

# EDUCAUSE의 IMS 명세서에 기반을 둔 개방형 교육 시스템

<sup>1</sup>황희정\*, 백영태\*, 강운구\*, 이세훈\*\*, 윤경섭\*\*, 왕창종\*  
\*인하대학교 전자계산공학과  
\*\*인하공업전문대학

## Open Distributed Learning System based on IMS Specification of EDUCAUSE

<sup>1</sup>H.J. Hwang\*, Y.T. Baek\*, U.G. Kang\*, S.H. Lee\*\*, K.S. Yoon\*\*, C.J.  
Wang\*

\*Dept. of Computer Science and Engineering, Inha University  
\*\*Inha Technical College

### 요약

본 논문에서는 교육 시스템의 국제적 표준 구조인 IMS 명세서를 기반으로 개방형 분산 교육 시스템을 설계하였다. 설계한 시스템은 IMS 관리 시스템의 개발 권고 사항인 HTTP/CORBA 모듈과 그룹 관리 기능, 개인 프로파일 관리 기능을 사용하였다. 제안된 시스템은 표준 개방 환경을 준수하고 있으며 관리 시스템이 IMS 관리 시스템에 기반하고 있으므로 상호 연동성을 확보할 수 있었으며, 향후 모든 IMS의 요구 사항의 반영이 가능하다.

### 1. 서론

인터넷의 WWW 서비스가 일반화됨에 따라 이를 기반으로 하는 다양한 애플리케이션에 대한 요구가 증가하고 있다. 특히 원격 교육 분야는 가장 큰 파급 효과를 가지는 응용으로 여겨지고 있다[1].

현재 교육 시스템에서의 구조나 콘텐츠가 개발자에 따라서 상이하기 때문에 상호 연동 하는데 많은 문제점을 내포하고 있으므로, 개방형 교육 시스템에 대한 표준화가 필요하다. 따라서 EDUCAUSE라는 단체에서는 IMS(Instructional Management System)라는 프로젝트를 통해 1998년 3월에 프로토타입을 발표하고 1999년 2월 표준적인 교육시스템 명세서를 발표하였다[1,2,3].

IMS에서는 분산 학습 시스템과 콘텐츠에 대한 표준을 정함으로써, 교육 시스템간의 상호 운용성을 보장할 수 있도록 하였다. 따라서 이러한 표준을 따

르는 시스템들은 콘텐츠의 재사용과 시스템간의 연동으로 인해 그 교육적, 경제적 효과를 극대화 할 수 있다.

따라서 본 논문에서는 IMS 명세서를 분석하고, 이명세서에 따른 교육 시스템을 설계한다. IMS에서는 현재까지의 다양한 컴퓨팅 기술들을 수용하고 있으며, 시스템 설계자가 각각에 맞는 기술들을 선택하여 설계할 수 있도록 하고 있다.

초기의 단순한 원격교육 서비스들은 표준 웹 프로그래밍 인터페이스인 CGI(Common Gateway Interface)를 사용하는 문제로 인해 전체 시스템의 효율이 감소하는 문제점을 가지고 있으며, 또한 웹 서비스를 구성하는 문서 포맷인 HTML(Hyper Text Markup Language) 과 통신 프로토콜인 HTTP(Hyper Text Transfer Protocol)의 한계로 대규모의 실시간 다중 사용자 시스템을 개발하는데 많은 문제점을 가지고 있다[4].

본 논문에서는 이러한 문제점을 해결하기 위해서는 개방형 분산객체 표준 기술인 CORBA(Common Object Request Broker Architecture)[5]를 기반으로 한 분산 교육 시스템을 설계하고 HTML이 제공하는 단순로운 사용자 인터페이스를 극복하기 위해 자바 애플릿 등의 향상된 기술을 적용한다.

CORBA는 이기종 분산 환경에서 객체 지향 개발 방식을 제공하므로 클라이언트는 분산 객체가 어디에 존재하는지 또는 객체를 실행하는 운영 체제가 무엇인지, 클라이언트는 서버 객체가 어떻게 구현되었는지 알 필요가 없다. 실제로 서버 객체는 C++, Java, COBOL 등의 코드로 구현한다.

## 2. EDUCAUSE의 IMS 고찰

EDUCAUSE는 미국의 교육 관련 협회인 CAUSE와 Educom의 회원들이 1998년 7월에 만든 통합 협회로 양질의 교육을 지원하기 위하여 컴퓨터시스템의 활용과 네트워크 및 정보자원의 관리와 이용에 관한 표준화에 초점을 두고 있다.

EDUCAUSE에서 지원하는 교육 시스템 표준화 프로젝트인 IMS(Instructional Management System)의 목표는 분산 학습 환경과 여러 저작자가 함께 콘텐츠를 만들 수 있는 방식에 관한 표준 명세서를 만드는 것이다[1,2,3].

### 2.1 IMS의 배경

인터넷과 월드와이드웹의 대중화에 따라 어디에서나 모든 종류의 교육이 가능해졌으며, 새로운 기술들이 교육에 적용되었다. 그러나 온라인 교육 콘텐츠를 관리하기 위한 접근 방법에 대한 표준이 없기 때문에 웹을 통합 효율적인 교수, 학습이 커다란 장애가 되고 있다.

IMS 프로젝트는 온라인 상에서의 여러 가지 교육 행위 즉, 교육 콘텐츠의 이용, 학습자의 진행 상황 추적, 학습 결과 레포팅, 행정 시스템 사이에 학습자 기록 상호 교환 등에 관한 개방적인 명세서를 개발하고 발전시키기 위한 프로젝트이다. 이 명세서는 분산된 환경에서 학습의 기회를 넓일 수 있으며, 교수나 학습자의 창의성과 생산성을 증가시킬 수 있을 것이다. IMS 프로젝트의 목표는 분산 학습 환경과 여러 저작자가 같이 콘텐츠를 개발할 수 있는 표준화된 구조를 만드는 것이다. 따라서 이 프로젝트가 성공하기 위해 많은 산·학계와 정부 기관으로부터 지원을 받고 있다. IMS는 현재 1999년 2월 버전 1.0을 발표하였으며, IEEE 표준을 위해 명세서를 검토 받고 있는 중이다.

### 2.2 IMS 명세서

그림 2는 IMS(Instructional Management System) 명세서에서 제시하는 IMS 환경의 내부 구조와 외부 환경과의 관계를 나타내고 있다.

#### (1) 그룹 관리 기능

그룹 관리는 그룹내의 멤버와 역할들을 관리하고, 멤버들에게 가능한 정보와 행위들을 지정할 수 있는 기능이다. 그룹 관리의 전형적인 예로는 코스 설계 과정에서 사전 요구 사항들을 정하고, 자원들을 선택하고, 그룹을 위한 콘텐츠와 행위들을 구조화하는 것이다. 상세한 그룹 관리 기능은 필수 요구 사항으로 21가지가 있으며, 선택 사항으로 2가지가 있다.

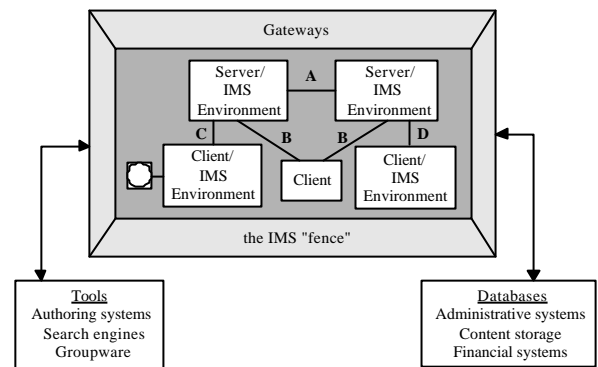


그림 2 IMS 환경의 구조

#### (2) 개인 프로파일 관리 기능

개별적 사용자 신상 및 학습 과정에 대한 기록들을 어떻게 관리하는가에 대한 기능이다. 개인 신상은 사용자 ID, 성취도, 학습 과정 정보, 기호 정보 등으로 구성된다. 필수 요구 사항은 12가지, 선택 사항은 4가지가 있다.

#### (3) 사용자 행위 관리 기능

사용자 행위 관리 기능은 사용자가 어떻게 실제로 IMS 환경과 작용하는지를 관리한다. 그룹관리가 추상적인 상호작용을 성립하는데 반해 사용자 행위 관리 기능은 상호작용의 특정한 사례 또는 구현을 관리하며, 학습자가 다른 학습자 혹은 교수 그리고 전문가와 의사교환을 가능하게 한다. 필수 요구 사항은 11가지, 선택 사항은 1가지가 있다.

#### (4) 평가 및 인증 관리 기능

평가 및 인증 관리 기능은 사용자의 진행과정과 수행성취도에 대한 정보를 기록하고 추적 및 보고하는 기능을 수반한다. 필수 요구 사항은 6가지, 선택 사항은 2가지가 있다.

#### (5) 콘텐츠 관리 기능

콘텐츠 관리 기능은 개인 혹은 그룹의 학습자료와

도구의 생성을 위한 요구사항들을 포함 한다. 필수 요구 사항은 28가지, 선택 사항은 6가지가 있다.

(6) 라이선스 관리 기능

라이선스 관리 기능은 콘텐츠의 사용과 배포 및 서비스의 제공과 관련된 요구사항을 지원한다. 필수 요구 사항은 26가지, 선택 사항은 4가지가 있다.

(7) 보안 관리 기능

보안 관리 기능은 신원확인, 인증, 권한부여에 대한 요구사항을 지원한다. 예를 들면 IMS 환경에 접근, IMS 자원의 사용에 대한 접근 제어 같은 기능이다. 필수 사항은 9가지, 선택 사항은 1가지가 있다.

(8) 기술적 행정 관리 기능

기술적 행정 관리 기능은 IMS 환경 및 콘텐츠의 구조 또는 전반적인 기반 구조에 대한 요구사항을 지원한다. 보안 관리 기능과 유사하게 여기에서 언급된 요구사항은 IMS 의 모든 특징들을 지원한다. 필수 요구 사항은 21가지, 선택 사항은 1가지가 있다.

### 3. IMS 명세서에 기반을 둔 분산 교육 시스템의 설계

#### 3.1 시스템 개요

시스템은 그림3 과 같이 관리 시스템을 중심으로 그룹/사용자/프로파일 부분을 IMS 권고 사항을 적용하여 설계한다. IMS 에서는 HTTP/COM 및 HTTP/CORBA/RMI 두 가지 구현 모델을 제시하고 있는데 HTTP/CORBA 모델을 선택했다[6].

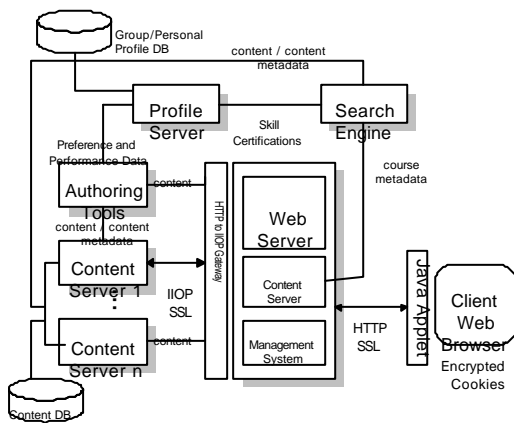


그림 3 전체 시스템 구조

서버 시스템을 구성하는 모듈은 크게 관리모듈과 서비스 모듈로 나눌 수 있다. 관리 모듈을 구성하는 요소들은 웹서버와 콘텐츠서버 그리고 IMS 관리 시

스템이며 서비스 모듈은 다수의 콘텐츠서버와 저작 도구 그리고 프로파일서버 및 검색엔진으로 구성되어 있다.

① 검색엔진 : 다수의 콘텐츠서버 에서 특별한 종류의 교육내용을 검색하기 위해 콘텐츠 메타데이터를 사용한다. 또한 프로파일 서버로 부터 요구되는 사용자를 검색한다.

② 저작도구 : 저작도구는 프로파일 서버로부터 해당 프레젠테이션을 최적화 하기 위한 기호 데이터 (Preference data)를 검색한다. 또한 IMS 관리 시스템과 콘텐츠를 이용해 상호작용을 한다.

③ 프로파일 서버 : 그룹 및 개인 데이터를 관리하는 서버로 검색 엔진과 콘텐츠 서버, 저작도구를 위한 기본정보를 제공한다.

④ 콘텐츠 서버 : 콘텐츠 서버는 개별적으로 관리되는 콘텐츠를 위한 다수의 서버와 IMS 관리 시스템내의 코스 데이터를 관리하는 콘텐츠 서버가 있다.

⑤ 웹 서버 : 사용자와 교육시스템의 중계 역할을 담당한다.

⑥ 클라이언트 : 사용자 인터페이스는 웹브라우저로 자바 애플릿을 통해 서버와 상호작용 한다.

⑦ 데이터베이스 : 콘텐츠 서버 들에 의해 관리되는 콘텐츠 DB와 그룹 및 사용자 정보를 가지고 있는 그룹 및 개인 프로파일 DB가 있다. 콘텐츠 DB 의 정보들은 저작도구에 의해 생성되고 콘텐츠 메타데이터로 저장, 관리된다.

시스템의 각 구성 요소들 사이의 통신은 IMS 관리 서버를 중심으로 이루어지는데 사용자 인터페이스와의 통신은 HTTP를 사용한다. 웹 서버는 실제 사용자 요청을 처리할 CORBA 객체에 HTTP to IIOIP 게이트웨이를 이용하여 분산 객체에 메시지를 전달한다. 시스템에서의 네트워크 데이터 보안은 1차적으로 사용자 인증을 사용하고 세션동안 이를 인증하기 위해 암호화 된 쿠키(cookie)를 사용한다. 2차적으로는 세션 유효 시간을 이용하여 유효 세션을 관리하며 3차적으로 서버간의 데이터 전송시 SSL(Secure Socket Layer)를 사용한다.

#### 3.2 관리 시스템의 설계

IMS 관리 시스템은 IMS 명세서를 준수하는 모든 서비스들과 연동하여 전체 시스템의 상호 운용을 가능하게 하는 시스템으로 중심적인 위치에 있다. IMS 관리 시스템은 HTTP/CORBA 모델로 구현하며 기본적으로 프로파일 서버의 관리를 위한 그룹 관리 시스템 과 개인 프로파일 관리 시스템을 포함

한다. 관리 시스템의 기본적인 IDL 은 다음과 같다.

```
Module IMSMgmt {
// 관리 시스템에서 사용자를 관리하기 위한 인터페이스
interface User{
    User(in String userName);
    String getName();
    long getID();
    User rProfile getUserProfile();
    Vector get Groups();
    String getStartGroup();
    void setStartGroup(in String groupName);
    void setProfile(in String profileServer, in long
profileID);
    getPersonalData();
    getPreferences();
    getRecords();
}
// 관리 시스템에서 사용자 프로파일을 관리하기
// 위한 인터페이스
interface UserProfile {
    UserProfile(in User user);
    PersonalData getPersonalData();
    Preferences getPreferences();
    Records getRecords();
}
// 관리 시스템에서 Group 관리를 위한 인터페이스
interface Group{
    Group(in String groupName, in long parentGroupID);
    Vector getSubgroups();
    Vector getResources();
    String getHomepage();
    void setHomepage(in String homepage);
    User getOwner();
    void setOwner();
    Vector getMembers();
    String getChannel();
    String getNChannel();
    void setNChannel(in String nChannel);
    boolean addMember(in User user);
    boolean removeMember(in User user);
    void addResource(in Resource resource);
    void removeResource(in Resource resource);
    void addSubGroup(in Group child);
    void removeSubGroup(in Group child);
}
}
```

### 3.4 사용자 인터페이스의 설계

전체적인 시스템은 기본적으로 웹 브라우저를 클라이언트 프로그램으로 사용하게 된다. 이때 기존의 HTML로 해결할 수 없는 다양한 사용자 인터페이스를 구현해 보다 효율적인 교육시스템이 될 수 있도록 하기 위해 자바 애플릿을 사용한다. 애플릿의 구현에는 자바 2에 포함된 Swing 컴포넌트를 사용하며 자바 애플릿을 통해 입력된 사용자 데이터들은 웹서버를 경유해 HTTP to IOP 게이트웨이를 통해 HTTP로 전달받은 내용이 IOP로 전달되어 메인 서버인 CORBA 서버로 전달된다. 클라이언트와 서버와의 접속을 줄이고 최적의 성능을 발휘하기 위해 가급적 클라이언트에서 처리할 수 있는 부분을 최대

한 활용하고 자바스크립트와 애플릿의 통신 기능을 이용한다.

```
<HTML>
<SCRIPT LANGUAGE=JavaScript>
function SetId(id) {
    var tmp_id = GetCookie(id);
    document.applets [0].SetnsID(tmp_id);
}
</SCRIPT>

<BODY onLoad=SetId("user_id")>
<APPLET CODE="Clientclass" width=660 height=910>
<param name=USE_ORB_LOCATOR value=true>
</applet>
</body>
</html>
```

그림 4 자바스크립트와 자바 애플릿과의 상호작용 구현 예

## 4. 결론

본 논문에서는 IMS 요구사항을 기반으로 개방형 분산 교육 시스템을 설계하였다. 기존 CGI 방식의 교육 시스템에서 발생하는 서버 과부하와 상호 연동 문제는 IMS 관리 시스템의 개발권고 사항인 HTTP/CORBA 모듈을 사용하는 것으로 해결하였다. 제안된 시스템은 표준 개방환경을 준수하고 있으며 관리 시스템이 IMS 관리 시스템에 기반하고 있으므로 향후 모든 IMS의 요구사항의 반영이 가능하다.

이후의 연구에서는 XML(eXtensible Markup Language)에 의한 코스 카탈로그 시스템 및 컨텐츠 관리 시스템을 구현하여 보다 확장된 IMS 기반 코스웨어 시스템을 구현할 수 있을 것으로 기대된다.

## 참고문헌

- [1] EDUCAUSE, IMS Design Requirements, EDUCAUSE Document, Feb., 1999.
- [2] EDUCOM/NLII, Instructional Management Systems Specifications Document, April, 1998.
- [3] Bob Carlton, Mark Resmer, Steve Griffin, "The Educom NLII Instructional Management System Project", Conference of EDUCOM97, 1997
- [4] Lazar,P. and Holfelder,p., "CGI vs. Server-Side JavaScript for Database Applications", Netscape DevEdge, July 1997.
- [5] OMG, "The Common Object Request Broker, Architecture and Specification Revision 2.3", OMG TC Document, 1998.

[6] 왕창종, 이세훈, Inside CORBA3 프로그래밍,  
도서출판 대림, 1999, 4