

Web-based Instruction을 위한 Object-Oriented Lecturelet의 설계

The design of Object-Oriented Lecturelet for Web-based Instruction

서대우, 이재호 (인터넷경영정보과)

Dae-Woo Seo, Jae-Ho Lee (Dept. of Internet and Management Information System)

Key words : Web-based Instruction/Education/Tutoring, Intelligent Tutoring System, Object-Oriented Technology, OOL, Lecturelet, contents reusing, IMS, XML, C#

ABSTRACT : The development of educational contents based on Internet is difficult because of insufficient qualified domain experts and authors and thus needs vast cost and effort. Besides, a content already developed has many problems when reused in similar knowledge domain.

To solve these problems, this paper proposes OOL(Object-Oriented Lecturelet). OOL follows IMS specification, which is the standard specification for Web-based instruction and, encapsulates an instructional content which includes learning materials, problems to diagnose students' knowledge and the function to assess the students' answer to an object. Thus, it allows the extension, reuse, distributed development and execution of contents.

This paper designed and implemented OOL for OOLM(Object-Oriented Lecturelet Model). Results showed that the implemented OOL is appropriately executed on Web environment and therefore proves the possibility of WBI system based on OOL Model.

1. 서론

인터넷이 일반인에게 본격적으로 도입되기 시작한 1990년대 중반부터 인터넷은 폭발적인 양적 성장을 계속해 왔다. 국내 환경에서만 보더라도 1994년 13.8만명에 불과하던 인터넷 인구는 2001년 6월 현재 2,223만명으로 기하급수적인 증가세를 유지하고 있으며 전세계적으로도 4억명 이상이 인터넷을 사용되는 것으로 추정된다[1].

이전의 단독실행형 교육 시스템인 ITS(Intelligent Tutoring System)는 설계에서 개발에 이르기까지 영역지식 전문가 및 저작자의 참여를 필요로 하며 월드와이드웹의 기본 특징인 개방성을 갖지 않음으로써 현재와 같은 다중 개발/사용 환경에는 적합하지 않은 부분이 많다. 또한, ITS시스템들간의 상호 호환성 및 유사 지식영역에서의 재사용에도 문제가 있다[2].

이러한 한계를 극복하고자 WBI(Web-Based Instruction) 시스템은 인터넷을 기반으로 하여 다중 사용자를 지원하도록 설계 및 개발되고 있으며, 학교간 상호 교류에도 이용하고 있다[3]. 웹기반 교육은 학습이 일어나거나 조장되는 유의미한 학습 환경을 조성하기 위하여

웹의 특성과 웹이 제공하는 자료들을 활용하여 전개하는 하이퍼미디어 기반의 교육 프로그램을 말한다[4]. 이것은 웹상의 다양한 자료를 활용하여 교육이 이루어지는 것을 강조하는 것으로서, 웹상의 수많은 도메인 안에는 현재 현실 세계에서 일어나고 있는 생생한 자료들이 무수히 존재하며 이를 이용해야 한다는 것이다. 결국 웹 기반 교육은 웹이 제공하는 풍부한 정보와 통합적 환경을 활용하여 이루어지는 원격교육의 일종이라고 할 수 있다. 이러한 웹의 장점을 충분히 살리고 개발 기간 및 비용을 최소화하기 위해서는 개발된 콘텐츠의 확장 및 재사용, 다양한 서버에 존재하는 콘텐츠의 검색 및 전송과 실행 기능이 필수적이다. 이러한 콘텐츠의 확장 재사용 및 분산 제작·실행을 지원하기 위해서는 콘텐츠 개발을 위한 표준 지침이 필요하며 이 지침을 따라 저작을 하고 재사용까지 가능한 모델이 제시되어야 한다.

이 논문에서는 위와 같은 문제점을 해결하기 위하여 지식 개념망의 노드로 사용될 수 있는 OOL(Object-Oriented Lecturelet)의 개념을 도입 및 설계한다. 실험에서는 설계된 OOL을 구현하고 웹 상에서 실행해 봄으로써 설계 및 제안의 타당성을 입증하며, OOL에 기반한 OOLM(object-Oriented Lecturelet Model) 프레임워크의 가능성을 보여주고자 한다.

이 논문의 2장에서는 현재 인터넷의 표준으로 떠오르고 있는 XML 및 관련 현황, 이를 이용한 WBI 시스템의 국제 표준화 프로젝트인 IMS 프로젝트, 현재 연구 및 서비스 중인 기존 WBI 시스템에 대하여 알아본다. 3장에서는 분산 개발 및 실행을 위한 기본 객체인 OOL을 설계하고 설계된 OOL을 구현하여 웹 상에서 실행해 봄으로써 OOLM의 가능성을 검토해보고 4장에서 결론 및 향후 연구방향을 제시한다.

2. 기존 연구

인터넷 기반의 WBI 시스템을 설계함에 있어서 중요한 이슈는 내용의 표준화, 검색, 데이터베이스화 등을 지원하는 것이다. 이를 위하여 IMS 프로젝트에서는 표현 가능한 정보를 표준 XML DTD 및 XML Schema로 정의하였으며 이를 이용한 WBI 시스템을 제작하도록 권고하고 있다. 이 장에서는 기존의 XML 기반 기술 및 IMS 프로젝트, 그리고 기존의 WBI 시스템에 대하여 알아본다.

2.1 XML 및 관련 기술

인터넷에서 자료 교환을 위한 표준 언어로 사용되어온 HTML(Hyper Text Markup Language)은 문서의 외양을 제어하는 데에는 유용하게 사용될 수 있지만 제한된 태그로만 분류되어 지정되지 않은 태그의 사용이 불가능하다거나 단일 링크형으로서 문서들 사이의 다양한 관계를 지원하지 못하여 검색하기가 쉽지 않다. 또한 구조 기반 설계가 아닌 표현 양식 기반 설계이기 때문에 표현 형식의 변환이 어렵다는 등의 한계를 드러내고 있다.

이러한 문제점들을 해결하기 위해서 CSS(Cascading Style Sheets), HTML DOM(Document Object Model), DHTML(Dynamic HTML) 등 많은 대안들이 등장하였으나 근본적인 문제점을 해결하지 못하고 있다. HTML이 대중적으로 보급되기 이전에, 절차적

markup이 아니라 기술적인 markup을 이용함으로써 하드웨어 및 소프트웨어의 종류에 관계없이 이 기종간의 정보 교환이나 검색, 처리를 지원하는 표준으로 SGML(Standard Generalized Markup Language)이 제시되었다. 그러나, SGML은 지나치게 복잡하여 관련 소프트웨어의 개발이 어렵고 인터넷상에서의 서비스를 위한 목적으로 만들어진 것이 아니므로 웹상에서 일반적인 기능을 수용하지 못하는 등의 이유로 대중화되지 못했었다.

이 같은 한계를 뛰어넘으며 등장한 것이 XML(eXtensible Markup Language)이다. XML은 1998년 2월 W3C 권고에 따르는 XML 1.0 사양이 발표되었다. XML은 HTML과 SGML이 갖는 단점은 보완하고 장점은 결합하고 있다. XML은 SGML의 복잡하고 어려운 과정을 대폭 완화시킨, 간단하고 매우 융통성이 있는 텍스트 형식이다. 또한 구조화된 문서를 정의하고 자유롭게 태그를 정의할 수 있는 등 SGML의 장점은 그대로 취합하고 있다. HTML은 제한된 태그로만 분류되어 지정되지 않은 태그의 사용이 불가능하던 단점을 보완하여 XML 문서를 만드는 개발자가 임의대로 항목을 정할 수 있도록 하였으며, 인터넷 상에서 손쉽게 하이퍼미디어 문서를 제공할 수 있는 HTML의 장점은 그대로 가져가고 있다[5].

Table 1. Comparing HTML / SGML / XML

항 목	HTML	SGML	XML
태그	DTD정의,사용자 정의 불가능, 제한적	사용자 정의 태그 지원 무제한	사용자 정의 태그지원 SGML 보다는 제한적
문서재사용	불가능	가능	가능
응용분야	단순한 구조의 문서 및 너무 길지 않은 문서	방대한 내용과 구조를 요하는 기술적 문서	SGML과 동일 웹상의 교환문서
문서작성	쉽고 간결함논리구조 작성 어려움	복잡하고 어려움	SGML을 단순화시켜 편리하게 작성기능
문서검색	효과적 검색 어려움	자료의 표현 구조와 자료의 내용 분리 정확한 검색 가능	정확한 검색 가능
출력형식언어	CSS	DSSSL	XSL

Table 1에서 보는 바와 같이, XML은 웹의 새로운 표준 형식으로 자리잡고 있으며 그 자체의 장점으로 인하여 B2B를 포함한 전자상거래, 온라인 banking, 푸시 기술, 검색엔진, 웹기반 제어 시스템, 에이전트, EDI, 전자도서관 등 그 응용 분야가 계속 확장되고 있다. 웹 기반 교육 분야에서의 활용 예로서 IMS Global Learning 컨소시엄에서 진행 중인 IMS 프로젝트가 있다[16].

XML이 웹의 많은 응용 영역에서 각광을 받고 있으나 XML 문서는 문서를 구조화할 수 있는 기능만을 가지고 있다. 이러한 문서들은 DOM(Document Object Model)을 통하여 XML 문서들을 마치 프로그램 객체처럼, 자신들의 콘텐츠(contents)나 객체 내에 감추어진 데이터를 가질 수 있도록 작성될 수 있게 한다. DOM을 이용함으로써 문서를 객체들과 마찬가지로 메서드(method)라고 불리는 객체지향 프로시저들을 함께 가지고 갈 수 있게 된다. DOM은 두 가지 계층의 인터페이스 구현을 제공하는데, 그 하나는 XML을 지원하는 DOM Core로서 그 다음 계층의 기반이 되며, 다른 하나는 DOM HTML로서 그 모델을 HTML

문서로 확장한 것이다. 즉, 어떠한 HTML이나 XML 요소에 대해서도 프로그래밍을 통한 개별적인 주소지정이 가능하며, 또한 이 규격은 IDL을 사용하여 설명하게 될 것이므로, 언어에 독립적이다[6].

XML Schema는 최근에 W3C로부터 인정받은 새로운 문서 구조 기술 언어로서, 현재 사용되고 있는 문서 구조 기술 명세인 XML DTD(Document Type Definition)의 단점을 보완하여 다양한 자료형을 선언 및 정의할 수 있는 메커니즘을 제공한다[5]. 이러한 장점으로 인하여 XML DTD를 점차 대체해 나갈 것으로 기대된다.

XML Query는 웹 상의 문서로부터 데이터를 추출하는 유연한 질의어로서, 웹과 데이터베이스를 상호작용시킬 수 있는 기능을 제공한다[5]. 따라서 XML 파일의 집합을 데이터베이스처럼 다룰 수 있도록 만든다. 이것은 OOLM 프레임워크의 구성 요소 중 OOL 검색 모듈에 이용될 수 있다.

XSL은 XML의 스타일시트를 표현하는 언어로서, 크게 XML 문서의 변환을 위한 언어인 XSLT(XSL Transformations)와 XSLT에 의해 XML 문서의 다른 부분을 참조하는데 사용되는 XPath(XML Path Language)로 구성된다. XSLT는 XML로 작성된 문서를 HTML이나 다른 언어로 변환하는 기능을 제공하므로 XML을 지원하지 않는 HTML 브라우저나 기존 문서와의 호환을 위해 사용된다[5]. OOL에서는 OOL에 의해 XML로 생성되는 콘텐츠 및 문서를 필요에 따라 HTML로 변환하거나 XML 문서를 다양한 형태로 표현할 때 사용한다.

2.2 IMS 프로젝트

개방된 환경인 웹 기반의 교육 시스템 개발에 있어 표준화는 매우 중요한 이슈이다. IEEE의 LTSC[7], IMS 프로젝트[8], Aviation Industry's AICC[9], DoD's ADL initiative[10], ARIADNE[11], GESTALT 프로젝트[12], PROMETEUS[13], CEN/ISSS/LT[14], GEM 프로젝트[15] 및 많은 표준화 작업이 진행 중이다.

IMS Global Learning 컨소시엄은 교육 콘텐츠와 관리 시스템 사이에서 학습자 진도, 학습자 효율성 및 학습자 기록의 교환 등, 온라인 분산 학습 활동을 위한 공개 명세를 개발 중이다[16]. IMS의 두 가지 목적은 첫째, 분산 학습 서비스와 어플리케이션 사이의 상호운용을 위한 기술적 명세서 정의이며 둘째, IMS 명세를 전세계에서 사용될 수 있도록 지원하는 것이다. IMS는 분산 학습 환경과 다중 저작자에 의해 저작 가능한 콘텐츠를 위한 명세서로 채택될 수 있도록 노력하고 있다.

IMS의 회원 기관은 캘리포니아 대학 등의 교육기관, IBM을 포함한 상용기관, 미국방성 등을 포함하는 정부기관 등을 망라하고 있으며 W3C를 비롯한 협력기관은 세계적인 ICT와 교육관련 표준화 기관들을 포함하고 있다. 따라서 IMS에서 발표하는 명세서는 교육 명세서의 세계적인 표준안이라고 할 수 있다. IMS 명세서는 현재까지 다음과 같은 5종의 명세를 발표하였고, 지속적인 발전이 진행 중이다[16].

1. IMS 학습자원 메타데이터 명세(Learning Resource Meta-data Specification)는 학습 콘텐츠를 기술하는 단일한 방법을 제공하여 학습 콘텐츠를

보다 쉽게 검색할 수 있게 한다. 메타데이터를 이용하여 검색함으로써 학습상황에서 발생하는 사용자의 고유한 요구에 부응하는 콘텐츠를 찾아낼 수 있다.

2. IMS 엔터프라이즈 명세(Enterprise Specification)는 코스, 성능, 학습자 등에 대한 정보를 플랫폼, 운영체제, 사용자 인터페이스 등에 걸쳐서 공유하기 위해 필요한 어플리케이션과 서비스를 목적으로 한다.
3. IMS 콘텐츠 패키징 명세(Content Packaging Specification)는 다양한 학습 시스템에서 사용될 수 있는 재사용 가능한 콘텐츠 객체를 생성하는 것을 쉽게 해준다.
4. IMS 문제와 시험 명세(Question & Test Interoperability Specification)는 시험 아이템들과 다른 평가 도구들을 다른 시스템들에 걸쳐서 공유할 수 있도록 해준다.
5. IMS 학습자 프로파일 명세(Learner Information Package Specification)는 학습 시스템이 각 학습자의 요구에 보다 지능적으로 반응하기 위해 학습자 정보를 조직하는 방안을 제공한다.

특히 콘텐츠에 관련된 표준화 작업인 IMS의 CPS(Content Packaging Specification)은 콘텐츠 생성 도구에 의해 생성되는 인터넷 기반의 콘텐츠와 학습관리 시스템 및 실행 환경에서 상호운용성을 제공하기 위한 자료 구조를 기술한다. CPS의 목적은 콘텐츠의 교환에 사용할 수 있는 구조의 표준 집합을 정의하는 것이다. 이 구조는 서로 다른 소프트웨어 개발자가 저작도구, LMS(Learning Management System), 실행 환경에 관계없이 상호운용이 가능한 교육 자료를 생성할 수 있는 표준 자료 구조를 제공한다.

IMS의 명세서는 자료 구조를 XML DTD 및 XML Schema로 표현하고 있다. 이 논문에서 제안하는 OOL의 출력은 콘텐츠의 표현에 관련된 IMS CPS를 따름으로써 호환 및 상호운용성을 극대화시키도록 한다.

2.3 기존의 Web-based Instruction에 관한 연구

KIE(Knowledge Integration Environment) 프로젝트는 캘리포니아 버클리대학교에서 미국 과학재단의 지원을 받아 1994년부터 추진해오고 있는 것으로, 중학생과 고등학생들이 인터넷에서 과학적 증거를 탐색하여 복잡한 과학 개념과 과정을 통합적으로 이해할 수 있게 하는 데 목적이 있다. 이 프로젝트는 인터넷에서 운영될 수 있는 과학교육용 소프트웨어를 개발하고 이를 미국 전역에 보급하는 형태로 진행되고 있다. KIE는 KIE에서 작업을 할 수 있는 컨트롤 패널인 KIE 툴 팔레트, 학생집단이 자신들의 프로젝트와 관련된 문서를 관리할 수 있는 NetBook, 수집된 과학적 증거 자료들로 구성된 증거 데이터베이스, 학습에 도움을 제공하는 온라인 안내, 논증을 구성하기 위한 도구인 SenseMaker, 토론 및 실시간 의사소통을 위한 SpeakEasy 등 6 가지 요소로 구성되어 있다[17].

SimulNet[18]는 웹 기반의 상호 작용 및 공동 작업이 가능한 교육 응용 분야를 위한 시스템으로서 콘텐츠의 개발보다는 시스템의 개발에 초점을 맞추고 있다. SimulNet은 소프트웨어의 재사용에 초점을 맞추고 있으며, 현재 컴포넌트 기반 방법론[19], COM(Component-Object Model)[20], 자바 빈즈(Java Beans)[21] 및

CORBA(Common Object Request Broker Architecture)[22] 등이 소프트웨어 재사용을 위한 기술로서 사용된다. 그러나 이 시스템은 콘텐츠 자체의 재사용은 지원하지 않는다.

IDEAL은 분산 학습 환경을 제공하는 시스템으로서 지능형 에이전트를 도입하여 학습자 중심의, 고도의 상호 작용과 학습자 관리를 제공하는 WBI이다[23]. IDEAL은 학습편(lecturelet)을 기반으로 하여 학습 콘텐츠를 저작한다. 학습편은 캡슐화, 자동화된 콘텐츠의 조각이다. 이러한 학습편은 Lecturelet 서버 내의 디지털 라이브러리에 저장되어 학습자의 요청에 의해 제공되어 진다. IDEAL에서 사용되는 학습편은 내부적으로 XML로 표현이 된다. Fig. 1은 IDEAL에서 사용되는 학습편의 내부 구성요소와 전달을 위한 프레임워크를 보여주고 있다.

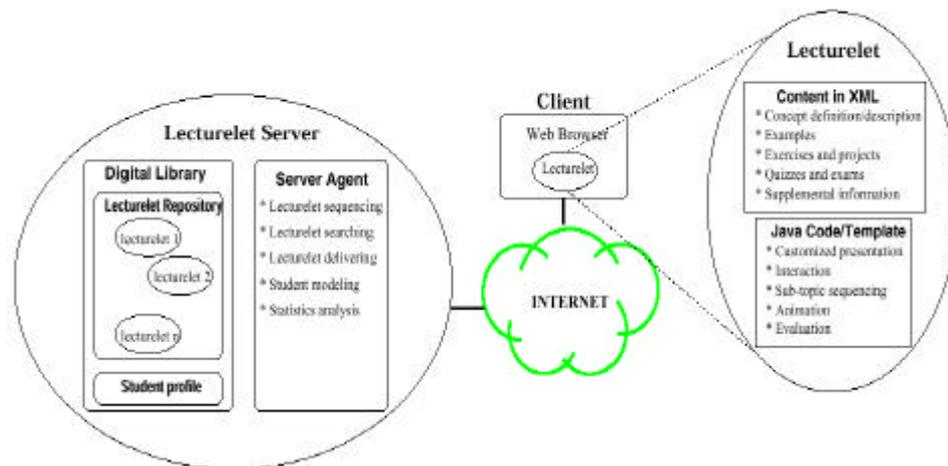


Fig. 1. Framework for lecturelet of IDEAL

그러나 IDEAL은 학습편의 분산 개발 및 재사용보다는 학습자 지원을 위한 지능형 에이전트에 초점을 맞추므로서 학습편의 효과적인 생성 및 관리와 다중 저작자 및 저작 서버를 위한 개념을 제공하지 않는다.

국내의 경우 대표적인 교육용 웹 사이트로 에듀넷을 들 수 있다. 에듀넷은 교육 정보의 효과적인 전달 체제를 구축함과 동시에 분산되어 있는 교육 관련 정보를 상호 연계함으로써 학생, 교사, 교육행정가, 학부모 등 모든 국민이 원하는 양질의 교육 정보를 컴퓨터 통신을 통하여 정보를 이용할 수 있도록 하기 위한 목적으로 한국교육학술정보원(KERIS)에서 운영하는 교육 정보 종합 서비스 시스템이다[24].

그러나 에듀넷은 교육 관련 정보들이 에듀넷 서버에 모여 있는 중앙 집중형 구조이다. 이와 같은 중앙 집중형 구조는 일원화된 자료 관리가 가능하므로 자료의 일관성을 유지시키기가 쉽다는 장점이 있지만 자료의 수와 사용자의 수가 증가하는 경우 서버의 과부하가 커진다. 또한 에듀넷의 콘텐츠들은 교과 과목별로 별개의 콘텐츠를 개발함으로써 유사 개념의 콘텐츠에 대해서는 재활용을 실시하지 않아 개발에 많은 비용이 소비되었다.

위에서 살펴본 시스템들은 IMS의 표준 명세를 따르지 않음으로서 분산 저작 및 실행에 필요한 상호운용성이 낮다. 이 논문에서는 기존의 WBI 시스템에서는 사용되지 않는 객체지향 학습편인 OOL(Object-Oriented Lecturelet)의 개념을 제안하며 IMS 명세서에 준하여

설계한다. 또한 설계된 OOL을 근거로 하여 구현 및 웹 상에서 실행해 봄으로써 설계의 타당성을 입증하고 OOLM(OOL Model) 기반 WBI 시스템의 가능성을 보여준다.

3. OOL의 설계

이 장에서는 OOLM의 기반이 되는 OOL을 설계한다. OOL은 개방형 분산 교육 시스템인 OOLM 프레임워크에서 핵심이 되는 개념이다. OOL을 이용하면 학습자가 학습하기 위한 교육용 콘텐츠를 개념 단위로 분할하여 동시에 콘텐츠 및 학습자를 평가하기 위한 문제를 개발할 수 있으며, 유사한 지식 영역에서의 재사용시 기존의 OOL을 검색하고 이를 확장/상속받음으로서 다른 지식 영역의 WBI 시스템 개발시 개발 기간 및 비용을 줄일 수 있다.

3.1 OOL의 구성

OOL은 크게 메타데이터, 콘텐츠, 문제 및 OOL 객체 자신을 표현하기 위한 XML 생성기로 구성된다. OOL의 구조는 Fig. 2와 같다.

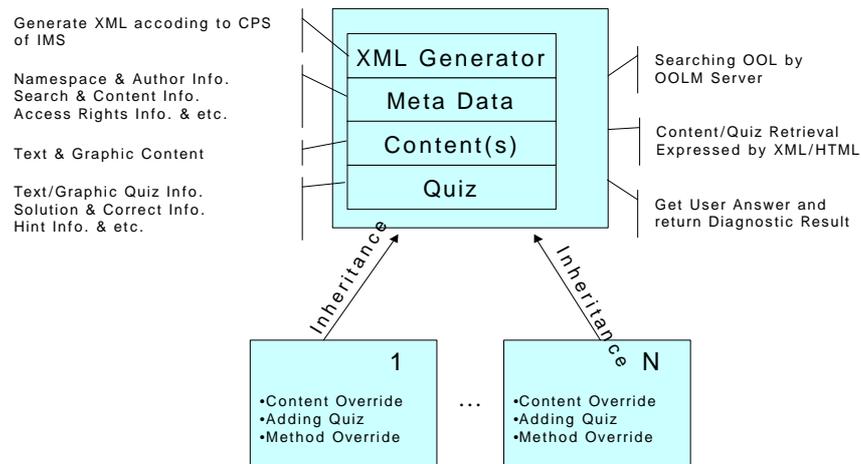


Fig. 2. Structure diagram of OOL(Object-Oriented Lecturelet)

OOL 내부의 구성 요소는 크게 OOL 자신을 IMS의 CPS에 따라 외부에 설명하고 검색을 위한 XML 생성기, 객체를 설명하는 메타데이터, 학습 개념을 설명하기 위한 콘텐츠, 학습자의 성취도를 평가하기 위한 문제 및 이에 대한 학습자의 응답을 진단하기 위한 진단 메서드 등으로 구성된다. OOL의 내부 구성 요소는 저작의 편의 및 표준화를 위해 기반 클래스로부터 상속받는다. Fig. 3은 Rumbaugh의 OMT에 따른 기반 클래스들과 OOL 클래스 사이의 관

계를 나타낸다.

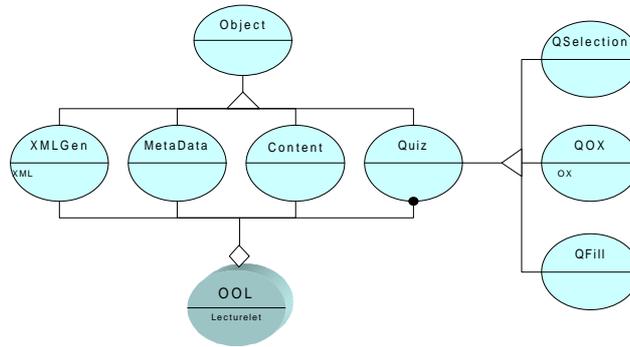


Fig. 3. Relationship diagram for OOL and base classes

1) XML 생성기

객체의 내부에는 자신을 설명하기 위한 메타데이터를 포함하고 있다. 외부에서 메타데이터에 접근하면 이 메타데이터를 XML 문서로 생성해 낸다. 생성된 XML 문서는 IMS CPS를 따르도록 함으로써 상호운용성을 높이고, 검색 엔진이나 원래의 저작자 외의 다른 저작자나 학습자를 위한 기초 자료로서 사용된다. IMS CPS는 Fig. 4에서 보여주는 것과 같은 요소(element) 구조를 갖는다.

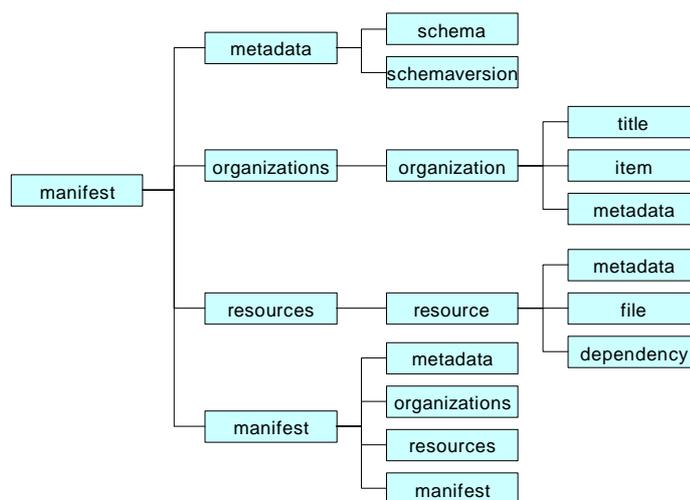


Fig. 4. IMS Content Packaging Specification elements structure

OOL은 IMS CPS의 XML DTD를 따르기 위해 table 2와 같은 XML DTD를 준수하여 XML 문서를 생성한다.

Table 2. XML DTD for Content Packaging Specification

```

<!ELEMENT schema (#PCDATA )>
<!ATTLIST schema e-dtype NMTOKEN #FIXED 'string' >
<!ELEMENT schemaversion (#PCDATA )>
<!ATTLIST schemaversion e-dtype NMTOKEN #FIXED 'string' >
<!ELEMENT title (#PCDATA )>
<!ATTLIST title e-dtype NMTOKEN #FIXED 'string' >
<!ELEMENT file (metadata? )>
<!ATTLIST file href CDATA #REQUIRED >
<!ELEMENT dependency EMPTY>
<!ATTLIST dependency identifierref CDATA #IMPLIED >
<!ELEMENT resource (metadata? , file+ , dependency* )>
<!ATTLIST resource identifier ID #REQUIRED type CDATA #REQUIRED href CDATA #IMPLIED >
<!ELEMENT resources (resource* )*>
<!ELEMENT item (title? , item* , metadata? )*>
<!ATTLIST item identifier ID #REQUIRED isvisible CDATA #IMPLIED parameters CDATA #IMPLIED
  identifierref CDATA #IMPLIED a-dtype NMTOKENS 'isvisible boolean' >
<!ELEMENT organization (title? , item* , metadata? )>
<!ATTLIST organization identifier ID #REQUIRED >
<!ELEMENT organizations (organization* )>
<!ATTLIST organizations default IDREF #IMPLIED >
<!ELEMENT metadata (schema? , schemaversion? )>
<!ELEMENT manifest (metadata? , organizations , resources , manifest* )>
<!ATTLIST manifest identifier ID #REQUIRED version CDATA #IMPLIED >

```

2) 메타 데이터 클래스

메타데이터 클래스는 강좌명, 개념명 등 개념망에서 OOL의 위치를 표현하고 자신을 설명하기 위한 데이터, 저작자와 관련된 데이터 및 Content 보호를 위한 signature, 콘텐츠 구분을 위한 namespace, 소유자 정보 및 사용자 제한 정보 등 IMS CPS에서 제안된 메타데이터를 포함한다. 이 정보는 객체의 소스를 보호하기 위하여 객체 내부에 hard-coded되어 외부에서 검색을 하려면 XML 생성기 및 관련 메서드로만 검색이 가능하다. 메타데이터에 포함되는 구체적인 정보는 table 3과 같다.

Table 3. Data types of meta-data

구분	포함 데이터
각종 코드	학문분야, 세부학문, 과목, 개념에 대한 코드
컨텐츠 관련정보	제목, 요약, 설명, 컨텐츠 키워드, 난이도, 중요도, 최소접수
관련 개념 코드	선수 개념 및 관련 개념에 대한 코드
저작자 신상정보	이름, 암호, 메일 주소 등의 저작자 신상에 관한 정보
접근권한 정보	호스트 IP, 접근 가능/불가능 IP
날짜	생성일, 최종 수정일, 소멸일 등 컨텐츠의 날짜에 관한 정보
상속 정보	부모 OOL(상속받은 경우)의 코드, 상속 가능 여부
기타 정보	컨텐츠 언어, 컨텐츠 구분(텍스트/멀티미디어) 정보, 문제 수 등

3) Content 클래스

컨텐츠 클래스는 하나의 학습 주제를 설명하기 위한 컨텐츠를 포함하는 클래스이며, 이는 텍스트 및 이미지/오디오/애니메이션/비디오 등 각종 멀티미디어 요소를 사용하여 저작된다. 컨텐츠는 학습자의 수준에 따라 그 표현을 달리하여 학습자에게 제시할 수 있는 메서드를 내재한다. 컨텐츠를 클라이언트 브라우저에 전송하기 위하여 XML/XSL 또는 HTML 문서를 생성한다.

Table 4. Information and methods included in Quiz class

구분	포함 데이터
문제 형태	사지선다, OX, 단답형
문제	문제 본문, 보기, 정답 (텍스트/멀티미디어)
평가 정보	학습자 평가를 위한 난이도, 중요도, 문제 점수
기타	필수 여부
상속 정보	오버라이드(override) 가능 여부
메서드	속성 설정(문제 표현 방식) 및 획득 내부표현을 XML/HTML로 생성, 사용자 응답 평가

4) Quiz 클래스

Quiz 클래스는 OOL에 포함되는 문제들의 집합이며, 이 문제는 자신을 표현하는 정보 및 정답에 대한 정보, 학습자에게 제시되는 문제의 표현 형태를 포함하고 있다. 문제도 컨텐츠 클래스와 같이 텍스트나 멀티미디어 요소를 사용하여 저작되며 사용자의 수준에 따라 같은 문제를 표현하는 방법이 달라지게 된다. 또한 제시된 문제를 진단하기 위한 응답 진단 메서드를 포함한다. 이 논문에서는 문제 풀이 결과의 용이한 평가를 위해 문제형을 사지선다형, 진위형, 단답형의 세 가지 문제형으로 제한한다. Quiz에 포함되는 정보는 table 4와 같다.

3.2 OOL의 저작 및 생성

OOL을 제작하기 위해서는 OOLM 프레임워크의 OOL 생성 모듈에 포함된 템플릿

(template)을 이용하여 저작의 전문가가 아니더라도 쉽게 제작할 수 있는 저작 방법을 제공해야 한다. OOL의 생성 과정은 Fig. 5와 같다.

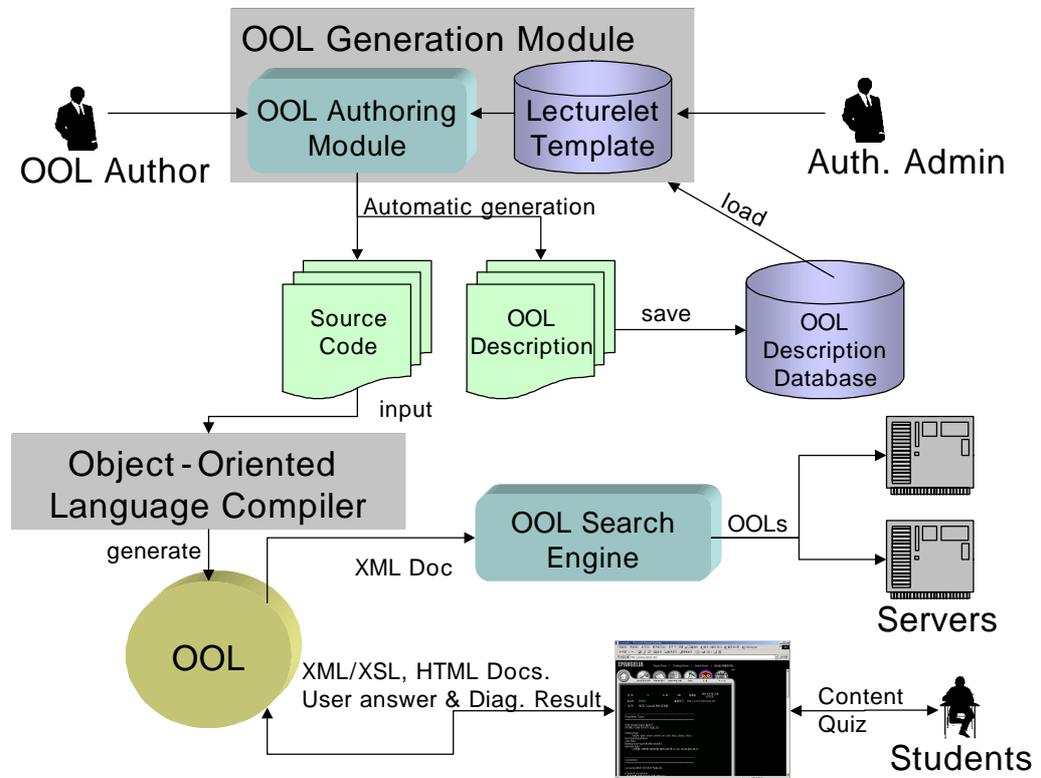


Fig. 5. Usage and generation process of OOL

템플릿은 저작 관리자가 미리 제작을 해서 입력해 두며, 저작자는 이를 이용하여 OOL을 쉽게 제작하게 된다. 특수한 인터페이스를 가지는 OOL을 생성하기 위해서는 새로운 템플릿을 미리 제작한 후 생성한다. OOL 생성 모듈에 포함된 OOL 저작 모듈은 이 템플릿을 이용하여 저작자에게 필요한 데이터를 입력받고 OOL이 제공하는 기능들을 선택하며, 최종적으로 Java/C++/C# 과 같은 객체지향 프로그래밍 언어의 소스 형태로 출력하게 된다. 이 소스 파일은 컴파일러를 통하여 이진 파일로 변환한다. 독립된 이진 파일 형태로 생성된 OOL은 내부의 정보를 숨길 수 있으며 네트워크간 공유시에도 쉽게 전달될 수 있도록 한다.

OOL 생성 모듈의 설계 및 구현은 OOLM 프레임워크에서 이루어 지므로 이 논문에서는 텍스트 에디터를 이용하여 OOL 소스를 편집하였다.

3.3 OOL의 구현

현재 객체지향 프로그래밍 언어로서 널리 사용되는 언어로 C++, Java, C# 등이 있다. C++은 C 언어로부터 파생된 하이브리드 언어로서 전통적인 C 언어를 지원하기 위해 완전한 객체지향의 특성을 갖지 못한다. Java는 전 세계의 많은 개발자가 이 언어를 사용하고 있

으나, 기본 데이터형(primitive types)와 객체 데이터형(혹은 사용자 정의형, object types)들이 매우 다른 방식으로 취급되고 처리된다. C# 언어는 마이크로소프트에서 개발된 .NET 프레임워크를 위한 새로운 언어로서 C++과 Java의 장점을 결합한 객체지향을 제대로 표현하는 프로그래밍 언어이다[25].

C#은 C/C++ 계열 언어중 최초의 컴포넌트 기반 언어로서 간단하고 객체지향적이며 형 안정성을 가지는 언어이다. C#은 CLR(Common Language Runtime) 및 CLS(Common Language Specification)을 기반으로 설계되었으므로 비주얼 베이직, C++, Jscript 등 다른 언어에서도 C# 언어로부터 상속받을 수 있는 장점이 있다. 이것은 개발 환경의 변화 등 상황에 따라 다양한 언어로 OOL을 생성할 수 있다는 것을 의미하는 것이다. 따라서 이 논문에서는 C# 언어를 사용하여 OOL을 구현한다. 이 절에서는 Fig. 3에서 설계한 OOL을 구성하는 기반 클래스와 OOL 클래스의 뼈대가 되는 코드를 소개하며, 실제 응용을 위하여 개발된 OOL의 실행 화면을 제시한다. OOL의 구현 및 실행 환경은 table 5 와 같다.

Table 5. The environment of OOL implementation and execution

OS	Windows 2000 Server
Web Server	IIS(Internet Information Server) 5.0
Languages	PHP, C#, HTML/XML/CSS/XSL, Javascript
Browser	Internet Explorer 5.5
Database	N/A (파일 시스템을 이용)

3.3.1 Skeleton code for classes

Fig 3에서 설계한 OOL을 위한 클래스들의 skeleton 코드는 아래와 같다. 2라인에서 보는 바와 같이 OOLs라는 이름공간(namespace)를 가진다. OOL 관련 클래스는 Quiz 클래스로부터 상속받은 여러 개의 QSelection, QOX, QFill 객체를 가지므로 36 라인에서와 같이 C#에서 제공하는 인덱서(Indexer) 기능을 이용하여 구현해야 한다. 부모 클래스는 자식 클래스로부터 객체화된 객체를 참조할 수 있으므로 OOL 클래스는 QSelection, QOX, QFill 클래스의 부모 객체인 Quiz 클래스 객체만을 멤버로서 포함하면 된다. Table 6은 skeleton 소스 코드이다.

Table 6. Skeleton code for base classes

```

01: using System.Collections;
02: namespace OOLs
03: {
04:     public class XMLGen
05:     {
06:         // properties and methods are go here.
07:         public override string ToString()
08:         {
09:             return "XMLGen class";
10:         }
11:     }
12:     public class Metadata
13:     {
14:         // properties and methods are go here.
15:         public override string ToString()
16:         {
17:             return "Metadata class";
18:         }
19:     }
20:     public class Content
21:     {
22:         // properties and methods are go here.
23:         public override string ToString()
24:         {
25:             return "Content class";
26:         }
27:     }
28:     public class Quiz
29:     {
30:         // properties and methods are go here.
31:         public override string ToString()
32:         {
33:             return "Quiz class";
34:         }
35:         protected ArrayList quiz = new ArrayList();
36:         public object this[int idx]
37:         {
38:             get
39:             {
40:                 if (idx > -1 && idx < quiz.Count)
41:                     return (quiz[idx]);
42:                 else
43:                     return null;
44:             }
45:             set
46:             {
47:                 if (idx > -1 && idx < quiz.Count)
48:                     quiz[idx] = value;
49:                 else if (idx == quiz.Count)
50:                     quiz.Add(value);
51:                 else
52:                     ; // throw exception
53:             }
54:         }
55:     }
56:     public class QSelection : Quiz
57:     {
58:         // properties and methods are go here.
59:         public override string ToString()
60:         {
61:             return "QSelection class";
62:         }
63:     }
64:     public class QOX : Quiz
65:     {
66:         // properties and methods are go here.
67:         public override string ToString()
68:         {
69:             return "QOX class";
70:         }
71:     }
72:     public class QFill : Quiz
73:     {
74:         // properties and methods are go here.
75:         public override string ToString()
76:         {
77:             return "QFill class";
78:         }
79:     }
80: }

```

3.3.2 PHP Wrapper

PHP Wrapper는 웹 서버 내에 산재한 OOL 클래스 인스턴스를 실행시켜주기 위한 PHP 코드이다. 학습자가 요구하는 OOL 인스턴스의 이름을 입력하면 PHP Wrapper가 이를 실행

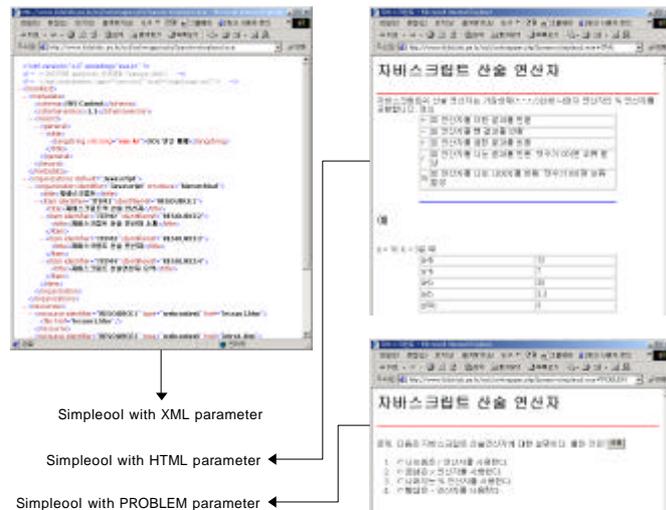


Fig. 7. Real execution example of OOL object

4. 결론 및 향후 연구 방향

이 논문에서는 개방형 분산 WBI를 위한 모델인 OOLM을 위한 OOL을 제안하고 설계 및 구현하였다. 설계한 OOL은 하나의 개념을 설명하고 학습자를 평가하기 위한 콘텐츠 및 문제를 포함하며, 검색 및 재사용을 위한 각종 메타데이터와 XML 생성기를 포함한다. 메타데이터는 XML 생성기에 의해 IMS의 표준 명세를 따르는 XML로 변환되며 이 XML 문서는 OOLM에서의 검색, 재사용 및 상속을 위한 자료로 사용된다. 설계된 OOL을 객체지향을 가장 잘 표현하는 C# 언어로 구현하였다. C# 언어로 컴파일된 OOL은 실행 파일만으로도 상속이 가능하므로 OOL의 상속 및 재사용에 자연스러운 해법이 된다. 이렇게 구현된 OOL 객체는 웹 서버에서 운영이 가능하며 설계대로 동작하는 것을 보여주었다.

OOL은 객체를 호출하는 매개변수에 따라 객체 설명 자료(XML 문서), 학습자료(XML/XSL, HTML 문서), 문제(HTML 문서) 등 서로 다른 문서를 출력할 수 있음을 보였다. 이것은 OOL이 OOL 검색엔진, 학습자료의 요청 등 상황에 따라 다르게 대응할 수 있다는 것을 보이는 것이다. 따라서 이 OOL 객체를 개방형 웹 기반 학습 시스템 모델인 OOLM에 사용할 수 있음을 의미한다.

향후 연구로는 이렇게 설계 및 개발된 OOL을 이용한 모델을 제시하고 이 모델을 이용하여 OOL을 개발, 검색, 전송, 실행할 수 있는 프레임워크를 설계 및 구현하는 것이다. OOLM 프레임워크는 재사용 및 검색을 위해 OOL을 하나의 노드로 사용하는 영역지식 개념망(domain knowledge conceptual network)를 저작하는 기능이 필수적으로 요구되며, OOL을 위한 템플릿을 이용한 저작모듈 및 저장 방법, XML Query를 이용하는 OOL 검색 모듈, 검색된 OOL을 전송 및 실행하기 위한 모듈, 학습자를 관리하기 위한 모듈 등을 제시해야 한다.

참고문헌

- [1] 한국인터넷정보센터 WWW site. <http://stat.nic.or.kr/iuser.html>
- [2] 황희정, 백영태, 강운구, 이세훈, 윤경섭, 왕창중, *EDUCUSE의 IMS 명세서에 기반을 둔 개방형 교육 시스템*, 춘계학술논문발표집, 한국정보처리학회, 1999. 4.
- [3] W.J.Trehwella, P.M.C. Davies, Benny K.K. Chan, Gray A. Williams, *Inter-institutional Collaborative Teaching In The Virtual School of Biodiversity*
- [4] Khan, B.(Ed.), *Web-based Instruction*, Englewood Cliffs, NJ: ET Publications.
- [5] XML W3C WWW site. <http://www.w3.org/>
- [6] Document Object Model W3C WWW site. <http://www.w3.org/DOM/>
- [7] IEEE. Learning Technologies Standardization Committee. WWW site. <http://ltsc.ieee.org>.
- [8] IMS. Instructional Management Systems Global Learning. WWW site. <http://www.imsproject.org>
- [9] AICC. Aviation Industrie CBT Committee. WWW site. <http://www.aicc.org>
- [10] U. D. of Defense. DoD Advanced Distributed Learning. WWW site. <http://www.adlnet.org>
- [11] ARIADNE. The Alliance of Remote Instructional Authoring and Distribution Networks for Europe. WWW site. <http://ariadne.unil.ch>.
- [12] GESTALT. Getting Educational System Talk Across Leading edge Technologies project. WWW site. <http://www.fdggroup.co.uk/gestalt>.
- [13] PROMETEUS. PROMoting Multimedia access to Education and Training in EUropean Society. WWW site. <http://prometeus.org>.
- [14] CEN. European Committee for Standardization (CEN), Information Society Standardization Systems (ISSS), Learning Technologies Workshop (LT). WWW site. <http://www.cenorm.be/iss/Workshop/lt/>.
- [15] GEM. Gateway to Educational Materials. WWW site. <http://www.geminfo.org>.
- [16] IMS Global Consortium Inc. WWW site. <http://www.imsproject.org>
- [17] 나일주, *웹기반교육*, 1999. 5. 교육과학사, pp208~pp212
- [18] L. Anodo-Rifn, M. Llamas-Nistal, M.J. Fernndez-Iglesias, *A Component Model for Standardized Web-based Education*, 2001. Proceedings of the 10th International Conference, pp86~pp95
- [19] W. Lim. *Managing Software Reuse: A Comprehensive Guide to Strategically Reengineering the Organization for Reusable Components*. Prentice-Hall, July 1998.
- [20] Microsoft. Components Object Model. WWW site. <http://www.microsoft.com/com/>.
- [21] S. Microsystems. Java Beans. WWW site. <http://www.sun.com/beans/spec.html>.
- [22] Object Management Group. CORBA 3.0. WWW site. <http://www.omg.org/news/pr98/compnent.html>.
- [23] Yi Shang, Hongchi Shi, Su-Shing Chen, *An Intelligent Distributed Environment for Active Learning*, 2001. Proceedings of the 10th International Conference, pp308~pp315

- [24] 김영애, 박근수, *에듀넷 활용성 제고방안 연구*, 1999. 12. 한국교육학술정보원(KERIS)
- [25] Tom Archer, *Inside C#*, 2001. 8. Microsoft Press