

# CAD 시스템간의 형상정보 교환을 위한 XML 이용에 관한 연구

박승현<sup>+</sup>, 최의성<sup>+</sup>, 정태형<sup>\*</sup>

(논문접수일 2003. 5. 28, 심사완료일 2003. 7. 21)

## The Exchange of Feature Data Among CAD Systems Using XML

Seung Hyun Park<sup>+</sup>, Eui Sung Choi<sup>+</sup>, Tae Hyong Chong<sup>\*</sup>

### Abstract

The exchange of model design data among heterogeneous CAD systems is very difficult because each CAD system has different data structures suitable for its own functions. STEP represents product information in a common computer-interpretable form that is required to remain complete and consistent when the product information is needed to be exchanged among different computer systems. However, STEP has complex architecture to represent point, line, curve and vectors of element. Moreover it can't represent geometry data of feature based models. In this study, a structure of XML document that represents geometry data of feature based models as neutral format has been developed. To use the developed XML document, a converter also has been developed to exchange modules so that it can exchange feature based data models among heterogeneous CAD systems. Developed XML document and Converter have been applied to commercial CAD systems.

**Key Words** : CAD(Computer Aided Design), XML(eXtensible Markup Language), DOM(Document Object Model), Feature, STEP (Standard for The Exchange of Product model data)

## 1. 서론

최근 많은 기업들은 제품 개발 기간과 비용을 줄이고 제품의 품질을 개선하기 위해 CAD 시스템에서 솔리드 모델을 이용한 제품설계를 하고 있다. 이러한 제품은 보통 수십

개에서 수백 개의 부품들로 이루어져 있다. 따라서 여러 명의 설계자가 참여하고 있으며 다른 환경의 CAD 시스템을 사용하기 때문에 제품 설계 데이터의 교환이 필요하다. 그러나 각각의 CAD 시스템마다 고유한 형식으로 데이터를 저장하기 때문에 같은 CAD 시스템에서 모델링한 파일이

<sup>+</sup> 한양대학교 대학원 기계설계학과

<sup>\*</sup> 교신저자, 한양대학교 기계공학과 (thchong@hanyang.ac.kr)  
주소: 133-791 서울시 성동구 행당동 17번지

아니면 데이터의 정확한 교환이 이루어 질 수 없다<sup>(1)</sup>.

이렇게 다른 환경의 CAD 시스템간의 데이터 교환 문제를 해결하기 위한 방안으로, 이미 1980년대에 들어오면서부터 각 나라별로 각기 중립 형태 표준(Neutral Format Standard)을 이용한 도면 정보 전송을 시도해 오고 있다.

IGES(Initial Graphics Exchange Specification)는 1980년 버전 1.0부터 시작하여 현재 버전 5.3까지 진행되어 왔으며, 처음에는 2차원 도면 정보의 호환에서 시작하여 3차원 정보의 호환에 이르기까지 적용범위가 넓어졌다. 최근에는 국제 표준인 STEP(Standard for The Exchange Product Model Data)을 이용한 교환이 적용되고 있다<sup>(2)</sup>.

STEP은 ISO에서 제안한 새로운 국제 표준으로, 제조업체에서 제품을 개발하고 생산할 때, 서로 다른 자동화 시스템들 간에 제품 정보를 교환하는 데 사용되는 공통의 언어 역할을 하는 인터페이스 기술이다.

그러나 STEP을 이용한 데이터 교환은 솔리드 모델의 형상 정보를 포인트와 벡터 중심의 포맷을 이용하여 점, 선, 곡선으로 표현하고 있기 때문에 기술한 내용을 쉽게 파악하기 어려운 문서 구조를 가지고 있다. 따라서 최근의 솔리드 기반의 CAD에서 사용되는 피쳐(feature) 정보를 표현할 수 없다는 문제점을 가지고 있다.

이에 본 논문에서는 XML(eXtensible Markup Language)을 사용하여 설계자가 이해하기 쉬운 피쳐(feature) 기반의 형상정보를 표현할 수 있는 문서구조를 개발하여 CAD 시스템 간에 교환하는 연구를 수행하고, 형상정보 교환의 검증을 위해 간단한 피쳐들로 이루어진 솔리드 모델을 CAD 시스템에 적용하였다.

## 2. XML을 이용한 형상 정보의 중립 형식 개발

### 2.1 STEP을 이용한 형상정보의 교환

STEP은 순수한 수학적 표현에 바탕을 둔 형상모델(geometric model)과 함께 가공과 생산을 위한 정보를 포함하는 확대된 개념의 제품모델(product model)을 포함한다<sup>(3)</sup>. 이러한 표준 문서는 58개 이상의 파트(part)로 되어 있으며 이 중에서 응용프로토콜을 사용하는 3차원 형상정보를 표현하는 AP203, 자동차 산업의 표준을 위한 AP214 등 각 분야별 특성에 맞게 제정되어 있다<sup>(4)</sup>.

기계 부품과 조립품에서 3차원의 설계를 위한 응용 시스템간의 설계 정보의 교환을 목적으로 하는 AP203의 표현 가능한 타입으로는 토폴로지(topology)정보가 없는 와이어

```
ISO-10303-21:
HEADER:
FILE_DESCRIPTION (( 'STEP AP203' ),
"1");
FILE_NAME ('box.STEP',
'2002-11-22T03:06:46',
( '최의성' ), ( '한양대학교' ),
'SwSTEP 2.0',
'SolidWorks 2001340',
"");
FILE_SCHEMA (( 'CONFIG_CONTROL_DESIGN' ));
ENDSEC:
DATA:
#1 = ORIENTED_EDGE ( 'NONE', *, *, #11, .T. );
#2 = VERTEX_POINT ( 'NONE', #105 );
#3 = ORIENTED_EDGE ( 'NONE', *, *, #65, .T. );
#4 = EDGE_CURVE ( 'NONE', #22, #78, #110, .T. );
#5 = ORIENTED_EDGE ( 'NONE', *, *, #26, .F. );
#6 = ORIENTED_EDGE ( 'NONE', *, *, #99, .T. );
#7 = EDGE_CURVE ( 'NONE', #389, #481, #121, .T. );
:
:
:
#485 = ORIENTED_EDGE ( 'NONE', *, *, #85, .T. );
#486 = EDGE_CURVE ( 'NONE', #78, #185, #117, .T. );
ENDSEC:
END-ISO-10303-21:
```

Fig. 1 Sample document of STEP AP203

프레임(wireframe)과 서피스(surface)의 표현, 토폴로지 정보를 가지는 와이어프레임의 지오메트리(geometry)정보의 표현, 토폴로지 정보가 있는 다양한 서피스 정보의 표현, 경계조건 정보 표현과 같이 4가지 정도로 표현할 수 있다. 이중 기하학적 형상 부분의 문서의 구조를 보면 점, 선, 벡터의 엔티티로 나타내고 있다. Fig. 1은 AP203의 형식에 따라 간단한 직육면체에 대한 STEP 문서를 표현한 것이다. 이렇게 나타낸 중립포맷의 문서는 설계자가 읽기 어려우며 솔리드 모델 기반 CAD에서 사용되는 피쳐의 정보를 표현할 수 없다.

### 2.2 XML과 XML 어플리케이션

XML(eXtensible Markup Language)은 1998년 W3C(World Wide Web Consortium)에서 표준으로 제안한 웹 문서 교환의 표준 형식으로서 HTML(HyperText Markup Language) 등과 같은 마크업 언어를 표현할 수 있는 메타마크업 언어(meta-markup language)이다<sup>(5)</sup>.

XML이 가장 일반적으로 사용되는 곳은 웹이겠지만, 웹 클라이언트에게 데이터를 전달하는 용도로만 XML을 생각 하는 것은 XML의 활용 범위를 너무 좁게 본 것이다. 최근

XML은 웹 문서 교환을 위해 사용하기 보다는 웹 기반 환경에서 데이터의 교환을 위한 중립파일 형식(neutral file format)으로서 활용하려는 연구가 많이 진행되고 있다<sup>6)</sup>. 최근 기업 간에 EDI(Electronic Data Exchange, 전자 자료 교환)가 구축되면서 기존의 EDI 포맷 대신 XML 포맷이 사용되기 시작한 것이 좋은 예이다.

특정 분야의 정보 표현을 위해 메타 마크업 언어인 XML로 작성한 마크업 언어(markup language)를 XML 어플리케이션(XML application)이라 한다. 따라서 전혀 다른 어플리케이션 사이에 데이터를 전달하기 위해서 XML이 이용될 수도 있다. 현재 개발되어 있는 XML 어플리케이션 중 공학적으로 활용할 수 있는 것에는 MathML(Mathematical Markup Language), CML(Chemical Markup Language), OSD(Open Software Description), SMIL(Synchronized Multimedia Integration Language) 등이 있다. MathML은 웹 브라우저에서 수학적식을 표현하기 위한 마크업 언어이고 CML은 화학식 및 분자식 등을 표현하기 위한 마크업 언어, OSD는 인터넷을 통한 분산 소프트웨어를 위한 표준 언어, SMIL은 플랫폼 간에 호환되며 독립적인 소프트웨어의 필요 없이 멀티미디어를 전송하는 표준 언어이다.

즉, XML은 태그(tag)의 이름을 임의로 정할 수 있으므로 HTML에서는 표현 할 수 없었던 수학적식, 화학식 등의 특정한 전문 분야에 대한 정보를 구조화하여 웹 기반 환경에서 공유할 수 있는 마크업 언어인 XML 어플리케이션을 생성할 수 있다.

### 2.3 피쳐 형상정보의 표현

대부분의 CAD 시스템에서의 모델링 작업은 시스템에 미리 정의된 피쳐(Pre-defined feature)와 사용자의 요구에 따라 새롭게 정의된 피쳐(user-defined feature)를 이용하여 수행된다.

피쳐를 기반으로 모델링을 하게 될 경우 설계자는 입체를 설계할 때에 각 피쳐의 주요치수와 위치를 파라미터로 입력하게 된다. 이러한 피쳐 기반의 모델링의 장점은 완성된 설계의 데이터에 내부적으로 그 입체를 구성하는 피쳐에 대한 정보가 정확하게 입력되어 있는 데 있다. 이 때문에 대부분의 CAD 시스템에서는 피쳐의 또 다른 특징인 완전 연관성에 따라 이 3차원 데이터를 이용하여 가공이나 조립체 설계, 설계 변경 등의 작업을 할 때에 작업 효율을 극대화시킬 수 있다.

이것은 파트 모델링의 기본 단위인 피쳐를 집을 지을 때

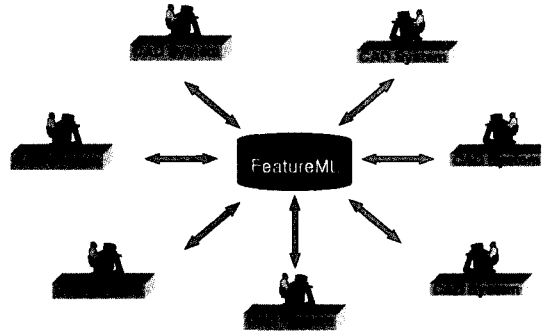


Fig. 2 The role of FeatureML

사용하는 벽돌과 같이 볼 수 있다. 벽돌을 쌓아 나가듯이 차곡차곡 생성하여 결과적으로 완성된 디자인을 창출할 수 있다. 쉬운 예로 구멍 뚫기(Hole)라는 피쳐가 존재할 경우에 이를 가공하기 위해서는 드릴링 작업이 필요하다는 것을 알 수 있으며, 후 공정에서 이 피쳐가 가진 가공정보를 이용해서 자동으로 드릴링 작업이 수행되도록 할 수 있다.

한편 피쳐 기반의 모델링은 각 CAD 시스템에서 제공되는 기본 피쳐 외에 그것이 사용되는 사용 분야나 사용자에 따라 피쳐의 종류가 제각기 다르다는 문제점이 있다. 그러나 각 CAD 시스템에서는 이러한 문제점을 해결하기 위해서 사용자의 용도에 맞는 피쳐를 임의로 정의하여, 필요 시 사용자가 지정할 수 있는 사용자 정의 피쳐(User-defined feature)를 제공하고 있다.

본 연구에서는 완성된 설계의 데이터에 내부적으로 그 입체를 구성하는 피쳐에 대한 정보가 정확하게 표현되는 피쳐 기반의 모델링의 장점을 활용하여, 피쳐 기반의 형상정보를 표현하는 중립 형식의 XML 문서를 개발하였고 이렇게 만들어진 문서를 FeatureML로 명명하였다.

Fig. 2와 같이 FeatureML은 서로 다른 CAD 시스템간의 형상 정보 교환을 위한 중립 포맷으로 사용된다. 상이한 CAD 시스템간의 정보 교환이 필요할 경우 FeatureML을 중간자로 이용함으로써 정보 교환이 가능해진다.

### 2.4 FeatureML의 구조와 문서

Fig. 3은 FeatureML의 구조를 보여준다. FeatureML의 구조는 각각의 피쳐(Point, Circle, Sketch, Extrusion, cut\_extrusion 등)로 구성되어 있다. 피쳐 형상정보를 기술하는 방법은 여러 가지가 될 수 있다. 본 논문에서는 최상단에 "<PART\_DOCUMENT>" 엘리먼트를 지정함으로써 단일

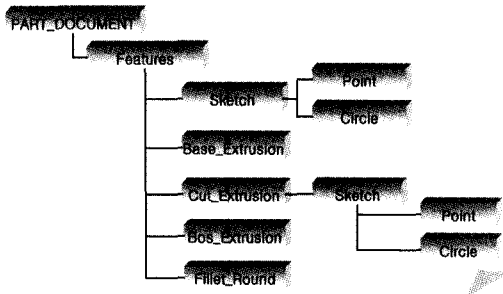


Fig. 3. The Structure of FeatureML

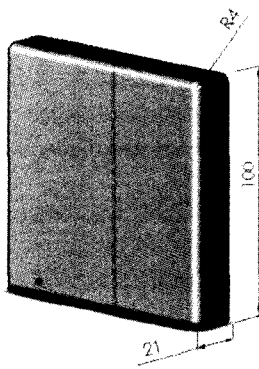


Fig. 4 The hexahedron with a fillet Feature

부품의 정보를 표현할 수 있도록 하였다. 최상위 요소 하위에는 각각의 피쳐들의 상위 요소로서 “<Features>” 엘리먼트가 존재한다. “<Sketch>” 엘리먼트의 자식요소로는 “<Point>” 나 “<circle>”과 같은 세부 피쳐 요소를 가질 수 있다. 이는 피쳐의 정보를 기초 평면에서 스케치한 부분이 하나의 컨벡스(convex)가 되어서 원하는 방향만큼 돌출시키거나 컷을 하는 형태를 가지고 있어서 <Point>와 <circle> 등으로 표시했다. 이외에도 스플라인 곡선, 자유곡선을 나타내는 엘리먼트를 추가할 수 있다.

Fig 4는 직육면체에 반지름 4mm인 필렛을 적용한 형상을 나타내고 있다. 이것은 먼저 XY 평면에 100\*100의 직사각형을 그리고 이를 Z축 방향으로 21mm 만큼 돌출시킨 돌출 피쳐를 생성한 후 마지막으로 반지름이 4mm인 필렛 피쳐를 생성하는 방법으로 3D 형상이 재구성된다.

Fig 5는 Fig. 4의 필렛이 적용된 직육면체의 형상 정보를 FeatureML 문서로 나타낸 것이다. 4행에서 31행은 돌출 피쳐의 형상정보를 표현한 것이고 32행은 필렛 피쳐의 반

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <PART_DOCUMENT>
3    <Features>
4      <Sketch>
5        <Point_1>
6          <x_point>0</x_point>
7          <y_point>0</y_point>
8          <z_point>0</z_point>
9        </Point_1>
10       <Point_2>
11         <x_point>100</x_point>
12         <y_point>0</y_point>
13         <z_point>0</z_point>
14       </Point_2>
15       <Point_3>
16         <x_point>100</x_point>
17         <y_point>100</y_point>
18         <z_point>0</z_point>
19       </Point_3>
20       <Point_4>
21         <x_point>0</x_point>
22         <y_point>100</y_point>
23         <z_point>0</z_point>
24       </Point_4>
25     </Sketch>
31     <Base_Extrusion direction="z_axis">21</Base_Extrusion>
32     <Fillet_Round>4</Fillet_Round>
33   </Features>
34 </PART_DOCUMENT>
    
```

Fig. 5 The FeatureML document for hexahedron with a fillet

지름 값을 표현한 것이다. Fig. 5와 같이 FeatureML은 피쳐의 정보를 설계자가 바로 이해할 수 있는 형식으로 표현되어 있는 것이 특징이다. 예를 들어 필렛의 반지름을 수정하고 싶을 경우 CAD 시스템을 실행할 필요 없이 FeatureML의 반지름 값을 바로 수정하면 된다.

### 3. CAD 시스템간의 교환 모듈 개발

#### 3.1 DOM과 CAD시스템의 API

XML 데이터를 표현하는 방법에는 DOM(Document Object Model)과 SAX(Simple API for XML)가 있다<sup>(7)</sup>. SAX는 DOM과는 달리 이벤트 기반의 처리 방법을 사용함으로써 메모리 사용을 최소화하고, 처리 속도가 빠른 것에 비해 문서의 변경이 용이하지 않다. 이에 비해 DOM은 트리 구조로 XML 문서를 매핑함으로써 XML 문서에 동적으로 접근하고 변경할 수 있는 프로그램 인터페이스이다. 따라서 객체 지향 프로그래밍 인터페이스인 DOM을 이용해서 XML 문서의 내용을 읽고, 변경하며, 새로운 XML 문서를 만드는 어플리케이션을 작성할 수 있다. 어플리케이션이 XML문서에 접근하면서 DOM 파서는 해당 XML 문서의 내용을 DOM 규격에 맞춰서 메모리에 저장하고, 어플리케이션은 DOM 규격에 정의된 방식으로 메모리에 로드된 DOM 트리에 접근할 수 있다. 본 논문에서는 MS XML 파서를 DOM 파서로 이용하였다.

CAD 시스템에 의해 생성된 모델은 각각의 CAD 시스템에 맞는 설계 방법으로 모델을 생성하더라도 그 결과는 같은 값을 가진다. 예를 들면 직육면체를 만드는 방법은 라인으로 사각형을 그리고 돌출시키는 방법과 사각형의 도형을 그리고 돌출시키는 방법이 있다. 그러나 이 둘의 결과 값은 같은 값을 가지게 된다. 즉, 모델 생성 방법의 차이일 뿐이다. 이렇게 생성된 모델의 형상 정보를 얻기 위해서 각각의 CAD 시스템에서 제공하는 어플리케이션을 이용하면 쉽게 얻을 수 있다.

SolidWorks API(Application Programming Interface)는 SolidWorks의 OLE 프로그래밍 인터페이스로 API에는 비주얼 베이직(visual basic)이나 C++과 같은 외부 프로그래밍 언어에서 SolidWorks를 제어할 수 있는 방법을 제공한다. SolidEdge API도 마찬가지로 SolidEdge를 제어하기 위한 다양한 API 함수를 제공하고 있어서 프로그래밍 언어를 이용해 SolidEdge 기능에 직접 접근 할 수 있다<sup>(8)</sup>.

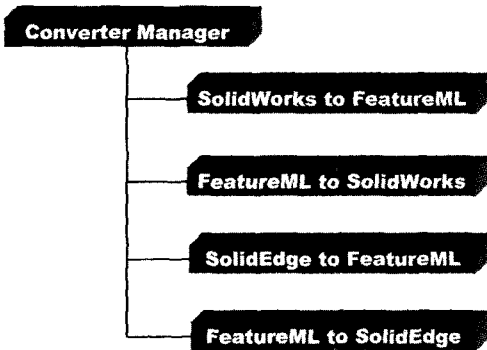


Fig. 6 The structure of FeatureML Converter

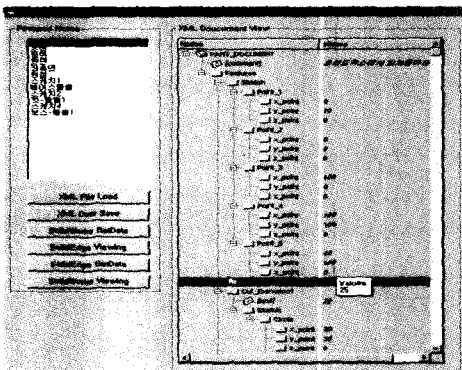


Fig. 7 FeatureML Converter

각각의 CAD시스템에서 제공하는 API를 이용하여 피쳐 정보를 추출하고, 이 정보들을 재가공하여 DOM 파서를 이용하여 FeatureML 문서를 생성함으로써 CAD 형상에서 FeatureML 문서로 변환되고, 반대의 작업을 통해서 FeatureML에서 CAD 형상이 재구성된다.

### 3.2 FeatureML 컨버터의 구조와 기능

Fig. 6과 같이 FeatureML 컨버터는 4개의 모듈로 구성되어 있다. 이러한 모듈단위의 구성은 각각의 어플리케이션을 연결하는 역할을 한다. 첫 번째 모듈은 SolidWorks에서 얻어진 피쳐 형상정보를 FeatureML의 문서를 만들게 된다. 두 번째 모듈은 이렇게 만들어진 FeatureML 문서를 SolidEdge로 피쳐 형상정보를 전달하게 된다. 전달된 정보를 가지고 SolidEdge를 통해 모델을 생성하고 볼 수 있다. 상호교환이 가능해야 하므로 세번째와 네번째 모듈은 SolidEdge에서 FeatureML, FeatureML에서 SolidWorks로 넘기는 역할을 하게 된다. 이렇게 생성된 모듈은 모듈의 추가로 다른 CAD 시스템에서도 적용할 수 있어서 확장이 가능한 장점을 가진다.

Fig. 7은 FeatureML 컨버터의 실행화면을 보여준다. SolidWorks GetData 버튼을 클릭하면 SolidWorks 형식의 파일을 읽어 들어서 API를 이용하여 피쳐 형상 정보를 분석한다 분석된 정보를 이용하여 Fig. 7의 왼쪽 상단에 있는 Features Name에 피쳐의 이름들을 추가하고 오른쪽 트리에는 SolidWorks 형상에 대응하는 FeatureML문서가 생성된다. 생성된 FeatureML 문서를 저장하고 난 후 SolidEdge Viewing을 통해 Solid Works에서 생성된 모델과 똑같은 모델을 SolidEdge에서 보여주게 한다. 화면 오른쪽에는 FeatureML 문서들을 트리 형태로 표현하여 FeatureML의 내용 파악 및 수정을 용이하게 하였다.

## 4. FeatureML의 적용

개발한 FeatureML을 SolidWorks와 SolidEdge에 적용하여 피쳐 기반의 형상정보 교환을 확인하였다.

Fig. 8은 개발된 FeatureML문서를 CAD 시스템 간에 컨버터를 통하여 데이터의 교환이 이루어지는 과정을 보여준다. 여기서 예제로 사용된 모델은 다면체, 원통, 직육면체의 3가지 피쳐의 조합으로 구성되어 있다. Solid Works를 이용해 표현된 형상정보를 FeatureML 컨버터를 통해서 추출하고 그 정보를 FeatureML의 구조에 맞게 XML 문서를 생성했다. 생성된 FeatureML 문서를 SolidEdge에서 문서를 열면 정확하게 모델이 생성된다. 또한 STEP의 중립형식의

로는 할 수 없었던 생성된 FeatureML 문서를 CAD 시스템이 없어도 직접 그 값을 수정할 수 있다. Fig. 9는 예제로 사용된 모델의 XML기반의 FeatureML 문서이다.

### 5. 결론

본 논문에서는 형상정보를 표현하는 중립형식의 XML 기반의 문서 구조를 개발하고, 이를 이용하여 다른 환경의 CAD 시스템 간에 교환이 가능하도록 하는 컨버터를 모듈 단위로 개발하였다.

개발된 피쳐 기반의 FeatureML 문서는 피쳐의 정보를 그대로 유지 할 수 있기 때문에 설계자가 읽고 수정하기 쉽고 확장과 관리가 편리하다. 이를 이용하여 다른 CAD 시스템 간에 데이터 교환이 가능하도록 하는 컨버터를 개발하였다. 개발된 컨버터는 피쳐 기반의 형상정보를 CAD 시스템 간에 교환하기 위해 피쳐 기반의 XML 공용문서를 각각의 CAD 시스템에 교환해 주는 모듈 단위로 구성하였다. 이것은 모듈의 추가로 다른 CAD 시스템과의 교환을 가능하게 한다. 컨버터는 XML 문서를 읽고 수정할 수 있으며 XML 문서를 불러오거나 저장이 가능하도록 하였다. 형상정보 교환의 검증을 위해 간단한 피쳐들로 이루어진 솔리드 모델을 SolidWorks와 SolidEdge에 적용하여 정확하게 데이터 교환이 이루어지는 것을 확인하였다.

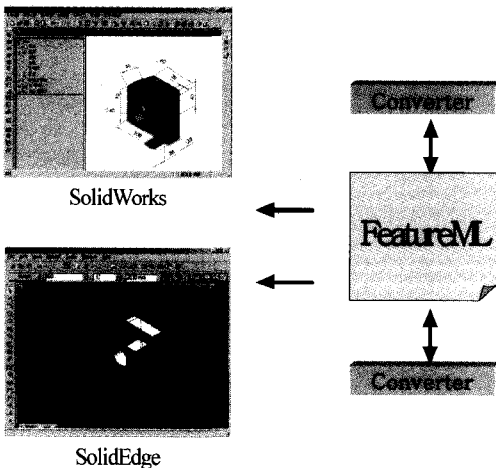


Fig. 8 The data exchange using FeatureML and FeatureML Converter

```

1<?xml version="1.0" encoding="UTF-8"?>
2<PART_DOCUMENT>
3  <Features>
4      <Sketch>
5          <Point_1>
6              <x_point>0</x_point>
7              <y_point>70</y_point>
8              <z_point>0</z_point>
9          </Point_1>
10         <Point_2>
11             <x_point>0</x_point>
12             <y_point>0</y_point>
13             <z_point>0</z_point>
14         </Point_2>
15         <Point_3>
16             <x_point>100</x_point>
17             <y_point>0</y_point>
18             <z_point>0</z_point>
19         </Point_3>
20         <Point_4>
21             <x_point>100</x_point>
22             <y_point>100</y_point>
23             <z_point>0</z_point>
24         </Point_4>
25         <Point_5>
26             <x_point>40</x_point>
27             <y_point>100</y_point>
28             <z_point>0</z_point>
29         </Point_5>
30     </Sketch>
31     <Base_Extrusion1 direction="z_axis">25
32     </Base_Extrusion>
33     <Cut_Extrusion1 direction="z_axis">25
34     <Sketch>
35         <Circle>
36             <x_point>30</x_point>
37             <y_point>30</y_point>
38             <z_point>0</z_point>
39             <Radius>20</Radius>
40         </Circle>
41     </Sketch>
42 </Cut_Extrusion1>
43 <Bos_Extrusion1 direction="z_axis">28
44 <Sketch>
45     <Point_1>
46         <x_point>70</x_point>
47         <y_point>80</y_point>
48         <z_point>53</z_point>
49     </Point_1>
50     <Point_2>
51         <x_point>70</x_point>
52         <y_point>40</y_point>
53         <z_point>53</z_point>
54     </Point_2>
55     <Point_3>
56         <x_point>90</x_point>
57         <y_point>40</y_point>
58         <z_point>53</z_point>
59     </Point_3>
60     <Point_4>
61         <x_point>90</x_point>
62         <y_point>80</y_point>
63         <z_point>53</z_point>
64     </Point_4>
65 </Sketch>
66 </Bos_Extrusion>
67 </Features>
68</PART_DOCUMENT>
    
```

Fig. 9 FeatureML document of the sample model

## 참고 문헌

- (1) Han, S. H., and Lee, S. K., 1995, "Data exchange between CAD systems using the STEP standard," *Journal of The Society of Naval Architects of Korea*, Vol. 32 No. 1, pp. 40~49.
- (2) Han, S. H., and Shin, Y. J., 1996, "Sharing of Structural Design Information Using STEP," *Proceedings of the Autumn Meeting : The Society of Naval Architects of Korea*, pp. 463~366.
- (3) Korean STEP Center, <http://kstep.or.kr>.
- (4) Na, H. S., Kim, S. S., Lim, K. I., and Kim, Y. J., 2002, "Information Model for 2D CAD Data based on STEP, " *2002 Conference : The Society of CAD/CAM of Korea*, pp. 371~382.
- (5) Extensible Markup Language(XML) 1.0, 1998, [www.w3c.org/XML](http://www.w3c.org/XML), W3C Recommendation.
- (6) Chong, T. H., and Park, S. H., 2002, "Extension of Legacy Gear Design Systems Using XML and XSLT," *Journal of Korean Society of Machine Tool Engineers*, Vol. 11 No. 4, pp. 19~25.
- (7) Document Object Model(DOM) Level 1 Specification, 1998, W3C Recommendation.
- (8) SolidWorks API Support, <http://www.solidworks.com/html/products/api/>.