Sharing CAD models based on feature ontology of commands history

Tae-Sul Seo (tsseo@kisti.re.kr)
Technical Support for Standardization Dept., Korea Institute of Science and Technology Information, Seoul, Korea

Yoonsook Lee (yoonsook.lee@samsung.com)
Design Innovation Group, Digital Appliance Division, Samsung Electronics, Suwon, Korea

Sang-Uk Cheon (csu@icad.kaist.ac.kr), Soonhung Han* (shhan@kaist.ac.kr)
Dept. of Mechanical Engineering, Korea Advanced Institute of Science and Technology, Daejeon, Korea

Lalit Patil (lpatil@umich.edu), Debasish Dutta (dutta@umich.edu)
Dept. of Mechanical Engineering, University of Michigan, Ann Arbor, MI, USA

Abstract

Different CAx systems are being utilized throughout the product lifecycle due to the practical reasons in the supply chain and design processes. One of the major problems facing enterprises of today is how to share and exchange data among heterogeneous applications. Since different software applications use different terminologies, it is difficult to share and exchange the product data with internal and external partners. This paper presents a method to enhance the CAD model interoperability based on feature ontology. The feature ontology has been constructed based on the feature definition of modeling commands of CAD systems. A method for integration of semantic data has been proposed, implemented, and tested with two commercial CAD systems.

Key Words: Feature-based modeling, history-based parametric, interoperability, ontology

1. Introduction

Various software systems are being used throughout the lifecycle of a product. CAx systems are used during the processes of design, engineering, and manufacturing. PDM (product data management) and ERP (enterprise resource planning) systems are also used to integrate and manage the engineering information. Due to practical reasons different CAx systems are being used. One of the major problems facing enterprises of today is how to share and exchange data among heterogeneous applications. RTI (Research Triangle Institute) estimated that interoperability problems in the product design phase resulted in one billion dollar yearly in the automotive industry of USA [1].

STEP (Standard for the Exchange of Product model data) is a set of international standards to

* Corresponding author:
Tel: +82-(0)42-869-3040, Fax: +82-(0)42-861-6080
Homepage: http://icad.kaist.ac.kr/~shhan/
Email: shhan@kaist.ac.kr

solve the interoperability problems between product models. STEP has has been successful as far as the explicit geometry is concerned. As a standard for sharing and exchange, STEP defines the generic geometric and non-geometric information required for the product data definition. However, it does not define semantics, which is the underlying information of features. There are on-going projects [2,3,4,5] inside the STEP committee of ISO (International Standard Organization) to share and exchange features. One of the problems with ISO 10303 STEP is that it does not provide a sound basis to reason with knowledge. To achieve collaboration in product development, representations of knowledge should support multiple levels of abstraction. To adequately achieve this we need a formal method for representing features, such as using formal ontologies. We propose a method of mapping modeling features based on ontology to enhance the interoperability of feature-based CAD systems.

Previous researches on features in CAx system can be categorized as: (i) (machining) feature-based modeling in CAD systems, (ii) (machining) feature recognition from B-Rep model in CAPP/CAM systems, (iii) feature data sharing and exchange among heterogeneous CAD systems. For the case of (i), most of commercial CAD systems support the feature-based modeling. For the case of (ii), in the past two decades some useful methods have been developed for the limited applications. For the case of (iii), the procedural representation [2], the feature resource [3,4,6] and the macro-parametric [5,7,8,9] projects have been in progress within the Parametrics group of ISO TC184/SC4.

The approach of this paper is different from above approaches of category (iii) in that this paper proposes a data integration method in terms of semantic interoperability. To achieve semantic integration of heterogeneous data, an ontology method is applied. This paper proposes: (1) a way to construct the feature ontology, (2) a pilot implementation that verifies interoperability between two commercial CAD systems, CATIA and SolidWorks.

2. Related works

Capturing and representing real world knowledge in information systems has been recognized in the domains of artificial intelligence, software reuse, and database management. Ontology has been proposed as means of representing knowledge for the development of database designs [10]. Ontology is defined as a specification of a conceptualization [11]. Fig. 1 shows vocabulary and structure of vocabulary of the corresponding domain. The structure of vocabulary is called taxonomy. Ontology consists of concept, relation, concept hierarchy, function relation, and axiom [12].

$O := \{ C, R, HC, rel, AO \}$

where

O : ontology, C : concept, R : relation, HC: concept hierarchy, rel : function relation, AO: axiom.

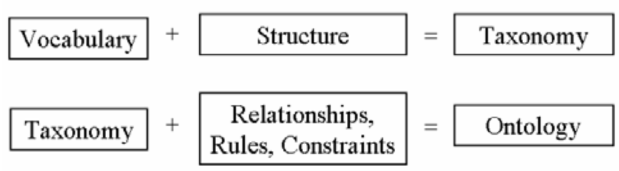

Fig.1. Concept of ontology[13]

Compared with the traditional classification structure, which consists of vocabulary and its structure, ontology endows semantics with data model by including additional rules, relations, constraints, and axioms. RDF, DAML+OIL, OWL [14], F-Logic [15], and KIF [16] are some representative languages for ontology representation.

Feature-based CAx applications have neither explicit feature taxonomy nor an explicit ontology. In order to exchange feature data between different CAD systems, it is necessary to categorize and organize features into families that are relatively independent of the application domain. Hierarchical structure of the feature taxonomy helps to facilitate the inheritance of feature properties and object-oriented implementation. Several feature taxonomy schemes have been proposed such as CAM-I project [17], rotational parts taxonomy [18], Part 48 and AP224 [28] of STEP. CAM-I constructed a feature taxonomy to derive the standard data representation for CAD systems, Kim [18] constructed a rotational feature taxonomy based on the features of PDES [20]. For feature-based data exchange, Spitz and Rappoport [21] introduced the concept of Universal Product Representation (UPR) architecture and presented a methodology for feature-based data exchange between different commercial CAD systems. Dartigues et al [22] proposed an ontological approach for integrating CAD and CAPP. They developed a shared ontology and domain specific ontologies in the KIF (Knowledge Interchange Format) language. Domain specific ontologies are developed after analyzing the CAD software and the CAPP software.

In the building and construction domain, there have been efforts to use feature ontology to estimate the construction cost. IFC (Industry Foundation Classes) defines modeling features from the viewpoint of designers of the construction domain [23]. Staub-French et al. [24] extended the scope of features by adding required features to IFC standard. Their ontology is formal, general, and system independent. They implemented a construction cost estimation system by adding supplementary features such as an opening and a turn which have to be considered for cost estimation.

For the manufacturing process, Ciocoiu et al. [25] used ontology to express semantic information among different applications that should be integrated. They presented an example of using a common ontology as an Interlingua for facilitating exchange of manufacturing process information between ProCap and ILOG. Gruninger et al. [26] used Process Specification Language (PSL) as a mediator ontology. The PSL defines a neutral representation for manufacturing processes. The axioms of PSL are organized into PSL-Core and a set of extensions. PSL-Core is the set of axioms written in KIF (Knowledge Interchange Format) and uses only the non-logical lexicon of PSL-Core. The purpose of PSL-Core is to axiomatize a set of intuitive semantic primitives that is adequate for describing the

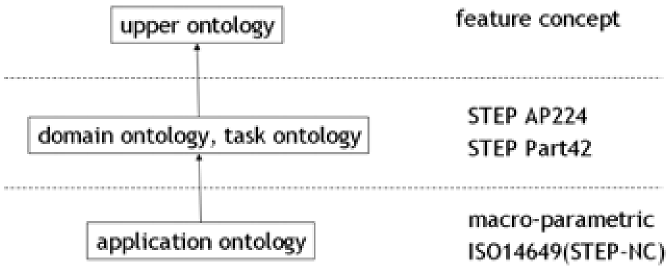

Fig. 2. Classification of ontologies

fundamental concepts of manufacturing processes.

Kim et al [27] proposed the product ontology and showed how to share information by semantic mapping based on the ontology. They focused on ontology design procedures, and semantic mapping. Patil et al [28] utilized a standards-based approach to develop a Product Semantic Representation Language (PSRL). To enable semantic interoperability, mathematical logic and corresponding reasoning has been used to determine semantic equivalences between the application ontology and the PSRL.

Choi [7] and Mun [8] define a neutral set of modeling commands that describes features and design history and then exchange data using the XML file of commands history. Although that method allows mappings between different terminologies which mean the same but syntactically different, the mappings can be done only grammatically, not semantically. To allow the semantic mappings, this paper extends the approach of [9] by using ontology to interface heterogeneous CAD systems in a semantic way.

Compared with the previous researches, this paper focuses on the practical CAD model interoperability with: i) the feature ontology based on the marco-parametrics approach; ii) the application of commercial CAD systems such as CATIA and SolidWorks.

3. Ontology based feature sharing

3.1 Layered ontology

To develop and apply ontologies, ontologies are classified into different levels. A classification system that uses the subject of conceptualization as the main criterion has been introduced by Guarino [29]. He suggests the development of different kinds of ontologies according to the level of generality such as shown Fig. 2. The upper ontology describes general concepts which are independent of a particular domain. The domain ontology describes the vocabulary of a domain by specializing the concepts introduced in the upper ontology. The application ontology is has a narrow scope for the particular problem.

The upper ontology corresponds to the underlying concept of design features and the domain ontology corresponds to ISO 10303 AP224 [19] for machining features and Part 42 [30] for shapes. The application ontology has been built using the neutral modeling commands of the macro-parametric method [7,8]. The macro-parametric method is intended to transfer parametric information by exchanging the macro (or journal, script) file which contains the modeling history.

3.2 Building the ontology

The top-down approach of Figure 3 is suitable for a new application area where the shared ontology defining the common terminologies is built first and the source ontologies are built by inheriting the shared ontology afterwards. This helps different applications to interoperate because their data models are related.

However, many applications already exist and are developed based on different data models. As shown in Fig. 3, the source ontology and the shared ontology can be bridged based on pre-exiting source ontologies. This bottom-up approach is necessary when heterogeneous data sets are integrated after applications have been established in the domain. The bridging is defined by axioms that specify the relations between different applications. The bridging describes the relations of syntactically different but semantically same data [30]. In this paper, the neutral commands of the macro-parametric approach, which have been defined by analyzing modeling commands of several commercial CAD systems, are formatted as the shared ontology using OntoEdit. Source ontologies are implemented by inheriting axioms from the shared ontology.

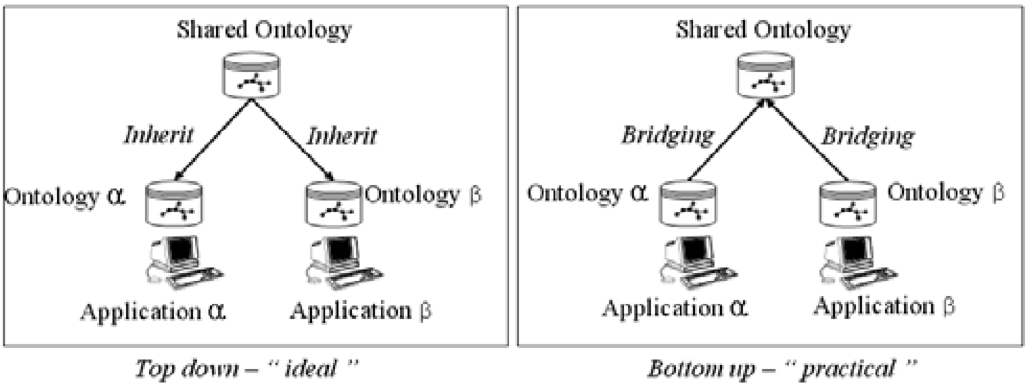

Fig. 3. Different approaches of building ontology

The macro-parametric approach defines a set of feature modeling commands [5,6] to exchange feature-based CAD files. Macro-parametric commands have semantics compared to the current STEP AP203 or Part 42 because modeling commands have more semantics than B-rep. Still, macro-parametric commands do not have enough semantics for automatic feature translations. We have implemented the shared ontology based on the feature commands defined in the macro-parametric by adding semantic information. We also constructed the system-specific source ontologies of two commercial CAD systems

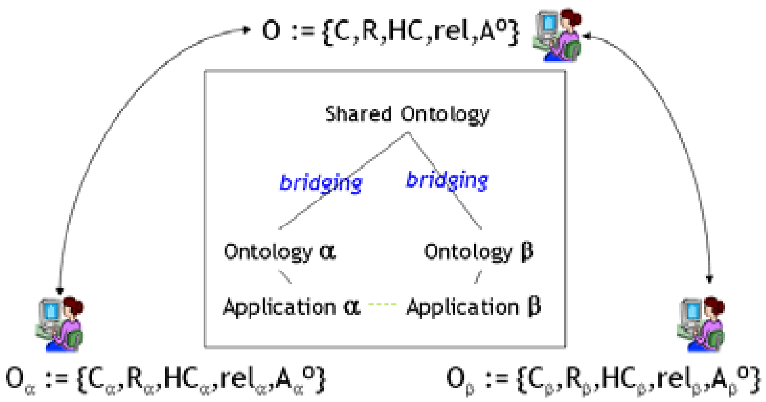

Fig. 4. Reasoning with ontology

and bridged them by defining axioms.

Because commercial CAD systems had been developed based on their own data models, they can be analyzed to build the source ontologies. As the previous studies [5,6] of authors have defined a set of neutral commands, they are translated to the shared ontology.

3.3 Reasoning using ontology

The shared ontology can be defined by the domain expert who knows the corresponding domain. The source ontologies can be defined by the application experts who know the application and also understand the shared ontology. Even if the expert of A knows only the application A and the shared ontology, while the expert of B knows only the application B and the shared ontology, the two applications can interoperate through the reasoning of axioms.

To share and exchange data between heterogeneous systems, the reasoning should map data whose syntactic definitions are different but semantics are same. If the shared ontology is bridged with the source ontology A and with the source ontology B respectively, then the application A and the application B can interoperate (See Fig. 4).

Bridging is possible by defining axioms. Axioms of the source ontology A and the source ontology B are defined by inheriting the axioms from the shared ontology. Additional axioms are defined for the application specific data whose semantics are different or whose semantics are deficient. This approach does not guarantee that there are no contradictions between the source ontology A and the source ontology B. If the axioms are strictly defined in accordance with the shared ontology, it can be assumed that there are no contradictions. The method to resolve these contradictions is beyond the scope of this paper at this time.

4. Ontology of modeling features

4.1 Taxonomy of modeling features

To build the ontologies of modeling features of CAD systems, the taxonomy of modeling features has been constructed by analyzing the manufacturing features defined in AP224, the feature resource [4], and the neutral commands set [7,8,9]. AP224 and the feature resource define form features, but the macro-parametric also defines operations required to build a solid model. "modify_operation" or "select_operation" are examples. The feature taxonomy of Fig. 5 is used to represent the feature ontology. The prismatic, rotational, auxiliary, operation, and sheet features are inherited from the solid_feature. The features of this level are abstract features that can not be instantiated. The instantiable features are prismatic_primitive_feature, extruded_feature, swept_feature, and lofted_feature.

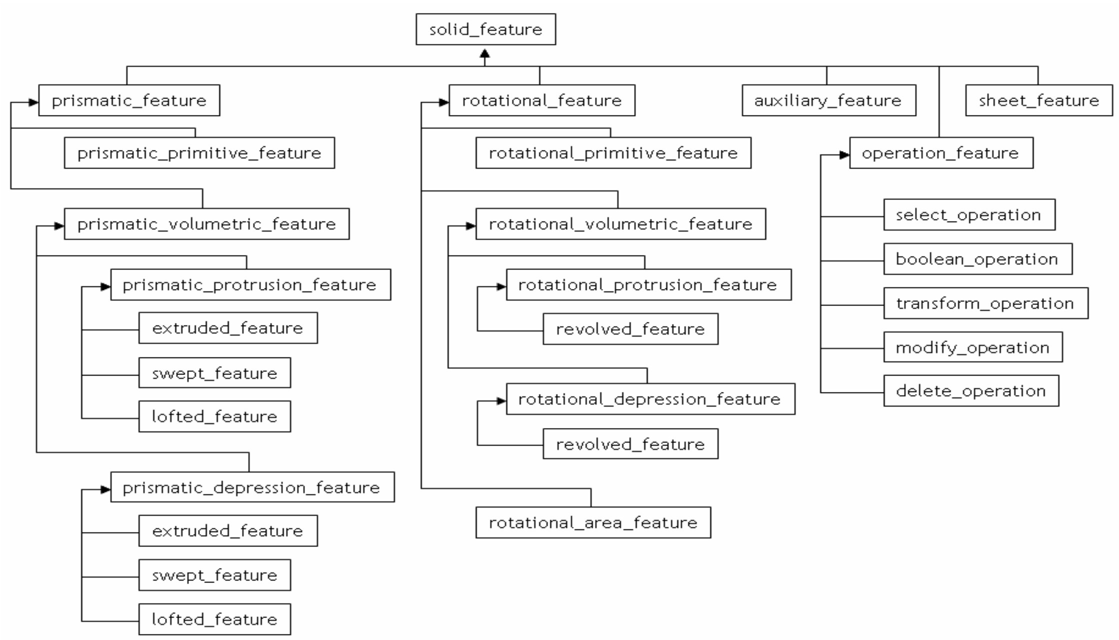

Fig. 5. Taxonomy of the modeling features

Based on the feature taxonomy, the other elements of ontology such as concepts, inheritance, relations, and range are defined. The pocket concept in Figure 6 contains the target face on which the sketch is drawn, the direction, the height, the length, and the width as relations. The range specifies the type of the data.

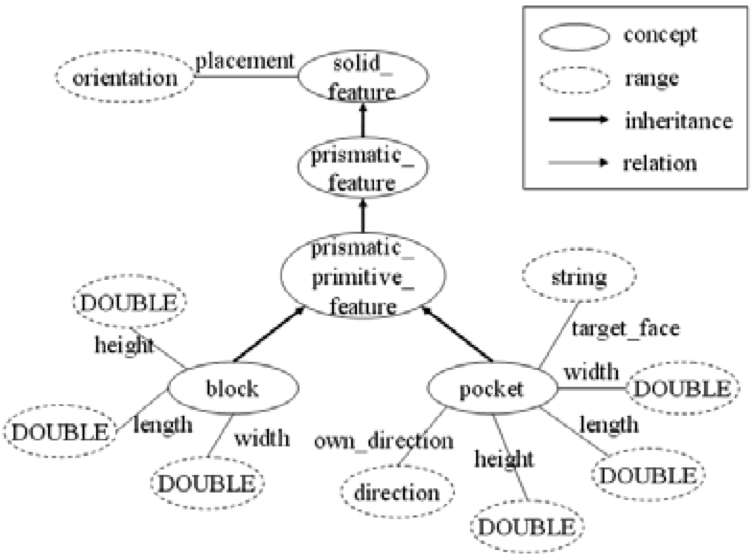

Fig. 6. An example of concept, relation, and range

4.2 Axiom

The axiom enables semantic query in the ontology. Humans can recognize the syntactically different facts of same meaning, but machine cannot do so without a n explicit description. An axiom provides knowledge with the data model so that it allows machine to understand the meaning of the fact. F-Logic is used for ontology representation in this paper. F-Logic is a deductive, object-oriented database language which combines the declarative semantics and expressiveness of deductive database languages with the rich data modeling capabilities supported by the object-oriented data model.

Definitions of modeling features of various CAD systems are slightly different. For example, in the case of the hole feature, the center position of the hole is represented by one Cartesian point in CATIA, whereas it is represented by three real numbers in SolidWorks. Fig. 7 shows how the definitions of hole features in the two CAD systems are different. In this context, axioms enable CAD systems to understand each other by specifying that two syntactically different variables are semantically same.

Hole definition in CATIA

```
Set reference1 = part1.CreateReferenceFromBRepName(
   "BorderFVertex:(BEdge:(Brp:(Sketch.2;1);None:(Limits1:();Limits2:()));
   WithTemporaryBody;WithoutBuildError;WithInitialFeatureSupport)", factory2D1)

Set reference2 = part1.CreateReferenceFromBRepName(
   "FSur:(Face:(Brp:(Pad.1;2);None:());
   WithTemporaryBody;WithoutBuildError;WithInitialFeatureSupport)", pad1)

Set hole1 = shapeFactory1.AddNewHoleFromRefPoint(reference1, reference2, 10.000000)
```

Hole definition in SolidWorks

```
Part.FeatureManager.HoleWizard 2, 0, 21, "1/16", 0, 0.007935, 0.01, 1, 2.059488517353,
   0, 0, 0, 0, 0, -1, -1, -1, -1, -1
```

Fig. 7. Different definitions of hole features in CATIA and SolidWorks

Fig. 8 shows an example of ontology definition where axioms are written in F-Logic format. In the concepts section, a hierarchy of a round_hole feature is defined. A rotational_feature is inherited from a solid_feature and a rotational_primitive_feature is inherited from a rotational_feature in turn. A round_hole is inherited from a hole. In the local relations section, properties and ranges of the round_hole is defined. This definition of round_hole is based on the set of neutral modeling commands proposed by Mun [8]. Two rules are defined in the axioms section, where the first rule states that if and only if a concept X has a property W whose range is Y and Y has three properties of DOUBLE type, then X has three properties of DOUBLE type. In F-Logic, 'X:Y' expresses that X belongs to class Y; 'X::Y' expresses that X is a subclass of Y; 'X[Y=>Z]' states that the single-valued method Y is defined as a member of the class X and the corresponding result object belongs to the class Z. The second axiom in Fig. 8 represents that a variable whose attribute is of a Y type and a Y has four variables equals a variable whose has same kind of four variables as the attributes.

```
// 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]).
```

Fig. 8. Ontology definition of the round_hole feature

4.3 Mapping and reasoning

One-to-one mapping can be defined for concepts between the shared ontology and the source ontology. The mappings for concepts are stored in the source ontology. After the shared ontology has been built based on the neutral commands set, the application expert builds the source ontology using the shared ontology. If the mappings for concepts are not defined, the reasoning mechanism detects the mapping. In addition, for concepts that cannot be mapped one-to-one, reasoning by the relation axioms is applied. Both the mappings for concepts and the relation axioms have been manually defined.

If the mapping between concepts is defined manually, it explicitly states that two syntactically different concepts represent the same data. The manual mapping is defined by local relations and axioms in this paper. The round_hole feature and the simple_hole feature in Fig. 9 represent the same feature information. The difference between the two in addition to the name difference is how they define their properties. The types of all properties of the simple_hole are primitive types whereas the types of some properties of the round_hole are concept types. A primitive type is a built-in object type defined in F-Logic. The mapping tells that the properties of two feature definitions are semantically identical by explicitly describing the difference of definition. The mapping is defined by axiom. The axiom in Fig. 9 describes that the center point of the hole feature can be defined by a Cartesian point or by three real numbers. The round_hole feature defined in Fig. 9 has the center property. The center is defined as a position in the Cartesian coordinates system. Some CAD systems define the center by three variables of the double type as the property. Other CAD systems define the center by the double array of size three. This kind of syntactic heterogeneity is bridged by the axiom that shows two concepts are semantically same. Fig. 9 shows that the axiom bridges syntactically different concepts.

```
// Hole definition in the shared ontology
round_hole[cs=>coord_sys;center=>position;depth=>DOUBLE;bottom_angle=>DO
UBLE;head_angle=>DOUBLE;bottom_type=>INTEGER].

// Hole definition in the source ontology for SolidWorks
simple_hole[ox=>DOUBLE;oy=>DOUBLE;oz=>DOUBLE;dx1=>DOUBLE;dx2=>
DOUBLE;dx3=>DOUBLE;dy1=>DOUBLE;dy2=>DOUBLE;dy3=>DOUBLE;dz1
=>DOUBLE;dz2=>DOUBLE;dz3=>DOUBLE;depth=>DOUBLE;diameter=>DOU
BLE;flip=>BOOLEAN;dir=>BOOLEAN;htype=>INTEGER;etype=>INTEGER;an
gle1=>DOUBLE;angle2=>DOUBLE].

coord_sys[origin=>position;dx=>direction;dy=>direction;dz=>direction].
direction[x=>DOUBLE;y=>DOUBLE;z=>DOUBLE].
position[x=>DOUBLE;y=>DOUBLE;z=>DOUBLE].

// The axiom to bridge the syntactic difference
FORALL X,Y (X:(simple_hole[center=>Y])) ↔
(X:(simple_hole[ox=>DOUBLE;oy=>DOUBLE;oz=>DOUBLE])).
```

Fig. 9. Axiom bridges syntactically different concepts

4.4 Semantic query

The ontology-based application can support semantic queries. Fig. 10 shows that the syntactically different but semantically same concepts are searched by the single query statement. The following query statement, described in F-Logic, means that 'search the feature which has the center as the property':

FORALL X, Y, W ← X :(Y[center□W]).

Where, FORALL X, Y ← X:Y means that "find the entities X and Y such that Y is the instance of X". In Fig. 10, the counterbored and the HoleWizard define the same feature but their terms and properties are different. By the query statement, the center(has_center) of the counterbored and the center(selectedbyX, selectedbyY, selectedbyZ) of the HoleWizard can be found through reasoning.

Query statements should be consistent to query concepts. The query for the upper-level search, the query for the lower-level search, the query for ownership as well as for simple terminology should be defined. The users want to query the application without the complex F-Logic format. They need a natural language representation of predefined queries in the graphical user interface.

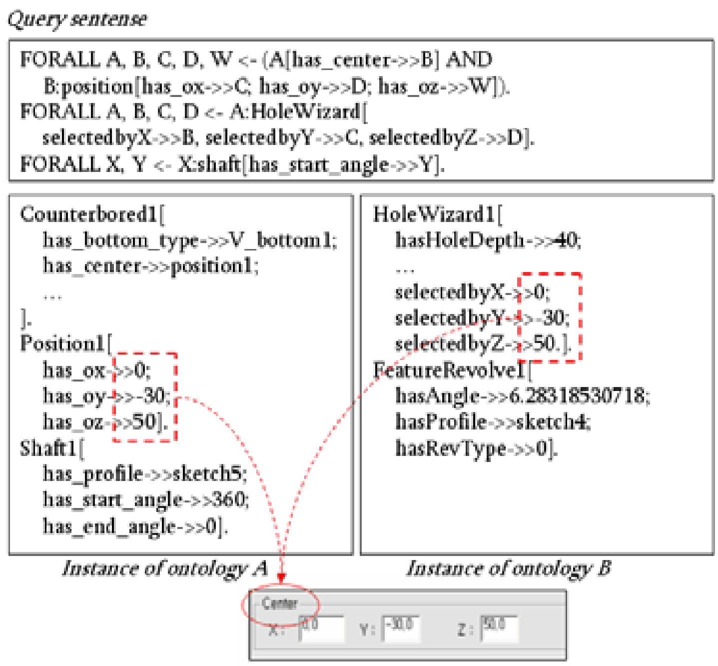

Fig. 10. Semantic query by axiom

5. Implementation and experiment

Fig. 11 shows the IDEF-0 activity diagram of the implemented application. In an IDEF-0 diagram, the arrows means, clockwise from the left, input, control, output, and mechanism. The implementation focuses on searching the same type of design features from different feature-based CAD systems, which is possible by a query based on the shared ontology. Also, feature editing such as hole removal or modification of the hole radius is possible. The input files are the commands history of commercial CAD systems such as CATIA or SolidWorks. The input commands history is translated into the instance of the feature ontology (A0). Design features in the instance file are searched and edited through the queries based on the feature ontology (A1). The modified modeling features are translated into the instance of the receiving CAD system (A2). The commercial ontology tools, OntoEdit and OntoBroker [31], are used for the implementation. OntoEdit is an ontology building tool and OntoBroker is a reasoning engine which enables semantic queries.

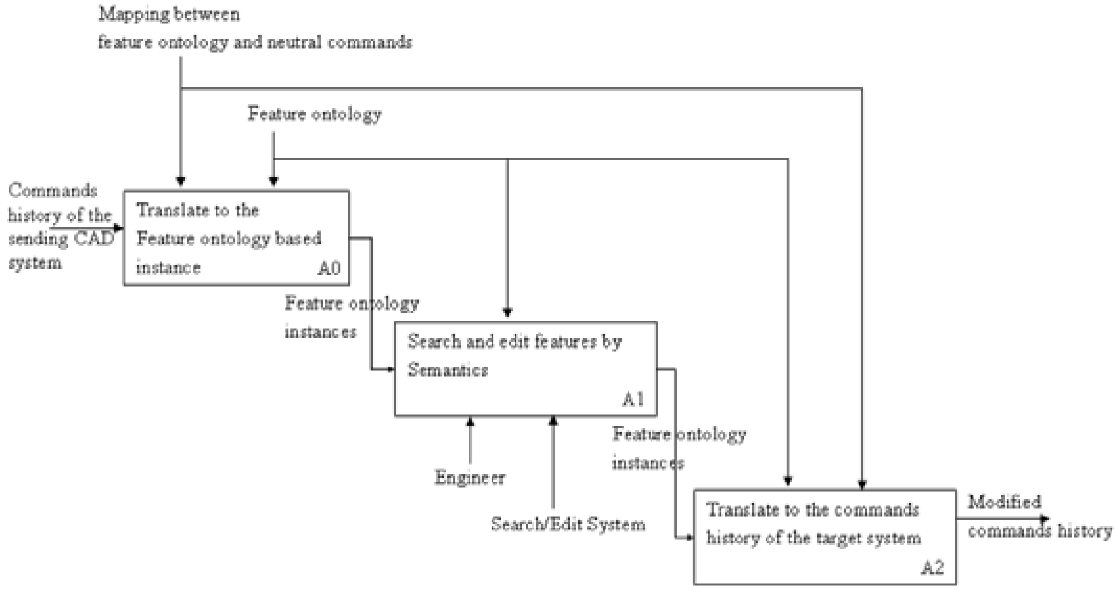

Fig. 11. IDEF-0 activity diagram of the implemented system

The implemented system is composed of three modules: OntoSmart-Translator, OntoSmart-Query, and OntoSmart-Editor. The OntoSmart-Translator module is the pre-processor that translates the commands history of a commercial CAD system into the ontology instances based on each system's feature ontology. The OntoSmart-Query is the module that queries the modeling features in terms of the shared ontology. The commercial tool, OntoBroker, is incorporated into the OntoSmart-Query module. The OntoBroker provides the communication mechanism via TCP/IP. Once the connection to the OntoBroker server is established, the OntoBroker server returns the query result of a string type with the query input of a string type. Once the specific feature is retrieved by a query, the OntoSmart-Editor modifies attributes of the feature. The OntoSmart-Editor can modify the properties of the feature or remove the feature.

Fig. 12 shows that the counterbored hole features from two Y-shaped parts are modified simultaneously by a single editing command. The parts are modeled by CATIA and SolidWorks respectively. The modeling result can be saved as script files of CATIA and SolidWorks. The script file consists of a series of modeling commands which are used to generate the part. A pilot system has been implemented which is able to modify the feature of a CATIA part file or a SolidWorks part file using the feature definition of the shared ontology.

The operational scenario of the implemented system is as follows: i) there exist the shared ontology of the feature based modeling system and source ontologies of CATIA and SolidWorks respectively. These ontologies are generated using the OntoEdit tool. Inputs are a CATIA script file and a SolidWorks script file; ii) OntoSmart-Translator translates each script file into the instance file which conforms to each source ontology; iii) translated instance files are loaded onto OntoSmart and user input features which will be modified through the graphical user interface; iv) F-Logic commands are generated based on the input features and feature information can be retrieved by OntoSmart-Query which uses the generated F-Logic commands. The generated F-Logic commands follow the definition of the shared ontology. OntoSmart-Query uses the functionality of OntoBroker; v) features can be modified according to user input, which results in modified ontology instance files. These files are translated back to script files of CATIA and SolidWorks respectively. This task is done by OntoSmart-Editor and OntoSmart-Translator respectively; vi) finally, modified parts can be displayed after CATIA and SolidWorks read the resulting script files.

The feature-based CAD systems are widely used in the mechanical industry these days. The method proposed in this paper is the novel solution to manage and integrate the heterogeneous commercial CAD data in the field. The company which needs to manage the CAD data modeled by several different CAD systems with a unified and consistent view will benefit by the approach of the paper. The limitation of current implementation is that it does not deal with native CAD part files but it deals only with the script file of a CAD system. To manage native CAD part files, a feature tree extraction module using APIs of commercial CAD systems should be implemented. That module can be a substitute for OntoSmart-Translator module.

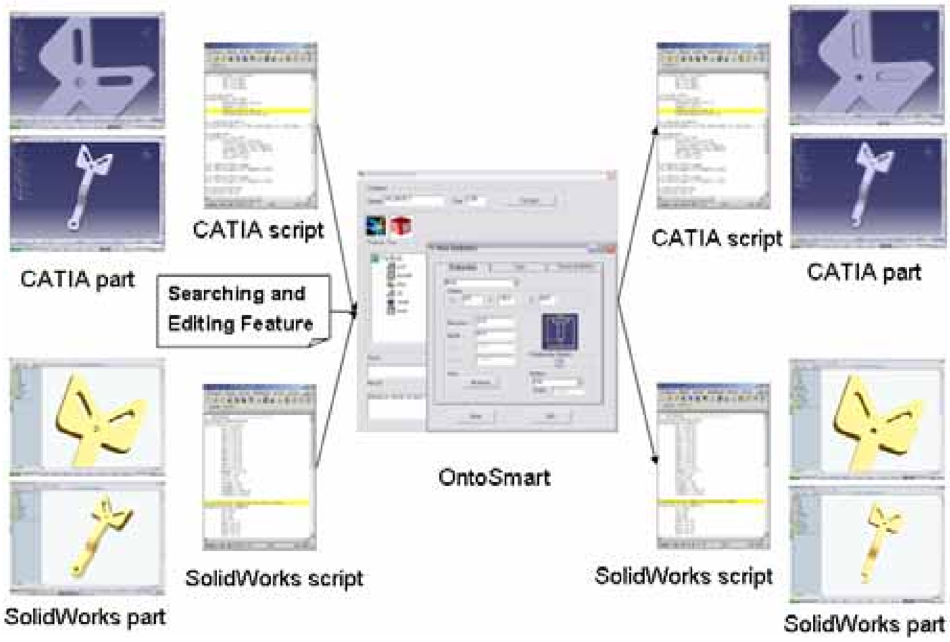

Fig. 12. Simultaneous modification of the counterbored features

6. Conclusion

An ontology-based method is proposed to enable the semantic interoperability of the feature-based CAD data. It allows commercial CAD systems to share and exchange feature-based CAD models semantically. The taxonomy of modeling features and the shared feature ontology have been constructed by analyzing the STEP AP224, the feature-resource and the macro-parametrics approaches. The macro-parametrics approach has primarily been used to define the shared ontology. To enable the semantic interoperability, the relational axioms have also been defined. The proposed system has been implemented and tested. Design features of the two commercial CAD systems have been queried and edited through the user interface where two systems work as single application. Further research may accesses the CAD systems using the application programming interface (API), and construct a more rigid and robust relational and functional axioms.

References

1. 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
2. ISO TC184/SC4, (2003) ISO 10303-55 DIS, *Integrated application resource: Procedural and hybrid representation*, International Standard Organization
3. Pratt, M. J., Anderson, B. D., Ranger, T. (2004), Towards the standardized exchange of parameterized feature-based CAD models, *Computer Aided Design*, Submitted
4. ISO TC184/SC4, (2003) ISO 10303-111 CD, *Integrated application resource: Construction history features*, International Standard Organization
5. ISO TC184/SC4, (2004) ISO 10303-112 CD, *Integrated application resource: Standard modelling commands for the procedural exchange of 2D CAD models*, International Standard Organization
6. Stiteler, M. (2004), CHAPS - construction history and parametrics, *CHAPS Program Final Report*, January 2004, Advanced Technology Institute
7. 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), 23-31.
8. Mun, D. H., Han, S., Kim, J. H., Oh, Y. C. (2003), A set of standard modeling commands for the history based parametric approach, *Computer Aided Design*, 35(13), 1175-1179.
9. Lee, Y., Cheon, S. U., Han, S. (2003), "Enhancement of CAD model interoperability based on feature ontology", *Proceedings of International Workshop on Frontier Technology in Ship and Ocean Engineering*, held in Seoul, Korea
10. Vijayan S., Veda C. S. (2002), Ontologies for conceptual modeling: their creation, use, and management, *Data & Knowledge Engineering*, 42, 251-271.
11. Gruber, T. (1992), A mechanism to support portable ontologies. *Technical Report KSL91-66*, Stanford Knowledge Systems Laboratory.
12. Maedche, A. (2002), *Ontology learning for the Semantic Web*, Kluwer Academic Publishers.
13. Sudarsan, R. (2003), A product information modeling framework for PLM, Presentation in SC4 meeting of ISO/TC184 held in Stuttgart, Germany

14. World Wide Web Consortium, http://www.w3.org/
15. Corcho, O., Lopez, M .F., and Perez, A. G. (2002), Methodologies, tools and languages for building ontologies, *Data and Knowledge Engineering*, 46(1), 41-64.
16. Genesereth, M., Fikes, R. (1992), "Knowledge interchange format version 3.0 reference manual", *Report Logic-92-1*, Stanford
17. 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.
18. 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), 341-350.
19. ISO TC184/SC4. (2001), ISO 10303-224: *Mechanical product definition for process planning using machining features*, 2nd Ed., International Standard Organization
20. PDES Inc., http://pdesinc.aticorp.org/
21. Spitz, S., Rappoport, A. (2004), Integrated feature-based and geometric CAD data exchange, *ACM Symposium on Solid Modeling and Applications*.
22. Dartigues, C., Ghodous, P., Gruninger, M., Pallez, D., Sriram, R. (2004), CAD/CAPP integration using feature ontology, *Computer Aided Design*, Submitted.
23. International Alliance of Interoperability (2001). IFC 2x extension modeling guide, http://www.iai.org.uk
24. 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.
25. Ciocoiu, M., Gruninger, M., and Nau, D. S. (2000), Ontologies for integrating engineering applications, *Journal of Computing and Information Science in Engineering*, 1(1), 12-22.
26. Gruninger, M. (2000), The process specification language (PSL): Overview and version 1.0 specification, *NISTIR 6459*, National Institute of Standards and Technology.
27. Kim, K. Y., Chae, S. H., and Suh, H. W. (2003), *An approach to semantic mapping using product ontology for CPC environment, CE Research and Applications: The vision for the future generation*, J. Cha, R. Jardm-Goncalves, and A. Steiger-Garcao, Eds., Swets and Zeitlinger, 291-298.
28. Patil, L., Dutta, D., Sriram, R. (2005), Ontology-based exchange of product data semantics, *IEEE Transactions on Automation Science and Engineering*, 2(3), 213-225.
29. Guarino, N. (1998), Formal ontology and information systems, In Proceedings of FOIS'98, *Formal Ontology in Information Systems*, Held in Trento, Italy, IOS Press.
30. ISO TC184/SC4, ISO 10303-42: *Integrated generic resources: geometric and topological representation*, International Standard Organization
31. Ontoprise Corp., http://www.ontoprise.de