SOTECH Vol. 9, No. 3, pp. 33 ~ 4, 2005

Enhancement of CAD model interoperability based on feature ontology

Lee, Y.S.[1], Cheon, S.U.[2] and Han, S.H.[2]
[1] Samsung Electronics,
[2] KAIST, Dept. of Mechanical Eng.

Abstract

As the networks connect the world, enterprises tend to move manufacturing activities into virtual spaces. Since different software applications use different data terminology, it becomes a problem to interoperate, interchange, and manage electronic data among heterogeneous systems. According to RTI, approximately one billion dollar has been being spent yearly for product data exchange and interoperability. As commercial CAD systems have brought in the concept of design feature for the sake of interoperability, terminologies of design features need to be harmonized. In order to define design feature terminology for integration, knowledge about feature definitions of different CAD systems should be considered. STEP standard have attempted to solve this problem, but it defines only syntactic data representation so that semantic data integration is not possible. This paper proposes a methodology for integrating modeling features of CAD systems. We utilize the ontology concept to build a data model of design features which can be a semantic standard of feature definitions of CAD systems. Using feature ontology, we implement an integrated virtual database and a simple system which searches and edits design features in a semantic way.
Keywords: data integration, feature ontology, semantic search

1 Introduction

Various CAx systems are utilized throughout the product life cycle from product design to delivery. In order to manage design, production and material handling information, systems like PDM (product data management) and ERP (enterprise resource planning) have been being used. One of the major problems facing enterprises today is how to share and exchange data among heterogeneous applications. It is said that approximately one billion dollar has been being spent yearly in USA for product data exchange and interoperability (Gallaher et al 2002). STEP (Standard for the Exchange of Product model data) have attempted to solve this problem, but it cannot define semantic data which include parameters, features and constraints. There are several on-going projects to establish a standard by which feature data can be exchanged. However they concentrate only on manufacturing processes so that they cannot integrate and manage product design data among heterogeneous applications.
Researches on features in CAx system can be categorized as; i) modeling by feature, ii) feature recognition from the B-Rep model, iii) feature data exchange among heterogeneous CAD systems. In case of (iii), Macro-Parametric (Choi et al 2002) and Feature Resource projects have been in progress within the Parametrics group of ISO TC184/SC4.

This paper is also related to (iii) category, but the differences are; i) Macro-Parametric and Feature Resource both focus on data exchange but this paper focuses on real-time interoperability; ii) it provides a method of integrating various CAD data in a semantic way.
To integrate data in a semantic way, an ontology method is applied. An ontology is an explicit specification of a conceptualization (Gruber 1992). Taxonomy consists of vocabulary and concept structure which are used in a specific domain. Ontology adds relation, rule and constraint to the taxonomy so that semantics can be represented in a data model. The Macro-Parametric method first defines standard features and the design history and then exchanges data using a neutral file format according to the standard. Although the existing method allows mappings between different terminologies, which mean the same but syntactically are different, the mappings can be done only grammatically, not semantically. Thus, we construct a feature ontology to transfer the semantic information so that we can manage heterogeneous data from different CAD systems.
This paper explains: i) the way to construct feature ontology, ii) the method of sharing feature information, iii) a pilot program that verifies interoperability between commercial CAD systems, CATIA and SolidWorks.

2 Related works

- Ontology

The importance of capturing and representing real world knowledge in information systems has long been recognized in artificial intelligence, software reuse, and database management. Ontologies have been proposed as an important and natural means of representing real world knowledge for the development of database designs (Vijayan and Veda 2002). The word ontology is defined by the AI community as "Ontology is a specification of a conceptualization " (Gruber 1992). Ontology consists of concept, relation, concept hierarchy, function relation and axiom. There are languages for expressing ontology such as RDF, DAML+OIL, first order logic and F-logic.

- Feature taxonomy

In order to exchange feature data between CAD and CAM systems, it may be a good approach to categorize features into families that are relatively independent of the intended application domain. Several feature taxonomy schemes have been proposed such as CAM-I project (Butterfield et al 1986), rotational parts taxonomy (Kim et al 1991), and Part 48(ISO 1992) of STEP.

- TOVE project

The goal of TOVE project is to develop a set of ontologies for the modeling of both commercial and public enterprises (Belli and Radermacher 1992). TOVE project provides a shared terminology for the enterprise that each agent can jointly understand and use. It also defines the meaning of each term in a precise and as unambiguous manner as possible so that it can implement semantics which enable to answer all the questions about the enterprise.

- A feature ontology to support construction cost estimating

Many architectural 3D modeling applications can export semantically rich product models using the industry standard Industry Foundation Classes (IAI 2001), which enables the sharing of product models with other software applications. However it is the standard not for a cost estimating but for a designer. Therefore Staub-French et al formalized an ontology using features to represent the different design conditions that affect construction costs (Staub-French et al 2002).

- Ontologies for integrating engineering applications

It has been a big problem to interoperate various software applications that a manufacturing process uses. This research uses ontology to express semantic information among different applications which should be integrated (Ciocoiu et al 2000). It provides an example of using a common ontology as an Interlingua for facilitating manufacturing process information exchange between ProCap and ILOG. PSL (process specific language) is used as a mediator ontology (Gruninger 2000).

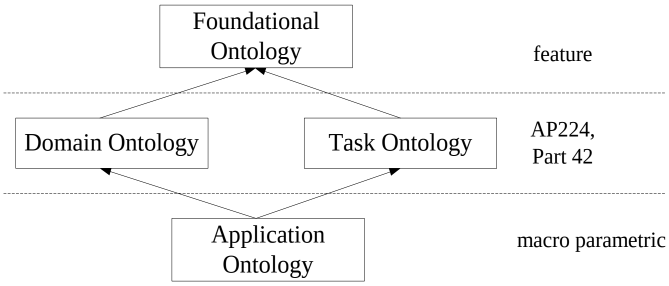

Figure 1: Layered ontology

3 Ontology application scenarios

In this section, a short overview on aspects of developing the feature ontology is given.

3.1 Layered ontology

In order to construct ontology which contains information and knowledge of applications so that it is reused and shared by human and computer at the same time, it is better to have layers such as application ontology, domain ontology and foundational ontology as shown in Figure 1.

If only one ontology engineer constructs the ontology, domain knowledge can be insufficient. Hence a domain expert should model, manage and repair the ontology. Before building a layered ontology, classification of ontology should be done. First of all we can build the foundational ontology which defines the basic concept, and using the foundational ontology, the domain ontology which represents the domain knowledge can be built. At last using the domain ontology, the application ontology can be designed for a specific application.

This paper defines the basic feature concept as the foundational ontology, and for the domain ontology, design features of ISO 10303 AP224 and Part 42 is referenced. An application ontology is built using design features which are based on a set of standard macro commands (Mun et al 2003) from the Macro-Parametic method. The Macro-Parametric method (Choi et al 2002) is intended to provide capabilities to transfer parametric information by means of exchanging macro files which contains designer's intend and design history.

3.2 Method to build ontology

The ideal situation when applications are being developed in a domain is that the standard ontology of terminologies should be built in advance. Inheriting the standard ontology, several application ontologies can be constructed respectively as shown in Figure 2. Even if several systems are developed, they can easily interoperate because they use the same terminology.

However in reality, existing systems already defined and used their own set of terminologies, so different applications cannot interoperate. In this case, as shown in Figure 2, we should make

the shared ontology based on existing application ontologies. After this, we can bridge between different application ontologies.
Bridging means that if defined axioms from various systems are different, we can define the axiom between them again, so that they can be reasoned to have the same meaning even though they are using different terminologies.

3.3 Reasoning
In order to share data between different systems, they should be reasoned. If we want to interoperate the system A with the system B, the shared ontology and the application ontology of system A should be connected, and the shared ontology and the application ontology of system B should be bridged. Then application A and application B can interoperate based on reasoning, even if they are not directly connected. Bridging is possible by describing axioms. Although we define axioms, if the shared ontology is not rigid and powerful, bridging between different ontologies is a difficult task.

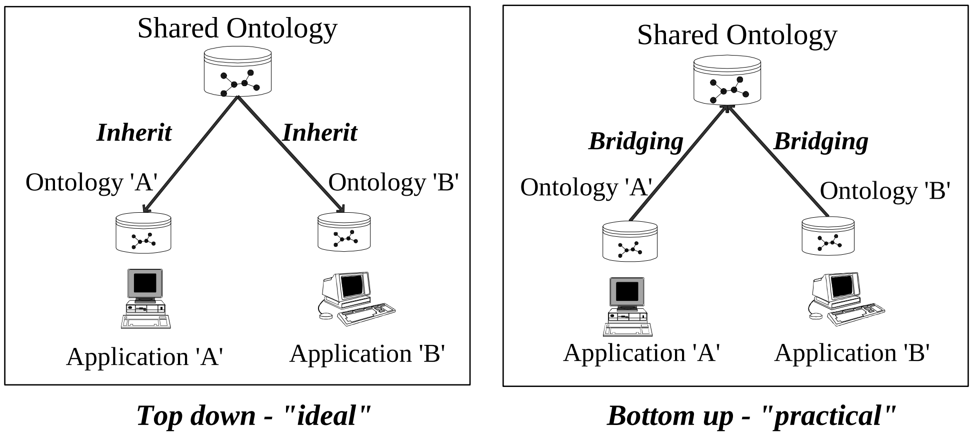

Figure 2: Different approaches of building ontology

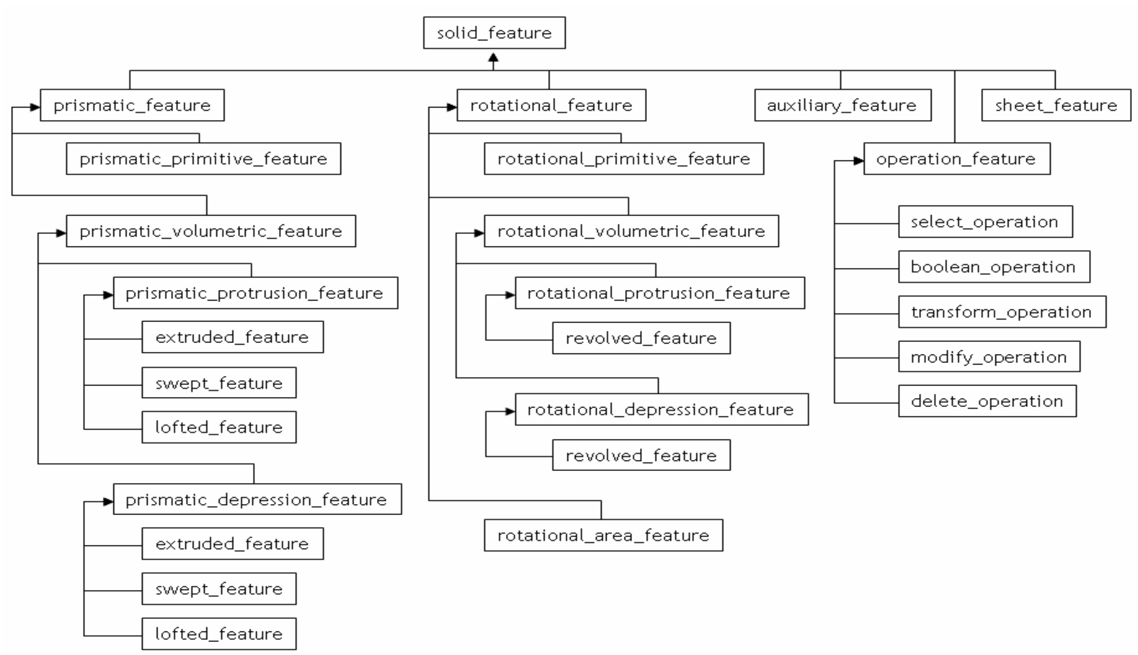

Figure 3: Feature taxonomy

4 Feature ontology

4.1 Feature taxonomy
The feature taxonomy has been constructed through an analysis of design features selected from features defined in AP224, the Feature Resource and Macro-Parametrics. The proposed feature taxonomy is shown in Figure 3. As the result of building taxonomy, definitions are inherited through hierarchy, so it is easy to expand and reuse them in different area. Based on the taxonomy, concept, relation and range are defined as the other elements of ontology. An example is shown in Figure 4.

4.2 Axiom
Axiom is a core element that enables semantic query. It is declarative and reasonably recognized without proof, so in logic world it is considered as a premise of reasoning. Human can understand even if there is not a specific description, but machines cannot understand without a description. Axiom can provide knowledge into a data model so that it allows machines to understand the meaning. Figure 5 shows the ontology file with axiom definitions, which is written in F-logic format.

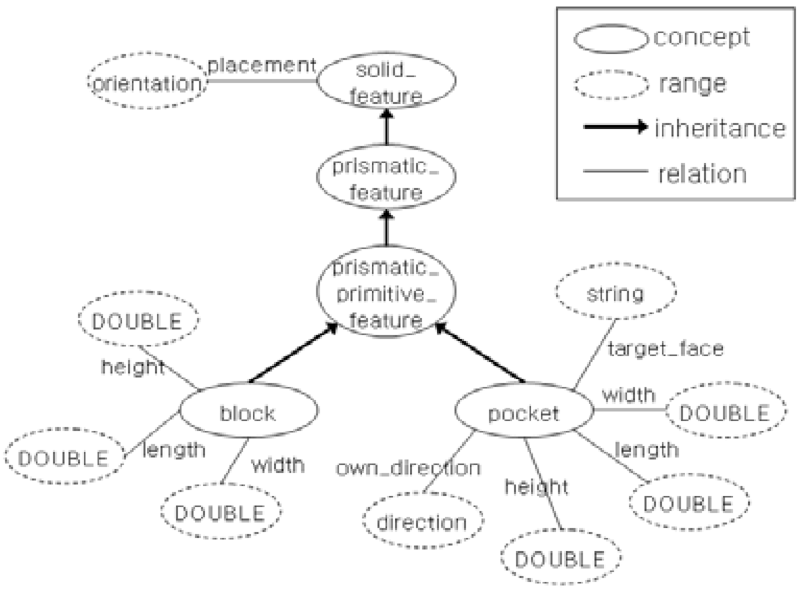

Figure 4: An example of concept, relation

```
// CONCEPTS ------------------
solid_feature::DEFAULT_ROOT_CONCEPT.
rotational_feature::solid_feature.
rotational_primitive_feature::rotational_feature.
hole::rotational_primitive_feature.
round_hole::hole.
// LOCAL RELATIONS ------------------
round_hole[cs⇒coord_sys;
            center⇒position;
            depth⇒DOUBLE;
            bottom_angle⇒DOUBLE;
            head_angle⇒DOUBLE;
            bottom_type⇒INTEGER].
// AXIOMS ------------------
rule _1: FORALL X,Y,W (X[W=>Y] AND
    Y[x=>DOUBLE;y=>DOUBLE;z=>DOUBLE] ) ↔
   (X[x=>DOUBLE;y=>DOUBLE;z=>DOUBLE]).
rule _2: FORALL X,Y  (X[cs⇒Y] AND
   Y[origin⇒position;dx⇒direction;dy⇒direction;dz⇒direction] ) ↔
   (X[origin⇒position;dx⇒direction;dy⇒direction;dz⇒direction]).
```

Figure 5: An example of ontology file (in F-logic file format)

4.3 Reasoning by axiom

As shown in Figure 6, there is syntactical and semantic heterogeneity among different ontology. Syntactic heterogeneity is settled by simple mapping between different terms. This is called mapping axioms. However, there is structural and semantic heterogeneity which cannot be connected by syntactic mapping. Hence we need to bridge them by defining relation axioms or symmetric axioms. An example of syntactic heterogeneity is between 'has_depth' and 'hasDepth'. In this case we can define mapping axiom as;

FORALL X,Y X [has_depth⇒Y] ↔X [hasDepth⇒Y].

In case of semantic heterogeneity, B ontology expresses the hole center concept directly as 'selectedbyX, selectedbyY, selectedbyY'. On the other side, the shared ontology defines 'has_center_point' and it also has the point concept. Therefore, if we query 'center point', only the shared ontology can give us the right answer. That is why we have to define axiom between them as

FORALL X,Y (X:(Hole_linear[has_center_point⇒Y]))
↔(X:(Hole_wizard[selectedbyX⇒DOUBLE;selectedbyX⇒DOUBLE;selectedbyX⇒DOUBLE])
).

We know how 'radius' and 'diameter' is different. But we have to give this semantics to the computer as;

FORALL CIRCLE,RADIUS,DIAMETER (CIRCLE[has_diameter–>>DIAMETER])
← ((CIRCLE[has_radius–>>RADIUS] and (DIAMETER,*(2.0,RADIUS)))).

4.3 Query example

After defining axiom between the shared ontology and the A ontology, between the shared ontology and the B ontology, Data A and data B can be integrated without any mapping between A ontology and B ontology. Figure 7 shows that A ontology and B ontology are different even though they mean the same. Because of the defined axiom which is explained in 4.2, B ontology data can be queried although it queries using the A ontology terminology.

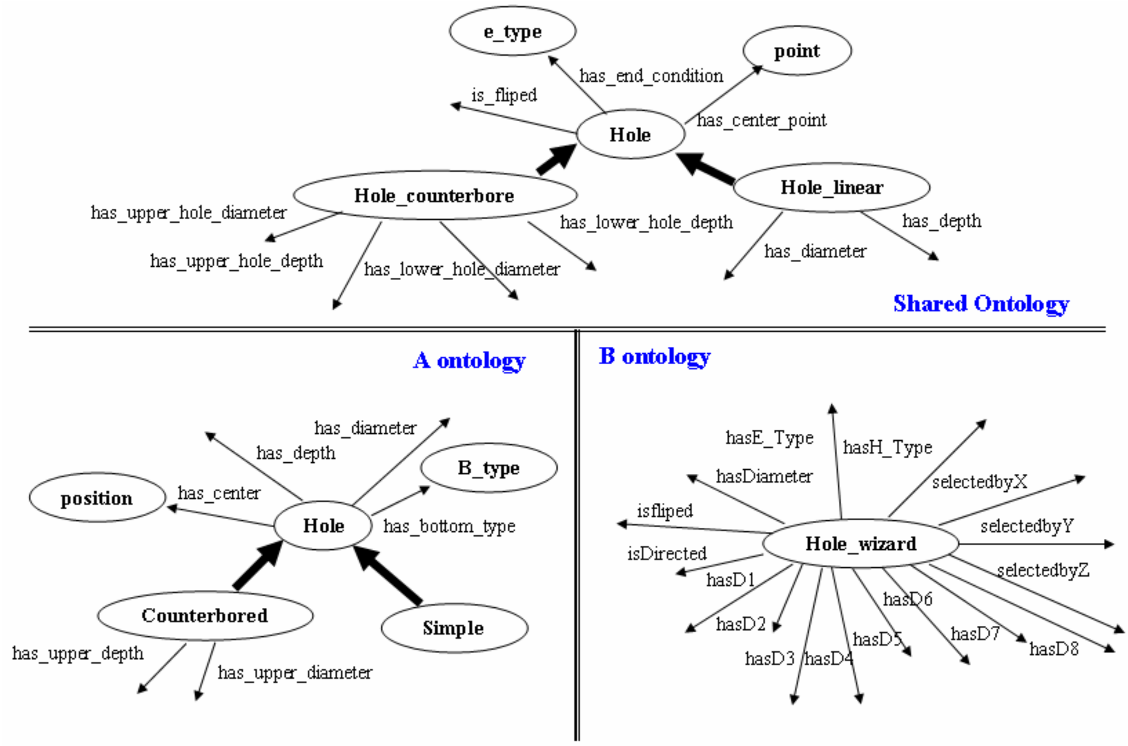

Figure 6: Heterogeneity among different ontology

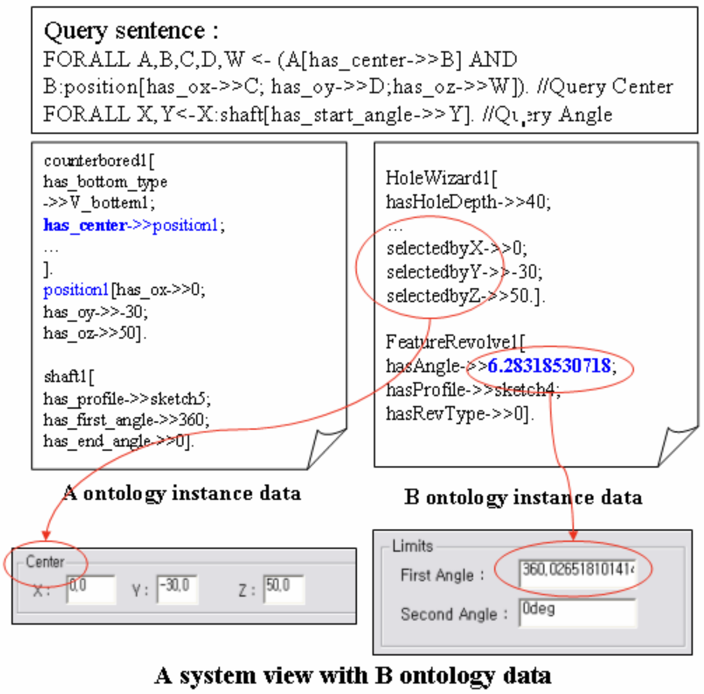

Figure 7: Semantic query from different ontology data

5 Pilot implementation

5.1 Implementation

This section provides an example of integrating different CAD systems; CATIA and SolidWorks. In this paper, we have used commercial programs OntoEdit and OntoBroker as ontology tool and query engine. Figure 8 explains the procedure based on the IDEF0 diagram.

5.2 OntoSmart

The implemented system is named as OntoSmart which is composed of OntoSmart_Translator (Macro file to Ontology file), OntoSmart_Query/Edit, OntoSmart_Translator (Ontology file to Macro file). The OntoSmart_Translator (Macro file to Ontology file) translates the macro file of a commercial CAD system into ontology data based on each system's feature ontology. The OntoSmart_Query/Edit module shows query result of a different system. It edits each ontology data in one view which is made only for one system terminology so that it verifies that two systems can interoperate. The OntoSmart_Translator (Ontology file to Macro file) translates ontology data into the macro file of another commercial CAD system so that they can be executed. Figure 9 shows a result of editing the Y shaped model. OntoSmart only uses the CATIA terminology but as shown at the left below of Fig. 9, the Solidworks file is also edited semantically.

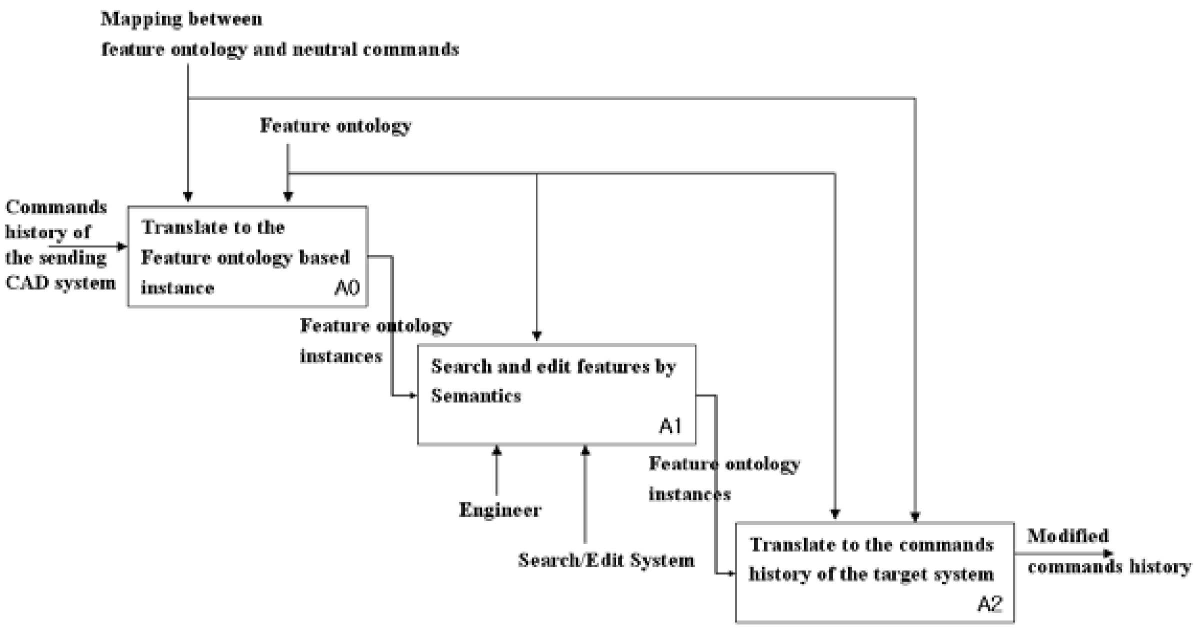

Figure 8: IDEF0 diagram of the implemented system

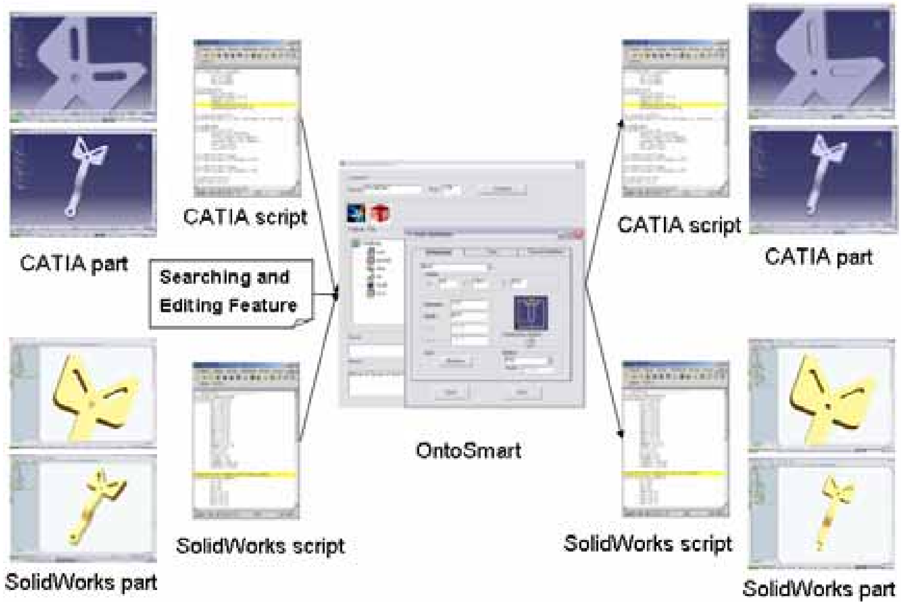

Figure 9: Pilot implementation (OntoSmart)

6 Conclusion

To solve the interoperability problem of sharing and exchanging heterogeneous data, we have constructed a feature ontology based on the ontology methodology. The data model containing knowledge is built using ontology, and features data among different CAD systems can be shared by the ontology which includes semantic information. We have showed the CAD system interoperability through a pilot implementation. That integrates CATIA and SolidWorks in an integrated environment and enables the semantic search and edit between two different data sets.
As a future work, the ontology of a commercial CAD system should be rigidly constructed. After that, the knowledge for interoperability should be added to share ontology so that the shared ontology should be rigid and powerful. With the shared ontology, a data exchange tool or other CAD integration tools can be built.

References

Belli, F. and Radermacher, F.J. 1992 The TOVE Project: A Common-sense Model of the Enterprise, Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Lecture Notes in Artificial Intelligence # 604, Berlin: Springer-Verlag, pp. 25-34
Butterfield, W. R., Green, M. K., Scott, D. C., and Stoker, W. J. 1986 Part features for process planning, CAM-I Report R-85-PPP-03.
Choi, G. H., Mun, D. H, and Han, S. H. 2002 Exchange of CAD part models based on the macro-parametric approach, International Journal of CAD/CAM, 2, 2, pp. 23-31.
Ciocoiu, M., Gruninger, M., and Nau, D. S. 2000 Ontologies for integrating engineering applications, Journal of Computing and Information Science in Engineering, 1, 1, pp. 12-22.
ISO TC184/SC4 1992 ISO10303-48 WD, Industrial Automation Systems and Integration - Product Data Representation and Exchange - Part 48 : Integrated Generic Resources : Form Features, second edition
Gallaher, M. P, O'Connor, A. C., and Phelps, T. 2002 Economic impact assessment of the international standard for the exchange of product model data (STEP) in transportation equipment industry, NIST

Gruber, T. 1992 A mechanism to support portable ontologies. Technical Report KSL91-66, Stanford Knowledge Systems Laboratory.

Gruninger, M. 2000 The process specification language (PSL): Overview and version 1.0 specification, NISTIR 6459, National Institute of Standards and Technology.

International Alliance of Interoperability. 2001 IFC 2x extension modeling guide, http://www.iai.org.uk

Kim, J. Y., O'Grady, P., and Young, R. E. 1991 Feature taxonomies for rotational parts: A review and proposed taxonomies, International Journal of Computer-Integrated Manufacturing, 4, 6, pp. 341-350.

Mun, D. H., Han, S., Kim, J. H., and Oh, Y. C. 2003 A set of standard modeling commands for the history based parametric approach, Computer Aided Design, 35, 13, pp. 1175-1179.

Staub-French, S., Fischer, M., Kunz, J., Paulson, B., and Ishii, K. 2002 A feature ontology to support construction cost estimating, CIFE Working Paper #69.

Vijayan, S. and Veda, C. S. 2002 Ontologies for conceptual modeling: their creation, use, and management, Data & Knowledge Engineering, 42, pp. 251-271.