

웹 서비스를 이용한 CAD 모델 정보의 획득

김병철*, 한순홍**

Retrieval of CAD Model Information Using Web Services

Byungchul Kim* and Soonhung Han**

ABSTRACT

Web Services are a distributed component technology based on XML and Web. The Web Services are used to integrate heterogeneous systems under the distributed environment. In this study, the concept of Web Services for CAD (WSC) is introduced, which aims at interoperability with CAD systems using the Web Services. This paper introduces one part of the WSC which enables remote users or applications to retrieve the geometries and the design history of CAD models using the Web Services. This paper proposes an architecture of the WSC, the data model and the interfaces for retrieving CAD model information, and the way to apply the WSC to commercial CAD systems.

Key words : Web Services, CAD, Internet CAD, Interoperability

1. 연구 배경

제품 설계 프로세스 상에서 CAD 데이터의 역할은 중요하다. CAD 데이터는 CAD 시스템을 사용하여 제품의 형상을 설계한 결과물일 뿐만 아니라, CAE, CAM, PDM과 같은 다른 시스템의 입력 데이터로 활용된다. 최근에는 기업 간의 협업이 증가하면서, CAD 데이터를 통한 설계 정보의 공유가 이슈로 되고 있다.

한 예로, GE는 항공기 엔진을 설계할 때 독자적으로 엔진의 모든 부분을 설계하는 것이 아니라 Parker 라는 기업과 작업을 나누어 설계를 진행한다. GE가 엔진의 전체적인 구조를 제안하고 설계를 하면, Parker에서는 노즐 부분에 대한 상세 설계를 한다. 이 두 기업이 진행하는 설계 내용은 인터넷을 통하여 수시로 정보가 교환되고, 자신들이 설계하고 있는 부분이 다른 기업이 설계하는 부분과 문제가 생기지 않도록 수시로 점검한다. 이 때 CAD 데이터는 두 회사가 설계하고 있는 부분이 서로 충돌하지 않는 지를 확인하기 위한 중요한 데이터가 된다. 또한 두 기업이 설계한 제품의 성능 해석을 할 경우에도, 원격지에 있는

CAD 시스템과 실시간으로 상호작용을 한다¹⁾.

GE와 Parker의 경우처럼 자동차나 항공기와 같은 거대한 제품을 설계하기 위해서는 단일 기업이 아닌 여러 기업이 한 제품의 설계 과정에 참여하게 된다. 이처럼 여러 기업이 참여하는 경우일수록 CAD 데이터를 통한 정보 공유는 중요한 문제가 된다. 빠른 CAD 데이터 공유는 제품 개발 시간을 단축시켜 주고, 이는 결국 제품 개발 비용의 감소로 이어지거나 때론 비용이다.

본 논문에서는, 협업 환경하에서 실시간으로 CAD 데이터를 공유하기 위한 한가지 방법을 제안한다.

본 연구의 최종 목표는 웹 서비스를 이용하여 CAD 데이터를 공유하는 것이다. 우리는 이를 WSC (Web Services for CAD)라 부른다. 웹 서비스는 SOA (Service-Oriented Architecture)를 가장 잘 표현하는 프레임워크이다²⁾. 따라서 본 연구의 결과는 CAD 시스템을 SOA 환경에서 활용하는 데 적용될 수 있다. 본 연구에서 이 논문은 웹 서비스를 이용하여 CAD 모델 정보를 획득하는 방법에 초점을 맞춘다. CAD 모델 정보를 CAD 시스템에서 가져오는 것은 CAD 데이터의 공유를 위한 가장 기본적인 기능이다.

본 연구의 결과는 앞의 GE와 Parker의 예에서처럼 기업 간 또는 한 기업 내의 부서 간에 CAD 데이터를 공유할 필요가 있을 때, 특히 SOA에 기반하여 정

*교신저자, 학생회원, 한국과학기술원 기계공학과
**중신회원, 한국과학기술원 기계공학과
- 논문투고일: 2007. 04. 09
- 심사완료일: 2007. 06. 12

보 시스템을 구성하였을 경우 유용하게 사용될 수 있다. 예를 들면, GE가 Parker로부터 받은 노즐 부분에 대한 CAD 파일이 포함하고 있는 형상 정보를 가져오기 위하여 WSC를 사용할 수 있다. 그리고 GE가 자신의 설계 시스템들을 SOA 기반으로 구축을 하였다면, WSC로부터 받은 결과를 해석 서비스로 전달하여 활용할 수 있다.

본 연구를 수행하는 데 있어서 가장 큰 이슈는 웹 서비스를 적용하기 위한 시스템 아키텍처와, XML에 기반한 데이터 모델의 정의 및 실제 시스템들이 상호 작용하기 위한 인터페이스를 정의하는 것이다. 본 연구의 이전 연구^[3]에서는 데이터 모델 및 인터페이스가 중립 모델이 아닌 특정 시스템에 의존적인 것이었다. 그러나 본 연구에서는 이를 개선하여 중립 형식의 데이터 모델 및 인터페이스를 제안한다.

본 논문은 이러한 내용을 위하여 다음과 같이 구성 되어있다. 우선 2절에서는 본 연구와 관련이 있는 연구에 대하여 살펴본다. 3절에서는 본 연구에서 사용한 웹 서비스의 개념에 대해서 간단히 설명한다. 4절에서는 현재 연구하고 있는 WSC의 개념과 제안된 아키텍처, 데이터 모델 및 인터페이스에 대하여 설명한다. 5절에서는 WSC를 Pro/ENGINEER에 적용하여 구현하고 실험한 내용을 보여준다. 마지막으로 6절에서는 향후 연구 방향에 대한 고찰이 있다.

2. 관련 연구

제품 개발에 참여하는 소프트웨어 시스템들이 CAD 데이터를 얻는 방법은 2가지가 있다. 그것은 데이터 교환(data exchange) 방법과 데이터 공유(data sharing) 방법이다^[4]. 데이터 교환은 데이터 파일을 이용하는 방법이고, 데이터 공유는 데이터에 접근하기 위한 API(Application Programming Interface)를 이용하는 방법이다. 두 방법에 대한 비교는 Table 1에 나와 있다.

Table 1. Data exchange and data sharing^[4]

| 데이터 교환 | 데이터 공유 |
|--------------------|----------------|
| 데이터 생성자가 교환 시작 | 데이터 사용자가 공유 요청 |
| 중립 형식으로 변환 | API 사용 |
| 데이터가 시간에 불연속적으로 변함 | 데이터가 실시간으로 변함 |
| 여러 개의 복사본 가능 | 하나의 데이터 원본만 존재 |

데이터 교환 방법은 일반적으로 표준 형식을 이용하게 되는데, 대표적인 표준 형식에는 국제 표준화 기

구(International Organization for Standardization, ISO)에서 제정한 ISO 10303 STEP(STandard for the Exchange of Product model data, 제품 모델 데이터 교환에 관한 표준)^[4,5]과 미국의 국가 표준인 IGES(Initial Graphics Exchange Specification)가 있다.

표준 형식을 이용하는 것 이외에도 CAD 모델의 매크로 파일을 교환하는 연구^[6,9]와 온톨로지를 활용하는 연구^[10,12]도 있었다.

데이터 공유 방법은 API를 구현하는 기술에 따라서 분류할 수 있다. 가장 기본적인 방법은 CAD 시스템이 제공하는 API를 사용하여 데이터를 공유하는 것이다. 그러나 이 경우 API는 대상 CAD 시스템에 특화되어 있기 때문에 여러 CAD 시스템과 데이터를 공유하는 경우 문제가 된다. 이를 위하여 제안된 것이 표준화된 CAD API이다. 표준화된 API로 제안된 것 중 대표적인 것이 CAM-I의 AIS(Application Interface Specification)^[13]와 OMG의 CAD Services^[14,15]이다. 이 중 CAD Services는 분산 환경에서 CAE 시스템과 CAD 시스템이 CAD 데이터를 공유할 수 있는 표준 인터페이스 규격이다. CAD Services에서는 분산 환경을 위하여 CORBA(Common Object Request Broker Architecture)^[16] 기술을 사용하여 인터페이스를 정의하였다.

인터페이스 표준을 정의하는 연구 이외에도 CORBA나 Java RMI(Remote Method Invocation), COM(Component Object Model)^[17]과 같은 미들웨어(middleware)를 사용하여 CAD 데이터를 공유하는 연구들도 있었다.

Chao^[18]는 인터넷 상에서 STEP 파일을 기반으로 CAD 데이터를 공유하기 위한 연구를 수행하였다. DMAC(Design and Manufacturing Automation Corporation)^[19]은 COM의 일종인 OLE에 기반하여 CAE와 CAD 시스템 데이터 공유를 위한 OLE for DM 표준 인터페이스를 제안하였다. Vahl^[20]은 CORBA에 기반을 둔 OMG의 CAD Services를 사용하여 CAD 시스템에서 얻어온 정보를 가상 환경 시스템에 가시화시키는 연구를 하였다.

이 연구들에서는 CORBA 또는 COM을 사용하여 CAD 데이터를 공유하기 위한 방법을 제시하였다. 그러나 CORBA나 Java RMI, COM과 같은 기존의 미들웨어들은 플랫폼 의존적이고 밀집한 커플링(coupling) 관계를 요구하기 때문에 인터넷에 기반을 둔 기업 간의 시스템 통합으로 확장할 수 없다는 한계점을 가지고 있다. 이를 해결하기 위하여 기업 간의 시스템 통합을 가능하게 하고, XML(Extensible

Table 2. Comparison of the related studies

| | Middleware | Domain | Interface design | Application to the commercial system |
|------------|--------------|------------------|------------------|--------------------------------------|
| Chao | N/A | Data Exchange | N/A | AutoCAD |
| DMAC | COM | Data Sharing | Object-oriented | SolidWorks/SolidEdge |
| Vahl | CORBA | VR Visualization | Object-oriented | MDT |
| This study | Web Services | Data Sharing | Service-oriented | Pro/Engineer |

Markup Language)과 인터넷에 기반을 둔 플랫폼 독립적인 통합 아키텍처를 지원하는 웹 서비스(Web Services)가 출현하였다.

본 연구와 앞의 연구들의 차이점은 1) 기반이 되는 프레임워크로 Java RMI, COM, CORBA가 아닌 웹 서비스를 적용했다는 것, 2) 웹 서비스가 연구에서 단순한 프로토콜 수단으로써의 의미 보다는 XML을 이용한 독립된 정보로써 의미를 가진다는 것, 3) CAD 시스템을 서비스화 했다는 것이다. Table 2는 본 연구와 관련 연구들의 비교표이다.

3. 웹 서비스

웹 서비스는 XML과 웹에 기반한 분산 컴포넌트 기술이다. 웹 서비스를 사용함으로써 시스템들은 서로 약한 결합을 가지게 되고, 이로 인하여 시스템들은 확장성을 가지게 된다.

웹 서비스는 다양한 표준 규격들로 정의되지만, 가장 기본적인 규격은 웹 서비스의 인터페이스를 정의하는 WSDL(Web Services Description Language)²¹⁾과 전송 프로토콜을 정의하는 SOAP(Simple Object Access Protocol)²²⁾이다. 이들은 기본적으로 XML 형식으로 되어 있다.

WSDL은 웹 서비스의 인터페이스를 기술하는 언어이다. WSDL은 주로 웹 서비스와 상호작용하기 위하여 필요한 프로토콜 바인딩이나 인터페이스, 메시지 형식을 XML 형식으로 기술한다. WSDL에는 지원되

는 인터페이스가 추상적으로 기술된 후에, 네트워크 프로토콜이나 메시지 형식에 바인딩된다. 또한 웹 서비스 요구자는 서비스 제공자가 어떤 기능을 제공하는지 알기 위하여 서비스의 WSDL을 필요로 한다. WSDL은 CORBA나 COM에서 인터페이스를 기술하기 위하여 사용하는 IDL(Interface Description Language) 또는 MIDL(Microsoft IDL)과 유사한 역할을 한다.

웹 서비스를 기술하기 위하여 WSDL 파일에는 데이터 형식, 메시지, 포트 형식, 바인딩, 서비스에 대한 내용이 XML 형식으로 기술되어 있다. 이 중, 데이터 형식은 XML 스키마(XML Schema)²³⁾를 사용하여 기술한다. XML 스키마는 XML 문서의 형식을 정의하는데 사용되는 언어이다.

SOAP는 컴퓨터 네트워크 상에서 XML에 기반하여 메시지를 교환하기 위한 프로토콜을 정의한다. SOAP는 웹을 통하여 웹 서비스에 사용되는 XML 문서를 주고 받는 것을 가능하게 한다.

웹 서비스를 적용하는 데 있어서 이슈는 WSDL, XML 스키마를 정의하는 방법이다. SOAP는 WSDL과 XML 스키마로부터 자동으로 바인딩을 시킬 수 있기 때문에 큰 문제가 되지는 않는다. 그러나 좀 더 나은 네트워크 전송 성능 향상을 바라다면 SOAP를 제어하는 부분까지 구현을 해야 할 필요가 있다. 그러나 본 논문에서는 이 부분까지는 다루지 않는다.

Table 3은 유사한 기술인 Java RMI, COM, CORBA를 웹 서비스와 비교한 표이다.

Table 3. Comparison of Java RMI, COM, CORBA and Web Services

| | Java RMI | COM | CORBA | Web Services |
|-----------------------|------------------------------------|---|--|----------------------|
| Protocol | JRMP (Java Remote Method Protocol) | ORPC (Open Network Computing Remote Procedure Call) | IIOP (Internet Inter-Object request broker Protocol) | SOAP |
| Interface Description | Java | MIDL | IDL | WSDL |
| Supporting Language | Java | Usually C++/VB | Usually C++ | Language Independent |
| Design Focus | Object | Object | Object | Service |
| Coupling | Tightly Coupled | Tightly Coupled | Tightly Coupled | Loosely Coupled |

4. CAD를 위한 웹 서비스(Web Services for CAD: WSC)

4.1 기본 개념

CAD 시스템의 API를 사용하여 데이터를 공유하는 경우는 다음과 같다.

- 1) CAD 모델의 생성을 자동화하기 위하여 프로그래밍적인 방법으로 CAD 모델을 생성하고자 할 경우
- 2) CAD 모델 파일이 주어졌을 때, 이로부터 CAD 모델의 정보(형상 및 위상 정보, 특징 형상 및 파라메트릭 정보 등)를 알고자 하는 경우
- 3) 파라메트릭 정보를 포함하는 CAD 모델 파일이 주어졌을 때, 이를 파라메트릭하게 변환하고자 할 경우

WSC는 이러한 경우들에 대하여 사용할 수 있는 기능들을 웹 서비스의 형태로 제공하는 것을 목적으로 한다. Fig. 1에서처럼 해석 프로그램이나 PDM 시스템과 같은 서비스 요구자들은 WSC가 제공하는 기능을 WSC가 제공하는 인터페이스를 사용하여 원격지에서 이용할 수 있다.

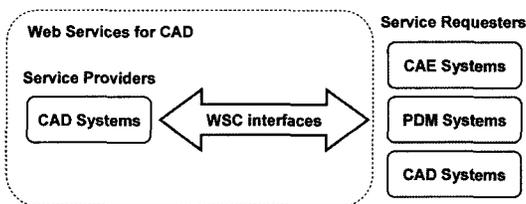


Fig. 1. Web Services for CAD.

서비스 요구자가 WSC를 이용하는 방법은 CAD 시스템이 제공하는 API나 OMG의 CAD Services가 제공하는 인터페이스와 같이 프로그래밍 언어에 바인딩이 된 함수 호출 형태로 접근한다. 그러나 실제로 데이터가 전달이 될 때에는 데이터 교환 방법과 같이 문서가 전달이 된다. 이 때, 이 문서는 XML 형태로 기술된다. 이러한 이유로 WSC가 제공하는 방법은 데이터 공유 방법과 데이터 교환 방법이 혼합된 형태라고 볼 수 있다. 데이터가 문서 형태로 전달이 되기 때문에 서비스와 관련된 모든 정보를 담아서 전달할 수 있고, XML 형태로 기술이 되기 때문에 복잡한 정보도 구조화된 형태의 문서로 전달할 수 있다. 이는 서비스가 복잡한 인터페이스를 정의하지 않아도 구조화된 XML 문서를 통하여 다양한 정보를 한 번에 전달할

수 있다는 것을 의미한다. 이렇게 두 가지 방법을 혼합할 경우, 인터페이스를 단순하게 설계할 수 있다.

이 연구에서 WSC를 이용하여 CAD 모델 정보를 가져오는 2)의 경우로 그 범위를 한정시켰다. 1)과 3)의 경우에 대해서는 아직 연구가 진행 중이다.

서비스 요구자가 2)의 경우에 대하여 WSC의 서비스를 제공받고 싶으면, 우선 서비스 요구자는 WSC 서비스를 제공하는 서버를 찾고, 이 서버로부터 WSC 인터페이스 정보를 가져와야 한다. 그리고 이 인터페이스 중 CAD 데이터 획득을 위한 특정 함수에 이진(binary) 형태의 CAD 모델 파일을 전달하면, WSC는 XML 문서 형태의 구조화된 정보를 서비스 요구자에게 반환한다. 일반적으로 CAD 모델은 면, 모서리, 꼭지점 순서의 계층적인 구조를 가지고 있기 때문에, CAD 모델의 하부 정보, 예를 들면 꼭지점의 좌표 정보에 접근하기 위해서는 면과 같은 상부 정보로부터 순차적으로 하위 단계의 정보에 접근해야 한다. OMG CAD Services와 같이 CORBA나 DCOM, Java RMI를 이용할 경우에는 상부 정보로부터 한 단계 아래의 정보에 접근할 때마다 원격 요청을 하기 때문에 많은 네트워크 부하가 걸릴 수 있다. 반면 WSC와 같이 웹 서비스를 사용하는 경우에는 한 번의 요청으로 CAD 모델에 관한 모든 정보를 XML 문서로 전달한다. 이처럼 XML 문서로 모든 정보를 한 번에 전달하게 되는 경우에는 XML 문서 자체가 많은 정보를 포함하고 있기 때문에 WSC와 서비스 요구자 사이에 의존성이 줄어든다. 또한 XML 문서를 다른 서비스를 통하여 가공하는 것도 쉽다. 최근 개발 환경에서는 XML 직렬화(serialization) 기술을 통하여 XML 문서 정보를 클래스 형태로 변환하여, 프로그래밍 언어에서도 쉽게 접근할 수 있다. 그러나 현재 WSC 개념에서는 전체 CAD 모델에서 원하는 정보를 선택적으로 가져올 수 있는 기능을 제공하지 않기 때문에, CAD 모델이 큰 경우에는 데이터를 가져오는 데 많은 시간이 걸릴 수 있다. 이를 해결하기 위해서는 선택적으로 CAD 모델의 일부 정보만을 가져올 수 있는 쿼리(query) 기능이 필요하다.

4.2 구성 컴포넌트

본 연구에서 제안하는 WSC를 구성하는 기본 컴포넌트는 Fig. 2와 같다.

Fig. 2에서 'CAD 시스템(CAD System)'은 우리가 사용하는 CAD 시스템을 의미한다. CAD 시스템은 상업용 CAD 시스템이 될 수도 있고, 형상 모델링 커널이 될 수도 있다. 'CAD 어댑터(CAD Adaptor)'는

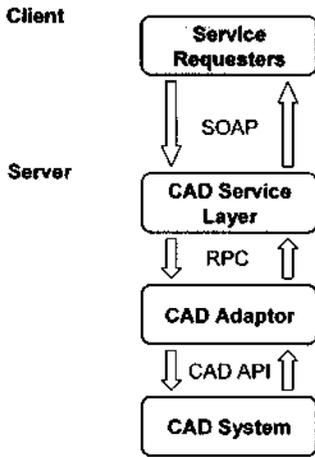


Fig. 2. CAD Services configuration components.

서비스 요구자가 전달해 준 CAD 모델 파일로부터 서비스 요구자에게 반환할 XML 형식의 서비스 결과 데이터를 가공하는 역할을 한다. 일반적으로 데이터를 가공하기 위해서는 CAD 시스템의 API를 사용한다. 여기서 생성되는 XML 문서는 ‘CAD 서비스 레이어 (CAD Service Layer)’를 통하여 서비스 요구자에게 전달된다. CAD 서비스 레이어는 웹 서비스로 CAD 어댑터의 기능을 제공하는 역할을 한다. CAD 서비스 레이어는 COM을 사용하여 CAD 어댑터와 통신을 한다. CAD 서비스는 서비스 요구자가 전달한 CAD 모델 파일을 CAD 어댑터로 전달하고 그 결과를 다시 서비스 요구자에게 전달한다. 이 때 서비스 요구자와 CAD 서비스 사이의 통신을 위하여 SOAP가 사용된다.

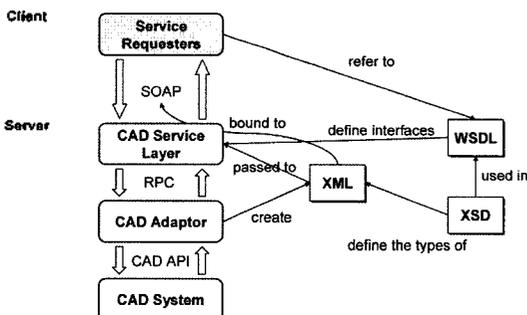


Fig. 3. Data for the configuration components.

Fig. 3은 WSC를 구성하는 컴포넌트들과, 이들이 사용하는 데이터를 표시한다. 앞에서 밝혔듯이 CAD 어댑터는 CAD 모델에 관한 XML 데이터를 생성한다. 이 때, XML 데이터는 XSD, 즉 XML 스키마로 정

의가 되는데, XML 스키마에는 CAD 모델 정보를 정의하는 데이터 형식뿐만 아니라, 웹 서비스 요구자가 서비스를 요구할 때 사용하는 입력 인자를 위한 데이터 형식도 정의되어 있다. CAD 모델 정보를 정의하는 데이터 형식은 WSDL 내에서 서비스의 반환 형식으로 사용된다. WSDL은 XSD 형태의 서비스 입력 및 출력 데이터 형식뿐만 아니라, 서비스가 제공하는 서비스 연산자를 정의한다. 또한 WSDL은 서비스 요구자가 서비스를 요청할 때 참조된다. CAD 서비스가 CAD 어댑터로부터 얻는 XML 데이터를 서비스 요구자에게 실제로 전달하기 위해서는 SOAP를 사용해야 한다. SOAP에는 앞서 정의한 WSDL을 수용하는 XML 데이터와 웹 서비스에 필요한 부가적인 정보들이 포함된다.

4.3 데이터 모델 정의

WSC에서는 CAD 모델 정보를 기술하기 위한 데이터 모델을 정의한다. 데이터 모델은 UML을 사용하여 표현된다. 이 UML 모델은 물리적 모델로 표현하기 위하여 XSD로 다시 표현이 된다. 그리고 XSD로 표현된 데이터 모델은 WSC의 결과를 XML 문서로 서비스 요구자에게 반환할 때 사용이 된다.

본 연구에서 제시하는 데이터 모델은 CAD 파트 모델을 명시적 모델과 절차적 모델로 동시에 표현한다. 이는 현재 상업용 CAD 시스템들이 사용하고 있는 일반적인 방법이다. 명시적 모델에는 솔리드 형상에 대한 정보뿐만 아니라, 참조 곡선, 참조 곡면 및 참조 평면과 같은 형상들도 정의되어 있다. 우리는 솔리드 형상 및 다양한 기하 형식을 정의하기 위하여 STEP Part 42^[24]를 참고하였다. 절차적 모델은 특징 형상들로 구성이 되는데, 특징 형상은 CAD 시스템마다 다르게 정의되어 있다. 따라서 본 연구에서는 절차적 모델과 특징 형상을 정의하는 데 있어서 STEP Part 55^[25]와 STEP Part 111^[26]을 참고하였다. STEP Part 55는 절차적 모델을 표현하는 방법을 기술하고 있고, STEP Part 111은 솔리드 특징, 형상에 관한 데이터 형식을 기술하고 있다. 또한 이와 관련이 있는 논문들^[27,28]도 참고하였다.

Fig. 4는 CAD 파트 모델 정보를 기술하는 데이터 모델의 전체적인 구조를 보여준다. 도식을 간략하게 표현하기 위하여 Fig. 4에는 상세한 속성이나 하위 클래스들 중 일부가 생략되었다. 데이터 모델의 최상위에는 Part 클래스가 있다. Part 클래스는 ExplicitShapeRepresentation 클래스와 ProceduralShapeRepresentation 클래스와 연관이 되어 있는데, Explicit

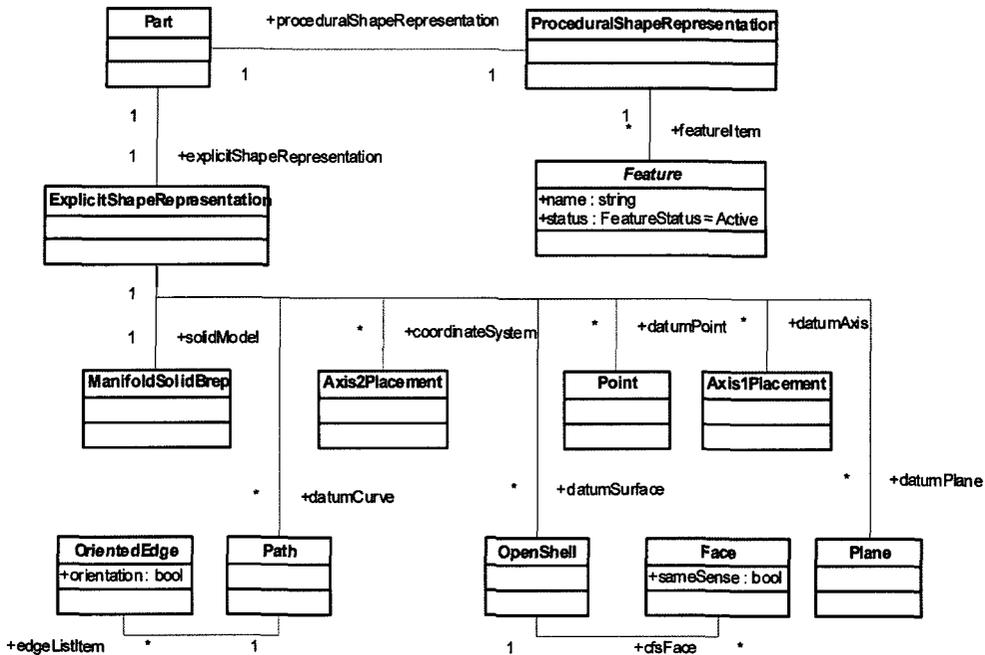


Fig. 4. The overall UML diagram of the WSC data model.

ShapeRepresentation 클래스는 파트 모델의 명시적 모델을 표현하고 ProceduralShapeRepresentation 클래스는 파트 모델의 절차적 모델을 표현한다. 절차적 모델을 표현하는 ProceduralShapeRepresentation 클래스는 특징 형상을 표현하는 Feature 클래스들로 구성되어 있다. 여기서 Feature 클래스의 하위 클래스에 대한 상세한 다이어그램은 생략되어 있다. 명시적 모델을 표현하는 ExplicitShapeRepresentation 클래스는 다시 솔리드를 표현하는 NonmanifoldSolidBrep 클래스, 참조점을 표현하는 Point 클래스, 좌표계를 표현하는 Axis2Placement 클래스, 축을 표현하는 Axis1Placement 클래스, 참조 평면을 표현하는 Plane 클래스, 참조 곡선을 표현하는 Path 클래스, 참조 곡면을 표현하는 OpenShell 클래스와 연관되어 있다.

Fig. 5는 ManifoldSolidBrep 클래스와 이와 관련된 클래스들을 UML 다이어그램으로 표현하고 있다. ManifoldSolidBrep 클래스는 다양체 솔리드 모델을 표현한다. 이는 STEP Part 42의 manifold_solid_brep 엔티티를 참고하여 정의되었다.

4.4 데이터 모델의 XSD 표현

UML로 표현한 데이터 모델을 웹 서비스에서 사용하기 위해서는 XSD로 구현을 해야 한다. 그런데 XSD로 표현할 때, 우선 고려해야 할 것이 클래스들

간의 연관 관계를 표현하는 방법이다. 예를 들면, 두 모서리(edge)가 한 개의 공통 꼭지점(vertex)을 참조할 때, 이를 XML에서 어떻게 표현할 지를 고려해야 한다. 본 연구에서는 참조될 필요가 있는 모든 인스턴스에 식별자를 부여하고, 같은 인스턴스를 재참조할 경우에는 이 식별자를 통하여 원래 인스턴스를 참조하도록 하였다.

우선 참조될 필요가 있는 모든 데이터 형식은 Table 4의 Entity 데이터 형식을 상속받는다. Entity로부터 상속받은 모든 데이터 형식은 id 속성을 가진다.

Table 4. The XML Schema of the Entity data type

```

<!-- Entity type -->
<xs:complexType name="Entity" abstract="true">
  <xs:attribute name="id" type="xs:string" />
</xs:complexType>

```

그리고 참조될 필요가 있는 데이터 형식들은 Table 5의 Vertex처럼 Entity로부터 상속을 받는다.

꼭지점은 실제 데이터를 정의하는 Vertex 데이터 형식과 참조 데이터를 표현하는 VertexRef 데이터 형식 두 가지로 표현이 된다. 그리고 이 두 데이터 형식을 선택적으로 표현하는 VertexHandle이 정의된다.

Table 5. The XML Schema of the Vertex data type

```

<!-- Vertex type -->
<xs:complexType name="Vertex">
  <xs:complexContent>
    <xs:extension base="Entity">
      <xs:sequence>
        ...
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="VertexHandle">
  <xs:choice>
    <xs:element name="Vertex" type="Vertex" />
    <xs:element name="VertexRef">
      <xs:complexType>
        <xs:attribute name="ref"
          type="xs:string" use="required" />
      </xs:complexType>
    </xs:element>
  </xs:choice>
</xs:complexType>
    
```

Table 6. The XML Schema of the Edge data type

```

<!-- Edge type -->
<xs:complexType name="Edge">
  <xs:complexContent>
    <xs:extension base="Entity">
      <xs:sequence>
        <xs:element name="EdgeStart"
          type="VertexHandle" />
        <xs:element name="EdgeEnd"
          type="VertexHandle" />
        ...
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
    
```

실제로 VertexHandle로 정의된 데이터 형식은 Table 6의 Edge 데이터 형식처럼 참조가 된다.

이렇게 정의된 Edge는 실제 XML 문서에서는 Table 7과 같이 표현된다. Table 7에서는 모서리 e44가 모서리 e185에서 이미 정의된 꼭지점 v1을 참조하기 위하여 VertexRef를 사용한다.

앞 절에서 UML로 표현한 데이터 모델은 이번 절에서 제시한 방법에 의해서 XSD로 표현된다.

Part 클래스에 대하여 정의된 XSD 데이터 모델은 서비스에 대한 결과를 서비스 요구자에게 반환하기 위하여 사용된다. 반면, 서비스 요구자가 입력 매개변수로 사용하기 위한 데이터 형식은 Table 8의

Table 7. The XML representation of the Edge instance

```

<Edge id="e185">
  <EdgeStart>
    <Vertex id="v1">...</Vertex>
  </EdgeStart>
  <EdgeEnd>
    <Vertex id="v2">...</Vertex>
  </EdgeEnd>
  ...
</Edge>
<Edge id="e44">
  <EdgeStart>
    <Vertex id="v7">...</Vertex>
  </EdgeStart>
  <EdgeEnd>
    <VertexRef ref="v1" />
  </EdgeEnd>
  ...
</Edge>
    
```

Table 8. The XML Schema of the PartBinaryContents

```

<!-- PartBinaryContents -->
<xs:element name="PartBinaryContents">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="FileName" type="xs:string" />
      <xs:element name="Contents"
        type="xs:base64Binary" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
    
```

PartBinaryContents와 같이 정의된다.

PartBinaryContents는 입력 CAD 파일에 대한 파일 이름과 파일의 내용을 바이너리(binary) 형식으로 저장한다.

4.5 인터페이스 정의

WSC가 서비스 요구자로부터 서비스를 요청받고 그 결과를 다시 서비스 요구자에게 반환하기 위해서는 서비스를 이용하기 위한 인터페이스가 필요하다. WSDL은 웹 서비스의 인터페이스를 정의하기 위한 웹 서비스 표준 규격이다.

웹 서비스의 인터페이스를 정의하기 위해서는 우선 데이터 형식이 정의되어야 하는데, 이는 앞에서 정의한 XSD를 사용한다. 그리고 XSD를 이용해서 서비스 요구자와 서비스 제공자 사이에 전달이 되는 메시지를 정의해야 한다.

Table 9에서는 WSC가 사용하는 두 개의 메시지가 정의되어 있다. RetrievePartDataRequestMessage는

Table 9. The WSDL messages definition

```

<definitions
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  targetNamespace="http://mnjsoft.net/ProeServices"
  xmlns:s0="http://mnjsoft.net/CADServicesTypes.xsd" ... >
  ...
  <message name="RetrievePartDataRequestMessage">
    <part name="RequestParameter"
      element="s0:PartBinaryContents" />
  </message>

  <message name="RetrievePartDataResponseMessage">
    <part name="ResponseParameter" element="s0:Part" />
  </message>
  ...
</definitions>

```

Table 10. The interface definition

```

<definitions
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  targetNamespace="http://mnjsoft.net/ProeServices"
  xmlns:tns="http://mnjsoft.net/ProeServices" ... >
  ...
  <portType name="ProeServicesInterface">
    <operation name="RetrievePartData">
      <input
        message="tns:RetrievePartDataRequestMessage" />
      <output
        message="tns:RetrievePartDataResponseMessage" />
    </operation>
  </portType>
  ...
</definitions>

```

서비스에 대한 입력 매개변수로 사용되는 메시지로 XSD에서 정의한 PartBinaryContents 데이터 형식을 사용한다. RetrievePartDataResponseMessage는 서비스의 결과를 전달하기 위한 메시지로 XSD의 Part 데이터 형식을 사용한다. 이 두 개의 메시지를 가지고 WSC가 제공하는 인터페이스를 Table 10과 같이 정의하였다.

WSC는 CAD 모델 정보의 획득을 위해서 한 개의 연산자인 'RetrievePartData'를 정의한다. 이 연산자는 입력 매개변수로 앞에서 정의한 RetrievePartDataRequestMessage를 사용하고, 출력 매개변수로 RetrievePartDataResponseMessage를 사용한다.

WSC는 여기서 정의된 WSDL을 실제로 구현하고 SOAP를 통하여 데이터를 전달하게 된다. WSDL을 프로그래밍 언어에 바인딩을 시키고, SOAP를 통하여 데이터를 네트워크를 통하여 전달하는 것은 일반적으로 개발 플랫폼이 그 기능을 제공한다.

5. 구현 및 실험

5.1 실험 내용

WSC를 실제 시스템에 적용하기 위하여, WSC를 Pro/ENGINEER Wildfire 3.0(이하 Pro/E)에 구현하였다. 그리고 Pro/E에 WSC를 적용한 시스템(이하 ProeServices)이 제공하는 서비스를 테스트하기 위하여 클라이언트 프로그램을 구현하였다. 클라이언트 프로그램은 WSC에서 정의한 인터페이스를 사용하여 Pro/E의 파트 파일의 정보를 가져온다. 그리고 가져온 정보를 형상 모델링 커널인 ACIS를 사용하여 다시 솔리드 모델을 구성하고 이를 가시화 라이브러리인 HOOPS를 사용하여 화면에 가시화를 한다. 가시화가 정상적으로 되는 것을 통하여 본 연구에서 정의한 데이터 모델이 솔리드 모델을 잘 표현하고 있다는 것을 확인할 수 있고, 웹 서비스를 통하여 CAD 시스템과 상호운용 될 수 있다는 것을 보여줄 수 있다.

5.2 ProeServices 구성 컴포넌트

실험을 위하여 Fig. 6의 내용을 구현하였다. Fig. 6은 Fig. 2의 WSC 구성 컴포넌트에 해당하는 ProeServices 구성 컴포넌트를 보여준다.

ProeServicesProxy는 ProeServices와 SOAP를 통하여 상호 운용되는 클라이언트 측 컴포넌트이다. WSCClient는 사용자로부터 Pro/E 모델 파일을 입력으로 받고 결과를 화면에 보여주는 프로그램이다. WSCClient는 사용자로부터 Pro/E 모델을 입력 받으면, 파일 정보를 가져오기 위하여 ProeServicesProxy를 호출한다. 그리고 ProeServicesProxy는 ProeServices에 서비스를 요청한다. 이 때, WSCClient는 자신이 사용하는 서비스가 Pro/E에 구현된 서비스인지 알지 못한다. 단순히 WSC가 정의한 인터페이스를 사용할 뿐이다.

서비스를 요청받은 ProeServices는 ProeServicesProxy로부터 CAD 모델 파일을 입력으로 받고 이를 ProeAdaptor에게 넘겨준다. ProeAdaptor는 Pro/E의 API인 Pro/TOOLKIT을 이용하여 Pro/E로부터 CAD 모델 정보를 가져온다. 그리고 이 정보를 XML 문서로 가공을 하고, ProeServices에게 넘겨준다. ProeServices는 다시 이 XML 문서를 SOAP를 통하여 ProeServicesProxy에게 반환한다. ProeServices와 ProeAdaptor의 인터랙션은 ProeAdaptor의 API를 사용하여 행해진다. ProeAdaptor의 API는 COM으로 구현된다.

ProServices로부터 XML 문서를 넘겨받은 ProServicesProxy는 이를 다시 WSCClient에게 넘겨주고 WSCClient는 XML 문서로부터 솔리드 모델을 구성하고 화면에 가시화를 한다.

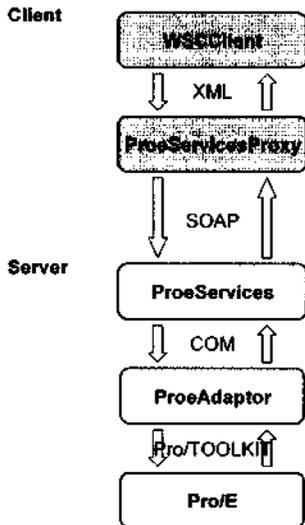


Fig. 6. ProServices configuration components.

5.3 구현 환경

ProServices는 다음과 같은 환경에서 구현되었다.

- 운영체제: Windows XP
- CAD 시스템: Pro/ENGINEER Wildfire 3.0
- ProAdaptor
 - Visual C++ 2003
 - WTL (Windows Template Library)
 - ATL (Active Template Library)
 - Pro/TOOLKIT
- ProServices
 - Visual C# 2005
 - .NET Framework 2.0
 - WSE (Web Services Enhancements) 3.0
 - IIS (Internet Information Services)
- ProServicesProxy
 - Visual C# 2005
 - .NET Framework 2.0
 - WSE (Web Services Enhancements) 3.0
- WSCClient
 - Visual C++ 2003
 - MFC (Microsoft Foundation Classes)
 - ACIS R15
 - HOOPS 12.0

ProAdaptor는 Pro/E의 API인 Pro/TOOLKIT를 사용하기 때문에, Pro/TOOLKIT에 가장 적합한 MS Visual C++ 2003을 개발 환경으로 사용하였다. 처음에는 MS Visual C++ 2005를 사용하려고 했으나 Pro/TOOLKIT과 잘 호환이 되지 않았다. ProAdaptor의 응용 프로그램 프레임워크로는 일반적으로 Visual C++에서 많이 사용하는 MFC (Microsoft Foundation Classes)가 아닌 WTL을 사용하였다. WTL 역시 Microsoft에서 개발한 라이브러리로 MFC보다 작고 가볍게 설계되었다. 또한 WTL은 COM을 구현하기 위하여 사용된 ATL과 잘 호환된다.

ProServices는 웹 서비스를 쉽게 구현할 수 있도록 .NET Framework와 C#을 사용하여 구현되었다. 또한 확장된 웹 서비스 기능을 사용하기 위하여 WSE 3.0을 사용하였다. 그리고 웹 서비스를 지원하기 위한 웹 서버로 IIS를 사용하였다.

ProServicesProxy 또한 웹 서비스를 잘 지원하도록 ProServices와 같은 환경에서 구현되었다. 그러나 웹 서비스 요구자는 웹 서비스를 지원하는 어떠한 환경에서도 구현될 수 있다.

WSCClient는 ACIS와 HOOPS를 가장 잘 사용할 수 있는 MS Visual C++ 2003에서 구현되었다. ACIS는 솔리드 모델을 구성하기 위하여 사용되었고, HOOPS는 ACIS로 구성된 솔리드 모델을 가시화하기 위하여 사용되었다.

.NET Framework는 WSDL을 해석하여 이에 해당하는 내용을 C# 클래스로 생성해 준다. 여기에 ProAdaptor가 생성한 XML 파일을 직렬화(serialization)시키면 XML 파일과 일치하는 C# 클래스의 인스턴스를 생성할 수 있다. 이를 해당 연산자의 반환값으로 반환을 하면 SOAP를 통하여 값이 반환된다.

ProServicesProxy가 Pro/E 파일을 ProServices에 전달하기 위해서는 앞에서 정의한 PartBinaryContents 요소를 사용해야 하는데, 이는 바이트의 스트림(stream)으로 정의가 되어 있다. 따라서 ProServicesProxy는 Pro/E 파일을 읽어서 바이트 스트림으로 변환을 하여 ProServices에 전달을 한다. 그러면 ProServices는 서버에 다시 같은 내용의 Pro/E 파일을 생성하고, 이를 가지고 작업을 한다.

5.4 구현 및 실험 결과

실험에는 Fig. 7에 보인 간단한 L 형상의 Pro/E 파트 모델이 사용되었다.

ProServicesProxy는 ProServices를 사용하기 위하여 우선 ProServices를 검색해야 한다. 이를 Visual

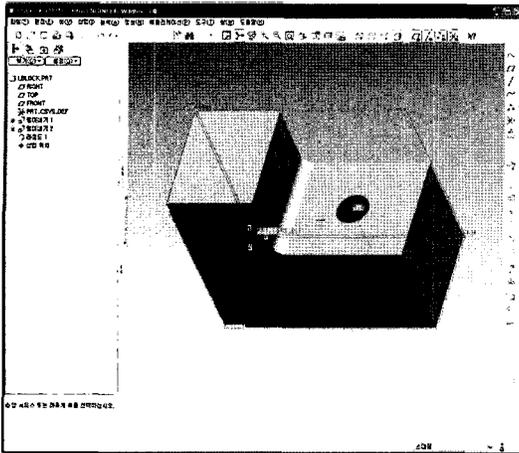


Fig. 7. L-shaped Pro/E part model.

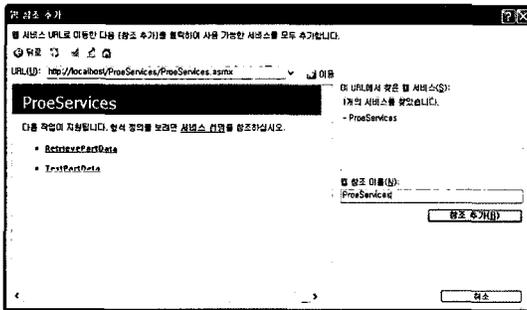


Fig. 8. Search for the ProeServices on the web.

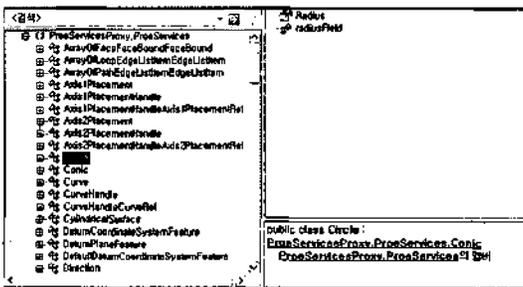


Fig. 9. The proxy classes generated from the WSDL.

Studio의 웹 서비스 검색기능을 통하여 수행할 수 있다. Fig. 8은 ProeServices 검색 화면을 보여준다.

ProeServices가 검색이 되면 이로부터 WSDL 파일을 가져올 수 있다. 그리고 가져온 WSDL 과 일치하는 프록시(proxy) 클래스들을 생성한다. Fig. 9는 WSDL로부터 생성된 프록시 클래스들을 보여준다. ProeServicesProxy는 이 프록시 클래스를 사용하여 자기가 웹 서비스를 사용하고 있다는 것을 인식하지 않아도 ProeServices의 기능을 사용할 수 있다.

```
<?xml version="1.0" encoding="EUC-KR" ?>
<Part xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
- <ExplicitShapeRepresentation>
- <SolidModel xmlns="http://mnjsoft.net/CADServicesTypes.xsd">
- <Outer>
- <Shell>
- <CfsFaces>
- <CfsFace>
- <Face id="F42">
- <FaceBounds>
- <FaceBound>
- <Bound>
- <Loop>
- <EdgeList>
- <EdgeListItem>
- <EdgeElement>
- <Edge id="e185">
- <EdgeStart>
- <Vertex id="v1">
- <VertexGeometry>
- <Point>
<X>200</X>
<Y>100</Y>
<Z>0</Z>
</Point>
</VertexGeometry>
</Vertex>
</EdgeElement>
</EdgeListItem>
</EdgeList>
</FaceBound>
</FaceBound>
</Face>
</CfsFace>
</CfsFaces>
</Shell>
</Outer>
</SolidModel>
</ExplicitShapeRepresentation>
</Part>
```

Fig. 10. The result XML document obtained from the ProeServices.

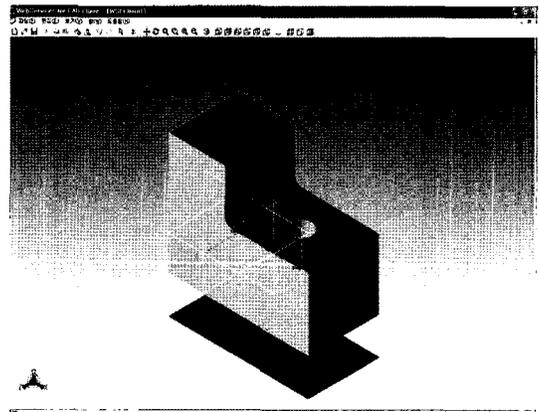


Fig. 11. The visualization of the result XML document.

이 클래스들을 이용하여 ProeServices에 서비스를 요청하면 Fig. 10의 XML 문서를 반환한다.

그리고 이를 ACIS를 사용하여 솔리드 모델을 구성하고 HOOPS로 가시화한 화면이 Fig. 11이다. Fig. 11의 형상이 Fig. 7의 형상과 동일한 것을 확인할 수 있다. 또한 웹 서비스를 통한 CAD 모델 정보 획득이 의도한 대로 수행된 것을 확인할 수 있다.

6. 결론 및 향후 연구

본 연구에서는 CAD 파일을 입력으로 받아서 이 파일에 대한 정보를 생성하는 WSC 개념 중 일부를 소개하였다. 본 논문에서는 WSC에 대한 개념 및 구성 요소에 대하여 기술을 하였고, 이를 Pro/E에 적용하여 ProeServices를 구현하는 과정을 기술하였다. 또한 구

현 및 실험을 통하여 그 결과를 보여주었다.

연구 과정에서 어려웠던 것은, 상업용 CAD 시스템이 대부분 C/C++ 기반 API를 제공하였던 반면, 웹 서비스를 구현하기 위한 환경은 C#이나 Java 기반으로 구현하기가 쉽게 되어 있어, 이 둘 사이의 갭을 해결하는 것이었다. 본 연구에서는 이를 해결하기 위하여 CAD 어댑터에서 CAD 시스템과의 인터페이스를 단순화 시키고 이를 COM으로 구현하였다. COM으로 구현된 인터페이스는 C#에서 쉽게 사용할 수 있다. 다른 어려움으로는 CAD 어댑터를 CAD 서비스 레이어를 통하여 웹 서비스로 제공할 때 사용자 권한을 설정하는 것이었다. 본 연구에서는 웹 서비스에 'impersonate'라는 속성을 제공해서, 외부 사용자가 마치 서버 계정을 통하여 웹 서비스를 사용하는 것처럼 할 수 있도록 하였다.

본 연구에서는 CAD 모델에서 원하는 정보를 선택적으로 가져올 수 있는 기능을 제공하지 않기 때문에, CAD 모델이 큰 경우에는 데이터를 가져오는 데 많은 시간이 걸릴 수 있다. 이를 해결하기 위해서는 선택적으로 CAD 모델의 일부 정보만을 가져올 수 있는 쿼리(query) 기능에 대한 연구가 필요하다. 또한 남아있는 WSC의 연구 부분인, WSC를 통한 모델 생성과 파라메트릭 변경에 대한 연구가 필요하다. 또한 Pro/E가 아닌 다른 CAD 시스템에서 그 결과를 적용해볼 필요가 있다.

감사의 글

본 논문은 과학기술부 특정연구개발사업 과제인 "자동차 모듈 설계용 e-엔지니어링 프레임워크 개발"의 일부이며, 연구수행에 지원해 주신 관계자 여러분께 감사 드립니다.

참고문헌

1. Kao, K. J., Seeley, C. E., Yin, S., Kolonay, R. M., Rus, T. and Paradis, M., "Business-to-Business Virtual Collaboration of Aircraft Engine Combustor Design", *Journal of Computing and Information Science in Engineering*, Vol. 4, No. 4, pp. 365-371, Dec. 2004.
2. Newcomer, E., *Understanding Web Services: XML, WSDL, SOAP, and UDDI*, Addison-Wesley, Sep. 2002.
3. 김병철, 한순홍, "웹 서비스를 이용한 CAD 모델 정보의 획득", 2007 한국CAD/CAM학회 학술발표회 논문집(2007년 1월 31일-2월 2일, 강원도 평창군 보광 워닉스 컨벤션센터), pp. 337-342, 2007년.
4. Kemmerer, S. J., *STEP: The Grand Experience*, National Institute of Standards and Technology Special Publication 939, U. S. Government Printing Office, Jul. 1999.
5. Owen, J., *STEP: An Introduction*, 2ed., Information Geometers, Jul. 1997.
6. 문두환, 한순홍, "매크로 파라메트릭 방법론을 이용한 CAD 모델의 교환", 한국 CAD/CAM 학회 논문집, 제6권, 제4호, pp. 254-262, 2001년 12월.
7. 문두환, 김병철, 한순홍, "피쳐 트리와 매크로 파일을 이용하는 하이브리드 파라메트릭 번역기", 한국 CAD/CAM 학회 논문집, 제7권, 제4호, pp. 240-247, 2002년 12월.
8. Choi, G.-H., Mun, D. and Han, S., "Exchange of CAD Part Models Based on the Macro-Parametric Approach", *International Journal of CAD/CAM* (Homepage: <http://www.ijcc.org>), Vol. 2, No. 1, 2002.
9. Mun, D., Han, S., Kim, J. and Oh, Y., "A Set of Standard Modeling Commands for the History-Based Parametric Approach", *Computer-Aided Design*, Vol. 35, No. 13, pp. 1171-1179, Nov. 2003.
10. 이윤숙, 천상욱, 한순홍, "CAD 시스템 간의 상호 운용성을 위한 설계 특징형상의 온톨로지 구축", 한국 CAD/CAM 학회 논문집, 제9권, 제2호, pp. 164-174, 2004년 6월.
11. Patil, L., Dutta, D. and Sriram, R., "Ontology-Based Exchange of Product Data Semantics", *IEEE Transactions on Automation Science and Engineering*, Vol. 2, No. 3, pp. 213-225, Jul. 2005.
12. Seo, T.-S., Lee, Y., Cheon, S.-U., Han, S., Patil, L. and Dutta, D., "Sharing CAD Models Based on Feature Ontology of Commands History", *International Journal of CAD/CAM* (Homepage: <http://www.ijcc.org>), Vol. 5, No. 1, pp. 39-48, Dec. 2005.
13. CAM-I, *Application Interface Specification (AIS) 2.0*, Technical Report, R-90-PM-03, May 1991.
14. Claus, R. and Weizer, I., "CAD Services - An Industry Standard Interface for Mechanical CAD Interoperability", *NASA Glenn Research Center Technical Reports*, NASA/TM-2002-211902, NASA Glenn Research Center, Oct. 2002, Homepage: <http://gltrs.grc.nasa.gov/cgi-bin/GLTRS/browse.pl?2002/TM-2002-211902.html>.
15. OMG, *Computer Aided Design Services Specification V1.2*, Jan. 2005, Homepage: <http://www.omg.org/technology/documents/formal/cad.htm>.
16. Siegel, J., *CORBA: Fundamentals and Programming*, John Wiley & Sons, Inc., Apr. 1996.
17. Rogerson, D., *Inside COM*, Microsoft Press, Feb. 1997.
18. Chao, P.-Y. and Wang, Y., "A Data Exchange Framework for Networked CAD/CAM", *Computers in Industry*, Vol. 44, No. 2, pp. 131-140, Mar. 2001.

19. Design and Manufacturing Automation Corporation, Inc., Homepage: <http://www.dmac.org/>.
20. Vahl, M. and Lukas, U., "Integration of Virtual Reality and CAD Based on OMG's CAD Services Interfaces", 10th Annual European Concurrent Engineering Conference'2003 (Apr. 14-16, 2003, Plymouth, United Kingdom), pp. 54-61, Apr. 2003.
21. W3C, Web Services Description Working Group, Homepage: <http://www.w3.org/2002/ws/desc/>.
22. W3C, XML Protocol Working Group, Homepage: <http://www.w3.org/2000/xmlp/Group/>.
23. W3C, XML Schema, Homepage: <http://www.w3.org/XML/Schema>.
24. ISO, "ISO 10303-42:2003, Industrial Automation Systems and Integration — Product Data Representation and Exchange — Part 42: Integrated Generic Resource: Geometric and Topological Representation", Geneva, Switzerland: International Organization for Standardization (ISO): 2003.
25. ISO, "ISO 10303-55:2005, Industrial Automation Systems and Integration — Product Data Representation and Exchange — Part 55: Integrated Generic Resource: Procedural and Hybrid Representation", Geneva, Switzerland: International Organization for Standardization (ISO): 2005.
26. ISO, "ISO/DIS 10303-111, Industrial Automation Systems and Integration — Product Data Representation and Exchange — Part 111: Integrated Application Resource: Elements for the Procedural Modelling of Solid Shapes", Geneva, Switzerland: International Organization for Standardization (ISO): 2005.
27. 양정삼, 한순홍, 김병철, 박찬국, "CAD 모델 교환을 위한 매크로 파라메트릭 정보의 XML 표현", 대한기계학회논문집 A권, 제27권, 제12호, pp. 2061-2071, 2003년 12월.
28. Yang, J., Han, S., Cho, J., Kim, B. and Lee, H. Y., "An XML-Based Macro Data Representation for a Parametric CAD Model Exchange", in Proceedings of the 2004 International CAD Conferences and Exhibition-CAD'04 (Pattaya, Thailand, May 24-28, 2004), Computer Aided Design and Its Application, Vol. 1, pp. 153-161, May 2004.



김 병 철

2001년 고려대학교 기계공학과 학사
2003년 한국과학기술원 기계공학과 석사
2003년-현재 한국과학기술원 기계공학과
박사과정

관심분야: Feature and Parametric Design,
CAD Data Exchange, Intelligent
CAD, Internet CAD



한 순 홍

한국과학기술원 기계공학과 교수이며,
2004년까지 웹 저널인 International
Journal of CAD/CAM(www.ijcc.org)의
편집장으로 활동하였다. 2003년까지
STEP센터(www.kstep.or.kr)의 회장과 전
자거래학회(www.calscc.or.kr)의 회장을
맡았으며, 관심분야는 STEP, 가상현실 응용,
지능형 CAD이다. 연락처는 shhan@kaist.ac.kr, 홈페이지 <http://icad.kaist.ac.kr>,
미국 미시건 대학에서 1990년 박사학위