

컴포넌트 기반 SCORM 표준 LMS의 개발 방법론 연구

김강석[†] · 김기석^{†*}

요 약

본 논문에서는 e-learning의 콘텐츠 표준화 연구에 있어 국내외적으로 관심이 집중되고 있는 ADL의 SCORM 표준을 따르는 LMS를 개발하기 위해 컴포넌트 구조에 기반한 개발 방법론을 제시한다. 또한 JSP, EJB 등 J2EE 기반의 분산 객체 컴포넌트에 기반하여 개발된 LMS인 iOneLMS를 SCORM 표준을 따르는 LMS로 변환하기 위해 제시된 방법론에 따라 SCORM 핵심 모듈을 설계하고 J2EE 기반의 EJB 컴포넌트로 구현하였다. SCORM 컴포넌트 모듈은 콘텐츠 들여오기, 강좌등록하기, 수강 및 학습 데이터 수집 등 세 모듈로 나누어 구현하도록 하였으며, 구현 과정에서 ADL 및 AICC에서 제공하는 클래스 라이브러리를 일부 재사용 하였다.

A Study on Development Methodology of SCORM Standard LMS Based on the Component Architecture

Kang-Suk Kim[†] · Kiseok Kim^{†*}

ABSTRACT

In this paper, we propose a development methodology based on the component architecture to develop the LMS that follows ADL's SCORM standard, which is interested internally and externally in the e-learning contents standardization research. In like manner, we designed the SCORM core module and implemented the EJB component based on J2EE through the suggested methodology for the conversion of SCORM Conformance LMS, from iOneLMS which was developed based on distributed object component like JSP, EJB, etc. The SCORM component modules were implemented by three modules - content importing, course registering, and taking course and gathering the learning data. In the process of implementation, we used the part of the class library that was supported by the ADL and AICC again.

1. 서 론

IT 기술 및 인프라의 구축에 따라 사이버교육 부분 역시 기술과 시장에 있어서 많은 발전을 이루고 있으며 이와 동시에 사이버교육에 참여하고

있는 각 기관 및 단체들에서 개별적으로 이루어 지던 기술 개발 및 방법론 연구가 개별 연구에 따르는 비효율성을 줄이고 동시에 교육의 질을 높이기 위해 미국과 유럽 등을 중심으로 기술 표준화를 위한 노력으로 초점이 맞춰지고 있다. 미국의 경우 사이버교육 표준화 기구들인 IEEE, AICC, IMS, ARIADNE 등의 연구들을 토대로 정부주도 연구기관인 ADL에서 SCORM이라는

† 준 회 원: 한동대학교 정보통신학과 석사과정
 ** 종신회원: 한동대학교 전산전자공학부 교수
 논문접수: 2002년 12월 30일, 심사완료: 2003년 1월 18일
 * 본 논문은 2002년 산학협동재단의 학술연구비 지원으로 수행된 연구 과제임

표준화 스펙을 제시하여 각 교육계 및 산업계에 서 이러한 표준화 스펙을 따르도록 지원하고 있다. [1,2]

우리나라의 교육환경에 적합한 표준화 안이 제시되지 않은 현 시점에서 국제적 표준화의 흐름을 주도하는 ADL의 SCORM을 적용하는 LMS 및 콘텐츠들의 개발이 필요하다고 여겨진다.

국내에서도 사이버교육 기관 및 시스템들이 표준화에 따르고자하는 더 많은 노력들이 있기를 기대하면서 본 논문에서는 표준화를 고려하지 않고 개발된 LMS인 iOneLMS를 사이버교육 표준화에서 주도적 흐름을 잡고 있는 ADL의 SCORM 스펙을 따르는 LMS로 변환하였다.

SCORM 스펙 중 LMS에 대한 표준을 제시하는 Run-Time Environment는 세 가지 요소로 구성된다. 첫째는 진수(Launch)이다. 이는 학습 객체가 LMS상에서 어떻게 시작 또는 실행되는지에 대한 메카니즘을 말한다. 둘째는 API인데 이는 학습이 진행되는 동안 학습객체와 LMS간의 정보 전달을 가능하게 하는 함수들로 구성된다. 이 API를 통하여 학습자의 학습 정보가 실시간으로 LMS에 전송되고 LMS로부터 학습자 개인 정보가 Client로 전달되어 학습 과정에 활용될 수 있게 된다. 셋째가 Data Model이다. 이는 API가 정보를 전달할 때 어떠한 포맷으로 정보를 전달할 것인가에 대한 정의라 할 수 있다. 공통의 Data 모델과 표준화된 API를 사용함으로써 서로 다른 LMS간에서 콘텐츠의 공유 및 재사용이 가능하게 되는 것이다.[2]

이상의 SCORM Run-Time Environment 세 요소들을 기능적인 측면에서 분석하고 ADL과 AICC에서 제공하는 클래스들을 재사용함과 동시에 iOneLMS와 동일한 개발환경 즉, EJB, JSP 등을 사용하여 SCORM 표준화를 위한 컴포넌트를 구현하여 iOneLMS를 SCORM에 순용하는 LMS로 변환하기 위한 구현방법을 제시하며 구현시 고려사항 등에 대해 이하에서 다루고자 한다.

2. 선행연구

2.1. SCORM Run-time Environment 구성요소

SCORM RTE(Run-Time Environment)는 콘텐츠의 재사용 및 서로 다른 LMS에서의 상호운용을 위해 콘텐츠의 실행 절차 및 방법에 있어 아래 세 가지 스펙을 제시한다.

Launch

런치는 학습 활동을 하기 위해 최초로 학습자료를 불러오는 것에 있어 특정 LMS의 구현방법에 구애받지 않고 학습자료의 전달이 가능한 방법을 제시한다. SCORM 1.2에서는 학습자료들의 순서화(Sequencing)와 이동(Navigation)에 대한 표준안에 대해서는 언급하지 않으나 이러한 순서화와 콘텐츠 이동의 책임을 LMS가 가져야 하는 것에 대해서는 분명히 하고 있다. 즉, 어떠한 학습자료를 런치시켜야 하는가에 대해서는 콘텐츠 구조, 학습자 정보 등을 고려하여 LMS가 결정하게 되는 것이다. 콘텐츠의 순서화와 이동을 LMS가 담당 함으로서 콘텐츠간의 독립성을 유지하여 콘텐츠의 재사용이 가능하게 되는 것이다.[3]

LMS는 한 번에 한개의 SCO를 런치시켜야 하며 동시에 두 개 이상의 SCO가 활성화 된 상태로 있지 못한다. SCO가 일단 런치 되면 SCO는 LMS의 클라이언트가 되는 인터넷 브라우저에서 플러그인으로 구현된 API Adapter를 찾고 통신을 위해 API Adapter를 초기화 한다.

API Adapter

SCORM은 AICC의 CMI001 Guidelines for Interoperability에 정의된 실행환경을 기초로 한다. API는 LMS와 SCO간의 표준화된 통신 방법을 제공하는데 SCO의 입장에서 볼 때에 LMS와 통신하기 위해 LMS가 어떻게 구현되어 있는지 전혀 알 필요가 없게된다. SCO는 단지 표준화된 API를 구현한 API Adapter와 만 통신을 하면된다. 따라서 콘텐츠 개발자는 LMS의 구현에 구애받지 않고 콘텐츠의 내용에만 집중하여 훨씬 효율적으로 콘텐츠를 개발 할 수 있을 뿐 아니라 재사용 가능한 콘텐츠를 개발할 수 있게 된다.

API Adapter가 구현해야 할 함수들은 세 영역으로 나누어지는데, LMSInitialize(""), LMSFinish("")을 포함하는 실행상태 함수들과 LMSGetLastError(), LMSGetErrorString(parameter), LMSGetDiagnostic(parameter) 등 에러 처리를 위한 상태관리 함수 그리고, LMSGetValue(data model element), LMSSetValue(data model element), LMSCommit("") 등 데이터 전송 함수들이다.

이 API 함수들을 통하여 LMS와 콘텐츠가 통신할 수 있기 위하여 몇가지 지켜야 할 사항들이 있다. 첫째는 API Adapter를 포함하는 API 함수들은 SCORM 스펙에서 제시하는 정확한 함수 이름으로 구현되어 있어야 하며 둘째, API Adapter의 구현은 각 LMS에 따라 다를 수 있으나 LMS는 SCO로 하여금 API Adapter를 찾을 수 있도록 하기 위해 DOM(Document Object Model) 윈도우 내에 "API" 라는 이름으로 API Adapter를 위치 시켜야 한다. 셋째, 최초 SCO가 Client 브라우저로 전달되었을 때 SCO는 LMS와의 통신을 위해 API Adapter를 찾고 API Adapter를 초기화 해야 한다. 넷째, SCO에서 API 함수들을 호출하는 것은 일반적으로 javascript를 통해서 이루어 진다.[2]

CMI Data Model

컨텐츠와 LMS간의 데이터 전달에 있어 표준화된 데이터 모델을 사용함으로써 특정 콘텐츠가 LMS의 환경에 좌우되지 않고 콘텐츠 정보를 LMS와 교환할 수 있게 된다. API에서와 마찬가지로 SCORM에서는 데이터 모델에 있어서도 AICC의 CMI Data Model을 따르고 있다. 따라서 현재 SCORM에서 사용하는 데이터 모델에서 모든 요소들은 'cmi'라는 문자로 시작된다.

SCORM은 데이터 모델을 필수요소와 선택요소로 구분 하였다. SCO의 입장에서 LMS에서 실행되기 위한 최소한의 요구는 DOM 윈도우 내에서 API를 찾고 LMSInitialize()와 LMSFinish() 두 함수만을 정상적으로 호출할 수 있으면 된다. LMSInitialize()와 LMSFinish()는 함수 인자 없이 사용되는 함수이기 때문에 어떤 데이터 모델도 사용하지 않고 LMS에서 실행될 수는 있는 것이다. 하지만 LMS에서는 데이터 모델 요소들

중 필수 요소에 해당하는 것은 최소한 구현되어 있어야 한다.

데이터 모델 중 필수로 구현되어야 할 요소들은 다음과 같은 것들이다. cmi.core, cmi.core_studentid, cmi.core.lesson_location, cmi.core.exit, cmi.core.credit, cmi.core.lesson_status, cmi.core.score 이다.[2]

(그림 1) SCORM RTE 개념도

이상의 SCORM RTE의 구성요소들 간의 실행 메카니즘을 보면 (그림 1)과 같다

2.2. iOneLMS의 기능 및 구성

iOneLMS의 개발 환경 및 Platform

iOneLMS는 웹을 기반으로 하는 사이버 교육 운영시스템으로 인터넷/인트라넷을 통해 관리자, 교수자, 학습자 등 누구나 접근 가능하며 학습활동과 함께 관리가 가능하다.

iOneLMS는 J2EE(Java 2 Enterprise Edition)를 기반으로 개발되었다. J2EE는 기본적으로 J2SE 환경을 기반으로 대규모 기업환경의 서버 애플리케이션 개발에 적합한 API 라이브러리 및 서버 애플리케이션 요구사항 등을 정의하고 있는데, J2EE의 핵심기술이라 할 수 있는 EJB를 비롯하여 iOneLMS에서 적용된 J2EE 엔터프라이즈 API들은 엔터프라이즈 자바빈즈(EJBs), JDBC, 서블릿/JSP, JNDI(Java Naming and Directory Interface) 등이다. iOneLMS는 J2EE 서버로서 HP-as를 채택하였고 HP-as는 리소스 풀링, 퍼포먼스 등 J2EE 서버가 제공해 주는 시스템 레

벨의 서비스를 안정적으로 제공해 준다.

iOneLMS는 또한 EJB 컴포넌트 기반으로 설계 및 구현 되었다. EJB 스펙에서는 EJB 아키텍처를 '컴포넌트에 기반한 분산 비즈니스 애플리케이션 개발과 실제 운용(Deployment)을 위한 컴포넌트 아키텍처'로 정의하고 있다. 즉 EJB는 분산 비즈니스 애플리케이션 작성시에 사용될 소프트웨어 모듈(컴포넌트 모델)이며 동시에 EJB 자체의 기능 구현과 실행환경에 배치되어 실행되는 속성 설정(Deployment Descriptor)을 분리함으로써 특정 실행환경에 국한되지 않는 확장성 및 재사용성을 갖는 빈을 개발할 수 있게 한다.[4]

iOneLMS가 이러한 확장성 및 재사용성을 갖는 EJB 컴포넌트에 기반하여 개발됨으로서 SCORM 표준화에 순응하는 LMS로 변환함에 있어 새로운 기능에 해당하는 컴포넌트를 독립적으로 개발 및 재사용하여 구현할 수 있었다.

iOneLMS의 기능

iOneLMS는 학습기능과 커뮤니티 형성기능, 테스트 기능 및 평가기능을 보유함으로써 실제 교실 수업에서 할 수 있는 모든 기능을 인터넷/인트라넷상의 가상공간에서 실현 가능하게 구현하였다. 또한 사용자의 권한에 따라 관리자, 강사, 학과 운영자, 학습자 네가지 모드를 지원하며 각 모드에 맞는 메뉴 체계에 따라 관리 및 강의와 학습이 쉽게 이루어지도록 구성되어 있다.

2.3. LMS 표준화의 의미

LMS를 표준화 한다는 것은 모든 LMS 시스템들이 같은 기술을 사용하고 동일한 기능, 동일한 인터페이스로 구현되어야 한다는 것을 의미하지는 않는다. SCORM은 Run-Time Environment 스펙에서 LMS의 표준화에 대해 다룰 때 LMS의 모든 기능에 대해 표준화를 언급한 것이 아니라 단지 콘텐츠의 재사용과 상호운용성, 접근성, 지속성 등의 SCORM 목표 달성을 위한 구현 방법론 및 구현 예제만을 제시하고 있다.

SCORM은 LMS 측면에서 이러한 목표를 달성하기 위한 핵심 개념을 콘텐츠와 LMS간의 일반화된 방법에 의한 커뮤니케이션과 상호작용으로

보고 있다. 따라서 이를 제외한 기타 LMS가 갖추어야 하는 기능과 사용자 인터페이스 및 구현을 위한 적용 기술 등은 얼마든지 개별 LMS의 특징을 가지고 개발 될 수 있다.

따라서 표준화를 고려하지 않고 개발된 LMS라 하더라도 SCORM RTE의 API와 데이터모델을 추가로 개발하고 콘텐츠의 진수(launch)과정을 조정하게 되면 SCORM 표준화에 상응하는 LMS로의 변환이 가능하다는 개념적 결론을 취할 수 있게된다. 이러한 개념에 근거하여 iOneLMS의 변환을 시도하였고, 실제 구현에 있어서도 같은 결론에 이를수 있었다.

본 논문에서 다루게 될 SCORM 표준화예로의 변환은 교수자에 의한 강의등록과 학습자의 강의 수강을 중심으로 이루어 지게 된다.

3. SCORM RTE 컴포넌트의 구현

SCORM RTE 컴포넌트의 구현은 다음과 같은 세 단계로 나누어 구분지을 수 있다. 첫째, 콘텐츠 들여오기(Content Importing) 단계이다. 이는 zip 형태로 압축된 PIF(Package Interchange File) 파일의 기 제작된 SCORM 표준화 콘텐츠를 LMS 또는 LCMS에 가져오고 그것이 가지는 콘텐츠 정보를 추출 및 저장하는 과정이다. 둘째, 강의등록 단계이다. 이는 개설된 특정 과목에서 강좌로서 SCORM 콘텐츠를 등록하는 과정을 말한다. 셋째, 강의 수강 및 학습 Data 수집 과정이다. 학습이 이루어 지면서 발생하는 학습 데이터의 수집과 이것을 가능하게 하는 LMS와 콘텐츠 간의 상호작용이 여기에 해당된다. 그 외에도 수집된 학습 데이터의 활용 등의 기능이 포함되어 있다.

이와같이 구현된 RTE 컴포넌트를 iOneLMS가 통합함으로써 SCORM 표준에 순응하는 LMS로의 기능을 갖출 수 있게 되었다.

3.1. 콘텐츠 들여오기

클라이언트에서 서버로 콘텐츠 전송 및 언패키징

(그림 2)에서와 같이 교수자는 iOneLMS의 학습객체관리 메뉴를 통하여 콘텐츠 파일을 서버로 올릴 수 있다.

(그림 2) 콘텐츠 들여오기 화면

SCORM 스펙 중 Content Aggregation Model에서는 콘텐츠의 패키징을 필수사항으로 두지는 않았으나 콘텐츠를 패키지화 할 것을 권하고 있다. 이렇게 압축파일의 형태로 묶여진 콘텐츠 덩어리를 PIF(Package Interchange File)파일이라 한다. 한 강좌를 이루는 콘텐츠 내에는 html, jpg, text 등 여러 형태의 여러 가지 파일들이 포함 될 수 있다. 이때 해당하는 모든 학습자료들을 하나의 패키지 형태로 만들 경우 이동성이 쉬워질 뿐 아니라 콘텐츠의 관리 역시 효과적으로 이루어질 수 있기 때문이다.[5]

클라이언트에서 서버로의 파일 업로드는 서블릿(Servlet) 즉, java로 작성된 CGI 프로그램으로 구현되었다. 이를 위해 java package "com.jspsmart.upload"와 그 패키지 내에 SmartUpload.class를 비롯한 다섯 개의 java class를 생성하였다. 이 java class들은 java.lang.Object를 상속하였고, 아파치에서 제공하는 class 라이브러리인 servlet.jar 내에 있는 javax.sevlet.http.*을 import하였다. 이렇게 작성된 SmartUpload.class가 콘텐츠 들여오기 기능에서 호출되는 JSP page에 importing되어 지정된 콘텐츠 파일을 업로드 하게된다. 업로드된 콘텐츠 파일은 1차적으로 서버의 임시 디렉토리 내에 저장되었다가 imsmanifest.xml을 파싱한 후 다시 서버상의 지정된 콘텐츠 리포지토리로 옮겨진

다.[6]

SCORM 콘텐츠 패키지 모델에서는 패키지화된 콘텐츠의 모든 정보를 imsmanifest.xml 파일에 포함하도록 하고 있으며 PIF 파일 내에서 imsmanifest.xml 파일을 루트 디렉토리 상에 위치하도록 하고 있다. 학습객체들의 상세 정보를 알아내기 위해 콘텐츠 패키지에서 imsmanifest.xml 파일을 찾고 패키지 파일 외부로 추출하기 위해 zip 형태의 파일 포맷에서 특정 파일을 찾고 풀수 있도록 하는 java.util.zip.*을 import한 PackageHandler.class를 생성하여 사용한다.

콘텐츠정보 파일(imsmanifest.xml) 파싱

imsmanifest.xml에 대한 접근이 가능하게 된 후에는 PIF가 포함하는 학습객체들의 정보를 LMS가 활용할 수 있도록 하기 위해 데이터베이스에 저장하는 과정이 필요한데 이에 앞서 정보추출을 위해 파서를 이용하여 imsmanifest.xml 파일을 파싱한다.

ADL은 imsmanifest.xml의 작성을 위해 자체에서 정의한 XML 스키마인 adl_cp_rootv1pl.xsd를 제시할 뿐 아니라 메니페스트 파일을 파싱할 수 있는 class가 포함된 라이브러리로 ADLParser.jar을 제공하고 있으며, 본 연구에서는 메니페스트 파일의 파싱을 위해 ADLParser.jar가 포함하는 class들을 포함하고 기타 여러 class들을 import하여 ManifestHandler.class를 생성하여 사용하였다.

ManifestHandler.class는 ADLParser.jar 내의 CPDOMParser class를 상속하였고, CPDOMParser는 ADLDOMParser를 상속하였는데 이들 두 클래스를 포함한 org.adl.parsers.dom 패키지는 org.apache.xerces.parsers 패키지 내의 DOMParser 클래스를 상속하여 구현되었다. 즉, iOneLMS의 ManifestHandler.class는 아파치(Apache)에서 제공하는 라이브러리인 xerces.jar의 DOMParser 및 XMLParser Class를 상속하여 만들어 졌다고 할 수 있다. 이들 클래스들이 org.w3c.dom과 org.xml.sac 패키지를 포함하고 있기 때문에 파싱이 완료된 imsmanifest.xml을 W3C DOM의 Document object로 접근할 수 있

게 된다.

```
try {
    // Parse the file
    parse( sourceToParse );
    document = getDocument();
    :
}
```

(그림 3) ManifestHandler.class (parse)

(그림 3)에서와 같이 CPDOMParser 클래스에서 상속받은 parse() 메소드로 메니페스트 파일을 파싱하고 getDocument() 메소드를 통해 W3C DOM의 Document 인터페이스를 획득하게 된다.

```
Node contentNode = document.getDocumentElement();
NodeList contentChildren = contentNode.getChildNodes();
this.manifest.fillManifest( contentNode );
:
for ( int i = 0; i < contentChildren.getLength(); i++ ) {
    Node currentNode = contentChildren.item(i);
}
```

(그림 4) ManifestHandler.class (DOM)

Document 인터페이스는 또한 W3C Node 오브젝트를 상속받아 구현되었기 때문에 (그림 4)에서 보는 바와 같이 메니페스트 파일 내의 각 아이تم들을 노드로 접근할 수 있게 된다. 최종적으로 콘텐츠 정보들은 java.util.Vetor 타입으로 리턴된다.

학습객체정보 관리 및 순서화 파일 생성

메니페스트 파일이 가지는 xml 태그 중 <Organizations> 아이تم은 콘텐츠의 구조를 나타내며, <Resources> 아이تم은 SCO, Asset 등 학습객체에 대한 정보를 가진다. <Resources> 태그 및 그 하위 태그가 가지는 콘텐츠 및 콘텐츠가 포함하는 SCO들에 대한 상세 정보를 LMS가 쉽게 접근하고 활용할 수 있도록 데이터베이스에 저장하고 관리하기 위하여 데이터베이스에 관련 테이블을 추가한다.

콘텐츠 전체에 대한 정보 즉, PIF 파일 자체에 대한 정보를 위하여 COURSEINFO 테이블을 생성하였고, 개개의 SCO 또는 Asset에 대한 정보의 관리를 위하여 ITEMINFO 테이블을 생성하

였다. COURSEINFO 테이블은 코스 아이디, 코스 제목, 코스 네비게이션 방식 및 활성화 유무 등의 필드를 가지고, ITEMINFO 테이블에서 관리하는 정보는 아이تم 제목, 콘텐츠 리포지토리 상에서 실제 위치를 나타내는 URI, 콘텐츠 타입, 요구되는 선행학습 등에 대한 정보이다.

COURSEINFO 테이블과 ITEMINFO 테이블에 대한 데이터 관리를 위하여 이들과 매칭되는 ECourseinfo EJB 엔티티빈과 EIteminfo EJB 엔티티빈을 생성하여 디플로이 하였고, 학습객체정보 관리에 있어서의 비즈니스 로직과 트랜잭션 처리를 위해 SScormImportTx, SIteminfoTx 등 두 개의 EJB 세션빈을 작성하였다.

ManifestHandler.class로부터 Vector 타입으로 콘텐츠 정보를 받은 JSP 페이지가 콘텐츠 정보를 데이터 베이스에 저장하기 위한 EJB 클라이언트 MScormImport.class의 ScormImport() 메소드를 호출하고 MScormImport.class는 엔티티빈의 DAO(Data Access Object) 객체를 생성하여 세션빈의 홈 인터페이스를 찾은다음 리모트 인터페이스를 통하여 적절한 세션빈의 메서드를 호출하게 된다. 다시 세션빈은 관련 엔티티빈의 객체를 생성하고 최종적으로 홈 인터페이스의 create() 메소드를 호출함으로 데이터베이스에 콘텐츠 정보 및 SCO정보들을 생성한다.

(그림 5) 콘텐츠 들여오기 컴포넌트 구성도

이상의 콘텐츠 들여오기 단계에서 구현된 컴포넌트 구성도를 보면 (그림 5)와 같다.

3.2. 강의 등록

강의 등록이란 개설된 과목에 강좌를 추가하는 과정을 말한다. 강의 등록은 첫째는 등록할 강좌를 LMS로 들여오는 동시에 단 하나의 지정된 과목의 강좌로 등록하는 것과 둘째, 강의 들여오기 과정을 통해 콘텐츠 리포지토리에 저장된 학습 콘텐츠를 조회하여 지정한 과목의 강좌로 등록하는 것이다. 이렇게 콘텐츠 리포지토리 내의 강좌를 과목에 등록할 경우는 과목 수에 상관 없이 하나의 콘텐츠를 복수개의 과목에 등록할 수 있을뿐 아니라 강좌의 내용 구성 변경이 용이해진다.

표준화로 변환하기 전의 iOneLMS는 첫 번째 방식의 강의 등록만 가능하였으나 SCORM 표준으로 변환하면서 콘텐츠 재사용의 기회를 높이기 위해 두 번째 방식 즉, 강의 들여오기와 강좌 등록 기능을 분리하여 구현하였다. 이렇게 함으로써 최소한의 기능을 갖춘 콘텐츠 리포지토리를 추가로 구현할 수 있게되었다.

변환 전의 iOneLMS는 데이터베이스 LECTURES 테이블을 중심으로 강좌정보를 관리하고 있었다. 따라서 콘텐츠 리포지토리가 보유하고 있는 콘텐츠를 강좌로 등록하고 학습으로까지 이어지도록 하기 위해서 리포지토리의 콘텐츠 아이디를 LECTURES 테이블에서 외부키로 필드에 포함하였다. 이렇게 함으로써 기존의 데이터베이스 및 프로세스의 변경을 최소화할 수 있었다.

리포지토리에 저장된 콘텐츠를 강좌로 등록하기에 앞서 리포지토리 조회 및 콘텐츠 상세 조회 기능을 제공한다. 리포지토리가 보유하고 있는 콘텐츠를 조회하기 위하여 이미 작성된 ECourseinfo EJB 엔티티빈과 Elteminfo EJB 엔티티빈을 사용하며 SScormImport, SIteminfo 두 개의 비즈니스 로직을 구현한 EJB 세션빈을 작성하였다. EJB 클라이언트로서 MViweScormInfo.class는 해당 비즈니스 로직을 가지는 세션빈을 호출하고 세션빈은 엔티티 빈으로부터 클라이언트가 요청하는 콘텐츠 정보를 ResultSet으로 반환해 준다.

3.3. 강의 수강 및 학습 Data 수집

강의수강은 과목에서 강좌로 등록된 SCORM Contents를 학습자가 오픈하고 학습하는 일련의 과정 가운데 발생하는 모든 활동들을 포함한다.

콘텐츠의 실행 및 초기화

일반적으로 LMS는 html을 통한 학습 및 특정 형태의 콘텐츠를 실행하기 위해 자체의 학습용 Player나 플러그인 프로그램을 사용한다. 그러나 표준화된 LMS와 콘텐츠를 통한 학습이 이루어지기 위해서는 단순히 html을 불러오거나 플러그인 프로그램을 실행하는 것 만으로는 되지 않는다. 선행연구에서 살펴 보았듯이 콘텐츠와 LMS 간의 커뮤니케이션을 위해 클라이언트와 서버측에서 동시에 프로그램이 초기화 되어야 한다.

클라이언트에 콘텐츠가 로드되면 콘텐츠는

```

<object classid ="clsid:8AD9C840-044E-11D1-B3E9-00805F499D93"
width ="0" height="0" id="APIAdapter"
codebase ="http://java.sun.com/products/plugin/1.3/jinstall-13-win32.cab#Version=1.3.0.0">
<param name = "code" value = "org/adl/samplerte/client/APIAdapterApplet.class" >
<param name = "codebase" value = "/student/scormlect" >
<param name = "type" value="application/x-java-applet:version=1.3">
<param name = "mayscript" value="true" >
<param name = "scriptable" value="true" >
<param name = "archive" value = "cmdatamodel.jar,lmsclient.jar,debug.jar" >
</object>
    
```

(그림 6) API Adapter 객체 선언

API Adapter를 찾는데 콘텐츠에 포함된 자바스크립트가 이 역할을 하게 된다. 어떠한 콘텐츠라도 이 API Adapter를 찾을 수 있도록 하기 위해 SCORM은 브라우저 상의 API Adapter를 "API"라는 이름을 가지는 DOM 객체로 접근하도록 하였으며 아래와 같은 코드로 구현 가능하다

```
API = this.document.APIAdapter;
```

콘텐츠에 포함된 자바 스크립트는 DOM 객체로 선언된 API를 아래의 코드와 같은 함수를 사용하여 찾은 다음 API Adapter의 메서드인 LMSInitialize("")를 호출함으로써 통신을 초기화할 수 있다

```
var api = getAPIHandle();
var result = api.LMSInitialize("");
```

iOneLMS는 SCORM 콘텐츠의 실행을 위해 세 개의 프레임을 가지는 하나의 윈도우를 제공한다. 상위 프레임에는 (그림 6)과 같이 <applet> 또는 <object> 태그를 사용하여 API Adapter Applet을 포함하고, 메니페스트 파일의 Organization Item에 기록된 순서에 의하여 학습 흐름을 제어할 수 있도록 사용자 인터페이스를

보여준다. 마지막으로 아래 오른쪽 프레임에는 jsp로 작성된 네비게이션 엔진과 함께 학습할 학습객체를 디스플레이 하도록 하였다. (그림 7)은 iOneLMS에서 강의수강 화면을 보여준다.

학습자가 콘텐츠에 대한 학습을 처음 시작하는 시점에 iOneLMS는 콘텐츠에 대한 학습자의 학습상황을 트래킹하기 위한 초기 정보를 데이터베이스 테이블 USERCOURSEINFO와 USERSCOINFO에 기록한다. USERCOURSEINFO 테이블은 학습자가 수강중인 코스 리스트를 관리하고, USERSCOINFO는 코스내에 포함된 개개의 SCO에 대한 학습진도율, 시험점수 등 학습 상태를 관리한다. 이상의 두 DB 테이블은 SCORM으로의 변환을 위해 새롭게 추가 생성된 것으로 각 테이블에 대한 EJB 엔티티빈을 생성하고 학습상태 정보 관리를 위한 비즈니스 로직을 담은 세션빈을 생성하여 정보조회 및 추가가 가능하게 하였다.

또한 학습이 진행되는 동안 LMS와 콘텐츠가 통신하는 과정에서 CMI 데이터 모델을 통하여 표준화된 통신이 가능하게 하기 위하여 코스내에 포함되는 SCO에 대한 CMI 데이터 모델 오브젝트 파일을 생성하였다. 생성된 CMI 데이터 모델 오브젝트 파일은 SCORM 스펙에서 제시하는 모든 아이템들을 포함하고 있으며, SCO가 런치된 후 API Adapter의 LMSInitialize() 메서드가 호출되는 시점에 LMSCMIServlet.class 서블릿을 통하여 서버에서 읽혀진 후 다시 클라이언트의 APIAdapter 애플릿에 오브젝트로 기록되어 클라이언트에서 접근이 가능하게 된다.

학습의 흐름 및 학습객체간 이동

콘텐츠가 LMS에 의해 런치되고 API가 초기화되고나면 학습자의 학습 진도 및 SCO간의 이동은 학습자 자신에 의해 컨트롤될 수 있을뿐 아니라 LMS에 의해서 컨트롤될 수도 있다.

iOneLMS는 콘텐츠 패키지 내의 메니페스트 파일이 제공하는 콘텐츠 정보를 기반으로 순차적인 학습흐름을 제공할 뿐 아니라 SCO들에 대한 계층적 구조를 학습자에게 제시하여 주고 학습자로 하여금 자유롭게 SCO들 간의 이동이 가능할 수 있도록 하였다.

(그림 7) 강의수강

제공한다. 아래 왼쪽 프레임에는 Organization Item에 기록된 순서화에 무관하게 학습자 임의로 네비게이션이 가능하도록 콘텐츠의 트리구조를

현재 런치된 SCO에 대한 학습이 완료되면 콘텐츠는 학습 결과를 LMS 서버에 알리고 LMS는 이 정보에 근거하여 적당한 다른 SCO를 런치시킴으로서 LMS 주도의 학습흐름의 제어가 가능하나 LMS에 의한 학습객체간의 이동이 이루어지기 위해서는 콘텐츠 역시 저작시 이것을 고려하여 저작되어야 하며 그렇지 못할 경우는 오히려 학습의 흐름 제어가 불가능하여 학습 자체가 어려워질 수 있으므로 주의 하여야 한다.

학습 정보 수집

SCO가 런치되면 API함수 중 LMSInitialize() 함수를 호출하는데 이때 API Adapter는 서버의 서블릿으로부터 CMI 데이터 객체를 전달받아 생성한다.

학습이 진행되는 동안 콘텐츠는 콘텐츠 내에 포함된 javascript를 통하여 API Adapter의 LMSSetValue() 또는 LMSGetValue() 함수를 호출하면서 CMI 데이터 객체에게 정보를 요청하거나 기록한다. LMSFinish() 또는 LMSCommit() 등의 API 함수가 호출될 때 API Adapter는 현재까지의 학습상태를 보유하고 있는 CMI 데이터 오브젝트를 ObjectOutputStream으로 서버의 서블릿에게로 전달하고 서블릿은 ObjectInputStream으로 받아 오브젝트 파일로 기록함과 동시에 데이터베이스에 저장한다. LMS 서버와 클라이언트가 상호간에 Object Input/Output Stream을 통하여 학습상태정보를 교환하게 함으로서 서버의 보안을 강화하고 성능의 향상을 기하도록 하였다.

학습 정보는 데이터베이스의 USERCOURSEINFO 테이블에 SCO 단위로 기록된다. LMSCMIServlet 서블릿은 데이터베이스와의 트랜잭션을 위하여 MUpdateScoinfo 클라이언트 빈을 통하여 EJB 세션빈에 접근한다.

CMI 데이터 모델은 ADL에서 제공하는 org.adl.datamodels 패키지 내의 모든 클래스를 그대로 재사용하여 구현 가능하다

3.4 학습 데이터의 활용

하나의 코스에 대해 학습이 진행되면서 LMS에 의해 추적되는 학습정보는 SCO 단위의 학습

정보이다. 따라서 많은 SCO들을 포함하는 코스들에서는 많은 양의 학습 데이터 수집이 이루어질 수 있다. 또한 개별 SCO들에 대한 학습정보 수집의 주체는 LMS가 아니라 콘텐츠이다. 현재의 SCORM 스펙에서는 LMS와 콘텐츠간의 통신의 시작은 콘텐츠가 먼저 하도록 되어 있기 때문이다. SCORM 버전 1.2에서는 LMS가 콘텐츠에 먼저 데이터를 요청하는 메카니즘은 없다. 따라서 콘텐츠에서 LMS로 정보를 주지 않으면 LMS는 학습정보를 가질 수 없게 된다.

(그림 8) 학습진행정보 보기

iOneLMS는 이러한 학습 데이터의 수집이 콘텐츠에 의존적인 것을 고려하여 (그림 8)에서와 같이 SCO 단위의 학습상태를 나타내는 "cmi.core.lesson_status" 정보를 제공하도록 하였다.

4. 결론 및 향후 연구

Java의 객체지향 특성 뿐 아니라 J2EE의 분산 컴포넌트 기술에 기반하여 API Adapter, 데이터 모델, 데이터 관리 등의 SCORM 표준화 핵심 기능들을 EJB와 java 서블릿, java 애플릿 및 JSP 기술들을 사용하여 컴포넌트로 개발하였다. 이 과정에서 ADL에서 제공되는 CMI 데이터 모델, DOM Parser 등의 라이브러리를 충분히 활용하였다. 개발된 컴포넌트를 iOneLMS에 적용하는 과정에서 표준화 콘텐츠 및 학습진행 상태 관리를 위해 데이터베이스에서 관리해야할 데이터의 많은 부분을 기존의 데이터베이스 정보와는 관련

성을 줄이고 독립성을 높여 구현할 수 있었다.

본 논문에서 구현한 SCORM에 순용하는 LMS는 SCORM 표준화 콘텐츠의 효과적인 관리방안에 대해서는 충분한 연구가 이루어지지 못했다. 본 연구에서는 콘텐츠를 체계적인 분류에 의해 관리하지 못했으나 SCORM 콘텐츠를 저장하는 콘텐츠 리포지토리에서 콘텐츠가 담고있는 학습 내용, 학습 대상자, 콘텐츠의 난위도 등에 따른 체계적인 콘텐츠 관리에 대한 연구가 필요할 것으로 보여진다.

또한 콘텐츠와 LMS의 통신을 통해 콘텐츠로부터 받아들이는 학습상태 정보의 활용적인 측면에서 코스 네비게이션, 추가 학습정보 제공 등에 대한 더 많은 연구가 이루어질 것을 기대한다. ADL에서 2002년 11월 27일에 'ADL SCORM Version 1.3 Application Profile' Working Draft Version을 발표하여 IMS Simple Sequencing Specification에 기초한 Sequencing과 코스 네비게이션을 적용하기 위한 공개 테스트가 이루어지도록 하였다.[7] SCORM Version 1.3의 발표에 따라 Sequencing과 네비게이션에 대한 더 깊은 연구가 이루어 질 것으로 기대한다.

참 고 문 헌

[1] 이준, LCMS(Learning Content Management System) 기반의 e-Learning 개발과 적용, 한국교육학술정보원, 2002

[2] ADL, Sharable Content Object Reference Model version 1.2, 2002 http://www.adlnet.org/ADLDOCS/Documents/SCORM_1.2_RunTimeEnv.pdf, 2001

[3] 김용만 · 김현철, SCORM 스펙을 이용한 학습 관리 시스템 설계. 한국컴퓨터교육학회 하계 학술발표논문집 제6권제2호, 173 - 174, 2002

[4] Sun Microsystems. Enterprise JavaBeans(TM) Specification Proposed Final Draft 2.1, <http://java.sun.com/products/ejb/docs.html#specs>, 2002

[5] ADL, Sharable Content Object Reference Model version 1.2, http://www.adlnet.org/ADLDOCS/Documents/SCORM_1.2_CAM.pdf, 2001

[6] ADL, ADL Sample Run-time Environment version 1.2.1, [http://www.adlnet.org/ADLDOCS/OtherSCORM\(TM\)Version1_2SampleRTEVersion1_2_1.zip](http://www.adlnet.org/ADLDOCS/OtherSCORM(TM)Version1_2SampleRTEVersion1_2_1.zip), 2002

[7] ADL, ADL SCORM Version 1.3 Application Profile Working Draft 0.9, <http://www.adlnet.org>, 2002

김 기 석

1984 서울대학교
전자계산기공학과(공학사)

1987 서울대학교
컴퓨터공학과(공학석사)

1992 서울대학교 컴퓨터공학과
(공학박사)

1994~1998 삼성 SDS 정보기술연구소 책임연구원
2000~현재 한동대학교 전산전자공학부 교수
관심분야: Adaptive Education, e-learning 표준화
멀티미디어교육

E-Mail: peterkim@handong.edu

김 강 석

1997 계명대학교
경영정보학과(학사)

2001~현재 한동대학교
정보통신학과 석사과정

관심분야: 가상교육, e-learning 표준화
E-Mail: jdpeter@hanmail.net