1. CHALLENGE IN SOFTWARE ENGINEERING: SEMANTIC INTEROPERABILITY

With the continuous development of information techniques, information systems are now merging into our life and even becoming an essential and indispensable part in the social infrastructure. From general computing system to complex database management system and Enterprise Resource Planning (ERP) system, various kinds of information systems are scattered in different application domains and areas so that they can be typically deemed as distributed systems. Meanwhile, rapid progress of information techniques drives the evolution of software development paradigm. More specifically, with the shifting from object-oriented techniques to component based development method, not only the granularity of software modular grows bigger, but also the middle-ware technologies and component-based development become the main and popular techniques for software development. Furthermore, the raising of web services (Curbera, 2001; Newcomer E., 2002), service-oriented architecture (SOA) (Thomas, 2004; Newcomer, 2005), and semantic web (Berners-Lee, 2001; Daconta, 2003) cause great changes in both ingredient and development method of web-based information systems. On one hand, web services and semantic web services with greater granularity and more complicated structure are now regarded as the unit of current information systems; on the other hand, it is recommended to create information systems by dynamically linking and integrating existing information resources and service resources on the web. This situation leads to the fact that complexity and scale of information systems are expanding increasingly, which consequently brings our research focus on how to ensure efficient interconnection, intercommunication and interoperation between each part of a common information system.

In the realm of software engineering, when the studied objects take radical changes in the scale, the essential problem of our research will change correspondingly. Considering a single information system, the primary task is to achieve the given functionality of it. Otherwise, if the whole functionality of a system will be realized through several information resources or other systems, this kind of system will be created by linking and integrating specified resources. Here, the key issue of developing current information system is how to organize and manage varied
information resources systemically, enhance accessible rate of them, and finally promote knowledge sharing and interchange between them. However, different development methods and platforms make information resources differ in syntax and semantics, which might hamper the understanding and interacting between them. Therefore, effective solutions should be taken as a bridge to connect information resources and implement interoperation between them.

Generally speaking, interoperability is the ability to communicate and share data across programming languages and platforms (ISO, 1993). On information domain, interoperability will be defined as “the ability of two or more systems or components to exchange information and to use the information that has been exchanged (IEEE, 1990)” or “the ability of a collection of communicating entities to (a) share specified information and (b) operate on that information according to an agreed operational semantics (Lisa, 2004)”. While we talk about interoperation between information systems, we should pay more attention to their behavior of interaction with each other. So in terms of the previous definitions on interoperability, interoperation between information systems can be defined as: the capability that the message sent by one information system can be received, understood and processed by the other information system so that they can cooperate with each other to perform a specific task.

Since different information systems adopt different description languages and have different service capability, interoperation between them can be implemented at different levels, as Fig. 1 shows. Actually, people expect information systems can provide stronger service capability: from simple search to intelligent query and even to knowledge discovery, software is expected to provide appropriate services so that they can meet personalized requirements from various kinds of users. For this purpose, information description languages progress rapidly and their capability for representing semantics is strengthened accordingly, which can be used to support interoperation on different levels.

![Fig.1. Information description languages and interoperation](image)

- Data dictionary, Entity-Relationship (ER) model and XML can help data level interoperation.
- Metadata abstract datatype and data element as the meta information of data, which can facilitate syntax and structure based interoperation between information resources.
- Meta-model is used to describe the structure of complex information, which is more complicated than metadata. Meta-model abstracts syntax and structure of various models so that it can also support syntax-based interoperation between information systems. For example, UML (Unified Modeling Language) meta-model defines modeling structure and principles of UML models, which can realize syntax interoperation between them.
- Ontology conceptualizes concepts and the relationship between them within a universe of domain. It expresses explicit meaning of exchanged information and the inherent relation between them. Consequently, ontology is the basis of semantic interoperation.

Suppose that there are two information systems named A and B respectively. If information models they adopted are the same, then the interoperation between them can be conducted in an easy way. That is, the same data descriptive and modeling language can make the system A completely understand the messages sent by system B and then realize the data-level interoperation between A and B. On the other hand, if information models used in two systems are different, it is necessary to create the ontology mapping between them. Generally speaking, the process of modeling information model should obey specified modeling rules and adopt the corresponding modeling languages. Therefore, it is feasible to ensure the syntax interoperation between them based on the common meta-model and mapping rules between these two modeling languages. Furthermore, if the exchanged messages are meaningful, interconnection between these two information systems can be treated as semantic interoperation. Especially, Fig. 1 implies that efficient semantic interoperation will considerably enhance the service capability of systems. So undoubtedly, semantic interoperation is one of the active topics in current research and ontology is the key technique to realize semantic interoperation.

2. THEORY OF ONTOLOGY & META-MODELING

To realize semantic interoperation between diverse information systems and resources, meta-modeling methodology in software engineering and ontology originating from philosophy are merged in this paper as a novel methodology, Theory of Ontology & Meta-modeling (He, 2005a; He, 2005c). First of all, it treats ontology as the classification mechanism of information resources to register and manage them in an orderly manner. Second, meta-modeling technology enables syntax-level interoperation on the abstract meta-level. The most important is that by means of semantic annotation, ontology can provide a comprehensive description, merging structural, syntax and semantic information, for models and meta-models in the meta-modeling architecture. Hence, theory of ontology & meta-modeling can guide effectively the sharing and exchanging of complex information resources on the web, consequently it is also the basis of semantic interoperation between heterogeneous information resources.

2.1 What is Ontology

The concept “Ontology” comes from philosophy, in which it is the study of the kinds of things that exist and can be used to describe the characteristic of “beings”. In 20th century, ontology was formally introduced into computer science domain, involving knowledge engineering,
natural language processing, knowledge representation, etc., because ontology explicitly expresses the implication of concepts in a specific domain. Especially, those concepts with specified implication can unify the common understanding of the universe of domain and then establish the foundation for communications between people.

Well, what is ontology? In information domain, the most popular and widely acceptable definition of ontology is, ontology is an explicit specification of a conceptualization (Gruber, 1993). In this definition, term “conceptualization” means that ontology can be treated as an abstract model of a phenomenon in the given world, and “explicit” implies that ontology specifies the precision of concepts, types of concept and their relations. Furthermore, Borst extended the previous definition as “An ontology is a formal specification of a shared conceptualization (Borst, 1997)”. In his opinion, the term “sharing” illustrates an agreement between ontology users so that different people can get consensus on the same thing and communicate with each other, and “formal” hints that the accurate and precise mathematical description that ontology adopts can be understood and processed by machines to realize human-machine interaction.

Actually, ontologies are content theories about the sorts of objects, properties of objects and relations between objects that are possible in a specified domain of knowledge (Chandrasekaran, 1999). Currently, information resources are now augmenting increasingly and become more and more complicated, so we can use ontology to capture the semantics of information from various sources, give them a concise, uniform and declarative description (Fensel, 2001), which later can be treated as the semantic foundation of knowledge sharing and exchanging between people, people and machine as well as machines.

Due to the strong support that ontology provides to semantic interoperation between information systems, deep researches have been initiated on developing ontology modeling and descriptive languages, creating standard ontology for specific domains, or ontology inference, etc. What’s more, most of them got some progresses in practice. Meanwhile, although issues on ontology registration, ontology evolution management, ontology mapping and ontology-based semantic annotation are very complex, relevant researches have been underway all around the world. More specifically, Karlsruhe University of Germany analyzed the process of ontology evolving and then proposed a base model for ontology lifecycle management (Baclawski, 2002; Stejanovic, 2004; Haase, 2004); State Key Lab. of Software Engineering (SKLSE), Wuhan University, China leaded the project on international standard, ISO/IEC 19763-3(1st Edition), to register administrative information of ontologies (ISO/IEC 19763-3, 2007) (He, 2005b; Wang, 2006). Especially, the developing standard named ISO/IEC 19763-3(2nd Edition) concentrated on evolution information of ontologies that are registered based on the first version(ISO, 2006) (He, 2007b; He, 2006). Therefore, these two versions can collaborate with each other to form a comprehensive solution for ontology management (He, 2007a).

### 2.2 Meta-modeling technique and meta-model

Meta-modeling is a kind of meta computing method for abstract construction and aggregation of complex information structure. It can be used to manage information resources in a normative way and realize deep sharing of resources in Internet. So both academia and industry treat meta-modeling as the key technique to ensure semantic interoperation between
information resources.

If meta-modeling technique is applied to objects in a specific research domain, we can abstract their characteristics, use modeling notations to represent them as models and finally get the corresponding meta-models. And accordingly, the features of meta-models are listed as follows:

(1) generality: the meta-model can define the common characteristics in structure and behavior of multiple base objects;
(2) reversibility: the meta-model can be extracted from the instance model; and instantiation of meta-model is the corresponding normative model.
(3) extensibility: pre-defined extension mechanism, and the meta-model can be extended due to the practical requirements without modifying original structures and constraints;
(4) transformability and mapping ability: mutual transformation between the models based on same meta-model using transformation and mapping rules.

Obviously, the meta-model plays important roles in two aspects: on one hand, it greatly promotes the standardization and unification of models; on the other hand, it supplies with the guarantee for sharing information and interoperability among models, according to the common features among models constructed from the same meta-model. The MOF (Meta Object Facility)(OMG, 2006) is the classic meta-model specification recommended by OMG (Object Manage Group). It is a well-extensible data management framework and defines a well-known four-layer meta-modeling architecture (as shown in Fig.2).

![Fig 2. The four-layer meta-modeling architecture (OMG, 2006)](image)

- M0 Layer: it is the instance layer in MOF, and is composed of abundant desired instances including model instances (for instance, the computer in real world) and application data instances (for example, Tom). Although objects may be greatly different from each other in their states, values etc, their structures and value ranges may have some similarities. Based on these, the model can be gotten by abstraction.
- M1 Layer: it is the model layer, the abstraction of object instances, to describe the common features of the instances of M0 layer, for instance, UML model. Modeling structures may be of analogous, although modeling objects and languages are probably distinct. Thus, the meta-model can be attained by constraining the model structures and concepts.
- M2 Layer: it is meta-model layer, which is extracted the common modeling elements and constraints among elements of various models from M1 layer. For instance, the UML meta-model, CWM meta-model etc.
- M3 Layer: it is metameta-model layer depicting the common features of structures and semantics among meta-models of different modeling languages on the M2 Layer. Consequently, the MOF model, defined by the language based on the meta-language, would be on the layer of M3.

### 2.3 Details of Ontology & Meta-modeling Theory

Ontology theory is a meta theory in nature, and it is constructed based on description logic. Ontology is an explicit specification of a conceptualization. The advantage of ontology can be summarized in two points: on one hand, ontology concept is similar to the thinking of human beings, and easy to understand; on the other hand, ontology theory is based on description logic, which is the formalization basis for the deduction. Thereby, the combination of ontology theory and meta-modeling method in software engineering, which forms the theoretical basis of the meta-modeling, will strengthen the modeling and representation capability of meta-models, and facilitate considerably the semantic sharing of information resources (Cao, 2002). Especially, UML technique and MOF meta-modeling mechanism have already been applied widely in the software engineering community, while ODM (Ontology Definition Meta-model) makes UML Profile as one of feasible techniques to support the ontology development and maintenance, which is the basis for the relevant integration of ontology modeling and meta-modeling theories. Taking Ref. (Brockmans, 2006) as example, it illustrates how to guide the construction of ontology model based on MOF meta-model and UML Profile. The detailed steps are illustrated in Fig.3: (1) defining ODM and UOP based on MOF; (2) defining the visual symbolic notations for OWL DL ontologies through UML Profile, thereby bidirectional mapping can be created between ODM and UOP; (3) getting the two sorts of instances of OWL ontology model and UML model respectively by instantiation, and implementing the transformation between them based on the above defined mapping. Therefore, one of the effective ways to achieve the theory of Ontology & Meta-modeling is that, providing the appropriate mechanism for introducing ontology on the basis of existing meta-modeling theory and technique, and then making use of the rich semantics of ontology model to help us. Two ontology introducing mechanisms are presented as follows:

![Fig3. An Ontology Definition Meta-model of OWL within the MOF Framework (Brockmans, 2006).](image)

(1) Construction method (heavy-weighted method): during the meta-modeling process, a descriptive ontology concept is introduced as a classifier. The meta-model defined in this method is related with ontology model directly, and the meta-model can be affected by the
evolution of ontology model. This method is generally used in the situation in which the meta classes in meta-model are defined by top level ontology for its stability. For instance, OWL-s (Chris, 2006) is a kind of meta-model of Web service, and describes for Web service the related concepts and relationships among them by using OWL concepts, e.g., Class, ObjectProperty, DataProperty, subClass etc. In the OWL-s meta-model, Output and Parameter are the classes, and Output is the subclass of Parameter.

(2) Annotation method (light-weighted method): in the meta-modeling process, the property set of classifier in this meta-model is defined by descriptive ontology introduced by annotation method. The relationship between the ontology model and meta-model defined by this method is a loose-coupled relationship. The properties of classifier in the meta-model are generally annotated by domain ontology and application ontology. For instance, Registry Information Model (RIM) is the meta-model of ebXML (OASIS, 2002), and defines the relations among the core concepts in the electronic business area. For the on-registry object, the relationship between the object and ontology can be built through the three classifiers of Classification, ClassificationScheme, and ClassificationNode. Fig. 4 shows the registry information instances of registry items extended based on RIM, where gasGuzzlerInc is a registry item. Based on ontologies of Geography and Industry, their properties, US and Automotive, can be extended respectively.

Fig4. Registry information instances of registry items based on RIM (OASIS, 2002)

3. META-MODEL FRAMEWORK FOR INTEROPERABILITY (MFI):
INTERNATIONAL STANDARD ISO/IEC 19763

3.1 Overview of MFI
Due to the spread of E-Business and E-Commerce over the Internet, the effective exchange of business transactions and other related information across countries and cultures has been a great concern for people both inside and outside the IT industry. To follow these trends, many
standardization activities have focused on the common facilities or generic modeling methods, such as e-business exchange format (e.g. XMI(OMG, 2000), SOAP(W3C, 2003)), description facility of information resources (e.g. RDF(W3C, 2004), XML(W3C, 2006)), Business process integration facilities (e.g. BPEL or BPMN), Registry facilities (e.g. MDR(ISO, 2004), ebXML R&R(OASIS, 2002)), etc. Obviously, these specifications enable the wide business cooperation between different organizations.

For this purpose, the contents described with these standards should be stored in the registries. Currently, many registries and repositories have been developed by defining suitable meta-models, which can be used to unify the format of registered objects and the relations between them. In general, meta-model is the model describing a series of models. That is, meta-model abstracts and standardizes the common syntax, semantic and structural features of models, modeling tools and modeling rules. Moreover, metaclasses and meta-relationships designated in a meta-model can be generalized as models in different ways; and vice verse, models generated from the same meta-model have inherent relations, and can realize the interoperation to some degree. In recent years, a great number of registries and repositories have been created to meet requirements of various organizations. However, structural and semantic differences in their meta-models hamper effective collaboration among communities. Therefore, a novel facility is needed to promote interoperation between existing heterogeneous registries, especially the interoperation between meta-models they adopt, and generate a harmonized registry federation. To satisfy these requirements, this Meta Model Framework for Interoperability(MFI) family provides the facilities for describing various types of registries or meta-models as a consolidated set of meta-model frameworks and facilitates semantic interoperation between those meta-models.

More specifically, MFI proposes a series of mechanisms to describe, register and manage different types of meta-models and registries, including registration and description facility for various kinds of reusable modeling constructs, description and registration mechanisms for rules of model mapping and transformation to enable the harmonization of registry contents, etc. Fig.5 illustrates the overview of MFI.

**Fig.5. Overview of MFI**
MFI-2 (Core model) (ISO, 2007a) is not only the essential part of MFI family but the foundation for defining other parts of MFI. Core Model designates a set of normative modeling elements and rules in the basic registration mechanism for models, so the rest of MFI family have to inherit some concepts and modeling elements from Core model. MFI-3(Meta-model for ontology registration, MOR) (ISO, 2007b) describes a meta-model that provides a facility to register administrative information of ontologies. MFI-4(Meta-model for model mapping) (ISO, 2007c) can be used to register any sort of mapping rules between objects, such as meta-models, model elements or data elements, so that it can help model and data transformation at different levels. MFI-5(Meta-model for process model registration) (ISO, 2007d; Wang, 2007; Wang, 2008) provides a meta-model to register administrative, structural and semantic information of heterogeneous process models.

3.2 Meta-model for ontology registration

Meta-model for ontology registration (ISO/IEC 19763-3) (ISO, 2007) is part 3 in MFI family, and its main objective is to register and manage administrative information with respect to structure and semantics of ontologies. MFI-3 provides a generic mechanism for ontology to promote semantic interoperation between ontologies, especially ontology-based interoperation between information systems (He, 2005c; Wang, 2005). In 2003, ISO/IEC JTC1 SC32 approved that the work on the project ISO/IEC 19763-3 was to be taken on Prof. He Keqing’s research group of SKLSE, Wuhan University, China, and Prof. He Keqing was requested as the editor of this project. In Dec, 2007, it was officially published as an international standard.

3.2.1 Overall structure of ISO/IEC 19763-3

The differences in ontology descriptive languages and ontology development techniques make it difficult to fulfill semantic interoperation between ontologies. Fortunately, overall structure of ISO/IEC 19763-3 illustrates a comprehensive solution, as Fig. 6 illustrates.

With the vertical view of Fig. 6, ISO/IEC 19763-3 defines a three-layer structure “Ontology Whole- Ontology Component-Ontology Atomic_Construct” to register common information of ontologies. It implies that ontology consists of ontology components and ontology component is composed of ontology atomic construct, which is treated as the smallest ingredient of ontology. Here, the three-layer structure emphasizes only on the language-independent information of ontologies and ignore their differences caused by representations. So for any ontology to be registered, ontology component and ontology atomic construct will respectively correspond to sentences and non-logical symbols (such as concepts, instances) of the ontology.

With the horizontal view of Fig. 6, ISO/IEC 19763-3 also specifies two kinds of ontology, Reference_Ontology Whole(RO) and Local_Ontology Whole (LO) to distinguish different roles that ontology plays in different cases. RO is used to represent common ontology in some domains, which is created and maintained by authorities and/or relevant domain experts to make it stable. Different from RO, LO is often used by particular information systems, which reuses some elements of ROs and makes some changes to meet different needs. As a result, if two LOs are derived from the same RO, there must be some inherent semantic relations
between them. Then the information systems respectively adopting these two LOs can interoperate with each other on the basis of the common RO.

Fig.6. Overall structure of ISO/IEC 19763-3

In addition, LO can also reuse some parts of RO, modify the reused part as well as add new elements in ISO/IEC 19763-3. So in Fig.6, it can be found that Reference_Ontology_Component (ROC) forms RO but LO is composed of two kinds of ontology component: one is ROC that is borrowed from RO directly; the other is Local_Ontology_Component(LOC) that is special for LO and generated by modifying or adding. Similarly, ROC consists of Reference_Ontology_Atomic_Construct(ROAC), while LOC is composed of both ROAC and Local_Ontology_Atomic_Construct(LOAC).

3.2.2 Examples

In this section, a case study in (Wang, 2006) will be demonstrated to show how to realize semantic interoperation between two information systems based on the same RO. In this case, RO M is the ontology that is released by some standardized organization and expressed with RDFs. Ontology M says that “Credit card is used to pay purchases.” and “Customer uses credit card”. Then OWL-based ontology A and ontology B will be adopted by two specified information systems as the corresponding LOs. More specifically, Ontology A reuses “Credit Card” in M and changes “customer” to “buyer”. Here we will take LO A as an example to show that it is feasible not only to use ISO/IEC 19763-3 for registration of ontology, ontology component and ontology atomic construct, but to record the reusing relation between LO and RO.

Table 1 lists the registration information of Ontology A, which means that the name of the registered ontology is “A” and it is described with OWL. Ontology A consists of two LO, i.e ROC01 from ontology M and LOC01 defined by A itself.
Table 1. Registration information of Ontology A

<table>
<thead>
<tr>
<th>Attribute/Reference</th>
<th>Literal/Instance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administration_record</td>
<td>Administraton_Record03</td>
</tr>
<tr>
<td>URI</td>
<td>uri_A</td>
</tr>
<tr>
<td>ontologyName</td>
<td>A</td>
</tr>
<tr>
<td>modelType</td>
<td>OWL</td>
</tr>
<tr>
<td>consistsOf</td>
<td>OID of uri_M#ROC01</td>
</tr>
<tr>
<td></td>
<td>OID of uri_A#LOC01</td>
</tr>
</tbody>
</table>

Table 2 shows the registration information of LOC01. It means that the name of the registered ontology component is LOC01, which contains three ontology atomic constructs, i.e. one concept “buyer” from LO A, two concepts “Credit_Card” and “uses” from RO M.

Table 2. Registration information of Ontology_Component LOC01

<table>
<thead>
<tr>
<th>Attribute/Reference</th>
<th>Literal/Instance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administration_record</td>
<td>Administraton_Record04</td>
</tr>
<tr>
<td>namespace</td>
<td>uri_A#</td>
</tr>
<tr>
<td>sentenceIdentifier</td>
<td>LOC01</td>
</tr>
<tr>
<td>consistsOf</td>
<td>OID of uri_A#buyer</td>
</tr>
<tr>
<td></td>
<td>OID of uri_M#Credit_Card</td>
</tr>
<tr>
<td></td>
<td>OID of uri_M#uses</td>
</tr>
</tbody>
</table>

Table 3 implies the registration information of ontology atomic construct named “Credit_Card”. It is a non-logical symbol from ontology M.

Table 3. Registration information of Ontology_Atomic_Construct “Credit_Card”

<table>
<thead>
<tr>
<th>Attribute/Reference</th>
<th>Literal/Instance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administration_record</td>
<td>Administraton_Record02</td>
</tr>
<tr>
<td>namespace</td>
<td>uri_M#</td>
</tr>
<tr>
<td>nonLogicalSymbol</td>
<td>Credit_Card</td>
</tr>
</tbody>
</table>

4. APPLICATION OF MFI-3 IN THE MANUFACTURING INFORMATIONALIZATION

Informationalization is the trend in the global economic and social development, and the software industry is fundamental and strategic industry in this trend. In the last few years, the environment and policies for the software industry in China are ameliorated gradually, and the growth rate is increased rapidly. The impact of the software industry on the economic development is strengthened. Meantime, it is also a key age for the transformation of software development methods and the adjustment of the software industrial structure and division. The component-based software development (CBSD) is becoming the main stream in the software development methods. The CBSD method provides the technical support for the production of software in industrial way, and has a great impact to the transformation of software development methods and the adjustment of the software industrial structure and division. The
CBSD method is also regarded as one of the key factors for advancing the software industry. With the maturation of related technologies, the software component resources are becoming pervasive over the Internet environment; on the contrary, the research on the software component resource management gets limited achievement. The existing methods and techniques for the software component classification, registration (e.g. ComponentSource, ebxmlrr, UDDI(Accenture, Ariba, Inc., 2000)), repository, and interoperability management are not capable to meet the requirements posed by the complex information structure (attribute, interface) of software component resources (e.g. composition, circulation, management, and reuse), which consequently hinders the development of software enterprises in China.

To this end, it is necessary to investigate the key methods and techniques for software component development and management, to make standards in the international, national and industrial level for the standardization of the production, testing, circulation, and sharing of software components; to establish the infrastructure for software component development for the purpose of application of CBSD methods and techniques. For this purpose, we developed the “Management and Service platform for Semantic Interoperability on Manufacturing Informationalization Software Component Repository (SCR)” based on the ISO standard ISO/IEC 19763-3. This platform can support the encapsulation, classification, registration, repository management of software components. This platform is based on the Client/Server architecture, and provides the programmable interface based on web services, which can be extended flexibly with SOA (Service-Oriented Architecture). The major difference between this platform and other existing software component repositories (e.g. Shanghai software component repository, Peking University JadeBird software component repository) is the ontology-based software component attributes classification, registration and management using ISO/IEC 19763-3 standard, and implementation of semantic (ontology) based software component query and retrieval. Based on above mentioned techniques, this platform is quite open to other platforms, and facilitates the management of semantic interoperability, which provides the reliable infrastructure for the reusing and sharing of heterogeneous software component resources. The detailed architecture of this platform is shown in Fig. 7.

Fig. 7. Overview of software component repository
The detailed architecture of software component repository platform is shown in Fig. 8. First, we specify the domain model in the manufacturing informationalization using ontology technique, and use the ontology as the logical mechanism in the repository storage, which makes the software component repository as a knowledge repository on software component. The semantic-based query and deduction can be executed on this knowledge repository, which implements the query and management of software components, and ensure the synchronized evolution between domain knowledge and information model of software component repository. The other advantage of ontology-based classification and registration is the user-friendly services for users, which realizes the simple declarative language or query interface in natural language. Second, we employ the UML as the uniform notation for the design, development, publication, registration and application process of software component, and tailor the latest UML 2.0 specification to accommodate the characteristics of software component in the domain of KAIMU manufacturing, which integrates the component diagram, state chart diagram, class diagram and use case diagram effectively to represent the interface and behavior of software component. Third, we design the service interface based on web service, which makes the platform independent from operating system and programming languages. This is fundamental for the heterogeneous system integration and third party software component trading. This platform can implement easily the query in natural language due to the employment of ontology technique, which is a great benefit for common user. For example, user queries the software component repository using web service interface, and gets the query result. In this query process, the predefined component ontology and domain ontology can monitor the query optimization and execute the query, i.e. the query can be optimized according to the software component attributes defined in ontology, and filter out the attributes which are not defined in the software component ontology. The software component repository can perform the ontology deduction and query based on the software component knowledge described by ontology, and return the query result according to the query condition.

In the system development, we use Java as the implementation language, and Eclipse for the GUI development of software component repository. We adopt OWL (Web Ontology
Language) as the ontology description language, and Protégé as the ontology editor tool for the development of manufacturing domain ontology and software component ontology. Jena is employed for the ontology model operation and ontology model database. The ontology defined in OWL is parsed into ontology model, and stored in the database. This platform can also support the IBM DB2 database by the extension of the database connection classes (database types, connected users and password, etc) in Jena.

This project is theoretically based on the theory of ontology & meta-modeling, which is a concrete application of the ISO/IEC 19763-3 standard in the semantic interoperability management and service (ontology-based classification, registration, repository management, model mapping, ontology query, etc) for complex information resources integration. As latest information till Oct, 2007, the BOM (Bill of Material) products constructed by software components have been applied in mechanics, electronics, light industry, and textile industry, etc, including 239 enterprises, which advances considerably the technical level of manufacturing enterprises.

REFERENCE


**KEY TERMS & DEFINITIONS**

Metamodel framework for interoperability (MFI): framework for registering artifacts that are based on meta-model and model semantic interoperability: the ability to exchange and use information between two or more entities meta-modeling: a methodology of how to extract common information from models and create a meta-model universe of domain: all those things of interest that are concrete or abstract and that have been, are, or ever might be ontology: description of a universe of domain in a language that a computer can process reference ontology: ontology that is usable and sharable by a community of interest local ontology: ontology that is specialized for defined applications and based on at least one reference ontology