

XML 기반 데이터 공유 Interface 설계 및 구현

김철원* · 김상영* · 박종훈**

*호남대학교 · **중부대학교

A Design and Implementation for Data Sharing Interface in based XML

Sang-young Kim* · Chul-won Kim* · Jong-hun Park*

*Honam University · **Joongbu University

E-mail : hrsksy@nate.com

요 약

XML 문서를 저장하고 검색하는 시스템에 관련된 연구가 활발하게 이루어지고 있으며, 이러한 시스템들은 효율적으로 XML 문서를 저장하고 검색할 수 있는 기능에 중점을 둔 경우가 많다. 또한 XML 문서를 저장하기 위해서 특별히 설계된 테이블이나 저장 구조를 가지고 있어서 XML 문서의 내용뿐만 아니라 문서의 구조 정보 등도 같이 저장할 수 있으며 이를 바탕으로 XML 문서의 내용 검색이나 구조 검색 등을 효율적으로 할 수 있도록 하고 있다. 본 논문은 기존에 사용해 왔던 여러 종류의 데이터베이스들이 갖고 있는 데이터들을 그대로 재활용 하여 XML 형식으로 변환하고, 이 XML 파일을 웹에서 공유하여 여러 가지 다양한 Interface를 통해 출력 될 수 있도록 설계, 구현하였다.

ABSTRACT

Study related to a system that saves a XML document, and to search is consisting actively and has a lot of cases to have left emphasis in the function that these systems efficiently save a XML document and can search. Also, It has a table or the storage structure which was especially designed in order to save a XML document and can save structure information of a document in addition to contents of a XML document together and can efficiently do content retrieval of a XML document or a structure search with an early base. As for this paper, a design implemented the data which the many different kinds of database that had currently used had with you so that did recycling, and shared conversion and this XML file in Web, and output can become a XML format through various Interface.

키워드

translation, XML Data Query, Interface

1. 서 론

현재 우리는 인터넷을 기반으로 하는 웹의 급속한 성장으로 인해 다양한 분야의 정보를 인터넷을 통해 얻을 수 있게 되었다. 전세계의 모든 사람들의 통신이 가능하게 하는 인터넷은 주로 HTML을 사용하고 있으나, 인터넷상의 정보를 표현하기 위한 HTML이 구조화된 문서를 표현하기가 어렵고, 인터넷이 요구하는 다양한 역할을 감당할 수 없는 단점을 가지고 있음에 따라 SGML과 HTML의 단점을 보완하여 W3C에서는 HTML과는 달리 Tag를 정의할 수 있고 데이터를 기술할 수 있는 차세대 웹문서의 표준으로 XML을 제안하게 되었다. XML은 Extensible Markup Language의 약자로

HTML과 같이 Markup Language이다. 그러나 정확한 뜻은 마크업언어가 아니라 마크업언어를 정의하기 위한 언어이다. 뿐만 아니라 웹상에서 구조화된 문서를 전송 가능하도록 설계된 표준화된 텍스트 형식의 마크업 언어로써 SGML의 Subset이며 SGML보다 훨씬 간결하고 인터넷에서 바로 사용할 수 있는 문서를 표현하는 표준이다.

XML은 SGML의 부분집합으로서 HTML이 갖는 문서 구조의 확장성과 문서의 포맷 검사 기능에 대한 문제점을 해결하였다. 뿐만 아니라 SGML에서의 태그의 최소화나 태그의 생략과 같은 해결하기 어려웠던 부분을 제거함으로써 문서의 처리가 쉽고 명확하게 되었다. 이는 다양한 플랫폼에 내용을 더 구체적으로 표현하고, 더욱 산별된 결과를 추출

하게 해 줄 것이다.

본 논문에서는 웹 상의 다중 Interface를 기반으로 기존에 다른 응용프로그램에 사용하던 데이터베이스를 XML Data로 변환하는 것에 대한 설계 및 구현을 함으로써 공유된 XML 데이터의 수정, 추가, 삭제 및 검색 기능이 효율적으로 이뤄질 수 있는 인터페이스를 제안하였다.

II. 관련연구

2.1 기존 웹 환경에서의 DB 활용

인터넷이 처음 등장한 시기에는 정보를 제공하기 위해 고정된 문자 데이터나 이미지, 그리고 사운드 데이터를 HTML문서를 통해 보여주고, 소량의 데이터나 C, C++, Perl 등을 사용해 파일에 저장하는 CGI(Common Gateway Interface)프로그래밍이 주로 사용되었다. 그러나 전자상거래나 기업간의 비즈니스등에 활용하는 단계로까지 발전하면서 웹과 데이터베이스의 연구는 더욱 가속화 되었다.

홈페이지를 만들 때 HTML과 자바스크립트는 웹브라우저에서 해석되는 클라이언트측의 스크립트다. 또한 데이터 저장 관리를 위해 C, C++등의 언어를 사용하여 기본 문법을 배우기 어렵고 오류에 대한 디버깅과 DB 연동이 어려운 단점이 있다. 반면 PHP나 ASP, JSP와 같은 서버측의 스크립트들을 서버에서 해석되고 결과만을 HTML형태로 웹브라우저에 전송함으로써 보안면에서 유용하고 DB 연동이 쉽다는 장점을 가지고 있다.

2.2 XML의 활용성

인터넷상에 존재하는 모든 데이터를 XML 형식으로 변환 후 이 파일을 공유해 모든 플랫폼과 응용 프로그램으로 재사용이 가능하다. 데이터 전송과 처리라는 측면에서 XML은 확실한 표준으로 자리잡으며 웹의 큰 축이 되고 있다. XML은 문서나 콘텐츠를 잘 표현할 수 있는 공통 포맷 언어이고 여러 다양한 시스템 간, 어플리케이션간의 데이터 교환을 위한 인터페이스 포맷으로도 유용하게 쓰여질 수 있기 때문에 EDI/CALS의 전자상거래와 국가 행정 업무간 문서 전송과 저장, 자료 검색 등 인터넷과 DB를 사용하는 범위를 뛰어 넘는 다양한 분야로 확장되고 있다.

본 논문에서는 기존의 다른 응용 프로그램에 의해 생성된 데이터를 XML 데이터 포맷형식으로 변환함과 동시에 서로 다른 플랫폼에서 효율적으로 출력하기 위해 사용할 수 있는 XSL 사용 방법을 제안하였다.

III. XML기반 인터페이스 설계 및 구현

3.1 XML 기반 Interface 설계

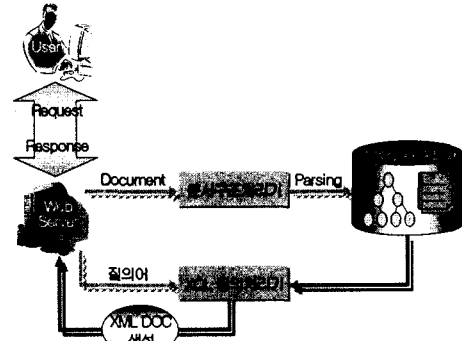


그림 1. XML 문서 검색 시스템 구성도

XML 편집기에 의해 작성된 문서의 내용은 파싱한 후 JDBC/ODBC 인터페이스에 의해 데이터베이스에 저장을 한다. 질의어 처리 과정은 Well-formed XML 문서를 입력 받아서 문서 구조 처리기에 의해 파싱한 후, start/end document, start/end element, 텍스트들을 분류하여, XML 문서를 인덱스와 문서구조 정보를 구성하고 그 정보를 XML 질의어 처리기의 문서 검색을 위해 저장한다. 질의 결과에 대한 구조 정보는 인덱스 정보와 함께 스택에 유지되어지며, 결과에 대한 질의를 계속하여 처리할 수 있도록 인덱스 정보를 계속 유지한다. 이 시스템에서는 XML 문서의 엘리먼트, 속성과 내용에 대하여 검색 할 수 있도록 시스템을 구성하였고, 그 질의 결과로는 XML 문서를 출력할 수 있도록 하였다.

본 논문에서 설계 및 구현한 데이터 공유 시스템에서 실험 1단계에서는 데이터 소스를 응용프로그램과 운영체제에 종속되지 않고 자유롭게 사용하기 위해 XML Data로 변환 후 웹에 게시하였고, 실험 2단계에서는 웹에 있는 XML Data를 공유하여 여러 가지 출력 인터페이스를 제작하는 과정이다.

3.2 Data Translation

데이터의 Translation에서는 접근 응용 프로그램과 운영체제에 영향을 받지 않는 형식으로 변환하는 것이 중요하다. 다양한 인터넷 접근 프로그램이 있으나 여기서는 비주일베이직을 이용한 Translation과 ASP를 이용한 Translation을 사용하고자 한다.

3.2.1 비주일베이직을 이용한 Translation

비주일베이직을 이용한 Translation은 여러 가지 DB의제공자를 선택하고 사용할 DB를 선택한 후 선택되어진 제공자와 DB 이름을 다음 출력 화면으로 넘기기 위해 Property Procedure를 작성한 후 ADO 개체를 이용해 다양한 데이터 소스에 데이터를 연결해 XML 데이터로 변환하는 1단계 Interface와 1단계 Interface에서 넘겨 받은 제공자와 DB를 이용해 XML 데이터로 변환 생성하는 2

단계 Interface로 구성되어 있다.

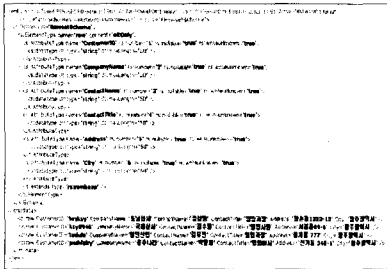


그림 2. VB를 이용하여 변환된 XML 문서의 구조

3.2.2 ASP를 이용한 Translation

이 단계는 비주얼베이직을 거치지 않고 ASP에서 XML Data를 웹에 게시하기 위한 XML 데이터를 생성한 Translation이다. DOM을 ASP 문서내에서 이용하기 위해 MSXML 인스턴스를 만들어야 한다.

```
Set objXML=Server.CreateObject("Microsoft.XMLDOM")
objXML.load(Server.MapPath("Customers.xml"))
Set objRootElement = objXML.documentElement
```

그림 3. MSXML 인스턴스 생성 코드

<그림3>과 같은 ASP를 실행하면 XML 문서를 파싱하면서 인덱스 구조 정보를 생성시킨다.

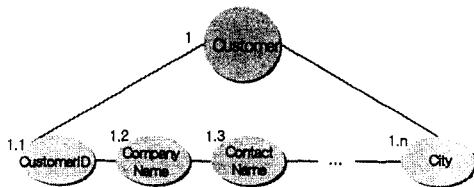


그림 4. MSXML 파싱후 생성된 DOM트리

ASP에서 DOM이 함께하면 문서를 적재하기, 검증하기, 트리 구조 이동하기, 내용 생성하기 등이 가능하다.

3.3 XML Data Query

웹에 공유되어 있는 XML 데이터를 사용하기 위한 방법을 제시하는 단계이다. XML 데이터는 XML 파서를 이용하여 데이터를 사용하는 방법과 직접 XML 데이터로 접근하여 데이터를 가지고 가는 방법(DDSQ)이 있다.

3.4 웹 환경의 출력 인터페이스

웹 환경에 시스템의 전반적인 구성요소는 그 시스템의 성능을 결정하는 것만큼 중요하다고 할 수 있다. 그러나 이것은 개발자와 시스템 설계자의 측면에서 보는 관점일 것이다. 다음은 A, B, C 세가지의 인터페이스의 특징과 출력 화면을 살펴보고 룩 한다.

3.4.1 A-Type Interface

A-Type Interface는 XML 데이터를 DSO 개체를 이용해 HTML로 작성한 인터페이스로서 공유되어 있는 XML데이터에 DSO 개체를 이용하여 직접 액세스한 후 출력 인터페이스에 결과를 나타낸다. 출력 ASP에서 ASP 스크립트로 클라이언트에서 요구하는 쿼리를 DDSQ 형식으로 공유된 XML Data로 접근 후 결과를 출력하는 Interface이다. HTML 페이지에서 XML 데이터 소스를 이동하는 출력 인터페이스를 작성하기 위한 순서는 아래와 같다.

- ① 텍스트 편집기를 실행한다. XML 데이터 내용을 작성한다.
- ② HTML의 <XML> 태그를 사용하여 분리되어 있는 데이터를 통합하는 HTML 페이지를 만든다.
- ③ 테이블 형식을 나타내는 HTML 태그를 작성한다. 그리고 XML 데이터를 결합하기 위해 dataFld와 dataSrc를 이용한다. dataSrc 속성값은 앞에 #기호가 붙은 DSO의 이름이다. DataFld 속성은 레코드셋의 어느 컬럼이 페이지 요소에 연결해야 하는지를 지정한다.
- ④ 데이터의 이동 및 데이터를 추가하기 위한 단추를 추가하는 HTML을 작성한다.
- ⑤ 위의 단계대로 HTML 페이지를 완성하면 A-Type Interface가 된다.

3.4.2 B-Type Interface

B-Type Interface는 XML Data는 XSL 문서가 같은 파일 내에 위치한 Interface로서 CSS와 XSL 형식의 Style Sheet를 가지고 있어 XML Data를 다양한 형태로 출력이 가능한 Interface이다. 텍스트편집기로 출력을 위한 HTML태그를 입력하고 사용하고자 하는 데이터를 XML 문서 구조에 맞추어 입력한다.

3.4.3 C-Type Interface

C-Type Interface는 다른 곳에 위치한 XML Data와 XSL을 결합시켜 출력하는 인터페이스로서 기존에 한정된 문서의 다양성, 재사용성, 활용성이 플랫폼과 응용프로그램에 종속되지 않고 바로 해결할 수 있는 Interface이다.

C-Interface를 출력하기 위한 알고리즘은 다음과 같다.

- ① XML데이터는 기존에 공유되어 있는 XML 데이터를 사용한다. 출력하고자 하는 XSL을 각각 작성하여 저장한다. XML 데이터, 그리고 테이블 형식으로 출력하고자 하는 각각의 XSL을 출력하기 위해 HTML을 작성한다. 먼저 사용할 언어를 VBScript로 선언한다.
- ② <HEAD>부분에 Function을 작성한다. 세가지의 Function을 transformNode를 이용하여 XSL을 변경하여 출력하도록 한다.
- ③ XML 문서와 XSL 파일을 사용하기 위해

- <XML> 태그를 사용한다.
- ④ <BODY> 부분에는 XSL을 선택하기 위한 단추를 세 개 입력한 후 <DIV> 태그를 이용하여 하단에 XSL 데이터와 XSL파일이 결합된 결과 인터페이스를 출력하여 준다.
 - ⑤ 위 단계를 거쳐 완성된 C-Type Interface 즉, 테이블 형식의 XSL을 XML 데이터와 연결하여 출력하면 아래 그림과 같은 C-type Interface가 완성된다.

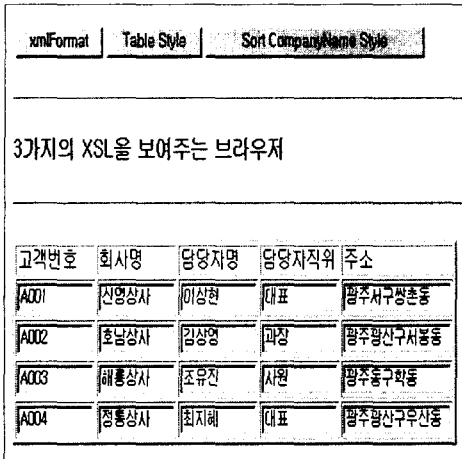


그림 5. C-Type Interface

IV. 비교 분석

설계부분에서 RDB를 이용한 웹 환경과 XML 데이터를 기본으로 한 웹 환경의 비교했을 때 어떤 스크립트 언어이든 DB와 연동하려면 ODBC와 JDBC를 이용하기 위해 DB를 연결하는 중간계층이 존재해야만 가능하다. 그러나 이런 RDB와 별개로 XML은 텍스트 형식이므로 문자열을 처리할 수 있는 시스템과 응용 프로그램에서는 공유할 수도 있고, 검색, 추가 등의 데이터베이스 작업을 지원한다는 것이다. 구현에 대한 결과로서 XML 데이터의 공유와 출력 인터페이스를 비교했을 때 XML 데이터는 텍스트 기반의 형식이므로 텍스트와 연결되는 모든 응용 프로그램 혹은 플랫폼에서 활용될 수 있기 때문에 기존에 자료에 비해 매우 호환성이 높다. 단지 인터페이스 부분에서 본다면 윈애플과 같이 사용자의 다양한 취향에 따른 스킨제공을 하는 비슷한 기능으로서 간단한 멀티인터페이스가 가능한 것도 XML의 장점이라고 본다. 본 논문에서 구현된 3가지 인터페이스를 간단하게 비교하면 아래 <표1>과 같다.

표 1. 인터페이스의 비교항목

	A-type Interface	B-type Interface	C-type Interface
XML 데이터 위치	별도의 파일로 저장	HTML 문서에 포함	별도의 파일로 저장
XML 공유	가능	불가능	가능
XML 데이터 확장성	확장성 용이	불가능	가능
XSL 작성 방법	HTML 태그에서 DDSO 개체 활용	HTML 태그를 이용하여 출력	별도의 XSL 작성 저장
문서 재사용성	가능	불가능	가능
플랫폼	HTML 브라우저에만 출력	HTML 브라우저에만 출력	모든 웹 시스템 적용

V. 결론 및 향후 연구과제

데이터로서의 XML은 SGML 및 HTML의 장점을 모두 수용한 것이라 할 수 있으며, 단지 웹 브라우저에 표시하기 위한 것이기 보다는 어떠한 종류의 응용 프로그램과도 통합될 수 있는 범용적인 데이터베이스라고 할 수 있다. XML이 XML문서만을 의미하는 것이라면 DB와 다르지만 만약 XML이 문서뿐만 아니라 연관된 모든 XML도구 및 기술을 의미하는 것이라면 DB와 유사하다고 할 수 있다. 이것은 XML이 데이터베이스에서 볼 수 있는 storage(the XML document), schemas(DTDs, schema languages), query languages(XQL, XML-QL, QUILT), programming interfaces(SAX, DOM)등을 포함하기 때문이다. 하지만 실제 DBMS가 가지는 효율적 저장, 인덱싱, 보안, 트랜잭션과 데이터 무결성, 다중사용자 접근, 트리거, 다중문서 질의 등은 가지지 않기 때문에 실제 데이터베이스와 완전히 동일하지는 않다. 데이터베이스는 방대한 양의 데이터를 다루는 데 있어 매우 효과적이며 전송에 민감한 응용 프로그램들이 갖추어야 될 견고함, 무결성, 일치성, 가용성과 같은 기본적인 특성들을 제공하지만 서로 다른 데이터베이스들간에 데이터를 전송할 땐 공통의 데이터 교환 양식으로서 표준화된 것이 아직 존재하지 않다. 바로 이 부분에서 XML이 사용될 수 있다. 즉, XML은 데이터를 교환하는데 가장 잘 사용할 수 있고, 데이터베이스는 데이터를 저장하고 불러오는데 가장 잘 사용할 수 있다는 점에서 두 기술은 서로 보완이 가능해야 한다.

현재의 최적 솔루션을 확보하는 것도 중요하지만 장기적인 안목을 가지고 XML 기반 체반 요소 기술 및 각종 솔루션 개발에 노력을 기울여야 할 것이다.

참고문헌

- [1] PROFESSIONAL XML 2nd Edition XML Schemas, SOAP, SXLT, DOM, SAX 2.0,

- David Hunter 외 5인 공저 천주선 외 3인
공역, 정보문화사, 2002
- [2] PROFESSIONAL XML Database, Kevin
Williams 외 9인 공저, 최효진, 안성욱 외 공
역, 정보문화사, 2001
- [3] Visual Basic6 XML, James Britt & Teun
Duynstee 공저, 이태용 역, 정보문화사, 2000
- [4] XML Link Requirements Vwesion 1.0
<http://www.w3.org/TR/xlink>, Working Draft
- [5] ASP와 XML을 활용한 차세대 웹 프로그래
밍, 손진곤, 한연희 공저, (주)이한디지털리,
2001
- [6] Microsoft XML developer center,
<http://msdn.microsoft.com/xml/default.asp>