International Journal of CAD/CAM Vol. 2, No. 1, pp. 13~21 (2002)

INTERNATIONAL JOURNAL OF CAD/CAM
www.ijcc.org

Exchange of CAD Part Models Based on the Macro-Parametric Approach

Guk-Heon Choi[1], Duhwan Mun[2] and Soonhung Han[2]*
[1]*Dong Seoul College, Seoul, Korea*
[2]*KAIST, Yusong-Ku, Daejon, Korea*

Abstract – It is not possible to exchange parametric information of CAD models based on the current version of STEP. The design intent can be lost during the STEP transfer of CAD models. The Parametrics Group of ISO/TC184/SC4 has proposed the SMCH schema, which includes constructs for exchange of parametric information. This paper proposes a macro-parametric approach that is intended to provide capabilities to transfer parametric information including design intents. In this approach, CAD models are exchanged in the form of macro files. The macro file contains the history of user commands, which are used in the modeling phase. To exchange CAD models using the macro-parametric approach, the modeling commands of several commercial CAD systems are analyzed. Those commands are classified and a set of standard modeling commands has been defined. Mapping relations between the standard modeling commands and the native modeling commands of commercial CAD systems are defined. The scope of the current version is limited to parts modeling, not assemblies.

***Keywords*: CAD, exchange of design intents, parametric, STEP**

---

1. Introduction

More and more enterprises are building systems, that can manage in-house resource flows or information flows, such as PDM (Product Data Management), ERP (Enterprise Resource Planning), and DMU (Digital MockUp) systems. For product design and manufacturing, companies are enhancing their competitiveness by reducing the time-to-market and improving the quality of design using CAD/CAM/CAE.

There are two major approaches for information sharing among different CAD/CAM/CAE systems. One is the static interface of standard data exchange such as STEP and IGES, which translate models through a neutral file. A snap shot of the model is exchanged. The other is to implement the dynamic interface by standardizing API (Application Program Interface). AIS (Application Interface Specification) of CAM-I (Consortium for Advanced Manufacturing International) provides a standard programming interface for users to interface their own application with CAD/CAM systems.

Whereas the static interface deals with the content and structure of data, the dynamic interface provides solid modeling and geometric operations [9]. AIS is the standardized interface of geometric modelers. Version 2.1 of AIS has not yet implemented the API for features, tolerances, parameters, or constraints [2].

There are two methods for product data exchange among different CAD systems. One is direct translation and the other is through a neutral format. The method of using a neutral format starts from a pre-processor, which generates the neutral file from a native format. A post-processor receives the neutral file and converts it into the native format of the receiving CAD system.

Examples of existing neutral CAD formats are STEP, IGES (Initial Graphics Exchange Specifications), and DXF (Drawing Exchange Format). These formats have their own limitations. DXF cannot be used for 3D solid models, and IGES cannot handle non-geometric data. STEP (Standard for Exchange of Product Model Data: ISO 10303) includes the product geometric information as well as the lifecycle data. However, the parametric information cannot be transferred in the STEP data exchange [1].

2. Existing Approaches of Exchanging Parametric Information

2.1. Problems of STEP Representation for Product Data

STEP activity began in 1984 and the first set of standards was released in 1994 [21, 22]. Because the standardization process needs to freeze the technical contents at the time of voting, STEP could not reflect the latest functionalities of commercial CAD systems. In particular, STEP cannot represent design parameters, constraints, and features. Furthermore, it cannot exchange design intent between different CAD systems.

To solve these problems, it is necessary to define a new STEP schema is needed to solve this problem. The schema should include the history-based model and the

---

*Corresponding author:
Tel : +82-42-869-3040
Fax : +82-42-869-3210
E-mail : shhan@kaist.ac.kr
http://icad.kaist.ac.kr/~shhan

explicit model, which is the parametric method adopted by commercial CAD systems [5].

2.1.1. Explicit model approach

For the explicit model approach, entities for the representing geometry of a model are defined in STEP Part 42 [21]. New resources define the relationships between the parameters and the constraints between geometry elements. Thus, this approach is essentially constraint-based parametrics. The schema and the classes for constraints are already defined in Part 108 [4, 7, 12].

2.1.2. History-based model approach

Although entities for representing a CSG (Constructive Solid Geometry) model are also defined in STEP Part 42 [21], the entities that define a CSG model cannot describe parametric dimensions. Part 42 includes the entities for Boolean operation, protrusion, and rotation, which are used for a model construction operation, but it does not include functions (e.g. shelling and draft) for history-based modeling that are supported by commercial CAD systems [4].

2.2. ENGEN Project

The ENGEN (Enabling Next GENeration mechanical design) project proposed EDM (ENGEN data model), which is a product data model with parameters, features, design history, and constraints based on STEP Part 42 [3]. The purpose of the ENGEN project is to verify exchange capability of design intent, which is represented by design parameters, constraints, and features. The exchange experiments of product information including the design intent are among CAD systems such as Pro/E(PTC), I-DEAS(SDRC), and CADDS5(CV).

Because the main focus of the ENGEN project is to exchange models with constraints, EDM does not have a sufficient number of entities for the design history [8]. EDM cannot be used to represent the history-based model. Nor does EDM handle the persistent naming problem. This is a problem of referencing entities, which are implicitly created by modeling operations.

2.3. Approaches of STEP Parametrics Group

At the ISO STEP meeting in October 2001, Bill Anderson proposed the SMCH (Solid Model Construction History) schema, and provided an updated version of the Implementor's Guide of SMCH [6]. The SMCH schema [6] consists of a data structure that enables exchange of parameters, constraints, features, and design history. This schema covers the following scope.

- Geometrically constrained solid models containing parametric features.
- Representation of implicit entities and operations to enable exchange of history-based models.
- Structures from Part 42 Edition 2 for exchange of Constructive Solid Geometry using (regularized) Boolean operations of union, intersection, and difference on solid primitives, manifold Breps, and other solids.
- Exchange of *current result* of advanced Brep solid models

Hua Jiang at NIST achieved exchange of the L-block test model between Pro/Engineer and SolidWorks using an EXPRESS schema of his own. Bob Tildsley of Theorem Solutions translated the construction history Part 21 file bi-directionally between CATIA and UG [15].

To solve the persistent naming problem, Tony Ranger proposed a method that uses explicit entities to represent reference model elements. In the case of selecting one edge for filleting, the method generates and exchanges the explicit geometric entity of the selected edge. The CAD system that receives the model data can find the reference entities (or the selected edge) based on a geometry comparison.

2.4. Others

The Manufacturing Task Force of OMG (Object Management Group) published a CAD Services document for the standardized CAD system interface in October 2001 [13]. Hoffmann suggested E-Rep (Editable Representation), which is an interface for creating feature-based CAD models [10].

3. Macro-Parametric Approach

3.1. Concept of the Macro-Parametric Approach

The macro-parametric approach is another kind of history-based parametric method. To transfer parametric information including design history, a set of standard commands is defined and used as a neutral format. A macro file that records the modeling command sequence or user's modeling history is exchanged. The history of user commands, which define a high-level dynamic interface, is recorded in a macro file, and the macro file is used for the static model exchange.

We get an idea from the database recovery process where the transaction log file is used to restore the database after a crash. A set of user commands issued by a CAD designer during the design task is recorded as the modeling history, which implicitly includes the CAD designer's intent. This is another method of product data exchange because the exchanged macro file regenerates the same model inside the receiving CAD system [11].

Following Bill Anderson's definition [3], we define the term *design intent* as "some functional requirements provided by customers, that is, a set of geometric and functional rules which the final product have to satisfy". Therefore the design intent is represented by constraints, parameters, design history, and features. By translating the command sequence, designer intent can

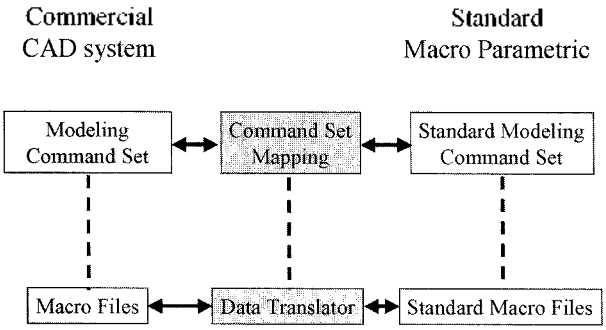


Fig. 1. Mapping in macro-parametric.

also be implicitly translated.

Fig. 1 shows the data exchange model used in the macro-parametric approach. Mapping in the macro-parametric is comprised of two levels. One is the schema mapping between the user commands set of a CAD system and the standard commands set. The other is the actual data translation between the macro file of the commercial CAD system and the standard macro file. To translate data models between CAD systems, the macro file generated by a commercial CAD system is translated into a standard macro file, which is again translated into a macro file of the receiving CAD system.

3.2. Standard Commands Set

To standardize the user modeling commands of commercial CAD systems we surveyed the modeling commands of 6 commercial CAD systems, CATIA, Pro/Engineer, UG, IDEAS, SolidWorks, and SolidEdge. We defined a set of 167 standard modeling commands. As can be seen from Table 1 and Fig. 2, the set of standard modeling commands has 4 groups at the uppermost root level - SKETCH, SOLID, SURFACE, and CONSTRAINT - and the modeling commands are further classified into 4 levels [8].

The standard commands set is a common set of modeling commands that are used in part modeling modules of major commercial CAD systems. It does not include commands of assembly or sheet metal modules. The macro-parametric approach does not deal with assembly level such as a mate condition or CAD system's specific and sophisticated commands such as the ToroidalBend command of Pro/Engineer.

Table 1. Grouping of standard modeling commands

Root : 4
SKETCH, SOLID, SURFACE, CONSTRAINTS
LEVEL 1: 20
Create, Operate, Modify, ...
LEVEL 2: 63
LEVEL 3: 100
LEVEL 4: 22
TOTAL : 167

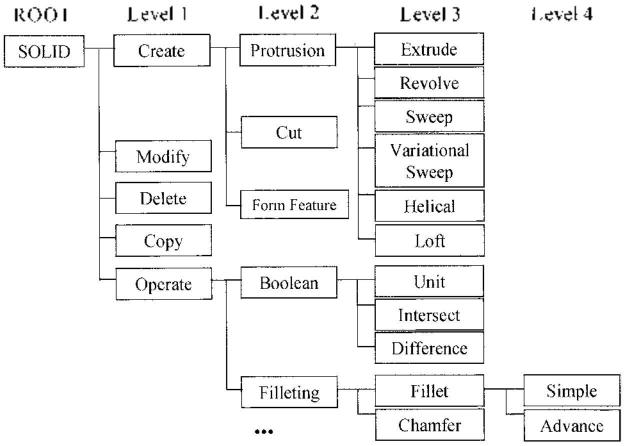


Fig. 2. Classification of standard modeling commands.

3.3. Macro Files

Commercial CAD systems use different names for the macro files; the trail file of Pro/Engineer, the program file of IDEAS, the macro file of UG, the script file of CATIA, and the swb file of SolidWorks. Storage formats are also different. Macro files of CATIA and SolidWorks are written in visual basic code. Macro files of Pro/E, UG, and IDEAS, are saved as text files at the level of GUI (Graphical User Interface).

During the macro data exchange, the following are identified as difficult to convert because recoding of these items is system-dependent.

- Information of selected modeling entities
- Coordinate system which is applied
- Information required to generate a local coordinate

Pro/E records the names of selected entities in its macro file and uses the 3D coordinates. IDEAS records the names of selected entities and uses the screen coordinates. UG does not record the names of selected entities during the modeling process but it does use the screen coordinates. CATIA records the names of entities during the modeling process and uses the 3D coordinates. For the recording of entity names, the topological naming method is used in CATIA. The macro file of SolidWorks records the names of entities that the user selected during the 2D profile stage, but does not record names during the 3D solid stage. It does not record the local coordinates used in the 2D profile stage; instead it records the 3D world coordinates.

Constraint, parameterization, and feature information are also recorded in the macro file in addition to the operation history. For example, the following constraint commands are defined in the standard commands set.

CONSTRAINTS_Create_Constraint_Vertical,
CONSTRAINTS_Create_Constraint_Horizontal,
CONSTRAINTS_Create_Constraint_Perpendicular

3.4. A Comparison of Macro-Parametric with SMCH

While SMCH uses a hybrid method, the macro-parametric schema uses a pure history-based approach. SMCH is composed of two parts, B-rep and construction history [2], whereas the macro-parametric employs only the modeling history.

In SMCH, for example, a 2D profile, in which the explicit relationships among entities are defined, is used for the protrusion operation. However, the macro-parametric approach uses the history-based method even in creating a 2D profile. Whereas SMCH explicitly defines geometric data, topological data, and their relations to create a 2D profile, the macro-parametric approach defines a sequence of modeling commands which have attribute values as arguments without any geometric or topological data structure.

For example, to create a line segment which is determined by two points, only the SKETCH_Create _2D_Line_2Points command with two CARTESIAN _POINTs is required in macro-parametric, whereas entities such as CARTESIAN_POINT, VERTEX_POINT, DIRECTION, VECTOR, LINE, EDGE_CURVE, ORIENTED_EDGE should be defined in SMCH.

The exchange file size of the macro-parametric is smaller than that of SMCH. The macro-parametric also has an advantage in Web applications. It is easy to develop a translator using the macro-parametric approach because most commercial CAD systems support their native macro files even when there is missing information.

4. Implementation and Experiments

4.1. Architecture of the Translation System

Exchange of macro files with parametric information has been implemented and tested between CATIA and SolidWorks. The translation test is accomplished in one-way from SolidWorks to CATIA. It has been implemented on Microsoft Windows 2000 and MS Visual C++. An ACIS 4.0 kernel is used as the geometric modeling module of the translation system.

Features (extrusion, pad, blend, and cut) and 2D profile entities (line, arc, circle, parallel constraint, and perpendicular constraint) can be handled in the system. Because we defined only the commands for part modeling as the standard modeling commands set, an assembly model cannot be translated by the system. Attributes such as color and material cannot be translated either.

Fig. 3 shows the architecture of the macro-parametric translator. The translation system has a *macro input module*, a *macro output module*, a *graphical user interface*, and a *macro data translation module*. The *macro data translation module* uses the ACIS geometric modeling kernel and table of commands mapping. For the translation, an internal geometric model has been created using the ACIS kernel. It is used to translate commands that are difficult to map directly.

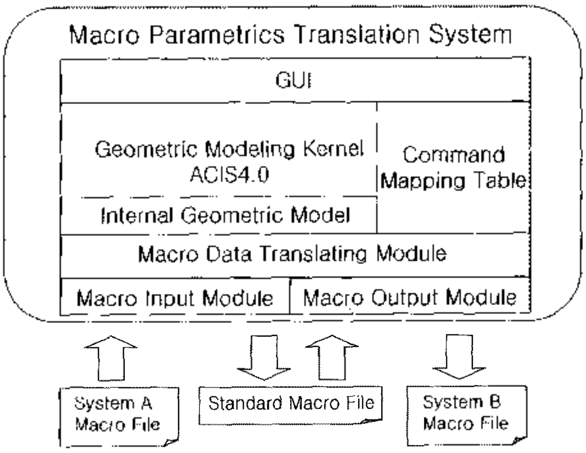


Fig. 3. Architecture of the macro-parametric translator.

4.2. Translation Process

Fig. 4 shows the translation process between two different CAD systems. Modeling commands of the input macro file is grouped into sections of construction work. The commands in each section are further classified into two groups; one is the group of commands that can be directly translated, and the other is the group of commands that cannot be directly translated.

The commands of indirect translation are ones of which arguments are difficult to translate. An internal geometric

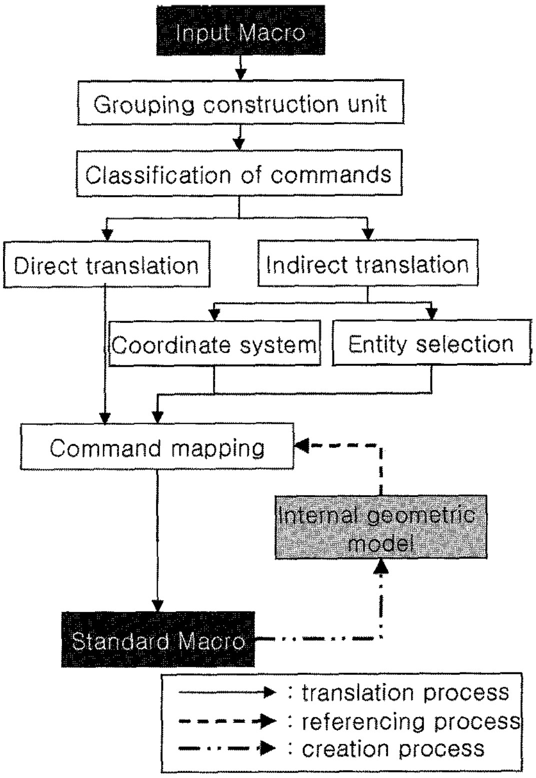


Fig. 4. Translation process.

model based on ACIS has been used to extract necessary geometric information to translate the arguments. Examples of *indirect translation* are commands related to the coordinates system, entity selection, and commands that are not supported by the receiving system.

4.3. Usage of the Internal Geometric Model

Compared with API functions of a modeling kernel, user commands that are recorded in a macro file constitute a relatively high level interface. Because of this characteristic, a macro file can transfer the design intent. But it is difficult to obtain detailed geometric and topological information such as point, edge, or face from a macro file. Detailed information of a CAD model is required to compute the local coordinates of a feature or to solve the persistent naming problem. The persistent naming problem originated from different naming conventions for entities generated implicitly inside CAD systems.

The entity selection command, SELECT_Reference _Entity, needs as arguments, persistent identifiers of entities, which are internally generated by a construction operation. Although those arguments semantically correspond to each other, they have different formats for different commercial CAD systems. The internal geometric model contains the detailed geometric information of the CAD model. During the translation process, the translator obtains the missing geometric information from this ACIS model.

Fig. 5 shows an example of using the internal geometric model to compute the local coordinates of a feature. When we make a cut feature, we need to select the sketch plane and to decide on the local coordinates of the selected surface. In general, the z vector of the local coordinates is determined by a normal vector of the selected plane. The internal geometric model is used to regenerate the local coordinate information of SolidWorks.

There are differences in the arguments of the entity selection commands of commercial CAD systems. For example, SolidWorks records coordinates of 3D points and the type of selected entity, while CATIA records the selected entity information using a topological naming method.

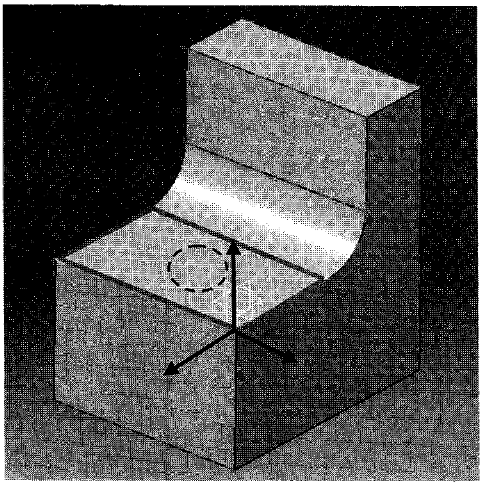
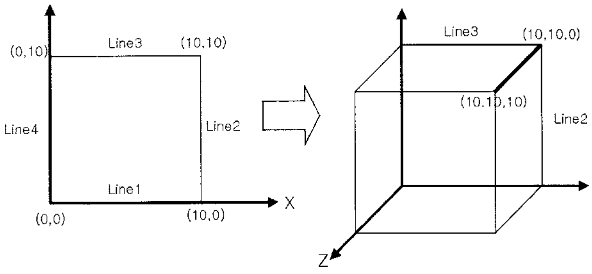


Fig. 5. Detail geometric data from ACIS.



SolidWorks : Edge located by (10,10,5)
CATIA : Edge created by extrusion of intersection between Line2 and Line3

Fig. 6. Arguments for entity selection command.

The differences are outlined in Fig. 6. In the case of selecting one edge that has been created implicitly during a protrusion operation after a sketch, SolidWorks records the selection information as "*EDGE*", *(10,10,5)*, while CATIA records as ("REdge:(Edge:(Face:(Brp:(Pad.1;0:(Brp:(Sketch.1;**Line2**)));None:());Face:(Brp:(Pad.1;0:(Brp:(Sketch.1;**Line3**)));None:());None:(Limits1:();Limits2:()));WithTemporaryBody;WithoutBuildError)", Pad1).

To map these different arguments between SolidWorks and CATIA, detailed geometric information is required. For the translation from SolidWorks to CATIA, the procedure to map arguments using the internal geometric model is listed as follows:

- Inside the protrusion feature, find the nearest edge from (10,10,5) using the ACIS model.
- Obtain the local coordinates of the starting point and end point of the found edge. If both of the z values of the two points are 0, the edge is located on the sketch plane. If both of the two z values are non-zero, the edge is located one the opposite side of the sketch plane. Otherwise, the edge is located on one of the sides of the protrusion feature. In Fig. 6, the selected edge belongs to the last case.
- Comparing the x, y values of the selected edge with those of the edges in the sketch plane using ACIS API, we find that Line2 and Line3 have the same x, y values with the selected edge in this case.
- Define arguments for the selection command of CATIA according to the topological naming method of CATIA.

4.4. Exchange of Sketch Information

In a macro file, the sketch section starts with defining the sketch plane, and ends with defining a feature of which the sketch is a part. For example, in a standard macro file, the sketch section begins with selecting the sketch plane using the plane referencing command, 'CONSTRAINTS_Create_3DReference_Plane', and it comes to an end with a feature creation command such as 'SOLID_Create_Protrusion_Extrude'. During the sketch section, the sequence of user commands of 2D sketch modeling is recorded.

For the constraints solving problem that occurs in a CAD file translation, the macro file simply describes the fact that 2D constraints are imposed, and the translator

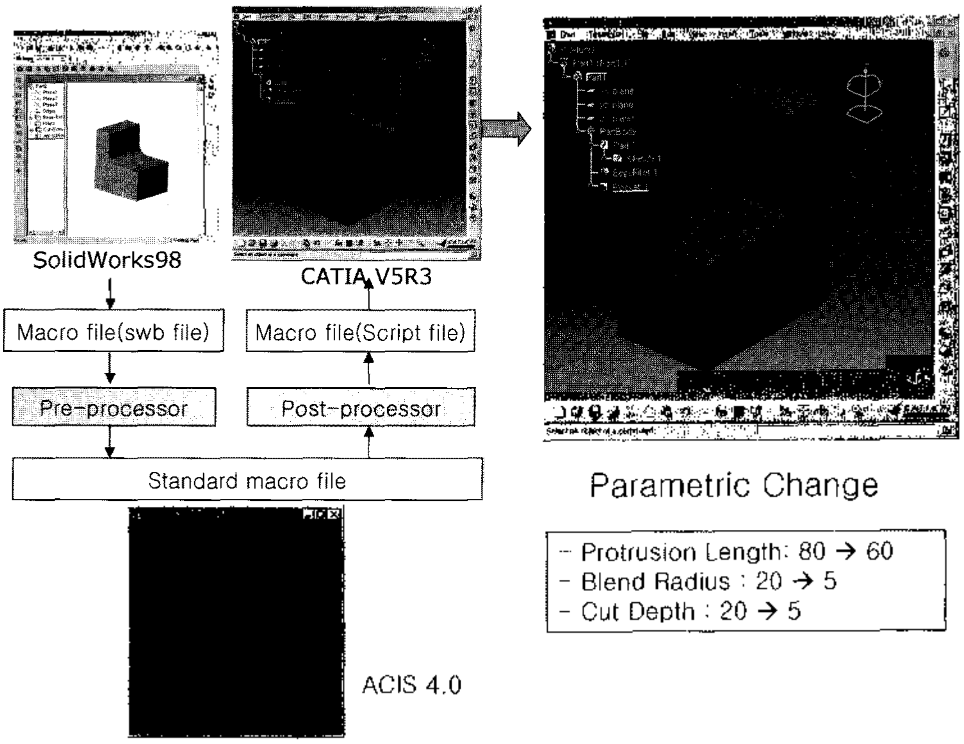
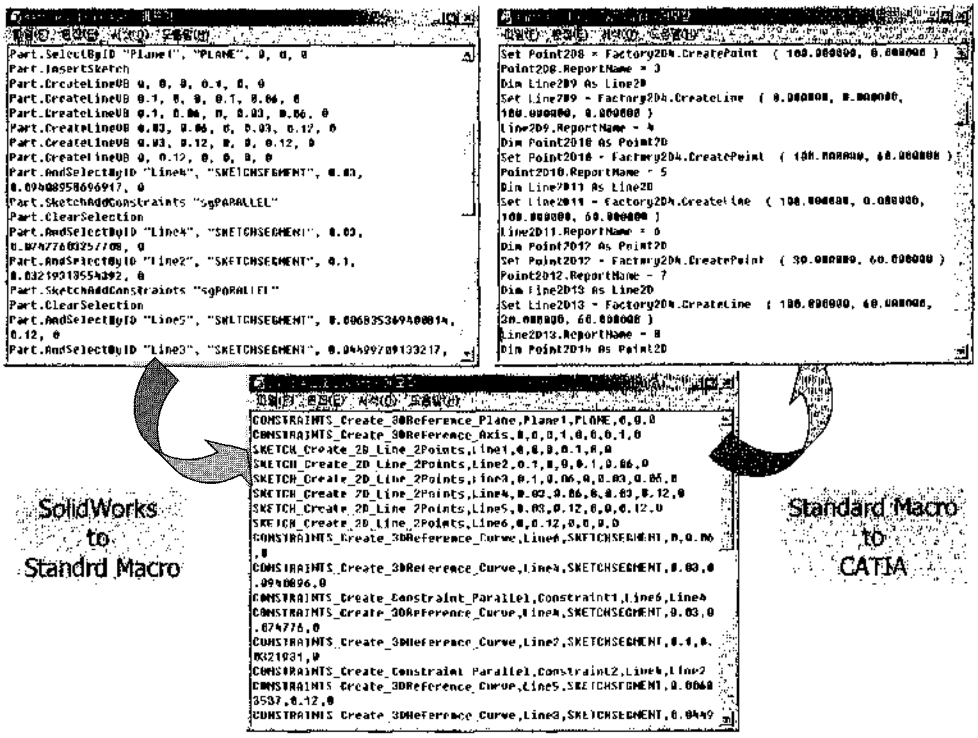


Fig. 7. Experiments of CAD model exchange.


```
Part.SelectByID "Plane1", "PLANE", 0, 0, 0
Part.InsertSketch
Part.CreateLineVB 0, 0, 0, 0.1, 0, 0
Part.CreateLineVB 0.1, 0, 0, 0.1, 0.06, 0
Part.CreateLineVB 0.1, 0.06, 0, 0.03, 0.06, 0
Part.CreateLineVB 0.03, 0.06, 0, 0.03, 0.12, 0
Part.CreateLineVB 0.03, 0.12, 0, 0, 0.12, 0
Part.CreateLineVB 0, 0.12, 0, 0, 0, 0
Part.AndSelectByID "Line4", "SKETCHSEGMENT", 0.03,
0.09408958696917, 0
Part.SketchAddConstraints "sgPARALLEL"
Part.ClearSelection
Part.AndSelectByID "Line4", "SKETCHSEGMENT", 0.03,
0.07477600257708, 0
Part.AndSelectByID "Line2", "SKETCHSEGMENT", 0.1,
0.03219318554392, 0
Part.SketchAddConstraints "sgPARALLEL"
Part.ClearSelection
Part.AndSelectByID "Line5", "SKETCHSEGMENT", 0.006835349400014,
0.12, 0
Part.AndSelectByID "Line3", "SKETCHSEGMENT", 0.04499709133217,
```

```
Set Point2D8 = Factory2D4.CreatePoint  ( 100.000000, 0.000000 )
Point2D8.ReportName = 3
Dim Line2D9 As Line2D
Set Line2D9 = Factory2D4.CreateLine  ( 0.000000, 0.000000,
100.000000, 0.000000 )
Line2D9.ReportName = 4
Dim Point2D10 As Point2D
Set Point2D10 = Factory2D4.CreatePoint  ( 100.000000, 60.000000 )
Point2D10.ReportName = 5
Dim Line2D11 As Line2D
Set Line2D11 = Factory2D4.CreateLine  ( 100.000000, 0.000000,
100.000000, 60.000000 )
Line2D11.ReportName = 6
Dim Point2D12 As Point2D
Set Point2D12 = Factory2D4.CreatePoint  ( 30.000000, 60.000000 )
Point2D12.ReportName = 7
Dim Line2D13 As Line2D
Set Line2D13 = Factory2D4.CreateLine  ( 100.000000, 60.000000,
30.000000, 60.000000 )
Line2D13.ReportName = 8
Dim Point2D14 As Point2D
```

```
CONSTRAINTS_Create_3DReference_Plane,Plane1,PLANE,0,0,0
CONSTRAINTS_Create_3DReference_Axis,0,0,0,1,0,0,0,1,0
SKETCH_Create_2D_Line_2Points,Line1,0,0,0,0.1,0,0
SKETCH_Create_2D_Line_2Points,Line2,0.1,0,0,0.1,0.06,0
SKETCH_Create_2D_Line_2Points,Line3,0.1,0.06,0,0.03,0.06,0
SKETCH_Create_2D_Line_2Points,Line4,0.03,0.06,0,0.03,0.12,0
SKETCH_Create_2D_Line_2Points,Line5,0.03,0.12,0,0,0.12,0
SKETCH_Create_2D_Line_2Points,Line6,0,0.12,0,0,0,0
CONSTRAINTS_Create_3DReference_Curve,Line6,SKETCHSEGMENT,0,0.06
,0
CONSTRAINTS_Create_3DReference_Curve,Line4,SKETCHSEGMENT,0.03,0
.0940896,0
CONSTRAINTS_Create_Constraint_Parallel,Constraint1,Line6,Line4
CONSTRAINTS_Create_3DReference_Curve,Line4,SKETCHSEGMENT,0.03,0
.074776,0
CONSTRAINTS_Create_3DReference_Curve,Line2,SKETCHSEGMENT,0.1,0.
0321931,0
CONSTRAINTS_Create_Constraint_Parallel,Constraint2,Line4,Line2
CONSTRAINTS_Create_3DReference_Curve,Line5,SKETCHSEGMENT,0.0068
3537,0.12,0
CONSTRAINTS_Create_3DReference_Curve,Line3,SKETCHSEGMENT,0.0449
```

Fig. 8. Example of CATIA, SolidWorks, and Standard macro files.

does not solve the constraints at all. Instead, the constraints are handled by the constraint solver of the receiving CAD system.

4.5. Translation Test from SolidWorks to CATIA

A translation experiment between SolidWorks and CATIA using the macro-parametric method has been performed. We used the L-block as a sample model, which is also used by the STEP Parametrics Group for the SMCH (Solid Model Construction History) project. The L-block has 2D sketch entities such as line, arc, circle, parallel constraint, perpendicular constraint, and features such as protrusion, cut, and blend.

For the macro file translation, which includes translation of design intent, we have to solve two problems. First, it is necessary to resolve differences in entity selection commands and their argument types. The second issue is that the macro file of SolidWorks does not record the local coordinates of a feature. In the experiments, we have generated an internal geometric model to map the edge information, which is selected for blending, and the face information, which is selected to make the cut feature. To obtain the local coordinates that are required by the CATIA macro file, the internal geometric model is also used.

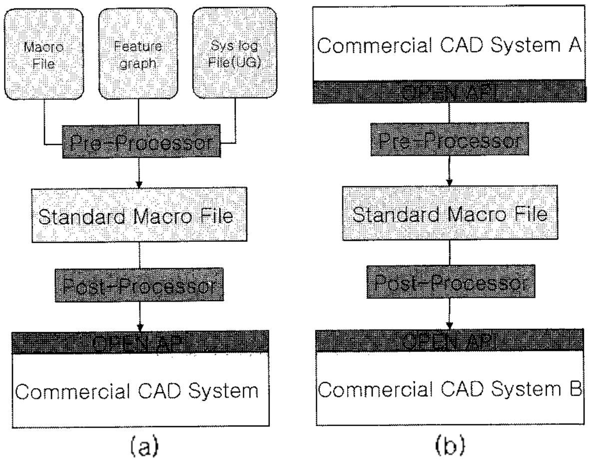
The translation result of the L-block model from SolidWorks to CATIA is shown in Fig. 7. The feature tree inside the CATIA window confirms that parametric information is transmitted from SolidWorks to CATIA. The right side of Fig. 7 shows the parametrically changed model after the translation. We have changed protrusion depth, blend radius, and cut depth. Fig. 8 shows three macro files of SolidWorks, CATIA, and the standard commands.

5. Conclusions

We propose a macro-parametric approach to exchange CAD model data between different CAD systems. In this approach CAD models are exchanged in the form of macro files, which is a sequence of modeling commands used in the modeling process. We have experimentally verified the capabilities of the macro-parametric approach by exchanging a CAD model between CATIA and SolidWorks.

The standard set of modeling commands is derived from the user commands of part modeling modules of major commercial CAD systems. In order to implement the macro-parametric exchange system, we grouped modeling commands of commercial CAD systems into categories such as SKETCH, SOLID, SURFACE, and CONSTRAINT.

We use the solid modeling kernel ACIS 4.0 to generate an internal geometric model, because some mappings between modeling commands from a commercial CAD and standard commands could not be resolved directly. The problem stems from the differences in commands of each CAD system, such as the commands for local coordinate systems, or the commands for selecting an entity. We have utilized the internal geometric model to map arguments between mismatching modeling commands. Those arguments semantically correspond to each other but have different formats.



(a) (b)

Fig. 9. Systems using API functions.

Modeling functions of part modeling modules of commercial CAD systems are quite stable (or mature). There will be few changes in the basic modeling functionality coming from software version upgrades or extensions. However, the format and contents of a macro file may change considerably. In addition, the macro-parametric method is problematic CAD systems that do not generate macro files.

Future research may focus on using API and auxiliary data. One solution is to use API to develop the pre-processor and post-processor. This idea is illustrated in Fig. 9. We may utilize auxiliary files from commercial CAD systems, for instance, the *syslog* file of a UG system, to extract missing information.

The macro files vary from designer to designer, because the modeling steps vary accordingly. We may need to clean up or to optimize the raw macro files into more compact and easy-to-convert formats.

In this study, we surveyed only part modeling commands of 6 commercial CAD systems to define a set of 167 standard modeling commands. However, assembly modeling is important in design. The standard modeling commands set should be expanded to include assembly level commands. Some compound commands or CAD system specific modeling commands such as ToroidalBend of Pro/E can be assembled with multiple standard modeling commands, which can be a separate and reusable macro file.

Previous studies on persistent naming problem generally deal with how to define persistent naming and matching of implicitly generated entities when construction and editing operations are issued in a modeling phase [16, 17, 18]. To exchange data with design intent, we need to know how to map persistent identifiers between different persistent naming schemes of commercial CAD systems [20].

References

[1] STEP Center of Korea, "ISO 10303 STEP (in Korean)", Sung An Dang, June 1997.

[2] CAM-I, "Application Interface Specification (AIS) Version 2.1", Technical Report R-94-PM-01, Consortium for Advanced Manufacturing International, Inc., Bedford, TX, 1994.

[3] B. Anderson, "ENGEN Data Model: a neutral model to capture design intent", In: G. Jacucci, G. J. Olling, K. Preiss, M. J. Wozny (Editors), PROLAMAT98, Kluwer, 1998, CDROM Proceedings of the Tenth International IFIP WG 5.2/5.3 Conference.

[4] M. J. Pratt, "Extension of the Standard ISO10303 (STEP) for the exchange of parametric and variational CAD models", CDROM Proceedings of the Tenth International IFIP WG 5.2/5.3 Conference PROLAMAT98, 1998.

[5] ISO TC184/SC4/WG12 N189, "Parametric Representation and Exchange: Preparatory knowledge about history based parametric model", http://www.nist.gov/sc4/wg_qc/wg12/n189/, 1998.

[6] Bill Anderson, "Implementor's Guide, Solid Model Construction History", PDES, Inc., April 2001.

[7] Michael J. Pratt, "Provision of an Explicit Constraints Schema in the STEP Standard", In W. Strasser, R. Klein, R. Rau (Editors), *Geometric Modeling: Theory and Practice*. Springer-Verlag, 1997.

[8] Yuchon Oh, Duhwan Mun, Soonhung Han, "Standard Modeling Command Set Based on History-Based Approach (in Korean)", KIIE Conference Proceedings, November 2000.

[9] S. P. Magleby, D. B. Jackson, "A Standardized Application Interface for Geometric Modelers". In: J. Turner, J. Pegna, M. Wozny (Editors), Product Modeling for Computer-Aided Design and Manufacturing, North-Holland, 1991, Proc. IFIP TC5/WG5.2 Working Conference on *Geometric Modeling for Product Engineering*, 1990.

[10] C. M. Hoffmann, R. Juan, "EREP: An editable high-level representation for geometric design and analysis", In: P. R. Wilson, M. J. Wozny, M. J. Pratt (Editors), *Geometric Modeling for Product Realization*, North-Holland, 1992, Proc. IFIP WG5.2 Workshop on Geometric Modeling in CAD, 1992.

[11] Duhwan Mun, "CAD Model Exchange Using Macro-Parametric Approach (in Korean)", Master thesis, KAIST, February 2001.

[12] "ISO/CD 10303-108: Product data representation and exchange: Integrated application resource: Parameterization and constraints for explicit geometric product models", Nov. 2001.

[13] OMG CAD Services Specification, http://www.omg.org/cgi-bin/doc?dtc/2001-10-06

[14] Michael J. Pratt, Bill D. Anderson, "A shape modeling applications programming interface for the STEP standard", Computer-Aided Design, **33**, 531-543, 2001.

[15] ISO TC184/SC4/WG12 N950, "WG12 Minutes from San Francisco", http://www.nist.gov/sc4/wg_qc/wg12/n950/, 2001.

[16] J. Kripac, "A mechanism for persistently naming topological entities in history-based parametric solid models", Computer-Aided Design, **29**(2), 113-122, 1997.

[17] V. Capoyleas, X. Chen, C. M. Hoffmann, "Generic naming in generative, constraint-based design", Computer-Aided Design, **28**(1), 17-26, 1996.

[18] S. Raghothama, V. Shapiro, "Boundary representation deformation in parametric solid modeling", ACM Trans. on Computer Graphics, **17**(4), 259-286, 1998.

[19] D. Agbodan, D. Marcheix, G. Pierra, "Persistent naming for parametric models", WSCG 2000.

[20] ISO TC184/SC4/WG12 N680, "On Exchangeable Construction History of Geometric Models in STEP", 2001.

[21] ISO TC184/SC4, ISO 10303-42-Part 42: Industrial automation systems and integration - Product data representation and exchange - Integrated generic resources: Geometric and topological representation, ISO, 1994.

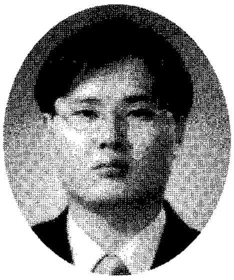
[22] ISO TC184/SC4, ISO 10303-21-Part 21: Industrial automation systems and integration - Product data representation and exchange - Implementation methods: Clear text encoding of the exchange structure, ISO, 1994.

Guk-Heon Choi is an Associate Professor at Dong-Seoul College. His research interests are geometric modeling, virtual prototype modeling, and computer graphics. He received a BS and an MS in mechanical engineering from the Yon-Sei University, Korea, and a PhD from the KAIST (Korea Advanced Institute of Science and Technology).

Soonhung Han is an Associate Professor in the Department of Mechanical Engineering at the KAIST (www.kaist.ac.kr). He is leading the Intelligent CAD laboratory (icad.kaist.ac.kr) at the KAIST and the STEP community of Korea (www.kstep.or.kr). He is the editor of the new web-based journal IJCC (www.ijcc.org). His research interests include STEP, geometric modeling kernel, VR application in design, and knowledge-based design system. He has a BS and a MS from the Seoul National University, Korea, and a PhD from the University of Michigan, USA.

Duhwan Mun is a Ph.D. candidate in the Department of Mechanical Engineering at the KAIST (Korea Advanced Institute of Science and Technology). His research interests include CAD, parametric technology, data exchange in CAD and search technology of mechanical and electronic parts, STEP, and database. He received a BS from Korea University, Korea and a MS from KAIST, Korea.

Guk-Heon Choi



Duhwan Mun

Soonhung Han