_measure parameter_value
context_dependent_measure descriptive_measure positive_length_measure positive_plane_angle_measure
count_measure)
;+          (cardinality 0 1)
            (create-accessor read-write))
        (single-slot week_in_year_number
            (type INTEGER)
            (range 1 53)
;+          (cardinality 0 1)
            (create-accessor read-write))
        (single-slot %3ADEFINITION
            (type STRING)
;+          (cardinality 0 1)
            (create-accessor read-write))
        (single-slot curve_form
            (type INSTANCE)
;+          (allowed-classes _T_b_spline_curve_form)
;+          (cardinality 0 1)
            (create-accessor read-write))
        (single-slot value_qualifier
            (type SYMBOL)
            (allowed-values precision_qualifier type_qualifier uncertainty_qualifier)
;+          (cardinality 0 1)
            (create-accessor read-write))
        (single-slot feature_based_pp_action_request_item
            (type STRING)
;+          (value "product_definition_formation")
;+          (cardinality 0 1)
            (create-accessor read-write))
        (single-slot characterized_product_definition
            (type SYMBOL)
            (allowed-values product_definition_relationship product_definition)
;+          (cardinality 0 1)
            (create-accessor read-write))
        (single-slot unit
            (type SYMBOL)

```

```

        (allowed-values named_unit derived_unit)
;+      (cardinality 0 1)
        (create-accessor read-write))
(single-slot si_prefix
  (type SYMBOL)
  (allowed-values exa peta tera giga mega kilo hecto deca deci centi milli micro nano pico femto
atto)
;+      (cardinality 0 1)
        (create-accessor read-write))
(single-slot related_action
  (type INSTANCE)
;+      (allowed-classes action)
;+      (cardinality 0 1)
        (create-accessor read-write))
(single-slot ref_direction
  (type INSTANCE)
;+      (allowed-classes direction)
;+      (cardinality 0 1)
        (create-accessor read-write))
(single-slot description
;+      (comment "OPTIONAL text")
        (type STRING)
;+      (cardinality 0 1)
        (create-accessor read-write))
(single-slot count_measure
  (type STRING)
  (default "NUMBER")
;+      (cardinality 0 1)
        (create-accessor read-write))
(single-slot tolerance_method_definition
  (type SYMBOL)
  (allowed-values tolerance_value limits_and_fits)
;+      (cardinality 0 1)
        (create-accessor read-write))
(single-slot length_measure
  (type FLOAT)
;+      (cardinality 0 1)
        (create-accessor read-write))
(single-slot requests
  (type INSTANCE)
;+      (allowed-classes versioned_action_request)
;+      (cardinality 0 1)
        (create-accessor read-write))
(single-slot document_reference_item
  (type INSTANCE)
;+      (allowed-classes _T_document_reference_item)
;+      (cardinality 0 1)
        (create-accessor read-write))
(single-slot assigned_action_request
  (type INSTANCE)
;+      (allowed-classes versioned_action_request)
;+      (cardinality 0 1)
        (create-accessor read-write))
(single-slot name_
;+      (comment "label")
        (type STRING)
;+      (cardinality 0 1)
        (create-accessor read-write))
(single-slot v_closed
  (type SYMBOL)
  (allowed-values FALSE TRUE)
;+      (cardinality 0 1)
        (create-accessor read-write))
(single-slot descriptive_measure

```

```

;+      (type STRING)
        (cardinality 0 1)
        (create-accessor read-write))
(multislot upper_index_on_knots
  (type INTEGER)
  (cardinality 0 2)
  (create-accessor read-write))
(single-slot method
  (type INSTANCE)
;+      (allowed-classes action_method)
;+      (cardinality 0 1)
        (create-accessor read-write))
(multislot knots
  (type FLOAT)
  (cardinality 0 2)
  (create-accessor read-write))
(single-slot vector_or_direction

```

COMMENT:

The function classes below are simply an abstract definition for the system classes which instantiate functions and rules within the protégé system. These are core to the protégé system and the classes which define the 'rule' entities of the STEP standard and the inter-enterprise knowledge base are instances of the subclasses of class JESS-Definition below.

```

(defclass %3AFUNCTION
  (is-a %3ASYSTEM-CLASS)
  (role abstract)
  (single-slot %3ADEFINITION-NAME
    (type STRING)
;+    (cardinality 0 1)
    (create-accessor read-write))
  (single-slot %3ADEFINITION
    (type STRING)
;+    (cardinality 0 1)
    (create-accessor read-write)))

```

This abstract class is the root class for the main rule engine classes. It declares the global variables which are included in the concrete subclasses. The declared variables simply include the name of the 'object' a string containing the source code as a string and whether the object is saved in the knowledge base or flushed.

```

(defclass %3AJESS-DEFINITION
  (is-a %3ASYSTEM-CLASS)
  (role abstract)
  (single-slot %3ADEFINITION-NAME
    (type STRING)
;+    (cardinality 0 1)
    (create-accessor read-write))
  (single-slot %3ADEFINITION
    (type STRING)
;+    (cardinality 0 1)
    (create-accessor read-write))

```

```

      (single-slot %3AKB-SAVE
        (type SYMBOL)
        (allowed-values FALSE TRUE)
;+      (cardinality 0 1)
        (create-accessor read-write)))

```

Deffacts are subclasses of JESS_definitions that define concrete ‘facts’ within the system (constants). It is a concrete subclass of jess-definition and the objects instantiated are the constants within the knowledge base.

```

(defclass %3AJESS-DEFFACTS
  (is-a %3AJESS-DEFINITION)
  (role concrete))

```

The defglobal function within jess defines static and global constants within the knowledge base. It’s a concrete class and the instantiated objects are stored in the knowledge base.

```

(defclass %3AJESS-DEFGLOBAL
  (is-a %3AJESS-DEFINITION)
  (role concrete))

```

```

(defclass %3AJESS-DEFMESSAGE-HANDLER
  (is-a %3AJESS-DEFINITION)
  (role concrete))

```

Defmethods are routines which can run on knowledge base (defined in KIF). The routines are instantiated as objects of the class, and also inherit the base properties from jess-definition.

```

(defclass %3AJESS-DEFMETHOD
  (is-a %3AJESS-DEFINITION)
  (role concrete))

```

The classes below are used by the JESS engine, and included for reference purposes.

```

(defclass %3AJESS-DEFTEMPLATE
  (is-a %3AJESS-DEFINITION)
  (role concrete))

```

```

(defclass %3AJESS-FUNCTION
  (is-a %3AFUNCTION %3AJESS-DEFINITION)
  (role concrete))

```

```

(defclass %3AJESS-ENGINE
  (is-a %3ASYSTEM-CLASS)
  (role concrete)
  (single-slot %3ASTARTUP-EXPRESSION
    (type STRING)
;+    (cardinality 0 1)
    (create-accessor read-write))
  (single-slot %3ASTARTUP-FILE
    (type STRING)
;+    (cardinality 0 1)

```

```

                (create-accessor read-write)))

(defclass %3ARULE
  (is-a %3ASYSTEM-CLASS)
  (role abstract)
  (single-slot %3ADEFINITION-NAME
    (type STRING)
    (cardinality 0 1)
    (create-accessor read-write))
  ;+
  (single-slot %3ADEFINITION
    (type STRING)
    (cardinality 0 1)
    (create-accessor read-write)))
  ;+
  (create-accessor read-write)))

```

The JESS-RULE class below is a concrete class which is a subclass of JESS-Definition. This is the class which is used for instantiating rules in the JESS expert system engine. Each rule is an object of the class below and inherits the variables found above. The instances are for both the STEP standard rules and the generic knowledge base workflow and rule engine.

```

(defclass %3AJESS-RULE
  (is-a %3AJESS-DEFINITION %3ARULE)
  (role concrete))

```

The following concrete classes below define the Express schema for the STEP standard AP-224. The class names, types are directly equivalent to the STEP standard entities, and have been translated to KIF knowledge base as a component of the integrated knowledge base. The STEP documentation fully documents each class (entity) in the schema, however a number here are also documented for the sake of brevity and to clearly understand how the STEP schema can be translated from Express to KIF.

```

(defclass _TYPE
  (is-a USER)
  (role abstract))

(defclass _T_identifier
  (is-a _TYPE)
  (role concrete)
  (single-slot identifier
    (type STRING)
    (cardinality 0 1)
    (create-accessor read-write)))
  ;+

(defclass _T_feature_based_pp_ordered_item
  (is-a _TYPE)
  (role concrete)
  (single-slot feature_based_pp_ordered_item
    (type STRING)
    (default "product_definition_formation")
    (cardinality 0 1))
  ;+

```

```

        (create-accessor read-write)))

(defclass _T_set_of_reversible_topology_item
  (is-a _TYPE)
  (role concrete)
  (single-slot set_of_reversible_topology_item
    (type SYMBOL)
    (allowed-values FALSE TRUE)
    (cardinality 0 1)
    (create-accessor read-write)))
;+

(defclass _T_value_qualifier
  (is-a _TYPE)
  (role concrete)
  (single-slot value_qualifier
    (type SYMBOL)
    (allowed-values precision_qualifier type_qualifier uncertainty_qualifier)
    (cardinality 0 1)
    (create-accessor read-write)))
;+

(defclass _T_positive_length_measure
  (is-a _TYPE)
  (role concrete)
  (single-slot positive_length_measure
    (type FLOAT)
    (range 0.0 %3FVARIABLE)
    (cardinality 0 1)
    (create-accessor read-write)))
;+

(defclass _T_si_prefix
  (is-a _TYPE)
  (role concrete)
  (single-slot si_prefix
    (type SYMBOL)
    (allowed-values exa peta tera giga mega kilo hecto deca deci centi milli micro nano pico femto
atto)
    (cardinality 0 1)
    (create-accessor read-write)))
;+

(defclass _T_ratio_measure
  (is-a _TYPE)
  (role concrete)
  (single-slot ratio_measure
    (type FLOAT)
    (cardinality 0 1)
    (create-accessor read-write)))
;+

(defclass _T_year_number
  (is-a _TYPE)
  (role concrete)
  (single-slot year_number
    (type INTEGER)
    (cardinality 0 1)
    (create-accessor read-write)))
;+

(defclass _T_shape_definition
  (is-a _TYPE)
  (role concrete)
  (single-slot shape_definition
    (type SYMBOL)
    (allowed-values product_definition_shape shape_aspect shape_aspect_relationship)
    (cardinality 0 1)
    (create-accessor read-write)))
;+

```



```

(defclass _T_characterised_material_property
  (is-a _TYPE)
  (role concrete)
  (single-slot characterized_material_property
    (type STRING)
    (value "material_property_representation")
    (cardinality 0 1)
    (create-accessor read-write)))

(defclass _T_document_reference_item
  (is-a _TYPE)
  (role concrete)
  (single-slot document_reference_item
    (type SYMBOL)
    (allowed-values directed_action property_definition action_method
externally_defined_feature_definition dimensional_characteristic_representation)
    (cardinality 0 1)
    (create-accessor read-write)))

(defclass _T_limit_condition
  (is-a _TYPE)
  (role concrete)
  (single-slot limit_condition
    (type SYMBOL)
    (allowed-values regardless_of_feature_size least_material_condition
maximum_material_condition)
    (cardinality 0 1)
    (create-accessor read-write)))

(defclass _T_person_organisation_select
  (is-a _TYPE)
  (role concrete)
  (single-slot person_organization_select
    (type SYMBOL)
    (allowed-values person organization person_and_organization)
    (cardinality 0 1)
    (create-accessor read-write)))

(defclass _T_date_time_or_event_occurrence
  (is-a _TYPE)
  (role concrete)
  (single-slot date_time_or_event_occurrence
    (type STRING)
    (value "date_time_select")
    (cardinality 0 1)
    (create-accessor read-write)))

(defclass _T_feature_based_pp_classified_item
  (is-a _TYPE)
  (role concrete)
  (single-slot feature_based_pp_classified_item
    (type STRING)
    (default "product_definition_formation")
    (cardinality 0 1)
    (create-accessor read-write)))

(defclass _T_shell
  (is-a _TYPE)
  (role concrete)
  (single-slot shell
    (type SYMBOL)
    (allowed-values open_shell closed_shell)
    (cardinality 0 1)
    (create-accessor read-write)))

```

```

(defclass _T_transition_code
  (is-a _TYPE)
  (role concrete)
  (single-slot transition_code
    (type SYMBOL)
    (allowed-values discontinuous continuous cont_same_gradient
cont_same_gradient_same_curvature)
    (cardinality 0 1)
    (create-accessor read-write)))

;+

(defclass _T_list_of_reversible_topology_item
  (is-a _TYPE)
  (role concrete)
  (single-slot list_of_reversible_topology_item
    (type INSTANCE)
    (allowed-classes)
    (cardinality 0 1)
    (create-accessor read-write)))

;+
;+

(defclass _T_feature_based_pp_action_item
  (is-a _TYPE)
  (role concrete)
  (single-slot feature_based_pp_action_item
    (type STRING)
    (value "product_definition_formation")
    (cardinality 0 1)
    (create-accessor read-write)))

;+
;+

(defclass _T_feature_based_pp_person_and_organisation_item
  (is-a _TYPE)
  (role concrete)
  (single-slot feature_based_pp_person_and_organization_item
    (type SYMBOL)
    (allowed-values action_directive product_definition_formation)
    (cardinality 0 1)
    (create-accessor read-write)))

;+

(defclass _T_name_attribute_select
  (is-a _TYPE)
  (role concrete)
  (single-slot name_attribute_select
    (type SYMBOL)
    (allowed-values action_request_solution derived_unit person_and_organization
product_definition property_definition_representation)
    (cardinality 0 1)
    (create-accessor read-write)))

;+

(defclass _T_positive_plane_angle_measure
  (is-a _TYPE)
  (role concrete)
  (single-slot positive_plane_angle_measure
    (type FLOAT)
    (range 0.0 %3FVARIABLE)
    (cardinality 0 1)
    (create-accessor read-write)))

;+

(defclass _T_b_spline_surface_form
  (is-a _TYPE)
  (role concrete)
  (single-slot b_spline_surface_form
    (type SYMBOL)
    (allowed-values conical_surf surf_of_linear_extrusion ruled_surf plane_surf spherical_surf
surf_of_revolution toroidal_surf unspecified cylindrical_surf generalised_cone quadric_surf)

```

```

;+          (cardinality 0 1)
            (create-accessor read-write)))

(defclass _T_transformation
  (is-a _TYPE)
  (role concrete)
  (single-slot transformation
;+          (comment "functionally_defined_transformation")
            (type STRING)
;+          (cardinality 0 1)
            (create-accessor read-write)))

(defclass _T_angle_relator
  (is-a _TYPE)
  (role concrete)
  (single-slot angle_relator
            (type SYMBOL)
            (allowed-values large small equal)
;+          (cardinality 0 1)
            (create-accessor read-write)))

(defclass _T_feature_based_pp_action_request_item
  (is-a _TYPE)
  (role concrete)
  (single-slot feature_based_pp_action_request_item
            (type STRING)
;+          (value "product_definition_formation")
;+          (cardinality 0 1)
            (create-accessor read-write)))

(defclass _T_day_in_week_number
  (is-a _TYPE)
  (role concrete)
  (single-slot day_in_week_number
            (type INTEGER)
            (range 1 7)
;+          (cardinality 0 1)
            (create-accessor read-write)))

(defclass _T_dimensional_characteristics
  (is-a _TYPE)
  (role concrete)
  (single-slot dimensional_characteristic
            (type SYMBOL)
            (allowed-values dimensional_size dimensional_location)
;+          (cardinality 0 1)
            (create-accessor read-write)))

(defclass _T_pcurve_or_surface
  (is-a _TYPE)
  (role concrete)
  (single-slot pcurve_or_surface
            (type SYMBOL)
            (allowed-values pcurve surface)
;+          (cardinality 0 1)
            (create-accessor read-write)))

(defclass _T_reversible_topology_item
  (is-a _TYPE)
  (role concrete)
  (single-slot reversible_topology_item
            (type SYMBOL)
            (allowed-values edge path face face_bound closed_shell open_shell)
;+          (cardinality 0 1)

```

```

        (create-accessor read-write)))

(defclass _T_month_in_year_number
  (is-a _TYPE)
  (role concrete)
  (single-slot month_in_year_number
    (type INTEGER)
    (range 1 12)
    (cardinality 0 1)
    (create-accessor read-write)))
;+

(defclass _T_label
  (is-a _TYPE)
  (role concrete)
  (single-slot label
    (type STRING)
    (cardinality 0 1)
    (create-accessor read-write)))
;+

(defclass _T_derived_property_select
  (is-a _TYPE)
  (role concrete)
  (single-slot derived_property_select
    (type STRING)
    (value "property_definition")
    (cardinality 0 1)
    (create-accessor read-write)))
;+

(defclass _T_vector_or_direction
  (is-a _TYPE)
  (role concrete)
  (single-slot vector_or_direction
    (type SYMBOL)
    (allowed-values vector direction)
    (cardinality 0 1)
    (create-accessor read-write)))
;+

(defclass _T_founded_item_select
  (is-a _TYPE)
  (role concrete)
  (single-slot founded_item_select
    (type SYMBOL)
    (allowed-values representation_item founded_item)
    (cardinality 0 1)
    (create-accessor read-write)))
;+

(defclass _T_plane_angle_measure
  (is-a _TYPE)
  (role concrete)
  (single-slot plane_angle_measure
    (type FLOAT)
    (cardinality 0 1)
    (create-accessor read-write)))
;+

(defclass _T_measure_value
  (is-a _TYPE)
  (role concrete)
  (single-slot measure_value
    (type SYMBOL)
    (allowed-values length_measure plane_angle_measure ratio_measure parameter_value
context_dependent_measure descriptive_measure positive_length_measure positive_plane_angle_measure
count_measure)
    (cardinality 0 1)
    (create-accessor read-write)))
;+

```

```

(defclass _T_axis2_placement
  (is-a _TYPE)
  (role concrete)
  (single-slot axis2_placement
    (type SYMBOL)
    (allowed-values axis2_placement_2d axis2_placement_3d)
    (cardinality 0 1)
    (create-accessor read-write)))

;+

(defclass _T_characterized_definition
  (is-a _TYPE)
  (role concrete)
  (single-slot characterized_definition
    (type SYMBOL)
    (allowed-values characterized_product_definition characterized_object shape_definition)
    (cardinality 0 1)
    (create-accessor read-write)))

;+

(defclass _T_represented_definition
  (is-a _TYPE)
  (role concrete)
  (single-slot represented_definition
    (type SYMBOL)
    (allowed-values property_definition property_definition_relationship shape_aspect
shape_aspect_relationship)
    (cardinality 0 1)
    (create-accessor read-write)))

;+

(defclass _T_preferred_surface_curve_representation
  (is-a _TYPE)
  (role concrete)
  (single-slot preferred_surface_curve_representation
    (type SYMBOL)
    (allowed-values curve_3d pcurve_s1 pcurve_s2)
    (cardinality 0 1)
    (create-accessor read-write)))

;+

(defclass _T_characterized_product_definition
  (is-a _TYPE)
  (role concrete)
  (single-slot characterized_product_definition
    (type SYMBOL)
    (allowed-values product_definition_relationship product_definition)
    (cardinality 0 1)
    (create-accessor read-write)))

;+

(defclass _T_feature_based_pp_organization_item
  (is-a _TYPE)
  (role concrete)
  (single-slot feature_based_pp_organization_item
    (type SYMBOL)
    (allowed-values action_directive product_definition_formation)
    (cardinality 0 1)
    (create-accessor read-write)))

;+

(defclass _T_feature_based_pp_approved_item
  (is-a _TYPE)
  (role concrete)
  (single-slot feature_based_pp_approved_item
    (type SYMBOL)
    (allowed-values directed_action product_definition_formation)
    (cardinality 0 1)
    (create-accessor read-write)))

;+

```

```

(defclass _T_role_select
  (is-a _TYPE)
  (role concrete)
  (single-slot role_select
    (type SYMBOL)
    (allowed-values action_assignment action_request_assignment approval_assignment
approval_date_time document_reference group_assignment security_classification_assignment)
;+ (cardinality 0 1)
    (create-accessor read-write)))

(defclass _T_parameter_value
  (is-a _TYPE)
  (role concrete)
  (single-slot parameter_value
    (type FLOAT)
;+ (cardinality 0 1)
    (create-accessor read-write)))

(defclass _T_description_attribute_select
  (is-a _TYPE)
  (role concrete)
  (single-slot description_attribute_select
    (type SYMBOL)
    (allowed-values property_definition_representation external_source_representation date_role
person_and_organization approval_role organization_role action_request_solution person_and_organization_role
application_context)
;+ (cardinality 0 1)
    (create-accessor read-write)))

(defclass _T_source_item
  (is-a _TYPE)
  (role concrete)
  (single-slot source_item
    (type STRING)
;+ (cardinality 0 1)
    (create-accessor read-write)))

(defclass _T_feature_based_pp_dated_item
  (is-a _TYPE)
  (role concrete)
  (single-slot feature_based_pp_dated_item
    (type SYMBOL)
    (allowed-values versioned_action_request directed_action document)
;+ (cardinality 0 1)
    (create-accessor read-write)))

(defclass _T_geometric_set_select
  (is-a _TYPE)
  (role concrete)
  (single-slot geometric_set_select
    (type SYMBOL)
    (allowed-values curve surface point)
;+ (cardinality 0 1)
    (create-accessor read-write)))

(defclass _T_tolerance_method_definition
  (is-a _TYPE)
  (role concrete)
  (single-slot tolerance_method_definition
    (type SYMBOL)
    (allowed-values tolerance_value limits_and_fits)
;+ (cardinality 0 1)
    (create-accessor read-write)))

```

```

(defclass _T_group_item
  (is-a _TYPE)
  (role concrete)
  (single-slot group_item
    (type SYMBOL)
    (allowed-values instanced_feature replicate_feature transition_feature)
    (cardinality 0 1)
    (create-accessor read-write)))

;+

(defclass _T_text
  (is-a _TYPE)
  (role concrete)
  (single-slot text
    (type STRING)
    (cardinality 0 1)
    (create-accessor read-write)))

;+

(defclass _T_id_attribute_select
  (is-a _TYPE)
  (role concrete)
  (single-slot id_attribute_select
    (type SYMBOL)
    (allowed-values representation action application_context)
    (cardinality 0 1)
    (create-accessor read-write)))

;+

(defclass _T_reversible_topology
  (is-a _TYPE)
  (role concrete)
  (single-slot reversible_topology
    (type SYMBOL)
    (allowed-values reversible_topology_item list_of_reversible_topology_item
      set_of_reversible_topology_item)
    (cardinality 0 1)
    (create-accessor read-write)))

;+

(defclass _T_trimming_select
  (is-a _TYPE)
  (role concrete)
  (single-slot trimming_select
    (type SYMBOL)
    (allowed-values cartesian_point parameter_value)
    (cardinality 0 1)
    (create-accessor read-write)))

;+

(defclass _T_descriptive_measure
  (is-a _TYPE)
  (role concrete)
  (single-slot descriptive_measure
    (type STRING)
    (cardinality 0 1)
    (create-accessor read-write)))

;+

(defclass _T_supported_items
  (is-a _TYPE)
  (role concrete)
  (single-slot supported_item
    (type SYMBOL)
    (allowed-values action_directive action action_method)
    (cardinality 0 1)
    (create-accessor read-write)))

;+

(defclass action_assignment

```

```

        (is-a AP224_FILE)
        (role abstract)
        (single-slot assigned_action
          (type INSTANCE)
          (allowed-classes action)
          (cardinality 0 1)
          (create-accessor read-write))
        (single-slot role
          (type INSTANCE)
          (allowed-classes object_role)
          (cardinality 0 1)
          (create-accessor read-write)))

(defclass object_role
  (is-a AP224_FILE)
  (role concrete)
  (single-slot name_
    (comment "label")
    (type INSTANCE)
    (allowed-classes _T_label)
    (cardinality 0 1)
    (create-accessor read-write))
  (single-slot description
    (comment "OPTIONAL text")
    (type INSTANCE)
    (allowed-classes _T_text)
    (cardinality 0 1)
    (create-accessor read-write)))

(defclass bezier_curve
  (is-a b_spline_curve)
  (role concrete))

(defclass point
  (is-a geometric_representation_item)
  (role concrete))

(defclass cartesian_point
  (is-a point)
  (role concrete)
  (multislot coordinates
    (type FLOAT)
    (cardinality 0 3)
    (create-accessor read-write)))

(defclass surface
  (is-a geometric_representation_item)
  (role concrete))

(defclass bounded_surface
  (is-a surface)
  (role concrete))

(defclass b_spline_surface
  (is-a bounded_surface)
  (role concrete)
  (single-slot u_closed
    (type SYMBOL)
    (allowed-values FALSE TRUE)
    (cardinality 0 1)
    (create-accessor read-write))
  (single-slot u_degree
    (type INTEGER)

```



```

;+          (cardinality 0 1)
            (create-accessor read-write))
(single-slot v_degree
  (type INTEGER)
;+          (cardinality 0 1)
            (create-accessor read-write))
(single-slot v_closed
  (type SYMBOL)
  (allowed-values FALSE TRUE)
;+          (cardinality 0 1)
            (create-accessor read-write))
(single-slot self_intersect
  (type SYMBOL)
  (allowed-values FALSE TRUE)
;+          (cardinality 0 1)
            (create-accessor read-write))
(multislot control_points_list
  (type INSTANCE)
;+          (allowed-classes cartesian_point)
            (cardinality 0 2)
            (create-accessor read-write))
(single-slot v_upper
  (type INTEGER)
;+          (cardinality 0 1)
            (create-accessor read-write))
(single-slot u_upper
  (type INTEGER)
;+          (cardinality 0 1)
            (create-accessor read-write))
(single-slot surface_form
  (type INSTANCE)
;+          (allowed-classes _T_b_spline_surface_form)
;+          (cardinality 0 1)
            (create-accessor read-write))
(multislot control_points
  (type FLOAT)
  (create-accessor read-write)))

(defclass b_spline_surface_with_knots
  (is-a b_spline_surface)
  (role concrete)
  (single-slot knot_spec
    (type INSTANCE)
;+          (allowed-classes _T_knot_type)
;+          (cardinality 0 1)
            (create-accessor read-write))
  (multislot v_knots
    (type FLOAT)
    (cardinality 0 2)
    (create-accessor read-write))
  (single-slot knot_v_upper
    (type INTEGER)
;+          (cardinality 0 1)
            (create-accessor read-write))
  (multislot v_multiplicities
    (type INTEGER)
    (cardinality 0 2)
    (create-accessor read-write))
  (single-slot knot_u_upper
    (type INTEGER)
;+          (cardinality 0 1)
            (create-accessor read-write))
  (multislot u_multiplicities
    (type INTEGER)

```

```

                (cardinality 0 2)
                (create-accessor read-write))
(multislot u_knots
  (type FLOAT)
  (cardinality 0 2)
  (create-accessor read-write)))

(defclass bezier_surface
  (is-a b_spline_surface)
  (role concrete))

(defclass quasi_uniform_surface
  (is-a b_spline_surface)
  (role concrete))

(defclass uniform_surface
  (is-a b_spline_surface)
  (role concrete))

(defclass rational_b_spline_surface
  (is-a b_spline_surface)
  (role concrete))

(defclass elementary_surface
  (is-a surface)
  (role concrete))

(defclass toroidal_surface
  (is-a elementary_surface)
  (role concrete))

(defclass degenerate_toroidal_surface
  (is-a toroidal_surface)
  (role concrete))

(defclass cylindrical_surface
  (is-a elementary_surface)
  (role concrete))

(defclass conical_surface
  (is-a elementary_surface)
  (role concrete))

(defclass plane
  (is-a elementary_surface)
  (role concrete))

(defclass spherical_surface
  (is-a elementary_surface)
  (role concrete))

(defclass swept_surface
  (is-a surface)
  (role concrete))

(defclass edge_curve
  (is-a geometric_representation_item)
  (role concrete))

(defclass cartesian_transformation_operator
  (is-a geometric_representation_item)
  (role concrete))

(defclass cartesian_transformation_operator_3d

```

```

        (is-a cartesian_transformation_operator)
        (role concrete))

(defclass solid_model
  (is-a geometric_representation_item)
  (role concrete))

(defclass face_surface
  (is-a face_geometric_representation_item)
  (role concrete))

(defclass advanced_face
  (is-a face_surface)
  (role concrete)
  (multislot hayders_generic_geometry
;+      (comment "can be used for definition of all generic instances")
        (type FLOAT)
        (cardinality 0 10)
        (create-accessor read-write)))

(defclass vector
  (is-a geometric_representation_item)
  (role concrete))

(defclass vertex_point
  (is-a geometric_representation_item)
  (role concrete))

(defclass qualified_representation_item
  (is-a representation_item)
  (role concrete))

(defclass mapped_item
  (is-a representation_item)
  (role concrete))

(defclass measure_representation_item
  (is-a representation_item)
  (role concrete))

(defclass descriptive_representation_item
  (is-a representation_item)
  (role concrete))

(defclass geometric_tolerance
  (is-a AP224_FILE)
  (role abstract))

(defclass geometric_tolerance_with_datum_reference
  (is-a geometric_tolerance)
  (role concrete))

(defclass angularity_tolerance
  (is-a geometric_tolerance_with_datum_reference)
  (role concrete)
  (single-slot hayders_generic_geometry
;+      (comment "can be used for definition of all generic instances")
        (type FLOAT)
        (range %3FVARIABLE 2.9999)
;+      (cardinality 0 1)
        (create-accessor read-write)))

(defclass shape_aspect_relationship
  (is-a AP224_FILE)

```

```

(role abstract))

(defclass document_usage_constraint_assignment
  (is-a AP224_FILE)
  (role abstract))

(defclass applied_document_usage_constraint_assignment
  (is-a document_usage_constraint_assignment)
  (role concrete)
  (single-slot document_reference_item
    (type INSTANCE)
    ;+ (allowed-classes _T_document_reference_item)
    ;+ (cardinality 0 1)
    (create-accessor read-write)))

(defclass group_assignment
  (is-a AP224_FILE)
  (role abstract))

(defclass applied_group_assignment
  (is-a group_assignment)
  (role concrete)
  (single-slot group_item
    (type SYMBOL)
    (allowed-values instanced_feature replicate_feature transition_feature)
    ;+ (cardinality 0 1)
    (create-accessor read-write)))

(defclass approval
  (is-a AP224_FILE)
  (role concrete)
  (single-slot level
    (type STRING)
    ;+ (cardinality 0 1)
    (create-accessor read-write))
  (single-slot status
    ;+ (comment "label")
    (type INSTANCE)
    ;+ (allowed-classes approval_status)
    ;+ (cardinality 0 1)
    (create-accessor read-write)))

(defclass approval_status
  (is-a AP224_FILE)
  (role concrete)
  (single-slot name_
    ;+ (comment "label")
    (type STRING)
    ;+ (cardinality 0 1)
    (create-accessor read-write)))

(defclass approval_assignment
  (is-a AP224_FILE)
  (role abstract)
  (single-slot assigned_approval
    ;+ (comment "approval")
    (type INSTANCE)
    ;+ (allowed-classes approval)
    ;+ (cardinality 0 1)
    (create-accessor read-write))
  (single-slot role
    (type INSTANCE)
    ;+ (allowed-classes object_role)
    ;+ (cardinality 0 1)

```

```

        (create-accessor read-write)))

(defclass approval_date_time
  (is-a AP224_FILE)
  (role concrete)
  (single-slot role
    (type INSTANCE)
    (allowed-classes object_role)
    (cardinality 0 1)
    (create-accessor read-write))
  (single-slot dated_approval
    (type INSTANCE)
    (allowed-classes approval)
    (cardinality 0 1)
    (create-accessor read-write))
  (single-slot date_time
    (type INSTANCE)
    (allowed-classes _T_date_time_select)
    (cardinality 0 1)
    (create-accessor read-write)))

(defclass approval_person_organisation
  (is-a AP224_FILE)
  (role concrete)
  (single-slot person_organisation
    (type INSTANCE)
    (allowed-classes _T_person_organisation_select)
    (cardinality 0 1)
    (create-accessor read-write))
  (single-slot role
    (type INSTANCE)
    (allowed-classes approval_role)
    (cardinality 0 1)
    (create-accessor read-write))
  (single-slot authorized_approval
    (type INSTANCE)
    (allowed-classes approval)
    (cardinality 0 1)
    (create-accessor read-write)))

(defclass description_attribute
  (is-a AP224_FILE)
  (role concrete)
  (single-slot described_item
    (type INSTANCE)
    (allowed-classes _T_description_attribute_select)
    (cardinality 0 1)
    (create-accessor read-write))
  (single-slot attribute_value
    (type STRING)
    (cardinality 0 1)
    (create-accessor read-write)))

(defclass product_definition_relationship
  (is-a AP224_FILE)
  (role abstract))

(defclass assembly_component_usage
  (is-a product_definition_usage)
  (role concrete)
  (single-slot reference_designator
    (type STRING)
    (cardinality 0 1)
    (create-accessor read-write)))

```

```

(defclass advanced_brep_shape_representation
  (is-a shape_representation)
  (role concrete)
  (multislot hayders_generic_geometry
;+      (comment "can be used for definition of all generic instances")
        (type FLOAT)
        (cardinality 0 6)
        (create-accessor read-write)))

(defclass shape_representation_with_parameters
  (is-a shape_representation)
  (role concrete))

(defclass block_shape_representation
  (is-a shape_representation_with_parameters)
  (role concrete)
  (multislot hayders_generic_geometry
;+      (comment "can be used for definition of all generic instances")
        (type FLOAT)
        (cardinality 0 5)
        (create-accessor read-write)))

(defclass characterized_object
  (is-a AP224_FILE)
  (role concrete))

(defclass feature_definition
  (is-a characterized_object)
  (role concrete))

(defclass boss
  (is-a feature_definition)
  (role concrete)
  (multislot hayders_generic_geometry
;+      (comment "can be used for definition of all generic instances")
        (type FLOAT)
        (cardinality 0 12)
        (create-accessor read-write)))

```

❖ Appendix (D) Generic Collaborative ontology, ArvinMeritor test example

As per the STEP ontology above, the generic collaborative ontology is also defined using the KIF like language in Protégé which can be interpreted by agent based systems RDF based data-mining systems and expert systems.

```
(defclass %3ACLIPS_TOP_LEVEL_SLOT_CLASS
  (is-a USER)
  (role concrete)
  (single-slot quantity
    ;+ (comment "quantity of items in BoM")
        (type INTEGER)
        (default 1)
        (cardinality 1 1)
        (create-accessor read-write))
  (single-slot Phase_Evidence
    ;+ (type SYMBOL)
        (allowed-values Fully_Verified None Incomplete)
        (cardinality 0 1)
        (create-accessor read-write))
  (single-slot Phase_plan
    ;+ (type SYMBOL)
        (allowed-values In_plan Being_Planned Not_planned)
        (cardinality 1 1)
        (create-accessor read-write))
  (single-slot requirement_priority
    ;+ (type SYMBOL)
        (allowed-values specification standard)
        (cardinality 0 1)
        (create-accessor read-write))
  (single-slot company_name
    ;+ (comment "name of the supplier or customer organisation")
        (type STRING)
        (cardinality 1 1)
        (create-accessor read-write))
  (multislot BoM_items_used
    ;+ (type INSTANCE)
        (allowed-classes Bill_of_Material_items)
        (create-accessor read-write))
  (single-slot part_no
    ;+ (comment "component part number")
        (type INTEGER)
        (cardinality 0 1)
        (create-accessor read-write))
  (single-slot frequency_of_use_manual_powered
    ;+ (type SYMBOL)
        (allowed-values manual powered)
        (cardinality 0 1)
        (create-accessor read-write))
  (single-slot req_title
    ;+ (type STRING)
        (cardinality 0 1)
        (create-accessor read-write))
  (single-slot Assembly_number
    ;+ (type STRING)
        (cardinality 1 1)
        (create-accessor read-write))
```

```

(single-slot %3ADEFINITION-NAME
  (type STRING)
;+ (cardinality 0 1)
    (create-accessor read-write))
(single-slot nomination_date
  (type INTEGER)
;+ (cardinality 0 1)
    (create-accessor read-write))
(single-slot specification_page_no
  (type STRING)
;+ (cardinality 0 1)
    (create-accessor read-write))
(multislot potential_use
  (type INSTANCE)
;+ (allowed-classes)
    (create-accessor read-write))
(single-slot material_type
;+ (comment "the generic property of the material")
  (type STRING)
;+ (cardinality 1 1)
    (create-accessor read-write))
(single-slot issue_level
;+ (type STRING)
  (cardinality 1 1)
  (create-accessor read-write))
(multislot potential_supplier
  (type INSTANCE)
;+ (allowed-classes suppliers)
    (create-accessor read-write))
(single-slot required_by
;+ (comment "the entity that has stipulated this specific requirement.")
  (type SYMBOL)
  (allowed-values customer corporation Regulations DRLT)
;+ (cardinality 0 1)
    (create-accessor read-write))
(single-slot operation
  (type SYMBOL)
  (allowed-values opening closing unlocking locking superlocking unsuperlocking childsafty
control_ajar_switch control_key_switch control_lock_switch control_release_switch)
;+ (cardinality 1 1)
    (create-accessor read-write))
(single-slot %3ASTARTUP-EXPRESSION
  (type STRING)
;+ (cardinality 0 1)
    (create-accessor read-write))
(multislot timesheet_instance
;+ (comment "instances of a timesheet for personel in a project")
  (type INSTANCE)
;+ (allowed-classes timesheet_template)
    (create-accessor read-write))
(single-slot requirement_version
  (type INTEGER)
;+ (cardinality 0 1)
    (create-accessor read-write))
(single-slot Request_no
  (type INTEGER)
;+ (cardinality 0 1)
    (create-accessor read-write))
(single-slot requirement_no
  (type INTEGER)
;+ (cardinality 0 1)
    (create-accessor read-write))
(single-slot specification_number
  (type STRING)

```



```

;+      (cardinality 0 1)
        (create-accessor read-write))
(single-slot bom_oem_part_no
;+      (comment "optional part number of the customer")
        (type STRING)
;+      (cardinality 0 1)
        (create-accessor read-write))
(single-slot Specification_ref
        (type INSTANCE)
;+      (allowed-classes Specification_reference)
;+      (cardinality 0 1)
        (create-accessor read-write))
(single-slot %3ASTARTUP-FILE
        (type STRING)
;+      (cardinality 0 1)
        (create-accessor read-write))
(single-slot product_desc
        (type STRING)
;+      (cardinality 0 1)
        (create-accessor read-write))
(single-slot bom_reference_no
        (type INTEGER)
;+      (cardinality 1 1)
        (create-accessor read-write))
(single-slot contact_number
;+      (comment "the telephone number of the contact person")
        (type INTEGER)
;+      (cardinality 0 1)
        (create-accessor read-write))
(single-slot BoM_Material
        (type INSTANCE)
;+      (allowed-classes Materials)
;+      (cardinality 0 1)
        (create-accessor read-write))
(single-slot %3AKB-SAVE
        (type SYMBOL)
        (allowed-values FALSE TRUE)
;+      (cardinality 0 1)
        (create-accessor read-write))
(single-slot production_supplier
        (type INSTANCE)
;+      (allowed-classes suppliers)
;+      (cardinality 0 1)
        (create-accessor read-write))
(multislot req_type_instance
        (type INSTANCE)
;+      (allowed-classes req_type)
        (create-accessor read-write))
(single-slot deadline_date
        (type INTEGER)
;+      (cardinality 0 1)
        (create-accessor read-write))
(single-slot requirement_description
        (type STRING)
;+      (cardinality 0 1)
        (create-accessor read-write))
(single-slot description
        (type STRING)
;+      (cardinality 0 1)
        (create-accessor read-write))
(single-slot Assembly_weight
;+      (comment "total weight of the items in the assembly")
        (type INTEGER)
;+      (cardinality 0 1)

```

```

        (create-accessor read-write))
(single-slot year
;+      (comment "the year in which the week number is assigned")
        (type INTEGER)
        (range 2002 2020)
;+      (cardinality 1 1)
        (create-accessor read-write))
(single-slot %3ADEFINITION
;+      (type STRING)
        (cardinality 0 1)
        (create-accessor read-write))
(single-slot apqp_number
;+      (type INTEGER)
        (default 1)
        (cardinality 1 1)
        (create-accessor read-write))
(single-slot Latch_location
;+      (type SYMBOL)
        (allowed-values front-right front-left rear-right rear-left)
        (cardinality 0 1)
        (create-accessor read-write))
(single-slot requirement_source
;+      (comment "should be a link to the origin of the requirement... perhaps a hyperlink of some sort")
        (type STRING)
;+      (cardinality 0 1)
        (create-accessor read-write))
(single-slot issue_date
;+      (type STRING)
        (cardinality 0 1)
        (create-accessor read-write))
(single-slot requirement_owner
;+      (comment "the entity that handles the processing of this requirement")
        (type STRING)
;+      (cardinality 0 1)
        (create-accessor read-write))
(single-slot spec_item
;+      (type STRING)
        (cardinality 0 1)
        (create-accessor read-write))
(single-slot project_name
;+      (comment "the name assigned to the project by default")
        (type STRING)
;+      (cardinality 1 1)
        (create-accessor read-write))
(single-slot week_number
;+      (comment "the week number during the project lifecycle during which tasks were carried out.")
        (type INTEGER)
        (range 1 52)
;+      (cardinality 1 1)
        (create-accessor read-write))
(single-slot verification_method
;+      (type INSTANCE)
        (allowed-classes verification_methods)
;+      (cardinality 0 1)
        (create-accessor read-write))
(single-slot specification_critical
;+      (type SYMBOL)
        (allowed-values FALSE TRUE)
        (cardinality 0 1)
        (create-accessor read-write))
(single-slot BoM_weight
;+      (comment "in grammes")
        (type FLOAT)
;+      (cardinality 0 1)

```

```

        (create-accessor read-write))
(single-slot Phase_Status
  (type SYMBOL)
  (allowed-values On_target Risk Late)
;+
  (cardinality 1 1)
  (create-accessor read-write))
(single-slot required_of
;+
  (comment "the actual components of the system that have to adhere to the set requirement.")
  (type INSTANCE)
;+
  (allowed-classes)
;+
  (cardinality 0 1)
  (create-accessor read-write))
(single-slot type_of_striker
  (type SYMBOL)
  (allowed-values loop pin wedge)
;+
  (cardinality 0 1)
  (create-accessor read-write))
(single-slot Request_date
;+
  (type INTEGER)
  (cardinality 0 1)
  (create-accessor read-write))
(single-slot verification_method_desc
;+
  (type STRING)
  (cardinality 0 1)
  (create-accessor read-write))
(single-slot day_of_week
;+
  (comment "the day on which a task is done")
  (type SYMBOL)
  (allowed-values monday tuesday thursday friday wednesday)
;+
  (cardinality 1 1)
  (create-accessor read-write))
(single-slot specification_value
;+
  (type SYMBOL)
  (allowed-values FALSE TRUE)
  (cardinality 0 1)
  (create-accessor read-write))
(single-slot contact_name
;+
  (comment "the name of the person to contact")
  (type STRING)
;+
  (cardinality 0 1)
  (create-accessor read-write))
(single-slot Phase_item
;+
  (type SYMBOL)
  (cardinality 1 1)
  (create-accessor read-write))
(single-slot bom_part_no
;+
  (type STRING)
  (cardinality 0 1)
  (create-accessor read-write))
(single-slot prototype_supplier
;+
  (type INSTANCE)
  (allowed-classes suppliers)
;+
  (cardinality 0 1)
  (create-accessor read-write))
(single-slot contact_address
;+
  (comment "the address of the contact person")
  (type STRING)
;+
  (cardinality 0 1)
  (create-accessor read-write))
(single-slot role
;+
  (comment "the role of the person in the project")
  (type STRING)
;+
  (cardinality 0 1)
  (create-accessor read-write))

```

```

(single-slot requirement_type
;+      (comment "the reason for stipulating a particular requirement.")
;+      (type STRING)
;+      (cardinality 1 1)
      (create-accessor read-write))
(single-slot req_ver_no_subtype
      (type SYMBOL)
      (allowed-values DL DT DY EC ED EL FS GE IT MA PA SD SO TW)
;+      (cardinality 0 1)
      (create-accessor read-write))
(single-slot verification_no
;+      (comment "unique verification number")
      (type INTEGER)
;+      (cardinality 0 1)
      (create-accessor read-write))
(single-slot Specification_item_value
      (type INSTANCE)
;+      (allowed-classes Specification_items)
;+      (cardinality 0 1)
      (create-accessor read-write))
(multislot req_associated
      (type INSTANCE)
;+      (allowed-classes Requirements)
      (create-accessor read-write))
(single-slot hours_during_day
;+      (comment "the number of hours spent on project during the day")
      (type INTEGER)
      (range 1 8)
      (default 1)
;+      (cardinality 1 1)
      (create-accessor read-write))
(multislot Customer
;+      (comment "the customer name")
      (type INSTANCE)
;+      (allowed-classes OEM)
      (create-accessor read-write))
(multislot latch_type
      (type INSTANCE)
;+      (allowed-classes latch_types)
      (create-accessor read-write)))

(defclass %3AJESS-RULE
  (is-a %3AJESS-DEFINITION %3ARULE)
  (role concrete))

(defclass Specifications
  (is-a USER)
  (role abstract))

(defclass safety
  (is-a Specifications)
  (role abstract)
  (single-slot specification_value
    (type SYMBOL)
    (allowed-values FALSE TRUE)
;+    (cardinality 0 1)
    (create-accessor read-write))
  (single-slot specification_critical
    (type SYMBOL)
    (allowed-values FALSE TRUE)
;+    (cardinality 0 1)
    (create-accessor read-write))
  (single-slot Specification_item_value

```

```

                (type INSTANCE)
;+                (allowed-classes Specification_items)
;+                (cardinality 0 1)
                (create-accessor read-write))
        (single-slot Specification_ref
                (type INSTANCE)
;+                (allowed-classes Specification_reference)
;+                (cardinality 0 1)
                (create-accessor read-write)))

(defclass failsafe_operation
  (is-a safety)
  (role abstract))

(defclass power_loading
  (is-a failsafe_operation)
  (role concrete))

(defclass abuse_loading
  (is-a failsafe_operation)
  (role concrete))

(defclass environmental_abuse
  (is-a failsafe_operation)
  (role concrete))

(defclass Crashworthiness
  (is-a safety)
  (role concrete))

(defclass vehicle_dynamics
  (is-a safety)
  (role concrete))

(defclass mis-assembly
  (is-a safety)
  (role concrete))

(defclass personal
  (is-a safety)
  (role concrete))

(defclass product
  (is-a safety)
  (role concrete))

(defclass security
  (is-a Specifications)
  (role abstract)
  (single-slot specification_value
    (type SYMBOL)
    (allowed-values FALSE TRUE)
;+    (cardinality 0 1)
    (create-accessor read-write))
  (single-slot specification_critical
    (type SYMBOL)
    (allowed-values FALSE TRUE)
;+    (cardinality 0 1)
    (create-accessor read-write))
  (single-slot Specification_item_value
    (type INSTANCE)
;+    (allowed-classes Specification_items)
;+    (cardinality 0 1)
    (create-accessor read-write)))

```

```

        (single-slot Specification_ref
          (type INSTANCE)
          (allowed-classes Specification_reference)
          (cardinality 0 1)
          (create-accessor read-write)))

(defclass unauthorised_entry
  (is-a security)
  (role concrete))

(defclass percieved
  (is-a security)
  (role concrete))

(defclass reliability
  (is-a Specifications)
  (role abstract)
  (single-slot specification_value
    (type SYMBOL)
    (allowed-values FALSE TRUE)
    (cardinality 0 1)
    (create-accessor read-write))
  (single-slot specification_critical
    (type SYMBOL)
    (allowed-values FALSE TRUE)
    (cardinality 0 1)
    (create-accessor read-write))
  (single-slot Specification_item_value
    (type INSTANCE)
    (allowed-classes Specification_items)
    (cardinality 0 1)
    (create-accessor read-write))
  (single-slot Specification_ref
    (type INSTANCE)
    (allowed-classes Specification_reference)
    (cardinality 0 1)
    (create-accessor read-write)))

(defclass control
  (is-a reliability)
  (role abstract))

(defclass ajar_switch
  (is-a control)
  (role concrete))

(defclass key_switch
  (is-a control)
  (role concrete))

(defclass lock_switch
  (is-a control)
  (role concrete))

(defclass release_switch
  (is-a control)
  (role concrete))

(defclass system_integration
  (is-a Specifications)
  (role abstract)
  (single-slot specification_value
    (type SYMBOL)
    (allowed-values FALSE TRUE))

```

```

;+          (cardinality 0 1)
            (create-accessor read-write))
  (single-slot specification_critical
    (type SYMBOL)
    (allowed-values FALSE TRUE)
;+          (cardinality 0 1)
            (create-accessor read-write))
  (single-slot Specification_item_value
    (type INSTANCE)
;+          (allowed-classes Specification_items)
;+          (cardinality 0 1)
            (create-accessor read-write))
  (single-slot Specification_ref
    (type INSTANCE)
;+          (allowed-classes Specification_reference)
;+          (cardinality 0 1)
            (create-accessor read-write)))

(defclass weight
  (is-a system_integration)
  (role concrete))

(defclass packaging
  (is-a system_integration)
  (role concrete))

(defclass service_and_repair
  (is-a system_integration)
  (role concrete))

(defclass sensitivity
  (is-a system_integration)
  (role concrete))

(defclass striker_mis_alignment
  (is-a sensitivity)
  (role concrete))

(defclass noise_specification "note this is different to noise testing"
  (is-a Specifications)
  (role abstract)
  (single-slot specification_value
    (type SYMBOL)
    (allowed-values FALSE TRUE)
;+          (cardinality 0 1)
            (create-accessor read-write))
  (single-slot specification_critical
    (type SYMBOL)
    (allowed-values FALSE TRUE)
;+          (cardinality 0 1)
            (create-accessor read-write))
  (single-slot Specification_item_value
    (type INSTANCE)
;+          (allowed-classes Specification_items)
;+          (cardinality 0 1)
            (create-accessor read-write))
  (single-slot Specification_ref
    (type INSTANCE)
;+          (allowed-classes Specification_reference)
;+          (cardinality 0 1)
            (create-accessor read-write)))

(defclass on_road
  (is-a noise_specification))

```

```

(role concrete))

(defclass aesthetics_specification
  (is-a Specifications)
  (role abstract)
  (single-slot specification_value
    (type SYMBOL)
    (allowed-values FALSE TRUE)
;+    (cardinality 0 1)
    (create-accessor read-write))
  (single-slot specification_critical
    (type SYMBOL)
    (allowed-values FALSE TRUE)
;+    (cardinality 0 1)
    (create-accessor read-write))
  (single-slot Specification_item_value
    (type INSTANCE)
;+    (allowed-classes Specification_items)
;+    (cardinality 0 1)
    (create-accessor read-write))
  (single-slot Specification_ref
    (type INSTANCE)
;+    (allowed-classes Specification_reference)
;+    (cardinality 0 1)
    (create-accessor read-write)))

(defclass interior_styling
  (is-a aesthetics_specification)
  (role abstract))

(defclass exterior_styling
  (is-a aesthetics_specification)
  (role abstract))

(defclass striker
  (is-a aesthetics_specification)
  (role abstract))

(defclass throat
  (is-a aesthetics_specification)
  (role abstract))

(defclass Specification_reference "Contains the specification number issue date and page no"
  (is-a Specifications)
  (role concrete)
  (single-slot specification_page_no
    (type STRING)
;+    (cardinality 0 1)
    (create-accessor read-write))
  (single-slot issue_date
    (type STRING)
;+    (cardinality 0 1)
    (create-accessor read-write))
  (single-slot specification_number
    (type STRING)
;+    (cardinality 0 1)
    (create-accessor read-write)))

(defclass ease_of_operation
  (is-a Specifications)
  (role abstract)
  (single-slot specification_value
    (type SYMBOL)
    (allowed-values FALSE TRUE))

```



```

;+          (cardinality 0 1)
            (create-accessor read-write))
  (single-slot specification_critical
    (type SYMBOL)
    (allowed-values FALSE TRUE)
;+          (cardinality 0 1)
            (create-accessor read-write))
  (single-slot Specification_item_value
    (type INSTANCE)
;+          (allowed-classes Specification_items)
;+          (cardinality 0 1)
            (create-accessor read-write))
  (single-slot Specification_ref
    (type INSTANCE)
;+          (allowed-classes Specification_reference)
;+          (cardinality 0 1)
            (create-accessor read-write)))

(defclass opening
  (is-a reliability noise_specification ease_of_operation)
  (role concrete))

(defclass closing
  (is-a reliability noise_specification ease_of_operation)
  (role concrete))

(defclass unlocking
  (is-a reliability noise_specification ease_of_operation)
  (role concrete))

(defclass locking
  (is-a reliability noise_specification ease_of_operation)
  (role concrete))

(defclass superlocking
  (is-a reliability noise_specification ease_of_operation)
  (role concrete))

(defclass unsuperlocking
  (is-a reliability noise_specification ease_of_operation)
  (role concrete))

(defclass childsafety
  (is-a reliability noise_specification aesthetics_specification ease_of_operation)
  (role concrete))

(defclass intuitive_operation
  (is-a Specifications)
  (role abstract)
  (single-slot specification_value
    (type SYMBOL)
    (allowed-values FALSE TRUE)
;+          (cardinality 0 1)
            (create-accessor read-write))
  (single-slot specification_critical
    (type SYMBOL)
    (allowed-values FALSE TRUE)
;+          (cardinality 0 1)
            (create-accessor read-write))
  (single-slot Specification_item_value
    (type INSTANCE)
;+          (allowed-classes Specification_items)
;+          (cardinality 0 1)
            (create-accessor read-write)))

```

```

        (single-slot Specification_ref
          (type INSTANCE)
          (allowed-classes Specification_reference)
          (cardinality 0 1)
          (create-accessor read-write)))

(defclass opening_intuitive
  (is-a intuitive_operation)
  (role concrete))

(defclass unlocking_intuitive
  (is-a intuitive_operation)
  (role concrete))

(defclass locking_intuitive
  (is-a intuitive_operation)
  (role concrete))

(defclass superlocking_intuitive
  (is-a intuitive_operation)
  (role concrete))

(defclass unsuperlocking_intuitive
  (is-a intuitive_operation)
  (role concrete))

(defclass child_safety_intuitive
  (is-a intuitive_operation)
  (role concrete))

(defclass lock_status_intuitive
  (is-a intuitive_operation)
  (role concrete))

(defclass Specification_items
  (is-a Specifications)
  (role concrete)
  (single-slot spec_item
    (type STRING)
    (cardinality 0 1)
    (create-accessor read-write)))

(defclass Workflows "The definition of workflow items in Arvin Meritor's C2C processes"
  (is-a USER)
  (role abstract)
  (single-slot Phase_Evidence
    (type SYMBOL)
    (allowed-values Fully_Verified None Incomplete)
    (cardinality 0 1)
    (create-accessor read-write))
  (single-slot Phase_item
    (type SYMBOL)
    (cardinality 1 1)
    (create-accessor read-write))
  (single-slot Phase_plan
    (type SYMBOL)
    (allowed-values In_plan Being_Planned Not_planned)
    (cardinality 1 1)
    (create-accessor read-write))
  (single-slot Phase_Status
    (type SYMBOL)
    (allowed-values On_target Risk Late)
    (cardinality 1 1)
    (create-accessor read-write)))

```

```

(defclass Phase_1 "concept design and definition"
  (is-a Workflows)
  (role abstract))

(defclass 10_Plan_project
  (is-a Phase_1)
  (role concrete)
  (single-slot Phase_item
    (type SYMBOL)
    (allowed-values initiate_project_action_plan Define_project_record_retention_system
Define_project_activities_and_outputs Establish_project_cost_tracking_method Develop_communication_plan
Develop_performance_metrics Establish_engineering_parts_and_test%2Fanalysis_cost
develop_design_reviews_requirements_and_expectations Establish_preliminary_baseline_timeline
Develop_project_team_meeting_requirements_and_expectations
Develop_management_review_requirements_and_expectations Establish_project_change_control_practice)
;+ (cardinality 1 1)
    (create-accessor read-write)))

(defclass 14.3_Develop_preliminary_AR
  (is-a Phase_1)
  (role concrete)
  (single-slot Phase_item
    (type SYMBOL)
    (allowed-values Establish_project_budget Refine_project_feasibility)
;+ (cardinality 1 1)
    (create-accessor read-write)))

(defclass 2_Product_concept_engineering_documentation
  (is-a Phase_1)
  (role concrete)
  (single-slot Phase_item
    (type SYMBOL)
    (allowed-values 2_Obtain_vehicle_packaging_requirements
1_refine_product_performance_requirements_and_select_product_concepts
3_Develop_product_concept_CAD_model 9_Develop_Reliability_Targets
6_Develop_product_concept_specifications_EBoM 4_Develop_preliminary_system_concept_DFMEA
7_Develop_Intellectual_Property_Rights_plan 5_Identify_preliminary_special_product_characteristics)
;+ (cardinality 1 1)
    (create-accessor read-write)))

(defclass 3.6_Perform_functional_concept_review
  (is-a Phase_1)
  (role concrete)
  (single-slot Phase_item
    (type SYMBOL)
    (allowed-values Perform_functional_concept_review)
;+ (cardinality 1 1)
    (create-accessor read-write)))

(defclass 3_Verify_product_concept
  (is-a Phase_1)
  (role concrete)
  (single-slot Phase_item
    (type SYMBOL)
    (allowed-values 3_Perform_initial_DfM_analysis 1.05.08_Evaluate_product_feasibility
4_Test_product_concepts 5_Assess_design_alternatives 1_Perform_critical_design_analysis
2_Verify_key_tolerances 1.05.04_Develop_preliminary_DVP)
;+ (cardinality 1 1)
    (create-accessor read-write)))

(defclass 4.04_Develop_process_capability_analysis_of_product_design_classification
  (is-a Phase_1)
  (role concrete)

```

```

        (single-slot Phase_item
          (type SYMBOL)
          (allowed-values Analyse_classified_prints)
;+      (cardinality 1 1)
          (create-accessor read-write)))

(defclass 4_Develop_preliminary_industrialisation_plan
  (is-a Phase_1)
  (role concrete)
  (single-slot Phase_item
    (type SYMBOL)
    (allowed-values
      1_Refine_functional_process_flow
      2_Develop_preliminary_PFMEA
      3_Identify_preliminary_special_process_characteristics
      5_Develop_process_needs_checklist 9_Refine_product_manufacturing_cost_estimates)
;+    (cardinality 1 1)
    (create-accessor read-write)))

(defclass 5.07_Functional_design_concept_review
  (is-a Phase_1)
  (role concrete)
  (single-slot Phase_item
    (type SYMBOL)
    (allowed-values Team_meeting_for_concurrence_of_functional_design)
;+    (cardinality 1 1)
    (create-accessor read-write)))

(defclass 5.09_Determine_project_go_no_go_recommendation
  (is-a Phase_1)
  (role concrete)
  (single-slot Phase_item
    (type SYMBOL)
    (allowed-values
      2_Update_product_marketing_strategy
      5_Refine_project_costs_and_assumptions 1_Refine_project_scope_statement)
;+    (cardinality 1 1)
    (create-accessor read-write)))

(defclass 6_Obtain_project_authorisation
  (is-a Phase_1)
  (role concrete)
  (single-slot Phase_item
    (type SYMBOL)
    (allowed-values Management_project_continuation_received)
;+    (cardinality 1 1)
    (create-accessor read-write)))

(defclass 7_Customer_information
  (is-a Phase_1)
  (role concrete)
  (single-slot Phase_item
    (type SYMBOL)
    (allowed-values
      Refine_system_DFMEA_and_vehicle_assumptions
      Confirm_government_regulations Obtain_customer_inputs_and_expectations)
;+    (cardinality 1 1)
    (create-accessor read-write)))

(defclass Phase_2 "engineering design of concept"
  (is-a Workflows)
  (role abstract))

(defclass Phase_3 "Testing and analysis of final design, set up of manufacturing plant"
  (is-a Workflows)
  (role abstract))

(defclass Phase_4 "This phase is for mass production only and has no workflow or gate checklist defined"

```

```

(is-a Workflows)
(role abstract))

(defclass Phase_5 "decommissioning phase"
  (is-a Workflows)
  (role concrete))

(defclass Bill_of_Materials
  (is-a USER)
  (role concrete))

(defclass Bill_of_Material_items
  (is-a Bill_of_Materials)
  (role concrete)
  (single-slot quantity
    ;+      (comment "quantity of items in BoM")
            (type INTEGER)
            (default 1)
    ;+      (cardinality 1 1)
            (create-accessor read-write))
  (single-slot bom_part_no
    ;+      (type STRING)
            (cardinality 0 1)
            (create-accessor read-write))
  (single-slot prototype_supplier
    ;+      (type INSTANCE)
            (allowed-classes suppliers)
    ;+      (cardinality 0 1)
            (create-accessor read-write))
  (single-slot bom_oem_part_no
    ;+      (comment "optional part number of the customer")
            (type STRING)
    ;+      (cardinality 0 1)
            (create-accessor read-write))
  (single-slot product_desc
    ;+      (type STRING)
            (cardinality 0 1)
            (create-accessor read-write))
  (single-slot bom_reference_no
    ;+      (type INTEGER)
            (cardinality 1 1)
            (create-accessor read-write))
  (single-slot BoM_Material
    ;+      (type INSTANCE)
            (allowed-classes Materials)
    ;+      (cardinality 0 1)
            (create-accessor read-write))
  (single-slot production_supplier
    ;+      (type INSTANCE)
            (allowed-classes suppliers)
    ;+      (cardinality 0 1)
            (create-accessor read-write))
  (single-slot BoM_weight
    ;+      (comment "in grammes")
            (type FLOAT)
    ;+      (cardinality 0 1)
            (create-accessor read-write))
  (single-slot issue_level
    ;+      (type STRING)
            (cardinality 1 1)
            (create-accessor read-write))
  (single-slot Latch_location
    (type SYMBOL)
    (allowed-values front-right front-left rear-right rear-left RIGHT LEFT))

```

```

;+          (cardinality 0 1)
            (create-accessor read-write))
(multislot Customer
;+          (comment "the customer name")
            (type INSTANCE)
;+          (allowed-classes OEM)
            (create-accessor read-write))
(multislot latch_type
;+          (type INSTANCE)
            (allowed-classes latch_types)
            (create-accessor read-write)))

(defclass orders
  (is-a Bill_of_Material_items)
  (role concrete))

(defclass cad_control
  (is-a Bill_of_Material_items)
  (role concrete))

(defclass Bill_of_Material_Assembly "consists of an aggregate of BoM items."
  (is-a Bill_of_Materials)
  (role concrete)
  (single-slot Assembly_weight
;+          (comment "total weight of the items in the assembly")
            (type INTEGER)
;+          (cardinality 0 1)
            (create-accessor read-write))
  (multislot BoM_items_used
;+          (type INSTANCE)
            (allowed-classes Bill_of_Material_items)
            (create-accessor read-write))
  (single-slot Latch_location
;+          (type SYMBOL)
            (allowed-values front-right front-left rear-right rear-left)
            (cardinality 0 1)
            (create-accessor read-write))
  (single-slot Assembly_number
;+          (type STRING)
            (cardinality 1 1)
            (create-accessor read-write))
  (multislot latch_type
;+          (type INSTANCE)
            (allowed-classes latch_types)
            (create-accessor read-write)))

(defclass latch_types
  (is-a Bill_of_Materials)
  (role concrete)
  (single-slot product_desc
;+          (type STRING)
            (cardinality 0 1)
            (create-accessor read-write)))

(defclass Materials "the types of materials used in production, associated properties, what items can be manufactured
with them, and also potential suppliers."
  (is-a Bill_of_Materials)
  (role concrete)
  (multislot potential_use
;+          (type INSTANCE)
            (allowed-classes)
            (create-accessor read-write))
  (single-slot material_type
;+          (comment "the generic property of the material")

```

```

                (type STRING)
;+                (cardinality 1 1)
                (create-accessor read-write))
        (multislot potential_supplier
                (type INSTANCE)
;+                (allowed-classes suppliers)
                (create-accessor read-write)))

(defclass Finish
  (is-a Materials)
  (role concrete))

(defclass rfq_header_info
  (is-a USER)
  (role concrete)
  (single-slot deadline_date
    (type INTEGER)
;+    (cardinality 0 1)
    (create-accessor read-write))
  (single-slot Request_no
    (type INTEGER)
;+    (cardinality 0 1)
    (create-accessor read-write))
  (single-slot part_no
;+    (comment "component part number")
    (type INTEGER)
;+    (cardinality 0 1)
    (create-accessor read-write))
  (single-slot product_desc
    (type STRING)
;+    (cardinality 0 1)
    (create-accessor read-write))
  (single-slot nomination_date
    (type INTEGER)
;+    (cardinality 0 1)
    (create-accessor read-write))
  (single-slot project_name
;+    (comment "the name assigned to the project by default")
    (type STRING)
;+    (cardinality 1 1)
    (create-accessor read-write))
  (single-slot Request_date
    (type INTEGER)
;+    (cardinality 0 1)
    (create-accessor read-write)))

(defclass personnel "abstract class for all personnel data"
  (is-a USER)
  (role abstract)
  (single-slot contact_name
;+    (comment "the name of the person to contact")
    (type STRING)
;+    (cardinality 0 1)
    (create-accessor read-write))
  (single-slot contact_address
;+    (comment "the address of the contact person")
    (type STRING)
;+    (cardinality 0 1)
    (create-accessor read-write))
  (single-slot role
;+    (comment "the role of the person in the project")
    (type STRING)
;+    (cardinality 0 1)
    (create-accessor read-write))

```

```

        (single-slot contact_number
;+          (comment "the telephone number of the contact person")
            (type INTEGER)
;+          (cardinality 0 1)
            (create-accessor read-write))
        (multislot timesheet_instance
;+          (comment "instances of a timesheet for personnel in a project")
            (type INSTANCE)
;+          (allowed-classes timesheet_template)
            (create-accessor read-write)))

(defclass advanced_engineering "concept design personnel"
  (is-a personnel)
  (role concrete))

(defclass client "client personnel involved in project"
  (is-a personnel)
  (role concrete))

(defclass engineering "embodiment design and testing personnel"
  (is-a personnel)
  (role concrete))

(defclass manufacturing "personnel liaising with manufacturing suppliers and design of assembly jigs"
  (is-a personnel)
  (role concrete))

(defclass marketing "marketing and pre-sales personnel"
  (is-a personnel)
  (role concrete))

(defclass subcontractor "the subcontractor personnel involved in project"
  (is-a personnel)
  (role concrete))

(defclass testing_legislation "the testing and legislation personnel"
  (is-a personnel)
  (role concrete))

(defclass timesheet_template "the template for managing timesheets"
  (is-a personnel)
  (role concrete)
  (single-slot year
;+          (comment "the year in which the week number is assigned")
            (type INTEGER)
            (range 2002 2020)
;+          (cardinality 1 1)
            (create-accessor read-write))
  (single-slot hours_during_day
;+          (comment "the number of hours spent on project during the day")
            (type INTEGER)
            (range 1 8)
            (default 1)
;+          (cardinality 1 1)
            (create-accessor read-write))
  (single-slot day_of_week
;+          (comment "the day on which a task is done")
            (type SYMBOL)
            (allowed-values monday tuesday thursday friday wednesday)
;+          (cardinality 1 1)
            (create-accessor read-write))
  (single-slot week_number
;+          (comment "the week number during the project lifecycle during which tasks were carried out.")
            (type INTEGER)

```



```

      (range 1 52)
;+      (cardinality 1 1)
      (create-accessor read-write)))

(defclass Customer_supplier_mgmt "subclasses used to manage all customer and supplier related information"
  (is-a USER)
  (role abstract)
  (single-slot contact_name
;+      (comment "the name of the person to contact")
      (type STRING)
;+      (cardinality 0 1)
      (create-accessor read-write))
  (single-slot company_name
;+      (comment "name of the supplier or customer organisation")
      (type STRING)
;+      (cardinality 1 1)
      (create-accessor read-write))
  (single-slot contact_address
;+      (comment "the address of the contact person")
      (type STRING)
;+      (cardinality 0 1)
      (create-accessor read-write))
  (single-slot contact_number
;+      (comment "the telephone number of the contact person")
      (type INTEGER)
;+      (cardinality 0 1)
      (create-accessor read-write)))

(defclass suppliers "at client or supplier company"
  (is-a Customer_supplier_mgmt)
  (role concrete))

(defclass OEM "details of OEMs working with arvin, they are normally customers"
  (is-a Customer_supplier_mgmt)
  (role concrete))

(defclass unresolved
  (is-a USER)
  (role concrete))

(defclass Tolerance_study "list of issues with tolerances in the design"
  (is-a unresolved)
  (role concrete))

(defclass Statement_of_work "Commercial info, Quality, Teams selection, Location, CAD system / standard,
Delivery / logistics details, Warranty, Payment"
  (is-a unresolved)
  (role abstract))

(defclass ProtoDesControl "design and production of prototype documents inc changes, requests, supplier data"
  (is-a unresolved)
  (role concrete))

(defclass prototype_build_log
  (is-a ProtoDesControl)
  (role concrete))

(defclass prototype_builds
  (is-a ProtoDesControl)
  (role concrete))

(defclass PCN_and_feasibility "Directions containing problems and feasibility requests to-from customer and
suppliers"
  (is-a unresolved))

```

```

(role concrete))

(defclass PCN_approved "here are the lists of Engineering changes to go ahead with"
  (is-a PCN_and_feasibility)
  (role concrete))

(defclass PCN_pending
  (is-a PCN_and_feasibility)
  (role concrete))

(defclass feasibility_request
  (is-a PCN_and_feasibility)
  (role concrete))

(defclass returned_from_customer
  (is-a PCN_and_feasibility)
  (role concrete))

(defclass Open_issues "problems to be solved"
  (is-a unresolved)
  (role concrete))

(defclass customer
  (is-a Open_issues)
  (role concrete))

(defclass internal
  (is-a Open_issues)
  (role concrete))

(defclass supplier_open_issues
  (is-a Open_issues)
  (role concrete))

(defclass Testing
  (is-a unresolved)
  (role abstract))

(defclass DVP%26R "sub-modle test results"
  (is-a Testing)
  (role concrete))

(defclass fire_experiment
  (is-a Testing)
  (role concrete))

(defclass frg "testing and quality control documents"
  (is-a Testing)
  (role concrete))

(defclass heat_test
  (is-a Testing)
  (role concrete))

(defclass noise
  (is-a Testing)
  (role abstract))

(defclass latch_noisy_component_pictures
  (is-a noise)
  (role concrete))

(defclass modification_request
  (is-a noise)

```

```

        (role concrete))

(defclass prototype_orders
  (is-a noise)
  (role concrete))

(defclass OEM_specific
  (is-a unresolved)
  (role abstract))

(defclass ford
  (is-a OEM_specific)
  (role abstract))

(defclass drawings
  (is-a ford)
  (role concrete))

(defclass ford_testing_info
  (is-a ford)
  (role concrete))

(defclass presentations_work_methods
  (is-a ford)
  (role concrete))

(defclass statement_of_work
  (is-a ford)
  (role concrete))

(defclass Minutes_of_meetings
  (is-a unresolved)
  (role concrete))

(defclass PDP_minutes "Minutes of meetings of the product development project"
  (is-a Minutes_of_meetings)
  (role concrete))

(defclass Manufacturing
  (is-a unresolved)
  (role concrete))

(defclass control_plans
  (is-a Manufacturing)
  (role concrete))

(defclass customer_standard_operations_list
  (is-a Manufacturing)
  (role concrete))

(defclass dimensional_reports
  (is-a Manufacturing)
  (role concrete))

(defclass in-house_work_files "not to be released to supplier or customer"
  (is-a Manufacturing)
  (role concrete))

(defclass manufacturing_drawings "images"
  (is-a Manufacturing)
  (role concrete))

(defclass split_FMEA "contains separated FMEA analysis"
  (is-a Manufacturing)

```

```

(role concrete))

(defclass trial_build_reports
  (is-a Manufacturing)
  (role concrete))

(defclass Lifecycle
  (is-a unresolved)
  (role concrete))

(defclass Feasibility_study
  (is-a unresolved)
  (role concrete))

(defclass design_review "geometry details and review"
  (is-a Feasibility_study)
  (role concrete))

(defclass manufacturing_risk "potential problems with latch in production"
  (is-a Feasibility_study)
  (role concrete))

(defclass FSS "Full Service supplier confirmation document"
  (is-a unresolved)
  (role concrete))

(defclass FMEA_and_robustness "Failure mode effect analysis testing requirements and results"
  (is-a unresolved)
  (role concrete))

(defclass FMEA_analyses
  (is-a FMEA_and_robustness)
  (role concrete))

(defclass supplier_specific_tests
  (is-a FMEA_analyses)
  (role concrete))

(defclass comparing_prevention
  (is-a FMEA_and_robustness)
  (role concrete))

(defclass customer_specific_instructions
  (is-a FMEA_and_robustness)
  (role concrete))

(defclass project_specific_analysis
  (is-a FMEA_and_robustness)
  (role concrete))

(defclass Diagnostic
  (is-a unresolved)
  (role concrete))

(defclass Design_calculations
  (is-a unresolved)
  (role concrete))

(defclass CAD
  (is-a unresolved)
  (role concrete))

(defclass archive
  (is-a CAD))

```

```

(role concrete))

(defclass configuration_specific
  (is-a CAD)
  (role concrete))

(defclass master_CAD_data_forms
  (is-a CAD)
  (role concrete))

(defclass released
  (is-a CAD)
  (role abstract))

(defclass designs_for_supplier
  (is-a released)
  (role concrete))

(defclass request
  (is-a CAD)
  (role concrete))

(defclass Budgets
  (is-a unresolved)
  (role concrete))

(defclass target_agreement_document "financial info on costing"
  (is-a Budgets)
  (role concrete))

(defclass Action_plan
  (is-a unresolved)
  (role concrete))

(defclass APQP "Advanced Product Quality Planning. This is structured method for defining & evaluating the actions
necessary to ensure a product that satisfies the customers. APQP is supplier led and is required for all system sub-
system and component manufacturing locations"
  (is-a Action_plan)
  (role concrete)
  (single-slot apqp_number
    (type INTEGER)
    (default 1)
    (cardinality 1 1)
    (create-accessor read-write)))
;+

(defclass Requirement "for specifying the product requirements at phase 0"
  (is-a USER)
  (role abstract))

(defclass Requirements "details of product requirements from the OEM."
  (is-a Requirement)
  (role concrete)
  (single-slot requirement_version
    (type INTEGER)
    (cardinality 0 1)
    (create-accessor read-write))
;+
  (single-slot requirement_priority
    (type SYMBOL)
    (allowed-values specification standard)
    (cardinality 0 1)
    (create-accessor read-write))
;+
  (single-slot requirement_no
    (type INTEGER)
    (cardinality 0 1)
    (create-accessor read-write))
;+

```

```

        (create-accessor read-write))
(single-slot req_title
  (type STRING)
;+    (cardinality 0 1)
        (create-accessor read-write))
(single-slot requirement_owner
;+    (comment "the entity that handles the processing of this requirement")
        (type STRING)
;+    (cardinality 0 1)
        (create-accessor read-write))
(multislot req_type_instance
;+    (type INSTANCE)
        (allowed-classes req_type)
        (create-accessor read-write))
(single-slot req_ver_no_subtype
;+    (type SYMBOL)
        (allowed-values DL DT DY EC ED EL FS GE IT MA PA SD SO TW)
        (cardinality 0 1)
        (create-accessor read-write))
(single-slot requirement_description
;+    (type STRING)
        (cardinality 0 1)
        (create-accessor read-write))
(single-slot verification_method
;+    (type INSTANCE)
;+    (allowed-classes verification_methods)
;+    (cardinality 0 1)
        (create-accessor read-write))
(single-slot required_by
;+    (comment "the entity that has stipulated this specific requirement.")
        (type SYMBOL)
        (allowed-values customer corporation Regulations DRLT)
;+    (cardinality 0 1)
        (create-accessor read-write))
(single-slot required_of
;+    (comment "the actual components of the system that have to adhere to the set requirement.")
        (type INSTANCE)
;+    (allowed-classes)
;+    (cardinality 0 1)
        (create-accessor read-write))
(single-slot requirement_source
;+    (comment "should be a link to the origin of the requirement... perhaps a hyperlink of some sort")
        (type STRING)
;+    (cardinality 0 1)
        (create-accessor read-write))
(multislot req_associated
;+    (type INSTANCE)
        (allowed-classes Requirements)
        (create-accessor read-write)))

(defclass req_type "list of the various requirement types"
  (is-a Requirement)
  (role concrete)
  (single-slot requirement_type
;+    (comment "the reason for stipulating a particular requirement.")
        (type STRING)
;+    (cardinality 1 1)
        (create-accessor read-write)))

(defclass verification_methods "requirements verification methods DVM"
  (is-a Requirement)
  (role concrete)
  (single-slot req_ver_no_subtype
    (type SYMBOL))

```

❖ Appendix (E) Generic Collaborative ontology, test data

The items below are the test objects instantiated from the knowledge base. In this example ArvinMeritor company's Ford Motor company project was used as a test example. The following 90 pages are simply a definition of all the test objects used for the example described in the main text. A number of Bills of materials and a series of existing components are defined within the system, linked together with the full lifecycle information and STEP defined entities. There are two types of objects within the collaborative knowledge base 'knowledge' and 'rules'. The knowledge objects are defined below. Please see page 179 for the JESS rule definitions and instantiations.

To better understand the code, here is a simple breakdown of the syntax. The [Arvin_XXXXX] defines the unique object number (allowing versioning of the objects and maintaining easy synchronisation for the knowledge base. Following this 'of' defines what class the object is instantiated from e.g. in example below Bill_of_Material_item is the name of the class which object Arvin_01961 is instantiated from. Following this, every item is contained in a () separated by a carriage return. Each () contains a slot and its value or values (for slots which allow multiple values per object). The structure is simple (SLOT value), e.g. in the example below (bom_part_no "FA11B00R") bom_part_no is the slot name and its value is FA11B00R. Each object is separated from the other by two carriage returns.

```
([arvin_00023] of advanced_engineering
  (contact_name "Gurbinder Kalsi")
  (contact_address "birmingham")
  (role "concept design engineer")
  (contact_number 1234111104)
  (timesheet_instance [arvin_00107]))

([arvin_00056] of %3APAL-CONSTRAINT)

([arvin_00097] of %3APAL-CONSTRAINT)

([arvin_00101] of timesheet_template
  (year 2003)
  (hours_during_day 3)
  (day_of_week tuesday)
  (week_number 34))

([arvin_00102] of %3APAL-CONSTRAINT

                                (%3APAL-DESCRIPTION "constrains the
                                selection of a single weekday for a particular
                                timesheet")
                                (%3APAL-NAME "week_day_constraint")
                                (%3APAL-STATEMENT "(Predicate)")
                                (%3APAL-RANGE
                                "monday\tuesday\twednesday\tthursday\tfriday"))

([arvin_00107] of timesheet_template
  (year 2002)
  (hours_during_day 1)
  (day_of_week thursday)
  (week_number 23))

([arvin_00115] of %3APAL-CONSTRAINT)

([arvin_00117] of APQP
  (apqp_number 7887))
```

([arvin_00155] of APQP
 (apqp_number 2))

([arvin_00156] of client
 (timesheet_instance [arvin_00157]))

([arvin_00157] of timesheet_template
 (hours_during_day 1)
 (day_of_week thursday))

([arvin_00181] of 10_Plan_project
 (Phase_Evidence Fully_Verified)
 (Phase_item
 Develop_project_team_meeting_requirements_and_e
 xpectations)
 (Phase_plan In_plan)
 (Phase_Status Late))

([arvin_00182] of 14.3_Develop_preliminary_AR
 (Phase_Evidence Incomplete)
 (Phase_item Establish_project_budget)
 (Phase_plan Not_planned)
 (Phase_Status On_target))

([arvin_00183] of 7_Customer_information
 (Phase_Evidence None)
 (Phase_item
 Confirm_government_regulations)
 (Phase_plan Being_Planned)
 (Phase_Status Risk))

([arvin_00490] of rfq_header_info)

([arvin_00919] of Action_plan)

([arvin_01493] of Materials
 (material_type
 "CR1_Aluminium_killed_FeP13")
 (potential_supplier [arvin_01494]))

([arvin_01494] of suppliers
 (contact_name "Mr Smith")
 (company_name "gammastamp")
 (contact_address "birmingham"))

([arvin_01497] of Materials
 (material_type "Actetal")
 (potential_supplier [arvin_01494]))

([arvin_01498] of Materials
 (material_type
 3111_type_9/2_con_B")
 (potential_supplier
 [arvin_01494]
 [arvin_01499]))

([arvin_01499] of suppliers
 (contact_name "Mr Jones")
 (company_name "Barton")
 (contact_address "Birmingham"))

([arvin_01500] of Materials
 (material_type "Stainless_steel_BS2056")
 (potential_supplier [arvin_01494]))

"BS-

([arvin_01501] of Materials
 (material_type "blank_XCO_0.2-
 0.3%Cr_Overmould_Hytral_5526_black")
 (potential_supplier [arvin_01494]))

([arvin_01502] of Materials
 (material_type "EPDM_Santropene_101-
 80")
 (potential_supplier [arvin_01494]))

([arvin_01503] of Materials
 (material_type "Nye_Rheolube")
 (potential_supplier [arvin_01494]))

([arvin_01504] of Materials
 (material_type "Hytrel_G3548L")
 (potential_supplier
 [arvin_01505]
 [arvin_01506]))

([arvin_01505] of suppliers
 (contact_name "Mr Davis")
 (company_name "Barkley"))

([arvin_01506] of suppliers
 (contact_name "Mrs Thomason")
 (company_name "SMPI"))

([arvin_01508] of Materials
 (material_type "Acetal_Delrin_900")
 (potential_supplier
 [arvin_01505]
 [arvin_01506]))

([arvin_01509] of Materials
 (material_type "Acetal_Delrin_500")
 (potential_supplier
 [arvin_01505]
 [arvin_01506]))

([arvin_01510] of Materials
 (material_type "Acetal_delrin_525")
 (potential_supplier
 [arvin_01505]
 [arvin_01506]))

([arvin_01511] of Materials
 (material_type "Zinc_alloy")
 (potential_supplier [arvin_01512]))

([arvin_01512] of suppliers
 (contact_name "Mr Smithers")
 (company_name "Dynacast")
 (contact_address "Springfield"))

([arvin_01513] of Materials
 (material_type
 "Stainless_steel_BS5770_part4_301s21")
 (potential_supplier
 [arvin_01512]
 [arvin_01514]))

([arvin_01514] of suppliers

(contact_name "Mr Howard")
 (company_name "Peterson_Springs")
 (contact_address "Coventry"))

([arvin_01515] of Materials
 (material_type "BS3111_type_0/1_con_B")
 (potential_supplier
 [arvin_01494]
 [arvin_01517]))

([arvin_01516] of Materials
 (material_type
 "Celanex_PBT_2300GV1/30")
 (potential_supplier
 [arvin_01505]
 [arvin_01494]))

([arvin_01517] of suppliers
 (contact_name "Mr Houston")
 (company_name "RT_Palmer")
 (contact_address "Leeds"))

([arvin_01518] of Materials
 (material_type "Stainless_steel_shaft")
 (potential_supplier [arvin_01519]))

([arvin_01519] of suppliers
 (contact_name "Mr Johnson")
 (company_name "Johnson")
 (contact_address "Leicester"))

([arvin_01520] of Materials
 (material_type
 "C35300_High_leaded_brass")
 (potential_supplier [arvin_01506]))

([arvin_01521] of Materials
 (material_type
 "Ticona_hostaform_C9021")
 (potential_supplier
 [arvin_01522]
 [arvin_01506]))

([arvin_01522] of suppliers
 (contact_name "Mr Sergejovic")
 (company_name "Mikron")
 (contact_address "St_Petersburg"))

([arvin_01523] of Materials
 (material_type "Zinc_alloy_ZL2")
 (potential_supplier [arvin_01524]))

([arvin_01524] of suppliers
 (contact_name "Mr Clark")
 (company_name "Carford")
 (contact_address "Derby"))

([arvin_01525] of Materials
 (material_type "Acetal_delrin_100")
 (potential_supplier
 [arvin_01505]
 [arvin_01506]))

([arvin_01526] of Materials
 (material_type "PBT_30%GF")
 (potential_supplier [arvin_01527]))

([arvin_01527] of suppliers
 (contact_name "Miss Lovely")
 (company_name "Burgess")
 (contact_address "Devon"))

([arvin_01528] of Materials
 (material_type
 "Steel_cs_60mod_0.4c_min")
 (potential_supplier
 [arvin_01529]
 [arvin_01527]))

([arvin_01529] of suppliers
 (contact_name "Mr Patel")
 (company_name "bacol")
 (contact_address "Leicester"))

([arvin_01530] of Materials
 (material_type "Crastin_LW9020")
 (potential_supplier
 [arvin_01505]
 [arvin_01506]))

([arvin_01531] of Materials
 (material_type "QSTE_500")
 (potential_supplier [arvin_01494]))

([arvin_01532] of Materials
 (material_type
 "42CrMo4_harden_and_temper_HRC_27-31")
 (potential_supplier [arvin_01494]))

([arvin_01533] of Materials
 (material_type "POM")
 (potential_supplier [arvin_01494]))

([arvin_01534] of Finish
 (material_type "Nitrotec_NO40")
 (potential_supplier [arvin_01494]))

([arvin_01535] of Finish
 (material_type "deltacol_black_8-
 12_microns_zinc_6_microns_max_deltacol")
 (potential_supplier
 [arvin_01494]
 [arvin_01499]
 [arvin_01517]))

([arvin_01536] of Finish
 (material_type "Blank_depton_10-
 20_micron")
 (potential_supplier [arvin_01494]))

([arvin_01537] of Finish
 (material_type
 "White_zinc_plate_without_Cr6")
 (potential_supplier [arvin_01494]))

([arvin_01538] of Finish
 (material_type "Delt_at_one+Delta_seal")
 (potential_supplier [arvin_01494]))

<p>([arvin_01546] of Bill_of_Material_items (quantity 1) (bom_part_no "FA10B00Q") (product_desc "Retention_Plate_sub_assy_RH")</p>	<p>(bom_reference_no 1) (production_supplier [arvin_01494]) (BoM_weight 184.0) (issue_level "1") (Latch_location RIGHT))</p>
<p>([arvin_01961] of Bill_of_Material_items (quantity 1) (bom_part_no "FA11B00R") (product_desc "retention_plate_sub_assy_LH") (bom_reference_no 2) (production_supplier [arvin_01494]) (BoM_weight 184.0) (issue_level "1") (Latch_location LEFT))</p>	<p>(BoM_Material [arvin_01493]) (production_supplier [arvin_01494]) (BoM_weight 116.0) (issue_level "8") (Latch_location LEFT))</p>
<p>([arvin_01962] of Bill_of_Material_items (quantity 1) (bom_part_no "FA10B18R") (product_desc "Retention_plate_sub_assy_FMS_RH") (bom_reference_no 3) (production_supplier [arvin_01494]) (issue_level "1") (Latch_location RIGHT))</p>	<p>([arvin_01966] of Bill_of_Material_items (quantity 1) (bom_part_no "FA02C14T") (prototype_supplier [arvin_01494]) (product_desc "Pawl_spring_holder_RH") (bom_reference_no 7) (BoM_Material [arvin_01497]) (production_supplier [arvin_01494]) (BoM_weight 1.0) (issue_level "4") (Latch_location RIGHT))</p>
<p>([arvin_01963] of Bill_of_Material_items (quantity 1) (bom_part_no "FA11B18G") (product_desc "Retention_plate_sub_assy_FMS_LH") (bom_reference_no 4) (production_supplier [arvin_01494]) (issue_level "1") (Latch_location LEFT))</p>	<p>([arvin_01967] of Bill_of_Material_items (quantity 1) (bom_part_no "FA03C14H") (prototype_supplier [arvin_01494]) (product_desc "Pawl_spring_holder_LH") (bom_reference_no 8) (BoM_Material [arvin_01497]) (production_supplier [arvin_01494]) (BoM_weight 1.0) (issue_level "4") (Latch_location LEFT))</p>
<p>([arvin_01964] of Bill_of_Material_items (quantity 1) (bom_part_no "FA02C20T") (prototype_supplier [arvin_01494]) (bom_oem_part_no "GO3M5A- R21812_PIA001") (product_desc "Retention_plate_RH") (bom_reference_no 5) (BoM_Material [arvin_01493]) (production_supplier [arvin_01494]) (BoM_weight 116.0) (issue_level "8") (Latch_location RIGHT))</p>	<p>([arvin_01968] of Bill_of_Material_items (quantity 2) (bom_part_no "FA02Z70Y") (prototype_supplier [arvin_01499]) (bom_oem_part_no "GO3M5A- R21812_PIA003") (product_desc "Claw_and_pawl_rivet_GM") (bom_reference_no 9) (BoM_Material [arvin_01498]) (production_supplier [arvin_01494]) (BoM_weight 9.1) (issue_level "2"))</p>
<p>([arvin_01965] of Bill_of_Material_items (quantity 1) (bom_part_no "FA03C20H") (prototype_supplier [arvin_01494]) (bom_oem_part_no "GO3M5A- R21813_PIA001") (product_desc "Retention_plate_LH") (bom_reference_no 6)</p>	<p>([arvin_01969] of Bill_of_Material_items (quantity 1) (bom_part_no "327890") (prototype_supplier [arvin_01494]) (product_desc "Claw_spring") (bom_reference_no 10) (BoM_Material [arvin_01500]) (production_supplier [arvin_01494]) (BoM_weight 0.39)</p>

(issue_level "B"))

[[arvin_01970] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA02C00X")
 (prototype_supplier [arvin_01494])
 (bom_oem_part_no "GO3M5A-R21812")
 (product_desc "Claw_overmoulded_RH")
 (bom_reference_no 11)
 (BoM_Material [arvin_01501])
 (production_supplier [arvin_01494])
 (BoM_weight 42.21)
 (issue_level "14")
 (Latch_location RIGHT))

[[arvin_01971] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA03C00J")
 (prototype_supplier [arvin_01494])
 (bom_oem_part_no "GO3M5A-R21813_PIA002")
 (product_desc "Claw_overmould_LH")
 (bom_reference_no 12)
 (BoM_Material [arvin_01501])
 (production_supplier [arvin_01494])
 (BoM_weight 42.21)
 (issue_level "14")
 (Latch_location LEFT))

[[arvin_01972] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA02C13N")
 (prototype_supplier [arvin_01494])
 (product_desc "Pawl_overmoulded_RH")
 (bom_reference_no 13)
 (BoM_Material [arvin_01501])
 (production_supplier [arvin_01494])
 (BoM_weight 12.26)
 (issue_level "28")
 (Latch_location RIGHT))

[[arvin_01973] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA03C13L")
 (prototype_supplier [arvin_01494])
 (product_desc "Pawl_overmoulded_LH")
 (bom_reference_no 14)
 (BoM_Material [arvin_01501])
 (production_supplier [arvin_01494])
 (BoM_weight 12.26)
 (issue_level "28")
 (Latch_location LEFT))

[[arvin_01974] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA02C16W")
 (prototype_supplier [arvin_01494])
 (product_desc "Cam_plastic_insert")
 (bom_reference_no 15)
 (production_supplier [arvin_01494])
 (issue_level "1")
 (Latch_location RIGHT))

[[arvin_01975] of Bill_of_Material_items

(quantity 1)
 (bom_part_no "FA03C15F")
 (prototype_supplier [arvin_01494])
 (product_desc "Cam_plastic_insert_LH")
 (bom_reference_no 16)
 (production_supplier [arvin_01494])
 (issue_level "1")
 (Latch_location LEFT))

[[arvin_01976] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA02Z24H")
 (prototype_supplier [arvin_01494])
 (product_desc "Pawl_spring")
 (bom_reference_no 17)
 (BoM_Material [arvin_01500])
 (production_supplier [arvin_01494])
 (BoM_weight 0.23)
 (issue_level "2"))

[[arvin_01977] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA02K60Z")
 (prototype_supplier [arvin_01494])
 (product_desc "Seal-fishmouth_RH")
 (bom_reference_no 18)
 (BoM_Material [arvin_01502])
 (production_supplier [arvin_01494])
 (BoM_weight 1.16)
 (issue_level "1")
 (Latch_location RIGHT))

[[arvin_01978] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA03K60B")
 (prototype_supplier [arvin_01494])
 (product_desc "Seal-fishmouth_LH")
 (bom_reference_no 19)
 (BoM_Material [arvin_01502])
 (production_supplier [arvin_01494])
 (BoM_weight 1.16)
 (issue_level "1")
 (Latch_location LEFT))

[[arvin_01979] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA10Z90P")
 (product_desc "Grease")
 (bom_reference_no 20)
 (BoM_Material [arvin_01503])
 (production_supplier [arvin_01494])
 (issue_level "1"))

[[arvin_01980] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA02B00Y")
 (product_desc "Body_sub_assy_FRT_RH")
 (bom_reference_no 21)
 (production_supplier [arvin_01506])
 (issue_level "1")
 (Latch_location front-right))

[[arvin_01981] of Bill_of_Material_items
 (quantity 1)

(bom_part_no "FA03B00K")
 (product_desc "Body_sub_assy_FRT_LH")
 (bom_reference_no 22)
 (production_supplier [arvin_01506])
 (issue_level "1")
 (Latch_location front-left))

([arvin_01982] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA10B12Y")
 (product_desc "Body_sub_assy_RR_RH")
 (bom_reference_no 23)
 (production_supplier [arvin_01506])
 (issue_level "1")
 (Latch_location rear-right))

([arvin_01983] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA11B12K")
 (product_desc "Body_sub_assy_RR_LH")
 (bom_reference_no 24)
 (production_supplier [arvin_01506])
 (issue_level "1")
 (Latch_location rear-left))

([arvin_01985] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA10B05W")
 (product_desc
 "Body_sub_assy_RR_manual_RH")
 (bom_reference_no 25)
 (production_supplier [arvin_01506])
 (issue_level "1")
 (Latch_location rear-right)
 (latch_type [arvin_02619]))

([arvin_01986] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA11B05Z")
 (product_desc
 "Body_sub_assy_RR_manual_LH")
 (bom_reference_no 26)
 (production_supplier [arvin_01506])
 (issue_level "1")
 (Latch_location rear-left)
 (latch_type [arvin_02619]))

([arvin_01987] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA10B13J")
 (product_desc
 "Body_sub_assy_FRT_keyless_RH")
 (bom_reference_no 27)
 (production_supplier [arvin_01506])
 (issue_level "1")
 (Latch_location front-right)
 (latch_type [arvin_02620]))

([arvin_01988] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA11B13S")
 (product_desc
 "Body_sub_assy_FR_keyless_LH")
 (bom_reference_no 28)
 (production_supplier [arvin_01506])
 (issue_level "1")
 (Latch_location front-left)
 (latch_type [arvin_02620]))

([arvin_01989] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA10B16O")
 (product_desc
 "Body_sub_assy_FRT_rod_RH")
 (bom_reference_no 29)
 (production_supplier [arvin_01506])
 (issue_level "1")
 (Latch_location front-right)
 (latch_type [arvin_02621]))

([arvin_01990] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA11B16D")
 (product_desc
 "Body_sub_assy_FRT_rod_LH")
 (bom_reference_no 30)
 (production_supplier [arvin_01506])
 (issue_level "1")
 (Latch_location front-left)
 (latch_type [arvin_02621]))

([arvin_01991] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA10B17M")
 (product_desc "Body_sub_assy_FRT_non-
 slam_RH")
 (bom_reference_no 31)
 (production_supplier [arvin_01506])
 (issue_level "1")
 (Latch_location front-right)
 (latch_type [arvin_02622]))

([arvin_01992] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA11B17A")
 (product_desc "Body_sub_assy_FRT_non-
 slam_LH")
 (bom_reference_no 32)
 (production_supplier [arvin_01506])
 (issue_level "1")
 (Latch_location front-left)
 (latch_type [arvin_02622]))

([arvin_01993] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA10B19U")
 (product_desc "Body_sub_assy_FRT_non-
 slam-keyless_RH")
 (bom_reference_no 33)
 (production_supplier [arvin_01506])
 (issue_level "1")
 (Latch_location front-right)
 (latch_type
 [arvin_02620]
 [arvin_02622]))

([arvin_01994] of Bill_of_Material_items
 (quantity 1)

(bom_part_no "FA11B19Q")
 (product_desc "Body_sub_assy_FRT_non-
 slam-keyless_LH")
 (bom_reference_no 34)
 (production_supplier [arvin_01506])
 (issue_level "1")
 (Latch_location front-left)
 (latch_type
 [arvin_02620]
 [arvin_02622]))

([arvin_01995] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA10B20P")
 (product_desc "Body_sub_assy_FRT_self-
 cancel_RH")
 (bom_reference_no 35)
 (production_supplier [arvin_01506])
 (issue_level "1")
 (Latch_location front-right)
 (latch_type [arvin_02623]))

([arvin_01996] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA11B20X")
 (product_desc "Body_sub_assy_FRT_self-
 cancel_LH")
 (bom_reference_no 36)
 (production_supplier [arvin_01506])
 (issue_level "1")
 (Latch_location front-left)
 (latch_type [arvin_02623]))

([arvin_01997] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA10B21C")
 (product_desc
 "Body_sub_assy_FRT_power-release_RH")
 (bom_reference_no 37)
 (production_supplier [arvin_01506])
 (issue_level "1")
 (Latch_location front-right)
 (latch_type [arvin_02618]))

([arvin_01998] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA11B21V")
 (product_desc
 "Body_sub_assy_FRT_power-release_LH")
 (bom_reference_no 38)
 (production_supplier [arvin_01506])
 (issue_level "1")
 (Latch_location front-left)
 (latch_type [arvin_02618]))

([arvin_01999] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA10B22O")
 (product_desc
 "Body_sub_assy_RR_power-release_RH")
 (bom_reference_no 39)
 (production_supplier [arvin_01506])
 (issue_level "1")
 (Latch_location rear-right))

(latch_type [arvin_02618]))
 ([arvin_02000] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA11B22D")
 (product_desc
 "Body_sub_assy_RR_power-release_LH")
 (bom_reference_no 40)
 (production_supplier [arvin_01506])
 (issue_level "1")
 (Latch_location rear-left)
 (latch_type [arvin_02618]))

([arvin_02001] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA02Q00N")
 (prototype_supplier [arvin_01505])
 (product_desc "Striker_buffer_RH")
 (bom_reference_no 41)
 (BoM_Material [arvin_01504])
 (production_supplier [arvin_01506])
 (BoM_weight 2.72)
 (issue_level "4")
 (Latch_location RIGHT))

([arvin_02002] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA03Q00L")
 (prototype_supplier [arvin_01505])
 (product_desc "Striker_buffer_LH")
 (bom_reference_no 42)
 (BoM_Material [arvin_01504])
 (production_supplier [arvin_01506])
 (BoM_weight 2.72)
 (issue_level "4")
 (Latch_location LEFT))

([arvin_02003] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA02K00K")
 (prototype_supplier [arvin_01505])
 (bom_oem_part_no "GO3M5A-
 R21812_PIA018")
 (product_desc "Body_latch_RH")
 (bom_reference_no 43)
 (BoM_Material [arvin_01508])
 (production_supplier [arvin_01506])
 (BoM_weight 22.9)
 (issue_level "11")
 (Latch_location RIGHT))

([arvin_02004] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA03K00W")
 (prototype_supplier [arvin_01505])
 (bom_oem_part_no "GO3M5A-R21813")
 (product_desc "Body_latch_LH")
 (bom_reference_no 44)
 (BoM_Material [arvin_01508])
 (production_supplier [arvin_01506])
 (BoM_weight 22.9)
 (issue_level "11")
 (Latch_location LEFT))

([arvin_02005] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA16P40X")
 (prototype_supplier [arvin_01505])
 (product_desc "Power-release_body_RH")
 (bom_reference_no 254)
 (BoM_Material [arvin_01508])
 (production_supplier [arvin_01506])
 (issue_level "1")
 (Latch_location RIGHT)
 (latch_type [arvin_02618]))

([arvin_02006] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA17P40J")
 (prototype_supplier [arvin_01505])
 (product_desc "Power-release_body_LH")
 (bom_reference_no 255)
 (BoM_Material [arvin_01508])
 (production_supplier [arvin_01506])
 (issue_level "1")
 (Latch_location LEFT)
 (latch_type [arvin_02618]))

([arvin_02007] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA06G31R")
 (prototype_supplier [arvin_01505])
 (bom_oem_part_no "GO4N5A-A21812-PIA009")
 (product_desc "Key_lever-Cable_RH")
 (bom_reference_no 45)
 (BoM_Material [arvin_01509])
 (production_supplier [arvin_01506])
 (BoM_weight 4.4)
 (issue_level "10")
 (Latch_location RIGHT))

([arvin_02008] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA07G31G")
 (prototype_supplier [arvin_01505])
 (bom_oem_part_no "GO4N5A-A21813-PIA009")
 (product_desc "Key_lever-cable_LH")
 (bom_reference_no 46)
 (BoM_Material [arvin_01509])
 (production_supplier [arvin_01506])
 (BoM_weight 4.4)
 (issue_level "10")
 (Latch_location LEFT))

([arvin_02009] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA10H05L")
 (prototype_supplier [arvin_01505])
 (product_desc "Key_lever_rod_RH")
 (bom_reference_no 47)
 (BoM_Material [arvin_01509])
 (production_supplier [arvin_01506])
 (BoM_weight 4.2)
 (issue_level "5")
 (Latch_location RIGHT)
 (latch_type [arvin_02621]))

([arvin_02010] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA11H05E")
 (prototype_supplier [arvin_01505])
 (product_desc "Key_lever_rod_LH")
 (bom_reference_no 48)
 (BoM_Material [arvin_01509])
 (production_supplier [arvin_01506])
 (BoM_weight 4.2)
 (issue_level "5")
 (Latch_location LEFT)
 (latch_type [arvin_02621]))

([arvin_02011] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA02J43A")
 (prototype_supplier [arvin_01505])
 (bom_oem_part_no "GO3M5A-R26412-PIA027")
 (product_desc "Child_lock_barrel_RH")
 (bom_reference_no 49)
 (BoM_Material [arvin_01509])
 (production_supplier [arvin_01506])
 (BoM_weight 1.2)
 (issue_level "7")
 (Latch_location RIGHT)
 (latch_type [arvin_02627]))

([arvin_02012] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA03J43O")
 (prototype_supplier [arvin_01505])
 (bom_oem_part_no "GO3M5A-R26413-PIA027")
 (product_desc "Child_lock_barrel_LH")
 (bom_reference_no 50)
 (BoM_Material [arvin_01509])
 (production_supplier [arvin_01506])
 (BoM_weight 1.2)
 (issue_level "7")
 (Latch_location LEFT)
 (latch_type [arvin_02627]))

([arvin_02013] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA02J44G")
 (prototype_supplier [arvin_01505])
 (product_desc "Child_safety_lever_Manual_RH")
 (bom_reference_no 51)
 (BoM_Material [arvin_01510])
 (production_supplier [arvin_01506])
 (BoM_weight 2.1)
 (issue_level "7")
 (Latch_location RIGHT)
 (latch_type [arvin_02627]
 [arvin_02619]))

([arvin_02014] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA03J44M")
 (prototype_supplier [arvin_01505])

```

        (product_desc
"Child_safety_lever_manual_LH")
        (bom_reference_no 52)
        (BoM_Material [arvin_01510])
        (production_supplier [arvin_01506])
        (BoM_weight 2.1)
        (issue_level "7")
        (Latch_location LEFT)
        (latch_type
            [arvin_02627]
            [arvin_02619]))

([arvin_02015] of Bill_of_Material_items
    (quantity 1)
    (bom_part_no "FA06D00J")
    (prototype_supplier [arvin_01505])
    (product_desc "Self_cancel_lever_RH")
    (bom_reference_no 53)
    (BoM_Material [arvin_01509])
    (production_supplier [arvin_01506])
    (BoM_weight 1.1)
    (issue_level "1")
    (Latch_location RIGHT))

([arvin_02016] of Bill_of_Material_items
    (quantity 1)
    (bom_part_no "FA07D00S")
    (prototype_supplier [arvin_01505])
    (product_desc "Self_cancel_lever_LH")
    (bom_reference_no 54)
    (BoM_Material [arvin_01509])
    (production_supplier [arvin_01506])
    (BoM_weight 1.1)
    (issue_level "1")
    (Latch_location LEFT))

([arvin_02017] of Bill_of_Material_items
    (quantity 1)
    (bom_part_no "FA06F20K")
    (prototype_supplier [arvin_01505])
    (product_desc "Self_cancel_link_RH")
    (bom_reference_no 55)
    (BoM_Material [arvin_01509])
    (production_supplier [arvin_01506])
    (BoM_weight 0.8)
    (issue_level "1")
    (Latch_location RIGHT))

([arvin_02018] of Bill_of_Material_items
    (quantity 1)
    (bom_part_no "FA07F20W")
    (prototype_supplier [arvin_01505])
    (product_desc "Self_cancel_link_LH")
    (bom_reference_no 56)
    (BoM_Material [arvin_01509])
    (production_supplier [arvin_01506])
    (BoM_weight 0.8)
    (issue_level "1")
    (Latch_location LEFT))

([arvin_02019] of Bill_of_Material_items
    (quantity 1)
    (bom_part_no "FA06A00H")
    (prototype_supplier [arvin_01505])

```

```

        (product_desc "Non-slam_lever_RH")
        (bom_reference_no 57)
        (BoM_Material [arvin_01509])
        (production_supplier [arvin_01506])
        (BoM_weight 1.6)
        (issue_level "4")
        (Latch_location RIGHT))

([arvin_02020] of Bill_of_Material_items
    (quantity 1)
    (bom_part_no "FA07A00N")
    (prototype_supplier [arvin_01505])
    (product_desc "Non-slam_lever_LH")
    (bom_reference_no 58)
    (BoM_Material [arvin_01509])
    (production_supplier [arvin_01506])
    (BoM_weight 1.6)
    (issue_level "4")
    (Latch_location LEFT))

([arvin_02021] of Bill_of_Material_items
    (quantity 1)
    (bom_part_no "FA06J01T")
    (prototype_supplier [arvin_01514])
    (product_desc "Non-slam_spring_RH")
    (bom_reference_no 59)
    (BoM_Material [arvin_01500])
    (production_supplier [arvin_01506])
    (BoM_weight 0.12)
    (issue_level "2")
    (Latch_location RIGHT))

([arvin_02022] of Bill_of_Material_items
    (quantity 1)
    (bom_part_no "Non-slam_spring_LH")
    (prototype_supplier [arvin_01514])
    (product_desc "Non-slam_spring_LH")
    (bom_reference_no 60)
    (BoM_Material [arvin_01500])
    (production_supplier [arvin_01506])
    (BoM_weight 0.12)
    (issue_level "2")
    (Latch_location LEFT))

([arvin_02023] of Bill_of_Material_items
    (quantity 2)
    (bom_part_no "FA02Z55T")
    (product_desc "Power-release_gear")
    (issue_level "1")
    (latch_type [arvin_02618]))

([arvin_02024] of Bill_of_Material_items
    (quantity 2)
    (bom_part_no "FA16Z70D")
    (product_desc "Rivet_P/L")
    (issue_level "1"))

([arvin_02025] of Bill_of_Material_items
    (quantity 1)
    (bom_part_no "FA02F20X")
    (prototype_supplier [arvin_01505])
    (product_desc "Power-
release_quadrant_RH")
    (bom_reference_no 69)

```

(production_supplier [arvin_01506]) (issue_level "1") (Latch_location RIGHT) (latch_type [arvin_02618]))	(production_supplier [arvin_01512]) (BoM_weight 13.0) (issue_level "8") (Latch_location RIGHT))
([arvin_02026] of Bill_of_Material_items (quantity 1) (bom_part_no "FA03F20J") (prototype_supplier [arvin_01505]) (product_desc "Power-release_quadrant_LH") (bom_reference_no 70) (production_supplier [arvin_01506]) (issue_level "1") (Latch_location LEFT) (latch_type [arvin_02618]))	([arvin_02032] of Bill_of_Material_items (quantity 1) (bom_part_no "FA03C11R") (prototype_supplier [arvin_01512]) (product_desc "Pawl_lifter_LH") (bom_reference_no 74) (BoM_Material [arvin_01511]) (production_supplier [arvin_01512]) (BoM_weight 13.0) (issue_level "8") (Latch_location LEFT))
([arvin_02027] of Bill_of_Material_items (quantity 1) (bom_part_no "FA02B01J") (prototype_supplier [arvin_01512]) (product_desc "Pawl_lifter_sub_assy_RH") (bom_reference_no 71) (production_supplier [arvin_01512]) (issue_level "1") (Latch_location RIGHT))	([arvin_02033] of Bill_of_Material_items (quantity 1) (bom_part_no "FA02C16W") (prototype_supplier [arvin_01512]) (product_desc "Power-release_pawl_lifter") (production_supplier [arvin_01512]) (issue_level "1") (Latch_location RIGHT) (latch_type [arvin_02618]))
([arvin_02028] of Bill_of_Material_items (quantity 1) (bom_part_no "FA03B01S") (prototype_supplier [arvin_01512]) (product_desc "Pawl_lifter_sub_assy_LH") (bom_reference_no 72) (production_supplier [arvin_01512]) (issue_level "1") (Latch_location LEFT))	([arvin_02034] of Bill_of_Material_items (quantity 1) (bom_part_no "FA03C16Z") (prototype_supplier [arvin_01512]) (product_desc "Power-release_pawl_lifter_LH") (production_supplier [arvin_01512]) (issue_level "1") (Latch_location LEFT) (latch_type [arvin_02618]))
([arvin_02029] of Bill_of_Material_items (quantity 1) (prototype_supplier [arvin_01512]) (product_desc "Power-release_pawl_lifter_sub_assy_RH") (production_supplier [arvin_01512]) (issue_level "1") (Latch_location RIGHT) (latch_type [arvin_02618]))	([arvin_02035] of Bill_of_Material_items (quantity 2) (bom_part_no "FA10H03R") (prototype_supplier [arvin_01505]) (product_desc "Lock-link_RH") (bom_reference_no 75) (BoM_Material [arvin_01509]) (production_supplier [arvin_01512]) (BoM_weight 0.6) (issue_level "4") (Latch_location RIGHT))
([arvin_02030] of Bill_of_Material_items (quantity 1) (prototype_supplier [arvin_01512]) (product_desc "Power-release_pawl_lifter_sub_assy_LH") (production_supplier [arvin_01512]) (issue_level "1") (Latch_location LEFT) (latch_type [arvin_02618]))	([arvin_02036] of Bill_of_Material_items (quantity 2) (bom_part_no "FA11H03G") (prototype_supplier [arvin_01505]) (product_desc "Lock_link_LH") (bom_reference_no 76) (BoM_Material [arvin_01509]) (production_supplier [arvin_01512]) (BoM_weight 0.6) (issue_level "4") (Latch_location LEFT))
([arvin_02031] of Bill_of_Material_items (quantity 1) (bom_part_no "FA02C11Q") (prototype_supplier [arvin_01512]) (product_desc "Pawl_lifter_RH") (bom_reference_no 73) (BoM_Material [arvin_01511])	([arvin_02038] of %3APAL-CONSTRAINT)

([arvin_02039] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA02Z21R")
 (prototype_supplier [arvin_01514])
 (product_desc "Lock_link_retainer_RH")
 (bom_reference_no 77)
 (BoM_Material [arvin_01513])
 (production_supplier [arvin_01512])
 (BoM_weight 2.0)
 (issue_level "5")
 (Latch_location RIGHT))

([arvin_02040] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA03Z21G")
 (prototype_supplier [arvin_01514])
 (product_desc "Lock_link_retainer_LH")
 (bom_reference_no 78)
 (BoM_Material [arvin_01513])
 (production_supplier [arvin_01512])
 (BoM_weight 2.0)
 (issue_level "5")
 (Latch_location LEFT))

([arvin_02041] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA02G01N")
 (prototype_supplier [arvin_01505])
 (product_desc "Cam-index_RH")
 (bom_reference_no 79)
 (BoM_Material [arvin_01509])
 (production_supplier [arvin_01512])
 (BoM_weight 2.0)
 (issue_level "8")
 (Latch_location RIGHT))

([arvin_02042] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA03G01L")
 (prototype_supplier [arvin_01505])
 (product_desc "Cam-index_LH")
 (bom_reference_no 80)
 (BoM_Material [arvin_01509])
 (production_supplier [arvin_01512])
 (BoM_weight 2.0)
 (issue_level "8")
 (Latch_location LEFT))

([arvin_02043] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA10Z20W")
 (prototype_supplier [arvin_01514])
 (product_desc "Index_spring_RH")
 (bom_reference_no 81)
 (BoM_Material [arvin_01500])
 (production_supplier [arvin_01512])
 (BoM_weight 0.8)
 (issue_level "2")
 (Latch_location RIGHT))

([arvin_02044] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA11Z20Z")
 (prototype_supplier [arvin_01514])

(product_desc "Index_spring_LH")
 (bom_reference_no 82)
 (BoM_Material [arvin_01500])
 (production_supplier [arvin_01512])
 (BoM_weight 0.8)
 (issue_level "2")
 (Latch_location LEFT))

([arvin_02045] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA02B02P")
 (prototype_supplier [arvin_01494])
 (product_desc "Back_plate_sub_assy-
 cable_FRT_RH")
 (bom_reference_no 83)
 (production_supplier [arvin_01494])
 (BoM_weight 137.0)
 (issue_level "3")
 (Latch_location front-right))

([arvin_02046] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA03B02X")
 (prototype_supplier [arvin_01494])
 (product_desc "Back_plate_sub_assy-
 cable_FRT_LH")
 (bom_reference_no 84)
 (production_supplier [arvin_01494])
 (BoM_weight 137.0)
 (issue_level "3")
 (Latch_location front-left))

([arvin_02047] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA02B04O")
 (prototype_supplier [arvin_01494])
 (product_desc "Back_plate_sub_assy-
 cable_RR_RH")
 (bom_reference_no 85)
 (production_supplier [arvin_01494])
 (BoM_weight 136.5)
 (issue_level "1")
 (Latch_location rear-right))

([arvin_02048] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA03B04D")
 (prototype_supplier [arvin_01494])
 (product_desc "Back_plate_sub_assy-
 cable_RR_LH")
 (bom_reference_no 86)
 (production_supplier [arvin_01494])
 (BoM_weight 136.5)
 (issue_level "1")
 (Latch_location rear-left))

([arvin_02049] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA10B01E")
 (prototype_supplier [arvin_01494])
 (product_desc "Back_plate_sub_assy-
 rod_push_pull_RH")
 (bom_reference_no 87)
 (production_supplier [arvin_01494])

```

(issue_level "1")
(Latch_location RIGHT)
(latch_type
  [arvin_02626]
  [arvin_02621]))

([arvin_02050] of Bill_of_Material_items
  (quantity 1)
  (bom_part_no "FA11B01U")
  (prototype_supplier [arvin_01494])
  (product_desc "Back_plate_sub_assy-
rod_push_pull_LH")
  (bom_reference_no 88)
  (production_supplier [arvin_01494])
  (issue_level "1")
  (Latch_location LEFT)
  (latch_type
    [arvin_02621]
    [arvin_02626]))

([arvin_02051] of Bill_of_Material_items
  (quantity 1)
  (bom_part_no "FA10B23M")
  (prototype_supplier [arvin_01494])
  (product_desc "Back_plate_sub_assy-
rod_mazda_RH")
  (bom_reference_no 89)
  (production_supplier [arvin_01494])
  (issue_level "1")
  (Latch_location RIGHT)
  (Customer [arvin_02611])
  (latch_type [arvin_02621]))

([arvin_02052] of Bill_of_Material_items
  (quantity 1)
  (bom_part_no "FA11B23A")
  (prototype_supplier [arvin_01494])
  (product_desc "Back_plate_sub_assy-
rod_mazda_LH")
  (bom_reference_no 90)
  (production_supplier [arvin_01494])
  (issue_level "1")
  (Latch_location LEFT)
  (Customer [arvin_02611])
  (latch_type [arvin_02621]))

([arvin_02053] of Bill_of_Material_items
  (quantity 1)
  (bom_part_no "FA10B25U")
  (prototype_supplier [arvin_01494])
  (product_desc
    "Back_plate_sub_assy_FRT_power_release_RH_vol
vo_single_pull")
  (bom_reference_no 93)
  (production_supplier [arvin_01494])
  (issue_level "1")
  (Latch_location front-right)
  (Customer [arvin_02613])
  (latch_type
    [arvin_02624]
    [arvin_02618]))

([arvin_02054] of Bill_of_Material_items
  (quantity 1)

```

```

(bom_part_no "FA11B25Q")
(prototype_supplier [arvin_01494])
(product_desc
  "Back_plate_sub_assy_FRT_power_release_LH_vol
vo_single_pull")
  (bom_reference_no 94)
  (production_supplier [arvin_01494])
  (issue_level "1")
  (Latch_location front-left)
  (Customer [arvin_02613])
  (latch_type
    [arvin_02624]
    [arvin_02618]))

([arvin_02055] of Bill_of_Material_items
  (quantity 1)
  (bom_part_no "FA10B24R")
  (product_desc
    "Back_plate_sub_assy_RR_power_release_RH_volv
o_double_pull")
  (issue_level "1")
  (Latch_location rear-right)
  (Customer [arvin_02613])
  (latch_type
    [arvin_02625]
    [arvin_02618]))

([arvin_02056] of Bill_of_Material_items
  (quantity 1)
  (bom_part_no "FA11B24G")
  (product_desc
    "Back_plate_sub_assy_RR_power_release_LH_volv
o_double_pull")
  (issue_level "1")
  (Latch_location rear-left)
  (Customer [arvin_02613])
  (latch_type
    [arvin_02625]
    [arvin_02618]))

([arvin_02057] of Bill_of_Material_items
  (quantity 1)
  (product_desc
    "Back_plate_sub_assy_FRT_power_release_RH_for
d_push_pull")
  (issue_level "1")
  (Latch_location front-right)
  (Customer [arvin_02612])
  (latch_type
    [arvin_02626]
    [arvin_02618]))

([arvin_02058] of Bill_of_Material_items
  (quantity 1)
  (product_desc
    "Back_plate_sub_assy_FRT_power_release_LH_for
d_push_pull")
  (issue_level "1")
  (Latch_location front-left)
  (Customer [arvin_02612])
  (latch_type
    [arvin_02626]
    [arvin_02618]))

```

([arvin_02059] of Bill_of_Material_items
 (quantity 1)
 (product_desc
 "Back_plate_sub_assy_RR_power_release_RH_ford
 _push_pull")
 (issue_level "1")
 (Latch_location rear-right)
 (Customer [arvin_02612])
 (latch_type
 [arvin_02618]
 [arvin_02626]))

([arvin_02060] of Bill_of_Material_items
 (quantity 1)
 (product_desc
 "Back_plate_sub_assy_RR_power_release_LH_ford
 _push_pull")
 (issue_level "1")
 (Latch_location rear-left)
 (Customer [arvin_02612])
 (latch_type
 [arvin_02626]
 [arvin_02618]))

([arvin_02061] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA02F21V")
 (prototype_supplier [arvin_01494])
 (bom_oem_part_no "GO3M5A-R21812-
 PIA006")
 (product_desc
 "Inside_handle_lever_RH_push_pull_override")
 (bom_reference_no 99)
 (BoM_Material [arvin_01493])
 (production_supplier [arvin_01494])
 (BoM_weight 12.6)
 (issue_level "6")
 (Latch_location RIGHT)
 (latch_type [arvin_02626]))

([arvin_02062] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA03F21P")
 (prototype_supplier [arvin_01494])
 (bom_oem_part_no "GO3M5A-R21813-
 PIA006")
 (product_desc
 "Inside_handle_lever_LH_push_pull_override")
 (bom_reference_no 100)
 (BoM_Material [arvin_01493])
 (production_supplier [arvin_01494])
 (BoM_weight 12.6)
 (issue_level "6")
 (Latch_location LEFT)
 (latch_type [arvin_02626]))

([arvin_02485] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA10D21H")
 (prototype_supplier [arvin_01494])
 (bom_oem_part_no "GO4N5A-R21812-
 PIA008")
 (product_desc
 "Inside_handle_lever_RH_single_double_pull")

(bom_reference_no 101)
 (BoM_Material [arvin_01493])
 (production_supplier [arvin_01494])
 (BoM_weight 14.0)
 (issue_level "19")
 (Latch_location RIGHT)
 (latch_type
 [arvin_02625]
 [arvin_02624]))

([arvin_02486] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA11F21N")
 (prototype_supplier [arvin_01494])
 (bom_oem_part_no "GA4N5A-R21813-
 PIA008")
 (product_desc
 "Inside_handle_lever_LH_single_double_pull")
 (bom_reference_no 19)
 (BoM_Material [arvin_01493])
 (production_supplier [arvin_01494])
 (BoM_weight 14.0)
 (issue_level "19")
 (Latch_location LEFT)
 (latch_type
 [arvin_02625]
 [arvin_02624]))

([arvin_02487] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA06G01Q")
 (prototype_supplier [arvin_01494])
 (bom_oem_part_no "GO3N6A-R21812")
 (product_desc
 "Inside_handle_lever_RH_independent")
 (bom_reference_no 105)
 (BoM_Material [arvin_01493])
 (production_supplier [arvin_01494])
 (BoM_weight 10.5)
 (issue_level "1")
 (Latch_location RIGHT)
 (latch_type [arvin_02628]))

([arvin_02488] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA07G01R")
 (prototype_supplier [arvin_01494])
 (bom_oem_part_no "GO3N6A-R21813-
 PIA001")
 (product_desc
 "Inside_handle_lever_LH_independent")
 (bom_reference_no 1)
 (BoM_Material [arvin_01493])
 (production_supplier [arvin_01494])
 (BoM_weight 10.5)
 (issue_level "1")
 (Latch_location LEFT)
 (latch_type [arvin_02628]))

([arvin_02489] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA10Z21S")
 (prototype_supplier [arvin_01494])

(product_desc
 "Inside_handle_lever_spring_RH")
 (bom_reference_no 107)
 (BoM_Material [arvin_01500])
 (production_supplier [arvin_01494])
 (BoM_weight 3.82)
 (issue_level "9")
 (Latch_location RIGHT))

 ([arvin_02490] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA11Z21K")
 (prototype_supplier [arvin_01494])
 (product_desc
 "Inside_handle_lever_spring_LH")
 (bom_reference_no 9)
 (BoM_Material [arvin_01500])
 (production_supplier [arvin_01494])
 (BoM_weight 3.82)
 (issue_level "9")
 (Latch_location LEFT))

 ([arvin_02491] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA02G00E")
 (prototype_supplier [arvin_01514])
 (product_desc
 "Inside_handle_lever_spring_push_pull_RH")
 (bom_reference_no 109)
 (BoM_Material [arvin_01500])
 (production_supplier [arvin_01494])
 (BoM_weight 2.1)
 (issue_level "3")
 (Latch_location RIGHT)
 (latch_type [arvin_02626]))

 ([arvin_02492] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA03G00U")
 (prototype_supplier [arvin_01514])
 (product_desc
 "Inside_handle_lever_spring_push_pull_LH")
 (bom_reference_no 110)
 (BoM_Material [arvin_01500])
 (production_supplier [arvin_01494])
 (BoM_weight 2.1)
 (issue_level "3")
 (Latch_location LEFT)
 (latch_type [arvin_02626]))

 ([arvin_02493] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA02Z71J")
 (prototype_supplier [arvin_01494])
 (bom_oem_part_no "GO4N5A-A21812-
 PIA012")
 (product_desc
 "Inside_handle_rivet_single_double_push_pull")
 (bom_reference_no 111)
 (BoM_Material [arvin_01515])
 (production_supplier [arvin_01494])
 (BoM_weight 13.1)
 (issue_level "11")
 (latch_type

[arvin_02625]
 [arvin_02626]
 [arvin_02624]))

 ([arvin_02494] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA10L21O")
 (prototype_supplier [arvin_01494])
 (bom_oem_part_no "GO3M5A-R21812-
 PIA014")
 (product_desc "Inside_handle_rivet_rod")
 (bom_reference_no 112)
 (BoM_Material [arvin_01515])
 (production_supplier [arvin_01494])
 (BoM_weight 5.7)
 (issue_level "3")
 (latch_type [arvin_02621]))

 ([arvin_02495] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA10E20Q")
 (prototype_supplier [arvin_01494])
 (bom_oem_part_no "GA3M5A-R21812-
 PIA010")
 (product_desc
 "Outside_handle_lever_rod_RH")
 (bom_reference_no 113)
 (BoM_Material [arvin_01493])
 (production_supplier [arvin_01494])
 (BoM_weight 1.0)
 (issue_level "1")
 (Latch_location RIGHT)
 (latch_type [arvin_02621]))

 ([arvin_02496] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "114")
 (prototype_supplier [arvin_01494])
 (bom_oem_part_no "GO3M5A-R21813-
 PIA010")
 (product_desc
 "Outside_handle_lever_rod_LH")
 (bom_reference_no 114)
 (BoM_Material [arvin_01493])
 (production_supplier [arvin_01494])
 (BoM_weight 5.9)
 (issue_level "1")
 (Latch_location LEFT)
 (latch_type [arvin_02621]))

 ([arvin_02497] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA10E21Q")
 (prototype_supplier [arvin_01494])
 (bom_oem_part_no "GO4N5A-A21812-
 PIA006")
 (product_desc
 "Outside_handle_lever_cable_RH")
 (bom_reference_no 115)
 (BoM_Material [arvin_01493])
 (production_supplier [arvin_01494])
 (BoM_weight 7.0)
 (issue_level "8")
 (Latch_location RIGHT))

([arvin_02498] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA11E21U")
 (prototype_supplier [arvin_01494])
 (bom_oem_part_no "GO4N5A-A21813-
 PIA006")
 (product_desc
 "Outside_handle_lever_cable_LH")
 (bom_reference_no 116)
 (BoM_Material [arvin_01493])
 (production_supplier [arvin_01494])
 (BoM_weight 7.0)
 (issue_level "8")
 (Latch_location LEFT))

([arvin_02499] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA11E25W")
 (prototype_supplier [arvin_01517])
 (product_desc "Intermediate_lever_rivet")
 (bom_reference_no 117)
 (BoM_Material [arvin_01515])
 (production_supplier [arvin_01494])
 (BoM_weight 0.71)
 (issue_level "3"))

([arvin_02500] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA10E23T")
 (prototype_supplier [arvin_01505])
 (product_desc
 "Outside_handle_intermediate_lever_RH")
 (bom_reference_no 118)
 (BoM_Material [arvin_01516])
 (production_supplier [arvin_01494])
 (BoM_weight 2.0)
 (issue_level "5")
 (Latch_location RIGHT))

([arvin_02501] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA11E23H")
 (prototype_supplier [arvin_01505])
 (product_desc
 "Outside_handle_intermediate_lever_LH")
 (bom_reference_no 119)
 (BoM_Material [arvin_01516])
 (production_supplier [arvin_01494])
 (BoM_weight 2.0)
 (issue_level "5")
 (Latch_location LEFT))

([arvin_02502] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA10E25W")
 (prototype_supplier [arvin_01494])
 (bom_oem_part_no "GO4N5A-A21812-
 PIA013")
 (product_desc
 "Outside_handle_lever_rivet")
 (bom_reference_no 120)
 (BoM_Material [arvin_01515])
 (production_supplier [arvin_01494])

(BoM_weight 5.0)
 (issue_level "8"))

([arvin_02503] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA10L22M")
 (prototype_supplier [arvin_01494])
 (bom_oem_part_no "GO3M5A-R21812")
 (product_desc
 "Outside_handle_lever_rivet_rod")
 (bom_reference_no 121)
 (BoM_Material [arvin_01515])
 (production_supplier [arvin_01494])
 (BoM_weight 5.0)
 (issue_level "1"))

([arvin_02504] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA10Z22J")
 (prototype_supplier [arvin_01494])
 (product_desc
 "Outside_handle_spring_RH")
 (bom_reference_no 122)
 (BoM_Material [arvin_01500])
 (production_supplier [arvin_01494])
 (BoM_weight 1.56)
 (issue_level "5")
 (Latch_location RIGHT))

([arvin_02505] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA11Z22S")
 (prototype_supplier [arvin_01494])
 (product_desc
 "Outside_handle_spring_LH")
 (bom_reference_no 123)
 (BoM_Material [arvin_01500])
 (production_supplier [arvin_01494])
 (BoM_weight 1.56)
 (issue_level "5")
 (Latch_location LEFT))

([arvin_02506] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA02H01F")
 (prototype_supplier [arvin_01514])
 (product_desc
 "Outside_handle_spring_rod_RH")
 (bom_reference_no 124)
 (BoM_Material [arvin_01500])
 (production_supplier [arvin_01494])
 (BoM_weight 1.0)
 (issue_level "1")
 (Latch_location RIGHT)
 (latch_type [arvin_02621]))

([arvin_02507] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA03H01T")
 (prototype_supplier [arvin_01514])
 (product_desc
 "Outside_handle_spring_rod_LH")
 (bom_reference_no 125)
 (BoM_Material [arvin_01500])

(production_supplier [arvin_01494])
 (BoM_weight 1.0)
 (issue_level "1")
 (Latch_location LEFT)
 (latch_type [arvin_02621]))

([arvin_02508] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA02C30J")
 (prototype_supplier [arvin_01494])
 (bom_oem_part_no "GO3M5A-R21812-PIA024")
 (product_desc
 "Back_plate_cable_FRT_RH")
 (bom_reference_no 126)
 (BoM_Material [arvin_01493])
 (production_supplier [arvin_01494])
 (BoM_weight 102.0)
 (issue_level "58")
 (Latch_location front-right))

([arvin_02509] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA03C30S")
 (prototype_supplier [arvin_01494])
 (bom_oem_part_no "GO3M5A-R21812-PIA024")
 (product_desc
 "Back_plate_cable_FRT_LH")
 (bom_reference_no 127)
 (BoM_Material [arvin_01493])
 (production_supplier [arvin_01494])
 (BoM_weight 102.0)
 (issue_level "58")
 (Latch_location front-left))

([arvin_02510] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA10C00L")
 (prototype_supplier [arvin_01494])
 (product_desc
 "Back_plate_cable_RR_RH")
 (bom_reference_no 10)
 (BoM_Material [arvin_01493])
 (production_supplier [arvin_01494])
 (BoM_weight 90.0)
 (issue_level "10")
 (Latch_location rear-right))

([arvin_02511] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA11C00E")
 (prototype_supplier [arvin_01494])
 (product_desc
 "Back_plate_cable_RR_LH")
 (bom_reference_no 129)
 (BoM_Material [arvin_01493])
 (production_supplier [arvin_01494])
 (BoM_weight 90.0)
 (issue_level "10")
 (Latch_location rear-left))

([arvin_02512] of Bill_of_Material_items
 (quantity 1)

(bom_part_no "FA10B03T")
 (bom_oem_part_no "GO3M5A-R21812-PIA017")
 (product_desc "Back_plate_rod_RH")
 (bom_reference_no 130)
 (BoM_Material [arvin_01493])
 (BoM_weight 60.0)
 (issue_level "3")
 (Latch_location RIGHT)
 (latch_type [arvin_02621]))

([arvin_02513] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA11B03H")
 (prototype_supplier [arvin_01494])
 (bom_oem_part_no "GO3M5A-R21813-PIA017")
 (product_desc "Back_plate_rod_LH")
 (bom_reference_no 131)
 (BoM_Material [arvin_01493])
 (production_supplier [arvin_01494])
 (BoM_weight 60.0)
 (issue_level "3")
 (Latch_location LEFT)
 (latch_type [arvin_02621]))

([arvin_02514] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA02H02Z")
 (prototype_supplier [arvin_01494])
 (product_desc
 "Back_plate_power_FRT_RH")
 (bom_reference_no 134)
 (BoM_Material [arvin_01493])
 (production_supplier [arvin_01494])
 (BoM_weight 60.0)
 (issue_level "1")
 (Latch_location front-right)
 (latch_type [arvin_02616]))

([arvin_02515] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA03H02B")
 (prototype_supplier [arvin_01494])
 (product_desc
 "Back_plate_power_FRT_LH")
 (bom_reference_no 135)
 (BoM_Material [arvin_01493])
 (production_supplier [arvin_01494])
 (BoM_weight 60.0)
 (issue_level "1")
 (Latch_location front-left)
 (latch_type [arvin_02616]))

([arvin_02516] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA10C30E")
 (prototype_supplier [arvin_01494])
 (product_desc
 "Back_plate_power_RR_RH")
 (BoM_Material [arvin_01493])
 (production_supplier [arvin_01494])
 (issue_level "1")
 (Latch_location rear-right))

(latch_type [arvin_02616]))	(latch_type [arvin_02618]))	
([arvin_02517] of Bill_of_Material_items (quantity 1) (bom_part_no "FA11C30U") (prototype_supplier [arvin_01494]) (product_desc "Back_plate_power_RR_LH") (BoM_Material [arvin_01493]) (production_supplier [arvin_01494]) (issue_level "1") (Latch_location rear-left) (latch_type [arvin_02616]))	([arvin_02525] of Bill_of_Material_items (bom_part_no "FA03D20B") (product_desc "Power_release_engagement_lever_LH") (issue_level "1") (Latch_location LEFT) (latch_type [arvin_02618]))	
([arvin_02518] of Bill_of_Material_items (bom_part_no "FA02A00Z") (product_desc "Power_release_link") (issue_level "1") (Latch_location RIGHT) (latch_type [arvin_02618]))	([arvin_02526] of Bill_of_Material_items (bom_part_no "Power_release_engagement_rivet") (product_desc "Power_release_engagement_rivet") (issue_level "1") (latch_type [arvin_02618]))	
([arvin_02519] of Bill_of_Material_items (bom_part_no "FA03A00B") (product_desc "Power_release_link_LH") (issue_level "1") (Latch_location LEFT) (latch_type [arvin_02618]))	([arvin_02527] of Bill_of_Material_items (bom_part_no "FA03Z76G") (product_desc "Power_release_lever_rivet_1") (issue_level "1") (latch_type [arvin_02618]))	
([arvin_02520] of Bill_of_Material_items (bom_part_no "FA02C12E") (product_desc "Power_release_lever_RH") (issue_level "1") (Latch_location RIGHT) (latch_type [arvin_02618]))	([arvin_02528] of Bill_of_Material_items (bom_part_no "FA03Z77Q") (product_desc "Power_release_lever_rivet_2") (issue_level "1") (latch_type [arvin_02618]))	"Power-
([arvin_02521] of Bill_of_Material_items (bom_part_no "FA03C12U") (product_desc "Power_release_lever_LH") (issue_level "1") (Latch_location LEFT) (latch_type [arvin_02618]))	([arvin_02529] of Bill_of_Material_items (bom_part_no "FA02D40Q") (product_desc "Power_release_switch_cam_RH") (issue_level "1") (Latch_location RIGHT) (latch_type [arvin_02618]))	
([arvin_02522] of Bill_of_Material_items (bom_part_no "FA02E20Y") (product_desc "Power_release_spring_RH") (issue_level "1") (Latch_location RIGHT) (latch_type [arvin_02618]))	([arvin_02530] of Bill_of_Material_items (bom_part_no "FA03D40R") (product_desc "Power_release_switch_cam_LH") (issue_level "1") (Latch_location LEFT) (latch_type [arvin_02618]))	
([arvin_02523] of Bill_of_Material_items (bom_part_no "FA03E20K") (product_desc "Power_release_spring_LH") (issue_level "1") (Latch_location LEFT) (latch_type [arvin_02618]))	([arvin_02531] of Bill_of_Material_items (quantity 1) (bom_part_no "FA06B00B") (product_desc "Lower_case_sub_assy_P/L_base_level_ford_RH") (bom_reference_no 136) (issue_level "1") (Latch_location RIGHT) (Customer [arvin_02612]))	
([arvin_02524] of Bill_of_Material_items (bom_part_no "FA02D20Z") (product_desc "Power_release_engagement_lever_RH") (issue_level "1") (Latch_location RIGHT)	([arvin_02532] of Bill_of_Material_items (quantity 1) (bom_part_no "FA07B00F") (product_desc "Lower_case_sub_assy_P/L_base_level_ford_LH") (bom_reference_no 137)	

(issue_level "1")
 (Latch_location LEFT)
 (Customer [arvin_02612]))

([arvin_02533] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA10B14P")
 (product_desc
 "Lower_case_sub_assy_P/L_BL_volvo_RH")
 (bom_reference_no 138)
 (issue_level "1")
 (Latch_location RIGHT)
 (Customer [arvin_02613]))

([arvin_02534] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA11B14X")
 (product_desc
 "Lower_case_sub_assy_P/L_BL_volvo_LH")
 (bom_reference_no 139)
 (issue_level "1")
 (Latch_location LEFT)
 (Customer [arvin_02613]))

([arvin_02535] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA06B04P")
 (product_desc
 "Lower_case_sub_assy_P/L_BL_mazda_RH")
 (bom_reference_no 140)
 (issue_level "1")
 (Latch_location RIGHT)
 (Customer [arvin_02611]))

([arvin_02536] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA07B04X")
 (product_desc
 "Lower_case_sub_assy_P/L_BL_mazda_LH")
 (bom_reference_no 141)
 (issue_level "1")
 (Latch_location LEFT)
 (Customer [arvin_02611]))

([arvin_02537] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA10B32L")
 (product_desc
 "Lower_case_sub_assy_P/L_child_lock_RR_volvo_
 RH")
 (bom_reference_no 142)
 (issue_level "1")
 (Latch_location rear-right)
 (Customer [arvin_02613])
 (latch_type [arvin_02627]))

([arvin_02538] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA11B32F")
 (product_desc
 "Lower_case_sub_assy_P/L_child_lock_RR_volvo_
 LH")
 (bom_reference_no 143)
 (issue_level "1")

(Latch_location rear-left)
 (Customer [arvin_02613])
 (latch_type [arvin_02627]))

([arvin_02539] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA10B26L")
 (product_desc
 "Lower_case_sub_assy_P/L_ford_RH")
 (bom_reference_no 144)
 (issue_level "1")
 (Latch_location RIGHT)
 (Customer [arvin_02612]))

([arvin_02540] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA11B26E")
 (product_desc
 "Lower_case_sub_assy_P/L_ford_LH")
 (bom_reference_no 145)
 (issue_level "1")
 (Latch_location LEFT)
 (Customer [arvin_02612]))

([arvin_02541] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA10B27H")
 (product_desc
 "Lower_case_sub_assy_P/L_mazda_RH")
 (bom_reference_no 146)
 (issue_level "1")
 (Latch_location RIGHT)
 (Customer [arvin_02611]))

([arvin_02542] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA11B27N")
 (product_desc
 "Lower_case_sub_assy_P/L_mazda_LH")
 (bom_reference_no 147)
 (issue_level "1")
 (Latch_location LEFT)
 (Customer [arvin_02611]))

([arvin_02543] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA10B28F")
 (product_desc
 "Lower_case_sub_assy_P/L_volvo_RH")
 (bom_reference_no 148)
 (issue_level "1")
 (Latch_location RIGHT)
 (Customer [arvin_02613]))

([arvin_02544] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA11B28T")
 (product_desc
 "Lower_case_sub_assy_P/L_volvo_LH")
 (bom_reference_no 149)
 (issue_level "1")
 (Latch_location LEFT)
 (Customer [arvin_02613]))

([arvin_02545] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA10B02N")
 (product_desc
 "Lower_case_DL_sub_assy_volvo_RH")
 (bom_reference_no 150)
 (issue_level "1")
 (Latch_location RIGHT)
 (Customer [arvin_02613]))

([arvin_02546] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA11B02L")
 (product_desc
 "Lower_case_DL_sub_assy_volvo_LH")
 (bom_reference_no 151)
 (issue_level "1")
 (Latch_location LEFT)
 (Customer [arvin_02613]))

([arvin_02547] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA10B29Z")
 (product_desc
 "Lower_case_DL_sub_assy_ford_RH")
 (bom_reference_no 152)
 (issue_level "1")
 (Latch_location RIGHT)
 (Customer [arvin_02612]))

([arvin_02549] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA11B29B")
 (product_desc
 "Lower_case_DL_sub_assy_ford_LH")
 (bom_reference_no 153)
 (issue_level "1")
 (Latch_location LEFT)
 (Customer [arvin_02612]))

([arvin_02550] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA10B30R")
 (product_desc
 "Lower_case_DL_sub_assy_ford_RH")
 (bom_reference_no 154)
 (issue_level "1")
 (Latch_location RIGHT)
 (Customer [arvin_02612]))

([arvin_02551] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA11B30G")
 (product_desc
 "Lower_case_DL_sub_assy_ford_LH")
 (bom_reference_no 155)
 (issue_level "1")
 (Latch_location LEFT)
 (Customer [arvin_02612]))

([arvin_02552] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA10B33H")

(product_desc
 "Lowr_case_DL_sub_assy_mazda_RR_RH")
 (bom_reference_no 156)
 (issue_level "1")
 (Latch_location rear-right)
 (Customer [arvin_02611]))

([arvin_02553] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA11B33N")
 (product_desc
 "Lower_case_DL_sub_assy_mazda_RR_RH")
 (bom_reference_no 157)
 (issue_level "1")
 (Latch_location rear-left)
 (Customer [arvin_02611]))

([arvin_02554] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA10B34F")
 (product_desc
 "Lower_case_DL_sub_assy_mazda_FRT_RH")
 (bom_reference_no 158)
 (issue_level "1")
 (Latch_location front-right)
 (Customer [arvin_02611]))

([arvin_02555] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA11B34T")
 (product_desc
 "Lower_case_DL_sub_assy_mazda_FRT_RH")
 (bom_reference_no 159)
 (issue_level "1")
 (Latch_location front-left)
 (Customer [arvin_02611]))

([arvin_02556] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA10B31U")
 (product_desc
 "Lower_case_DL_sub_assy_manual_chil_lock_volv
 o_RR_RH")
 (bom_reference_no 160)
 (issue_level "1")
 (Latch_location rear-right)
 (Customer [arvin_02613])
 (latch_type
 [arvin_02627]
 [arvin_02619]))

([arvin_02557] of Bill_of_Material_items
 (quantity 1)
 (bom_part_no "FA11B31Q")
 (product_desc
 "Lower_case_DL_sub_assy_manual_child_lock_volv
 o_RR_LH")
 (bom_reference_no 161)
 (issue_level "1")
 (Latch_location rear-left)
 (Customer [arvin_02613])
 (latch_type
 [arvin_02627]
 [arvin_02619]))

([arvin_02558] of Bill_of_Material_items
(quantity 1)
(bom_part_no "FA10B04B")
(product_desc
"Lower_case_DL_sub_assy_child_lock_volvo_RR_
RH")

(bom_reference_no 162)
(issue_level "1")
(Latch_location rear-right)
(Customer [arvin_02613])
(latch_type [arvin_02627]))

([arvin_02559] of Bill_of_Material_items
(quantity 1)
(bom_part_no "FA11B04F")
(product_desc
"Lower_case_DL_sub_assy_child_lock_volvo_RR_
LH")

(bom_reference_no 163)
(issue_level "1")
(Latch_location rear-left)
(Customer [arvin_02613])
(latch_type [arvin_02627]))

([arvin_02560] of Bill_of_Material_items
(quantity 1)
(bom_part_no "FA06Z40Z")
(prototype_supplier [arvin_01519])
(product_desc "Lock_motor")
(bom_reference_no 164)
(BoM_Material [arvin_01518])
(production_supplier [arvin_01519])
(BoM_weight 39.7)
(issue_level "2"))

([arvin_02561] of Bill_of_Material_items
(quantity 1)
(bom_part_no "GA07N00E")
(product_desc "Worm_wheel")
(bom_reference_no 165)
(BoM_Material [arvin_01520])
(production_supplier [arvin_01506])
(issue_level "A"))

([arvin_02562] of Bill_of_Material_items
(quantity 1)
(bom_part_no "FA06G34H")
(product_desc "Worm_wheel")
(bom_reference_no 166)
(BoM_Material [arvin_01497])
(production_supplier [arvin_01506])
(BoM_weight 2.2)
(issue_level "4"))

([arvin_02563] of Bill_of_Material_items
(quantity 1)
(bom_part_no "GA01G03A")
(prototype_supplier [arvin_01522])
(product_desc "Stop_lever")
(bom_reference_no 167)
(BoM_Material [arvin_01521])
(production_supplier [arvin_01506])
(issue_level "C"))

([arvin_02564] of Bill_of_Material_items
(bom_part_no "330389")
(product_desc "Lever_holder")
(bom_reference_no 168)
(issue_level "K"))

([arvin_02565] of Bill_of_Material_items
(quantity 1)
(bom_part_no "FA06G03N")
(prototype_supplier [arvin_01505])
(product_desc "Locking_drive_lever_RH")
(bom_reference_no 169)
(BoM_Material [arvin_01510])
(production_supplier [arvin_01506])
(BoM_weight 1.1)
(issue_level "5")
(Latch_location RIGHT))

([arvin_02566] of Bill_of_Material_items
(quantity 1)
(bom_part_no "FA07G03L")
(prototype_supplier [arvin_01505])
(product_desc "Locking_drive_lever_LH")
(bom_reference_no 170)
(BoM_Material [arvin_01510])
(production_supplier [arvin_01506])
(BoM_weight 1.1)
(issue_level "5")
(Latch_location LEFT))

([arvin_02567] of Bill_of_Material_items
(quantity 1)
(bom_part_no "FA06G30M")
(prototype_supplier [arvin_01505])
(product_desc "Locking_lever_RH")
(bom_reference_no 171)
(BoM_Material [arvin_01509])
(production_supplier [arvin_01506])
(BoM_weight 6.0)
(issue_level "9")
(Latch_location RIGHT))

([arvin_02568] of Bill_of_Material_items
(quantity 1)
(bom_part_no "FA07G33E")
(prototype_supplier [arvin_01505])
(product_desc "Locking_lever_LH")
(bom_reference_no 172)
(BoM_Material [arvin_01509])
(production_supplier [arvin_01506])
(BoM_weight 6.0)
(issue_level "9")
(Latch_location LEFT))

([arvin_02569] of Bill_of_Material_items
(quantity 1)
(bom_part_no "FA06G36Z")
(prototype_supplier [arvin_01505])
(product_desc "Locking_lever_CDL_RH")
(bom_reference_no 173)
(production_supplier [arvin_01506])
(issue_level "2")
(Latch_location RIGHT))