http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet

Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 8 (6): 871-878, 2009
ISSN 1812-5638

Ontology-Based Model for Software Resources Interoperability

Bo Ding and Li-juan Sun
Research Institute of Computer Applied Techniques, Harbin University of Science and Technology,
Harbin 150080, China

Abstract: In this study, we propose an ontology-based framework to provide an integrated view, which could integrate various software resources and realize semantic interoperability between different software resources. Ontologies are divided into shared ontology and domain ontology. The design of shared ontology is described in detail. The shared ontology which has explicit ontological semantics, implements the uniform representation of heterogeneous information and helps to shield the heterogeneity of software resources systematically. The domain ontology is a domain-specific functional design ontology repository, in which, the invoking functions of the specific platform is encapsulated. The Collaborative Functional Design Environment (CFDE) is built through the shared ontology and the domain ontology. The CFDE facilitates the semantic interoperability among diverse software resources, which provides more software resources and better service to users.

Key words: Semantic interoperability, shared ontology, domain ontology, functional modeling, CFDE

INTRODUCTION

Functional design (Usher *et al*., 1998) is a new perspective towards the research of this upstream design activity and its objective is to provide computer tools to link design functions with the structural (physical) embodiments used to realize the functions. The existing functional modeling frameworks have shown their effectiveness in various standalone functional design applications over the last three decades. But, upstream design information has semantic heterogeneity, it is difficult to enable semantic interoperability in different functional applications. Not only the geometric information about the design, but also the activities such as designing, manufacturing and marketing that affect product development right from the stage of conceptual design are all exchanged by designers. Thus, a product development activity requires the expertise and interaction of a broad range of disciplines. In order to utilize the various software resources, an intermediary which is capable of providing an integrated view of them is needed (Handschuh *et al*., 1997). An effort of significant relevance is the development of the Process Specification Language (PSL) at the National Institute of Standards and Technology (NIST). Process Specification Language (PSL) defines a neutral representation for interoperability of information relevant to manufacturing processes. It considers the representation of process data used throughout the life cycle of a product, from early indications of the manufacturing process flagged during design, through process planning, production scheduling and controlling. The ontology is being developed to facilitate exchange of information among various manufacturing process related software. Because ontologies are explicit and formal specifications of the knowledge, especially implicit or hidden knowledge (Cho *et al*., 2006).

In this study, we propose a framework of shared ontology to facilitate semantic interoperability among different software resources. The shared ontology is a standard functional modeling as intermediate by Web Ontology Language (OWL) (Antoniou *et al*., 2004). It effectively manages heterogeneous information and presents the exchange of semantics of product data among the domain ontologies. The shared ontology provides only high-level concepts and attributes based on meta-concepts and meta-properties. The implicit or hidden knowledge is explicated by specializing the basic concepts of the shared ontology into sub-concepts in each domain ontology. Those high-level concepts guide the domain ontology to have similar aggregation and granularity. The domain ontology is an ontology-based functional modeling, which uses OWL to build a domain-specific functional design ontology repository. The CFDE is implenmented based on the shared ontology and the domain ontology, which provides convenience for user to transparently use various software resources. Nowadays, about sixty percent of software resoures has been shared based on the framework of CFDE.

Corresponding Author: Bo Ding, Research Institute of Computer Applied Techniques, Harbin University of Science and Technology, Harbin 150080, China

Ontologies can realize integration of information and semantic interoperability. Various ontology-based information integration methods have been developed. They are single ontology approach, multiple ontologies approach and hybrid approach. In the single ontology approach each information source is related to the same global domain ontology. As a result, a new information source can not bring in any new or specific concept without requiring change in the global ontology. In the multiple ontologies approach, each information source has its own ontology developed irrespective of other information sources. In this case, the inter-ontology mapping is very difficult to define as the different ontologies may use different aggregations and granularities of the ontology concepts. In the hybrid approach, to overcome the drawback of single or multiple ontology approaches, each information source has its own ontology and all source ontologies are connected by some means to a common shared ontology. In this study, the hybrid approach is adopted.

However, ontologies themselves could be heterogeneous. The ontology heterogeneities are obstacles to integrating and sharing product information. The ontology heterogeneities can be subdivided into four categories: system heterogeneity, syntax heterogeneity, structure heterogeneity and semantic heterogeneity. The semantic heterogeneity is the most difficult to be shielded. There are four reasons to cause semantic heterogeneity.

- The different information sources represent identical concepts using different terms
- The identical concept expresses different meanings in different information sources
- The different information sources represent identical (similar) information using different semantic structure
- The concepts might be relevant in different information sources, because of distributed character of information sources, the implicit connections can not be embodied. Therefore, the need for rigorous and unambiguous description shields ontology heterogeneities

An approach toward the development of a product ontology and semantic mapping using first-order logic is presented by Kim *et al.* (2003). This effort proposes the development of a shared ontology. The shared ontology as a medium, which implements data exchange between different software resources. However, no details are provided on the development of the ontologies and how the system enables semantic interoperability.

An ontology-based framework is proposed to enable semantic interoperability by Patil *et al.* (2005). A standards-based approach is used to develop a Product Semantic Representation Language (PSRL). Formal description logic (DAML+OIL) is used to encode the PSRL. The preliminary exchange of product data between the PSRL and other application ontologies is achieved. But, the PSRL alone is not enough for identifying domain concepts consistently and structuring them systematically. The mappings are manually specified and the mappings between arbitrary expressions are hard to handle.

An ontology-based model based on the IEC62264 standard is proposed by Dassisti *et al.* (2008). The ontology-based model solves the information interoperability (I^2) problem by specifying the information related to products, manufacturing resources and processes. The required function is decomposed into a set of eight sub-function models. Each model concerns a particular view of the integration problem. However, no details are provided on the development of the eight ontologies.

Present method follows the hybrid approach, each information source has its own domain ontology and all domain ontologies are connected by some means to the shared ontology. The shared ontology is organized into layered ontologies. An ontology in a layer plays different roles. The shared ontology defines certain concepts and properties and has an intelligent semantic inference mechanism. It shields the heterogeneity of semantic information and supports a meaningful representation across various software resources.

A LAYERED FRAMEWORK OF SHARED ONTOLOGY

The shared ontology is a functional modeling as a standard intermediate. The goal of the shared ontology is to provide a uniform representation for heterogeneous information. The benefits of ontology representation lie in its capabilities in explicitly defining concepts, attributes and relationships. The explicit concepts and structure can contribute to machine-readable and human-perceived meaning of semantic information to distributed, synchronous software sources. Therefore, the shared ontology is the foundation to automatically operate and use software sources. In this study, the shared ontology which provides only high-level concepts and attributes, decreases the cost of knowledge collaboration. Meanwhile, the shared ontology which provides an intelligent semantic inference mechanism, effectively implements ontology inference and ontology study. The collaborative functional design is accurately and effectively implemented based on the shared ontology across different software resources. Figure 1 shows the hierarchical organization of ontologies.

The basic process of constructing the shared ontology can be described as follows:

Step 1: The design of concept model. The concept model governs all ontologies with the most fundamental meta-concepts. It gives fundamental distinctions of conceptions and a proper taxonomy.

Step 2: The design of semantic feature library. The semantic feature library which divides the selected concepts into a hierarchical organization, informs the appropriate parts properties of the product modeling according to the meta-properties.

Step 3: The design of semantic inference model. The semantic inference model provides an intelligent semantic inference mechanism, which can quickly acquire semantic information and implement data exchange in the upper layer between the shared ontology and the domain ontology.

The concept model provides high-level concepts and gives fundamental distinctions of concepts in the real world. The concept model should include several feature vectors such as products, processes and resources. Table 1 shows the categories of feature vectors. The values of technology features, which are irrelevant to

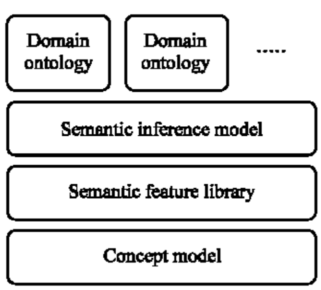

Fig. 1: Hierarchical organization of ontologies

environment and state, are independent relatively. The values of service features are closely relevant to environment and state. For product and service, the demand of user no longer only reflecting technology features of product performance itself, service features are also important aspects of competitions, such as delivery time and maintenance, etc. These vectors are mutual influence and restriction.

The representation method of concept model is:

$$A_i = (V_i, D_i, C_i)$$

Definition domain $V_i = (v_{i1}, v_{i2}, \ldots, v_{iN_i})$, $0<N_i<\infty$, is open interval. The number of feature variables tends to become infinite with the increasing of user demands, the continuous improvements of the technology and the increasing complexity of the products. It is impossible that every enterprise has all feature variables in the definition domain. They possess only part of product process and resource variables based on their own core capability and compose distributed business units-domain ontologies.

Value domain $D_i = (d_{i1}, d_{i2}, \ldots, d_{iN_i})$. The variety of user demands and service features requires the shared ontology with good adaptive capability. The feature values of state variables should consider redundancy, such as building replacement parts, replacement resources and replacement processes, etc. when the transformation of product state occurs, The state variables can adapt the changes rapidly utilizing redundancy features.

Constraint $C_i = (c_{i1}, c_{i2}, \ldots, c_{ik})$. All properties of features, including their geometric parameters and validity conditions, are declared by means of constraints. The specification of validity condition can be classified into three categories: geometric, topologic and functional. The different validity conditions can be saved by different constraint types. For instance, to specify the geometric condition, the width and length of a square section passage feature can be simple equalities between two parameters, for this, we use algebraic constraints.

The product semantic feature which is the parameter of effecting technical performance and service performance, includes product, all relevant process and

Table 1: The catagories of feature vectors

	Feature vectors		
Catagories	Product	Process	Resource
Technology feature	Competitiveness		Resource capability
	Quality cost time		Personnel capability
Self-correlation of relation feature	Product product matrix	Process process matrix	Resource resource matrix
Cross-correlation	Product process matrix	Process product matrix	Resource product matrix
	Product resource matrix	Process resource matrix	Resource process matrix

resource related to product. The product semantic feature which combines with distributed, irrelevant knowledge, which is easy to manage the knowledge and information of product life cycle. The semantic feature library is built based on meta-properties and meta-concepts. The parts classes of product modeling are given according to the meta-concepts. The appropriate parts properties of product modeling are given according to the meta-properties. Form view of essence the semantic feature library gives the definition domain and the value domain of the shared ontology based on the domain knowledge and reflects definite knowledge accumulation. The standards-based definition domain and value domain are easy to share and reuse of the product knowledge and decrease the cost of knowledge collaboration. The semantic feature library for the shared ontology plays the role of shared dictionary. Figure 2 shows part of semantic feature primitives and their structure.

The semantic feature library applies OWL to describing semantic features in the paper. The OWL is ontology description language, which is recommended by semantic internet. It is used to clearly express the meaning of terms and the relationship among them as a vocabulary. And it defines the knowledge of feature field, such as concept, axiom, judgment, rule, etc. The OWL supports fine grained product semantic information description and information sharing of function layer.

The class product is the parent of all classes. The basic structured definition of Product described by OWL schema is presented as below:

```
<owl:Class rdf:ID = “Product”
 <rdfs:subClassOf>
   <owl:Restriction><owl:someValuesForm rdf:resource =“#Part”/>
    <owl:onProperty><owl:ObjectProperty rdf:ID =“hasPart”/>
    </owl:onProperty></owl:Restriction>
  <owl:Restriction><owl:onProperty rdf:resource =“#hasPart”/>
    <owl:someValuesForm><owl:Class rdf:ID =“Assembly”/>
    </owl:someValuesForm></owl:Restriction>
 <rdfs:subClassOf>
 <owl:Restriction>
   <owl:onProperty rdf:resource =“ObjectID”>
   <owl:cardinality rdf:datatype =“&xsd:NonNegaveIntger”>1
   </owl:cardinality>
 </owl:Restriction>
 <owl:Restriction>
  </owl:onProperty rdf: resource =“Name”>
  </owl:cardinality rdf:datatype =“&xsd:NonNegaveInteger>1
  </owl:cardinality>
 </owl:Restriction>
......
</ow1:Class>
```

The semantic inference model can quickly acquire semantic information by semantic mapping and semantic inference mechanism, which implements data exchange in the upper layer between the shared ontology and the domain ontology. Semantic inference adopts a mixing mapping method based on Open World Assumption (OWA) and Closed World Assumption (CWA) (Reiter, 1987). The method is not rigidly controlled by the OWA

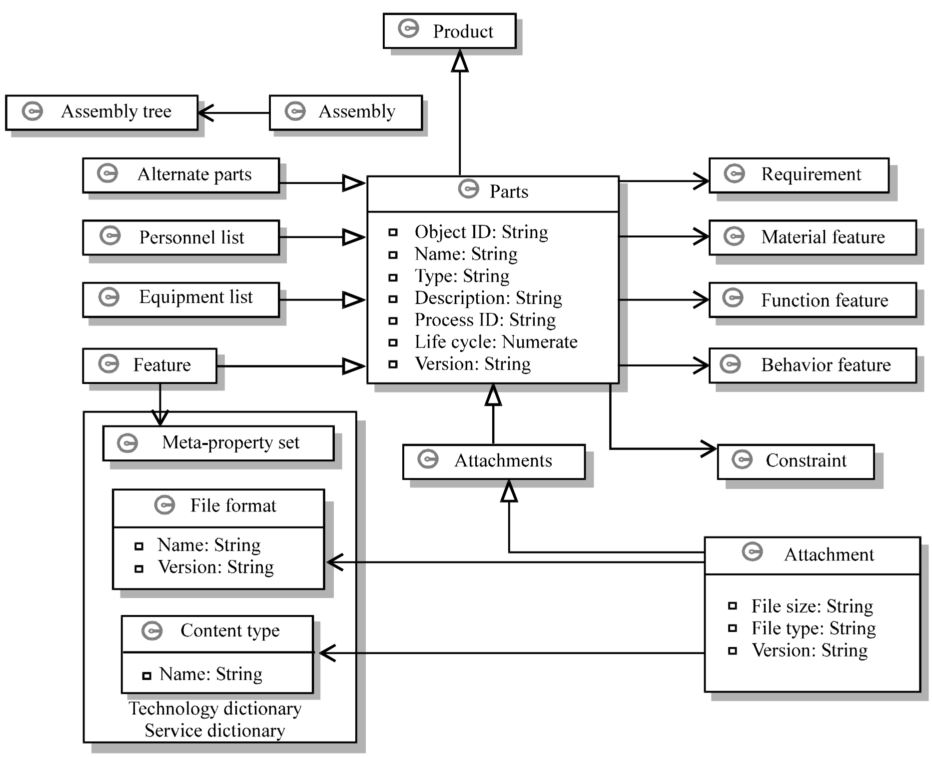

Fig. 2: Semantic feature primitives and their structure

or by the CWA. In this method, we permit nulls, more generally, attributes in targets-to be open or closed. Open attributes can be instantiated by many values, but for closed, only one value is permitted.

Let σ and τ be two relational database schemas; σ is thought of as a source schema and τ as a target schema. A mapping M between schemas σ and τ is a condition that states how instances of σ and τ are related. In data exchange, mappings are typically specified by sets of Source-to-Target Dependencies (STDs) of the form:

$$\psi_\tau(\bar{x},\bar{z}) :- \varphi_\sigma(\bar{x},\bar{y}) \tag{1}$$

where, φ_σ is a First-Order (FO) formula over vocabulary σ and Ψ_τ is a conjunction of atomic τ-formulae. A mapping for us is thus a triple (σ, τ, Σ), where Σ is a set of STDs. If S is a source instance, then a target τ-instance T is called a solution for S. That is to say S and T is equivalent on semantic layer. If there is an operation P, S and T is input, the result P(S) and P(T) is similar to a certain extent.

Target instances can be populated by two different kinds of elements: constants and nulls. Constants are elements that come from the source and nulls are new elements created in targets.

We shall allow each variable in the left-hand side Ψ of an STD to be annotated with an element of the set (op, cl), referring to them as open or closed variables, respectively.

To determine equivalent relationship of source ontology and target ontology, concept must have definite equivalence definition between two ontologies. To every concept of domain ontology, semantic inference model should have a definite equivalence definition. Supposes any individual S of σ in source ontology, any individual T of τ in target ontology, T has two binary relations and the annotated STDs is:

$$T(x^{cl}, y^{op}) :- S(x) \tag{2}$$

Closed annotations specify one-to-one relationships, so closed nulls behave just as nulls in CWA-solutions. Open annotations specify one-to-many relationships and exhibit the behavior of solutions of Fagin *et al.* (2005).

A database instance with incomplete information is an instance whose domain is a subset of Const∪Null. Nulls are treated as existing but unknown values. A valuation is a partial map ν: Null→Const. Given an instance v with incomplete information and a valuation ν defined on all of its nulls, ν(T) stands for the instance over Const in which every null ⊥ in T is replaced by ν(⊥). The semantics of T, denoted by Rep(T), consists of all such instances:

$$Rep(T) = \nu(T) \tag{3}$$

where, ν is a valuation.

Semantic inference generates a series of mapping results, so we should find certain answers. Query answering in data exchange normally means finding certain answers. By analyzing background and constraint knowledge to design query answering. Since, the notion of Q(T), where T is a solution, is not well-defined due to T containing nulls, we must find certain answers to Q over each solution T and then find tuples that belong to such certain answers over all solutions T. That is, given an annotated mapping with STDs Σ_α, a source instance S and positive relational algebra query Q, we define:

$$Certain_{\Sigma_\alpha}(Q,S) = \bigcap_{T \text{ is a } \Sigma_a\text{-solution}} \; \bigcap_{R \in Rep(T)} Q(R) \tag{4}$$

If there are semantic equivalence relationships between a source ontology and the shared ontology, a Semantic Equivalence Matrix (SEM_{sc}) is determined. The matrix is an ordered concept list. SEM_{sc} between source ontology and shared ontology is:

$$SEM_{sc}\{(\sigma,\theta) : \sigma \in s, \theta \in c, \sigma^c \equiv \theta\} \tag{5}$$

If there are semantic equivalence relationships between a target ontology and the shared ontology, the SEM_{tc} is:

$$SEM_{tc}\{(\tau,\theta) : \tau \in t, \theta \in c, \tau^c \equiv \theta\} \tag{6}$$

That is to say, any individual of σ is effective individual of θ and any individual of τ is effective individual of θ. Therefore, Sα⇔θ⇔Tτ. σ and τ has the mutual semantic equivalence relationship. Any individual of σ is effective individual of τ.

The shared ontology supports explicit and formal specifications of the knowledge, especially implicit or hidden knowledge. The upper layer exchange of product data is implemented by the shared ontology across different software resources.

Shared ontology = Concept model + semantic feature library + semantic inference model

The shared ontology is extensible with the application software enhancing their capabilities over a period of time. The meta-concepts and meta-properties can be continually increased, support the sharing among different software resources.

A SEMANTIC WEB-BASED ARCHITECTURE FOR COLLABORATIVE FUNCTIONAL DESIGN

Here, a semantic web-based architecture for CFDE is proposed, which combines ontology-based functional modeling to support distributed, synchronous collaborative functional design. The architecture is composed of two visiting levels: global level and local level. Figure 3 shows the proposed architecture.

The global level provides a collaborative functional design environment for various domain ontology. The different functional models are linked together on the global level, each model concerns a particular view of the integration problem. The global level serves as a middleware to support collaborative management and ontology communication. The global level mainly includes a shared ontology and four specific functional models. The shared ontology is a standardized functional modeling as intermediate, which implements the uniform representation of product model data. The heterogeneity is shielded by shared ontology. The four functional models are: product information model, product schedule model, product capability model and ontology index model. The product information model presents what was made? and what was used? It lists the specific personnel, equipment and material needed. The product schedule model corresponds to the product information, which splits product information into simpler function requirements by Functional Supportive Synthesis (FSS) (Zhang *et al*., 2004) or Functional Decomposition (FD) (Umeda *et al*., 1996). A product schedule shall be made up of one or more production requests. A request for production for a single product identified by a production rule shall be shown as a production request. Besides, product schedule model answers the questions: when is the function requirement executed? and when is it due? The product capability model answers the question what is available? It defines what the collaborative functional design system is capable of doing. The product capability model is the collection of information about all resources for production. This is made up of information about equipment, material, personnel and process segments. It provides detail required for planning and scheduling, takes into account maintenance activities and available capacity. The ontology index model provides an ontology registration and a quick location mechanism to find the required domain ontology.

The local level is the basis for collaborative functional modeling. This local framework is mainly composed of two layers: domain ontology layer and ontology visit layer. The domain ontology layer includes various software resources. The traditional standalone functional models are extended and are designed to be domain ontologies having a specific function. The domain ontology uses OWL as a content language, which facilitates information sharing and retrieval. The semantic communication is implemented upon different domain ontologies. The top layer is the ontology visit layer, supported by a format exchanging unit. The semantic information with OWL format can be encapsulated to facilitate communication and sharing between the different ontologies.

Collaborative Functional Design Environment (CFDE) supports real-time exchange of product model data

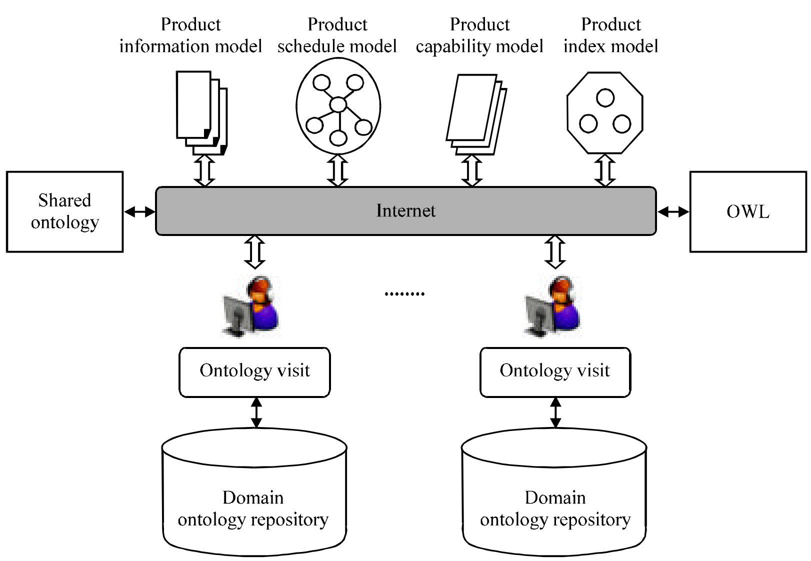

Fig. 3: A semantic web-based architecture for CFDE

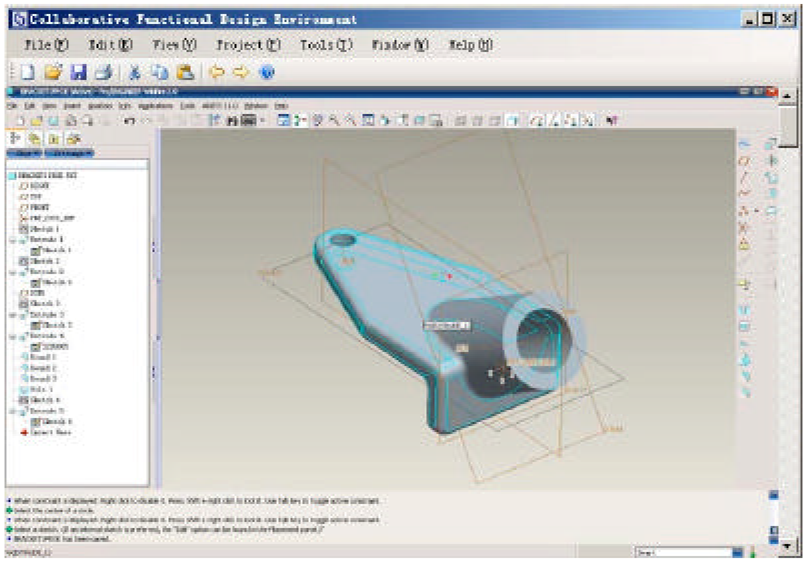

Fig. 4: Pro/E Wildfire 2.0 view

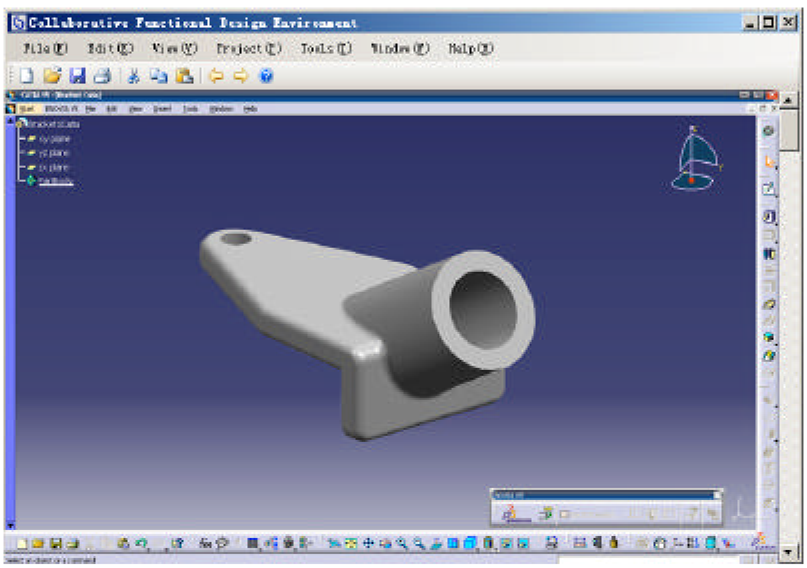

Fig. 5: CATIA P3 V5R17 view

between different software resources and which facilitates the interaction between users and diverse software resources. The CFDE can choose the best software resources according to personalized user demands and show to terminal users in a proper way. This study adopts virtual desktop as a present pattern of resources. The virtual desktop transparently uses distributed software resources in a succinct, consistent way. Pro/E Wildfire 2.0 is being accessed as shown in Fig. 4. CATIA P3 V5R17 is being accessed as shown in Fig. 5.

CONCLUSION

An increasing trend toward product development in CFDE has resulted in the use of various software tools to enhance the product design. In order to manage heterogeneous information, it is appropriate to develop standard-based models able to trace all information related to the product life cycle. The shared ontology and the domain ontology are designed in this paper. The shared ontology is a standardized functional modeling as intermediate. It provides foundations for systematization of domain knowledge and its usefulness in automated information integration. The domain ontology is a domain-specific functional design ontology repository. Based on the shared ontology and the domain ontology, about sixty percent of software resoures has been shared. Both of them implement semantic interoperability of product information and provide convenience for user to transparently use various software resources.

ACKNOWLEDGMENTS

The authors are very grateful to the National Natural Science Foundation of China (No. 60173055) and the helping of Xue-Yao Gao and Xian-Guo Liu.

REFERENCES

Antoniou, G. and F.V. Harmelen, 2004. Web Ontology Language: OWL. In: Handbook on Ontologies, Staab, S. and R. Studer (Eds.). Birkhäuser, Berline, ISBN: 3540408347, pp: 67-92.

Cho, J., S. Han and H. Kim, 2006. Meta-ontology for automated information integration of parts libraries. Comput. Aided Des., 38: 713-725.

Dassisti, M., H. Panetto, A. Tursi and M. De Nicolo, 2008. Ontology-based model for production-control systems interoperability. Proceedings of the 5th International Conference on Digital Enterprise Technology, Oct. 22-24, Nantes, France, pp: 1-11.

Fagin, R., P.G. Kolaitis, R.J. Miller and L. Popa, 2005. Data Exchange: Semantics and Query Answering. In: Database Theory-ICDT 2003, Calvanese, D. *et al*. (Eds.). LNCS 2572, Springer Verlag, Berlin, Heidelberg, ISBN: 978-3-540-00323-6, pp: 207-224.

Handschuh, S., B.F. Schmid and K. Stanoevska-Slabeva, 1997. The concept of a mediating electronic product catalog. EM-Electronic Markets, 7: 32-35.

Kim, K.Y., S.H. Chae and H.W. Suh, 2003. An approach to semantic mapping using product ontology for CPC environment. Proceedings of the 10th ISPE International Conference on Concurrent Engineering Research and Applications, July 26-30, A.A. Balkema Publishers, Madeira, Portugal, pp: 291-298.

Patil, L., D. Dutta and R. Sriram, 2005. Ontology-based exchange of product data semantics. IEEE Trans. Automation Sci. Eng., 2: 213-225.

Reiter, R., 1987. On Closed World Databases. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA., ISBN-10: 0934613451, pp: 300-310.

Umeda, Y., M. Ishii, M. Yoshioka, Y. Shimomuray and T. Tomiyama, 1996. Supporting conceptual design based on the function-behavior-state modeler. Artificial Intelli. Engin. Design Analysis Manufa., 10: 275-288.

Usher, J.M., R. Utpal and P. Hamid, 1998. Integrated Product and Process Development: Methods, Tools and Technologies. 1st Edn., John Wiley and Sons, New York, ISBN-10: 0471155977.

Zhang, W.Y., S.B. Tor and G.A. Britton, 2004. A graph and matrix representation scheme for functional design of mechanical products. Int. J. Adv. Manufa. Technol., 25: 221-232.