

CAD 모델 교환을 위한 매크로 파라메트릭 정보의 XML 표현

양정삼[†] · 한순흥* · 김병철** · 박찬국***

(2003년 5월 19일 접수, 2003년 10월 20일 심사완료)

A Macro Parametric Data Representation for CAD Model Exchange using XML

Jeongsam Yang, Soonhung Han, Byungchul Kim, Chan-Cook Park

Key Words : CAD Model Exchange(CAD 모델 교환), Macro-Parametric(매크로 파라메트릭), Modeling Commands(모델링 명령어), XML(eXtensible Markup Language)**Abstract**

The macro-parametric approach, which is a method of CAD model exchange, has recently been proposed. CAD models can be exchanged in the form of a macro file, which is a sequence of modeling commands. As an event-driven commands set, the standard macro file can transfer design intents such as parameters, features and constraints. Moreover it is suitable for the network environment because the standard macro commands are open, explicit, and the data size is small. This paper introduces the concept of the macro-parametric method and proposes its representation using XML technology. Representing the macro-parametric data using XML allows managing vast amount of dynamic contents, Web-enabled distributed applications, and inherent characteristic of structure and validation.

1. 연구 배경

과거 10 여 년간 전세계 제조업체들은 그들이 생산하는 제품 모델을 디지털로 구축하기 위하여 1 조 억 달러에 달하는 비용을 투자하였다.⁽¹⁾ CAD 도면, 3 차원 솔리드 모델, 어셈블리, BOM (bill of material) 정보, 해석과 시뮬레이션 모델 등은 제조업체의 중요한 자산이 되었다. 그러나, 시대의 흐름은 C3PE (CAD, CAM, CAE, PDM, ERP)를 통합한 디지털 기업화(digital enterprise)를 요구함에 따라, 제조업체들은 기존에 구축된 제품 모델의 재 사용 문제에 있어서 어려움을 겪고 있다. 미국 자동차 시장에서 제품 모델 데이터의 교환에 따라 발생하는 문제를 해결하기 위해 1 년에 10 억 달러의 비

용을 지출하고 있다.⁽²⁾

CAD 모델을 교환하기 위해 다음과 같은 세가지 방법이 사용되고 있다.

첫 번째는, 전사적으로 단일 CAD 시스템을 도입해서 동일한 제품 모델 파일을 생산하는 것이다. 즉, 제품 개발에서 생산에 이르기 까지 CATIA 와 같은 한가지 범용 CAD 시스템을 전 과정에 사용하는 것이다. 그러나, 실제 부서간에 특화된 CAD 시스템을 활용하는 현실과, 특정 CAD 시스템에 회사 전체가 종속되는 문제점을 갖고 있다. 특히, 최근 분산 환경에서 다양한 조직이 참여하는 공급망 관리 (supply chain management)와, 분산 개발 환경을 요구하는 흐름에서는 궁극적인 해결책이 안 된다. 또한, 확장 기업(extended enterprise) 개념에서 단일 시스템이 어렵다.

두 번째 방법은, 서로 다른 CAD 시스템 간에 직접 번역기를 사용하는 방법으로, 국내 현업에서 일반적으로 사용하는 방법이다. 그러나, CAD 시스템의 핵심에 해당하는 CAD 형상 커널(kernel, geometry engine)이 CAD 시스템 마다 서로 다르기 때문에, 번역하는 과정에서 형상의 왜곡과 데이터 손실이 발생하기 때문에 100% 번역은 불가능하다. 또한, N 개의 CAD 시스템을 활용하는 기업에서는

[†] 책임저자, ㈜부품디비 기술연구소

E-mail : jsyang@partdb.com
TEL : (042)862-9225 FAX : (042)862-9224

* 회원, 한국과학기술원 기계공학과 shhan@kaist.ac.kr

** 한국과학기술원 기계공학과 mir7942@icad.kaist.ac.kr

*** 고등기술연구원 정보기술센터 parkck@iae.re.kr

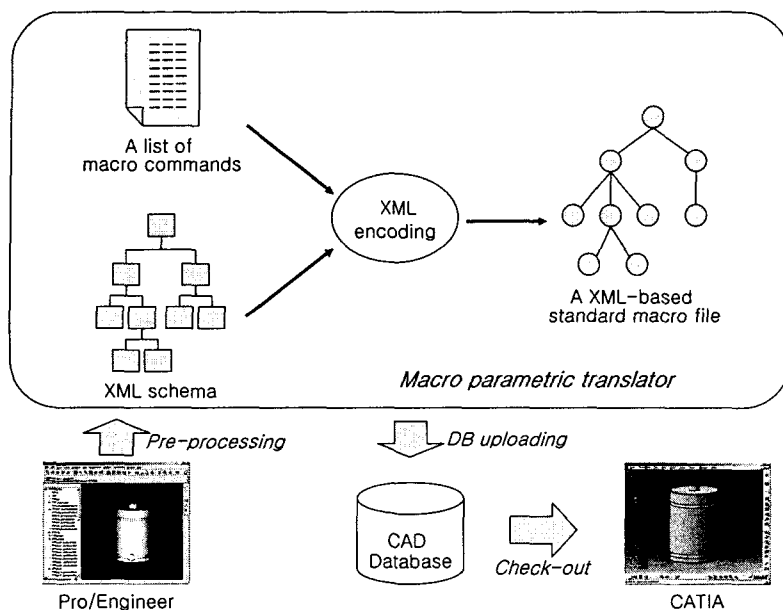


Fig. 1 An overview for XML-based macro-parametric representation

$N(N-1)/2$ 개의 번역기가 필요하기 때문에 추가적인 비용 지출이 요구된다.

세 번째는, 중립 CAD 포맷으로 변환하는 방법이다. 대표적인 중립 CAD 파일 포맷은, 국제표준(ISO 10303)인 STEP, 미국표준(ANSI)인 IGES, 독일표준인 VDA-FS, 그리고 업계 표준(de facto)인 DXF (Data eXchange File) 등이 있다. 중립 포맷의 공통적인 단점은, 설계자의 설계 의도가 포함된 파라메트릭 정보를 전달하지 못하고, CAD 모델의 외형 (pure boundary representation) 만을 전달한다는 데 있다. 파라메트릭 정보가 포함되지 않은 CAD 모델은 협업 (collaborative) 설계와 편집 (configuration) 설계 과정에서 설계 변경이 어렵고 형상의 오류를 초래 할 수 있다.

이와 같은 문제를 해결하기 위해, CAD 모델에 포함된 파라메트릭 정보를 전달할 수 있는, 설계 이력(design history) 기반의 매크로 파라메트릭 방법이 제안 되고, 연구가 진행 중이다.^(3-6, 16)

본 논문에서는, Fig. 1에서 보는 바와 같이, 매크로 파라메트릭 방법론에 따라, 표준 모델링 명령어 집합으로 번역된 CAD 모델 데이터를, XML 기술을 이용해서 표현하고, 서로 다른 CAD 시스템 사이에서 공유할 수 있는 방법을 제안한다. 또한 데이터베이스 연동을 통해 XML 과 표준 모델링 명령어를 기반으로 CAD 데이터베이스 구축의 예를 보여 준다.

2. 관련 연구

2.1 파라메트릭 정보의 교환

국제 표준인 STEP 에는, CAD 모델의 형상을 정의하기 위해서 Part 42 에 형상의 엔터티들이 정의되어 있다. 그러나 매개변수, 제약조건, 특징형상에 대한 표현을 제공하지 못하여, CAD 모델 교환 시 설계 의도를 전달하지 못하고 있다. CSG (constructive solid geometry)를 위한 엔터티들이 Part 42 에 제한적으로 정의되어 있지만, CSG 관련 엔터티들은 치수를 매개변수화(parameterization) 할 수 없다.⁽⁷⁾ 또한, 상용 CAD 시스템에서 제공하는 shelling 이나 drafting 과 같은, 설계 이력 기반의 모델링 기능에 대한 정의는 없다.

미국에서 수행된 ENGEN (Enabling Next GENERation mechanical design) 프로젝트는, STEP Part 42 를 기반으로 매개변수, 설계이력, 제약조건, 특징형상 정보를 표현하기 위해, 제품 데이터 모델인 EDM (ENGEN Data Model)을 제안하였다.⁽⁸⁾ 이 프로젝트는 서로 다른 CAD 시스템에서 모델 교환 실험을 하였지만, 2 차원 제약조건만을 표현할 수 있고, 암묵적으로 파라메트릭 모델을 표현하기 위해서 발생하는 중요한 문제인, 고유 명칭 (persistent naming) 문제를 다루지 않았다.

STEP 파라메트릭스 그룹의 Varra 와 Anderson 은 매개변수, 제약조건, 특징형상, 설계이력정보를 교환할 수 있는 SMCH (Solid Model Construction

History) 스키마를 제안하였다.⁽⁹⁾ 이 스키마는 매개 변수화 된 특징형상을 포함하는, 기하학적으로 구속된 솔리드와 설계 이력 기반 모델을 교환 할 수 있는, 암묵적(implicit) 엔티티와 작업의 표현에 대한 구조를 포함하고 있다.

2.2 데이터 표현을 위한 XML 의 적용

XML 은 Web 환경에서 구조화된 문서와 데이터를 교환하기 위한 표준 포맷이다. XML 은 기업과 기업간 (business-to-business), 응용프로그램과 응용 프로그램간 (application-to-application), 응용프로그램과 인간 상호간 (application-to-human)에 정보의 교환을 위해 여러 분야에 적용되고 있다.

Web 상에서 수학적 표현과 화학적 표현을 위해 MathML (Mathematical Markup Language)과 CML (Chemical Markup Language)이 제안이 되었고, 그 밖에, BSML (Bioinformatic Sequence Markup Language), AML (Astronomical Markup Language) 등이 제안되었다.⁽¹⁰⁾

CAD 정보의 교환을 위한 XML 의 적용은 Rezayat 에 의해 연구 되었다.⁽¹¹⁾ Rezayat 는 CAD 정보를 공유하기 위해 KBPD (knowledge-based product development) 라는 지식 베이스에, CAD 형상 정보를 XML 형태로 변환하여 사용하였다. 이를 위해, CADML (CAD Markup Language) 라는 DTD 형태의 스키마를 제안하였다. 이 스키마는 제한적인 B-Rep 형태의 CAD 형상을 표현하기 위한 구조를 가지고 있다.

ISO/TC184/SC4/WG (working group) 11 에서는 EXPRESS 를 사용해서 정의된 STEP 스키마들을 XML 로 표현하는 연구가 진행되고 있다.⁽¹³⁾ EXPRESS 스키마로 정의된 STEP 구조는 명확하고 직관적인 정보 시스템을 제공하지만, 상세한

규격에 있어서 구현의 한계가 있다.

Fig. 2 에서 보는 바와 같이, EXPRESS 스키마에서 지식 기반의 제한 조건을 표현하기 위해 WHERE 만을 사용할 수 있지만, XML 스키마는 다양한 형태의 제한 조건을 표현할 수 있는 연산자를 제공한다. 최근, WG 11 에서는 Part 21 (product data representation)의 EXPRESS 스키마에 대응되는 XML markup 선언자를 Part 28 에 정의하였다.

3. 매크로 파라메트릭 방법론

3.1 설계 이력 기반 파라메트릭

매크로 파라메트릭 방법론은 한국과학기술원과 (주)부품디비에서 진행중인 연구 내용으로, 설계자가 모델링 하는 과정에서 사용한 명령어의 히스토리에 기초하여 설계 의도를 교환하는 방법이다.^(3-6,16) 이 방법은 트랜잭션 로그 (transaction log) 파일을 이용한 데이터베이스의 복구에서 힌트를 얻었다. 설계자가 모델링 하는 과정에서 사용한 명령어들은 매크로 파일로 기록된다. 매크로 파일에는 특징 형상, 매개 변수, 구속 조건 등에 대한 정보가, 명령어 형태로 설계 프로세스와 함께 들어 있다. 이 명령어를 다른 CAD 시스템에서 사용하는 명령어들로 매핑시켜, 그 시스템에서 사용할 수 있는 매크로 파일로 변환을 시켜 주면, 설계 의도가 들어 있는 파라메트릭 모델을 재생할 수 있다.

Fig. 3 은 매크로 파라메트릭 방법론을 이용한 데이터 교환 방법을 보여준다. 매크로 파라메트릭은 크게, 상용 CAD 시스템의 명령어 집합과 표준 모델링 명령어 집합 간에 스키마 매핑이 이루어지는 부분과, 상용 CAD 시스템의 매크로 파일과 표준 매크로 파일의 변환이 이루어지는 데이터 변환 부분이 있다.

모델링 명령어 집합은 특정 상용 CAD 시스템에서 사용되는 모델링 명령어의 집합이고, 표준 매크로 명령어 집합은, 각종 상용 CAD 시스템의

SCHEMA example_part_schema;

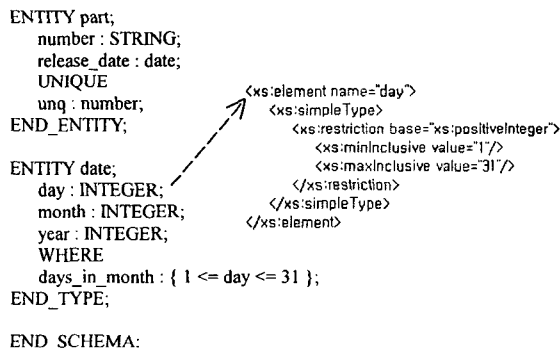


Fig. 2 Examples of EXPRESS schema and XML schema

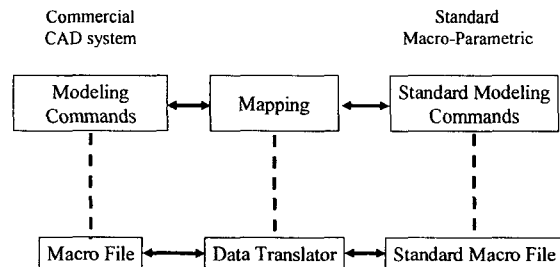


Fig. 3 Concept of the macro-parametric mapping⁽⁵⁾

모델링 명령어 집합을 표준화하여 만든 모델링 명령어의 집합이다. 매크로 파일은 모델링 과정 동안 CAD 시스템에서 사용한 명령들을 모아 놓은 것으로, CAD 시스템의 모델링 명령어 집합의 부분 집합이다. 매크로 파일은 모델링 이력만을 교환함으로써, 어느 CAD 시스템에서나 같은 모델을 재생할 수 있다.

일반 상용 CAD 시스템들에서, 모델링 명령어 목록과 API 목록들의 공통분모를 찾아 표준화하여, 표준 모델링 명령어 집합을 만들었다. 각 CAD 시스템들의 명령어 집합과의 매핑 관계를 정의한 후, 사용자가 각 CAD 시스템에서 사용한 명령어를 모은, 매크로 파일을 표준 명령어 매크로로 변환하여, 각 CAD 시스템들 사이에서 설계 정보를 교환한다.

3.2 표준 모델링 명령어 집합

모델링 명령어를 표준화 하기 위해 5 개의 상용 CAD 시스템 (CATIA, Unigraphics, SolidWorks, Pro/Engineer, IDEAS)에서 사용하고 있는 명령어들의 공통 분모를 추출하여, 144 개 (스케치 명령어 57, 솔리드 모델링 명령어 40 개, 곡면 모델링 명령어 23 개, 구속 조건 명령어 24 개)의 명령어 집합으로 나누고, EXPRESS 를 사용하여 정의하였다 [부록 A, B 참조]. 표준 모델링 명령어 집합은 ISO TC184/SC4/WG12 에 NWI (new work item)로 제안 하였다.

Fig. 4 는 표준 모델링 명령어 집합을 4 개의 단계로 분류한 그림으로서, 2 차원 스케치 명령어 그룹(SKETCH), 곡면 모델링 명령어 그룹(SURFACE), 솔리드 명령어 그룹(SOLID), 구속 조건 명령어 그룹(CONSTRAINT)으로 분류된다.

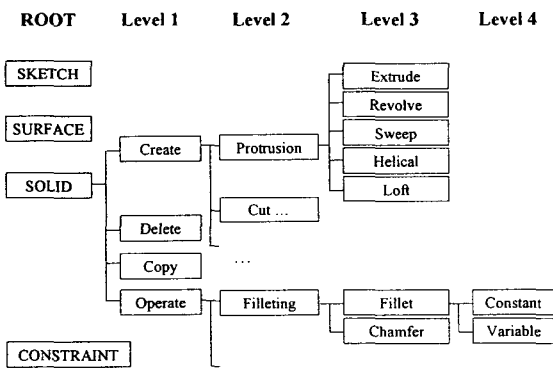


Fig. 4 Classification tree of standard modeling commands

각각의 그룹들은 생성 명령어(Create), 삭제 명령어>Delete), 연산 명령어(Operator) 등과 같이 상세하게 분류되고, 그 하위 분류는 돌출 명령어(Protrusion), 라운딩 명령어(Filleting) 등 구체적인 작업으로 구성된다.

한 개의 표준 모델링 명령어의 이름은, 분류 명칭의 연속으로 만든다. 예를 들면, 직선 돌출에 대한 명령어는 "SOLID_Create_Protrusion_Extrude" 이다. Fig. 5 는 XML 스키마를 사용하여 명령어 "SOLID_Create_Protrusion_Extrude"를 표현 한 것이다.

여기서, result_object_name 은 명령어를 수행해서 생기는 돌출 특징 형상의 이름, profile_sketch 는 돌출 형상의 단면, flip 은 돌출 방향에 대한 인자, start_condition, end_condition 은 돌출의 시작과 끝나는 부분의 상태, start_depth, end_depth 는 돌출의 길이를 나타낸다. 이 명령을 수행한 결과는 Fig. 6 과 같다.

```
<xs:element name="SOLID_Create_Protrusion_Extrude">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="result_object_name"/>
      <xs:element name="profile_sketch" type="xs:string"/>
      <xs:element name="flip" type="xs:boolean"/>
      <xs:element name="start_condition" type="end_type"/>
      <xs:element name="start_depth" type="length_measure"/>
      <xs:element name="end_condition" type="end_type"/>
      <xs:element name="end_depth" type="length_measure"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:simpleType name="length_measure">
  <xs:restriction base="xs:double"/>
</xs:simpleType>
<xs:simpleType name="end_type">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Blind"/>
    <xs:enumeration value="ThroughAll"/>
    <xs:enumeration value="ThroughNext"/>
    <xs:enumeration value="UpToVertex"/>
    <xs:enumeration value="UpToSurface"/>
    <xs:enumeration value="OffsetFromSurface"/>
    <xs:enumeration value="MidPlane"/>
  </xs:restriction>
</xs:simpleType>
```

Fig. 5 XML schema representation of SOLID_Create_Protrusion_Extrude

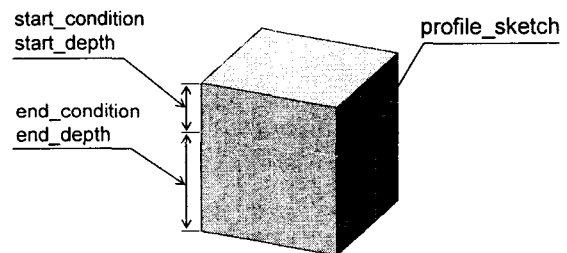


Fig. 6 A generated shape from the command SOLID_Create_Protrusion_Extrude

4. XML 을 이용한 명령어 정의와 실험

4.1 XML 을 이용한 정보의 표현

매크로 파라메트릭 방법론에 따라 정의된 표준 모델링 명령어를, XML 기술을 사용하여 표현하였다. 단순한 텍스트 포맷(flat file format)을 사용하지 않고 XML 을 사용하는 이유는, 부가적인 정보 없이 의미론적(semantics)으로 구조화된 데이터를 표현할 수 있기 때문이다. 즉, 데이터를 저장하는 내용(content) 부분과, 설계 이력과 같은 동적이고 확장성이 제공되는, 구조(structure) 부분이 분리 되어 있다. 또한, Web 기반의 분산 환경하에서 다양한 응용 프로그램에 적용이 가능하기 때문이다.

XML 을 이용해서 정보를 표현하는 방법은 *자료 중심의 XML (Data-centric use of XML)* 표현 방식과 *문서 중심의 XML (Document-centric use of XML)* 표현 방식으로 구분된다.⁽¹²⁾ 데이터 중심의 XML 표현 방식은 매우 정적(static)인 구조로서, 내용(content)과 구조(structure)가 혼재 되어 있는 정도가 낮고, 형제들 간의 순서가 중요하지 않다. 주로 전자상거래용 문서에 적용되는 방식으로, 엘리먼트를 저장, 조작, 변경하는데 중요한 의미를 가진다.

문서 중심의 XML 표현 방식은 비 정형적인 구조로서, 형제 엘리먼트 상호간의 순서가 중요하다. Fig. 4 와 같이 상·하의 명령어들이 종속적으로 연결되는 설계 이력 정보를 표현하기 위해서, 문서 중심의 XML 표현 방식이 필요하다. 이 방식을 적용하여 Fig. 7 에서 보는 바와 같이 표준 매크로 명령어를 표현하는 XML 스키마를 설계하였다.

4.2 XML 스키마의 설계

Table 1 에서 보는 바와 같이 표준 모델링 명령어는 모델링 과정에서 명령어 사용 빈도가 높은 Core commands (CC)와 사용 빈도가 낮은 Non-core commands (NCC) 분류하였다 [부록 A, B 참조].

표준 모델링 명령어를 위해 설계된 XML 스키마는 3 개의 레이어로 구성된다.

첫 번째 레이어는, Fig. 7 에서 보는 바와 같이, 루트

Table 1 Grouping of standard modeling commands

	Num. of CC	Num. of NCC	Sum
Sketch commands	30	27	57
Surface commands	17	6	23
Solid commands	22	18	40
Constraint commands	21	3	24
	90	54	144

(root) 엘리먼트(MACRO_PARAMETRICS)를 포함하여 스케치 명령어 엘리먼트(SKETCH_COMMANDS), 솔리드 명령어 엘리먼트(SOLID_COMMANDS), 곡면 명령어 (SURFACE_COMMANDS) 그리고 구속조건 엘리먼트(CONSTRAINT_COMMANDS)로 구성되었다.

두 번째 레이어는 실제로 수행되는 명령어들에 대한 엘리먼트들로 이루어 졌다. 스케치 명령어는 30 개의 엘리먼트, 솔리드 명령어는 22 개의 엘리먼트, 곡면 명령어는 17 개의 엘리먼트, 구속조건에 대한 명령어는 21 개의 엘리먼트로 구성되었다. 이 명령어들은 선택적으로 XML 파일 내에 기록된다.

세 번째 레이어는 각 명령어에 대한 매개변수를 정의한 엘리먼트들로 구성되었다. 이들 엘리먼트는 명령어의 매개변수에 따른 데이터 타입을 정의해서, XML 의 적합성(validation) 검증에 적용된다.

Fig. 7 은 스케치 엘리먼트와 구속조건 엘리먼트의 하위 노드들은 collapse 하고, 솔리드 엘리먼트의 하위 노드들에 대해서만 펼쳐(explode) 놓은 XML 스키마의 구조를 보여 주고 있다.

Fig. 7 에서 락선(dashed line)으로 둘러싸인 "SOLID_Create_Protrusion_Revolve" 명령어는, Fig. 8 같이 스케치를 회전해서 솔리드를 생성하는 명령어이다.

"SOLID_Create_Protrusion_Revolve" 명령어를 정의한 XML 스키마 구조는 Fig. 7 과 Fig. 9 에서 보는 바와 같이, 5 개의 하위 엘리먼트로 구성되어 있다. profile_sketch 엘리먼트는 회전 단면 형상과

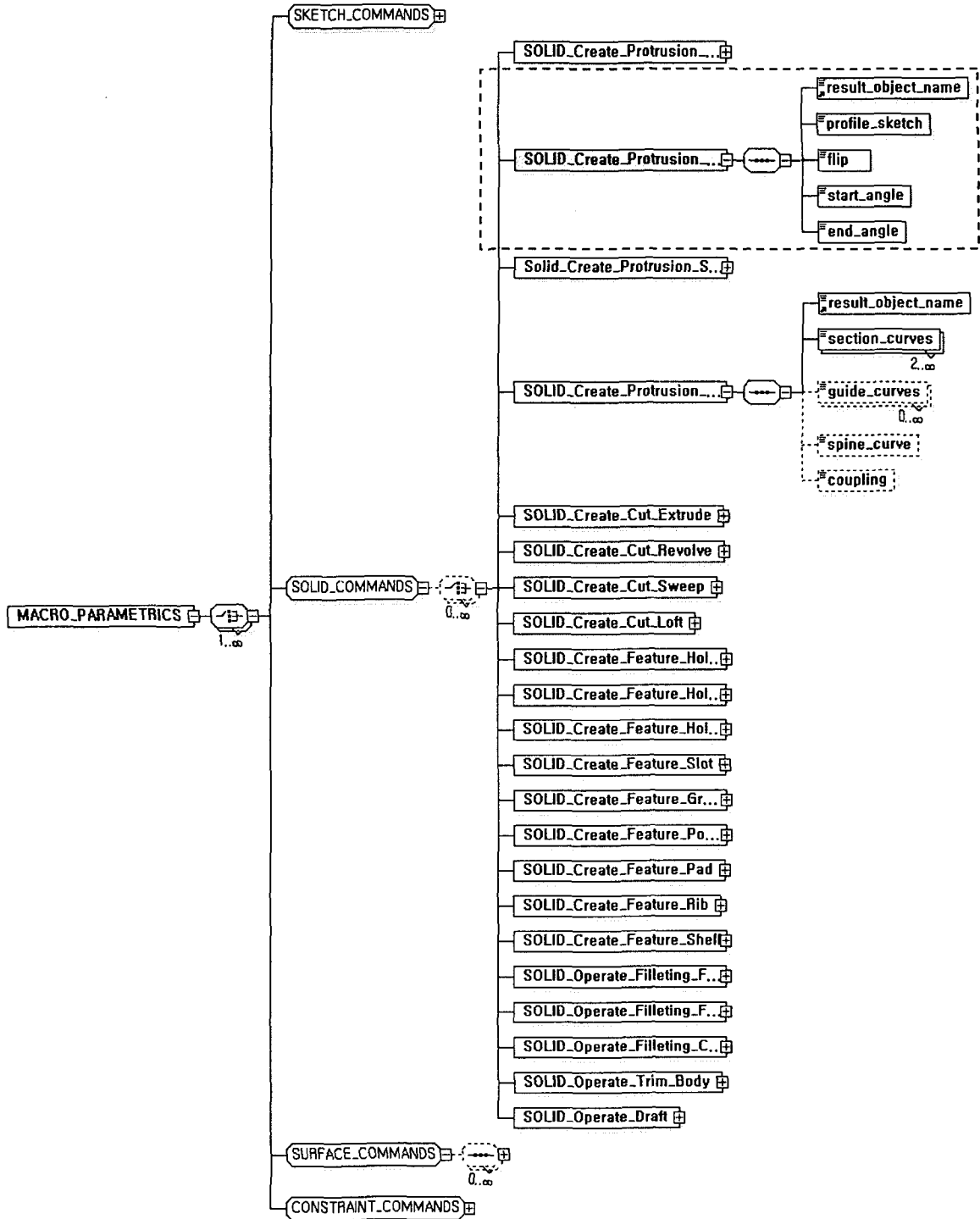


Fig. 7 The XML schema structure designed for the standard modeling commands

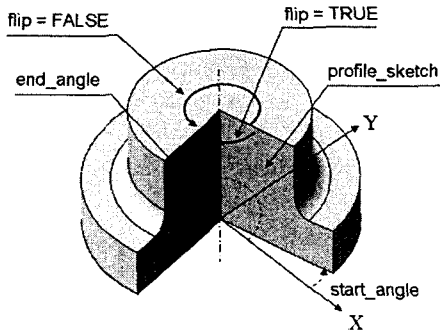


Fig. 8 Parameters of SOLID_Create_Protrusion_Revolve

```
<xs:element name="SOLID_Create_Protrusion_Revolve">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="result_object_name"/>
      <xs:element name="profile_sketch" type="STRING"/>
      <xs:element name="flip" type="xs:boolean"/>
      <xs:element name="start_angle" type="plane_angle_measure"/>
      <xs:element name="end_angle" type="plane_angle_measure"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Fig. 9 XML schema of SOLID_Create_Protrusion_Revolve

함께, 회전축을 포함하고 있는 스케치 평면을 나타낸다. flip 은 부울(boolean) 타입의 회전 방향을 나타내는 엘리먼트이고, start_angle 은 회전의 시작 각을, end_angle 은 회전의 끝 각을 나타내는 double 타입의 엘리먼트가 된다. 마지막으로 STRING 타입의 result_object_name 엘리먼트는, 회전으로 생성된 오브젝트의 이름을 나타낸다.

4.3 교환 실험

XML 스키마의 적용 실험은, 기존에 연구 개발된 번역기⁽⁶⁾와, 표준 모델링 명령어를 기반으로 하는 CAD 모델러 MirCAD⁽⁶⁾를 사용하였고, 데이터베이스와 연동을 위해 MS-Access 를 사용하였다.

Fig. 10 은 CATIA V5R9 에서 모델링 된 Y-model 을, 매크로 번역기를 통해 번역하고, 표준 매크로 명령어를, 본 논문에서 설계된 XML 스키마의 구조에 따라, XML 파일로 생성하는 내용을 보여 준다. 실험에 사용된 Y-model 은 54 개의 매크로 명령어가 사용 되었다. Fig. 10 에서 보여주는 XML 파일은, 54 개의 매크로 명령어 가운데 4 개의 명령어에 대한 내용을 보여준다.

“SELECT_Reference_Plane” 명령어는 2 차원 스케치 평면을 설정하기 위해 사용되었다. “SKETCH_Create_2D_Line_2Points”는 스케치 평면 상에서 2 개의 점으로 이루어지는 직선을 정의한

다. “SOLID_Create_Protrusion_Sweep”는 스케치를 임의의 경로에 따라 돌출 시키는 명령어이고, “SOLID_Create_Cut_Extrude”는 단면을 크게 돌출 시켜서 잘라내는 명령어이다.

번역된 명령어들과 매개변수들은, 기존 형상 모델 관점에서 개발된 CAD 데이터베이스 보다, 정형화 된 형태의 데이터베이스 구축을 위해서, MS-Access 를 사용해서 XML 스키마 기반의 데이터베이스 저장 실험을 하였다. 데이터베이스에 저장된 내용은 Fig. 10 과 같이 데이터베이스 테이블 형태로 확인할 수 있다.

XML 파일의 헤더(header)에는 XML 스키마 문서가 정의 되어 있다. 이를 통해 이 XML 문서가 검증(validation)된 문서임을 알 수 있다. 또한 번역된 XML 결과 파일을 자체 개발된 MirCAD 와, 4 개의 상용 CAD 시스템 (Pro/Engineer, CATIA V5, SolidWorks)에서 양방향으로 번역할 수 있다.

5. 결론

EXPRESS 로 표현된 데이터 포맷만으로도 유용한 가치가 있지만, XML 과 같은 의미론적으로 구조화된 문서를 포함하는 표준은, 응용프로그램 개발과 인터넷 데이터 교환에 있어서 강력한 기능을 제공한다.

본 논문에서는 매크로 파라메트릭 방법론에 따라 정의된 표준 모델링 명령어를, 구조적인 형태로 표현하기 위해 XML 기술을 적용 하였다. 이를 검증하기 위해 4 개의 상용 CAD 시스템과, 자체 개발된 MirCAD 사이에 CAD 모델을 번역/교환 실험을 하였다. 또한, 데이터베이스 MS-Access 와 연동을 통해, CAD 데이터를 데이터베이스로 구축하는 시범 시스템을 만들었다.

매크로 파라메트릭 데이터를 XML 을 이용해서 표현하는 것은, 다음과 같은 장점을 가지고 있다.

- 표준 매크로 명령어를 기록하는 내용(content) 과, 논리적인 골격을 이루는 구조(structure)를 분리해서, 동적인 형태의 데이터를 표현할 수 있다.
- Web 환경에서 TCP/IP 를 통해, CAD 모델이 쉽게 전송될 수 있고, 특정 응용프로그램에 구속되지 않는다.
- 데이터베이스 스키마와 연동시켜 CAD 데이터베이스를 구축할 수 있다.
- 제 3 의 번역기 개발자가 매크로 번역기를 개발할 경우, 개방된 형태의 자료 구조 (open data architecture)를 파악할 수 있다.
- 다양한 XML 관련 응용프로그램에서 활용할

수 있다.
향후 진행될 내용은, 현재 완료되지 않은 곡면

모델링 명령어(SURFACE_COMMANDS)와 Non-core commands 에 대한 내용을 보완 해야 한다.

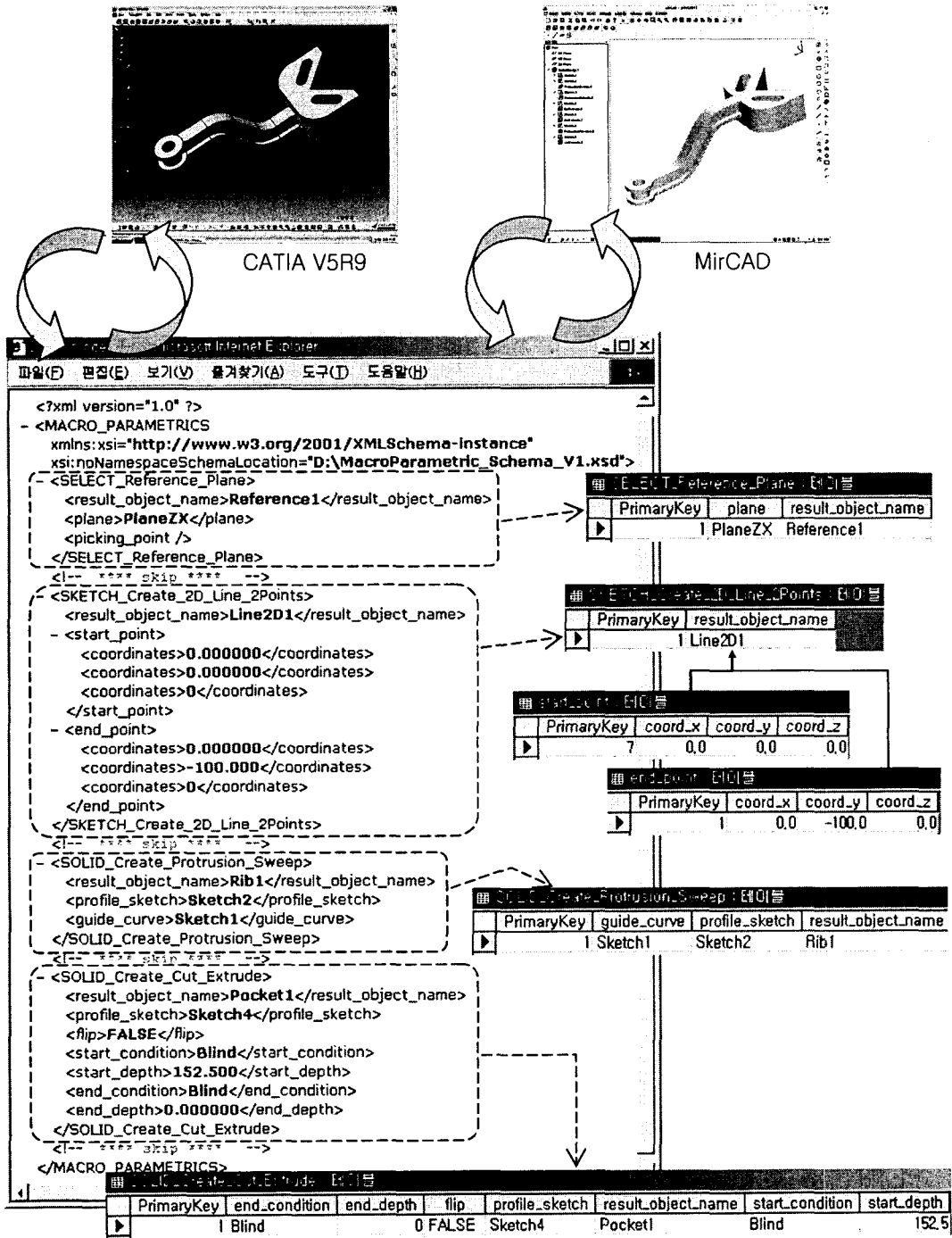


Fig. 10 Experiment of CAD model exchange using an XML file

참고문헌

- (1) Crabb, H.C., 1998, *The Virtual Engineer*, 21st Century Product Development, SME/ASME Press, New York.
- (2) Gregory, T., 1999, *Interoperability Cost Analysis of the U.S. Automotive Supply Chain – Final report*, RTI Project Number 7007-03, Research Triangle Institute.
- (3) Mun, D.H. and Han, S., 2001, "Exchange of CAD Models Using Macro Parametric Approach (in Korean)," *Transactions of the Society of CAD/CAM Engineers*, Vol. 6, No. 4, pp. 254~262.
- (4) Mun, D., Kim, B. and Han, S., 2002, "A Hybrid Parametric Translator Using the Feature tree and the Macro File (in Korean)," *Transactions of the Society of CAD/CAM Engineers*, Vol. 7, No. 4, pp. 240~247.
- (5) Choi, K.H., Mun, D.H. and Han, S., 2002, "Exchange of CAD Part Models Based on the Macro-parametric Approach," *International Journal of CAD/CAM* (www.ijcc.org), Vol. 2(2), pp. 23~31.
- (6) Kim, B.C., 2002, "Verification of the Standard Modeling Commands Set by Developing a Geometric Modeler (in Korean)," Master's thesis, Korea Advanced Institute of Sci. and Tech.
- (7) Pratt, M.J., 1998, "Extension of the Standard ISO10303 (STEP) for the Exchange of Parametric and Variational CAD Models," PROLAMAT98.
- (8) Anderson, B., 1998, "ENGEN Data Model: a Neutral Model to Capture Design Intent," PROLAMAT98.
- (9) Barra, R. and Anderson, B., 2000, "Draft Implementor's Guide Solid Model Construction History," DRAFT Minutes of WG12 Parametrics Meeting – October 2000.
- (10) Kerer, C., Kirda, E. and Kruegel, C., 2002, "XGuide – A Practical Guide to XML-Based Web Engineering," LNCS2376, pp. 104~117.
- (11) Rezayat, M., 2002, "Knowledge-based Product Development Using XML and KCs," *Computer Aided Design*, Vol. 32, pp. 299~309.
- (12) Nambiar, U., Lacroix, Z., Bressan, S., Lee, M.L. and Li, Y., 2002, "Current Approaches to XML Management," *IEEE Internet computing* July-August.
- (13) ISO/WD 10303-28e2, ISO TC184/SC4/WG11 N202, 2002, *Product Data Representation and Exchange: Implementation Methods: XML Schema Governed Representation of EXPRESS Schema Governed Data*, http://www.tc184-sc4.org/SC4_Open/SC4_and_Working_Groups/WG11/.
- (14) Kim, J. and Han, S., 2002, "Manipulating Geometry Instances in an STEP-based OODB from commercial CAD systems (in Korean)," *Journal of the Korean Institute of Industrial Engineers*, Vol. 28, No. 4, pp. 436-443.
- (15) Kim, J., Han, S. and Kim, Y., 2003, "An Evaluation of Access Performance of STEP-Based CAD Database (in Korean)," *Conference Proceeding of Society of Naval Architects of Korea*, Feb., 2003, Hanjin Heavy Industry.
- (16) Mun, D., Han, S. and Oh, Y.C., 2003, "A Set of Standard Modeling Commands for the History-Based Parametric Approach," *Computer Aided Design*, Vol. 35, pp. 1171~1179.
- (17) Jang, S.-H. and Kim C.-W., 2001, "An Intelligent CAD System for Development of Controllers of Active Magnetic Bearings," *KSME International Journal*, Vol. 15, No. 8, pp. 1108~1118.

부록

A. Core Commands List

	Commands list	
Sketch commands	SKETCH_Create_2D_Line_2Points SKETCH_Create_2D_Polyline SKETCH_Create_2D_Centerline SKETCH_Create_2D_Rectangle SKETCH_Create_2D_Point SKETCH_Create_2D_Arc_Concentric SKETCH_Create_2D_Arc_3Tangents SKETCH_Create_2D_Arc_CenterEnds SKETCH_Create_2D_Arc_3Points SKETCH_Create_2D_Circle_CenterPoint SKETCH_Create_2D_Circle_Concentric SKETCH_Create_2D_Circle_3Tangents SKETCH_Create_2D_Circle_3Points SKETCH_Create_2D_Arc_Angles SKETCH_Create_2D_Ellipse_3Points	SKETCH_Create_2D_Ellipse_CenterPoint SKETCH_Create_2D_Spline SKETCH_Create_2D_Conic SKETCH_Create_2D_Text SKETCH_Create_2D_AxisPoint SKETCH_Create_SketchName SKETCH_Operate_Transform_Move SKETCH_Operate_Transform_Rotate SKETCH_Operate_Transform_Mirror SKETCH_Operate_Transform_Scale SKETCH_Operate_Offset SKETCH_Operate_Fillet SKETCH_Operate_Chamfer SKETCH_Open SKETCH_Close
Surface commands	SURFACE_Create_ThroughPointFromPoles SURFACE_Create_FromPointCloud SURFACE_Create_Sweep SURFACE_Create_Section SURFACE_Create_Offset SURFACE_Create_Offset_DraftOffset SURFACE_Create_Extrude SURFACE_Create_Revolve SURFACE_Create_Plane_3Points	SURFACE_Operate_Split SURFACE_Operate_Merge SURFACE_Operate_MidSurface SURFACE_Operate_Trim SURFACE_Operate_Extension SURFACE_Operate_Filleting_Fillet_Constant SURFACE_Operate_Filleting_Fillet_Variable SURFACE_Operate_Filleting_Chamfer
Solid commands	SOLID_Create_Protrusion_Extrude SOLID_Create_Protrusion_Revolve SOLID_Create_Protrusion_Sweep SOLID_Create_Protrusion_Loft SOLID_Create_Cut_Extrude SOLID_Create_Cut_Revolve SOLID_Create_Cut_Sweep SOLID_Create_Cut_Loft SOLID_Create_Feature_Hole_Linear SOLID_Create_Feature_Hole_Counterbore SOLID_Create_Feature_Hole_Countersunk	SOLID_Create_Feature_Slot SOLID_Create_Feature_Groove SOLID_Create_Feature_Pocket SOLID_Create_Feature_Pad SOLID_Create_Feature_Rib SOLID_Create_Feature_Shell SOLID_Operate_Filleting_Fillet_Constant SOLID_Operate_Filleting_Fillet_Variable SOLID_Operate_Filleting_Chamfer SOLID_Operate_Trim_Body SOLID_Operate_Draft
Constraint commands	CONSTRAINTS_Create_Dimension_Horizontal CONSTRAINTS_Create_Dimension_Vertical CONSTRAINTS_Create_Dimension_Arbitrary CONSTRAINTS_Create_Dimension_Circular CONSTRAINTS_Create_3DReference_CoordSys CONSTRAINTS_Create_Constraint_Perpendicular CONSTRAINTS_Create_Constraint_Tangent CONSTRAINTS_Create_Constraint_Coincident_SamePoints CONSTRAINTS_Create_Constraint_Coincident_Collinear CONSTRAINTS_Create_3DReference_OffsetPlanes	CONSTRAINTS_Create_Constraint_Concentric CONSTRAINTS_Create_Constraint_Horizontal CONSTRAINTS_Create_Constraint_Vertical CONSTRAINTS_Create_Constraint_Symmetric CONSTRAINTS_Create_3DReference_Plane CONSTRAINTS_Create_3DReference_Axis CONSTRAINTS_Create_3DReference_Curve CONSTRAINTS_Create_Constraint_Parallel CONSTRAINTS_Create_3DReference_Point SELECT_Reference_Plane SELECT_Object

B. Non-Core Commands List

	Commands list	
Sketch commands	SKETCH_Create_3D_Line SKETCH_Create_3D_Point SKETCH_Create_3D_Arc_3Points SKETCH_Create_3D_Circle SKETCH_Create_3D_Ellipse SKETCH_Create_3D_Curve_IntersectionCurve SKETCH_Create_3D_Curve_ProjectionCurve SKETCH_Create_3D_Curve_Spline SKETCH_Create_3D_Spline SKETCH_Create_3D_PointSet SKETCH_Create_3D_Helix SKETCH_Operate_Pattern_Rectangular SKETCH_Operate_Pattern_Circular SKETCH_Operate_Pattern_UserDefined	SKETCH_Operate_Intersect SKETCH_Operate_Divide SKETCH_Operate_Merge SKETCH_Delete SKETCH_Copy SKETCH_Reference_Plane_ReferencePlane SKETCH_Reference_Axis SKETCH_Reference_CoordSys SKETCH_Reference_Curve SKETCH_Reference_Line SKETCH_Reference_Point SKETCH_Reference_Section_Attach SKETCH_Reference_Section_CrossSection
Surface commands	SURFACE_Operate_Transform_Move SURFACE_Operate_Transform_Rotate SURFACE_Operate_Transform_Mirror	SURFACE_Operate_Transform_Scale SURFACE_Delete SURFACE_Copy
Solid commands	SOLID_Create_Protrusion_Helical SOLID_Create_Cut_Helical SOLID_Create_Feature_Pipe SOLID_Create_Feature_Thread SOLID_Create_Feature_UserDefined SOLID_Operate_Boolean_Union SOLID_Operate_Boolean_Intersect SOLID_Operate_Boolean_Difference SOLID_Operate_Pattern_Rectangular	SOLID_Operate_Pattern_Circular SOLID_Operate_Transform_Move SOLID_Operate_Transform_Rotate SOLID_Operate_Transform_Mirror SOLID_Operate_Transform_Scale SOLID_Delete_Object SOLID_Delete_Pattern SOLID_Delete_Parameter SOLID_Copy
Constraint commands	CONSTRAINTS_Create_Equation CONSTRAINTS_Delete	CONSTRAINTS_Copy