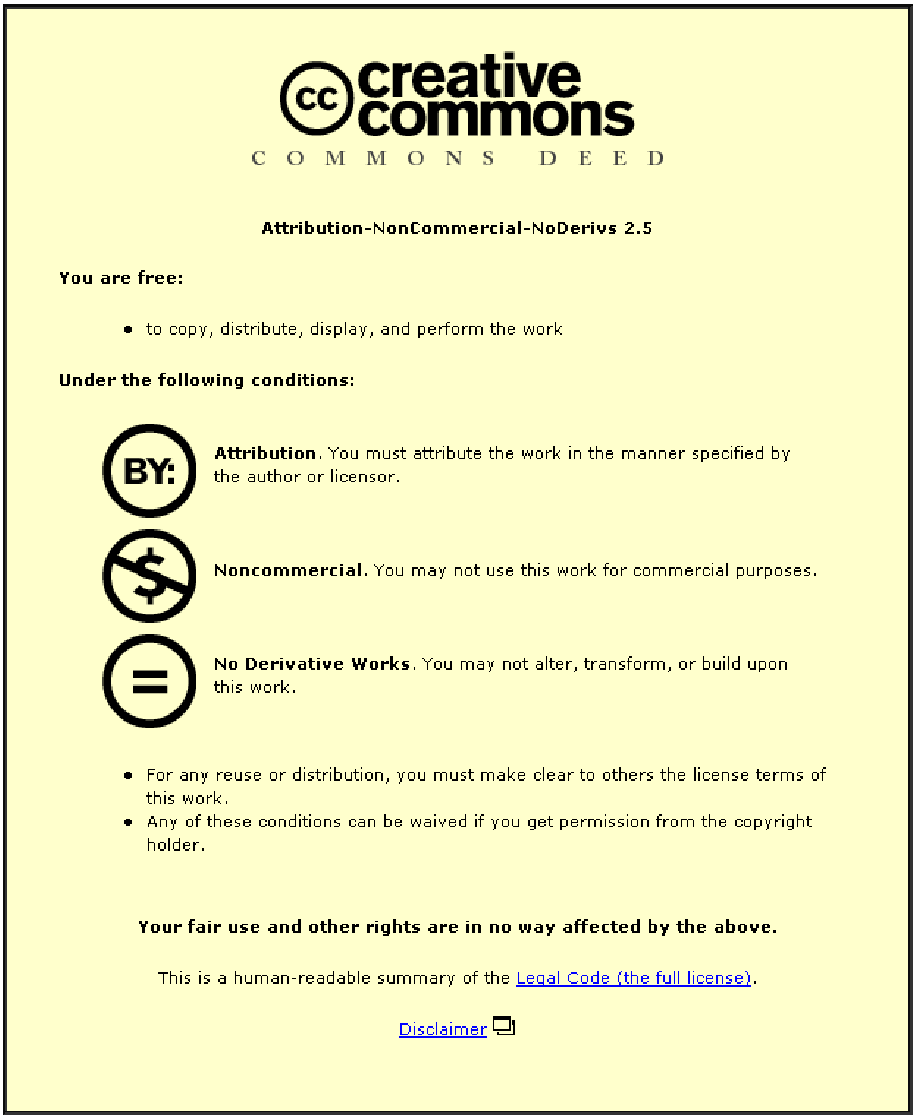


Title Manufacturing System Engineering Ontology for Semantic Interoperability Across Extended Project Teams

Author Hsiao-Kang Lin, J. A. Harding, M. Shahbaz

Wolfson School of Mechanical and Manufacturing Engineering,
Loughborough University, Loughborough, Leicestershire LE11 3TU, UK.

Abstract:

Communication, knowledge sharing and awareness of available expertise are complex issues for any multi-discipline team. Complexity increases substantially in extended enterprise environments. The concepts of an MSE moderator have previously been considered in environments with shared information models and vocabularies. These concepts are now translated to the realm of extended enterprises where inevitably individual partners will have their own terminology and information sources. An MSE ontology is proposed to enable the operation of an extended enterprise MSE Moderator, to provide common understanding of manufacturing-related terms, and therefore to enhance the semantic interoperability and reuse of knowledge resources within globally extended manufacturing teams.

Keywords: Manufacturing System Engineering Ontology, Extended Project Teams, Semantic Web, Resource Description Framework (RDF), RDF-Schema, Inference Rule.

1. Introduction

Communication between project teams and different organizations within extended enterprises 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***, (i.e. projects taking place in a virtual enterprise or extended enterprise, that include participants from different companies as members of a global extended manufacturing 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). Even the international standards in the area of enterprise engineering and integration are developed independently by different standards organizations using incompatible and inconsistent terminologies (Kosanke and de Meer 2001).

A solution to this problem is the development of a taxonomy of manufacturing concepts and terms to make design knowledge effectively accessible across extended enterprise team 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, is proposed in this paper.

The context for the proposed MSE Ontology is to provide an environment for the application of a Manufacturing System Engineering (MSE) Moderator within extended enterprises, called EEMSE Moderator. The concepts and examples of Moderators (to support both Product Design and Manufacturing System Engineering) have been previously reported in (Harding and Popplewell 1996; Harding et al. 1999b; Harding et al. 2003; Lin and Harding 2003). A Moderator is an intelligent support application that is designed to facilitate and improve concurrent engineering design by enhancing the degree of awareness, cooperation, and coordination among engineering team members. This is a complex task for any multi-disciplined team, particularly, in large enterprises when the team members may be located at different global locations. However, the task is further complicated when team members come from an extended or virtual enterprise, 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. There is therefore an important requirement to make design knowledge effectively accessible across virtual enterprise team members, by using an explicit and accepted well-defined terminology (Lin and Harding, 2003).

The proposed MSE Ontology has been designed to provide a common understanding of manufacturing-related terms, and therefore to enhance the semantic interoperability and reuse of knowledge resources within global extended manufacturing teams. This paper contains several important concepts; firstly the MSE Ontology model that defines a terminology for building knowledge representation in a manufacturing system domain is introduced. This is illustrated using Protégé http://protege.stanford.edu/, which is a set of tools designed to automate the process of building domain-specific knowledge acquisition and knowledge based systems. The paper then demonstrates how the defined MSE Ontology is converted into the formal ontology language, Resource Description Framework (RDF), and Resource Description Framework Schema (RDFS), to serve as ontology metadata that may be used to create, delete, modify, and query the MSE Ontology. The application ontology is input to the RDF Gateway http://www.intellidimension.com/, an application server built around a powerful RDF-based deductive database. Finally, the subject of interest is extended beyond just knowledge representation and knowledge manipulation. A method of ontology inference is proposed by building sets of declarative mapping rules that could be applied to map all shareable semantic metadata between the common MSE Ontology and any manufacturing system model for their semantic and syntax integration.

2. Manufacturing Systems Engineering (MSE) Ontology Model

Manufacturing system information models, such as CIMOSA (Kosanke et al. 1999), MOSES (Ellis et al. 1994), (Molina and Bell 1999), FDM (Harding and Yu 1999a) 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. 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 conceptualization 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 and knowledge. PSL creates a neutral, standard language for process specification to integrate multiple process-related applications throughout the manufacturing life cycle. The Standard for the Exchange of Product Model Data (STEP) effort also supports the exchange of information by aiming to create an interlingua for exchanging manufacturing product data. However, 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 extended-enterprise MSE Moderator. Manufacturing Systems Engineering 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 (Harding et al. 2003) 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 MSE Ontology model is presented, using an ontology modeling technique. The term ontology is defined in Collins English dictionary as: " Ontology is a philosophical discipline, a branch of metaphysics that deals with the nature of being." This definition provides an abstract philosophical notion of ontology. Interest has recently been increasing in the concepts of ontology and this is partly due to the growing needs of the artificial intelligence community to develop a terminology for building knowledge bases for particular domains at a level which can be understood by machines. 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). 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. catalogs), glossary of simple terms, class taxonomies (an informal is-a relation / a formal is-a relation), frames (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.

The MSE Ontology model uses Protégé and its Plugins as a basis for expressing ontologies and converting the informal vocabularies into the formal language – RDF/RDFS/OWL. Protégé has been chosen for use in our experimental environment as it allows the user to construct a domain ontology, so that classes and class hierarchy, properties (Protégé calls these slots) and slot-value restrictions, relationships between classes and properties of these relationships can all be defined. The Protégé Plugins http://protege.stanford.edu/plugins.html are built on top of Protégé as the extension of the ontology-modeling tool for various possible applications requiring creation and edition of ontologies and ontology instances. For example, a range of Plugins exist such as the RDF storage backend (to create, import and save RDF(S) files in Protégé), OWL Plugin (to load, save, and edit OWL ontolgies in Protégé) to support semantic web application, and OntoViz Plugin (that enables the user to visualize Protégé ontologies with the help of graph drawing) for visualization.

Comment [J1]: I assume that you want to change or add something here.

As previously stated, 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 MSE Moderator has therefore been designed to monitor design decisions, evaluate their significance to individual project team members and communicate with any team members deemed necessary (Harding et al. 2003) . However, the original MSE Moderator reported in (Harding et al. 2003) was designed to operate within a single enterprise using shared (common) information and terminology. The situation is very different within an extended enterprise 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 ontology of manufacturing within an extended enterprise.

2.1 Class Vocabularies

All manufacturing enterprises are different, but they do have natural, common characteristics, which have been captured in seven key base classes within the MSE ontology model. These key base classes have been determined using the knowledge and experiences of published Manufacturing system information models, (Harding et al, 1999c; Zoha et al, 1999; Kosanke et al, 1999; Harding et al, 2003), in addition to the Extended_Enterprise class to support the extended enterprise environment. The seven top-level classes: Project, Flow, Process, Enterprise, Extended_Enterprise, 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 1 shows elements of the class structure and some relationships between classes that have been captured using protégé and are displayed using its visualisation ontologies, OntoViz Plugin.

[Insert figure 1]

The extended enterprise is a dynamic network or loose coalition system of organisations. The business processes of participating companies 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 extended enterprise is disbanded when the goal has been achieved, and participating companies go their individual ways, or recombine to form further extended enterprises. Goranson described the lifecycle of an agile virtual enterprise (extended enterprise) in terms of its various stages: opportunity identification, partner identification, formation, operation and reconfiguration / dissolution (Goranson 1999).

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, contracts, or program. The Project class and a section of its hierarchy is represented in Figure 2. Each instance of the Project class travels along one or more flows (instances of the

Flow class) that connect independent processes or activities into a system with a purpose.

[Insert figure 2]

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 2 illustrates the common business processes in a manufacturing enterprise and a section of the Process class hierarchy. 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).

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). He 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 superclass 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 and this 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 superclass of classes Factory, Shop, Cell, and Station. In addition, the aggregation relationships defined by Zhao are also included in MSE Ontology. Later in this section, a representation of a section of the Enterprises Class hierarchy and aggregation relationships and instances will be explained and shown in figure 6.

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 1999a), the Enterprise Ontology (Uschold et al. 1998)…etc. The *Resources 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 machinery tools, raw materials …etc (see figure 2). 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).

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 (Harding et al, 1999 IJPR Paper) supports 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 1996). In the MSE Ontology, the strategy concept is implemented from the FDM model. Figure 3 shows a section of the *Strategy class* hierarchy and the slots (properties) definition from the MSE Ontology in Protégé-2000.

[Insert figure 3]

2.2 Property Vocabularies and Instances

Each class has ***properties*** that may be thought of as attributes of the class and can also represent relationships between classes. Figure 4 shows some of the elements of the classes and properties structure and relationships for executing a new order flow for an extended project. Figure 5 fills in the values for properties for instances.

[Insert figure 4]

An example of planning and control of order flow for semiconductor manufacturing through an extended enterprise project, using the MSE Ontology model are now presented. Initially, a new customer’s order (order_number: LU3223-1) shown in figure 5 as ‘LU3223-1’ is defined as an instance of the Customer_order class (subclass of the Documents class and the Project class). Each instance of this class contains the properties of order_number, order_date, and quantity, …etc, and inherits all the properties of its superclass, such as project_name, project_leader, project_ team, and travels_along. Additionally, the slot-value type[1] of the travels-along property is an Instance-type slot that allows definition of relationships between the Project class and Flow class. Slots with value type Instance must also define a list of allowed classes from which the instances can come. Therefore each travels_along's instance points to an instance of the Flow class to build the relationships between Project class and Flow class. 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_procurement_process class in order to obtain new raw material for the production, and the linked_by property connects the Production_planning_process class for the production scheduling. Furthermore, production-planning process requires several resources for the process, e.g. production resource, storage resource, through uses_resource property links to the Resource class. Figure 5 illustrates the graph-drawing instance of the example with protégé OntoViz Plugin.

[Insert figure 5]

Figure 6 demonstrates the classes' hierarchy, properties attributes and relations, and instances of a section of the capacity of an individual participating enterprise and its responsibility for the specific process for the extended enterprise project shown in figure 5. For example, the extended_enterprise (RISC processors – 3223-1 project) has two enterprises, TSME and AREM. TSME has two factories USA Fab 12 and

Taiwan Fab 3. The Material_procurement process is the responsibility of the Taiwan Fab3 factory and the Production_planning process is the responsibility of the AREM enterprise.

[Insert figure 6]

3. MSE Ontology Database (Semantic Metadata)

An ontology must be encoded in some language to express the concepts in the domain in a manner that computers can manipulate meaningfully. There are a number of ontology specification languages such as Classic Knowledge Representation System http://www.bell-labs.com/project/classic/ (Ronald J. Brachman et al. 1991), Description-Logic Knowledge Representation System Specification (KRSS) (Patel-Schneider P. F. and Swartout 1993) and Knowledge Interchange Format (KIF) (KIF 1999). More recently in the research literature for semantic web technologies (Lassila et al. 2000; McGuinness et al. 2002; McGuinness and Van Harmelen 2003) languages have been used to represent instantiated ontologies and to structure collections of data and sets of inference rules for semantic browsers. Figure 7 shows the web ontology representation language architecture for the semantic web framework.

[Insert figure 7]

The lowest layer, the syntax layer provides a syntactic representation of the ontology and the knowledge base using the Extensible Markup Language (XML). XML presents text structure for humans to read on the web, but does not contain markup information about the contents of the page for computer manipulation. The next layers, the data layer and the ontology layer, are based on the RDF, RDF-Schema (RDFS) and different language primitive vocabularies (e.g. OWL, DAML+OIL, and F-Logic …etc) that provide simple data model defined terms and their relationships to other terms. Finally, the logic layer provides formal semantics that allow the implications of the term definitions and relationships to be deduced.

In the MSE Ontology, RDF and RDFS are used for the metadata model. RDF is a W3C http://www.w3c.org/RDF standard for describing machine-processable semantics of metadata which is represented by the *subject-predicate-object* triples, in a similar fashion to semantic nets or frame-based systems. RDF/XML builds upon XML syntax to provide a mechanism for exchanging semantics over the Internet. Figure 8 [a] illustrates an example of the RDF model from the MSE Ontology, the subject (on[2]:order) is shown as an ellipse connected by two arcs that link named properties (predicates) (c:order_date and c:quantity) plus the values for those properties (29 Oct 2003 and 5000). It does this in a way that allows the computers that process this information to understand its meaning. The meaning in RDF is expressed through reference to a RDFS. A RDFS is the place where definitions concept (or class) hierarchies and restrictions of usage for properties are documented. Figure 8 [b] shows a section of RDFS from the model given in figure 8[a]. The RDFS elements specification can be found at http://www.w3.org/2000/01/rdf-schema#. Several complementary technologies are associated with RDF, such as Jena (McBride 2002), a Java-based API that supports the creation, manipulation, and query RDF graphs; KAON http://kaon.semanticweb.org/, provides programmatic access to

RDF models; Sesame http://sesame.aidministrator.nl/, an open source RDF Schema-based repository and querying facility; and RDF Gateway http://www.intellidimension.com/, a database and integrated web server.

[Insert figure 8]

3.1 RDF and the Relational Databases (RDB) Model

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. RDF is a model of entities (nodes) and relationships, which is basically an opening (i.e. increasing accessibility) 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. Every table has columns, 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. However, a relational schema is created independently for each database. This makes it difficult to share information between systems that do not share the identical relational 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, and this in fact is what provides the interoperability. Figure 9 shows the difference between RDF statement and a relational database structure.

[Insert figure 9]

There is a large quantity of existing data already stored using relational database technology, and this needs to be exploited by the new RDF technology. Tools are currently available for mapping the RDB schemas onto RDF, and these include: Jena relational database interface, which stores its data in a relational database; D2R Map - Database to RDF Mapping Language and Processor http://www.wiwiss.fu-berlin.de/suhl/bizer/d2rmap/ D2Rmap.htm, which is a declarative language to describe mappings between relational database schemata and OWL ontologies. The mappings can be used by a D2R processor to export data from a relational database into RDF and RDF Gateway uses the SQL Data Service to get the RDF schema for a relational database.

3.2 RDF Database Query Language

There are many RDF database query languages, such as Jena's RDQL(McBride 2002) which is an SQL-like syntax for this query model derived from SquishQL http://swordfish.rdfweb.org/ rdfquery/ and rdfDB http://guha.com/rdfdb/query.html, Sesame's SeRQL (pronounced as "circle") http://sesame.aidministrator.nl/, and RDF Gateway's RDFQL. The following is an example of the RDFQL query language which enables databases to be created, rows to be inserted or deleted and the data to be queried. A row in this language would be an RDFQL statement {Predicate, Subject, Object}. For instance, to create a database, a table, and insert rows, the following syntax can be used (RDFGateway 2003).

```
CREATE DATABASE mse;
CREATE TABLE  mse_ontology;

INSERT  {[rdf:type] [on:order] [c:Customer_order]}

        {[c:order_date] [on:order] '29Oct2003'}

        {[c:travels_along] [on:order] [oe:OrderEntry]}

        {[rdf:type] [oe:OrderEntry] [c:Flow]}

        {[c:carries] [oe:Order Entry] 'LU3223-1'}

INTO  mse_ontology;
```

The SELECT command is used to query RDFQL’s deductive database. It returns bindings for the variables in the variable list that meet the conditions specified in the WHERE clause. Only the datasources (tables) specified in the USING clause are queried against. The following syntax shows SELECT query.

```
SELECT    ?Odate

USING     mse_ontology

WHERE    {[c:order_date] [c:order] ?Odate};
```

When the query is examined, a pattern {[c:order_date] [c:order] ?Odate} can be seen for triples in the RDF database. This pattern is matched against each triple in the database and the results collected together (in the example, there is only one such match {[c:order_date] [on:order] '29Oct2003'}, therefore the return result for the variable ?Odate is ‘29Oct2003”).

4. The Semantic and Syntax Integration

4.1 The Mapping Process Steps

In an extended enterprise, MSE systems have been created independently, and do not share the same semantics for the terminology of their manufacturing models. For example, different information models may be used by different parts of the extended enterprise project teams. Examples from two existing MSE models, the FDM and the MISSION model, are shown in figure 10[a], and each of these have been built to meet the objectives of different research project needs. The examples will now be used to demonstrate the operation of the MSE ontology model. Both the Token class in the FDM application and the MSE-Item class in the MISSION application correspond to a common concept of an object, e.g. a project. This syntax problem of applications could be parsed using the 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 from n*m to n+m by requiring that an application only maps its concepts to the concepts of a common ontology rather than mapping it to all the other applications. In contrast, both the FDM and MISSION models have the same term Process, however, this term can have a different meaning, where Process in FDM is applied to a range of types of processes, including business function entities (e.g. sales, marketing, and production, etc) and more traditional manufacturing processing (e.g. milling, grinding, etc). In contrast, the process in MISSION model is a route entity (e.g. connecting). As a result, the semantic problems identified in section 1 occur for Process. 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. Figure 10[a] and figure 10[b] show the semantic and syntax integration by mapping to the common MSE Ontology.

[Insert figure 10]

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 RDF model in our research.
2. Then the equivalence mappings between the terminologies and the common MSE Ontology are specified. (see figure 10)
3. Each model is then converted into the RDF database to serve as ontology metadata that may be used to create, delete, modify, and query their ontology.
4. Finally, a set of reuse inference rules are developed that encode the mappings between classes and their properties. These are represented in the RDF Schema (RDFS) (see figure 12), to enable automatic deduction from the RDF database for sharing or exchanging different terminologies.

In this research, the MSE Ontology and the other models (e.g. FDM model and Mission model …etc) have been tested using Protégé 2000 and the RDF Gateway and these have proved valuable for the experimental implementations. The first three steps of the mapping process scenario were demonstrated in the previous section. The final step, the development of inference rules, provides mechanisms for deduction of information, and this is an important characteristic of ontology-based systems. The definition of semantic patterns is now demonstrated with some instantiation examples from our research. Computers that process RDF can share disparate information by mapping from one schema to another. The mapping approach adopted in this research is to model the axiom specification in the RDFS in an object-oriented manner. Following the object-oriented tradition, RDFS provides the special property rdfs:subClassOf that defines the subclass relationship 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, this semantic mapping defines two inference rules, the subclass rule and 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:Customer_order class if and only if it is an instance of the f:Product_order class and the f:Product_order class is a subclass of the c:Customer_order class (see figure 11[a]). The following additional example demonstrates the subproperty rule. A value (e.g. 09Jan2004) is a instance of the subproperty of the c:order_date property if and only if it is the instance of the f:product_order_date property and the f:product_order_date is subproperty of the c:order_date (see figure 11[b]). In this example, a set of mapping rules are provided that can be used in a query in RDFQL. The rules should ensure that when someone queries the MSE Ontology model for the instances of the c:order_date, the result includes all instances of the f:product_order_date from the FDM model.

[Insert figure 11]

Figure 12 illustrates the semantic and syntax integration for all systems to map to the common MSE Ontology schema.

[Insert figure 12]

4.2 Application of the MSE Ontology

The mapping process scenario presented in the previous section completes the description of the four steps that are essential for the application of the MSE Ontology. A short discussion is now presented to summarise how the MSE ontology would work with the EEMSE Moderator, in the context of an extended enterprise project.

The scenario that will be considered involves several companies coming together to form an extended enterprise, called ExeCorp. The objective of ExeCorp is to design and manufacture a specialist product, which contains parts (components and assemblies) which are to be provided by individual members of the extended enterprise, to best exploit the skills and experiences of each of the individual partners. ExeCorp contains several members, but to simplify this discussion, only 3 will be considered here. L U Technology Ltd (who will act as overall project managers and also accommodate the final product assembly process), TSME (an American company with manufacturing facilities in the Far East) and AREM (an SME which is a specialist design house based in Europe), see figure 5 and figure 6. LU Technology Ltd., TSME and AREM each have their individual business interests in addition to their involvement with ExeCorp. They also form parts of other extended enterprises and supply chains. Each company therefore also 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 practises and experiences in particular industry sectors, the culture in their particular organisation, and many other contributory factors.

Each partner within ExeCorp 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 naturally do not work with a common language or common information models or structures. The role of a Moderator in this context is therefore more complex than in the earlier reported projects (Harding and Popplewell, 1996 and Harding et al, 2003) where common information models were utilised. The general concepts of design moderation and the modular design of the Moderator's knowledge base, which enables it to contain and structure knowledge of individual contributors' (or design agents') areas of interest, will not be discussed here. The MSE Moderator structure has been used as the basis for the initial work on the EEMSE Moderator, and hence, details can be found in the earlier papers. However, two substantial differences do exists in the case of the EEMSE Moderator, because partners in ExeCorp (or in any other extended team) may

(1) express design information changes (including additions or deletions) in different languages and terminology and

(2) express information or knowledge of what they consider to be important aspects of the design (eg key variables or values) in different languages or terminology.

The first difference directly affects the EEMSE Moderator's design moderation process and the second difference affects both the EEMSE Moderator'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 EEMSE Moderator possible, by providing a mechanism for dealing with these differences.

The EEMSE Moderator's design moderation process should be activated whenever a change is made to information related to the extended enterprise's joint project. A change could occur for any item of information, for example, LUTechnology may need to change an order number for some administrative reason. The other partners should also be made aware of this change, and the EEMSE Moderator may therefore need to communicate with them. Assume initially that some participants in ExeCorp use information models in their business. If, for example, LUTechnology used information models based on the FDM model and TSME used information models based on the Mission Model, it has already been shown (fig 10) that information from these could be translated to the neutral MSE Ontology. As can be seen in figure 10, LUTechnology would use the term "Product_order" when making the change, and the EEMSE Moderator would then use the MSE Ontology to determine that the change had been made to a "Customer_order". It could then use this fact to process its knowledge base to see who needed to be notified of the change. The EEMSE Moderator could apply translators to change information provided by these participants into the neutral MSE Ontology format, and then check its knowledge base to see which partners should be made aware of the change. To then communicate with the relevant partner, the EEMSE could again invoke translators, this time from the MSE Ontology into the partner's language. Hence TSME could be notified of a change to an "Order"

The MSE Ontology could also be used by the EEMSE in its Knowledge Acquisition mode (Harding et al, 2003) when it needs to update its knowledge base with knowledge or information about new or existing partners. In this case knowledge of the extended enterprise partners would be translated into the neutral MSE Ontology format to be inserted into the EEMSE Moderator's knowledge base.

Assume now that some partners in ExeCorp do not use established information models. For example, AREM is an SME and is therefore likely to be limited in the investments that it can afford to make in information systems. However, the use of standard tools in the design of the MSE Ontology should enable translations to be made between the MSE Ontology (which is RDF based) and the relational databases, which are commonly used in businesses. Therefore even smaller partners should potentially be able to participate in the EEMSE environment.

5. Conclusion and Discussion

In this paper, an ontology based approach for an extended enterprise MSE Moderator has been introduced. This has necessitated the examination and development of common terminologies for manufacturing system engineering design. A comprehensive semantic web language based on RDF and RDFS was defined and further extended to axioms and rules. This is required to solve the syntax and semantic problems in extended project teamwork environment. The focus of this paper is directed more on the usage of RDF and RDFS than on the theoretical semantic web (see figure 7). However, our research is ongoing and will continue to improve and to keep in line with current semantic web technology, such as OWL

Future implementation should therefore support more powerful inference and query.

REFERENCE

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.

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), June, pp 377-387.

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,

Fox, M. S. and M. Gruninger, (1997), 'On Ontologies and Enterprise Modelling', International Conference on Enterprise Integration Modelling Technology, Springer-Verlag,

Goranson, H. T., (1999), 'The Agile Virtual Enterprise: Cases, Metrics, Tools', Quorum Books, ISBN: 1567202640.

Gruber, T. R., (1993), 'A Translation Approach to Portable Ontology Specifications', Knowledge Acquisition, Volume: 5(2), pp 199-220.

Harding, J. A., (1996), 'A Knowledge Representation Model to Support Concurrent Engineering Team Working', PhD Thesis, Loughborough University, UK.,

Harding, J. A., A. R. Omar and K. Popplewell, (1999b), 'Application of QFD within a concurrent engineering environment 1(2)', International Journal of Agile Management Systems, Volume: 1(2), pp 88-98.

Harding, Yu and Popplewell (1999, I think) IJPR Paper on FDM Views

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), ISBN 0020 7543, 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), pp 1973-1986.

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

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

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

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

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), Nov/Dec, pp 67-73.

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.

McBride, B., (2002), 'Jena: A Semantic Web Toolkit', Internet Computing, IEEE, Volume: 6, Nov/Dec, 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), Sep/Oct, 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,

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,

Patel-Schneider P. F. and B. Swartout, (1993), 'Description-Logic Knowledge Representation System Specification', The KRSS Group of the DARPA Knowledge Sharing Effort

RDFGateway, (2003), 'Intellidimension RDF Gateway Documentation', http://www.intellidimension.com/,

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', Morgan Kaufmann, ISBN: 1558600884.

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

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,

NOTES

[1] The slot-value type in protégé 2000 includes: Any, Boolean, Class, Float, Instance, Integer, String, Symbol

[2] The XML namespace mechanism plays a crucial role for the development of RDF(S) and its application. The detail of XML namespace can be found in http://www.w3.org/TR/REC-xml-names.

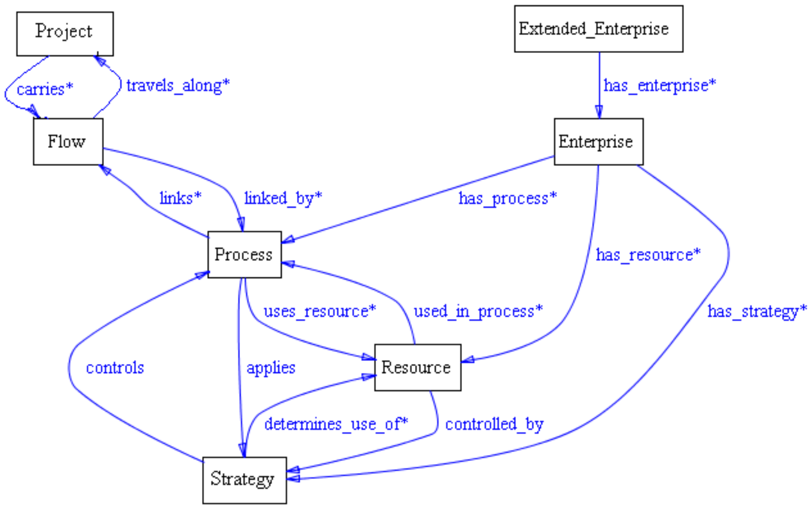
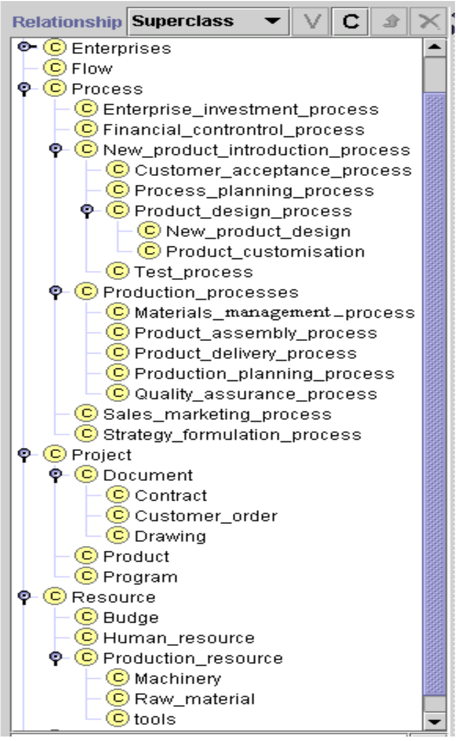


Figure 1: Top-level abstract classes from the MSE Ontology model

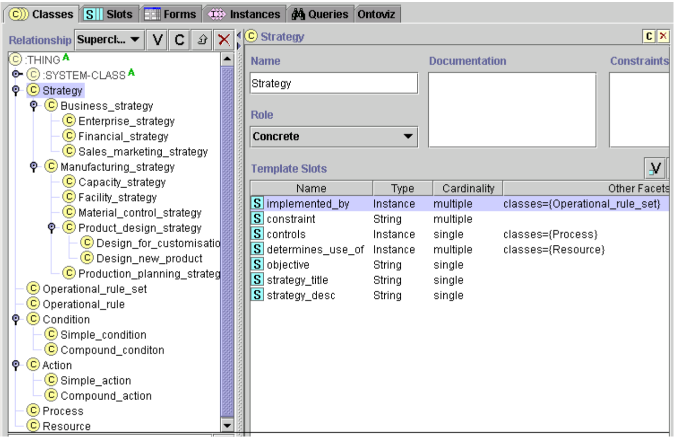
Boxes represent classes and arrows represent relations.

* Multiple cardinality



```
C Enterprises
C Flow
C Process
    C Enterprise_investment_process
    C Financial_controntrol_process
    C New_product_introduction_process
        C Customer_acceptance_process
        C Process_planning_process
        C Product_design_process
            C New_product_design
            C Product_customisation
        C Test_process
    C Production_processes
        C Materials_management_process
        C Product_assembly_process
        C Product_delivery_process
        C Production_planning_process
        C Quality_assurance_process
    C Sales_marketing_process
    C Strategy_formulation_process
C Project
    C Document
        C Contract
        C Customer_order
        C Drawing
    C Product
    C Program
C Resource
    C Budge
    C Human_resource
    C Production_resource
        C Machinery
        C Raw_material
        C tools
```

Figure 2: A section of the Class hierarchy from the MSE Ontology



- :THING
- :SYSTEM-CLASS
- Strategy
  - Business_strategy
    - Enterprise_strategy
    - Financial_strategy
    - Sales_marketing_strategy
  - Manufacturing_strategy
    - Capacity_strategy
    - Facility_strategy
    - Material_control_strategy
    - Product_design_strategy
      - Design_for_customisatio
      - Design_new_product
    - Production_planning_strateg
- Operational_rule_set
- Operational_rule
- Condition
  - Simple_condition
  - Compound_conditon
- Action
  - Simple_action
  - Compound_action
- Process
- Resource

Strategy

Name: Strategy

Role: Concrete

Documentation

Constraints

Name	Type	Cardinality	Other Facets
implemented_by	Instance	multiple	classes={Operational_rule_set}
constraint	String	multiple	
controls	Instance	single	classes={Process}
determines_use_of	Instance	multiple	classes={Resource}
objective	String	single	
strategy_title	String	single	
strategy_desc	String	single	

Figure 3: Editing a section of the Strategy Ontology in Protégé-2000

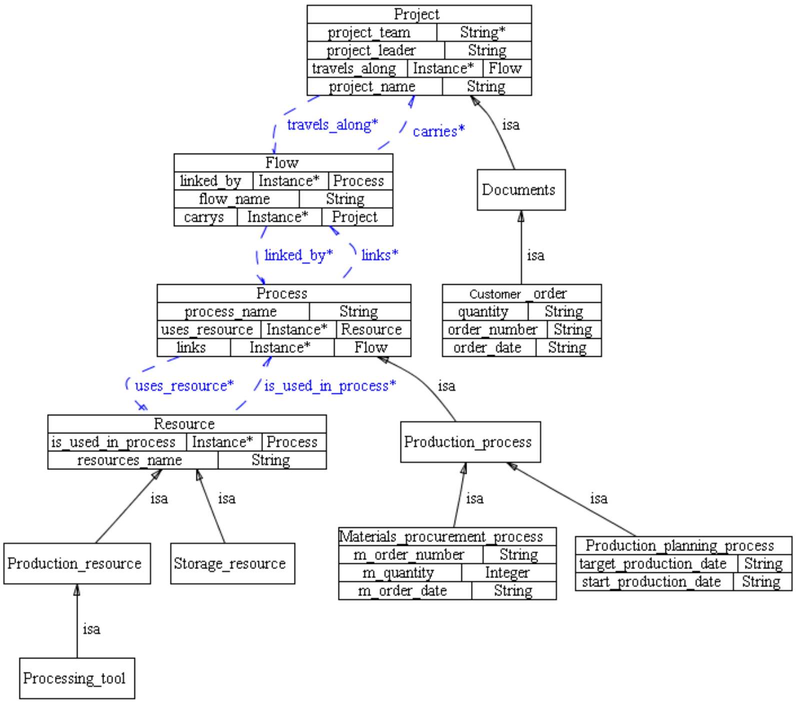
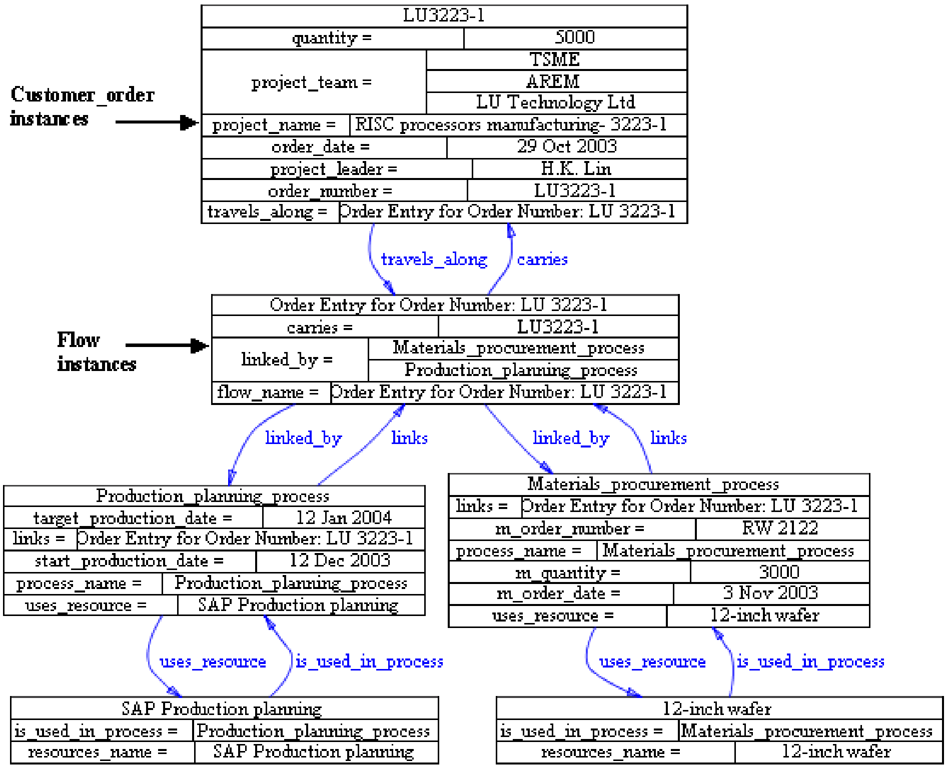


Figure 4: The attributes and relationships of the properties between

the Product, Flow, Process, and Resource classes



LU3223-1
quantity = 5000
project_team = TSME
AREM
LU Technology Ltd
project_name = RISC processors manufacturing- 3223-1
order_date = 29 Oct 2003
project_leader = H.K. Lin
order_number = LU3223-1
travels_along = Order Entry for Order Number: LU 3223-1

Order Entry for Order Number: LU 3223-1
carries = LU3223-1
linked_by = Materials_procurement_process
Production_planning_process
flow_name = Order Entry for Order Number: LU 3223-1

Production_planning_process
target_production_date = 12 Jan 2004
links = Order Entry for Order Number: LU 3223-1
start_production_date = 12 Dec 2003
process_name = Production_planning_process
uses_resource = SAP Production planning

Materials_procurement_process
links = Order Entry for Order Number: LU 3223-1
m_order_number = RW 2122
process_name = Materials_procurement_process
m_quantity = 3000
m_order_date = 3 Nov 2003
uses_resource = 12-inch wafer

SAP Production planning
is_used_in_process = Production_planning_process
resources_name = SAP Production planning

12-inch wafer
is_used_in_process = Materials_procurement_process
resources_name = 12-inch wafer

Figure 5: Instances of executing a new order for extended project

(Based on figure 4 classes and properties)

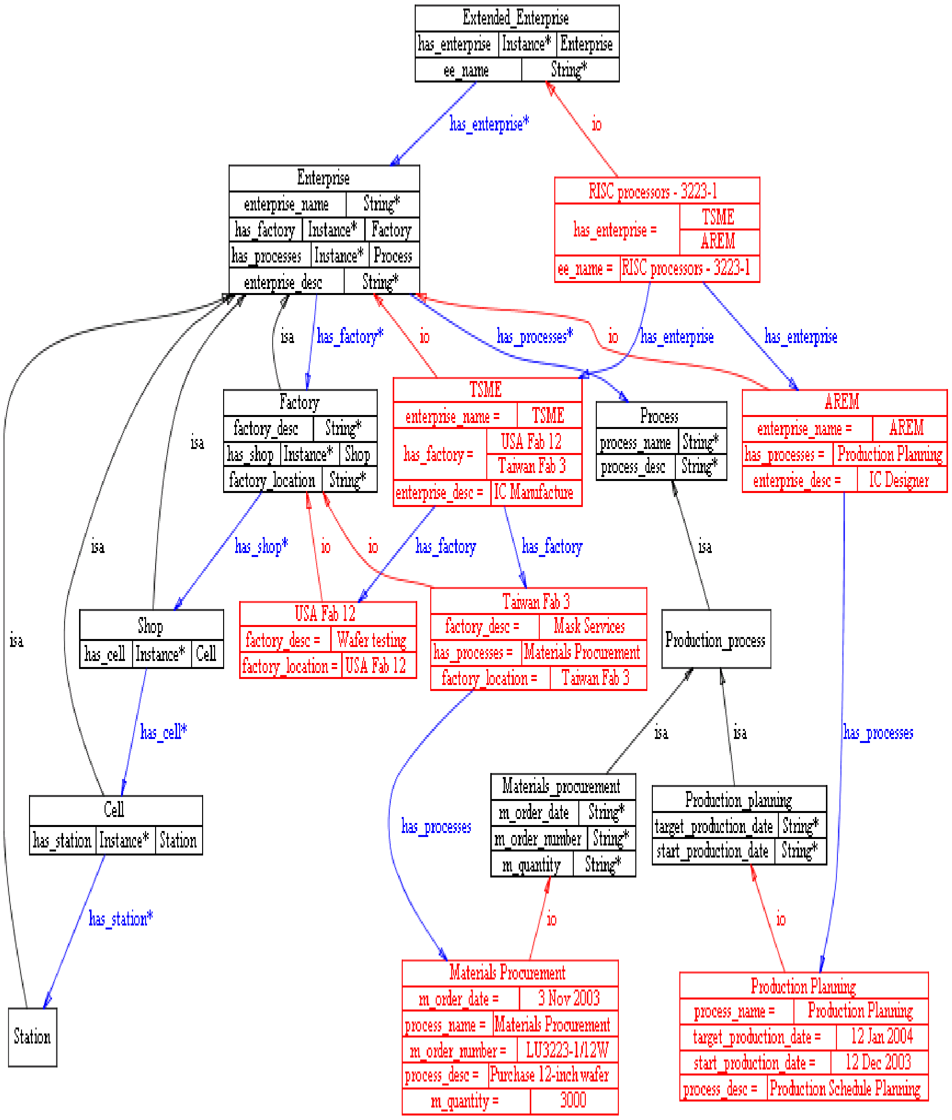

Extended_Enterprise
has_enterprise | Instance* | Enterprise
ee_name | String*
has_enterprise*
io
Enterprise
enterprise_name | String*
has_factory | Instance* | Factory
has_processes | Instance* | Process
enterprise_desc | String*
RISC processors - 3223-1
has_enterprise = | TSME | AREM
ee_name = | RISC processors - 3223-1
isa
has_factory*
io
has_processes*
io
has_enterprise
has_enterprise
Factory
factory_desc | String*
has_shop | Instance* | Shop
factory_location | String*
TSME
enterprise_name = | TSME
has_factory = | USA Fab 12 | Taiwan Fab 3
enterprise_desc = | IC Manufacture
Process
process_name | String*
process_desc | String*
AREM
enterprise_name = | AREM
has_processes = | Production Planning
enterprise_desc = | IC Designer
isa
isa
isa
has_shop*
io
io
has_factory
has_factory
isa
Shop
has_cell | Instance* | Cell
USA Fab 12
factory_desc = | Wafer testing
factory_location = | USA Fab 12
Taiwan Fab 3
factory_desc = | Mask Services
has_processes = | Materials Procurement
factory_location = | Taiwan Fab 3
Production_process
isa
has_cell*
isa
isa
has_processes
has_processes
Cell
has_station | Instance* | Station
Materials_procurement
m_order_date | String*
m_order_number | String*
m_quantity | String*
Production_planning
target_production_date | String*
start_production_date | String*
has_station*
io
io
Station
Materials Procurement
m_order_date = | 3 Nov 2003
process_name = | Materials Procurement
m_order_number = | LU3223-1/12W
process_desc = | Purchase 12-inch wafer
m_quantity = | 3000
Production Planning
process_name = | Production Planning
target_production_date = | 12 Jan 2004
start_production_date = | 12 Dec 2003
process_desc = | Production Schedule Planning

Figure 6: The classes hierarchy, properties, and instances

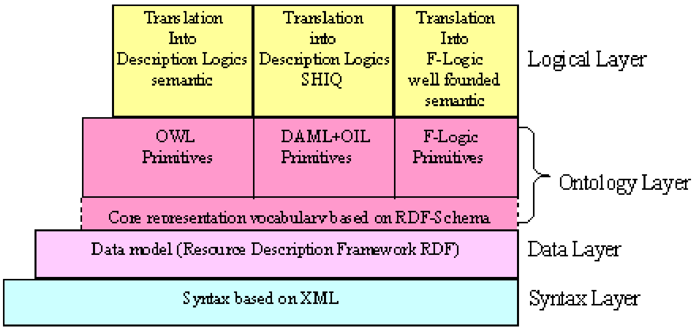

Translation Into Description Logics semantic
Translation into Description Logics SHIQ
Translation Into F-Logic well founded semantic
Logical Layer
OWL Primitives
DAML+OIL Primitives
F-Logic Primitives
Ontology Layer
Core representation vocabulary based on RDF-Schema
Data model (Resource Description Framework RDF)
Data Layer
Syntax based on XML
Syntax Layer

Figure 7: Web Ontology Representation Layers

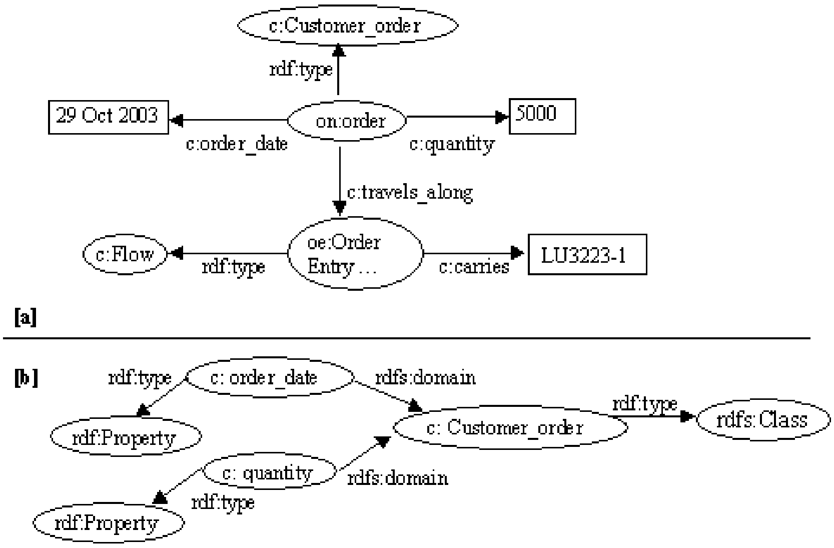
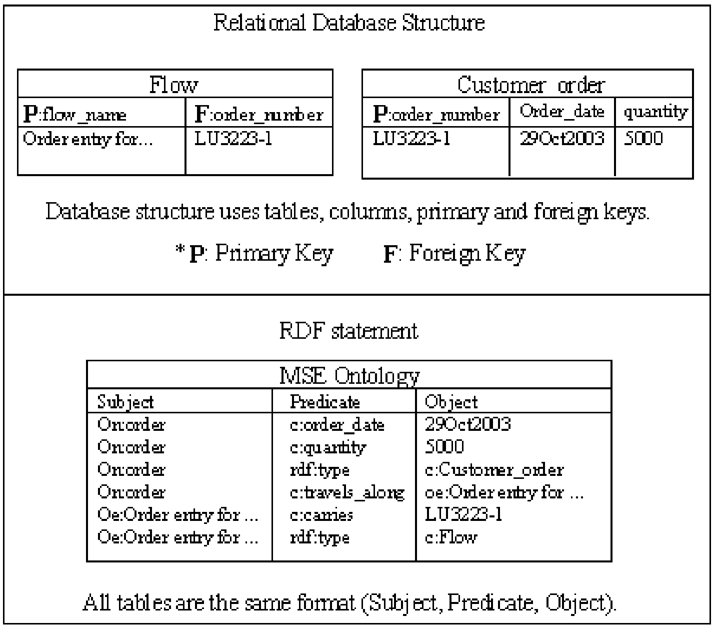


Figure 8: An example of the RDF Model and RDF-Schema



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

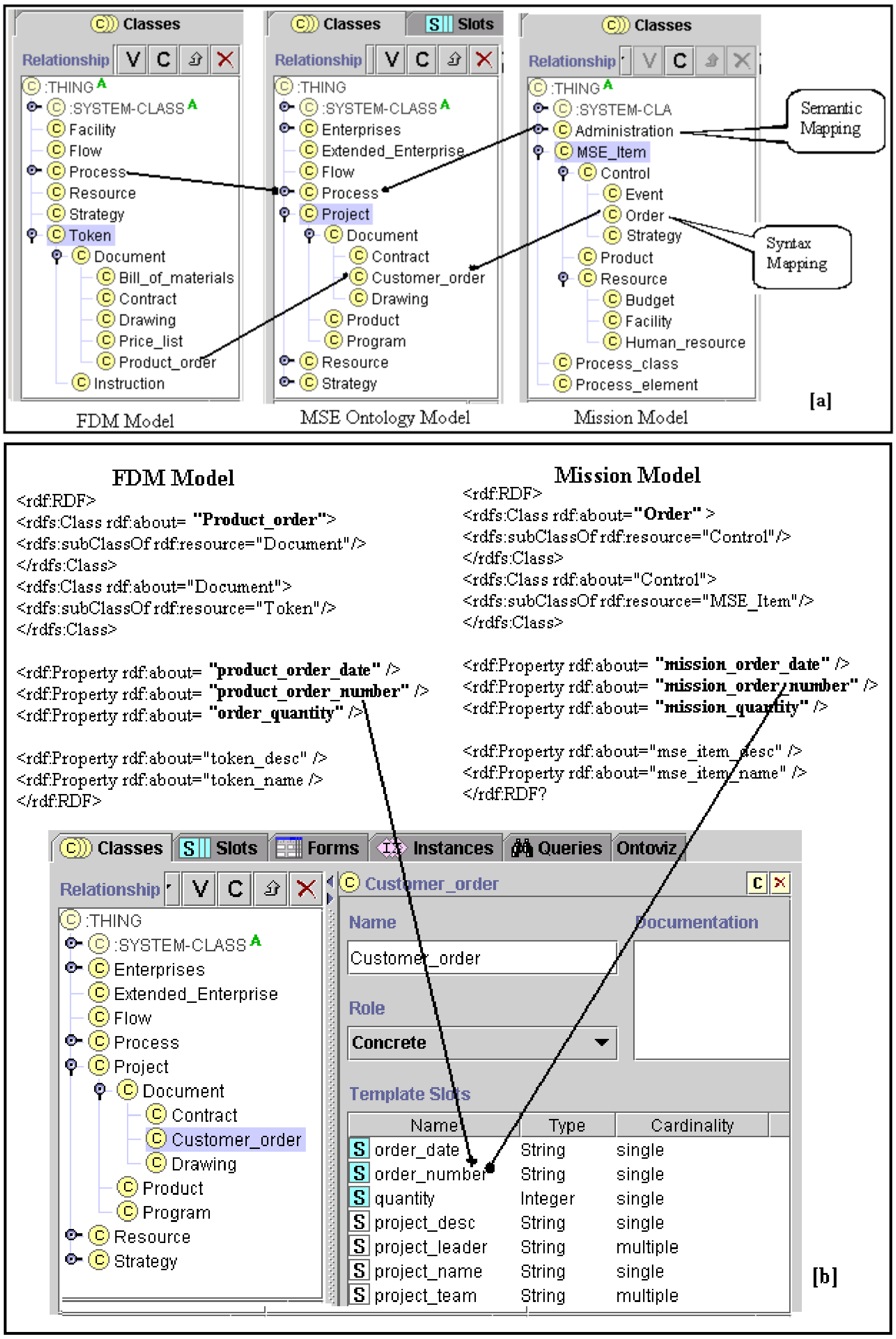
RDF statement

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 9: The difference between RDF and RDB model



FDM Model

```
<rdf:RDF>
<rdfs:Class rdf:about= "Product_order">
<rdfs:subClassOf rdf:resource="Document"/>
</rdfs:Class>
<rdfs:Class rdf:about="Document">
<rdfs:subClassOf rdf:resource="Token"/>
</rdfs:Class>

<rdf:Property rdf:about= "product_order_date" />
<rdf:Property rdf:about= "product_order_number" />
<rdf:Property rdf:about= "order_quantity" />

<rdf:Property rdf:about="token_desc" />
<rdf:Property rdf:about="token_name />
</rdf:RDF>
```

Mission Model

```
<rdf:RDF>
<rdfs:Class rdf:about="Order" >
<rdfs:subClassOf rdf:resource="Control"/>
</rdfs:Class>
<rdfs:Class rdf:about="Control">
<rdfs:subClassOf rdf:resource="MSE_Item"/>
</rdfs:Class>

<rdf:Property rdf:about= "mission_order_date" />
<rdf:Property rdf:about= "mission_order_number" />
<rdf:Property rdf:about= "mission_quantity" />

<rdf:Property rdf:about="mse_item_desc" />
<rdf:Property rdf:about="mse_item_name" />
</rdf:RDF?
```

Name: Customer_order

Role: Concrete

Template Slots

Name	Type	Cardinality
S order_date	String	single
S order_number	String	single
S quantity	Integer	single
S project_desc	String	single
S project_leader	String	multiple
S project_name	String	single
S project_team	String	multiple

Figure 10: The Common MSE Ontology

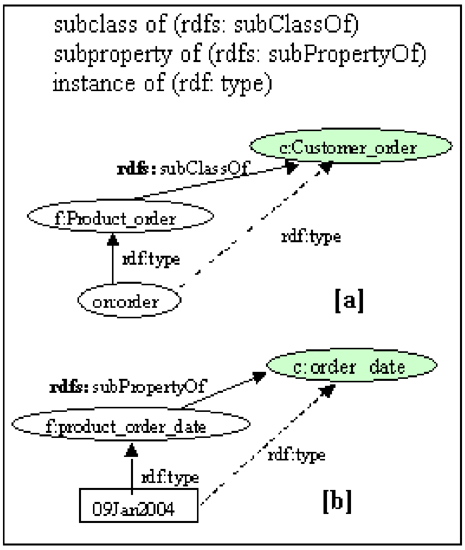


Figure 11: The recursive algorithms

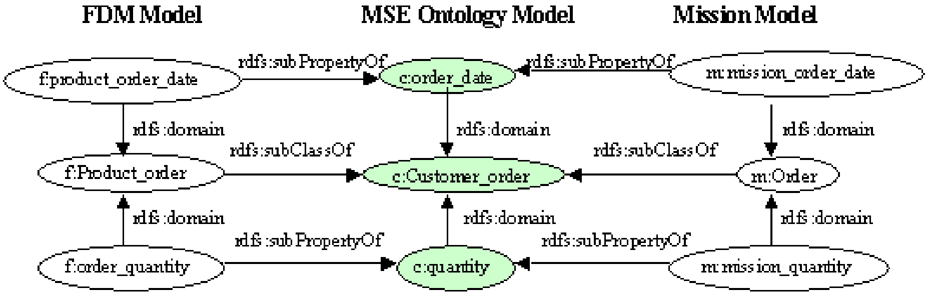


Figure 12: Mapping to the MSE Ontology Model using the RDF Schema