Approach for Information Systems Semantic Interoperability in Supply Chain Environment

Yan Lu

分类号：TP14　　　　　　　　　　　　单位代码：10335
密　级：公开　　　　　　　　　　　　学　　号：10508042

浙江大学

博士学位论文

中文论文题目：供应链环境下的信息系统语义互操作方法

英文论文题目：**Approach for Information Systems Semantic Interoperability in Supply Chain Environment**

申请人姓名：吕艳

指导教师：顾新建 教授

合作导师：Hervé　Panetto

专业名称：机械制造及自动化

研究方向：制造业信息化

所在学院：机械系

论文提交日期 **2012-04-17**

Abstract

Multiple enterprises are usually involved in supply chain network, while almost each enterprise has its own information systems and databases (enterprise legacy systems). There are large quantities of information and data intersecting or even repeating with each other, which calls for system integration. Meanwhile in the cooperative process of supply chain, the whole product line or product family is involved in the enterprise systems, including PDM, ERP, SCM, and CRM et al, which also need interaction and integration. Although many data integration methods and corresponding tools have been applied to the business cooperation in the environment of supply chain, however the interoperability of enterprise information systems is becoming more and more complex, because of the restriction from lots of factors, such as the complexity of enterprise application systems and the diversity of heterogeneous data sources, especially the new application demands of enterprises,.

In this dissertation, the interoperability method of networked enterprise information systems in the environment of supply chain is studied, from the point of theoretical modeling and application. The main research contents are as follows:

Chapter 1 states the research background of the dissertation, analyzes the domestic and overseas research status of supply chain integration, and proposed semantic interoperation is an effective solution to deal with the integration of enterprise information systems. Meanwhile it analyzes and summarizes the research status of correlated techniques involved in semantic interoperation. Finally, the organizational chart of the dissertation is provided, stating the foundation, content and significance of research.

Chapter 2 analyzes the SoS (System-of-Systems) characteristic of networked enterprises in the environment of supply chain, proposed the SoNE (Systems-of-Networked Enterprises) paradigm. And then it analyzes the distinction between the interoperability and integration of enterprise information systems. Finally, it proposes the information system interoperability framework of supply chain

enterprises based on the demand of information system interoperability in the environment of supply chain.

Chapter 3 studies the developing method of product ontology. Based on some international standards related to enterprise system integration (such as IEC 62264, ISO 10303 and STEP-PDM et al.), the model-driven ontology developing method is adopted to integrate the business information and technology data related to product life cycle, including product design, make, deliver and et al, into a common shared product ontology.

Chapter 4 discusses the building method of supply chain ontology. Adopting SCOR model as the basic concepts and framework for supply chain ontology, the SCOR supply chain ontology is built by Protégé seven-step method. Meanwhile the semantic description is made by OWL DL. Taking make-to-order of main machine bed as an example, an instance of SCOR supply chain ontology is given. Finally, a simple evaluation system is illustrated about SCOR supply chain ontology.

Chapter 5 researches the ontology merging and semantic interoperability methods. The method combining ontology mapping based on WordNet with reasoning based on rules is adopted to realize the merging of product ontology and supply chain ontology, to achieve a product-centric supply chain ontology. The information systems interoperability of supply chain enterprises is realized by the mapping among ontology and the mapping between databases and ontology. Finally, an instance is put forward in this chapter.

Chapter 6 takes the supply chain process of the double column CNC guideway&surface grinding machine in Machine Tool Company as an example to validate the method for information systems interoperability of supply chain enterprises. Meanwhile, according to interoperability principle, an information systems interoperability platform prototype of supply chain enterprises is developed, to provide as a feasible way for the information integration of supply chain enterprises.

Chapter 7 makes a summary for the whole dissertation, and looks into the future direction about this research topic.

Keywords: ontology; semantic interoperability; SCOR; supply chain; enterprise information system

Index

1 Introduction

1.1 Introduction

With the advent and continuously enhancing of global economic from the 1990s, the information industry has raised and develop continuously by the promoting of technological revolution represented by computer technology, biotechnology and microelectronics technology, which has leaded to the rapid transformation from traditional industrial economy to knowledge and information economy. In the global economic environment, the markets and global manufacturing are changing rapidly, which brings great changes to the competitive environment, management mode, business pattern, technical approach and other aspects, causing huge pressure for enterprises. In order to respond to such pressure, enterprises build closer relationship with suppliers and distributors. which exist not only in large international enterprises, but also appears in small and medium-sized enterprises. Enterprises cannot be viewed as being isolated, and enterprise collaboration is no longer just between two partners; they have evolved to what has been described as ‘enterprise networks’, in forms of supply chain partnerships, extended enterprises and virtual enterprises [Jagdev, 2001].

Supply chain is a system with organization, people, technology, activity, information and resource involved in, to deliver a product or service from suppliers to customers. Supply chain activity transforms natural resources, raw materials and components into final products, and delivers them to customers. A network is composed by the enterprises and enterprise departments involved in this process. The most basic request of supply chain operation is to minimize the inventory and preparation time of whole supply chain, while the following premises should be satisfied when achieving this goal [Jagdev, 2001]:

(1) The clear understanding among supply chain partners must be reached, while the opposite expectation and request should be comprehended clearly.

(2) Seamless material and information system.

(3) The effective communication must exist among the market, sale, purchase,

manufacturing plan and manufacturing control and et al. of each node.

(4) The information exchange among all nodes must be very effective, ensuring the high efficiency of supply chain.

(5) The information and decision support system of node must be fast response, satisfying the constantly changing demand and the communication with corresponding nodes.

(6) With the converting of industry from product orientation to client orientation, supply chain need make corresponding operation change.

(7) Some issues just like quality management and its incessant improvement consist of a part of the treaty, which are established at the time when supply chain is built.

In the premises mentioned above, the most core demand is that the enterprises in supply chain could interact effectively Although there has been substantial research literature on supply chain interactions, these perspectives and proposed approaches vary greatly. One common tenet is that competitive success depends on managers' ability to recognize changes in the competitive environment and then to structure organizational, and where appropriate, supply chain resources to effectively meets customers' real needs[Stanley,2001]. This is a fundamental tenet of Supply Chain Management (SCM)implementation.

Meanwhile in industrial companies, rapidly developed ICT (Information and Communications Technology) is being adopted to handle supply chain management issues, ranging from ERP (Enterprise Resource Planning) to MPC (Manufacturing Planning and Control) and MES (Manufacturing Execution System), facilitating various aspects of the supply chain [Kollberg,2006]. It provides a new way to store, process, distribute and exchange information both within companies and with customers and suppliers in the supply chain. This environment has increasing demands for information or knowledge exchange among enterprises. Therefore in informationalized supply chain, the effective interaction among enterprises mainly depends on the interaction among the information systems of enterprises, which is usually realized by enterprises integration or system integration.

1.2 Research Background

1.2.1 Problems of systems integration in supply chain environment

Supply chain network is usually involved in various enterprises, while almost each enterprise has its own applications and databases (enterprise legacy system). An approach is needed to make the existing applications and databases adjusted to new environment, bring benefit in new environment. There are plenty of information and data intersecting and even repeating with each other, which calls for system integration. Meanwhile during the cooperation process of supply chain, the involved enterprise system is not a single software, but the whole product line or product family about each aspect of enterprise management, including PDM, ERP, SCM and CRM et al, which all call for system interaction and integration. System heterogeneity among quantities of enterprise systems is caused by the difference of developers, application platform, development platform, ,communication protocol and data format used for external exchange etc. Integration of heterogeneous data source is the typical problem in database field. Although at present, there are lots of data integration methods and corresponding tools used for the business cooperation application in the environment of supply chain, the restriction exist from many aspects such as the complexity of enterprise information system, the diversity of heterogeneous data source. And especially the new application demands of enterprises lead to the complexity of enterprise data integration process.

Meanwhile the high demands of information exchange in supply chain environment, and mass information or knowledge spread out in various formats among different enterprise systems, which lead to semantic heterogeneity issues between the existing enterprises information systems. Semantic heterogeneity is a essential problem existing in any application involved two or more groups. The heterogeneity of these systems and the disunity of knowledge expression methods are becoming barriers for the knowledge acquisition by stakeholders. While these collaborations are being adopted, there would be excessive knowledge accumulated, unorganized and decentralized, which could lead to a considerably low efficiency and inconsistency in their treatments. All these issues will definitely affect everyone's comprehension when enterprises collaborate in the context of supply chain.

The system heterogeneity and information semantic heterogeneity obstruct the automation of information exchange among enterprises in supply chain, increase the cost of enterprise cooperation and block the development and innovation of supply chain. In the environment of global supply chain, one of the difficult points faced by enterprises is the lack of interoperability between software and information system [Farinha, 2007] [Jardim-Goncalves, 2006a] [Panetto, 2006a].

1.2.2 The state-of-art of theroy for supply chain integration

Manufacture supply chain is a complex network composed by manufacturing enterprises, logistics enterprises, customers and other entities in manufacture field, while the logistics flow, workflow, fiance flow and information flow are all exist in the network. These entities include some management companies, manufacturing enterprises, transshipment warehoouse, suppliers of raw materials and components, logistics companies, distribution center, retailers and end users. Integrated supply chain is the hot research topic both overseas and domestic all the time. Stanley et al. considered that building a completed integrated supply chain system, and dividing it into different types to serve for different market, which could improve the survival and competitive capability of enterprise [Stanley et al., 2002]. Supply chain integration brings positive effects to enterprises for enacting and adopting correct competition policy, which plays an irreplaceable role in enhancing enterprise working performance and improving achievement [Narasimhan, 2002].

There have various views about supply chain integration. Ming-gang Sui et al considered that integrated supply chain referred to a "virtual organization" composed of members with a common goal in supply chain, optimizing the organization target by information sharing and the coordination and cooperation of finance and resource among the members of the organization [Ming-gang Sui, 2002]. Prabir et al considered that supply chain integration is that the circumstance that in the certain environment of sociality, economy and technology, the market, resource, information and organization form related to enterprises are recognized and selected effectively by enterprises to achieve management integration, to make the supply chain networked, smart, flexible, individual and optimized [Prabir, 2005b]. Togar et al. emphasized the importance of supply chain integration and proposed four definition models: logistics synchronization, information sharing, incentive consistent and collective learning, which are interrelated, interdependent and interactive, and form a supply chain network system [Togar, 2004]. Barua et al considered that supply chain integration referred to the circumstance that based on the integration of data information and business process inside enterprise, the scope of integration is expanded to enterprise external to link information systems inside enterprise with systems from business partner. Each member in supply chain can coordinate its own business operation according to the correct information of whole supply chain, in order to realize the enterprise integrating including customer service and support, plan and forecast, product development, production manufacturing, purchase and human resource et al [Barua, 2011].

With the change of development pattern of manufacturing industriy, the appearing technology of integrated manufacturing, green manufacturing, agile manufacturing, intelligent manufacture, networked manufacturing and mass customization manufacturing enhance the outstanding effect of supply chain integration and management on aspects of resource management, logistic transportation, organization structure and performance measurement [Guo-ning Qi, 2001] [Guo-ning Qi, 2004] [Xin-jian Gu, 2001] [Xin-jian Gu, 2002]. Xin-jian Gu et al disused on the development strategy of Chinese enterprise cluster integrated manufacturing based on the comparison of various integrated manufacturing patterns of enterprise cluster such as supply chain integrated manufacturing system and strategic alliance integrated manufacturing system et al. And they pointed out the significance of supply chain integration and detailed summarized on the function structure and specific application of manufacturing supply chain integration system

[Xin-jian Gu, 2003].

About the supply chain integration models under various manufacturing modes, Yun-bo Zhang et al proposed the supply chain flexible system integration model for the supply chain in manufacturing. According to supply and demand principle, use systematic analytical method to analyze the flexible generation and development motive of supply chain in its life cycle, and the dynamic characteristics of supply chain flexibility. And then six flexible sub system models were built, including research, resource, manufacturing, logistics, information and decision, while the supply chain flexible system integration model was built based on the inner relationships among each sub systems [Yun-bo Zhang, 2004]. Yong-jun Sun et al fully borrowed ideals from available CIM modeling method, aiming at the characters of agile supply chain, analyzed about the essence of agile supply chain, proposed the integration modeling method on agile supply chain and general model of agile supply chain, which was based on hierarchical control thought and multiple views modeling theory. The multiple views model of agile supply chain was built to analyze the characteristics of each view model and the relationships among them.

Fundamentally speaking, the views mentioned above are all based on information integration. The information integration of supply chain was divided into three hierarchies in [Bu-tong Zhang, 2003] [Ting-bin Chen, 2005]: data level, information level and knowledge and decision level, while the integration details and integration process were illustrated in Table 1-1:

Table 1-1 Classification and content of supply chain information integration

Integration hierarchy	Integration process	Integration details
Data level	Inner data integration of bottom	The seamless integration of information flow is realized inside core enterprises of supply chain. The key points of integration are to settle down to the data acquiring, data flow and data consistency. Distributed and multilayer structures are adopted to realize the seamless integration.
Information level	Enterprise system integration	The concordance between enterprise heterogeneous systems and data sources is realized inside enterprise. The information exchange between new and old systems is realized by adopting asynchronous information mode inside enterprise, to realize the integration with Legacy system, which is the main component of traditional EAI (Enterprise Application Integration).
	System concordance of supply chain enterprise	The information exchange and transaction procedure automation among enterprises is realized based on Web service, XML-RPC and other ways. The synchronization and coordination of upstream and downstream enterprises are realized by adopting service-oriented integration, and realize the workflow of supply chain, in order to achieve the dynamic alliance of supply chain enterprises: B2Bi(Business to Business Integration).
knowledge and decision level	Business process management	Take the operational programs inside enterprise to coordinate with upstream and downstream enterprises and realize the customer-driven dynamic integration system. Make B2C, B2B, CRM, ERP and SCM integrated together and improve the response ability and flexible management of enterprise, to realize real-time and intelligent management for whole supply chain: BPM (Business Process Management).
	Knowledge discovery and decision support	Realize the sharing and management in knowledge level by integration of multi data sources, to support the whole decision and optimization of supply chain.

1.2.3 The state-of-art of technique for supply chain integraion

The research of supply chain integration method around the world are mainly about technology support and data representation at present. The relatively main methods include dynamic integration technology of loosely coupling modularization, supply chain intelligent integration method based on Agent and supply chain integration method based on knowledge representation.

The dynamic integration technology of loosely coupling modularization includes some workflow management technology such as Internet/Intranet technology, Web technology, electronic data interchange (EDI) and CIMS technology, distributed components and XML centric information technology et al. As to dynamic integration, the dynamic integration technology of loosely coupling modularization realizes the multilevel enterprise structure of inner enterprise mainly by modularization technology, and solves the problems of data flow integration inside supply chain enterprise, which provides an excellent solution for supply chain information integration [Bu-tong Zhang, 2003]. As to the integration of loosely coupling modularization supply chain system, a typical application was proposed by Sahuguet et al to realize the enterprise solution of complex problems by the combination of J2EE multi-layer structure and XML and Web Services [Sahuguet, 2001]. The enterprise application integration (EAI) based on internet or intranet is effectively supported by the combination of these technologies. The J2EE provides the support for Web service and realizes the dynamic integration based on XML among supply chain enterprises, which effectively supports Business to Business Integration (B2Bi). Thomas proposed using XML and relevant technologies to realize the message-oriented asynchronous mode, to solve the problems on EAI and realizing Service-Oriented Integration (SOI) based on Web service, which then solved the problems on B2Bi [Thomas, 2004]. After analyzing the research status of manufacturing industry supply chain integration systems, Chen et al. proposed a kind of expandable platform based on Service-Oriented Architecture (SOA) for the collaborative integration manufacturing of enterprises, to realize the information integration among manufacturing enterprises, which mainly through building the basic system structure of electronic business and supply chain enterprises application by Web Services to provide the dynamic application integration inside and outside enterprises based on service [Chen, 2006]. On the analysis the status of enterprise supply chain and the evolution of electronic pivot systems, Zeng et al proposed a way based on B2B electronic pivot systems to realize enterprise information integration, which makes the seamless merging between electronic pivot systems and mature information technology such as ERP and CRM, and then removes the difference and limit among enterprises and supply chain members during supply chain process to realize information integration [Zeng, 2003].

Supply chain intelligent integration method based on Agent is to simulate, optimize, implement and control the integrated working process of supply chain by using intelligent agents which have abilities of reasoning independently and deciding independently, and multi-agent systems composed. In 1993, Fox et al built the research project named by "Integrated Supply Chain Management", which image that managing the supply chain inside enterprise by intelligent and independent objects instead of people to realize the integration on the knowledge and decision of supply chain process, and build the corresponding multi-agent system framework model [Fox, 1993]. Nissen studied the supply chain integration based on Agent, proposed a suit of supply chain intelligent agent system to express and conduct the commercial activities among product users, buyers and suppliers automatically, and made comparison between this method and integration methods based on EDI and Web Service respectively, drawing a conclusion about the application necessity of different methods under different enterprise supply chain environment [Nissen, 2002]. On the base of analyzing the effective ways to deal with challenges under complex and changeful global market environment for manufacturing industry, Zhang et al discussed the defects of existing technology systems such as MES, SCM and ERP et al in responding to these challenges, and proposed combining the multi-agent systems with agile manufacturing systems and electronic manufacturing systems and building the physical model and simulation framework based on Agent to realize the effective integration for enterprise supply chain process and supply chain network was the best approach to deal with these problems [Zhang, 2006]. Wang et al studied the effective integration of enterprise supply chain under the environment of electronic business, proposing an approach based on agent intermediary to realize the supply chain integration of electronic business, which could coordinate and process the information and data flow effectively during the supply chain service process and could be combined to the business activities under the condition adding to various constraints, and then could respond to the changeful and uncertain operation environment of enterprise activity process effectively [Wang, 2008]. As to the complex and changeful environment of business process for each link of enterprise supply chain, virtual market was introduced by Kaihara to propose a kind of multi-agent system based on supply chain, which could be used to solve product configuration problems, build discrete resource configuration mathematical model and match corresponding algorithm under indeterminacy condition, and then realize the complex and accurate supply chain integration management [Kaihara, 2003].

Supply chain integration method based on knowledge representation is mainly embodied in using knowledge transforming technologies such as XML (Extensive Markup Language) to study the knowledge integration of Supply chain heterogeneous systems and realize the visit to enterprise information. XML is the markup language used to mark electronic documents and make them own constitutive property, which could be used to mark data and define data type. Meanwhile, XML is a kind of source language allowing users to define their own markup expression, providing unified approach to describe and exchange structured data independent in application softwares or suppliers, which is very fit for web transmission and information integration. Nurmilaakso et al studied the supply chain information integration under electronic business by using main XML technologies, constructed mathematical model, and discussed the application of information integration system based on XML in practical enterprise supply chain in detail, and how XML technologies could and in which conditions could realize the concordance for the business process of all partners in supply chain such as source, make, sale and finance et al, while conducting standardized archiving for the use of XML technologies in supply chain process and discussing the defects of present supply chain information integration application based on XML [Nurmilaakso, 2002a] [Nurmilaakso, 2004b]. Prabhir et al researched the integration of information technology and organizational management in enterprise supply chain process, pointed out the significance of knowledge transformation and data interaction in integration based on the analysis of the faced problems for some aspects of logistic process such as deliver, source, stock control, assignment management and customer service et al, and discussed the concrete application of supply chain process management and communication technologies in each link of supply chain, which include XML, RDF and EDI (Electronic Data Interchange) et al [Prabhir, 2003a].

The integration technologies of supply chain systems mentioned above could realize the supply chain integration to a certain extent and , while for solving the problems involved in supply chain system integration in wider scope or more effectively, it's necessary to concern about system integration on semantic.

1.3 Ontology and semantic interoparability

1.3.1 Ontology general question

Ontology is a subject originated from philosophy and formal logic. The concept of ontology is originated from philosophy field, being as a research branch of philosophy for a long time. Ontology is defined as "systematic description to the objective existence in real world, exactly the ontologie" by Aristotle, the ancient Greek philosopher, which means that ontology is the systematic explanation or statement of objective existence, concerning about abstract essence of objective reality.

In the 1980s, ontology was introduced in artificial intelligence field to describe the concepts in real world and given new definition. Thereinto, the most famous definition is from Gruber that an ontology is an explicit and formal specification of a shared conceptualization [Gruber, 1993a]. Ontology could formally express knowledge as a set of concepts and their relationships in certain area, to keep its the semantic. In detail, ontology provides lexical concepts in object domain and hierarchy classification of concepts for compute readable and processiable. Ontology not only defines the class, relationship and classification, but also uses formal ontology representation language to express them. It's a kind of structured language, which can support various consistency check and interoperability between different applications. Therefore, ontology is a promising method to describe knowledge of different types and forms through unifying the metamodels of knowledge representation.

According to the difference of application fields, ontologies of different types should be defined to meet applications' demands. Metadata ontology is used as knowledge expression language, such as Dublin Core [Weibel, 1995]. Upper ontology is used to provide the ontologies about common sense in objective world. Domain ontology is about ontologies in specific subject areas. Problem and method ontology is used to provide the terms involved in specific tasks or the concepts and relationships used for the problem reasoning of specific tasks. Application ontology contains the ontologies used for specific application knowledge modeling. Thereinto, domain ontology is the most widely defined and used in engineering applications.

In early artificial intelligence field, ontology representation language is based on first-order logic or description logic, such as KIF (Knowledge Interchange Format) [Genesereth, 1992], Ontolingua [Farquhar, 1997] and Loom [MacGregor, 1991]. At present the widely used ontology representation language is the semantic web ontology language established by W3C[1] group, including RDF, RDF Schema, OIL, DAML+OIL and OWL et al. Thereinto, OWL is the most widely used ontology representation language presently. The famous softwares and tools used for ontology development include Protégé[2], DUET (DAML UML Enhanced Tool) [DUET, 2008], XP [deVos, 2001a] [deVos, 2001b] and VOM (Visual Ontology Modeler)[3] et al.

The application fields of ontology are mainly involved in the following aspects:

(1) Knowledge management. A unified framework or a standard model is built by ontology to reduce the diversity of concepts and terms. Ontology can be reused to avoid the repeated analysis of knowledge domain. For different people have different knowledge structures especially on the multi-discipline knowledge management,

---

1 http://www.w3.org/
2 http://protege.stanford.edu/
3 http://www.sandsoft.com/products.html

ontology can provide a unified knowledge framework as a common and shared knowledge reference. In the application of intelligent information retrieval, ontology is usually acting as a domain model about customer interested domain. Besides, it also used as the knowledge representation language system and standard for annotating documents uniformly. Except for information retrieval, ontology can support the filtration and maintenance of knowledge and the generation of automation documents.

(2) Support the interoperability among different systems. Ontology can realize interoperability and integration among different systems by translating and mapping among different modeling methods, paradigms, languages, softwares and tools. The information from variously different and dispersive sources can be integrated by ontology. When the distributed applications need knowledge from different applications, forms and granularity, ontology can be used to support the information integration and automatic acquisition process. Ontology is the sharing and clear expression, so it can be used in the annotation of multiple data resources, not only in the form of webpage, but also XML document and relationship database et al.

(3) Support networked electronic business. In the Web based electronic business, the ontology exists as the tool providing knowledge sharing and reusing on knowledge layer and the support for semantic search. Building standard ontology in business fields can provide the transformation service of different data structures based on ontology and support the automatic and electronic business data transmission.

1.3.2 Ontology development method

The ontology development method contains a set of rules, processes, practices, approaches and activities of ontology design, construction, evaluation and application. Presently, the most of ontology development methods are about ontology construction, while few are about ontology reuse, maintenance and evolution. Gruber proposed the five principles of ontology construction: clarity, coherence, extendibility, minimal encoding bias and minimal ontological commitment, which have been the influential theory about ontology construction so far [Gruber 1995].

No any ontology development method is the best or most correct. Ontology development must be a repeated circular process, and calls for a strictly standard flow. However, on one hand, ontology modeling engineering has still stayed in a relatively immature stage; on the other hand, for the difference of fields, the scope and depth of ontology application in engineering are also different. Therefore many ontology modeling projects have their own unique methods. The common used ontology development methods include IDEF5, Skeletal Methodology, TOVE, Seven-step Method developed by Stanford University and Methontology Framework et al. Among them, Methontology Framework standardizes the whole life cycle of ontology from ontology definition, which not only supports the ontology construction starting from scratch, but also supports reusing other ontologies [Fernández-López, 1999].

There are some ontology development methods having great differences from typical ontology construction methods mentioned above. Devedžić proposed a method adopted object-oriented software analysis and design philosophy [Devedžić, 2002]. Its basic principle is that ontology should express the concept, property of concept and the value of property, meanwhile expressing the event and the reason, influence, process and time of the event. Ontology has the hierarchical relationship, and most of ontologies support the relationships such as generalization, inheritance, aggregation and instance. Therefore ontology construction begins with assembling domain vocabularies and adopts object-oriented analyzing method, and then generates domain ontology related to application systems. Paslaru et al proposed a way to develop ontology by reusing knowledge resource, which firstly expressed the concepts coming from outside knowledge resources, defined universal term vocabulary and concepts, and then selected corresponding relationships [Paslaru, 2005]. This approach could improve the efficiency of ontology development and reduce the cost of development. Web 2.0 technology raised in recent years was introduced into ontology development by O'Reilly, which adopted Folksonomy as the foundation of ontology development to reduce the time of development [O'Reilly, 2005]. Folksonomy comes from the tags by cooperative construction, which is the relatively fuzzy information classification for ontology. Although folksonomy can provide a kind of ecosystem to change the classification results dynamically, it can't be used effectively in the knowledge classification and sharing. Because it lacks of solid structure, only a set of terms [Gruber, 2005c].

Object-oriented software design philosophy is applied to ontology development by the model-driven modeling and developing of MDA. Introducing UML into ontology engineering was firstly proposed by Cranefield, which has become the knowledge representation and exchange approach to bridge software engineering and ontology engineering [Cranefield, 2001a]. Cranefield defined the ontology development method based on UML and used OCL to express ontology constraint. For UML lacked normative definition, Cranefield then proposed the method to integrate UML and RDF(S), which was based on a series of standards such as XML, RDF(S), XSLTs and XMI et al [Cranefield, 2001b]. Baclawski et al also proposed two ontology development methods based on UML [Baclawski, 2002a] [Baclawski, 2002b]. The first one defined the mapping rules of UML and DAML+OIL in concept level, and expanded UML metamodel based on the corresponding concepts of DAML+OIL. For the first method had introduced some new concepts into UML metamodel, then they developed an independent ontology metamodel UOF (Unified Ontology Language) based on MOF, which was an inspiration of current ODM proposal. Falkovych et al proposed the mapping method from UML to DAML+OIL by XSTL [Falkovych, 2003]. The problems involved in network ontology development and evolution life cycle in distributed application were researched in the NeOn[4] project of the 6th framework of European Union. It developed a series of tools and relevant methodologies and trying to provide a kind of economically feasible method to manage the lifecycle of new generation semantic application, especially developed some ontology metamodels relevant to OWL ontologies, rules, mapping and module ontologies [Haase, 2007].

Although the research on ontology development has lasted for a very long time

[4] http://www.neon-project.org

and various ontology development methods have been explored and practiced. For the ontology engineering still in the developing stage without mature application, there are quantities of problems existing in building and development process of domain ontology.

(1) Construction process is not normative.

Currently, the common construction methods are all summarized from the development process of specific domain ontologies, with limited application field. Most of them have few correlation techniques, and rough detail, while there is no method completely according to life cycle development method. In addition, for no unified construction principle existing as a guide, it's hard for the whole construction process of ontology to be organized normatively.

(2) Demand analysis is insufficient and construction process is planless.

The concrete demand of ontology development, especially the ontology development of a specific field, could not be described clearly, which will directly lead to the lack of plan and control for the whole ontology construction process.

(3) There is no unified evaluation criterion.

There is no unified evaluation criterion about ontology result, also any standard test set, which make it hard to evaluate the construction result of ontology reasonably and will bring negative effect to ontology application and further expansion.

(4) The method of ontology further expansion is absent.

With the continuous developing and changing of the domain, there must be more and more domain concepts and relationships to be introduced into core ontology. It will expand the existing ontology and provide stronger semantic representation ability. However, the maintenance and expansion of existing ontology have been not yet researched and supported.

(5) The sharing and reusing of ontology cannot be full ensured.

The purpose of domain ontology is to provide the semantic basis for different systems when communicating with each other. Meanwhile, ontology construction process is also the robotic accumulation process of human knowledge. Therefore, sharing and reusing are the essential requirements of ontology. How to ensure that during the construction process of domain ontology is very important.

1.3.3 Semantic matching

The research field of ontology is mainly involved in three aspects: agent-based software interoperation, knowledge acquiring and nature language processing. Among them, semantic matching is based on above-mentioned aspects and is the important means to employ ontology. Ontology matching methods can be divided into four types according to input data process: terminological technology, structural technology, extensional technology and semantic technology [Euzenat, 2007] [Giunchiglia, 2004].

(1) Terminological technology is a name based method, and seeks the similarity by comparing the character string of input elements such as tags, concept names and annotation et al. According to the difference of data process, it can be subdivided into string-based method and linguistic-based method. String-based method adopts some approaches such as string standardization, edit-distance and string comparison et al to estimate similarity. This method is suitable for the comparison of very similar words. Thereinto, a series of means of string standardization such as switching between uppercase and lowercase, abbreviation and removing space et al, are also the data pretreatment methods. The means adopted by linguistic-based method is nature language processing, which need recognize the grammatical structure and sequence of terms, usually with the help of external resources such as dictionary. It is situtable for processing the phrase similarity.

(2) Structure-based technology can be applied to calculate the similarity between ontologies, including the method on comparing internal structure and the relational structure. Internal structure contains concept names, annotation, property, property domain, data type of property and mulitiplicities et al. Internal structure provides a basis which algorithms can rely. Comparing one type of structure singly can’t reflect the similarity between ontologies, so this method is usually not used alone. Relational structure contains the relationship between class and subclass, hyponymy and

property relationship et al. The most universal structure relationship is the taxonomic relationship. The method on comparing relationship structure can add the relationships among entities of ontology to the similarity comparison, which is usually used together with terminological method.

(3) Expansion method is usually used in the situation that the ontologies share set of individuals, which seek the relationships among concepts by comparing individuals. Statistics and individuals similarity can be used here..

(4) The characteristic of semantic-based technology is that model-theoretic semantics is used to justify their results, which is a deductive method. It cannot be used alone, and usually acting as a complementary approach to ensure the completeness of mapping. Some common means contain: adopting external data source such as some upper ontologies including DOLCE, Cyc and SUMO et al [Lenat, 1990] [Niles, 2001] [Gangemi, 2003]; propositional method and description logic technology et al. There are a few such technology developed and used at present.

Based on the analysis of ontology matching technology from articles such as [Euzenat, 2007b] [Bruijn, 2003] [Kalfoglou, 2003b], the existing ontology matching methods is classified according to the criteria of above-mentioned four matching technologies and the matching automaticity, which is shown in Table 1-2.

Table 1-2 The classification of ontology matching technology

	Matching technology

		Terminologyical technology	Structural technology	Extensional technology	Semantic technology
Matching automaticity	Automatic matching	MOMIS[Bergamaschi, 2001] IF-Map[Kalfoglou, 2002] QOM[Ehrig, 2004] S-Match[Giunchiglia, 2004] Artemis[Castano, 2000] H-Match[Castano, 2006] Tess[Lerner, 2000] Cupid[Madhavan, 2001] Falcon-AO [Jian, 2005] MoA [Kim, 2005] XClust [Lee, 2002] ASCO [Bach, 2004] BayesOWL [Pan, 2005]	MOMIS[Bergamaschi, 2001] IF-Map[Kalfoglou, 2002] QOM[Ehrig, 2004] Artemis[Castano, 2000] H-Match[Castano, 2006] Tess[Lerner, 2000] Cupid[Madhavan, 2001] Falcon-AO [Jian, 2005] XClust [Lee, 2002] ASCO [Bach, 2004] BayesOWL [Pan, 2005] OMEN [Mitra, 2005] DCM [Chang, 2005]	QOM[Ehrig, 2004] S-Match[Giunchiglia, 2004] Artemis[Castano, 2000] H-Match[Castano, 2006] Cupid[Madhavan, 2001] Falcon-AO [Jian, 2005] MoA [Kim, 2005] XClust [Lee, 2002] ASCO [Bach, 2004] BayesOWL [Pan, 2005] OMEN [Mitra, 2005]	OntoMap[Kiryakov, 2001] S-Match[Giunchiglia, 2004]
	Semi-automatioc matching	MAFRA[Maedche, 2000] PROMPT[Natalya, 2003] GLUE [Doan, 2002] CAIMAN[Lacher, 2001] Hovy[Hovy, 1998] TranScm[Milo, 1998] DIKE[Palopoli, 2003] SKAT[Mitra, 1999] Clio [Miller, 2000]	MAFRA[Maedche, 2000] PROMPT [Natalya, 2003] OntoMorph[Chalupksy, 2000] Hovy [Hovy, 1998] TranScm [Milo, 1998] DIKE [Palopoli, 2003] SKAT [Mitra, 1999] Clio [Miller, 2000]	TranScm[Milo, 1998] DIKE[Palopoli, 2003] SKAT[Mitra, 1999]	ONION[Mitra, 2002] OntoMorph[Chalupksy, 2000]
	Manual	DELTA[Clifton, 1997] CtxMatch[Bouquet, 2003] OntoBuilder[Modica, 2001] COMA[Do, 2002]	FCA-Merge[Stumme, 2001] COMA[Do, 2002] ToMAS[Velegrakis, 2004] MapOnto[An, 2006]	RDFT[Omelayenko, 2001] OntoMerge[Dou, 2005] CtxMatch[Bouquet, 2003] ToMAS[Velegrakis, 2004] OntoBuilder[Modica, 2001] COMA[Do, 2002] MapOnto [An, 2006]	OBSERVER[Mena, 1998] CtxMatch[Bouquet, 2003]

1.3.4 Semantic interoperability

Semantic interoperability refers to that computer can understand the knowledge representation of various domains by methods referring to and mapping of knowledge system (representation for concept, constraint, relationship and axiom), to make information systems have the semantic interaction ability. As a kind of high-level interoperability idea and problems urgent for user demand, semantic interoperability is one of research hotspots currently. Realizing semantic interoperability can make the information between human, between machine and man and even between machine and machine to communicate with each other accessibly, and also can make machine understand the meaning of information. It would realize the high automation and intelligence of information exchange and sharing, improve and promote the exchange mode and quality of information thoroughly. Many scholars have done researches on semantic interoperability, mainly on structured and formal expression on knowledge.

Paolo et al bridged the gap between HTML based internet and RDF based semantic web by linking the words of original text to concepts of ontologies, and developed fully automated methods for mapping equivalent concepts of imported RDF ontologies (for this prototype WordNet, SUMO and OpenCyc). These methods will thus allow the seamless integration of domain specific ontologies for concept based information retrieval in different domains [Paolo, 2003]. Jacob et al proposed the knowledge-based approach to solve heterogonous problem, using fusion rule to manage the semi-structured information that is input for merging. The integrated usage of fusion rules with a knowledgebase offers a practical and valuable technology for merging conflicting information [Jacob, 2006]. Anthony firstly used a series of analysis techniques to distinguish similar model elements such as relationship and attribute. And then the statistical analysis technique was applied to a preliminary integrated data set to estimate the relationship among model elements more accurately, using repetitive process to realize the integration of heterogeneous data sources [Anthony, 2006]. Zhao et al presented algorithms to resolve schematic discrepancies by transforming metadata into the attribute values of entity types, keeping the information and constraints of original schemas. Although focusing on the resolution of schematic discrepancies, the technique works seamlessly with the existing techniques resolving other semantic heterogeneities in schema integration [Zhao, 2006].

Meanwhile, many researches introduced ontology into semantic interoperability. Qi proposed a semantic-driven integration method named a priori approach. The innovation was that each data source participated in integration process contains an ontology, and this method could ensure all data sources quote the sharing ontology, to realize the automation of integration process. Qi proposed two integration algorithms: one is sharing ontology, which would be expanded during the integration process; the other is the instance of ontology data source built on the sharing ontology. At the end, Qi pointed out that this method could integrate automatic electronic classification and enterprise engineering data by using PLIB ontology model [Qi, 2006]. Bellatreche proposed a kind of formalized ontology acting as a relevant role in interoperation framework of information system. Class and attribute were used by this ontology to describe the communication behavior among agents. Bellatreche also proposed communication behavior ontology provided interoperability support, by recognizing the communication behavior of a kind of agent communication language for the communication behavior instance of another communication language. This ontology used OWL to describe attribute and class [Bellatreche, 2006]. Bermúdez expounded the demand of enterprise business process and built the research foundation, proposing a concept model. And then, the current status and future trend of business process modeling and process interoperability were reviewed based on this model. At last, Bermúdez proposed a kind of innovative semantic web technology and a agent-based framework to facilitate the business process cooperation [Bermúdez, 2007]. Ruinan proposed the three-step method to enhance the interoperability of heterogeneous semantic resources. Firstly, referring to the original representation language for semantic representation, adopted OWL DL of fulcrum form to construct the heterogeneous representation. Secondly, mappings were built among these standard resource concepts, and then stored in an ontology named clarity. Thirdly, an approach to rank these mappings was needed to satisfy user’s demand, which was developing in Semantic Resources “Interoperabilisation” and Linking System (SRILS) [Ruinan, 2006]. Silva et al studied the structure of semantic XPath processor, which defined the mapping from XML to RDF and allowed the interoperability on semantic layer for XML documents. Model mapping method was used in RDF to express XML instance and XML mode. Opposite to the common structural mapping approaches, this expression could keep the structural order. And this processor could generate feedback for unlimited XML model and RDFS/OWL ontology [Silva, 2006].

1.4 Conclusion of literature review

Based on above-mentioned summary of research status about supply chain integration theory and related technology, there still exist some problems in current research:

(1) There are many theoretical researches on supply chain integration framework, but the not on semantic level, and there is no actually used system for semantic interoperability of heterogeneous information systems.

(2) There are many researches on ontology development method, but most of them are from bottom layer, with big construction workload and high difficulty. Meanwhile, the reuse degree of ontology is low, and there is not any good ontology putting into use inside the country.

(3) The research on ontology mapping is the foundation to realize semantic interoperability. But looking at the current research status, one heterogeneous problem is solved by one specific method. And as to the semantic interoperability of heterogeneous information systems in complex supply chain environment, there is not any method that adopts different mapping methods to solve semantic mapping according to different knowledge structures, knowledge objects and heterogeneous systems.

(4) The ontology covers all aspects of supply chain environment is rarely appeared.

1.5 Research contents and meanings

1.5.1 Research foundation and meanings

The dissertation is supported by supported by Chinese Science and Technology Support Plan Project (2006AA04Z157) “Research and application on ontology-based business cooperation oriented heterogeneous systems integration technology”, Chinese National Natural Science Foundation Project (61175125) “Semantic interoperability of heterogeneous systems in supply chain based on SCOR”, and Zhejiang Provincial Natural Science Foundation Project (Y107360) “Research and application on business cooperation oriented semantic interoperability mechanism”. Meanwhile this work combines with the research results from the research group of professor Panetto in CRAN at Nancy in French. and researches on the interoperability of enterprise information systems in the environment of supply chain.

The research group of professor Panetto in CRAN has the long-term research accumulation about product-driven systems and system interoperability. Product is set as the centre of enterprise system integration, and Product-Driven Paradigm is proposed for tracing the information of product life cycle, in order to promote the integration of product manufacturing and business process. Morel proposed applying SoS (System-of-Systems) paradigm to the research on enterprise integration and interoperability, which regards enterprise systems as components of system and studies the principle of enterprise system interoperability and guides the engineering application from SoS paradigm [Morel, 2007]. Under SoS paradigm, based on some international standards such as IEC 62264 and ISO 10303 STEP-PDM, Turis constructed the product ontology model ONTO-PDM to realize the product-driven interoperability of enterprise manufacturing systems [Tursi, 2009a].

Although there are quantities of researches on supply chain system integration, however realizing the semantic integration of networked enterprise information systems in supply chain environment is always a difficulty, and meanwhile there is not any effective integration project on semantic level applied to practice. Based on above foundation, the information system interoperability mechanism and related techniques of networked enterprises in the environment of manufacturing enterprise supply chain are studied in this dissertation. Adopting ontology and standard based semantic interoperability method to deal with the complex business cooperation and heterogeneous systems in supply chain, which is an effective approach to realize the semantic integration of networked enterprise information systems. The purpose of the dissertation is to solve the interaction of heterogeneous information systems of networked enterprises, improve the correctness and effectiveness of information interaction and send the correct feedbacks to enterprise decision level, which would not only improve the production value of single enterprise, but also optimize the whole supply chain network. Interoperation is used to deal with the information interaction among various enterprises in supply chain environment, and when a new system enters into the enterprise system network, the original enterprise system can work side by side with the new system, which would reduce the cost to introduce new technology. Adopting ontology to formalize and standardize production information and supply chain model can realize the supply chain information model combining product and process, and enhance the interoperability of enterprise information system to semantic level. The sharing ontology can concentrate the explicit knowledge resources from various media such as text, diagram, article, patent, standard, application system, source code, product data model and product drawing et al, and systems, together with the tacit knowledge from human brain to realize knowledge ordering and then improve the efficiency of information interaction, knowledge sharing and knowledge reusing among enterprises.

1.5.2 Main content and structure

In this dissertation, the interoperability method of networked enterprise information systems in the environment of supply chain was studied, from the point of theoretical modeling and application. The main research contents are as follows:

(1) Analyze the difference between enterprise information system integration and

interoperability in supply chain environment. Starting from the demand of information system interoperability, we propose the framework of supply chain enterprise information system interoperability.

(2) Based on the product ontology model ONTO-PDM developed by the CRAN (the laboratory of Nancy First University in France) and some international standards related to enterprise system integration such as IEC 62264 and ISO 10303 STEP-PDM et al, takeing product as the core of enterprise value chain in the environment of supply chain, we concentrate the related business information and technology data of about product life cycle, such as product design, manufacturing and deliver et al., to a commonly used product ontology.

(3) Research supply chain operation reference model SCOR, and take it as the foundation to build a supply chain ontology including definition of the whole supply chain process, basic elements involved in the process and performance metrics of supply chain.

(4) Adopt the method combining ontology mapping based on WordNet and the reasoning method based on the rule to realize ontology merging of product ontology and supply chain process ontology, in order to build the product-centric supply chain ontology. Realize the interoperability of supply chain enterprise information system from two aspects: the mapping among ontology bases and the mapping between data base and ontology base.

The structure of dissertation is arranged as Figure 1.1, and the contents are as follows:

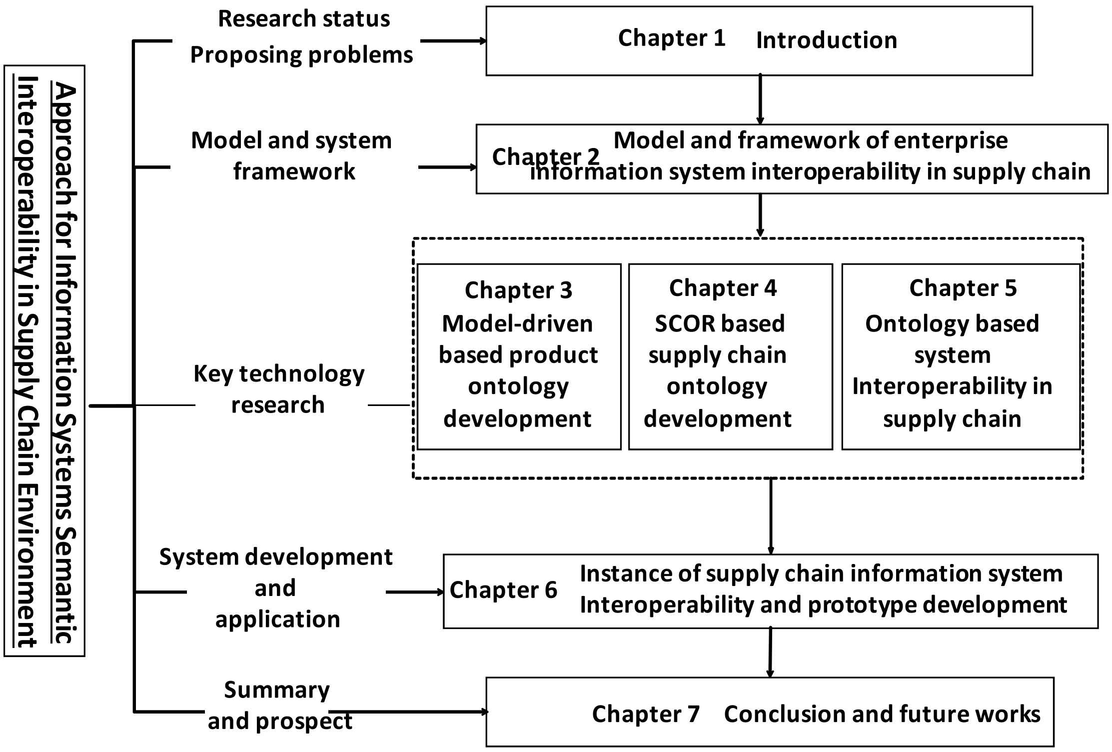


Figure 1.1 The structure of dissertation

Chapter 1 stated the research background of the dissertation, analyzed the domestic and overseas research status of supply chain integration, and proposed semantic interoperability is an effective solution to deal with the integration of enterprise information systems in supply chain environment. Meanwhile analyzed and summarized the research status of correlated techniques involved in semantic interoperation. Finally, the structure of dissertation was provided, stating the foundation, content and meaning of the research.

Chapter 2 analyzed the SoS (System-of-Systems) characteristic of networked enterprises in the environment of supply chain, proposed the SoNE (Systems-of-Networked Enterprises) paradigm. And then it analyzed the distinction between the interoperability and integration of enterprise information systems. Finally, proposed the information system interoperability framework of supply chain enterprises based on the demands in the environment of supply chain.

Chapter 3 studied the developing method of product ontology. Based on some international standards involved in enterprise system integration (such as IEC 62264, ISO 10303 and STEP-PDM et al.), the model-driven ontology developing method was adopted to integrate the business information and technology data related to product life cycle, including product design, manufacturing, deliver and et al, into a common shared product ontology.

Chapter 4 discussed the building method of supply chain ontology. Adopting SCOR model as the basic concepts and framework of supply chain ontology, the SCOR supply chain ontology was built by using Protégé seven-step method. Meanwhile the semantic meaning was described by OWL DL. Taking make-to-order of main machine bed as an example, an instance of SCOR supply chain ontology was given. Finally, an evaluation system of SCOR supply chain ontology was simply illustrated.

Chapter 5 researched the ontology merging and semantic interoperability methods. The method combining ontology mapping based on WordNet with reasoning based on rules was adopted to realize the merging of product ontology and supply chain ontology, to achieve product-centric supply chain ontology. The information systems interoperability of supply chain enterprises was realized by the mapping among ontology and the mapping between data bases and ontology. Finally, an instance was put forward in this chapter.

Chapter 6 took the supply chain process of numerical control guide grinding machine as an example to validate the method for information systems interoperability of supply chain enterprises. Meanwhile, according to interoperability principle, a prototype of information systems interoperability platform for supply chain enterprises was developed, providing as a feasible method for the information integration of supply chain enterprises.

Chapter 7 made a conclusion for the whole dissertation, and looked into the future works.

1.6 Conclusion

This chapter stated the research background of dissertation, analyzed the existing problems of enterprise information integration in supply chain environment and the domestic and overseas research status of supply chain integration, then proposed that semantic interoperability was an effective solution to deal with the enterprise information system integration in supply chain environment. After, it analysed and summarized about the research status of related technology involved in semantic interoperability. At last, the structure of dissertation was provided, stating the research foundation of dissertation, research content and research meanings.

2 Model and framework of enterprise information systems interoperability in supply chain

2.1 Introduction

Supply chain management is the complex and dynamic environment jointly participated by organization, human, technology, activity, information and resource et al to shift one product or service. In the past twenty years, supply chain management (SCM) has been devoting to better deploying supply chain according to different purposes, such as enterprise integration, outsourcing manufacturing and service et al. Presently, SCM not only concerns about the business process of traditional value chain, but also have an insight into the process network related to organization, which is also the form of cooperation and the relationship among partner organizations. On this occasion, the research emphasis of SCM turns from the inside enterprise to the relationships among enterprises, seeking the interoperability between operating system and applications among enterprises [Jardim-Goncalves, 2011]. The continuous change of market makes the demandings of enterprise, business, function and quality always being continuously changing, so enterprise system should have enough flexibility to adapt to this changes. The enterprises in supply chain are connected together and composed to a system network, thus the corresponding solution can be proposed by analyzing the characteristics of this system network.

2.2 System-of-Systems (SoS) Paradigm of networked enterprise in supply chain

The term System-of-Systems (SoS) is widely recognized and has become quite studied since a decade. Its application area spans from original military to other domains, especially system engineering. Researchers tried to formalise this new paradigm in the field of information system, complex system in military and enterprise since many years[Kotov, 1997][Pei, 2000][Sage, 2001][Carlock, 2001]. Further, various efforts have been made to give a common definition to specify the characteristics or principles of the paradigm. Widely cited definitions are for example Systems-of-Systems (SoS) are large-scale concurrent and distributed systems, the components of which are complex systems themselves." [Kotov, 1997][Sage, 2001]. Whichever definition is used, there are several principles that distinguish SoS from monolithic systems. The classical five principles are known as Maier's criteria: operational independence of the constituent systems, managerial independence of the constituent systems, geographical distribution of the constituent systems, evolutionary development, and emergent behavior. Based on the characteristics mentioned by Boardman [Boardman, 2006] and DeLaurentis [DeLaurentis, 2004], Auzelle [Auzelle, 2009]summarized and extracted six characteristics of SoS, as shown in Figure 2.1.

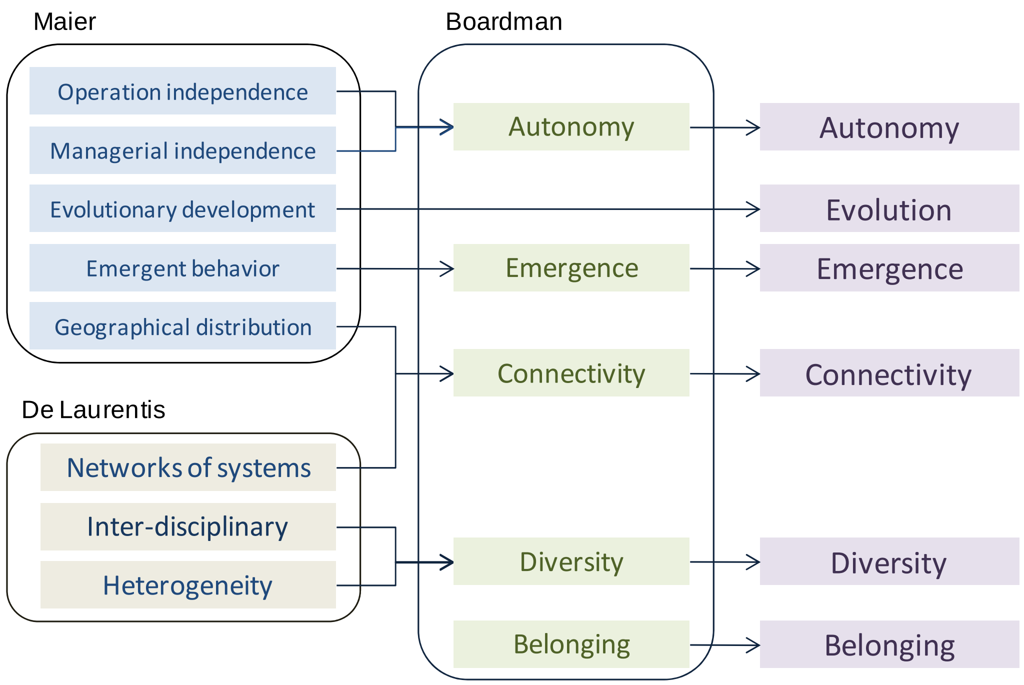


Figure 2.1 SoS characteristics

Each characteristic means respectively:

(1) Autonomy: exercised by constituent systems in order to fulfil the purpose of the SoS

(2) Evolution: The SoS adapts to fulfil its (possibly evolving) mission as a whole

as the underlying technologies evolve with time

(3) Emergence: Enhanced by deliberately not being foreseen, though its crucial importance. It creates an emergence capability climate that will support early detection and elimination of bad behaviours.

(4) Connectivity: Dynamically supplied by constituent systems with every possibility of myriad connections between constituent systems, possibly via a net-centric architecture, or by interoperability processes, to enhance SoS capability.

(5) Diversity: Increased diversity in SoS capability achieved by released autonomy, committed belonging, and open connectivity

(6) Belonging: Constituent systems choose to belong on a cost/benefits basis; also in order to cause greater fulfilment of their own purposes, and because of belief in the SoS supra purpose.

These characteristics represent the main distinguishes of fundamental components of a SoS. Thus, we could recognize a SoS by identifying whether the components are qualified with these characteristics or have capability to achieve these. A SoS is a concept at the core of research and development works to study the structure and dynamics of large scale collaboration between enterprise systems. The SoS approach does not advocate particular tools, methods, or practices; instead, it promotes a new way of thinking for solving grand challenges through the interactions of technology, business, even enterprises.

Along with the globalization process, the enterprise framework has developed gradually, and could be classified in 5 types: sub-enterprise, single enterprise, multi-sites enterprise, extended enterprise and virtual enterprise [Molina, 2007]. Table 2-1 shows an analysis of these different enterprise architecture crossed with the previously mentioned six SoS characteristics. At the sub-enterprise level and single-enterprise level, systems or applications are naturally belonging to a relatively homogeneous area, and normally systems do not have so much freedom to develop by themselves separately, they are usually bind together to execute a process for an enterprise. Meanwhile, multi-sites enterprises are generally an issue faced by large companies (e.g., Boeing, IBM, General Motors, and EADS), in integrating heterogeneous systems throughout their facilities [Panetto, 2008c]. A multi-sites enterprise has more autonomy, but its systems remain not fully independent. At a higher level, extended enterprises are loosely coupled and considered as a self-organizing network of firms that combine their economic output to provide products and services offerings to the market. Finally, virtual enterprises are a temporary alliance of enterprises that come together to share skills or core competencies and resources in order to better respond to business opportunities, and whose cooperation is supported by computer networks. So extended and virtual are not limited in one single enterprise, but span from enterprise to enterprise. They form a loosely or temporary network. Enterprises operate independently, share resources, skills, information, to achieve common goal or benefit. Related to these two kind of networked enterprises, autonomy, connectivity, and diversity SoS characteristics are obvious, while evolution and emergence characteristics appear as a result of each constituent. Based on the analysis of these SoS characteristics, we can conclude that extended and virtual enterprises fall into the paradigm of a SoS-like system, that we can call Systems-of-Networked Enterprises (SoNE).

Table. 2-1 Differentiating SoS characteristics for each kind of enterprise architecture.

Level of integration	Autonomy	Evolution	Emergence	Connectivity	Diversity	Belonging
Sub-enterprise	none	By itself	Depends on itself	Processed by sub-systems or none	None	nature
Single-enterprise	none	By itself	Depends on itself	Processed by sub-systems	None	nature
Multi-site enterprise	limited	By itself	Depends on itself	Processed by sites	None or exists among sites	nature
Extended enterprise	complete	Result of constituent enterprises	Achieved by constituent enterprises	Processed by constituent enterprises	Exists among constituent enterprises	Can Choose
Virtual	complete	Result of	Achieved	Processed by	Exists	Can

enterprise		constituent enterprises	by constituent enterprises	constituent enterprises	among constituent enterprises	Choose

Strictly speaking the enterprise network in the environment of supply chain is not a enterprise framework, but can be regarded as a kind of enterprise form existing between multi-site enterprise and extended enterprise. Enterprise can be dispersive in region, but forms a cooperation network to provide a common product or service and reach the common profit. Each enterprise or system can be considered as an independent system. And each system is connected together when calling for interaction to compose the paradigm of Systems-of-Networked Enterprises (SoNE). As the connectivity of SoNE enterprise requires the dynamic connectivity among composition systems, networked based interoperation can make sure enterprise information systems work coordinately in the premise of not changing themselves, and then enhance the system flexibility to adjust to market change and outside change, thus to strengthen the power of whole SoNE system.

2.3 System interoperability

2.3.1 Interoperability definition and classfication

The two relatively famous definitions of system interoperability are as follows:

(1) The IEEE defines interoperability as " the ability of two or more systems or components to exchange information and to use the information that has been exchanged" [IEEE, 1995].

(2) ISO standard defines manufacturing software interoperability as "the ability to share and exchange information by using unified sentence structure and semantic, and to satisfy the specific function relationship of application system by means of common interfaces" [ISO 16100, 2002].

More exactly speaking, interoperability is a kind of ability that different types of computers, networks, operating systems and applications can work together effectively without the communication, providing a practical and significant way to realize information exchange [Panetto, 2007].

Interoperability can be divided into several different levels: code, vocabulary, syntax, semantic and semiotics [Euzenat, 2001]. Code interoperability deals with the expression of char; vocabulary interoperability deals with the expression of word or symbol; syntax interoperability can deal with structural sentence, formula or axiom; semantic interoperability refers to constructing the expression about proposed meaning; semiotics interoperability refers to constructing the expression of practical meaning or the meaning in context environment. This kind of stepped classification is not very strict, but points out the evolutionary process of interoperability, while the latter interoperability can only be reached by completing the former interoperability. Thereinto, semiotics interoperability involves complex data processing process and method of artificial intelligence, so the interoperability researched in engineering field is usually stopped in semantic interoperability.

2.3.2 Difference between system integration and interoperability

Integration and interoperability are usually confused in researches. Based on the "concept and rule of enterprise model" of ISO 14528, it can be considered that models can contact with each other by three relationships: (1) when a standard or important form exists to express these models, integration is formed; (2) when a universal meta-level structure exists to build the semantic equivalence among models, unification is formed; (3) when each model exists separately, but concept mapping on ontology need to formalize semantic interoperability, federation is formed [ISO 14528, 1999].

As to the definitions of Integration, Interoperability, Compatibility and Portability, there are always some different discussions on them. Based on the definition of IEEE, integration refers to unite two components into a unified system. Interoperability refers to the ability of exchanging information and using the information by two or more systems or components. Compatibility is the ability that two or more systems or components executing the required functions in the same environment. Portability is the ease about removing system or component among different environments [IEEE 610, 1992].

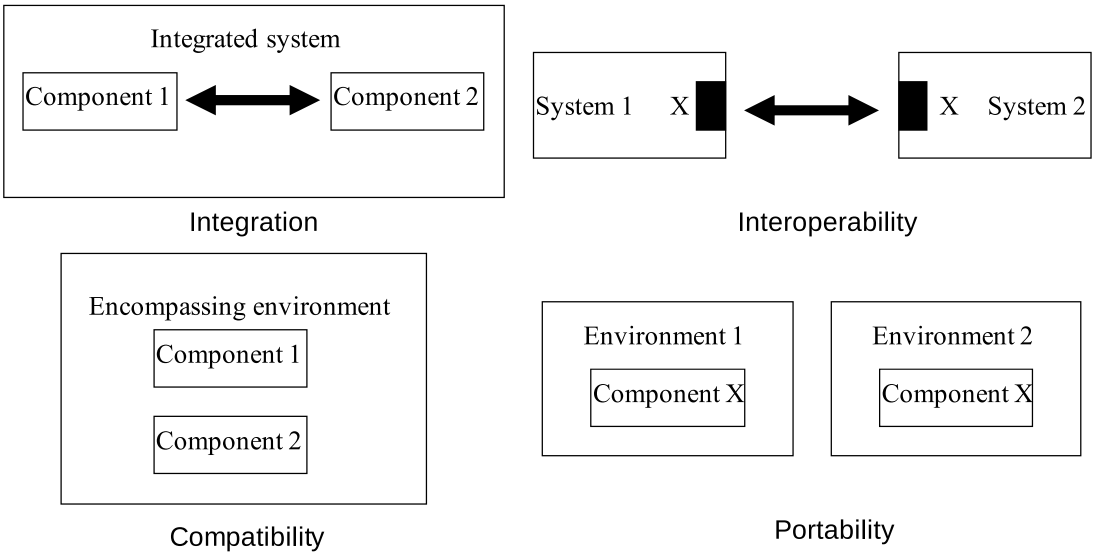


Figure 2.2 Graphical representation of Integration, Interoperability, Compatibility and Portability

Panetto considered that Interoperability was between completely Integration and Compatibility [Panetto, 2007]. Integration is generally considered to be slight beyond interoperability level, concerning about certain function support. Interoperability systems can be functional independent from each other, while once the service flow of one integrated system is interrupted, some significant functions would be lost. A series of integrated systems must be interoperable, while interoperability systems needn't be integrated. Integration can deal with the problems on organization and management by informal methods. Interoperablity is even more the problem on technology level, while integration is more difficult to be solved for involving organization and management. Compatibility is under interoperability, which refers to that system or unit has no interface to communicate with other systems or functions, not saying there is no exchange service ability. Interoperability systems must be compatible, but to the contrast it is not necessary. To display the advantage of networked by stable information exchange, systems must be above the compatibility level. Compatibility is the lowest demand, while interoperability and integration both require a series of associated systems or units driven by the basic operation of systems. Interoperability is a way to realize complete system integration [Vernadat, 2002].

2.3.3 Models of information system interoperability

The system interoperability model can describe the revolution trend of interoperability in linear dimension. Panetto analyzed several kinds of current relatively mature system interoperability models: LISI, OIM, LCIM, NATO and EIF, which are shown in Table 2-2 [Panetto, 2007].

LISI [C4ISR, 1998] divides interoperability models into five levels according to the systems involved in interoperability process: isolated systems, connected interoperability, functional interoperability, domain based interoperability and enterprise based interoperability. It mainly concerns about the technical interoperability and the complexity of interoperability between systems. The environmental and organizational problems that contribute to construction and maintenance of interoperable system are not involved in this model, such as defining interoperability requirement and maintaining interoperability across version by the sharing process.

OIM [Clark, 1999] extends LISI to the more abstract level supported by command and control. According to the organizational maturity, five levels are defined to describe the interoperation ability. The information exchange among systems is not involved in these levels.

NATO [NATO, 2003] concerns about the technical interoperability, and establishes interoperability degrees, and classifies the operation effectiveness by the structuring and automating the exchange and interpretation of data. According to the difference of degree, models are divided into four types: unstructural data exchange, structured data exchange, seamless sharing of data and seamless sharing of information.

LCIM [Tolk, 2003] is the interoperability model at a conceptual level, defining the levels of conceptual interoperability, whose purpose is to bridge conceptual design with technical design. According to the availability of data to be interchanged and the interface documentation, interoperability can be divided into four levels by this model.

EIF [EIF, 2004] divides interoperability into organizational interoperability, semantic interoperability and technical interoperability. The aspects of interoperability on organizational, technical and semantic concerned about by the above four models LISI, OIM, LCIM, NATO are coherent with the definition of EIF model.

Table 2-2 Interoperability model of information system [Panetto, 2007]

EIF					Organisational
			Semantic		
	Technical				
LISI	0-Isolated	1-Connected	2-Functional Distributed	3Domain Integrated	4-Enterprise Universal
OIM	0-Independent	1-Ad-hoc	2-Collaborated	3-Integrated	4-Unified
LCIM	0-System specific	1- Documented	2-Aligned static	3-Aligned Dynamic	4-Harmonised
NATO	1-Unstructured data	2-Structured data	3-Seamless data sharing	4-Seamless information sharing	

2.4 Models of Enterprise information system interoperability in supply chain

2.4.1 Problem model

In the environment of supply chain, various kinds of enterprise information systems are involved in the supply chain process among enterprises and departments, such as ERP, CAD and PDM et al, while the effective integration is difficult to be acquired among different information systems.

The main reasons causing this problem: (1) Semantic heterogeneity, means different description to the same object from different institutions (enterprises or departments); (2) syntax heterogeneity (including structure), means different description to the same object from different softwares. It's shown in Figure 2.3.

The reasons causing semantic heterogeneity: (1) the difference of usage, different institutions have different description usages about the same object (pragmatics); (2)

the difference of habit, different institutions have different description habits about the same object.

The reasons causing syntax heterogeneity: (1) the difference of information design, the designers of different softwares have different comprehension about the same object; (2) the difference of usage.

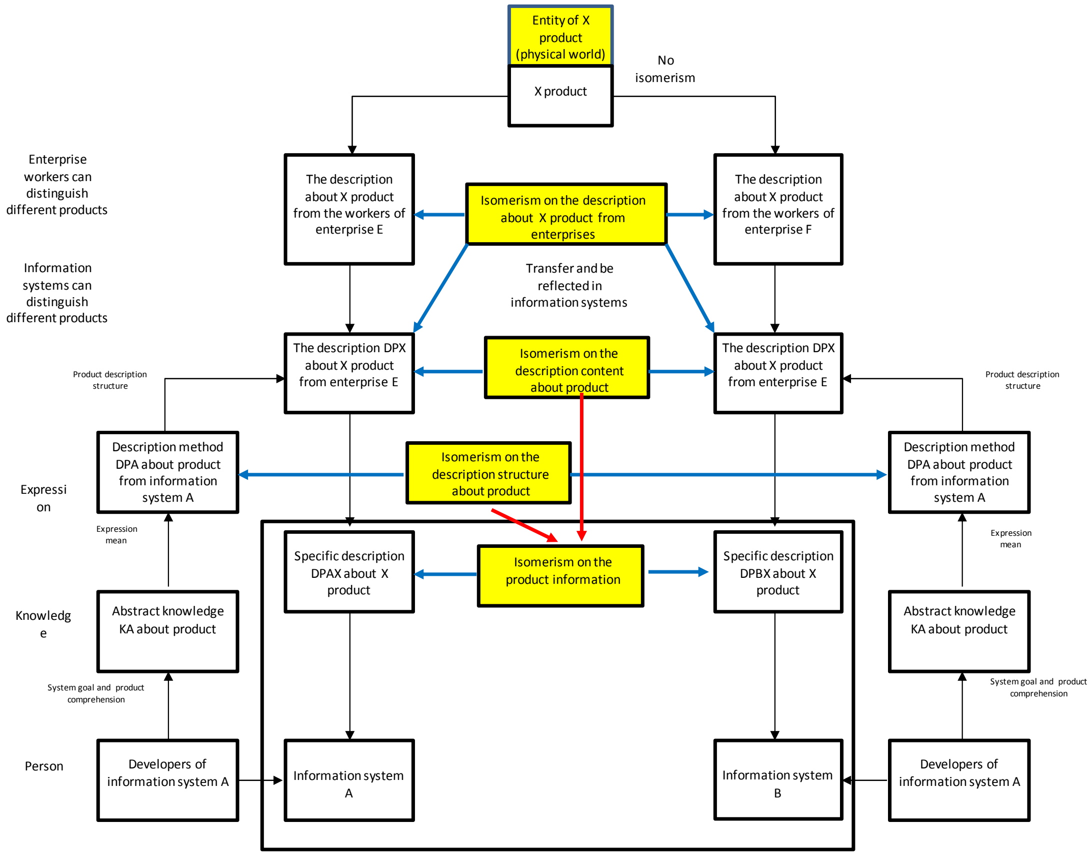


Figure 2.3 The reasons about information heterogeneity in business cooperation process

Supply chain oriented heterogeneous enterprise information system interoperability emphasis on solving the problem of semantic heterogeneity, without changing any information system. As during the supply chain process, the data structure and business logic of information system is the black box for users.

2.4.2 Solution and Models

Some techniques such as XML, CORBA and Web Service et al can preferably solve the syntax heterogeneous problem of external communication of information system, but can’t deal with the semantic heterogeneity inside information system; ontology is the formalized expression of sharing concept, providing new way to solve semantic heterogeneous problem. The thought about solutions for enterprise information system interoperability is shown in Figure 2.4.

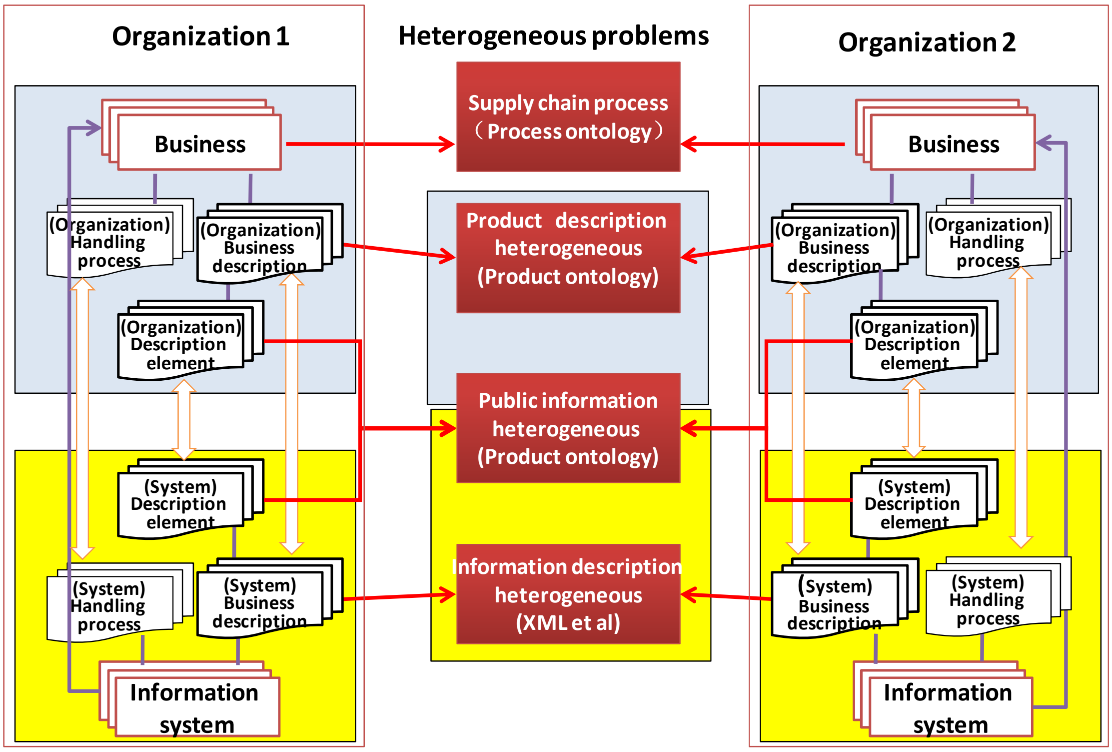


Figure 2.4 The thought about solutions for enterprise information system interoperability

In this thought, the supply chain ontology and product ontology are built separately according to the related information involved in supply chain process and the product in the environment of supply chain. And then the ontology merging of product ontology and supply chain ontology is realized by ontology mapping, to build a common shared product-centric supply chain ontology. In the paradigm of SoNE, each enterprise or enterprise information system is considered as a sub-system, and the product-centric supply chain ontology is a sub-system too. During the whole supply chain process from suppliers to customer, all kinds of enterprise information systems can reach the semantic consistency and realize semantic interoperability by interaction with this common shared product-centric supply chain ontology, which is shown in Figure 2.5.

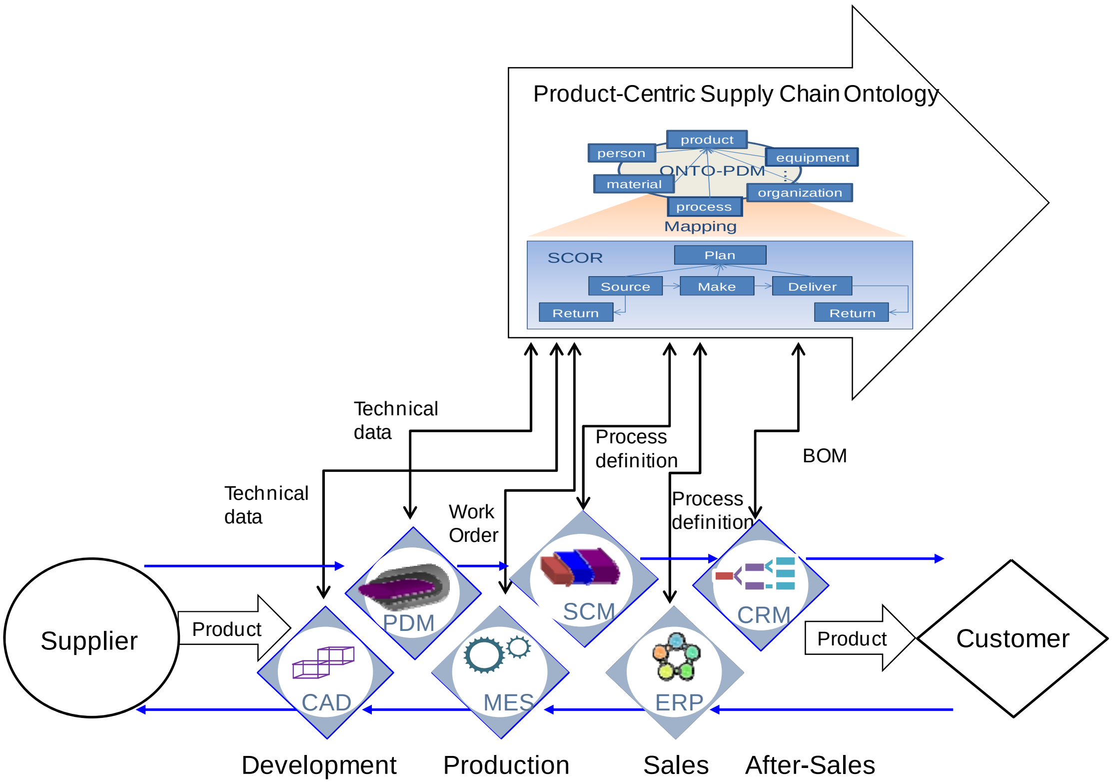


Figure 2.5 Interoperability model of supply chain enterprise information systems

2.5 Framework of product-centric supply chain information system interoperability

2.5.1 Key elements in supply chain environment

It is frequent to find that information are distributed within enterprises: in applications used to manage technical data (e.g.: Product Data Management systems (PDM)), in applications that manage business information (e.g.: Enterprise Resource Planning (ERP)) and, finally, in the applications that manage manufacturing information (e.g.: Manufacturing Execution Systems (MES)).Some research works have already been done to analyse the semantic problems in supply chain and make contribution to it. Millet et al. proposed an extended reference model based on SCOR operation reference model for alignment of business processes and information systems [Millet, 2009]. This model is built on a multi-view of business process mapping, including the informational dimension, and thus a more practical alignment of ERP systems with processes. Sakka proposed to represent the SCOR operation reference model as an

ARIS [Sheer, 2005] process model, and then transformed this ARIS model into a OWL format in order to add semantic into the original SCOR model [Sakka, 2010]. Zdravković proposed a SCOR model ontology for supporting knowledge management in supply chain operations [Zdravković, 2010a][Zdravković, 2010b]. SCOR+[5]is a commercial tool directed towards overcoming the limitations of the basic SCOR model through an ontology based tools, to supply chain process definition at four levels: the supply chain level, the enterprise level, the elements level, and the interaction level. There are also other works based on SCOR operation reference model to build supply chains for special purposes [Fayez, 2005] [Haller, 2008] [Yiqing, 2009]. All the approaches above adopt SCOR model, which is one of the most known supply chain operation reference models currently, as a base for supply chain process definition and configuration. This reference model could be considered as a standardization of domain knowledge by providing categorized concepts, to act a candidate solution for interoperability problem.

However, all these researches are mainly focusing on supply chain process definition and configuration. But supply chain is not only just a process, also involves product, human, resources etc. The change of manufacturing pattern was one of considerable reasons leaded to supply chain management emergence. Such change was actually due to product complexity, product module reuse extensively. Related works demonstrated that, while product is the centred value of enterprises processes, its information-based model may act as a common pivotal information system to make all enterprise systems interoperating [Vegetti, 2005][Terzi, 2007][Tursi, 2009b][Zdravković, 2009]. Hence product plays an important role in supply chain interaction. And it is much more meaningful to considerate them, especially product design and manufacturing stage of product life cycle, for systems integration within extended supply chain environment.

In the context of networked enterprises, and mainly in supply chain environment in which both business enterprises and manufacturing enterprises are involved,

[5]http://www.productivityapex.com/products/scor.asp

information exchange emphasizes more about inter-enterprise relationships, which are not concerning only products but also processes related to customers, market, service and so on. Information exchange focus is moving from integrated intra-enterprise application packages to internet-based and inter-enterprise application software. Improving Supply Chain Management (SCM) and Customer Relationships Management (CRM) are key processes to enable enterprise value chain [Kirchmer, 2004]. Thus in order to reach maximum comprehension between enterprises, and more effective information exchange, more knowledge is needed. Then we introduce SCOR operation reference model and ONTO-PDM product ontology.

2.5.2 Framework of product-centric supply chain information system interoperability

Building supply chain ontology for supply chain process and building product ontology for the product in supply chain environment, and merging ontologies of product ontology and supply chain ontology by adopting the method combining WordNet based ontology mapping with the rule based reasoning, to build the product-centric supply chain ontology. Supply chain enterprise information system interoperability is realized from two aspects: the matching between ontology bases and the matching between data bases and ontology bases. The framework of product-centric supply chain enterprise information system interoperability is shown in Figure 2.6.

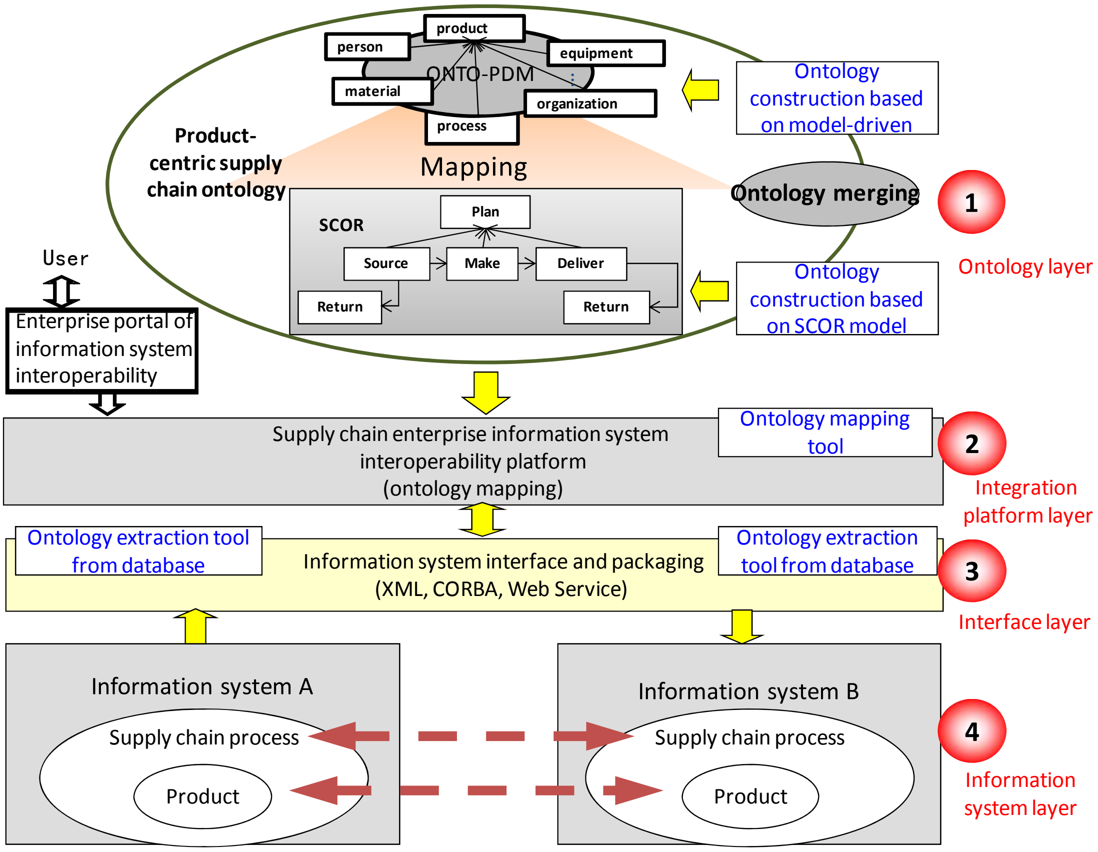


Figure 2.6 The framework of supply chain enterprise information system interoperability

This framework is composed by four layers: information system layer, interface layer, interoperability platform layer and ontology layer. The content of each layer is expressed in Table 2-3.

Table 2-3 The four layers of supply chain enterprise information system interoperability framework

Layer name	**Content explanation**
Ontology layer	This layer mainly deals with the construction of ontology base, which is the core to realize interoperability for enterprise information systems in supply chain environment. ● Based on the related international standards involved in enterprise information system integration, we adopt the model-driven ontology development method to concentrate the related business information and technical data of product life cycle such as product design, manufacturing and deliver et al to a unified product ontology. ● Adopting SCOR model as the basic concept and framework for supply chain ontology construction, we build supply chain ontology of SCOR according to the seven-step method of Protégé. ● Adopt the method combining WordNet based ontology mapping with the rule based reasoning to realize the ontology merging of product ontology and

	supply chain ontology, to build the product-centric supply chain ontology.
Interoperability platform layer	Users can complete the interoperation work of information system by the interoperability platform. The core technology and method is to adopt ontology mapping to realize semantic interoperability.
Interface layer	Apply some information techniques supporting the information system interoperability in supply chain environment, such as CORBA, XML, Web service, Agent technology, SET safety certificate technology and STEP product data exchange technology et al.
Information system layer	Researching object: the information systems of networked enterprises in the environment of supply chain.

In this framework, our research emphasis on the construction of model-driven based product ontology, the construction of SCOR model based supply chain ontology, ontology merging and heterogeneous system semantic interoperability.

(1) Construction of model-driven based product ontology

In the environment of supply chain, quantities of researches indicated that product could be regarded as the core of enterprise value chain. For there is no unified standard for product ontology model at present, product ontologies are numerous and complicated, which leads to the difficulty of ontology reuse. The established ontologies stayed on the theoretical stage and don't satisfy the universal international software standards, which are difficult to be applied. The dissertation adopted the product ontology model development method based on international standards, combining the research results of our research group and CRAN laboratory of Nancy University in France, based on the related international standards involved in enterprise system integration such as IEC 62264 and ISO 10303 STEP-PDM, to integrate the related business information and technical data of product life cycle such as product design, manufacturing and deliver et al to a commonly used product ontology model.

The IEC 62264 set of standards define an information exchange framework to facilitate the integration of business applications and manufacturing control applications, which related to product production phase. It can be used to integrate business enterprise applications such as ERP, supply chain management system with manufacturing enterprise applications such as MES. ISO 10303 STEP-PDM is a standard for computer interpretable representation and exchange of product definition data, which aims at providing a mechanism capable of describing product data throughout the life cycle of a product. It is used for exchanging information between ERP and PDM systems. It deals with typical product-related information including geometry, engineering drawings, project plans, part files, assembly diagrams, numerical control machine-tool programs, analysis results, correspondence, bills of material, engineering change orders and many more.

Therefore, the ONTO-PDM product ontology based on IEC 62264 and ISO 10303 STEP-PDM not only contains product technical data and geometric data, but also includes the information data related to business, concentrating almost all the related information span from product development to manufacturing and related business process.

(2) Construction of SCOR model based supply chain ontology

Supply chain is a dynamic process combination, while how to express the dynamic configuration process of supply chain process effectively and how to express the efficiency of supply chain process on conceptual are the difficulties for supply chain ontology development. The supply chain ontology development method based on SCOR model was adopted in the dissertation, taking SCOR model as the basic concept and framework for supply chain ontology development. Supply-Chain Operations Reference-model (SCOR) is a process reference model developed and endorsed by the Supply-Chain Council (SCC), applying to different industry field. SCOR can make enterprises communicate accurately with each other about supply chain, evaluate the performance objectively and ensure the goal of performance modification, and then affect the later development of supply chain management softwares. The SCOR model provides a unique framework that links business processes, metrics, best practices and technology features into a unified structure to support communication among supply chain partners and to improve the effectiveness of supply chain management and related supply chain improvement activities. SCOR is based on five distinct management processes: plan, source, make, deliver, and return. According to the pyramid supply chain process of SCOR, the supply chain process of the supply chain ontology in the dissertation was defined as three architectures: Process type, Process category and Process element, to express the configuration of supply chain process flexibly. Meanwhile, various evaluation metrics of supply chain process in SCOR model are included, providing the valuable references to enhance the executing efficiency of supply chain and optimize supply chain management.

(3) Ontology merging and heterogeneous system semantic interoperability

The semantic interoperabilityin the dissertation is completed by the two parts of ontology mappings, including the mapping among shared ontology bases and the mapping between data base and ontology. ONTO-PDM ontology and SCOR supply chain ontology are from two fields, and are developed separately. To construct the common shared product-centric ontology, the mapping relationships between them need be discovered. Semantic heterogeneous relationship also exists between external data base and shared ontology data base, so the mapping relationships should be discovered by matching.

Product ontology and supply chain ontology are both built on international standards. Although there already are mass of research on ontology development method, as mentioned in chapter 1.3.2, most ontologies developed today are small-scale, nonstandard, stable and hard to reuse. So ontology development is still a open question. The difficulties of ontology development are lack of methodology or standards for ontology development, configuration management, web service or other application storage support; little tools supporting ontology development; lack of existed ontology base. Meanwhile, ontology development is a circle innovation process, without strict range and scope. Two ontologies could be different in the same domain. It is difficult to build a standard ontology, even in a certain organization with uniformed terminology, As firstly, the standardization of terminology is a long period. And mass terminology is a block for organization to change which needs consensus by all people. Then cross-organizational ontology standard is a critical problem, as both consensus by people and conflicts on focus exist [Pepijn, 1998] [Uschold, 2000].

Standard are collection of knowledge recognized and development by domain experts, validated for a long time. It could be an assist for ontology development, and provide some solutions when encounter difficulties.

When the framework of product-centric supply chain information system interoperability proposed in the dissertation compared to the interoperability models in table2-2, it is mainly to realize interoperability in technical and semantic, as show in Figure 2.7.

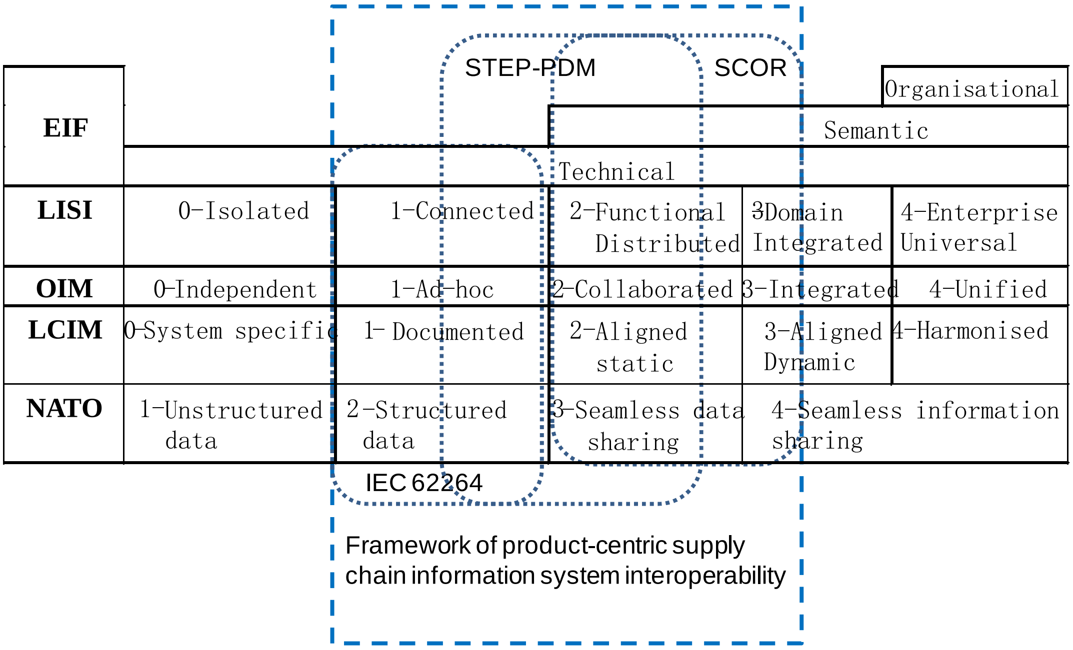


Figure 2.7 The position of the framework of product-centric supply chain information system interoperability in interoperability models

2.6 Conclusion

The SoS characteristics of networked enterprises in supply chain environment were analyzed in this chapter, the Systems-of-Networked Enterprises (SoNE) paradigm is proposed. The difference between enterprise information system integration and interoperability was analyzed, while the product-centric supply chain enterprise information system interoperability framework was proposed from the problem model of information system interoperability in the environment of supply chain.

3 Model-Driven based Product Ontology Development

3.1 Introduction

Product is the core of enterprise value or service, and the basic information which enterprise's manager and information system concerns in the environment of supply chain. There are mass of research on product ontology in specific environment or for specific product, which lead to difficulties of ontology reuse. Studying on product data integration standards related to enterprise information systems, and building a common shared product ontology based on standards is important for realizing enterprise information semantic integration in supply chain environment.

3.2 Standards based Product information model

3.2.1 Standards for Product Data Integration

Product information is the most basic information involved in all systems and applications interaction inside and among enterprises. Product model should express the production process conducted by manufacturing enterprises [Hegge, 1995]. It is valuable to construct a common shared product model, which stores information about product knowledge and process, including the information about manufacturing, storage, sales and distribution et al., for the information exchange among all enterprises involved in the value chain on supply chain. Nowadays, quantities of standards used to integrate enterprise product model through normalized definition of technical knowledge about product whole life cycle, in order to solve the problem of information interoperability. Among the rest, ISO 10303 and IEC 62264 are widely adopted international standards, which completely define models related to product and manufacturing. ONTO-PDM product ontology model has been built based on these standards.

(1) ISO 10303：STEP

STEP (Standard for Exchange of Product model data) is the standard recognized and used widely during product development phase. It is established by international organization for standardization, as an ISO standard (ISO 10303) for computer interpretable representation and exchange of product definition data,.This standard provides a neutral mechanism independent on any systems to describe product data throughout the product life cycle, acting as the foundation of the implementation and sharing product database and files. The typical implementation is computer application softwares used for product design such as CAD, CAM, and CAE, etc. The purpose of the STEP standard is to describe the product data as a kind of neuter files exchange among different software solutions, also in a distributed engineering or manufacturing environment. STEP integrates the product information from mechanical and electronic design, geometry dimension and tolerance, and analysis and manufacturing et al., also including additional information of various industries domain, such as automation, vehicle, vessel manufacturing, oil field and natural gas et al. STEP is helpful for the integration of industry manufacturing systems.

The most important aspect of STEP is its extensibility: STEP has a structure based on reconfigurable and modular, using Application Protocols (APs) to specify the representation of product information for different application systems. APs are subset of STEP， focused on specific issues or specific industrial sectors, which break the entire STEP standard into easily manageable views of quick implementation. STEP uses EXPRESS language to describe data types, data type constraints and the relations bewteen data types. APs adopts EXPRESS and EXPRESS-G to express information, while EXPRESS-G is a graphic technique supporting a subset of EXPRESS language.

A significant solution for PDM (Product Data Management) data exchange is Unified PDM Schema, which is a basic specification for exchange of administrative product definition data. It defines any product metadata through unifying all PDM data existing in STEP application protocols, and allows information exchange among PDM systems. To meet the increasing demand of the product model exchange, ISO 10303 provides a set of reusable STEP modules related to PDM, which has been published in the form of technical reports and consists of all the descriptive or additional product technical data such as product structure, configuration control, people and organizations, etc. Data integration must ensure the uniqueness of data definition description on product design, manufacturing and life cycle. STEP data integration eliminates the redundancy while reducing the problems caused by redundant information.

(2) IEC 62264

Product production phase contains product manufacturing, distribution and all other related sub-activities. IEC 62264 set of standards define a set of reference models extending the ANSI/ISA S95 specificatiions., define the information exchange framework for the integration of enterprise business applications and manufacturing control applications, and provide unified models and terminology to express these interfaces. The models and terminology emphasize the best integration implement of control systems and enterprise systems existing in the system life cycle, which could improve the system integration ability of existent manufacturing control systems and enterprise systems. Automaticity can be ignored during use, and the effectiveness of information exchange can be improved by standard term definitions and unified concepts and models [IEC, 2003].

According to the diversity of information exchange, the standard defines 8 models to unify all concepts applying for the enterprise control integration: the resource hierarchy related models (Personnel, Equipment, and Material), the process hierarcy related models (Process Segment, Product Definition), and the production related models (Product Schedule, Production Performance, and Capability Definition). Each model represents one particular aspect of integration problem, and are linked together in a logical sequence in order to define a hierarchy of models as show in Figure 3.1.

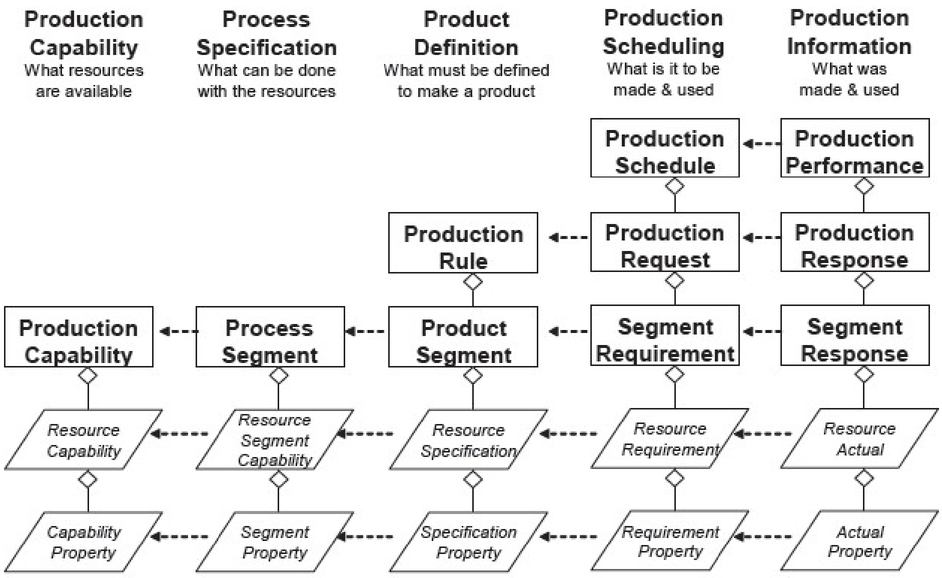


Figure 3.1 Model layers of IEC 62264

B2MML (Business to Manufacturing Makeup Language) is developed by WBF (World Batch Forum) and XML working group, and is an XML implementation of IEC 62264 part 1 [B2MML, 2008]. It is written using XML Schema to description on standardised resources and information flow. B2MML is meant to be a common data format to link enterprise business application (ex. ERP) with enterprise manufacturing application (ex. MES).

3.2.2 Standard based Product Ontology Model (ONTO-PDM)

Tursi et al. [Tursi et al., 2009] [Tursi, 2009] from CRAN in France, have researched on product-centric information systems interoperability in networked manufacturing enterprises, and proposed ONTO-PDM product ontology model for product data management and interoperation. This integrated and common model formalizes the knowledge related to product data management at the business and the manufacturing levels of enterprises (B2M, Business to Manufacturing), in order to achieve the interoperability between systems. It adopts two standards: the IEC 62264 [IEC, 2002] and the ISO 10303 STEP-PDM [STEP, 2004]. The IEC 62264 set of standards define an information exchange framework to facilitate the integration of business applications and manufacturing control applications, which related to product production phase. It can be used to integrate business enterprise applications such as ERP, supply chain management system with manufacturing enterprise applications such as MES.STEP-PDM is a standard for computer interpretable representation and exchange of product definition data, which aims at providing a mechanism capable of describing product data throughout the life cycle of a product. It is used for exchanging information between ERP and PDM systems. It deals with typical product-related information including geometry, engineering drawings, project plans, part files, assembly diagrams, numerical control machine-tool programs, analysis results, correspondence, bills of material, engineering change orders and many more. So ONTO-PDM concentrates most but not all product technical and geometrical data and business related information span from its development to its manufacturing and related business processes. Figure 3.2 shows an extract of the ONTO-PDM ontology concepts . Concept of the ontology model inherits from IEC 62264 and ISO 10303 through the concept mapping, and is expressed by UML diagram. The semantic of ontology model is clearly decribed by First Order Logic (FOL) axiom.

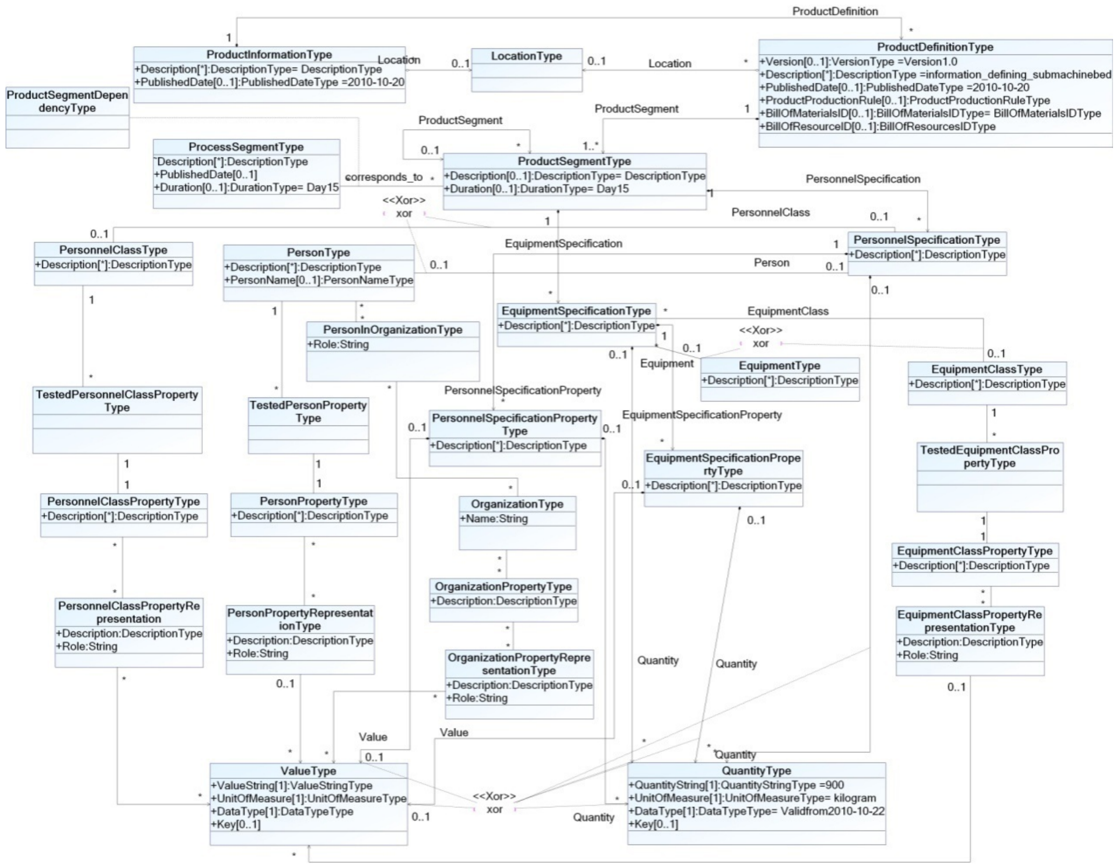


Figure 3.2 An extract of the ONTO-PDM product ontology model

ONTO-PDM model has been applied in distributed manufacturing enterprise environment, including PDM, CAD, ERP, MES, Heterogeneous systems involved in distributed manufacturing enterprise environment need exchanging product information to acquire and store real-time product information. It realizes the interaction among systems and tracking of product information through exchanging information with ONTO-PDM product ontology model, during all phases of product life cycle, from EBOM, MBOM ,customer request, supply and manufacturing product, to product transportation.

3.3 Model Driven Engineering and Ontology Development

The dissertation is the continuation based on the research by the CRAN in French. Concept structure is expressed by UML diagram and concept semantic is expressed by FOL in ONTO-PDM model, which makes it hard to ontology reusing and later ontology mapping. Therefore, this dissertation adopted model-driven method to development ontology, translating UML model into OWL ontology and unifying ontology representation methods to ease later research work.

3.3.1 Model Driven Ontology development method

With the completion of various industrial standards and the mature of object-oriented programming development, more and more business systems especially distributed systems have adopted object-oriented analysis, design and development method [Stephen and Martin, 1999]. UML (Unified Modeling Language) is the commonly used standard for modelling recently, with the primary design purpose of realizing a kind of people communication model for object-oriented programming language systems, and then helping developers participate in the communication of system models. At present, UML has already been used more often in descriptive design, such as RDF Schemas, data pattern and knowledge model et al. UML is not only used in the modelling about the structure, behaviour and architecture of application system, but also in the modelling of business process and data. UML has strong expression ability and extensive application population, especially in distributed information system. There have been quite a number of softwares supporting UML. Meanwhile, UML XMI is also one of OMG standards, and has been widely used in model transformation. MDA makes UML to be a more formalized and more suitable computer interpretable language . Therefore, UML can be used during compiling and running time, rather than a kind of graphical annotation for helping people communicating [Kogut et al., 2002].

OMG MDA and W3C are both devoting themselves to realizing semantic interoperation, while adopting different metadata approaches. OMG extracts and defines core element set based on existing complex metamodels, and then builds meta-metamodel MOF. W3C builds new model layers based on the exisiting model layers, making the model layer more complicated, such as from XML->RDF->RDF Schema->DAML [Kogut et al., 2002]. The knowledge representation languages such as KIF、RDF、OWL of AI (Artificial Intelligence) domain are known limited in research field, rarely supported by softwares. Some interfaces should be extra explored when applying to networked system, which would increase the complexity. At the same time, lots of research achievements are confined in laboratory and hardly to promote, thereby causing resource waste. The building of ODM (Ontology Definition Metamodel) makes the transformation between UML model and OWL model possible, and constructs the bridge between standards of model driven architecture and Semantic Web technologies.

Model-driven ontology development method, is a way to realize the language transformation from UML to OWL by model mapping based on model driven architecture, thus to convert UML model to OWL ontology. There are a few tools supporting the model interconversion recently, because different softwares are based on different metamodels. Model-driven ontology development method can improve the reuse of ontology, and the ontology built can be applied in application development. ODM decreases the number of mapping between different languages to improve the efficiency of ontology development. Although, UML still uses descriptive language to express semantic at present, its graphical model representation method makes it easier for user to discover ontology and its concept. On the contrary, Description Logic is a linear grammar, requiring lots of time to learn relevant grammar knowledge for non-professionals. Model-driven ontology development method can make more people of AI field to participate in the modelling stage of ontology, and take advantage of everyone's tacit knowledge.

3.3.2 Model Driven Engineering and Model Driven Architecture

Model-driven Engineering (MDE) originated from software engineering, is a software design method which through describing a designed system by model firstly, and then converts it into the actual system. Object-oriented technology, whose main principle is dealing with everything as an object, is applied to design pattern by MDE. Similarly, the main principle of MDE model paradigm is anything seen as a model [Bezivin, 2005]. Through reusing the standard model, MDE contributes to maximum optimize the product compatibility, simplify the design process, promote the communication among the developers or groups, and then to improve the efficiency of system development. Bran Selic considered that any system or software reflects and processes human ideas in essence, not physical artifacts. Hence MDE model paradigm efficiently reflects the intention of product users, including designers, developers, and customers et al who could promote product development. Recently, the main research results and innovation of MDE are Model-Driven Architecture (MDA) studied and established by Object Management Group (OMG)[6], and Eclipse[7] programming and modelling integrated software.

MDA initiative is an approach to system specification and interoperability architecture based on the use of formal models, with primary purpose on improving the portability, interoperability, and reusability of systems through architectural separation of concerns. In MDA, Platform-Independent Models (PIMs) was initially expressed in a platform-independent modelling language, such as UML. Then, PIM is subsequently mapped to some implementation languages or platforms (e,g. JAVA)

---

6 http://www.omg.org/mda/
7 http://www.eclipse.org/

by formal rules, which result in transforming from PIM to Platform-Specific Model (PSM). A series of OMG standards compose the core of MDA such as UML (Unified Modeling Language), MOF (Meta Object Facility), XMI (XML Metadata Interchange), and CWM (Common Warehouse Metamodel) et al., which have made tremendous contributions to the current state-of-art of system modelling [John, 2001] [MDA, 2003]. In other words, under the model-driven architecture, it expresses business related system's function and behaviour modules, such as authorization, distribution, and management modules, by OMG modelling standards (e.g. UML) replacing traditional programming languages, to realize the platform-independent virtual system, including Web services, .NET, CORBA, J2EE, etc.. On the technology view, MDA describes the system as standard modules and separates it from implementation technology and development platform, and provides a solution for the interoperability among heterogeneous systems on the other hand by making the interoperability independent on CORBA standards and component interfaces. On the function view, MDA separates business requirement from development technique. While function modules changes and refines itself continuously based on business requirement, the technology upgrades with dramatic technology development to satisfy business requirement.

MDA is a metamodel architecture of four-layer structure illustrated in Figure 3.3, with the change of linguistic instantiation in vertical axis and the change of ontological instantiation in horizontal axis. The topmost layer M3 is a meta-metamodel (MOF), It is an abstract, self-defined language and framework for specifying, constructing and managing technologically independent metamodels, such as UML, or MOF itself. MOF can also be considered as an upper descriptive language. Metamodels of UML, ODM, CWM in MDA as previously mentioned are all defined by MOF. All of the standard or user defined metamodels locate in M2 layer, and are defined in MOF. Models of the real world, which are represented by concepts from metamodel belonging to the M2 layer, are in the M1 layer. M0 layer is the instantiation of concept models of M1 layer, including the concept instance defined in M1 layer and things come from the real world.

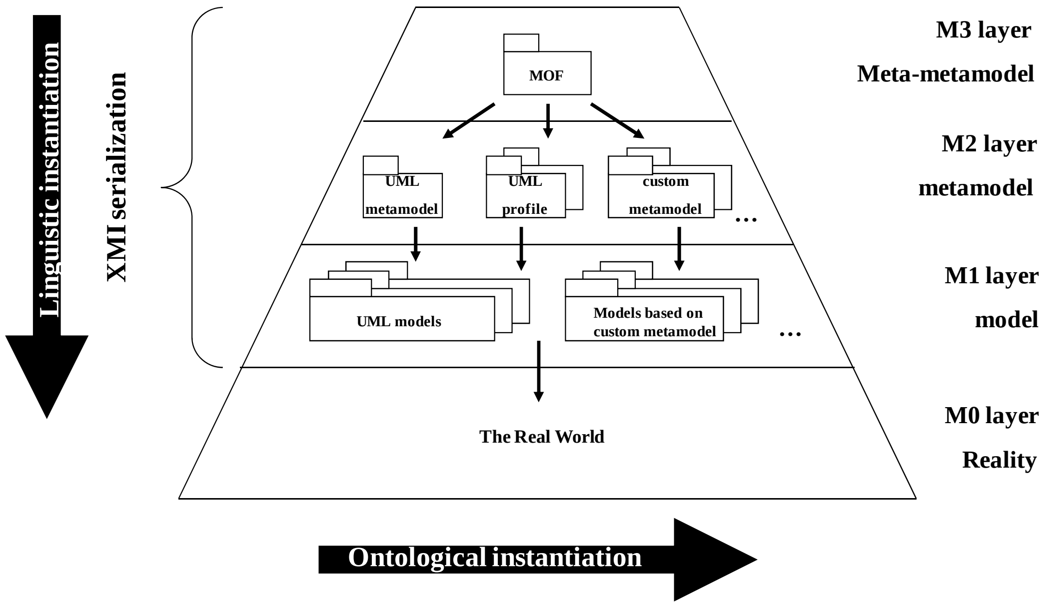


Figure 3.3 Four-layer structure of MDA [Gaševic et al., 2009]

3.3.3 Ontology Definition Metamodel (ODM)

Ontology Definition Metamodel (ODM) are specifications in MDA including a series of independent metamodels defined by MOF2, related profiles and mapping among metamodels corresponding to several international standards, aiming at supporting Semantic Web ontology modelling. The ODM is applicable to knowledge representation, conceptual modelling and formal taxonomy development, and enable utilizing enterprise model as starting points for ontology development through mapping to UML and MOF. The ontology based on ODM can be used to support interchange of knowledg among heterogeneous computer systems, representation of knowledge in ontology and knowledge base, and specification of expressions that are the input to or output from inference engine [ODM，2009]. The structure of ODM metamodel is illustrated in Figure 3.4. At the core are two metamodels thate represent formal logic language: DL (Description Logic)and CL (Common Logic). There are three metamodels that represent more structural or descriptive representations that are less expressive in nature than CL and DLs: metamodels of abstract syntax for RDF, OWL and TM. The RDF metamodel and OWL metamodel is based on ontology modeling languages RDF and OWL of W3C standards, while OWL metamodel inherits RDF metamodel. TM (Topic Maps) can be used as a standard for knowledge representation and knowledge exchange based on XML Schema. So through TM metamodel, ODM can support all ISO standard ontology languages except for W3C ontology language, such as ISO 13250 data model and XML serialization, ISO 18024 query language and ISO 19756 constraint language.

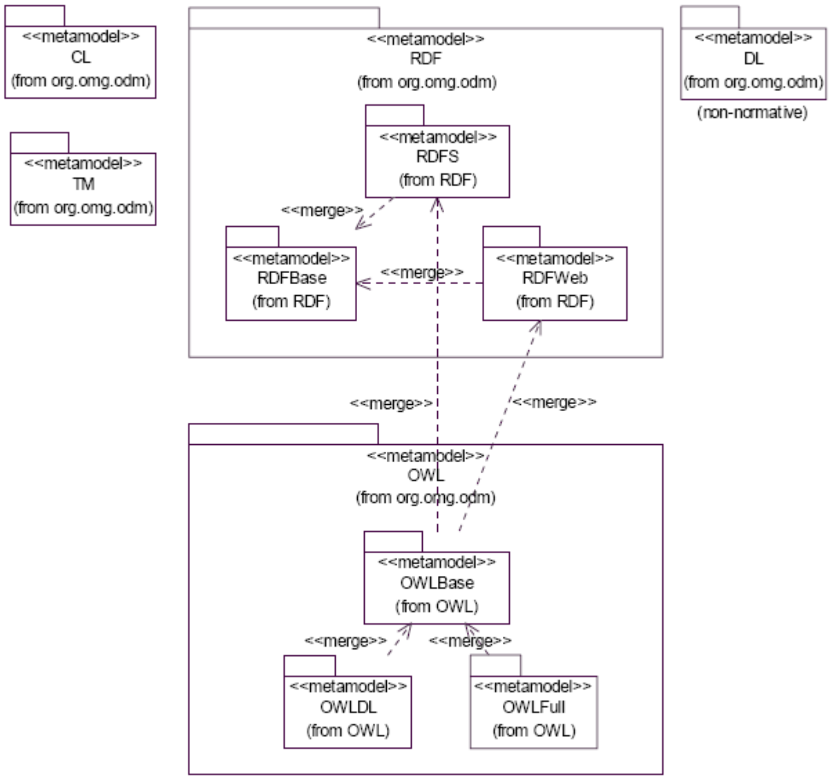


Figure 3.4 The Structure of ODM metamodel

OWL metamodel is one of main metamodels of ODM, forming the foundation of model-driven ontology modeling method. Part of OWL metamodel structure based on MOF is expressed by UML model in Figure 3.5. A series of class corresponding to OWL features are defined in OWL metamodel based on the definition of OWL by W3C in [Sean,2004]. OWLClass represents the set of ontology concepts of owl:Thing, having semantic relationships with other ontology concepts through relationships such as OWLcomplementOf, OWLunionOf, and OWLintersectionOf et al. Property represents rdf:Property in OWL language, meaning the attribute specification of concepts, and can be divided into OWLObjectProperty and OWLDatatypeProperty based on the difference of property objects. OWLRestriction represents constraints to property in OWL language: owl:Restriction, and can be divided into value constraint and cardinality constraint. Value constraint can be expressed through relationships such as SomeValueForClass, AllValueForClass and HasLiteralValue et al. Cardinality constraint can be expressed through relationships such as MinCardinalityForClass, MaxCardinalityForClass and CardinalityForClass et al. Different with ontology concepts defined by OWLClass, OWLmetamodel also defines Individual to express the instance of OWL.

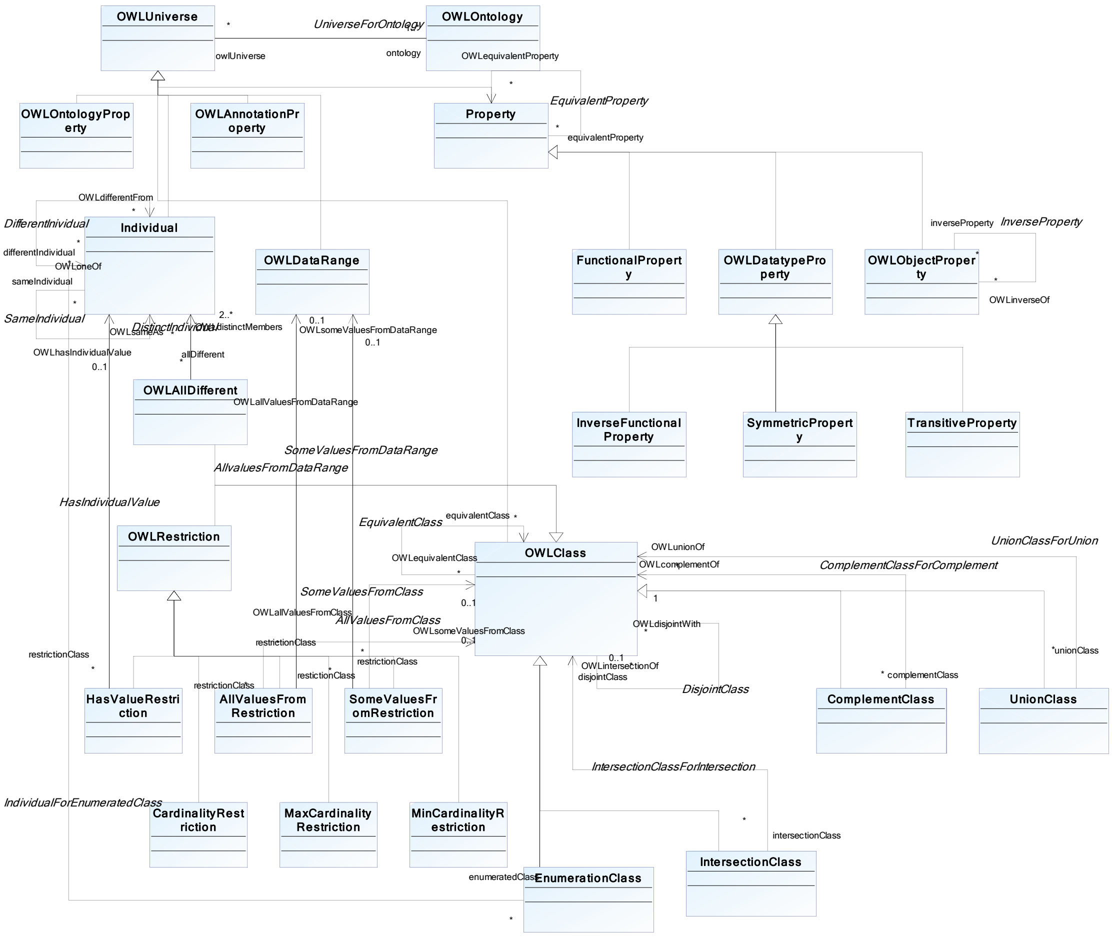


Figure 3.5 OWLBase Package of OWL metamodel

Furthermore, two additional important metamodels considered essential to the ODM represent more traditional to concept modelling is UML2 and ER (Entity Relationship) diagramming. UML2 metamodel has been one of OMG standards and the whole ER diagram metamodel will come in future. For supporting the reuse of legacy models and then takeing it as the starting point of ontology development, ODM defines the mapping among different models, mainly among RDF metamodel, OWL metamodel and TM metamodel.

3.3.4 Model transformation language (ATL)

ATL (Atlas Transformation Language) developed by ATLAS INRIA&LINA research group, is a model transformation language, conforming to OMG MOF and QVT and providing a way to generate the target model from source model for the developers in model-drive field [ATL, 2006]. ATL adopts the unidirectional programming mode combining declaration and imperative constructs, to match source model data to target model by some rules. ATL supports the model transformation between the MOF meta-metamodel defined by OMG and the Ecore meta-metamodel defined by EMF. In other words, any metamodel comforming to the semantics of MOF or ECORE can be transformed by ATL. The famework of model transformation process is illustrated in Figure 3.6 [Jouault et al., 2008]. Currently, there are some tools supporting ATL, which can be used as plugins integrated in the Eclipse platform and provide a series of toolkit supporting the transformation between XML documents or other standard documents and MOF or Ecore metamodel, such as XML injection, EBNF extraction, etc.

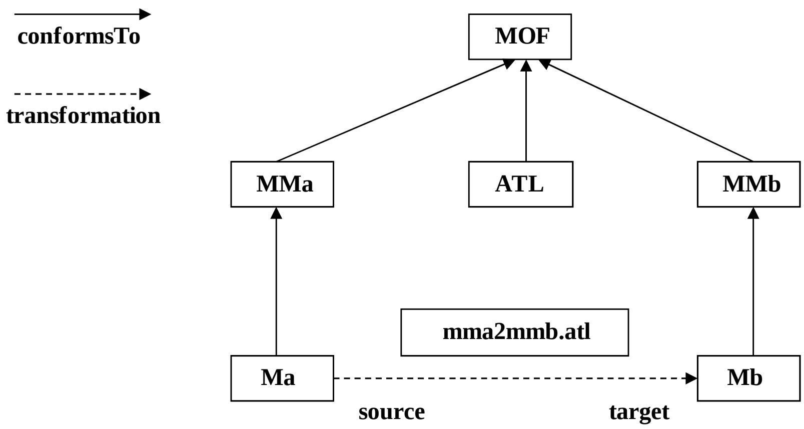


Figure 3.6 Framework of model transformation process of ATL

3.4 Model-Driven based Product Ontology Development

3.4.1 Model-Driven based Ontology development framework

Based above discussion, UML metamodel locates M2 layer in model-driven framework of MDA, which conform to MOF metamodel. OWL metamodel as a submodel of ODM also conform to MOF metamodel. So ATL can be used to support the transformation from UML model to OWL model. ODM provides a series of metamodels, such as OWL metamodel and UML metamodel, and mappings among these metamodels which combining metamodel domain with ontology domain. Therefore, the mapping from UML model to OWL ontology based on ODM metamodel can be realized through the model transforming process of ATL. Although Hillairet [Hillairet, 2007] developed the transforamtion approach of ATL from UML model to OWL model by using the previous version of OMG ontology definition metamodel, there are some shortcomings of this approach as the continuous development and improvement of ODM. It could not perform a complete model transformation and result in partial loss of model information, for example: the range information after conversion would be lost, when the attribute range of class in UML model is defined by entity type rather than the primitive type. The dissertation made improvement on the basis of Hillairet's method.

The process of transfer approach based on ODM from UML model to OWL ontology is illustrated in Figure 3.7. The input of this model transfer method is UML2 concept model while the output is OWL ontology model. For recently most of modelling softwares provide the function supporting that UML2 concept model diagram can be saved as UML XMI document, meanwhile OWL ontology language can use the expression syntax based on XML, XML XMI is adopted by to act as the intermediate format of document exchange. The main conversion processes include three steps:

1) Transforming UML2 model to OWL ontology model conforms to ODM metamodel (UML2OWL): this conversion process takes UML2

metamodel and OWL metamodel as the source metamodel and target metamodel of model transformation respectively, transforming by mapping rules of ATL based on the mappings relationship between UML metamodel and OWL metamodel defined by ODM.

2) Transforming ODM ontology model to the ontology model based on syntax expression of XML (OWL2XML): this conversion process takes OWL metamodel and XML metamodel as the source metamodel and target metamodel of model transformation respectively, transforming by the rules of ATL.

3) Transforming the ontology model based on syntax expression of XMI to OWL ontology model using XML Extractor.

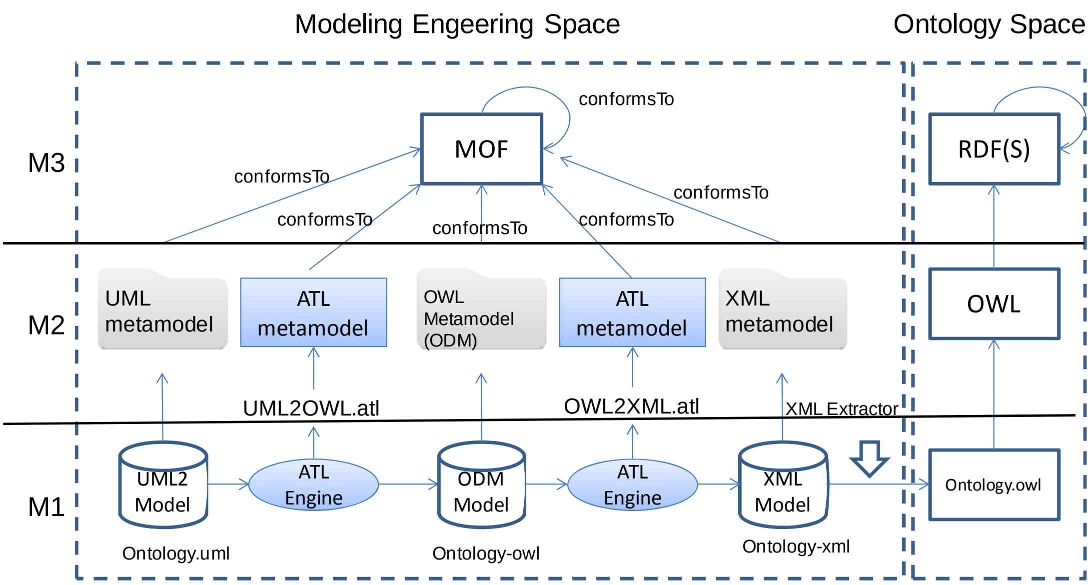


Figure 3.7 Conversion process of UML2OWL by ATL

3.4.2 Mapping between UML Metamodel and OWL Metamodel

During the model transformation process mentioned above, the mappings between UML metamodel and OWL metamodel are the key of whole convertion. Therefore, it is necessary to discuss the mapping before proposing the ontology development instance based on model-driven method. Because of the continuous development of UML and OWL standards, W3C has published OWL2 ontology language. The dissertation adopts UML2.0 and OWL DL to discuss mapping relationship. Although the graphical model expression mode and descriptive semantic explanation of UML are difficult for the semantic reasoning, the class diagram of UML can express the concept of ontology clearly.

Class diagram is mainly used for ontology concept modelling in UML. The representation of concept is mainly by elements in class diagrame, such as *Class*, *Association*, *Property* and *Generalization* et al. Through comparing of the elements in UML with the element in OWL one by one, Table 3-1 shows the corresponding relationships of elements between the two languages.

Table 3-1 Some OWL concepts corresponding to concepts of UML

UML element	OWL element
package	*Ontology*
class	*Class*
instance	*Individual*
property	*ObjectProperty/DatatypeProperty*
association	*ObjectProperty*
generalization/subclass	*rdfs:subClassOf*
generalization/meta-association	*rdfs:subPropertyOf*
multiplicity	*minCardinality/maxCardinality/ cardinality*
primitivetypes	*xsd datatypes*
enumeration	*oneOf*

Some Notable Problems during the transformation process between UML model and OWL model:

(1) UML and OWL both have namespaces. Objects of UML can be distinguished by name in the minimize environment. For example: classes, relationships and other objects of a package can be distinguished by name, while the names of attributes in different classes can be same, classes and relationships of different packages can be same. However, object is recognized by unique URI in OWL. Therefore, when UML objects are mapped to OWL objects, it needs to distinguish the objects with the same name in different classes.

(2) Four main relationship in UML class diagram include: *Association*, *Dependency*,

Generalization and *Realization*. *Realization* expresses the relationship between class and interface, which is not involved in ONTO-PDM product model. And there is no corresponding relationship of *Dependency* in OWL. Therefore, the dissertation mainly concerned about the mapping of *Association* and *Generalization*. At the same time, *Aggregation* is a special *Association* expressing whole–part relationship among classes, while *Composition* can be considered as a kind of stronger *Aggregation* [Booch et al., 2005]. Therefore, the *Aggregation* and *Composition* relationships involved in the mapping are all considered as a kind of *Association*.

(3) UML and OWL are both based on class, and allow the inheritation of class. All UML classes are disjoint and considered as the subclasses of *Thing* when mapped to OWL classes.

(4) Property of OWL is the subclass of *Thing*, which is an independent element of global scope used to describe any class. But the attribute and association of UML have scope constraints, limited only to the class they belong to. Therefor the domain and range of property must be defined when mapping from UML model to OWL model.

(5) Instance of UML is the specific demonstration of class, but the instance model isn't included in class model. And individual is an independent entity in OWL. So mapping of UML instances is not included in this dissertation.

(6) Some special identifications are used to describe properties in OWL, such as *InverseFunctionalProperty*, *FunctionalProperty*, *SymmetricProperty*, and *TransitiveProperty*. There are no corresponding elements in UML directly corresponding to the OWL elements as mentioned above, but properties of OWL after mapping can be adding corresponding *InverseFunctionalProperty*, *FunctionalProperty*, *SymmetricProperty* and TransitiveProperty based on the constraints of the property and relationship of UML.

(7) Relationships of *intersectionOf*, *unionOf* and *complementOf* in OWL are absent among UML classes.

3.4.3 Instance of Product Ontology Development based on Model-Driven Method

According to the mapping rules mentioned above from UML model to OWL model, UML model of ONTO-PDM product were mapped to OWL ontology by ATL model transformation language. Take a part of typical UML diagram including *class*, *attribute*, *association* and *multiplicity* from the ONTO-PDM UML model as a example, We will explain the mapping process from UML model to OWL ontology.

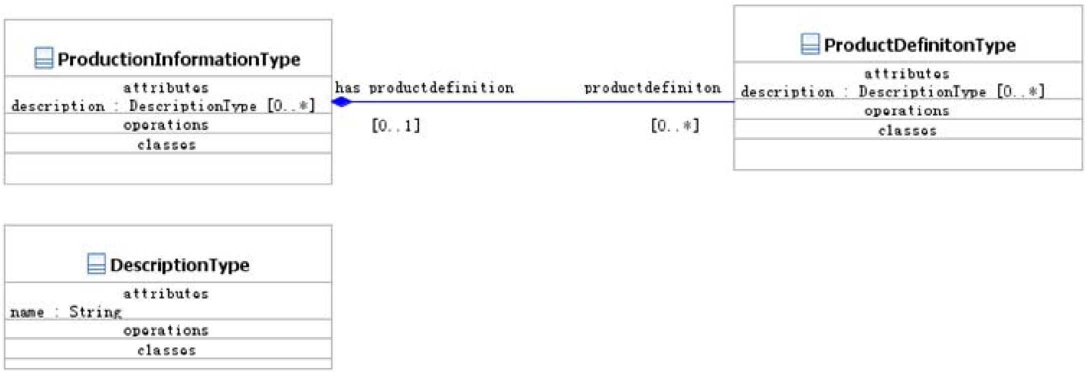


Figure 3.8 Part of ONTO-PDM UML model instance

A part of ONTO-PDM UML model is illustrated in Figure 3.8, with three classes: *ProductionInforamtionType*, *ProductDefinitionType* and *DescriptionType* included in this model. Thereinto, *ProductionInforamtionType* and *ProductDefinitionType* have the attribute“*description*”, whose type is entity type “*DescriptionType*”. The attribute type of *DescriptionType* is data type “*primitivetypes:string*”. *ProductionInforamtionType* and *ProductDefinitionType* have the mappings listed as follows:

(1) Mapping from the package of UML to OWL

UML and OWL are both modular structures owning namespace. The *package* of UML is mapped to the *Ontology* of OWL. And the *package* of above-mentioned product UML model is expressed as:

```
<uml:Package xmi:version="2.1"
    xmlns:xmi=http://schema.omg.org/spec/XMI/2.1
    xmlns:uml="http://www.eclipse.org/uml2/2.1.0/UML" name="Package">
```

```
    ……
</uml:Package>
```

Mapping to OWL is expressed as:

```
<owl:Ontology rdf:about = 'http://www.example.org/Package#'/>
```

(2) Mapping from the *class* of UML to OWL

The *class* in UML is mapped to the *class* in OWL, while the two classes of *generalization* relationship can be mapped to the relationship of class and subclass. The example provided by this chapter did not contain this relationship, so *ProductionInforamtionType*, *ProductDefinitionType* and *DescriptionType* in UML model are mapped to three independent classes in OWL.

Class in UML is expressed as:

```
<packagedElement xmi:type="uml:Class" xmi:id=" "
    name="ProductDefinitonType">
   ……
</packagedElement>
<packagedElement xmi:type="uml:Class" xmi:id=" "
    name="ProductionInformationType">
    ……
</packagedElement>
<packagedElement xmi:type="uml:Class" xmi:id=" " name="DescriptionType">
    ……
</packagedElement>
```

Mapping to OWL is expressed as:

```
<owl:Class rdf:about = '#ProductDefinitonType'>……</ owl:Class>
<owl:Class rdf:about = '#ProductionInformationType'>……</ owl:Class>
<owl:Class rdf:about = '#DescriptionType'>……</ owl:Class>
```

(3) Mapping from the *attribute* of UML to OWL

The mapping from the *attribute* in UML to OWL can be divided into two situations. Firstly, the *attribute* should be mapped to the *DatatypeProperty* of OWL if its type is primitive data types such as: String, Boolean and Integer et al.

DescriptionType of the instance provided in this chapter has attribute “*name*” with attribute type “*String*”. So the attribute “*name*” is mapped to the *DatatypeProperty* in OWL, with its domain *DescriptionType* and range *String*. It is expressed in UML as:

```
<packagedElement xmi:type="uml:Class" xmi:id=" " name="DescriptionType">
    <ownedAttribute xmi:id=" " name="name" aggregation="composite">
        <type xmi:type="uml:PrimitiveType"
href="pathmap://UML_LIBRARIES/UMLPrimitiveTypes.library.uml#String"/>
    </ownedAttribute>
</packagedElement>
```

Mapping to OWL is expressed as:

```
<owl:DatatypeProperty rdf:about = '#name'>
    <rdfs:domain rdf:resource = '#DescriptionType'/>
    <rdfs:range rdf:resource = 'http://www.w3.org/2001/XMLSchema#string'/>
</owl:DatatypeProperty>
```

Secondly, the *attribute* should be mapped to the *ObjectProperty* of OWL if its type is entity type such as *class*. *ProductionInforamtionType* and *ProductDefinitionType* of the instance provided in this chapter have the attribute “*description*” with its type “*DescriptionType*” and its domains *ProductionInforamtionType* and *ProductDefinitionType* respectively. It is expressed in UML as:

```
<ownedAttribute xmi:id=" " name="description" type=" "
  aggregation="composite">
……
</ownedAttribute>
```

Mapping to OWL is expressed as:

```
<owl:ObjectProperty rdf:about = '#description'>
    <rdfs:domain rdf:resource = '#ProductDefinitonType'/>
    <rdfs:domain rdf:resource = '#ProductionInformationType'/>
    <rdfs:range rdf:resource = '#DescriptionType'/>
</owl:ObjectProperty>
```

(4) Mapping from the *relationship* of UML to OWL

The mapping of *generalization* has been discussed in the mapping of *class* in UML. The main concerning content here is about the mapping from the *association* including *aggregation* and *composition* in class diagram of UML to OWL. *Association* relationship has four basic characters: *name*, *role*, *multiplicity* and *aggregation*. *Role* connected with class is mapped to the *ObjectProperty* of OWL, *multiplicity* is mapped to the *Cardinality* of OWL and *name* is not mapped to OWL. The UML instance of this chapter exisit *role*: *has_productdefinition* and *productdefiniton* are mapped to ObjectProperty of OWL, with the classes related to role mapped to domain and range. *Association* is expressed in UML as:

```
<packagedElement xmi:type="uml:Association" xmi:id=" " name="relation"
        memberEnd=" ">
    <ownedEnd xmi:id=" " name="has_productdefinition" type=" " association="
 ">
        ……
     </ownedEnd>
    <ownedEnd xmi:id=" " name="productdefiniton" type=" "
aggregation="composite" association=" ">
        ……
     </ownedEnd>
</packagedElement>
```

Mapping to OWL is expressed as:

```
<owl:ObjectProperty rdf:about = '#has_productdefinition'>
<rdfs:domain rdf:resource = '#ProductDefinitonType'/>
<rdfs:range rdf:resource = '#ProductionInformationType'/>
<rdf:type rdf:resource =
'http://www.w3.org/2002/07/owl#InverseFunctionalProperty'/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about = '#productdefiniton'>
<rdfs:domain rdf:resource = '#ProductionInformationType'/>
```

```
<rdfs:range rdf:resource = '#ProductDefinitonType'/>
<rdf:type rdf:resource =
'http://www.w3.org/2002/07/owl#InverseFunctionalProperty'/>
</owl:ObjectProperty
```

(5) Mapping from the *Multiplity* of UML to OWL

Multiplity in UML is used to define the scope of objects, and can be used to constrain class, attribute and association et al. The constraints to attribute and association are mainly discussed here. *Multiplity* can be mapped to the *Property restriction* in OWL, expressing the constraint to property and being a subclass of *owl:Class*,. The upper of *Multiplity* is corresponding to the *maxCardinality* in OWL, while the lower is corresponding to the *minCardinality* in OWL. In the instance of this chapter, *role: has_productdefinition* related to *Multiplity* [0..1], is mapped to the *ObjectProperty: has_productdefinition* in OWL, while *maxCardinality* is 1 and *minCardinality* is 0. Similarly, the *multiplity* of *productdefiniton* and *description* in the instance are mapped to *Property restriction*.

Role: has_productdefinition is expressed in UML as:

```
<ownedEnd xmi:id=" " name="has_productdefinition" type=" " association=" ">
    <upperValue xmi:type="uml:LiteralUnlimitedNatural" xmi:id=" "
value="1"/>
    <lowerValue xmi:type="uml:LiteralInteger" xmi:id=" "/>
</ownedEnd>
```

Mapping to OWL is expressed as:

```
<owl:Restriction>
        <owl:onProperty rdf:resource = '#has_productdefinition'/>
        <owl:maxCardinality rdf:datatype =
'http://www.w3.org/2001/XMLSchema#integer'>1</owl:maxCardinality >
    </owl:Restriction>
……
  <owl:Restriction>
        <owl:onProperty rdf:resource = '#has_productdefinition'/>
```

<owl:minCardinality rdf:datatype = 'http://www.w3.org/2001/XMLSchema#integer'>0</owl:minCardinality >

</owl:Restriction>

The ONTO-PDM UML model can be transformed to OWL product ontology by transformation from UML to OWL using the rules and method mentioned above. Figure 3.9 illustrates the partial relation among *ProductDefinitionType* and *ProductionInforamtionType* and other relevant concepts in transformed OWL ontology.

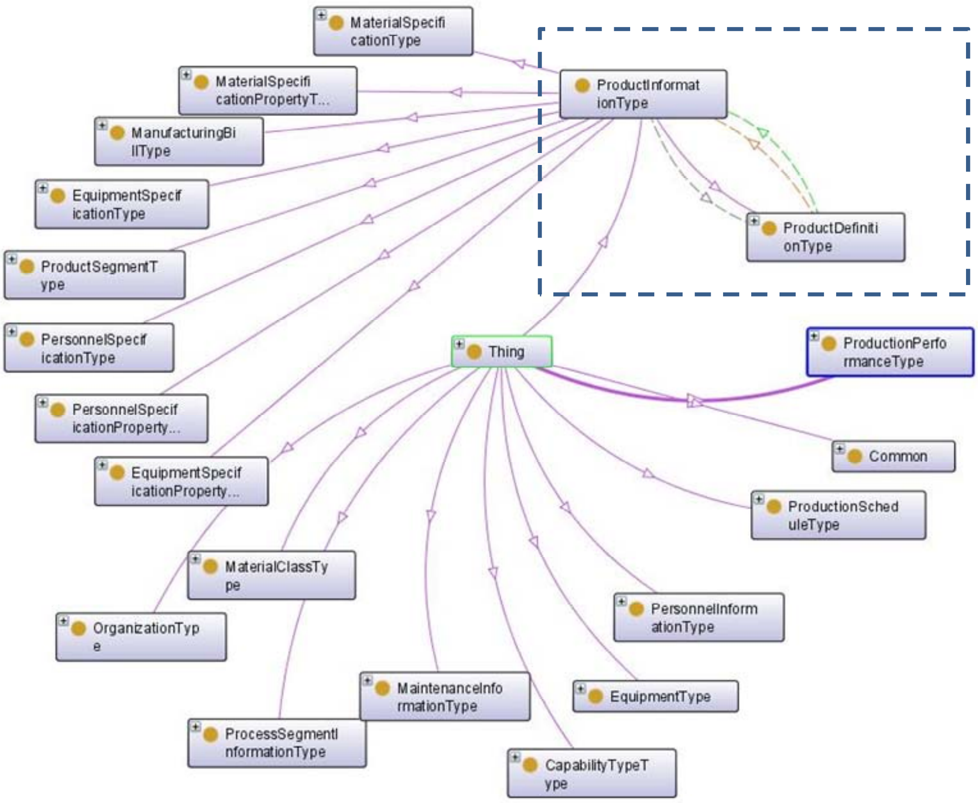


Figure 3.9 OWL product ontology segment

3.5 Conclusion

This chapter analysed the base of ONTO-PDM product ontology: international standards IEC 62264 and ISO 10302 STEP-PDM, studied model driven architecture and combined it with ontology development. Reused ONTO-PDM product model through model-driven ontology development method, and achieved a common shared product ontology integrating business information and technical data related to product life cycle, including product design, manufacturing, transformation etc.

4 SCOR based Supply Chain Ontology Development

4.1 Introduction

In the environment of supply chain, the information is not only limited to product data, but also supply chain process and related organization, person, etc. With the enterprise information systems involved in supply chain environment, supply chain integration needs the supporting of various data standards. To achieve semantic integration, it is more necessary to construct a supply chain model concentring whole information about supply chain process, person, organization etc. It is a complex process to construct such a model. And it is very hard to build a complete supply chain model by one person or a few groups in a short time. Therefore to handle mass of element and information in complex supply chain management (SCM), reference model plays an important role.

4.2 Standards of Supply Chain Integration

Standards play an important role in promoting the process of innovative product and service, providing a stable reference for developing innovative approach and broadening market [Jardim-Goncalves et al., 2011]. Not only in high-technical departments, but also in organizational management departments, the use of standards can accelerate organizational reformation and improve innovative process. During the process of supply chain integration and interoperation, various standards already exist to accelerate the progress, such as system architecture standards, system management standards, data exchange standards and logistics integration standards.

In supply chain system architecture, some reference models for configuration of supply chain process have been proposed recently. The most famous one is SCOR (Supply-Chain Operations Reference-model), developed and supported by SCC

(Supply-Chain Council)[8], for different industrial fields. This model integrats some famous concepts, such as business process reengineering, benchmarking and performance measurement et al., in a cross-functional architecture, which is a process reference model contributing to the effective communication of supply chain partners [Stewart,1997]. SCOR can be considered as a standard language to help managers focusing on management problems and the internal supply chain of enterprise. SCOR can also be used for describing, measuring and evaluating supply chain configuration. The SCM model proposed by Global Supply Chain Forum (GSCF)[9], is built on eight key business processes that are both cross-functional and cross-firm in nature. However all these models focus on supply chain process configure, and do not address any system semantic interoperation problem. RosettaNet [Damodaran and Ana, 2004] is the electronic business process standard developed by global leading high-tech companies. By following the standard, trade cooperation partners, solution providers and system integrator can utilize these technologies and experience. Its main objective is supply chain integration and optimization, which improve the efficiency and performance by enhanced B2B (Business To Business) integration. RosettaNet electronic business process standard is aimed at improving speed, efficiency and reliability, to allow a larger scale of cooperation and communication among trade cooperation partners. It provides a public communication platform or a kind of common language, and allows different trade cooperation partners participated in business process making automatic process and carrying out in internet.

The data exchange standards involved in supply chain include EDI, ebXML and ISO-10303 series standards et al. EDI (Electronic Data Interchange) is a structured transmission of data, such as information about trade, transportation, insurance, bank, customs and other industries, in an internationally recognized standard format. Through the computer communication network, so that all relevant departments, the company and enterprises can exchange and process data, and complete the full course of business focusing on trade [Jilovec, 2004]. .EDI includes the data exchange

8 http://www.supply-chain.org
9 http://scm-institute.org/

between buyer and seller and the data exchange inside enterprise etc., and be used in enterprise supply chain system integration. The ebXML (electronic business using extensible Markup Language) is a set of standards supporting modularization e-business framework [Gibb and Damodaran, 2002]. ebXML supports the global electronic market, makes enterprises of any sizes contact and handle business without the regional limit by exchanging the information based on XML. ISO-10303 series standards or STEP (standard for the exchange of product model data) standards are widely accepted international standards about product model data format exchange, including ISO-10303-21, ISO-10303-236, ISO-10303-28, etc., for different enterprise fields. It adopts the ASCII structure to edit data format, which is easy to be read and brings great help for systems integration both inside enterprise and among enterprises in the environment of supply chain [Jardim-Goncalves, 2011].

Except for the standards mentioned above, supply chain management standards ISO/PAS 28000:2005[Muñoz, 2011], and logistics integration standards: ISO17367:2009[Nambiar, 2010], EPCglobal [Traub et al., 2005] et al all provide specification for supply chain integration in different aspects.

The goal of dissertation is to build a common shared model for the whole supply chain system. Therefore, the standards of supply chain architecture have significant reference value. The reference standards in this aspect are not very much. Among the three relative famous standards mentioned above, although RosettaNet is aimed at supply chain integration and optimization, it is more like a about electronic data integration standard. As to SCOR model and SCM model, SCOR model is the most used supply chain reference model in supply chain management recently, and has a higher cognition in industrial fields. What's more, SCOR has a very detailed concept definition of supply chain process and clear hierarchical structure. Thus SCOR model is adopted by the dissertation as the foundation of constructing supply chain ontology.

4.3 Structure of Supply Chain Operations Reference Model (SCOR)

Supply-Chain Operations Reference-model (SCOR) is a process reference model developed and endorsed by Supply Chain Council (SCC) [SCC, 2001], and is also the first standard supply chain process reference model. The SCOR model provides a unique framework that links business processes, metrics, best practices and technology feature into a unified structure to support communication among supply chain participants and to improve the effectiveness of supply chain management and related supply chain improvement activities. SCOR is based on five distinct management processes: Plan, Source, Make, Deliver and Return, as shown in Figure 4.1.

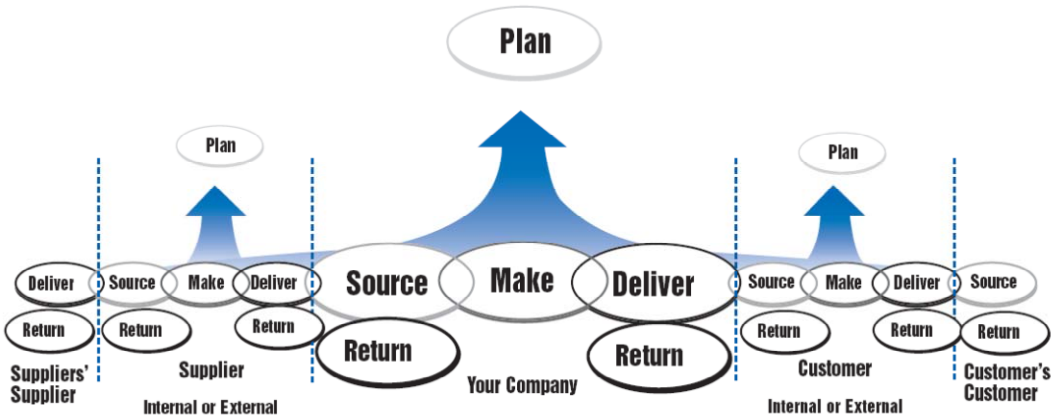


Figure 4.1 the basic management processes of SCOR operation reference model [SCC, 2001]

Viewed from its structure, it has a pyramid decomposition oriented process detail, as shown in Figure 4.2. Level 1defines five process types mentioned before: Plan, Source, Make, Deliver and Return as top level, presenting the range and content of the reference model and ensuring the foundation of competition performance goals of enterprises. Level 2 is a configuration level, and further divides each process types into 26 core process categories according to function and goals' difference, such as source stocked product, source make-to-order product, etc. Level 3 presents detailed process element information for each process category in Level 2.This level includes process flows, all inputs and outputs information, and also performance metrics and best practices for supply chain evaluation. Each process element of the Level 3 can be subdivided based on the practical implementation process of different enterprises and business, but the Council has focused on three process levels and does not attempt to prescribe how a particular organization should conduct its business or tailor its systems/information flow. So Level 4 is not included in SCOR scope, which is used for companies to implement its own SC process flow.

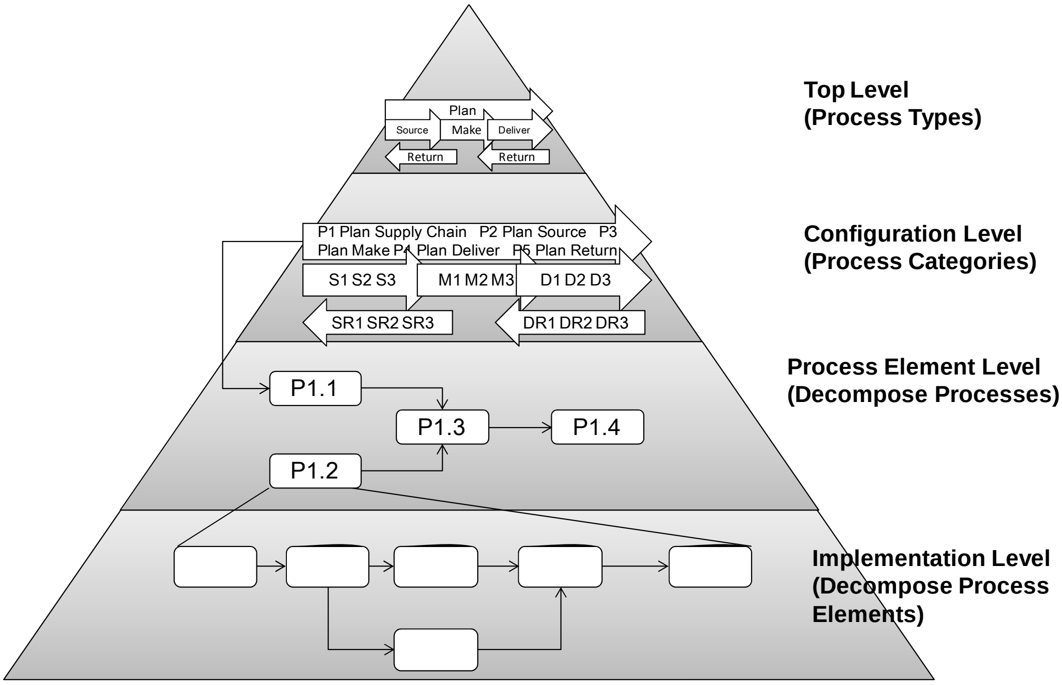


Figure 4.2 Structure model of SCOR

SCOR is designed for effective communication among supply chain partners. As a standard language, SCOR model helps SCM (Supply Chain Management), while as an industry standard，it also facilitates inter and intra supply chain collaboration, horizontal process integration by explaining the relationships between processes (i.e., Plan-Source, Plan-Make, etc.)[SCC,2001]. It specifies the following information:

(1) all customer interactions, from order entry through paid invoice;

(2) all physical material transactions, from supplier's supplier to customer's customer, including equipment, supplies, spare parts, bulk product, software, etc;

(3) all market interactions , from the understanding of aggregate demand to the fulfilment of each order;

(4) standardised process configuration model for supply chain process modelling;

(5) descriptions of best practices related to each process;

(6) standard measure metric for performance measures.

4.4 SCOR based Supply Chain Ontology

The dissertation focus on information flow among networked enterprises in supply chain environment, nevertheless the current SCOR operation reference model configures process focus on its physical supply chain process flow, not integration yet an “information view” [Millet, 2009] Therefore, in this chapter, we are trying to formalize an Ontology based on SCOR operation reference model to enrich it with semantic meanings.

4.4.1 Method of SCOR Supply Chain Ontology Development

As stated in the first chapter, the methods of ontology development frequently used currently include: IDEF5, Skeletal Methodology, TOVE, and METHONTOLOGY et al. No matter whichever method, it is restrained by some primitive rules: There is no method of ontology development that is ideal or most efficient. To judge the method is proper or not, depending on the development intention of ontology, knowledge domain of developers and whether the results accord with expectation. Ontology development must be a constantly updated process, and the concepts in ontology should accord with the description of objects (physical or logical) and relationships in the specific domain [Natalya , 2001].

According to the above rules, seven-step method of Protégé is adopted to develop the SCOR supply chain ontology based on SCOR in this chapter.

The main steps of developing SCOR supply chain ontology:

(1) Determine the domain and scope of ontology: The knowledge domain and knowledge scope of ontology should be ensured through confirming the scope, purpose and effect, users and maintainer of ontology and the information should be provided by ontology. The purpose of defining supply chain ontology in this chapter is to facilitate the enterprise interaction without any semantic barrier in the environment of supply chain. So the ontology should cover the information related to supply chain process, information carried by every supply chain process and

information about the interaction among enterprises et al, which mainly exist among enterprises rather than inside enterprise. Meanwhile the ontology is a domain ontology rather than upper ontology or common ontology. The problem about the scope of ontology can be solved by answering competency questions such as: How to build the supply chain process? What are the inputs and outputs of processes? How to evaluate the efficiency of supply chain? What are the metrics of performance attribute?

(2) Check if there is any ontology could be reused: With the developing of ontology research in various fields, lots of domain ontology bases have been built recently. Some of these ontologies are already mature enough and put into application, for example: Gene Ontology[10], which has been widely used in bioinformatics field. Reusing existing and mature ontology can contribute to the perfection and standardization of domain ontology, while improve the universality and develop efficiency of ontology.

(3) Determine the significant terms of ontology: it is the detail process of the ontology concepts, including class, attribute, relationship and et al. It is the most trivial and difficult step in ontology modeling, which calls for the participation of domain experts or developers qualified with extremely rich domain knowledge. Meanwhile, for the difference of developers' knowledge, the descriptions of the same concept may be different. Therefore, it is vitally necessary to adopt a wide, unified and standard knowledge base as the foundation of ontology concept, which meanwhile can improve the integrity, generality and reusability of ontology. In this chapter, SCOR is used as concept foundation of supply chain ontology, and the standard terms of SCOR are extracted to describe class, attribute and relationship of the ontology.

(4) Define class and the hierarchical relationship: Define the concept name of

[10] http://www.geneontology.org/

class and ensure the relationship between class and subclass. There are three ways: from top to bottom, from bottom to top and the mix of the two. Top-down approach means definiting starts from the most general concepts locating in the uppermost level, and then detailing the concepts gradually. Bottom-up approach refers to firstly defining the concrete concepts locating in the bottommost level, and then summarizing the upper concepts gradually. Mixing approach refers to the comprehensive use of top-down approach and bottom-up approach. Mixing approach is used to define class and the hierarchical relationship for the supply chain ontology in this chapter. Based on the structure of SCOR, the overriding classes and the hierarchical relationships among them are firstly defined by using top-down approach, while the detailed classification is summarized and generated by bottom-up approach.

(5) Define the property of class: The property of class expresses the particular characteristics different from other classes, including inner and outer properties of class itself and the relationship property between class and other classes, which mainly refer to the ObjectProperty and DatatypeProperty. In the supply chain ontology, the definition of property mainly comes from comprehending and extracting the hierarchical relationships of the pyramid structure in SCOR, while also including the relationships among business process, metrics, best practices and technique feature.

(6) Define the facet of property: This step includes the definitions of property feature, type of property value, property restriction, domain and range. Features of property refer to the property characters including inverseOf, TransitiveProperty, SymmetricProperty and FunctionalProperty and other features of property. Type of property value can be common data types (String, Boolean, Integer, etc) or entity type. Property restriction include the relationship between property value and class such as allValuesFrom and someValuesFrom, and the cardinality definition such as

minCardinality and maxCardinality. Domain and range of property limit the scope of property..

(7) Create instance: The instance of SCOR supply chain ontology will be presented in section 4.5 of this chapter.

During the practical building process, ontology development is usually a circulating process. Ni Yi-hua discussed and summarized the ontology process, and proposed an ontology development method facing the full life cycle as shown in Figure 4.3 [Ni Yi-hua, 2005]. It's a long-term process of accumulation and improvement from determing the domain and scope of ontology to ontology application, which needs domain experts to spend lots of time to build, and then cost time and energy for maintaining subsequently. At present, most of ways to create ontology are by hand, and creating ontology is far from becoming a kind of engineering activity. Each ontology development group has its own principle, design standard and development process. The absence of development method recognized and followed by all, is the main obstacle of ontology sharing, reusing and interoperating. The ontology development method adopted in the dissertation is an approach based on SCOR or saying an standard based approach, which largely reduces the work of collecting and determining for basic concepts and relationships when building ontology, and improves the development efficiency and generality of ontology immensely. As to the evaluation process of process, it's not included in the scope of the dissertation but could be the further research, so it will not be discussed here.

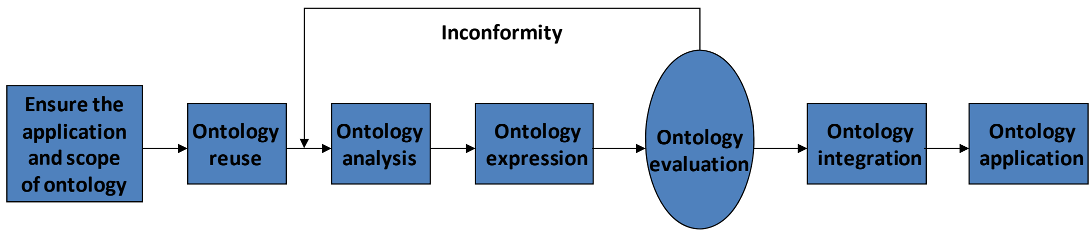
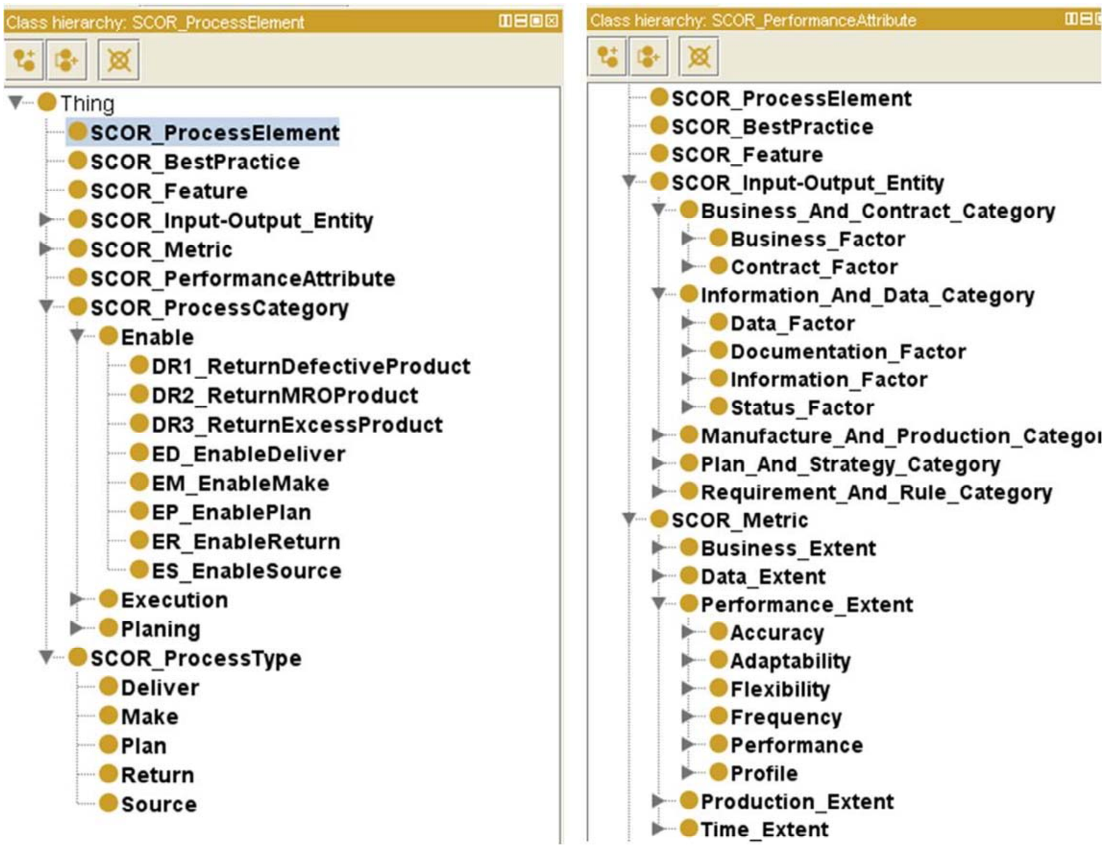


Figure 4.3 Modeling steps of ontology facing the full life cycle

4.4.2 Structure and semantic representation of SCOR Supply Chain Ontology

The dissertation adopted SCOR version 7.0 [SCOR, 2005] as the foundation of main concepts and structures, used Protégé 4.1[11] as the development tool and taked OWL DL as the ontology description language. SCOR supply chain ontology contains the three process hierarchies of the pyramid framework of SCOR: *SCOR_ProcessType*, *SCOR_ProcessCategory* and *SCOR_ProcessElement*, which are illustrated in Figure 4.4 (a). All of the inputs and outputs elements, performance attribute, metric, best practices and features et al related to process are shown in Figure 4.4 (b).



(a) (b)

Figure.4.4 Fragment of SCOR supply chain ontology related to (a) SCOR process (b) input/output entities

As the classes of SCOR supply chain ontology shown in Figure 4.4, *SCOR_ProcessType* represents the processes at level 1, including *plan*, *source*, *make*, *deliver*, *return*. *SCOR_ProcessCategory* represents level 2 processes' types: *Planning*

---

[11] http://protege.stanford.edu/

(including *P1_PlanSupplyChain*, *P2_PlanSource*, *P3_PlanMake*, *P4_PlanDeliver*, *P5_PlanReturn*), *Execution* (such as *D1_DeliverStockedProduct*, *M1_Make-to-Stock*, *S1_SourceStockedProduct*, *SR1_ReturnDefectiveProduct*, etc), *Enable* (such as *DR1_ReturnDefectiveProduct*, *ED_EnableDeliver*, *EM_EnableMake*, *ER_EnableReturn*, etc). *SCOR_ProcessElement* formalises the detail process element information for each Level 2 *SCOR_ProcessCategory*, such as the subdividing process of *P1_PlanSupplyChain*. Even if the SCOR operation reference model requires that any company should decompose its processes to more detailed levels (4 or more), we have limited our ontology to the level 3, because in a networked supply chain, the boundaries of the information flows are related only to interactions between companies. Thus, SCOR_ProcessElement is defined as the smallest process unit in our SCOR ontology.. For the subdividing for supply chain processes inside enterprise of the forth level, *SCOR_ProcessElement* is used to express the specific process unit. *SCOR_Input-output_Entity* is the aggregation of input and output element classes related to every *SCOR_ProcessElement*. Based on the definitions about every process element in SCOR vision 7.0, all the input and output elements of process are summarized, while hierarchical classification is made according to the contents of elements, such as *Business_And_Contract_Category*, *Information_And_Data_Category*, etc. *SCOR_PerformanceAttribute* represents the aggregation of performance evaluation attribute about supply chain process. *SCOR_Metric* is the aggregation of metric related to performance evaluation attribute. *SCOR_BestPractice* and *SCOR_Feature* express the aggregation of best practice and the aggregation of technical features of SCOR process respectively.

Meanwhile, a series of properties are defined to describe the relationships among classes in SCOR supply chain ontology, as shown in Figure 4.5. The property *is_configured_of* and *is_decomposed_of* are used to express the hierarchical relationships among *SCOR_ProcessType*, *SCOR_ProcessCategory* and *SCOR_ProcessElement*. For presenting the process of supply chain, the property *proceed_by* is defined to combine instance of *SCOR_ProcessElement* instance. Every *SCOR_ProcessElement* expresses the input and output elements of process units by property *has_input* and *has_output* combining with *SCOR_Input-output_Entity*. In the meantime, *SCOR_ProcessType*, *SCOR_ProcessCategory* and *SCOR_ProcessElement* all have the property *has_performanceattribute* expressing that the efficiency of process operation could be measured by *SCOR_PerformanceAttribute*. *SCOR_ProcessCategory* and *SCOR_ProcessElement* can express the best practice by *has_bestpractices*.

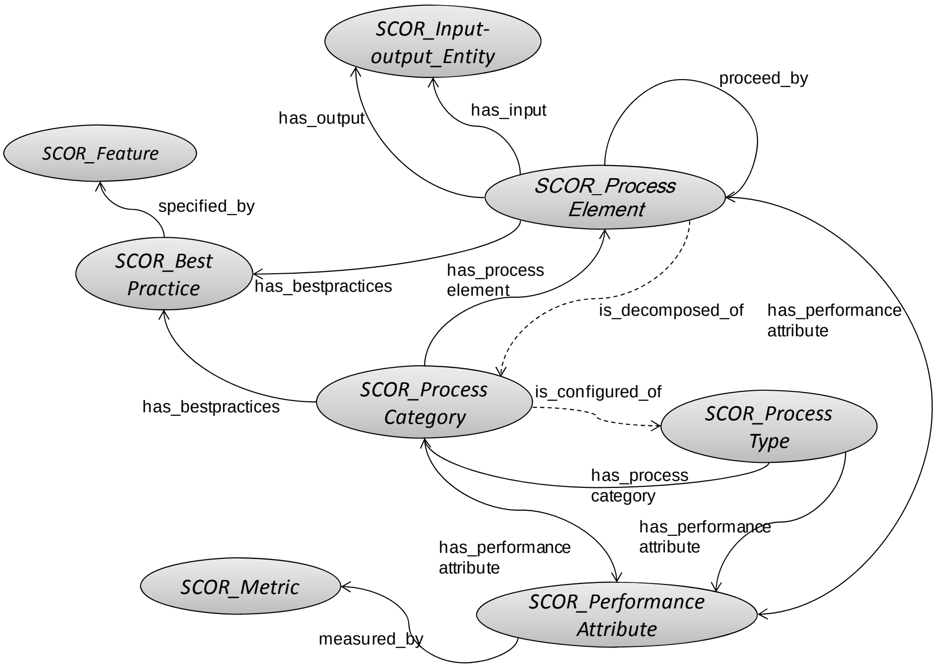


Figure 4.5 Part of property relationship of the SCOR supply chain ontology

SCOR supply chain ontology is represented by OWL DL, having explicit conceptual meaning, consistent and precise semantics.. OWL is the Semantic Web ontology language developed by W3C [Horrocks et al., 2003], providing three sub languages: OWL Lite, OWL DL and OWL Full. Thereinto, OWL DL has the strongest expressing ability, and was designed to support the existing Description Logic business segment and has desirable computation properties for reasoning systems. The ontology language OWLDL is based on Description logics (DL), which is a family of formal knowledge representation languages [Baader. 2003]. The above ontology model can be expressed by the following equivalent axioms in DL. For example:

Axiom 1: *SCOR_ProcessType* is the subclass of *Thing*, while All values of the *has_processcategory* of the *SCOR_ProcessType* class is the member of the class

SCOR_ProcessCategory.

$$SCOR\_ProcessType \sqsubseteq Thing \sqcap \forall has\_processcategory.SCOR\_ProcessCategory$$

Axiom 2: All value of the *has_processelement* of the *SCOR_ProcessCategory* class is the member of the class *SCOR_ProcessElement*.

$$SCOR\_ProcessCategory \sqsubseteq \forall has\_processelement.SCOR\_ProcessElement$$

Axiom 3: *Plan*, *Source*, *Make*, *Deliver* and *Return* are all subclasses of *SCOR_ProcessType*, and they are disjoint.

$$Plan \sqsubseteq SCOR\_ProcessType$$

$$Source \sqsubseteq SCOR\_ProcessType$$

$$Deliver \sqsubseteq SCOR\_ProcessType$$

$$Make \sqsubseteq SCOR\_ProcessType$$

$$Return \sqsubseteq SCOR\_ProcessType$$

$$Plan \sqcap Source \sqcap Deliver \sqcap Make \sqcap Return \equiv \perp$$

Axiom 4: All the members of *SCOR_ProcessElement* having the property *has_input* and *has_output* belong to *SCOR_InputOutput_Entity*.

$$SCOR\_ProcessElement \sqsubseteq \forall has\_input.SCOR\_InputOutput\_Entity$$

$$SCOR\_ProcessElement \sqsubseteq \forall has\_output.SCOR\_InputOutput\_Entity$$

Axiom 5: All value of the *proceed_by* of the *SCOR_ProcessElement* is a member of itself..

$$SCOR\_ProcessElement \subseteq \forall proceed\_by.SCOR\_ProcessElement$$

Axiom 6: Object property *is_configured_of* is an inverse of *has_processcategory* object property.

$$is\_configured\_of \equiv has\_processcategory^{-}$$

4.5 Process Instance of Supply Chain Ontology

In this section, we give an example from the main body manufacturing process of double column CNC guideway&surface grinding machine in Machine Tool Company, to explain the building process of instance of SCOR supply chain ontology. As grinding machine is usually made by order, the structure model of SCOR

Make-to-Order (level 2) process is built. The structure model of supply chain process inside enterprise is illustrated in Figure 4.6.

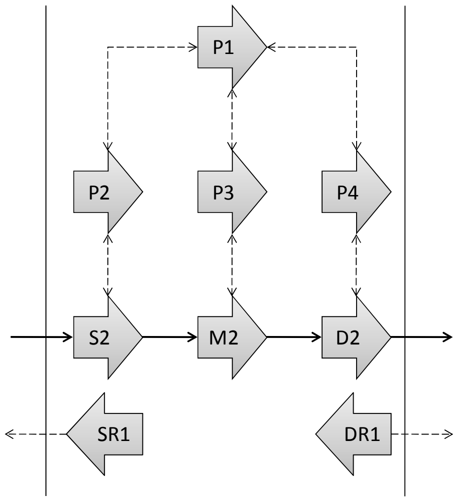


Figure 4.6 Structure model of internal supply chain process (SCOR level 2)

Based on the structure model of the second level of supply chain, the supply chain can be furthermore resolved into process elements according to the standard process definition of SCOR. M2 (Make-to-Order) process is picked up as an example to subdivide into supply chain process units as shown in Figure 4.7. This supply chain process expresses the detailed manufacturing process of main machine bed of double column CNC guideway&surface grinding machine, from accepting the order of customer to completing the bed manufacturing by the enterprise. The process of manufacturing and testing are then subdivided into process units such as casting, scribing, primary planing, primary milling, ageing treatment, scribing and finishing planing et al.

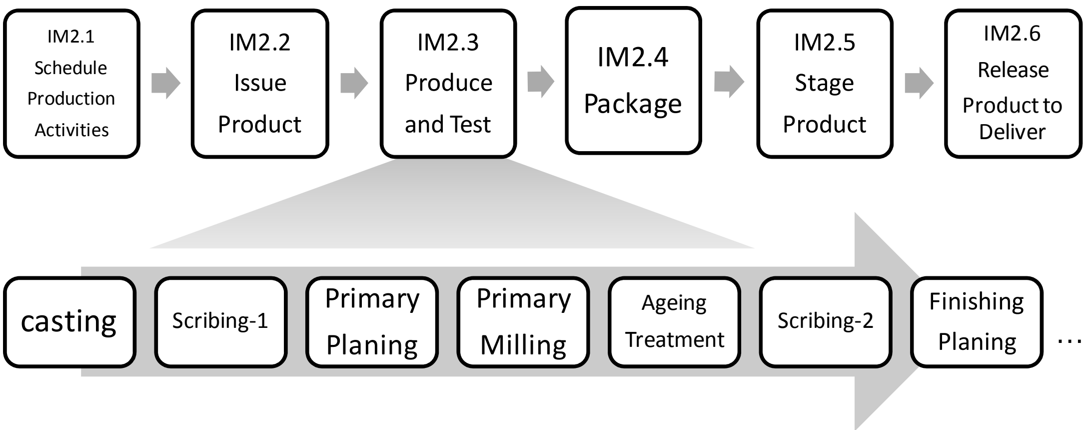


Figure 4.7 A detailed Make-to-Order process element of main mainchin bed

Based on the detailed supply chain process of Make-to-Order shown in Figure 4.7, the instance of SCOR supply chain ontology is built in Protégé shown in Figure 4.8. The instance contains the whole manufacturing process of M2 and the input and output elements of every process unit, for example, the process of *Schedule_Production_Activities* has the inputs of *Production_Plan*, *Scheduled_Receipts* and *Information_Feedback* et al, while having the output of *Production_Schedule*.

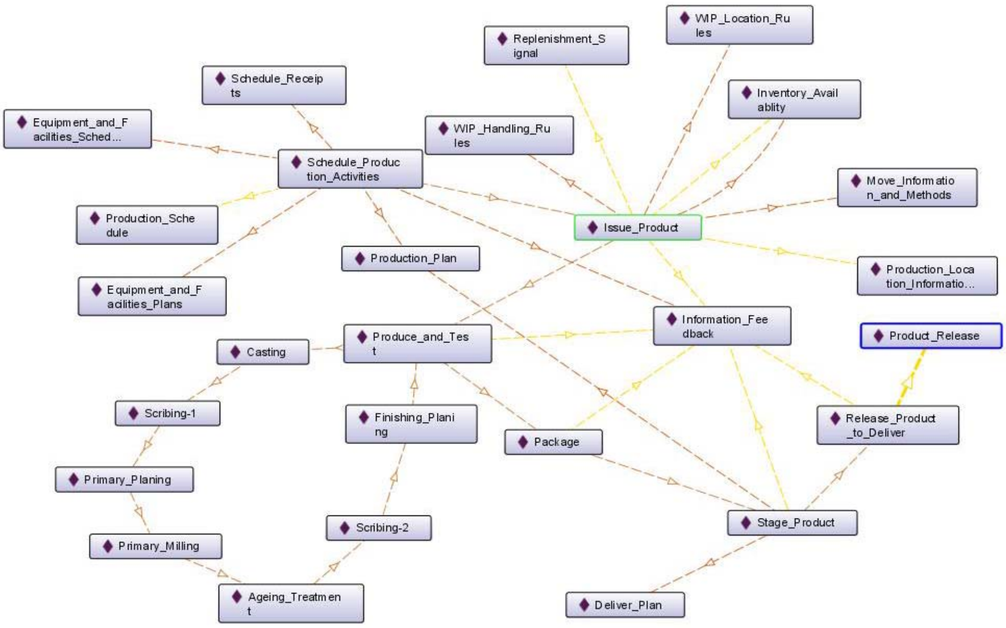


Figure 4.8 Instance of SCOR supply chain ontology of Make-to-Order in Protégé

4.6 Performance Measure based on SCOR Supply Chain Ontology

The SCOR model can help managers organize, analyse and evaluate the whole supply chain process. And performance evaluation system is one important component part of it, providing a standard approach to evaluate the execution of supply chain and then assisting managers solve problems. The SCOR supply chain ontology built in this chapter contains the concepts and relationships related to performance measure, and has primary ability of performance measure. For the measure of supply chain process is not included in the research scope of the dissertation, here we only presents some simple analysis of the performance measure of supply chain.

Performance measure was defined as the quantization of effectiveness and efficiency by Neely et al. Effectiveness is the extent to which a customer's requirements are met and efficiency measures how economically a firm's resources are utilised when providing a pre-specified level of customer satisfaction. Performance measurement systems are described as the overall set of metrics used to quantify both the efficiency and effectiveness of action [Neely et al., 1995]. Lots of researchers in engineering and management field have researched the methods of performance management recently, such as [Lindner, 2009] [Shepherd and Gunter, 2006] [Lockamy and McCormack, 2004] and so on, who studied supply chain performance measurement in the aspects of operation, design and strategy et al. SCOR_Metric which is aggregation of elements related to SCOR_PerformanceAttribute has been defined in detail in SCOR supply chain ontology, provides the metrics and methods of measuring performance.

The measurement process of SCOR supply chain ontology can be explained by the supply chain process inside an enterprise presented in section 4.5 for example. The dissertation selected two performance measurement indexes as instance: Order Fulfillment Cycle Time and Supply Chain Management Cost (SCMC).

The corresponding calculation methods are shown as follows:

Formula 1:

$$\text{Order Fulfillment Cycle Time} = \text{Order Fulfillment Process Time} + \text{Order Fulfillment Dwell Time}$$

Formula 2:

$$\text{Supply Chain Management Cost} = \text{Cost to (strategize and)Plan} + \text{Cost to Source} + \text{Cost to Deliver} + \text{Cost to Return}$$

According to the supply chain process P2-M2 involved in the supply chain inside an enterprise illustrated in Figure 4.7, the performance measurement metrics related to Order Fulfillment Cycle Time and Supply Chain Management Cost in every process have been summarized in Table 4.1.

Table 4.1 Summary of performance measurement metrics related to supply chain process

	Order Fulfillment Cycle Time	Supply Chain Management Cost（SCMC）
P2	None Identified	Total Source Costs % of (S+M+D) Costs Material planning costs % of purchased material
P3	None Identified	Total Make Costs % of (S+M+D) Costs Plan Make Costs % of Make Costs
P4	Order Management Cycle Time	Total Deliver Costs as % of (S+M+D) Costs Plan Deliver Costs as % of Deliver Costs
S2	Source Cycle Time Schedule Product Deliveries Cycle Time Receive Product Cycle Time Verify Product Cycle Time Transfer Product Cycle Time Authorize Supplier Payment Cycle Time	Product Acquisition Costs as % of Source (S2) Costs Schedule Deliveries Costs as a % of Product Acquisition Costs Receiving costs as a % of Product Acquisition Costs Verification costs as a % of Product Acquisition Costs Transfer & Product storage costs as a % of Product Acquisition Costs Cost per type of invoice
M2	Make Cycle Time Schedule Production Activities Cycle Time Issue Sourced/In-Process Product Cycle Time Produce and Test Cycle Time Package Cycle Time Stage Finished Product Cycle Time Release Finished Product To Deliver Cycle Time	None Identified
D2	Deliver Cycle Time Receive, Configure, Enter and Validate Order Cycle Time Reserve Resources and Determine Delivery Date Cycle Time Consolidate Orders Cycle Time Build Loads Cycle Time Route Shipments Cycle Time Select Carriers and Rate Shipments Cycle Time Receive Product from Make/Source Cycle Time Pick Product Cycle Time Pack product cycle time Load Vehicle and Generate Shipping Documentation Cycle Time Ship Product Cycle Time Receive and Verify Product Cycle Time Install Product Cycle Time Invoice Cycle Time	Order Management Costs as % of Deliver cost Cost / type of Inquiry & Quote Order cost / type of order Order Entry and Maintenance Costs as % of (S+M+D) cost Finished Goods Inventory Days of Supply Transportation Costs Receiving costs / type of receipt Product Picking Cost / type of order Loading costs / type of load Receive & Verify cost / type of receipt Cost of non-conformance Product Install cost / type of installation Customer Invoicing/Accounting Costs

4.7 Conclusion

In this chapter, we analysed the standards for supply chain integration, and enriched the semantic supply chain process and related information through building an supply chain ontology. The SCOR supply chain ontology was based on SCOR model, and followed the seven-step method of Protégé. The OWL DL was used to represent semantic meanings. Then an instance was illustrated on the make-to-order supply chain process of main machine bed. At the end, the performance measurement of SCOR supply chain was simply explained.

5 Ontology based System Interoperability in Supply Chain

5.1 Introduction

Semantic interoperation is usually realized by the matching between ontologies. The semantic interoperation in the dissertation was completed by two parts of ontology matching, including the matching among ontology bases and the mapping between ontology and external database. As ONTO-PDM product ontology and SCOR supply chain ontology are from two domains, and are developed separately. To construct a common shared product centric supply chain ontology, the mapping relationships between them should be discovered. Besides, semantic heterogeneity also exists between external database and product centric supply chain ontology, and the mapping relationships need to be discovered by matching.

5.2 Basic Theory of Ontology Matching

5.2.1 Concept of Ontology Matching

Ontology mapping, ontology matching, ontology merging and ontology aligning are all the methods of ontology processing, and many researches usually confuse with them. Based on the definitions from articles [Bruijn, 2006][Giunchiglia, 2004][Rahm, 2001] [Ding, 2002], we give a explanation about the four concepts in order to express the ontology processing method more clearly in the dissertation:

(1) Ontology mapping is to build the similarity relationship between two ontologies that are independent. The result of ontology mapping is set of semantic overlap relationships expressed by a independent and clear form which can be the axioms and formulas described by specific mapping languages. Mapping relationship doesn’t belong to any ontology, and can be used for heterogeneous knowledge bases query and transformation among different data structures et al.

(2) Ontology aligning refers to the process about discovering the similarity

relationship between ontologies automatically or semi-automatically, to make the ontologies reuse the information of each other. The result of ontology aligning is the expression about the similarity among ontologies, and also can be regarded as a kind of ontology mapping. Generally, ontology aligning is also called as ontology matching.

(3) Ontology merging, which can also be called as ontology integration, refers to building a new and single consistency ontology based on the resources of two or more existing ontologies. This new ontology integrates all the knowledge from resource ontologies, and usually request quite large adjustment and expansion. The difficulty of ontology merging is that the new merging ontology must reflect all the similarity and otherness among resource ontologies. Ontology merging is a complex and difficult process. The overlap parts among ontologies need be found, connect the semantic relevant concepts by relationships such as *subClassOf*, *equality*, *unionOf*, and *complementOf* et al. The new ontology must be consistency and continuity and removed the redundancy of concepts. The difference between ontology merging and ontology aligning is illustrated in Figure 5.1.

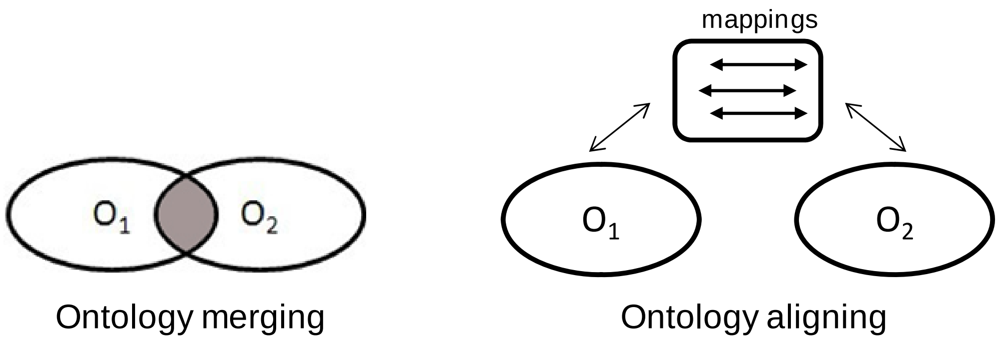


Figure 5.1 The difference between ontology merging and ontology aligning

For the difference exists in ontologies, the situation that ontologies are not matched often appears when ontologies interaction. The reasons causing ontologies unmatched are various, such as knowledge domains diversity of ontology, autonomy of ontology modeling and difference of description languages et al. Distinguishing different cases and acknowledging mismatch reasons clearly are the premise to deal with specific ontology mapping. According to some articles such as [Pepijn, 1997] [Klein, 2001], ontology mismatch case can be divided into following types, as shown in Table. 5-1.

Table. 5-1 The explanation of ontology mismatching case

<table>
<tr><td rowspan="7">Ontology layer</td><td rowspan="2">Conceptualization mismatch (different concepts in the same domain)</td><td colspan="2">Scope mismatch (classes have the same expansion intersection such as instance, but the concepts of class are different)</td></tr>
<tr><td colspan="2">Model coverage and granularity mismatch (the covering fields of ontology or the detailing extent of model are different)</td></tr>
<tr><td rowspan="5">Explication mismatch (difference about the ways of concept expression)</td><td rowspan="2">Style of modeling mismatch</td><td>Paradigm (the expression paradigms adopted by the same concept are different)</td></tr>
<tr><td>Concept description (the expressions on concept hierarchy are different)</td></tr>
<tr><td rowspan="2">Terminologcial mismatch</td><td>Synonym (the same object have different names, such as “car” and “automobile”)</td></tr>
<tr><td>Homonym (such as “conductor”, referring to “command” in music field but “conuctor” in electronic field)</td></tr>
<tr><td colspan="2">Encoding mismatch (the coding style of data in different ontologies are different, such as dd/mm/yyyy and mm-dd-yy)</td></tr>
<tr><td>Language layer</td><td colspan="3">The difference of grammar, logic expression, original semantic and language expressivity (using different ontology description languages, such as LOOM and RDF)</td></tr>
</table>

As to the ontology mappings discussed in the dissertation, the ontologies are both use OWL in language layer, so the mismatching condition of grammar or logic expression do not exist. The ontology mismatching mainly exists in ontology layer: model coverage mismatching, terminologcial mismatching and style of modeling mismatch Model coverage mismatching: ONTO-PDM product ontology mainly concerns the concepts related to product data, while the concepts related to process as complement; SCOR supply chain ontology is mainly about the description of supply chain process, but inevitably involves the related information of product during the supply chain process. Therefore, the two ontologies have the intersection in scope. Terminologcial mismatching: the concept *EquipmentType* in ONTO-PDM product ontology and the concept *Equipment* in SCOR supply chain ontology, the concept *ProductionCapabilityType* in ONTO-PDM product ontology and the concept *Production-Capability* in SCOR supply chain ontology, although they are expressed by different words, the concepts are the same, which is just the synonym case. Style of modeling mismatch: the concept *EquipmentType* in ONTO-PDM product ontology is the subclass *Thing*, while the concept *Equipment* in SCOR supply chain ontology is the subclass of *SCOR_Input-output_Entity*, which reflects the difference on the description of concept hierarchy.

5.2.2 Pretreatment of Ontology

No matter the ontology mapping between ONTO-PDM product ontology and SCOR supply chain ontology or between sharing ontology and database, the basic ontology pre-treatment is required for removing some factors that may affect the result of semantic matching, in order to improve the accuracy of automatic matching algorithm and the efficiency of semantic matching. Two ontology pretreatment methods were adopted in the dissertation: the standardization of character string and the standardized prototype of word. The two methods are both the basic ontology matching methods, while the former is mainly used for word treamtment, and the latter is mainly used for phrase treamtment..

The standardization of character string includes two aspects: ① remove the insignificant marks in the name of class and attribute such as hyphen, underline and punctuation et al. For example, change *SCOR_Input-Output_Entity* into *SCOR Input Output Entity*. It can be realized automatically by computer programming. ② expand the abbreviation in the names of class and attribute. For example, expand the concept *SKU* and *MRO* in SCOR supply chain ontology to *Stock Keeping Units* and *Maintenance Repair and Operations* respectively. The abbreviations in ontologies are few, so this work is completed by manual.

The standardized prototype of word is based on the lexicon of WordNet[12] and done automatically, including: ① Restore the part of speech of word . Restore the various forms of word such as past tense, present participle and plural form et al to prototype. For example, restore *Payments* to *Payment* and restore *Revised_Capital_Plan* to *Revise Capital Plan*. ② Remove the quantifiers, prepositions and adverbs that have no semantic meaning such as “a”, “of” and “the” et

---

[12] http://wordnet.princeton.edu

al. For example, change *Bill_of_Materials* into *Bill Material*.

5.3 Mapping between ONTO-PDM Product Ontology and SCOR Supply Chain Ontology

5.3.1 Strategy of Ontology Matching

ONTO-PDM product ontology mainly describes the technical data related to product manufacturing, and SCOR supply chain ontology mainly contains the information about process of product supply chain. The two ontologies are the knowledge sets about different fields, meanwhile the two ontologies are developed independently. ONTO-PDM product ontology is based on some international standards such as IEC 62264 and ISO 10303 STEP-PDM, and SCOR supply chain ontology takes Supply-Chain Operations Reference-model as the foundation. To acquire a common shared product-centric supply chain ontology, the mapping relationship between the above two ontologies need be established. The ontology matching method based on the combination of terminology, structure, semantic and extend technology is adopted in the dissertation to realize ontology merging.

Reviewing the content of SCOR supply chain ontology and ONTO-PDM product ontology, although they are two separate ontologies concerning different aspects, SCOR Ontology is specifically used for capturing complex supply chain management process into a standard process model, while ONTO-PDM is designed for integrating all relevant information related to products lifecycle. However, there are still some common concepts between them. And the concepts have obvious linguistic and terminological commons between the two ontologies can be sensibly distinguished. Table. 5-2 list a fragment of relevant concepts that have semantic correspondences.

Table. 5-2 Fragment of relevant concepts between ONTO-PDM and SCOR

Concepts in ONTO-PDM Product Ontology	Concepts in SCOR Supply Chain Ontology
ProcessSegmentType	*ProcessType/Process_Procedures*
ProductDefinitionType	*Product*
StatusType	*Status*

ProductionPerformanceType	*Performance*
CapabilityType	*Capability*
ProductionScheduleType	*ProductionSchedule*
PersonnelRequirementType	*Requirement*
ProductProductionRuleType	*Production_Rules*
MaterialType	*Material*
TimeType	*Time*
ProductionCapabilityType	*Production_Capability*

The properties contained in SCOR supply chain ontology are very few, thus the matching between the two ontologies are mainly based at concept level. The process of semantic matching is expressed as *f*, the mapping relationship among concepts is expressed as *A*, and the ontology to be matched is expressed as *O*, so the ontology mapping can be expressed as:

$$A = f(o, o', r, p) \tag{1}$$

The dissertation adopted the method combining semantic similarity algrithem with rule based inference to discover the mapping relationship between the two ontologies, with the help of the WordNet lexicon. Semantic similarity algrithem based on WordNet has good effect on discovering the semantic similarity among concept names, while inference based on rules can discover the implicit relationships between ontologies, which is a complement for semantic similarity algrithem method. The detail ontology merging process is mainly by the following steps shown as Figure. 5.2:

(1) Do pretreatment to the class concepts of ontologies, according to the method in Section 5.2.2;

(2) Discover the mapping relationships *A* between concepts by calculating semantic similarity algrithem;

(3) Formalise these concepts with DL;

(4) Identify SWRL mapping rules based on instance;

(5) Compute inference concepts using Pellet inference engine, and acquire the mapping relationship *A'*;

(6) Ontology merging based on the mapping relationships.

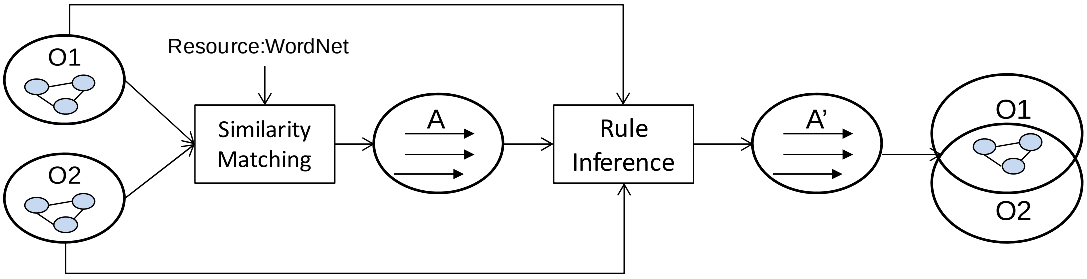


Figure. 5.2 ontology mapping discover strategy

According to the above six steps, the ontology merging process was mainly explained in three aspects: semantic similarity algrithem, rules based reasoning and the merging of mapping results.

5.3.2 Ontology Matching Process

5.3.2.1 Ontology Matching based on Semantic Algorithm

Sensibly, some obviously similar concepts can be found existing in SCOR supply chain ontology and ONTO-PDM product ontology, which are shown in Table. 5-2. Because the two ontologies contain thousands of classes and properties, it's difficult to be done by manual. We adopt WordNet semantic algrithem method based on perfect matching principle[13] to discover the concept correspondences between ontologies by calculating the similarity between two concepts.

The concepts in SCOR supply chain ontology and ONTO-PDM product ontology are mainly appeared in the form of phrase or word group, that is not good to calculate edit distance to get the semantic similarity between them directly. So in the dissertation, we adopt computer word segmentation method to divide concept names into word sets, expressed as $C = \{c_i | i = 1,2,3 \dots m\}$, after the pretreatment to ontologies. According to perfect matching principle, take the maximum semantic similarity value of word sets as the semantic similarity between words, and then define the semantic similarity $SIM(C_1, C_2)$ between two concepts as the average value of the sum of maximum semantic similarity value of word sets, with the computational formula as follow:

[13] http://en.wikipedia.org/wiki/Perfect_matching

$$SIM\left(C_{1,}C_{2}\right)=\left\{\sum_{i=1,2\ldots,m}\max_{j=1,2\ldots n} sim_{c}(c_{i,}c_{j})\right\}/m \tag{2}$$

In the formula, $sim_c(c_{1,}c_2)$ is a kind of semantic similarity algrithem based on WordNet; C_1, C_2 are the word sets of the concepts in two ontologies after word segmentation; m, n express the word number of word sets repectively; c_i, c_j express the single word of word sets repectively .

$sim_c(c_{1,}c_2)$ is the semantic similarity algrithem based on WordNet proposed by Jiang-Conrath [Jiang, 1997]. WordNet is a huge corpus, where nouns, verbs, adjectives and adverbs compose synsets, and then synsets form the tree structural word hierarchy system according to the concept semantic and vocabulary relationships. The special structure of WordNet provides a very effective method for automaticly language processing. Based on the word hierarchy system of WordNet, Jiang-Conrath algorithm combined vocabulary classification structure with corpus statistical information, with the formula shown as follow:

$$sim_{c}\left(c_{1},c_{2}\right)=2\log p\left(c_{0}\right)-\log p(c_{1})+\log p(c_{2}) \tag{3}$$

In the formula, c_1, c_2 express the two words requesting similarity calculation; c_0 is the nearest common hypernym of c_1 and c_2; $p(c)$ is the probability appearing in specific corpus.

Jiang-Conrath algorithm combined with two common methods for calculating semantic similarity, which is the method based on distance and the method based on information content. The method based on distance mainly utilizes the hyponymy between concepts, that means closer the distance between two concepts higher the similarity. The method based on information content is the complement for the method based on distance, and is based on information theory. It less depends on the hierarchical structure of terminology, but focuses on measuring the extent of closeness of the relationship between superclass and subclass. Budanitsky et al adopted three indexes including accuracy, recall and F measurement to make overall evaluation on five kinds of semantic similarity algorithms based on WordNet, and the experimental result also indicated that Jiang-Conrath algorithm had preferable matching effect.

According to Formula (3), calculate the concepts similarity between the two ontologies. Based on the calculation result, we adopt MDS[14] method to show the distribution of the similar concepts between SCOR supply chain ontology and ONTO-PDM product ontology, which is shown in Figure 5.3. MDS method uses distance to express the similarity degree between concepts, while the concepts with small distance have the big similarity degree, and the concepts with great distance have the little.

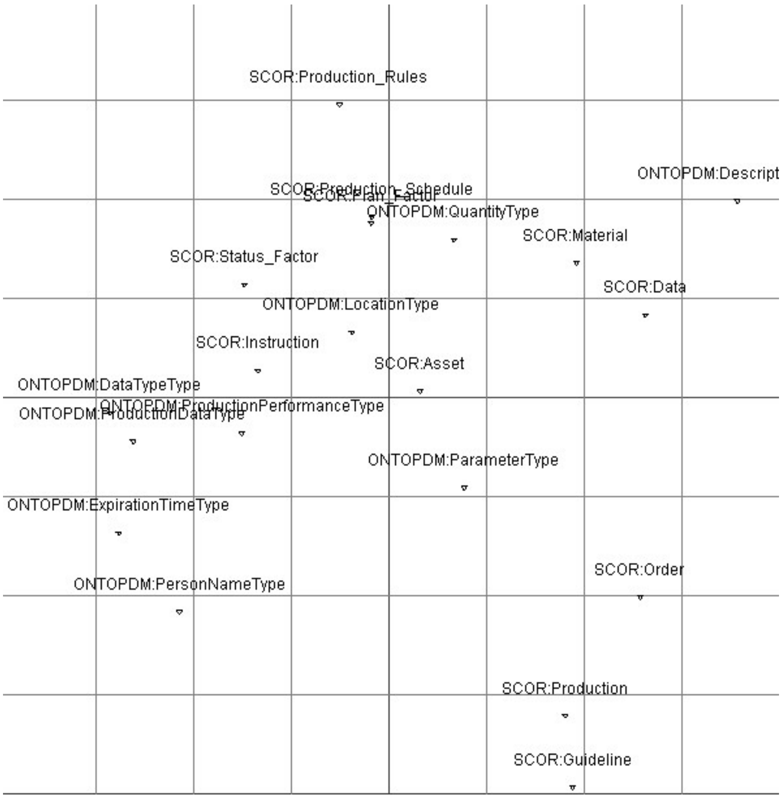


Figure 5.3 Fragment of distribution of ontology similar concepts

Based on the above calculation result, the following concept mapping relationships *A* can be acquired:

$$ActualEndTimeType \subset Time \subset SCOR\_Input\_Output\_Entity$$
$$ActualFinishTimeType \subset Time \subset SCOR\_Input\_Output\_Entity$$
$$ActualStartTimeType \subset Time \subset SCOR\_Input\_Output\_Entity$$
$$PublishedDateType \subset Date \subset SCOR\_Input\_Output\_Entity$$
$$RequestedCompletionDateType \subset Date \subset SCOR\_Input\_Output\_Entity$$
$$CapabilityType \subset Production\_Capability \subset SCOR\_Input\_Output\_Entity$$
$$ResponseStateType \subset Status \subset SCOR\_Input\_Output\_Entity$$
$$RequestStateType \subset Status \subset SCOR\_Input\_Output\_Entity$$
$$StorageLocationType \subset Location \subset SCOR\_Input\_Output\_Entity$$

[14] http://en.wikipedia.org/wiki/Multidimensional_scaling

$$ProductionPerformanceType \subset Performance \subset SCOR\_Input\_Output\_Entity$$
$$ProductProductionRuleType \equiv ProductionRules \subset SCOR\_Input\_Output\_Entity$$
$$LocationType \supset Location$$

5.3.2.2 Ontology Matching based on Rule Inference

Semantic similarity algrithem only concerns about the language similarity between the terminology and structure of ontology concepts, without considering the constraint of ontology properties, which is difficult to find some connotative relationships. The ontology matching method based on rules is a complement for above semantic similarity algrithem. This method adopts the mathematical method of Category theory, using Semantic Web Rule Language (SWRL) to define ontology mapping rules by manual. Based on the defined SWRL mapping rules, through Pellet[15] inference engine, the similar or same ontology concepts can be infered automatically, and then reclassified semi-automatically. This method can find the core concepts of model, and help find the related concepts between two ontologies.

Before defining the reasoning rules, the concepts should be formally described firstly. We adopted description logic DL to express, and Table 5-3 shows part of the formalized DL expression of ontology concepts.

Table 5-3 DL formal expression of concepts

ONTO-PDM: *ProcessSegmentType/ProductSegmentType* (define the attributes and relationships of *ProcessSegmentType* and *ProductSegmentType*) $ProcessSegmentType \subseteq \forall Description.DescriptionType$ $ProcessSegmentType \subseteq \forall Duration.DurationType$ $ProductSegmentType \subseteq \forall Description.DescriptionType$ $ProductSegmentType \subseteq \forall Duration.DurationType$ $ProcessSegmentType \subseteq \forall correspond\_to.ProductSegmentType$ $ProcessSegmentType \subseteq \forall is\_composed\_of.ProcessSegmentType$
SCOR: *SCOR_ProcessElement* (*SCOR_ProcessElement* is the subclass of *SCOR_ProcessCategory*, and only have the attribute relationships of input and output) $SCOR\_ProcessCategory \subseteq \forall has\_processelement.SCOR\_ProcessElement$ $SCOR\_ProcessElement \subseteq \forall connectBy.SCOR\_ProcessElement$ $SCOR\_ProcessElement \subseteq \forall has\_input.SCOR\_InputOutput\_Entity$ $SCOR\_ProcessElement \subseteq \forall has\_output.SCOR\_InputOutput\_Entity$

SWRL is proposed as a semantic web rule language combining OWL DL and

15 http://clarkparsia.com/pellet

RuleML (Rule Markup Language) [Horrocks I., 2004]. SWRL rules are really a kind of OWL axiom, which can make for more expressive property and class axioms, and and be applied to semantic transformation within inference engines. After analyzing of the two ontologies, we then postulate some mapping rules between concepts from both ontologies by SWRL, shown as follows:

Rule 1.

$SCOR{:}SCOR\_InputOutput\_Entity\ (?\,y)\ , SCOR\_InputOutput\_Entity\ (?\,z)\ , SCOR\_ProcessElement(?\,x1)\ ,$

$SCOR\_ProcessElement(?\,x2)\ , has\_input(?\,x1, ?\,y)\ , has\_output(?\,x1, ?\,z)\ , proceed\_by(?\,x1, ?\,x2)$

$\rightarrow ONTOPDM{:}ProductSegmentType(?\,x1)$

Rule 2.

$SCOR{:}SCOR\_InputOutput\_Entity\ , SCOR\_ProcessElement(?\,x1)\ ,$

$SCOR\_ProcessElement(?\,x2)\ , has\_output(?\,x1, ?\,y)\ , proceed\_by(?\,x1, ?\,x2)$

$\rightarrow ONTOPDM{:}ProductSegmentType(?\,x1)$

Rule 3.

$SCOR{:}SCOR\_InputOutput\_Entity\ (?\,y)\ \ ), SCOR\_ProcessElement(?\,x1)\ ,$

$SCOR\_ProcessElement(?\,x2)\ , has\_input(?\,x1, ?\,y)\ , proceed\_by(?\,x1, ?\,x2)$

$\rightarrow ONTOPDM{:}ProductSegmentType(?\,x1)$

Rule 4.

$ONTOPDM{:}ProductionScheduleType(?\,x), ProductionRequestType(?\,y),$

$has\_productrequest(?\,x, ?\,y) \rightarrow SCOR{:}Production\_Schedule(?\,x)$

Rule 5.

$ONTOPDM{:}ProductDefinitionType(?\,x), ProductSegmentType(?\,y), productsegment(?\,x, ?\,y),$

$ManufacturingBillType(?\,z), manufacturingbill(?\,x, ?\,z), ProductInformationType(?\,q),$

$has\_productdefinition(?\,x, ?\,q) \rightarrow SCOR{:}Production\_Plans(?\,x)$

Rule 6.

$ManufacturingBillType(?\,x), has\_manufacturingbill(?\,x, ?\,y), ProductDefinitionType(?\,y)$

$\rightarrow Bill\_of\_Materials(?\,x)$

SWRL is the reasoning method based on instance, deducing implication relationships between ontologies by the reasoning classification to instance. The make-to-order supply chain process of machine tool company in section 4.5 is set as an instance to explain Pellet reasoning. The instance of make-to-order process in Protégé is shown in Figure 5.4. According to above SWRL rules, the results of Pellet reasoning is shown in Figure 5.5.

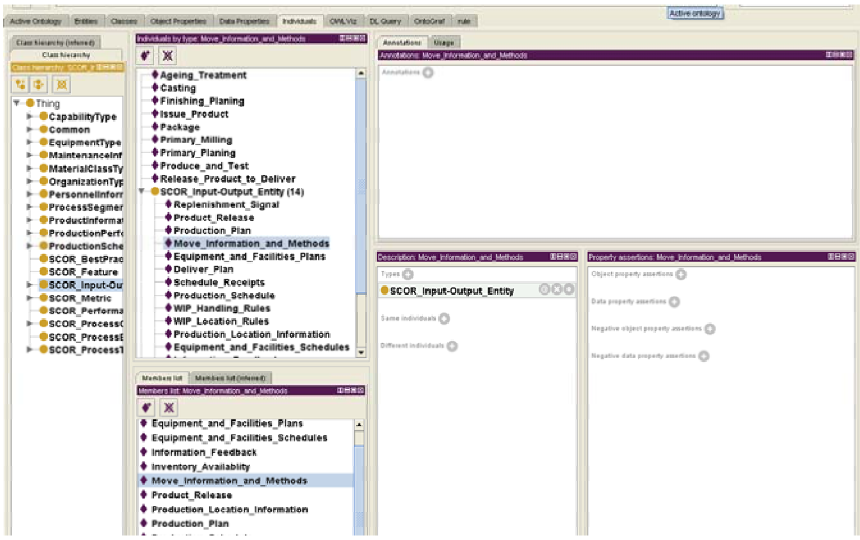


Figure 5.4 The instance of make-to-order process in Protégé

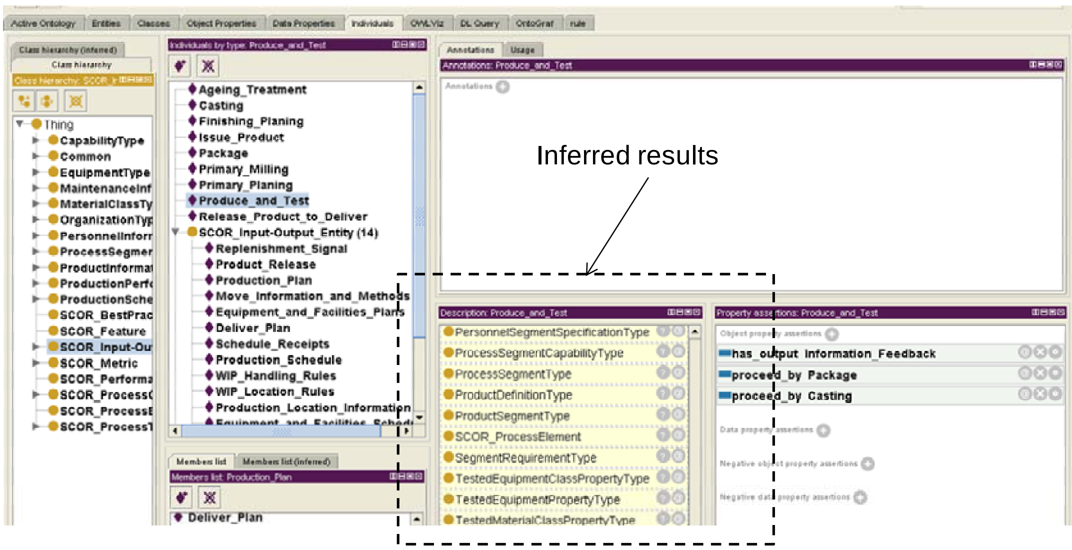


Figure 5.5(a) Result of Pellet reasoning

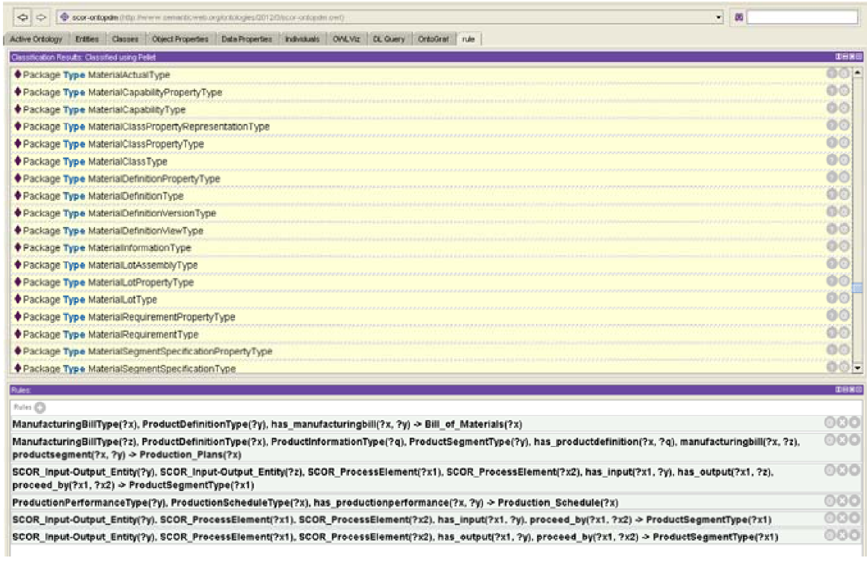


Figure 5.5(b) Result of Pellet reasoning

Through the analysis about above reasoning results, the following mapping relationships *A'* can be acquired:

$$SCOR\_ProcessElement \equiv ProductSegmentType$$

$$Production\_Plan \equiv ProductDefinitionType$$

$$ProductScheduleType \equiv Production\_Schedule$$

$$StatusType \supset Status$$

$$ManufacturingBillType \equiv Bill\_of\_Material$$

5.3.2.3 Ontology Merging

To build a common shared product-centric supply chain ontology, the ontology merging is based on the mapping relationships *A* and *A'* obtaining from semantic similarity calculation and rule reasoning. Ontology merging is in the environment of Protégé 4.1, and the followed the following principles: by adding the relationships of *equivalentClassOf* and *subClassOf* to reclassifiy the correspondence concepts; the properties and constraint relationships of class follow the moving of class; select the mapping relationship of superclass, when the superclass and subclass both have mapping relationships with one concept. The merging result of ONTO-PDM product ontology and SCOR supply chain ontology is shown in Figure 5.6, while the part inside dotted box expressed the concepts from ONTO-PDM product ontology.

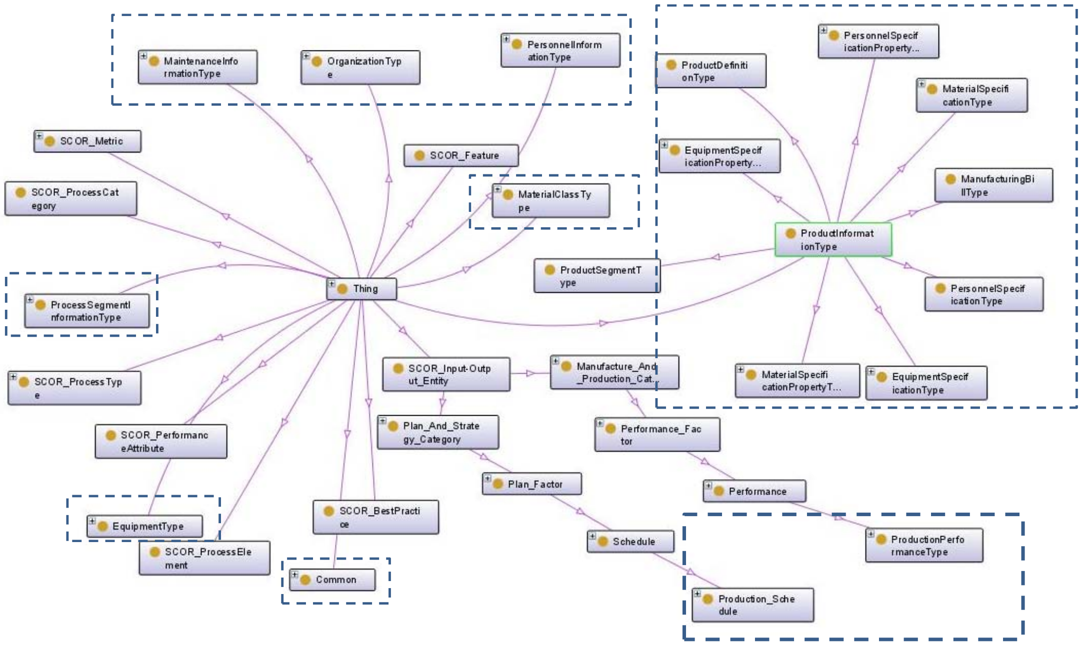


Fig. 5.6 Screenshot about the result of ontology merging

Table. 5-4 Statistics from the ONTO-PDM product ontology and SCOR supply chain ontology aligning process

	ONTO-PDM product ontology	SCOR supply chain ontology	Merging ontology (product-centric supply chain ontology)
Number of concepts	152	608	741
Number of relationships	228	600	823
Number of properties	188	11	239
Average number of properties per concept	3	1	2

How the merged product-centric supply chain ontology can realize the interoperability mong supply chain enterprise information systems well be illustrated in chapter 6, through one instance. It is expressed in detail in the scene of supply chain information system interoperability in section 6.2.

5.4 Mapping between Database and Ontology

XML is the extensible markup language, and already become the standard of Web data expression and exchange. XML uses user-defined mark to store data information, including the relationships among each mark such as the parent child relation and brotherhood. XML document can be seen as medium, widely used for data exchange, cross-platform application development and data transformation. XML is taken as the standard of data exchange among enterprise information systems, through the matching between XML and the product-centric supply chain ontology, to realize the interoperability of enterprise data. The mapping process between database and shared ontology contains two main steps: firstly, realize the transformation from enterprise XML data to OWL local ontology; and then realize the matching from enterprise OWL local ontology to the product-centric supply chain ontology by semantic matching algorithm.

5.4.1 Transformation from XML file to OWL ontology

In the environment of supply chain, building a local ontology by enterprise itself will cost lots of manpower, time and energy to organize the enterprise information which calls for interoperablity, and formalize semantic description. The heavy work of building local ontology makes it difficult to respond to the information system interoperability among enterprises with higher and higher demand. An automatic method is need to transform the XML information of enterprise information systems into OWL ontology, which is actually the enterprise local ontology. This ontology reflects the enterprise local information needed for semantic interoperability. Therefore, XML document which describe the information of enterprise business cooperation is adopted to build enterprise local ontology, and realize automatic method for building enterprise local ontology.

XML Schema is a recommended standard announced in May, 2011 by W3C group, and already become the preferred data modeling tool recognized globally in the environment of XML. XML Schema is the definition and description about the structure of XML document, with its main function is to restrain XML files and check the efficiency. It can define that what elements and attributes may appear in XML files and the data types of elements and attributes et al. Because XML Schema itself is XML document and accord with the grammar structure of XML, it can be analyzed by common XML parser. XML Schema provides certain semantic description for XML document, but the semantic of XML Schema is still hided in document. The pattern itself cannot describe its semantic. To realize dominant expression about the semantic of XML Schema, our research group proposed a method using ontology language to explicitly describe the semantic information inside XML Schema.

Based on what mentioned above, rule based method is adopted to realize the automatic mapping from XML document to ontology. The mapping transformation structure framework is shown in Figure 5.7.

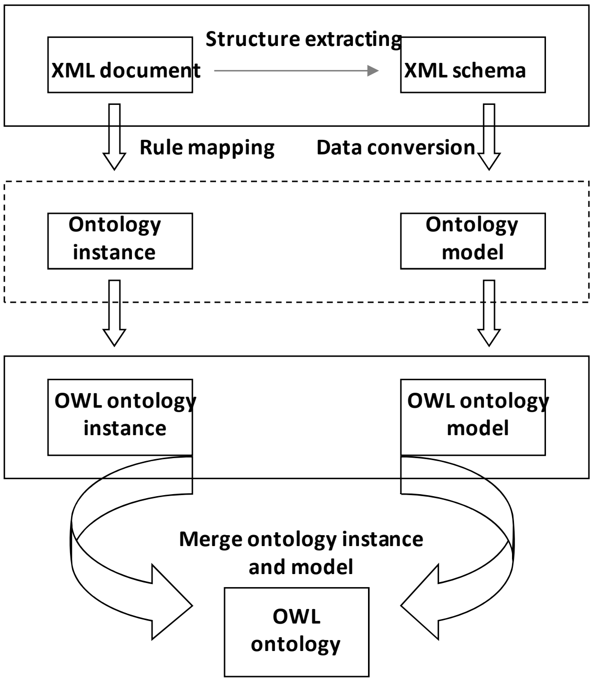


Figure 5.7 The mapping transformation process from XML document to OWL ontology

Based on this mapping framework, automatic generation of enterprise local ontology can be realized according to the following steps,:

(1) Extract the structure of the XML documents of original data source, generate XML Schema file with format as schema.xsd, making the preparation for further using XML Schema file to build ontology model;

(2) Through data migration and data transformation, use XML Schema file to generate the common ontology model model.owl. According to the mapping relationship listed in Table 5-5, establish the transformation rule from the elements in XML Schema to the elements in OWL, and then transform the original XML Schema file to ontology model by XSLT style sheet, realizing the automatic generation of ontology;

(3) Because the ontology model generated by XML Schema only contains concepts, properties and some constraints, no instance information is included, it's requested to transform original data source XML document into the common ontology instance by universal mapping rules, and then transformed directly into OWL ontology instance instance.owl.

(4) Merge OWL ontology instance and OWL ontology model effectively, that is merging model.owl file and instance.owl file. And then generate the final OWL ontology;

(5) Test, modify and evaluate the final OWL ontology in real application.

Table 5-5 The transformation rules between the elements of XML Schema and OWL

XML Schema elements	OWL elements
xsd:elements, containing subelement or at least one attribute	owl:Classes owl:ObjectProperties
xsd:elements, no subelement nor attribute； xsd:atrributes	owl:DatatypeProperties
xsd:minOccurs, xsd:maxOccurs	owl:minCardinality, owl:maxCardinality
xsd:sequence, xsd:all	owl:intersectionOf
xsd:choice	owl:intersectionOf, owl:unionOf, owl:complementOf
Inheritance mechanism： xsd:extension, xsd:restriction	owl:subClassof

5.4.2 Ontology Matching Algorithm

Aiming at the concept semantic similarity between enterprise local ontology and the product-centric supply chain ontology, the dissertation referred to the similarity algorithm studied by our research group. It concerns about the ontology characteristics such as concept, property, relationship and instance et al, synthesizes methods of pattern level, instance level and structure level, to calculation semantic similarity to ensure ontology mapping. The steps of the comprehensive concept similarity algorithm is shown as follows:

First, setting: w_1 is the weight based on concept similarity, and w_2 is the weight based on property similarity, w_3 is the weight based on concept structure similarity, and w_4 is the weight based on instance similarity. Therefore, under the premise of acquired concept similarity $SC(C_1,C_2)$, property similarity $SP(C_1,C_2)$ and structure similarity $SS(C_1,C_2)$, the comprehensive similarity of two concepts is expressed as Formula (4):

$$Sim(C_1,C_2)=\frac{w_1*SC(C_1,C_2)+w_2*SP(C_1,C_2)+w_3*SS(C_1,C_2)+w_4*SI(C_1,C_2)}{w_1+w_2+w_3+w_4} \quad (4)$$

The values of w_1, w_2, w_3 and w_4 should be ensured by domain experience. They can be appointed before specific calculation, while a set of normalized values do not exist. So in practical, based on a group of training samples, firstly each similarity component can be got by computing, and then they can be synthesized by changing w_1, w_2, w_3 and w_4. The rationality of results should be analyzed by experts, and at last the empirical values can be acquired. The specific computing methods about concept similarity $SC(C_1,C_2)$, property similarity $SP(C_1,C_2)$, structure similarity $SS(C_1,C_2)$ and instance similarity $SI(C_1,C_2)$ are explained in [Hai-bo Wang, 2010], which will be not explained in detail in the dissertation.

5.5 Mechanism of Semantic Interoperability

The construction of the product-centric supply chain ontology and the mapping method between database and ontology are the foundation for enterprise information systems semantic interoperability. Based on above foundation, information interoperability of business order is taken as an example to explain the semantic interoperability process of enterprise heterogeneous information systems under the product-centric supply chain information system interoperability framework, which is shown in Figure 5.8.

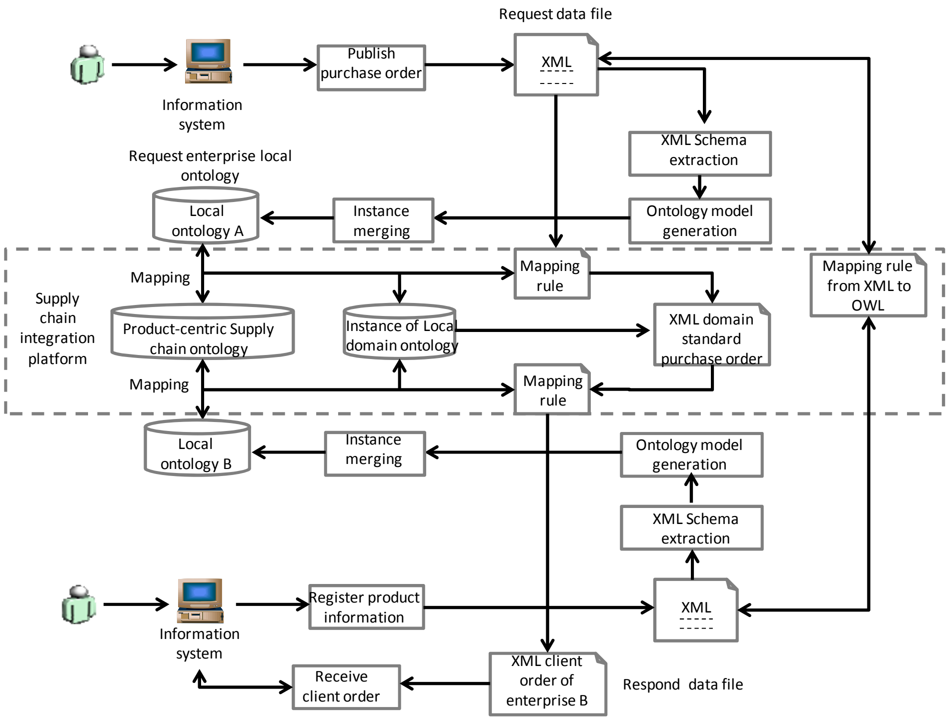


Figure 5.8 semantic interoperability process based on the product-centric supply chain ontology

During this process, the product-centric supply chain ontology is domain ontology, containing the shared concepts in supply chain. While the local ontology is the shared concepts within local scope such as certain enterprise, department or system. The mapping between domain ontology and local ontology is the key to realize the semantic interoperability of enterprise information systems in supply chain environment.

The semantic interoperability process based on the product-centric supply chain ontology can be summarized as follows:

(1) Domain experts build domain ontology: The domain ontology here is the product-centric supply chain ontology, which is based on some international standards such as IEC 62264 and ISO 10303 STEP-PDM and Supply-Chain Operations Reference-model. This ontology is the explicit expression of shared concept and the semantic expression of data mode, and it can be shared among enterprise information systems in the environment of supply chain, for supporting the semantic interoperability of information systems when business cooperation.

(2) Source information system generates the XML request data file: The enterprise users can transform the XML request data file into the request enterprise

local ontology by the website of supply chain information system interoperability, which can be completed by the above automatic transformation from XML data file to OWL ontology.

(3) Transform the request enterprise local ontology into the instance of local domain ontology: Through matching with the product-centric supply chain ontology by semantic similarity algorithm, realize the request enterprise local ontology transformed to instance of local domain ontology, and save the mapping rules.

(4) Transform the instance of local domain ontology into the XML response data file: According to the demand of target enterprise and the saved mapping rules, transform the instance of local domain ontology into the XML response data file satisfying the demand of target enterprise, and then import it into target information system through the user interface.Then the interoperability process between the enterprise information systems in supply chain environment is accomplished and the data integration is realized.

The semantic interoperability mechanism based on the product-centric supply chain ontology integrates the key technologies discussed in the dissertation such as domain ontology modeling, ontology matching and the matching technology between database and ontology, and is the proper solution to realize the enterprise information system interoperability in the environment of supply chain.

5.6 Conclusion

In this chapter, ontology merging between ONTO-PDM product ontology and SCOR supply chain ontology was realized to generate the product-centric supply chain ontology, through the matching between shared ontologies. Meanwhile it proposed the mapping discovery method between ontology and external database, stated the supply chain information system interoperability mechanism based on ontology, and then set business order as an example to explain the supply chain information system interoperability.

6 Instance of Supply Chain Information System Interoperability and Prototype Development

6.1 Introduction

This chapter takes the machine bed supply chain process of the double column CNC guideway&surface grinding machine in Machine Tool Company as an example to explain the feasibility about the supply chain information system Interoperability, which is proposed in chapter 5. And then according to the principle of product-centric supply chain Interoperability framework, the prototype system of supply chain enterprise information system Interoperability is developed to provide a feasible solution for the information integration of supply chain enterprises.

6.2 Instance of Supply Chain Information System Interoperability

6.2.1 A Scenario example of supply chain

The double column CNC guideway&surface grinding machine in Machine Tool Company is large-scale machine tool, whose raw materials and small components are purchase parts, while the main pieces such as machine bed is made to order. This chapter takes the machine bed as an example to explain interoperability of supply chain enterprise information system. Machine bed is large assembling unit composed of main machine bed, front sub machine bed and back sub machine bed, whose manufacturing processed are shown in Figure 6.1. Main machine bed, front sub machine bed and back sub machine bed are all castings, while the casting process is completed by external cooperation enterprises, and the finish machining processes are completed by enterprise itself. After, the assembly process is conducted. The Make-to-Order supply chain process of machine bed is built according to SCOR model, which is shown in Figure 6.2. Thereinto, the manufacturing of machine bed is a complex process. Taking the manufacturing process of main machine bed as an example, it contains casting, scribing, primary planning, primary milling, Ageing treatment, scribing and finishing planning, which are carried out in different places.

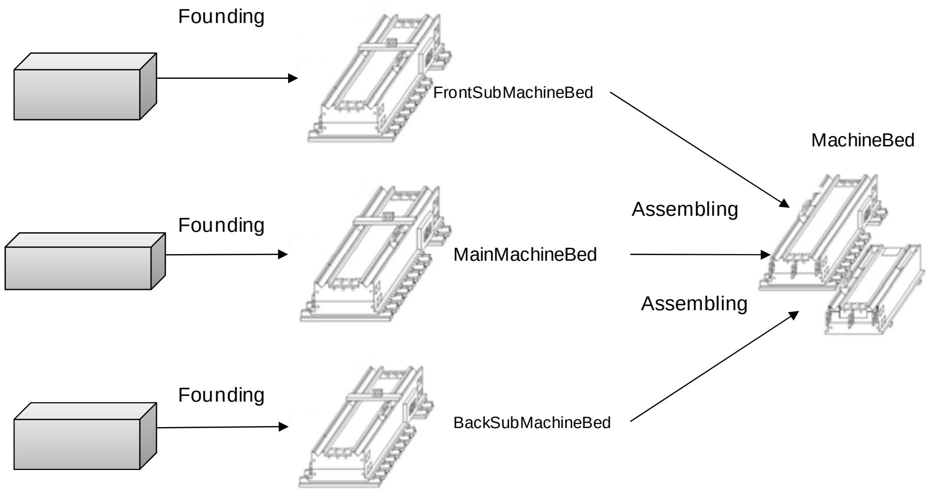


Figure 6.1 Manufacturing process of machine bed

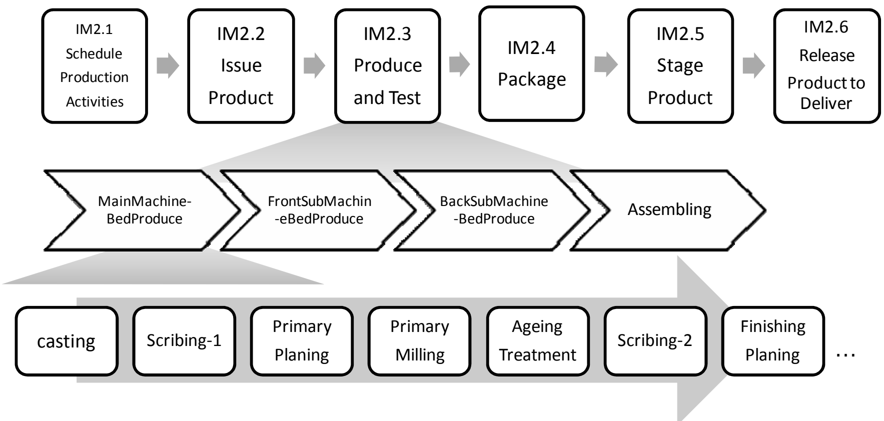


Figure 6.2 Make-to-Order supply chain process of machine bed

The Make-to-Order supply chain process of machine bed involves machine tool enterprise with its sales department, design department and manufacturing department, external cooperation enterprises, and the participant systems including ERP, CAPP, CAD and Pro-E et al. The interaction scenario is shown in Figure 6.3. Machine tool enterprise and external cooperation enterprises use their own CAPP systems, without integration. Inside machine tool enterprise, the complete automation on production data and order data management have not been realized, while many works still by manual. This interaction mode could be seen as system centric, and interfaces should be implemented between systems or total by manual work when systems need interoperability.

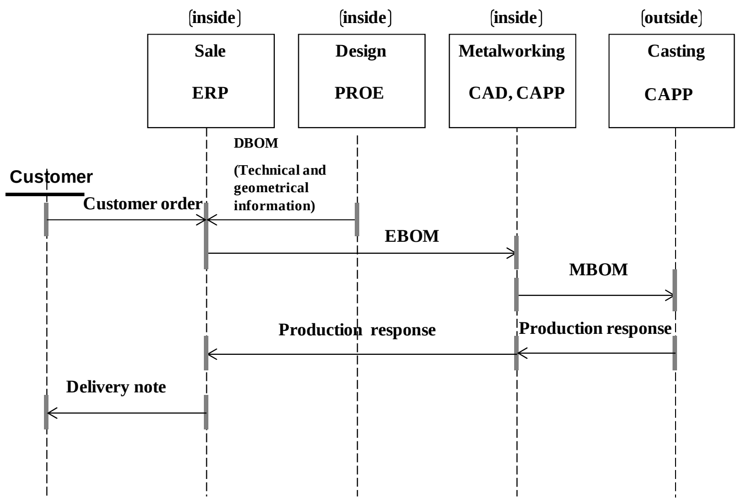


Figure 6.3 System centric interaction scenario

In the supply chain information interoperability framework proposed in the dissertation, it takes the product-centric supply chain ontology as a storage and exchange center of shared information to store product information during the whole supply chain process. And then each enterprise information system can realize interoperability by interacting XML format data with the ontology. In the supply chain information interoperability framework, the product-centric supply chain ontology can be regarded as a information system, and the enterprise systems such as ERP, CAPP and CAD et al can realize information interoperability by exchanging relevant information with it. Taking the manufacturing process of main machine bed in supply chain process shownin Figure 6.2 as an example, its casting process are by external cooperation enterprises, while the rest metalworking processes are completed inside machine tool enterprise. The supply chain process can be described as the scenario shown in Figure. 6.4.

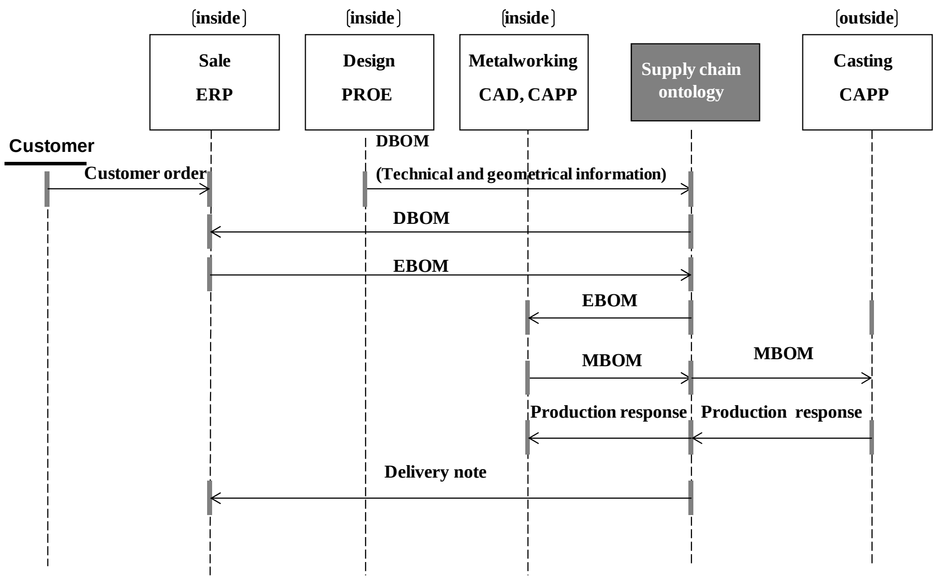


Figure 6.4 The product centric supply chain information system interoperability scenario

6.2.2 Interoperability process of supply chain information system

In the supply chain information system interoperability scenario shown in Figure 6.4, the Make-to-Order supply chain process of main machine bed can be divided into four stages:

(1) Receive client's order. The DBOM is achieved from Pro-E system in design department which stores product information about design, geometry, material etc. And then map DBOM to instance of product-centric supply chain ontology. The ERP system in sale department can acquire the DBOM information stored in Pro-E system by interaction with the ontology., and then establish EBOM information based on that.

(2) After receiving the EBOM information from the product-centric supply chain ontology, the metalworking department drafts detailed manufacturing process to generate MBOM, and then send the MBOM information to external cooperation enterprises by the product-centric supply chain ontology, to enable external cooperation processing.

(3) The metalworking department sends the production request to external cooperation casting room by the product-centric supply chain ontology. The external cooperation casting room responds to the request, and complete manufacturing process.

(4) The metalworking workshop of machine tool enterprise completes the rest of the product manufacturing process, and sends out the deliver message to sales department by the product-centric supply chain ontology.

First stage:

The Pro-E system of design department stores the design information of machine bed, including part information, material quantity, part drawing number, etc. All these information compose of the product DBOM information using XML format data file. According to the interoperability mechanism in section 5.5, the XML file can be transformed to local ontology automatically.


```xml
<?xml version="1.0" encoding="UTF-8"?>
<product_information>
  <MainMachineBed>
    <name>MainMachineBed</name>
    <PartNumber>1001/1</PartNumber>
    <material>HT250</material>
    <number>1</number>
    <weight>none</weight>
    <type>Casting piece</type>
  </MainMachineBed>
  <LeftWedge>
    <name>LeftWedge</name>
    <PartNumber>1001/10</PartNumber>
    <material>45</material>
    <number>4</number>
    <weight>13.06</weight>
    <type>Forgeable piece</type>
  </LeftWedge>
  <FrontSubMachineBed>
    <name>FrontSubMachineBed</name>
    <PartNumber>1001/2</PartNumber>
    <material>HT250</material>
    <number>1</number>
    <weight>none</weight>
    <type>Casting piece</type>
  </FrontSubMachineBed>
  <BackSubMachineBed>
    <name>BackSubMachineBed</name>
    <PartNumber>1001/3</PartNumber>
    <material>HT250</material>
    <number>1</number>
    <weight>none</weight>
    <type>Casting piece</type>
  </BackSubMachineBed>
</product_information>
```


DBOM.XML


```xml
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
xml:base="http://localhost/tmp/D:/workspace/.metadata/.plug
ins/com.genuitec.eclipse.easie.tomcat.myeclipse/tomcat/web
apps/business/results/onto057/model.owl#"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
……
<BackSubMachineBed
rdf:ID="BackSubMachineBed_N10043"><name>BackSubMa
chineBed</name><PartNumber>1001/3</PartNumber><mat
erial>HT250</material><number>1</number><weight>none
</weight><type>Casting
piece</type></BackSubMachineBed><FrontSubMachineBed
rdf:ID="FrontSubMachineBed_N1002E"><name>FrontSubM
achineBed</name><PartNumber>1001/2</PartNumber><m
aterial>HT250</material><number>1</number><weight>no
ne</weight><type>Casting
piece</type></FrontSubMachineBed><LeftWedge
rdf:ID="LeftWedge_N10019"><name>LeftWedge</name><P
artNumber>1001/10</PartNumber><material>45</material>
<number>4</number><weight>13.06</weight><type>Forge
able piece</type></LeftWedge><MainMachineBed
rdf:ID="MainMachineBed_N10004"><name>MainMachineBe
d</name><PartNumber>1001/1</PartNumber><material>H
T250</material><number>1</number><weight>none</weigh
t><type>Casting
piece</type></MainMachineBed><product_information
rdf:ID="product_information_N10001"><hasMainMachineBe
d
rdf:resource="#MainMachineBed_N10004"/><hasLeftWedge
rdf:resource="#LeftWedge_N10019"/><hasFrontSubMachine
Bed
rdf:resource="#FrontSubMachineBed_N1002E"/><hasBackS
ubMachineBed
rdf:resource="#BackSubMachineBed_N10043"/></product_i
nformation></rdf:RDF>
```


DBOM.OWL

Figure 6.5 Transform DBOM to local ontology

DBOM local ontology form Pro-E system can be stored in product-centric supply chain ontology by ontology matching. The ERP system in sale department achieve product compose information through interaction with product-centric supply chain ontology, and enact EBOM by adding information such as product type, product name, product description, enactment date of BOM and material bill et al. EBOM (Table 6-1) explains that machine bed is composed by assembly units, including main machine bed, front sub machine bed and back sub machine bed. The information contained in the material bill of main machine bed is listed in detail in Table. 6-2, including material type, description, material unit and quantity et al. This material bill is about the ingredient used to make one main machine bed, and the bed material is gray cast iron HT250. The correspondence relationships between product DBOM and EBOM are illustrated in Figure 6.6.

Table. 6-1 EBOM information of grinding machine bed

Product	BillOfMaterialID	ProductInformation	PublishDate	ManufacturingBill	Description	…
GrindingMachineBed	bom27627	HZ-KDL8025x20-1000	2010-10-20	MainMachineBed_Manufacturingbill FrontSubMachineBed_Manufacturingbill BackSubMachineBed_Manufacturingbill	InformationDefiningMachineBed	…
MainMachineBed	bom27635	HZ-KDL8025x20-1000	2010-10-20	MainMachineBed_Manufacturingbill	InformationDefiningMainMachineBed	…
FrontSubMachineBed	bom27637	HZ-KDL8025x20-1000	2010-10-20	FrontSubMachineBed_Manufacturingbill	InformationDefiningFrontSubMachineBed	…
BackSubMachineBed	bom27638	HZ-KDL8025x20-1000	2010-10-20	BackSubMachineBed_Manufacturingbill	InformationDefiningBackSubMachineBed	…

Table 6-2 Material bill information of main machine bed EBOM

ManufacturingBill	BillOfMaterialID	PublishDate	ViewContext	Material	Material Unit	quantity	MaterialVersion
MainMachineBed_Manufacturingbi	bom27635	2010-10-20	ProE	GreyCastIron_HT250	piece	one	MaterialVersion1.0

11							

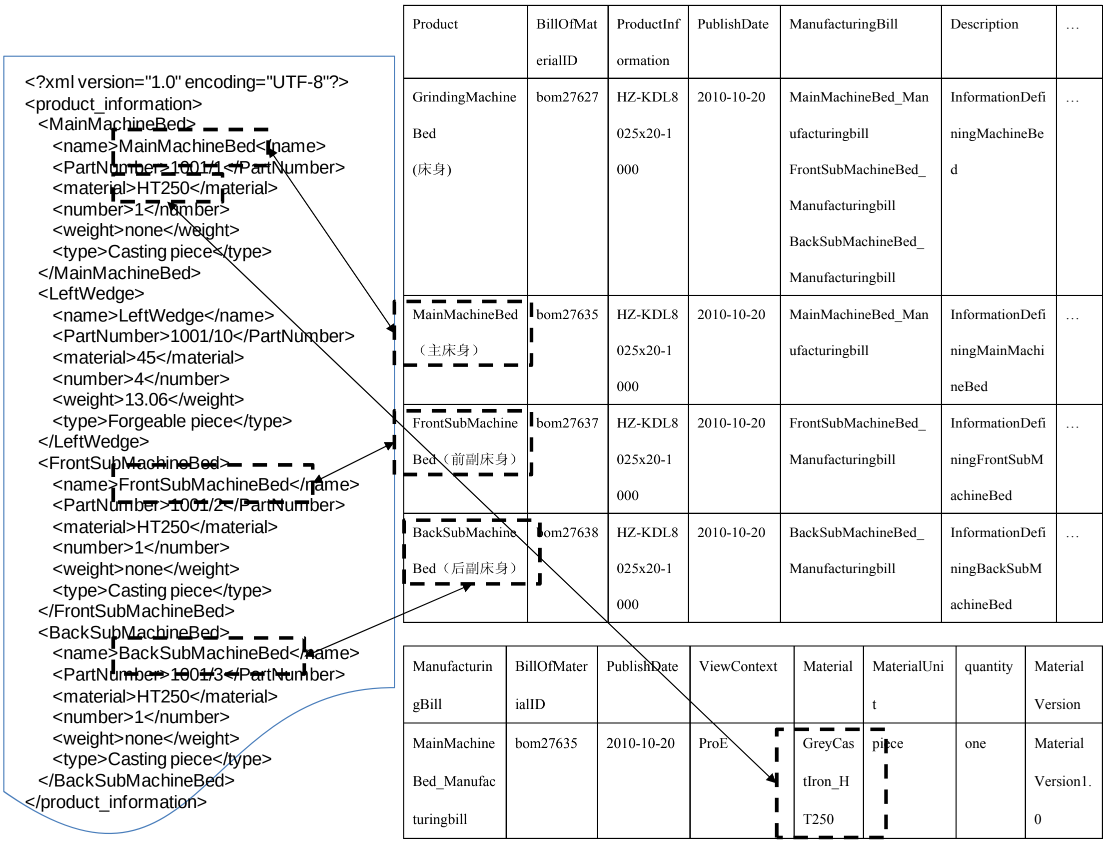


Figure 6.6 Relationship between DBOM and EBOM

The EBOM information of machine bed mapped to the instance of the product-centric supply chain ontology in Protégé, which is shown in Figure 6.7. The ontology concept model of EBOM information is shown in Figure 6.8.

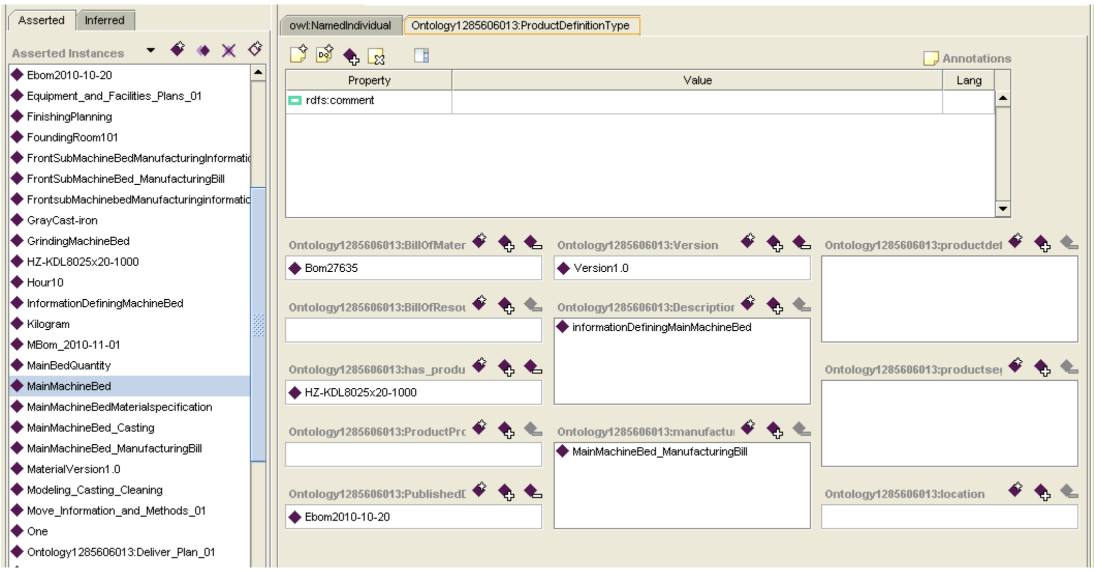


Figure 6.7(a) The instance of machine bed EBOM in Protégé

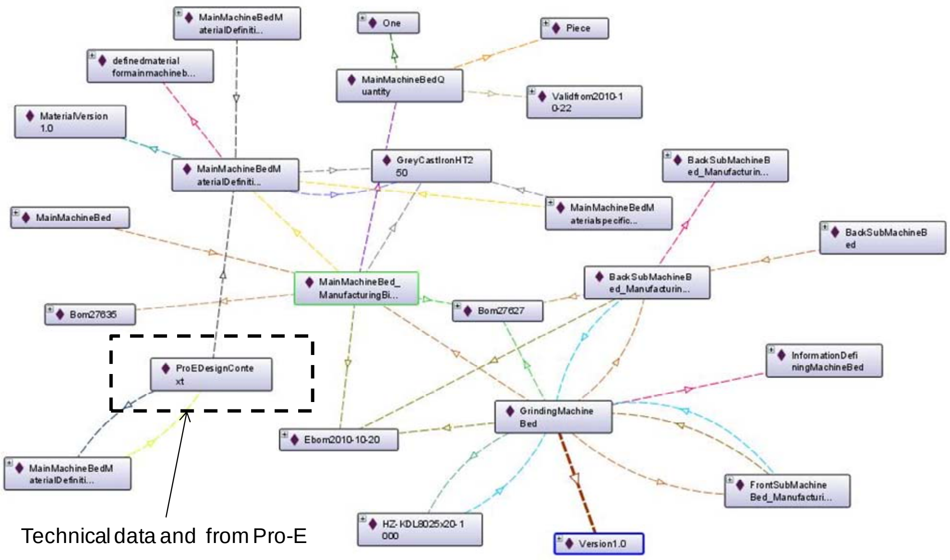


Figure 6.7(b) Part of the instance of machine bed EBOM in Protégé

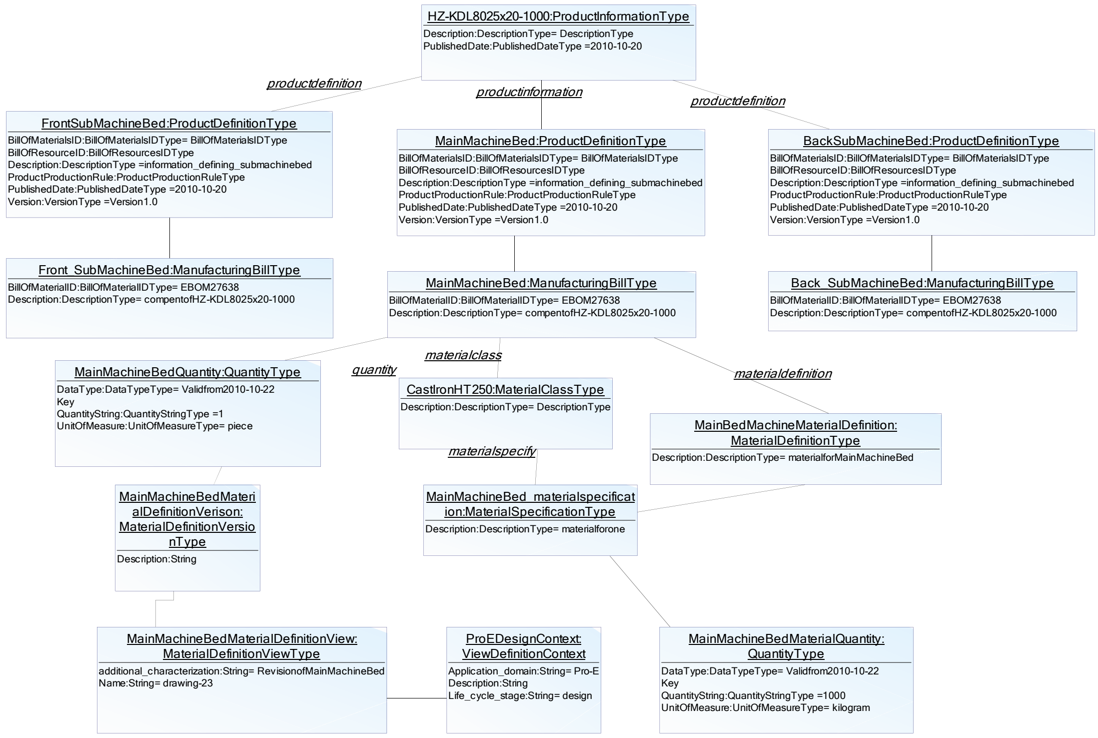


Figure 6.8 Part of ontology concept model of EBOM information

Second stage:

The product information in EBOM is only about the product composition, material quantity, and the technical and geometrical information on product et al, not including specific technical information such as manufacturing process, manufacturing parameter and procedure location et al. After receiving product EBOM, the metalworking department of machine tool enterprise may establish detailed MBOM for subsequent course of product manufacturing. MBOM stores specific product information including specific manufacturing process, manufacturing equipment, material, workshop location and procedure time et al. The machine tool enterprise can enable external cooperation processing through the product-centric supply chain ontology to send out product MBOM to external cooperation casting room.

The MBOM information of main machine bed is shown as Table. 6-3, including the manufacturing course and manufacturing place, procedure equipment, material, participant people and procedure time et al.

Table 6-3 MBOM information about the manufacturing technical process of main machine bed

Location	ProduceName	Process	Equipment	Material	Person	Time	…
CastingRoom（outside）	Casting	Modeling Casting Cleaning		GreyCastIron HT250	Group102	15Days	…
Metalworking01	Scribing-1			GreyCastIron HT250	Group101	5Hours	…
Metalworking01	PrimaryPlanning	Planningback Planningsurface	PlanningMachine01	GreyCastIron HT250	Group101	6Hours	…
Metalworking01	PrimaryMilling	Millingback Millingguiderail	Milling Machine 03	GreyCastIron HT250	Group101	8Hours	…
CastingRoom04	AgeingTreatment			GreyCastIron HT250	Group102	30Days	…
Metalworking01	Scribing-2			GreyCastIron HT250	Group101	4Hours	…
Metalworking01	FinishingPlaning	Planingguiderail		GreyCastIron HT250	Group101	7Hours	…
…	…	…	…	…	…		…

The manufacturing process of main machine bed in MBOM is mapped to the class *ProductSegment* in the product-centric supply chain ontology, and the instance in Protégé is shown in Figure 6.9. In Figure 6.9(b), the instance inside the dotted line are the manufacturing processes of main machine bed including casting, scribing and primary planning et al. The ontology concept model of MBOM information is shown in Figure 6.10.

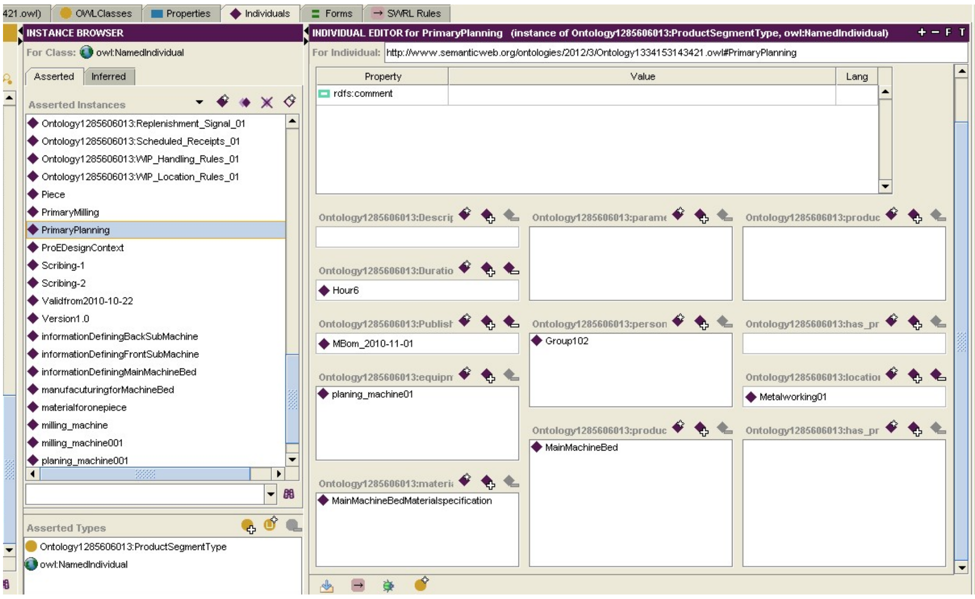


Figure 6.9(a) Part of the instance of machine bed MBOM in Protégé

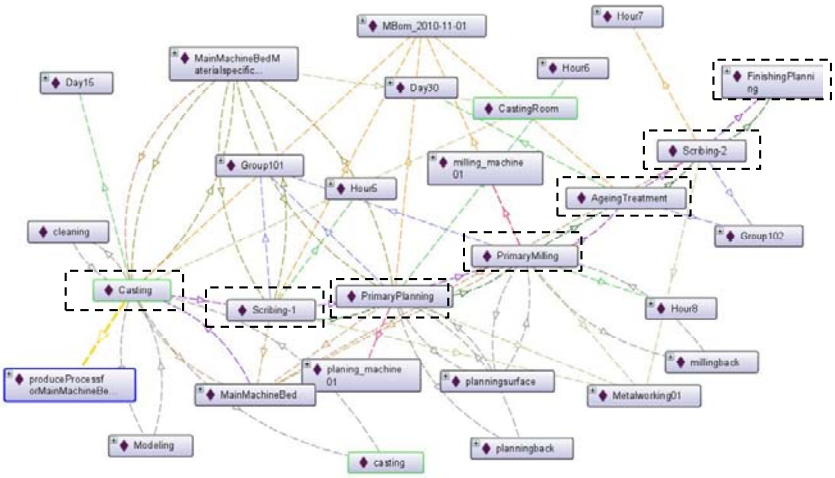


Figure 6.9(b) Part of the instance of machine bed MBOM in Protégé

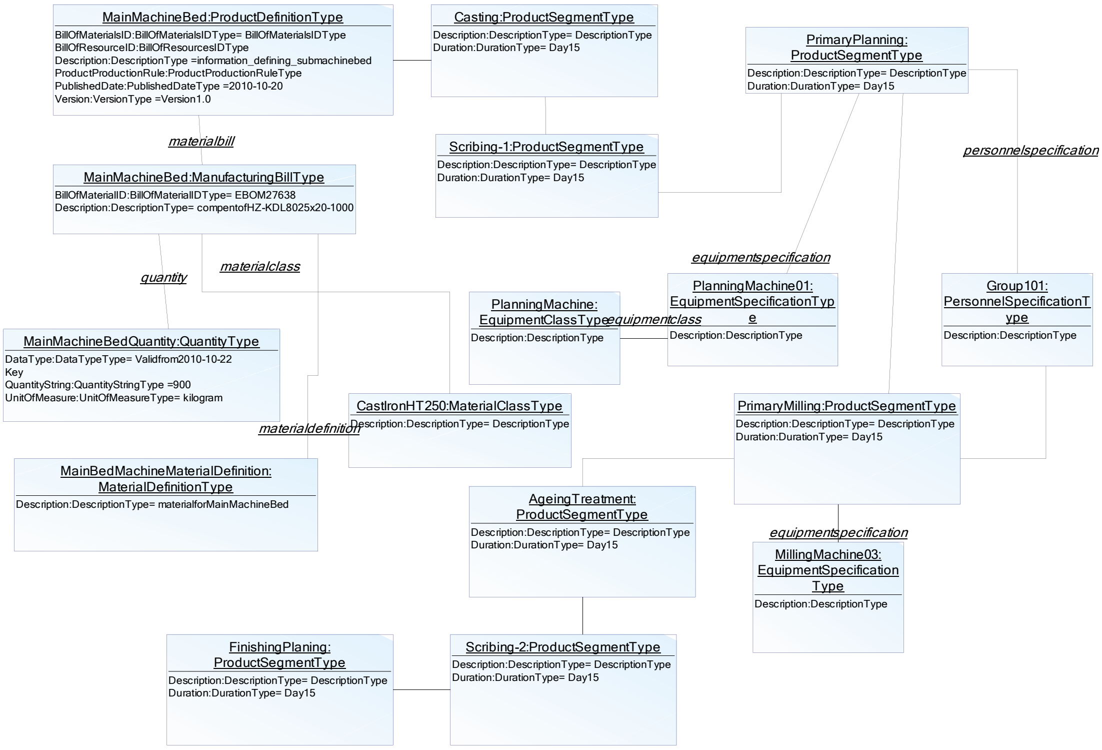


Figure 6.10 Part of ontology concept model of MBOM information

Third stage:

The metalworking workshop of machine tool enterprise can send out production request to external cooperation casting room though the product-centric supply chain ontology. And the external cooperation casting room responds to the production request process. The product request asks external cooperation casting room to complete the casting process of main machine bed, and its content is shown in Table 6-4. The external cooperation casting room responds to the production request, and its content is shown as Table 6-5. This course can be realized by *ProductionRequirment* and *ProductionResponse* in the product-centric supply chain ontology.

Table. 6-4 Production request of machine tool enterprise

Produce	Duaration	EarliestStart Time	LatestEnd Time	Material Quantity	Material	UnitOfMaterial	…
Casting	day15	2010-12-01	2010-12-15	1000	GreyCastIron HT250	kilogram	…

Table. 6-5 Production response of external cooperation casting room

Produce	StartTime	EndTime	MaterialUsed Quatity	MaterialActual	MaterialLot	..
Casting	2010-12-03	2010-12-08	900	GreyCastIronHT250	Hall300	..

The instance of production request in Protégé is shown as Figure 6.11. In Figure 6.11(b), the instance inside dotted line box represent production request and production response respectively. And the ontology concept model of production request and response are shown in Figure 6.12.

Figure 6.8(a) The instance of production request in Protégé

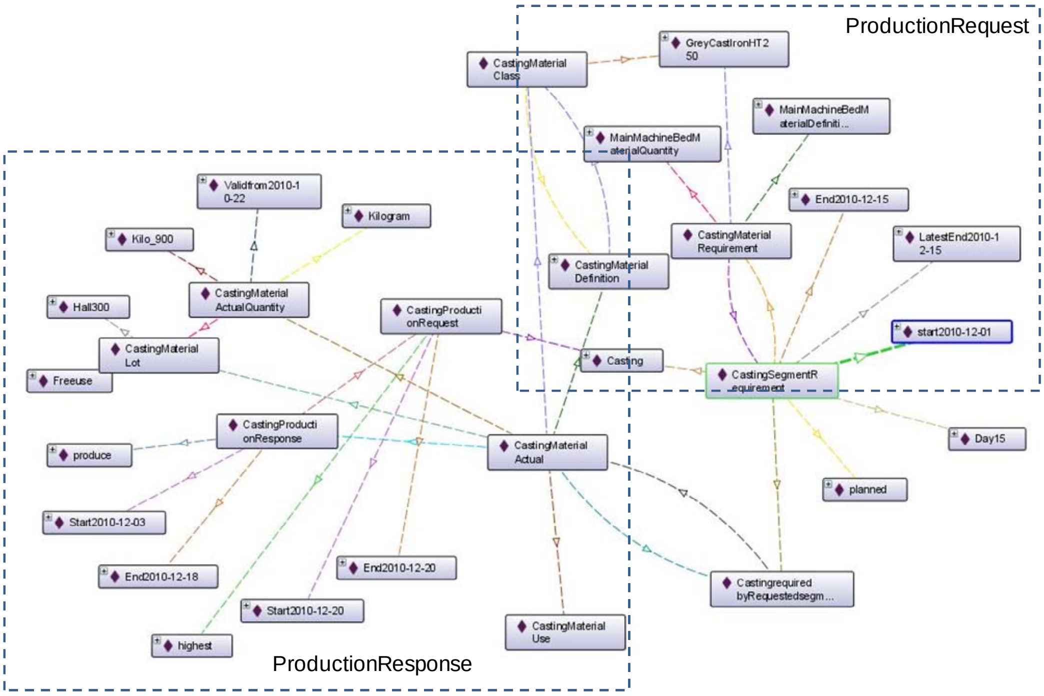


Figure 6.11(b) Part of the instance of production request in Protégé

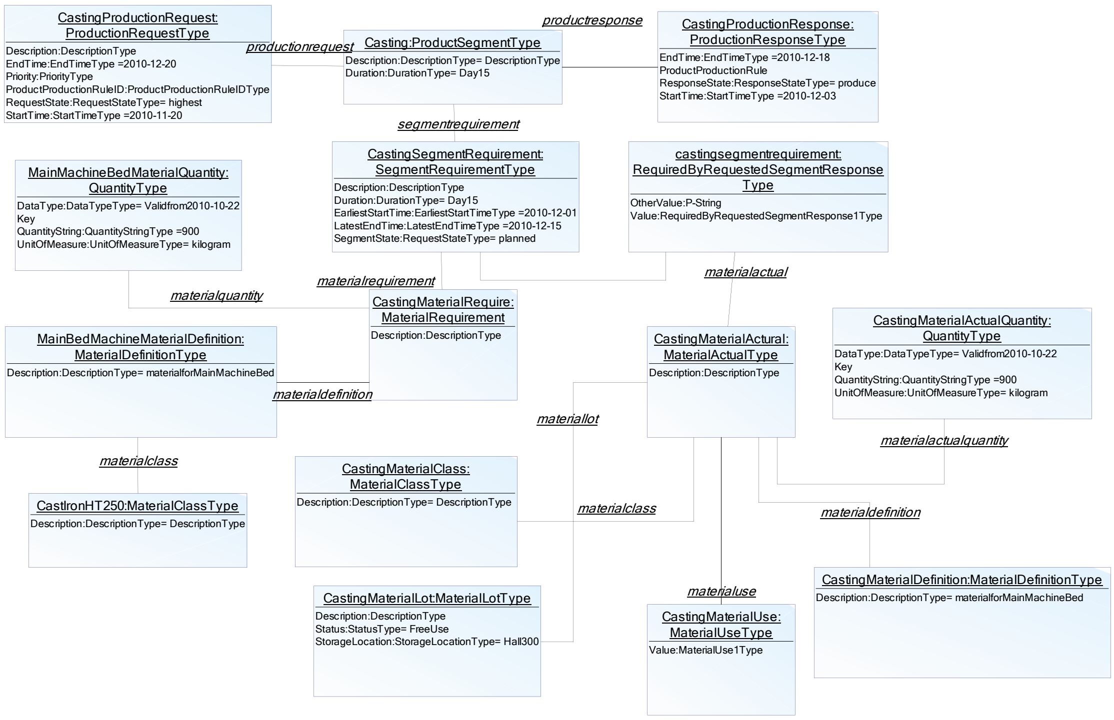


Figure 6.9 Part of ontology concept model of production request and response

Forth stage:

The metalworking workshop of machine tool enterprise completes the rest manufacturing process of product, sending out the deliver notification to enterprise sales department through the product-centric supply chain ontology. This course is similar to the third course, and can be realized by passing message with class *ProductionRequirment* and *ProductionResponse*. So here does not describe it in detail.

To better explain the information contained in the product-centric supply chain ontology includes the full stage of the Make-to-Order supply chain process of grinding machine shown as Figure 6.2, the instance about the Make-to-Order supply chain process of main machine bed is built in Protégé seeing in Figure 6.13. In Figure 6.13(a), the class *ProductSegment* of ONTO-PDM product ontology inside the dotted line includes the technique information of manufacturing process. In Figure 6.13(b), the class *ProductDefinition* of ONTO-PDM product ontology inside the dotted line box contain the technique information of product definition.

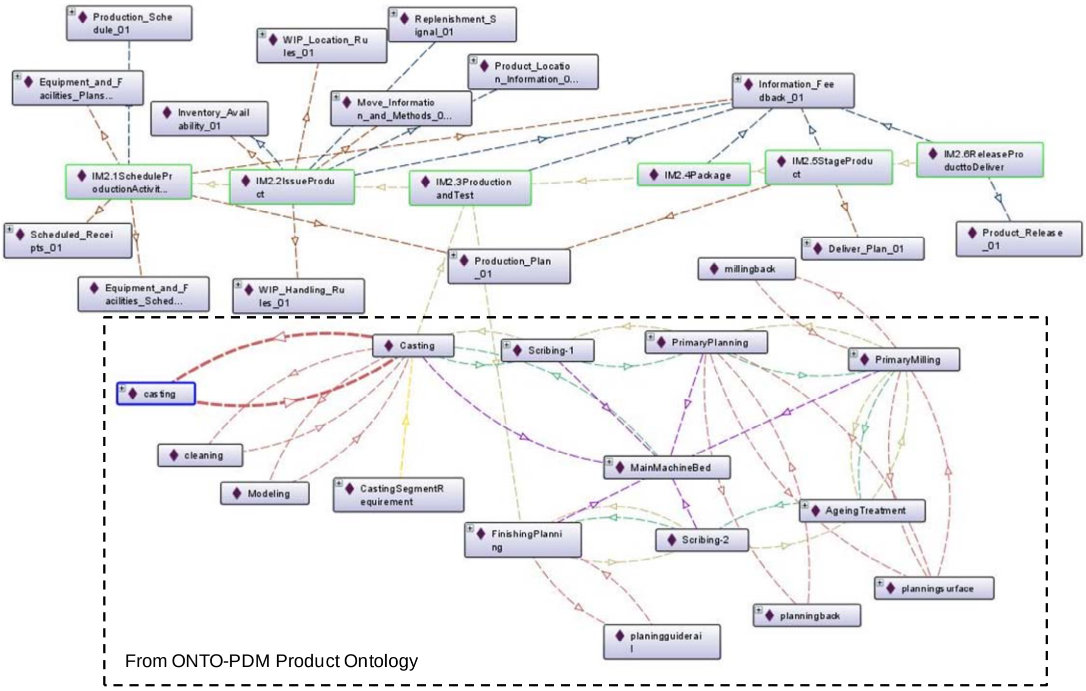


Figure 6.13(a) Part of the instance about Make-to-Order supply chain process of grinding machine in Protégé

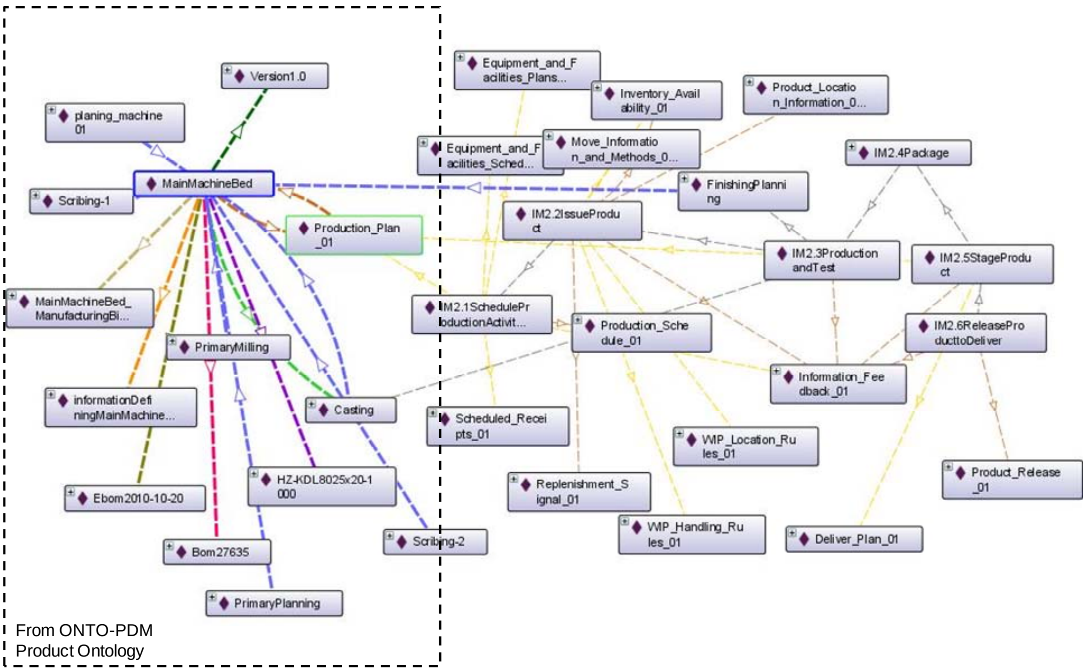


Figure 6.13(b) Part of the instance about Make-to-Order supply chain process of grinding product in Protégé

Because of condition limitation, the grinding machine company does not have SCM, MES, PDM systems. So we simulate a scenario of supply chain environment with ERP, CAPP, SCM, MES and PDM involved in, and all these systems use XML as medium to exchange information, seeing Figure 6.14. The product-centric supply chain ontology is considered as a common shared information center for store and exchange product related information in whole supply chain process, with which SCM system could exchange process information, PDM and ERP system could exchange product definition data, CAPP and MES could system exchange manufacturing process data. The product-centric supply chain ontology as an independent system record product related information in whole supply chain. New systems involved could realize interoperability with other enterprise systems through interaction with this shared ontology.

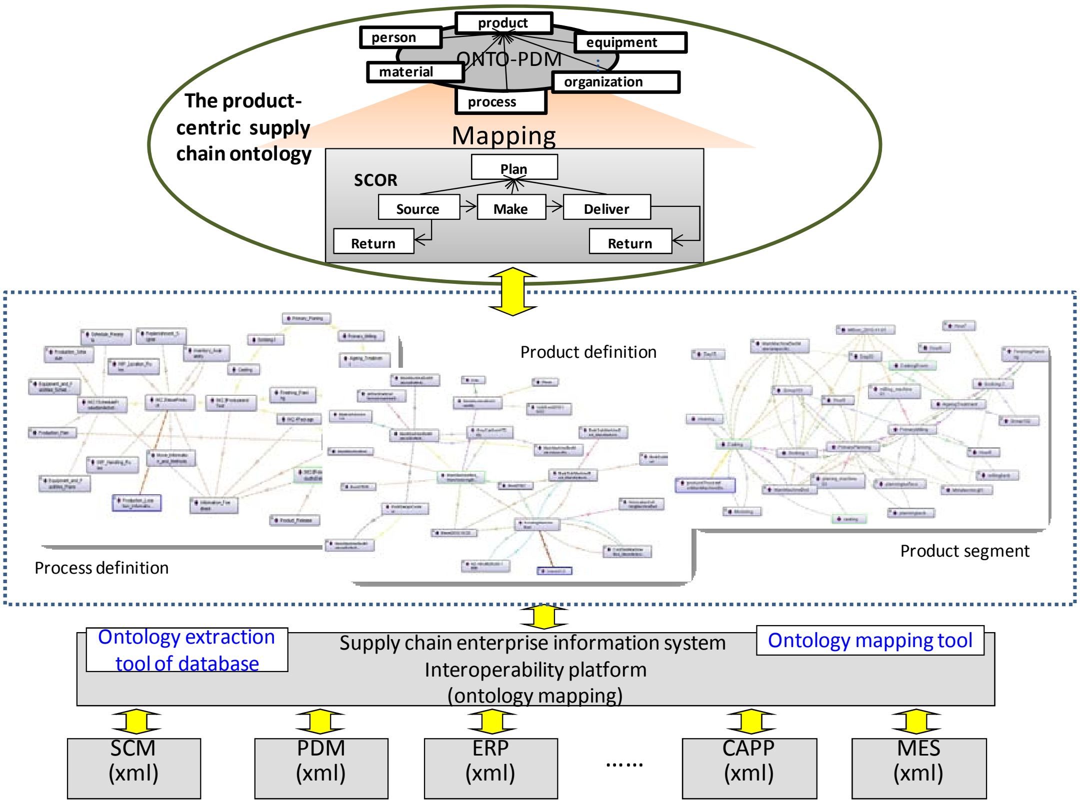


Figure 6.14 The interoperability of product-centric supply chain information systems

6.3 Prototype design and development

6.3.1 Framework of prototype system

According to the principle of product-centric supply chain information systems interoperability framework, we developed a prototype system of supply chain information system interoperability. Users can complete the interoperability of information systems via interoperability system, whose core technology and method is through ontology matching to realize semantic interoperability. Designing this system is practical application of the principle of product-centric supply chain information system interoperability .

The structure of the prototype system is illustrated in Figure 6.15. In the supply chain environment, enterprise information systems exchange XML data with the system interface, and then store the data as local ontologies by database ontology extraction tool. The product-centric supply chain ontology is the core shared ontology base of this system, supporting the ontology maintenance, semantic mapping and semantic transformation, in order to realizes the interoperability between different enterprise information systems.

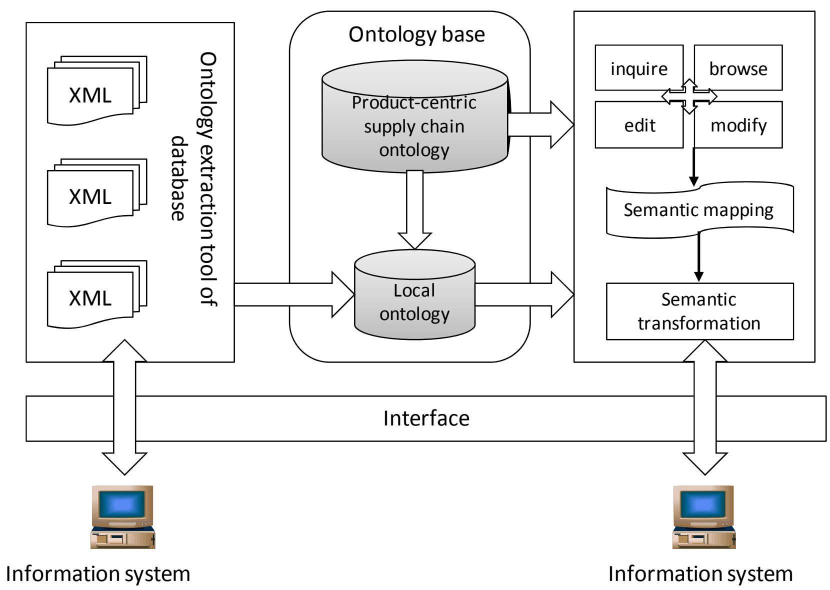


Figure 6.15 The structure of prototype system of supply chain information system interoperability

The core shared ontology base of supply chain information system interoperability platform adopts Protégé as the development tool. Protégé is the ontology development and edit tool most widely used currently developed by Stanford University, which support multiform saving of ontology file such as RDF(S), OWL and XML et al. The core shared ontology base in this system adopts OWL format file. It uses Jena software development kit to store OWL files of ontology base into database in persistence, and then develops reading, inquiring, matching, saving and other operations on ontology by Jena interface.

JAVA is adopted as the programming language of this platform, and MySQL is used as database.

6.3.2 Function module of prototype system

According to the principle of product-centric supply chain information system interoperability framework, the interoperability platform mainly provides three function modules: mapping between database and ontology base, ontology management and semantic interoperability.

Mapping between database and ontology module: This module provides the function that transforming enterprise data into enterprise local ontology. The XML data of enterprise can be automatically transformed and stored as OWL enterprise local ontology by the database ontology extraction tool integrated in the platform, preparing for the following semantic matching. Figure 6.16 and Figure 6.17 are screenshots of the uploaded XML data of enterprise and the transformed into OWL enterprise local ontology.

Figure 6.16 The screenshot of uploaded XML data of enterprise

Figure 6.17 The screenshot of OWL enterprise local ontology

Ontology management module: This module realizes product-centric supply chain ontology built by Protégé persistence in database through Jena API. Applet technology is used to display ontology concept dynamically, helping visually read, edit, inquire and modify ontology knowledge, and integrate the ontology maintenance work into the platform, improving the usability of system. This module can also be applied to other knowledge systems as a modularization ontology management tool. Figure 6.18 shows the screenshot of ontology visualization of ontology management module.

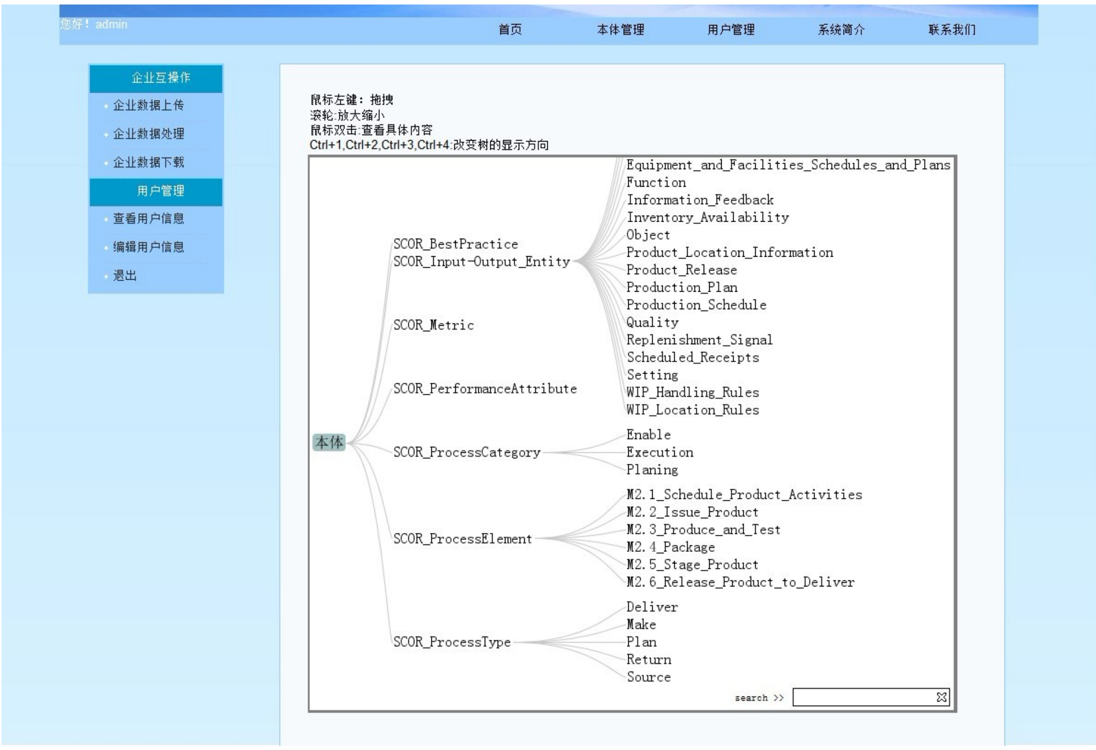


Figure 6.18 The visual guide interface of ontology

Semantic interoperabiltiy module: Aiming at local ontology and domain ontology, it adopts a method based on comprehensive similarity algrithem to realize the mapping between domain ontology and enterprise local ontology. Users can deal with enterprise local ontology by semantic information adding, iterative computation and calculation results adjustion to gain the matching results satisfied by users. The system stores the final matching results into ontology mapping files. Enterprises can download the matching results according to their demands. Figure .6.19 is the matching interface of enterprise local ontology, while the Figure 6.20 is the download interface of matching results.

Figure 6.19 The matching interface of enterprise local ontology

Figure 6.20 The download interface of matching results

6.4 Conclusion

This chapter took the main machine bed supply chain process of the double column CNC guideway&surface grinding machine in Machine Tool Company as an example, analyzed the supply chain process of main machine bed from ordering, external cooperation manufacturing to complement step by step, and test the feasibility of the product-centric supply chain information system interoperability. And then according to the interoperability mechanism, the prototype system of enterprise supply chain information system interoperability platform is developed. At the end, the chapter introduced the development tool and function modules of platform.

7 Conclusion and Future Works

7.1 The main work conclusion

The dissertation is supported by supported by Chinese Science and Technology Support Plan Project (2006AA04Z157) "Research and application on ontology-based business cooperation oriented heterogeneous systems integration technology", Chinese National Natural Science Foundation Project (61175125) "Semantic interoperability of heterogeneous systems in supply chain based on SCOR", and Zhejiang Provincial Natural Science Foundation Project (Y107360) "Research and application on business cooperation oriented semantic interoperability mechanism". Meanwhile this work combines with the research results from the research group of professor Panetto in CRAN at Nancy in French. and researches on the interoperability of enterprise information systems in the environment of supply chain. The research contents of dissertation contains: the semantic interoperability mechanism of networked enterprise information systems, the building of shared ontology and the semantic interoperability method based on ontology et al. At last, we evaluate the effectiveness of research results in a machine tool company.

The research works of dissertation can be summarized as follows:

(1) Based on the analysis of relationships and characteristics among the enterprises and their information systems in supply chain environment, it defined the SoNE (Systems-of-Networked Enterprises) paradigm if the networked enterprises in supply chain environment. Focusing on the Connectivity characters of this paradigm, it analyzed the solutions to current system connectivity, and proposed to adopt semantic interoperation to deal with the dynamic connectivity in systems. Meanwhile, it analyzed the characteristics and demands of supply chain network, proposing the semantic interoperability framework of networked enterprises in supply chain.

(2) It studied the model transformation method based on ODM. Through detailed comparison analysis on UML and OWL language, it provided the mapping relationships from UML to OWL and the ATL transformation method. Based on the research results of CRAN laboratory, it adopted model-driven ontology development method to reuse ontology and transform coding, and transformed the UML model of ONTO-PDM product ontology which was built based on international standards

IEC62264 and ISO 10303 STEP-PDM into OWL product ontology, to construct the common ontology to integrate the business information and technical data related to product life cycle such as product design, manufacturing and deliver et al.

(3) It analyzed the related standards of supply chain integration, and focused on discussing the supply chain system architecture standard: the Supply-Chain Operations Reference-model (SCOR). And then it built SCOR supply chain ontology, which contained the dynamic configuration mechanism of supply chain process and the process self-measurement mechanism, providing reference data for supply chain optimizing management. At the end it gave the SCOR supply chain ontology instance about the make-to-order process of machine tool product.

(4) It studied the semantic matching method between ontologies and the matching method between database and ontology. Merged ONTO-PDM product ontology and SCOR supply chain ontology into the shared product-centric supply chain ontology. And it took the supply chain process about the main bed of grinding machine as an example to validate the effectiveness of this ontology. Based on the interoperability framework of supply chain enterprise information systems, it stated the interoperability principle of enterprise information systems by an order instance, and then developed the prototype system of supply chain information system interoperability.

The innovation the work:

(1) It proposed the model-driven ontology development method that make automatic transformation from UML concept model to OWL ontology and add semantic information. It provided an effective method for ontology reuse, and accelerated ontology development progress.

(2) It proposed an method to build product ontology based on international standards IEC62264 and ISO 10303 STEP-PDM and the method to build supply chain ontology based on Supply-Chain Operations Reference-model. It could improve practical operation for ontology standardization, and have very strong capability to solve the difficulties about the practical application of ontology which was built by traditional methods.

(3) It adopted ontology merging technology to merge ONTO-PDM product ontology and SCOR supply chain ontology into the shared product-centric supply chain ontology. Taking this ontology as core, it proposed the product-centric supply chain information system interoperability framework, and achieved system

interoperability by mapping technology, which provided the feasible solution for the information integration of complex and heterogeneous systems.

7.2 Future works

Currently, global manufacturing and global supply chain are the main reality and trend. With the advance of standardization course, more and more large-scare enterprise application systems have adopted standard data structures and interfaces gradually, which provide foundation for the interoperability of enterprise heterogeneous systems. However, semantic interoperability is still a urgent issue to be solved. The research work of the dissertation acquired some useful results for the semantic interoperability of enterprise information systems. But due to limitation of time and other conditions, some potential works of this research should be further studied and completed. Look forward to the future research work:

(1) The transformation from UML model to OWL ontology in the dissertation is mainly based on the constituent elements and related constraint relationships of the two languages. Because the UML model in the dissertation is not involved in OCL constraint, how to transform OCL constraint into OWL is not considered, which need further research.

(2) The ontology development tool Protégé do not support the export of UML format currently. OWL ontologies applied in practical systems should be secondary developed by some Java API such as Jena and OWLAPI et al, which application development efficiency is low, and the system operating efficiency also need be further improved. Further development of ODM establishes the foundation for the standard transformation between ontology languages and software languages such as UML and XML. Develope the automatic mapping transformation method from OWL and RDF ontology to UML model would be helpful to improve the practical application of ontology, and curtail the time of ontology from laboratory to engineering field.

(3) The Supply-Chain Operations Reference-model is in developing, and the related process import and export and best implementation of SCOR model are mostly described by language, with no standard terms. Supply chain optimization is very important for improving the efficiency of whole enterprise network. With the further perfection of SCOR model, research deeply on best implementation is significant for

improving the efficiency of whole supply chain.

(4) The system developed in the dissertation is only a testing system prototype, and it need be more further perfected when put into practical application, which require concerning about many factors such as the operation efficiency of system and user experience et al.

Reference

[Ting-bin Chen, 2005] Ting-bin Chen. Research on Technology and Methods of Information Integration for Supply Chain Based on XML and Web Services [D].Dalian University of Technology, 2005.

[Xin-jian Gu, 2003] Xin-jian Gu,Chang-yue Ge,Yong-sheng Han. Discussion on the integrated manufacturing development strategy of Chinese enterprises group [J]. Group Technology & Production Modernization. 2003, 20(3).

[Xin-jian Gu, 2001] Xin-jian Gu, Xiang-hong Xu, Zi-chen Chen. Research on Network-based Manufacturing Paradigm [J]. China Industrial Economy, 2001 (1):30-34.

[Xin-jian Gu, 2002] Xin-jian Gu, Guo-jun Zheng, Yong Di. Agile supply chain management systems for mass customization) [J]. Engineering Geology Computer Application, 2002.

[Yi-hua Ni, 2005] Yi-hua Ni. The Research of Knowledge Integration Technology Based on Ontology for Manufacturing Enterprise [D]. Zhejiang University, 2005.

[Guo-ning Qi, 2001] Guo-ning Qi，Xin-jian Gu. New models and new methods of Manufacturing systems [J]. Electric Age, 2001(12).

[Guo-ning Qi, 2004] Guo-ning Qi，Hai-cheng Yang. Research on the Developing Trends of Manufacturing Informatization Technology. China Mechanical Engineering [J],2004(10): 1693-1696.

[Ming-gang Sui, 2002] Ming-gang Sui,，Jia-zheng Huo，Zhong-ying Liu. Integrated Supply Chain and the Status Quo of the Research [J]. Industrial Engineering and Management，2002.

[Hai-bo Hai, 2010] Hai-bo Hai. Business Cooperation-oriented Heterogeneous System Integration Platform and Its Implementatio. Zhejiang University[D], 2010.

[Bu-tong Zhang, 2003] Bu-tong Zhang，Ting-bin Chen. Multi-Agent Based Intelligent Integration and Decision of Supply Chain [J]. Management Science in China，2003,16(6) :52-56.

[Yun-bo Zhang, 2004] Yun-bo Zhang，Zhen-ye Wu，Cheng-lian Yang. Integrated System Model for Flexible Supply Chain. Journal of Southwest JiaoTong University [J],2004,39(2).

[Yong-jun Sun, 2002] Yong-jun Sun，Zheng-xiao Wang，Xiao-hong Pan et al. Research on Integrated Agile Supply Chain Modeling Method [J]. China Mechanical Engineering, 2002,13(19).

[An, 2006] An Y, Borgida A and Mylopoulos J. Discovering the semantics of relational tables through mappings[J]. Journal on Data Semantics VII, 2006, 7:1-32.

[Anthony, 2006] Anthony H and Rupert S. A knowledge-based approach to merging information[J]. Knowledge-Based Systems, 2006,19(8):647-674.

[Auzelle, 2009] Auzelle J P. Proposal a product-centred multi-level model for information system in enterprise[D]. University of Nancy, France, 2009.

[Atkinson, 2003] Atkinson C and Kühne T. Model-driven development: A metamodeling foundation[J]. IEEE Software, 2003, 20(5):36-41.

[ATLAS, 2006] ATLAS Group. The ATLAS Transformation Language. ATL User Manual version 0.7[EB/OL], 2006. Available: http://www.eclipse.org/gmt.

[Baader, 2003] Baader F, Calvanese D, McGuinness D L, and et al. The Description Logic Handbook: Theory, Implementation, Applications[M]. Cambridge University Press, UK, 2003.

[Bach, 2004] Bach T, Rose D and Gandon F. On ontology matching problems (for building a corporate semantic web in a multi-communities organization)[C]. Proceedings of 6th International Conference on Enterprise Information Systems (ICEIS), Porto, 2004:236-243.

[Barua, 2011] Barua A. Supply Chain Integration[P]. Patent Application Publication, USA, 2011.

[Baclawski, 2002a] Baclawski K, Kokar M, Kogut P, and et al. Extending the unified modeling language for ontology development[J]. Software and Systems Modeling, 2002, 1(2):142-156.

[Baclawski, 2002b] Baclawski K, Kokar M, Smith J E, and et al. UOL: Unified Ontology Language: Assorted Papers Discussed at the DC Ontology SIG Meeting [EB/OL], 2002. Available: http://www.omg.org/cgi-bin/doc?ontology/2002-11-02.

[Bermúdez, 2007] Bermúdez J, Goñi A, Illarramendi A, and et al. Interoperation among agent-based information systems through a communication acts ontology[J]. Information Systems. 2007,32(8):24.

[Bezivin, 2005] Bézivin J. On the unification power of models. Software and System Modeling[J], 2005, 4(2):171-188.

[Bellatreche, 2006] Bellatreche L, Dung N X and Pierra G. Contribution of ontology-based data modeling to automatic integration of electronic catalogues within engineering databases[J]. Computers in Industry, 2006,57(8): 711-724.

[B2MML, 2008] B2MML. Business To Manufacturing Markup Language Batch Markup Language Version 0401[EB/OL], 2008. Available: http://wbforg.affiniscape.com/displaycommon.cfm?an=1&subarticlenbr=99

[Booch, 2005] Booch G, Rumbaugh J and Jacobson I. The Unified Modeling Language User Guide 2th ed[M]. Addison Wesley Professional Press, 2005.

[Bergamaschi, 2001] Bergamaschi S, Castano S, Vincini M, and et al. Semantic integration of heterogeneous information sources[J]. Special Issue on Intelligent Information Integration, Data & Knowledge Engineering, 2001, 36(1):215-249.

[Bruijn, 2003] Bruijn J, Mart´ın-Recuerda F, Manov D, and et al. State-of-the-art survey on Ontology Merging and Aligning V1[R]. SEKT: Semantically Enabled Knowledge Technologies, 2003.

[Bouquet, 2003] Bouquet P, Serafini L, and Zanobini S. Semantic coordination: A new approach and an application[C]. Proceedings of 2nd International Semantic Web Conference (ISWC), Sanibel Island (FL US), 2003: 130-145.

[Boardman, 2006] Boardman J and Sauser B. System of Systems-the meaning of SoS[C]. Proceedings of 1st IEEE/SMC International Conference on System of Systems Engineering, John Wiley & Sons. 2006: 4-10.

[Bruijn, 2006] Bruijn J, Ehrig M, Feier C and et al. Ontology mediation. Merging and aligning[M]. Semantic Web Technologies: Trends and Research in Ontology-based Systems, 2006.

[Budanitsky, 2001] Budanitsky A and Hirst A. Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures[C]. Proceedings of NAACL 2001 Workshop on WordNet and other lexical resources, 2001.

[Castano, 2000] Castano S, Antonellis V D, and Sabrina D. Global viewing of heterogeneous data sources[J]. IEEE Transactions on Knowledge and Data Engineering, 2000, 13(2):277-297.

[Castano, 2006] Castano S, Ferrara A, and Montanelli S. Matching ontologies in open networked systems: Techniques and applications[J]. Journal on Data Semantics, 2006, 5:25–63.

[Carlock, 2001] Carlock P G and Fenton R E. System-of-Systems (SoS) Enterprise Systems for Information-Intensive Organizations[J]. Systems Engineering, 2001, 4:242-261.
[Chalupksy, 2000] Chalupksy H. OntoMorph: a translation system for symbolic knowledge[C]. Proceedings of the 17th International Conference on Knowledge Representation and Reasoning (KR-2000), 2000.
[Chang, 2005] Chang K, He B and Zhang Z. Toward large scale integration: Building a metaquerier over databases on the web[C]. Proceedings of 2nd Biennial Conference on Innovative Data Systems Research (CIDR), Asilomar (CA US), 2005, 44-55.
[Chen, 2006] Chen Q, Shen J, and et al. Building a Collaborative Manufacturing System on an Extensible SOA-based Platform[C]. Proceedings of 10th International Conference on Computer Supported Cooperative Work in Design, 2006: 1-6.
[Clark, 1999] Clark T and Jones R. Organisational Interoperability Maturity Model for C2. Proceedings of the Command and Control Research And Technology Symposium (CCRTS), Newport, RI, USA, 1999.
[C4ISR, 1998] C4ISR. C4ISR Architectures Working Group report[R]. Levels of Information Systems Interoperability (LISI), Washington, 1998.
[Clifton, 1997] Clifton C, Hausman E, and Rosenthal A. Experience with a combined approach to attribute matching across heterogeneous databases[C]. Proceedings of 7th IFIP Conference on Database Semantics, Leysin (CH), 1997:428-453.
[Cranefield, 2001a] Cranefield S. Networked knowledge representation and exchange using UML and RDF[J]. Journal of Digital Information, 2001. Available: http://jodi.ecs.soton.ac.uk/Articles/v01/i08/Cranefield/.
[Cranefield, 2001b] Cranefield S. UML and the Semantic Web[C]. Proceedings of the Semantic Web Working Symposium, Stanford University, 2001:113-130.
[Damodaran, 2004] Damodaran S and Ana S. B2B integration over the Internet with XML: RosettaNet successes and challenges[C]. Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters, 2004.
[DeVos, 2001a]deVos A and Rowbotham C T. Knowledge representation for power system modeling[C]. Proceedings of the 22nd International Conference on Power Industry Computer Applications, Sydney, 2001:50-56.
[DeVos, 2001b] deVos A, Widergren S E and Zhu J. XML for CIM Model Exchange[C]. Proceedings of the 22nd International Conference on Power Industry Computer Applications, Sydney, 2001:31-37.

[DeLaurentis, 2004] DeLaurentis D and Callaway R. A System-of-Systems Perspective for Public Policy Decisions[J]. Review of Policy Research, 2004, 21(6):829-837.
[Devedžić, 2002] Devedžić V. Understanding ontological engineering[M]. Communications of the ACM, 2002, 45(4):136-144.
[DUET, 2008] DUET. DAML-UML Enhanced Tool (DUET) Documents [EB/OL], 2008.Available:http://codip.grci.com/wwwlibrary/wwwlibrary/wwwlibrary/DUET_Docs/.
[Do, 2002] Do H and Rahm E. COMA-a system for flexible combination of schema matching approaches[C]. Proceedings of 28th International Conference on Very Large Data Bases (VLDB), Hong Kong (CN), 2002: 610-621.
[Doan, 2002] Doan A, Madhavan J, Domingos P, and et al. Learning to map between ontologies on the semantic web[C]. Proceedings of the 11th International World Wide Web Conference (WWW 2002), 2002:662-673.
[Dou, 2005] Dou D, McDermott D, and Qi P. Ontology translation on the semantic web[J]. Journal on Data Semantics, 2005:2:35-57.
[Ding, 2002] Ding Y, Fensel D, Michel C A and et al. The semantic web: yet another hip? [J]. Data Knowledge Engineering, 2002, 41(2-3):205-227.
[Ehrig, 2004] Ehrig M and Sure Y. Ontology mapping-an integrated approach[C]. Proceedings of 1st European Semantic Web Symposium (ESWS), Hersounisous (GR), 2004:76-91.
[Euzenat, 2001] Euzenat J. Towards a principled approach to semantic interoperability[C]. Proceedings of the IJCAI-01 Workshop on Ontologies and Information Sharing, Seattle, USA, 2001, 47:19-25.
[Euzenat, 2007] Euzenat J and Shvaiko P. Ontology Matching[M]. Springer Berlin Heidelberg press, New York, 2007.
[EIF, 2004] EIF. European Interoperability Framework for pan-European eGovernment Services, Interoperable Delivery of European eGovernment Services to public Administrations[EB/OL]. Businesses and Citizens (IDABC), Luxembourg, 2004.Available: http://ec.europa.eu/idabc/servlets/Docd552.pdf?id=19529
[Fayez, 2005] Fayez M, Rabelo L and Mollaghasemi M. Ontologies for supply chain simulation modelling[C]. Proceedings of 37th Conference on Winter Simulation Conference, Orlando, Florida, 2005:4-7.
[Farquhar, 1997] Farquhar A, Fikes R and Rice J. The Ontolingua Server: A Tool for

Collaborative Ontology Construction[J]. International Journal of Human-Computer Studies, 1997, 46:707-728.
[Falkovych, 2003] Falkovych K, Sabou M, Stuckenschmidt H. UML for the Semantic Web: Transformation-based approaches[C]. Int. Conf. on Knowledge Transformation for the Semantic Web, 2003, 95: 92-106.
[Farinha, 2007] Farinha F, Jardim-Goncalves R and Steiger-Garcao A. Integration of cooperative production and distributed design in AEC[J]. International Journal Advances in Engineering Software, 2007.
[Fernández-López, 1999] Fernández-López M, Gómez-Pérez A, Sierra J P and et al. Building a chemical ontology using Methontology and the Ontology DesignEnvironment[J]. IEEE Intelligent Systems, 1999, 14(1):37-46.
[Fox, 1993] Fox M S, Chionglo J F, and et al. The integrated supply chain management system, Internal Report[R]. Dept. of Industrial Engineering, Univ. of Toronto, 1993.
[Gangemi, 2003] Gangemi A, Guarino N, Masolo C, and et al. Sweetening WordNet with DOLCE[J]. AI Magazine, 2003, 24(3):13-24.
[Gaševic, 2009] Gaševic D, Djuric D and Devedžic V. Model Driven Engineering and Ontology Development[M]. SpringerVerlag press, Berlin Heidelberg, 2009.
[Genesereth, 1992] Genesereth M R and Fikes R E. Knowledge Interchange Format, Version 3.0, Reference Manual [EB/OL], 1992. Available: http://www-ksl.stanford.edu/knowledge-sharing/papers/kif.ps.
[Gibb, 2002] Gibb B K and Damodaran S. ebXML: Concepts and Application[M]. John Wiley & Sons, Inc. New York, NY, USA, 2002.
[Giunchiglia, 2004] Giunchiglia F and Shvaiko P. Semantic Matching. In The Knowledge Engineering Review journal, 20041, 8(3):265-280.
[Gruber, 1993] Gruber T R. A translation approach to portable ontology specifications[J]. Knowledge Acquisition, 1993, 5(2):199-229.
[Gruber, 1995] Gruber T R. Toward principles for the design of ontologies used for knowledge sharing[J]. International Journal of Human-Computer Studies, 1995, 43(5-6):907-928.
[Gruber, 2005] Gruber T R. Ontology of Folksonomy: A Mash-up of Apples and Oranges[EB/OL].Available:http://tomgruber.org/writing/ontology-of-folksonomy.htm.
[Giunchiglia, 2004] Giunchiglia F, Shvaiko P, and Yatskevich M. S-match: an algorithm and an implementation of semantic matching[C]. Proceedings of ESWS'04,

number 3053 in LNCS, Heraklion, Greece, 2004: 61-75.
[Giunchiglia, 2004] Giunchiglia F and Shvaiko P. Semantic Matching[J]. The Knowledge Engineering Review journal, 2004, 18(3):265-280.
[Hovy, 1998] Hovy E. Combining and standardizing large-scale, practical ontologies for machine translation and other uses[C]. Proceedings of. 1st International Conference on Language Resources and Evaluation (LREC), Granada (ES), 1998: 535-542.
[Haase, 2007] Haase P, Brockmans S, Palma R, and et al. Updated Version of the Networked Ontology Model. NeOn Project Deliverable D1.1.2[EB/OL], 2007. Available:http://www.neon-project.org/web-content/index.php?view=weblink&catid=17%3Aproject-reports&id=86%3Ad112-networkedontology-model-eval&option=com_weblinks.
[Haller, 2008] Haller A, Gontarczyk J and Kotinurm P. Towards a complete SCM ontology: the case of ontologisingRosettaNet[C]. Proceedings of the 2008 ACM Symposium on Applied Computing, Ceara, Brazil, 2008, 1467-1473.
[Hegge，1995] Hegge H. Intelligent Product Descriptions for business applications[D]. Eindhoven University of Technology, Netherlands, 1995.
[Hillairet, 2007] Hillairet G.. ATL Use Case-ODM Implementation (Bridging UML and OWL) [EB/OL], 2007. Available:
http://www.eclipse.org/m2m/atl/usecases/ODMImplementation/
[Horrocks, 2003] Horrocks I, Patel-Schneider P F and vanHarmelen F. From SHIQ and RDF to OWL: the making of a web ontology language[J]. Journal of Web Semantics , 2003, 1(1):7-26.
[Horrocks, 2004] Horrocks I, Patel-Schneider P F, Boley H and et al. SWRL: A Semantic Web Rule Language Combining OWL and RuleML[EB/OL]. W3C Member Submission, 2004. Available: http://www.w3.org/Submission/SWRL/.
[IEC, 2002] IEC. Enterprise-control system integration. Part 1: Models and terminology; Part 2: Model objectattributes[S]. ISO/IEC FDIS Standard, Genèva, Switzerland, 2002.
[IEC, 2003] IEC. International standard IEC 62264-1, 1th ed [S]. Genève, Switzerland, 2003.
[IEEE 610, 1992] IEEE 610. IEEE Std 610-IEEE Standard Glossary of Software Engineering Terminology[S]. Institute of Electrical and Electronics Engineers, 1990.

[IEEE, 1995] IEEE. Institute of Electrical and Electronics Engineers: IEEE Standard Computer Dictionary[S]. A Compilation of IEEE Standard Computer Glossaries, New York, 1990.
[ISO 16100, 2002] ISO 16100. Manufacturing Software Capability Profiling for interoperability[S]. Part 1: Framework, TC 184/SC5/WG4, Geneva, Switzerland, ICS 25.040.01, 2002.
[ISO 14528, 1999] ISO 14528. Industrial Automation Systems-Concepts and rules for Enterprise Models[S], TC 184/SC5/WG1, Geneva, Switzerland, 1999.
[Jagdev, 2001] Jagdev H S and Thoben K D. Anatomy of enterprise collaborations[J]. Production Planning and Control, 2001, 12(5):437-451.
[Jardim-Goncalves, 2006] Jardim-Goncalves R, Panetto R and Nuñez M J et al. SMART-fm: setting interoperability in SME-based industrial environments, Advanced Manufacturing: An ICT & Systems Perspective [M]. Taylor and Francis, 2006.
[Jardim-Goncalves, 2011] Jardim-Goncalves R et al. Standards Framework for Intelligent Manufacturing Systems Supply Chain[J]. Supply Chain Management-New Perspectives, Sanda Renko (Ed.), 2011:555-580.
[John, 2001] John D P. Model-Driven Architecture: Vision, Standards And Emerging Technologies[C]. Workshop on Metamodeling and Adaptive Object Models, Budapest, Hungary, 2001.
[Jacob, 2006] Jacob K, Stephan P, Michael S, and et al. Ontology based text indexing and querying for the semantic Web[J]. Knowledge-Based Systems, 2006, 19(8): 744-754..
[Jilovec, 2004] Jilovec N. EDI, UCCnet and RFID: Synchronizing the Supply Chain[M]. 29th Street Press, Loveland, CO, 2004.
[Jouault, 2008] Jouault J, Allilairea F, Bézivin J, and et al. ATL: A model transformation tool[J]. Science of Computer Programming, 2008, 72:31-39.
[Jian, 2005] Jian N, Hu W, Cheng G, and et al. Falcon-AO: Aligning Ontologies with Falcon[C]. K-Cap 2005 Workshop on Integrating Ontologies, 2005.
[Jiang, 1997] Jiang J J and Conrath D W. Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy[C]. Proceedings of International Conference Research on Computational Linguistics (ROCLING X), Taiwan, 1997.
[Kalfoglou, 2002] Kalfoglou Y and Schorlemmer M. Information-flow-based ontology mapping[C]. Proceedings of the 1st International Conference on Ontologies, Databases, 2002.

[Kalfoglou, 2003] Kalfoglou Y and Schorlemmer M. Ontology mapping: the state of the art[J]. The Knowledge Engineering Review, 2003, 18:1-31.
[Kim, 2005] Kim J, Jang M, Ha Y, and et al. MoA: OWL ontology merging and alignment tool for the semantic web[C]. Proceedings of 18th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE), Bari (IT), 2005:722-731.
[Kiryakov, 2001] Kiryakov A, Simov K and Dimitrov M. OntoMap: portal for upper-level ontologies[C]. Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS'01), 2001:47-58.
[Kaihara, 2003] Kaihara T. Multi-agent based supply chain modeling with dynamic environment[J]. International Journal of Production Economics, 2003, 85(2):263-269.
[Kirchmer, 2004] Kirchmer M. E-business process networks–successful value chains through standards[J]. Journal of Enterprise Information Management, 2004, 17(1):2-30.
[Kogut, 2002] Kogut P, Cranefield S, Hart L, and et al. UML for Ontology Development[J]. The Knowledge Engineering Review, 2002, 17(1):61-64.
[Kotov, 1997] Kotov V. Systems-of-Systems as Communicating Structures[R]. Hewlett Packard Computer Systems Laboratory, Paper HPL-97-124, 1997:1-15.
[Kollberg, 2006] Kollberg M and Dreyer H. Exploring the impact of ICT on integration in supply chain control: a research model[C]. Proceedings of the 13th International EurOMA Conference “Moving up the Value Chain", Glasgow, UK, 2006
[Klein, 2001] Klein M. Combining and relating ontologies: an analysis of problems and solutions[C]. In Workshop on Ontologies and Information Sharing, IJCAI'01, Seattle, USA, 2001.
[Lenat, 1990] Lenat D and Guha R. Building large knowledge based systems[J]. Addison Wesley, Reading (MA US), 1990.
[Lindner, 2009] Lindner C. Supply Chain Performance Measurement: A research of occurring problems and challenges[D], Jonkoping International Business School, 2009.
[Lockamy, 2004] Lockamy A and McCormack K. Linking SCOR planning practices to supply chain performance: An exploratory study[J]. International Journal of Operations & Production Management, 2004, 24(12):1192-1218
[Lacher, 2001] Lacher M and Groh G. Facilitating the exchange of explicit knowledge through ontology mappings[C]. Proceedings of the 14th International FLAIRS

Conference, 2001.
[Lee, 2002] Lee M L, Yang L H, Hsu W, and et al. XClust: clustering XML schemas for effective integration[C]. Proceedings of 11th International Conference on Information and Knowledge Management (CIKM), McLean (VA US), 2002:292-299.
[Lerner, 2000] Lerner B S. A model for compound type changes encountered in schema evolution[J]. ACM Transactions on Database Systems, 2000, 25(1):83-127.
[Madhavan, 2001] Madhavan J, Bernstein P and Rahm E. Generic schema matching with Cupid[C]. Proceedings of 27th International Conference on Very Large Data Bases (VLDB), Roma (IT), 2001: 48-58.
[Maedche, 2000] Maedche A and Staab S. Semi-automatic engineering of ontologies from texts[C]. Proceedings of the 12th International Conference on Software Engineering and Knowledge Engineering (SEKE 2000), 2000:231-239.
[MacGregor, 1991] MacGregor R. Inside the LOOM classifier[J]. ACM SIGART Bulletin-Special issue on implemented knowledge representation and reasoning systems, 1991, 12(3):70-76.
[Mena, 1998] Mena E, Kashyap V, Illarramendi A, and et al. Domain Specific Ontologies for Semantic Information Brokering on the Global Information Infrastructure[C]. Proceedings of the 1st International Conference on Formal Ontology in Information Systems (FOIS'98), 1998:269–283.
[Miller, 2000] Miller R, Haas L and Hernandez M. Schema mapping as query discovery[C]. Proceedings of 26th International Conference on Very Large Data Bases (VLDB), Cairo (EG), 2000: 77-88.
[Milo, 1998] Milo T and Zohar S. Using schema matching to simplify heterogeneous data translation[C]. Proceedings of 24th International Conference on Very Large Data Bases (VLDB), New York (NY US), 1998:122-133,.
[Mitra, 1999] Mitra P, Wiederhold G, and Jannink J. Semi-automatic integration of knowledge sources[C]. Proceedings of 2nd International Conference on Information Fusion, Sunnyvale (CA US), 1999: 572-581.
[Mitra, 2002] Mitra P and Wiederhold G. Resolving terminological heterogeneity in ontologies[C]. Proceedings of the ECAI'02 workshop on Ontologies and Semantic Interoperability, 2002.
[Mitra, 2005] Mitra P, Noy N and Jaiswal A. Ontology mapping discovery with uncertainty[C]. Proceedings of 4th International Semantic Web Conference (ISWC), Galway (IE), 2005: 537-547.

[Modica, 2001] Modica G, Gal A, and Jamil H. The use of machinegenerated ontologies in dynamic information seeking[C]. Proceedings of 9th International Conference on Cooperative Information Systems (CoopIS), Trento (IT), 2001:433-448.

[MDA, 2003] MDA. MDA Guide Version 1.0.1[EB/OL], 2003. Available: http://www.omg.org/cgi-bin/doc?omg/03-06-01.

[Molina, 2007] Molina A, Chen D, Panetto H and et al. Enterprise Integration and Networking: Issues，Trends and Vision[C]. IF AC TC 5.3 Enterprise Integration and Networking, 2007.

[Millet, 2009] Millet P A, Schmitt P and Botta-Genoulaz V. The SCOR model for the alignment of business processes and information systems[J]. Enterprise Information Systems, 2009, 3(4):393-407.

[Morel, 2007] Morel G, Panetto H, Mayer F And et al. System of enterprise-Systems integration issues: an engineering perspective[C]. IFAC Cost Effective Automation in Networked Product Development and Manufacturing, Monterrey, Mexico, 2007.

[Muñoz, 2011] Muñoz P and Jesús D. The ISO/EN 13606 Standard for the Interoperable Exchange of Electronic Health Records[J]. Journal of Healthcare Engineering, 2011, 2(1):1-25.

[Nambiar, 2010] Nambiar A N et al. USA Traceability in agri-food sector using RFID[J]. Information Technology (ITSim), 2010, 874-879.

[Natalya, 2001] Natalya F N and Deborah L M. Ontology Development 101. A Guide to Creating Your First Ontology[R], Stanford K, Issue: Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, 2001:1-25.

[Natalya, 2003] Natalya F N and Mark A M. The PROMPT suite: Interactive tools for ontology merging and mapping[J]. International Journal of Human-Computer Studies, 2003, 59(6):983-1024.

[Nissen, 2002] Nissen M E. Agent-based supply chain integration[J]. Information Technology and Management, 2002, 2:289-312.

[Neely, 1995] Neely A, Gregory M and Platts K. Performance measurement systems design: a literature review and research agenda[J]. International Journal of Operations & Production Management, 1995, 15(4):80–116.

[Niles, 2001] Niles I and Pease A. Towards a standard upper ontology[C].

Proceedings of 2nd International Conference on Formal Ontology in Information Systems (FOIS), Ogunquit (ME US), 2001:2–9.
[Nurmilaakso, 2002] Nurmilaakso J M, Kettunen J and et al. XML-based supply chain integration: a case study[J]. Integrated Manufacturing Systems, 2002, 13(8):586-595.
[Nurmilaakso, 2004] Nurmilaakso J M and Kotinurmi p. A review of XML-based supply-chain integration[J]. Production Planning & Control: The Management of Operations, 2004, (15):608-621.
[Narasimhan, 2002] Narasimhan R and Kim S W. Effect of supply chain integration on the relationship between diversification and performance: evidence from Japanese and Korean firms[J]. Journal of Operations Management, 2002, 20(3):303-323.
[NATO, 2003] NATO. NATO Allied Data Publication 34 (ADatP-34)[R]. NATO C3 Technical Architecture (NC3TA), Version 4.0, 2003.
[Omelayenko, 2001] Omelayenko B and Fensel D. A two-layered integration approach for product information in B2B e-commerce[C]. Proceedings of the Second Intenational Conference on Electronic Commerce and Web Technologies (EC WEB-2001), Munich, Germany, 2001.
[ODM，2009] ODM. Ontology Definition Metamodel Version 1.0[EB/OL], 2009. Available: http://www.eclipse.org/m2m/atl/usecases/ODMImplementation/
[O'Reilly, 2005] O'Reilly T. What Is Web 2.0? [EB/OL]. 2005. Available: http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html
[Paslaru, 2005] Paslaru B E and Mochol M. Towards a reuse-oriented methodology for ontology engineering[C]. Proceedings of the 7th International Conference on Terminology and Knowledge Engineering, Copenhagen, Denmark, 2005.
[Panetto, 2006] Panetto H, Jardim-Gonçalves R and Pereira C. E-Manufacturing and Web-Based Technology for Intelligent Manufacturing and Networked Enterprise[J]. Journal of Intelligent Manufacturing, 2006, 17(6):639-640.
[Panetto, 2007] Panetto H. Towards a Classification Framework for Interoperability of Enterprise Applications[J]. International Journal of Computer Integrated Manufacturing 20, 2007, 8:727-740.
[Panetto, 2008] Panetto H and Molina A. Enterprise Integration and Interoperability in Manufacturing Systems: trends and issues[J]. Special issue on Enterprise Integration and Interoperability in Manufacturing Systems. Computers in Industry, 2008, 59(7):641-646.

[Prabir, 2003] Prabir K B and Tage S L. Integration of Information Technology and Organizations in a Supply Chain[J]. International Journal of Logistics Management, 2003, 14(1):89-108.
[Prabir, 2005] Prabir K B, Byoung C H, and et al. Supply chain integration: a European survey[J]. International Journal of Logistics Management, 2005, 16(2):275-294.
[Paolo, 2003] Paolo B, Luciano S and Stefano Z. Semantic Coordination: A New Approach and an Application[R]. 2003:130-145.
[Pei, 2000] Pei R S. Systems-of-Systems Integration (SoSI)–A Smart Way of Acquiring Army C4I2WS Systems[C]. Proceedings of 2000 Summer Computer Simulation Conference, Society for Computer Simulation, Vancouver, 2000:574-579.
[Pepijn, 1998] Pepijn R S and Zhan C. On accepting heterogeneous ontologies in distributed architectures[C]. Proceedings of the ECAI'98 workshop on applications of ontologies and problem-solving methods, Brighton, UK, 1998.
[Pepijn, 1997] Pepijn R S, Dean M, Jones T J and et al. An analysis of ontological mismatches: Heterogeneity versus interoperability[C]. In AAAI 1997 Spring Symposium on Ontological Engineering, Stanford, USA, 1997.
[Palopoli, 2003] Palopoli L, Terracina G, and Ursino D. DIKE: a system supporting the semi-automatic construction of cooperative information systems from heterogeneous databases[J]. Software–Practice and Experience, 2003, 33(9):847-884.
[Pan, 2005] Pan R, Ding Z, Yu Y, and et al. A Bayesian network approach to ontology mapping[C]. Proceedings of 3rd International Semantic Web Conference (ISWC), Hiroshima (JP), 2005:563-577.
[Qi, 2006] Qi H and Tok W L. An ontology based approach to the integration of entity–relationship schemas[J]. Data & Knowledge Engineering, 2006, 58(3):299-326.
[Ruinan, 2006] Ruinan G et al. Business process collaboration using semantic interoperability: Review and framework[C]. Proceedings of the Semantic Web-ASWC 2006-First Asian Semantic Web Conference, 2006.
[Rahm, 2001] Rahm E and Bernstein P A. A survey of approaches to automatic schema matching[J]. VLDB Journal: Very Large Data Bases, 2001, 10(4):334-350.
[Sakka, 2010] Sakka O, Millet P A and Botta-Genoulaz V. A OWL based ontology of SCOR model: a prerequisitie for strategic alignment[C]. Proceedings of 8th International Conference of Modeling and Simulation (MOSIM'10), Hammamet, Tunisia, 2010.

[SCC, 2001] SCC. Supply chain operations reference model 7.0. Supply Chain Council[EB/OL]. Available: http://www.supply-chain.org.

[SCOR, 2005] Supply Chain Operations Reference Model Version 7.0[EB/OL]], 2005. Available: http://www.supply-chain.org.

[Sheer, 2005] Sheer A W and Schneider K. ARIS architecture of integrated Information Systems[M]. Handbook of Information Systems, Springer, Berlin, 2005.

[Shepherd, 2006] Shepherd C and Gunter H. Measuring supply chain performance: current research and future directions[J]. International Journal of Productivity and Performance Management, 2006, 55(3/4):242-258.

[Sean, 2004] Sean B, Frank V H, Jim H, and et al. OWL Web Ontology Language Reference[EB/OL]. W3C Recommendation 10 February 2004. Available: http://www.w3.org/TR/owl-ref/.

[Stanley, 2001] Stanley E F. Achieving World-Class Supply Chain Alignment: Benefits, Barriers, and BridgesCenter for Advanced Purchasing Studies[D]. Arizona: Tempe, Orlando, Florida, 2005.

[Stanley, 2002] Stanley E F. and Gregory M M. The rhetoric and reality of supply chain integration[J]. International Journal of Physical Distribution & Logistics Management, 2002, 32(5):339-361.

[STEP, 2004] STEP. ISO/TS 10303 STEP modules related to Product Data Management[S]. Industrial automation systems and integration-Product data representation and exchange, Geneva, 2004.

[Stephen, 1999] Stephen C and Martin P. UML as an ontology modelling language[C]. Proceedings of the Workshop on Intelligent Information Integration 16th International Joint Conference on Artificial Intelligence IJCAI99 (1999), Germany, 1999.

[Stumme, 2001] Stumme G and Maedche A. Ontology merging for federated ontologies on the semantic web[C]. Proceedings of the International Workshop for Foundations of Models for Information Integration (FMII-2001), 2001.

[Stewart, 1997] Stewart G. Supply-chain operations reference model (SCOR): the first cross-industry framework for integrated supply-chain management[J]. Logistics Information Management, 1997, 10(2):62-67.

[Sage, 2001] Sage A P and Cuppan C D. On the Systems Engineering and Management of Systems of Systems and Federations of Systems[J]. Information, Knowledge, Systems Management, 2001, 2(4):325-345.

[Sahuguet, 2001] Sahuguet, Arnaud, and et al. Building Intelligent Web Applications

Using Lightweight Wrappers[J]. Data and Knowledge Engineering, 2001, 36(3):283-316.

[Silva, 2006] Silva D F et al. Semantic interoperability of heterogeneous semantic resources[J]. Electronic Notes in Theoretical Computer Science, 2006.

[Thomas, 2004]Thomas E. Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services[R]. Prentice Hall PTR Upper Saddle River, NJ, USA, 2004.

[Terzi, 2007] Terzi S, Panetto H, Morel G and et al. Aholonicmetamodel for product lifecycle management[J]. International Journal of Product Lifecycle Management, 2007, 2(3):253-289.

[Traub, 2005] Traub K, Allgair G, Barthel H, and et al. The EPC global architecture framework[EB/OL]. EPCglobal,2005. Available: http://www.epcglobalinc.org/standards_technology/Final-epcglobal-arch-20050701.pdf.

[Togar, 2004] Togar M S, Alan C W and Ramaswami S. The knowledge of coordination for supply chain integration[J]. Business Process Management Journal, 2004, 8(3):289-308.

[Tolk, 2003] Tolk A. Beyond Technical Interoperability-Introducing a Reference Model for Measures of Merit for Coalition Interoperability[C]. Proceedings of the 8th International Command and Control Research and Technology Symposium (ICCRTS), Washington, 2003.

[Tursi, 2009a] Tursi A. Ontology-based approach for enterprise production systems interoperability[D]. University Henri Poincaré Nancy I, France, 2009.

[Tursi, 2009b] Tursi A, Panetto H, Morel G, and et al. Ontological approach for Products-Centric Information System Interoperability in Networked Manufacturing Enterprises[C]. IFAC Annual Reviews in Control, 2009, 33(2):238-245.

[Uschold, 2000] Uschold M. Creating, integrating, and maintaining local and global ontologies[C]. Proceedings of the First Workshop on Ontology Learning (OL-2000) in conjunction with the 14th European Conference on Artificial Intelligence (ECAI-2000), Berlin, Germany, 2000.

[Velegrakis, 2004] Velegrakis Y, Miller R, Popa L, and et al. ToMAS: A system for adapting mappings while schemas evolve[C]. Proceedings of 20th International Conference on Data Engineering (ICDE), Boston (MA US), 2004.

[Vernadat, 2002] Vernadat F B and Chen D. Enterprise Interoperability: a standardisation view[C]. IFIP International Conference on Enterprise Integration and Modelling Technology (ICEIMT'02), 2002:273-282.

[Vegetti, 2005] Vegetti M, Henning G P and Leone H P. Product ontology: definition of an ontology for the complex product modelling domain. Proceedings of 4th Mercosur Congress on Process Systems Engineering, Rio de Janeiro, Brazil, 2005.

[Wang, 2008] Wang M, Liu J and et al. On-demand e-supply chain integration: multi-agent constraint-based approach. Expert Systems with Applications, 2008, 34(4):2683-2692.

[Weibel, 1995] Weibel S, Gridby J and Miller E. OCLC/NCSA Metadata Workshop Report[R], Dublin, EUA, 1995.Avalabile:
http://www.oclc.org:5046/oclc/research/conferences/metadata/dublin_corereport.html.

[Yiqing, 2009] Yiqing L, Lu L and Chen L. Decision-Making for Supplier Selection Based on Ontology and Rules[C]. Proceedings of 2nd International Conference on Intelligent Computation Technology and Automation, Washington, USA, 2009.

[Zdravković, 2009] Zdravković M and Trajanović M. Integrated Product Ontologies for Inter-Organizational Networks[J]. Computer Science and Information Systems, 2009, 6(2):29-46.

[Zdravković, 2010a] Zdravković M, Panetto H and Trajanović M. Concept of semantic information pool for manufacturing supply networks[J]. International Journal of Total Quality Management and Excellence, 2010, 37(3):69-74.

[Zdravković, 2010b] Zdravković M, Panetto H and Trajanović M. Towards an Approach for Formalizing the Supply Chain Operations[C]. Proceedings of 6th International Conference on Semantic Systems, ACM ICP, Austria, 2010.

[Zeng, 2003] Zeng A Z and Pathak B K. Achieving information integration in supply chain management through B2B e-hubs: concepts and analyses[J]. Industrial Management & Data Systems, 2003, 103(9):657-665.

[Zhang, 2006] Zhang D Z, Anosike A I, and et al. An agent-based approach for e-manufacturing and supply chain integration[J]. Computers&Industrial Engineering, 2006, 51(2):343-360.

[Zhao, 2006] Zhao H and Ram S. Combining schema and instance information for integrating heterogeneous data sources[J]. Data & Knowledge Engineering, 2006: 281-303.