엔터프라이즈 자바 빈 환경에서의 교육시스템 구현

유재호, 김일민
한 성 대 학 교 컴퓨터 신기술 대학원 컴퓨터공학과
{panic_gray, ikim}@hansung.ac.kr

Implementing Learning System on Enterprise Java Bean

Jae-Ho Yu, Il-min Kim, Hansung Univ.

요 약

본 논문에서는 EJB환경에서의 강의시스템 구현에 관한 논문이다. 이 시스템은 하나는 교수 강의시스템을 구현하고, 다른 하나는 수강생 개개의 공동 작업형 환경을 구현함을 목적으로 하고있다. 이 시스템은 기존의 N-tier방식의 클라이언트/서버 방식에서 벗어나 분산 환경에서 작업이 가능하도록 함이 목적이다. 본 논문은 분산환경에서 가장 유용한 언어중 하나인 J2EE를 기반으로 구성하였으며 따라서 이 기종간의 컴퓨터 환경에서도 확장 및 포팅이 용이하고 장차 생길지도 모르는 새로운 요구 사항에서도 대처 할 수 있게끔 함이 주목적이라 할 수 있다.

1. 서 론

컴퓨터 하드웨어 및 소프트웨어 기술이 발전함과 동시에 컴퓨터 네트워크의 양적, 질적 발전 또한 계속 가속화되고 있다. 또한 인터넷의 팽창과 더불어 그에 상응하는 응용 소프트웨어 개발 또한 급속도로 진행중이다.

특히 네트워크 상에서 가장 도움을 주는 분야가 일련의 교육용 시스템이라고 할 수 있다. 컴퓨터 네트워크환경 상에서는 이전의 오프라인형 교육방식과 달리 시간, 공간 제약을 상당부분 극복함과 아울러 교육내용의 상호 교류를 통해 교육의 질적 수준을 높일 수 있다.

하지만 기존의 원격교육시스템은 N-tier 방식의 클라이언트/서버 방식을 채택함으로 인해 서버 확장성과 부하 및 안정성 면에서 많은 문제점을 낳고 있다. 또한, 대부분의 시스템이 일반적인 오프라인 강의 형태를 그대로 답습한 면이 많아 교육의 질적 측면에서는 아직 모자란 면이 있다.

이러한 제반 문제들을 해결하고자 본 논문은 엔터프라이즈 자바 빈 환경을 기반으로 한 새로운 교육시스템을 제안하고자 한다.

2. 엔터프라이즈 자바 빈 소개

Corba, DCOM, RMI등의 다양한 분산 컴퓨팅언어가 널리 사용되긴 하지만 분산 환경에서 실행되는 응용 프로그램을 작성하기에는 여러 가지 문제점이 있다. 이러한 문제점들은 표준화, 보안, 트랜잭션등의 서비스를 위해 개발자들이 직접 코드를 작성해야 하는 점으로 요약된다. 이러한 문제점들을 보완 및 해결하기 위해 개발된 것이 엔터프라이즈 자바 빈(Eneterprise Java Bean)이다.

엔터프라이즈 자바 빈은 3-tier 혹은 N-tier를 지원하는 컴포넌트 기반의 분산컴퓨팅을 위한 구조이다. EJB를 사용하면 간단한 작업만으로 분산 처리가 가능하고, 따라서 개발자는 위에서 제기된 제반문제들에 별다른 구애를 받지 않고 응용소프트웨어를 작성 할 수 있는 것이다.

EJB는 단독으로 실행되는 것이 아니라, EJB컨테이너 라는 소프트웨어에 설치되어야 실행 될 수 있다. EJB 컨테이너의 구성은 다음 그림과 같다.

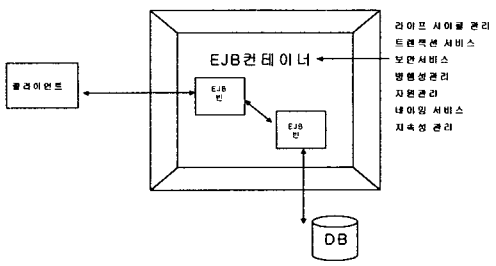

그림1. EJB 컨테이너 서비스

이러한 기본적인 서비스들은 EJB 컨테이너가 제공하기 때문에 다음과 같은 장점들을 얻을 수 있다.

① 분산 프로그래밍개발시간 단축
② 분산 프로그래밍의 복잡성 감소
③ 컴포넌트화된 객체 사용으로 재사용성 증가

또한 EJB는 컴포넌트화된 프로그래밍이 가능하므로 프로그램의 소스를 변경시키지 않고, 속성을 변경함으로서 커스터마이즈(Customize) 할 수 있는 특징이 있다. J2EE 명세서에는 클라이언트 컴포넌트, 웹컴포넌트, 비즈니스 컴포넌트 등의 종류를 정의하고 있다.
다음 그림은 J2EE 컴포넌트와 컨테이너의 구조이다.

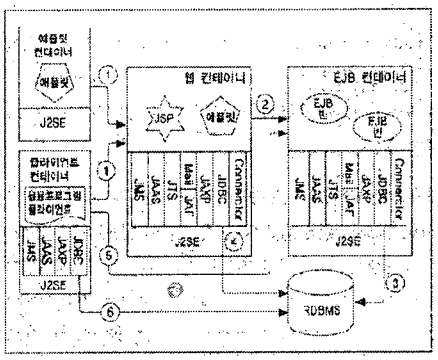

그림2. J2EE 구조

3. EJB 환경에서의 강의시스템 설계 및 구현

본 논문이 제안하는 교육시스템의 주목적은 네트워크에 연결된 이 기종의 컴퓨터들을 하나의 가상 컴퓨터로 인식하여 사용자에게 보다 쉬운 환경을 제공하는 것이다. 본 논문의 시스템에서는 EJB에서 사용할 수 있는 원격 객체란 개념을 도입해 다른 호스트에 있는 메소드 호출을 받아들일 수 있도록 한다.
즉, 공통적인 속성을 가진 객체들을 그룹으로 묶은 다음, 그룹을 대상으로 메소드를 호출하면 모든 객체들에게 응답을 보낼 수 있는 것이다. 또한 Vector와 비슷한 자료구조인 분산 리스트 자료구조를 이용해 각 클라이언트는 서버에게 객체 스트림(Object Stream)으로 캡슐화된 메시지를 전송하고 서버는 마찬가지로 캡슐화된 메세지로 응답한다. 이렇게 하면 다른 종류의 데이터 타입까지 처리 할 수 있다.

본 논문에서 구현하고자 하는 네트워크 시스템의 개요는 다음과 같다.

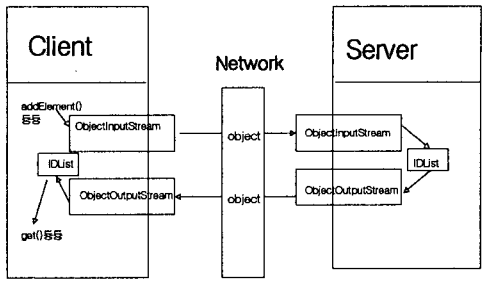

그림 3. 강의 시스템의 네트워크 구조

3.1.1 강의시스템구성 및 설계

본 논문의 강의 시스템은 크게 수강생 개개인의 공동 작업 모듈과 교수 강의형 모듈로 구성된다.

CoJobModule : 수강생 개개인의 채팅 및 공동 작업을 담당하는 객체.
CoWBModule: 교수강의용으로서 화이트보드 기능을 담당하는 객체

CoJobModule은 기본적으로 채팅방을 선택 할 수 있고, 필요한 경우에는 다른 채팅방을 만들 수 있다. 기본적인 채팅의 기능 외에 공동 작업이 가능한 기능을 부여 한 점이 특징이다. 즉, 참여자 모두가 주어진 과제를 임의로 수정 및 삭제 할 수 있고, 나아가서 공동의 문안을 작성 할 수 있게 할 수 있는 기능이 가능하게 했다.
다음의 그림은 CoJobServerEJB 및 CoJobRoom의 시스템의 요구사항이다.

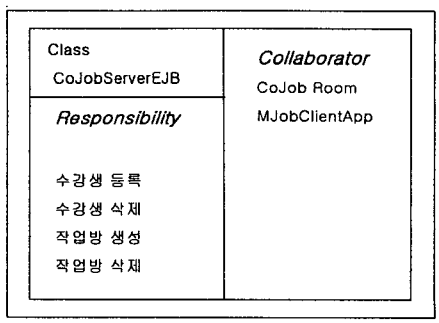

그림 4. CoJobServerEJB의 CRC

Class	Collaborator
CoJobRoom	CoJobServerEJB
Responsibility	MJobClientApp
수강생 등록	
수강생에게 메세지를 전달한다	
수강생등록을 삭제한다.	
수강생 목록을 확인	

그림 5. CoJobServerEJB의 CRC

Class	Collaborator
MJOBClientApp	CoJobRoom
Responsibility	CoJobServerEJB
수강생 등록	
수강생 삭제	
작업방 생성	
작업방 삭제	

그림 6. MJobClient의 CRC

CoJobModule의 구성은 CoJobServer 및 CoJobRoom, MJobClientApp의 세 개의 클래스로 구성된다.

CoJobServerEJB는 작업방을 관리하기 위한 여러 가지 메소드들이 포함되어있다. 또한 클라이언트 프로그램은 RMI 콜백을 사용하기 위해 setMessage(), updateMessage(), updateRooms() 메소드를 갖고 있다.

```
import java.rmi.*;
import javax.ejb.*;

public interface CoJobServer extends EJBObject {
 public void unregister()
          throws RemoteException;

 public void broadcast(String msg)
          throws RemoteException;

 public void makeRoom(String name) throws RemoteException;
 public void enterRoom(String room) throws RemoteException;
 public void leaveRoom() throws RemoteException;
 public String[] getRoomList() throws RemoteException;
 public String[] getUserList() throws RemoteException;
}
```

CoWBModule 역시 CoJobModule과 같은 방식으로 공동 작업이 가능하게끔 구현했다. 서로 수정이 가능한 점이 특징이며, 수정 즉시 모두에게 동일한 메시지를 보내 변경사항을 참여자 모두가 동일하게 볼 수 있는 구조이다.

특징은 분산 리스트 자료구조 형태를 취하고 있어 각각의 클래스들의 메시지가 객체스트림(Object Stream)으로 캡슐화된 형태로 전송되는 점이 특징이다. 이와 같은 형태를 취함으로서 전송 데이터의 종류에 구애받지 않고 전송이 가능한 점이 특징이다.

다음의 그림은 서버측 실행이다. 보기와 같이, 서버는 각 클라이언트들의 접속 여부를 확인하고 클라이언트간의 수정, 변경 사항을 접속된 모든 클라이언트에게 발송한다.

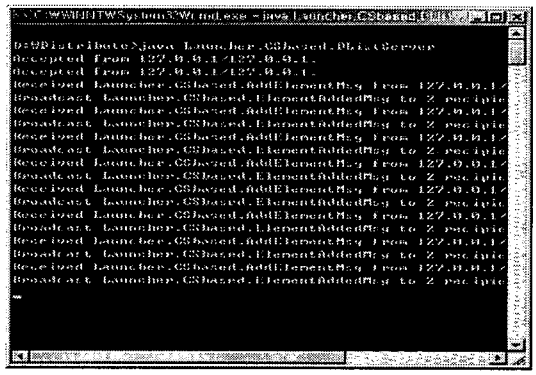

그림 7. 서버측 실행 예제

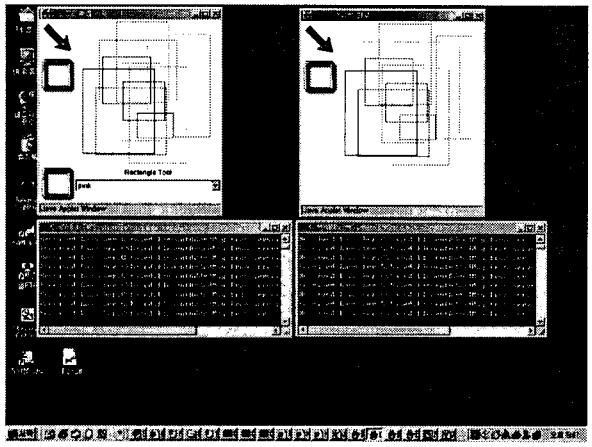

그림8. 분산 리스트로 구현한 화이트보드 프로그램(클라이언트 측)

그림에서 볼 수 있듯이 개개의 클라이언트는 접속 즉시 현재까지의 작업 현황을 그대로 볼 수 있다. 또한 1명의 클라이언트가 어떤 작업을 수행하면 그 결과가 모든 클라이언트에게 전달되며, 서버는 모든 작

업상황을 수집하여 순차적으로 전달 순서에 맞게 브로드캐스트한다.

4. 강의시스템 수행 시나리오

수강생이 강의 시스템에 참여하면 크게 두 가지 모듈로 접근이 가능하다. 위에서 언급했다시피, CoJobModule이나, CoWBModule로 접근을 할 수 가 있다. 즉, CoWBModule로 접근하면 교수가 진행하는 수업에 참여함을 의미하고, CoJobModule에 접근하게 되면, 개개의 수강생이 모두 참여 할 수 있는 작업이 가능하다. 물론 교수도 CoJobMoudle에 접근한다면, 공동 작업에 참여가 가능하다. 공동 작업은 여러 형태로 생각할 수 있는데 프로그램 언어 작성에서 어떤 수강생이 자신이 작성한 코드를 다른 수강생이 이 코드를 변경하거나 수정하는 과정을 접속한 모든 수강생이 동시에 그 과정을 볼 수 있는 경우도 가정 할 수 있다.

또한, CoWBModule에서의 경우도 교수가 아닌 수강생 발표자가 화이트보드에 작성한 예제를 올려놓으면 다른 수강생이 이에 대한 수정이나 다른 의견을 제시할 수 있는 형태로 응용이 가능하다.

5. 결론 및 향후 과제

기존의 교육시스템들은 클라이언트/서버 구조하에서 작동하기 때문에 확장성과 속도 및 안정성 면에서 많은 단점을 가지고 있었다. 본 논문에서는 이러한 단점을 해결하기 위해 EJB환경 하에서 구현된 형태의 교육시스템을 제안하였다.

또한 일반 수업형 강의에서 공동 작업형 강의 형태가 이뤄지면서 사용자간의 빠른 상호 작용과 협력적 관계를 증진시키게 하기 위함이 본 논문이 제시한 시스템의 특징이라고 할 수 있겠다.

원격교육의 목표 중 하나가 일반적인 전파형 수업이 아닌 상호작용인 만큼 본 논문의 시스템은 이에 부합한다고 할 수 있다.

그러나, 본 논문은 원격교육의 기본적인 기능만을 구현 한 점이 미흡한 점이라 할 수 있다. 장차 실질적이면서 유용한 교육환경을 제공하기 위해서라면 몇 가지의 개선요소와 요구되는 기능이 있는데 이를 정리하면 다음과 같다.

첫째, 음성채팅의 기능을 추가하여 시각적 효과 와 더불어 멀티미디어 환경에 부합하게끔 기능을 추가시켜야 한다.

둘째, 위에서 제시된 각 모듈의 인터페이스를 하나로 통합해 시스템 관리가 용이함과 동시에 사용자 인터페이스를 보다 쉽고 간편하게 구성해야 한다.

셋째, 특정 형태의 강의만이 아닌 다른 형태의 강의도 진행 할 수 있게끔 새로운 모듈의 개발이 필요하다.

[참고문헌]

[1] 김일민, 알기 쉬운 중급 자바, 홍릉 출판사 2001

[2] Merlin Hughes외 3인, JAVA NetWorking 프로그래밍 인포북 2000

[3] 유재우 외 3인, 프로그래머를 위한 EJB, 이한 출판사, 2002

[4] 김현철, 객체지향 분산 가상 환경 상에서의 원격교육시스템 정보과학회 2001

[5] 황대준 외 6인, "21세기형 첨단학교 · 가상대학 설립운영에 관한 연구", 교육부 교육정책과제 연구보고서. 1997