분산환경을 위한 교수법적 설계의 재사용 단위를 객체화한 강의 컨텐츠 시스템

신 행 자† · 박 경 환††

요 약

본 논문에서는 웹기반 강의 컨텐츠의 문제점을 알아보고 그 문제점을 해결할 수 있는 방법을 제안한다. 다시 말해서, 기존의 웹기반 강의 컨텐츠는 HTML 포맷 기반 코스웨어의 고정되고 획일적인 하나의 큰 파일이거나 미디어 제공 벤더에 종속된 저작도구로 작성된 파일이다. 이러한 강의 컨텐츠는 서로 다른 가상 교육 시스템에서 공유하거나 재사용하기가 어렵고, 학습자의 학습 활동 변경에 따른 적시 적격의 강의 컨텐츠 변경이 어렵다. 따라서 본 논문에서는 분산 컴퓨팅 환경에서 교수 설계자와 다양한 가상 교육 시스템들이 강의 컨텐츠를 공유할 수 있고 재사용할 수 있도록 강의 컨텐츠를 속성을 가진 더 작은 크기로 분해하여 객체화하는 방법을 제시한다. 특히 교수법적인 설계를 근거로 강의 컨텐츠의 재사용 단위를 지도 항목, 테스트, 케이스 예제, 토론, 문제, 디스커버리, 리소스, 시뮬레이션으로 추출하였다. 이것은 개요, 사실, 해보기, 퀴즈, 평가, 탐구 학습, 토론의 구성 요소로 설정하고 이를 CBD 방법으로 구현하였다. 이것은 웹기반 컨텐츠 시스템의 문제점을 해결할 뿐만 아니라 가상 교육 시스템을 이용하는 교수 설계자와 학습자의 컨텐츠 이해도를 높였다.

A Learning Content System which is Objectified with the Reusable Unit of Pedagogical Designs for Distributed Environments

Haeng-Ja Shin† · Kyung-Hwan Park††

ABSTRACT

In this paper, we investigate a problem with Web-based learning contents and introduce the solving method of the problem. To be more accurate, existing Web-based learning contents were one fixed and uniformed courseware file with a large size and HTML format. Also, They were written files with authoring tools of which depended upon providing a multimedia vender. These learning contents were difficult to reuse among cyber education systems and change the applicable contents to the learner for learning experiences in time. So in this paper, we produced reusable and interoperable learning contents among instruction designers and education systems. They were deconstructed into smaller chunks and added to its properties. For the purpose of this producing method, we used the pedagogical designs for units of reuse. These are just tutorial item, test, case example, discussion, problem, discovery, resource, and simulation. These set components of the introduction, fact, try, quiz, test, link-more and tell-more and was implemented with CBD method. As a result, The problem of existing Web-based learning contents system was resolved and then the power of understanding about objectified learning content was increased for the learner and instruction designers.

키워드 : 강의 컨텐츠 시스템(Learning Contents System), 분산 환경(Distributed Environments), 재사용(Reusable), 공유가능(Sharable), 교수법적 설계(Pedagogical Designs)

1. 서 론

웹 기반 교육 시장의 활성화로 각종 교육 사이트 및 가상 대학들이 늘어나면서 학습자가 시공간의 한계를 넘어 학습할 수 있는 기반이 조성되었다. 이러한 웹 기반의 학습 활동은 기존의 교육 패턴과는 다른 자발적으로 학습에 참여하여 학습 결과를 스스로 나타낼 수 있도록 한다거나, 피드백을 요구하여 학습자의 적극적인 참여를 유도하는 패러다임으로 변하고 있다. 이러한 교육 패러다임을 성공시키기 위해서는 컨텐츠의 질적 수준이 높아야 하며, 비선형적이고 동시 병행적인 학습 방식이 효과적이다[1].

학습자의 학습 방식을 중점적으로 다루던 초기의 학습 관리 시스템(Learning Management System)에서는 교수자 모드, 관리자 모드, 학습자 모드에 필요한 기능들을 중심으로 시스템이 나누어져 설계되었다[2]. 웹 기반의 LMS 시스템들은 단순히 웹 페이지만을 제공하는 것이 아니라 오디오

※ 이 논문은 2000학년도 동아대학교 학술 연구 조성비(공모과제)에 의하여 연구되었음.
† 정 회 원 : 동아대학교 기계산업시스템공학부 BK21 교수
†† 정 회 원 : 동아대학교 전기전자컴퓨터공학부 교수
논문접수 : 2003년 7월 14일, 심사완료 : 2003년 10월 9일

나 비디오 자료와 같은 풍부한 컨텐츠를 구비하여야 하므로 전문적인 관리와 유지 보수 기술이 필요하다.

그러나 이러한 교육 시스템의 문제점은 첫째, 코스웨어 중심의 설계로 학습 활동의 융통성이 부족하다는 것이다. 즉, 학습자가 원하지 않아도 코스대로 모든 교육 내용을 학습하도록 구성되어 있다. 둘째, 학습 내용의 가변성이 낮아 해당 분야의 정보나 지식이 빠른 시간 내 학습자가 요구하는 내용으로 수정, 보완되지 않는다. 셋째, 철저한 분석을 통해 완벽한 코스웨어를 만들어야하므로 컨텐츠 개발 시간과 비용이 많이 소요된다. 마지막으로, 웹 기반 교육 시스템들의 학습 컨텐츠는 다양한 학습 저작도구 벤더에 의존적이다. 이러한 여러 형태의 학습 컨텐츠는 학습 내용이 동일한 컨텐츠의 중복으로 컨텐츠의 양을 매우 빠르게 증가시켜서 관리 및 유지보수에 많은 비용이 소요되며, 특히 분산 컴퓨팅 환경에서의 시스템들간의 학습 컨텐츠 공유 및 교환을 더욱 어렵게 만든다.

본 연구에서는 이러한 현재의 웹 기반 LMS 시스템과 학습 컨텐츠의 문제점을 해결하기 위해 다른 학습 컨텍스트에서 재사용할 수 있으며, 특정 벤더에 종속되지 않아 다른 시스템 간에 학습 컨텐츠를 공유할 수 있고 교환 가능한 학습 객체를 제안한다. 특히 효과적인 학습을 위한 교수법적 설계(pedagogical designs) 개념을 근거로 재사용 단위를 설정하여 강의 컨텐츠를 분해하여 객체화하였다.

본 연구의 구성은 다음과 같다. 2장에서는 여러 교육 단체에서 정의한 학습 객체를 고찰하고 이것을 반영하는 학습 컨텐츠의 모델들을 고찰한다. 그리고 3장에서는 본 논문에서 제안한 강의 컨텐츠를 객체화한 이론적 근거와 설계 방법을 보이며 또한 제안한 강의 컨텐츠 객체를 어플리케이션 수준의 재사용 및 검색을 위한 메타데이터 설계 및 컨텐츠 패키징 방법을 언급한다. 4장에서는 객체화된 강의 컨텐츠를 위한 시스템 구조 설계 및 동작 방법과 그 인터페이스 구현을 보인다. 그리고 마지막 5장에서 결론과 향후 개발 방향으로 끝을 맺는다.

2. 관련 연구

분산 컴퓨팅 환경의 인프라를 변화하는 교육 패러다임과 접목시키면서 교육분야에서도 AICC(Aviation Industry CBT Committee), IMS(Instructional Management System), ADL (Advanced Distributed Learning), IEEE LTSC(Learning Technology Standards Committee) 등의 단체들이 학습자 관련 표준, 학습 컨텐츠 관련 표준, 그리고 기술 표준들을 위한 작업들을 활발히 진행시키고 있다.

이 장에서는 교육 종사자 및 단체들에서 연구되고 있는 학습 객체에 대한 정의를 살펴보고, AICC, IDC(Internet Data Centers), ADL 등의 학습 객체 모델링 방법들을 살펴본다.

2.1 학습 객체의 정의

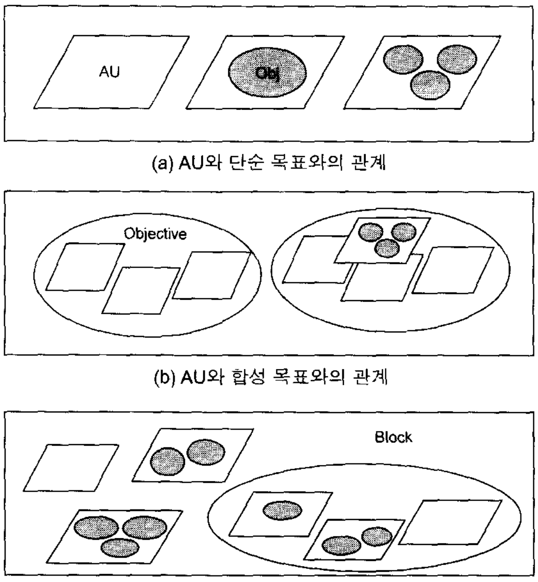
IEEE LTSC(Learning Technology Standards Committee)는 학습 객체(learning object)란 "학습을 하고 있는 중에 사용되거나 재사용 혹은 참조될 수 있는 어떠한 디지털 혹은 비디지털 엔티티"라고 정의하였다[3]. David A. Wiley에 의하면 학습 객체는 "컴퓨터 과학에서의 객체지향 패러다임을 근거로 한 컴퓨터 기반 교육의 새로운 타입의 요소"라고 하였다[4]. Quinn은 적어도 다음 4가지 하부 구성요소 즉, 내용(contents), 기능(function), 학습 목표(learning objective), 인터페이스(look and feel)를 가져야 학습 객체라고 할 수 있다고 하였다[5]. 또한 Albert Ip은 교수법적 패러다임에서 기술적으로 학습 객체를 정의하고자 하였다[6]. 이러한 용어들은 ADL SCORM(Sharable Content Object Model)에서 컨텐츠 객체 혹은 지식 객체(knowledge object)라고 하였고, IMS에서는 특별히 이 용어에 대한 언급은 없고 학습 리소스(learning resources)라는 개념을 사용하였다[7, 8].

이와 같은 학습 객체의 정의들을 분석해볼 때, LTSC의 정의는 너무 포괄적이고, IMS의 리소스란 용어는 "학습에 필요한 읽을 자료"의 수동적인 의미가 내포되어 있다. Quinn이 내린 정의는 S/W분야의 객체지향 접근법의 프로그래밍적 특성이 보인다. SCORM의 정의는 교육 분야 컨텐츠보다 넓은 분야의 컨텐츠를 포함하고 있다. 본 논문에서는 "학습 객체란 웹 기반 가상 대학 교육 시스템에서 어플리케이션 수준의 자동화된 재사용 가능한 디지털 컨텐츠"라고 정의한다.

2.2 학습 객체 모델링

2.2.1 AICC의 학습 객체 모델

AICC의 CMI(Computer Managed Instructions) 지침서에서 코스 구조(course structure)는 초기에 파일에 기반한 브라우저와 서버간의 통신 형태로 컨텐츠의 묶음(package)을 붙임(post) 형태로 설계가 되었기 때문에 학습 컨텐츠 모델은 서버가 코스 컨텐츠를 어떻게 로딩 혹은 브로드캐스팅할 것인가에 대한 통신 부분과 다음에 제공될 컨텐츠가 무엇인지에 중점을 두어 설계를 하였다[9]. AICC의 모든 코스는 학습 목표, 할당 단위(assignable unit), 블록(block) 이라는 요소를 사용한다. 또한 코스를 구축할 경우에는 레슨(lesson)이 할당 단위로 표현되기도 한다. 학습 목표는 단순 목표(simple objective)와 복합 목표(complex objective)로 정의되는데 전자는 AU(assignable unit) 레벨에서의 학습 목표이고 후자는 다양한 AU들, 블럭들 그리고 다른 학습 목표의 조합으로 정의한다. 이것을 (그림 1)로 보인다.



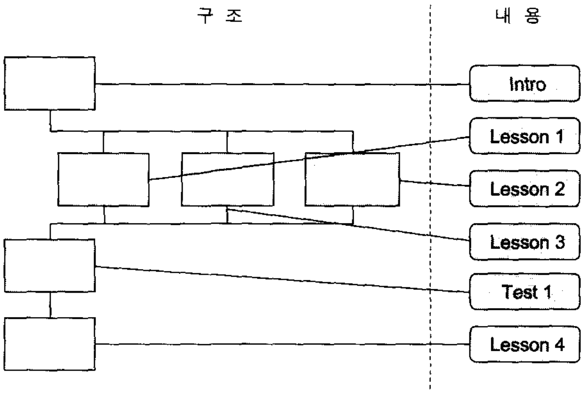
(a) AU와 단순 목표와의 관계

(b) AU와 합성 목표와의 관계

(c) AU 학습 목표 그리고 블럭

(그림 1) AU와 학습 목표들

AICC의 코스는 레슨들의 집합이다. 코스 구조는 간단히 여러 개의 레슨으로 차례로 구성되거나, 많은 레슨들의 트리 구조로 표현되는데, 레슨 단위로 트래킹하도록 한다. 코스 구조의 컨텐츠는 모든 레슨, 평가(test), 그리고 다른 할당 단위들로 이루어진다. (그림 2)는 AICC의 코스 구조와 컨텐츠들의 관계를 보인 것이다[10].

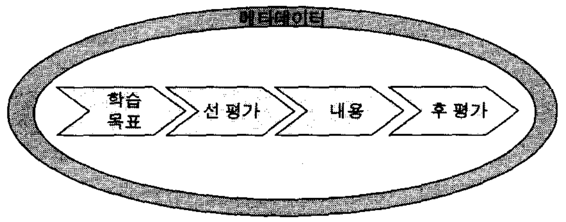


(그림 2) AICC의 코스 구조와 컨텐츠의 관계

2.2.2 IDC 학습 객체 모델

IDC에서는 특정 학습 목표에 기반한 내용과 평가를 포함하는 최소한의 학습 단위로 그 학습 단위를 상징화하는 메타데이터로 둘러싸인 것을 학습 객체라고 정의하였다[11]. 즉, 특정 학습 목표에 중점을 둔 학습 내용의 묶음(chunk)으로 한 개 이상의 정보 객체(텍스트, 이미지, 비디오, 자바 애플릿 등)를 포함하고 있다. (그림 3)는 IDC에서 정의한 학습 객체 모델이다.

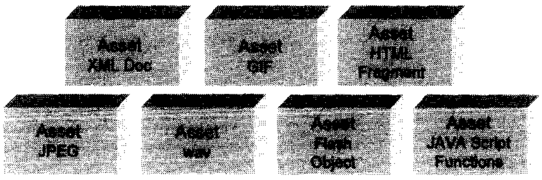
학습 목표는 학습자가 특정 교육적 목표를 도달하도록 도와주는데 그 명세 정도는 보는 관점에 따라 결정한다. 평가는 실제 내용을 학습하기 전의 평가는 학습 토픽에 따라서 생략 가능하고 학습을 마친 후의 평가는 실질적인 테스트 성격을 띤다. 내용은 실질적으로 학습할 내용으로 텍스트, 그래픽, 오디오, 어떠한 상호 작용의 형태, 개념적 어플리케이션을 포함한다. 이러한 내용을 생성하는 도구는 마이크로소프트사의 오피스 도구에서부터 매크로미디어사의 드림위버까지 매우 다양하다. 이러한 학습 객체 모델의 메타데이터는 객체가 포함하고 있는 것을 기술하는데 사용되는데 교육적 내용, 기간, 사용 언어, 객체를 다룰 때 필요한 선지식, 그리고 객체들간의 포함관계 등이 기술된다.



(그림 3) IDC의 학습 객체 모델

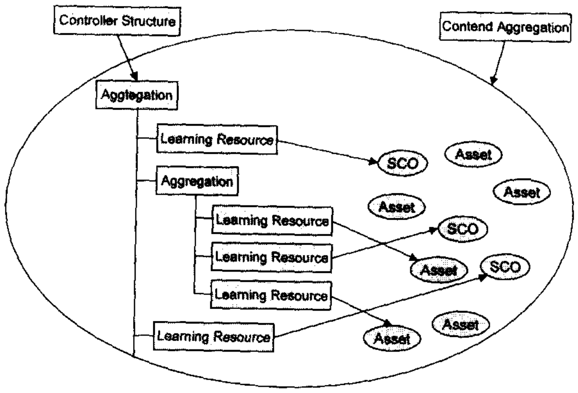
2.2.3 ADL SCORM

SCORM은 비교적 작고 재사용할 수 있는 학습 리소스를 도모할 수 있는 컨텐츠 개념을 지원한다. 여기서 학습 리소스는 교수 단위 형태로 수집될 수 있는 코스, 모듈, 장, 평가 등을 의미한다. SCORM의 객체 모델은 애셋(Asset), SCO (Sharable Content Object), 컨텐츠 집합(Content Aggregation)으로 구성되어 있다. (그림 4)에서 보인 애셋은 컨텐츠에서 대부분 기본이 되는 표현 즉, 텍스트, 이미지, 사운드, 웹 페이지 등 클라이언트로 전달되는 그 어떠한 것도 애셋으로 가능하다[7].



(그림 4) 애셋의 예

SCO는 하나 이상의 애셋 집합을 표현하는데, SCORM 런타임 환경을 사용하는 LMS에 의해 트래킹 되는 학습 리소스의 가장 낮은 입자(granularity) 레벨로 재사용을 위해 학습 컨텍스트에 독립적이어야 한다. (그림 5)에서 컨텐츠 집합은 특정 학습 경험(learning experience) 혹은 학습 컨텍스트를 구축하는데 필요한 학습 리소스, 즉 애셋이나 SCO 들을 수집한 것으로 컨텐츠 구조를 정의하게 한다. 이러한 컨텐츠 구조는 학습자에게 나타나야 할 학습 리소스의 시퀀스를 정의하는 기법을 제공한다.



(그림 5) SCORM의 컨텐츠 집합과 컨텐츠 구조

지금까지 AICC의 학습 객체 모델, IDC의 학습 객체 모델, SCORM의 객체 모델을 살펴보았다. 교육 시스템 표준화에 참여하고 있는 많은 단체들은 ADL SCORM의 CAM(Content Aggregation Model)[12], IMS Global Learning Consortium, Inc.의 CPM(Content Packaging Model)[13], IEEE LTSC의 LOM(Learning Object Metadata)[14]의 사양들을 서로 간에 참조하고 있다. 학습 객체 모델은 SCORM을 참조하고 있으며 컨텐츠 패키징은 IMS의 CPM을 참조하고 있다. 또한 학습 객체의 의미와 구조를 기술하는 메타데이터는 LOM을 참조하여 표준화를 진행하고 있다.

그러나 AICC의 경우, 용어 "할당 단위"란 단지 0번 이상의 단순 학습 목표를 가질 수 있다고만 밝혀 그 정의가 모호하다. 또한 학습 객체를 기술하기 위해 ASCII 텍스트에서의 콤마 구별자를 사용하여 데이터를 포맷하거나, 파일 단위로 저장하는데, 이러한 디스크립터들은 학습 시스템 특유의 의미를 잘 반영할 수 없다. IDC의 경우, 학습 객체는 학습 컨텐츠를 위한 교수 단위(unit of instruction) 개념 없이 단순화시켰다. SCORM 객체의 경우는 교육 분야의 학습 객체 개념이 없다. 또한 SCORM 객체에서 참조하는 LOM 메타데이터의 경우, 교육 컨텐츠 개념 이외의 매우 많은 요소들이 포함되어 있으며, 애셋 레벨에서부터 방대한 LOM의 메타데이터를 적용하여 컨텐츠 패키징은 매우 복잡하고 방대하다.

3. 강의 컨텐츠의 객체화

본 연구에서는 교육 표준화 단체들의 복잡하고 방대한 표준을 그대로 따르기보다는 실질적으로 가상 교육 시스템에서 강의 컨텐츠 저작자 혹은 제공자가 학습자의 학습활동 변화에 빠르게 대처하기 위해 강의 컨텐츠를 교수법적 설계 방법 근거로 모델링한다. 이러한 모델링 방법으로 객체화한 강의 컨텐츠는 분산 컴퓨팅 환경에서의 교육 시스템들과의 공유 및 재사용, 검색을 위하여 XML 문서로 그 정보를 작성한다.

3.1 개 요

재사용가능 컨텐츠란 학습 컨텍스트에 독립적인 컨텐츠를 의미한다. 즉 새로운 개발 도구나 전달 플랫폼을 지니는 다양한 학습 훈련 상황과 다양한 학습자들에게 사용될 수 있다는 것을 의미한다. 또한 상호작용 가능한(interoperable) 컨텐츠란 컨텐츠를 생성하는데 사용되는 도구나 전달될 플랫폼에 상관없이 여러 어플리케이션이나 환경, 하드웨어 및 소프트웨어 구성되어도 동작하는 컨텐츠를 의미한다.

웹을 기반으로 한 분산 컴퓨팅 환경에서 재사용 가능하고 상호작용 가능한 학습 컨텐츠를 설계하기 위해서는 기존의 가상 교육 시스템에서 코스웨어로 다루는 하나의 획일적인 큰 컨텐츠를 분해하여야 한다. 여기서 분해되는 학습 컨텐츠의 크기는 목적에 따라 나누어질 것이므로 상관하지 않는다. 컨텐츠의 분해를 위해 본 연구에서는 웹 기반 교육 시스템의 가상 강의실에서 효과적인 몇 개의 교수법적 설계 방법론을 복합적으로 도입한다.

3.2 이론적 근거

Tutorial, Drill and Practice 교수법에서는 지도-연습-평가의 형태로 학습이 이루어진다. 이러한 형태에서 재사용 가능한 단위는 지도를 위한 각각의 tutorial item들과 평가를 위한 테스트가 재사용 단위로 적합하다. Case Method 기법에서는 학습자들이 학습한 것을 예제로 분석하고 그대로 따라 해볼 수 있어야 한다. 그러므로 재사용 단위는 case example이 가능하다. Distributed Problem-based Learning 기법에서는 다양한 영역에서의 문제를 해결함으로써 학습하는 기법이므로 재사용 가능한 단위는 문제가 가능하다. Exploratory Learning 기법에서는 탐구하는 과정을 통해 학습하는 기법이므로 사실이나 개념을 찾을 수 있는 것, 즉 리소스나 디스커버리(discovery)가 재사용 단위이다. Resource-based Learning에서는 학습자 중심의 시스템 설비를 이용하는 동작을 통해 학습하는 방식이므로 리소스가 재사용의 단위이다. Goal-based Learning 기법에서는 문제를 해결하고 과제를 완수함으로써 학습하는 기법으로 그에 상응하는 시나리오 혹은 시뮬레이션이 필요하다. 그러므로 재사용 가능한 단위는 시뮬레이션이다[15]. 이상과 같은 고찰에서 교수법적 설계 방법을 고려한 재사용 가능한 단위는 tutorial item, test, case example, discussion, problem, discovery,

〈표 1〉 교수법적 설계에서의 재사용 가능 단위

교수법적 설계	재사용 단위
Tutorial, Drill and Practice	지도 항목, 테스트
Case Method	케이스 예제, 토론
Distributed Problem-based learning	문제
Exploratory Learning	리소스, 디스커버리
Resource-based Learning	리소스
Goal-based Learning	시뮬레이션

resource, simulation 이 가능하다. 이를 <표 1>에 정리하였다.

3.3 교수법적 개념의 재사용 단위를 적용시킨 학습 객체 모델링

본 연구에서는 표준의 SCORM 런타임 환경과 호환을 위해 SCORM의 애셋 개념은 도입한다. 다시 언급하자면, 애셋 요소는 각종 미디어 데이터, 텍스트, 이미지, 사운드, 웹 페이지 등 웹 클라이언트로 전달될 수 있는 모든 것의 전자적 표현을 의미하는데 가공되지 않은 리소스(raw resource)이다. 대부분의 경우 하나의 파일로 취급된다. SCORM의 경우 학습 객체 개념이 없는 SCO로 조합하였다. 그러나 본 연구에서는 애셋 요소들을 조합하여 교수법적 개념의 재사용 단위를 적용시킨 LIO(Lecture Item Object) 요소, 그리고 교수 단위(unit of instruction)의 개념을 적용시킨 것은 레슨 요소 구성하였다. 각 요소는 다음 절에서 설명한다.

3.3.1 LIO 요소

LIO 요소는 웹에서 클라이언트로 전달할 수 있는 데이터, 이미지, 사운드 등 가공되지 않은 리소스를 개념적으로 그룹화시킨 것으로 시스템에서 전송하고 트래킹할 수 있는 최소의 논리적 단위이다. LIO 요소는 교수법적 설계 방법에서 분석한 재사용 단위를 근거로 LIO 요소가 타입을 가지도록 하였다. 설계된 타입으로는 개요(introduction), 사실(fact), 퀴즈(quiz), 해보기(try), 탐구 학습(link-more), 토론(tell-more), 평가(test) 들이다.

개요, 사실은 학습할 실질적 본문으로 학습할 개요와 사실들을 나타낸 텍스트와 관련 여러 가지 그래픽 등으로 구성된 교수법적 설계에서 재사용 단위인 지도 항목이나 리소스로 볼 수 있다. 퀴즈는 간단한 자습 문제이고, 평가는 단원 학습 이해도를 측정하는 문제이므로 교수법적 설계의 재사용 단위의 테스트나 문제로 본다. 해보기는 재사용 단위의 케이스 예제, 시뮬레이션에 해당하며, 탐구 학습은 더 많은 정보를 얻어 탐구하도록 하는 URL 기반의 리소스, 디스커버리에 해당한다. 토론은 전문가 혹은 교수자와 토론이 가능하므로 재사용 단위 토론에 해당될 수 있다.

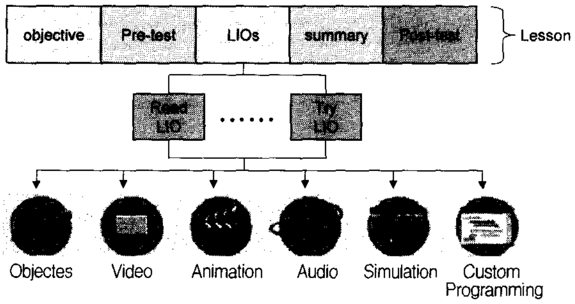
〈표 2〉 LIO 구성 요소

LIO 구분	LIO 구성요소	설 명
Read LIO	개 요	내용에 대한 개요
	사 실	지식의 사실 / 개념을 설명
Try LIO	해 보 기	단계별 동작표, 데모, 시뮬레이션
	퀴 즈	간단한 질문 (self-test)
	평 가	학습 평가
	탐구 학습	더 많은 지식의 탐구를 위한 링크
	토 론	교수자 등과의 질문과 답 / 토론

특히 강의 컨텐츠를 읽어서 지식을 습득하는 학습 과정과 실제 직접 해 봄으로써 학습하는 과정으로 분류하였다. 가상 강의실에서의 학습자는 정적인 학습으로 개념이나 사실을 설명한 것을 읽음으로써 지식을 습득한다. 이 부분을 정적 혹은 read LIO라고 한다. 이 LIO 요소에는 개요, 사실들의 타입이 포함된다. 학습자와 상호작용이 필요한 동적인 학습이 필요한 부분을 동적 혹은 try LIO라고 한다. 이 LIO 요소에는 해보기, 퀴즈, 탐구학습, 토론, 평가들의 타입이 포함된다.

3.3.2 레슨 요소

레슨 요소는 특정 학습 컨텍스트에 관련되어 어떠한 학습 목표를 가진 내용들을 구성하는 LIO 요소들의 묶음(chunk)으로 레슨 요소에는 Read LIO, Try LIO 뿐만 아니라 학습 목표와 선 평가(pre-test), 후 평가(post-test), 요약(summary)도 포함이 된다. 교수 설계자는 그 레슨의 학습 활동에서 성취해야 할 목표는 학습 목표, 레슨을 학습하기 전 학습자의 선행 지식은 선 평가, 실질적 학습 활동은 몇 개의 read LIO, try LIO들을 통해 학습하며, 학습한 후 요약 정리나 다음 단계에 필요한 학습 정보를 언급한 요약, 학습을 마친 후 학습 활동의 통과나 실패(pass/fail)를 결정하기 위한 후 평가들이 포함된다.

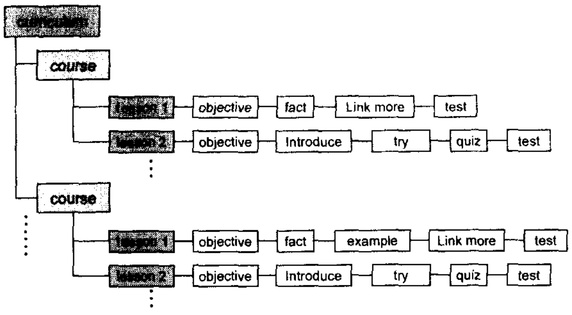


(그림 6) 레슨의 요소와 구성

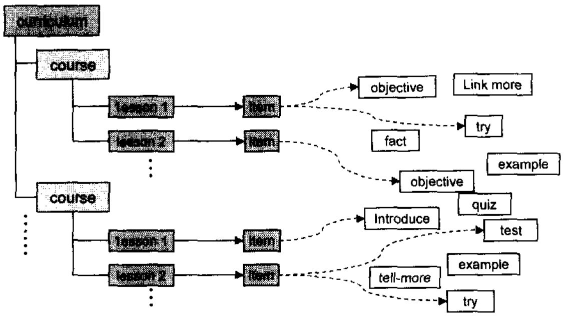
3.3.3 코스 구조

코스 구조(course structure)는 레슨 요소들의 집합이다. 코스의 학습 목표에 따라 시퀀스가 정의되어 코스 구조를 구성하게 된다. 과거 CBT(Computer Based Training)기반의 학습 저작 도구들은 전형적으로 그 도구에 특화된 데이터 포맷과 코스웨어를 지원했다. 이러한 전형적인 코스웨어는 학습 순서가 도구에서 정해진 대로 진행되어야 함을 의미한다. (그림 7)는 전형적인 코스웨어 구조를 보인 것이다. 학습 컨텍스트에 독립적인 내용들이 정적으로 포함되어 컨텐츠의 양을 증가시킨 결과를 초래했다.

강의 컨텐츠를 학습 컨텍스트에 독립적인 학습 객체로 설계하였을 경우 (그림 7)은 (그림 8)과 같이 설계 가능하다. 학습 컨텍스트에 독립적인 학습 객체는 여러 다른 레슨에서 공유할 수 있으며, 다른 학습 경험에서 재사용할 수 있어 컨텐츠의 양은 현저히 줄어든다.



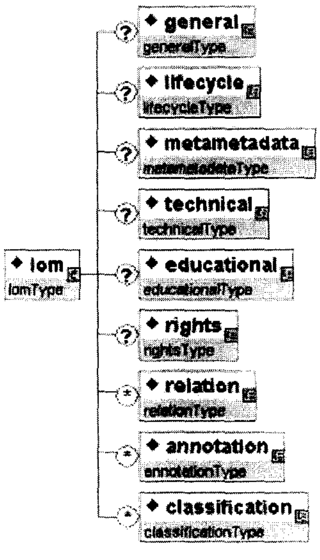
(그림 7) 전통적인 코스 구조



(그림 8) 객체화가 반영된 코스 구조

3.4 객체화된 강의 컨텐츠를 위한 메타데이터 설계

메타데이터는 데이터의 데이터로서 컨텐츠 모델을 구성하는 각 요소들의 속성을 기술하는 방법으로 컨텐츠에 대한 정보를 제공한다. 이러한 메타데이터는 컨텐츠를 더 쉽게 이용하거나 검색할 수 있도록 인덱스화된 레이블로 기술되는데, 정확하게 기술하기 위해 메타데이터 요소가 정밀하여야 한다. 교육 단체들의 메타데이터 표준안인 LOM(Learning Objects and Metadata)를 살펴보고 본 연구에서의 메타데이터 설계에 대해 고찰한다.



(그림 9) LOM 메타데이터 태그 및 구조

3.4.1 학습 객체 메타데이터(LOM) 개요

LOM은 다양한 형태의 교육 자료에 필요한 속성을 적용하여 폭 넓은 적용성과 확장성을 갖도록 한다. 그 의미에 따라 9가지 범주로 나누어 계층적으로 구조화 하였다. 그 9가지 범주는 <general>, <lifecycle>, <metametadata>, <technical>, <educational>, <rights>, <classification>, <relation>, <annotation>이 있으며, <educational> 범주 요소들이 교육 분야의 특성을 반영하는 정보를 가지고 있으며 <technical>범주 요소들이 리소스를 지닌 시스템 특성을 나타내는 정보를 가지고 있다[14].

3.4.2 객체화된 강의 컨텐츠를 위한 메타데이터 설계

본 연구에서는 객체화된 강의 컨텐츠 내용의 의미와 구조를 기술할 수 있는 XML 기반의 문서로 저장한다. 교육 표준화 시스템들과의 상호작용을 위하여 LOM의 메타데이터를 참조하여 설계하지만 9가지 범주를 모두 기술하는 태그를 지원하기 보다 학습 컨텍스트에서 자주 사용되는 것을 선택하여 메타데이터 편집기로 작성하여 XML 문서로 저장한다. 강의 컨텐츠의 가장 하위레벨 즉, 애셋에 대해서는 메타데이터를 지원하지 않으며, LIO 요소들에서 제공하는 메타데이터는 학습 컨텍스트에 독립적이다. 그러나 LIO 요소들의 집합인 레슨 메타데이터나, 레슨의 집합인 코스의 메타데이터는 학습 컨텍스트에 의존적이다.

본 연구에서의 객체화된 강의 컨텐츠 모델에서 메타데이터 편집기를 이용하여 작성하는 화면을 보인 것이 (그림 10)이다. LOM의 <general> 범주에 속하는 제목(Title)은 컨텐츠의 실제 제목이며 항목 식별자(Item Identifier)는 유일하게 식별 가능한 GUID(globally unique identifier)를 의미한다. 키워드(Keyword)는 컨텐츠를 정의하는데 사용되는 텍스트적인 단어나 구를 의미하며 기술(Description)은 우선하는 키워드를 사용하여 컨텐츠에 대한 텍스트적인 설명이며 구조(Structure)는 컨텐츠가 구성되는 방법으로 linear, tree의 값을 가진다. LOM의 <lifecycle>범주에 속하는 버전(Version), 배포(Contribute), 날짜(Date)는 컨텐츠의 현재 버전, 배포자, 생성되거나 제출된 날짜를 의미한다. LOM의 <metametadata>의 범주에 속하는 메타데이터 스킴(Metadata Scheme)은 시스템들간의 호환을 위해 LOM을 사용하므로 LOM이라는 값을 가지며 버전을 함께 명시한다. LOM의 <technical>범주에 속하는 포맷(Format)은 MIME 타입에서 지원하는 타입은 어떤 것이든 기술 가능한데, 크기(Size)를 바이트 단위로 표시하고 그 실제 위치를 로케이션(Location)에 명시한다. LOM의 <educational> 범주에 속하는 학습 리소스 타입(Learning Resource Type)은 LIO 요소의 교수법적 설계를 반영한 개요, 사실, 퀴즈, 해보기, 탐구학습, 토론 등이 가능하며, 인터랙티브 타입(Interactivity Type)은 학습 객체에 의해 지원되는 학습 모드로 active,

expositive, mixed 값을 가질 수 있다. Active는 학습자의 행동이 필요한 것(learning by doing)이고, expositive는 학습자가 주로 읽어서 지식을 습득하는 것(passive learning)을 의미한다. 용어 active와 expositive는 LOM의 용어를 따른 것으로 본 연구에서는 read LIO, try LIO라고 하였다. 그리고 인터랙티브 레벨(Interactivity Level)은 학습자에 대한 학습 컨텐츠의 난이도를 의미하는 것으로 low, medium, high의 값을 가진다. 마지막으로 LOM의 <rights>범주에 속하는 비용(Cost), 저작권(Copyright and Other Restrictions), 권한 기술(Right Description)은 학습 컨텐츠 사용에 대한 요금과 저작권, 접근 권한에 대한 정보를 기술한다.

Metadata		
Title : UML		Version : 1.0
Item Identifier: 001_item_uml_software_engineering		Date : 2003-6-30
Keyword : class, workflow model, component diagram, state		
Item Description : The UML defines model types that functional requirements and activity workflow models, to class structure design and component diagrams		
Contribute : Dong-A Univ.		Structure : Linear
Format : portable document format (pdf)		Size : 4100 bytes
Location : http://www.donga.ac.kr/cyberedu/s_w/uml/item_001.pdf		
Interactivity Type : expositive		Interactivity Level : low
Learning Resource Type : Fact		Typical Learning type : 2h
Cost : No	Copyright & Other Restrictions : Yes	Metadata Scheme: LOM1.0
Rights Description : permission		

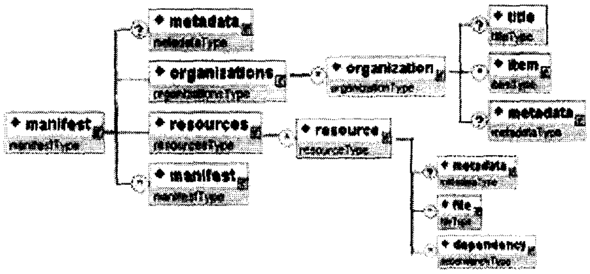
(그림 10) LIO 메타데이터 편집 인터페이스

3.5 객체화된 강의 컨텐츠의 패키징 설계

저작된 강의 컨텐츠는 다른 웹 기반 교육 시스템들에게 배포되어야 한다. 교육 단체들의 패키징 표준안인 IMS CPM (Content Packaging Model)을 살펴보고 본 연구에서의 학습 컨텐츠 패키징 설계에 대해 고찰한다.

3.5.1 IMS 컨텐츠 패키징 모델 개요

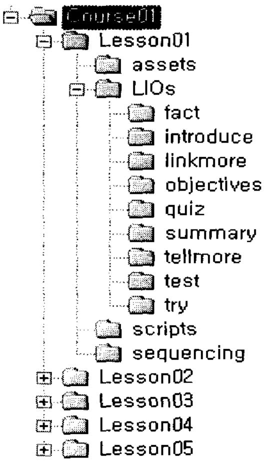
IMS의 CPM은 코스 구조에 관련된 컨텐츠를 하나의 파일(.zip or .jar)로 묶어 시스템들 사이 전달 가능하도록 했다. 묶여진 파일에는 컨텐츠가 어떻게 구성되었는지를 알 수 있는 매니페스트(manifest) 파일인 imsmanifest.xml과 실질적 컨텐츠 리소스 파일들로 구성되어 있다. 매니페스트 파일은 XML 기반의 문서이다. 이 문서에는 컨텐츠에 대한 메타데이터를 기술한 <metadata> 부분과 코스의 구조가 어떻게 구성되었는지를 알 수 있는 <organizations>, 그리고 물리적으로 학습 리소스가 위치하는 곳을 참조할 수 있고 그 리소스를 설명한 <resources> 부분으로 구성되어 있다. 또한 그 태그 요소들은 하위요소를 가진다[13].



(그림 11) IMS 컨텐츠 패키징 모델을 위한 메타데이터 태그와 그 구조

3.5.2 객체화된 강의 컨텐츠의 패키징 설계

본 논문에서는 시스템들 사이 배포 가능한 강의 컨텐츠 단위로 패키징하기 위해 다른 교육 표준 시스템들처럼 IMS CPM을 도입한다. 전송된 강의 컨텐츠의 패키징은 시스템 상에서 (그림 12)와 같은 파일 구조를 가진다. assets 디렉토리에는 각종 그림과 멀티미디어 파일들이 위치하고 LIOs에는 각 LIO 요소들과 그것을 기술한 메타데이터 파일이 위치한다. scripts 디렉토리에는 웹 브라우저와 동작할 자바스크립트 파일들, sequencing에서는 학습 시퀀스 정보 메타데이터들이 위치하며, 각 레슨에 대한 정보를 가진 메타데이터 파일은 각 Lesson 디렉토리에, 코스에 대한 정보를 가진 메타데이터 파일은 각 Course 디렉토리에 위치되며, 전체 강의 컨텐츠를 패키징 한 매니페스트(imsmanifest.xml) 파일 및 XML문서 태그와 구조를 인식하는 스키마 파일들은 루트 디렉토리에 위치된다.



(그림 12) 코스 컨텐츠를 패키징한 파일 구조

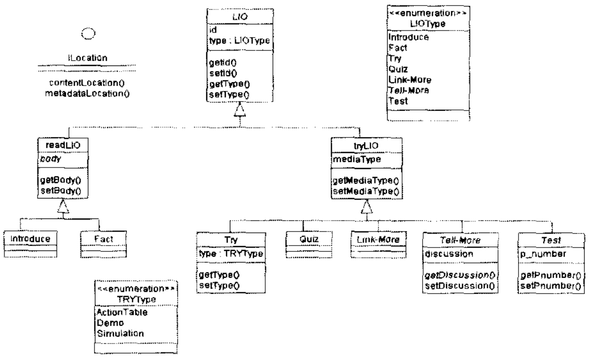
4. 객체화된 강의 컨텐츠 시스템 설계 및 구현

본 장에서는 교수법적 설계 개념을 근거로 분해하여 객체화한 강의 컨텐츠를 객체화하기 위해 교수법적 설계를 지원하는 LIO 객체를 클래스로 설계한다. 이 같은 LIO 객체를 지원하는 시스템에서는 컨텐츠 패키징 단위로 브라우저에 전송하며, LIO 단위로 트래킹할 수 있다. 또한 원격,

로컬의 객체 저장소에 저장된 강의 컨텐츠는 각 레슨에서 어플리케이션 간 LIO 단위로 공유 가능하며 코스 재구성이 LIO 단위의 재사용이 가능하다. 다음 절에서는 LIO 객체 설계 및 지원 시스템의 구조와 설계, 그리고 LIO 컨텐츠를 재사용하기 위한 교수 설계자 모듈과 학습자 모듈을 설명하고 그것을 구현한 인터페이스를 보인다.

4.1 LIO 학습 객체 설계

본 논문에서는 어플리케이션간 자동 교환 가능한 LIO 학습 객체를 위해 클래스로 설계하였다. 이것은 자바빈 컴포넌트로 구현된다. 사용자와의 상호작용을 고려해 정적 LIO와 동적 LIO를 구분하여 서버 접근 횟수를 줄이고자 했다. 개요 및 사실 LIO를 위해서는 Introduce 및 Fact 클래스로 해보기, 퀴즈, 탐구학습, 토론, 평가 LIO를 위해서는 Try, Quiz, Link-More, Tell-More, Test 클래스로 작성하였다. 정적 클래스는 읽어야 할 자료를 어플리케이션에서 사용할 수 있도록 getBody(), setBody() 메소드로 작성하고, 동적 클래스는 사용자의 이벤트에 반응할 수 있는 미디어 타입을 위한 메소드를 사용하였다. 이런 각각의 클래스는 어플리케이션에서 자유롭게 다루기 위해 Id와 타입을 위한 메소드를 마련하였다.



(그림 13) LIO 클래스 계층도

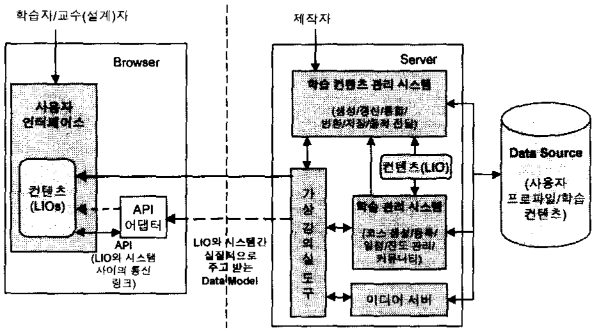
4.2 LIO 학습 객체 지원 시스템 구조 및 설계

분산 컴퓨팅 환경에서의 학습 컨텐츠 관리 시스템은 본질적으로 여러 형태의 원시 파일을 읽어 코스 구조를 트리 형태로 생성하고 이것을 DBMS나 파일로 저장한 후 LMS에서 교수 설계자 혹은 학습자에게 적시에 편집할 수 있도록 출력 인터페이스를 제공한다. 학습 컨텐츠 관리 시스템의 기본적인 기능을 <표 3>에서 보인다.

웹 기반 분산 환경에서 객체화된 강의 컨텐츠를 지원하는 시스템은 웹 브라우저에서 서버의 교육 시스템으로 시동(launch)하여 객체화된 컨텐츠를 코스웨어로 묶인 패키징 단위로 전송한다. 그러나 트래킹할 수 있는 최소의 논리적 단위는 LIO 단위로 LIO 학습 객체 실행을 위한 전체적인 시스템 구조는 (그림 14)와 같다.

〈표 3〉 학습 컨텐츠 관리 시스템의 기능

LCMS 기능	설 명
학습 객체 저작	웹 에디터, 오피스 도구 혹은 전용의 학습 객체 저작 도구 등을 사용하여 생성
학습 객체 저장	생성된 학습 객체를 중앙 객체 저장소에 저장
학습 객체 검색	학습 객체를 설명하는 메타 태그를 통해 로컬/원격 시스템의 중앙 객체 저장소에 있는 학습 객체를 검색
학습 객체 전달	객체 저장소에 저장된 학습 객체를 전달 엔진을 통해 학습자에게 전달
다양한 포맷지원	저장된 학습 객체는 XML 문서로 그 정보가 저장되어 웹 브라우저 뿐만 아니라 다양한 포맷 즉, CD, 인쇄물, PDA 등의 포맷으로 재사용



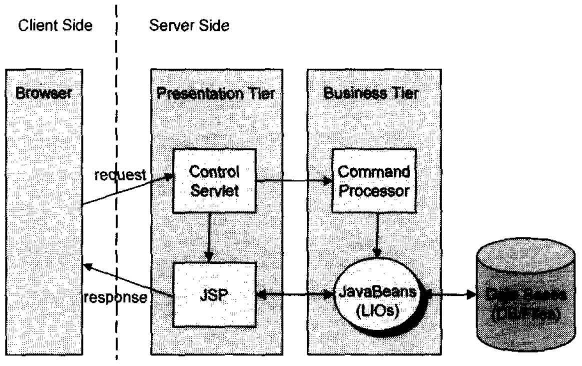
(그림 14) LIO 학습 객체 지원 시스템 구조

저작자에 의해 서버의 데이터 티어의 객체 저장소나 DB에 저장된 객체화된 강의 컨텐츠는 클라이언트측의 학습자에 의해 시동된다. 학습자의 학습 컨텍스트에 따라 해당 LIO 학습 객체들이 전달된다. 가상 강의실 도구와 통신되는 웹 인터페이스를 통해 객체와 서버의 시스템은 통신하게 된다. API 어댑터는 컨텐츠 객체와 LMS 혹은 LCMS와의 통신을 위한 중간 인터페이스로 제공되는데, 컨텐츠 객체의 실행, 상태 관리, 데이터 전달의 기능을 수행한다. 더욱 강화된 기능들은 학습 관리 시스템(LMS)에서는 교육 과정을 운영하기 위한 기능들을 제공하며, 학습 컨텐츠 관리 시스템(LCMS)에서는 학습 컨텐츠의 저작, 검색, 저장, 다양한 포맷 지원 및 동적 전달 등의 기능들을 제공한다.

웹 기반 분산 환경에서 LIO 학습 객체를 지원하는 시스템 구조에서 LIO 학습 객체를 컨텐츠로 다루는 시스템에 필요한 클래스만을 떼어서 서버측 클래스 구조를 보인 것이 (그림 15)이다. 이 클래스들은 특정 WAS(Web Application Server)를 필요로 하는 중량급 EJB로 구현하지 않고 웹 서버 내에 자바빈 컴포넌트만으로도 구현할 수 있는 경량급 프레임워크를 사용하였다[16, 17].

각 LIO 객체는 CBD(Component Based Development) 기반으로 생성한다. 웹 인터페이스에서의 특정 LIO 요청은 서블릿을 통해 서버의 백엔드로 전송한다. 이때 브라우저의 사용자에게 들어온 요청을 웹 화면의 렌더링 담당 부분과 실제적인 동작을 담당하는 비즈니스 로직 담당 부분으로 따

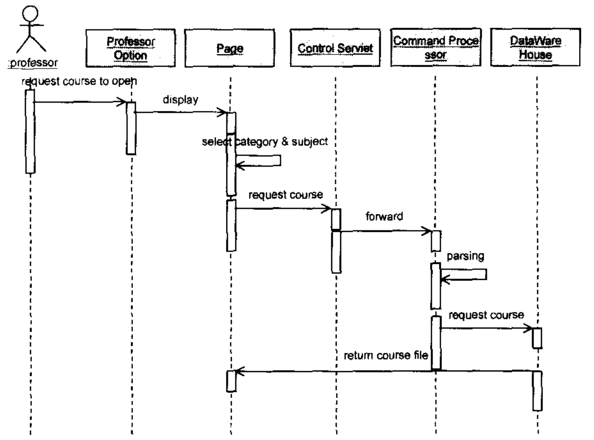
로 분리하여 설계하는 것은 MVC(Model View Controller) 패턴의 효과를 얻기 위해 View와 Control을 따로 구현하기 위한 것이다. 즉, 사용자 화면에 디스플레이하기 위한 렌더링은 JSP가 맡고 사용자의 요청과 그에 대한 응답은 ControlServlet이 맡도록 설계하였다. ControlServlet의 요청을 각 LIO객체를 구현한 각 자바빈이 직접 받는다면 새로운 형태의 LIO 학습 객체 타입을 생성하여 추가하기란 쉽지 않다. 그래서 새로운 LIO 학습 객체 타입으로의 확장을 위해 그 중간자로 CommandProcessor 인터페이스를 두어 각 LIO 객체는 그 인터페이스를 구현하도록 하였다. 이러한 Command Processor는 비즈니스 티어의 복잡함을 숨기고 요청한 LIO 객체를 호출할 수 있으며 이후 트랜잭션이나 보안을 위해 확장하기가 용이하다. 이상과 같은 프로세스 설계는 각 티어의 결합도를 줄이고, MVC 패턴을 사용하여 앞으로 특정 WAS의 EJB로 확장하여 더욱 복잡한 기능들을 추가 가능하다[18, 19].



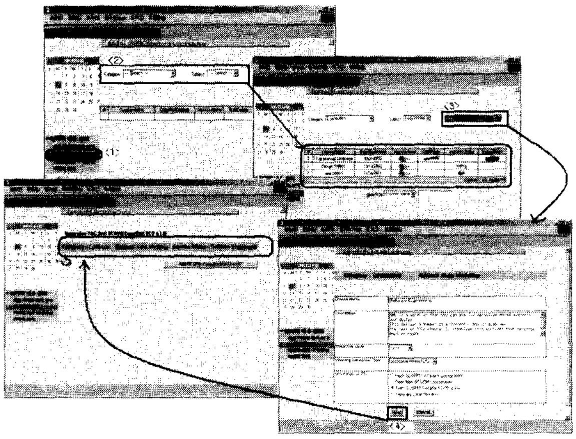
(그림 15) 확장성 있는 멀티-티어 LIO 클래스 설계 구조

4.3 교수 설계자 모듈과 인터페이스 구현

교수자 혹은 교수 설계자는 패키징 그대로의 코스웨어나 학습 객체의 논리적 결합을 가진 코스웨어를 얻어 학습자의 학습 활동에 따라 적시 적격의 학습 컨텐츠를 재구성해야 한다. (그림 16)은 교수 설계자가 객체 저장소에 저장된 코스웨어를 얻어 자신의 학습 컨텐츠로 재구성하기 위해 패키징 파일을 읽어 들이는 시퀀스 다이어그램이다[20]. 객체 저장소 혹은 DB에 저장된 학습 컨텐츠에서 원하는 분류(category)와 주제(subject)를 선택하여 코스를 요청하면 그 요청을 CommandProcessor에서 해당하는 분류와 주제에 해당하는 코스를 검색하여 관련 코스 파일을 되돌려준다. 이때 교수자는 관련 분야를 테이블 형태의 리스트로 얻는다. 얻은 코스의 전반적 개요를 보고 어떤 형태로 TOC를 사용할 것인지 결정하여 서버로 전송한다. 이를 구현한 인터페이스가 (그림 17)이다. 객체 저장소에서 저장된 학과목 메타데이터에서 "컴퓨터" 분야로 분류된 것 중 "데이터 모델"과 관련 주제를 검색하면 기존의 찾은 학과목의 이름, 작성일, 내용보기, 레슨의 편집의 유무 상태, 편집한 후 새로운 코스로 받아들여졌는지의 유무 등의 상태를 알 수 있는 테이블의 리스트를 얻으며 또한 관련된 새로운 학과목 정보를 가져온다. 새로운 학과목의 이름, 간략한 기술 정보, 학습자와의 상호작용 레벨, 교육 시스템에 의해 지원되는 학습모드의 정보를 확인하고 TOC를 얻는데 이때, 교육 시스템이 지원하는 여러 구조 중 원하는 코스 구조 형태를 선택한다.

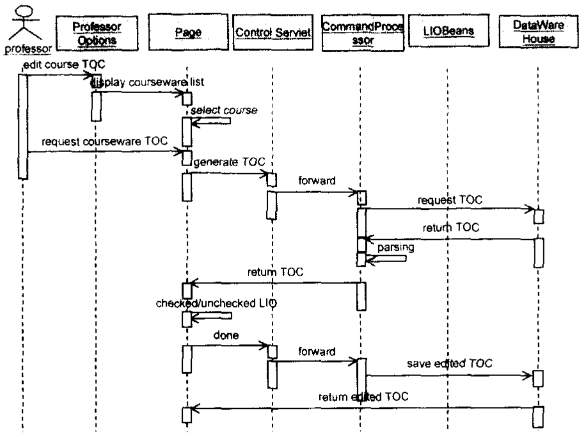


(그림 16) 객체 저장소에서 코스 얻기 시퀀스 다이어그램

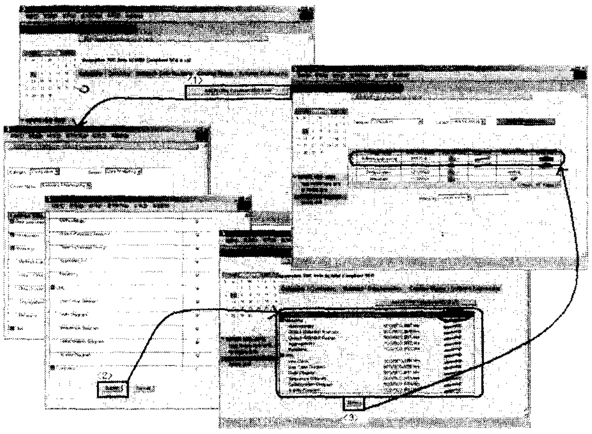


(그림 17) 원하는 코스 얻기 인터페이스

읽어 들인 코스의 TOC(Table of Content)를 이용하여 교수자가 새로운 학습 컨텍스트에 맞게 재구성하기 위해 레슨 혹은 LIO 단위로 편집할 수 있다. 예를 들어, (그림 19)에서 읽어 들인 코스의 리스트 형태의 TOC를 새로운 학습 컨텍스트에 맞게 교수자가 check/uncheck 하여 결정한 후 제출하면 각 레슨 혹은 LIO 단위의 TOC를 학습 컨텍스트에 맞게 내용을 편집할 수 있도록 테이블 인터페이스를 제공한다. 편집을 위해 리소스의 저장 위치와 편집 선택 아이콘으로 편집 가능하며, 최종적으로 수정된 TOC를 학과목 코스로 등록하기 위해 요청 상태로 만든다. 이후 요청 수락과 함께 객체 저장소에 재저장한 후 학습자가 학습할 수 있도록 한다.



(그림 18) 학습 컨텍스트에 맞게 코스의 LIO 편집하기 시퀀스 다이어그램

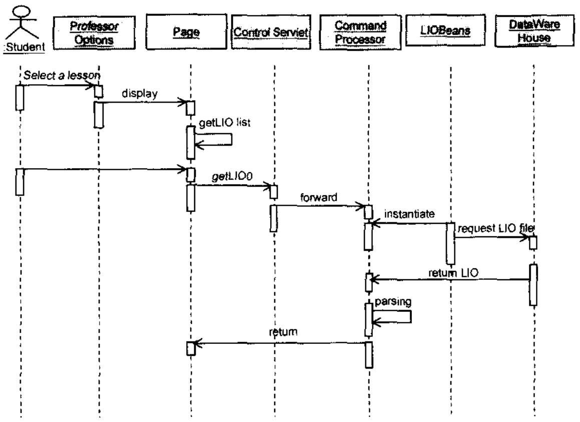


(그림 19) 얻어온 코스 편집하기 인터페이스

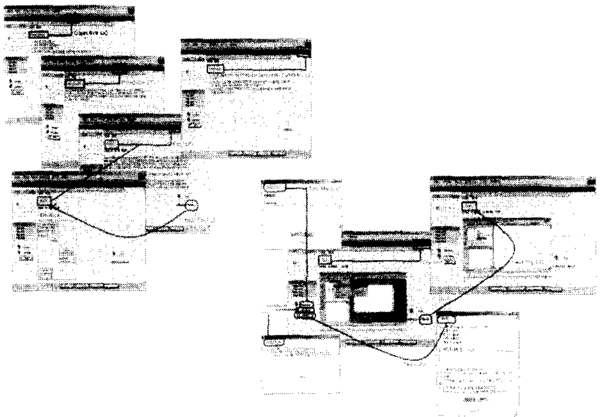
4.4 학습자 모듈과 인터페이스 구현

가상 강의실에서 학습자는 학습 컨텍스트의 특정 레슨에서 학습 트리 혹은 제공되는 웹 인터페이스를 이용해 학습하는 학습자의 동작 과정이다. 학습자가 Read 혹은 Try LIO들을 선택하여 학습하는 과정을 시퀀스 다이어그램으로 보인 것이 (그림 20)이다. 이 시퀀스 다이어그램에서 보여주듯이 사용자가 어떤 LIO를 선택했느냐에 따라 LIOBeans는 그 객체의 타입을 얻어올 수 있도록 인스턴스화 된다. CommandProcessor는 여러 타입의 LIO 객체를 추상화시켜 웹 요청을 전달하는 ControlServlet을 일괄적으로 다룰 수 있게 한다.

학습자가 학습 활동에 필요한 인터페이스를 구현한 것이 (그림 21)이다. 이미 교수자는 학습 컨텍스트에 맞게 학과목의 컨텐츠를 편집하고 객체 저장소에 저장하여 학습자가 이용할 수 있는 상태이다. 학습자는 정적 성질의 readLIO 즉, 학습 목표, 개요, 사실, 요약 메뉴를 통해 선택했을 경우 브라우저의 컨텐츠 영역에는 해당 객체 저장소의 위치에서 그 내용을 가져와 브라우저에 출력된다. 학습자가 서버와의 상호작용이 필요한 tryLIO 즉, 해보기, 탐구학습, 토론, 평가를 선택하여 학습하는 동안에는 서버의 다른 시스템과 상호작용이 필요하다. 해보기 메뉴를 선택하면 데모 형태이거나 혹은 단계적으로 따라하기의 경우 사용자의 다음 단계 선택에 맞추어 브라우저의 컨텐츠 영역에 출력되며, 탐구학습 메뉴를 선택하면 학과목의 심층학습을 위해 관련 사이트, 혹은 관련 컨텐츠로 링크를 걸어 학습하게 하였다. 또한 토론 메뉴는 전문가 혹은 교수자와의 묻고 답하기 형태를 학습자가 주제(제목)와 내용을 작성하여 보낼 수 있게 하였으며 평가 메뉴는 각 레슨의 객관식 주관식 문제에 답을 작성하여 서버로 보내도록 하였다. 그림의 왼쪽 위는 readLIO를 실행한 화면, 오른쪽 아래는 tryLIO를 실행한 화면이다.



(그림 20) LIO 학습 객체 학습하기 시퀀스 다이어그램



(그림 21) 학습자의 read LIO 및 try LIO 학습 객체 실행 인터페이스

5. 결 론

본 연구에서는 웹기반 코스웨어 중심의 강의 컨텐츠를 재사용 가능하고 시스템들 간의 공유 가능한 컨텐츠 객체로 분해하는 방법을 제안하고, 객체화한 강의 컨텐츠를 지원하는 시스템을 설계 및 구현하였다. 많은 교육 시스템 벤더들이나 교육 관련 표준화 단체에서 학습 객체를 제안하

였으나 학습 자료(text 및 multimedia material)들을 교육적 개념 없이 단순하게 조합하여 재사용 단위로 제시하였다. 그러나 본 연구에서는 교육 분야의 특성을 고려하여 교수법적 설계 개념을 사용하여 가상 교육 시스템의 코스웨어 의존적인 강의 컨텐츠를 분해하는 방법을 사용하였다. 즉, 여러 가지 교수법적 설계를 고찰하여 재사용 가능한 단위를 분석하고 추출하였는데, 재사용 가능한 단위로는 지도 항목, 평가, 케이스 예제, 문제, 디스커버리, 리소스, 시뮬레이션 등이 가능하며, 이것을 웹 기반 가상 교육 시스템의 강의 컨텐츠 시스템에 적용시켰다.

읽음으로써 사실과 지식을 습득하는 정적 성질의 read LIO로는 개요, 사실 LIO 요소로 구분하였는데 이것은 교수법적 설계의 재사용 단위의 지도 항목에 해당한다. 학습자와 상호작용이 필요한 동적 성질의 try LIO로는 해보기, 퀴즈, 탐구 학습, 토론, 평가 LIO 요소로 구분하였는데 이것은 교수법적 설계의 재사용 단위인 케이스 예제, 디스커버리, 리소스, 시뮬레이션, 테스트에 해당한다.

이러한 교수법적 설계 개념을 지원하는 강의 컨텐츠 시스템이 가지는 장점은 우선, 컨텐츠 저작자 및 교수 설계자에게 교수법적 설계 개념으로 구분된 강의 컨텐츠에 쉽게 접근할 수 있어 학습 활동의 변경으로 인한 강의 컨텐츠 재설계시 적시 적격의 변경이 가능하다. 둘째, 교수법적 개념을 근거로 분해한 방법의 각 LIO 학습 객체는 CBD 방법으로 구현하였다. 이것은 어플리케이션 간의 학습 객체를 쉽게 공유하고 재사용할 수 있어 학습 컨텐츠의 유지보수 및 관리 비용을 절약할 수 있다. 셋째, LIO 학습 객체 및 레슨 구조와 의미를 XML 문서로 저장하여 객체 저장소나 DB에 저장함으로써 분산 교육 시스템간의 자동 검색이 가능하여 학습 활동에 필요한 적격의 학습 컨텐츠를 제공할 수 있다. 마지막으로 학습 컨텐츠를 제공하는 미디어 벤더에 의존하지 않고 학습자는 웹 브라우저만으로 학습 활동과 관련한 연관성을 쉽게 얻으므로 학습 효과는 극대화될 수 있다.

향후 제시된 객체화된 강의 컨텐츠 시스템은 학습자 개인화 서비스를 위해 학습자의 프로필이나 관심사항 정보를 가진 LMS와 연동되어 컨텐츠 관리 시스템에 전달되기 위해 동적인 전달 메커니즘이 필요하다. 또한 각 레슨의 선평가, 후 평가에서 레슨 단위의 학습자별 pass/fail 등의 정보를 저장하고 구현할 수 있는 메커니즘이 필요하며, 자동화된 저작 프로그램의 개발을 위해 교수 설계 원리를 적용시킨 스토리보드 기능과 템플릿을 컨텐츠 저작자에게 제공할 수 있어야 한다. 이러한 것은 향후 교육 웹 서비스를 위한 설계와 구현의 초석으로 이용될 것이다.

참 고 문 헌

[1] 유영만, "e세상 e러닝 : e보양 e삶의 e러닝", 한언, 2002.

[2] 박경환, 문석원, "사용자간 상호작용 지향적 통합 가상교육 시스템의 설계 및 구현", 한국멀티미디어학회논문지, 제1권 제2호, 1998.

[3] Learning Technology Standards Committee, "Draft Standard for Data Model for Content Object Communication," http://ltsc.ieee.org/wg11, 2002.

[4] David A. Wiely, "Connecting Learning Objects to Instructional Design Theory : A Definition, A Metaphor, and A Taxonomy," 2000.

[5] Quinn C., "Learning Objects and Instruction Components," Educational Technology & Society, Vol.3, No.2, 2000.

[6] Albert Ip, Iain Morrison, and Mike Currie, "What Is a Learning Object, Technically?," 2000.

[7] Advanced Distributed Learning Initiative, "The SCORM Overview (Version 1.2)," http://www.ADLnet.org, 2001.

[8] IMS Global Learning Consortium, Inc., "IMS Content Packaging Specification," http://www.imsproject.org/, 2001.

[9] Scott Bergstrom and et al., "Handling Objectives in the AICC CMI Guideline," http://www.aicc.org/, 1998.

[10] Scott Bergstrom and et al., "CMI Guidelines for Interoperability AICC," http://www.aicc.org/, 2001.

[11] Michael Brennan, Susan Funke, and Cusbing Anderson, "The Learning Content Management Systems : A New eLearning Market Segment Emerges," An IDC White Paper, 2001.

[12] Advanced Distributed Learning Initiative, "The SCORM Overview and The SCORM Content Aggregation Model," http://www.ADLnet.org, 2002.

[13] IMS Global Learning Consortium, Inc., "IMS Learning Resource Meta-data Specification," http://www.imsproject.org/, 2001.

[14] IEEE Learning Technology Standards Committee, "Standard for Information Technology - Education and Training Systems - Learning Objects and Metadata," http://itlc.ieee.org/wg12/, 2002.

[15] Albert Ip and Iain Morrison, "Learning Objects in Different Pedagogical Paradigms," 2001.

[16] Deepak Alur, John Crupi, Dan Malks, "Core J2EE Patterns Best Practices and Design Strategies," Sun Microsystems, 2001.

[17] Scott Lee, "Building Framework with Design Patterns Step by Step Guide to Build a Lightweight Framework with Design Patterns," 2001.

[18] James W. Coper, "Java Design Patterns : A Tutorial," Addison-Wesley, 2000.

[19] Mark Grand, "Java Enterprise Design Patterns," John Wesley & Sons, Inc, 2002.

[20] Joseph Schmuller, "Teach Yourself UML in 24 Hours," SAMS, 1999.

신 행 자

e-mail : hjshin@daunet.donga.ac.kr

1991년 동아대학교 컴퓨터공학과(학사)

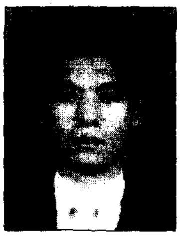
1995년 동아대학교 대학원 컴퓨터공학과 (공학석사)

2000년 동아대학교 대학원 컴퓨터공학과 박사수료

1996년~2000년 동아대학교 시간강사

2000년~현재 동아대학교 기계산업시스템공학부 BK21 교수

관심분야 : XML Technology, Internet Application, Distributed Object System



박 경 환

e-mail : khpark@daunet.donga.ac.kr

1981년 경북대학교 컴퓨터공학과(학사)

1983년 서울대학교 대학원 컴퓨터공학과 (공학석사)

1990년 서울대학교 대학원 컴퓨터공학과 (공학박사)

1984년~1986년 서울대학교 공과대학 시간 강사

1998년 University of California, Irvine 객원교수

1987년~현재 동아대학교 전기전자컴퓨터공학부 교수

관심분야 : 멀티미디어 시스템, 원격 교육, 전자상거래 시스템