



XML 기반 문제은행 시스템 구현

양 광 모*, 최 성**

● 목 차 ●

1. 서 론
2. 문제은행 시스템 개요
3. XML기반 문제은행 시스템 구축
4. 결 론

1. 서 론

국내 교육시장 규모는 굳이 강조할 필요가 없을 만큼 크게 성장해 있다. 인터넷이 발전함에 따라 기존의 오프라인 서비스에서 온라인 학습 서비스로 확장되고 있는 추세이다. 따라서 대부분의 오프라인 출판사나 관련 업체들은 온라인 서비스를 구축중이거나 구축을 계획하고 있다. 하지만 대부분의 문제은행 솔루션은 HTML로 제작되거나 RDB기반의 데이터 형태로 구축함으로써 기존의 출판물을 재활용하지 못해 온라인과 오프라인이 유기적으로 연동되지 못하는 문제점을 안고 있다. 따라서 출판물을 재사용해 온라인과 오프라인을 유기적으로 통합하는 솔루션은 교육 출판 시장에서 필수적이다. 학습자료를 XML 데이터 형식으로 저장, 관리해 학습자의 성적이나 일정에 맞는 맞춤형 서비스가 가능해 지는 것이다. 현재 (주)넥스텍은 lya.com 학습포탈 사이트, 에듀토피아 맞춤 학습서비스, Wislearn 문제은행 솔루션에 XML기반 문제은행 시스템을 적용 시켰다.

2. 문제은행 시스템 개요

2.1 기존 문제은행 시스템의 한계

온라인 문제은행 시스템은 사용자의 수준과 난이도에 맞는 문제를 온라인에서 제공하고 자신이 틀린 문제에 대한 평가와 점수를 실시간으로 서비스하는 기능이 필요하다. 또한 문제와 관련된 학습 자료를 제공하는 기능도 필요하다. 이 시스템 개발은 오프라인에서 가지고 있는 콘텐츠를 어떻게 효과적으로 사용자에게 서비스하느냐에 맞추어 개발상의 모든 Business Logic이 만들어진다.

2.1.1 기존 학습 시스템

기존의 Web 기반 온라인 문제은행 솔루션은 이미지 형태로 시험지를 스캔해 제공하거나, HTML로 시험지를 작성해 해당 문제를 제공하고 있다. 이러한 시스템은 매우 정적인 시스템으로 다음과 같은 기능적 한계를 갖고 있다.

- 1) 사용자 수준에 맞는 문제 서비스가 어렵다.
- 2) 문제 조합을 통한 다양한 문제지 생성이 어렵다.
- 3) 기존 출판물의 재활용이 어렵다.
- 4) HTML의 한계로 인한 정보 손실이 많다.

문제 작성과정에서도 출판물을 보고 HTML로

* (주)넥스텍 연구개발부 주임연구원

** 남서울대학교 컴퓨터학과 교수

코딩해야 하기 때문에 모든 콘텐츠를 변화하는 데 상당한 시간이 필요하다. 또한 새로운 문제가 발생할 때마다 다시 HTML로 시험지를 작성해 서비스할 수밖에 없다. 관리 측면에서도 파일 시스템으로 관리함에 따라 데이터의 양이 늘어날수록 관리와 해당 문제에 대한 검색이 어려워진다.

2.2 XML 선택

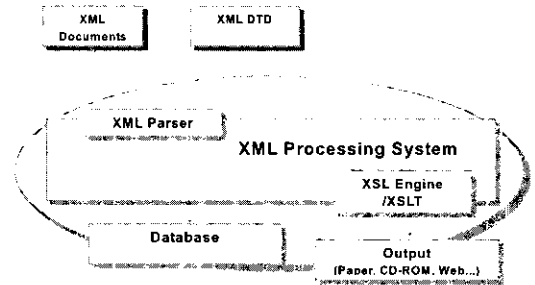
문제은행 시스템을 개발 할 당시 고객의 요구는 기존의 출판업체가 가지고 있는 콘텐츠를 그대로 사용하고, 사용자 수준에 맞게 임의로 문제를 조합해 실시간으로 서비스 할 수 있어야 하는 것이었다. 기존 출판업체들 대부분은 출판을 위해 매킨토시용 워드프로세서인 퀴크 익스프레스(Quark Express)를 사용해 책을 디자인하고 해당 콘텐츠를 작성한다. 이러한 기존의 Quark 파일을 변환해서 웹으로 서비스하면 변환을 위한 문서 작성에 드는 시간과 비용을 절감할 수 있다.

문제가 가지고 있는 고유한 속성(학년, 학기, 단원, 영역 등)을 데이터로 유지하고 해당 콘텐츠를 데이터베이스화 할 수 있어야 한다. 웹으로 서비스하기 위한 스타일링을 자유롭게 적용해야 하면 사용자 검색 조건에 따라 자유롭게 문제를 조합(Assemble)해야 한다. 이러한 요구 조건을 만족시키기 위해서는 데이터 타입을 XML로 가져가는 것이 최상이었다.

2.3 XML 시스템 구성

일반적인 XML시스템 구성은 (그림 1)과 같다. XML 시스템은 다음과 같이 크게 4가지 부분으로 나뉘 볼 수 있다.

- 1) XML Document Authoring / Parsing
- 2) XML Content Management (저장관리)
- 3) XML Information Retrieval (검색)
- 4) XML Presentation (웹,CD-ROM,종이 등)



(그림 1) XML 시스템 구성

먼저 Document Authoring/Parsing 은 문서를 생성하고 해당 문서의 Validation을 체크하는 부분이다. 문서 생성은 여러 가지 방법이 있는데 XML 전용 에디터나 To-XML 변화 프로그램을 통해 생성할 수 있다. Form 형태로 일정 규칙에 맞춰 만들 수 있다. 문서의 검증은 XML 파서를 통해 이뤄지며, 해당 문서와 DTD(Document Type Definition)를 비교하여 문서의 구조가 DTD를 따랐는지 검증하게 된다. XML 데이터를 데이터베이스화하고 관리하기 위해서는 XML Contents 관리를 필요로 한다. 프리젠테이션 단계에서 XSL(Extensible Stylesheet Language)를 이용하여 문서의 스타일링을 할 수 있다. XML 검색을 위해선 구조적 검색이 가능한 검색 엔진을 사용해야 한다.

XML구조 검색 시스템의 특징

- 1) 기존 Full-Text 검색 기능을 제공
 - 형태소 분석기를 통한 한글, 영문 형태소 분석 필요
 - Document에 대한 Index추출 및 관리
- 2) 엘리먼트 단위의 구조 검색 기능 제공
 - 엘리먼트의 트리 구조에 의한 검색
- 3) 속성 검색 기능 제공
 - XML 속성에 의한 문서 검색
- 4) 혼합 검색 기능 제공
 - TEXT, 엘리먼트, 속성에 대한 혼합 검색

3. XML 기반 문제 은행 시스템 구축

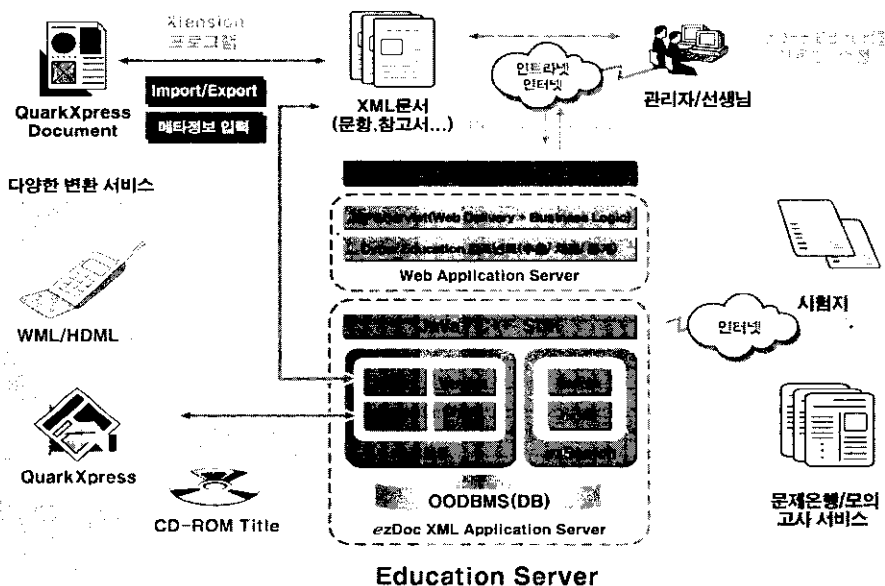
3.1 시스템 구성

본 시스템 개발은 세 개의 사이트를 대상으로 진행되었으며 각각의 고객사는 기존에 오프라인으로 문제지를 서비스하던 출판회사였다. 서비스 대상은 초,중,고 학생을 대상으로 하였으며, 출판사에서 출판물을 위해 작성한 Quark 데이터 파일을 XML로 변환해 데이터베이스로 구축하고, 이를 각 사의 서비스 시나리오에 맞게 서비스 하는 것이었다. 다음은 XML 문제은행 시스템의 구성과 전체 시스템의 구성도(그림 2)이다.

- 1) 시스템 운영체제 : Unix Solaris 2.7
- 2) 구현 언어 : Java
- 3) 데이터베이스 : O2 DBMS 5.1
- 4) 웹서버 : Apache 1.3.2
- 5) JSP/서블릿 엔진 : Gemstone J 4.1 or Tomcat 3.1
- 6) XSLT(XSL Transformation) 프로세서 : Xalan
- 7) XML CMS(Content Management System) : ezDoc XML Application Server

XML CMS(Content Management System)은 XML 문서를 저장 관리하기 위한 Repository Manager를 말한다. DOM이나 SAX를 이용해 시스템을 개발할 경우 직접 XML 데이터를 저장하고 핸들링하는 프로그램을 개발자가 개발해야 한다. CMS는 이런 저장과 데이터 핸들링에 관한 일련의 과정을 처리할 수 있는 다양한 기능을 제공한다. XML 문서는 루트 엘리먼트를 최상위로 하는 트리 구조를 갖는다. 이로 인해 XML을 구조적 언어라고 한다. XML의 구조적 특성을 이용한 검색과 버전 관리를 위해서는 문서를 트리 구조로 저장, 관리하는 것이 가장 효과적이다. 따라서 대부분의 CMS는 객체지향 데이터베이스를 스토리지로 사용하고 있다. CMS는 데이터베이스에 매우 의존적이기 때문에 어떤 스토리지를 사용하느냐에 따라 구현 방식이 매우 달라지게 된다. 일반적인 CMS의 기능은 다음과 같다.

- 1) XML 파서 내장
- 2) Blob 데이터(외부 엔티티) 저장관리
- 3) 개발자를 위한 API 제공
- 4) XML 문서의 색인과 검색을 위한 구조 검색



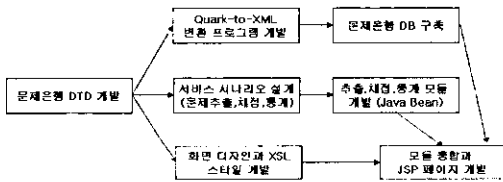
(그림 2) 문제은행 시스템 구성도

시스템과의 연동 기능

- 5) 사용자 권한에 의한 문서 생성, 삭제, 열람
- 6) 문서 Import/Export 인터페이스
- 7) 문서 Check-In / Check-Out 인터페이스 (문서의 버전관리)

3.2 시스템 개발

문제은행 시스템 개발은 다음과 같은 순서로 진행되었다. 데이터 구조 정의를 위해 3종의 DTD를 개발하고 DTD에 맞춰 Quark 파일을 XML 형식으로 변환하는 프로그램과 서비스 시나리오 설계, 화면 디자인과 XSL 스타일 개발 순으로 이뤄졌다. (그림 3)은 시스템 개발 순서이다.



(그림 3) 시스템 개발 프로세스

1) 문제은행 DTD 개발

문제은행 DTD는 각 과목별 문제집을 분석해 공통으로 적용 가능한 엘리먼트 구조를 추출했다. 문제를 검색하기 위한 Meta Data는 속성값으로 처리했는데 문제를 검색하는 조건은 학년, 학기, 단원, 영역, 문제 고유 ID 등이 검색 조건이 될 수 있다. 실제 스타일에서 처리를 위한 데이터도 속성으로 처리하였다. 예를 들면 문제의 박스 처리나 정답 빈칸 수 등에 대한 정보들도 DTD 작성 시에 고려해야 한다.

2) Quark 데이터를 XML로 변환

개발된 DTD를 기본으로 Quark 파일을 XML로 변환하는 프로그램을 만들게 되었다. Quark으로 작

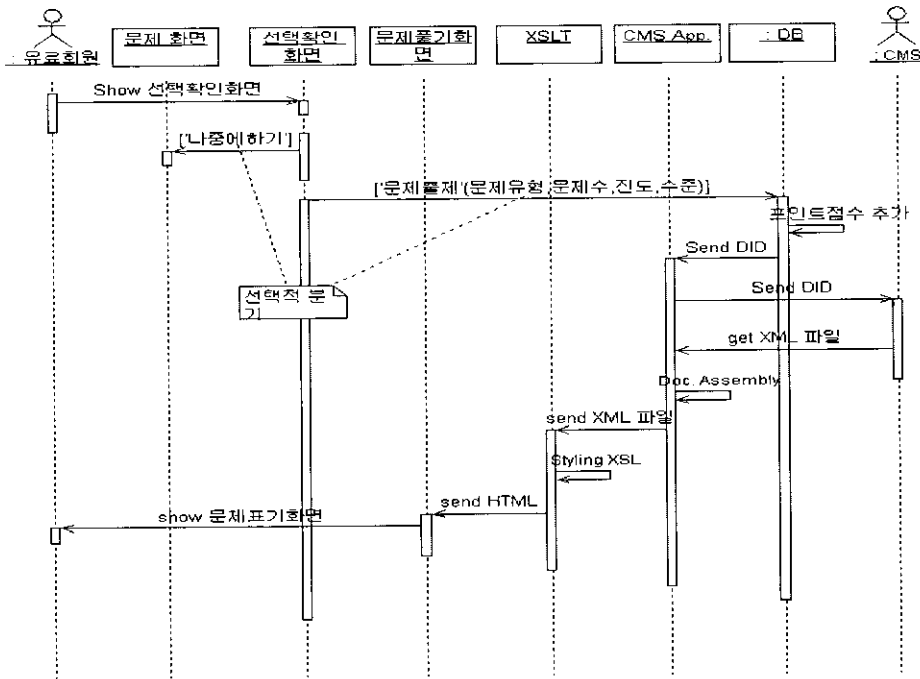
성된 문서를 XML로 변환하기 위해서는 Quark 데이터에 대한 사전 작업이 필요하다. 문항 DTD는 XML 문제를 만들기 위한 작성 규칙이 된다. Quark 파일은 일정 규칙으로 작성된 것이 아니므로 XML 파일로 변환하기 위해서는 Quark파일 내부 객체들을 일정한 형식으로 편집해 줄 필요가 있다. 이를 사전 작업이라 한다. Quark 파일을 정리한 뒤 Quark 문서의 각 객체에 DTD의 해당 태그들을 붙여 XML 파일로 변환하는 프로그램을 개발하였다. 이는 매킨토시용 데이터변환 프로그램 전문 업체를 통해 개발하였다.

3) 문제은행 DB 구축

생성된 XML 파일은 CMS를 거쳐 데이터 베이스에 저장되게 된다. 저장과정은 CMS Bulk Import 유틸리티를 이용해 이뤄진다. DBMS는 O2를 사용하였다. 이 시스템에서 사용한 CMS는 O2 데이터베이스를 기반으로 만들어진 저장관리기이다. O2는 ODMG-93표준을 적용한 객체지향 데이터 베이스로 Object를 하나의 저장 단위로 한다. CMS는 먼저 문서를 내장된 XML 파서로 파싱하여 Validate를 체크한다. Validate가 끝나면 문서의 선언부에서 외부 엔티티(이미지, 멀티미디어 파일)를 읽어 Blob타입의 객체로 데이터베이스에 저장하게 된다. 그 다음 문서 본문을 트리 형태로 분해한 후 각각의 엘리먼트와 속성을 하나의 객체로 데이터베이스에 저장하게된다. 이러한 과정을 거쳐 문서와 각각의 엘리먼트는 고유한 OID를 갖게 된다.

4) 서비스 시나리오 설계 및 서비스 모듈 개발

서비스 시나리오는 문제 추출 후 해당 문제를 해결한 후 채점된 결과를 확인하는 일련의 과정으로 설계했다. 설계 적용은 UML(Unified Modeling Language)를 이용해 Use-Case를 적용하고 이에 맞는 Sequence-Diagram을 작성했다. 가장 중요한 시나리오 흐름은 추출, 채점, 통계 처리를 위한 설계이



(그림 4) 문제 추출 Sequence Diagram

다. 각각의 시나리오는 JavaBean 형태의 모듈로 생성하고 이는 여러 서비스에 공통으로 적용되게 된다. 문제 추출 시나리오(그림 4)를 보도록 하자.

사용자가 문제 화면에서 자신의 진도, 수준, 문제 수를 선택하면 해당 값은 추출모듈을 통해 DB에서 검색되어지고 검색 결과 해당 문서의 DID(Document ID : Import 시 생성된 Object ID)리스트를 CMS로 보내면 CMS는 해당 리스트의 문서들을 통합하여 Assemble XML 문서를 생성하게 된다. XML 문서는 XSLT(Xml Stylesheet Language Transformation)엔진을 통해 HTML로 변환해 사용자에게 서비스하게 된다.

5) XSL 스타일 개발

DTD 개발이 완료되고 문서가 생성된 후 문서에 맞는 XSL 스타일을 개발하였다. XSL 파일은 XSLT 엔진을 이용해 HTML로 컨버전된다. 현재 본 시스템은 Apache Group의 Xalan-Java version 1.2.2를 이

용해 변환하였다.

4. 결론

XML 문제은행 시스템은 콘텐츠(학습문제)를 유료화 하여 서비스하는 사이트에서 사용하는 경우가 많다. 그것은 출판사들에겐 콘텐츠만으로도 큰 자산 가치가 있기 때문이다. 유료 사이트인 만큼 사용자의 요구 조건에 따라 해당문제를 빠른 속도로 제공하는 것이 개발 상의 관건이었다. 특히 수준별로 문제를 분류하고 학생의 수준을 문제 채점 후 계산하여 다음 추출부터 수준이 적용된 문제를 추출하는 알고리즘 등이 잘 설계되어야 동적인 학습 서비스를 제공할 수 있다.

DTD는 단순히 구조만을 표현하는 것이 아니라, 스타일 상의 필요한 정보와 검색 정보 등 문항만의 고유한 속성을 찾아내 DTD에 반영해야 한다. 또한 텍스트 지향적인 XML 문서의 경우, 어느 단계로

문서를 나눠 저장할 지도 중요하다. 이는 시스템의 성능과 연관되기 때문이다. XML 시스템의 경우 기존의 HTML 시스템에 비해 문서 뷰잉 속도가 느릴 수밖에 없다. 중간에 스타일을 처리하기 위한 변환 시간이 발생하기 때문이다. 특히 XSLT와 같은 변환 엔진을 사용하는 경우 한글처리와 속도에 대한 부분을 고려해서 선택해야 한다.

참고문헌

- [1] W3C, Extensible Markup Language(XML) 1.0 (Second Edition), W3C Recommendation 2000.10. 6, <http://www.w3.org/TR/2000/REC-xml-20001006>
- [2] W3C Working Group, XSL Transformations (XSLT) Version 1.0, 1999.11, <http://www.w3.org/TR/xslt.html>
- [3] ODMG(Object Data Management Group), ODMG 2.0:A Standard for Object Storage, 1998.7, <http://www.odmg.org/>

저자약력



양 광 모

1998년 중앙대학교 정보시스템학과 졸
 1998년 (주)넥스텍 연구소 입사
 1999년 한국도로공사 XML IETM 시스템 개발 업무
 2000년 lyah.com XML 문체은행 시스템 개발팀장
 2001년 XML 저장 관리 시스템 개발 업무
 2001년 (주)넥스텍 연구개발부 주임연구원
 관심분야: 데이터베이스, XML, CBD, 무선 인터넷
 e-mail : kmyang@nextech.co.kr



최 성

1975년-1980년 기업은행 전산개발부
 1983년 연세대학교 산업대학원 전자계산학과 공학석사
 1983년-1985년 제주은행 전산실장
 1986년-1994년 한국생산성본부 OA추진 사무국장
 1999년 강원대학교 대학원 컴퓨터학과 이학박사
 1994년-현재 남서울대학교 컴퓨터학과 교수
 현재 한국정보처리전문가협회 이사, 한국정보통신 기술사회 이사
 관심분야: EC, ERP, S/W Engineering, VR영상게임 등