

# SCORM 스펙을 이용한 학습관리 시스템 설계 및 구현

김용만\* · 김현철\*\*

## 1. 서론

ADL(2001)의 SCORM 보고서에 따르면 e-Learning 기술 표준에 관한 연구는 전통적인 CBI(Computer Based Instruction)부터 e-Learning에 이르기까지의 개발 방법 및 학습 효과에 대한 반성에서 비롯되었다. 그간 컴퓨터 환경에서의 수업개발에 너무나 많은시간과 비용이 소요된 것이 사실이다. 이렇게 고비용적인 개발 구조를 극복하기 위해서, 한번 만들어진 콘텐츠의 재사용과 제3자가 만든 콘텐츠의 공유를 모색하게 되었다. 전통적인 CBI에서는 콘텐츠와 그 콘텐츠를 만든 애플리케이션(application)이 결합되기 때문에 콘텐츠의 재사용이 매우 어려웠으며 제3자가 제작한 콘텐츠를 사용하는 것은 더욱더 힘든 일이었다. e-Learning에 있어서도 이러한 문제는 동일하게 나타난다. 콘텐츠가 그것을 제작하는데 사용된 특정 LMS에 종속됨으로써 다른 LMS에 사용할 수 없으며, 다른 LMS에서 사용되던 콘텐츠를 공유할 수가 없는 것이다. 콘텐츠의 재사용 및 공유를 위해서는 우선 콘텐츠가 특정 플랫폼으로부터 분리되어야 하며, 어떠한 플랫폼에서도 사용될 수 있어야한다. e-Learning 기술표준은 콘텐츠 및 플랫폼에 일정한 기준을 세워줌으로써 이 기준에

의해 제작된 콘텐츠는 플랫폼에 구애받지 않고 재사용 및 공유가 가능하도록 도와주며 플랫폼간 연동이 쉽게 이루어지게 하는 역할을 한다. 콘텐츠의 재사용 및 공유 외에 콘텐츠를 플랫폼으로부터 분리함으로써 얻을 수 있는 이점은 학습자의 특성에 맞는 적응적인 수업의 구현이 보다 용이해진다는 점이다.

e-Learning 플랫폼으로서 LMS는 몇 가지 제한점을 안고 있다. 첫째, 외부에서 제작된 콘텐츠가 LMS에 자유롭게 탑재되려면 LMS가 콘텐츠의 특성을 기술적으로 이해할 수 있어야 한다는 점이다. LMS의 종류가 매우 다양하고 콘텐츠 저작 도구들 역시 다양하다는 점을 고려하면 이는 해결되어야 할 문제이다. 둘째, LMS와 콘텐츠가 자유롭게 탑재될 수 있는 범위가 제한적일 경우 콘텐츠의 공유가 어려워진다. 기술적 환경만 지원된다면 제3자가 만든 콘텐츠를 가져다 사용할 수 있는 경우에도 외부에서 제작한 콘텐츠의 수용이 어려워 처음부터 새롭게 콘텐츠를 제작함으로써 시간과 비용을 낭비하게 되는 문제점이 있다. 셋째, LMS는 기본적으로 학습이 잘 일어날 수 있도록 지원하는 기능에 초점을 맞추고 있기 때문에 콘텐츠의 양이 일정 수준 이상 증가하게 될 때 이를 효율적으로 관리하기 어렵다. 콘텐츠의 저장, 검색 기능이 제한적이어서 대량의 콘텐츠를 관리하는데 적합하지 않다. 넷째, LMS는 외부에

\* 고려대학교 교육대학원 컴퓨터교육전공 석사과정  
 \*\* 고려대학교 컴퓨터교육과 부교수

서 만들어진 콘텐츠를 학습자에게 그대로 전달만 하기 때문에 학습자의 특성에 맞는 적응력 있는 학습 환경을 제공하기 어렵다.[3] 이러한 LMS의 문제점을 해결하고자 ADL(Advanced Distributed Learning)에서는 IMS(Instructional Management System Global Learning Consortium), AICC(Aviation Industry CBT Committee) 등 e-Learning 기술표준을 중심으로 SCORM(Sharable Content Object Reference Model)이라는 표준안을 내놓기에 이르렀다. SCORM 스펙의 핵심은 공유가능한 콘텐츠 객체(Sharable Content Object)를 통한 자원의 재사용 가능성을 높이고, 학습객체를 적절하게 관리하여 수업개발의 효율성을 높이는데 있다고 볼 수 있다.

ADL-SCORM은 교육과정에 적합한 구조를 제시하며 학습 객체를 통합하여 어떻게 과정을 만들 수 있을 지에 대한 방법과 평가를 제공한다. 또한 각각의 LO(Learning Object)는 무엇이며, 어떻게 시작되었으며, LMS와 어떻게 상호작용하는지, LMS와 상호 교환할 수 있는 LO에는 어떤 종류가 있는지에 대한 지침을 보여준다.

본 연구에서는 ADL에서 제시한 SCORM 스펙을 이용하여 학습자원의 집합체를 구조적으로 구성한 XML형식의 manifest.xml파일을 포함하는 콘텐츠 패키지(contents package)를 가지고 특정 LMS에만 국한되어 사용되는 학습자원이 아닌 이기종(異機種) 시스템이나 툴(Tool)간의 디지털 학습자원을 교환하는 표준화된 방법을 제공하는 시스템 설계를 기반으로 구현하고자 한다.

본 연구에서는 SCORM 스펙에 맞게 작성된 콘텐츠 패키지를 포함한 ZIP형태의 파일을 LMS에서 어떻게 SCO(Sharable Content Object)로 표현하며 SCO와 LMS간의 통신 매커니즘을 보이기 위해 LMS를 구현한다.

## 2. SCORM의 핵심 구성 요소

SCORM(Sharable Content Object Reference Model)은 미 백악관과 국방성이 군 및 국가기관의 e-Learning을 위한 표준안을 마련하기 위해 설립한 ADL(Advanced Distributed Learning Initiative)에서 개발한 모델로 2001년 10월 1.2버전을 공개했다. SCORM은 AICC(미 항공산업 CBT 위원회)와 IMS 등에서 이미 개발된 많은 스펙과 표준안들을 기반으로 개발했으며, 콘텐츠, e-Learning 기술, 시스템간의 통신 및 데이터 교환의 구현을 위한 프레임워크(framework)를 제시하고 있다. 이의 궁극적인 목적은 교육콘텐츠와 시스템간의 상호운영성, 재사용성, 제어성 등을 확보함으로써 사용자에게 대한 질 높은 서비스를 제공하는데 있다.

2001년 10월에 발표된 SCORM version 1.2는 크게 개요, Content Aggregation Model, 그리고 Run-Time Environment의 세 부분으로 이루어져 있다. Content Aggregation Model부분에서는 SCORM이라는 콘텐츠 모델에 대한 설명, 학습콘텐츠에 대한 정보인 메타 데이터에 대한 스펙, 코스웨어를 유통하기 위한 표준 포맷(Package)에 대한 스펙 등을 담고 있으며, Run-Time Environment부분에서는 학습객체의 실행과 종료 방식, 콘텐츠와 시스템간의 데이터 교환을 위한 기능(function), 데이터모델(학습자 메타 데이터) 등에 대한 스펙을 담고 있다. 본 연구에서는 SCORM version1.2에서 소개된 Content Aggregation Model과 Run-Time Environment에 대하여 간략하게 알아보고 이들을 이용한 LMS의 설계를 기반으로 구현을 통해 확인하고자 한다.

### 2.1 SCORM 콘텐츠 집합 모델(SCORM Content Aggregation Model)

교수설계자(Instruction Designer)와 개발자

(Developer)에게 학습 경험을 전달하기 위해 학습 자원(resources)을 통합(aggregation)하는 방법을 제공하는 교육학에 중립적인 방법론이 바로 SCORM 콘텐츠 집합 모델(SCORM Content Aggregation Model)이다. 콘텐츠 집합 모델은 콘텐츠 모델(Content Model), 메타 데이터(Meta-data), 콘텐츠 패키징(Content Packaging)으로 구성되어 있다.

콘텐츠 모델은 학습 경험을 담고 있는 콘텐츠 구성 요소를 정의하는 명명법(nomenclature)이라고 할 수 있다. 이 모델은 Asset, SCO(Sharable Content Object), Content Aggregation인 세 요소로 구성된다. 이들 구성요소에서 Asset은 학습 콘텐츠를 구성하는 미디어, 텍스트, 이미지, 사운드, 웹 페이지, 평가문항, 웹 클라이언트로 전달 가능한 데이터 등의 전자적 표현집합을 말한다. 또한 학습 자원을 Asset으로 표현한다.[4]

패키지(Package) 자체에 대한 정의 및 패키지에 대한 메타 데이터, 콘텐츠 구조, 행위를 정의하는 선택 구성 부문, 패키지안에서 자원을 가리키는 리스트 등과 같은 내용을 포함하고 있는 적하목록 파일(manifest file)이다. 또한 XML 기반의 적하목록을 생성하는 방법을 포함하고 있으며 적하목록 및 관련된 모든 물리적 파일을 압축파일(ZIP-files)이나 CD-ROM등으로 패키지화(packaging)하는 것에 대한 지침을 포함한다[1].

## 2.2 SCORM 실행 환경(SCORM Runtime Environment)

SCORM 실행환경은 공유 가능한 콘텐츠 객체 기반의 학습 콘텐츠와 LMS간의 상호운영성을 위한 기술적 방법을 제공한다. SCORM의 필수사항은 콘텐츠를 만들어 내는데 사용되는 도구가 무엇이든지 간에 다양한 LMS사이에서 상호 운영 가능한 학습 콘텐츠가 되어야 한다는 것이다. 이를 가능하게 하기 위해서는 LMS를 실행하는

동안 LMS와 콘텐츠사이에 상호교환되는 미리 정의된 데이터 요소와 LMS간의 통신을 위해 콘텐츠에 대한 공통된 방법, 콘텐츠를 시작하기 위한 공통된 방법이 존재해야 한다.

SCORM 실행 환경의 세 가지 요소는 진수(Launch), Application Program Interface(API) 그리고 데이터 모델(Data Model)로 정의한다.

진수(Launch) 매카니즘은 LMS가 웹 기반 학습자원을 시작하기 위한 공통의 방법을 정의한다. 이 매카니즘은 LMS와 전달된 학습자원간의 통신을 수립하기 위한 책임과 절차를 정의한다. 통신 프로토콜은 공통 API의 사용을 통해 표준화된다. API는 LMS와 공유가능한 콘텐츠 객체(SCO)사이에서 접수나 시간 제한과 같은 자료를 가져오고 설정하기 위해 사용되며 학습자원의 상태(예를 들자면 초기화, 종료, 에러 상황의 상태)를 LMS에게 알리는 통신 매카니즘이다[1].

API는 SCO를 이용할 수 있는 미리 정의된 기능의 유일한 집합이다. API는 SCO로부터 구현물의 상세 내용을 숨기고 재사용과 상호운영성을 증진시킨다. API의 목적은 크게 네 가지로 나뉘 볼 수 있다. 첫째, API는 단지 SCO가 사용하는 사전 정의된 기능의 집합이다. 둘째, SCO에서 구현의 상세한 내용을 숨김으로써, 재사용과 상호운영성을 촉진한다. 셋째, API Adapter는 기능적 소프트웨어의 일종이며, API의 기능을 구현한다. 넷째, API Adapter는 세 계층으로 구분된다. 실행 상태(Execution State), 상태 관리(State Management), 데이터 전송(Data Transfer)으로 나뉜다[4].

## 3. SCORM 스펙을 이용한 LMS 시스템 설계 및 구현

### 3.1 시스템의 설계

SCORM 스펙을 이용한 학습관리시스템은 클

라이언트측 요소인 웹브라우저(Web Browser)와 통신하는 서버측 요소인 LMS를 설계한다. (그림 1)은 SCORM 스펙을 이용하여 설계된 학습관리 시스템의 구성도를 나타내고 있다. 이 시스템에서 보듯이 학습 과정을 통한 학습자의 진행상황을 트래킹(tracking)하고 학습과정을 진수(launch) 시키고, 온라인 과정 전체에 대한 학습자 등록과 학습 계획대로 학습할 수 있도록 하는 중앙 집중적이고, 조직적인 접근을 가능케 하는 실행환경 서버측 요소(Run-Time Environment Server Component)와 실질적으로 LMS 서버와 SCO간의 통신을 가능케 하는 실행 환경 클라이언트 요소(Run-Time Environment Client Component), 그리고 실행환경 데이터 모델(Run-Time Environment Data Model)로 나누어 설계한다[4].

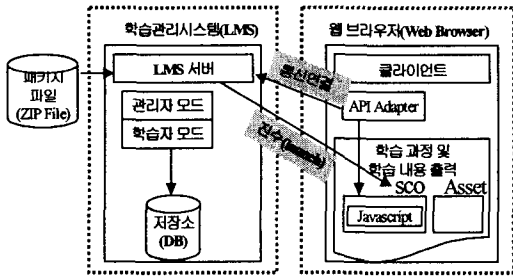


그림 1. SCORM을 이용한 LMS 구성도

### 3.1.1 실행 환경 서버 요소

본 연구에서 실행 환경 서버 요소(Run-Time Environment Server Component)는 실행환경 클라이언트로부터 오는 요청에 대한 반응을 위한 요소이다. 또한 데이터 모델에 대한 신뢰성과 제어성을 보장하고 학습자의 과정 메뉴와 과정별 수업메뉴에 대한 서비스를 제공하고 선택적 수업을 가능하게 하는 일을 담당하게 된다. 서버 요소는 imsmanifest.xml 파일내의 정보를 접근하고 데이터베이스(Database)내에 정보를 저장한다.

또한 PIF(Package Interchange File)로 표현된 콘텐츠 패키지(Contents Package)로부터 ims-manifest.xml 파일을 압축해제하고 접근할 수 있는 zip 유틸리티를 제공한다. 또한 실행환경 서버 요소와 SCO사이에서 통신을 위한 매카니즘을 제공한다[4].

### 3.1.2 실행 환경 클라이언트 요소

본 연구에서 실행환경 클라이언트 측 요소는 자바 애플릿(Java Applet)으로 구현된 실행환경 API 어댑터와 HTML, 자바스크립트로 구현된 사용자 인터페이스로 구성한다. 이 애플릿은 웹 브라우저를 통해 실행환경 메인 페이지(시작페이지)에 사용자가 접근할 때 클라이언트에 다운로드 된다. 그리고 API 어댑터 애플릿은 데이터 모델 요소의 제어성을 위해 실행환경 서버 요소와 통신을 제공한다. SCO는 자바스크립트를 통해 이 API함수를 호출한다. SCO는 실행환경 서버 요소에 대한 세부사항에 관하여 알 필요가 없다.

본 연구에서 클라이언트 요소의 기능을 다음과 같이 정의한다. SCORM으로 표현되어 있는 API 어댑터의 전체 구현물을 포함시킨다. 즉 이는 웹 브라우저를 종료시키기 전까지는 해당하는 API 어댑터가 실행중이라는 말이다[4].

### 3.1.3 실행 환경 데이터 모델

본 연구에서 SCORM 실행환경 데이터 모델은 SCO와 실행 환경간의 통신에 현재 사용되는 유일한 데이터 모델이다. 모든 데이터 요소가 실행 환경에서 실행된다. 데이터 모델에서 데이터 요소를 표현하기 위해서는 서블릿과 API 어댑터를 통해 통신을 하게 된다.

ZIP형태의 패키지 파일인 ADL\_Maritime\_Navigation은 신입 항해사교육용 학습자원이다. 이 자료는 학습자원에대한 메타 정보를 가지고 있는 SCO01.XML 파일과 학습 내용을 출력하기

위한 표현 기법을 정의한 XSD파일, 그리고 각종 학습자원에 대한 총괄 정보를 가지고 있는 IMSMANIFEST.XML, 그리고 최종 자원인 SCO01.HTM, 이미지 자원, 그리고 웹 브라우저에서 실행되는 SCOFuncton, APIWrapper와 같은 Javascript파일로 구성하고 있다. 이들은 최초 관리자에 의해 학습과정 가져오기(import) 작업을 통해 시스템내부에 이미 존재하는 압축해제 기능을 가진 유틸리티를 통해 압축해제 된다. 이후 가장 중요한 적하 목록파일인 IMSMANIFEST.XML 파일을 통해 위에서 언급한 메타 데이터 파일과 출력 폼을 지정한 XSD파일, 그리고 최하위의 SCO, ASSET을 통합 관리하게 된다[4].

3.2 시스템의 구현

본 시스템의 개발 환경은 <표 1>과 같다.

표 1. 구현 시스템의 개발 환경

H / W	OS	Windows 2000 Advanced Server
	Web Server	Apache Tomcat/4.0.1
S / W	JSP Engine	Servlet 2.3 and JSP 1.2 specifications.
	Java Compiler	Java 2 SDK Standard Edition 1.3.1_02
	DBMS	MS Access 2000
	JDBC	JDBC-ODBC
	Tool	Kawa, EditPlus

본 시스템의 전체적인 프로그램 모듈은 (그림 2)와 (그림 3)과 같다.

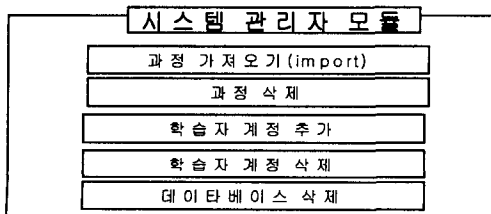


그림 2. 시스템 관리자 프로그램 모듈

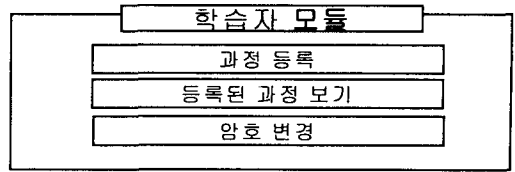


그림 3. 학습자 프로그램 모듈

본 시스템은 LMS에서 관리자 기능과 학습자 기능으로 나누어 구현하고자 한다.

3.2.1 LMS에서 관리자 기능

웹 기반 학습에서는 학습자의 활동뿐만 아니라 교수자의 역할도 중요하게 다루어진다. 특정한 사이트의 기획과 운영 및 관리를 총괄 담당하는 자를 웹마스터(Administrator, Web-Master)라고 한다. 이러한 관리자 역할을 수행하는 것이 교수자의 역할이라고 볼 수 있다. 본 연구에서 교수자의 역할을 수행하기 위한 LMS의 기능적 인터페이스를 살펴보면 다음과 같이 크게 분류할 수 있다. 사용자 선택사항과 관리자 선택사항으로 나눌 수 있다. 관리자 선택사항의 메뉴를 보면 과정 가져오기, 과정 삭제하기, 사용자 계정 추가, 데이터베이스 삭제, 사용자 계정 삭제로 구성되어 있다. (그림 4)에서 보는 바와 같이 ADL의 SCORM Version 1.2의 Contents aggregation Model을 따르는 manifest.xml을 포함한 ZIP형식의 파일을 로컬 시스템(local system)에서 LMS시스템으로

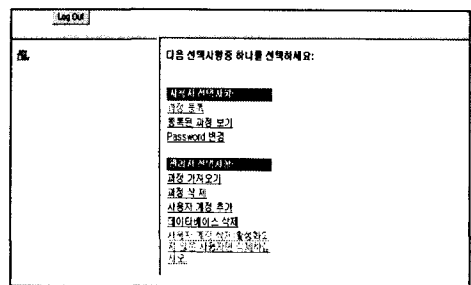


그림 4. 관리자 인터페이스 화면



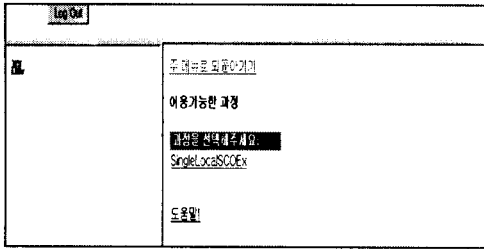


그림 7. 학습자 선택메뉴 중 과정등록

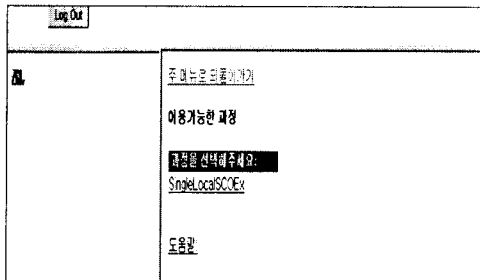


그림 8. 학습자 선택메뉴 중 등록된 과정 보기

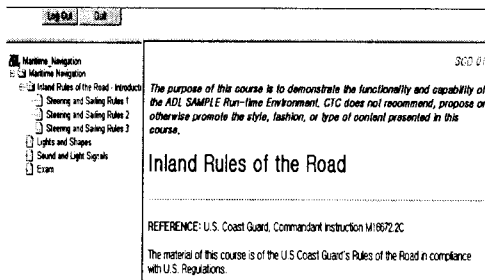


그림 9. 학습자 선택메뉴 중 과정 시작하기

로 데이터베이스내의 ItemInfo라는 테이블은 등록된 학습과정에 관한 정보를 담고 있는 CourseInfo 테이블내의 과정 식별번호(CourseID)를 매개(媒介)로 하여 학습 과정에 관한 세부 항목정보를 저장하게 된다.

물론 이 자료는 과정 가져오기에서 사용한 ZIP 형식의 파일내에 존재하는 imsmanifest.xml(그림 10,11)내의 각 요소에 대한 속성(attribute)값이나 요소(element)값이 데이터베이스내의 ItemInfo라는 테이블에 동시 저장된다.

```
<?xml version="1.0"?>
<manifest identifier="SingleScorableResource_MultipleManifest" version="1.1"
xmlns="http://www.imscp.org/xsd/imscp_rscd/v1/p1p2"
xmlns:adcp="http://www.adnet.org/xsd/adcp_rscd/v1/p2"
xmlns:xi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.imscp.org/xsd/imscp_rscd/v1/p1p2imscp_rscd/v1/p1p2xsd
http://www.imsglobal.org/xsd/imsmd_rscd/v1/p2p1/imsmd_rscd/v1/p2p1xsd
http://www.adnet.org/xsd/adcp_rscd/v1/p2/adcp_rscd/v1/p2xsd">
<metadata>
<organizations default="EBC">
<organization identifier="EBC">
<title>Maritime Navigation Course</title>
<item identifier="E1">
<title>Maritime Navigation</title>
<item identifier="S1" identifierref="sc0_1">
<title>Inland Rules of the Road - Introduction</title>
<item identifier="S2" identifierref="sc0_2">
<title>Steering and Sailing Rules 1</title>
</item>
<item identifier="S3" identifierref="sc0_3">
<title>Steering and Sailing Rules 2</title>
</item>
</item>
</item>
</organization>
</organizations>
</metadata>
<schema>AOL_SCORM/schema>
<schemaVersion>1.2</schemaVersion>
<adcpLocation>Lesson01.xml</adcpLocation>
</metadata>
</organization>
</organizations>
```

그림 10. imsmanifest.xml의 메타데이터 정보

```
<resources>
<!-- ***** -->
** sc01
*****-->
<resource identifier="sc0_1" type="webcontent" adcp:scormtype="sc0" href="sc001.html">
<metadata>
<schema>AOL_SCORM/schema>
<schemaVersion>1.2</schemaVersion>
<adcpLocation>sc001.xml</adcpLocation>
</metadata>
<file href="sc001.html"/>
<dependency identifierref="wrapper" />
<dependency identifierref="functions" />
</resource>
<!-- ***** -->
** sc02
*****-->
<resource identifier="sc0_2" type="webcontent" adcp:scormtype="sc0" href="sc002.html">
<metadata>
<schema>AOL_SCORM/schema>
<schemaVersion>1.2</schemaVersion>
<adcpLocation>sc002.xml</adcpLocation>
</metadata>
<file href="sc002.html"/>
</resource>
```

그림 11. imsmanifest.xml의 메타데이터 정보

학습객체 저장소(Learning Object Repository)는 LMS 시스템에서 학습컨텐츠를 저장하고 관

리하는 중앙 데이터 베이스 역할을 한다. 또한 교수 매체는 Web이나 CD-ROM, 또는 인쇄물을 통해 전달되어왔다. 전달하고자하는 플랫폼에 독립성을 보장하는 콘텐츠의 무결성의 문제는 이들 교수매체의 재사용에 문제를 발생시킨다. 이런 문제를 해결하고자 프로그래밍 로직(logic)과 코드(code)로부터 콘텐츠를 분리하는 기능을 제공하는 XML이 대두되었다. 이런 이유로 SCORM 스펙은 학습객체 저장에 있어 XML을 선택하고 있다. 학습자원에 대한 패키지 파일내에는 imsmanifest.xml이라는 파일이 내포되어 있다.

그 내용을 (그림 10)과 (그림 11)을 통해 살펴보면 <organizations>요소의 identifier 속성 값은 데이터베이스내의 ItemInfo 테이블내의 identifier 열의 값으로 저장되고, <title>요소의 값은 title 열의 값으로 저장되며, <item identifier="S1" identifierref="sco\_1">요소의 경우 identifier 열에는 S1가, Title 열에는 'Inland Rules of the Road - Introduction' 값이 저장되며 identifierref 속성 값이 sco\_1인 것은 이 SCO에 대한 식별자가 참조하는 자원을 가리키는데, (그림 11)에서 <resource identifier="sco\_1" type="webcontent" adlcp:scormtype="sco" href="sco01.html">요소에서 나타난 자원의 identifier가 동일한 sco\_1을 찾아 해당하는 웹 콘텐츠의 type 속성값을 추출하여 type 열에 저장하며, href 속성 값이 가진 값의 경로를 추출하여 launch 열에 저장하게 된다. 기타 저장되는 항목으로는 파라미터(parameter) 문자열 정보를 추출하는 ParameterString 열, 학습관리 시스템으로부터 자료를 추출하는 DataFrom LMS, 선수학습에 관한 정보를 추출하는 Prerequisites, 학습과정에서 완성에 이를 수 있는 점수를 추출하는 MasteryScore, 학습할 수 있는 최대 허용시간을 추출하는 MaxTimeAllowed, 제한 시간을 추출하는 TimeLimitAction, 학습과정의 연

속성을 위해 순서를 매기는 Sequence, 학습 레벨을 나타내는 TheLevel이 있다. 여기서 sequence 값은 0부터 시작하여 동일한 과정에 대해서 순서를 차례대로 부여함으로써 학습자의 학습에 대한 연속성을 보장한다.

(그림 11)에서 보는 바와 같이 <file href="sco01.html">요소에 나타난 href 속성값에는 실질적으로 학습자가 학습해야할 학습 내용이 html 형태로 존재한다. 그리고 <metadata>요소내의 <adlcp:location>에는 sco01.html에 관한 학습 과정에 대한 메타 데이터, 즉 학습 객체가 소유하고 있는 정보를 가지고 있다. sco01.xml에는 SCO에 대한 분류ID, 검색을 위한 keyword, 저작자 및 기판명, 그리고 자료 수명(lifecycle), 언어, 기술적인 면에서 text/html과 같은 형식(format), 실질적인 자원이 위치하고 있는 위치 정보(URI:Uniform Resource Identifier), 사용 가능한 클라이언트 측 도구와 버전(version) 등에 관한 정보를 갖고 있다.

### 3.2.3 학습자원의 LMS로 가져오기

학습자원을 온라인상에서 LMS로 가져오기 하기 위한 기능을 수행하기 위해서 JSP(Java Server Page)를 통해 해당 로컬에 존재하는 zip 형태의 SCORM 스펙을 따르는 학습자원을 LMS에 가져오기함과 동시에 서버의 컴퓨터에 Course Imports라는 폴더가 생성되고 이 폴더내에 Course-1, Course-2, Course-3과 같이 하부 폴더가 생성된다. 또한 Course-1의 하부에는 sco01.xml 과 같은 메타 데이터 파일과 학습 자원, 즉 이미지 파일이나 sco01.html, imsmanifest.xml 파일이 존재하게 된다.

(그림 12)는 SmartUpload 컴포넌트를 이용하여 zip파일을 서버에 업로드함으로써 LMS에 학습자원을 들여오기(import)하는 기능을 가진 코드를 보여주고 있다.



```

// 세션ID, 업로드디렉토리, 사용자 디렉토리, 에러 객체 생성
String sessionId = new String();
String uploadDir = new String();
String userDir = new String();
String error = new String();
LMSManifestHandler myManifestHandler;
LMSPackageHandler myPackageHandler;

try
{
    sessionId = session.getId();
    String theWebPath = getServletContext().getRealPath( "/" );
    myUpload.initialize(pageContext);
    myUpload.upload();
    uploadDir = "WWWSampleRTFFilesWWWTempUploadsWWW" + sessionId;
    java.io.File theRTUploadDir = new java.io.File( uploadDir );

    // 그 과정 디렉토리가 존재하지 않아야 한다.
    if ( !theRTUploadDir.isDirectory() )
    {
        theRTUploadDir.mkdirs();
    }

    // 웹서버의 가상패스내에 파일 저장
    int count2 = myUpload.save( uploadDir, myUpload.SAVE_PHYSICAL );

    String courseTitle = myUpload.getRequest().getParameter( "courseName" );
    String zipFile = myUpload.getRequest().getParameter( "theZipFile" );
    String controlType = myUpload.getRequest().getParameter( "controlType" );

    // 패키지 파일로부터 manifest.xml을 압축해제 하여 업로드 디렉토리에 위치시킨다
    myPackageHandler = new LMSPackageHandler();
    myPackageHandler.extract( zipFile, "imsmanifest.xml", uploadDir );

    String manifestFile = uploadDir + "WWW" + "imsmanifest.xml";
    String newZip = zipFile.substring( zipFile.lastIndexOf( "WWW" ) + 1 );
    zipFile = uploadDir + "WWW" + newZip;

    // manifest handler 인스턴스 생성
    myManifestHandler = new LMSManifestHandler();
    InputSource fileToParse = setUpInputSource( manifestFile );
    myManifestHandler.setCourseName( courseTitle );
    myManifestHandler.setFileToParse( fileToParse );
    myManifestHandler.setControl( controlType );

    // manifest를 파싱하고 객체 구조를 구성한다
    boolean result = myManifestHandler.processManifest();

    // 과정ID값을 가져온다
    String courseID = myManifestHandler.getCourseID();

    ZipFile archive = new ZipFile( zipFile );
}

```

그림 12. LMS로 학습자원 패키지파일 들어오기 (import) JSP code

해당 세션ID와 업로드 디렉토리, 사용자 디렉토리, 에러 객체를 생성후 LMS 서버측에 새로 가져오기 할 학습 자원에 대한 Course 목록이 존재하지 않는다면 새롭게 디렉토리를 생성한 후 SmartUpload 컴포넌트를 이용하여 웹서버의 가상 패스내에 파일을 저장하게 된다. 학습 자원인 패키지 형태의 파일로부터 적하목록 파일인 manifest.xml을 압축해제하여 업로드 디렉토리에 위치시킨다. 그 다음, manifest handler 인스턴스를 생성한 후, 과정명과 가져오기(import) 한 파일을 파싱(parsing)하고 서브디렉토리(sub-directory)내에 객체 구조를 생성한 후 Course id 하부에 압축 해제한 학습자원들을 위치시킴으로써 학습자원이 LMS서버에 가져오기(import) 작

업이 수행되게 된다.

#### 4. 결론 및 향후 연구 과제

본 연구에서는 전 세계적으로 e-Learning분야에서 대두되고 있는 표준화 문제를 해결할 수 있는 SCORM 스펙을 이용한 학습관리시스템(LMS)을 실행환경 서버/클라이언트측 요소와 데이터 모델 측면에서 설계 및 구현하였다.

이러한 SCORM 스펙을 이용한 시스템은 다음과 같은 기대효과를 가져올 수 있다고 본다.

첫째, 어떠한 LMS에서도 공유 가능한 학습자원의 개발이 용이해 진다(Reusability).

둘째, 특정 지역, 특정 플랫폼, 특정 저작도구를 사용하여 개발했다라도 다른 지역에서 다른 저작도구와 플랫폼을 통해서도 사용이 가능한 상호운영성이 보장된다(Interoperability).

셋째, 기반 기술의 변화 시 학습 자원들이 제대로 작동 가능한 영속성을 보장한다(Durability).

넷째, 특정 학습 자료의 온라인 검색이 쉽고, 다양한 저작도구로 개발한 콘텐츠들과 데이터 교환이 가능한 LMS를 통한 접근이 용이하다(Accessibility).

이는 SCORM 스펙의 핵심요소인 실행환경 및 데이터 모델을 중심으로 학습객체(Learning Object) 혹은 SCO와 LMS간의 통신 매카니즘을 중심으로 한 설계 와 구현을 수행하였다. 하지만 이 시스템을 설계시 ADL-SCORM(2001) 스펙을 따라야만 하는 것은 아니다. 결국 SCORM 스펙을 따르는 학습자원을 개발하고, 이 파일을 LMS에서 들여오기 하여 작동하기 위해서는 반드시 ADL에서 제공하는 LMSCMIServlet, LMSPackageHandler, LMSManifestHandler와 같은 Web class 파일을 이용하여야 한다는 점에서 SCORM에 완전 독립적인 LMS를 구현하기가

어렵다는 제한점이 있다고 본다. 또한 복잡한 구조를 가지는 학습자원의 패키지화에 있어서 과연 어느정도 최소화된 Asset을 만들어 사용할 수 있을지에 대한 문제도 발생할 수 있다. 하지만 후자의 경우는 LMS 서버내에 SCO를 재구성할 수 있는 Applet 기술을 이용한 저작도구 어플리케이션을 포함함으로써 해결할 수 있으리라 본다.

향후 연구 과제로는 SCORM 스펙을 따르는 학습 자원의 개발과 현재 설계된 SCORM 스펙을 따르는 학습관리시스템에 몇 개의 SCO를 조합함으로써 새로운 SCO를 재구성할 수 있는 저작도구(Authoring Tool)의 기능이 보완되어야 할 것으로 본다.

## 참 고 문 헌

- [ 1 ] ADL(2002). Sharable Content Object Reference Model version 1.2, Advanced Distributed Learning [http://www.adlnet.org/ADLDOCS/Document/SCORM\\_1.2\\_RunTimeEnv.pdf](http://www.adlnet.org/ADLDOCS/Document/SCORM_1.2_RunTimeEnv.pdf), [http://www.adlnet.org/ADLDOCS/Document/SCORM\\_1.2\\_CAM.pdf](http://www.adlnet.org/ADLDOCS/Document/SCORM_1.2_CAM.pdf).
- [ 2 ] ADL(2002). Sharable Content Object Reference Model version 1.2, Advanced Distributed Learning [http://www.adlnet.org/ADLDOCS/Document/SCORM\\_1.2\\_Overview.pdf](http://www.adlnet.org/ADLDOCS/Document/SCORM_1.2_Overview.pdf).
- [ 3 ] 이준(2001). LCMS(Learning Content Management System) 기반의 e-Learning 개발과 적용. 한국교육학술정보원.
- [ 4 ] 김용만, 김현철(2002). SCORM 스펙을 이용한 학습관리 시스템 설계. 한국컴퓨터교육학회 하계 학술대회, pp.173-178.
- [ 5 ] Michael Brennan, Susan Funke, and Cushing Anderson(2001). The Learning Content Management System. An IDC White Paper.



김 용 만

- 2000년 한국의국어대학교 컴퓨터공학과(공학사)
- 2001년~현재 고려대학교 교육대학원 컴퓨터교육전공 석사과정
- 2001년~현재 우신 정보통신 인력개발원 교사
- 관심분야 : SCORM, LMS, XML EDI
- E-Mail : uaangel@korea.ac.kr



김 현 철

- 1988년 고려대학교 전산학과(이학사)
- 1990년 미조리 주립대학(Rolla) (전산학 석사)
- 1998년 플로리다 대학 (전산정보학 박사)
- 1998년 GTE Data Services, Inc. 시스템 분석가
- 1998년~1999년 미국 GTE Data Services, Inc와 삼성 SDS 근무
- 1999년~현재 고려대학교 컴퓨터교육과 조교수
- 관심분야 : 웹기반교육시스템, 기계학습이론, 데이터마이닝
- E-Mail : hkim@comedu.korea.ac.kr