

네트워크 기반의 3D 컨퍼런싱 시스템

최영·양상욱

중앙대학교 기계설계학과

요약문

The international standard STEP for representation and exchange of product model data, and CORBA as a standard for distributed object computing environment are expected to play very important role in the future manufacturing environment. Platform independent network programming language Java is also becoming a standard environment for developing Internet/Intranet applications. Furthermore, the ObjectWeb environment supported by the combination of CORBA and Java technology provides very efficient and convenient way of developing network applications and user interfaces. In this research, we developed a framework for the real time sharing of STEP data and 3D conference in the ObjectWeb environment and tested our proposal by implementing prototype software which we call STEPShare.

I. 서론

인터넷과 웹 기술의 보급은 데이터의 공유와 글로벌 협력을 위한 새로운 소프트웨어 도구의 출현을 가능하게 하고 있으며 불과 2년 전인 1996년에 비전으로서 제시되던 **Web-enabled engineering**은, 다수의 개발자나 사용자가 단지 웹브라우저와 같은 최소한의 클라이언트로 제품의 개발에 참여하거나 정보를 공유할 수 있다는 이점으로 인하여, 짧은 기간동안 급속한 발전을 거듭해왔다. 이 과정에서 CAD 벤더들은 다양한 방식으로 **Web-enabled** 제품 개발을 진행시켜 오고있는데, 이러한 시도는 3D CAD 데이터를 VRML(Virtual Reality Modeling Language)로 변환하여 웹상에서 보여준 다든지 DWF(Drawing Web Format), CGM(Computer Graphics Metafile), SAT 또는 STEP 등의 포맷으로 된 데이터를 웹에 게시하여 웹 브라우저의 **plug-in** 프로그램을 통하여 브라우징할수 있게 하는 방식, 클라이언트 요구에 따라 현재의 데이터베이스로부터 검색한 결과에 대한 HTML 문서를 생성하여 전달하는 방식 등으로 진행되었으며, 추후 웹을 통한 설계와 해석등 공동 작업을 지원할 수 있도록 하는 프로젝트도 현재 진행중에 있다.

이처럼 웹을 통한 엔지니어링은 현재 다양한 접근 방식으로 시도중이지만, 차세대의 **Web-enabled engineering**은 사업자간이나 사업자와 개발자 또는 사용자간의 트랜잭션에서 생기는 다중계층의 복잡한 요구를 수용할 수 있어야 하며, 이를 위해서 클라이언트와 서버간 풍부한 인터랙션의 중계가 가능해야 한다. 현재 널리 사용되고 있는 HTTP/CGI 패러다임은 쿼리전송-실행-결과전달의 불연속적인 동작 방식에 의존하므로 이러한 요구 사양을 해결하기에는 부적절하며, 이에 대한 해

결책으로 네트워크를 통해 프로그램 코드가 이동하여 실행되는 자바(Java)와, 분산 객체 환경의 표준인 CORBA를 이용하는 "Object Web"의 사용을 생각할 수 있다.

본 연구에서는 데이터의 표현 방법으로써 STEP을 이용하고 시스템의 구조로써 CORBA를, 구현 방법과 사용자 인터페이스로써 자바를 사용하여 분산 CAD 환경에서 제품 모델의 형상과 비형상 정보를 공유하고, 뷰 공유와 마크업, 의견 교환을 위한 채팅을 가능하도록 하는 3D 컨퍼런싱 시스템인 STEPShare를 개발하였다.

STEPShare는 클라이언트/서버의 구조로 이루어져 있으며, 제품 데이터와 이를 다루기 위한 시스템은 서버쪽에 위치하지만, 사용자 인터페이스를 제공하는 클라이언트 부분은 자바 애플릿으로 작성되어 이 애플릿을 포함하고 있는 웹문서를 사용자가 웹 브라우저를 통해 읽어 들이면 사용자의 컴퓨터로 실시간 다운로드되어 실행된다. 즉 사용자 측에서는 별도의 프로그램을 설치할 필요 없이 웹 브라우저만으로 전체 시스템의 사용이 가능하다.

서버는 클라이언트의 요구에 의해 STEP 물리 파일을 읽어 들여 객체 단위로 검색하고, 클라이언트는 사용자의 요구에 따라 서버로부터 넘겨받은 형상 데이터에 대해 회전, 확대, 이동 등의 기능을 이용해 보여주는 3차원 와이어프레임 뷰를 제공하며, STEP 파일에 대한 간략한 문서 요약 정보를 보여 주거나 엔티티 트리 컨트롤을 통해 비형상정보 또한 브라우징 할 수 있는 기능을 제공한다. 사용자는 같은 세션에 있는 다른 사용자들과 제품 데이터를 공유할 수 있으며, 뷰와 마크업 데이터의 실시간 공유와 채팅을 통하여 컨퍼런스에 참여할 수 있게 된다.

II. 관련 표준 및 기술

2.1 STEP

제품 데이터 교환의 새로운 표준으로 추진되고 있는 ISO 10303(STEP)은 모든 산업에서 제품 생명 주기의 모든 측면을 다루는 중립적인 메커니즘을 제공하는 것을 목표로 하고 있다. 즉, 모든 산업에서 제품 생명 주기의 모든 측면을 다루는 우수한 표준을 하나로 만드는 것을 그 목표로 하고 있다.

STEP은 생산정보와 설계정보를 모두 포함하고 있다는 점에서 기존 IGES나, DXF 같은 기존의 표준들과 차이를 가지며, 현재 메이저 벤더들은 경쟁적으로 STEP의 표준화된 파트에 대한 입출력 기능을 추가하거나, 기존 포맷과의 변환을 위한 프로그램이나 서비스를 제공하고 있다.

STEP 표준의 근간이 되는 것은 EXPRESS 정보 모델링 언어로써, STEP 자체는 EXPRESS 언어로 작성된 스키마에 기초를 두는 광범위한 데이터베이스라 할 수 있다. EXPRESS는 객체지향적으로 설계되었으며, 타 언어로의 바인딩에 대해 명시하고 있기 때문에 정보 모델링에 기초한 응용 프로그램을 작성한다든지, 기존의 프로그램에 새로운 기술들을 첨가하는데 매우 적합하다.

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 생성

하기 위해서 통합 자원들이 해석되는데, 통합 자원의 규정적인 정보 모델은 EXPRESS 언어로 기술한다. EXPRESS 언어는 part 11에서 정의되고 있으며 실질적인 제품 데이터, 즉, STEP 물리적 파일은 part 21 "Clear Text Encoding of Exchange String" 이 정한 규정에 따라 기술된다.

2.2 CORBA

CORBA는 분산 컴퓨팅 시스템의 개발을 위한 객체 지향 표준을 제정하기 위해 700여 이상의 컴퓨터 관련 단체가 모여서 결성한 컨소시엄인 OMG가 이종의 분산된 환경 하에서 응용 프로그램들이 서로 통합될 수 있는 표준 기술로써 제안한 OMA(Object Management Architecture) 중 네트워크 시스템에서 객체간의 통신을 가능하게 해주는 아키텍처를 문서화하여 발표한 것이다. OMA는 응용 프로그램 간의 결합 뿐만 아니라 객체의 생성, 소멸에서부터 저장, 트랜잭션 기능에 이르기까지 분산객체 환경에서 필요한 모든 서비스를 총칭하는 것이며 이들 기능 중 CORBA는 서로 다른 프로그램들 사이의 버스 역할을 하는 모듈로서 OMA의 가장 중요한 요소이다. OMA가 응용 프로그램 간의 통합을 나타내는 것이라면 CORBA가 객체 간의 상호작용을 나타내는 것이라고 할 수 있다.

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

2.3 Java

자바는 객체 지향적이고, 플랫폼에 독립적이며, HTTP를 통하여 코드가 이동할 수 있는 특징으로 인하여 WWW상의 표준적인 언어로 자리잡았다. 이종 분산 환경 프로그래밍 언어의 관점에서 본다면 다음과 같은 것들이 장점으로 작용한다고 말할 수 있다.

- 단순하다: 자바 인터프리터는 단지 215KB 램만을 사용하여 동작이 가능하다. 또한 자바로 작성된 후 컴파일된 바이너리는 디버그 코드를 포함하고도 소스와 비슷한 크기이거나 더 작다. 이로 인하여 코드 자체가 네트워크를 통해 이동한다 하더라도 필요한 대역폭과 로딩시간이 짧아, 높은 효율성을 가지게 된다.
- 객체지향 언어이다: 자바는 객체 또는 컴포넌트 단위로 존재하고, 그 자체가 객체 지향으로 설계되었기 때문에 STEP이나 CORBA와 같은 다른 객체지향 구조들과의 잘 어울린다.
- 분산되어있다: 자바는 초기 디자인부터 네트워크 환경을 고려했기 때문에 TCP, UDP 등 기본 네트워크 라이브러리를 비롯해 HTTP나 FTP 같은 대표적인 인터넷 프로토콜을 지원한다. 또한 자바는 필요한 객체를 글로벌한 네트워크 환경에서 참조할 수 있다.
- 아키텍처에 중립적이다: 특정 하드웨어와 운영체제에 상관없이 수행이 가능하며 이와 같은 특징으로는 컴파일로 생성된 코드는 상위수준이자 기계 독립적인 바이트 코드이며, 생성된 바이트 코드는 플랫폼 독립적인 가상 기계(JVM; Java Virtual Machine)에서 실행되고, 문자열 처리시 유니코드 타입을 지원하는 등 기본 데이터 타입이 하드웨어 독립적이라는 점을 들 수 있다.
- 안전하다: 자바 가상 기계는 코드를 실행하기 전 코드에 대한 검사(Byte code verification)을 수행하여 안전한 코드만을 실행한다. 이로써 네트워크의 다른 곳에서 전송되어 온 자바 프로그램이 로컬 컴퓨터의 파일을 조작하거나 다른 컴퓨터에 연결하여 정보를 전송하는 등의 보안 문제를 방지

한다.

· 동적이다: 자바는 런타임 시까지 모듈들을 링크하지 않는다. 따라서 라이브러리가 변하더라도 영향을 받지 않으며 재컴파일 할 필요가 없다. 이러한 특징은 네트워크를 통하여 공유된 소프트웨어 자원의 버전관리를 매우 용이하게 한다.

2.4 Object-web

Object-web 은 종종 JavaORB 로도 불리는데, CORBA 와 Java 를 근간으로 하는 객체 지향의 분산 환경이 WWW 에서 구현된 것을 일컫는다. Fig. 1 에는 HTTP(Hyper Text Transfer Protocol)를 통하여 HTML 문서와 함께 자바 애플릿이 다운로드 되어서 클라이언트쪽에서 실행되고, 일단 실행이 된 애플릿이 IIOP(Internet Inter-Operability Protocol)를 통하여 서버와 상호 작용을 하게 되는 Object-web 의 실행 과정과 그 구조가 나타나있다.

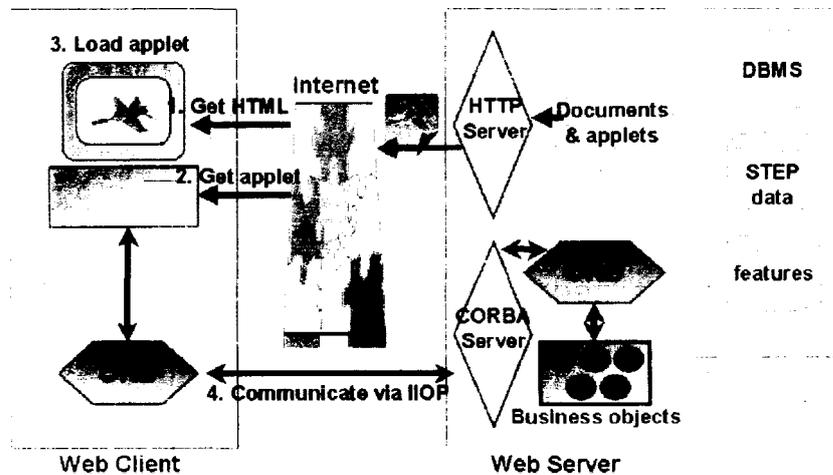


Fig 1. WWW 환경에서 Object-Web 의 작용 방식

III. 시스템 기능 및 구성

STEPShare 는 서버 사이트의 STEP 데이터의 형상 정보를 다수의 사용자가 3D 와이어프레임 디스플레이를 통해 상호 회전 또는 확대해서 살펴보는 것을 실시간으로 공유할 수 있게 하고, 비형상 정보를 구조적으로 탐색할 수 있게 하는 기능과, 마크업 데이터의 공유, 사용자간의 채팅 등을 통한 컨퍼런스를 지원하는 시스템이다. 이 때 사용자와의 인터페이스를 수행하는 부분은 클라이언트 부분으로서, 네트워크를 통하여 다운로드 된 후 사용자의 웹브라우저 안에서 실행되고, 서버와 통신하지만 사용자에게는 자신의 컴퓨터에서 하나의 응용 프로그램을 실행하는 것 처럼 네트워크에 대한 투명성을 가진다. Fig. 2 는 STEPShare 를 이용하여 컨퍼런스를 실행하는 과정을 나타낸다.

서버는 서버 호스트에 있는 STEP 데이터 파일들로부터 클라이언트에서 접근 가능한 데이터 목록을 만들고, 선택받은 STEP 물리 파일을 읽어들이어 인스턴싱한 후 STEP 의 각 엔티티에 해당하는 객체 단위의 메소드를 구현한다. 또한 경계 표현형식의 솔리드 모델로부터 와이어프레임 데이터를 추출하여 클라이언트에 넘기기도 하고, 인스턴싱된 엔티티를 구조적으로 검색할 수 있도록 하기

위해 객체간의 포함관계를 분석하여 최상위 객체들의 목록을 작성한다. 이러한 서버의 기능들과 서버 객체의 메소드들은 클라이언트측의 요구에 의해 수행된다. Fig. 3 은 서버의 세션 객체와 클라이언트 모듈간의 연결을 도시한 것이다.

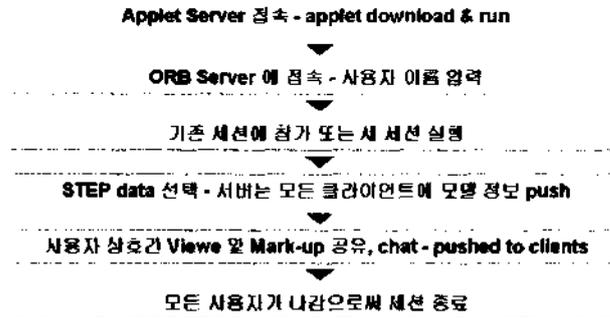


Fig. 2 STEPSHare 의 실행 시나리오

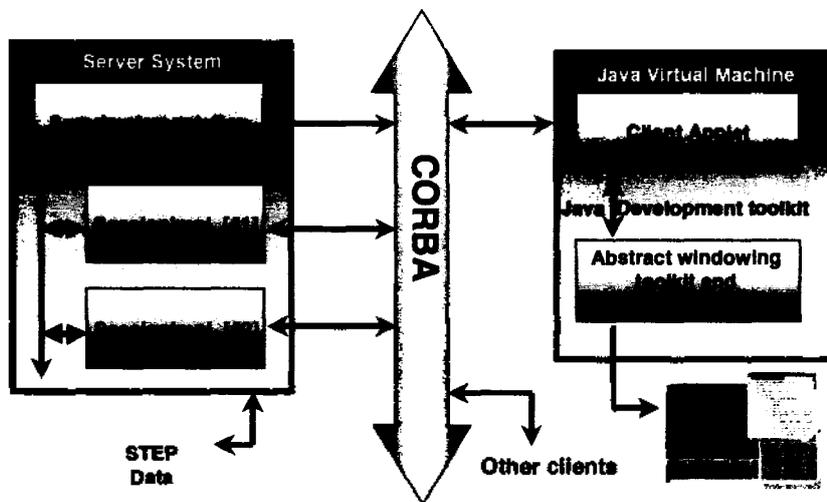


Fig. 3. Client/Sever 간 모듈 구성

Fig. 4.와 Fig. 5. 는 클라이언트의 실행 화면으로, 마크업 정보를 추가하고 채팅 기능을 보여주는 화면인데, 같은 세션의 사용자는 모두 같은 뷰와 마크업 정보를 보게 된다.

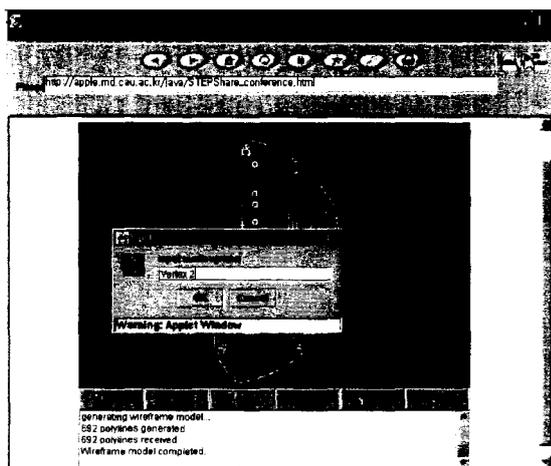


Fig. 4. Mark-up 데이터의 추가

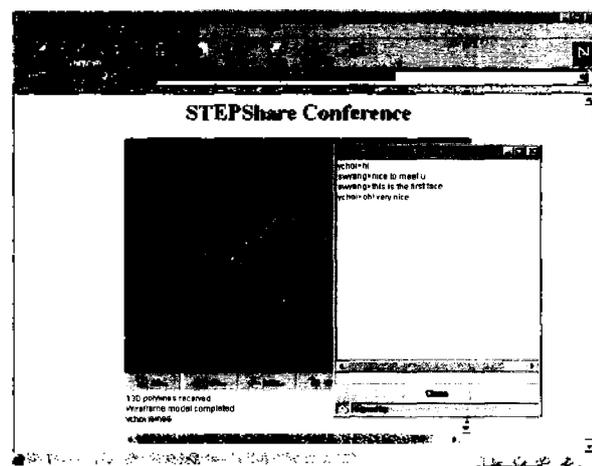


Fig.5. 채팅 기능을 이용한 의견 교환

클라이언트는 기본적으로 GUI를 통한 사용자 인터페이스와, 서버로부터 받은 데이터 리스트에서 사용자가 선택할 수 있게 하는 기능, 서버에서 전달한 받은 형상 정보 데이터를 렌, 줌, 회전 등의 대화형 인터페이스를 통해 3차원 와이어프레임으로 보여주는 기능, 데이터의 저장나 기관, 작성

일자, 작성 도구 등의 요약정보를 보여주는 기능, 트리 컨트롤을 통해 엔티티의 구조 및 속성을 보여주는 기능, 도움말을 웹 브라우저를 통해 보여 주는 기능 등을 가진다. 또한, 사용자가 현재 보고 있는 모델의 뷰를 바꾸거나, 마크업 데이터를 추가하는 등 컨퍼런스 관련 이벤트를 발생시키면 클라이언트는 이러한 데이터를 서버에 넘겨 세션의 다른 멤버들과 똑같은 관점을 유지하도록 한다.

IV. 구현 방법

STEPShare의 개발 과정은 Fig. 5와 같다. 개발과정에 사용된 도구는 서버와 클라이언트 공히 JDK(Java Development Kit) 1.1.5 버전과 OrbixWeb 2.0이다. 개발과정에서 보듯이 먼저 클라이언트 서버의 상호 작용을 위한 인터페이스를 IDL로 정의한 후 OrbixWeb에 포함된 IDL-to-Java 컴파일러를 이용하여 자바로 된 스티브와 스케레톤을 생성하여 이들과 CORBA API를 이용하여 양측의 프로그램을 작성하였다.

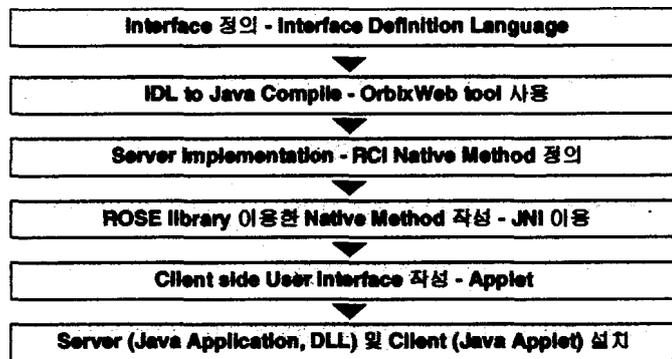


Fig. 5 STEPShare 개발 과정

서버쪽에서 STEP 데이터를 다루기 위해서 인터페이스로 정의된 객체의 구현 부분을 먼저 작성하고, STEP 파일의 처리 등 현재 자바로 구현하기 힘든 부분은 C++ 네이티브 메소드로 구현하기 위한 JNI(Java Native Interface) 스티브를 생성하여 STEP Tools사의 ST-Developer에 포함된 Rose 라이브러리를 사용하는 코어와 연결된다. 이 때 네이티브 메소드의 컴파일에는 Microsoft Visual C++ 4.2가 사용되며 이는 JNI를 이용하여 자바로 작성된 오브젝트 구현 부분과 연결된다. 따라서, 서버의 개발에는 JDK 1.1.5, OrbixWeb 2.0, ST-Developer 1.5와 MS VC++ 4.2가 사용되었으며, 이는 MS WindowsNT 4.0을 운용하는 펜티엄 PC에서 이루어졌다.

실행환경에 있어서 클라이언트는 자바 애플릿의 형태로 되어 있으므로 사용자의 컴퓨터에는 웹 브라우저외의 어떠한 프로그램도 설치 되어 있을 필요가 없으며 어떤 기종에 어떤 운영체제를 사용하고 있는지 상관없다. 단지 웹 브라우저를 통해 STEPShare의 클라이언트 애플릿을 포함하는 웹사이트에 접속하기만 하면 된다.

STEPShare에서 사용되는 서버는 WWW 서버, ORB 서버, 객체 구현(object implementation)등 세 가지 범주로 생각될 수 있다. WWW 서버는 STEPShare 클라이언트를 포함하는 HTML 문서를 제공하는 웹 서버이고, ORB 서버는 STEPShare가 사용하는 ORB인 OrbixWeb 데몬이 실행되고 있는 호스트이다. 또한 객체 구현 부분은 개발된 STEPShare의 서버 실행파일에 의해 제공된다. 개념적으로는 이 세 가지 서버가 서로 다른 호스트에 있을 수 있으나 STEPShare에서 아직 보안에 대한 기능이 마련되어 있지 않은 상태이기 때문에 하나의 호스트에서 세 가지 서비스를 구동한다.

V. 결론 및 향후 연구 과제

제품의 전주기 데이터를 표현하기 위한 국제표준인 STEP 과 분산 컴퓨팅 환경에서 객체간의 풍부한 상호 작용을 위한 CORBA, WWW 상의 표준 프로그래밍 언어로 자리잡아 가고 있는 자바 등과 같이, STEPShare 의 개발에 있어서 이슈가 된 이 세 가지 기술들은 객체 지향과 플랫폼에 독립적인 성격에서 공통점을 가지고 있으며, 각기 여러 산업 분야에서 정보 인프라 구축에 중심적인 역할을 할 수 있을것으로 기대되고 있다. 이러한 새로운 기술들을 CAD 환경에 어떻게 실질적인 방법으로 적용을 해볼 수 있을 것인가 하는 문제에 대한 답을 얻기 위한 노력의 한 방편으로 STEPShare 를 개발 하였다.

개발된 STEPShare 는 플랫폼에 독립적이고, 사용을 위하여 복잡한 설치과정을 필요로 하지 않으며 완전한 객체지향의 개념으로 작성되어 그 컴포넌트들은 다시 컴파일하는 일 없이 타 프로그램의 구성 요소로 사용될 수 있다. 또한 실행 시간동안 분산 객체간의 풍부한 상호 작용으로 엔드 유저의 측면에서는, 데이터의 이동에 소요되는 시간을 무시한다면, 로컬 머신에서 실행하는 프로그램처럼 작동한다.

현재 STEPShare 는 STEP 의 PART 들 중 AP203 만을 대상으로 하고 있으나 서버 쪽에서 다룰 수 있는 범위를 넓힘으로써 그 처리 범위를 확장할 것이다. 또한 단지 데이터를 보여 주는 기능에서 벗어나 데이터 서버쪽의 데이터를 편집하거나 다운로드 받을 수 있는 기능, 채팅룸과 같이 세션을 만들어 세션의 참가자들이 뷰나 엔티티 브라우저를 공유하는 방법으로 다른 사용자와 동시에 함께 작업할 수 있는 기능의 추가도 생각할 수 있다. 클라이언트와 서버 객체의 관계를 정의하는 인터페이스 부분은 현재 자체적으로 마련한 기준에 의해 작성되었지만 현재 제정중인 STEP part 26, SDAI-IDL 의 지원에 대한 연구를 통해 표준화 하여야 할 것이다. 또한 앞으로는 OMG 의 MDTF (Manufacturing Domain Task Force)에서 추진중인 PDM 인터페이스를 채택함으로써 CORBA 의 수직적 기능(vertical facility)으로 사용될 수 있어야 할 것이다.

활용 방안으로써는 인터넷/인트라넷을 이용한 PDM 구축을 우선 생각해 볼수 있으며, 컨퍼런스 기능 뿐 아니라 동시 협력 설계 시스템 등에 응용할 수 있다

참고문헌

1. W. C. Regli, "Internet-Enabled Computer-Aided Design", <http://www.computer.org/internet/9701/regli9701.html>
2. Caren Potter, "Web-Enabled Engineering: step-by-step", Computer Graphics World, pp. 64-69. Nov., 1997
3. 김철영외, "웹과 STEP 을 이용한 제품 설계 공유 정보 시스템", 한국 CAD/CAM 학회 논문집 제 1 권 제 3 호, pp. 203-214, 1996. 12 월
4. Y. Choi and H.Y. Shin "Sharing STEP data on the Internet" Proceedings of the 8th International Conference on Production Engineering, pp. 397-405, Sapporo, Japan, Aug. 18-20, 1997
5. 정운용외, "인터넷에서 STEP 을 이용한 자동차 PDM 과 형상 가시화", '97 한국 CAD/CAM 학회 학술발표회 논문집, pp. 131-139, 1997. 2. 14
6. R. Orfali et al., "CORBA, Java, and the Object Web", Byte magazine, <http://www.byte.com/art/9710/sec6/art3.htm>, Oct., 1997

7. Jon Siegel, CORBA fundamentals and programming, John Wiley & Sons, Inc., 1996
8. E. Evans and D. Rogers, "Using Java Applets and CORBA for Multi-user Distributed Applications", IEEE Internet Computing, pp. 43-55, Jun., 1997
9. STEP 연구회, 제품 모델 정보 교환을 위한 국제 표준 (ISO 10303) 스텝, 성안당, 1996
10. 박재현, "인터넷 애플리케이션 개발환경", 월간 마이크로 소프트웨어, pp. 464-472, 1996. 6.
11. Andres Vogel and Keith Duddy, "Java Programming with CORBA", John Wiley & Sons, Inc., 1997
12. Michael Morrison, et. al. "Java UNLEASHED", Sams.net Publishing, 1996
13. ISO 10303-203 - Part 203: Application Protocol: Configuration controlled design, ISO, 1994
14. Orbix 2 programming & reference guide, IONA Ltd. 1995
15. Beth Stearns, "Using the JNI to Integrate Native Code and Java Programs", <http://java.sun.com:81/docs/books/tutorial/native1.1/index.html>
16. Sun Microsystems, "Using the JFC/Swing Packages", <http://java.sun.com/docs/books/tutorial/ui/swing/index.html>
17. STEP Tools Inc., "STEP Programmer's Toolkit - ROSE Library reference manual", 1996