DBpia

효율적인 이러닝 시스템 개발을 위한 SCORM API 상호 연결

정회원 정 화 영*, 홍 봉 화**°

The Interconnection of SCORM API for the Efficient E-Learning System Development

Hwayoung Jeong*, Bonghwa Hong** *Regular Member*

요 약

본 연구는 이러닝 시스템 개발을 위하여 SCORM의 각 API를 상호 연결하였다. 이는 학습자가 다양한 학습 콘텐츠를 요구하였을 경우를 생각하여 LMS가 여러 개인 다중 LMS 환경도 고려하였다. SCORM API의 상호연결 방법으로는 ADL의 Wright 방법을 이용하였다. 이를 통해 학습 콘텐츠 프로세스들에 대한 연결 구조를 정형적으로 명세하였다. 또한 다중의 LMS 환경에서도 학습자가 원하는 콘텐츠를 서비스하기 위하여 각 LMS를 연결하는 Communicator를 두었다. 적용결과로서 본 연구는 다중 LMS 환경에서 많은 SCORM API들을 연결할 수 있었고, 연결된 프로세스들을 ADL을 통해 분석함으로서 각 로직들이 정상적으로 수행할 수 있음을 알 수 있었다.

Key Words : 웹 기반 학습 시스템, 상호연결, SCORM API, ADL, Wright

ABSTRACT

In this paper, we had inter-conntected between SCORM APIs for development of e-Learning system. That is, we had considered the environment of multi LMS because it might happen situation that learner was able to request various learning contents to LMS. And we used the ADL based Wright to interconnected between SCORM APIs. By this methods, we were able to make a specification of connection structure between each learning content processes. We made communicator to connect each LMS for service learning contents in multi LMS environment. In this result, we can interconnect a lot of SCORM APIs in multi LMS environment and know each logics can perform the process completely as analysis the connected process by ADL.

Ⅰ. 서 론

실시간 대면교육은 전통 교육에 비해서 공간적인 접근성을 극대화한 교육방법인 반면, 교육 컨텐츠와 학생간의 상호작용이 중심이 되는 WBI는 인터넷을 기반으로 시간과 공간적인 제약을 해소할 수 있다는 점에서 긍정적인 평가를 받고 있다[1]. 이러한 이러닝 콘텐츠 설계의 핵심요소는 내용 설계와 상호작용 설계이다[2]. 또한 이러닝 콘텐츠 제작시의 표준안은 SCORM(Sharable Content Object Reference Model)을 따른다. SCORM 2004 스펙의 핵심은 공유 가능한 컨텐츠 객체를 통해 자원의 재사용 가능성을 높이고, 학습객체를 적절하게 관리하여 수업의 효율성을 높이는데 있다. 또한 SCORM의 학습객체들을 처리하기 위한 각 프로세스의 관리는 이러닝 시스템이 많은 학습객체들을 사용하는 SCORM 기

* 경희대학교 교양학부(hyjeong@khu.ac.kr), ** 경희사이버대학교 정보통신학과(bhhong@khcu.ac.kr), (° : 교신저자)
논문번호 : 08069-1106, 접수일자 : 2008년 11월 6일

www.dbpia.co.kr

반이었을 경우 매우 필요하게 된다. 이는 이러닝 시스템이 다양한 학습 객체들을 요구하는 상황에서. 각 학습객체들의 처리를 효율적으로 관리 및 처리하지 않으면 이러닝 프로세스 운영에 비효율성 및 문제점을 가질 수 있기 때문이다.

본 연구에서는 효율적인 이러닝 시스템의 개발을 위해 다중의 SCORM 기반 학습객체들을 관리 및 처리할 수 있는 학습 프로세스 모형을 제시하고자 한다. 이를 위해 SCORM 인터페이스의 상호연결 및 처리를 담당하는 학습 시스템 서버를 두었으며, 다중 LMS 환경에서 학습자가 요청한 학습 콘텐츠의 처리를 위해 SCORM의 각 학습객체를 연결하는 역할을 수행하고, 학습객체 요구와 응답을 처리한다. 또한 학습객체의 정형적인 연결 및 처리를 위하여 ADL(Architecture Description Language)기반의 Wright[3]를 이용하였다. 이는 정형기법을 통하여 각 프로세스의 분석 및 설계를 분명히 할 수 있으며, 분명하고 효과적인 설계를 구현할 수 있기 때문이다.

Ⅱ. 관련연구

2.1 SCORM

SCORM은 컨텐츠와 LMS 환경간의 상호연동을 정의한 것이며, 특정한 LMS를 구현하는 기능에 대해서는 기술하지 않는다. 그림 1과 같이 LMS는 학습자에게 학습 컨텐츠를 전달하는 방법으로 여러 가지 서비스를 가지고 있는데 무엇을 언제 전달할 것인지를 결정하고(Delivery), 학습 컨텐츠를 통해 학습과정을 추적하는 능력을 가지고 있으며(Tracking), 정의된 규칙에 의해서 학습자에게 전달될 순서가 결정된다[4].

SCORM의 학습객체 인터페이스는 RTE(Run-Time Environment)의 Launch, API, Data Model로 나타내며 그림 2와 같이 나타낸다[5].

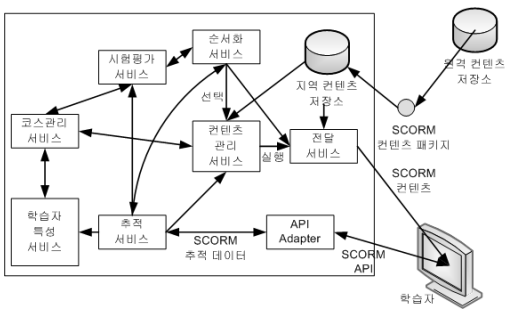

그림 1. SCORM기반의 LMS 구조

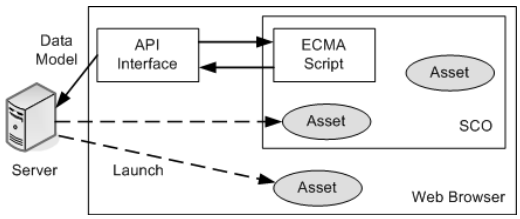

그림 2. SCORM의 RTE 구성요소

그림 3. Wright 인터페이스의 간단한 예

기본 컴포넌트는 콘텐츠 구성의 최소 단위인 asset과 이들의 집합체로 하나의 학습을 할 수 있는 학습객체인 SCO이다. 학습객체는 '학습목표와 내용을 가지고 있는 독립적인 단위로 디지털 기술을 이용하여 만들어진 그 자체로도 학습이 가능한 가장 작은 단위를 말한다. 각 학습객체간의 인터페이스는 공통 API로 표준화되어 있다[6].

2.2 Wright

ADL은 소프트웨어 시스템 모델링 컨셉 구조를 위한 요소들을 제공하는 언어이며, 정형방법으로 소프트웨어 구조를 기술하고 특정 소스 모듈의 세부사항을 구현하는 것 보다 더 높은 단계의 소프트웨어 설계를 표현한다[7]. 그림 2는 ADL의 하나인 Wright[3]의 간단한 예를 나타낸다.

위 구조를 기반으로 Wright 명세를 나타내면 다음과 같다. 이때 request 은 이벤트의 발생을 나타내며, ⊓은 프로세스의 내부선택, □은 프로세스의 외부선택을 나타내고, 프로세스의 §은 종료를 나타낸다.

Style Client-Server
  Component Client
    Port p = $\overline{\text{request}}$ → reply → p⊓§
    Computation = internalCompute →
      $\overline{\text{p.request}}$ → p.reply →
Computation⊓§
  Component Server
    Port p = request → $\overline{\text{reply}}$ → p□§
    Computation = p.request →
      internalCompute → $\overline{\text{p.reply}}$ →
        Computation □ §
  Connector Link
    Role c = $\overline{\text{request}}$ → reply → c⊓§

www.dbpia.co.kr

Role s = request → $\overline{\text{reply}}$ → s□§
Glue = c.request → $\overline{\text{s.request}}$ → **Glue**
□ s.reply → $\overline{\text{c.reply}}$ → **Glue**
□ §
Constraints
∃! s ∈ Component, ∀c ∈ Component : TypeServer(s) ∧ TypeClient(c) ⇒
connected(c,s)
EndStyle
Configuration Simple
Style Client-Server
Instances C : Client ; L : Link ; S : Server
Attachments C.p **as** L.c ; S.p **as** L.s
EndConfiguration

2.3 선행연구

한경섭[4]의 연구에서는 SCORM을 기반으로 학습자의 특성을 반영한 학습 시스템을 구현하였다. 각 학습처리 절차는 교수와 학습자 사이에 학습선호, 평가정보, 반응문맥, 학습특성 등의 요인으로 처리하였다. 그러나 각 학습객체들 사이의 인터페이스 절차나 처리에 대해 고려하지 않고 학습자와 교수자 사이의 학습모듈로만 구성하고 있어 구체적인 학습처리에 대해서는 알 수 없었다. 백영태[8]의 연구에서는 SCORM 인터페이스를 위한 각 API 함수 및 기능적 범주를 자세히 소개하였으나 학습관리시스템의 구현 시 고려하여야 하는 각 학습객체들의 상호연결에 대해서는 설명하지 못하였다. 이세훈[9]의 연구에서는 SCORM을 기반으로 자동화 저작도구를 통해 생성된 명세서를 통해 실행모듈과 웹 서버로 서비스 할 수 있도록 하였다. 이때 각 모듈들은 영역, 개념, 에셋, 실행모듈로 분류하고 이들 특성에 따라 학습 객체들 간의 연결 등을 고려하고 있다. 그러나 학습객체들의 연결에서 정형적인 분석이나 설계 없이 각 기능모델에 대해서만 절차에 따라 연결함으로서 프로세스들의 실행에 대한 효율성을 검증할 수 없었다. 정화영[10]의 연구에서는 SCORM을 기반으로 자기주도적 학습의 교수학습 콘텐츠 모형을 설계하였다. 그러나 자기주도적 학습의 각 학습객체들은 학습모형의 절차에 따라 운용되고 있을뿐 학습객체들 간의 인터페이스는 명시하지 못하였다. 정현숙[11]의 연구에서는 온톨로지와 SCORM의 관계를 통해 교육 콘텐츠 개발전략을 수립하고, 이를 원격대학에 적용하였다. 그러나 각 학습객체들의 인터페이스를 고려하지 않고 온톨로지와 SCORM의 특성만을 고려함으로서 학습객체들의 구체적인 상호작용을 나타내지 못하였다. Kwang-Hoon Kim[12]의 연구에서는 각 학습객체들을 Activity, Role, Repository를 통해 상호 연결하였으며, 이들에 대한 상호연결을 정의하였다. 그러나 학습객체를 처리하는 프로세스 로직에 대한 연결은 고려하지 않았다.

Ⅲ. 학습 시스템 개발을 위한 SCORM API 연결

3.1 SCORM API 연결

SCORM API에는 다음과 같은 함수들이 있어서 이를 통해 각 학습 객체들을 호출하고 응답을 받는다.

```
LMSInitialize();
strFindLocation = objAPI.LMSGetValue("사용자 학습 콘텐츠");
bSuccess = objAPI.LMSSetValue("사용자 학습 콘텐츠", "page");
bSuccess = objAPI.LMSCommit("");
LMSFinish();
```

이를 위한 다중 LMS에서 SCORM을 통한 사용 환경은 그림 4와 같다. 이때 여러 개의 학습 콘텐츠를 지원하는 다중 LMS 환경이 고려되었으며, 학습자의 요청에 따라 학습 시스템 서버에서 해당 LMS에 학습 콘텐츠 서비스를 요청하고 응답을 처리하도록 하였다. 각 LMS에는 해당 학습 내용에 따른 학습

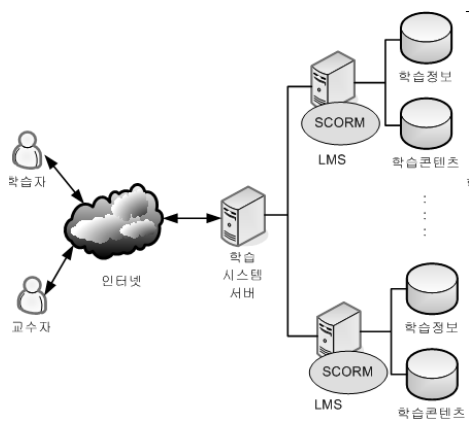

그림 4. 다중 LMS에 연동된 학습 시스템

www.dbpia.co.kr

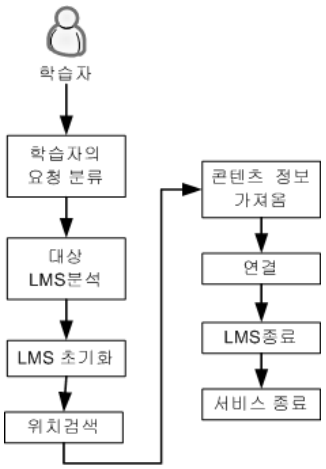

그림 5. 다중 LMS 환경의 SCORM 연결 순서도

콘텐츠 데이터베이스를 각각 가지고 있다. 그림 5는 다중 LMS환경에서 학습자의 요청에 따라 학습 환경을 지원하는 순서도를 나타낸다. 즉 학습자의 학습 콘텐츠 요구가 발생하면 학습자의 요청을 분류하게 된다. 이후 요청된 학습의 분류에 따라 대상 LMS를 분석하고 결정된다. 대상 LMS가 선정되면 해당 LMS를 호출하기 위한 API를 수행하게 된다. LMS 초기화에서는 LMSInitialize()를 통해 수행된다. LMS가 초기화되면 학습 콘텐츠를 찾기 위해 위치검색이 수행되고 objAPI.LMSGetValue()를 호출한다.

콘텐츠 정보 가져옴의 과정에서는 objAPI.LMSSetValue()을 통해 요청된 학습 콘텐츠 정보를 가져오게 된다. 연결의 과정에서는 objAPI.LMSCommit()을 통해 실제 학습 콘텐츠를 가지는 LMS와 연결하게 되어 학습 콘텐츠를 가져올 수 있으며, 이를 학습자에게 제공하게 된다. 해당 학습콘텐츠의 처리가 종료되면 LMSFinish()를 통해 LMS종료를 수행하고 전체 서비스를 종료하게 된다.

3.2 Wright을 이용한 연결 명세

본 연구에서는 학습자가 요청할 수 있는 다양한 학습 콘텐츠의 요구에 응답하고자 다중 LMS 환경에서 고려되었다. 다음은 이러한 다중 학습 콘텐츠 요구 및 처리에 대한 Wright 기반의 정형 명세를 나타낸다.

```
Style E-LearningSystem
  ContentsService LearningService1
    SourceURL = LMS001.hanto.kr□§
    Port p = request → reply → p□§
    Computation =
      FindContents → p.LMSInitialize →
      p.LMSGetValue → p.LMSSetValue →
      p.LMSCommit → Computation□§
      □ FinishContents → p.LMSFinish →
      p.finishService → Computation□§
  ContentsService LearningService2
    SourceURL = LMS002.hanto.kr□§
    Port p = request → reply → p□§
    Computation =
      FindContents → p.LMSInitialize →
      p.LMSGetValue → p.LMSSetValue →
      p.LMSCommit → Computation□§
      □ FinishContents → p.LMSFinish →
      p.finishService → Computation□§
            :                :
            :                :
  ContentsService LearningServicen
    SourceURL = LMS00n.hanto.kr□§
    Port p = request → reply → p□§
    Computation =
      FindContents → p.LMSInitialize →
      p.LMSGetValue → p.LMSSetValue →
      p.LMSCommit → Computation□§
      □ FinishContents → p.LMSFinish →
      p.finishService → Computation□§
  ServiceCommunication Communicator
    Role s1 = request → reply → s1□§
    Role s2 = request → reply → s2□§
            :                :
            :                :
    Role sn = request → reply → sn□§
    Glue = s1.request → s1.getLMSURL →
           s2.request → Glue
      □ s2.reply → s2.getLMSURL →
        s2.connecLMStURL → s1.reply
          → Glue □
            :                :
            :                :
      □ sn.reply → sn.getLMSURL →
        sn.connectLMSURL → sn-1.reply
          → Glue □ §
    Constraints
      ∃! s1 ∈ LearningService1,
```


www.dbpia.co.kr

```
        ∀s2 ∈ LearningService2 :
          :                    :
          :                    :
        ∀sn-1 ∈ LearningServicen-1 :
        ∀sn ∈ LearningServicen :
          TypeServer(s1)  ∧  TypeClient(s2)
          ...TypeServer(sn-1) ∧ TypeClient(sn)
          ⇒ connected(s1,s2) ; ... ;
                    connected(sn-1,sn)
EndStyle
Configuration Communicator
  Style ServiceCommunicator
  Instances S1 : LearningService1 ;
            C : Communicator ;
            S2 : LearningService2 ;
                 :          :
                 :          :
            Sn-1 : LearningServicen-1
           Sn : LearningServicen
  Attachments S1 as C ; C as S2 ; ... ;
            Sn-1 as C ; C as Sn
EndConfiguration
```

각 LMS가 1 ~ n 까지 있을 때 이를 러닝서비스1 ~ n 까지 명시한 뒤, SourceURL에서는 LMS의 위치 정보를 가지게 함으로서 LMS를 분류 및 검색할 수 있도록 하였다. 각 LMS에는 LMS 초기화, 콘넨츠 검색 및 연결, LMS 종료등의 프로세스가 포함된다. ServiceCommunication에서는 각 LMS와의 연동을 준비하였다. 이는 각 LMS의 위치를 getLMSURL을 통해 검색할 수 있도록 하였다. Configuration에서는 ServiceCommunication인 Communicator를 중심으로 각 LMS가 연결되어 서비스 될 수 있도록 하였다.

Ⅳ. 비교 평가

본 연구는 SCORM API를 연결함으로서 학습자가 다중의 학습콘텐츠를 요청시에 이를 효율적으로 관리 및 처리할 수 있도록 설계하였다. 이에 대한 결과로서 기존연구와의 비교를 나타내면 표 1과 같다. 이를 통해 기존의 연구에서도 모두 LMS와의 연동이나 SCORM을 사용하였으나, SCORM을 이용한 학습콘텐츠 처리를 담당하는 API 프로세스들에 대한 처리에 대해서는 충분히 고려가 되지 않은

표 1. 기존연구와의 비교

비교항목	백영태[8]	이세훈[9]	정화영[10]	제안 기법
학습 시스템에서 LMS 연동	○	○	○	○
학습 시스템에서 SCOM 적용	○	○	○	○
SCORM API의 프로세스의 기능적 연결 분석 및 설계	×	×	×	○
학습 콘텐츠 처리의 정형적 분석	×	×	×	○
다중 LMS 환경 고려	×	×	×	○

것을 알 수 있다. 이는 학습자의 학습 콘텐츠 요청 및 처리를 수행하는 중요한 역할을 수행하기 때문에 원활하고 효율적인 학습 콘텐츠 처리를 위해서는 각 프로세스의 연결 및 분석이 필요하다. 본 연구에서는 이를 위하여 ADL 기법에서 Wright를 사용함으로서 각 프로세스의 연결을 정형화하였고 다중 LMS 환경도 고려하였다.

Ⅴ. 결 론

본 연구는 효율적인 이러닝 시스템 개발을 위하여 SCORM API를 상호연결 하였으며, 정형화된 연결 명세를 위하여 ADL중의 하나인 Wright를 이용하였다. 이를 통해 각 학습 콘텐츠 처리를 담당하는 SCORM API 함수들을 LMS의 명세에 포함하였고, 이를 ServiceCommunication인 Communicator를 통해 상호 연결함으로서 LMS가 여러 개인 다중 환경에서도 학습자가 원하는 학습 콘텐츠를 효율적으로 처리할 수 있도록 하였다. 각 LMS에는 각각의 학습 콘텐츠를 요청 및 응답처리를 위하여 LMS 초기화와 학습 콘텐츠 요청, 연결, 정보를 가져옴 등을 수행하고 모든 처리가 종료되면 LMS 종료의 프로세스까지 포함하였다. Communicator의 역할에서는 각 LMS들이 가지는 SCORM API의 프로세스들을 연결함으로서 다중 LMS 환경에서 학습자가 원하는 학습 콘텐츠를 검색하고 요청 및 응답 처리할 수 있도록 상호 연결하였다.

향후 연구과제로는 Wright를 통해 분석 및 명세한 각 프로세스들을 실제 구현 및 개발하고 이를 실제 적용함으로서 실증에 대한 검증과정이 필요하다.

참 고 문 헌

〔1〕 최상균, “컴포넌트 기반 개발을 이용한

www.dbpia.co.kr

LCMS 기반의 e-Learning 시스템 개발", 한국전자거래학회지 제9권 제1호, 2003.
〔2〕 정은정, "교수자-학습자의 상호작용 촉진을 위한 e러닝 수업모형 연구",
〔3〕 Robert Allen, Rémi Douence, David Garlan, "Specifying and Analyzing Dynamic Software Architectures", Proceedings of 1998 Conference on Fundamental Approaches to Software Engineering, 1998.
〔4〕 한경섭, 서정만, 정순기, "SCORM 기반의 적응형 학습관리 시스템의 설계 및 구현", 한국컴퓨터정보학회 논문지 제9권 제3호, 2004.
〔5〕 Sharable Content Object Reference Model(SCORM) Version 2004, The SCORM Overview, Advanced Distributed Learning http://www.adlnet.org
〔6〕 장재경, 김선혜, 김호성, "SCORM 기반 동영상 콘텐츠의 재사용 전략", 한국콘텐츠학회논문지 Vol.8 No.1, 2008.
〔7〕 Zhenhua Yu and Yuanli Cai, Object-Oriented Petri nets Based Architecture Description Language for Multi-agent Systems, International Journal of Computer Science and Network Security, VOL.6 No.1B, 2006.
〔8〕 백영태, 이세훈, "SCORM 지원 공개 소프트웨어 학습 관린 시스템", 한국컴퓨터정보학회 학회지 제14권 제1호, 2006.
〔9〕 이세훈, 서대우, 왕창종, "EduCODE : ADL SCORM 자동 생성을 위한 콘텐츠 개발 지원 시스템", e-Learning 학술연구 제1권 제1호, 2002.
〔10〕 정화영, 홍봉화, "자기주도적 학습을 위한 학습자 지향의 교수학습 콘텐츠 모형", 한국통신학회논문지 제33권 제6호(통신산업응용), 2008.
〔11〕 정현숙, "온톨로지 기반의 교육 콘텐츠 제작 기법", 한국콘텐츠학회논문지 제5권 제2호, 2005.
〔12〕 Kwang-Hoon Kim, Hyun-Ah Kim, and Chang-Min Kim, "SCO Control Net for the Process-Driven SCORM Content Aggregation Model", LNCS 3483, 2005.

정 화 영 (Hwa-Young Jeong) 정회원

1994년 2월 경희대학교 전자계산공학과 공학석사
2004년 8월 경희대학교 전자계산공학과 공학박사
2000년 3월~2005년 2월 예원예술대학교 만화게임영상학부 조교수
2004년 5월~현재 (사)한국인터넷정보학회 논문지 편집위원
2005년 3월~현재 경희대학교 교양학부 조교수
<관심분야> 소프트웨어공학, CBD, 교육공학

홍 봉 화 (Bong-Hwa Hong) 정회원

1983년 3월 경희대학교 전자공학과 공학사
1992년 8월 경희대학교 전자공학과 공학석사
2001년 8월 경희대학교 전자공학과 공학박사
2002년 9월~2004년 2월 세명대학교 컴퓨터수리정보학과 조교수
2005년 5월~현재 경희사이버대학교 정보통신학과 부교수
<관심분야> 전자공학, 정보통신공학, 방송통신공학

www.dbpia.co.kr