

## 네트워크와 STEP 표준을 이용한 설계 정보 공유 시스템의 개발

조성욱\*, 최영\*, 권기억\*\*, 박명진\*\*, 양상욱\*\*

### Development of a Design Information Sharing System Using Network and STEP

Seong Wook Cho\*, Young Choi\*, Ki Eak Kwon\*\*, Myung Jin Park\*\*, Sang Wook Yang\*\*

#### ABSTRACT

An international standard for the product model data, STEP, and a standard for the distributed object technology, CORBA, will play a very important role in the future manufacturing environment. These two technologies provide background for the sharing of product data and the integration of applications on the network. This paper describes a prototype CAD/CAE environment that is integrated on the network by STEP and CORBA. Several application servers and client software were developed to verify the proposed concept. The present CAD/CAE environments are composed of several individual software components which are not tightly integrated. They also do not utilize the rapidly expanding network and object technologies for the collaboration in the product design process. In the design process in a large organization, sharing of application resources, design data and analysis data through the network will greatly enhance the productivity. The integration between applications can be supported by two key technologies, CORBA(Common Object Request Broker Architecture) and STEP(Standard for the Exchange of Product Model Data). The CORBA provides interoperability between applications on different machines in heterogeneous distributed environments and seamlessly interconnects distributed object systems. Moreover, if all the data in the CAD/CAE environment are based on the STEP, then we can exclude all the data conversion problems between the application systems.

**Key Words** : Object technology(객체기술), Distributed application(분산응용), STEP(스텝), CORBA(코바), CAD, CAE.

\* 중앙대학교 기계설계학과  
\*\* 중앙대학교 대학원 기계설계학과

## 1. 서론

현재의 CAD/CAE 시스템들은 대부분 독립적으로 운영되며 각 시스템마다 고유의 데이터 포맷을 갖는다. 또한 이들 각각의 시스템들은 여러 벤더들에 의해서 생산된 이기종의 시스템들로 구성되어 있다. 만일 네트워크를 통해 공동의 프로젝트를 수행하는 협력업체들이 서로 다른 CAD/CAE 시스템들을 보유하고 있다면 프로젝트 진행에 있어서의 비효율성을 피할 수 없다. 또한, 이기종의 시스템들로 이루어진 네트워크 환경에서 정보를 공유하는 분산 시스템을 구축하기 위해서는 어플리케이션(application)의 통합(integration)과 분산 프로세싱(processing)이 동시에 고려되어야 한다. 즉 응용프로그램에 네트워크 기능을 추가시켜야 하고 네트워크 시스템간의 상호작용(interoperation)을 고려해야 한다. 그리고 각각의 시스템에서 사용되는 응용프로그램들은 하드웨어 종류, 운영 체제, 네트워크 프로토콜과 어플리케이션 포맷에 무관하게 제작되어야 할 것이다. 이렇게 기업마다 상이한 CAD/CAE 시스템들을 분산 객체 기술의 표준인 CORBA를 이용하여 네트워크로 연결 한다면, 원격지에 있는 시스템 간의 협동 작업을 통하여 서로의 자원을 공유하거나, 필요한 정보를 주고 받는 등 일련의 상호작용이 가능해진다. 또한 CAD/CAE 정보를 제품 데이터 교환의 표준인 STEP으로 통일 한다면 시스템간의 데이터 변환이 필요 없이 제품의 생명주기의 모든 측면을 처리할 수 있다. 따라서 이와 같이 CORBA와 STEP을 지원하는 CAD/CAE 시스템에 대한 연구는 앞으로의 새로운 경향에 대비한다는 면에서 꼭 필요한 일이다.

본 연구의 목적은 설계·제조 환경에서 사용할 수 있는 분산 환경(STEP을 기반으로 하는 제품 데이터 관리 모듈, CAD 모듈, CAE 응용 모듈들이 CORBA를 통해 네트워크 상에서 연결되어 동시에 상호작용(interoperation)하는 네트워크 환경)을 구현하여 기업내의 설계·제조 파트를 구축하는 것이다. 이러한 분산 환경은 CAD/CAE 부문에만 한정되도록 설계되어 있지 않기 때문에 기업내의 전 영역으로 확장시킨다면 가상 기업(virtual enterprise) (네트워크를 통하여 정보를 공유하며, 생산 활동에 참여하는 기업들의 모임)을 구현하는 데에 매우 중요한 핵심기술이 될 것이다. Fig. 1은 STEP과 CORBA를 이용한 가상 기업의 구조를 나타낸 것이다. (1, 2, 3, 4, 5)

Fig. 1에서 CAD 작업과 해석 작업을 수행해야 될 작업자들은 클라이언트 전용 프로그램이나 웹브라우저를

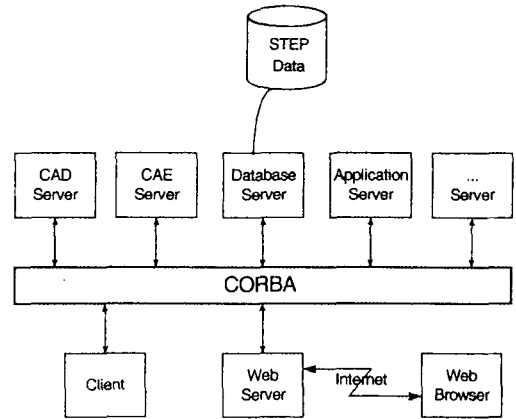


Fig. 1 가상 기업의 구조

이용한 클라이언트 프로그램 등을 사용하여 CORBA를 통해 각각의 서버와 연결을 하여 작업을 할 수 있으며 공동의 작업을 수행하는 여러 작업자들은 일하는 위치와 사용하는 하드웨어, 운영체제와 상관없이 다른 작업자들과 공동의 프로젝트를 진행시킬 수 있다. 이때 사용자들이 작업한 모든 내용은 데이터베이스 서버에서 관리하는 STEP 물리 파일로 저장된다. 각각의 서버들은 필요에 따라 기능이 추가되고 종류가 확장될 수 있다. 즉 사용자들의 요구에 따라 프로그래머가 필요한 서버 프로그램을 제작하여 기존의 시스템에 추가할 수도 있고, CORBA와 STEP을 지원하는 상용 프로그램을 구입하여 사용하던 시스템에 추가를 하여 계속 시스템을 확장할 수 있다. 이러한 방식으로 STEP의 각 파트들에 대한 서버들을 계속 추가시켜 나간다면 기업의 모든 부문을 커버할 수 있는 가상기업을 구현할 수가 있다.

이와 유사한 연구로는 STEP을 이용한 제품 데이터 공유를 위한 연구<sup>(23,24,25)</sup>와 CORBA를 이용한 분산 객체 환경에 대한 연구<sup>(26,27)</sup>가 진행된 바 있으나, 본 논문에서와 같이 STEP과 CORBA를 이용한 분산 CAD/CAE환경과 유사한 연구는 아직 발표된 것이 없다.

## 2. 관련 표준 기술

### 2.1 CORBA

분산객체는 네트워크를 통해서 자유롭게 접근할 수 있는 소프트웨어 객체를 말한다. 이것은 다른 객체와 마찬가지로 데이터와 그 데이터를 조작할 수 있는 메소드들로

이루어져 있으며 이러한 객체들은 프라퍼티와 메소드들에 의해 네트워크상에서 사용 가능하다. 이러한 객체들이 다양한 플랫폼과 프로그래밍 언어를 복합적으로 사용할 수 있도록 해주는데 이것을 분산객체라고 한다.

분산 컴퓨팅 시스템의 개발을 위한 객체 지향 표준을 제정하기 위해 700여 이상의 컴퓨터 관련 단체가 모여 OMG(Object Management Group)를 결성하였고 OMG에서 이종의 분산된 환경 하에서 응용 프로그램들이 서로 통합될 수 있는 표준 기술인 OMA(Object Management Architecture)를 제안하였다.

CORBA(Common Object Request Broker Architecture)는 OMA에서 이종의 네트워크 시스템에서 객체 간의 통신을 가능하게 해주는 아키텍처를 문서화하여 발표한 것이다. 1991년에 발표된 CORBA 1.1은 IDL과 구현된 ORB에서 클라이언트/서버간의 상호작용을 가능하게 하는 API를 정의 하였다. 그 이후 1994년 12월에 채택된 CORBA 2.0은 서로 다른 벤더들에 의해 생산된 ORB간의 상호작용을 가능하게 하였다.<sup>(6)</sup>

이종의 분산된 환경 하에서 응용 프로그램들이 서로 통합할 수 있는 표준 기술이 OMA이다. OMA는 응용 프로그램 간의 결합 뿐만 아니라 객체의 생성, 소멸에서부터 저장, 트랜잭션 기능에 이르기 까지 분산객체 환경에서 필요한 모든 서비스를 총칭하는 것이다. 이들 기능중 CORBA는 컴퓨터 내부의 버스처럼 서로 다른 프로그램들 사이의 버스 역할을 하는 모듈로서 OMA의 가장 중요한 요소이다. CORBA가 객체 간의 상호작용을 나타내는 것이라며 OMA는 응용 프로그램 간의 통합을 나타내는 것이다. OMA는 객체에 대한 서비스를 제공하는 하위 레벨의 CORBAservices와 어플리케이션에 대한 서비스를 제공하는 중간 레벨의 CORBAfacilities로 나뉜다.<sup>(7)</sup>

CORBA를 사용하는 이유는 다음과 같다.

CORBA는 기존에 사용하던 하드웨어와 개발 소프트웨어를 그대로 사용할 수 있으며, 추상화, 캡슐화, 상속성, 다형성 같은 객체지향 프로그램의 이득을 얻을 수 있다. 또한 프로젝트 개발을 위하여 여러 가지 혼합된 개발 툴들을 사용할 수 있고, CORBA에서 제공하는 서비스를 이용하여 작업량이 감소하므로 생산성을 극대화할 수 있다.

CORBA의 핵심 기술은 ORB(Object Request Broker)이다. ORB는 객체간에 클라이언트/서버 환경을 구축해주는 미들웨어이다. 즉 ORB는 클라이언트의 요청 메시지를 적당한 객체들로 보내준 다음 그 객체들로부터 나온 결과 메시지를 클라이언트로 보내주는 역할을 한다. 또한 ORB는 다중 오브젝트 시스템(multiple object systems)을 서로 연결할 수 있다. ORB를 사용하여 클라이언트는 로컬 머신이나 네트워크 건너편에 있는 서버 객체의 method를 호출할 수 있다. 클라이언트는 객체들이 어느 네트워크에 있는지, 그들과 어떻게 통신 할 것인지, 그들이 어떠한 프로그래밍 언어로 구현되었는지, 그들의 OS가 무엇인지, 그들이 어떻게 저장되었는지, 그리고 그들이 어떻게 실행 되는지 등에 대하여 사전 지식이 없어도 객체들의 서비스를 이용할 수 있다. CORBA의 클라이언트에는 클라이언트 스텝(Stub)과 프로그램 수행 시 원하는 메소드를 호출하는 동적 호출(Dynamic invocation) 기능을 통해 구현 객체를 호출할 수 있다. 일단 클라이언트의 호출이 발생하면 통신을 담당하는 ORB 기능을 통해 구현 객체에게 전달된다. 이때 구현 객체쪽의 객체 어댑터(Object Adaptor)는 기본적으로 해당 객체를 호출 가능하게 생성하고 원하는 호출에 해당하는 메소드를 실행시킨다. 일단 실행된 메소드에 의해서 처리된 결과는 다시 클라이언트에게 전달된다.<sup>(8)</sup>

CORBA에서 각각의 객체는 인터페이스를 필요로 한다. 이러한 객체의 인터페이스는 각 객체가 제공하는 서비스의 타입을 의미하며 IDL (Interface Definition Language)로 기술된다. IDL은 분산 환경에서 C, C++, Java, 스크립트처럼 언어나 컴파일러 등에 상관 없이 CORBA 응용 프로그램을 작성할 수 있게 해준다. 일단 IDL로 작성된 CORBA 응용 프로그램은 IDL 컴파일러를 통해 원하는 프로그래밍 언어에 적합하도록 변환된다. 사용자는 이 코드를 이용해 원하는 응용 프로그램을 작성하면 된다. IDL은 기본 데이터 타입들(short, long, float, double, 그리고 Boolean)과 구조 타입들

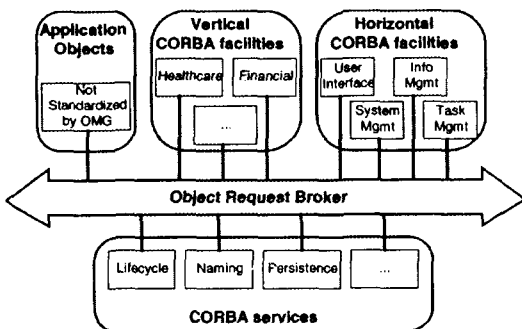


Fig. 2 OMG' s OMA

(struct, union)과 템플릿 타입들(sequence와 string)을 제공한다. 그리고 각각의 인터페이스 내에는 여러 개의 오퍼레이션들이 있을 수 있다. 바로 이 오퍼레이션이 객체가 제공하는 서비스를 의미하는 것이다. IDL은 특정 프로그램 언어의 컴파일러에 독립적이며, 다중 상속을 지원하는 등 대부분의 객체지향 개념을 지원하고, 동적 호출 메카니즘을 제공한다. 그러나 IDL은 구현을 위한 언어가 아니다. 즉 프로그래밍 언어가 아니다.

## 2.2 STEP

제품 데이터 교환의 새로운 표준으로 추진되고 있는 ISO 10303 STEP (Standard for the Exchange of Product Model Data)은 모든 산업에서 제품 생명 주기의 모든 측면을 다루는 중립적인 메카니즘을 제공하는 것을 목표로 하고 있다. 즉, 여러 나라들이 각각의 표준을 따로 가지고 있는 것 보다는, 모든 산업에서 제품 생명 주기의 모든 측면을 다루는 우수한 표준을 하나로 만드는 것을 그 목표로 하고 있다.

STEP과 다른 표준과의 한 가지 근본적인 차이점은 STEP이 미래 지향적이라는 점이다. STEP 개발에 연관되어 있는 많은 것들이 연구 개발의 결과물인 반면에, 다른 표준들은 CAD 시스템 개발자들에 의해서 만들어진 것이다. 또한 STEP은 생산정보와 설계정보를 모두 포함하고 있다는 점에서 다른 표준들과 차이를 갖는다. STEP 표준의 근간이 되는 것은 EXPRESS 정보 모델링 언어이다. EXPRESS는 객체지향적으로 설계되었기 때문에 기존의 STEP에 새로운 기술들을 첨가하는데 매우 적합하다.<sup>(9, 10)</sup>

STEP은 표준 자체의 구조를 반영하여 주는 파트들의 여러 그룹으로 나뉘어져 있는데 이러한 그룹 구조에서는 파트 번호가 작은 것일수록 다른 파트들에서 공통적으로 사용되는 내용을 담고 있다. 그룹의 범주에는 응용 프로토콜(application protocols), 통합 자원(integrated resources), 서술 방법(description methods), 구현 방법(implementation methods), 적합성 시험 방법 및 골격(conformance testing methodology and framework), 그리고 가상 시험 세트(abstract test suites)가 있다. 이 중 응용 프로토콜은 다른 모든 파트를 이용하게 되어있다. 특정 응용 분야에 적용되는 응용 프로토콜을 생성하기 위해서 통합 자원들이 해석되는데, 통합 자원의 규정적인 정보 모델은 EXPRESS 언어로 기술한다. EXPRESS 언어는 part 11에서 정의되고 있으며 실질적

인 제품 데이터, 즉, STEP 물리적 파일은 part 21 "Clear Text Encoding of Exchange String" 이 정한 규정에 따라 기술된다.

Fig. 3은 STEP의 전체적인 구조를 나타낸 것인데 이 중 응용 프로토콜(application protocol, AP)은 각 산업 응용 분야의 정보 요구(information requirement)를 명시하여, 신뢰성 있는 정보 교환을 위한 메카니즘을 정의하는 것이다. 본 연구에서 개발된 DICCES(Distributed CAD/CAE Environment using STEP & CORBA)는 STEP AP중에서 CAD와 관련된 AP 203 구성 제어 설계(Configuration controlled design)와 CAE와 관련된 AP 209 복합 금속 구조물의 해석을 이용한 설계(Design through analysis of composite and metallic structures)를 사용하였다.<sup>(11, 12)</sup>

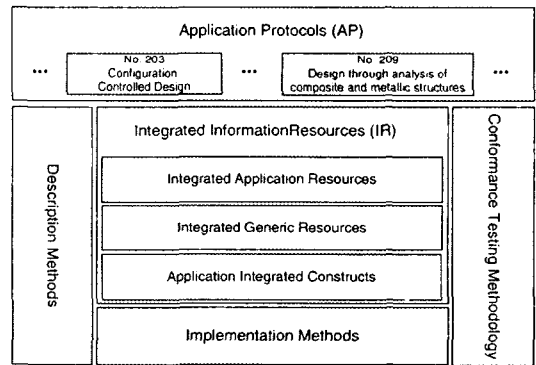


Fig. 3 STEP의 구조

EXPRESS는 일반화와 제약 조건을 표현할 수 있으며 엔티티-속성-관계(Entity-attribute-relationship) 모델에 기초한 정보 모델링 언어이다. STEP에서 개발되는 모든 공통 자원과 응용 프로토콜은 EXPRESS로 표현된 정보 모델을 반드시 포함하여야 한다. 따라서 STEP의 다양한 구현 방법들은 모두 EXPRESS를 기본으로 하고 있다.

이 언어는 사람이 읽기도 하고 컴퓨터에 의하여 번역되어야 하므로 정해진 문법에 따라 써어진다. EXPRESS는 공학 데이터를 모델링 하기 위하여 개발되었지만, STEP 표준화 작업 외의 많은 산업 또는 연구 프로젝트에 사용되고 있다. EXPRESS는 단지 방법론이 아니며 프로그

래밍 언어와도 다르다. 이것은 모든 정보 시스템의 설계 및 개발에 필요한 정보 모델을 설계하는 언어이다.

EXPRESS는 스키마(schema), 타입(type), 엔티티(entity), 상수, 함수(function), 프로시저(procedure), 규칙(rule) 등 7 개의 선언 구조체를 가지고 있고, EXPRESS로 표현된 엔티티 간의 상속, 포함 관계 등은 쉽게 C++ 코드로 표현할 수 있다. 본 연구에서는 C++ 언어로 구현하기 위하여 ST-Developer의 ROSE compiler를 사용하여 AP203 과 AP209 스키마로부터 각각의 엔티티에 해당하는 C++ 클래스를 생성하였다. 이들은 STEP part 23 "C++ programming language binding to the SDAI specification"에서 정한 규약의 실질적인 구현 방법이다.

### 2.3 VRML

인터넷에서 심심치 않게 등장하는 기술 중 하나가 VRML(Virtual Reality Modeling Language)이다. 인터넷을 통해 처리하는 3차원 기술이 논의되면서 1995년 8월에 VRML 1.0이 정의되었다. 하지만 VRML 1.0은 단순히 물체나 장면을 3차원으로 보여주고 시점을 인터넷의 다른 곳으로 옮겨 이동하는 기술밖에 제공하지 못했다. 따라서 3차원 그래픽을 인터넷에서 사용할 수 있는 표준이 결정되었다는 것 외에는 별다르게 주목받지 못했다.

최근에 발표된 VRML 2.0의 향상된 기능은 다양한 배경 표현과 효과음의 표현, 사용자와의 상호작용, 움직임의 표현, 스크립트 삽입, 프로토타이핑(Prototyping) 등 다섯 가지로 분류할 수 있다. 이상과 같은 사항들은 VRML 1.0에 비해서 기능적으로 월등히 나아졌다는 것을 의미한다. 이러한 기능의 추가 이외에도 VRML 2.0은 구조적으로 개선된 사항들이 많다. 결론적으로 말하면 VRML 2.0에 이르러서야 가상 공간을 표현하기 위한 대부분의 사항들이 갖추어졌다고 볼 수 있다.

VRML은 인터넷을 통해 3차원 그래픽을 표현하기 위한 표준으로 ISO/IEC 14772에 등록되어 있다. 하지만 단순히 3차원 그림을 보여주지만 하는 것으로 그치지 않는다. 다양한 형태의 표현이 가능하며 여러 가지 기능을 수행할 수 있다.

3차원 그래픽을 보여주는 VRML 파일은 텍스트(Text) 형태의 파일이다. 따라서 텍스트 에디터만 있어도 VRML 파일을 만들 수 있다. 웹 브라우저가 VRML 파일임을 알아내는 유일한 단서는 파일 이름이다. VRML 파일은 확장자가 ".wrl"로 보통 도트 월드(Dot

World)라고 읽는다. (13, 14, 15, 16)

## 3. 분산 CAD/CAE 환경

### 3.1 CORBA를 이용한 분산 CAD/CAE 환경

본 논문에서는 이러한 네트워크 시스템에서 정보를 공유하는 분산 시스템을 구축하기 위해 앞서 설명한 CORBA를 이용하였다. 그리고 CORBA를 이용한 분산 CAD/CAE 시스템을 구현하는 경우에도 마찬가지로 분산 시스템을 구축하기 위한 요구 사항이 적용되며 특히 설계부터 해석까지 일괄적인 작업이 가능하고, 작업자들이 효율적으로 업무를 수행할 수 있도록 프로젝트를 관리할 수 있어야 한다. 또한 분산 CAD/CAE 시스템을 구축하기 위해서는 네트워크 서비스, 관련 개발 도구와 기반 기술의 통합, 시스템 환경 구축을 위한 투자, 네트워크 어플리케이션 등이 필요하다. 이러한 요소들 중에서 네트워크 어플리케이션은 데이터베이스 서버와 각각의 응용 서버들로 이루어져 있으며 사용자가 STEP 데이터베이스를 완전하게 제어하거나 사용하기 위해서는 데이터베이스 서버와 사용자간에 데이터베이스 서버를 관리하기 위한 오퍼레이션을 IDL로 정의하는 관리 프로토콜(management protocol), STEP 데이터베이스를 파트 모델들로 분할하는 추상 프로토콜(abstraction protocol), 클라이언트들이 (SDAI를 이용하여) 네트워크를 통해 STEP 데이터베이스를 액세스할 수 있게 해주는 분산 프로토콜(distribution protocol)등이 필요하다. 본 연구에서는 분산 프로토콜에서 SDAI를 사용하는 대신에 STEP Tools사의 ST-Developer에서 제공하는 ROSE를 사용하였다. 한편 부품 정보를 위해 STEP 데이터베이스를 파트 모델들로 분할하는 추상 프로토콜은 본 연구에서 아직 구현되어 있지 않다.

본 연구는 기존의 CAD/CAE 시스템에 CORBA라는 분산 객체의 표준과 STEP이라는 제품 데이터 교환의 표준을 도입하였다. 그렇다면 CAD/CAE 시스템에 CORBA와 STEP을 이용하면 기존의 CAD/CAE 시스템보다 어떠한 장점이 생기는지 살펴볼 필요가 있다. 우선 분산 시스템의 장점을 이용할 수 있다. 하나의 컴퓨터에서 처리하던 작업을 여러 개의 서버에서 나누어 동시에 작업을 함으로써 작업 속도를 증가시킬 수 있다. 또한 여러 서버들 중에서 한 서버가 다운되거나 작업을 할 수 없는 경우가 생기더라도 다른 서버들은 계속 사용할 수가 있기 때문에 신뢰성이 높아진다. 그리고 작업자간에 데이

터를 공유할 수 있고 통신이 가능하기 때문에 효과적으로 프로젝트를 진행시킬 수가 있다. 두 번째로 분산 객체 기술을 이용하면 플랫폼에 대해 독립성을 가질 수 있다. 플랫폼 독립성은 다양한 서버들로 이루어진 네트워크 환경에서 많은 이점을 가지고 또한 프로그램 언어에 독립적인 면은 기존 코드를 재사용할 수 있고 개발자들에게 기술 습득을 용이하며, 시스템 개발시 데이터가 반드시 한 서버에 존재해야 하는 부담감을 없애준다. 마지막으로 STEP을 이용하면 각각의 응용 프로그램에 종속되는 데이터간의 교환을 위한 번역기가 필요 없으며 디자인부터 해석까지 일괄적인 작업이 가능해진다.

### 3.2 분산 CAD/CAE 시스템의 구조

분산 CAD/CAE 시스템은 데이터베이스 서버와 응용 프로그램 서버를 사용자들이 사용하는 클라이언트 부분과 완전히 분리하고 이들 사이에 연결 기능을 제공하는 미들웨어, 즉 ORB를 위치시킴으로써 좀더 유연하고 확장 가능한 시스템을 구현하였다. 분산 CAD/CAE 시스템은 STEP을 기반으로 한 데이터베이스 서버를 기점으로 하여 여러 종류의 응용 서버들(CAD 서버, CAE 서버 등)을 네트워크로 연결하여 데이터와 응용 시스템들을 공유하게 된다. 기본적으로 클라이언트는 모든 서버와 연결하여 작업할 수 있으며, 각각의 응용 서버들은 필요한 STEP 데이터 파일을 데이터베이스 서버에 요청을 하고 이를 얻을 수 있다. 그리고 각각의 응용 서버들은 C++를 이용하여 구현되었기 때문에 얻어진 STEP데이터 파일을 처리하기 위해 ROSE Library를 이용하였다. Fig. 4는 본 연구에서 개발된 분산 CAD/CAE 시스템의 구조를 나타낸다.

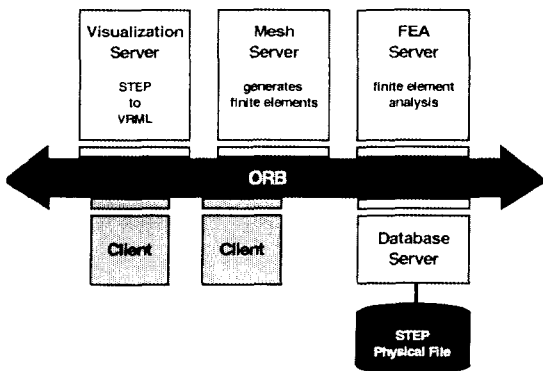


Fig. 4 분산 CAD/CAE 시스템의 구조

Fig. 4는 서론에서 설명한 가상기업의 구조에서 제품의 모델링과 해석을 위한 분산 CAD/CAE 시스템만을 나타낸 것이다. ORB를 통하여 여러 사용자들이 동시에 서버와 작업을 할 수 있고, CAD와 CAE 서버는 데이터베이스 서버로부터 작업 데이터의 정보를 얻고 작업 결과 데이터를 데이터베이스 서버에 저장할 수 있다.

### 3.3 분산 CAD/CAE 시스템의 작업 시나리오

일반적으로 분산 CAD/CAE 시스템에서는 일괄적으로 정해진 작업 순서 없이, 사용자는 원하는 시간에 ORB를 통하여 작업을 하기 위한 서버에 연결을 시도하여 자유롭게 일을 진행할 수가 있다. 그러나 이러한 작업에도 일정한 순서가 존재하게 된다. 클라이언트가 서버에 연결하여 작업을 하는 순서는 다음과 같다.

사용자가 우선 데이터베이스 서버에 연결하면 데이터베이스 서버가 작업 대상 STEP 물리 파일(AP203, AP209, ...) 리스트를 클라이언트에 보내준다. 이 리스트에서 사용자는 작업을 할 STEP 물리 파일을 선택한다.

작업 대상 STEP 물리 파일을 선택한 후 작업을 수행할 응용 서버에 연결하여 선택한 STEP 물리 파일의 정보를 넘겨주면, 응용 서버는 데이터베이스 서버에 연결하여 데이터베이스 서버로부터 선택된 STEP 물리 파일을 인스턴스한다. 클라이언트가 응용 서버에 작업(가시화, 메쉬 생성, 해석, ...)을 요구하면 해당 응용 서버는 클라이언트가 요구한 작업을 수행한다. 응용 서버의 계산이 끝나면, 클라이언트는 응용 서버에서 결과를 받을 수 있다. 모든 작업이 성공적으로 수행되어 응용 서버에 저장 명령을 내리게 되면 응용 서버는 변경된 내용을 데이터베이스 서버에 넘겨준다. 데이터베이스 서버에서는 작업 대상인 STEP 물리 파일에 변경된 내용을 갱신시켜서 저장하고, 또한 파일 리스트를 갱신시킨다.

여기서 주목할 점은 서버와 클라이언트간의 연결이 직접 이루어지지 않고 ORB를 통하여 이루어진다는 것이다. 즉, 클라이언트는 서버의 위치나 종류에 상관없이 ORB에 서비스를 요청하기만 하면 된다.

## 4. 프로토타입 시스템 개발

### 4.1 개발 환경

DICCES는 MS Windows NT 4.0 환경에서 개발되었으며, MS Windows 95/NT에서 운용된다. 프로그램

의 컴파일러로는 마이크로소프트사의 Visual C++ 4.2를 사용하였고, TGS사의 OpenInventor 2.21은 기본 윈도우에 형상정보를 display하기 위한 라이브러리로서 사용되었다. STEP Tools사의 ST-Developer 1.5는 STEP 물리파일을 읽어들이고 해석하기 위해 사용된 툴이고 Iona 사의 Orbix 2.02는 CORBA 툴로서 사용되었다. 변환 과정에서 사용되는 라이브러리는 공개 솔리드 모델링 라이브러리인 Irit 7.0을 이용하였다. (17, 18, 19)

#### 4.2 개발 프로세스

CORBA를 이용한 분산 프로그램의 개발은 일반적으로 맨 처음 해당 문제에 대한 클래스를 디자인하고, 해당 클래스를 IDL 문법에 맞게 작성한다. 일단 작성된 IDL 파일은 IDL 컴파일러에 의해서 클라이언트용 스텝 코드와 서버의 스키텔론 코드를 생성해낸다. 일단 CORBA에 의해서 자동으로 생성된 코드는 네트워크 부분을 위한 코드만 제공되기 때문에 클라이언트와 응용 서버 간의 실제 인터페이스를 위한 로직을 구현해야 한다. 마지막으로 이 모든 것을 컴파일하고 링크한 후 원하는 바이너리를 생성한다. 다음 Fig. 5는 DICCES의 전체적인 개발 과정을 나타낸 것이다.

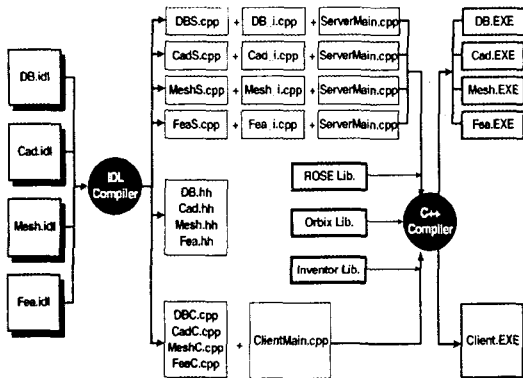


Fig. 5 개발 과정

다음은 DICCES의 개발 과정이다.

① 클라이언트와 서버간의 인터페이스를 IDL로 작성한 뒤 컴파일하면 헤더 파일, 서버 와 클라이언트쪽 C++ 소스(IDL2C++에 의해 생성된 C++ 소스) 파일이 생성된다.

② 클라이언트를 위해 생성된 C++ 소스 파일에 GUI

를 위한 소스 파일에 라이브러리를 추가하여 클라이언트 프로그램을 완성한다.

③ 각각의 응용 서버를 위한 소스 파일을 작성한다.

④ AP 203 과 AP 209 EXPRESS 파일을 컴파일하여 C++ 소스(EXPRESS2C++에 의해 생성된 C++ 소스) 파일을 생성한다.

⑤ IDL2C++ 에 의해 생성된 파일과 이에 대한 구현 파일 그리고 EXPRESS2C++에 의해 생성된 파일에 ROSE라이브러리, Orbix 라이브러리를 추가하여 데이터베이스 서버를 완성한다.

⑥ 응용 서버의 기능을 구현하기 위한 코드에 서버쪽의 IDL2C++ 파일, IDL2C++ 구현파일, Orbix 라이브러리를 추가하여 각각의 응용 서버를 완성한다.

⑦ 완성된 응용 서버를 Orbix daemon에 등록시킨다.

#### 4.3 서버 프로그램

##### ■ 데이터베이스 서버

데이터베이스 서버는 각각의 응용 서버에서 사용될 STEP물리 파일에 대한 기본 정보를 저장하고 관리한다. 즉, 임의의 작업(설계 및 해석)에 대한 프로젝트를 관리하고, 저장된 프로젝트 목록 및 작업 대상 파일 위치(로컬 머신 + 로컬 경로) 등의 정보를 클라이언트를 통해 사용자에게 알려주며, 또한 데이터 파일을 필요로 하는 응용 서버에 전달하는 역할을 한다. 사용자는 데이터베이스 서버의 STEP 물리 파일을 이용하여 필요한 작업을 진행하고, 작업 결과를 다시 데이터베이스 서버에 저장할 수 있다. 현재는 파일 단위로 관리가 되고있지만 데이터베이스 프로그램을 이용한다면 엔터티 단위의 서비스를 이용할 수가 있다.

##### ■ 가시화 서버

가시화 서버는 데이터베이스 서버로부터 STEP 모델링 데이터를 얻은 후 이 데이터를 클라이언트에서 Open Inventor를 이용하여 사용자에게 보여주기 위한 VRML 데이터로 변환시킨 후 클라이언트에 전송하는 기능을 한다. 경계 표현법 (Boundary Representation)으로 정의된 솔리드 데이터중의 페이스와 그 경계를 검색해서 페이스는 NURBS 곡면으로 변환하고, 페이스의 경계를 표현하는 형상은 NURBS 곡면상의 파라메트릭 도메인에서 표현되는 트리밍 곡선으로 변환하는 과정을 거쳐 페이스 단위의 중립적 데이터를 얻어낸다. 결과적으로 가시화 서버는 STEP물리 파일의 형상 데이터가 표현하고 있는 여러 종류의 페이스들이 클라이언트에서 가시화 될 수 있

도록 IDL로 정의한 VRML 인터페이스의 정의에 의한 VRML 리스트로 변환시킨 데이터를 클라이언트에 넘겨주는 서버이다. 이처럼 STEP 203 물리 파일을 VRML 2.0 파일로 변환시키는 이유는 STEP AP 203 스키마 중에서 형상 정보만을 선택하여 VRML 2.0으로 변환하므로 데이터의 양이 줄어들고, 변환된 VRML 파일의 구조가 삼각형 메쉬 구조로 되어 있기 때문에 IDL 인터페이스의 구조가 복잡하지 않아도 된다. 또한 사용자가 웹 브라우저만 있으면 간단히 VRML 데이터를 가시화할 수 있다는 장점도 있다. 또한 VRML 2.0에서는 여러 가지 멀티미디어 기능을 제공하기 때문에 좀 더 생동감 있는 장면을 구성할 수가 있다. 하지만 VRML 파일은 STEP 물리 파일이 가지고 있던 형상정보들을 상당 부분 상실하기 때문에 영구적인 보존 수단으로는 적당하지 않다.

VRML 2.0에는 형상을 표현하기 위해 50가지가 넘는 노드들이 있지만 STEP의 형상 엔티티들에 대응되지 못하는 부분도 존재한다. 대표적인 것이 NURBS 곡면과 곡선이다. 이러한 문제점으로 인해서 STEP AP 203 스키마로부터 VRML 2.0 노드로의 1 대 1 변환이 불가능하게 된다. 이러한 문제점을 해결하기 위한 방법으로는 VRML 2.0에서 제공하는 프로토타입을 이용하여 새로운 노드를 만들어서 사용하는 방법과 STEP 물리파일의 모든 형상 데이터들을 삼각화 알고리즘을 이용하여 삼각형으로 만들어서 VRML 형식에 맞게 저장시키는 방법이 있다. 본 논문에서는 후자의 방법을 이용하여 변환을 시도하여 STEP 물리 파일을 삼각형들로 이루어진 VRML 2.0 파일로 변환하였다. 이 과정에서 가장 중요한 것은 곡면을 삼각형으로 만드는 것이다. 본 논문에서는 공개 솔리드 모델링 라이브러리인 Irit 7.0에서 제공하는 함수를 이용하여 삼각형을 생성하였다.

■ 메쉬 서버

메쉬 서버는 데이터베이스 서버로부터 유한 요소로 분할하기 위한 모델링 데이터를 얻은 후, 클라이언트로부터 입력된 메쉬 사이즈 정보를 통하여 요소를 생성한다. 현재 여러 가지 이론에 기초하여 유한요소 격자를 생성하는 기술이 연구되고 있다. 각각의 이론과 방법은 그 장단점이 있다. 그러나 여러 가지 효율적인 측면을 고려하여 실제의 상용 프로그램에서 주로 사용하는 격자 생성방법은 사상법에 기초한 방법이다. 이 방법은 먼저 해석 영역을 세 변이나 네 변으로 둘러싸인 여러 개의 영역으로 분할해야 한다. 각 영역에 대한 요소의 생성은 매개 변수 영역에서 생성된 요소를 사상함으로써 이루어진다. 본 논문에서

서는 위에서 정의한 기본 모델링 데이터로 정의된 2차원 혹은 3차원 블록들로 이루어진 임의의 구조물을 Coons Patch를 위한 기본 식을 이용하여 영역의 경계선을 기준으로 내부의 요소와 절점을 생성한다.

■ 해석 서버

유한 요소 해석을 위한 해석 서버는 데이터베이스 서버로부터 분할된 유한 요소 데이터를 얻고 클라이언트로부터 입력된 경계 조건(하중, 변위)에 대한 정보를 사용하여 해를 구한다. 계산된 결과는 클라이언트의 요구에 의하여 데이터베이스 서버에 저장되거나 클라이언트에 다시 전송되어 보여진다. 유한 요소 해석을 위한 기본 데이터에는 절점과 절점으로 이루어진 요소와 이들 요소의 물성값과 각각의 절점과 요소에서의 결과값 등이 정의되었으며, 주어진 문제의 해(각 절점에서의 변위)를 계산하기 위한 해석 방법으로는 Skyline Solver를 사용하였다.<sup>(20, 21)</sup>

4.4 클라이언트 프로그램

서버와 클라이언트와의 통신은 IDL로 정의된 오퍼레이션으로 이루어지며 클라이언트 프로그램은 각 인터페이스의 오퍼레이션에 해당하는 사용자 인터페이스를 제공한다. 클라이언트 프로그램은 데이터베이스 관리, 가시화, 메쉬, 해석을 위한 기능을 모두 통합할 수도 있고 일부 기능만을 제공하는 형태로도 제공될 수 있다.

본 연구의 클라이언트 프로그램은 작업 파일을 선택할 수 있도록 프로젝트 목록 윈도우를 제공하며, 선택한 파일에 대한 가시화를 위하여 Open Inventor를 이용한 디스플레이 윈도우를 제공하고, 유한 요소 생성 및 해석을 수행하기 위한 요소 생성 조건과 해석 조건 등을 주기 위한 사용자 입력 윈도우가 제공된다. 또한 해석 결과를 출력하기 위한 텍스트 기반의 해석 결과 윈도우도 제공한다.

클라이언트는 각 응용 서버와 연결하여 사용자의 입력 정보를 연결된 응용 서버에 전달하는 역할과 응용 서버로부터의 메시지나 결과를 사용자에게 보여주는 역할을 한다. 이를 위해 보다 편리하고 명확한 사용자 인터페이스를 제공하여 이를 통해 입력된 정보를 응용 서버에서 사용되는 데이터로 변환시키는 기능과 연결된 서버로부터 전송된 데이터를 해석하여 디스플레이 해주는 기능을 갖는다.

앞서도 설명을 했지만 분산 객체에서 클라이언트 부분은 서버에 정의된 클래스의 메소드를 원격지에서 호출하고 이 결과를 전달받는 과정을 통해 작업한다. 즉 클리



엔트 부분은 정의된 메쏘드만을 호출하고 서버에서는 이 메쏘드 등 클래스를 구현해야 한다. 따라서 클라이언트 프로그램에는 실제로 계산을 하는 로직은 존재하지 않고 사용자로부터 입력을 받거나 결과값을 출력하는 GUI를 위한 기능이 추가 된다.

### 5. 프로그램 실행 결과

Fig. 6은 데이터베이스 서버에 연결하여 프로젝트 목록과 작업대상 STEP 물리 파일, 그리고 파일에 대한 정보를 보여주고 있다.

Fig. 7은 메쉬 서버에 연결하여 유한 요소로 분할하기 위한 모델링 데이터와 사용자 입력 정보를 통해 메쉬를 생성한 화면을 보여주고 있다.

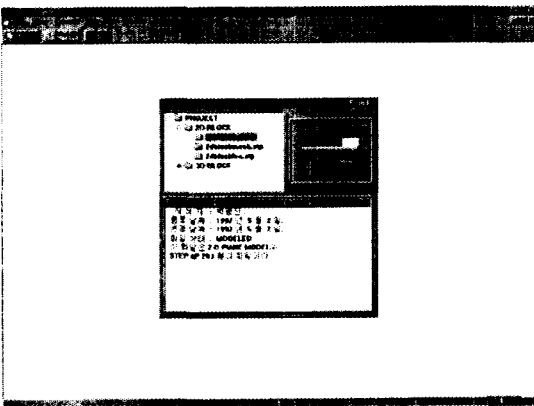


Fig. 6 데이터베이스서버 작업 윈도우

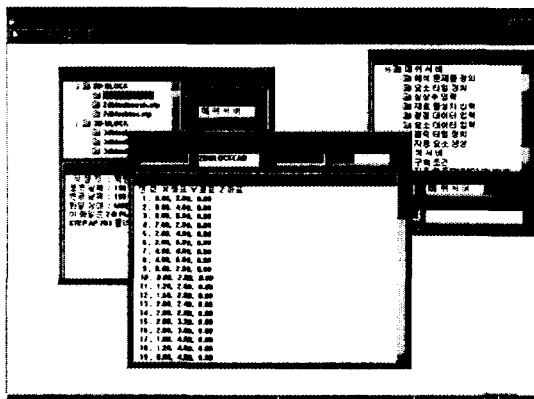


Fig. 7 메쉬 서버 작업 윈도우

Fig. 8은 DICCES환경에서 클라이언트가 구현된 예이며 우측 상단에 위치한 다이얼로그 박스 윈도우는 데이터베이스 서버에 연결하여 사용자가 원하는 STEP 물리 파일을 선택할 수 있도록 한 것이다. Fig. 8의 좌측 상단에 위치한 윈도우로부터 반시계 방향으로 각각 가시화 서버에서 전송된 모델링 데이터를 보여주는 윈도우, 메쉬 서버에서 전송된 메쉬된 모델링 데이터를 보여주는 윈도우, 해석 서버에서 전송된 해석 결과를 텍스트로 보여주는 윈도우를 보여 주고 있다.

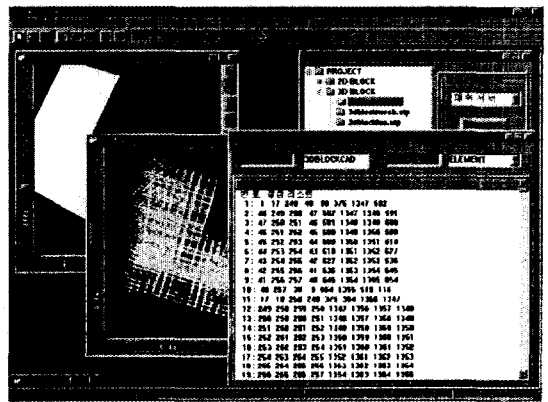


Fig. 8 클라이언트 작업 윈도우

Fig. 8은 3차원 모델링 데이터에 하중을 주어 해를 구하는 과정을 나타내고 있다. 작업자가 데이터베이스 서버에서 작업을 하기 위한 프로젝트를 선택한 후 가시화를 위한 STEP AP 203 물리 파일을 열면 가시화 서버에서 STEP 물리 파일을 VRML 파일로 변환시켜 클라이언트(사용자)에게 전송하게 되고 사용자는 이를 Open Inventor 윈도우를 통하여 확인하게 된다. 확인된 모델링 데이터를 해석하기 위해 메쉬 서버에 연결하여 데이터베이스 서버로부터 유한 요소로 분할하기 위한 모델링 데이터를 얻은 후, 사용자가 입력한 메쉬 사이즈 정보를 통하여 요소를 생성한다. 마지막으로 유한 요소 해석을 위한 해석 서버에 연결하여 데이터베이스 서버로부터 분할된 유한 요소 데이터를 얻고 클라이언트로부터 입력된 경계 조건(하중, 변위)에 대한 정보를 사용하여 해를 구한다. 계산된 결과는 클라이언트의 요구에 의하여 클라이언트에 다시 전송되어 텍스트로 결과값을 보여준다.

## 6. 결 론

본 논문에서는 네트워크를 기반으로 하는 제품정보의 공유 기술과 분산객체환경 기술을 연구하고 DB 서버, 가시화 서버, 메쉬 서버, 해석 서버를 클라이언트를 통하여 이용하는 DICCES를 구현하였다. 여기서 중요한 점은 S/W나 H/W 환경의 통일에 의한, 혹은 기업별 자체 표준 DB의 구축에 의한 기업별 정보 공유의 개념이 아니라는 것이다. 생산 데이터의 저장 및 응용 방식에 있어서 국제 표준을 사용하고 모든 응용 프로그램은 이를 이용하는 것이다. 또한 DICCES에서는 새로운 STEP을 기반으로 한 응용 서버를 개발하여 기능을 확장시키는 것이 가능하다. STEP에 등록되어 있는 모든 AP에 대한 응용 서버를 개발하여 DICCES에 추가하는 것이 가능하므로 CAD/CAE 분야 이외의 산업 부문에도 이용이 가능하다.

본 연구에서 개발된 각각의 시스템을 구성하는 요소들은 모두 표준을 사용하게 되는데, 이는 기업활동에서 적어도 생산정보 공유의 기술적인 측면에서는 국가간의 경제도 허물 수 있는 가능성을 보여주는 것이다. 따라서 본 연구의 결과로 나오게 되는 각 표준에 대한 응용 기술 및 개별 기술의 통합에 의한 프로토타입 시스템의 구현은 미래의 생산정보 인프라 구축에 대한 인식의 확산에 기여함은 물론 앞으로의 이 분야에 대한 기술적인 토대를 제공하게 될 것으로 기대된다. 또한 개발자의 입장에서 CORBA가 제공하는 서비스의 장점을 이용하여 보다 효과적으로 프로그램을 개발할 수 있으며, 작업자들은 분산환경하에서 STEP을 이용하여 생산성을 높일 수 있다.

향후 연구 방향은 본 논문에서 제시한 DICCES의 응용 서버의 기능을 강화하고 적용 범위를 확장시키는 것이다. STEP 물리 파일에 근거한 DB 서버를 오라클과 같은 관계형 DB나 객체지향 DB로 교체하여 생산, 설계 정보를 기반으로 하는 DB 서버를 구축하는 연구도 필수적이다. 또한 자바(Java)를 기반으로 하는 클라이언트 소프트웨어 구현에 의해 웹 브라우저를 이용하는 분산 환경을 구현하는 방안도 연구되어야 한다. 그리고, 분산환경에서 설계 및 해석에 참여하는 각각의 사용자(클라이언트)간의 원격 회의(conferencing) 시스템에 대한 연구도 함께 진행되어야 할 것이다.

## 후 기

본 논문은 1996년도 학술진흥재단 대학 부설 연구소

지원사업(96 우수, 중앙대학교 생산공학 연구소)에 의하여 지원되었습니다.

## 참고 문헌

1. Hardwick, M. et al., "Sharing manufacturing information in virtual enterprises," Communications of the ACM, Vol 39, pp. 46-54, 1996.
2. Marache, M. et al., "A CORBA-based infrastructure managing STEP distributed models for virtual reality applications," Proceedings of European Conference on Product Data Technology Days 1997, pp. 119-128, 1997.
3. Hardwick, M. et al., "Data protocols for the industrial virtual enterprise," Internet Computing, pp. 20-29, January - February 1997
4. Regli, W. C., "Internet enabled computer aided design," Internet Computing, pp. 39-50, January - February 1997.
5. Han, J. H., Requicha, A. G. "Modeler independent feature recognition in a distributed environment," Network-centric CAD, special issue of Computer Aided Design
6. "The Common Object Request Broker: Architecture and Specification 2.0," OMG, 1995.
7. Siegel, J., "CORBA fundamentals and programming," John Wiley & Sons, Inc., 1996.
8. 박재현, "분산객체 기술을 꿈꾸며, CORBA," 마이크로 소프트웨어, 1996년 3월호 ~ 1996년 6월호
9. STEP 연구회, "제품 모델 정보 교환을 위한 국제 표준 (ISO 10303) 스텝," 성안당, 1996.
10. "ISO 10303-11 - Part 11: Description Method : EXPRESS language reference manual," ISO, 1994.
11. "ISO 10303-203 - Part 203: Application Protocol: Configuration controlled design," ISO, 1994.
12. "ISO 10303-209 - Part 209: Application Protocol: Composite and metallic structural

- analysis and related design." ISO, 1996.
13. 김태한, "인터넷과 가상현실 세계의 오작교, VRML 2.0," 프로그램 세계, 10월, 1997.
  14. 문정수, "새로운 인터넷 세계로의 향해, VRML 2.0," 마이크로 소프트웨어, 6월, 8월, 1997.
  15. Ames, A. L., Nadeau, D. R. and Moreland, J. L., "VRML 2.0 Sourcebook," John Wiley & Sons, Inc., 1997.
  16. Roehl, B., Couch, J., Reed-Ballreich, C., Rohaly, T. and Brown, G., "Late night VRML 2.0 with java," Ziff-Davis Press, 1997.
  17. "STEP Programmer's toolkit - ROSE library reference manual," STEP Tools Inc., 1996.
  18. "Orbix 2 programming & reference guide," IONA Ltd. 1995.
  19. Gerson Elber, "Irit 7.0 Programmer's Manual," 1996.
  20. 이원양, "CAE 시스템을 위한 격자 생성기 및 열해석 모듈의 개발," 석사 학위 논문, 중앙대학교, 1997.
  21. Bathe, K. J., "Finite element procedures," Prentice Hall, 1996.
  22. 김철영, "웹과 STEP을 이용한 제품 설계 정보 공유 시스템" 한국 CAD/CAM 학회 논문집, 제1권 제3호 pp. 203-204, 12월, 1996.
  23. 서정우, "STEP을 이용한 선박 구획 형상 및 속성 정보의 공유방법," 한국 CAD/CAM학회 학술발표회 논문집 pp.283-290, 1997.
  24. 최 영, "WWW를 이용한 제품정보의 공유," 산업 공학회지 Vol. 23, NO. 3, September, 1997.
  25. 유상봉, 안만진, "표준 제품 데이터 변환 시스템," 한국 CAD/CAM 학회 학술발표회, 1998.
  26. 최 영, 양상욱, "ACORBA-based 3D STEP Viewer on the Internet," 한국 CAD/CAM 학회 학술발표회, 1998.
  27. 김준환, 한순홍, "CORBA를 통한 STEP 데이터베이스의 인터넷 검색," 한국 CAD/CAM 학회 학술발표회, 1998.