A set of standard modeling commands for the history-based parametric approach

Duhwan Mun*, Soonhung Han*, Junhwan Kim*, Youchon Oh**,

mun@icad.kaist.ac.kr, shhan@kaist.ac.kr,

everwind@icad.kaist.ac.kr, youchon@ats.go.kr

* Department of Mechanical Engineering

Korea Advanced Institute of Science & Technology

373-1, Gusong-Dong, Yusong-Gu,

Daejeon 305-701, Korea

Tel: +82-42-869-3080

Fax: +82-42-869-5210

**Automation and Components Division

Korean Agency for Technology and Standards, MOCIE

2, Jungang-Dong, Gwacheon,

Gyonggi-Do 427-716, Korea

Tel: +82-2-509-7350

Fax: +82-2-509-7423

ABSTRACT

The current version of STEP standard cannot exchange the parametric information of CAD models. Only pure boundary representations that cannot be parametrically edited are transferrable[1]. There are two approaches for the exchange of design intents such as parameters, features, and constraints. The first is an explicit approach based on constraints between pre-defined parameters and features. The second is a procedural approach based on the sequence of operations issued to construct the models. The authors have previously proposed a macro-parametric approach[2], which is a variation of the procedural approach. In this approach, CAD models can be exchanged in the form of macro files, which include the history of modeling commands. To exchange CAD models using the macro-parametric approach, a set of standard modeling commands should be defined. This paper introduces a set of standard commands and explains the process of developing the set.

Key words: CAD model exchange, History-based parametrics, Modeling command, Parametric STEP

1. INTRODUCTION

Recent CAD systems can generate product models with parameters, features and constraints. Such models can easily be edited for a variety of downstream applications. Parameters are dimensional variables within a geometric model. They provide indications of what dimensions are permissible to change. Features are high level shape constructs that make it possible for a shape designer to avoid having to work from the low level or individual curve and surface elements. Constraints and specified relationships between geometric or topological elements together provide invariant characteristics in the model, often in the interest of maintaining product functionality during modification. The designer's choice of parameters, features, and constraints constitutes an important part of what is known as design intent[3]. Following Bill Anderson's definition[4], we define the term *design intent* as the functional requirements provided by customers; that is, a set of geometric and functional rules which the final product have to satisfy. The design intent is represented by parameters, constraints, features and design history. By translating the commands sequence, the designer intent can be implicitly translated.

None of the information concerning parameters, features, and constraints can be transferred by the current version of STEP (STandard for the Exchange of Product model data) AP203 (Application Protocol)[5]. Only the basic geometry and topology of shape models of the boundary representation (B-rep) can be transferred. Therefore, it is necessary to extend the standard to capture and exchange design intent.

There are two approaches for the exchange of parametric models[6]. The first is the explicit approach using constraints between parameters and features. The second is the procedural approach, which translates the sequence of operations used to construct the model. This is based on the constructional history of the geometric model. In this procedural approach, which we also refer to as an implicit or history-based approach, the sequence of operations used in generating the product model is exchanged. Most major CAD systems use combinations of the two approaches, explicit and implicit.

This paper proposes a set of modeling commands required by the macro-parametric approach[2], which is a form of procedural approach. Section 2 analyzes related works, Section 3 briefly introduces the macro-parametric approach, Section 4 presents how the set of standard modeling commands has been developed, and Section 5 reports on an implementation and experiments.

2. RELATED WORKS

Previous studies are surveyed to highlight the differences between the macro-parametric approach and related works. Because the APIs (applications programming interface) of CAD modelers are similar to the set of standard modeling commands, they are also surveyed.

2.1 EDM

The ENGEN[4] (Enabling Next GENeration mechanical design) project proposed EDM (ENGEN data model), which is a product data model including parameters, features, constraints, and design history based on STEP Part 42[7]. The purpose of the ENGEN project is to verify exchange capability of design intent. Implementations and tests of EDM have involved the following CAD systems: Pro/E of PTC, IDEAS of SDRC and CADDS5 of CV.

Organizations, companies, and universities that have participated in the program are: Ford Motor Company, SCRA, PDES (Product Data Exchange using STEP), Purdue University, Arizona State University, ITI (International TechneGroup, Inc.), Pacific STEP, PTC and CV. Although they successfully transfer a crankshaft model between three CAD systems, the EDM, because it focuses on exchanging models with constraints, does not have a sufficient number of entities for the exchange of design history.

2.2 ISO 10303-108

ISO 10303-108 (parameterization and constraints for explicit geometric product models) has passed the CD (committee draft) voting stage[8]. It provides facilities for representing parameterization and geometric constraints as they apply to explicit shape models. It supports the explicit approach for parameters and constraints. Part 108 covers the following scope:

- The application of parameterization and constraints to explicit models
- Parameterization and constraints applicable to two- and three-dimensional models
- Means for the representation of the mathematical relationships
- Specific representations for geometric constraints

2.3 ISO 10303-55

ISO 10303-55 (procedural and hybrid representation) is currently in the balloting state as a NWI (new work item) - CD[9]. It provides the resource constructs for the representation of procedural models of construction history type, defined in terms of the sequence of constructional operations used to build them. The procedural model in this document assumes that the feature tree of a CAD system database is interrogated through API so that the final version of what has been designed is translated [10]. Conversely, the macro-parametric approach transfers a neutral representation of the journal file of commands executed during a design session. It can contain manipulation commands such as *undo* or *redo*, environment setting commands, and administrative commands that may not be important for the receiving systems. Part 55 covers the following scope:

- The specification of sequences of constructional operations

- The hierarchical structuring of constructional sequence

- The definition of a dual representation by association of a procedural model with an explicit *current result* model

- Parameterization and constraints applicable to two- and three-dimensional models

2.4 SMCH

Bill Anderson proposed SMCH (solid model construction history), which includes structures for exchange of parametric and geometrically constrained features on solids and history-based information[11]. The objective of SMCH is to develop modular capabilities to exchange history-based parametric features and constraints to enable the modification or editing of a design model in a receiving system. It is a hybrid approach, because it uses not only history-based models but also advanced B-rep solid models to validate the procedural model created in the receiving system. Construction history for curved surface and wireframe models is outside the scope of SMCH. SMCH covers the following scope:

- Solid models generated by procedural, or construction history, operations such as extruding or revolving planar profiles.

- Basic mechanisms to capture the operations (e.g., filleting) and related entities (e.g., filleted/deleted edge) to facilitate exchange with the receiving system.

- Basic features of filleting, rounding, and general swept solids. Hole features are created using primitive solids and Boolean operations.

- Representation of implicit entities and operations to enable exchange of history-based models.

- Structures from STEP Part 42 ed.2 for exchange of CSG (constructive solid geometry), manifold B-Reps and other solids. CSG uses Boolean operations (regularized) of union, intersection and difference.
- Exchange of *current result*, which is an advanced manifold boundary representation solid model.

The CHAPS (construction history and parametrics) project of PDES[12] uses SMCH to develop the edition 2 of AP203 of STEP. To enable the process, another resource module of *construction history features* is proposed[13]. This features resource has similar constructs with the solid commands of this paper[14]. Figure 1 shows the relationships between activities and documents related to the history-based parametrics. SMCH is a hybrid method where an explicit 2D sketch is used to create a procedural 3D solid model, whereas the macro-parametric method uses the procedural way to create both the sketch and the solid model. Modeling commands that are proposed in this paper support the macro-parametric method. Together with STEP Part 55, features resource, Part 108, and Part 109[15], the commands set can be a resource part of the parametrics capability of STEP standards.

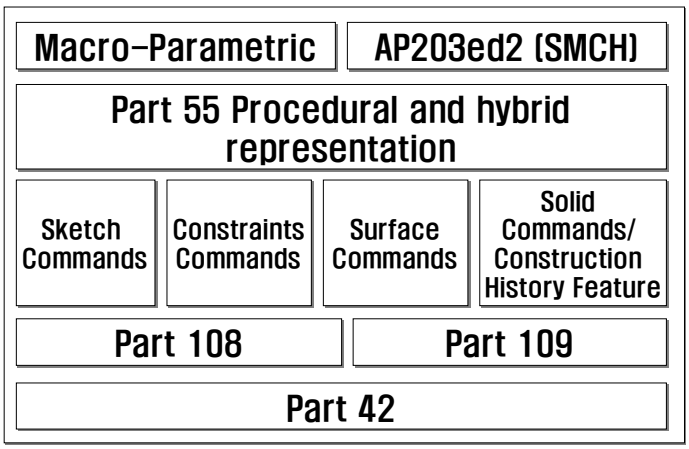


Figure 1 Relationships among related works

2.5 OMG CAD Services

This specification is an interface standard for mechanical CAD systems that enable the interoperability of CAD, CAM, and CAE tools. Developed by OMG (object management group)[16], its aim is to provide users of design and engineering systems with the ability to seamlessly integrate software across a wide variety of CAD, CAM, and CAE applications through the CORBA (common object request broker architecture) interface. This standard interface enables a distributed product design environment.

This specification provides the geometry and topology data of CAD to analysis and manufacturing applications. The objective is to establish a series of API level engineering interfaces that do not require low level data structures to answer mechanical engineering queries. To avoid many of the problems associated with data translation, this specification provides CORBA interfaces with consistent functionality across native CAD implementations. Solid models with parametric features that allow shape variation through the interface can be handled in the *CadFeature* module.

3. MACRO-PARAMETRIC APPROACH

The macro-parametric approach allows the exchange of design models based on commands history[2]. It is a history-based or implicit parametric approach. The authors find some direction in this approach from the database recovery procedure, where the transaction log file is used to recover the database after a crash. SQL (structured query language) queries are recorded in the database log file. Similarly a set of user commands used by a CAD designer during the design session is recorded as the modeling history, which implicitly includes the designer intent. The translated macro file regenerates the same model inside the receiving CAD system.

3.1 Mapping Relations

The mappings in the macro-parametric approach are shown in Figure 2, which consists of two levels. The upper semantic, schema, or conceptual level is the level of modeling commands set. The lower implementation or instance level is the level of macro files. A macro file records the sequence of commands used in a CAD system during a design session. Mapping of command sets provides the correspondences between the modeling commands of a CAD system and the standard modeling commands. Because the goal of providing user commands in a CAD system is to support users in the creation of geometric models, we can assume that modeling commands of each CAD system are almost the same at the semantic level. Granularity problems and diversity mismatch may impede or prevent one to one mapping. Hence we may need 1:N or N:1 mappings between commands from different CAD systems. After the mapping is defined, it is

then possible to develop a macro-parametric translator between the macro file of a commercial CAD system and a standard macro file [17].

An inherent incompatibility between different CAx models exists in CAD translation due to system specific modeling functionalities[20]. For example, the hole creation command of Pro/E cannot be directly translated to CATIA because Pro/E provides more diverse ways to generate hole features than CATIA. Such incompatible modeling commands can be mapped through 1:N or N:1 mappings. For example, any hole feature can be translated into CATIA by using the revolution feature and the sectional profile of the hole.

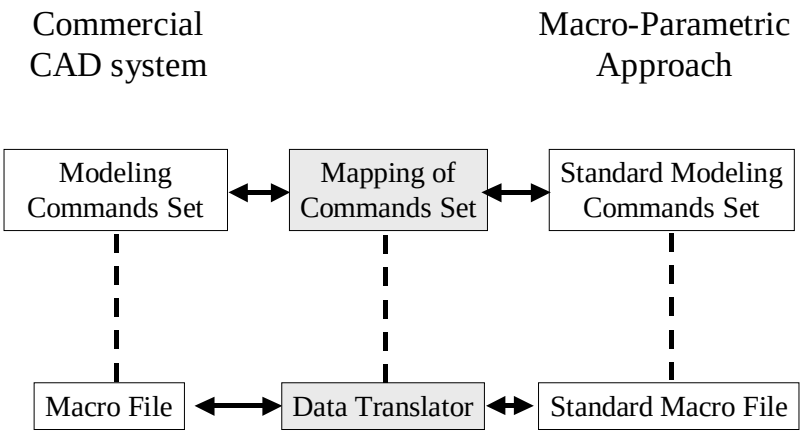


Figure 2 Mappings in the macro-parametric approach

3.2 Data Exchange Scenario

To translate CAD models between heterogeneous CAD systems using the macro-parametric approach, we need two translations, pre- and post-processing. The translation from a macro file generated by a commercial CAD system into a standard

macro file is the pre-processing, and the other translation from a standard macro file to a macro file of the receiving CAD system is the post-processing. The scenario of data exchange is shown in Figure 3. The translated model of the receiving CAD system can be edited parametrically.

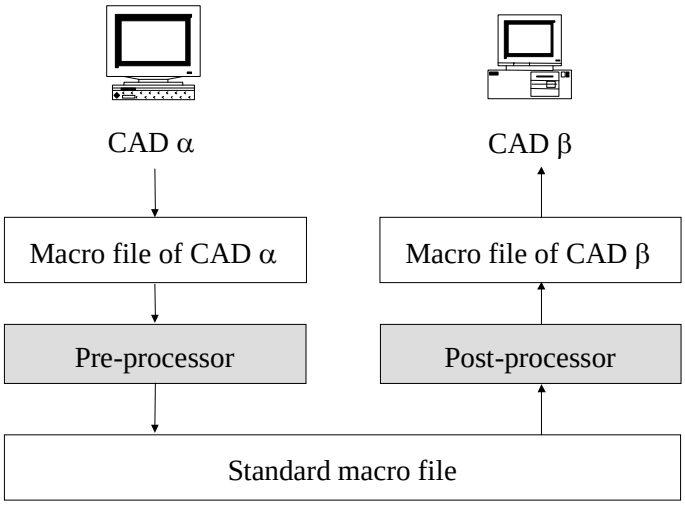


Figure 3 Pre-processor and post-processor

Figure 4 shows the translation process between two different CAD systems. Modeling commands of the input macro file are grouped into model construction tasks. The commands in each group are further classified into two subgroups; a group of commands that can be directly translated and a group of commands that cannot be directly translated.

Indirect translation often occurs when entity selection commands are to be mapped. Arguments of these commands such as persistent identifier, entity type, and coordinate information are used to identify the selected entity. There are differences among the arguments of the entity selection commands between commercial CAD systems. For

example, SolidWorks records coordinates of 3D points and the type of the selected entity, while CATIA records the selected entity using a topological naming method.

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 implicitly transfer the designer intention. But it is difficult to obtain detailed geometric and topological information such as points, edges, or faces from a macro file. Detailed information of a CAD model is sometimes required to compute the local coordinates of a feature or to solve a persistent naming problem. Persistent naming problems originate from different naming conventions for entities which are implicitly generated inside a CAD system.

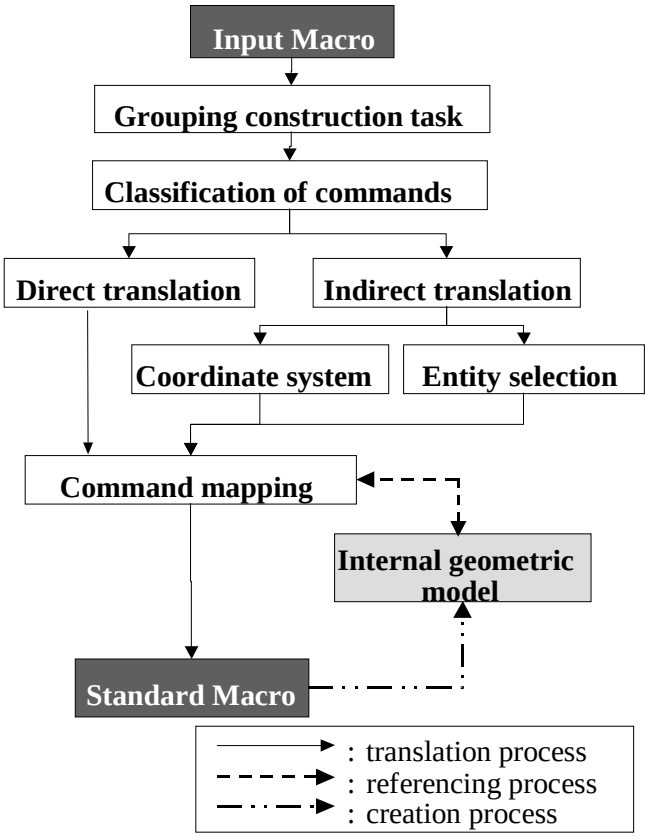


Figure 4 The translation process [2]

We have constructed an internal geometric model based on the standard macro file to obtain the detail geometric information. The detail geometry is required to map arguments of the entity selection commands which need indirect translation[2]. The persistent naming problem is an important issue of the parametric modeling approaches. Details of how to solve the problem have been treated elsewhere [18,19].

With regard to the constraints solving problem that occurs in a CAD file translation, the macro-parametric method simply translates the commands that impose constraints, and the translator does not solve the constraints. Instead, the constraints are handled by the constraint solver of the receiving CAD system.

4. A SET OF STANDARD MODELING COMMANDS

To exchange CAD models using the macro-parametric approach, a set of standard modeling commands has been developed. Figure 5 shows the process of developing the set of standard modeling commands. Modeling commands of famous commercial CAD systems have been analyzed. These modeling commands are grouped into several types. A set of *standard* modeling commands has been generated. Mapping relationships between the standard commands and commands of commercial CAD systems are defined. Based on the translation experiments of test models, the set of standard modeling commands can be finalized.

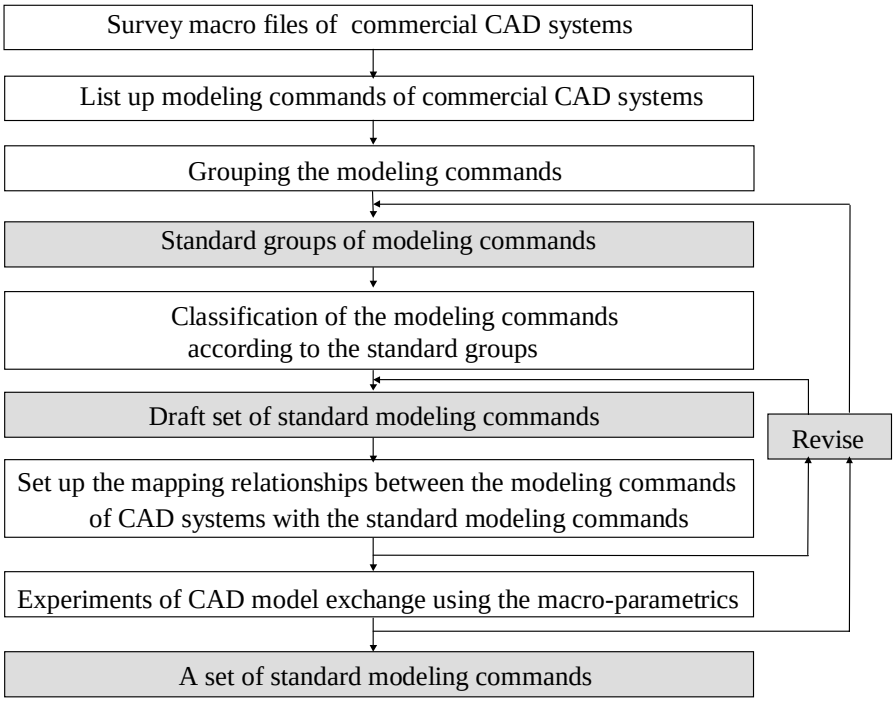


Figure 5 Process of developing a set of standard modeling commands

The proposed modeling commands are defined from a common set of user commands of commercial CAD systems such as CATIA, Pro/E, UG, IDEAS, SolidWorks, and SolidEdge. Each CAD system has its own set of modeling commands; however they have the common goal of providing an efficient user interface to define geometry. There are many similar commands such as *protrusion* or *revolve*. We can assume that commands required to model simple parts or features are similar enough to define a set of standard modeling commands. The first target of the macro-parametric translation is simple mechanical parts. The commands set can be later extended to model a sheet metal or an assembly model, but such complex models are beyond scope of this research.

4.1 Macro Files

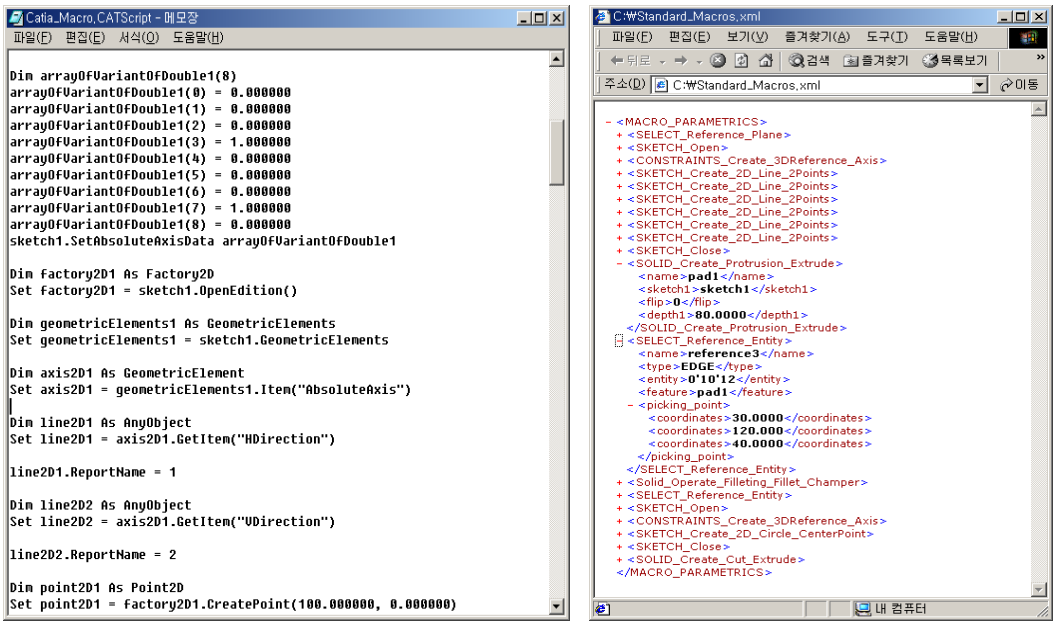
Most CAD systems use hybrid models, which contain both the B-rep information and the procedural information in their data structures. The macro file or the log file corresponds to the procedural model. A macro file is the history of modeling commands issued by a designer. Most commercial CAD systems support macro file formats; these include 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.

Formats of macro files are different for each CAD system. Macro files of CATIA and SolidWorks are written as Visual Basic codes. Macro files of Pro/E, UG and IDEAS are text files at the level of GUI (graphical user interface). Figure 6 shows two sample macro files. The left one is a *script* file of CATIA and the right one is the proposed

standard (neutral) macro file in XML format.

Pro/E records the names of selected entities in its macro file and uses the 3D coordinates. CATIA records the names of entities used during the modeling process and uses the 3D coordinates. To record entity names, a 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 it does not record names during the 3D solid stage. Nor does it record the local coordinates used in the 2D profile stage; instead it records the 3D world coordinates. Macro files of IDEAS and UG record the screen coordinates that come from the entity selection by a pointing device. It is difficult to translate such macro files.

Because the structure of the macro file of Pro/E is not well documented, we need to extract parametric information from the *Feature Tree* of the system using its API. While UG and IDEAS record *Undo/Redo* history in addition to modification history and default settings in their macro files, CATIA and SolidWorks do not record such information because their macro files are *Visual Basic* script files. They record the final history excluding *Undo/Redo* actions. Because we have experimented only with CATIA and SolidWorks, we are not concerned herein with the *Undo/Redo* history and default settings. For other CAD systems we may need to implement a cleaning algorithm for the macro file before the translation.



Macro file of CATIA **Standard macro file**

Figure 6 Examples of macro files

4.2 Grouping of Modeling Commands

The modeling commands of six commercial CAD systems such as CATIA (29 groups, 232 commands), Pro/E (4 groups, 239 commands), UG (10 groups, 390 commands), IDEAS (19 groups, 251 commands), SolidWorks (46 groups, 367 commands) and SolidEdge (8 groups, 125 commands) have been analyzed. The analyzed user commands of CAD systems are classified into 4 command groups; 2D sketch commands, surface commands, solid commands, and constraint commands. Figure 7 shows the 4 groups of the root level, SKETCH, SOLID, SURFACE, and CONSTRAINT, and they are further classified in detail. Naming of standard commands is set according to the classification. For example, the standard command for extrusion

is *SOLID_Create_Protrusion_Extrude*.

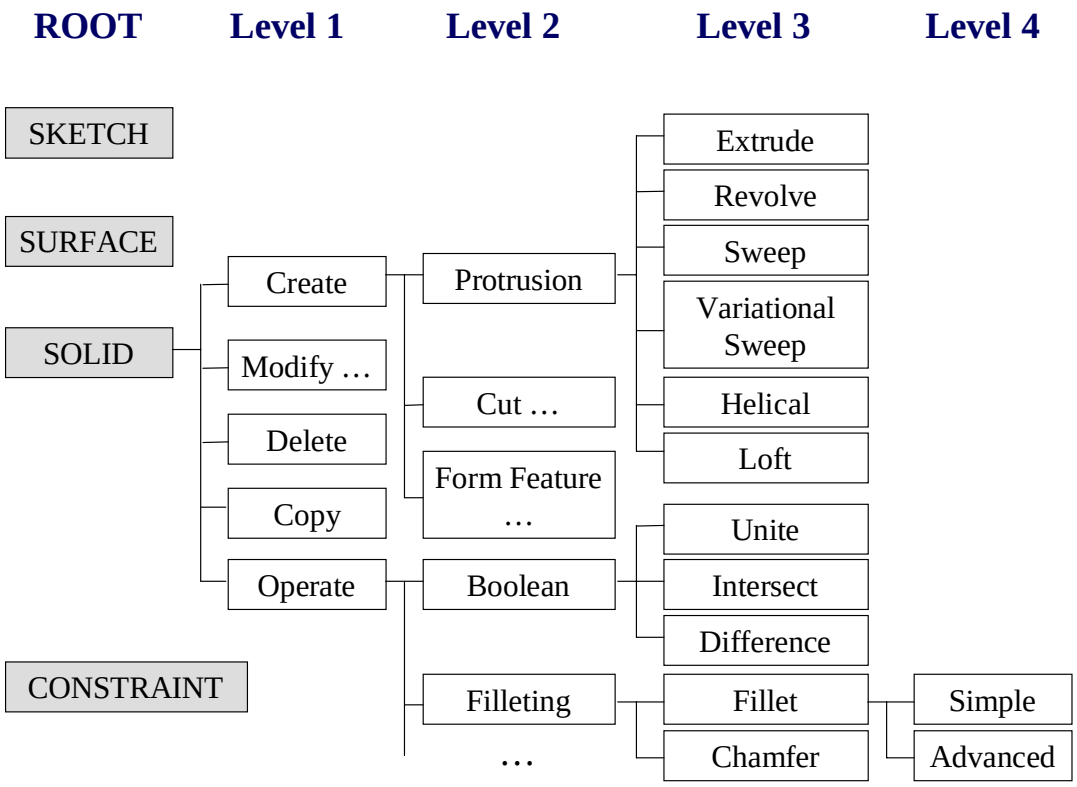


Figure 7 Classification of standard modeling commands

4.3 Draft Set of Standard Modeling Commands

The draft set of standard modeling commands, WD0 (Working Draft 0), was produced in Aug. 2001. A top-down approach has been used to define the set, which contains only 107 commands. The commands are grouped into four groups at the root level, eighteen groups of level 1, sixty groups of level 2, and fifty-five groups of level 3. The scope of the standard modeling commands is shown in Figure 8. It is a semantic common set of surveyed modeling commands. While it includes sketch and constraint commands, commands for assembly models are not addressed. The latest WD4, which is still under

refinement, is composed of 144 commands including arguments.

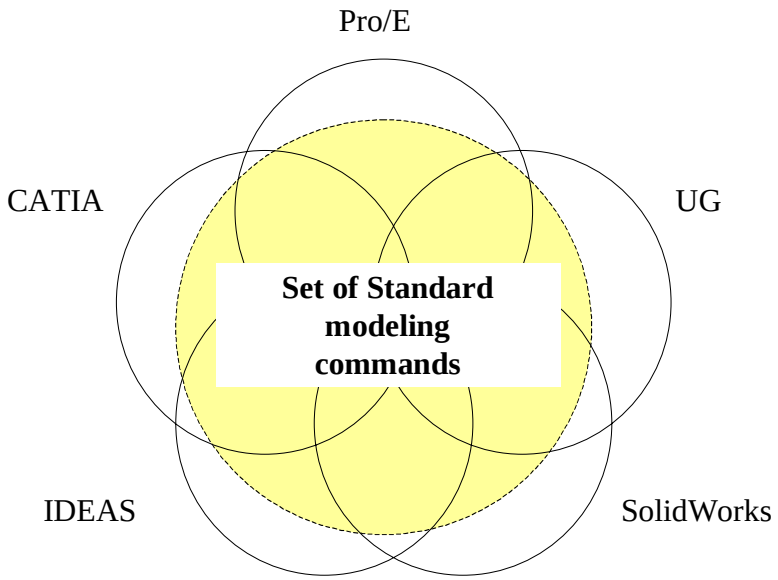


Figure 8 Scope of the standard modeling commands

5. IMPLEMENTATION AND TESTS

5.1 Revision of the Commands through Tests

According to the mapping between the standard modeling commands and modeling commands of commercial CAD systems, there have been several experiments of CAD model exchange, which are of aid in revising the draft set of standard modeling commands. Table 1 shows test classes for the experiments. Each conformance class has its own test model and a list of accumulative SOLID commands to be tested. The WD4 commands set reflects the test results up to CC4.

Conformance classes	Name of the test model	SOLID commands to be tested (accumulative)
CC 1	L Block	SOLID_Create_Protrusion_Extrude SOLID_Create_Cut_Extrude SOLID_Operate_Filleting_Fillet_Chamfer
CC 2	Y Model	SOLID_Create_Protrusion_Revolve SOLID_Create_Protrusion_Sweep
CC 3	Gas Spring	SOLID_Create_Cut_Revolve
CC 4	Linear Sensor	SOLID_Create_Cut_Sweep SOLID_Create_Protrusion_Variational_Sweep SOLID_Create_Feature_Hole_Linear SOLID_Create_Feature_Slot
CC 5	Engine Airfilter Housing	SOLID_Operate_Boolean_Union SOLID_Operate_Boolean_Intersect SOLID_Operate_Boolean_Difference SOLID_Operate_Pattern_Rectangular SOLID_Operate_Pattern_Circular
CC 6	Pneumatic Cylinder	SOLID_Create_Feature_Hole_Counterbored SOLID_Create_Feature_Hole_Countersunk SOLID_Create_Feature_Groove SOLID_Create_Feature_Pocket SOLID_Create_Feature_Pad

CC 7	Boat	SOLID_Operate_Transform_Move SOLID_Operate_Transform_Rotate SOLID_Operate_Transform_Mirror SOLID_Create_Protrusion_Loft SOLID_Create_Cut_Loft

Table 1 Test models for the standard modeling commands

5.2 Standard Modeling Commands

Standard modeling commands are defined in EXPRESS language [21] of STEP. For example, *SOLID_Create_Protrusion_Extrude* is defined as Table 2. It creates an extruded solid with *a profile_sketch* through an orthogonal linear path; where the *name* argument is the extrusion instance name, *profile_sketch* is the sketch name, *flip* is TRUE to reverse the extrusion direction, and *start_condition* is the condition of start face such as Blind, ThroughAll, or UpToFace. *start_depth* is the start depth of extrusion, *end_condition* is the condition of the end face such as Blind, ThroughAll, or UpToFace, and *end_depth* is the end depth of extrusion. Although it utilizes the EXPRESS syntax, it does not conform to the related *Parts* of STEP standard. Rather it reflects the common practices of commercial CAD systems.

```
ENTITY SOLID_Create_Protrusion_Extrude;
        name            :       STRING;
        profile_sketch  :       STRING;
        flip            :       BOOLEAN;
        start_condition :       end_type;
        start_depth     :       REAL;
        end_condition   :       end_type;
        end_depth       :       REAL;
END_ENTITY;

TYPE end_type = ENUMERATION OF (Blind, ThroughAll, ThroughNext,
UpToVertex, UpToSurface, OffsetFromSurface, MidPlane) ;
END_TYPE;
```

Table 2 Definition of a command in EXPRESS

5.3 Pilot Implementation

To exchange macro files which have parametric information, a pilot translator has been implemented and tested between CATIA and SolidWorks[2]. The translation result of the test model (CC2, Y Model) from SolidWorks to CATIA is shown in Figure 9. The feature tree inside the CATIA window confirms that the parametric information is properly transmitted from SolidWorks to CATIA. The right side of Figure 9 shows the parametrically changed model after a translation where the blend radius has been changed.

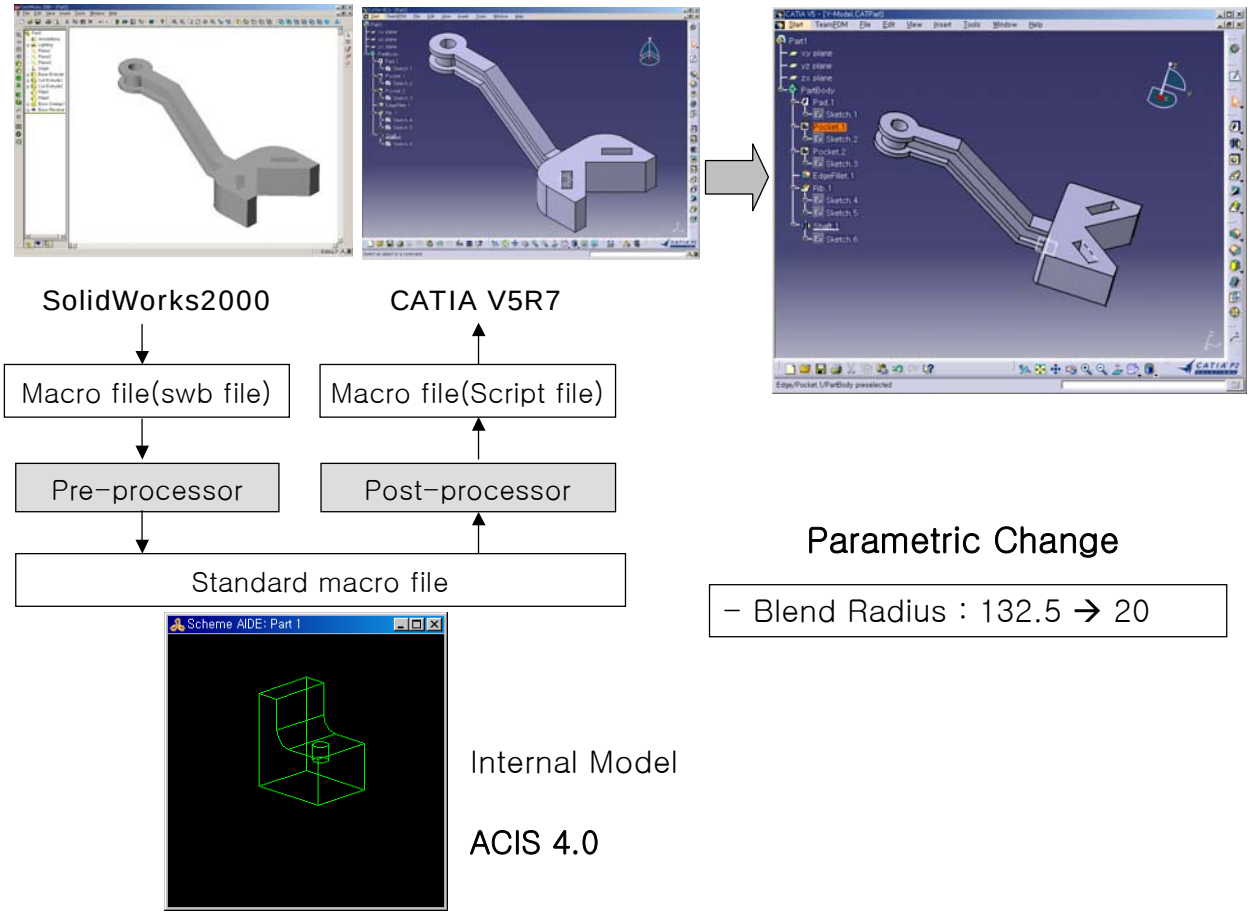


Figure 9 Experiments of CAD model exchange[2]

6. CONCLUSIONS

A set of modeling commands to support the history-based parametric approach is proposed to exchange parametric models among heterogeneous CAD systems. In this approach, CAD models can be exchanged in the form of macro files, which are sequences of high-level modeling commands. The macro-parametric approach has several advantages. It can implicitly transfer designer intents, because the macro file contains the design history. The macro file is easy to edit to generate a family of parts. As it is a pure procedural approach, the translation mechanism is simpler than that of SMCH. It is possible to develop a translator without close support from vendors because most commercial CAD systems provide macro files and user level commands that are open to the public. It is also suitable for network environments, because the macro file size is small.

The process of developing the standard modeling commands has been explained. Modeling commands of six commercial CAD systems have been analyzed. These modeling commands are classified into 4 groups at the root level, SKETCH, SOLID, SURFACE, and CONSTRAINT. The modeling commands are further classified into detail levels. In addition, mapping relationships between the standard modeling commands and modeling commands of commercial CAD systems are defined. After several experiments, the set of standard modeling commands has been refined. The fourth revision of WD4 is composed of 144 standard modeling commands.

ACKNOWLEDGEMENTS

This research has been partially supported by the Korea Maritime STEP project (NRL).

7. REFERENCES

[1] Pratt, Michael J., “Provision of an explicit constraints schema in the STEP standard”, In Strasser, W., Klein, R., Rau, R. (Editors), *Geometric Modeling: Theory and Practice*. Springer-Verlag, 1997.

[2] Choi, G., Mun, D., Han, S., “Exchange of CAD part models based on the macro-parametric approach”, International Journal of CAD/CAM (www.ijcc.org), 2(2):23-31, Feb. 2002

[3] Pratt, M.J., Anderson, B.D., “A shape modeling applications programming interface for the STEP standard”, Computer-Aided Design, 33 (2001), 531-543.

[4] Anderson, B., “ENGEN data model: a neutral model to capture design intent”, PROLAMAT98, 1998.

[5] ISO 10303-203, Product data representation and exchange: Application protocols: Configuration controlled design, 1994.

[6] Pratt, M. J., “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.

[7] 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.

[8] ISO/CD 10303-108, Product data representation and exchange: Integrated application resource: Parametrization and constraints for explicit geometric product models, ISO TC184/SC4/WG12 N940, 2001.

[9] ISO/NWI-CD 10303-55, Integrated generic resource: Procedural and hybrid representation, ISO TC184/SC4/SC4N1416, 2002.

[10] Greco, J., "Working magic with translation and healing", CADENCEweb, http://www.cadenceweb.com/magazine, October, 2002

[11] Anderson, B., "Implementor's guide - Solid model construction history", http://www.cax-if.org/documents/IMPl_GUIDE_May3.pdf, May 2002.

[12] SPANS (Supply-Chain Practices for Affordable Navy Systems) homepage, "CHAPS Project", http://www.spans.org/projects/chaps/description.html, 2002.

[13] ISO TC184/SC4/WG12 N1407, Proposal for a new a resource part for "Construction history features", 2002.

[14] ISO TC184/SC4/WG12 N1???, “Minutes of WG12 parametrics meeting from Seoul, KOREA”, 2002.

[15] ISO/CD 10303-109, Product data representation and exchange: Integrated application resource: Kinematic and geometric constraints for assembly models, ISO TC184/SC4/WG12 N1382, 2002.

[16] OMG, “CAD services specification”, http://cgi.omg.org/cgi-bin/doc?mfg/2001-06-03, Oct. 2001.

[17]Oh, Y., Han, S., Suh, H., “Mapping product structures between CAD and PDM systems using UML”, Computer-Aided Design, 33 (2001), 521-529.

[18] ISO TC184/SC4/WG12 N680, “On exchangeable construction history of geometric models in STEP”, 2001.

[19] Marcheix, D., Pierra, G., “A survey of the persistent naming problem”, 7th ACM Symposium on Solid Modeling and Applications, SM2002, 2002.

[20] Vergeest, J.S.M., Horvath, I., “Where interoperability ends”, Proc. of the 2001 Computers and Information in Engineering Conference, DETC'01/CIE-21233, ASME, New York, 2001.

[21] ISO TC184/SC4, ISO 10303-11 - Part 11: Industrial automation systems and integration – Product data representation and exchange – Implementation methods: The EXPRESS language reference manual, ISO, 1994.