

Ajax 기반의 SCORM 2004 시퀀싱 엔진 및 데이터 모델의 설계 및 구현

홍철기*, 장상현**, 황종선*
*고려대학교 디지털정보공학과
**동국대학교 컴퓨터공학과

e-mail : henricus@keris.or.kr, shjang@keris.or.kr, hwang@disys.korea.ac.kr

A Design and Implementation of a SCORM 2004 Sequencing Engine and Data Model Based on Ajax

Chul-Ki Hong*, Sang-Hyun Jang**, Chong-Sun Hwang*
*Dept. of Digital Information Engineering, Korea University
**Dept. of Computer Engineering, Dong-Guk University

요 약

본 논문에서는 기존 학습 관리 시스템(Learning Management System : LMS)의 JAVA Applet 으로 구현된 SCORM 2004 시퀀싱 엔진 및 데이터 모델에서 나타난 문제점을 해결할 수 있는 Ajax(Asynchronous JavaScript and Xml) 기반의 SCORM 2004 시퀀싱 엔진 및 데이터 모델을 제안한다. 기존 JAVA Applet 으로 구현된 시퀀싱 엔진 및 데이터 모델에서의 VM(Virtual Machine) 구동상에 발생하는 보안 및 인증 문제, VM 구동에 따른 제한점 및 시스템 처리속도의 문제점을 해결하였다.

1. 서론

최근 Ajax(Asynchronous JavaScript and Xml) 기술을 도입한 연구 및 적용사례가 각 분야에서 활발히 진행되고 있으며, Ajax 기술의 빠른 처리속도, 표준, 확장성, 호환성, 개방성 등의 장점 때문에 웹 어플리케이션의 새로운 모델로 부각되고 있다.

이러한 Ajax 기술은 e-Learning 환경에서도 요구되고 있으나, 아직까지 Ajax 기술을 접목한 e-Learning 연구 및 적용 사례가 거의 없다. e-Learning 분야의 표준안인 SCORM(Sharable Content Object Reference Model)[1]은 여러 다양한 표준화 그룹에서 개발하고 제안된 기술적인 요소들과 다른 스펙들을 모두 수집, 통합하여 하나의 구현 가능한 모델로 개발한 ADL(Advanced Distributed Learning)의 참조모델이다.[2] 현재 e-Learning 분야에서 기존 교육기관 및 전문업체의 학습관리 시스템(Learning Management System : LMS)이나, ADL의 sample LMS 인 SCORM SampleRTE 에서는 대부분 Java Applet 기반의 SCORM 시퀀싱 엔진 및 데이터 모델 엔진을 사용하고 있으나, Java VM(Virtual Machine)

구동상에 발생하는 보안 및 인증 문제와 VM 사용에 따른 제약, 시스템 부하 등의 문제점이 발생한다.

이러한 문제점을 해결하고자 Ajax 어플리케이션 기술을 활용한 SCORM 2004 시퀀싱 엔진 및 데이터 모델 엔진의 설계 및 구현 방식을 제안한다.

본 연구에서 제안하는 Ajax 기반의 SCORM 2004 시퀀싱 및 데이터 모델 엔진 구현을 통해 Java Applet 기반에서 나타나는 VM 문제점 및 시스템 부하 문제를 해결하였다.

본 논문의 구성은 다음과 같다. 2 장에서는 Ajax 의 관련 기술 및 적용사례를 살펴보고, 3 장에서는 Java Applet 기반에서의 SCORM 2004 시퀀싱 및 데이터 모델 엔진의 문제점을 제시하였다. 4 장에서는 이러한 문제점을 해결할 수 있는 방안으로 Ajax 기술을 활용한 학습관리 시스템에서의 SCORM 2004 시퀀싱 엔진과 데이터 모델 엔진을 제안한다. 5 장에서는 본 논문에서 제안한 방식의 분석과 기대효과, 향후 연구과제를 제시한다.

2. Ajax

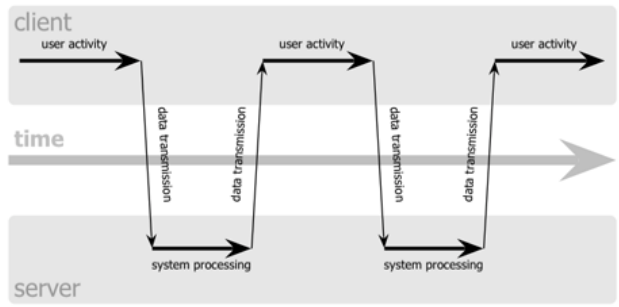
Ajax 는 Asynchronous JavaScript and XML 의 약자이며 RIA(Rich Internet Application : Lazlo application, Flex, JSF) 같이 웹에서 어플리케이션과 같은 풍부하고 동적인 웹 어플리케이션을 제작하기 위한 요구에서 시작되었다. Ajax 의 특징은 다음과 같다.

- XHTML, CSS 를 이용한 표준기반의 프리젠테이션
- DOM(Document Object Model)을 이용한 동적 표시 및 상호작용
- XML 과 XSLT 를 이용한 데이터 교환 및 제어
- XMLHttpRequest 를 이용한 비동기식 데이터 처리
- 상기 모든 기술을 결합해주는 자바스크립트[3]

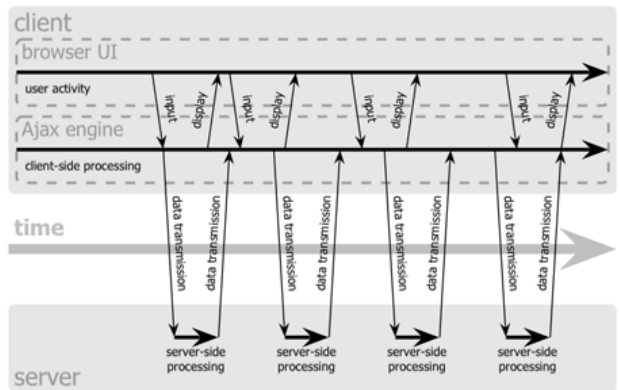
기존의 웹 어플리케이션은 클라이언트가 전송하고자 하는 내용을 웹 서버 측에 요청하고 웹 서버 측은 클라이언트 측이 요청한 내용에 따라서 새로운 웹 페이지를 작성하여 클라이언트 측에 되돌려준다. 이 때 최초의 클라이언트 측이 가지고 있는 내용과 서버가 되돌려 주는 두 내용 사이에 중복되는 HTML 코드로 인해 많은 대역폭을 낭비하게 된다. 대역폭의 낭비는 단순히 회선의 낭비를 넘어서서 많은 초과 투자를 받게 된다. 게다가 이러한 방식으로는 네이티브 어플리케이션과 비교할 때 복잡한 대화형 사용자 인터페이스를 작성하기가 어렵게 된다.

반면에 Ajax 어플리케이션은 클라이언트 측에 필요한 데이터만을 주도록 웹 서버 측에 요청할 수 있다. 보통 SOAP 이나 XML 기반의 웹 서비스 프로토콜을 사용하며, 웹 서버의 응답을 처리하기 위해 클라이언트 측에서 자바스크립트를 쓴다. 그 결과로 클라이언트의 웹 브라우저와 웹 서버 사이의 교환되는 데이터량이 줄어들기 때문에 어플리케이션의 응답성이 좋아진다. 요청을 주는 수많은 컴퓨터에서 이 같은 일이 일어나기 때문에, 전체적인 웹 서버 처리량도 줄어들게 된다.

classic web application model (synchronous)

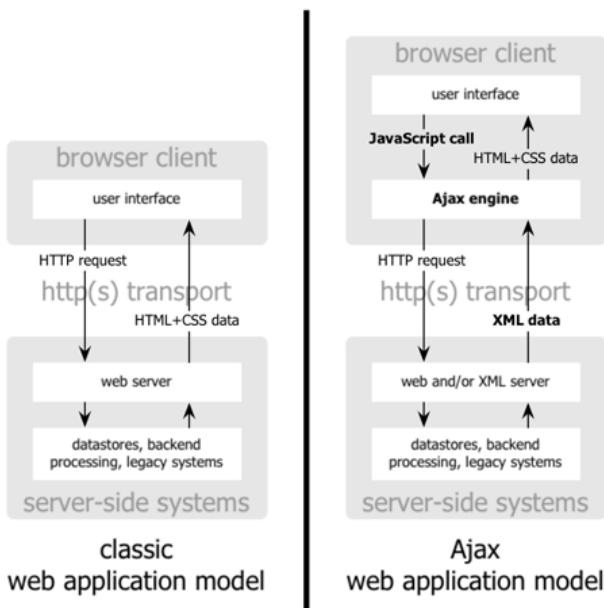


Ajax web application model (asynchronous)

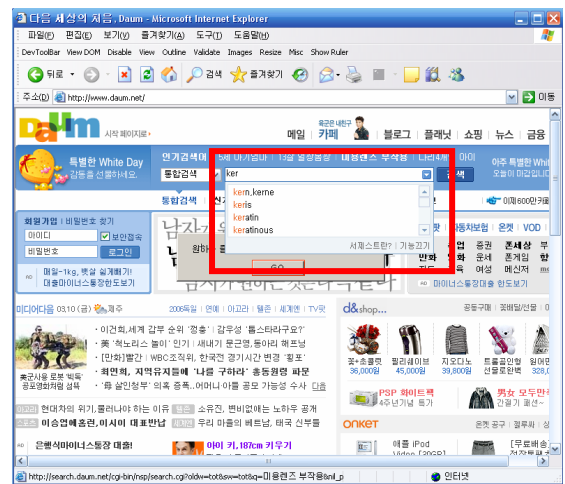


(그림 2) 전통적인 웹 어플리케이션(상)과 Ajax 어플리케이션(하)의 통신 방법 비교

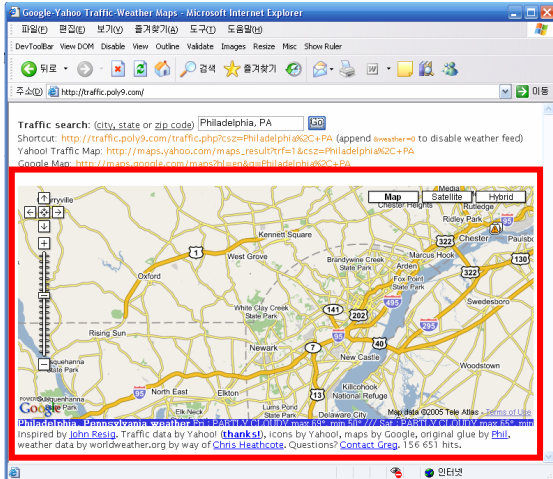
국내에서 Ajax 를 사용하는 웹 사이트로는 Daum, Naver, Empas 등이며 검색단어 서비스 제공과 지도 검색 서비스에서 Ajax 를 사용하고 있으며, 해외에서는 Google, Yahoo, Amazon 등에서 검색단어 서비스 제공과, 블로그, 메일, 지도 검색, 상품검색 등 다양한 분야에서 사용하고 있다.



(그림 1) 전통적인 웹 어플리케이션 모델(좌)과 Ajax 웹 어플리케이션 모델(우) 비교



(그림 3) Daum 의 검색서비스



(그림 4) Google의 지도서비스

3. Java Applet 기반의 SCORM 2004 시퀀싱 엔진 및 데이터 모델의 문제점

Java Applet은 Java VM으로 구동되는 Java Web Application이다. Java Applet 기반의 SCORM 2004 시퀀싱 및 데이터 모델 엔진에서의 문제점은 다음과 같다.

첫째, 대부분 Applet을 사용해야 할 구현 부분에는 보안 인증이 필요하다. 운영체제가 Windows XP SP2 버전 또는 그 이상이라면 공인 인증기관의 인증서가 없는 Applet 사용에 제한이 생긴다.

둘째, Windows XP의 IE(Internet Explorer)는 자체적으로 VM을 내장하고 있지 않기 때문에, 개인 사용자 또는 S/W 제공자가 VM을 설치하거나, VM 설치 안내문을 별도로 배포, 안내해야 한다.

셋째, Sun Java VM이 IE에 Plug-in되어 기본 VM으로 설정되면 MS VM 인증 메커니즘으로 구현된 Applet은 다운로드할 수가 없다. 기본 VM 설정이 MS VM인 경우도 마찬가지이다.

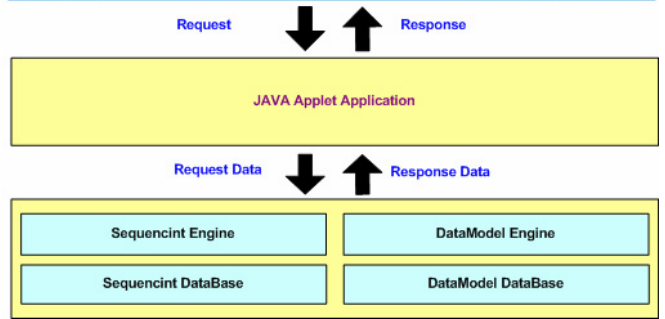
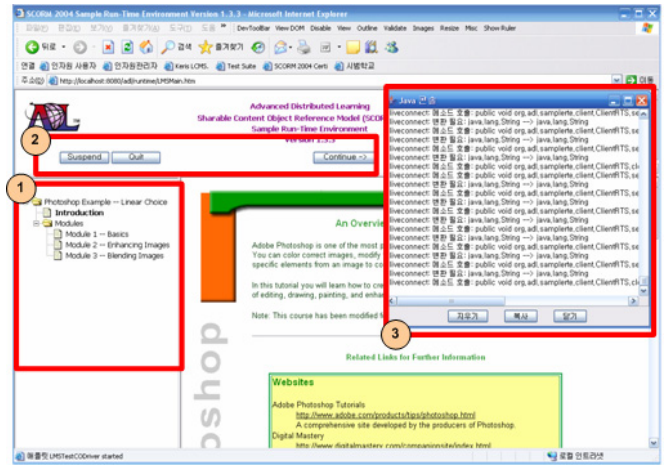
넷째, VM 패치에 따른 일부 어플리케이션의 기능 제약 및 주기적인 패치에 따른 시스템 변경 작업이 불가피하다.

마지막으로 Java Applet은 VM을 구동하는 방식으로 클라이언트 측에 시간적인 부하를 일으키며, 통신 어플리케이션일 경우는 클라이언트-서버 통신에 대한 시스템 부하를 일으키는 문제점이 발생한다.

4. Ajax 기반의 SCORM 2004 시퀀싱 엔진 및 데이터 모델 제안

JAVA Applet를 통한 클라이언트와 서버의 통신 프로세스는 그림 5와 같다. 그림 5는 ADL에서 제공하는 RTE(Run-Time Environment)의 화면 모습이다. RTE에서는 사용자의 요구에 의한 서버의 응답을 수용하기 위해서는 모든 자원이 다시 로드되는 과정을 거친다. 이런 과정을 통해 클라이언트 및 서버는 많은 대역폭을 낭비하게 된다. 또한 로드되는 과정에서 ③과 같은 VM의 응답을 기다려야 함으로써 클라이언트의 부하가 생기게 되고 클라이언트와 서버의 통

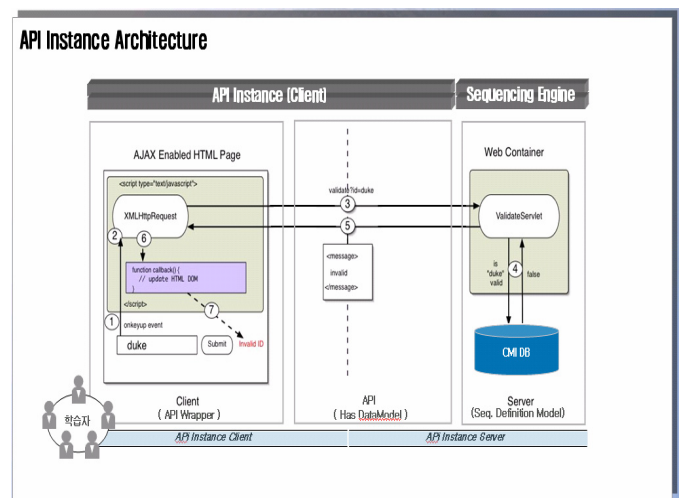
신 과정에서 많은 사용자들이 응답을 요청할 경우 서버의 부하가 많아지게 된다.



(그림 5) JAVA Applet 기반 클라이언트와 서버 통신

이러한 단점을 Ajax를 이용함으로써 VM의 보안, 인증 및 시스템의 성능 개선이 이루어졌다.

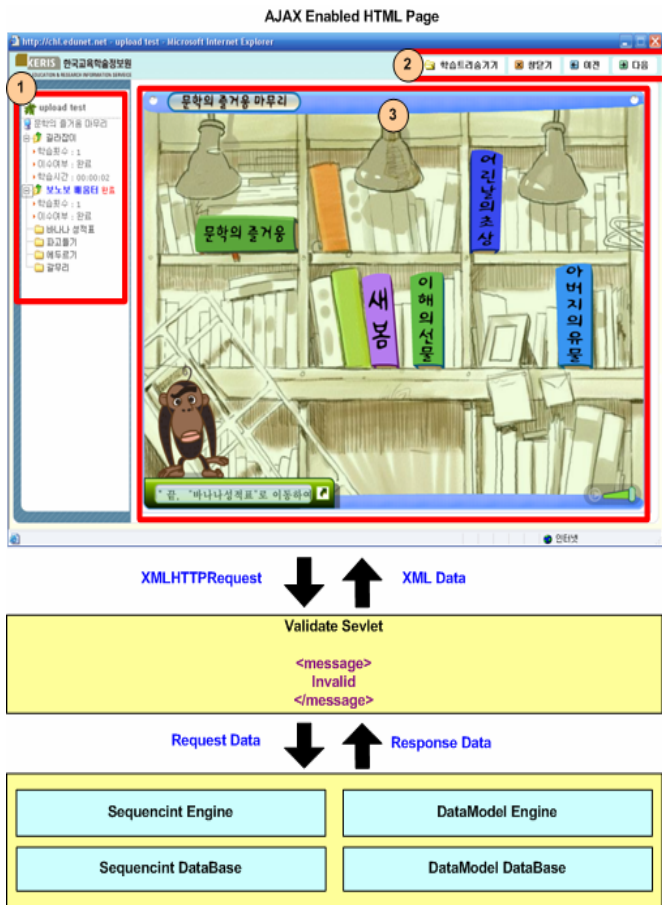
그림 6은 Ajax 기술의 XMLHttpRequest를 이용하여 API(Application Program Interface) 및 API Instance를 통해 실행, view, 서버통신이 이루어지는 전체적인 아키텍처 화면이다.



(그림 6) Ajax를 이용한 SCORM 2004 API Instance Architecture

Ajax로 개발되어진 시퀀싱 및 데이터 모델 엔진은 사용자의 요구에 의한 수많은 데이터를 서버에서 엔

진을 통해 프로세싱 되어 데이터베이스에 저장되고 저장된 데이터를 클라이언트 측에 전달하는 일련의 모든 프로세싱이 이루어진다. 이런 점에서 볼 때 Ajax 를 통해 개발되어진 시퀀싱 및 데이터 모델 엔진은 Ajax 의 목적에 맞는 RIA(Rich Internet Application)화 되었다고 말할 수 있다. 아래의 그림은 각 엔진에서 사용하고 있는 Ajax 를 통한 클라이언트와 서버의 통신의 프로세스이다.



(그림 7) Ajax 기반 클라이언트와 서버 통신

①은 시퀀싱 엔진을 통해서 얻어온 ActivityTree 이고 Ajax Object 를 통하여 ActivityTree 의 정보를 가지고 온다. ActivityTree 의 정보는 Activity 이름, 학습 횟수, 이수여부, 학습시간 등의 정보를 사용자에게 보여준다. ①의 이벤트를 통해서 서버에서 가져오는 응답값은 Ajax 를 통해 모든 프로세싱이 이루어지고 변경된 모든 사용자의 반응값이 보여진다.

②는 시퀀싱 엔진을 통해서 얻어온 버튼 활성화 유무의 Object 정보를 얻어온다. 학습 버튼에 대한 모든 정보들은 SCORM 2004 콘텐츠 구조정보 파일인 manifest.xml 의 <adlnav:hideLMSUI>에 정의되어져 있다. ②의 버튼 이벤트를 통해서 서버에서 가져오는 응답값은 Ajax 를 통해 모든 프로세싱이 이루어지고 시퀀싱의 응답값에 의한 반응값이 보여진다.

③은 ①과 ②를 통한 시퀀싱 엔진의 응답으로 해당 SCO(Sharable Content Object)에 대한 실제적인 파일의 경로를 가지고 온다. 가지고 온 파일의 경로를 통해

변경된 SCO 의 내용이 보여지고 SCO 는 SCORM 2004 API 를 통해 LMS 와 통신하게 된다. 이 때 데이터 모델 엔진과 통신하는 모든 프로세싱은 Ajax 를 통해 이루어지고 실시간으로 데이터 모델 엔진의 데이터들이 데이터베이스에 저장 된다. 저장된 데이터가 시퀀싱 엔진에 영향을 미칠 경우 데이터 모델 엔진은 시퀀싱 엔진과의 상호작용을 통해 프로세싱에 맞는 시퀀싱을 처리한다.

이로써 Java Applet 을 사용하였던 기존 방식과는 달리, 순수 XMLHttpRequest 기반으로 API Instance 를 통해 학습관리 시스템(LMS)의 학습창을 그림 8 과 같이 구현하여, 처리속도 개선 및 Applet 기반에서 발생하는 문제점을 해결하였다.



(그림 8) Ajax 기반 SCORM 2004 API Instance Client (LMS Viewer)

5. 결론

본 연구에서는 Java Applet 기반 SCORM 시퀀싱 및 데이터 모델 엔진을 내재한 학습관리 시스템에서의 문제점을 해결하기 위해서, 그 중요성과 적용범위가 점차 증가하고 있는 Ajax 기술을 기반으로 한 SCORM 2004 시퀀싱 및 데이터 모델 엔진을 제안하였다. 지금까지 연구 및 제안한 결론은 기존 Java Applet, ActiveX 등의 웹 어플리케이션의 모델보다 Ajax 를 통한 시퀀싱, 데이터 모델 엔진과의 통신이 클라이언트 및 서버측에 가장 안정적이었으며, 이로 인한 시스템의 부하 및 안정성이 보다 향상되었다. 앞으로 이러한 Ajax 기술을 이용한 e-Learning 표준화 연구 및 e-Learning 솔루션과 시스템 개발이 요구된다.

참고문헌

- [1] ADL Initiative, SCORM 2004 Overview, 2004.
- [2] 홍지영. “학습객체 기반 개별화 학습 모형 연구”, 박사학위 논문, pp.26, 2004.
- [3] Jesse James Garrett. “Ajax: A New Approach to Web Applications”, Essay, 2005.