

SOAP-EJB 기반 협업 지원 시스템의 설계 및 구현

김형선

한국전자통신연구원 컴퓨터, 소프트웨어연구소

A Design and Implementation of SOAP-EJB Based Collaboration Supporting System

Hyoung Sun Kim

ETRI/CSTL

A collaboration supporting system, which enables multi-user collaboration and communication in Internet environment, is being a new indispensable tool to compete and collaborate with business peers. A collaboration supporting system in distributed environment requires scalable and open system architecture to rapidly deploy a new technology and trends. In this paper, we propose such a collaboration supporting system where communication API is using XML(Extended Markup Language)-based SOAP(Simple Object Access Protocol)protocols, which is independent on programming language and platform. In our system, server components are developed with EJB(Enterprise Java Beans). System analysis, design, and development are based on UML(Unified Modeling Language).

Keywords : collaboration, SOAP, XML, EJB, Internet, UML

1. 서론

오늘날 인터넷과 웹의 발전은 사회생활 전반에 걸쳐 커다란 영향을 끼치고 있고, 이러한 상황에서 웹 기반 협업 지원시스템의 개발이 필수적으로 확산되면서 대규모의 웹 트랜잭션을 안정적으로 처리해 줄 수 있는 기본 플랫폼의 필요성이 절실히 요구되고 있다.

원격지간 다수의 참여자가 제품의 설계·개발의 복잡한 업무처리 과정을 처리하고 인터넷상에서 다양한 형태의 문서·이미지·도면을 공유할 수 있게 하고, 또한 도면 수정을 위한 마크업, 화상회의를 통한 정확한 의사전달을 통해 비즈니스의 생산성을 대폭 향상시킬 수 있다.

분산환경 하에서 온라인을 통한 협업과정에서 데이터의 전송한계를 극복하고, 각종 설계도면, 동영상, CAD 데이터 등 기반으로 한 메카트로닉스 분야는 물론 제조

업 등 산업전반에 걸쳐 제품개발의 생산성을 향상하기 위한 협업을 지원할 수 있는 환경을 제공하여야 한다[3].

이러한 요구를 만족시켜 줄 수 있는 것들 중의 하나로 분산 컴포넌트 트랜잭션 서비스가 있는데, 첫째로 SOAP은 언어와 플랫폼에 관계없이 클라이언트와 서버간의 데이터 교환을 지원하는 표준 프로토콜이며[13], 엔터프라이즈 자바빈즈(Enterprise JavaBeans; EJB)[12]는 분산 컴포넌트 트랜잭션 서비스를 지원하기 위한 새로운 플랫폼으로써, 기존에는 컴포넌트로 만들어주지 않아도 개발자들이 일일이 신경 써야 많은 기능들을 하부 구조에서 지원해주어, 개발자는 시스템 로직에 신경 쓰지 않으면서 더 쉽고 빠르게 동일한 기능을 구현할 수 있게 한다[3].

본 논문에서는 이러한 EJB와 SOAP을 이용하여 인터넷 기반 협업 지원 시스템의 설계방향을 제시하고, 시스

템의 분석 설계는 UML(Unified Modeling Language) 기반으로 설명한다.

이 논문의 구성은 다음과 같이 구성되는데, 2장에서는 SOAP과 EJB의 전반적인 개요에 대하여 살펴보고, 3장에서는 협업 지원 시스템을 구현하기 위하여 전체적인 시스템의 프레임워크를 제시하고, 4장에서는 UML을 이용한 협업 지원 시스템의 설계 및 구현을 하고, 5장에서 결론을 맺는다.

2. SOAP과 EJB의 개요

2.1 SOAP의 개요

SOAP은 CORBA와 같이 ORB(Object Request Broker)의 통신 프로토콜로 XML을 사용하는 것으로 객체에 대해 원격지에서 정해진 XML 규약에 맞추어 보내면 그 컴포넌트가 지시에 맞추어 동작하게 된다. 이와 같은 일은 원래 분산객체 미들웨어인 CORBA나 DCOM에서도 가능했던 것이지만 SOAP은 XML을 기반으로 하여 분산환경에서 분산객체를 위하여 정보를 교환하기 위한 텍스트 기반 프로토콜이다[11]. 또한 SOAP은 메시지에 있는 사항과 그 사항을 처리하는 방법을 설명하기 위해 프레임워크를 정의한 봉투, 응용 프로그램에서 정의한 데이터 형식의 인스턴스를 나타내는 일련의 인코딩 규칙, 원격 프로시저 호출 및 응답을 나타내는 규칙으로 구성된 XML 기반의 프로토콜이다[13]. SOAP 메시지는 XML을 이용하여 인코드(encode) 되어 있고 SOAP 규격은 추상화된 다른 계층으로 나타낸다. 클라이언트응용 프로그램(웹브라우저)으로부터 파생된 SOAP 메시지는 SOAP Requests로 서버에 요구하고, 서버가 접수를 받고 요청을 처리한 후에, SOAP Response는 클라이언트로 반환된다. 이 프로세스는 아래의 <그림 1>과 같이 나타난다. <그림 1>은 전형적인 SOAP 메시지의 흐름을 그린 것이며, 서버는 Listening 장치로서 역할을 하

고, SOAP의 요청을 찾고 있다. 여기서는 한 개의 서버만 있지만, Listening 서버는 실행을 위해서 다른 서버로 클라이언트의 요청을 할 수 있다[5]. 이 모델이 작동하면, 클라이언트와 서버 모두 어떻게 XML을 분석해야 하하고 식별하는지를 알아야 한다.

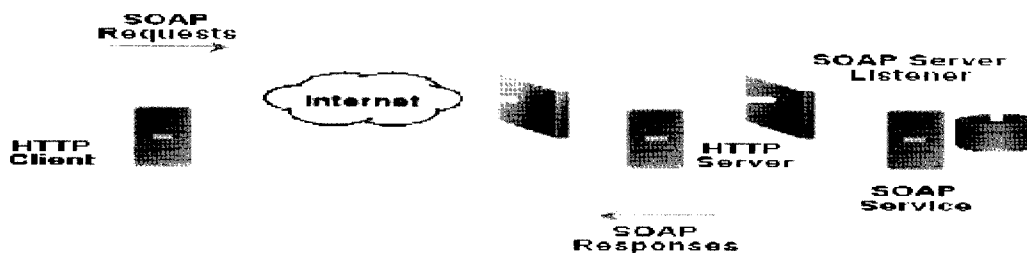
XML-speak에서 SOAP 메시지는 Envelope(봉투)라고 불리는 Root(루트) 요소를 이루고, 루트 노드는 Header(머리부분) 요소를 가지며, 루트 노드의 안쪽에서 Body(몸체) 요소로 구성되는 것이 필수적이다. 전형적으로, 이런 요소들 각각은 XML Namespace로 첫머리에 놓이게 되며, SOAP 규격은 우리에게 이런 Namespace Identifier(식별자)를 사용하고, SOAP-ENV의 앞에 놓일 수 있도록 해준다. Namespace는 우리에게 즉시 SOAP 메시지 즉, 우리가 프로토콜을 요구(Requirement) 하면 메소드와 데이터 콘텐츠 같은 것들에서 SOAP 요소들을 즉시 구별할 수 있도록 해주는 매우 유용한 것이다[5].

클라이언트와 서버의 커뮤니케이션은 HTTP(Hypertext Transfer Protocol)를 통해서 이루어지고, 다른 전송 메커니즘인 소켓(socket)이나 메세지 큐, E-mail 과 같은 것으로 사용될 수도 있다. 전형적인 SOAP 메시지의 예는 다음과 같다.

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV=
"http://schemas.xmlsoap.org/soap/envelope">
  <SOAP-ENV:Body>
    <m:getEmployeeDetail xmlns:m=
"http://www.craigmurphy.com/hrweb">
      <emp_no>5</emp_no></m:getEmployeeDetail>
    </SOAP-ENV:Body>
  </SOAP-ENV:Envelope>
```

2.2 EJB의 개요

EJB는 서버 상의 Java 컴포넌트와 그 컴포넌트를 위한 동작 환경을 제공하는 전체 프레임워크(Framework)이다.



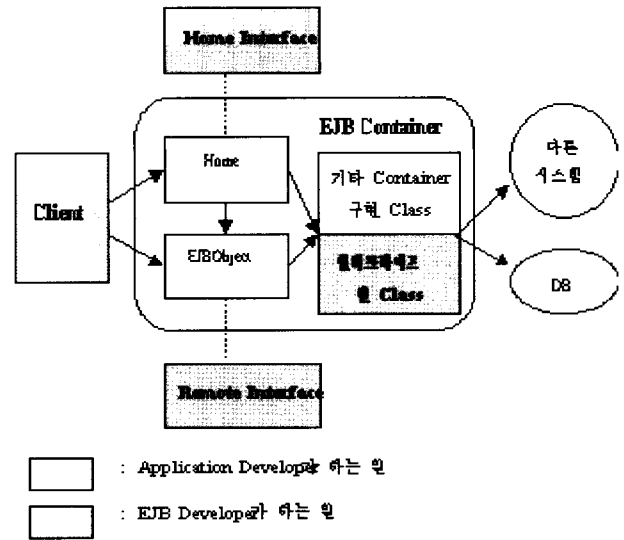
<그림 1 > SOAP 메시지 구조

EJB에서 동작하는 컴포넌트를 엔터프라이즈 빈(Enterprise Bean)이라고 하고, 그 컴포넌트가 동작하기 위한 환경을 컨테이너(Container)라 한다. EJB는 CORBA가 언어에 독립적인 특성을 가지고 있는데 반해 Java로만 구현되어 언어 독립성의 장점을 잃지만, EJB에서는 기존에 개발자가 직접 컴포넌트로 제공해 주어야 하는 데이터베이스 트랜잭션, 보안문제, 데이터베이스 연결 풀링, 쓰레딩 관리 같은 기능을 컨테이너에서 처리해 줌으로 일반 개발자는 시스템 로직에 신경 쓰지 않으면서 더 쉽고 빠르게 동일한 기능을 구현할 수 있게 된다[3].

2.2.1. EJB의 컴포넌트

EJB는 세션 빈(Session Bean)과 엔터티 빈(Entity Bean)이라는 두 가지 형태의 엔터프라이즈 빈을 제공한다. 세션 빈은 CORBA 등에서 말하는 컴포넌트와 같은 개념이며, 일반 비즈니스 로직을 같는다. 세션 빈은 데이터베이스와 같은 지속적인 저장장치에 연결되는 빈이 아니기 때문에 컨테이너가 재시작되면 이전에 있던 세션 빈은 사라진다. 반면에 엔터티 빈은 데이터베이스의 데이터를 의미하고 있으며, 간단한 형태의 엔터티 빈은 데이터베이스 테이블의 한 레코드를 나타낸다. 그리고, 복잡한 형태의 엔터티 빈은 특정한 테이블 간의 조인(Join)된 형태를 나타낼 수도 있다[9]. 엔터티 빈은 데이터베이스의 레코드에 연결되기 때문에 컨테이너가 재시작되더라도 이전에 작업했던 내용을 그대로 유지한다. 이 엔터티 빈은 CORBA와 같은 경우에는 개발자가 일일이 만들어 주어야 하는 컴포넌트였으며, EJB의 가장 큰 특징 중의 하나라고 할 수 있다. 컨테이너는 하나의 엔터티 빈에 대한 여러 사용자의 동시 접근을 허용하고, 이때 발생하는 동기화 처리 등을 알아서 해결해 준다[8]. 또한, 엔터티 빈은 세션 빈과 달리 특정 데이터를 의미하기 때문에 지속적으로 보존하여야 할 필요가 있다. EJB 명세는 이를 위한 두 가지 방법을 제공하고 있는데 그것이 컨테이너 관리 지속성 방법(Container Managed Persistence)와 빈 관리 지속성 방법 (Bean Managed Persistence)이다. 컨테이너 관리 지속성은 전개 문서(Deployment Descriptor) 상에 보존하여야 할 대상을 지정하면 개발자의 추가적인 작업 없이 Bean의 보존이 이루어진다. 빈 관리 지속성의 경우에는 엔터티 빈의 저장에 관련된 일련의 코딩 작업을 빈상에 개발자가 직접 수행하여 한다. 개발자의 입장에서 보면 컨테이너 관리 지속성이 편리하나 복잡한 질의 문이나 개발자가 Bean에 대한 강한 핸들링 권한이 필요한 경우에는 빈 관리 지속성 방법이 사용된다.

엔터프라이즈 빈 개발자 입장에서 보면 각 빈들은 <그림 2>와 같이 EJB Home, EJB Object, EJB 클래스



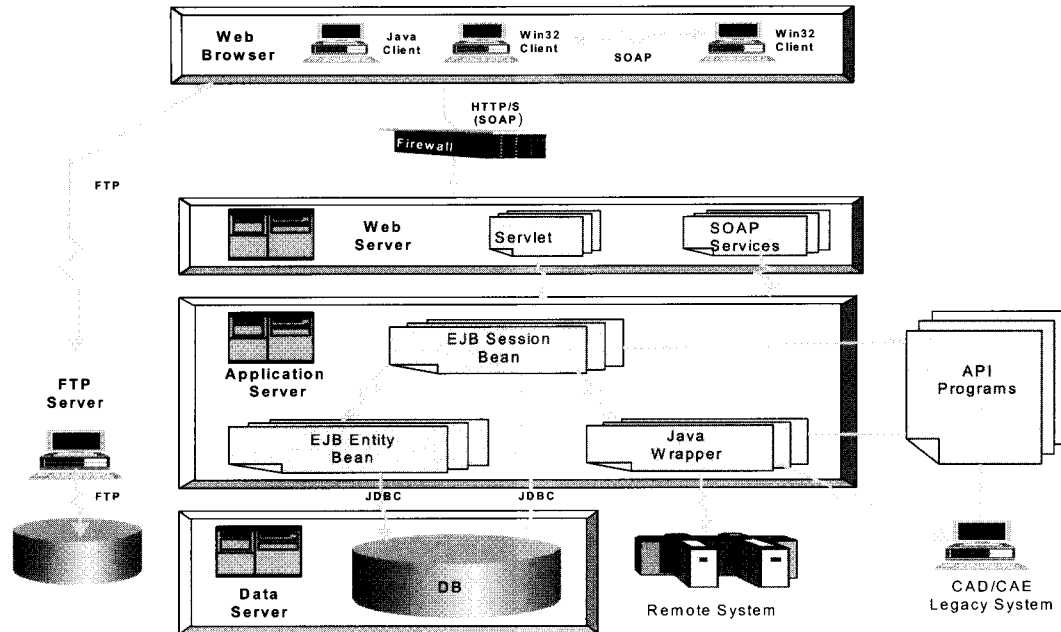
<그림 2> EJB 컴포넌트의 동작

만을 고려하면 된다. 여기서 EJB Home과 EJBObject는 각각 Home 인터페이스(Interface)와 리모트 인터페이스를 구현한 오브젝트이며[7,12], 실제 비즈니스 로직은 엔터프라이즈 빈 Class에서 구현한다. Home 인터페이스와 Home 오브젝트는 엔터프라이즈 빈 Class를 생성하고 제거하는 Factory 메소드를 열거한다. 즉, Home 인터페이스는 클라이언트들이 요구하는 Bean의 인스턴스(Instance)를 생성할 수 있는 하나 혹은 그 이상의 create() 메소드들을 내재한다. 또한, 엔터티빈은 데이터베이스의 레코드와 연계되기 때문에 Finder 메소드를 가지고 있는데 이를 이용하여 클라이언트는 특정한 엔터티 빈을 접근할 수 있다[7].

리모트 인터페이스는 빈 클래스의 비즈니스 메소드를 나열하고 있으며, 이의 구현인 EJB 오브젝트는 Bean의 대리자(Proxy)역할을 수행하는 네트워크상에 보여지는 오브젝트로서, 클라이언트가 EJB 인스턴스의 비즈니스 로직을 접근하기 위해서는 이 EJB 오브젝트를 사용해야 한다. 즉, 클라이언트는 EJB 인스턴스의 참조를 얻는 것이 아니라 단지 EJB 오브젝트의 인스턴스를 얻는다. 클라이언트가 메소드를 호출하면 EJB 오브젝트가 받아들여 그것을 EJB 인스턴스에 전달하여 실제 로직을 처리한다.

3. 협업 지원시스템의 하부구조

인터넷 기반 협업 지원시스템이란 분야마다 정의가 다르겠지만 건설분야에서 생각하여 볼 때 인터넷을 통해 건설현장과 발주, 설계, 시공, 감리 업체들을 연결하



<그림 3> 협업 지원 시스템의 하부구조

는 네트워크를 구축, 건설현장에서 일어나는 자재 조달과 인원 출입, 공구별 작업 진척도 등을 실시간으로 파악해 프로젝트에 관계돼 있는 사람들간의 회의를 실시간으로 진행할 수 있게 한 시스템이다.

이러한 협업 지원 시스템은 서비스를 제공받는 고객(clients), 서비스 제공자(service providers) 및 이들을 연계하는 서비스 중개자(service broker)로 구성될 수 있다. 고객은 데이터를 생성하여 이를 서버에 등록하기 위하여 인터넷을 통해 서비스 중개자에게 작업을 요청하고, 서비스 중개자는 역시 인터넷을 통해 서비스 제공자로 하여금 이 작업을 수행할 수 있도록 하며, 작업의 결과를 고객에게 제공한다[1].

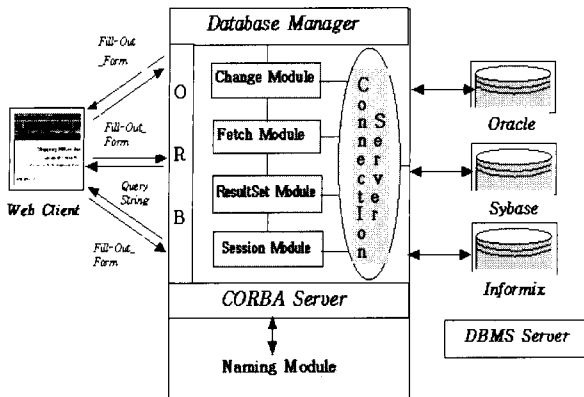
<그림 3>은 본 논문에서 제시하는 협업 지원 시스템의 개념적 구조를 보여준다. 제시된 구조는 웹(World Wide Web)의 강건한 통신환경 위에서 인터넷 컴퓨팅 기술과 분산객체기술을 이용한 클라이언트/서버 방식의 개방형 구조를 갖고 있다. 서버 측은 엔터프라이즈 자바빈즈(EJB) 구조를 가지는 컴포넌트로 개발하였으며, 클라이언트 측 응용프로그램은 마이크로소프트사의 운영환경하에서 비주얼 베이직으로 개발하였다. 특히 제안된 구조의 특이사항은 서버 쪽의 소스 프로그램을 전혀 수정하지 않고도 클라이언트 프로그램은 어떤 언어로도 구현할 수 있는 매우 큰 장점이 있다.

서버 측은 고객이 원하는 서비스를 유효하게 서비스 받을 수 있도록 지원하는 객체로써, 서비스 관리자(service manager), 데이터베이스 관리자(Database manager)[4], 통신 관리자(communication manager) 등

으로 구성된다[2]. 또한 서비스 제공자에 의해 제공되는 Legacy Application을 통합하기 위한 래퍼(Wrapper)를 포함한다. 클라이언트와 서버, 서버간의 통신은 XML SOAP[11,13] 프로토콜을 이용함으로써 지리적으로 분산된 이질적인 컴퓨팅 환경을 갖는 고객들이 웹을 기반으로 서비스를 제공받을 수 있도록 하였다.

3.1 서비스 관리자

서비스 제공자에 의해 제공될 수 있는 도구들은 다양한 플랫폼(platform)에서 다양한 형태로 존재한다. 또한 원시코드가 존재하는 응용 프로그램, 실행파일과 함께 API가 제공되는 소프트웨어, 실행파일만으로 수행되는 소프트웨어 등이 있다. 데이터의 입출력 역시 입력 파라메타(parameter)를 받아서 작업을 수행하고 출력 파라메타를 생성하는 응용 프로그램, 정형화된 입력 파일을 받아서 작업을 수행하고 그 결과를 파일 형태로 출력해주는 소프트웨어, GUI(Graphic User Interface)를 통해서 사용자가 직접 작업을 수행해야만 결과를 출력할 수 있는 소프트웨어 등 다양한 형태로 사용자에게 제공될 수 있다. 서비스 관리자는 이러한 여러 다양한 형태의 도구들을 분산환경에서 통합적으로 관리하여 고객에게 필요한 협업 지원서비스를 제공하는 역할을 담당한다. 서비스 제공자는 자신이 서비스할 수 있는 서비스 객체를 서비스 관리자에 등록을 함으로써 자신이 수행할 수 있는 작업이 무엇인지를 고객들에게 알릴 수 있다. 또한 고객은 서비스 관리자를 통해서 사용 가능한 서비스 객



<그림 4> 데이터베이스 관리자 구조

제가 무엇이 있으며, 어떠한 작업이 가능한 지를 알 수 있어 자신이 필요로 하는 서비스를 분산 환경에서 제공 받을 수 있다[2]. 등록된 협업 서비스 객체들은 서비스 관리자의 서비스 객체 저장소에 객체 형태로 저장되며, 추가 및 삭제가 가능하다. 서비스 관리자는 고객들이 현재 등록되어 있는 서비스 객체에 대한 정보를 요구할 때 이에 대한 정보를 제공할 수 있는 함수를 제공해준다. 또한, 서비스 관리자에는 등록된 서비스 객체가 가지고 있는 함수를 호출하기 위해 XML Message를 SOAP 프로토콜로 전달하여 사용자가 자신이 원하는 서비스 객체와 함수를 선택하여 실행시킬 수 있다.

3.2 데이터베이스 관리자

데이터베이스 관리자는 협업 지원시스템에 참조되는 다양한 정보를 고객과 협업 지원 서비스 객체에게 제공하는 역할을 담당한다. 협업 지원 시스템에서 제공되는 모든 정보는 데이터 베이스관리자를 통해 서비스되며, 서비스 중에 새롭게 생성되는 정보와 변경된 정보 역시 데이터베이스 관리자를 통해 관리된다. 즉 고객은 데이터베이스 관리자를 통해서 표준 정보들을 참조할 수 있으며 서비스 객체들의 수행 결과를 받아볼 수 있다. 협업 지원시스템에서 사용되는 프로젝트 정보, 문서정보, 시큐리티 정보, 사용자 객체정보 등의 정보를 제공하는 객체들은 데이터베이스 관리자에 등록되어 사용자들에게 제공되며, 서비스 객체들의 수행 결과에 대한 데이터 역시 메타 데이터(meta data) 형태로 데이터 관리자에 등록되어 서비스된다[4].

<그림 4>는 데이터베이스 관리자를 구성하는 모듈과 각 모듈과의 관계를 나타내고 있으며, 데이터베이스 관리자가 <그림 2>의 협업 지원시스템의 하부구조의 EJB Entity Bean에서 처리되는 각종 데이터베이스에 상호 연동되는 관계를 보여주고 있다.

3.3 통신 관리자

통신 관리자는 고객과 서비스 제공자 또는 관련 고객들 간에 대상이 되는 협업 지원시스템에서 사용되는 모든 정보를 공유하고 문자 또는 마크업 정보를 전송하면서 실시간 대화를 할 수 있는 기능을 제공한다. 통신 관리자는 대화를 위한 세션 객체의 생성과 종료, 사용자 로그인/로그아웃 등의 함수와 문자, 마크업 및 대상 모델을 가리키는 객체를 전송하는 함수를 EJB Session Bean에서 SOAP 서비스를 통해 세션 참가자들에게 제공한다. 대화의 대상이 되는 모델은 마크업이나 메시지 등의 정보가 부가되어 세션 참가자에게 전송됨으로써 통신 서비스를 하게 된다. 대상 모델의 전송은 해당 모델의 객체를 전송하여 수신한 세션 참가자가 전송된 모델을 가시화 함으로써 이루어진다.

4. 협업 지원시스템의 설계 및 구현

4.1 협업 지원시스템의 시나리오

본 논문에서의 제안한 협업 지원시스템은 실시간 회의 자료관리, 사용자관리, 메시지 관리, 화이트보드 관리, 리거시 시스템과의 연계관리, 텍스트 채팅, 오디오/비디오 컨퍼런싱, 참가자 초대 등 여러 기능들이 있으나 이중 가장 핵심이 되는 실시간 회의에 대하여 집중적으로 논의하고자 하며, 실시간 회의의 시나리오를 구상해보면 다음과 같고 이것을 Use Case Diagram [10,14]으로 도시한 것은 <그림 5>와 같다.

(1) 사용자가 아이디, 패스워드 등 자신의 정보를 서버로 보내고 서버는 User Access Verification을 수행한다. Verification이 성공적인 경우 서버는 현재 클라이언트가 갖고 있는 소프트웨어 정보를 이용하여 이를 업데이트할 것인지를 결정하고 업데이트가 필요 시에는 FTP를 통해 해당 파일을 받고 클라이언트 시스템 환경 정보를 갱신한다.

(2) 서버는 로그 온 한 사용자의 캐쉬 정보를 읽어서 변경된 부분을 체크한 후에 변경된 부분에 대한 데이터를 보낸다.

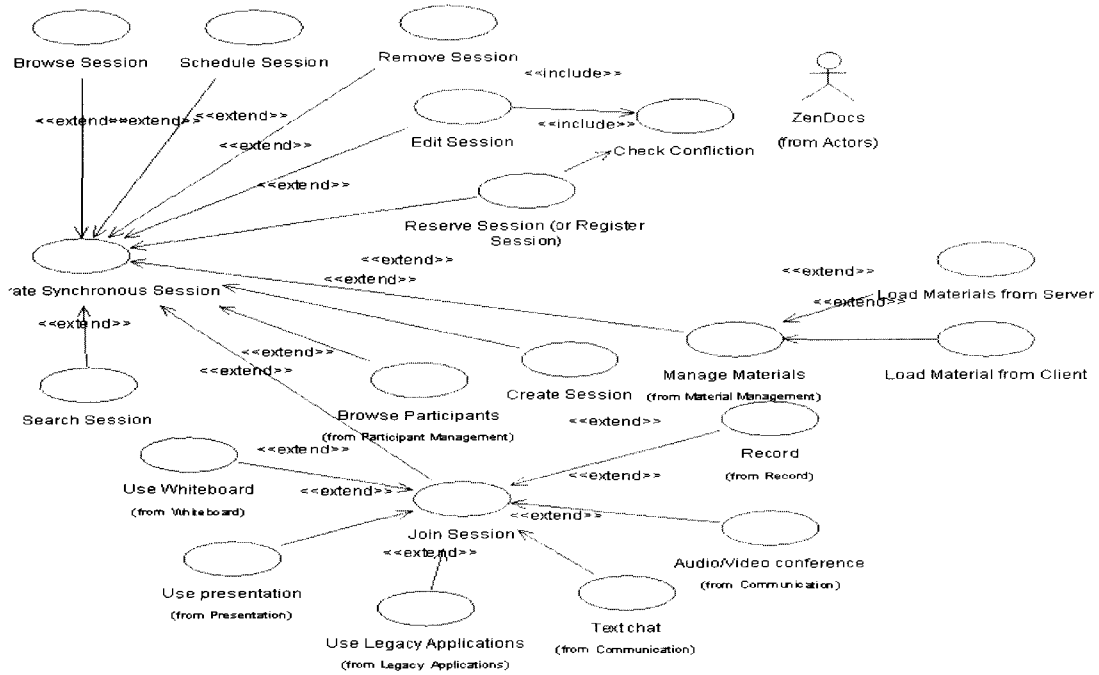
(3) 로그인과 함께 자신에게 온 메시지들을 서버로부터 받아오고, 자신을 액티브 유저(ActiveUser)로 서버에 등록한다.

(4) 자신의 친구(Friends)와 그들의 로그 온 상태를 서버로부터 받아온다.

(5) 현재 로그인 되어있는 모든 친구에게 자신이 로그인 했다는 정보를 메시지를 통해 보낸다.

(6) 실시간 컨퍼런스를 위하여 스케줄(Schedule)을 예

Use Case Diagram: Synchronous Session / Main



<그림 5> 협업 지원 시스템의 실시간 회의 Use Case Diagram

약하고 등록한다.

(7) 실시간 컨퍼런스를 위한 자료(Material)를 준비하고 등록한다.

(8) 실시간 컨퍼런스에 참가할 참가자(Participants)를 선택한다.

(9) 실시간 컨퍼런스를 위한 예약작업(Schedule, materials, participants 등)이 끝난 후 이 정보를 BBS나 웹 페이지에 올리거나 참가자들에게 메일을 보낸다.

(10) 실시간 컨퍼런스를 위해 실시간으로 참가자(기존 예약된 참가자 또는 새로운 참가자)를 초대한다

(11) 초대된 참가자(Participants)들이 실시간 컨퍼런스 세션에(Session)에 조인(Join)한다.

(12) 실시간 컨퍼런스를 위해 필요한 자료(Materials)를 화이트보드(Workspace)로 로딩>Loading) 한다.

(13) 화이트보드 상에서 텍스트 채팅, 오디오/비디오 컨퍼런스, 모델 뷰, 모델 마크 업 등으로 실시간 컨퍼런스를 진행한다.

(14) 실시간 컨퍼런스를 레코딩 하고 레포팅 한다.

4.2 협업 지원시스템에서 빈(Been)의 구성

협업 지원시스템 중에서 실시간 컨퍼런스를 진행하기 위한 필요한 EJB 빈들은 여러 가지 종류가 있으나 모두 소개할 수는 없고, 본 논문에서는 핵심 빈들만 소개하고자 한다. 실시간 회의를 하기 위하여 회의 스케줄을 예

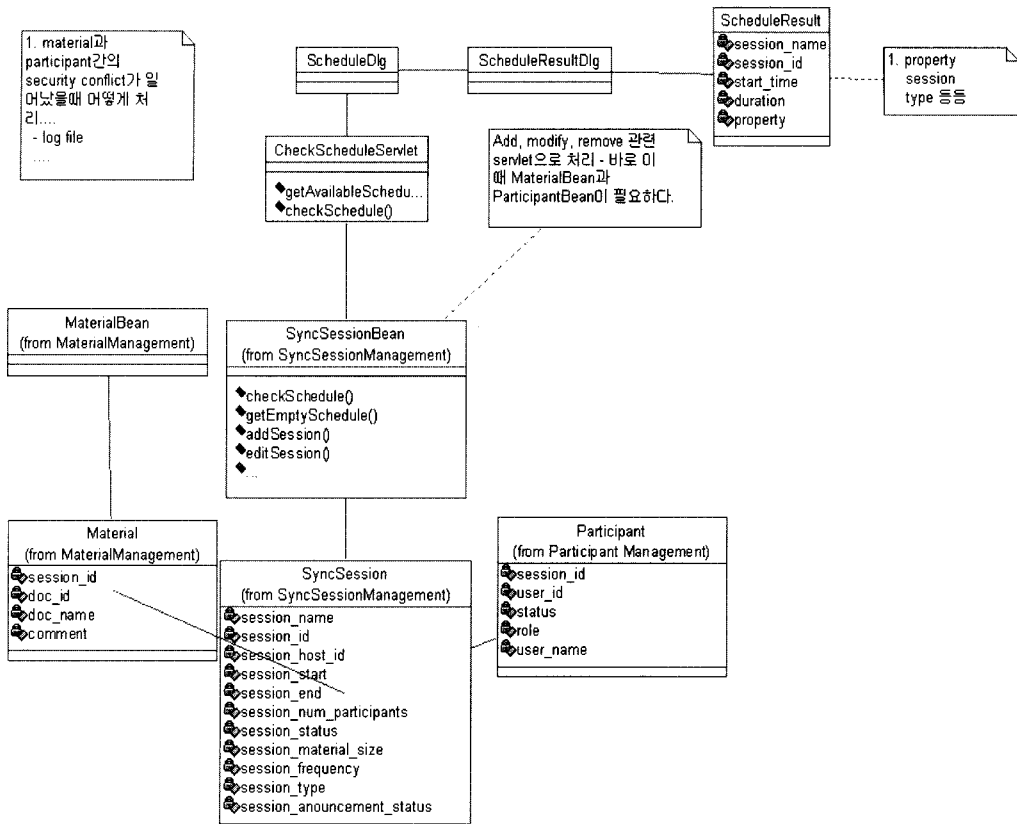
약하기 위한 실시간 세션 빈(SyncSession Bean), 화이트보드에 자료를 올리기 위한 자료 빈(Material Bean), 회의에 참가자를 초대하기 위한 초대자 빈(Participant Bean), 메시지 관리 빈 등으로 구성되어 있으며 각각의 빈들의 관계는 <그림 6>과 같다. 각각의 로직을 담당하는 세션 빈들은 내부적으로 데이터베이스에 접근할 때에는 엔터티 빈을 이용하도록 하고 있다[9].

4.3 시스템 구현

본 논문에서 제안한 협업 지원시스템은 분산환경 하에서 플랫폼과 언어에 관계없이 시스템을 설계하고 구현하는 방법을 제시하였다. 클라이언트 측은 마이크로소프트사의 운영체제와 개발 환경 하에서 프로그램을 개발하고, 서버 측은 EJB 컴포넌트를 이용하여 자바 서버를 개발하였으며, 특히 EJB 자바 서버와 윈도우즈 클라이언트의 통신을 위하여 XML 기반 SOAP 프로토콜을 사용하여 클라이언트와 서버, 서버와 서버간의 데이터 통신을 원활하게 하였다.

SOAP을 사용한 이유는 첫째, 소프트웨어 컴포넌트를 이용한 객체 기반의 이 기종 컴포넌트간의 표준화된 데이터 표현 및 통신 프로토콜의 부재로 인해 많은 솔루션들은 통합성이 그다지 좋지 않은 플랫폼 의존적인 경향이 있으며, DCOM, CORBA 등과 같은 다른 솔루션들은 방화벽 때문에 커뮤니케이션의 문제가 발생하는 문

Class Diagram: Schedule / ScheduleMain

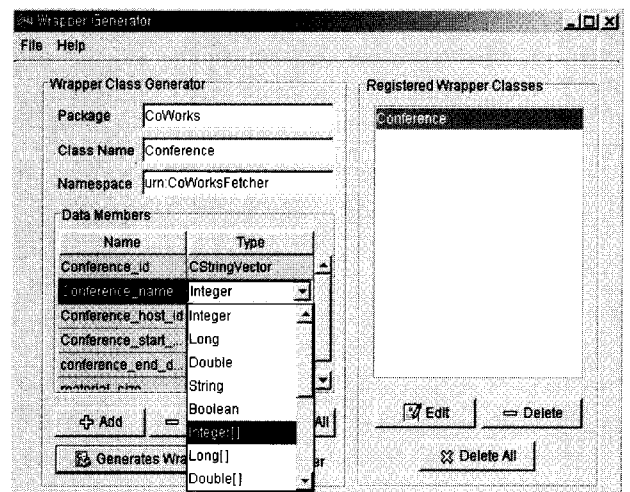


<그림 6> 빈들의 관계를 나타내는 클래스 다이어그램

제점을 해결하고자 사용하였으며, 둘째는 구현하고자 하는 협업 지원 시스템의 서버는 EJB 기반의 자바를 유지하려 하고, 클라이언트 프로그램은 마이크로소프트 기반으로 가려고 하다가 보니 클라이언트와 서버 사이의 통신은 XML 기반의 SOAP을 이용하면 편리하고 무리 없이 통신이 가능하기 때문이다.

마이크로소프트 윈도우 환경을 위한 pocketSOAP[11]으로 개발된 클라이언트와 Apache 서버[13]가 통신할 때 일반적으로 비주얼 베이직 클래스와 자바 클래스를 자동으로 Serialize/Deserialize할 수 없다는 문제가 발생하는데 이를 해결하고자 SOAP 기반 새로운 SOAP API를 개발하여 이용하였다. 또한 이 새로운 SOAP API를 이용하기 위하여 Wrapper Class Generator를 개발하였는데 이것은 아파치 자바(Apache Java) SOAP 서버와 비주얼 베이직 SOAP 클라이언트 사이에 통신을 하기 위한 자바 객체와 비주얼 베이직 XML Wrapper 객체를 자동으로 생성한다. 개발된 Wrapper Class Generator를 이용하여 협업지원 시스템의 실시간 컨퍼런스 진행하는 과정의 클래스를 만들어 보면 총 5개의 파일이 자동 생성되는데 4개의 *.cls 파일은 클라이언트용 어플리케이션 작성시 비주얼 베이직 프로젝트에 추가하여 사

용하면 되고, 1개의 *.java파일은 서버 측에서 사용되는 자바 객체이다. 개발한 Wrapper Class Generator를 실행하여 자동으로 객체를 실행하는 예는 <그림 7>과 같고 자동 생성된 파일은 아래 <그림 8>과 같다.



<그림 7> Wrapper Class Generator를 이용하여 객체를 생성하는 예

이름	크기	종류	수정된 날짜
Conference.cls	9KB	Visual Basic Cla...	2001-08-16 오전 ...
Conference.java	2KB	Java Source	2001-08-16 오전 ...
ConferenceHashtable.cls	8KB	Visual Basic Cla...	2001-08-16 오전 ...
ConferenceVector.cls	6KB	Visual Basic Cla...	2001-08-16 오전 ...
ConferenceVectorHashtable.cls	8KB	Visual Basic Cla...	2001-08-16 오전 ...

<그림 8> Wrapper Class Generator를 이용하여 생성된 객체 파일

위에서 개발한 SOAP 컴포넌트와 Wrapper Class Generator를 실행하여 만든 클래스와 객체를 이용하여 클라이언트와 서버 프로그램들을 구현한다. 구현하여 실행한 협업 지원시스템의 실시간 컨퍼런스 화면은 아래 <그림 9>와 같다. 구현된 시스템의 구동환경은 서버 측에서는 자바 서버 컴포넌트와 SOAP Service가 등록된 웹 어플리케이션 서버(Orion Server)가 구동되어 있어야

하고, 비주얼 베이직으로 개발된 클라이언트 쪽 프로그램을 실행하면 SOAP 서비스를 통하여 데이터를 원활하게 주고받을 수 있는 것을 입증하였다.

SOAP 서비스를 이용하여 구현한 시스템에서 클라이언트와 서버가 통신하기 위하여 보낸 각각의 SOAP 메시지의 예를 보면 아래와 같다[5].

본 시스템은 n-tier 클라이언트/서버 구조를 가지고 있으며, 개발 환경은 다음과 같다.

서버 측 개발환경

- Orion Server1.4.5
(<http://www.orionserver.com/>)
- JDK 1.2.2, J2EE 1.2.1
- Apache SOAP 2.0
(<http://www.apache.org/dist/soap/>)

Client가 보낸 SOAP 메시지

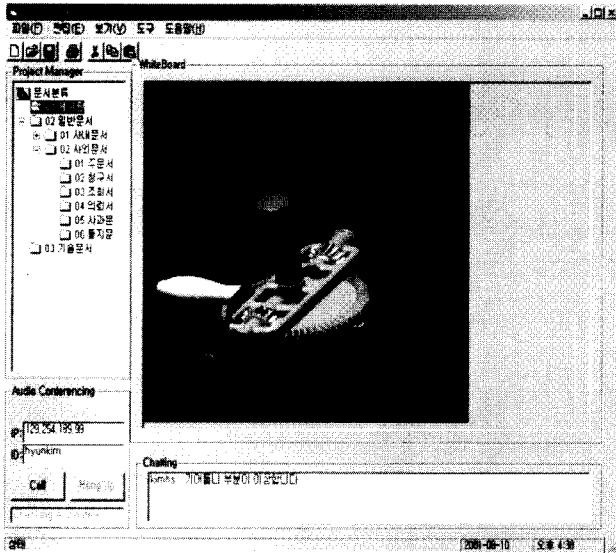
```
<?xml version='1.0' encoding='ISO-8859-1'?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/'
xmlns:xsi='http://www.w3.org/1999/XMLSchema-instance' xmlns:xsd='http://www.w3.org/1999/XMLSchema'>
<SOAP-ENV:Body>
<ns1:getAddressFromName xmlns:ns1='urn:urn:Callaboration:Sync:conference' SOAP-ENV:encodingStyle='http://schemas.xmlsoap.org/soap/encoding/'>
<nameToConference xsi:type='xsd:string'>3DModel</nameToConference>
</ns1:getAddressFromName>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Server가 보낸 SOAP 메시지

```
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/'
xmlns:xsi='http://www.w3.org/1999/XMLSchema-instance' xmlns:xsd='http://www.w3.org/1999/XMLSchema'>
<SOAP-ENV:Body>
<ns1:get3DModelFromNameResponse xmlns:ns1='urn:urn:Callaboration:Sync:conference' SOAP-ENV:encodingStyle='http://schemas.xmlsoap.org/soap/encoding/'>
<return xsi:type='ns1:3DModel'>
<SessionId xsi:type='xsd:string'>01334</SessionId>
<SessionName xsi:type='xsd:string'></SessionName>
<3DModelNumber xsi:type='ns1:3DModelNumber'>
```


클라이언트 측 개발환경

- Poket SOAP 0.9
(<http://pocketsoap.com/pocketsoap/>)
- WrapperClass Generator,
개발된 SOAPAPI
- Visual Basic 6.0



<그림 9> 협업 지원 시스템의 실시간 회의 화면

5. 결론

본 논문에서는 분산환경에서 협동으로 작업할 수 있는 구조를 제시하고 협업 지원 시스템 개발에 관해 논의하였다. 개발된 시스템은 분산된 환경에서 원격으로 떨어져 있는 다양한 사용자들 또는 응용 프로그램이 통합적이고 협업적으로 프로젝트를 진행할 수 있도록 지원하며 이 과정 중에 발생하는 모든 정보를 통합 운용하기 위한 기능을 제공한다. 제안한 시스템은 첨단 정보 기술을 수용하여 SOAP-EJB 기반의 협업 지원 시스템을 개발하였기 때문에 서버 측의 소스 프로그램은 전혀 수정하지 않고도 클라이언트 측의 소스프로그램은 플랫폼, 언어에 관계없이 개발할 수 있는 최대한의 장점을 가지고 있다. 그러나 SOAP을 이용한 협업 지원 서비스의 문제점은 인터넷 자체의 문제점과 많은 부분이 일치한다. 특히 느린 속도, 제한적인 서비스, 보안 문제 등은 향후 인터넷의 발전과 함께 반드시 보완되어야 할 문제이다.

참고문헌

- [1] 김현, 김형선 외, "인터넷 기반의 분산협동 설계",

대한기계학회 춘계학술발표회 논문 집, 2001

- [2] 김현, 명재형외, "가상공학 구현을 위한 웹기반 엔지니어링 서비스", 한국 CAD/CAM학회 학술발표회 논문집, 2000
- [3] 백동석, "인터넷 기반 분산 협업 설계 지원 프레임워크의 구축", 서울대 대학원 석사학 위 논문, 2001
- [4] 한재선 외 3인, "웹 어플리케이션 서버 : 인터넷 전자상거래를 위한 공용 플랫폼", EC/CALS 기술워크샵, pp. 63~69
- [5] 한성배, 김형선, "웹 환경에서 원격 데이터베이스에 효율적인 접속을 위한 DB 브로커개발", 대한산업공학회, 한국경영과학회 춘계학술대회 논문집, p.770-773, 2000
- [6] Craig Murphy, *Creating Application using SOAP and XMLHTTP*, 2001
- [7] Dan Chang, Dan Harkey, *Client/Server Data Access with Java and XML*, John Wiley & Sons, Inc, 1999
- [8] IONA, EJB Home, <http://www.ejhome.com>
- [9] Java Naming Directory Inteface, JavaSoft Inc., <http://java.sun.com/products/index.html>
- [10] J2EE hosting by The Enterprise Java Internet Provider, <http://www.orionserver.com>, 2001
- [11] Rumbaugh, J., M. Blaha, W. Premerlani, F.Eddy and W. Lorensen ;*Object-Oriented Modeling and Design*, Prentice, 1999
- [12] Simon Fell, "pocketSOAP WSDL Wizard", <http://www.pocketsoap.com/pocketsoap>, 2001
- [13] Sun Microsystems, "*Enterprise javaBeans Specification version 1.0*", 2000
- [14] The Apache Software Foundation, "*Apache XML Project*", <http://xml.apache.org/dist/soap>, 2001
- [15] UML Notation Guide, "<http://www.rational.com>", Rational Co. 1999.