Manufacturing system engineering ontology model for global extended project team

Manufacturing System Engineering Ontology Model For Global Extended Project Team

by
Hsiao-Kang Lin

A Doctoral Thesis

Submitted in partial fulfilment of the requirements for the award of
The Degree of Doctor of Philosophy of the Loughborough University

October 2004

Dedicated to my parents,

Your love, sacrifices and supports made this possible.

Particularly,

In Loving memory of My Father – Chiu Chiang Lin.

Acknowledgements

I would like to take this opportunity to express my sincere thanks a few of many people who help and support I have received from my friends, colleagues and family in the process of this research.

I owe my deepest gratitude to Dr. Jenny Harding who have provide both academic and expertise advice and guidance enable this research to be carried out efficiently, but also encouragement through the Ups-and-Downs of emotions that are associated with a PhD.

My special thank is due to Dr. P.C. Teoh of Motorola technology, Malaysia, who provided and coordinated the case studies on which the work is based, but more importantly, I value the friendship of you and your family.

I wish to thank the Engineering and Physical Sciences Research Council (EPSRC) and the Wolfson School of Loughborough University, UK in support of studentship to undertake this research. Especially convey my appreciation to all my friends in the Product Realisation Technologies Research Group and Manufacturing Technology Research Group at Wolfson School who have helped me in many ways.

Finally, my heart felt thanks go to my brother Jyh-Horng for his love and support my family in Taiwan, particularly take care my long term illness father who always has confidence in me to pursue a PhD.

Abstract

Communication, knowledge sharing and awareness of available expertise between multi-discipline project teams are complex issues. Complexity increases substantially in Extended Enterprises (EEs) / Virtual Enterprises (VEs) environments. The concepts of a Manufacturing System Engineering (MSE) moderator have previously been explored to facilitate and improve concurrent engineering design by enhancing the degree of awareness, cooperation, and coordination among engineering team members who are using shared information models and vocabularies. These concepts are now extended and adapted to the realm of EEs / VEs where inevitably individual partners will have their own terminology and information sources and may face problems and misunderstanding when different terminologies are used by particular team members.

This thesis is motivated by the achievement of ontology approaches to provide common underlying standardized meta-models for semantic and syntactic interoperability. Much research has been carried out and many commercial tools have been introduced to enhance the method. However, research related to Ontologies and ontology modelling has primarily concentrated on the collection of terms and definitions relevant to general business enterprises, and therefore has not focused specifically on the Manufacturing System Engineering (MSE) domain.

This thesis illustrates that semantic interoperability, can be achieved through an MSE Ontology Model which is proposed to enable the operation of an Extended Project Team MSE Moderator (EEMSEM). An EEMSEM framework for ontology acquisition, ontology mapping, knowledge collection, reuse, maintenance and moderation has also been illustrated. The proposed MSE Ontology Model has been designed using the semantic web technologies, Resource Description Framework (RDF), RDF Schema and Web Ontology Language (OWL), and verified with case studies to demonstrate that a common ontology approach and an integrated knowledge-sharing framework have potential for exploitation in new EEMSEM applications.

Table of Contents

Glossary of Terms

CAD	Computer Aided Design
CAE	Computer Aided Engineering
CAM	Computer Aided Manufacture
CSCW	Computer-Supported Cooperative Work
DAML+OIL	DARPA Agent Markup Language + Ontology Inference Layer
EE	Extended Enterprise
EEMSEM	Extended Projects Team Manufacturing Systems Engineering Moderator
HTTP	Hypertext Transfer Protocol
ICT	Information and Communication Technologies
MOSES	Model Oriented Simultaneous Engineering Systems
MISSION	Modelling and Simulation Environments for Design, Planning and Operation of Globally Distributed Enterprise
MSE	Manufacturing Systems Engineering
MSEM	Manufacturing Systems Engineering Moderator
OWL	Web Ontology Language
PDM	Product Data Management
PSL	Process Specification Language – ISO/CD 18629
RDF	Resource Description Framework
RDFs	Resource Description Framework Schema
SOAP	Simple Object Access Protocol
STEP	Standard for the Exchange of Product Model Data – ISO 10303
TCP/IP	Transport Control Protocol / Internet Protocol
VE	Virtual Enterprise
XML	Extensible Markup Language
XSL	Extensible Stylesheet Language
XSLT	Extensible Stylesheet Transformation Language

Chapter 1 : Introduction

The volatility and rapid change in technology and business environments require that organizations continuously adapt and adjust their structure and processes to remain competitive. Manufacturing Organizations, such as USA National Research Council and USA National Science Foundation, have introduced strategies and working methods intended to promote the adoption of Enterprise Integration. Intensive partnerships between many enterprises demand sophisticated logistic chain management that leads to the concept of the "Extended Enterprise (EE)" [USA National Research Council and Committee on Visionary Manufacturing Challenges 1998; Jordan and Michel 2000]. Increasingly an enterprise may be part of several logistics' partnerships, which constitute together a complicated "Virtuality" network and form of Virtual Enterprises (VE) [Goranson 1999; Camarinha-Matos et al. 2000].

The EE / VE approaches have been widely applied as part of many manufacturing enterprises' business strategy. The global webs of supply chain offer an alternative tactic to gain competitive advantage, to exploit market opportunities and to outsource external competencies as they occur. This globally distributed inter-enterprise teamwork requires integration approaches. Integration is about coordination processes and sharing information across the logistic chain.

Coordination integration is the redeployment of decision rights, work resources and process interaction within the inter-manufacturing-enterprises teams. The process of designing a globally distributed manufacturing inter-connection requires new Information Technology (IT) and approaches to support its operations across a number of collaborative organizations. The concept of the moderator was designed to facilitate and improve Concurrent Engineering (CE) design by enhancing the degree of awareness, cooperation, and coordination among engineering team members and it has been previously researched and explored in major research projects [MOSES 1992-1995; MISSION 1998-2001]. The Manufacturing System Engineering (MSE) process is further complicated when extended project team members come from an EE / VE, where several companies may have been brought together for a relatively short period of time, and different individuals within the team may communicate using different terminologies. An MSE Moderator which supports this type of team working in MSE design within an EE / VE environment is now proposed, and this has been called the Extended Project Team Manufacturing System Engineering Moderator (EEMSEM). The central objective of the EEMSEM is to improve and facilitate information management so that knowledge and information from multiple internal and external resources can automatically be integrated.

Information integration refers to the sharing and exchanging of information and knowledge among partners. In all types of communication, the ability to share information is often hindered because the meaning of information can be substantially affected by the context in which it is viewed and interpreted. This is especially true in manufacturing because of the growing complexity of manufacturing information and the increasing amount of knowledge and information that needs to be shared and exchanged between companies. Manufacturing extended project teams may face further problems when different terminologies are used by particular team members. In order to make design knowledge effectively accessible across EE / VE team members, the knowledge needs to be explicit in a well-defined terminology acceptable to all participating engineers. An approach for doing this, based on an MSE Ontology Model, is proposed in this thesis.

In this research,

"An MSE ontology is proposed to enable the operation of an Extended Project Team MSE Moderator (EEMSEM), to provide common understanding of manufacturing-related terms, and therefore to enhance the semantic interoperability and reuse of knowledge resources within globally extended / virtual manufacturing enterprises."

Chapter 2 : Scope Of Research

This chapter describes the scope of the research reported in this thesis. It establishes the aims and objectives of the research and provides an overview of the scopes, the focus and limitations of the research undertaken by the author. It ascertains the research novelty, addresses the contribution to knowledge, and outlines the structure of the thesis.

2.1 Research Aims And Objectives

The overall aim of this research is:

To determine how the existing moderator concepts might be extended or modified to make them applicable to Extended Enterprises and Virtual Enterprise environments.

This aim has been achieved by analysing and developing a common understanding of manufacturing-related terms, and thereby enhancing the semantic interoperability of a MSE Ontology meta-model for an EEMSE Moderator that stores manufacturing engineering knowledge and provides services for the coordination of shared and interchanged design knowledge among EE / VE team members. In order to achieve the research goal and to attain the expected results, five main objectives have therefore been undertaken:

I. Identify methods and technical solutions to enhance coordination and information integration between partners within inter-enterprises project team,

II. Develop a generic MSE Ontology model that accommodates an EEMSE Moderator,

III. Propose a suitable framework for EEMSE Moderator knowledge collection and moderation design,

IV. Evaluate the experimental MSE ontology model within the EEMSE Moderator context via a number of test cases based on relevant industrial data,

V. Consolidate the lessons learned as the basis for future development.

2.2 Research Novelties And Contribution To Knowledge

In the literature, manufacturing system information models, such as CIMOSA [Kosanke et al. 1999], MOSES [Ellis et al. 1994; Molina and Bell 1999], FDM [Harding and Yu 1999] and MISSION [Harding et al. 2003], describe the structure and relationships of data and information elements within manufacturing enterprise information systems. However, these models have mainly been developed for intra-enterprise integration. Where users may be expected to share the same terminology which is associated with information objects within the shared information models. To extend the operational scope to extended / virtual enterprise environments, research projects, including the Enterprise Project [Uschold et al. 1998] and the TOVE project [Fox and Gruninger 1997], have focused on the concepts of ontology for developing a taxonomy and have defined an explicit specification of conceptualisation for virtual enterprise modelling. However, these virtual enterprise ontologies have put effort into the collection of terms and definitions relevant to general business enterprises, and are not focused specifically on the manufacturing system domain.

The Process Specification Language (PSL) project [ISO/CD18629 2002] tries to develop a general ontology for representing manufacturing processes for the exchange of process information. PSL creates a neutral, standard language for process specification to integrate multiple process-related applications throughout the manufacturing life cycle. In a similar manner to PSL, the Standard for the Exchange of Product Data (STEP) effort aims to create an interlingua for exchanging manufacturing product data. Hence both PSL and STEP are focused on particular areas of manufacturing systems and therefore do not cover all the terminology aspects and needs that are necessary for the introduction of an EEMSE Moderator.

The author considered that it may be possible to implement a MSE Ontology Model, based on ontology approach and semantic web technology, to provide common understanding of manufacturing concepts and terms to make design knowledge effectively accessible across EE / VE team members. The model would need to be analysed and designed to comply with the needs of EEMSE Moderator. It therefore needed to bridge across multiple functional areas and meet the requirements for information semantic and syntactic integration between different MSE applications. The author believes that this is a novel area of research.

The main contribution of the research lies in the development of a new methodology for manufacturing information models within EE / VE environments. The new MSE Ontology Model, described in chapter 7, extends both the functionality and the information sharing and exchange capability for an EEMSE Moderator. Further major contributions are made by the EEMSE Moderator design, which includes the Ontology Acquisition Module and the Ontology Mapping Module that extend the functionality of the earlier generations of moderators. These are described in chapter 8 of this thesis.

2.3 The Overview Scope Of The Research

This research concentrates mainly on the ontology model within the MSE domain, in order to achieve information interoperation for an EEMSE Moderator within the inter-enterprises environment. The MSE ontology model may involve simple logical reasoning for semantic and syntax mapping. However, the EEMSE Moderator framework will be limited to a knowledge-based approach for the extraction of useful information based on the established knowledge. It will not include artificial intelligence (AI) branches that involve the discovery of new knowledge, such as data mining and machine learning. The scope of this research is listed and briefly described in the next 6 sections.

2.3.1 Literature Review Of Concepts And Structures Of Moderators (Chapter 3)

The complexity of product design and MSE processes generally requires expert contributions from many different disciplines within project teams. This section presents the key elements of moderators, which are support tools for inherently interdisciplinary design project team work. The literature review of concepts and structures of moderators covers two major research projects, and the MOSES engineering moderator and the MISSION MSE moderator, are addressed and discussed. Further challenges have been introduced by the need for manufacturing systems to be engineered, or re-engineered by Extended Project Teams, which take place in an EE or VE environment. The MSE moderator concept is therefore extended and aims to provide inter-enterprise knowledge and information exchange in MSE design through a new type of moderator called the Extended Project Team MSE Moderator (EEMSEM).

2.3.2 Literature Review From IT Requirements For Extended Project Team MSE Moderator (EEMSEM) (Chapter 4)

In recent years, VE / EE approaches have developed into critical success factors for corporations. Information Technology (IT) plays a vital role as the enabler of several functionalities of respective solutions. Surprisingly in this important area comprehensive IT support still does not exist to facilitate inter-enterprise teamwork. In this context, this section presents key elements of extended projects teamwork which needs to be supported by appropriate IT solutions. Literature reviews stress the essence and importance of EE / VE and the technologies for supporting interdisciplinary multi-enterprises integration are discussed. In particular, the present issues in the area of information interoperability through new ICT for the EEMSEM have been investigated. These reviews enabled areas of possible contribution for this research to be identified and potential applications and suitable methods for improving and reducing the barriers to inter-enterprise operations were found.

2.3.3 Literature Review Of Ontologies Approach (Chapter 5)

Since this research aims to enhance information integration in the EE / VE environment by providing a common semantic meta-data model, it was necessary to thoroughly review the current state-of-the-art in ontology approaches. Research into the basic concepts of ontology theory, ontology representation languages, semantic web technologies (RDF, RDF Schema, and OWL) and current ontology application areas in information search, semantic web, information integration, and knowledge management will therefore be discussed in this chapter.

2.3.4 Research Into IT Solutions For Integration In Manufacturing Systems (Chapter 6)

Various integration approaches in manufacturing systems, architectures and applications processes have been identified which show valuable contributions towards supporting inter-enterprise team working. Each application or system is considered individually so that particular strengths or weaknesses, in the current context, may best be explored.

2.3.5 The Manufacturing System Engineering (MSE) Ontology Model: Analysis, Design, Specification And Implementation (Chapter 7)

This research has explored the analysis, design, and development of a novel MSE Ontology Model, which provides a common ontology model for improving the semantic and syntactic interoperability between different MSE applications. The MSE Ontology Model specifies a range of classes and properties based on the emerging semantic web technologies, such as RDF, RDF-Schema, and OWL. In order to assess the validity of the MSE Ontology Model, a number of case studies have been designed and implemented using the protégé ontology editor tool and its plugins.

2.3.6 Architecture of The EEMSE Moderator Prototype (Chapter 8)

The architecture for formation of the EEMSE Moderator prototype which includes four major modules: *Ontology Acquisition Module, Ontology Mapping Module, Knowledge Acquisition Module* and *Design Moderation Module* has been illustrated. This architecture is also demonstrated in conflict moderation work between the EE / VE partners' software agents through an e-purchasing case example.

2.4 Structure Of The Thesis

The structure of this thesis is divided into four sections, as show in figure 2.1:

I. Background

The background to the research is comprised of 2 chapters. Chapter 1 is the main introduction to the thesis, presenting the background to the research area, and identifying the research problem. The principle aims, objectives, scope of research, the research novelty, the contribution to knowledge and the structure of the thesis are outlined in chapter 2.

II. Theoretical and Research Review

The theoretical / research review is comprised of 4 chapters. A literature review was first carried out on the concept and structure of moderators, such as MOSES Engineering Moderator, MISSION MSE Moderator, and the proposed Extended Projects Team MSE Moderator (EEMSEM) in chapter 3. Semantic interoperability requirements from IT to support the EEMSEM are discussed in chapter 4, to identify the relevant research area. After establishing the research topic, the research review was continued to find possible application areas and suitable methods. These were established based on the review in chapter 5, Ontologies approach for information integration. Finally, in chapter 6, the completed and current European research projects in IT solutions for integration in manufacturing systems were discussed.

III. Experimental Research

The experimental research is discussed in chapter 7 and chapter 8 where the MSE Ontology model has been analysed, designed and tested through a series of case studies, following the design of the EEMSE Moderator framework within a selected industrial environment.

IV. Research Conclusions

The research conclusions are presented in chapter 9 which provides an overview of the research, summarizes the novelty of the work and provides a concluding discussion with relating to further work.

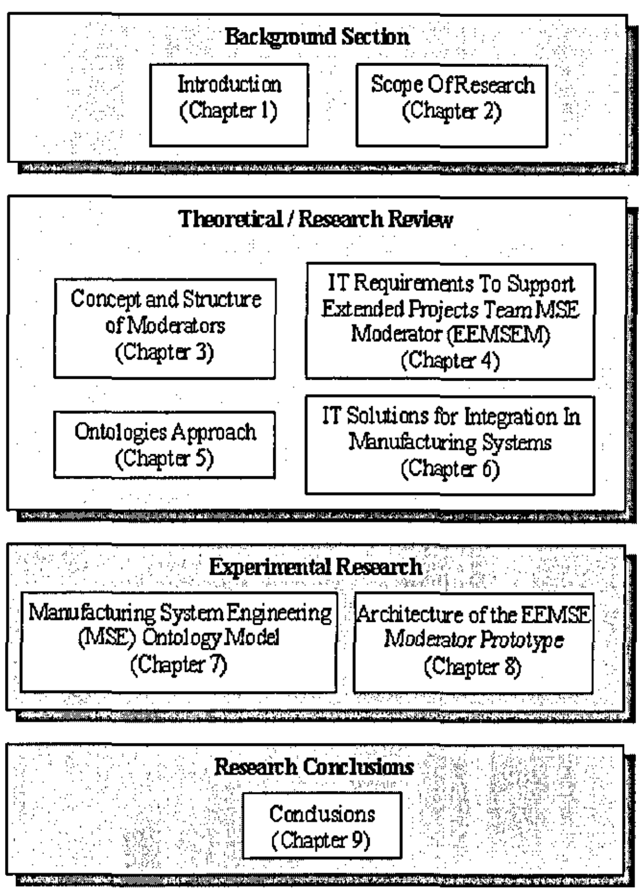


Figure 2.1: The Structure Of The Chapters Within The Thesis

Chapter 3 : The Concept And Structure Of Moderators

This chapter presents the key elements of Moderators. The Moderator concept was first proposed in the MOSES research project as a support tool for design project teams. It was coordinating software for Concurrent Engineering (CE) design, to raise awareness among the inter-working cross disciplinary participants that exist and need to co-operate in modern day engineering teams.

A literature review and discussion of concepts and structures of moderators in two major research projects, the EPSRC funded MOSES engineering moderator and IMS / ESPRIT funded MISSION MSE moderator is also given. In both the MOSES and the MISSION projects, the Moderators that were designed and implemented to work in environments where design teams used shared information models. In current manufacturing scenarios this is often not possible as increasing use is made of virtual or extended enterprises and supply chains. Therefore, at the end of this chapter a new type of Moderator, called an Extended Project Team MSE Moderator (EEMSEM), is proposed. A key aim of the EEMSEM is to provide inter-enterprise knowledge and information exchange in MSE design.

3.1 Moderators Concept

Typically the distributed CE team design project is seen to contain the multi-disciplines of specialist engineers with various different types of expertise. These may include, for example product development, process selection, equipment selection, project management, performance prediction (perhaps by simulation), and potentially many others. Hence the team brings together all the skills needed to develop a product design or manufacturing system to meet defined project and/or enterprise objectives.

In most inter-disciplinary CE team design activities, there will be periods when team members can work on their own individual contributions to the design and other times when several contributors will need to collaborate to achieve good acceptable compromise solutions. Hence both asynchronous and synchronous modes of working are required and inescapably there will be conflicts between the objectives of the different functions. Synchronous working is when two or more specialists (from different functions) within the design team co-ordinate their activities and work together on some aspect of the design. In contrast, in asynchronous working, team members will be working individually to contribute to the design. In order to achieve a balanced design, compromises have to be made to satisfy all the requirements and objectives of the project.

The real difficulty is when a change between asynchronous and synchronous activity should be made, but team members are not immediately aware that a point has been reached when compromises are required, since one or more aspect of the design is compromising other aspects. This problem is particularly difficult when the design team is highly distributed (see table 3.1).

High

Awareness

Time / Place	**"Asynchronous"** Different time	**"Synchronous"** Same time
" Co-located " Same Place	High awareness Low interaction	High awareness High interaction
" Distance" Different Place	Low awareness Low interaction	Low awareness High interaction

Low — Interaction — High

Table 3.1: The Level of Awareness and Interaction Between Different Types of CE Team Working

This is because if the project team is small and located in a single site, it is conceivable that team members can meet together regularly, exchange information and discuss progress. However, it is more complex when the team is part of a global distributed, extended / virtual organisation where team members are likely to be based in multiple distributed sites, possibly located across many countries.

Globally distributed manufacturing systems require the co-operation of different CE engineers working in a team, as well as using different software tools. The concepts and examples of Moderators (to support both Product Design and Manufacturing System Engineering) have been prototyped as coordinating software between different MSE design functions and these implementations have been previously reported in [Harding and Popplewell 1996; Popplewell and Harding 2001; Harding et al. 2003; Lin and Harding 2003].

The primary purposes of Moderators are to raise awareness, cooperation, and coordination among engineers in design team activities. The role of a Moderator is to identify each occurrence of a design conflict, and to orchestrate a dialogue between the interested design functions until the conflict is resolved. However, the Moderator should not be expected to solve design problems independently, as it is not an expert in any of the individual design functions, and it is also not an engineering arbitrator, and therefore does not automatically generate compromise solutions to design problems. A Moderator is included in CAE systems to raise the awareness of human designers within the CE team of how their decisions may affect, or be affected by actions of other team members. In this way it supports and empowers the human designer [Harding and Popplewell 1996].

There are growing demands for core competencies to be moved from large, UK single company design and manufacture scenarios, and be distributed to multiple companies in the logistic chain for product development and production. This move exploits smaller, more specialist design and manufacture units and cheaper wage rates that exist worldwide [Dwyer 2004]. Manufacturing systems have therefore been moving from distributed manufacturing and global manufacturing towards cross globally distributed organization manufacturing, using inter-connected systems. Additionally, in the second part of the1990s the Internet has been increasingly used as a communication backbone of the manufacturing industry. Through Web browsers and other standard tools, design and planning communication and information exchange take place.

The concepts of moderators therefore need to be evolved along with the manufacturing system and information evolution, as shown in figure 3.1. An EEMSE Moderator will be introduced and the IT requirements of information interoperability for an EEMSE moderator are discussed at the end of this chapter, and in subsequent chapters of this thesis. However, this discussion must be set in the context of previous research that has been carried out on moderators. Therefore, the structures and requirements of each generation of moderators are now described in the following sections.

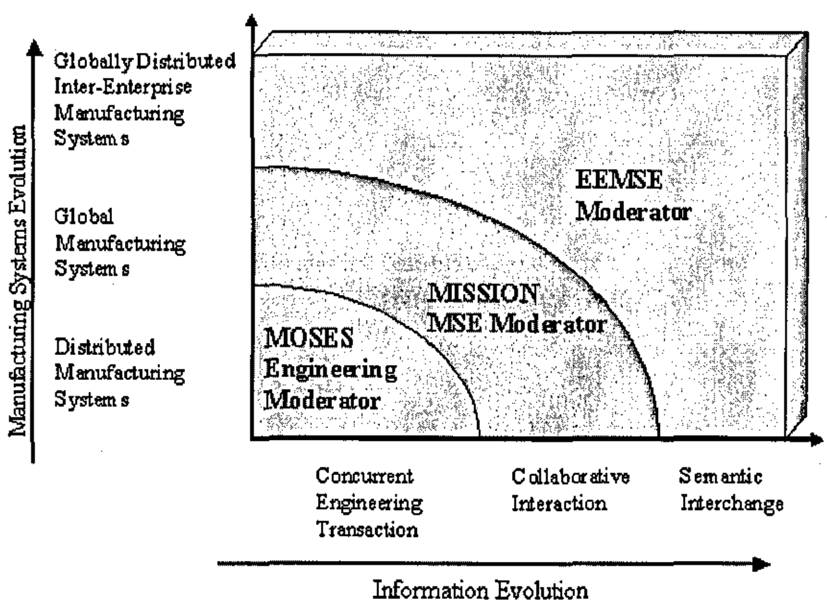


Figure 3.1: The Evolution of Manufacturing Engineering Moderator

3.2 The MOSES Architecture For Engineering Moderator (EM)

MOSES http://leva.leeds.ac.uk/www_moses/moses.html

Model Oriented Simultaneous Engineering Systems (EPSRC) (1992-1995)

The MOSES project was a joint EPSRC research project undertaken at the Department of Manufacturing Engineering of Loughborough University and the

School of Mechanical Engineering of University of Leeds. This research focused on a Computer Aided Engineering (CAE) system to provide product and manufacturing information, enable decision support based on these information sources and coordinate design activities in a manner that makes it suitable for operation in a distributed Concurrent Engineering (CE) environment.

In order to make effective use of information supported systems in design and manufacture, [Ellis et al. 1994] suggested a flexible, structure with data integrity for CAE applications throughout separating the information content from the software applications that drive them. Later [Young et al. 1998] termed these applications as data model driven applications. The fact that the data model is separated from the applications makes any specific application easy to replace as long as the underlying information model is maintained. Hence, the concept was that all applications within the MOSES environment were "loosely coupled" enabling a "Data model driven" approach to be used for the MOSES research.

The MOSES architecture (figure 3.2) for CAE systems is based on the use of two information models, a Product Model and a Manufacturing Model, which can be accessed by an open set of application programs via an integration environment.

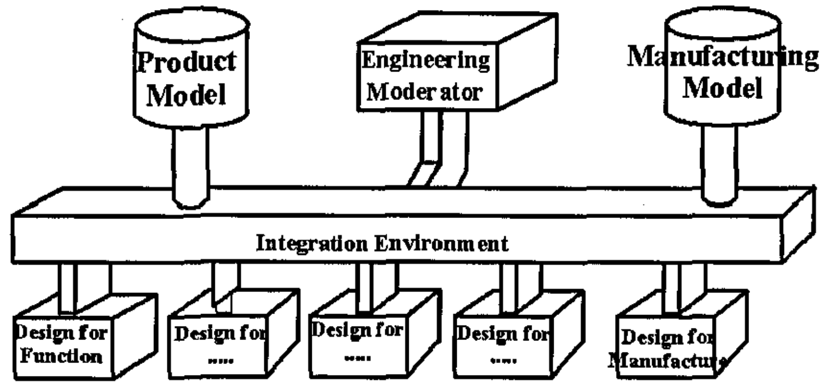


Figure 3.2: MOSES Architecture [MOSES 1992-1995]

The Product Model captures the information related to a product throughout its life cycle. The Manufacturing Model describes and captures the information about the manufacturing facility and capabilities at different levels of abstraction [Molina et al. 1994]. Manufacturing capability information modeling involves mainly how to represent a manufacturing facility through its manufacturing processes, resources, and the constraints imposed on them, their relationships and strategies. Design for Function and Design for Manufacture were the application areas particularly studied during the MOSES project and in figure 3.2, these are shown as just two of a range of possible Design for 'X' applications which may be utilized by the CE team.

A specialist manager or co-ordinating application was also included whose role was to drive concurrency within the MOSES system, and this was called the Engineering Moderator (EM). According to [Harding and Popplewell 1996], in order to identify and signal conflict in product design moderation within the MOSES system, the EM must be capable of performing the following duties:

- The EM is to promote communication and negotiation between design agents.
- The EM is to identify that a significant problem may have occurred in the design.
- The EM must determine the course of action to follow when a significant problem is identified.
- The EM must be able to maintain communication between interested agents until the conflict of interests has been resolved.

To facilitate execution of these duties requires the EM to be able to use certain types of knowledge, and to have access to particular types of information on which to apply its knowledge. The EM in the MOSES system included three main sections, Design Expert Knowledge, Knowledge Acquisition Module, and Design Moderation Module.

3.2.1 Design Expert Knowledge

The expertise within the CE project environment comes from many different disciplines and skills. [Harding 1996] therefore explains that the EM needs a mental model of each design expert (design agent), or knowledge of each type of expertise that is required within the CAE system. In the MOSES system, the design expert knowledge consisted of three main sections [Harding and Popplewell 1996]:

- Personal profile details. This section contains information to enable the EM to identify the agent. It would include the name, names or identifier by which the agent is known in the system. It could also include the type of agent, e.g. human or software.

- Main design criteria. The main content of this section is the knowledge of which design decisions are of relevance to the design agent. Alternatively, this could be thought of as which variables or parameters in the design are determined by, influence or constrain individual contributors within the design team. The knowledge is structured to enable the EM to decide whether or not the agent would be interested in the design step which has been taken and to assess the level of his interest. Therefore the knowledge is not structured to enable the EM to make design decisions from the agent's perspective, rather it is structured to enable the EM to decide whether the design agent should be consulted and whether the agent is likely to be able to identify any problems within the design, resulting from the change that has been made.

- Communication methods. This section contains any information required to enable the EM to communicate with the agent. For example, the location of agent, details of any translation programs required, etc.

It is important to note that the term "Design Agent" as used by Harding and Popplewell, refers to a contributor to the design with expertise in some relevant discipline. Hence a "design agent" in this context is not an autonomous software agent, but is most likely to be a combination of a human expert and a computer program, which can support the human expert's design activities, and access and update the current design which is stored in the shared product model.

3.2.2 Knowledge Acquisition Module

In order to perform the moderation duties as mentioned earlier, the EM must therefore retain and apply knowledge about the knowledge used by each of the design agents. Also the knowledge required by a design agent should be captured in whatever way best suits the design agent's requirements, since the best approach depends on specifics of the problem [Knaus and Jay 1990; Harding 1996]. Therefore, the EM must be able to collect, or have access to, contact information and knowledge for each of the design agents.

The function of the Knowledge Acquisition Module is to enable the EM to update its knowledge relating to particular design agents or to add new agents to the EM's Design Expert Knowledge stores, within the CAE system. Clearly this knowledge needs to be modelled and stored and used by the EM.

In the MOSES project, the EM's knowledge also had to be compatible with the stored product and manufacturing models, which were implemented using Object Oriented Databases (OODB). The approach adopted for the design and implementation of the EM's knowledge was therefore to also use an OODB to store the Moderator's current knowledge so that it was easily accessible and reusable by the EM throughout the design process. This functionality required the use of the OODB as a knowledge base, which was processed using C++ programs and the required functionality of the knowledge base was achieved through the use of a Knowledge Representation Model (KRM) which was first introduced by [Harding 1996].

3.2.2.1 Knowledge Representation Model (KRM)

The hub of the EM is a KRM as this provides a foundation for the structure of the moderator and enables the construction of generic and re-usable knowledge for modelling the design expertise to be applied and stored within the CAE system. [Harding 1996] took an Object Oriented (OO) approach for modelling the KRM, enabling it to be compatible with the other elements of the MOSES CAE system. In the early 1990s, the OO approach gained significant attention for its advantages in handling complexity, modularity, encapsulation, reusability, extensibility and abstraction of real-world objects [Booch and Graham 1993; Yourdon 1994].

According to Harding (1996) the KRM concept enables software expertise to be represented by one or more expert modules, as shown in Figure 3.3. Each module can be associated with one or more *knowledge base objects*, an *inference engine object* and one or more *working memory objects* within a production system metaphor [Jackson 1990].

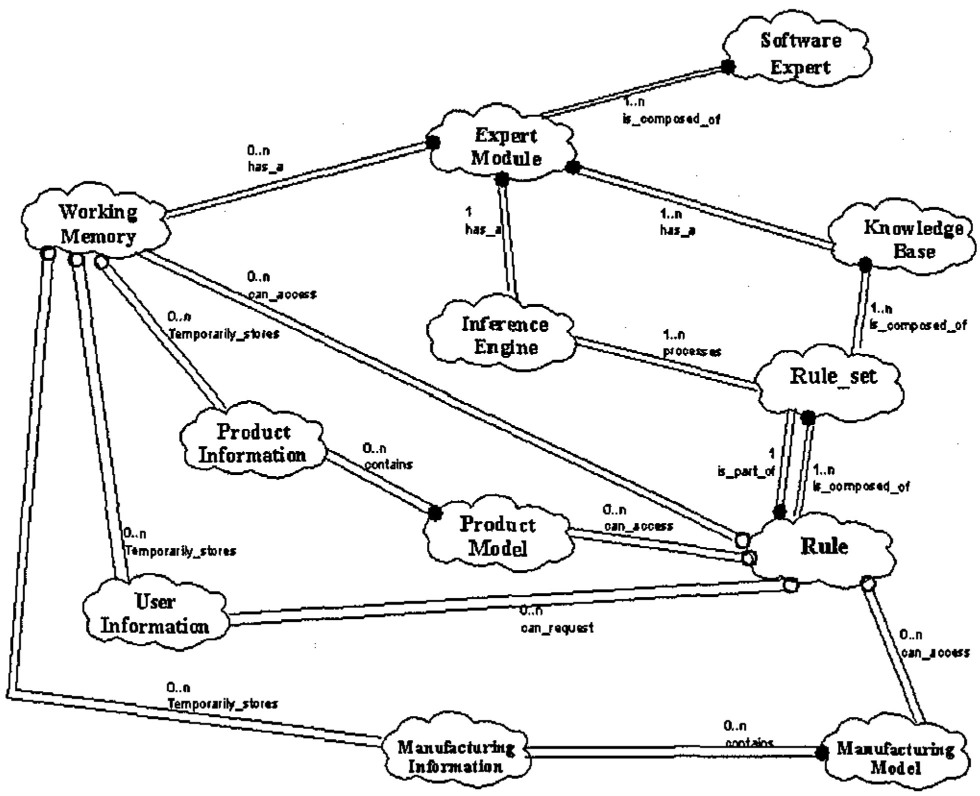


Figure 3.3: A Representation of Software Expertise Using Booch Object Oriented Design Graphical Notation. [Harding 1996]

The Knowledge base object contains knowledge of a particular type, or related to a specific type of expertise or domain. The Inference engine object carries out the processing of knowledge from one or more knowledge base objects. The Working Memory object is a store of variable information, which is possibly only of temporary value, to be used in association with the expert's domain knowledge, possibly to facilitate the processing of that knowledge. The production system metaphor is continued to allow storage of knowledge associated with any particular knowledge base object, through the definition of ruleset and rule objects. The details of KRM can be found in [Harding 1996].

The KRM was used to produce knowledge bases, by storing the objects as persistent objects within an object oriented database. It is however essential that a true object oriented database, which supports processing of object methods (or member functions), inheritance and polymorphism be used. In this way, the KRM concept makes use of database technology, whilst the KRM objects are able to collect and process information, generally by passing messages.

3.2.3 Design Moderation Module

This is used in the EM's normal mode of operation, and it enables the EM to moderate the current design. To all intents and purposes, this module embodies the EM's own expertise, as this module includes the EM's knowledge of how to carry out the tasks required in the design moderation process. For example the EM could detect the design change which has been made in the shared product model database, then carry out moderation on the design, by applying the knowledge it has about existing design agents, from its Design Expert Knowledge. The Design Expert Knowledge may be updated at any time, by using the Knowledge Acquisition Module.

The main sources of information used for this description of the MOSES project and the EM are[Harding 1996; Harding and Popplewell 1996], and the MOSES project web pages [MOSES 1992-1995].

3.3 The MISSION Architecture For Manufacturing System Engineering (MSE) Moderator

http://www.ims-mission.de/

Modelling and Simulation Environments for Design, Planning and Operation of Globally Distributed Enterprises (IMS 29656) 1998 - 2001

The MISSION project was an international project with partners in Europe, USA and Japan. It examined the process of designing the Manufacturing Systems (MS) which would span several sites and which were possibly globally distributed. The general goal of the research was to support of the Manufacturing System Engineering (MSE) process by integrating the appropriate software applications. This research investigated many applications of simulation and intelligent support systems within manufacturing system design and operation. An intelligent support application, the MSE Moderator, was developed at Coventry University and Loughborough University, UK. The primary function of the MSE Moderator was to support globally distributed MS design and enhance the degree of awareness, cooperation, and coordination between members of the CE team within the MISSION environment. The various activities performed by the MSE Moderator to achieve its main function are listed below:

- The MSE Moderator must know whenever a change is made to the MS design;
- The MSE Moderator must be able to identify when a design change may cause conflict[1];
- The MSE Moderator must communicate the detection of possible conflict to all MSE agents which it deems to have an interest in resolving the conflict, and when necessary remain in dialogue with these agents until resolution is achieved.

[1] In MISSION research, if a design change made by Designer A has implications, or causes problems for Designer B, it is said to cause ***conflict***.

Figure 3.4 shows the MISSION MSE Moderator includes two major programs to achieve its objectives. These were the Knowledge Acquisition Module and Design Moderation Module. Both of these modules also interact with multiple MSE Design Agent Modules (see figure 3.4), to support the activities described as above. Each Design Agent Module represents a contributor to the MSE process, three examples are shown, i.e. Project Agent, Supply Chain Agent and Simulation Manager.

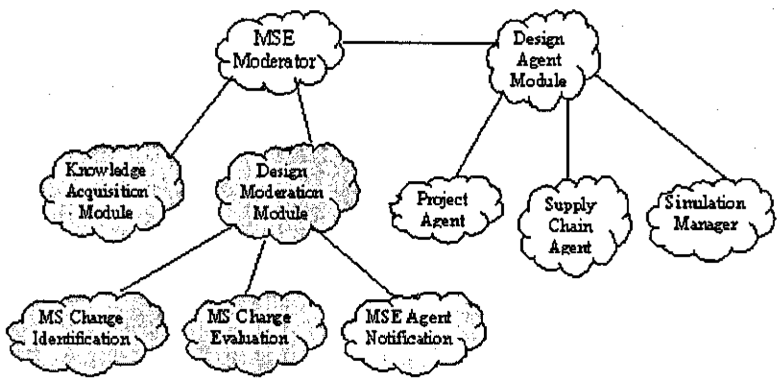


Figure 3.4: Structure Of Manufacturing Systems Engineering Moderator [Popplewell and Harding 2001]

3.3.1 MSE Design Agent Module

In the MISSION project, the term "MSE Agent" was used to refer to each combination of engineer(s) and supporting software performing an identifiable function to contribute to the developing MSE design. For example, the project management function may be fulfilled by a Project Agent which may be expected to include software tools to support both strategic management and project planning. The globally distributed developing design is shared between agents by a common communication platform. The platform includes communication protocols, information models and software tools. In this context the MISSION Modelling Platform (MMP) [Popplewell et al. 2001], shown in the figure 3.5, was implemented as an MSE Integration Infrastructure based on the HLA-RTI technology [McLean et al. 2000]

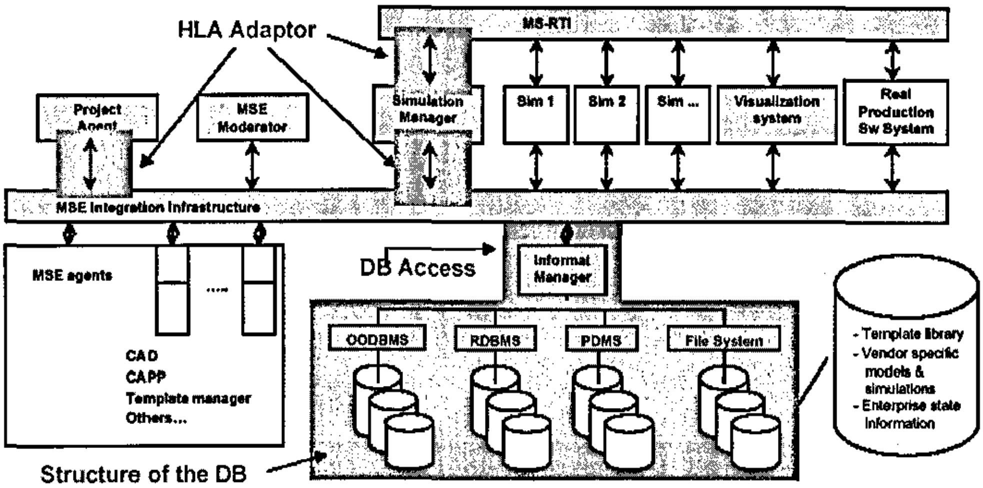


Figure 3.5: MISSION MMP Architecture [Popplewell et al. 2001]

In the prototype MMP, one of the applications developed during the project was an Information Manager (IM) Agent which contributed to the manufacturing system engineering (MSE) process by maintaining the information relating to the MS being designed. This was stored, accessed and shared in a common MSE database (implemented as an Oracle, object-relational database). The IM supported the information exchange between all the involved MSE Agents. Hence, in the MMP, whenever a change was recorded in the common project database, the MSEM could retrieve all the design information that it required to perform its design moderation duties, via the IM. Therefore, the moderator was able to know whenever a change was made to the MS design, and could moderate that change as required.

3.3.2 MSE Knowledge Acquisition Module

In the Mission project, the MSE Moderator was used to raise awareness within the MSE team of when problems were arising that required particular Design Agents to become involved. This was referred to as identifying when a design change may cause conflict. The moderator therefore needed to retain and apply knowledge about the knowledge used by each of the MSE Design Agents. Therefore, the MSE Moderator also had to be able to store, or have access to, contact information for each of the MSE Design Agents. This was available from the relevant Design Agent

Modules in the MSEM's knowledge base. This functionality was again achieved through the KRM concept and object oriented knowledge database, as in the MOSES project.

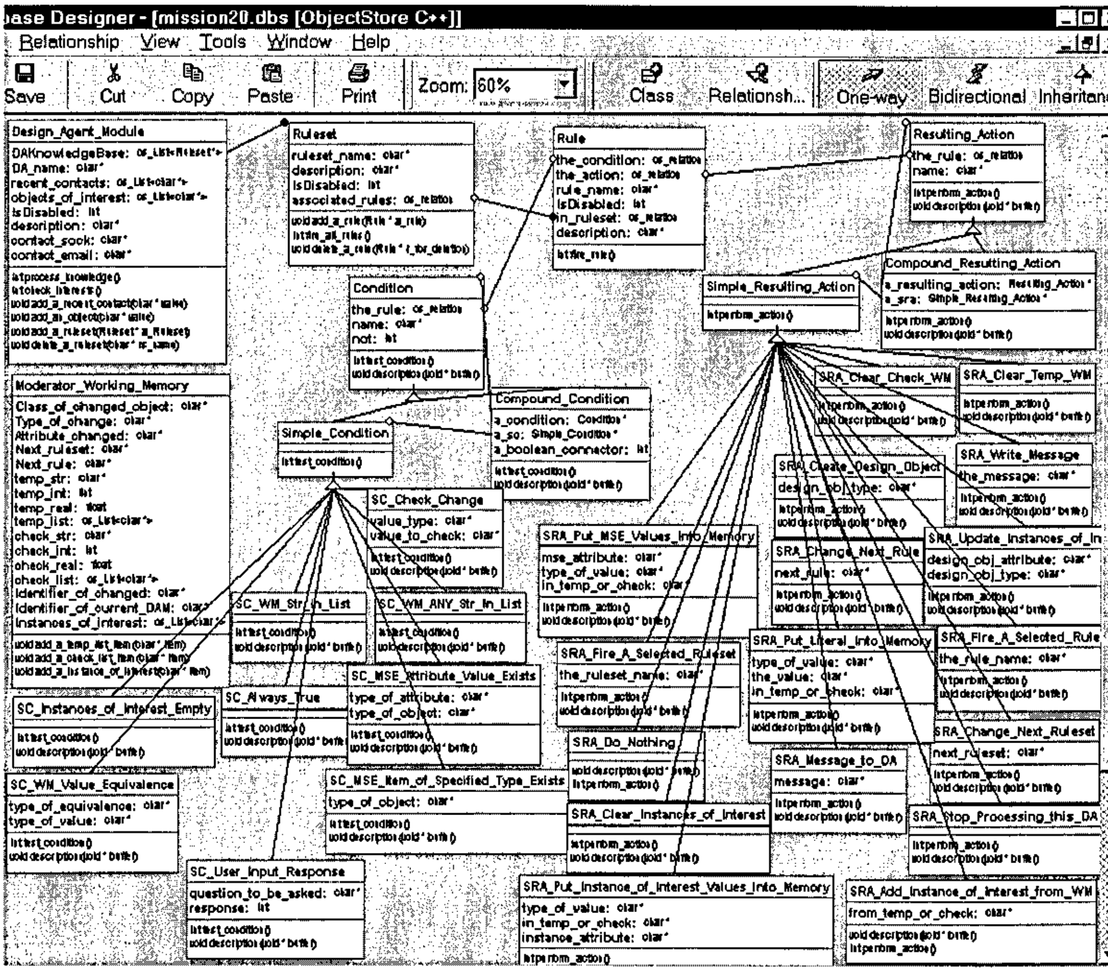


Figure 3.6: KRM Class Structure used to Implement Prototype MSE Moderator [Harding et al. 2003]

In the MISSION project, the MSEM's knowledge was stored in an Objectstore, object oriented database. The Design Agent Modules in the MSE Moderator used KRM objects to store knowledge about individual project members, and the knowledge about what changes are important to them, and what actions should be taken if such changes occur. Each Design Agent Module knew how to process its own knowledge, as this behaviour was implemented in methods of the class. The processing of knowledge was achieved by message passing between instances of various classes, including Ruleset, Rule, Condition and Action objects. Figure 3.6 shows the KRM class structure used to implement the prototype MISSION MSE Moderator.

The Knowledge Acquisition Module was therefore designed to collect, store and evaluate knowledge about what is important to individual design team members (MSE Agents). Once again, it was important that it be straightforward to modify this knowledge during the course of a project as team members may join, leave the project or the relevance of particular types of decision may also change. As a result, the Knowledge Acquisition Module, illustrated in figure 3.7, could be used to create, delete or modify MSE Agent Module objects, and their associated knowledge objects. The resulting objects were stored, as persistent objects in an object oriented knowledge database.

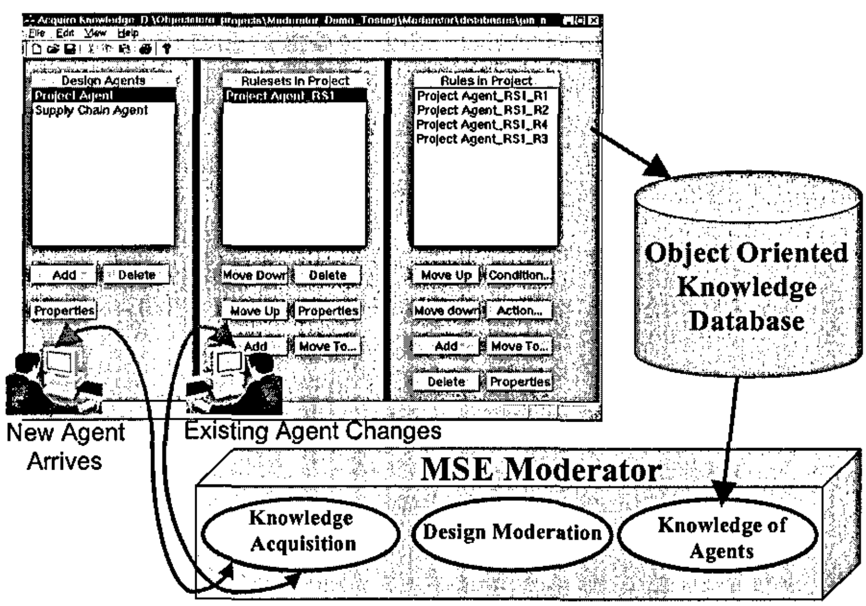


Figure 3.7: Knowledge Acquisition Module [Harding et al. 2003]

3.3.3 MSE Design Moderation Module

The shared MISSION information model, managed by the IM facilitated the communication activities in the MMP through the provision of communication message class objects which could be passed between the Moderator and MSE agents through the MSE infrastructure. The moderator also knew how to contact each MSE agent by email, or by notification directly to an appropriate personal computer. Hence the MSEM was able to communicate the detection of possible conflict to all MSE Design Agents who were needed to resolve the conflict. The MSEM should also remain in dialogue with these agents until resolution was achieved. The Design Moderation Module, shown in figure 3.8, was implemented to identify potential design conflicts and to perform moderation activities. The moderation process was activated, whenever a change was made to the information held in the shared MMP (Oracle) databases administered by the IM. When the MSEM had been notified of a change, it connected to its Object Oriented Knowledge Database (ObjectStore database), which contains the MSE Agent Modules and the MSEM's Working Memory Object. The Working Memory Object was used by the MSEM to keep track of changes made to the MSE design, and to record and manage its interactions with MSE Agent Modules.

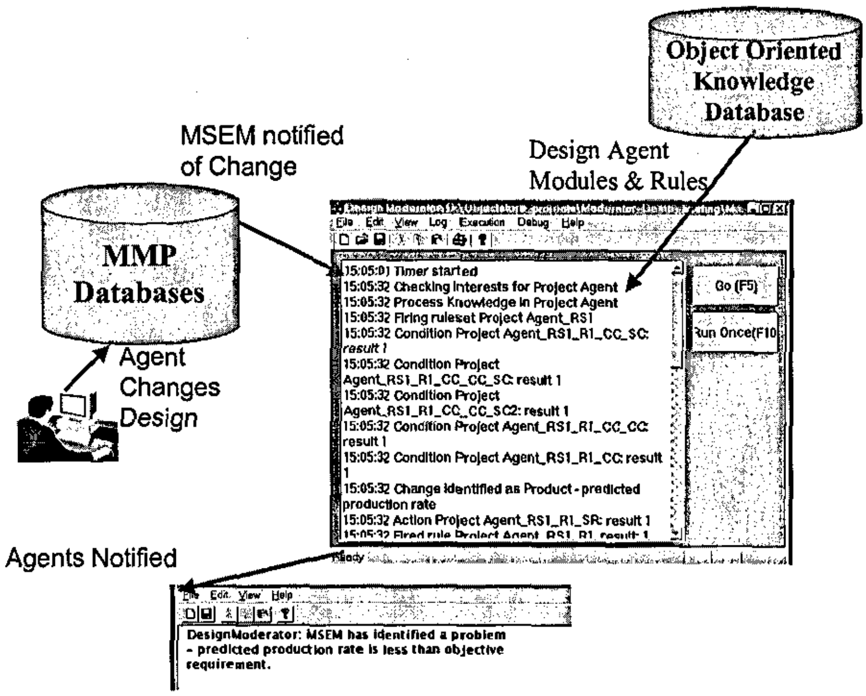


Figure 3.8: Design Moderation Module [Harding et al. 2003]

The MSEM checked the interests of all the MSE design agent modules in its knowledge base. If no design agents were interested in changes of that type, the moderation of the current change stopped (and the MSEM moved on the next change). However if the MSEM found that one or more design agents did have interest in the current type of change, a ***conflict*** had been identified, and the moderation process therefore proceeded and all the interested design agents were contacted by the MSEM. The interested design agents were then required to solve the conflict.

An example of this moderation process is given in the following explanation of figure 3.8. In the prototype MISSION system, the trace on the screen (see figure 3.8) shows the MSEM is activated when a change is made to the shared (Oracle) databases. When MSEM is informed that a change has taken place, it checks the knowledge database to see if any agents might be interested in changes to objects of that type. For example, the Project Agent has been identified as potentially being interested and as a result of processing its knowledge about the Project Agent. The information change may therefore effect the Project Agent and the MSE Moderator determines that a warning message should be sent to the Project Agent.

The main sources of information used for this description of the MISSION project and MSEM are [Harding et al. 2003], and the MISSION project web pages [MISSION 1998-2001].

3.4 The EEMSE Moderator (EEMSEM)

The MOSES Engineering Moderator and MISSION MSE Moderator were designed to support CE and assist the intra-enterprise distributed manufacturing system design team by identifying potential design conflicts. However, recently there is a need for manufacturing systems be designed, or re-designed by extended project teams. Therefore design is taking place in an EE / VE environment that can bring about both added value and additional complications to various aspects of engineering product development.

This EE /VE collaborative setting is a major challenge for the application of IT, as it typically leads to loosely structured, strongly decentralized organisation structures and at the same time weakly integrated IT environments in which information requirements are not likely to be known beforehand. Hence, shared information models and common understanding of terminology and vocabularies cannot be assumed. Therefore, to achieve high efficiency and quality of collaboration in each manufacturing system project, it is essential not only to consider the coordination and cooperation of teamwork, but also to support the communication processes within the project. For example, efficient integration of application tools, improved data exchange and sharing, common model repositories etc. This *inter-enterprise approach* is the baseline of the conceptual architecture of the proposed novel IT environment presented in this research. The EEMSEM is aimed at developing technologies for the next generation of such inter-enterprise collaboration tools. The IT requirements for the EEMSEM, especially focus on the information interoperability that will be described in the next chapter. Details of the design and prototype structure of EEMSEM are given in chapter 8.

Chapter 4 : IT Requirements For EEMSEM

Manufacturing System Engineering (MSE) with its increasing dependence on multi-disciplines, multi-departments, multi-enterprises, and multi-national contributors has moved towards Extended Projects Team MSE. These take place in an EE or VE environment, as a form of inter-enterprise collaborative working. Surprisingly in this important area, comprehensive information technology (IT) support still does not effectively facilitate inter-enterprises collaboration. The limitations of technologies, such as Electronic Data Interchange (EDI) and Enterprise Resource Planning systems (ERP), in meeting the demands of such complex, extended / virtual environments, have been reported [Dwyer 2004]. In this context, this chapter introduces literature reviews that stress the essence and importance of EE / VE and the technologies for supporting interdisciplinary multi-enterprise integration are discussed. In particular, the present issues in the area of information interoperability through new ICT for an EEMSEM have been investigated.

4.1 Extended Projects Team As A Form Of Inter-Enterprise Collaborative Working

Current project configurations and associated operations for product developments and services are delivered through complementary competence sharing between different project participants which may come from different organizations, possibly across different industrial sectors [Cutting-Decelle et al. 2003]. New trends are emerging and priorities are consequently changing the way organisations function and collaborate with each other. For example, transparency of information, inter-enterprise coordination, knowledge sharing and collaboration, and the increase in knowledge-intensive work, electronic business and globally distributed teamwork enabled by new communication technology. [Kazi and Hannus 2000] presented a distinct move towards inter-enterprise collaboration, as shown in Table 4.1.

From	To
Centralised planning	Transparency of information
Enterprise resource planning	Inter-enterprise coordination
Document management	Object management
In-house operative systems	Inter-enterprise collaborative system
Supply chain management	Demand change management
Workflow management	Group work support
Scheduling	Schedule synchronisation
Management information systems	Decision and negotiation support
Reporting	Forecasting and coordination
Electronic commerce	Elimination of ordering
Access control	Knowledge sharing
Integrated systems	Flexible interfaces

Table 4.1: Changing trends and priorities for inter-enterprise collaboration [Kazi and Hannus 2000]

The findings from Table 4.1 clearly point towards the operational paradigm of knowledge sharing inter-enterprise collaboration. So far, there is no unified definition for this paradigm and a number of different terms are even competing in the literature that either refers to the same concept or to its different perspectives. Among others, the terms: Extended Enterprise (EE), Virtual Enterprise (VE), Supply Chain Management (SCM), electronic commerce, cross border enterprise, network of enterprises, or virtual corporation, are commonly used. These terms, although not necessarily synonymous, represent related concepts [Camarinha-Matos et al. 1997].

A number of research projects, worldwide, are addressing different aspects of EE / VE, such as the [NATO-CALS; NGM; PRODNET-II 1996-1999; VEGA 1996-1999; NIIIP 1998; GLOBEMEN 2000-2003]. The concept can be found in the above literature, and many authors have elaborated it further, as shown in the following examples:

- An Extended Enterprise is the seamless integration of a group of companies and suppliers (industrial, educational, investment, and governmental) that collaborates to create and support a timely and cost-effective service or product, that responds to the customers' needs. [Jordan and Michel 2000].

- An agile virtual enterprise in terms of its various stages: opportunity identification, partner identification, formation, operation and reconfiguration / dissolution [Goranson 1999].

- Virtual Enterprise is a temporary alliance of enterprises that come together to share skills and resources in order to better respond to business opportunities and whose cooperation is supported by computer networks, challenges the way industrial production systems are planned and managed [Camarinha-Matos et al. 1997].

- Early efforts in the area of VE were strongly constrained by the need to design and develop horizontal infrastructures aimed at supporting the basic collaboration needs of consortia of enterprises. Current trends, however, are more and more directed to the development of new vertical business models. There is a shift towards business-to-business solutions, as a way to effectively enable E-commerce.[Camarinha-Matos et al. 2000]

- A key step towards achieving a VE is to create a set of standards and conventions that lets software automatically find partners, markets, and services as needed and then integrate them without prior agreement [Petrie and Bussler 2003].

In addition to these definitions, it has been said that the Virtual Enterprise concept has emerged as a more agile and responsive business model that is enabled by advanced ICTs network infrastructures. [King and K. Moon 1999] considered the outsourcing approach in Agile Virtual Enterprises (AVEs) is not just a buyer-vendor relationship, but also a quick response collaboration partnership. [Petrie and Bussler 2003] highlighted that one of the ideas driving VE creation is that of processes dynamically constructed out of available Internet-bases services as needed at runtime. Table 4.2 lists comparisons between Traditional Industrial Enterprise, Extended Enterprise and

Agile Virtual Enterprise, which has been coupled based on information from [Camarinha-Matos et al. 1997; King and K. Moon 1999; Kazi and Charoenngam 2003; Petrie and Bussler 2003].

	Agile Virtual Enterprise	**Extended Enterprise**	**Traditional Industrial Enterprise**
Market Strategy	Quick response and accurate response to market needs.	Partially customisation products.	Low degree level of responsiveness to emerging market requirements.
Collaboration Strategy	Collaboration through Temporary alliance	Collaboration through optimised relationship between the members of supply chain.	Collaboration through Long-term alliances.
Production	Focus on core competence and share knowledge with the temporary business partners to create value.	OEM is the leader of the chain, and often incorpora-tes its suppliers within the design and production planning to improve the production efficiency.	Mass production
Procurement	Consignment purchases. Long-term agreements for all material groups	Directly interfaced with ERP systems	MRP based procurements
IT Support	Fully decentralised and communicate through Internet, Extranet, and advance ICTs network infrastructure	Partially centralised, and using EDI to link and share information along it's own supply chain members.	Centralised decision with limited EDI.
Business Transaction	Electronic commerce for global integration across the entire supply-base.	Electronic commerce among qualification of key suppliers based on price, quality and history.	Buyer – Vendor transactions

Table 4.2: Comparison between Traditional, Extended and Agile Virtual Enterprise

4.2 Information Interoperability Requirements for EEMSEM

The extended projects team takes place in an EE or VE environment, which is a form of inter-enterprise collaborative working. When enterprises collaborate with each other, there is a need to have mechanisms to support collaborative work for dynamic, geographically and organizationally dispersed project teams. The Computer-Supported Cooperative Work (CSCW) technologies and modern ICT allow enterprises to work closely with each other even when operating in various industries worldwide. It is interesting to explore this further with regard to some core concepts involving the IT and may be best understood through a summary of the common characteristics of EE / VE shown as follows:

- Strategic temporary alliances cross organizational units,
- Some members are not known in advance,
- Communication between distributed locations enabled by ICT.
- Loosely couple network,
- Quickly reconfigured and short set-up times,
- Asynchronous information updating, autonomous repositories,
- Information integration through agreed standards, industry standards, and ontology,
- Interdisciplinary tasks, members may participate in several other concurrent EEs / VEs.

The interaction between EE / VE could be achieved by employing advanced technologies in communication and information exchange management. The use of EDI and ERP systems have been applied by major companies to exchange documents such as specifications, orders and invoices electronically with their suppliers and customers and the whole network in real time. Furthermore, excellent communications, a paperless paradise, incomparable order accuracy, noticeably reduced lead times and improved delivery scheduling as well as mandatory quality control requirements have been achieved.

However EDI, traditionally used by large organisations, is expensive, both in terms of applications and system running costs, requiring considerable transaction volumes before there is a financial payback. For smaller companies, running an EDI system is generally found to be too costly, raising an insurmountable barrier to supply chain extension. [Dwyer 2004] pointed out that EDI does not effectively transmit orders or schedules automatically to the whole tiers supply chain. This identification of the weakest link of IT as EDI is a major challenge to the global competitive advantage.

Recent information technology developments have emerged in a manner that permits moving from EDI to WWW E-commerce, making it more attractive for Small and Medium sized enterprises (SMEs) to use e-commerce for business to business (B2B) and business to customer (B2C) transactions. EDI functions for B2B transaction can now be moved to the lower cost new ICT, e.g. Internet platform, which provides the crucial turning point for e-commerce. The new ICT challenge is in the development of systems for information interoperability, from distributed database systems to global information systems.

Traditionally, while using ICT, collaboration has been primarily through simple document exchange between individuals. This point-to-point form of communication, as shown in figure 4.1(a), has both led to data / information redundancy and inconsistency. Additionally, collaboration knowledge is unorganised and not shared.

One solution to the problem of data / information redundancy and shared information has been through the introduction of the client / server approach [Berson 1992]. Here, information is stored in a central information repository that is accessible by the relevant information providers and users, as shown in figure 4.2(b). Client /server architectures also provide a set of remote services to several clients that are interconnected by distributed processing. According to [Ozsu and Valduriez 1999], architectures for distributed DBMS depend on at least three parameters: distribution, heterogeneity, and autonomy. With WWW and related Internet working technologies, there is no distribution, heterogeneity information management because anyone can put up a Web page and make data available on the Web.

Autonomous repositories that store different types of digital data in multiple formats are becoming available for use on fast evolving global information systems infrastructure. Organizations prefer to at times only release "partial" information, while keeping and maintaining the whole "internally". The way forward, as described by [Kazi and Hannus 2000], would be through flexible links between enterprise systems, as shown in figure 4.1(c). Here, an individual would communicate with the central repository of the enterprise, for which the individual is working, this would then release the relevant portion of this information to a shared project server. As such, enterprise specific systems / repositories would transfer and receive information packages on a periodic or per request basis to / from the VE specific project server.

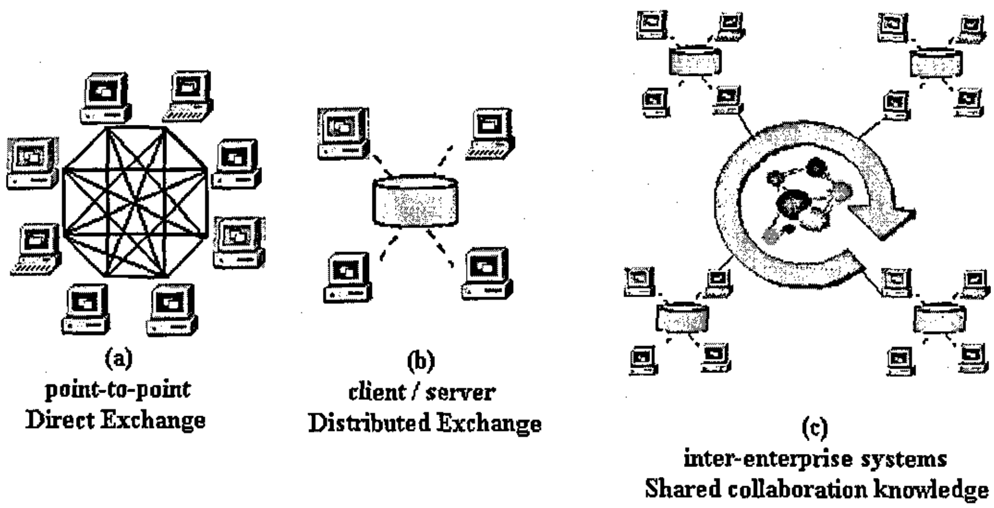


Figure 4.1: Inter-enterprise Information Exchange Mechanisms

However, the individual enterprise system and VE repository are unlikely to share a common data model. Information exchange across different data organizations, through data models, requires easy communication between the different enterprise systems. [Harding 1996] observed if two computers are to successfully communicate together, they must be able to understand each other. This may be achieved in many ways, e.g. by a common language and/or culture, effective translations, or use of common sources of information which may be individually accessed and comprehended by each. Information standardization, semantic ontologies and syntactic standardization have been generally considered as the common ground of information for data exchange among collaborative partners for inter-enterprise system integration [Lee et al. 1996; Stouffs and Krishnamurti 2001].

4.2.1 Information Standardization

Ideally, all enterprise systems would use a common language so that the common information model could be used on different software applications with no misunderstanding or interpretation required. However, people working within a particular company or team will inevitably develop their own vocabulary and terminology that they often work for particular purposes. This identifies the need for basic standard languages into which the various design and manufacture software can be converted to enable the models to be transferred between various software applications.

As a result, in order to resolve the information exchange problem, a standardization approach has been at the core of most research efforts. For example, technical standards for product information and CAD/CAM documents have been realized by efforts like Product Data Management and Product Lifecycle Management (PDM/PLM) and the Standard for the Exchange of Product Model Data - STEP [ISO 10303-1 1994]. Business documents standards for procurement applications have been defined by organization such as Commerce XML Resource [cXML], and XML Common Business Library [xCBL].

There are also some organizations and consortiums that develop and deploy standard electronic business interfaces to specific industries. These are already available such as Open Financial Exchange [OFX] for banking services, Health Level Seven [HL7] for health care, and RosettaNet [RosettaNet] in the area of information technology, electronic components, and semiconductor manufacturing working to create and implement industry-wide, open e-business process standards. These standards form a common e-business language, aligning processes between supply chain partners on a global basis.

A standardized terminology needs to be semantically consistent across organization boundaries, since the communication aspects of information require that communicating parties have the same understanding of the meaning of the exchanged information. This assumption is simple: if everyone adopts the same concepts, vocabulary, and language, any data expressed within this language will be accessible to everyone. However, recent practice has shown that establishing comprehensive and compatible standardized product data models can prove to be a long and complicated process.

According to [Turk 1998], the problems in the development of standardized, large-scale product data models are due to the difficulties of getting the interested parties to agree on a common representation and also to the incompleteness of the models. It is infinitely more difficult to design a global standard. [Kosanke and de Meer 2001] also consider that there are too many overlapping groups developing international standards independently using incompatible and inconsistent terminologies. Furthermore, [Stouffs and Krishnamurti 2001] question whether standardization will improve the design process through effective data exchange, or instead, would it hinder the process by imposing a specific language for designers to express their ideas and conceptualisations? They believe that whilst a standard vocabulary will enable all participants to effectively communicate and exchange data within the context of this standard, it will not support flexibility and extensibility from outside their design domain.

To tackle these issues a semantic representational model of conceptual entities and their relationships, called semantic ontologies, can encourage participants to express their design information, in their own ways, by providing support for exploring alternative design representations and providing support for mapping design information between representations. Hence, this is becoming an important area for information interoperability.

4.2.2 Semantic Ontologies

One of the common goals in developing ontologies is to share common understanding of the structure of information among people or software agents. Gruber provides widely quoted definitions of an ontology, as "an explicit specification of a conceptualisation" and "a specification of a representational vocabulary for a shared domain of discourse – definitions of classes, relations, functions, and other objects – is called an ontology" [Gruber 1993]. According to [Gruninger and Fox 1995], ontologies are a technique that is intended to provide an "easy to re-use" library of class objects for modelling the problems and domains. The ultimate goal of this approach is the construction of a library of ontolgies which can be reused and adapted to integrate the formalization of the underlying logical theories for specifying the semantics of object classes and relations in the ontology.

[Mena and Illarramendi 2001] considered that the information available in the different repositories should be described by semantic views in global information systems. Ontologies have been accepted as powerful description tools, and for this reason they are appropriate for playing the role of semantic views. [Lee et al. 1996] pointed out that there is a great deal of interest in the development of ontologies to facilitate knowledge sharing in general and database integration in particular.

For EE / VE in operation, information sharing and collaboration within their participants can typically be done through agreed EE / VE common standards with an ontologies approach that may provide semantic and syntactic mapping between an organization's information and the shared EE / VE standards. The detailed literature review of ontologies approach will be illustrated in chapter 5.

4.2.3 Syntactic Standardization

When considering ontology-based applications, inevitably the issue of data structuring syntax for presentation of conceptualisation will arise. An ontology must be encoded in some language to express the concepts in the domain in a manner that computers can manipulate meaningfully. Additionally, in order to facilitate the effective interoperation, a formal representational framework / syntax must be conceived.

The rapid rise in popularity of the Extended Markup Language (XML) provides web-friendly data structuring syntax for presentation. XML can be considered as a meta-language that serves to define markup languages for specific purposes. When project partners agree on tags, they can exchange data described in any markup language based on these tags, even when their own markup language differs in scope or composition. XML has the advantages that it is readable both by humans and by the computer. Markup languages based on XML can easily be adapted or extended to specific purposes or needs. In this way, XML allows for syntactic standardization, providing all participants with the ability to define or adopt their own data model, and consider ways of translating these different models between one another at later stages, using tools developed for this purpose [Stouffs and Krishnamurti 2001]. The Extensible Stylesheet Transformation Language (XSLT) is most generally used to transform an XML document from one form to another.

However, if every business uses its own XML definition for describing its data, and then the trading partners must transform their data to a common XML data format to be able to communicate with each other. Hence, the approach is inadequate to achieve real interoperability. Additionally, XML provides semantic information as a by-product of the structure of the document.

e.g.

```
<Desk>
    <Length> 29 </Length>
</Desk>
```

Tags define the semantics of the data. That is, structure and semantics of document are interconnected. Without "tag-centric" syntax, e.g. 29 could be the length or weight or height of a desk or something else, XML is deficient to express semantics for description data. XML provides a common syntax for data interchange, but XML does not define the meaning of the information.

Recently the World Wide Web Consortium [W3C Semantic Web] and several research groups [AIFB; DARPA] have been involved in the development of semantic web standards build upon XML syntax to provide a mechanism for exchanging data over the Internet. These semantic web standard languages [2], such as RDF, RDFS, OWL, will be discussed later in the chapter 5.

[2] The W3C announced final approval of two key Semantic Web technologies in 10 February 2004, the revised Resource Description Framework (RDF) and the Web Ontology Language (OWL) are Semantic Web standards.

Chapter 5 : Literature Review Of Ontologies Approach

This research aims to support semantic interoperability and enhance information integration in the inter-enterprises environment by providing a common semantic meta-data model. In this chapter, research into ontology approaches have been explored through publications relating to ontologies theory, ontologies representation languages, semantic web technologies (RDF, RDF Schema, and OWL), and current ontologies application areas in information search, semantic web, information integration, and knowledge management, will be discussed.

5.1 Ontologies Theory

> " Ontology" is a philosophical discipline, a branch of metaphysics that deals with the nature of being.
>
> Collins English Dictionary.

How thoughts, words and things relate to one another has been a recurrent subject in philosophy and linguistics? [Ogden and Richards 1923] introduced the interaction between symbols (or words), thoughts (or concepts) and things (or referents) of the real world as the meaning triangle (Figure 5.1). A concept is only an idea until it can be expressed by a symbol in a way that others can understand it. A symbol cannot completely capture the essence of a concept or of a referent; there is a relationship between them.

The meaning triangle is a model that linguistic expressions relate to a referent and to a concept: the direct identity relationship can only be derived from the mutual identification with a mediating concept, in the mind of some individual. The referential complexities are hidden in the human language triangle leading to ambiguities in communication where multiple terms may refer to the same thing and a single term may refer ambiguously to more than one thing. This is not suitable for building models in machine communication.

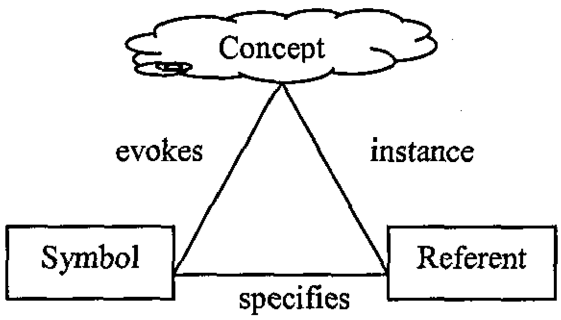


Figure 5.1: The Meaning Triangle[Ogden and Richards 1923]

[Maedche 2002] supposed the idea that underlies the meaning triangle has been combined with a "semiotics structure" on ontologies. He considered ontologies as models that are used to communicate meaning between machines and human beings. Figure 5.2 depicts the overall ontologies setting for communication between human and machine agents.

Consider the scenario of Figure 5.2; two human agents HA1 and HA2 exchange a specific sign (e.g. a word like "Jaguar"). Given their own internal model each of them will associate the sign to their own concept (or thought) referring to possibly two completely different existing things in the world, e.g. the animal vs. the car.

On the other hand, the machine agents MA1 and MA2 use the ontology to have a common semantic basis. When agent MA1 uses the term "Jaguar", the other agent MA2 may use the ontology just mentioned as background knowledge and rule out incorrect references, e.g. ones that let "Jaguar" stand for the car. Human and machine agents use their concepts and their inference processes, respectively, in order to narrow down the choice of referents (e.g., because animals do not have wheels, but cars have).

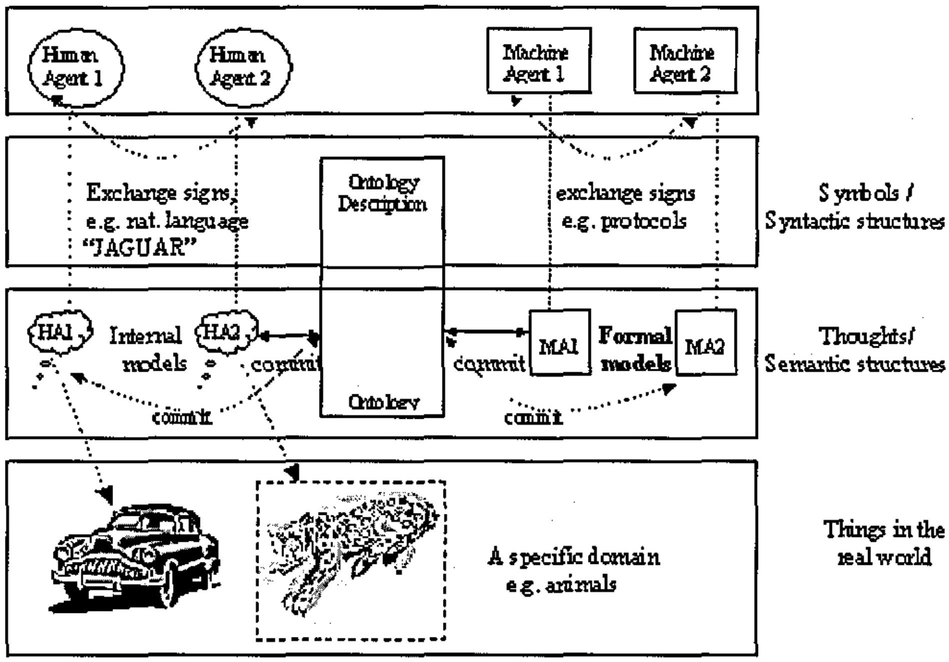


Figure 5.2: Ontologies for Communication [Maedche 2002]

Therefore, the formal languages of ontology are used for explicit representations of the real world and they arose from the needs of the artificial intelligence (AI) community to develop a terminology for building knowledge bases for particular domains in machine processable forms. Ontologies have been gaining interest in computational research, in addition to philosophical research.

Gruber provides widely quoted definitions of an ontology, as *"a formal, explicit specification of a conceptualisation" [Gruber 1993a]*. " *Conceptualisation*" refers to an abstract model of some phenomenon in the real world that identifies the relevant concepts of that phenomenon. "*Explicit*" means that the type of concepts used and constraints on their use are explicitly defined. "*Formal*" refers to the fact that the ontology should be computer understandable, written in a set of rigidly defined logical sentences or axioms regarding the intended meaning of the vocabulary used to describe a certain reality.

At the specification level, McGuinnes found that people encountered many forms of specifications that different people termed ontologies. The formalisms used can range from a finite list of terms (e.g. catalogue), glossary of simple terms, class taxonomies[3] (an informal is-a relation / a formal is-a relation), frames [4](classes and properties), value restrictions, to general logical constraints [McGuinness 2002]. Ontologies can therefore be conceived and applied at different levels, ranging from simple, informal developments to formal ontologies which can be strictly applied to enable automatic, machine use and reuse.

5.2 Ontologies Representation languages

When considering ontology-based applications, inevitably the issue of ontology language will arise. An ontology must be encoded in some language. The language does not only have to be able to express the concepts in the domain, but it also needs to consider the reasoning that may be supported in the language. Some fields such as Description Logics (DL) make this a central focus in language design. [McGuinness 2002] pointed out that the sets of formal constraining axioms and the logical reasoning theory view of ontology need to be considered, to express the concepts in the domain for computers to manipulate meaningfully.

There are a number of ontology specification languages including Classic Knowledge Representation System http://www.bell-labs.com/project/classic/ [Ronald J. Brachman et al. 1991], Description-Logic Knowledge Representation System Specification (KRSS) [Peter F. Patel-Schneider and Swartout. 1993] and Knowledge Interchange Format [KIF 1999]. More recently in this research area semantic web technologies [Lassila et al. 2000; McBride 2002; McGuinness et al. 2002; McGuinness and Van Harmelen 2003] have used languages to represent instantiated ontology and to structure collections of data and sets of inference rules for semantic browsers. The Semantic Web is based on two fundamental concepts: the explicit representation of the meaning of the content on the web and machine-processing these

[3] A taxonomy is commonly used as a hierarchical structure defined by "type" or "is-a" relationships. For example, a car is a type of transportation. It can also represent a part-whole relationship, e.g. a wheel is a part of the car.

[4]The frame system involves defining what kinds of classes, class hierarchy, properties of class and restrictions on property.

meanings in automatic way by rules, logic and inference engines. The materialization of this vision is supported by the incorporation of the many Semantic Web tools and technologies currently in the development. Tim Berners-Lee offered the architecture diagram (see Figure 5.3) of the Semantic Web Architecture in his digital paper " Semantic Web Road Map" [Berners-Lee 1998] and his presentation at the XML 2000 conference [Berners-Lee 2000] provides a basic foundation.

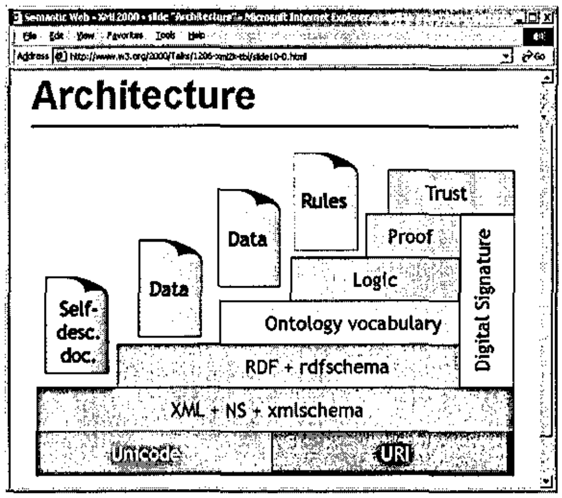


Figure 5.3: Berners-Lee's Semantic Web Architecture [Berners-Lee 2000]

The syntax layer provides a syntactic representation of the ontology and the knowledge base using the Extensible Markup Language (XML). XML present text structure for humans to read on the web, but does not contain markup information about the contents of the page for the computer to manipulate. The data model layer, the schema layer and ontology vocabulary layer are based on the Resource Description Framework (RDF), RDF-Schema (RDFS) and different language primitive vocabularies (e.g. DAML+OIL, OWL, and F-Logic ...etc) that provide a simple data model to define terms and their relationships to other terms. Currently the most advanced layer that has reached maturity is the ontology vocabulary layer, in the form of the OWL which corresponds to a rich DL. Hence, the next step will be the realization of logical rule systems on top of the ontology layer. The logic layer provides formal semantics that allow us to deduce implications of the term definitions and relationships. Finally, the proof and trust layer for monitoring and validating of logical steps, but no models have been defined in these layers yet. The details of XML, RDF, RDFS, and OWL are described in the following sections.

5.2.1 Syntax Layer - Extensible Markup Language (XML)

The XML provides web-friendly data structuring syntax for presentation and exchange of data over the Internet, like Hypertext Markup Language (HTML) it is a human-readable text and a markup language on the Web. XML supports an extensible set of features; such as user defined tags for specific contexts and users can define what they mean in Document Type Definitions (DTDs) or XML Schema. XML Schema was approved as a W3C Recommendation on 2 May 2001. The reasons why XML schema is better than DTD are [Fensel 2002]:

- XML schemas use XML syntax. XML definitions of schemas are XML documents and can be validated and rendered by the same software tools;
- XML schemas provide a rich set of elementary datatypes that can be used to define the values of elementary tags;
- XML schemas are extensible. e.g XML schemas provide much richer means for defining nested tags (i.e. tags with sub-tags);
- XML schemas provide the namespace mechanism to combine XML documents with heterogeneous vocabularies.

Goldfarb points out that the term XML text refers to the combination of character data and markup, not character data alone. Character data + markup = text. [Goldfarb 2001]. Actually XML itself only has content, any markup has to come from elsewhere. The Extensible Stylesheet Language (XSL) can be used to markup XML documents, which provide an alternative to Cascading Style Sheets (CSS) for formatting and styling an XML document. Another part of XSL is the XSL Transformation Language (XSLT), which is used to transform an XML document from one form to another. The resulting document may be XML, HTML, plain text or any other text-based document. That is the beauty of XSLT. One of the design goals for XSL was to make it possible to **transform data** from one format to another on a server, returning readable data to all kinds of browsers. These capabilities provide platform independence and Web-friendly data structuring syntax for presentation makes XML a common data format for data interchange between computer systems and applications.

5.2.2 Data Model Layer - Resource Description Framework (RDF)

RDF/XML builds upon XML syntax to provide a mechanism for exchanging semantics over the Internet. W3C describes, "RDF is a foundation for processing metadata (data about data)[5]; it provides interoperability between applications that exchange machine-understandable information on the Web." RDF provides a standardised data model on top of XML.

RDF is a standard for describing resources. What is a resource? That is rather a deep question and the precise definition is still the subject of debate. RDF is the W3C standard that is the foundation for the Semantic Web; so strictly speaking, an RDF resource is identified by a Uniform Resource Identifier (URI) reference. For the purposes of this research, it can be thought of as anything that can be identified.

An RDF description is a list of triples: resource, property, and value. A ***resource (the subject)*** is shown as an ellipse and is identified by a URI. Resources have ***properties (the predicate)*** that may be thought of as attributes of resources and also represent relationships between resources. That is, resources may be related to each other or to ***values (the object)*** via properties.

In figure 5.4 two resources are defined with a URI as their unique global identifier, each carrying a order_No and a order_date, quantity property with literal values, identifying the resources as http://www.eemse.co.uk/ order no/LU3223-1 and http://www.speedwell.co.tw/order/, correspondingly. These two resources are related via property order_by. RDF is based on a triple model.

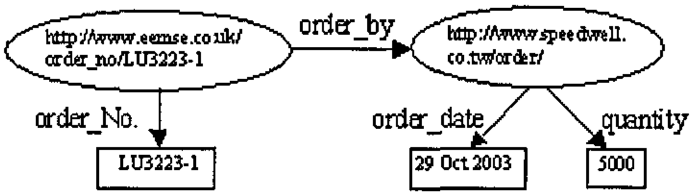


Figure 5.4: An example of the RDF Data Model

[5] Figure 5.4 and 5.5 clarify these elements.

All information is described using one or more RDF statements. In a RDF statement, the source of the relationship is called the **subject**, the labelled arc is the **predicate**, and the relationship's destination is the **object**. The statement can be documented using several different techniques. One of the most popular techniques, for instance, uses a 3-tuple to represent of the RDF triple as: {Subject, Predicate, Object}. A 3-tuple representation of a RDF statement from the figure 5.4 example becomes:

{http://www.eemse.co.uk/order_no/LU3223-1, order_No, "LU3223-1"}

5.2.3 Schema Layer (Metadata Repository) - RDF Schema (RDFS)

The RDF provides the meaning of information enabling semantics to be added to a document by using the triple model {Subject, Predicate, Object}, in a similar fashion to semantic nets or to frame-based systems. Meaning in RDF is expressed through reference to a schema. A schema is the place where definitions and restrictions of usage for ***classes*** and ***properties*** are documented. That is, it provides the means to define concept (or classes) hierarchies, and domain and range restrictions for properties. Thus, RDFS defines the terms that will be used in RDF statements and gives specific meanings and constraints to them.

According to [Powers 2003], if RDF is a way of describing data, then the RDFS can be considered as a domain-neutral way of describing the metadata that can then be used to describe the data for a domain-specific vocabulary. The best way to fully understand how the RDFS works is by looking at the elements that make up the schema. In the following the most relevant RDFS primitives are given. The detail of the RDFS elements specification can be found at http://www.w3.org/2000/01/rdf-schema#.

RDFS classes:

- *rdfs:Resource*, all things described by RDF are called resources and are instances of the class rdfs:Resource. Figure 5.5 shows two subclasses, namely rdfs:Class and rdf:Property.

- *rdfs:Class* denotes the set of all classes in an object-oriented sense. That means that classes like appl:Person or appl:Organisation are instances of rdfs:Class.
- *rdfs:Property* defines in the same way as rdfs:Class, e.g property like appl:cooperateWith is an instance of rdf:Property.

RDFS Properties:

- *rdfs:subClassOf* defines the subclass relationship between classes.
- *rdfs:subPropertyOf* similar to rdfs:subClassof which defines a hierarchy of properties.
- *rdfs:range* is used to specify the classes the property can reference as values.
- *rdfs:domain* associates a property with the class by stating that the property has a given class as its domain.

Figure 5.5 illustrates an RDFS example from Figure 5.4 RDF Data Model.

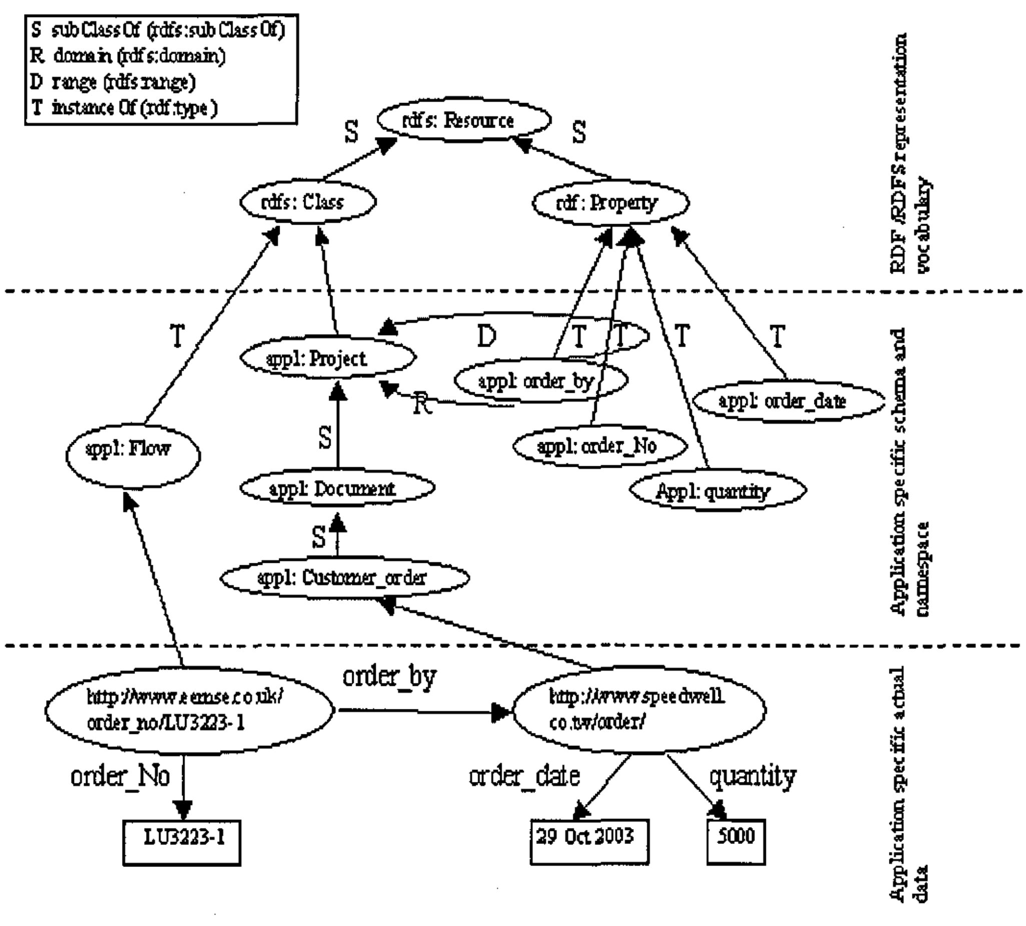


Figure 5.5: An RDF-Schema Example

5.2.3.1 Difference Between RDFS And Relational Database Schema

RDF provides a very simple triple (*subject-predicate-object*) model that consists of a set of nodes connected by arcs, forming a pattern of node-arc-node. The nodes come in three varieties: URI reference, blank nodes and literals. The arcs used to describe attributes of nodes and relationships between nodes. The RDF is a model of entities (nodes) and relationships, which is basically an opening of the " Entity-Relationship Model " [Chen 1976] to work on the Web.

Typically, relational database (RDB) models are generated from entity-relationship models. Therefore, the RDF model is very directly connected with the RDB model in this respect. The RDB model, first introduced by [Code 1970], represents the data in a database as a collection of relations. Informally, each relation resembles a table; each row in the table represents a collection of related data values. These values can be interpreted as a fact describing an entity or relationship instance. The table name and column names are used to help in interpreting the meaning of the values in each row of the table [Elmasri and Navathe 1989].

Within RDB table like structure, every table has columns (contain a column for each element within the domain being described), data types (the types of values that can appear in each column), a primary key (value that uniquely identifies the entity) and foreign keys (values that identify and refer to entities in other tables), which are defined as a relational schema (metadata).

However, a relational schema is created independently for each database. For instance, two relational databases have the same values created by two different domain-specific schemas, as shown in Figure 5.6. This makes it difficult to share information between systems that do not share the identical relational schema.

Customer_order

order_number	order_date
LU 3223-1	29 Oct 2003
SP 45633-23	26 Jan 2004

Sale

sale_number	sale_date
LU 3223-1	29 Oct 2003
SP 45633-23	26 Jan 2004

Figure 5.6: The incompatible RDB Schema

The RDF model is different from a RDB model in respect of its structure, with RDF all the tables have the same format (Subject, Predicate and Object) and keys are not needed. Based on this domain-neutral approach, it is this fact that provides the interoperability. Figure 5.7 shows the difference between RDF schema and a RDB schema.

Relational Database Structure

Flow

P:flow_name	F:order_number
Order entry for...	LU3223-1

Customer order

P:order_number	Order_date	quantity
LU3223-1	29Oct2003	5000

Database structure uses tables, columns, primary and foreign keys.

*P: Primary Key F: Foreign Key

RDF statement

MSE Ontology

Subject	Predicate	Object
On:order	c:order_date	29Oct2003
On:order	c:quantity	5000
On:order	rdf:type	c:Customer_order
On:order	c:travels_along	oe:Order entry for ...
Oe:Order entry for...	c:carries	LU3223-1
Oe:Order entry for ...	rdf:type	c:Flow

All tables are the same format (Subject, Predicate, Object).

Figure 5.7: The difference between RDFS and RDB Schema

Computers that process RDF can share disparate information by mapping from one schema to another through a common schema, as show in figure 5.8, and by using inference rules.

Customer order

order number	order date
LU 3223-1	29 Oct 2003
SP 45633-23	26 Jan 2004

mapping to ⟷

Order

number	date
LU 3223-1	29 Oct 2003
SP 45633-23	26 Jan 2004

mapping to ⟷

Sale

sale number	sale date
LU 3223-1	29 Oct 2003
SP 45633-23	26 Jan 2004

Figure 5.8: A common Schema

The mapping approach adopted is to model the axiom specification in the RDFS in an object-oriented manner. Following the object-oriented tradition, RDFS provides the special primitive rdfs:subClassOf that defines the subclass relationships between classes. There is a further special type of relation that is similar to refs:subClassOf, and this is refs:subPropertyOf, which defines a hierarchy of properties.

Hence, the schema mapping defines two inference rules, the subclass rule and the subproperty rule for RDFS. The RDFS rules are very recursive in a logical sense, that is, if the relation relates objects part way down the inheritance tree of the class hierarchy then it must be possible to traverse upward to objects that are higher in the hierarchy.

For example, The subclass rule, a resource (on:order) is an instance of the subclass of the c:Order class if and only if it is an instance of the f:Customer_order class and the f:Customer_order class is a subclass of the c:Order class (see figure 5.9[a]). The following additional example demonstrates the subproperty rule. A value (e.g. 29 Oct 2003) is a instance of the subproperty of the c: date property if and only if it is an instance of the f: order_date property and the f: order_date is a subproperty of the c: date (see figure 5.9[b]). The same subclass rule and subproperty rule apply to another resource (on:sale). In this example, the rules should ensure that when someone queries for the common ontology for instances of the c: date, the result includes all instances of the f: order_date or sale_date from the Customer_order calss or the Sale class. Hence, RDF provides the interoperability, regardless of the domain name as in RDB.

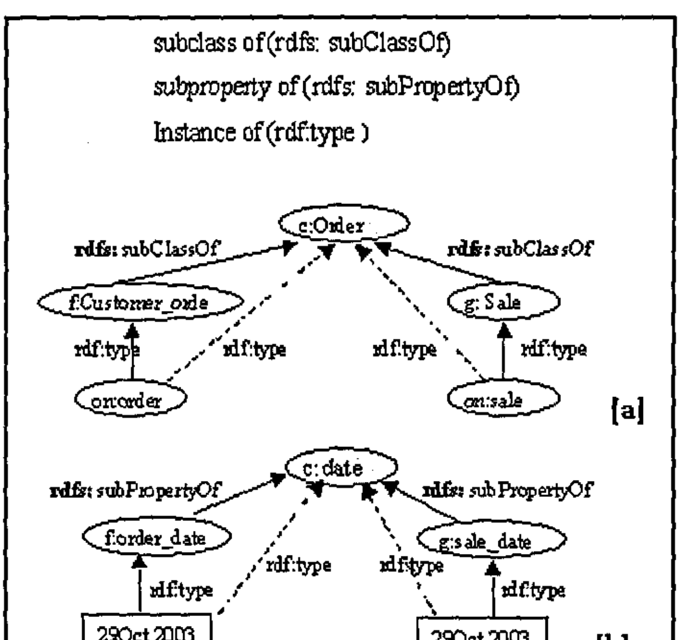


Figure 5.9: The recursive algorithms

5.2.4 Ontology Layer - Web Ontology Language (OWL)

RDF provides a simple data model. RDFS defines a simple ontology language with classes, sub-classes, properties, sub-properties, and domain and range restrictions in RDF for expressing metadata. However, RDFS is not explicit (formal) enough and still does not provide exact semantics when it comes to representing complex constraints. Formal semantics for the primitives defined in RDFS are not provided, and the expressivity of these primitives is not enough for full-fledged ontological modelling and reasoning. To perform these tasks, an additional layer on top of RDF Schema is needed [Broekstra et al. 2001].

One of standard semantic web technologies approved by W3C in this layer is Web Ontology Language (OWL), which is designed as an extension of RDF/ RDFS and is derived from the DAML+OIL (DARPA Agent Markup Language + Ontology Inference Layer). OWL facilitates greater machine readability of Web content than that supported by XML, RDF and RDFS by providing additional vocabulary along with a formal semantics (e.g. enumerations, restrictions, and logical statements) [McGuinness and Van Harmelen 2003]. The coding of data with semantic metadata allows users to access various kinds of heterogeneous data, including semantic heterogeneity (different vocabularies, logical schemas), and structural heterogeneity (different data structures: plain files, database, and WWW documents). The basic modelling elements of OWL are given in following. The detail of OWL primitives can be found at http://www.w3.org/TR/2004/REC-owl-ref-20040210/.

- ***Classes*** represent domain concepts and can be arranged in inheritance hierarchies, have properties to describe the attributes of the class and their relationships to other classes, and can have individuals (instances).

- ***Restrictions*** represent constraints on a certain property. OWL distinguishes two kinds of property restrictions: value constraints and cardinality constraints.

- A value constraint puts constraints on the range of the property *when applied to this particular class description*, e.g. owl:allValuesFrom, owl:someValuesFrom, and owl:hasValue.
- A cardinality constraint puts constraints on the number of values a property can take, e.g. owl:maxCardinality, owl:minCardinality, and owl:cardinality.

- ***Complex class expresses*** can also be defined by logically combining statements (e.g. intersection, union, and complement) about other classes, They can be viewed as representing the AND, OR and NOT operators on classes. These three operators get the standard set-operator names: owl:intersectionOf, owl:unionOf and owl:complementOf. Using OWL it is also possible to state that two classes are the same (owl:sameClassAs), equivalent (owl:equivalentClass) or disjoint (owl:disjointWith).

5.3 Ontologies Application Areas

An Ontology approach, based on formal specification, provides flexible and personalized access to the knowledge sources by allowing a group of individuals to structure and model a domain conceptually. It has been used to support the sharing and reuse of formally represented knowledge among AI [Gruber 1993]. Ontologies are the appropriate modelling structure for representing knowledge and are critical components in Semantic Web, knowledge management, electronic business applications, and several other application areas, e.g. in

- Web Service [McIlraith and Martin 2003; Staab et al. 2003; Zaijun Hu et al. 2003; Arpinar et al. 2004]
- Data Mining [Li and Zhong 2003; Priebe and Pernul 2003]
- Ontology Learning / Machine Learning [Maedche 2002]
- Process-related applications integration [Cutting-Decelle et al. 2003; Pouchard and Cutting-Decelle 2003]
- Intelligent agents [Hendler 2001]

The following provides more elaborated examples that are worthy of special attention in, Semantic Web, knowledge management (knowledge representation, interpretation, retrieval, query, extraction, maintain, and integration), and Business-to-Business (B2B) E-Commerce.

5.3.1 Knowledge Management

Knowledge management is concerned with facilitating acquisition, access, maintenance, and reuse of an organisation's knowledge and information, typically using advanced technology – knowledge based systems. Owing to globalisation, an evolution of distributed data management systems has taken place, depending on the degree of heterogeneity, distribution, and autonomy existing in the underlying data repositories, and the existence of a global schema. Furthermore, because of the Internet's impact, autonomous repositories that store different types of digital data in multiple formats are becoming available for use on the fast-evolving global information systems infrastructure. This information overload makes it impossible for users to be aware of the locations, organization or structure, query languages, and semantics of the information that exists in various repositories. Using ontologies as semantics-driven information of the data repositories is the key to hiding the heterogeneity from users as well as to allowing autonomy.

Formal knowledge management systems contain knowledge bases and ontologies, which could provide completely new possibilities: document exchange between departments through ontology-mediated mappings, definitions of views on documents, facilitate communication between its multiple users and links between multiple knowledge bases. Applications of ontologies in knowledge management are described in [Fensel 2003]. Research projects for Knowledge management, include [On-To-Knowledge 2000-2002; OntoWeb 2001 -2004].

5.3.2 Information Integration In B2B E-Commerce

E-commerce is about electronically exchanging business information – including product descriptions with information about vendor, the manufacturer, the lead time required and numerous other business-related considerations. In order to exchange business transactions electronically the sender and the receiver have to agree on a common standard (a protocol for transmitting content and a language for describing content) [Fensel 2002]. A number of standards arose for this purpose – e.g. the Electronic Data Interchange for Administration, Commerce, and Transport (EDIFACT) by UN and USA Federal EDI standard. However, the traditionally used EDIFACT / EDI on Virtual Private Network (VPN) is expensive both in terms of applications and system running costs and requires large maintenance efforts.

EDI functions for B2B transaction can now be moved to the lower cost WWW and Internet platform that provides the crucial turning point for e-commerce. The ubiquity of Internet standards such as TCP/IP, HTTP, HTML, and XML has enhanced the information interoperability between business partners. However, although XML provides a standard structuring syntax for presentation and exchange of data, it does not provide semantic terminologies to describe business processes and exchanged products.

B2B marketplaces have to deal with serous problems of heterogeneity. [Fensel et al. 2001] consider that this heterogeneity arises in at least three levels: the content, product catalogue structure, and document structure. The content of the exchange information must be modelled. They suggested that successful content management for B2B electronic commerce must deal with several challenges:

- Extracting information from rough sources;
- Classifying information to make product data maintainable and accessible;
- Reclassifying product data;
- Personalizing information; and
- Creating mappings between different information presentations.

To overcome the heterogeneity problems, the current B2B e-commerce needs intelligent solutions for mechanizing the process of structuring and standardizing, in addition to the content management. Ontologies provide much richer modeling means with classes and properties organized into is-a hierarchies and enriched with axioms and relations processable with inference, which may play a key role in content management.

However, constructing a shared domain ontology from scratch is a difficult task. Therefore, in the B2B web-commerce industry, efficient XML-based e-commerce information exchange needs ontologies in two important ways. Firstly, standard ontologies, as there are a number of specific parts of the business integration domain that have been carefully modeled within several standardization initiatives driven by large consortiums, e.g. [cXML; RosettaNet; xCBL]. Secondly, ontology-mediated translation services, in which ontologies serve to model the negotiability between each personalized product descriptions or link into the standard ontologies. [Omelayenko 2002] proposed an architecture for an ontology-based business integration service relying on a composite mediating ontology constructed from several business, a temporal, and a mapping ontologies. A comprehensive overview on applying ontologies E-commerce and its relationships to existing standards is given in [Fensel 2002]

5.3.3 Semantic Web

The current web technology, such as Internet / Intranet / Extranet, has provided platform independence for users to access data anywhere and anytime to support the global inter-enterprises operation. However, this enormous amount of various heterogeneous data (e.g. semantic heterogeneity or structural heterogeneity) has made it increasingly difficult to share and exchange information required by a wide variety of users.

Furthermore, currently web searching is done using keyword matching. Problems with keyword-based search are:

- The human user has to manually extract and interpret the information;
- It can retrieve irrelevant information that uses a certain word in a different context, and;
- It might miss information when different words are used about the desired context.

In response to these problems and to achieve true interoperability, the concept of "Semantic Web" - machine-processable semantics of data on the web start to emerge [Berners-Lee 1998]. The Semantic Web is based on two fundamental concepts: the explicit representation of the meaning of the content on the web and machine-processing these meanings in automatic ways by rules, logic and inference engines.

A Semantic Web is not about pages and links, but rather, it is about relationships between web pages indicating, for example whether one thing is a part of another. Web pages are annotated by ontology-based meta-data and logical rules so that an automatic system can follow the structure of the relationships and find, extract, represent, interpret, and maintain relevant information. This web content management is enhanced with link semantics, which provide ontology-based search instead of keyword matching, and details can be found in [Hyvönen et al. 2004; Varlamis et al. 2004]

Chapter 6 : IT Solutions For Integration In Manufacturing Systems

This chapter includes various integration approaches in manufacturing tools, systems, architectures and current research to support inter-enterprise interoperability. Each application or system is considered individually so that particular strengths or weaknesses, in the current context, may best be explored. The chapter also provides the author view of how an MSE Moderator, which is an intelligent support application for moderation work between extended projects team, can be implemented in an extended / virtual organization. The research reported in chapters 3 to 6 therefore resluts in the development of an experimental MSE ontology model for the EEMSE Moderator.

6.1 VEGA Project

Vega

http://cic.cstb.fr/ILC/ECPROJEC/VEGA/HOME.HTM

Virtual Enterprises Using Groupware Tools And Distributed Architecture
(ESPRIT 20408) (1996-1999)

This project aims to integrate business and technical processes, adopting the concept of virtual enterprise. It targeted the Large Scale Engineering (LSE) industry, which works on shorter-term business relationships and supply chains geared to specific projects. The VEGA project provides an integration software architecture in distribution of information, information sharing and concurrent activity, which supports the exchange of project and product information between disparate organizations.

This research bridged the gap between four standards (STEP, SGML, CORBA, EDI), using the COAST (Corba Access to STep Models) platform and the associated services (Persitance, Workflow, Documentary Support, EDI Messages), and coupled with the Workflow applications management, for distributing CAD applications. [Zarli and Amar 1997; Stephens 1999].

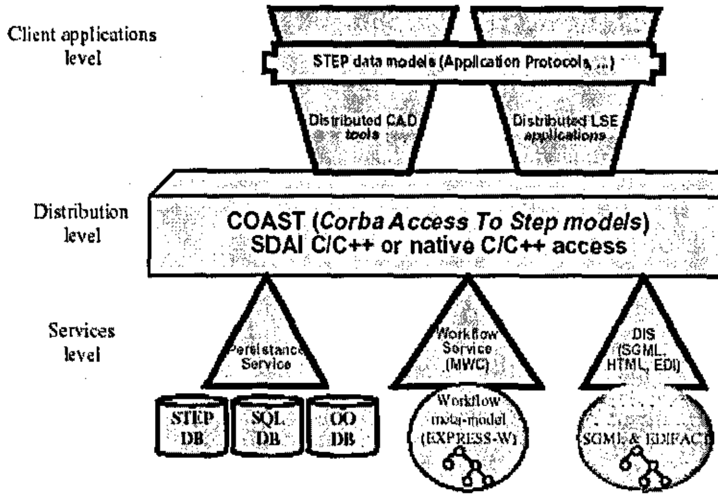


Figure 6.1: VEGA Platform [Debras et al. 1998]

Discussion:

This research is relevant for integration issues related to distributed databases. However, as shown in figure 6.1, the transporting of distributed information in this research is supported by COAST architecture based on CORBA (Common Object Request Broker Architecture) between remote data / applications. CORBA does not work very well for the Internet and works only for synchronous exchange of messages. Also CORBA has a symmetrical requirement, meaning that both ends of the communication link would typically need to have implemented the same distributed object model. Simple Object Access Protocol (SOAP) is not tied to a specific object model. SOAP offers interoperability across a variety of platforms because it is not tied to a specific object. For example, a client written in Microsoft VisualBasic could use SOAP to access a method in CORBA object running on a Unix platform. SOAP cleans up interoperability problems on the Web [Jepsen 2001]. At the time of this project, SOAP was not mature. In later research projects, e.g e-

COGNOS project, SOAP had reached maturity and was implemented and formed the core distributed infrastructure of e-COGNOS.

Additionally, data exchange is required to be the exchange of neutral format data files, supported by STEP format using EXPRESS language, between computer systems in this research. Disparate organizations have to send their information to their partners who then transform the data into STEP format by a converter (e.g. SGML/STEP converter, EDIFACT/STEP converter). The communication and exchange data between participants was used a standard vocabulary (STEP) and imposed a converter. It did not support directing data exchange. The semantic ontology approach does bridge the gap for directing data exchange, as can be seen in the more recent research projects, e.g. GENIAL, GLOBEMEN, WIDE that are presented in this chapter.

6.2 GENIAL Project

http://www.c-lab.de/genial/

Global Engineering Network Intelligent Access Libraries

(*ESPRIT 22284*) (1996 –1999)

GENIAL is the key project for the GEN [6] establishing a Common Semantic Infrastructure for global engineering market places, enabling enterprises from different engineering sectors to combine internal knowledge with global engineering knowledge and allowing them to acquire, migrate, publish, search, present, and administer information or services equally in an internal network of companies or Intranet, Extranet and Internet. Three main objectives of the project were [Gausemeier et al. 1997]:

- Establishing a Common Semantic model for describing products, users, etc.

[6] In 1994 a group of European engineering companies and organisations established the Global Engineering Networking (GEN) Initiative. The GEN Initiative is an open co-operation of industry and academia with the mission to provide a global electronic marketplace for users and suppliers of engineering products and services.

- Realization a software infrastructure for information acquisition, migration, presentation, administration etc (figure 6.2).
- Validation of the logical framework in real end-user pilots.

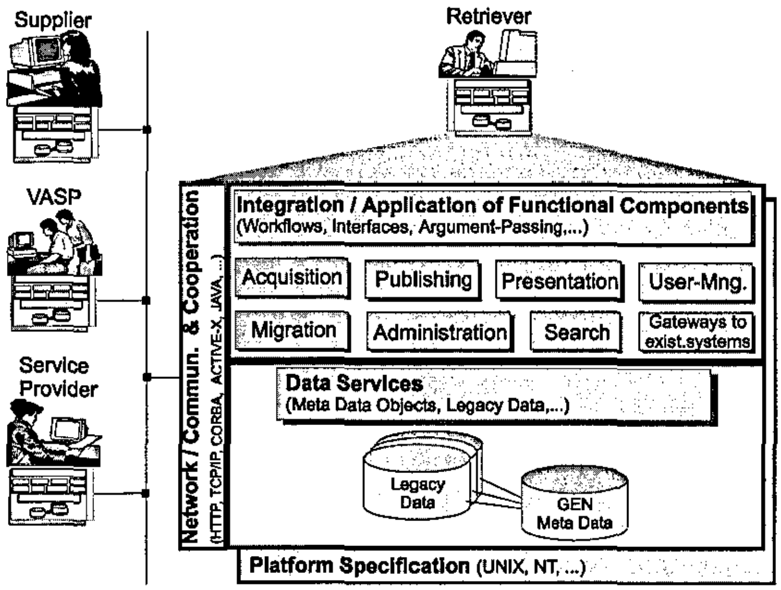


Figure 6.2: GENIAL Architecture [Debras et al. 1998]

The "backbone" of the GEN concept is a public collaborative network with intelligent nodes and services for large-scale distribution and controlled access to engineering knowledge by global network (Internet). However, most Internet-based information networks available gathering engineering knowledge are based on a centralised server approach. Suppliers have to send their information to the provider who then transforms the data into a specific format. The aim of GENIAL project was to provide a solution where any company could insert its data individually. Even existing data from other formats could be migrated into the information network without format conversions.

The critical factor of this research was to define a common information framework [Grabowski et al. 1997; Debras et al. 1998] and a distributed (logical) global information system which builds up a network of GEN-database. The GEN Meta data defines a common semantic model enabling a uniform view on various kind of information. This enables enterprises from different sectors to combines internal and external knowledge with knowledge acquisition and efficient searches according to standard (common) engineering classifications, shown in figure 6.3.

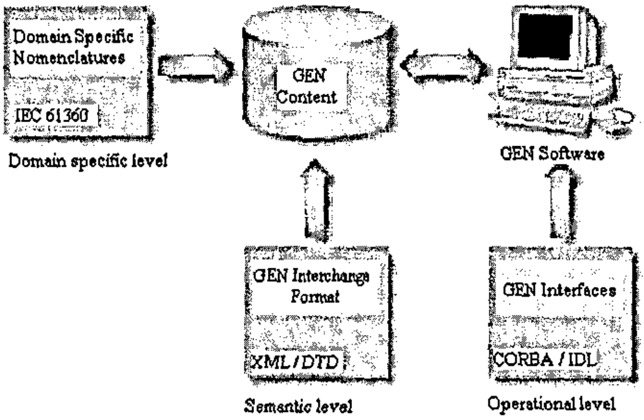


Figure6.3: Standardization Aspects for GEN [Radeke 1999]

Discussion:

The limitation of the semantic level in this research is the interchange format. GEN Meta data was based in XML / DTD format that requires manual acquisition of the meta data and semi-automatic extraction of the meta data. In fact, XML provides a common syntax for data interchange, but does not define the meaning of the information. Therefore the semantic mapping from Extra data to GEN Meta data must be performed manually.

Currently W3C work in semantic web technologies, such as RDF RDFS, and OWL, provides a framework for fully automatic information exchange, sharing and reuse on the Web. RDF/XML builds upon XML syntax to provide a mechanism for exchanging semantics over the Internet. Furthermore, classical ontologies need to allow the semantics to be precisely specified and complete inference to be viable. Many existing ontologies languages (e.g. OIL, DAML + OIL, and OWL) provide automatic mapping and seamlessly share disparate information by inference mechanism. Their success will depend heavily on the underlying logic foundation, especially on the description logics (DLs), and reasoning services that can be provided. Some of the latest research projects, such as WIDE and OntoWeb have adopted the semantic web technologies (RDF, DAML+OIL, OWL) as the information interchange format and these will be presented later in this chapter.

6.3 GLOBEMEN Project

http://globemen.vtt.fi/

Global Engineering and Manufacturing in Enterprise Networks

(ESPRIT IMS 99004) (2000- 2003)

This project built and expanded the key elements of the previous IMS project Globeman 21 - Global Manufacturing in the 21st Century. The GLOBEMEN project aims to support integration of business and engineering processes executed by a VE in a global and multicultural environment. GLOBEMEN is organized to address three main aspects of VE operation: sales and services (knowledge management, etc), inter-enterprise management (interfacing of enterprise systems) and distributed engineering (product, process, workflow management etc).

One of the major issues in GLOBEMEN was the identification of an inter-enterprise architecture [Kazi and Charoenngam 2003]. The Virtual Enterprise Reference Architecture and Methodology (VERAM) was its core finding and the basis for inter-enterprise collaboration. VERAM is about those modelling, technologies, standards, applications methodology and VE implementation that can be used during the formation and operation of VEs. More elaborate presentation of VERAM is provided in [GLOBEMEN 2000-2003; Kazi et al. 2001].

The modular approach used in VERAM was to identify a ICT layered architecture for cross enterprise teamwork in a global engineering and manufacturing setting, shown in figure 6.4. The main purpose of ICT architecture, including seven layers: Presentation layer, Application layer, Interoperability layer, Communication layer, Access layer, Service layer, and Storage layer, is to act as a mapping template upon which organizations could map their in-house applications and interface to shared VE environment. The functions and examples of each layer are discussed in table 6.1.

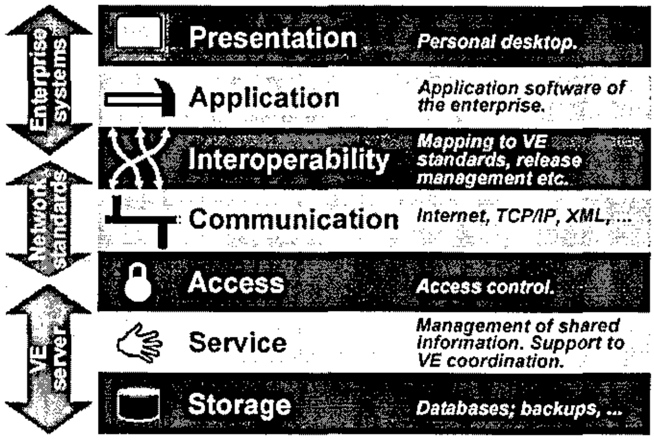


Figure 6.4: ICT architecture for inter-enterprise collaboration [GLOBEMEN 2000-2003]

Layer	Functions	Example
Presentation	User interface through which individuals gain access to VE information depending on their roles in the VE partner organization.	WWW browser (HTML + XML pages).
Application	Application that a user needs to perform tasks for specific VEs, including application software of the VE partner.	ERP, CAD.
Interoperability	The data / information mapping and translation mechanisms between an organization's applications and the shared VE environment. It may provide semantic and syntactic mapping to VE standards.	Conversion software that translates proprietary data format to a standard format to the VE.
Communication	Communication between an organization and the shared workspace of the VE. Thereby, this layer addresses both geographic and organizational distribution of VE partners. This layer relies mostly on standards and commonly available technologies.	Internet, communication protocols, middleware technologies.
Access	Controls the access to the shared VE workspace and information.	User identification, access rights management.
Service	Access to and management of shared information and services to the VE.	Inter-enterprise workflow management.
Storage	Hosts the main system registry and repository.	Database.

Table 6.1: Functions of ICT layers

Discussion:

Businesses today are becoming more dynamic and multicultural. The relationships between companies in networks are changing with increasing speed. Dynamic global networking cannot be efficient without guidelines, reference architectures and tools allowing true concurrency for all partners in the network. Therefore of this research is that it aims to provide a generic reference architecture specified through the required methods and tools for configuration and instantiation of virtual manufacturing enterprises.

6.4 e-COGNOS Project

http://www.e-cognos.org/

Electronic **CO**nsistent knowled**G**e ma**N**agement across pr**O**jects and between enterpri**S**es in construction domain

(IST 28671) 2000- 2002

This project aims to specify and develop an open web-based infrastructure and a set of tools that promote consistent Knowledge Management (KM) within collaborative construction environments. The research addresses four main issues: KM, web services, ontology, and construction industry [e-COGNOS Consortium 2002].

In this research, KM addresses the knowledge requirements of construction end-users while supporting their existing practices and taking into account the contractual, legal IPR (Intellectual Property Rights), security, and confidentiality constraints, which referred to as information and can be classified in to three following categories:

- *Domain knowledge*: It includes administrative information, standards, technical rules, and product databases, etc.
- *Organisational knowledge*: this is company specific, and is the intellectual capital of the firm.
- *Project knowledge*: this is both knowledge each company has about the project and the knowledge that is created by the interaction between firms.

A major e-COGNOS objective is to develop a web-centred and ontology-enabled solution that has been implemented following the Web Services model, incorporating the Simple Object Access Protocol (SOAP), Universal Discovery Description and

Integration (UDDI), and Web Services Description Language (WDSL), to manage the communication and relationships between these services. The e-COGNOS knowledge management services (figure 6.5), which include the creation, capture, indexing, retrieval and dissemination of knowledge [Wetherill et al. 2002]. A construction-specific ontology is used in conjunction with algorithms selected and developed by the consortium, as the primary mechanism to manage document consistency and broker the various knowledge related services.

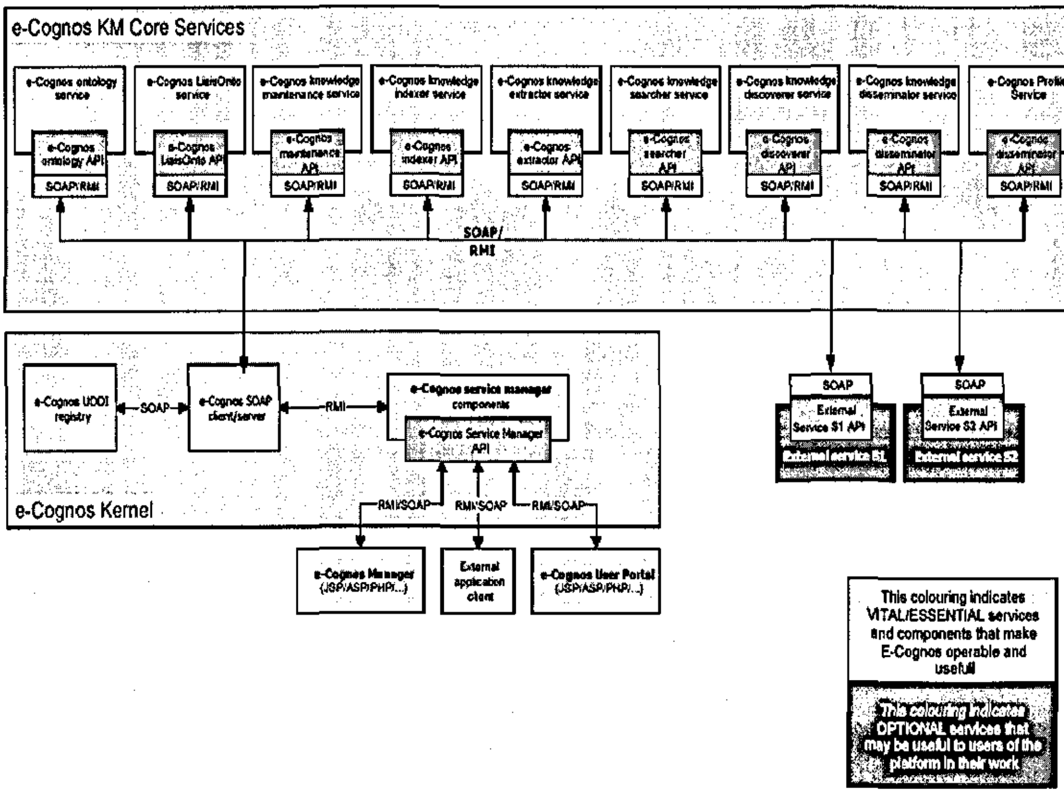


Figure 6.5: e-COGNOS Knowledge Management Services
[Wetherill et al. 2002]

Unlike [eConstruct 2000-2002] project, the e-COGNOS research did not actually define a building construction vocabulary / taxonomy. Instead, this project reused the current available construction industry standards such as Industry Foundation Classes (IFC) [Bazjanac 1998] http://www.iai-international.org, BS6100, SUMO http://suo.ieee.org and bcXML Schema (under the e-Construct project) http://www.bcxml.org/default_frame.htm taxonomy and has developed over 700 concepts, basic set of relationship in DAML+OIL format http://www.daml.org.

Discussion:

The application area for this project is essentially that of Web Services (SOAP, UDDI, WDSL) and taxonomy for construction industries knowledge management. This research explored the open web-based infrastructure and has provided an environment for the efficient and effective implementation of applications integration. The main focus on this research is relevant for integration issues related to distributed applications and distributed databases. Although the ontology approach has been adopted in the project [Lima et al. 2002]. The database interoperation is built on the standards-based translation mechanisms by mapping to the common construction industry standards.

6.5 IST for CE

http://www.istforce.com/

Intelligent Services and Tools for Concurrent Engineering
(IST-1999-11508) 2000-2002

This project is an Internet-based platform providing intelligent services and tools for an engineer participating in parallel in *multiple projects*. The developed approach enables plug-in of different IT tools on the platform, directly or as extended rented engineering services. An important part of the research work is concentrated on multi-project workflow management. Another important aspect of this work is to provide a user-centred services platform for CE, providing customisable user-friendly capabilities for management and modification of the data. For this purpose, an engineering ontology, which provides language interoperability service and allows the user to keep his own individual language, has been developed and implemented.

Discussion:

This research does address many of the issues related to the requirements for CE team working, practically with respect to the personal needs of the individual users. The research provides a common medium for communication with a shared, reusable product database to various distributed platforms, users, applications and network protocols and has developed methods and specifications for an engineering-friendly ontology framework that can bridge the gaps between users, data models and software applications. However, the XML-based ontology format, the mapping process between the common medium model (IFC model [Bazjanac 1998]) and the ontology is complicated, and often too slow.

6.6 WIDE Project

wide Semantic Web-Based Information Management and Knowledge Sharing for Innovative Product Design and Engineering http://www.ist-wide.info/

(IST-2001-34417) 2002-2005

The research tries to bridge the gap between different interdisciplinary teams by offering them an easy and effective way to access commonly used information sources without having all to speak a common language that is not natural for them. This means that sufficient understanding of the other's terminology is required. The project takes this multi-language plus cross-understanding idea as the basis for effective collaborative working and for sharing knowledge [WIDE 2002-2005]. Figure 6.6 shows an example here of the cooperation of designers and engineers in the automobile industry. Although working in the same domain these different user groups have totally different backgrounds and use different terminologies to talk about things like cars.

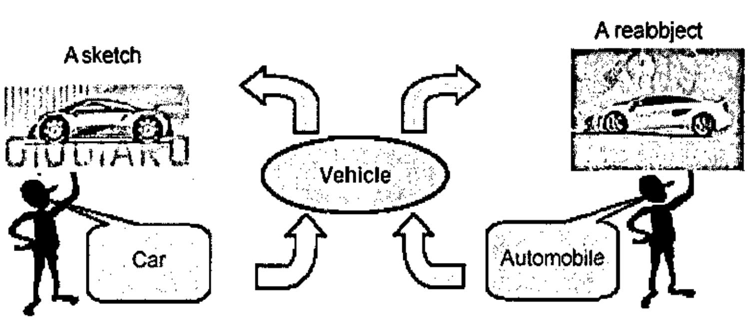


Figure 6.6: WIDE Motivation [WIDE 2002-2005]

The goal of WIDE is to develop an *intelligent search engine* to support an effective information management and knowledge sharing system for inter-enterprises of multi-disciplinary design teams by offering a natural and coherent environment for:

- Identifying information needs;
- Finding and assessing different information sources;
- Receiving and viewing information from different sources;

In doing so, the WIDE system tries to handle different terminologies using technologies from the domains of Knowledge Engineering and Semantic Web (SW). By making use of metadata, semantic annotation of documents and ontologies, user queries are interpreted and automatically connected to the corresponding information sources. Technologies, such as JAVA, XML, RDF, DAML+OIL, natural language processing, and text search, are used. The results for the query stemming from different information sources are then being integrated on a semantic level. Figure 6.7 [WIDE 2002-2005] shows the WIDE system architecture.

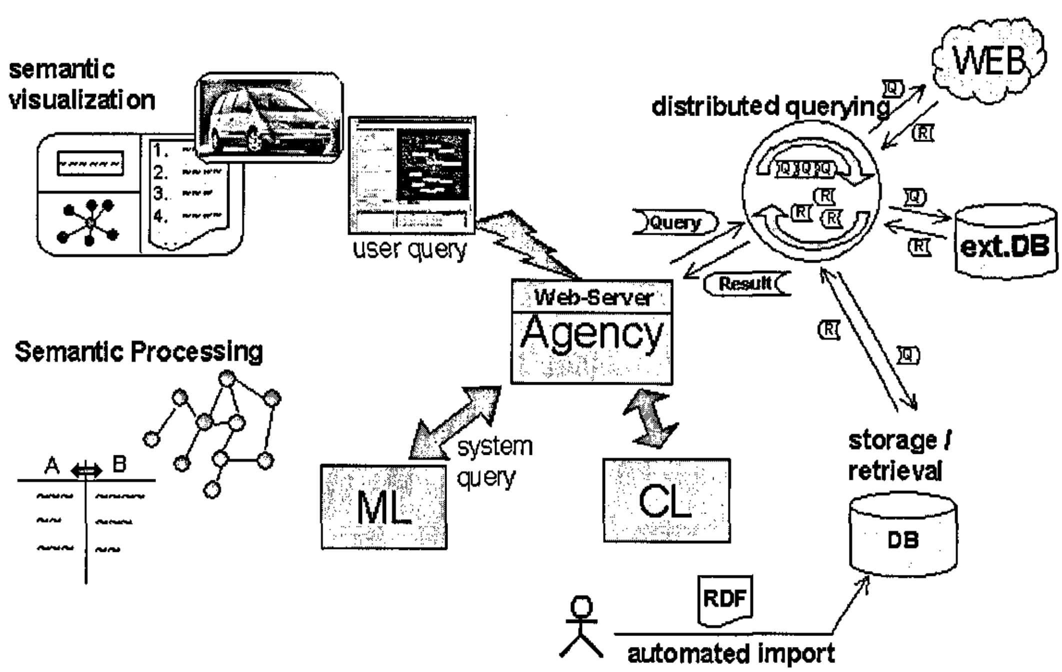


Figure 6.7: WIDE Architecture [WIDE 2002-2005]

The WIDE architecture provides the following functions:

- Semantic visualization of search results through a conceptual tree view presentation for documents clustered according to an underlying domain ontology
- Distributed querying over several heterogeneous information sources and result assembly
- Knowledge sharing without forcing the participants to speak a common language by utilizing semantic processing techniques for queries and results involving domain ontologies, thesauri and dictionaries
- Generic meta data modelling, storage and retrieval.

Discussion:

This research attempts to define a general knowledge representation which can be used to facilitate communication problems in the engineering design applications, using ontologies and semantic web technologies. Although this research is not into a coordination and moderation of concurrent engineering team working, it does include important group communication issues. The research does address many of the inter-enterprise and inter-working issues related to the requirements of information semantic interoperability as identified in the earlier chapters. The application area for this research is similar to the GENIAL project for information access, extraction, representation, interpretation and maintenance. The main difference is the information interchange format. The WIDE project applies the state of art in the semantic web technologies which will enable much more automated services based on machine-processable semantics for data and heuristics that make use of the metadata.

Chapter 7 : Manufacturing System Engineering (MSE) Ontology Model

In this chapter, the features of the proposed common ontology model to support information communication within the extended project team are discussed. Based on the literature review in previous chapters, an underlying standardized Manufacturing System Engineering (MSE) Ontology meta-model for the EEMSE Moderator is proposed, analysed and developed, which provides a common ontology model for improving the semantic and syntactic interoperability between different MSE applications. The MSE Ontology Model specifies a range of classes and properties based on the emerging semantic web language, RDF, RDF-Schema, and OWL. The implementation of instances of the MSE Ontology Model for formation of the EEMSE Moderator is also discussed. The MSE Ontology Model has been tested through case study work that was carried out using knowledge instances of an extended project from factories in Motorola Technology Malaysia PLC and its participants for two-way radio design and manufacture.

7.1 The Common Ontology For Syntactic And Semantic Integration

Communication between project teams and different organizations within cross-disciplinary inter-working groups is often hindered by lack of clarity in the terms and vocabulary used. The context in which information is exchanged between individuals or companies can substantially affect its overall meaning and the way in which individual parties view and interpret the shared implicit and explicit knowledge. This is especially true in manufacturing because of the growing complexity of manufacturing information and the increasing amount of knowledge and information that needs to be shared and exchanged between companies. Manufacturing projects generally, but particularly extended projects team, may face problems when different terminologies are used by particular team members.

Commonly, people working within a particular company or group will develop their own vocabulary, or common terms for particular issues, elements or activities that they often work with. Hence, when people are brought together from different groups or companies, two common types of problem in communication can occur, firstly, that the same term is being applied to different concepts (semantic problem) and secondly, that different terms may be used to denote the same entity (syntax problem) [Lin et al. 2004].

A solution to this problem is the development of a taxonomy of manufacturing concepts and terms to make design knowledge effectively accessible across inter-related working group members. The knowledge needs to be explicit in a well-defined terminology that is accepted by all participating engineers. An approach for doing this, based on a Manufacturing System Engineering (MSE) Ontology that provides a common understanding of basic manufacturing concepts, properties of concepts, relationships and constraints among concepts, is proposed in this research.

7.2 The Structure Of MSE Ontology Model

Manufacturing System Information models are discussed in Chapter 2, and these include CIMOSA [Kosanke et al. 1999], MOSES [Ellis et al. 1994; Molina and Bell 1999], FDM [Harding et al. 1999] and MISSION [Harding et al. 2003]. These models describe the structure and relationships of data and information elements within manufacturing enterprise information systems.

However, these models have mainly been developed for intra-enterprise integration. Research projects, including the Enterprise Project [Uschold et al. 1998] and the TOVE project [Fox and Gruninger 1997], have focused on the concepts of ontology for developing a taxonomy and have defined an explicit specification of conceptualisation for virtual enterprise modelling. However, these virtual enterprise ontologies have put effort into the collection of terms and definitions relevant to general business enterprises, and are not focused specifically on the manufacturing system domain. The Process Specification Language (PSL) project [ISO/CD18629 2002] tries to develop a general ontology for representing manufacturing processes for the exchange of process information. PSL creates a neutral, standard language for process specification to integrate multiple process-related applications throughout the manufacturing life cycle. In a similar manner to PSL, the Standard for the Exchange of Product Model Data (STEP) effort aims to create an interlingua for exchanging manufacturing product data. Hence both PSL and STEP are focused on particular areas of manufacturing systems and therefore do not cover all the terminology aspects and needs that are necessary for the introduction of an EEMSE Moderator.

MSE is complex and covers many wide-ranging aspects [Hitomi 1996], requiring inputs from many skills and disciplines. A fundamental requirement of an MSE Moderator is that it should be able to support a multi-discipline team and therefore communication between team members may include terminology from several functional areas. Therefore, an MSE Ontology model is needed to bridge across multiple functional areas and the approach taken in this research is based on the combination of the above formalisms.

The objective of the MSE Ontology model is to support an EEMSE Moderator, which has been designed to support concurrent engineering and MSE within an extended enterprise environment. MSE is very complex and is generally performed by multi-discipline project teams. The design or redesign of a Manufacturing System (MS) must satisfy many different requirements and objectives so compromises generally have to be made to achieve a balanced design for the new or re-engineered MS. Project team members must therefore be aware (or be made aware) when decisions are taken which have a significant effect on other team members. When teams are large and located in multiple sites, this can be very difficult to achieve, and intelligent support systems are necessary.

The Moderator concepts aim to raise awareness and facilitate and improve team working by monitoring design decisions, evaluating their significance to individual project team members and communicating with any team members deemed necessary.

However, as explained in chapter 3, the original MSE Moderator reported in [Harding et al. 2003] was designed to operate within either a single enterprise or cooperating enterprises all using shared (common) information and terminology. The situation is very different within an inter-enterprises environment where many inconsistent and incompatible terminologies may exist and an MSE Ontology model is necessary to enable the EEMSE Moderator to proceed with its support activities. The MSE Ontology model therefore needs to enable the EEMSE Moderator to perform these activities by integrating the information and knowledge requirements of the required set of 'manufacturing' software applications through the shared and reused common manufacturing ontology.

The MSE Ontology model is presented, using an ontology modelling technique. This technique was discussed in the Chapter 5 which reviewed the Ontologies approach. Ontology modelling can be a useful method to develop and specify a representational vocabulary for a particular domain. In the context of this research, a manufacturing enterprise model has been developed.

All manufacturing enterprises are different, but they do have natural, common characteristics. The MSE ontology model has been captured in seven key base classes using the knowledge and experiences of published Manufacturing System Information models [Harding et al. 1999; Kosanke et al. 1999; Molina and Bell 1999; Zhao et al. 1999(a); Harding et al. 2003], in addition to the Extended_Enterprise class to support the inter-related enterprise environment.

The seven top-level classes: **Project, Flow, Extended_Enterprise, Enterprise, Process, Resource**, and **Strategy** are all abstract classes, so each represents a hierarchy of subclasses which are detailed and classified according to their main characteristics. Figure 7.1 shows elements of the class structure, relationships between classes and constraints on the valid values of a certain property that have been captured using Protégé / OWL and are displayed using its visualization plugin, ezOWL. The details of the software design and implementation environments will be described in the next chapter.

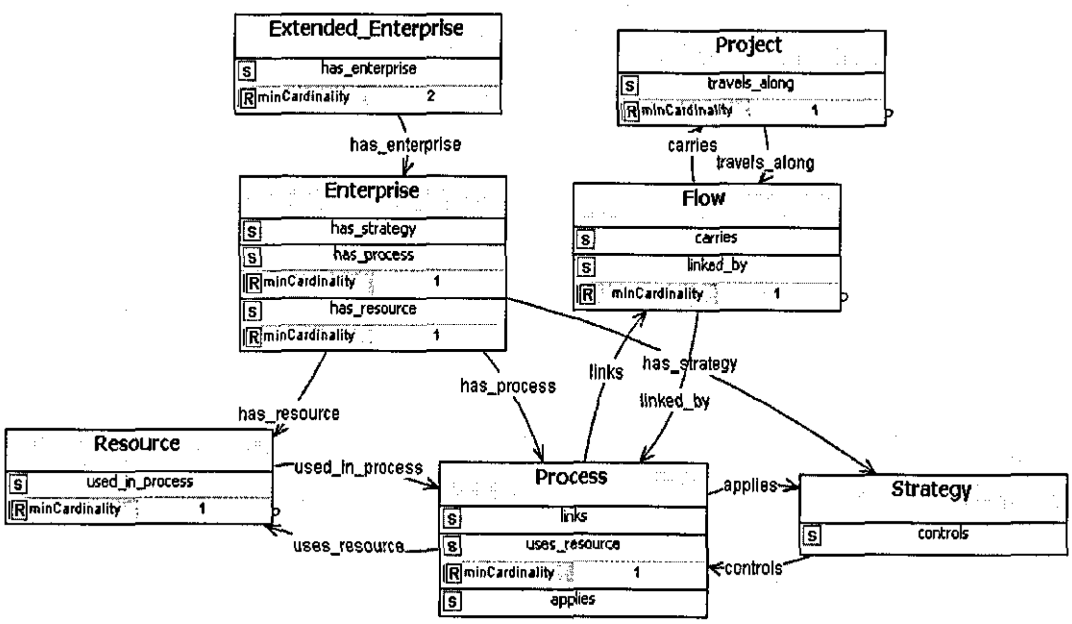


Figure 7.1: Top-level abstract classes from the MSE Ontology model

Boxes represent classes and arrows represent relations.

S Slot R Restriction

7.2.1 Project Class and Flow Class

As previously stated in the Chapter 1, the extended enterprise is formed to pursue a market opportunity and to achieve competitive advantage, since individual companies concentrate on their core competencies and outsource other business and service elements. In an extended manufacturing enterprise, several independent companies assemble a temporary consortium of partners and services for one or a limited number of specific projects in order to perform product development, design, engineering, and production preparation in close co-operation. The definition of the *Project class* is important as this can be considered as triggering the formation and operation of the extended enterprise MSE process.

The Project class hierarchy is used to represent the business objects, i.e. the things that flow through the manufacturing systems and processes. These can be either physical items, such as products or non-physical items, such as documents, or program. The Project class and a section of its hierarchy are represented in Figure 7.2.

Physical_item class, Program class, and Document class are the sub-class of the Project class. In addition, there are three sub-classes within the Document class: Customer_order class, Contract class and Drawing class.

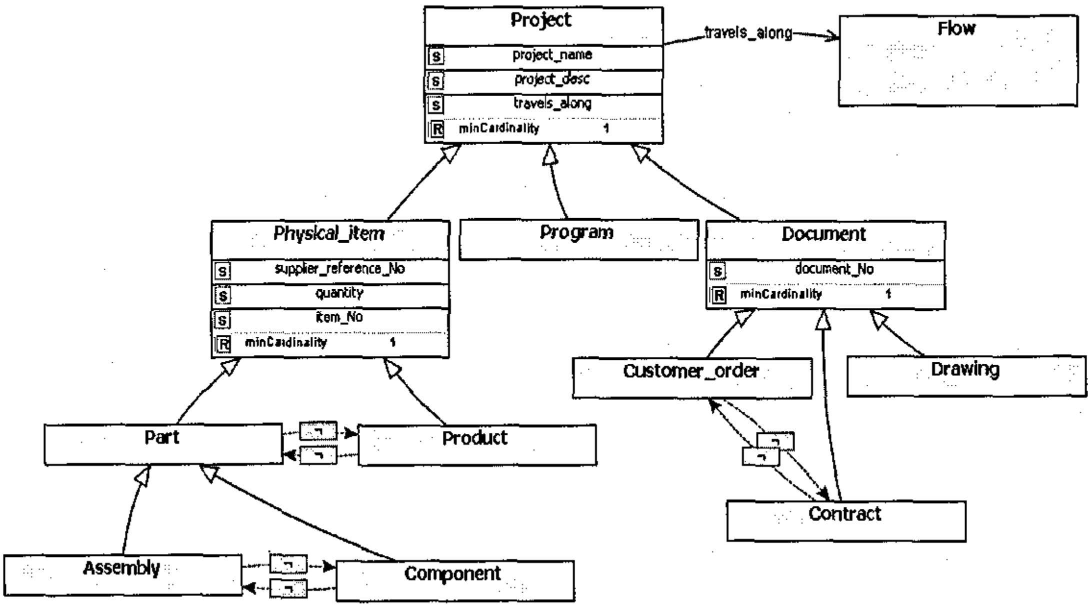
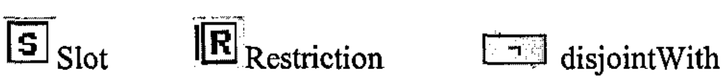


Figure 7.2: The Project Class Hierarchy from MSE Ontology Model



Using OWL, several constraints have been defined on the Project class and its properties. For example, the data in the Contract class will not be collected in the Customer_order class, as there is (owl:disjointWith) a constraint between both these classes. This can be illustrated by the following example where the Contract class captures information about the working practice for the consignment program which is a partnership with a selected customer to provide, schedule, and share items on a just-in-time basis that is not paid for until it is disbursed to customer. The intent of this contract is to reduce inventory levels, improve cycle time and provide the customer with flexibility to control the material. Therefore data in the Contract class are other than data in the Customer_order class that commits company funds and/or other resources. Documents such as Letters of Intent and Letters of Agreement are considered to be contractual documents that are the data in the Contract_class. If items are to be dispatched to a customer, a customer order will be generated. Documents such as customer order are captured as data in the Customer_order class.

Each instance of the Project class travels along at least (owl: minCardinality) one (but probably more) flows (instances of the *Flow class*) that connect independent processes or activities into a system with a purpose.

7.2.2 Process Class

All enterprises have functions, or processes, or perform activities as an essential part of their business [Bravoco and Yadav 1985]. The *Process class* describes something that can be done or a transformation that can be performed; there are business functions or activities that are essential to the operation of the extended enterprise. Figure 7.3 illustrates the common business processes in a manufacturing enterprise and a section of the Process class hierarchy. The Process class is the superclass of classes Production, New_product_introduction, Sales_marketing, and Financial_ control etc. The Production class includes several sub-classes, such as, Inventory_ management, Materials_management, Material_purchasing, Production_ planning, Product_assembly, Product_delivery, and Quality_assurance etc.

Process objects are defined and described by various important pieces of information, e.g. what resources are required for the process (through links to resources). The classes also capture how the process is measured and controlled (through links to strategies), and where the process is located, or the area of responsibility where the process takes place (captured by including links to enterprises).

Several class axioms have been defined in the Process class, such as the owl:intersectionOf axiom (Test class $\cap$ Customer_acceptance class) on the Quality_assurance class. That is, the product quality assurance depends not only on passing the quality test but also on being accepted as meeting the customer's requirements. Another axiom example is the owl:unionOf axiom (Raw_material $\cup$ Parts), see next section in Figure 7.4, on the Material_management class. This means, the information in either Raw_material class or Parts class (both are subclasses of the Resource class) will automatically link to Materials_management class (subclass of the Process class). Therefore, a semantically enabled MSE that could understand the manufacturing requirements of a particular design and link directly to a materials inventory system could then automatically generate overall materials requirements.

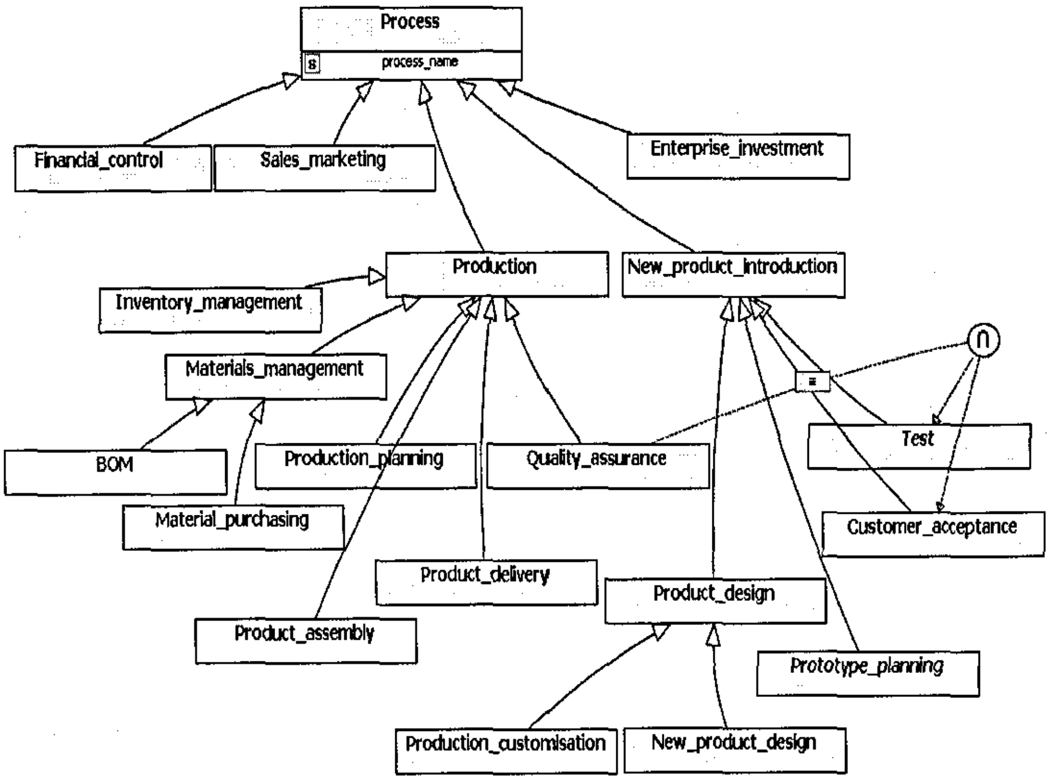


Figure 7.3: The Process Class Hierarchy from MSE Ontology Model

∩ IntersectionOf

7.2.3 Resource Class

Resources represent an important part of an enterprise's capability and have therefore been identified as fundamental entities in many other architectures, such as CIMOSA [Kosanke et al. 1999], FDM [Harding and Yu 1999] and the Enterprise Ontology [Uschold et al. 1998]. The *Resource class* describes mechanisms that enable a process to be executed. At a high level of abstraction, it could be a human resource, or a manufacturing resource, at a lower, more detailed level of abstraction, it could be a machinery tools, raw materials …etc (see figure 7.4). Resources may be described by various pieces of information, which may include: what the resource can do (through links to process), where it is located (through links to enterprises) and how it is allocated (through links to strategy).

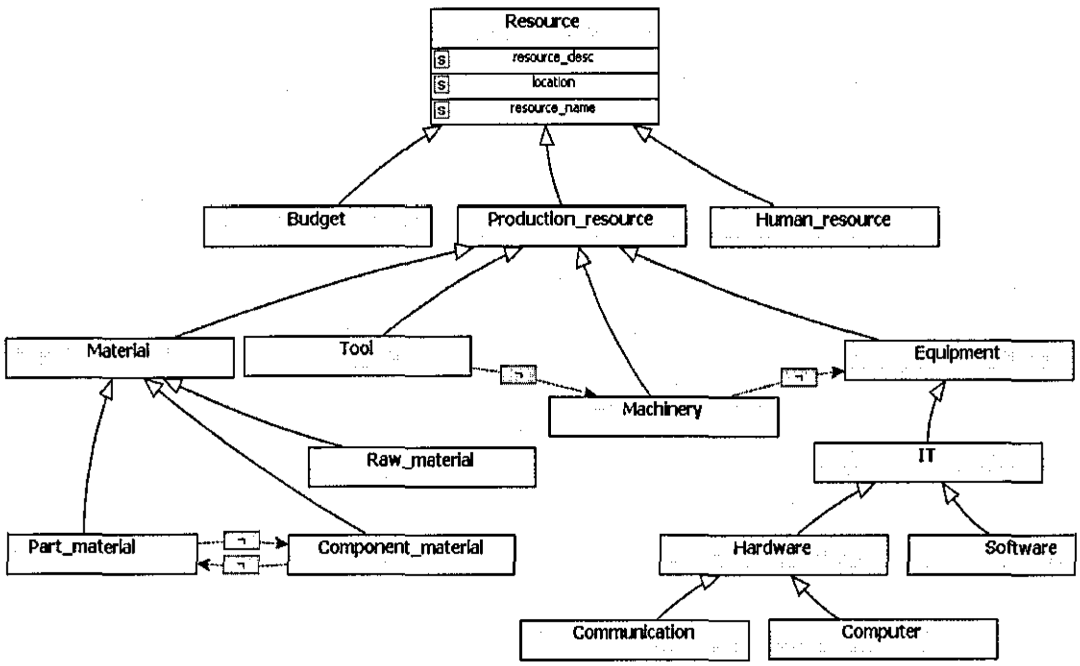


Figure 7.4: The Resource Class Hierarchy from MSE Ontology Model

7.2.4 Strategy Class

An important part of modern design and manufacture is to ensure that effective use is made of available manufacturing capability to achieve business and enterprise goals. Manufacturing and business strategy enables the enterprise to contribute to the long-term competitiveness. There is a need to represent strategy within the ontology, because the strategies represent the constraints, objectives, heuristics and other knowledge that can influence decisions made by the enterprise relating to the use of enterprise facilities, resources and process. For example, knowledge relating to operating costs of particular machines may affect choice of resources made for the manufacture of particular batch sizes of products. Similarly, knowledge relating to the current overall performances of its various facilities may influence a participating enterprise to dedicate output from one particular factory to meet the objectives of the current extended enterprise.

Molina [Molina 1995] believed that it was necessary to represent a company's strategic decisions and operational rules, in addition to its resources and process. The FDM model includes both a Strategic view and a Performance view, to ensure that developing designs can be regularly checked and their performance evaluated against strategic plans so that management can be confident that the proposed factory will meet their business objectives. The performance of an enterprise is significantly affected by the operational rules it adopts; therefore the determination of operational rules is an important part of enterprise redesign. In addition, the FDM research enables knowledge to be represented in a variety of ways and links Strategy objects with a knowledge representation model [Harding et al. 1999]which was discussed in section 3.2.4. In the MSE Ontology, the strategy concept is implemented from the FDM model. Figure 7.5 shows a section of the *Strategy class* hierarchy and the slots (properties) definition from the MSE Ontology in Protégé-2000.

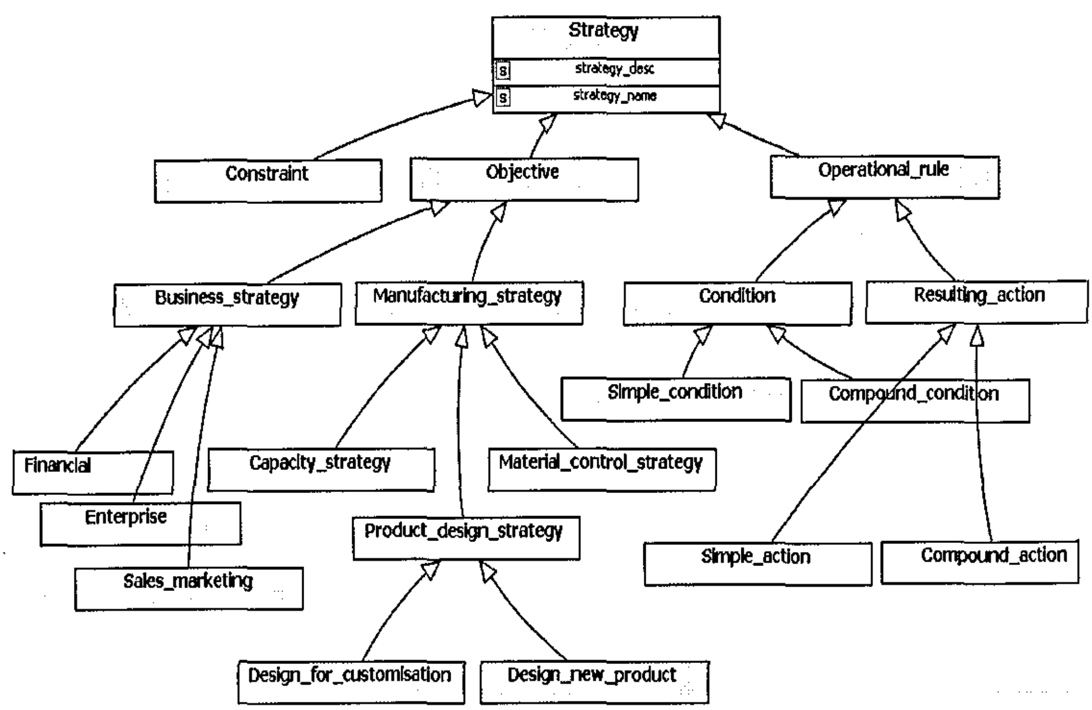


Figure 7.5: The Strategy Class Hierarchy from MSE Ontology Model

7.2.5 Extended_Enterprise class and Enterprise class

The *Enterprises class* is concerned with the representation of the capabilities and information within the extended enterprise. This is because in any specific virtual enterprise system, processes, resources, and strategies are arranged into different enterprises, related to their individual business objective and function. Zhao pointed out that global competition highlights the need for a more co-ordinated concurrent product development process within a multi-factory global manufacturing enterprise environment [Zhao et al. 1999]. Building on Molina's manufacturing model[Molina 1995], Zhao proposed that in the manufacturing data model, a manufacturing Facility can be considered to be either an individual machine (Station) at its lowest level, or a manufacturing Cell, Shop or Factory at higher levels, or a manufacturing Enterprise at the highest level. The class Facility is the super class of classes Enterprise, Factory, Shop, Cell and Station. The aggregation relationships between Enterprise, Factory, Shop, Cell and Station indicate that one enterprise object (e.g. a global enterprise) can consist of one or many factory objects, a factory object may have one or many shop objects and so on.

Zhao's manufacturing data model is intended to enable the manufacturing capacity of a particular facility to be reliably represented. However, his model focuses on the single multi-facilities global enterprise environment. As mentioned earlier, within the extended enterprises environment, the business processes of participating enterprises are aligned to external demands and their capabilities and resources are united and shared for a specific period of time for a specific business objective.

The MSE Ontology model encompasses multiple enterprises within an extended enterprise that produces products and provides services, be that in industrial, commercial, financial, educational or government sectors. It is intended to enable the manufacturing capacity and business capacity of a particular extended-enterprise and of each individual enterprise to be reliably represented. Therefore, the *Extended_ Enterprise class* has been defined which is an aggregation of Enterprise objects, each of which can be represented by its available facilities (e.g. factory, shop, cell, and station). The Enterprise class is therefore the super class of classes Factory, Shop, Cell, and Station. In addition, the aggregation relationships by Zhao are also included in the MSE Ontology. A representation of a section of the Enterprises Class hierarchy and aggregation relationships and instances will be explained and shown in figure 7.6.

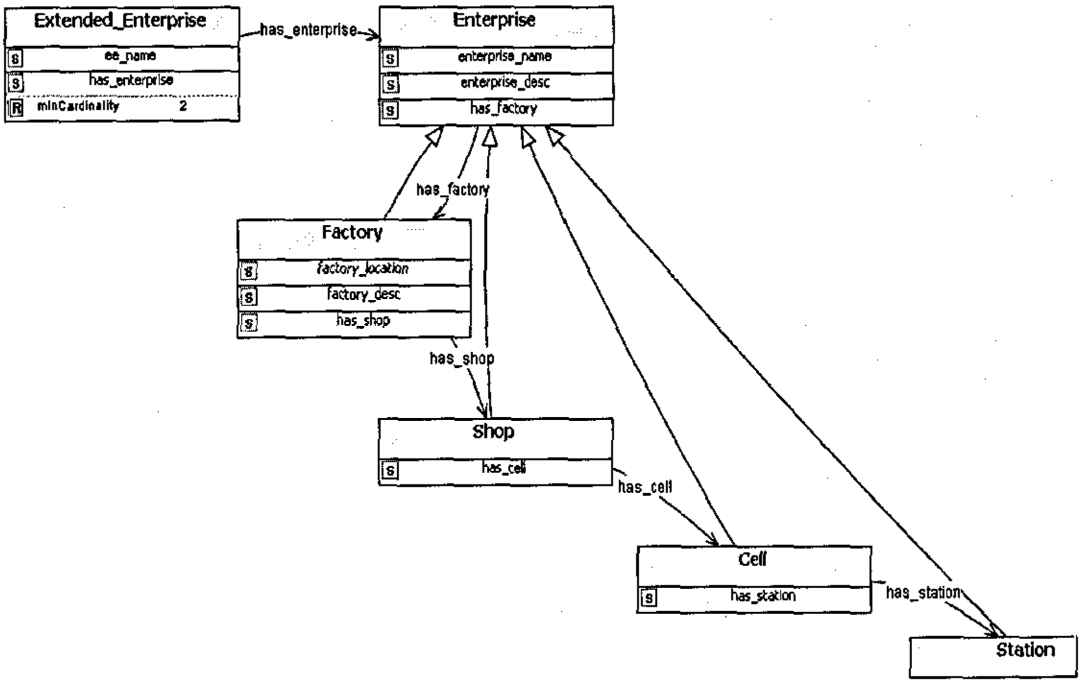


Figure 7.6: The Extended_Enterprise Class and Enterprise Class Hierarchy From MSE Ontology Model

7.3 Implementation Of The Instances Of The MSE Ontology Model

The MSE Ontology model uses Protégé http://protégé.stanford.edu/ and its Plugins as a basis for expressing ontologies and converting the informal vocabularies into the formal language – RDF/RDFS/OWL. This will also be illustrated using its visualization plugins, such as OntoViz Plugin and ezOWL, as shown in figure 7.7.

Protégé 2000:

Protégé is a graphical tool designed to automate the process of building domain-specific knowledge acquisition and knowledge based systems. It was chosen because it provided all the required functionality and is widely used by academic researchers. Protégé is an ontology editor, which can be used to define classes and class hierarchy, properties (Protégé calls these slots) and slot-value restrictions, relationships between classes and properties of these relationships. The instances tab is a knowledge-acquisition tool which can be used to acquire instances of the classes defined in the ontology. In addition to creating a Protégé-based editor for a new Semantic Web language [Noy et al. 2001], developers can plug in other applications in the knowledge-base-editing environment. In this research, the Semantic Web Language plugin and Visualization plugin were used as follows:

OWL Plugin is an extension of Protégé with support for the Web Ontology Language (OWL), which enables users to:
Load and save OWL and RDF ontologies.
Edit and visualize OWL classes and their properties.
Define logical class characteristics as OWL expressions.
Execute reasoners such as description logic classifiers.
Edit OWL individuals for Semantic Web markup.

ezOWL Plugin is a Visual OWL Editor for Protégé-2000.

Onto Viz Plugin allows Protégé ontologies to be visualized with the help of highly sophisticated graph visualization software (Graphviz from AT&T). The visualization is highly configurable and includes:
Picking a set of classes or instances to visualize parts of an ontology.
Displaying slots and slot edges.
Specifying colours for nodes and edges.

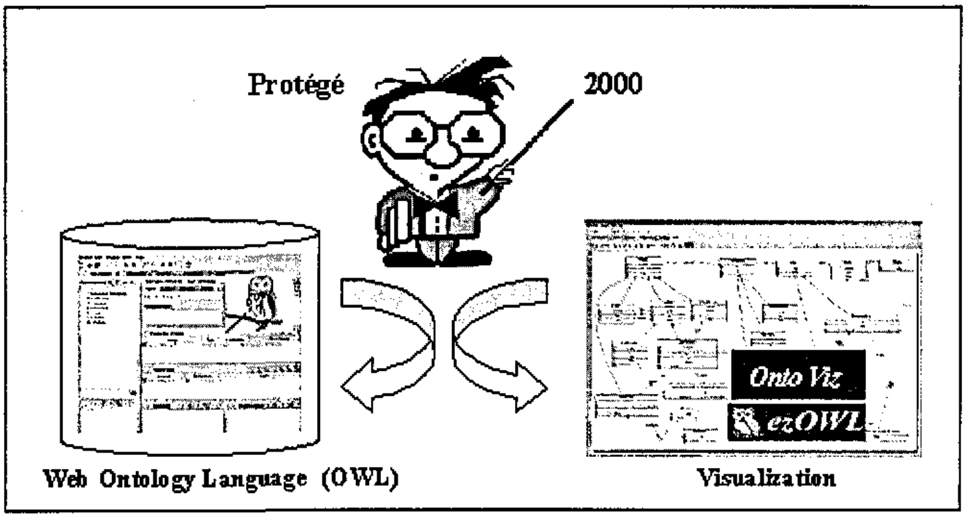


Figure 7.7: The Ontology Software Implementation Environment

7.3.1 The Knowledge Base Terminology in Protégé-2000

The following is a list of terms specifically used in the description of Protégé-2000 http://protégé.stanford.edu/ and to explain the proposed MSE Ontology Model:

Knowledge-based system	A computer system that includes a knowledge base about a domain and programs that include rules for processing the knowledge and for solving problems relating to the domain.
Problem-solving method	A computer program that is used in conjunction with a knowledge base to answer questions or solve problems.
Knowledge-acquisition tool	A tool used to build a knowledge base by acquiring instances.
Ontology	A model of a particular field of knowledge - the concepts and their attributes, as well as the relationships between the concepts. In Protégé-2000, an ontology is represented as a set of classes with their associated slots.
Domain	A particular field of knowledge, such as a manufacturing enterprise system.
Class	An abstract representation of a concept in a domain as a collection of related classes. For example, a manufacturing enterprise model might have *enterprise*, *factory*, and *shop data* as classes. A class can have a set of slots that represent the attributes of the class.
Inheritance	A parent-child (superclass-subclass) relationship between two classes. A child (subclass) inherits the slots of its parent classes (super classes).

Slot	An attribute of a class. For example, an *enterprise class* might have *name*, *title*, and *phone number* as slots.
Inherited Slot	A slot is attached to a class via inheritance from a parent class.
Slot Type	A slot that identifies the kind of values a slot may have - Boolean, Float, Instance, Integer, String.
Instance Type (slot type)	Type of slot whose value is the instance of a class.
Cardinality	A slot facet that describes whether the slot has just one value (*single*) or more than one value (*multiple*).
Instance (KB value)	Concrete occurrence of information about a domain is entered into a knowledge base. For example, "Fran Smith" might be an instance for a Name slot.

7.3.2 Case Study Backgrounds

In order to illustrate the MSE Ontology model, an extended project from factories in Motorola Technology Malaysia PLC and its participants for two-way radio (shown in figure 7.8) design and manufacture has been used.

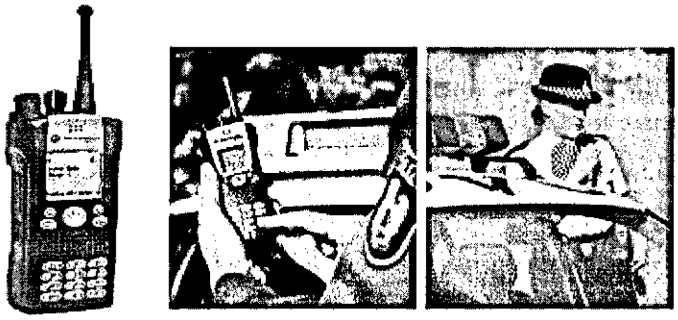


Figure 7.8: Two-way Radio

A two-way radio is a portable communication device used for short distance applications. The main functions of a radio are to transmit and receive audio signal to and from another radio or a group of radios tuned to the same frequency. In addition to the main functions, there are other supporting functions such as interfacing with the user, and securing and protecting the devices in the radio during various operating environments. For example, the MTP 700 is the TETRA[7] portable radio combined with the integration of voice, data and encryption in one unit for operational users in mission-critical environments, such as public safety agencies and transportation operators. Figure 7.9 illustrates an exploded view of a radio with the component list of MTP 700 [Motorola Inc 2002].

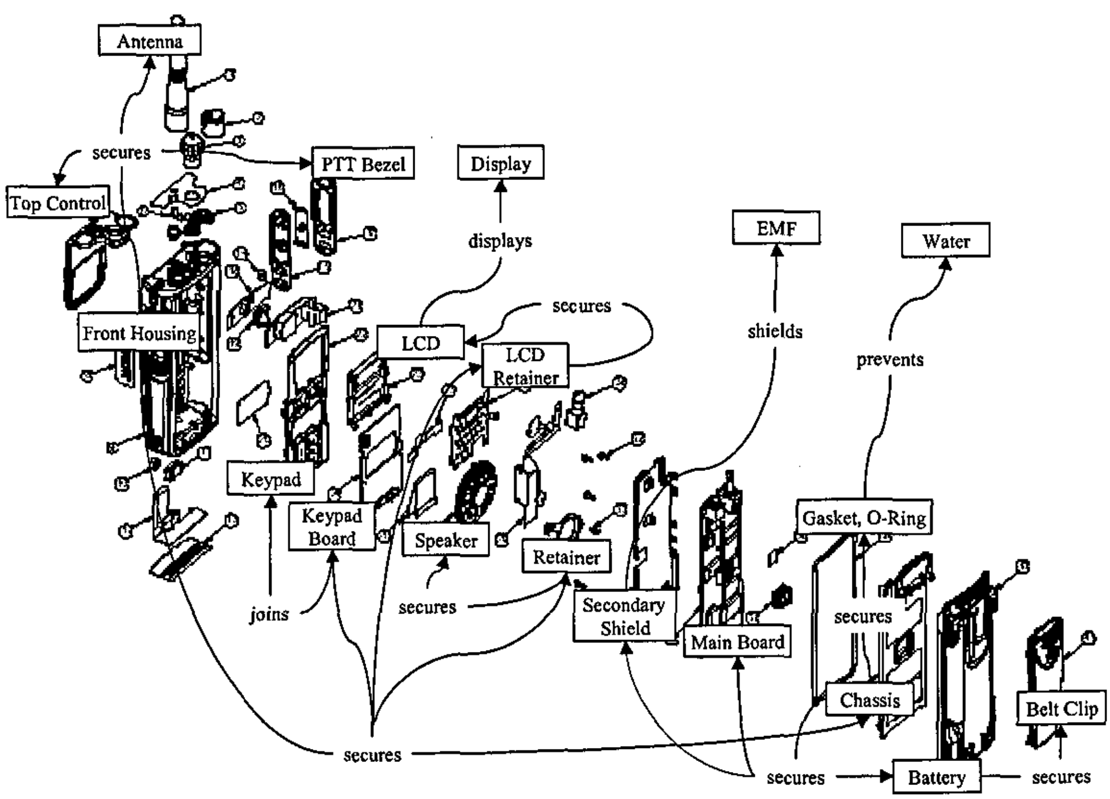


Figure 7.9: An exploded view of a Two-way Radio

[7] TErrestrial Trunked RAdio (TETRA) is an open digital trunked radio standard defined by the European Telecommunications Standardisation Institute (ETSI) to meet the needs of the most demanding professional mobile radio users.

The case studies are carried out in two stages; the first stage is the knowledge instances acquisition using MSE Ontology model with OWL primitives. The objective is to test the common meta-models for semantic and syntax interoperability between different MSE applications. The second stage is to evaluate the possible moderation functions provided by EE MSE moderator.

7.3.3 Case Study Examples

The case study examples were carried out by telephone interview, e-mail correspondence, and Web information from Motorola Technology PLC and Unitech Printed Circuit Board Corporation. Not all possible instances were covered by the case examples due to the availability of the data. Furthermore some of business data, was deemed confidential and could not be released by the company. Hence some approximate values have had to be assumed by the researchers to complete the scenarios. For example, costing information, and the scenario of the EE project: MTP 700 Two-way radio. However, it is felt that the approximated values are sufficiently representative to verify the MSE ontology model.

<u>Example 1: Planning and control of order flow for two-way radio extended project</u>

The extended two-way radio assembly project (eeproject_name: MTP 700 Two-way radio) using the MSE ontology model is now presented. Initially, a new contract (contract_no : MTP700/16/06/03) shown in figure 7.10 is defined as an instance of the Contract class (the subclass of the Documents class and Project class). Each instance of this class contains the properties of contract_no, and contract_date, ...etc, and inherits all the properties of its super class, such as project_name, project_team, and travels_along. Additionally, the slot type of the travels_along property is an instance type that allows definition of relationships between the Contract class and Flow class. Therefore each instance of travels_along points to an instance (flow_name = Order entry for contract number: MTP700/16/06/03 in figure 7.10) of the Flow class to build the relationships between these two classes. The linked_by property of Flow class then connects the independent processes into a system with a purpose. For example, the linked_by property connects the material purchasing process in order to obtain new parts (e. g. Main PCB, speaker) for the production. Another example, the linked_by property connects the production planning process for the production scheduling. This new order entry for contract number: MTP700/16/06/03 had the production planning process with String-value type instances, such as start_production_date = 09/05/04 and end_production_date = 09/08/04. Furthermore, production-planning process requires several resources for the process, e.g. production resource, human resource, through the uses_resource property attached to the Resource class.

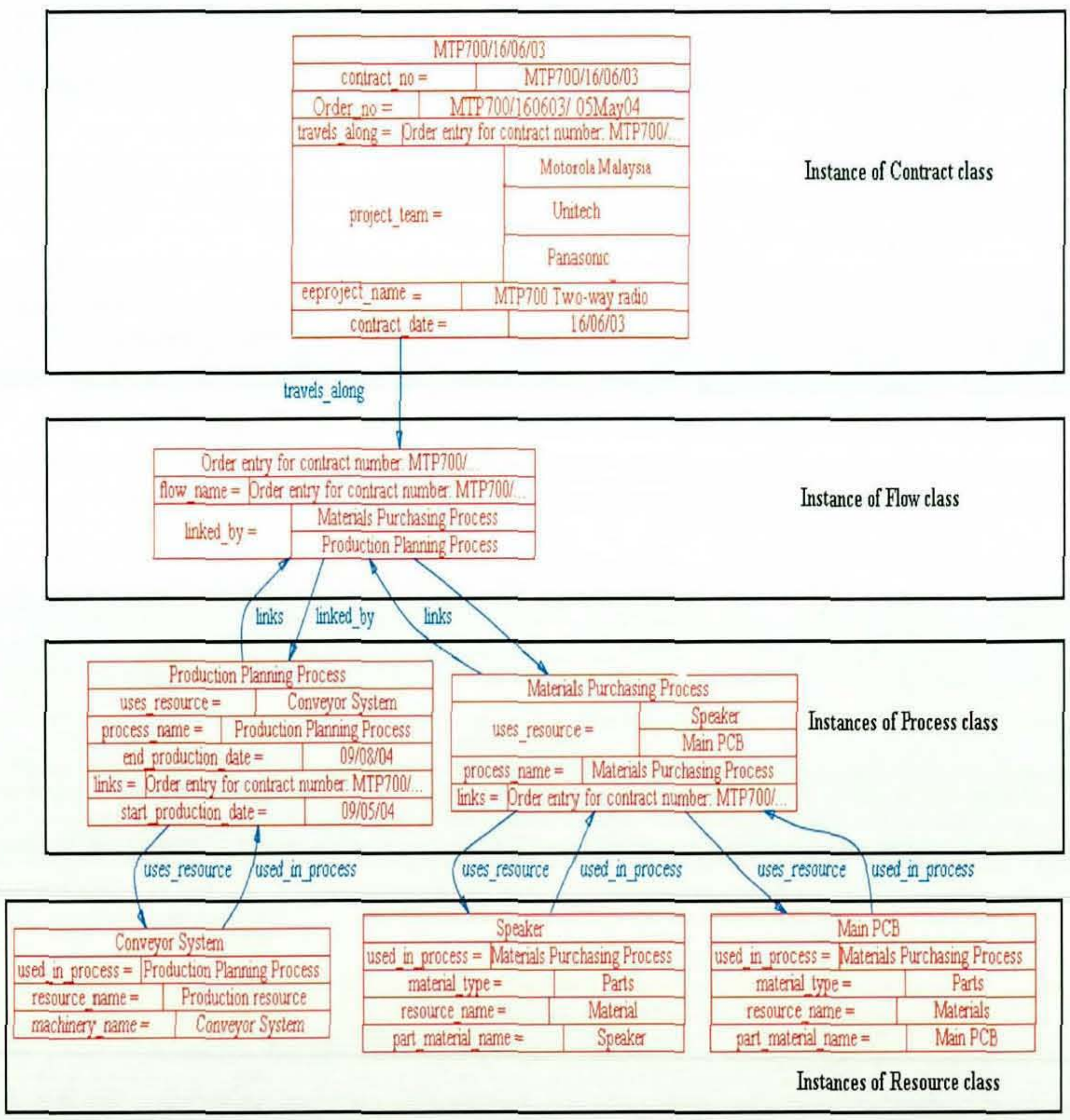


Figure 7.10: Instances of planning and control of order flow for MTP700 extended project

(Red for instances, blue lines as relationship)

Example 2: The extended project team structure

The project (ee_name: MTP 700/16/06/03) shown in figure 7.11 is defined as an instance of the Extended_Enterprise class, which has several enterprises involved in this extended project. Only three enterprises have been listed in this example, i.e. Motorola Malaysia Inc., Unitech Printed Circuit Board Corporation, and Panasonic and these are created as instances of the Enterprise class. Unitech is one of the PCB suppliers and Panasonic PLC is one of the battery suppliers.

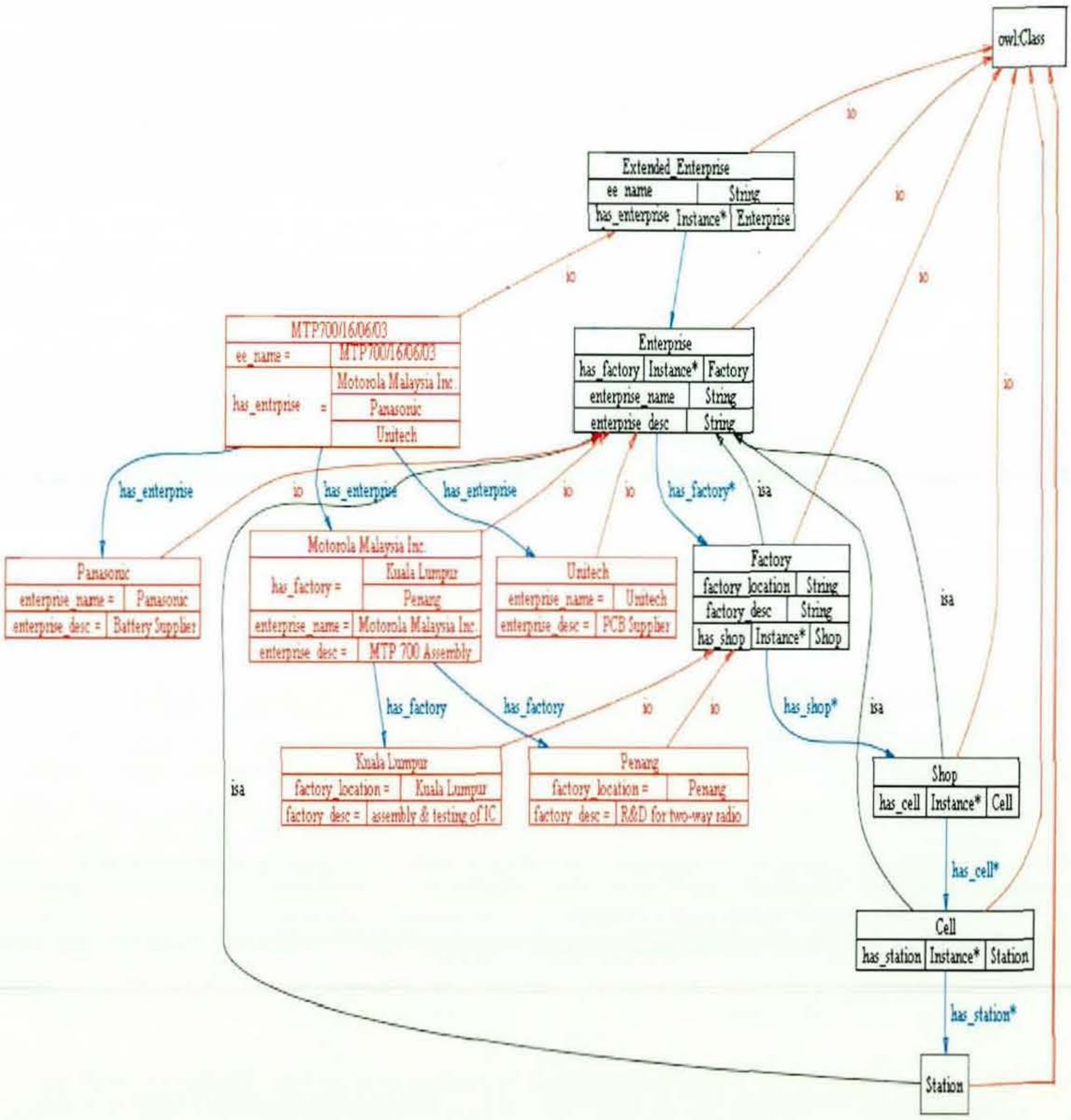


Figure 7.11: Classes, instances and relations among the extended two-way radio assembly project team.
(Black for classes, red for instances, blue lines as relationship, isa lines as subclass-of , io lines as instance-of)

Motorola Malaysia Inc. is the leader of the project and the distribution centre for two-way radio. It has responsibility for the total value chain of manufacturing that includes distribution and sales, of products manufactured or products procured. It is also responsible for the design and development of the two-way radio and provides after-sales repairs and parts replacement service to customers. Motorola Malaysia Inc. has two factories, Kuala Lumpur and Penang as instances of the Factory class. The main two-way radio production is located in its Penang factory.

Example 3: Materials Management Process

There are several business activities or processes that are essential to the operation of the project (ee_name: MTP 700/16/06/03). Individual participating enterprises (or factories, cell, ...etc) have their responsibility for specific processes for the MTP 700/16/06/03 project. This example shows Motorola Malasysia's Penang factory after the material purchasing process. Figure 7.12 shows examples of Material_purchasing class's properties and instances: process_name = Material Purchasing, purchasing_no = MTP700_023323, purchasing_group = 6, order_date = 12/05/04, delivery_date = 22/08/04, vender = Unitech, and has_item = mtp2 and mtp3.

Figure 7.12 also shows the one to many relationships between Material_purchasing process class and BOM process class via the relationship property (has_items). Therefore, the details of purchasing items are displayed and collected in BOM class. The BOM class's properties and instances are, for example, item_no =mtp2 (purchasing line-item number), material_no = BGA045 (vendor's part's number), materials_desc = BGA, and unit_prices = 8.0.

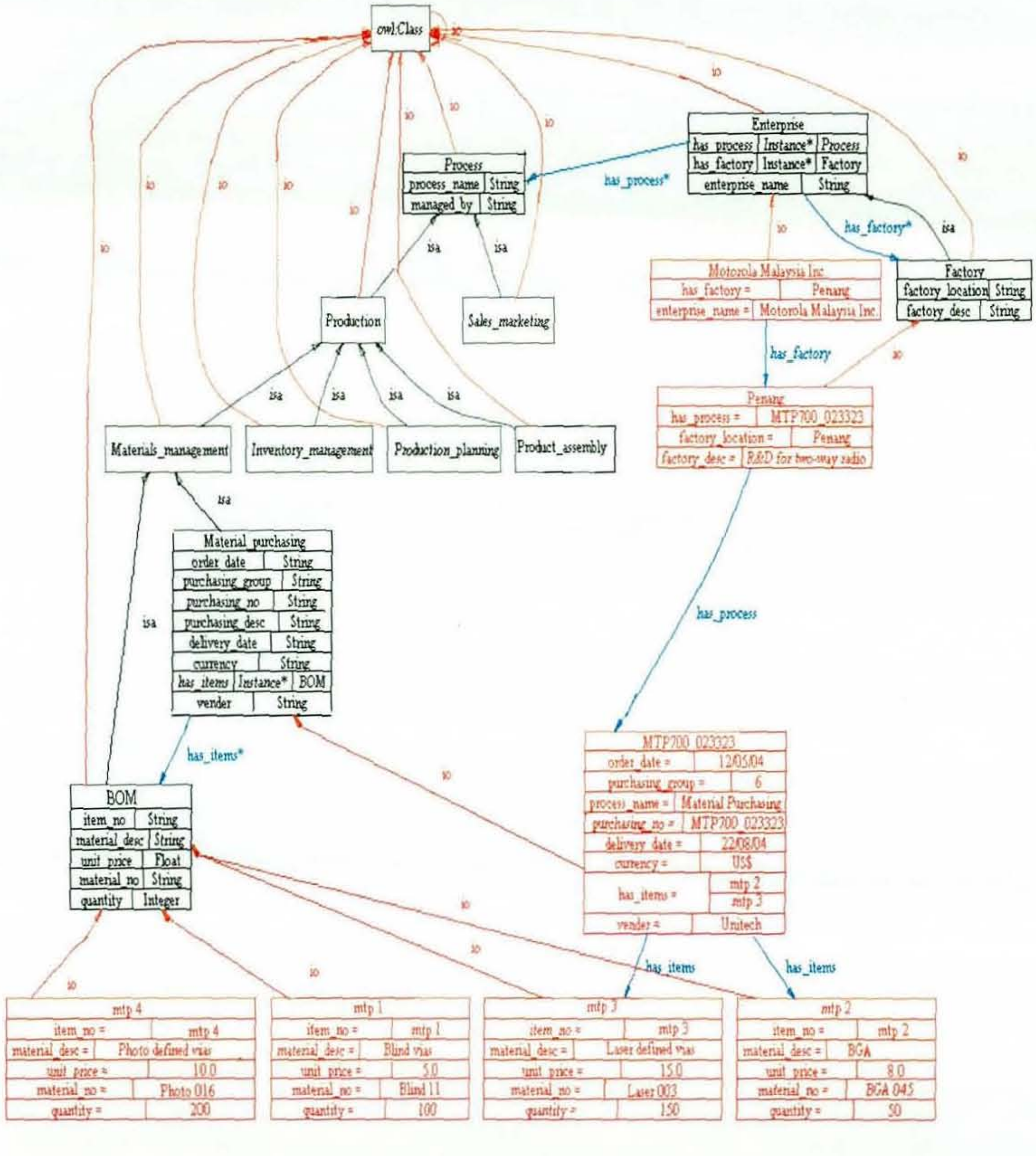


Figure 7.12: Instances of the materials management process (Black for classes, red for instances, blue lines as relationship, isa lines as subclass-of , io lines as instance-of)

Example 4: Production resource location

Figure 7.13 shows the resource (Production resource: Machinery, Equipment, Tool, Material) location. Particularly it illustrates the location of some production resources for the two-way radio. As previously stated in figure 7.13, an exploded view of a Two-way Radio, shows that it is composed of several parts, such as, antenna, main PCB, LCD display, speaker, keypad, keypad board, battery, belt clip, ...etc. Not all possible resource instances were covered due to the availability of the data, but a representative example set of instances is shown here. In this example, the instances of Part_material, keypad and keypad board, are located at Motorola Malaysia 's Penang factory's assembly store. In addition, another Part_material resource, Main PCB, and Machinery resource, Conveyor System, are located at the same factory, but in a different location, this time being on the production floor.

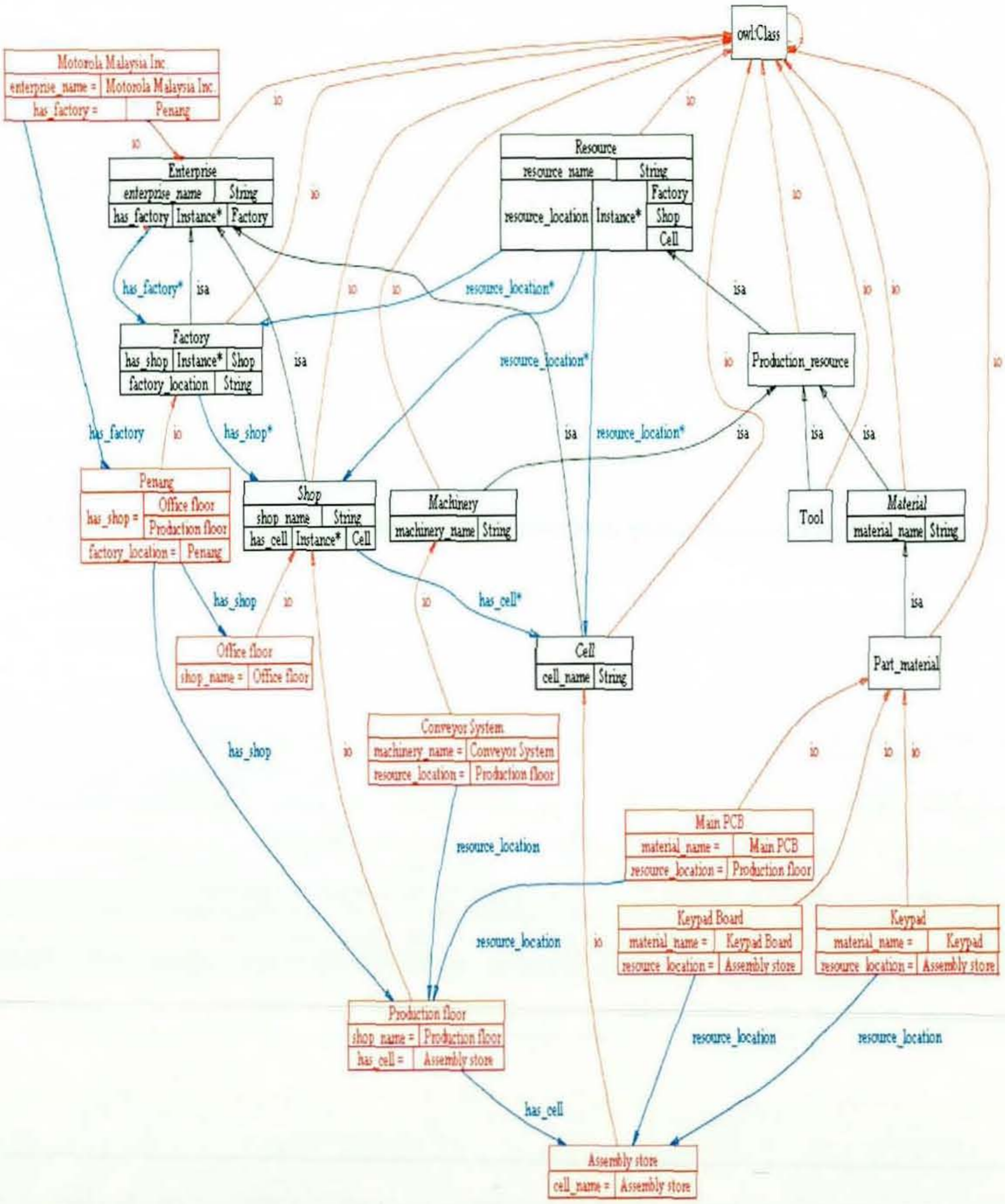


Figure 7.13: Classes, instances and relationship of resources location (Black for classes, red for instances, blue lines as relationship, isa lines as subclass-of , io lines as instance-of)

Example 5: Integration example

As can be seen from the integration example in figure 7.14, MSE systems have been created independently in an extended enterprise, and do not share the same semantics for the terminology of their manufacturing models. Figure 7.14 shows the semantic and syntactic integration by mapping to the common MSE Ontology. The mapping process scenario requires the following steps:

1. First, all the participating models are presented with the documented conceptual model in the common ontology language, i.e. the OWL model in this research.

2. Then equivalence mappings between the terminologies and the common MSE Ontology are specified.

3. Finally, a set of reuse inference rules are developed that encode the mappings between classes and their properties.

OWL provides built-in ontology mapping support, that is, a particular class or property in one ontology is the same as a class or property in another ontology (owl:sameClassAs, owl:samePropertyAs). These OWL primitives and our MSE Ontology model have been applied as a mediate service for enhancing information semantic and syntax integration within an extended enterprise community.

For example, different information models may be used by different parts of the extended enterprise project teams. Assume initially that some participants in MTP 700_090504 project use information models in their business. If, for example, Motorola used information models based on the FDM model and Unitech used information models based on the Mission model, are shown in figure 7.14 and each of these has been built to meet the objectives of different companies needs.

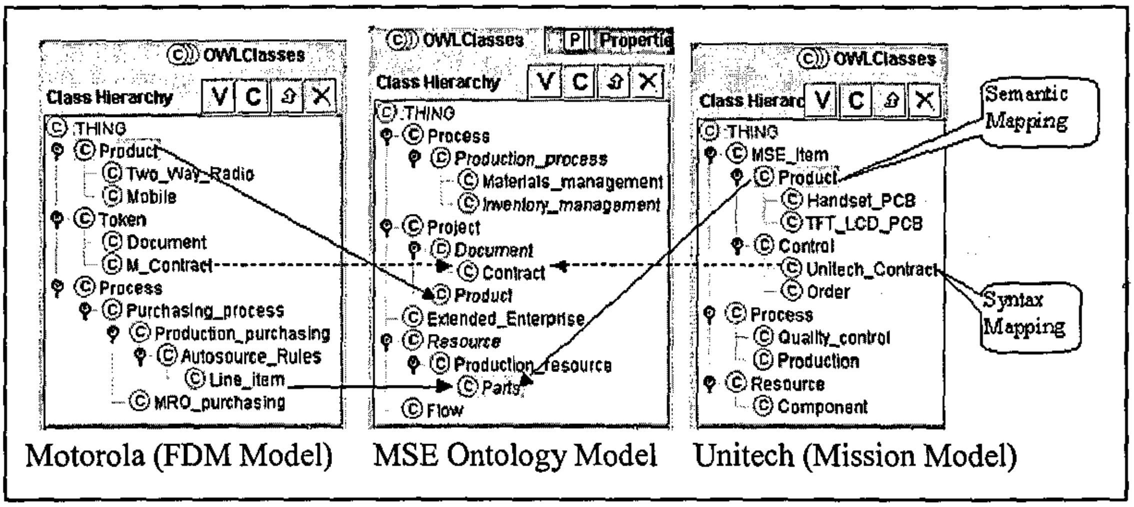


Figure 7.14: Ontology Mapping To The Common MSE Ontology Model

The aim of the extended project (ee_name: MTP 700/16/06/03) is to plan a contract flow for building MTP700 Two-Way Radio. Motorola and Unitech and other extended participators will work together to fulfil this contract. Both the M_Contract class in Motorola and Unitech_Contract class in Unitech models correspond to a common concept of an *object*, the Contract class in the extended project (ee_name: MTP 700/16/06/03). The syntax problem will occur in the M_Contract class and Unitech_Contract class within the extended project environment. This syntax problem of applications can be parsed by Extensible Stylesheet Language (XSL) to transform an XML document from one form to another. However, by using an ontology approach, an intermediate *communicator* is adopted, and this reduces the number of mappings by requiring that an application only map its concepts to the concepts of a common ontology rather than mapping to all the other applications, see figure 7.15.

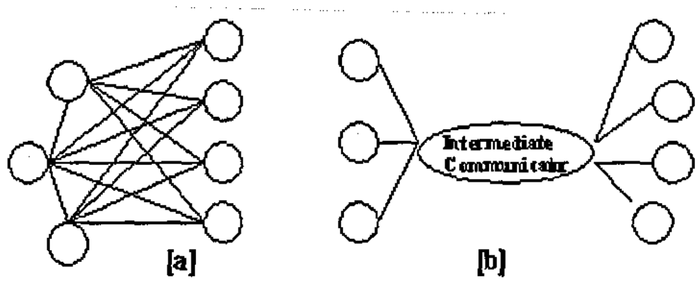


Figure 7.15: Reduced mappings due to intermediate communicator

Both the FDM and Mission models include the same term, i.e. Product class. However, the information stored in each Product class is different. These two classes collect different data and therefore represent different meanings, since Product in the Mission model is applied to the collection of the data relating to Unitech's core products (e.g. Handset PCB, TFT_LCD PCB) as Unitech is a major PCB manufacturer. The products, such as Handset PCB, TFT_LCD PCB, are parts to produce Two-Way Radio. In contrast, the Product class in Motorola model is designed to collect Motorola's finished product, which are Two-Way Radio or Mobile. As a result, the semantic problem occurs for the Product class.

The MSE ontology is proposed to facilitate application interoperability by developing a common ontology to interpret the MSE design concepts for meeting the needs of those applications. For example, Unitech's Product class links to the Parts class in the MSE Ontology model, as it is a production resource for the extended project. On the other hand, the Product class in Motorola needs to link to the Product class in MSE Ontology model.

OWL's built-in ontology mapping axioms (owl:sameClassAs, owl:samePropertyAs) are applied in our implementation. Figure 7.16 illustrates the semantic and syntax integration for all systems to map into the common MSE Ontology using OWL primitives.

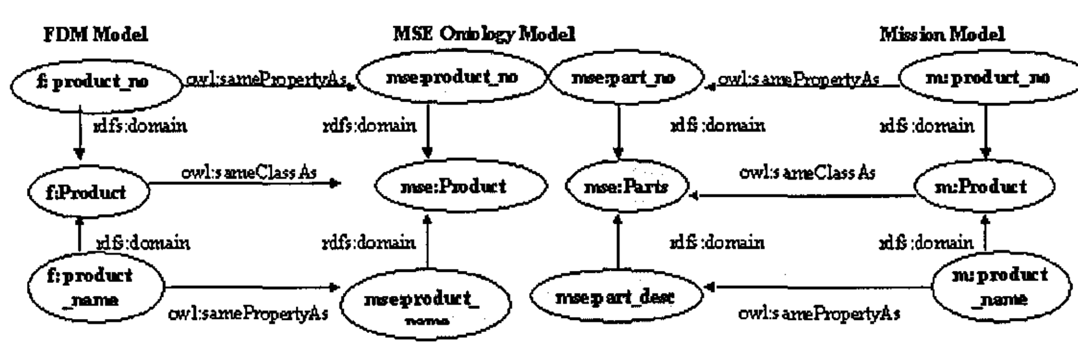


Figure 7.16: Mapping to the MSE Ontology Model using OWL primitives

Chapter 8 : The EEMSE Moderator Prototype

This chapter mainly discusses the architecture for formation of the EEMSE Moderator (EEMSEM) prototype which includes four major modules: *Ontology Acquisition Module*, *Ontology Mapping Module*, *Knowledge Acquisition Module* and *Design Moderation Module*. The integrated inter-enterprise system architecture, focusing on how to support ontology-based knowledge management and the conflict moderation work of the EEMSEM has also been demonstrated through an e-purchasing case example.

8.1 The Concept Of The EEMSE Moderator

The main function of the EEMSEM is to coordinate expertise and support the role of concurrency within the engineering activities of the inter-enterprises environment. The growing complexities of engineered systems are generally performed by multi-discipline project teams. The design or redesign of any part of the manufacturing system must satisfy many different requirements and objectives so compromises generally have to be made to achieve a balanced design for the new or re-engineered manufacturing system. Project team members must therefore be aware (or be made aware) when decisions they are taking may have a significant effect on other team members, such as constraining or even compromising other contributions to the re-engineering process. When teams are small and can meet regularly to discuss the project, team members are easily made aware of other peoples' requirements and views. However, when teams are large and located at multiple sites (or different global locations) this awareness can be difficult to achieve, and the task is further complicated when team members come from an inter-enterprise environment.

The MISSION MSE Moderator structure has been used as the basis for the initial work on the EEMSEM, and hence, details of the basic moderator concepts and structure can be found in the chapter 3. However, two substantial differences do exists in the case of the EEMSEM, and these are: [Lin et al. 2004]:

1. Design information changes (including additions or deletions) are expressed in different languages and terminology and

2. Information or knowledge of what team participants consider being important aspects of the design (e.g. key variables or values) is expressed in different languages or terminology.

The first difference directly affects the EEMSEM's design moderation process and the second difference affects both the EEMSEM's design moderation process and its knowledge acquisition process. The MSE Ontology has therefore been proposed and experimental implementations undertaken, to make the concept of an EEMSEM possible, by providing a mechanism for dealing with these differences. Therefore, the major goals of the EEMSEM are:

- To provide an interoperability mechanism with well-defined semantic definitions of an MSE Ontology Model, which is committed to by all participating extended project team partners. The model allows each of the partners to keep his own individual language via mapping to the cross-understanding MSE Ontology to support information autonomy.

- To reduce the complexity of EEMSE systems by providing a set of knowledge of the profiles and characteristics of participants within the extended enterprise group and communication mechanisms to orchestrate dialogues between them. The communications mechanisms are used to disseminate information about detected conflict or potential conflict[8].

[8] If a design change made by designer A has implications or causes problems for designer B, we say that it causes ***conflict.***

8.2 Architecture Of The EEMSE Moderator Prototype

The proposed design of the EEMSEM is as an intelligent software system operating on an extranet-based platform which is open and supports execution of distributed web applications on the WWW. Therefore the set of MSE software applications can work together in a global EE / VE environment. The developed approach enables plug-in of the EEMSEM onto any extended project team's extranet platform directly, as shown in Figure 8.1. Each of the MSE software applications performs a different role in the design and operation of the EE/VE and consequently supports a different area of expertise. Each MSE application contributes to the EE/VE, whilst functioning from a different enterprise, which is part of the current EE1 or EE2 or ...EE*n setting.

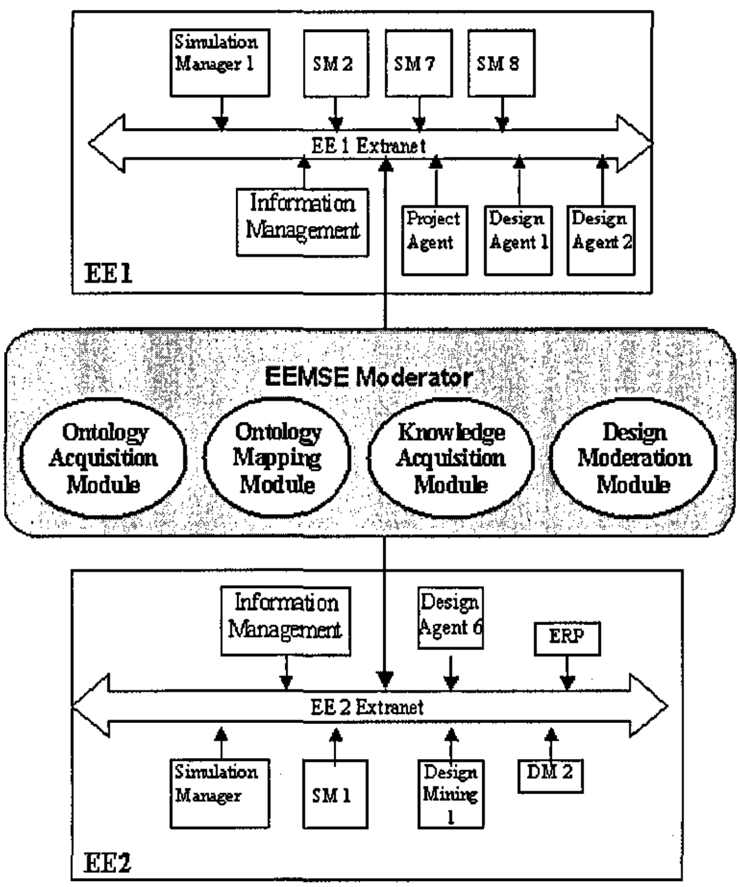


Figure 8.1: The general architecture for the EEMSE Moderator

The EEMSEM includes four major modules: *Ontology Acquisition Module*, *Ontology Mapping Module*, *Knowledge Acquisition Module* and *Design Moderation Module*.

The designs of Knowledge Acquisition Module and Design Moderation Module are largely from the implementation of MSE Moderator in the MISSION project as described in chapter 3. The main contribution to the EEMSEM in this thesis lies in the application of a new ontology approach and semantic web technology for knowledge and information integration. For this purpose, an Ontology Acquisition Module and the Ontology Mapping Module provide a language interoperability service and allow the individual enterprise to keep its own individual language. These two new modules therefore contribute strongly to the major novelty of this research. The details of each module will be discussed in the following sections.

8.2.1 Ontology Acquisition Module (OAM)

The EEMSEM's design moderation process should be activated whenever a change is made to information related to the inter-enterprises' joint project. The proposed design of the OAM is to establish a common, mediated, or integrated ontology which allows MSE users to access various heterogeneous data repositories from the domain of manufacturing engineering reference. Since different MSE information models have been independently developed by different enterprises or MSE design agents, they will include semantic heterogeneity (different vocabularies, logical schemas), structural heterogeneity (different data structures: plain files, databases, and WWW documents), and operational heterogeneity (some data repositories are accessed using SQL commands, others by Web browsers, and some of them do not have a standard query language).

One of the first steps in developing the EEMSEM is to acquire the common / mediated ontology created by a particular EE / VE group, describing explicit knowledge in a well-defined terminology that is accepted by all participating engineers, and this is called the Extended Project Team Ontology (EE Ontology). The EE Ontology needs to be built to meet the needs and objectives of the particular interdisciplinary project. Additionally the EE Ontology should be extensible and changed as necessary, as EEs / VEs in general, must be able to handle contingencies and new opportunities. However, any particular EE Ontology needs to be primarily focused on the needs of the current project since the EE / VE is disbanded when the goal has been achieved, the project is completed and participating companies go their individual ways, or recombine to form further EE / VE. In the proposed architecture, the MSE Ontology model (as described in chapter 7) is used to illustrate the manufacturing system domain and cover all the terminology aspects and needs for an EEMSE Moderator. It therefore serves as a core for the complete, extensible or reorganise structure of the individual EE Ontology, as shown in figure 8.2.

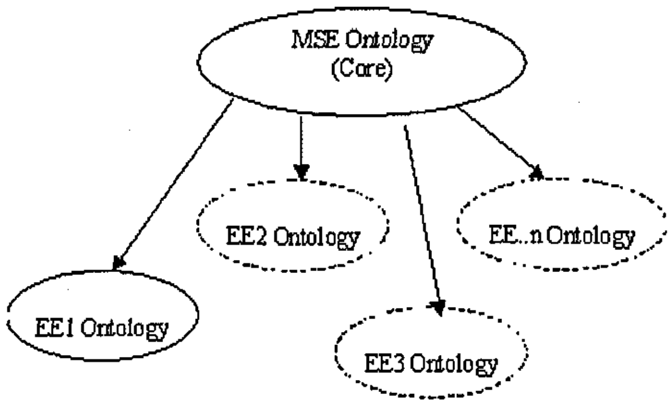


Figure 8.2: Proposed EE Ontology Architecture

The complete definition of the EE Ontology, mapping information, and mapped target ontology are stored and accessed through the EE Ontology Server. Figure 3 shows the instantiation of the general architecture for the EE1 or EE2 or EE*n Ontology Server of OAM on the EE1 or EE2 or EE*n extranet platform.

The EE Ontology Server provides the mediated terminology for the individual enterprise's documents within this particular EE group and therefore each enterprise can use its own individual language through mapping into the mediated EE ontology. In addition, the EE Ontology Server also stores the information about the mapped target ontology. Therefore, individual MSE design agents from different enterprises could share information and exchange documents through the EE Ontology Server. That is, the proposed design of the EEMSEM enables it to see and interpret the information stored in the EE Ontology Server and use the content to perform its moderation activities. Mapping details of any identified 'change' into the neutral EE Ontology enables the EEMSEM to perform most of its moderation activities by using its own, single chosen language. The mapping is carried out by the Ontology

Mapping Module, and the functionality and structure of this module will be discussed in the next section.

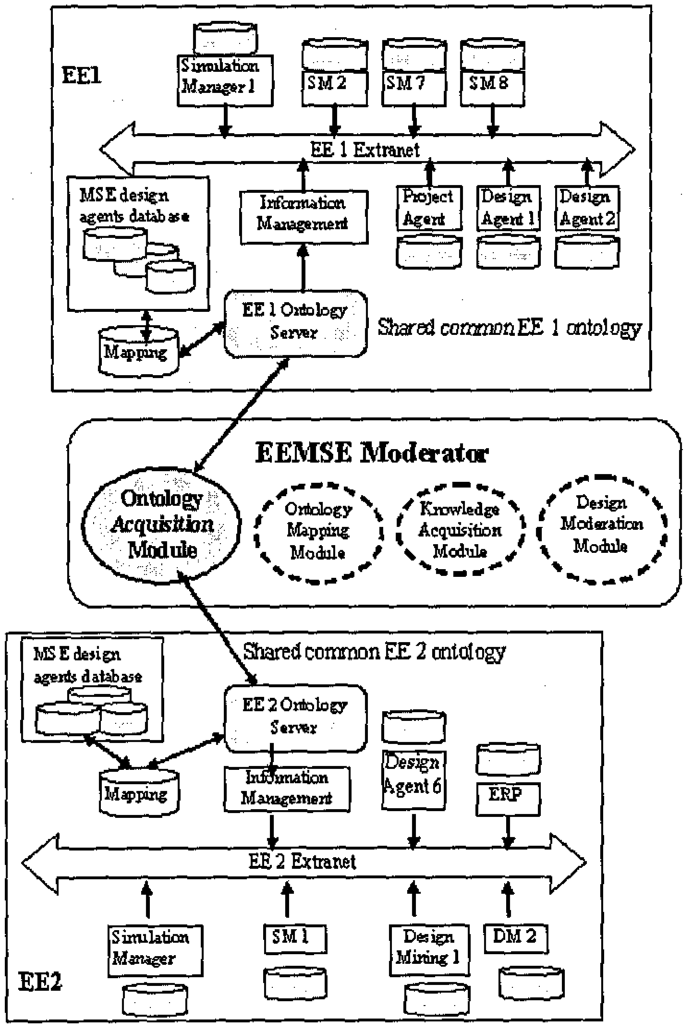


Figure 8.3: Ontology Acquisition Module of the EEMSE Moderator

8.2.2 Ontology Mapping Module (OMM)

Ontology mapping is the process by which two ontologies are semantically related at conceptual level with a portion of the source ontology to the target ontology's entities, transforming instances from the sources ontology into instances in the target ontology according to those semantic relations [Maedche et al. 2003; Noy and Musen 2004]. As shown in chapter 7, figure7.14 and 7.15 the mapping examples; all the individual ontologies must be mapped to the mediating ontology that specifies the shared semantics of the concepts that are to be used by the integration service. Two steps have been identified and embedded into the Ontology Mapping Module (OMM) (displayed in figure 8.4) of the EEMSEM: Normalization, Ontology Mapping Rules.

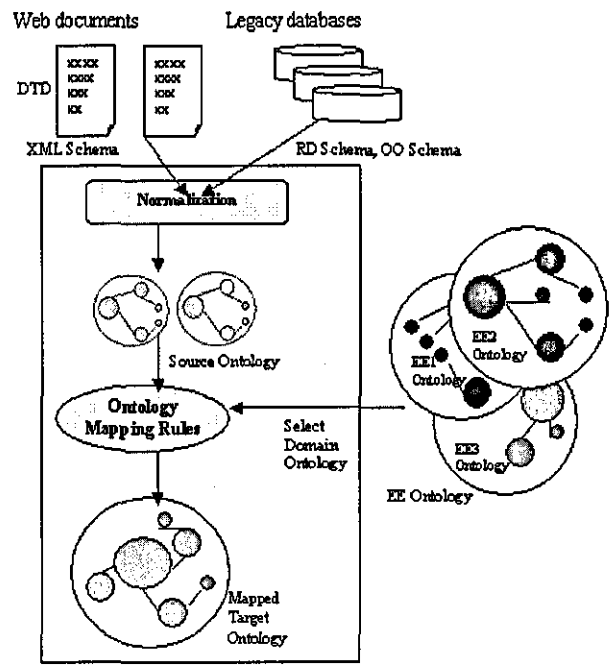


Figure 8.4: Architecture for Ontology Mapping Module

Normalization:

[Maedche et al. 2003] pointed out that normalization extends the ontology-mapping problem somewhat to the problem of integrating existing information sources that are not ontology based. For example, in most industries, there are large quantities of existing data already stored using relational database technology. Therefore, information presented in the documents needs to be transformed into a specific ontology format, for example, the transformation of free text, web documents, and legacy database into the ontology level is the first step for the OMM. Tools are currently available for mapping the RDB schemas onto RDF/ RDFS/OWL, such as Jena [McBride 2002] is a declarative language to describe mappings between relational database schemata and OWL ontologies.

Ontology Mapping Rules:

This step is to define and specify mapping rules between different ontologies and versions. These mapping rules define how to transform source-ontology instances into target-ontology instances. The mapping rules of the OMM have adopted several approaches from [Maedche et al. 2003; Noy and Musen 2004] in their managing multiple ontologies researches. The mappings according to:

- *Type of related entities.* Mapping rules can be established between concepts, attributes, and relations.
- *Cardinality.* Mapping rules can have 1:1, 1:n, or n:1 cardinality.
- *Condition.* Mapping rules can include conditions on the instances being transformed.
- *Transformation function.* A mapping rule can include a transformation function that, when applied to the source information in the source ontology, will produce the required information in the target ontology.

Within the EEMSE Moderator, all the individual MSE software's information must be normalized into the OWL primitives and be mapped to the mediating ontology, called the **Domain Ontology** which is selected from the EE Ontology of OAM. This is because OWL provides built-in ontology mapping support. For example, a particular class or property in one ontology is the same as a class or property in another ontology (owl:sameClassAs, owl:samePropertyAs).

8.2.3 Knowledge Acquisition Module (KAM) And Design Moderation Module (DMM)

Both the KAM and DMM in the EEMSE Moderator perform the same functionality as in the MISSION MSE Moderator, which was illustrated in chapter 3. The KAM is used to create, delete or amend knowledge about what is important to any individual EE / VE team members (these will be referred to as design agents here, to maintain consistency of terminology with the earlier MISSION MSE Moderator research). Therefore, it is important to modify this knowledge when new design agents join or if existing agents are changed significantly, resulting in changes to their associated knowledge which the EEMSEM uses to identify potential design conflicts. The knowledge structures repose in an object oriental knowledge rules database based on the knowledge representation model as in the MISSION MSE Moderator.

However, the KAM in the EEMSEM would be translated into the neutral format (EE Ontology) for dealing with any syntactic and semantic differences in the terminology that may be used by different project team members. This is achieved through the OAM and the OMM and then this knowledge about design agents can repose as mapped results in the Knowledge Rules Ontology Server, as shown in Figure 8.5. Additionally, it is recommended that the KAM in the EEMSEM should be a web browser interface, so that the design agents could add, delete or edit the knowledge rules about their interests 24 hours a day and 7 days a week around the world.

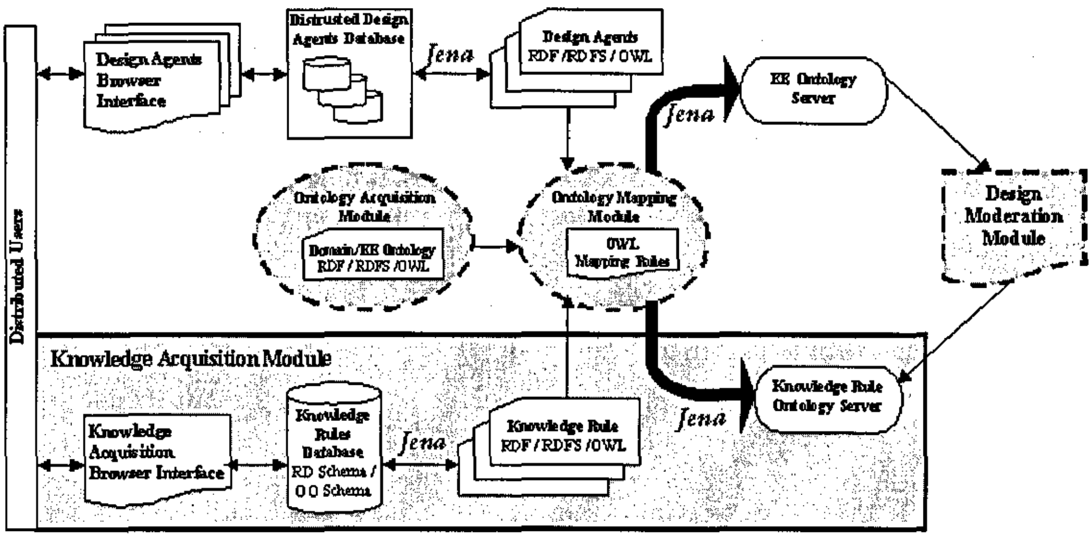


Figure 8.5: The Structure Of KAM In EEMSE Moderator

The DMM is used to assist and keep track of changes made to the MSE design documents and identify whether any current design agent may be interested in the change. The change details should therefore also go through the translation process into the neutral format as described above for the KAM and the mapped result of the change details will be reposed in the EE Ontology server, as shown in figure 8.5. Therefore the DMM should be activated whenever a change is made to any information that may be related to interests recorded in any design agent module.

These changes can then be passed through the translation process, through the OAM and the OMM and into the EE Ontology Server. If Information changes in the EE Ontology Server have been identified, the DMM will be notified of change and also connected to the Knowledge Rule Ontology Server which is needed for the moderation process of conflict detection.

8.3 EEMSE Moderator Case Examples

In order to illustrate the functionality of the EEMSEM, a manufacturing e-purchasing example study has been used. The purchasing cycle encompasses: raising a requisition ⇒ approving the requisition ⇒ producing the purchase order ⇒ approving the purchase order ⇒ issuing the purchase order ⇒ receiving the goods or services ⇒ returning goods or services (if goods or services don't match what was ordered) and ⇒paying the invoice. Requisitions are electronically generated, approved and passed through the purchasing system. Figure 8.6 outlines the requisition process for a production item.

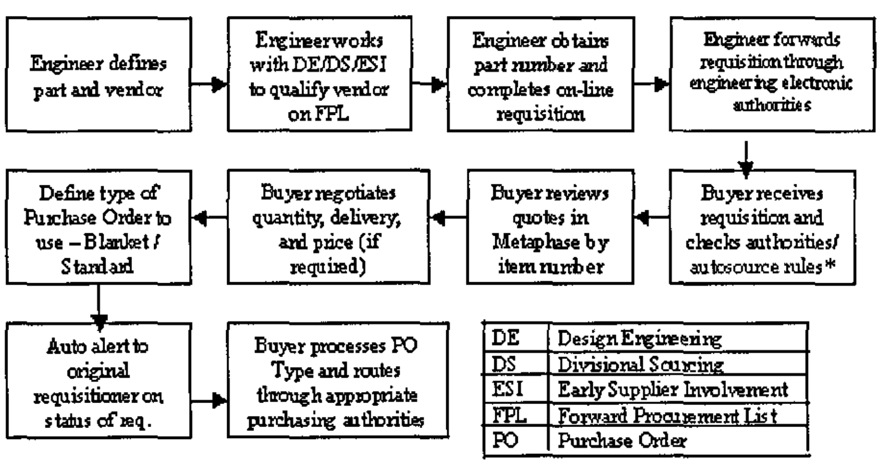


Figure 8.6: Requisition Process For A Production Item.

*Autosource Rules in the e-purchasing system allow predefined items, such as a list of approved vendors to be specified and source documents for these vendors to be associated with the current activity. This is the place that the line-item[9],vendor,

[9] Line-item is a term that describes the place that supports the purchase of an item, such as item number, item description, price, quantity ordered, etc.

schedule sharing splits[10] for consignment purchases[11], price and vendor part number (if applicable) come together.

This case example was used to demonstrate the conflict moderation work between the extended project teams' MSE agents (e.g. the Motorola's Enterprise Resource Planning (ERP) purchasing agent, the Unitech's Supply Chain Management (SCM) agents, and other MSE agents within this project. For example, Unitech' SCM is one of the participant systems in the EE project. As part of the extended project, Unitech's SCM determines that there should be a minimum quantities limitation of not less than 3000 units on their parts order. However, at some point during the operation of the extended project, there is a policy change in Motorola, for their ERP's purchase orders system that determines that the electronic signature approval levels are reset to permit a maximum quantity on each line-item of 2000 units.

The EEMSEM here must be able to identify when the ERP's purchase agent changes the approval levels for the electronic signature in the quantity attribute of the line-item object as this change may cause conflict, hence the moderator must communicate the detection of this possible conflict to all interested MSE agents. When the above information change is made, the EEMSEM should identify that the Unitech SCM is the design agent that will be affected and problems may occur with the quantity attribute of the part object. Therefore the EEMSEM should issue an appropriate warning message to the Unitech SCM (e.g. via e-mails).

This EE project example shows that each company has their own processes, databases, information and knowledge systems in place. Inevitably, each will also use their own languages and terminologies, which will have developed over a period of time through their working practices and experiences in particular industry sectors, the culture in their particular organization, and many other contributory factors. Each

[10] Schedule Sharing is a partnership program between Motorola and its suppliers aimed at improving both parties' operational performance by electronically sharing Motorola's customer forecast and reporting the inventory status.

[11] The Consignment purchases is a partnership between Motorola and selected suppliers to provide schedule share items on a just-in-time basis that is not paid for until it is disbursed to production. The intent is to reduce inventory levels in Plantation to less than one week, improve cycle time and provide the supplier with flexibility to control the material.

partner within the EE project will need to exchange and share some information and knowledge related to the project they are working on together, but this is inherently complex because they do not automatically work with a common language or common information models or structures. Figure 8.7 shows an example of two different identifiers existing with different models or databases, but having the same meaning. Hence, Motorola's ERP identifier, line-item number and Unitech's SCM variable, part number both have equivalent meanings in the purchasing process. They are therefore both mapped to component in the agreed EEMTP700 ontology.

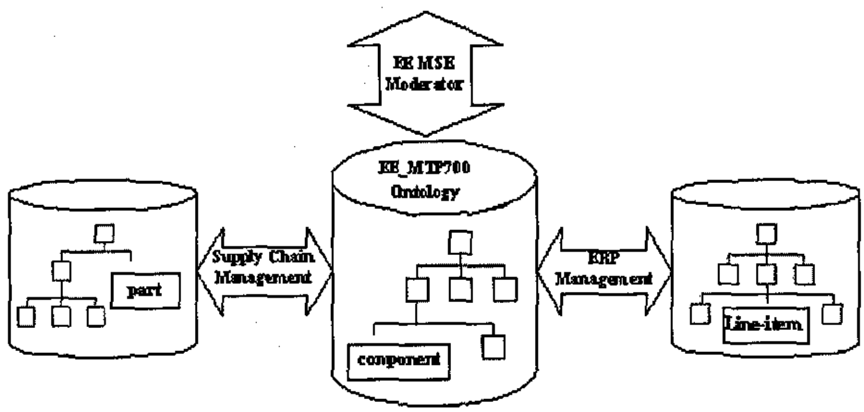


Figure 8.7: Ontology mapping into the common ontology model

The role of the OAM and the OMM have therefore been proposed and experimental implementations undertaken to make the concept of knowledge and information integration possible, by providing an interoperability mechanism for dealing with the above differences. It is assumed that the EE_MTP700 ontology derived from the proposed MSE Ontology Model has been chosen as the domain ontology (common ontology) for the MTP 700_090504 project. The ontology mapping to EE_MTP700 was undertaken using OWL primitive mapping as shows in table 1.

scm:part → ee_mtp700:component←erp:line-item	scm:quantity →ee_mtp700:quantity←erp:quantity
<owl:Class rdf:ID="scm.Part"> <owl:sameAs ref.resource ="#ee_mtp700_Component"/> </owl:Class> <owl:Class rdf:ID="erp.Lineitem"> <owl:sameAs ref.resource ="#ee_mtp700.Component"/> </owl:Class>	<owl:DatatypeProperty rdf:ID="scm.quantity"> <rdfs:domain rdf:resource=#scm_Part"> <owl:sameAs rdf:resource="#ee_mtp700:quantity"/> </owl:DatatypeProperty> <owl:DatatypeProperty rdf:ID="erp.quantity"> <rdfs:domain rdf:resource=#erp.Lineitem"> <owl:sameAs rdf:resource="#ee_mtp700:quantity"/> </owl:DatatypeProperty>

Table1: Owl:sameAs axioms for semantically mapping into EE_MTP 700

The built-in owl:sameAs statement links an individual to an individual that actually refer to the same thing: the individuals therefore have the same "identity". The owl:sameAs axioms are often used in defining mappings between ontologies. In this case, the concepts from scm:part have the same meaning as the concepts from ee_mtp700:component. Moreover, the concepts from erp:line-item also have the same meaning as the concepts from ee-mtp700:component. The axioms should ensure that when someone queries the SCM for the instances of the Part, the result includes all instances of the component from the ee_mtp700. Also, the instances of the Line-item will have the identity instances of the component from the ee_mtp700.

As explained, there are similarities between the proposed EEMSEM KAM and the KAMs introduced in earlier moderators, since the EEMSEM must be able to acquire knowledge about individual design team members, the knowledge about what changes are important to them, and what actions should be taken if such changes occur. This prototype version of the MISSION MSE Moderator [Harding et al. 2003] has been reused for the experimental implementation of the KAM and DMM elements of the EEMSEM, since the functionality of these elements of the EEMSEM is very similar to the functionality of the MISSION MSE Moderator. Figure 8.8 shows the interface from the KAM, which can be used whenever new agents join a project or existing agents are changed in any way.

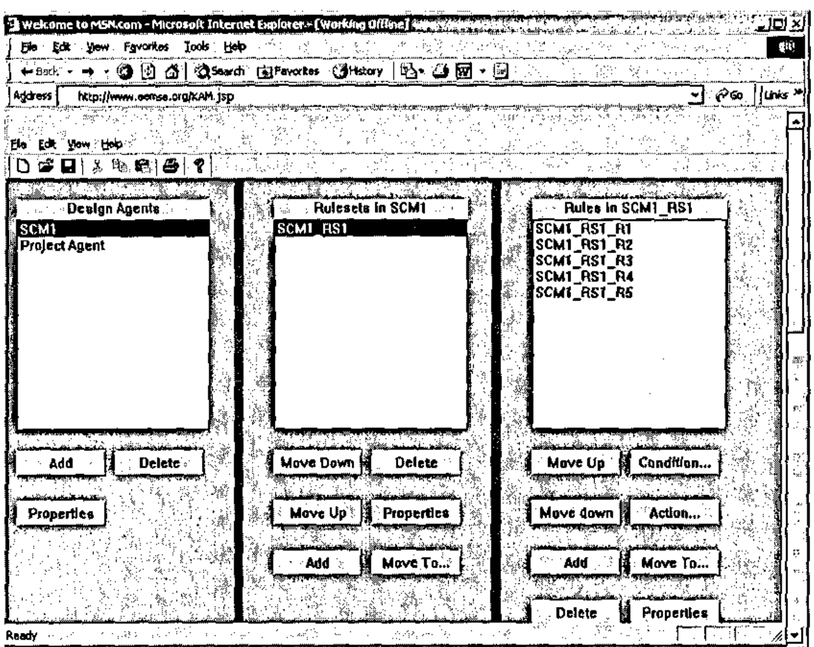


Figure 8.8: Knowledge Acquisition Module for Unitech's SCM Agent

In this case study example, Unitech have put their minimum order quantities constraint into the KAM by creating a new Design Agent Module called (SCM1). The SCM1 design agent module knows how to process its own knowledge, as this behaviour is implemented in methods of the various classes, including Ruleset, Rule, Condition and Action objects (see fig 3.6). As described in chapter 3, the KRM class structure has been used to implement the KAM. The knowledge base for SCM1 is captured in the database as a list of Ruleset objects. Each Ruleset object can be associated with any number of Rule objects. Figure 8.8 shows the SCM1_RS1 Rulesets includes several rules (SCM1_RS1_R1, SCM1_RS1_R2, SCM1_RS1_R3, SCM1_RS1_R4, SCM1_RS1_ R5) which are part of the SCM1 design agent module, and which embody the SCM1's interests in details of the minimum order quantities constraint.

Each Rule is associated with a Condition object and a Resulting Action Object, see fig 3.6, and the method of populating these into a design agent module is illustrated in figure 8.9. Condition and Resulting Action objects can be either simple or compound. A Compound Condition contains a Simple Condition and a Condition, which are connected using either AND or OR, and any Condition can be negated if required. Compound Resulting Actions contain a Simple Resulting Action and a Resulting Action and these are connected using AND.

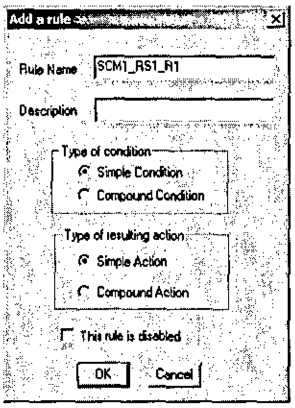


Figure 8.9: Add A Rule

The real processing power of each rule comes through the sub-classes of the Simple Condition and the sub-classes of the Simple Resulting Action. Figure 8.10 shows several examples of different sub-classes of Simple Condition, in the figure; these classes all have the prefix SC (Simple Condition).

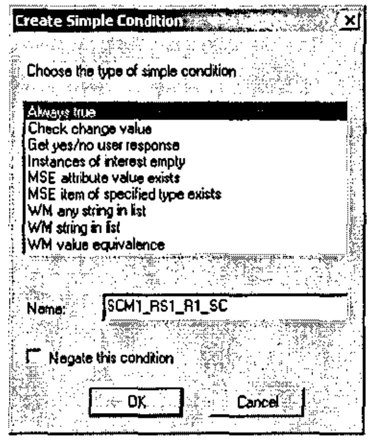


Figure 8.10: Choose The Type Of Simple Condition

Similarly, Resulting Actions are created by combining instances of the sub-classes of Simple Resulting Action (shown with prefix SR). Figure 8.11 shows the several type of Simple Resulting Action. Further explanations of these objects can be found in the reports of the earlier implementations of the KRM (Harding, 1996, and Harding et al, 2003) that are being exploited here.

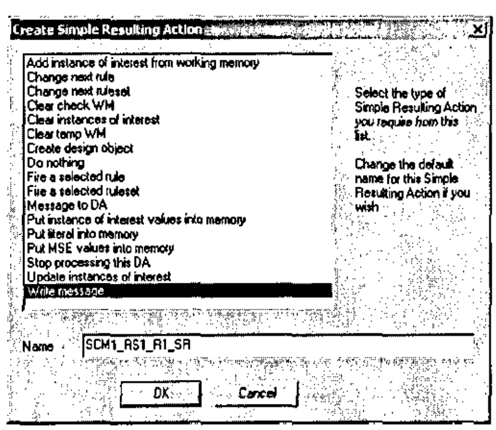


Figure 8.11: Create And Select The Type Of Simple Resulting Action

In the current case study, the SCM1_RS1 Rulesets includes five rules, SCM1_RS1_R1, SCM1_RS1_R2, SCM1_RS1_R3, SCM1_RS1_R4, SCM1_RS1_R5, which have been populated into the SCM1 design agent module. The details of each rule with its associated type of the Simple Condition and the type of the Simple

Resulting Action are illustrated in Table 8.1. and saved into the object knowledge database base (ObjectStore):HKL_demo.db.

Add a rule	Create Condition and select type of condition	Detail of the condition	Create Resulting Action and select type of action	Detail of the action
SCM1_RS1_ R1	Always true	-----	Write message	This message means the first rule has been activated
SCM1_RS1_ R2	Check change value	Type of value: Class of changed object Value: SCM_Part	Write message	A SCM_Part Object has been changed
SCM1_RS1_ R3	Check change value	Type of value: Attribute change Value: quantity	Put MSE value into memory	Attribute: quantity Type of value: *int* Working memory: in Temp
SCM1_RS1_ R4	Always true	-----	Put Literal into memory	Value: 3000 Type of value: int Working memory: in Check
SCM1_RS1_ R5	WM value equivalence	Type of equivalence: Check > Temp	Write message	*Minimums* quantity 3000 has been changed by other MSE agent

Table 8.1: The Details Of Each Rule In The SCM1_RS1 Rulesets

So, assuming that information is changed at Motorola, resulting in the electronic signature approval levels being reset to 2000 units. The EEMSEM here must be able to identify when Motorola's ERP's purchase agent changes the approval levels for the electronic signature in the quantity attribute of the line-item object. The EEMSEM will then pass and translate details of this change through the OAM and the OMM of the EEMSEM to eventually recognize that the information change is the quantity attribute of the component object in the EE_MTP700 Ontology Server.

When the information change in the EE_MTP700 Ontology Server have been identified, the EEMSEM will then execute its DMM and also connect to the Knowledge Rule Ontology Server which repose the mapped results from OAM and OMM of knowledge rules in KAM early, and then process this change information to determine which, (if any) of the participants in the extended project are interested in the change to the quantity attribute of the component object change. If the EEMSEM identifies any interested participants, it should then (still using the DMM) communicate the detection of the possible conflict to all the interested participants in the extended project (referred to here as MSE agents).

Figure 8.12 shows the operation of the DMM on this case study example; initially (on the top 2 lines) finding that Unitech's supply chain management agent (SCM1) is the one that will be interested. As the SCM1 stores the constraint on the quantity attribute of the part object which matches the quantity attribute of the component object through OAM and OMM in the Knowledge Rule Ontology Server. Therefore, the DMM identifies that the SCM1 is the design agent which will be affected if the information change in the EE_MTP700 Ontology Server, and then the DMM processes the knowledge in the SCM1 design agent module. Finally a warning message, minimums quantity 3000 has been changed by other MSE agent, should be sent to the Unitech (e.g. via e-mails).

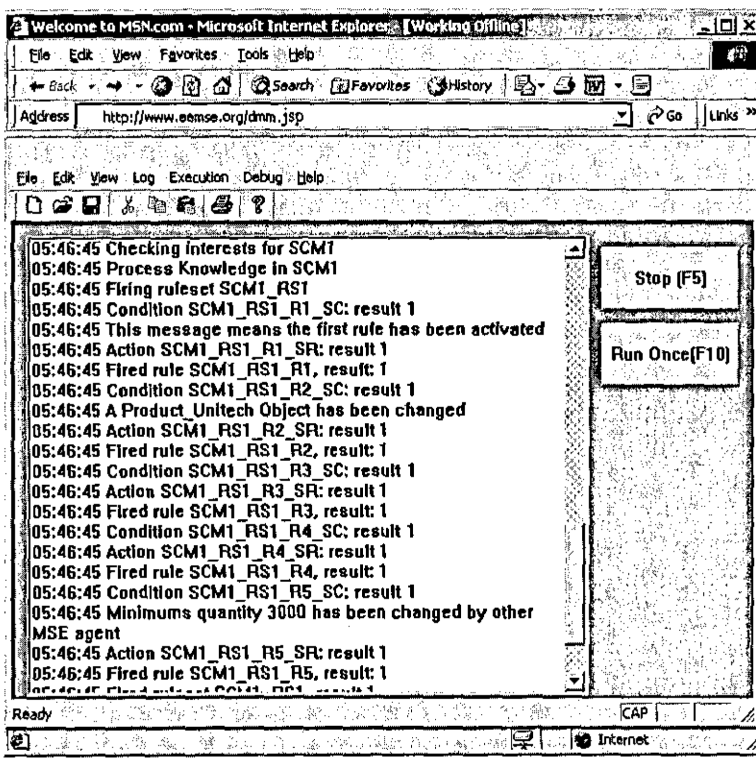


Figure 8.12: The Design Moderation Module in the EEMSEM

Chapter 9 : Conclusions

This chapter concludes the manufacturing system engineering ontology model for global extended project team research, which has been discussed in the previous chapters. The first section provides a summary of the thesis, following by the recommendations for future development. Finally, the conclusions are drawn against the research objectives set in Chapter 2.

9.1 Summary Of The Thesis

A common ontology methodology for MSE model has been proposed and its application demonstrated through the design of an EEMSE moderator to enhance coordination and information integration within a multi disciplinary inter-enterprises environment. The model is created based on an ontology approach and semantic web techniques for knowledge modelling and reasoning. The ontology model is implemented by semantic web languages (RDF, RDFS, and OWL), using ontology editor tool: Protégé-2000 and its plugins. The common ontology-based model acts as a mediating ontology and therefore provides the integration service for different MSE information models.

The concepts and implementations of previous moderators have been discussed and the challenges of extending this technology for use in EE / VE environments have been illustrated. This has led to the proposed framework for an EEMSE moderator which includes an ontology acquisition module, ontology mapping module, knowledge acquisition module, and design moderation module. The purpose of the ontology acquisition module is to acquire a common ontology based on the MSE ontology model and created by a particular EE / VE group and called the EE ontology. The EE ontology serves as a mediating ontology in conjunction with the ontology mapping module to provide translation services where a particular MSE team member needs a translation from his or her terminology into the EE ontology. The knowledge acquisition module is used to collect, delete or amend knowledge rules which will then be used for identifying potential design conflicts, when they are processed by the design moderation module. This proposed structure for the EEMSEM is novel and should provide a powerful tool for inter-enterprise team working by providing semantic and syntactic information integration as discussed in this thesis.

The feasibility of the MSE ontology model for use with an EEMSE moderator has been tested through prototype implementations and demonstration through an extended project case study provided by Motorola Technology Malaysia and its supply chains for two-way radio design and manufacturing. The case studies reported in this thesis also have demonstrated the conflict moderation work of the EEMSE moderator through an e-purchasing case example between the EE / VE partners' software agents.

9.2 Recommendations For Future Work

The MSE ontology model to enable the operation of the EEMSE Moderator framework is proposed as a possible way to integrate different information models via an ontology-mediated translation service. However, there are issues that need further improvements. Future development is recommended to improve the current method and identified requirements include:

1. Semi automated features for ontology mapping. Currently, a manual mapping process has been used and is based on the concept of declarative mapping relations, which are explicit specifications for the syntactic and semantic connections between individual enterprise terminology and their common ontology. The possibility of partially automated mappings is recommended for future development.

2. The MSE ontology model may involve simple logical reasoning for semantic and syntax mapping. However, the EEMSE Moderator is limited to a knowledge-based approach for the extraction of useful information based on the established object oriented knowledge database. It does not involve the discovery of new knowledge, such as ontology learning, automatic knowledge creation and automatic knowledge retrieval by logical axioms. Future development is recommended, such as Ontology-based information extraction towards the automatic knowledge creation and knowledge retrieval by rules, logic and proof, to improve and extend the application of the EEMSE moderator.

3. Continuous search for supporting software tools or technologies to enhance the system implementation. The research is ongoing and will continue to improve and keep in line with current semantic web technology. Future implementation should therefore support more powerful inference engine.

9.3 *Conclusion*

The research has shown the potential of the proposed MSE Ontology Model and the general architecture for the EEMSE Moderator to meet the objectives of enterprise integration for global extended project teams working. This collaborative system architecture focuses on how to support information autonomy that allows individual enterprises to keep their own preferred terminology or languages rather than requiring them all to adopt a single standardized vocabulary. Different engineering information terminologies are interpreted and automatically connected to the corresponding terminologies through mapping into the mediated ontology model.

The improvement in information semantic interoperability in IT will contribute to global competitive advantage in developing consistent inter-enterprises cooperation and enhancing supply chains globally. Particularly, the use of a standard and low cost Internet platform and semantic web technologies in this research should make the approach potentially viable for small and medium size enterprises (SMEs) and therefore help to break down the barriers of supply chain extension in IT.

Reference

AIFB "Institute for Applied Informatics and Formal Description Methods, University of Karlsruhe". http://www.aifb.uni-karlsruhe.de/english on 23/07/2004

Arpinar, I. B., B. Aleman-Meza, Ruoyan Zhang and A. Maduko (2004). "Ontology-driven web services composition platform". IEEE International Conference on e-Commerce Technology, 6-9 July, pp:146 - 152.

Bazjanac, V. (1998). "Industry Foundation Classes: Bringing software interoperability to the building industry." The Construction Specifier, Volume:6, pp:47-54.

Berners-Lee, T. (1998). "Semantic Web Road Map". available on-line as http://www.w3.org/DesignIssues/Semantic.html

Berners-Lee, T. (2000). "Semantic Web on XML". Slides available on-line as *http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html*

Berson, A. (1992). Client/Server Architecture (McGraw-Hill Series on Computer Communications), McGraw-Hill Education, ISBN:0070050767.

Booch, G. and I. Graham (1993). Object Oriented Methods, Addison Wesley, ISBN:0201593718.

Bravoco, R. R. and S. B. Yadav (1985). "A Methodology to Model the Functional Structure of an Organization." Computers In Industry, Volume:6, pp 345-361.

Broekstra, J., M. Klein, S. Decker, D. Fensel and I. Horrocks (2001). "Adding formal semantics to the Web building on top of RDF Schema". In Proceedings of the 10th International Web Conference (WWW-10), Hong Kong, China, 1-5 May

Camarinha-Matos, L. M., H. Afsarmanesh, C. Garita and C. P. Lima (1997). "Towards an Architecture for Virtual Enterprises". 2nd World Congress on Intelligent Manufacturing Processes & Systems, Budapest, June

Camarinha-Matos, L. M., H. Afsarmanesh and R. J. Rabelo, (editor) (2000). E-business and Virtual Enterprises: Managing Business-to-business Cooperation, Kluwer Academic Publishers, ISBN:0792372050.

Camarinha-Matos, L. M., C. P. Lima and A. L. Osorio (1997). "The PRODNET Platform for Production Planning and Management in Virtual Enterprises". Proceedings of International Conference On Concurrent Enterprising (ICE'97), Nottingham, UK., PRODNET-II Project (ESPRIT 22647), Oct

Chen, P. (1976). "The entity-relationship model-toward a unified view of data." ACM Transactions on Database Systems, Volume:1(1), pp 9-36.

Code, E. F. (1970). "A Relational Model of Data for Large Shared Data Banks." Communications of the ACM, Volume:13(6), pp 377-387.

Cutting-Decelle, A. F., R. I. Young, C. J. Anumba, A. N. Baldwin and N. M. Bouchlaghem (2003). "The Application of PSL to Product Design across Construction and Manufacturing." Concurrent Engineering: Research and Applications (CERA), Volume:11(1), pp 65-75.

cXML "Commerce XML Resources". http://www.cxml.org/ on 23/07/2004

DARPA "US Defense Advanced Research Projects Agency". http://www.darpa.mil/ on 23/07/2004

Debras, P., D. Hagemann, M. Joosten, Z. Pocsai, L. Seifert and D. Stumpf (1998). "The Common Semantic Model in GENIAL: From Conceptualisation towards Engineering Networks". Conference on Integration in Manufacturing (IiM 98), Goteburg, 6-8 Oct

Debras, P., A. Zarli, V. Amar and P. Poyet (1998). The Distributed Information Service in the VEGA project. Deliverable D401.

Dwyer, J. (2004). "The Weakest Link." IEE Review, Special Report Manufacturing.

e-COGNOS Consortium (2002). e-COGNOS Evaluation Results.

eConstruct (2000-2002). "eCommerce and eBusiness in the European Building and Construction Industry:Exploiting the Next Generation Internet", http://www.econstruct.org/default_frame.htm on 19/07/2004

Ellis, T. I. A., A. Molina, R. I. M. Young and R. Bell (1994). "The Development of an Information Sharing Platform for Concurrent Engineering". Proceedings of the International Manufacturing Systems Engineering Workshop, Grenoble, France

Elmasri, R. and S. B. Navathe (1989). Fundamentals of Database System, The Benjamin / Cummings Publishing Company, Inc, ISBN:0805301453.

Fensel, D. (2003). Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce, Springer Verlag, ISBN:3540009663.

Fensel, D., (Editor). (2002). Intelligent Information Integration in B2B Electronic Commerce, Kluwer Academic Publishers, ISBN:1402071906.

Fensel, D., Ying Ding, B. Omelayenko, E. Schulten, G. Botquin, M. Brown and A. Flett (2001). "Product data integration in B2B e-commerce." IEEE Intelligent Systems, Volume:16 , Issue: 4, pp:54 - 59.

Fox, M. S. and M. Gruninger (1997). "On Ontologies and Enterprise Modelling". International Conference on Enterprise Integration and Modelling Technology, Torino, Italy, Springer-Verlag, October 28-30

Gausemeier, J., L. Seifert, M. Joosten, P. Kesteloot and J.-L. Monceyron (1997). "GENIAL: The Logical Framework for the Systematisation of Internal and External Engineering Knowledge". International Conference on Engineering Design (ICED 97), Tampere, Finland, 19-21 August

GLOBEMEN (2000-2003). Global Engineering and Manufacturing in Enterprise Networks, Esprit IMS 99004, IST-1999-60002.

Goldfarb, C., Prescod, P. (2001). The XML Handbook (Third Edition), ISBN:0-13-055068-X.

Goranson, H. T. (1999). The Agile Virtual Enterprise: Cases, Metrics, Tools, ISBN:1567202640, pp 109-156.

Grabowski, H., S. Rude and Z. Pocsai (1997). "Ontology Technology: The Key for Intelligent Migration and Retrieval of Inforamtion in Engineering Networks". Internation Conference on Engineering Design (ICED 97), Tampere, Finland, 19-21,August

Gruber, T. R. (1993). "A Translation Approach to Portable Ontology Specifications." Knowledge Acquisition, Elsevier Science, Volume:5(2), pp 199-220.

Gruber, T. R. (1993a). "Toward Principles for the Design of Ontologies Used for Knowledge Sharing". International Workshop on Formal Ontology, Padova, Italy

Gruninger, M. and M. S. Fox (1995). "Methodology for The Design and Evaluation of Ontologies". Proceedings of the Whorkshop on Basic Ontological Issues in Knowledge Sharing, IJCAI-95, Montreal

Harding, J. A. (1996). "A Knowledge Representation Model to Support Concurrent Engineering Team Working". PhD Thesis

Harding, J. A. and K. Popplewell (1996). "Driving Concurrency in a Distributed Concurrent Engineering Project Team: A Specification for an Engineering Moderator." International Journal of Production Research, Volume:34(3), pp 841-861.

Harding, J. A., K. Popplewell and D. Cook (2003). "A Manufacturing System Engineering Moderator: An Aid for Multi-discipline Project Teams." International Journal of Production Research, Volume:41(9), 9, pp 1973-1986.

Harding, J. A. and B. Yu (1999). "Information-centred Enterprise Design Supported by a Factory Data Model and Data Warehousing." Computers In Industry, Volume:40, pp 23-36.

Harding, J. A., B. Yu and K. Popplewell (1999). "Information modelling: an integration of views of a manufacturing enterprise." International Journal of Production Research, Volume:37(12).

Hendler, J. (2001). "Agents and the Semantic Web." IEEE Intelligent Systems, Volume:16 , Issue: 2, Pages:30 - 37.

Hitomi, K. (1996). Manufacturing Systems Engineering, Taylor and Francis Ltd, London, ISBN:0748403248.

HL7 "Health Level Seven". http://www.hl7.org/ on 23/07/2004

Hyvönen, E., S. Saarela and K. Viljanen (2004). "Application of Ontology Techniques to View-Based Semantic Search and Browsing". Proceedings of the First European Semantic Web Symposium, Heraklion, Greece, 10-12 May

ISO 10303-1 (1994). "Industrial Automation and Integration - Product Data Representation and Exchange - Part1: Overview and Fundamental Principles."

ISO/CD18629 (2002). Industrial automation system and integration - Process specification language: Part 1: Overview and basic principles, ISO TC184/SC4/JWG8: International Organization for Standardization (ISO).

Jackson, P. (1990). Introduction to Expert Systems, Addison-Wesley, ISBN:0-201-17578-9.

Jepsen, T. (2001). "SOAP cleans up interoperability problems on the Web." IT Professional, Volume:3, Issue:1.

Jordan, J. and F. Michel (2000). Next Generation Manufacturing Methods and Techniques, John Wiley & Sons Inc, ISBN:0471360066.

Kazi, A. S. and C. Charoenngam (2003). "Facilitating Inter-enterprise Information Exchange in One-of-a-kind Settings." ITcon, Volume:8.

Kazi, A. S. and M. Hannus (2000). Functional Requirements for Inter-enterprise Intranet Services. Global Engineering, Manufacturing and Enterprise Networks. Nemes L.(Editors), Kluwer Academic Publishers: Pages: 54-60.

Kazi, A. S., M. Hannus, J. Laitinen and O. Nummelin (2001). "Distributed Engineering In Construction: Findings from the IMS GLOBEMEN Project." Journal of Information Technology in Construction, Volume:6.

KIF (1999). Knowledge Interchange Format, Part 1:KIF-Core. ISO/JTC1/SC32/WG2, WD.

King, R. E. and K. Moon (1999). "Quick response replenishment: a case study". Winter Simulation Conference Proceedings, Pages:1341-1349.

Knaus, R. and C. Jay (1990). "Transporting Knowlledge Bases: A Standard." AI Expert, November.

Kosanke, K. and J. de Meer (2001). "Consistent terminology-a problem in standardization / State of art report of enterprise engineering". 2nd IEEE Conference on Standardization and Innovation in Information Technology, Boulder, CO, USA, 3-6 Oct, pp 56-61.

Kosanke, K., F. Vernadat and M. Zelm (1999). "CIMOSA: enterprise engineering and integration." Computers In Industry, Volume:40, pp 83-97.

Lassila, O., F. van Harmelen, I. Horrocks, J. Hendler and D. L. McGuinness (2000). "The Semantic Web and its Languages." Intelligent Systems, IEEE, Volume:15, 6, pp 67-73.

Lee, J. L., S. E. Madnick and M. D. Siegel (1996). "Conceptualizing semantic interoperability: A perspective from the knowledge level." International Journal on Cooperative Information Systems (IJCIS), Volume:4(4).

Li, Y. and N. Zhong (2003). "Ontology-based Web mining model: representations of user profiles". Proceedings IEEE International Conference on Web Intelligence (WI 2003), 13-17 Oct, Pages: 96 - 103.

Lima, C., B. Fies, A. Zarli, M. Bourdeau, M. Wetherill and Y. Rezgui (2002). "Towards an IFC-enabled ontology for the Building and Construction Industry: the e-COGNOS approach". eSMART 2002 conference, Salford, UK, 18-21 November

Lin, H. K. and J. A. Harding (2003). "An Ontology Driven Manufacturing System Engineering Moderator for Global Virtual Enterprise Teams". Advances in Manufacturing Technology XVII, Professional Engineering Publishing Ltd, UK,, pp 365-370.

Lin, H. K., J. A. Harding and M. Shahbaz (2004). "Manufacturing System Engineering Ontology for Semantic Interoperability Across Extended Project Teams." International Journal of Production Research, Vol 42, No. 24, pp 5099-5118, ISBN 0020-7543.

Maedche, A., B. Motik, L. Stojanovic, R. Studer and R. Volz (2003). "Ontologies for Enterprise Knowledge Management." IEEE Intelligent System.

Maedche, A. D. (2002). Ontology Learning for the Semantic Web, Kluwer Academic Publishers, ISBN:0-7923-7656-0.

McBride, B. (2002). "Jena: A Semantic Web Toolkit." Internet Computing, IEEE, Volume:6, 6, pp 55-59.

McGuinness, D. L. (2002). "Ontologies Come of Age". In Dieter Fensel, Jim Hendler, Henry Lieberman, and Wolfgang Wahlster, editors. Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential, MIT Press

McGuinness, D. L., R. Fikes, J. Hendler and L. A. Stein (2002). "DAML+OIL: an ontology language for the Semantic Web." Intelligent Systems, IEEE, Volume:17(5), pp 72-80.

McGuinness, D. L. and F. Van Harmelen (2003). "OWL Web Ontology Language Overview", W3C Working Draft 31 March 2003, http://www.w3.org/TR/owl-features.

McIlraith, S. A. and D. L. Martin (2003). "Bringing semantics to Web services." IEEE Intelligent Systems, Volume:18 , Issue: 1, Pages:90 - 93.

McLean, C., F. Riddick and S. Leong (2000). "Architecture for Modelling and Simulation of Global Distributed Enterpries". 9th ASIM Dedicated Conference, Berlin, The New Simulation in Production and Logistics, Prospects, Views, and Attitudes, IPK Berlin, Eigenverlag, pp375-384.

Mena, E. and A. Illarramendi (2001). Ontology-Based Query Processing for Global Information Systems, Kluwer Academic Publishers, ISBN:0792373758.

MISSION (1998-2001). "Modelling and Simulation Environments for Design, Planning and Operation of Globally Distributed Enterprises", (ESPRIT IMS 29656), http://www.ims-mission.de/ on 16/07/2004.

Molina, A. (1995). "A Manufacturing Model to Support Data-Driven Applications for Design and Manufacture". PhD Thesis, Dept of Manufacturing Engineering, Loughborough University, UK.

Molina, A. and R. Bell (1999). "A manufacturing model representation of a flexible manufacturing facility". Proceedings of the Institution of Mechanical Engineers, Part B, 213, pp 225-246.

Molina, A., T. I. Ellis, R. I. Young and R. Bell (1994). "Modelling Manufacturing Resources, Processes and Strategies to Support Concurrent Engineering". Proceedings of CE94, Conference on Concurrent Engineering: Research and Applications, Pittsburgh, PA, USA, pp 51-60.

MOSES (1992-1995). "Model Oriented Simultaneous Engineering Systems." http://leva.leeds.ac.uk/www_moses/moses.html on 16/07/2004.

Motorola Inc (2002). "MTP700 Portable Radio Basic Training Manual". Chapter 6 Maintenance, Personal Communication

NATO-CALS "Commerce at Light Speed)". http://www.dcnicn.com/ncmb/ on 15/07/2004

NGM (1995-1996). "Next Generation Manufacturing". http://www.nsf.gov/od/lpa/news/publicat/nsf0050/manufacturing/nextgen.htm on 15/07/2004,

NIIIP (1998). "National Industrial Information Infrastructure Protocols". http://www.niiip.org/public/home.nsf/fProjectHome?openform&PID=NIIIP_TRP on 19/08/2004

Noy, N. F. and M. A. Musen (2004). "Ontology versioning in an ontology management framework." IEEE Intelligent Systems, Volume:19 , Issue: 4, Pages:6 - 13.

Noy, N. F., M. Sintek, S. Decker, M. Crubezy, R. W. Fergerson and M. A. Musen (2001). "Creating Semantic Web Contents With Protege-2000." IEEE intelligent SystemS, Volume:March / April.

OFX "Open Financial Exchange". http://www.ofx.net/ofx/default.asp on 23/07/2004

Ogden, C. and I. Richards (1923). The Meaning of Meaning: A study of the Influence of Language upon Thought and of the Science of Symbolism, Routledge & Kegan Paul Ltd, London, 10 Edition.

Omelayenko, B. (2002). "Ontology-Mediated Business Integration". Proceedings of the 13th EKAW 2002 Conference, Siguenza, Spain, © Springer-Verlag, October 1-4, Pages:264-269.

On-To-Knowledge (2000-2002). "Content-driven Knowledge-Management Tools through Evolving Ontologeis". http://www.ontoknowledge.org on 21/07/2004

OntoWeb (2001 -2004). "Ontology-based information exchange for knowledge management and electronic commerce". http://ontoweb.aifb.uni-karlsruhe.de

Ozsu, M. T. and P. Valduriez (1999). Principles of Distributed Database Systems, Prentice Hall.

Peter F. Patel-Schneider and B. Swartout. (1993). "Description-Logic Knowledge Representation System Specification." from the KRSS Group of the DARPA Knowledge Sharing Effort. Technical Report.

Petrie, C. and C. Bussler (2003). "Service agents and virtial enterprises: A survey." IEEE Internet Computing, Volume:7 , Issue: 4, Pages:68 - 78.

Popplewell, K. and J. A. Harding (2001). "A Manufacturing Systems Engineering Moderator: Identifying Conflicts in The Design of Distributed Manufacturing Systems". The 4th International EUROSIM: Shaping Future with Simulation., The Netherlands

Popplewell, K., J. A. Harding and M. Rabe, (editors) (2001). MISSION Project Final Report.

Pouchard, L. C. and A. F. Cutting-Decelle (2003). "Benefits of AI Techniques for Process Specification: Application to the PSL Language". Open Technical Forum, International Standards Organization TC184-SC4 Meeting, Poitiers, France, November

Powers, S. (2003). Practical RDF, O'Reilly & Associates, Inc, ISBN:0596002637.

Priebe, T. and G. Pernul (2003). "Ontology-based Integration of OLAP and Information Retrieval". Proceedings14th International Workshop on Database and Expert Systems Applications, 1-5 Sept, Pages: 610-614.

PRODNET-II (1996-1999). "Production Planning and Management in an Extended Enterprise". http://www.uninova.pt/~prodnet/ on 19/08/2004, ESPRIT 22.647

Radeke, E., (Editor) (1999). Final Report of GENIAL: Global Engineering Networking Intelligent Access Libraries, EP 22.284.

Ronald J. Brachman, Deborah L. McGuinness, Peter F. Patel-Schneider, Lori Alperin Resnick and A. Borgida (1991). Principles of Semantic Networks: Explorations in the Representation of Knowledge. Living with CLASSIC: When and How to Use a KL-ONE-Like Language, Morgan Kaufmann, ISBN:1558600884, pages 401-456.

RosettaNet. http://www.rosettanet.org/RosettaNet/Rooms/DisplayPages/LayoutInitial on 23/07/2004

Staab, S., W. van der Aalst, V. R. Benjamins, A. Sheth, J. A. Miller, C. Bussler, A. Maedche, D. Fensel and D. Gannon (2003). "Web services: been there, done that?" IEEE Intelligent Systems, Volume:18 , Issue: 1, Pages:72 - 85.

Stephens, J. (1999). VEGA - Virtual Enterprise Using Groupware Tools and Distributed Architectures. ESPRIT 20408.

Stouffs, R. and R. Krishnamurti (2001). "Standardization: A Critical View". International Conference IT in Construction in Africa, Mpumalanga, South Africa, 30 May – 1 June

Turk, Z. (1998). "Limits of Information Technology in Engineering (Or Why Computers Might Not Replace the Engineers)". Proceedings of the 4th Symposium on Intelligent Systems, Zagreb, Croatia

USA National Research Council and Committee on Visionary Manufacturing Challenges (1998). Visionary Manufacturing Challenges For 2020, National Academy Press, Washington, D.C., ISBN:0-309-06182-2.

Uschold, M., M. King, S. Moralee and Y. Zorgios (1998). "The Enterprise Ontology." The Knowledge Engineering Review, Volume:13(01), pp 31-89.

Varlamis, I., M. Vazirgiannis, M. Halkidi and B. Nguyen (2004). "THESUS, a closer view on Web content management enhanced with link semantics." IEEE Transactions on Knowledge and Data Engineering, Volume:16(6), ISSN: 1041-4347, pp. 685 - 700.

VEGA (1996-1999). "Virtual Enterprises Using Groupware Tools And Distributed Architecture". http://cic.cstb.fr/ILC/ECPROJEC/VEGA/HOME.HTM on 19/08/2004

W3C Semantic Web. http://www.w3.org/2001/sw/ on23/07/2004

Wetherill, M., Y. Rezgui, C. Lima and A. Zarli (2002). "Knowledge Management for the construction industry: the e-COGNOS project." ITCON Journal, special issue on KM.

WIDE (2002-2005). "Semantic Web-based Information Management and Knowledge Sharing for Innovative Product Design and Engineering." http://www.ist-wide.info/ on 18/07/2004.

xCBL "XML Common Business Library". http://www.xcbl.org/ on 23/07/2004

Young, R. I. M., O. Canciglieri-Jnr and C. A. Costa (1998). Information interactions in data model driven design for manufacture. Globalization of Manufacturing in the Digital Communications era of the 21st Century: Innovation, Agility and the Virtual Enterprise, Kluwer Academic Publishers, ISBN:0 412 83540 1, Pages: 313-324.

Yourdon, E. (1994). Object-oriented Systems Design: An Integrated Approach, Prentice Hall, ISBN:0136363253.

Zaijun Hu, E. Kruse and L. Draws (2003). "Intelligent binding in the engineering of automation systems using ontology and Web services." IEEE Transactions on Systems, Man and Cybernetics, Part C,, Volume:33 , Issue: 3, Pages:403 - 412.

Zarli, A. and V. Amar (1997). "Integrating STEP and CORBA for applications interoperability in the future virtual enterprises computer-based infrastructures". Proceedings IASTED International Conference on Intelligent Information Services 97, Grand Bahama Island

Zhao, J., W. M. Cheung, R. I. Young and R. Bell (1999). "An Object Oriented Manufacturing Data Model for a Global Enterprise". Proceedings of the 15th International Conference on Computer-Aided Production Engineering (CAPE '99), Durham, UK, April, pp 582-588.

Zhao, J., W. M. Cheung and R. I. M. Young (1999(a)). "A Consistent Manufacturing Data Model to Support Virtual Enterprises." International Journal of Agile Management Systems, Volume:1(3), pp 150-158.

Appendix I: Definitions For MSE Ontology Model in OWL

MSE Ontology

The top-level abstract classes for the manufacturing system engineering ontology provides taxonomy of manufacturing system engineering domain with focus on the operation of an EEMSE moderator.

Classes

Project, Flow, Process, Resource, Strategy, Entended_Enterprise, Enterprise

Properties

Travels_along, carries, links, linked_by, uses_resource, used_in_process, applies, controls, has_enterprise, has_strategy, has_process, has_resource.

OWL File

MSE_TOP.owl

```
<rdf:RDF
  xmlns="http://owl.protege.stanford.edu#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#">
 <owl:Ontology rdf:about=""/>
 <owl:Class rdf:ID="Strategy"/>
 <owl:Class rdf:ID="Project">
  <rdfs:subClassOf>
   <owl:Restriction>
    <owl:onProperty>
     <owl:ObjectProperty rdf:about="#travels_along"/>
    </owl:onProperty>
```

```
      <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
      >1</owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Process">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#uses_resource"/>
      </owl:onProperty>
      <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
      >1</owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Enterprise">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#has_process"/>
      </owl:onProperty>
      <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
      >1</owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#has_resource"/>
      </owl:onProperty>
      <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
      >1</owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Resource">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#used_in_process"/>
      </owl:onProperty>
      <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
      >1</owl:minCardinality>
    </owl:Restriction>
```

```
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Flow">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#linked_by"/>
      </owl:onProperty>
      <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
      >1</owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Extended_Enterprise">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#has_enterprise"/>
      </owl:onProperty>
      <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
      >2</owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:ObjectProperty rdf:ID="has_strategy">
  <rdfs:domain rdf:resource="#Enterprise"/>
  <rdfs:range rdf:resource="#Strategy"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="applies">
  <rdfs:domain rdf:resource="#Process"/>
  <rdfs:range rdf:resource="#Strategy"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="has_process">
  <rdfs:domain rdf:resource="#Enterprise"/>
  <rdfs:range rdf:resource="#Process"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="uses_resource">
  <rdfs:domain rdf:resource="#Process"/>
  <rdfs:range rdf:resource="#Resource"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="travels_along">
  <rdfs:domain rdf:resource="#Project"/>
  <rdfs:range rdf:resource="#Flow"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="linked_by">
```

```
  <rdfs:domain rdf:resource="#Flow"/>
  <rdfs:range rdf:resource="#Process"/>
 </owl:ObjectProperty>
 <owl:ObjectProperty rdf:ID="carries">
  <rdfs:domain rdf:resource="#Flow"/>
  <rdfs:range rdf:resource="#Project"/>
 </owl:ObjectProperty>
 <owl:ObjectProperty rdf:ID="links">
  <rdfs:domain rdf:resource="#Process"/>
  <rdfs:range rdf:resource="#Flow"/>
 </owl:ObjectProperty>
 <owl:ObjectProperty rdf:ID="used_in_process">
  <rdfs:domain rdf:resource="#Resource"/>
  <rdfs:range rdf:resource="#Process"/>
 </owl:ObjectProperty>
 <owl:ObjectProperty rdf:ID="has_resource">
  <rdfs:domain rdf:resource="#Enterprise"/>
  <rdfs:range rdf:resource="#Resource"/>
 </owl:ObjectProperty>
 <owl:ObjectProperty rdf:ID="controls">
  <rdfs:domain rdf:resource="#Strategy"/>
  <rdfs:range rdf:resource="#Process"/>
 </owl:ObjectProperty>
 <owl:ObjectProperty rdf:ID="has_enterprise">
  <rdfs:domain rdf:resource="#Extended_Enterprise"/>
  <rdfs:range rdf:resource="#Enterprise"/>
 </owl:ObjectProperty>
</rdf:RDF>
```

Project Ontology

Extended Project taxonomy for triggering the formation and operation of the extended / virtual manufacturing process and describing the project team of a building project.

Classes

Project, Physical_item, Program, Document, Part, Product, Assembly, Component, Customer_order, Drawing, Contract, Flow

Properties

project_name, project_desc, travels_along, supplier_reference_no, quantity, item_no, document_no, …etc.

OWL File

MSE_Project.owl

```
<rdf:RDF
  xmlns="http://owl.protege.stanford.edu#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#">
 <owl:Ontology rdf:about=""/>
 <owl:Class rdf:ID="Document">
  <rdfs:subClassOf>
   <owl:Restriction>
    <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
    >1</owl:minCardinality>
    <owl:onProperty>
     <owl:DatatypeProperty rdf:about="#document_no"/>
    </owl:onProperty>
   </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
   <owl:Class rdf:about="#Project"/>
  </rdfs:subClassOf>
 </owl:Class>
 <owl:Class rdf:ID="Physical_item">
  <rdfs:subClassOf>
   <owl:Class rdf:about="#Project"/>
```

```
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Customer_order">
  <rdfs:subClassOf rdf:resource="#Document"/>
  <owl:disjointWith>
    <owl:Class rdf:about="#Contract"/>
  </owl:disjointWith>
</owl:Class>
<owl:Class rdf:ID="Drawing">
  <rdfs:subClassOf rdf:resource="#Document"/>
</owl:Class>
<owl:Class rdf:ID="Program">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Project"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Component">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Part"/>
  </rdfs:subClassOf>
  <owl:disjointWith>
    <owl:Class rdf:about="#Assembly"/>
  </owl:disjointWith>
</owl:Class>
<owl:Class rdf:ID="Product">
  <rdfs:subClassOf rdf:resource="#Physical_item"/>
  <owl:disjointWith>
    <owl:Class rdf:about="#Part"/>
  </owl:disjointWith>
</owl:Class>
<owl:Class rdf:ID="Project">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
      >1</owl:minCardinality>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#travels_along"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Flow"/>
<owl:Class rdf:ID="Contract">
  <rdfs:subClassOf rdf:resource="#Document"/>
  <owl:disjointWith rdf:resource="#Customer_order"/>
```

```
 </owl:Class>
 <owl:Class rdf:ID="Part">
  <rdfs:subClassOf rdf:resource="#Physical_item"/>
  <owl:disjointWith rdf:resource="#Product"/>
 </owl:Class>
 <owl:Class rdf:ID="Assembly">
  <rdfs:subClassOf rdf:resource="#Part"/>
  <owl:disjointWith rdf:resource="#Component"/>
 </owl:Class>
 <owl:ObjectProperty rdf:ID="travels_along">
  <rdfs:domain rdf:resource="#Project"/>
  <rdfs:range rdf:resource="#Flow"/>
 </owl:ObjectProperty>
 <owl:DatatypeProperty rdf:ID="document_no">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#Document"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="item_no">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#Physical_item"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="project_name"
   rdf:type="http://www.w3.org/2002/07/owl#FunctionalProperty">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#Project"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="supplier_reference_no">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#Physical_item"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="project_desc"
   rdf:type="http://www.w3.org/2002/07/owl#FunctionalProperty">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#Project"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="quantity">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdfs:domain rdf:resource="#Physical_item"/>
 </owl:DatatypeProperty>
</rdf:RDF>
```

Process Ontology

Process ontology provides taxonomy of business and manufacturing functions or activities that are essential to the operation of all enterprises.

OWL File

MSE_Process.owl

```
<rdf:RDF
  xmlns="http://owl.protege.stanford.edu#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#">
 <owl:Ontology rdf:about=""/>
 <owl:Class rdf:ID="Quality_assurance">
  <rdfs:subClassOf>
   <owl:Class rdf:about="#Production"/>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="#Quality_assurance"/>
 </owl:Class>
 <owl:Class rdf:ID="Prototype_planning">
  <rdfs:subClassOf>
   <owl:Class rdf:about="#New_product_introduction"/>
  </rdfs:subClassOf>
 </owl:Class>
 <owl:Class rdf:ID="Production_planning">
  <rdfs:subClassOf><owl:Class rdf:about="#Production"/>
  </rdfs:subClassOf>
 </owl:Class>
 <owl:Class rdf:ID="Material_purchasing">
  <rdfs:subClassOf><owl:Class rdf:about="#Materials_management"/>
  </rdfs:subClassOf>
 </owl:Class>
 <owl:Class rdf:ID="Product_assembly">
  <rdfs:subClassOf><owl:Class rdf:about="#Production"/>
  </rdfs:subClassOf>
 </owl:Class>
 <owl:Class rdf:ID="Sales_marketing">
  <rdfs:subClassOf rdf:resource="#Process"
   rdf:type="http://www.w3.org/2002/07/owl#Class"/>
 </owl:Class>
 <owl:Class rdf:ID="Production">
```

```
    <rdfs:subClassOf rdf:resource="#Process"/>
  </owl:Class>
  <owl:Class rdf:ID="Materials_management">
    <rdfs:subClassOf rdf:resource="#Production"/>
  </owl:Class>
  <owl:Class rdf:ID="Customer_acceptance">
    <rdfs:subClassOf> <owl:Class rdf:about="#New_product_introduction"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Test">
    <rdfs:subClassOf><owl:Class rdf:about="#New_product_introduction"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Product_delivery">
    <rdfs:subClassOf rdf:resource="#Production"/>
  </owl:Class>
  <owl:Class rdf:ID="Inventory_management">
    <rdfs:subClassOf rdf:resource="#Production"/>
  </owl:Class>
  <owl:Class rdf:ID="New_product-design">
    <rdfs:subClassOf><owl:Class rdf:about="#Product_design"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="New_product_introduction">
    <rdfs:subClassOf rdf:resource="#Process"/>
</owl:Class>
  <owl:Class rdf:ID="Product_design">
    <rdfs:subClassOf rdf:resource="#New_product_introduction"/>
  </owl:Class>
  <owl:Class rdf:ID="Enterprise_investment">
    <rdfs:subClassOf rdf:resource="#Process"/>
  </owl:Class>
  <owl:Class rdf:ID="Financial_control">
    <rdfs:subClassOf rdf:resource="#Process"/>
  </owl:Class>
  <owl:Class rdf:ID="Production_customisation">
    <rdfs:subClassOf rdf:resource="#Product_design"/>
  </owl:Class>
  <owl:DatatypeProperty rdf:ID="process_name">
    <rdfs:domain rdf:resource="#Process"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  </owl:DatatypeProperty>
</rdf:RDF>
```

Resource Ontology

Resource terminology describes mechanisms that enable a MSE process to be executed.

OWL File

MSE_Resource.owl

```
<rdf:RDF
  xmlns="http://owl.protege.stanford.edu#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#">
 <owl:Ontology rdf:about=""/>
 <owl:Class rdf:ID="Human_resource">
  <rdfs:subClassOf rdf:resource="#Resource"
    rdf:type="http://www.w3.org/2002/07/owl#Class"/>
 </owl:Class>
 <owl:Class rdf:ID="IT">
  <rdfs:subClassOf>
   <owl:Class rdf:about="#Equipment"/>
  </rdfs:subClassOf>
 </owl:Class>
 <owl:Class rdf:ID="Hardware">
  <rdfs:subClassOf rdf:resource="#IT"/>
 </owl:Class>
 <owl:Class rdf:ID="Raw_material">
  <rdfs:subClassOf>
   <owl:Class rdf:about="#Material"/>
  </rdfs:subClassOf>
 </owl:Class>
 <owl:Class rdf:ID="Budget">
  <rdfs:subClassOf rdf:resource="#Resource"/>
 </owl:Class>
 <owl:Class rdf:ID="Part_material">
  <rdfs:subClassOf>
   <owl:Class rdf:about="#Material"/>
  </rdfs:subClassOf>
  <owl:disjointWith>
   <owl:Class rdf:about="#Component_material"/>
  </owl:disjointWith>
 </owl:Class>
```

```
<owl:Class rdf:ID="Software">
  <rdfs:subClassOf rdf:resource="#IT"/>
</owl:Class>
<owl:Class rdf:ID="Component_material">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Material"/>
  </rdfs:subClassOf>
  <owl:disjointWith rdf:resource="#Part_material"/>
</owl:Class>
<owl:Class rdf:ID="Machinery">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Production_resource"/>
  </rdfs:subClassOf>
  <owl:disjointWith>
    <owl:Class rdf:about="#Tool"/>
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:about="#Equipment"/>
  </owl:disjointWith>
</owl:Class>
<owl:Class rdf:ID="Communication">
  <rdfs:subClassOf rdf:resource="#Hardware"/>
</owl:Class>
<owl:Class rdf:ID="Computer">
  <rdfs:subClassOf rdf:resource="#Hardware"/>
</owl:Class>
<owl:Class rdf:ID="Equipment">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Production_resource"/>
  </rdfs:subClassOf>
  <owl:disjointWith rdf:resource="#Machinery"/>
</owl:Class>
<owl:Class rdf:ID="Production_resource">
  <rdfs:subClassOf rdf:resource="#Resource"/>
</owl:Class>
<owl:Class rdf:ID="Material">
  <rdfs:subClassOf rdf:resource="#Production_resource"/>
</owl:Class>
<owl:Class rdf:ID="Tool">
  <rdfs:subClassOf rdf:resource="#Production_resource"/>
  <owl:disjointWith rdf:resource="#Machinery"/>
</owl:Class>
</rdf:RDF>
```

Strategy Ontology

Strategy ontology provides terminologies of the objectives, constraints and operation rules that can influence decisions made by the enterprise relating to the use of enterprise facilities, resources and process.

OWL File

MSEStrategy.owl


```
<rdf:RDF
  xmlns="http://owl.protege.stanford.edu#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#">
 <owl:Ontology rdf:about=""/>
 <owl:Class rdf:ID="Product_design_strategy">
  <rdfs:subClassOf>
   <owl:Class rdf:about="#Manufacturing_strategy"/>
  </rdfs:subClassOf>
 </owl:Class>
 <owl:Class rdf:ID="Manufacturing_strategy">
  <rdfs:subClassOf><owl:Class rdf:about="#Objective"/>
  </rdfs:subClassOf>
 </owl:Class>
 <owl:Class rdf:ID="Operational_rule">
  <rdfs:subClassOf rdf:resource="#Strategy"
    rdf:type="http://www.w3.org/2002/07/owl#Class"/>
 </owl:Class>
 <owl:Class rdf:ID="Simple_action">
  <rdfs:subClassOf>
   <owl:Class rdf:about="#Resulting_action"/>
  </rdfs:subClassOf>
 </owl:Class>
 <owl:Class rdf:ID="Objective">
  <rdfs:subClassOf rdf:resource="#Strategy"/>
 </owl:Class>
 <owl:Class rdf:ID="Sales_marketing">
  <rdfs:subClassOf>
   <owl:Class rdf:about="#Business_strategy"/>
  </rdfs:subClassOf>
 </owl:Class>
 <owl:Class rdf:ID="Constraint">
```

```
  <rdfs:subClassOf rdf:resource="#Strategy"/>
 </owl:Class>
 <owl:Class rdf:ID="Material_control_strategy">
  <rdfs:subClassOf rdf:resource="#Manufacturing_strategy"/>
 </owl:Class>
 <owl:Class rdf:ID="Design_new_product">
  <rdfs:subClassOf rdf:resource="#Product_design_strategy"/>
 </owl:Class>
 <owl:Class rdf:ID="Resulting_action">
  <rdfs:subClassOf rdf:resource="#Operational_rule"/>
 </owl:Class>
 <owl:Class rdf:ID="Condition">
  <rdfs:subClassOf rdf:resource="#Operational_rule"/>
 </owl:Class>
 <owl:Class rdf:ID="Design_for_customisation">
  <rdfs:subClassOf rdf:resource="#Product_design_strategy"/>
 </owl:Class>
 <owl:Class rdf:ID="Enterprise">
  <rdfs:subClassOf>
   <owl:Class rdf:about="#Business_strategy"/>
  </rdfs:subClassOf>
 </owl:Class>
 <owl:Class rdf:ID="Compound_condition">
  <rdfs:subClassOf rdf:resource="#Condition"/>
 </owl:Class>
 <owl:Class rdf:ID="Simple_condition">
  <rdfs:subClassOf rdf:resource="#Condition"/>
 </owl:Class>
 <owl:Class rdf:ID="Financial">
  <rdfs:subClassOf>
   <owl:Class rdf:about="#Business_strategy"/>
  </rdfs:subClassOf>
 </owl:Class>
 <owl:Class rdf:ID="Compound_action">
  <rdfs:subClassOf rdf:resource="#Resulting_action"/>
 </owl:Class>
 <owl:Class rdf:ID="Capacity_strategy">
  <rdfs:subClassOf rdf:resource="#Manufacturing_strategy"/>
 </owl:Class>
 <owl:Class rdf:ID="Business_strategy">
  <rdfs:subClassOf rdf:resource="#Objective"/>
 </owl:Class>
</rdf:RDF>
```

Extended_Enterprise Ontology

Extended_Enterprise ontology provides classification and relationship between enterprise, factory, shop, cell and station within the extended / virtual enterprises environment.

OWL File

MSE_EE.owl

```
<rdf:RDF
  xmlns="http://owl.protege.stanford.edu#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#">
 <owl:Ontology rdf:about=""/>
 <owl:Class rdf:ID="Factory">
  <rdfs:subClassOf rdf:resource="#Enterprise"
    rdf:type="http://www.w3.org/2002/07/owl#Class"/>
 </owl:Class>
 <owl:Class rdf:ID="Cell">
  <rdfs:subClassOf rdf:resource="#Enterprise"/>
 </owl:Class>
 <owl:Class rdf:ID="Extended_Enterprise">
  <rdfs:subClassOf>
   <owl:Restriction>
    <owl:onProperty>
     <owl:ObjectProperty rdf:about="#has_enterprise"/>
    </owl:onProperty>
    <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
    >2</owl:minCardinality>
   </owl:Restriction>
  </rdfs:subClassOf>
 </owl:Class>
 <owl:Class rdf:ID="Shop">
  <rdfs:subClassOf rdf:resource="#Enterprise"/>
 </owl:Class>
 <owl:Class rdf:ID="Station">
  <rdfs:subClassOf rdf:resource="#Enterprise"/>
 </owl:Class>
 <owl:ObjectProperty rdf:ID="has_cell">
  <rdfs:domain rdf:resource="#Shop"/>
  <rdfs:range rdf:resource="#Cell"/>
```

```
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="has_factory">
  <rdfs:domain rdf:resource="#Enterprise"/>
  <rdfs:range rdf:resource="#Factory"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="has_station">
  <rdfs:domain rdf:resource="#Cell"/>
  <rdfs:range rdf:resource="#Station"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="has_shop">
  <rdfs:domain rdf:resource="#Factory"/>
  <rdfs:range rdf:resource="#Shop"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="has_enterprise">
  <rdfs:domain rdf:resource="#Extended_Enterprise"/>
  <rdfs:range rdf:resource="#Enterprise"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:ID="factory_location">
  <rdfs:domain rdf:resource="#Factory"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="ee_name">
  <rdfs:domain rdf:resource="#Extended_Enterprise"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="enterprise_name">
  <rdfs:domain rdf:resource="#Enterprise"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="enterprise_desc">
  <rdfs:domain rdf:resource="#Enterprise"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="factory_desc">
  <rdfs:domain rdf:resource="#Factory"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
</rdf:RDF>
```

Publications

The following papers have been published, or submitted for publication related to this research:

1. Lin, H. K. and J. A. Harding, (2003), '**An Ontology Driven Manufacturing System Engineering Moderator for Global Virtual Enterprise Teams**', Advances in Manufacturing Technology XVII, Proceedings of the International Conference on Manufacturing Research, Strathclyde, UK, September, Y. Qin and N. Juster, (eds), Professional Engineering Publishing Ltd, UK, pp 365-370, ISBN 1-86058-412-8.

2. Lin, H.K., J.A. Harding, and M. Shahbaz, (2004) '**Manufacturing System Engineering Ontology for Semantic Interoperability Across Extended Project Teams**', International Journal of Production Research, Vol 42, No. 24, pp 5099-5118, ISBN 0020-7543.

3. Lin, H.K., J.A. Harding (2004) '**A Manufacturing System Engineering Web Ontology Model for Inter-enterprise Collaborative Working**', submitted to Computers In Industry.