

XML 기반의 정보 처리 시스템 설계 및 구현

남윤영⁰ 황인준
아주대학교 정보통신전문대학원 정보통신공학과
{youngman, ehwang}@madang.ajou.ac.kr

Implementation of XML-based Information Processing system

Yunyoung Nam⁰ Eenjun Hwang
The Graduate School of Information and Communication, Ajou University

요약

최근 들어 기업이나 관공서, 기관 등에서 상호간에 정보를 서로 교환하거나 사용자에게 제시할 때, 대부분 웹을 통해 브라우저에서 볼 수 있도록 HTML(HyperText Markup Language)형태로 제작하게 된다. 그러나 HTML로 제작된 웹 페이지는 구조적인 정보나 다양한 정보를 표시할 수 없다는 문제점이 있으며, 빠르게 변하는 정보인 경우 상세한 정보 표현 면에서 취약하다는 단점이 있다. XML(eXtensible Markup Language)은 이러한 문제점을 해결할 수 있는 마크업(markup) 언어로써 차세대 데이터 교환의 표준으로 채택하고 있다. 한편, XML이 HTML보다 더 나은 구조와 기능을 제공하고 있으나 XML 자체만으로는 XML이 가지는 여러 가지 장점들을 충분히 활용하기 어렵기 때문에 스크립트 언어의 사용이나 애플리케이션의 제작이 필수적이다. 본 논문에서는 효율적인 정보의 교환과 공유를 위해 XForm과 XML Query를 사용하여 정보를 데이터베이스에 저장하고 XML 형태로 추출, 교환할 수 있으며, 서버(server)와 클라이언트(client) 사이에서의 정보 교류뿐만 아니라 유사한 콘텐츠를 제공하는 여러 서버들간의 정보 공유를 지원하는 정보 처리 시스템을 제안한다.

1. 서론

웹 페이지 제작시 주로 사용되는 언어인 HTML은 프레젠테이션용으로 사용하기에 용이하지만 구조적인 정보의 표현이 어려워 상호간의 정보 교류가 어렵다. 이러한 문제점을 해결하기 위해 W3C에서는 XML[1]을 표준으로 채택하였다. XML은 HTML이 지니는 한계를 극복할 수 있으며, 다양한 형태의 정보 교류에 효과적이라는 커다란 이점이 있다.

현재 기업이나 관공서, 기관에서 사용되는 정보들은 빈번하게 생성, 갱신, 소멸된다. 하지만 표기되는 정보의 구조적인 형태가 서로 다르기 때문에, 정보 교류시 비효율적이며 시간과 인력의 낭비가 심하다. 또한 정보 생성자가 다수이고 정보가 일관성이 없는 형태로 생성되는 경우에는 많은 문제들이 발생된다. 예를 들어, 여러 부품들이 결합하여 여러 종류의 완성품을 제작하는 기업의 경우, 부품에 대한 세부 내용이 정보가 되고 부품을 납품하는 공급자와 부품을 조립하는 제조자가 이 정보를 이용할 때, 통일된 DTD(Data Type Definition)나 XML Schema를 사용하여 XML 형태로 표현하면 정보의 통일성과 일관성을 가질 수 있고 효율적인 작업 환경을 갖출 수 있다. 더 나아가, 기관들이 상이한 DTD와 XML Schema를 사용할 때, 이를 매핑(mapping)시켜 통일된 형태로 만들어줄 수 있다면, 정보 교환시 향상된 형태로 발전될 수 있다.

따라서, 본 논문은 효율적인 정보 교류를 할 수 있도록 정보를 XML 형태로 생성하여 신속하게 정보를 교환할 수 있고, 스키마가 다른 XML 문서를 매핑·변환시켜 정보 공유를 할 수 있는 XML 기반 정보처리 시스템을 제안한다.

2. 관련연구

2.1. HTML Forms

HTML은 사용자에게 입력을 받아 사용될 수 있는 폼(form)이라는 엘리먼트(element)로 구성하여 정의될 수 있다. 이 때 폼은 웹에서 사용자와 상호 작용하도록 하는 중요하고 유용한 형태이다.

XML이 웹에서 애플리케이션간에 데이터를 교환하기 위한 표준 형태가 됨에 따라 폼이 갖는 역할과 XML context에서 폼이 어떻게 작동될 수 있는 지에 대해 관심이 모아지고 있다. 이에 따라, W3C에서는 XForm[4]이라 불리는 HTML 폼과 다른 XML 폼에 대한 연구가 진행되고 있다.

HTML에서의 폼은 폼 엘리먼트의 인스턴스(instance)이다. HTML 폼은 마크업과 text fields, checkboxes, menus 와 같은 폼 controls 인스턴스로 구성된다. 사용자들은 폼의 control을 통해서 상호작용하며 처리를 위해 웹서버 같은 원격 개체로 작성된 폼을 제출한다.

2.2 XML Forms

HTML이 폼을 위해 특별한 용어(form, input, button, etc)로 정의되어 진다면 XML은 언어를 정의하기 위한 언어이다. XML은 HTML과 다른 형태에서 작동되며 XML 폼 또한 HTML 폼과 다르게 작동된다[2]. XML 폼에서 폼 시트(formsheets)는 폼 값으로 어떻게 변환하는지를 나타낸다. 폼 시트의 중요한 장점은 XML 문서를 정렬을 이용하여 폼의 값을 구성할 수 있다는 것이다. 이것은 일반적인 인코딩

(encoding)이 아닌 typed data를 제출할 수 있다. 폼 데이터는 폼 시트를 생성하여 네임스페이스(namespace)를 할당하며 MIME type는 enctype 속성에서 주어진다.[3]

클라이언트는 웹 서버에서 XML 문서와 스타일시트(stylesheets)를 적용하여 변환된 HTML을 웹 브라우저에서 볼 수 있다. 이 문서는 사용자의 입력에 반응하여 값을 변경한 후 서버에 제출하면 XML 문서의 데이터를 결과로 출력한다. 폼 시트를 이용하여 폼의 값을 구성하는 XML 문서와 폼 시트의 예는 표 1, 표 2에 나타나 있다.

표 1 폼시트를 이용한 폼 값 구성

```
<form xf:form="scoped" action="/JSP/login" method="post"
enctype="text/x-userid"
formsheet="login.xml"
form-lang="text/xml">
  User ID: <input type="text" name="userid"/><br/>
  Password: <input type="text" name="passwd"/><br/>
  <button xf:form="submit"/>
</form>
...
```

표 2 login.xml

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/xsl/">
<xsl:template match="/">
  <user>
    <id>
      <xsl:value-of expr="/form/input[attribute(name)='userid']/
attribute(value)"/>
    </id>
    <password>
  ...
```

3. 시스템 구성도

본 논문에서 제안하는 시스템은 크게 다수의 클라이언트와 서버로 구성된 환경에서 서비스를 제공한다. 클라이언트는 서버를 통해 정보를 추가, 갱신, 삭제, 검색을 할 수 있으며, 서버는 클라이언트에 정보를 제공하며 서버간에도 정보를 주고 받는다. 개략적인 시스템 구성도는 그림 1과 같다.

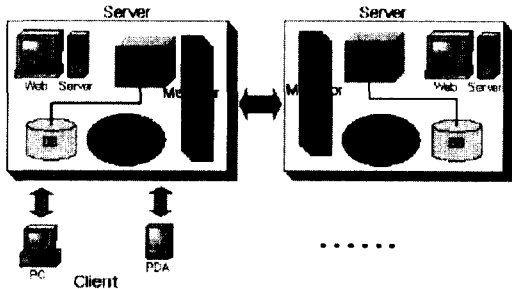


그림 1 XML 기반 정보 처리 시스템 구성도

3.1 시스템 처리 구성도

그림 2는 XML 정보 처리 시스템이 XML 문서를 처리하는 과정을 보여준다. 구조적인 정보를 담고 있는 DTD, XML Schema가 있으며, XML 문서는 XSLT[6] Template와 폼을 적용하여 브라우저로 보여진다. 폼을 통해 얻은 입력 값으로 데이터베이스에 질의를 하여 원하는 정보를 얻거나 추가, 갱신, 삭제를 할 수 있다.

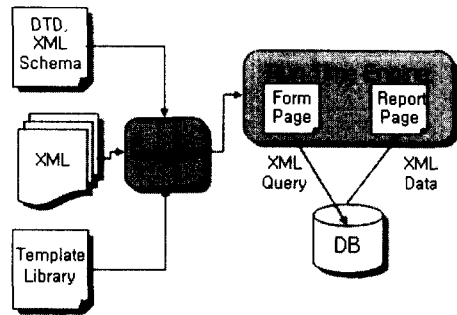


그림 2 XML 폼 처리 구성도

데이터베이스와의 연결은 JDBC를 사용하여 이루어지고 오라클 데이터베이스를 사용할 경우 XDK[5] 패키지에 있는 XSQL[6]을 사용하며, XML 문서를 데이터베이스에 저장하고 불러오기 위해 XSU(XML SQL Utility for java)를 사용한다.

XSU는 XML 문서의 생성을 위한 API와 삽입, 갱신, 삭제를 위한 API가 정의된 두개의 클래스로 구성된다. 우선 XML문서를 생성하기 위해서는 데이터베이스와 연결을 하고 XML Query 인스턴스를 실행한 후, DOM tree 또는 XML string를 결과 값으로 가져온다. 그림 3은 XSU를 이용한 XML 문서의 생성 과정을 나타낸다.

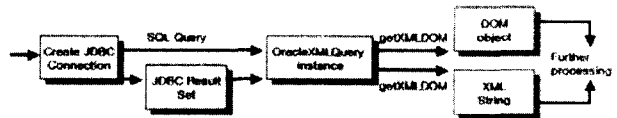


그림 3 XSU를 이용하여 XML을 생성하는 처리도

XML 문서의 구조가 잘 정의되어지고 자주 변경이 가해지는 경우는 XML 문서의 전체 또는 일부를 DB의 관계형 테이블 형태로 저장하는 것이 효율적이다. XSU는 XML 문서의 전체 또는 일부를 테이블로 매핑하여 저장하는 기능과 테이블의 데이터를 검색하는 기능을 제공한다. 또한 XML 데이터가 관계형 DB에 저장될 때 XML 문서의 각 엘리먼트의 이름들은 관계형 DB의 테이블에서의 컬럼명으로 매핑되고 이것은 XSL를 이용하여 다른 XML문서로 변환할 수 있다. XML 문서와 오라클 DB가 연동되는 내부모습은 그림 4와 같다.

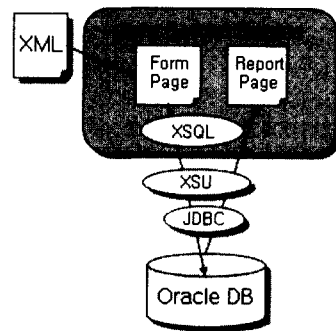


그림 4 오라클 DB와 연동한 XML문서 처리도

4. 시스템의 주요 기능

4.1 서버와 클라이언트간의 통신

사용자는 웹상에서 폼을 통해 질의를 보내고 그 결과로 생성된 XML data를 브라우저 상에서 확인할 수 있으며 애플리케이션을 통해서도 실시간으로 업데이트되는 정보들을 받아 볼 수 있다. 애플리케이션간에 교환되는 메시지는 정형화된 프로토콜 DTD를 따르며 표 3은 프로토콜 DTD의 일부분을 보여 준다.

표 3 프로토콜 DTD

```
<?xml version="1.0" encoding="EUC-KR" ?>
...
<!ELEMENT message((body?[*]*error*|subject? | thread?*))>
<!ATTLIST message to CDATA #IMPLIED from CDATA #IMPLIED
...
<!ELEMENT subject (#PCDATA)>
...
<!ELEMENT iq (#PCDATA)>
<!ATTLIST iq to CDATA #IMPLIED from CDATA #IMPLIED
id CDATA #IMPLIED type(get | set | result | error) #IMPLIED>
...
```

표 4는 프로토콜 DTD에 맞춰 생성된 XML형태의 메시지의 일부분이다.

표 4 새로운 정보를 알리는 XML형태의 메시지

```
<iq type="result" from="server001" id="1001">
<query xmlns:xips="http://adtl.ajou.ac.kr/xips">
<xips-release priority="optional">
...
<message type="normal" from="server001" to="client001">
<subject>welcome to XIPS</subject>
...
```

정형화된 프로토콜 DTD에 따라 생성된 XML 메시지를 사용하여 정보가 갱신되거나 새로운 정보가 생성되었을 때 현재 접속된 각 클라이언트에게 정보가 변경되었음을 알려주며 클라이언트가 원할 경우 열람할 수 있다.

4.2 서버와 서버간의 통신

유사한 콘텐츠를 제공하는 기관들간의 정보 교류나 데이터베이스의 저장되어 있는 데이터 교환시 데이터베이스의 스키마가 다르고, 데이터베이스에서 추출한 데이터를 HTML로 표현한다면 구조적인 정보의 표기가 어렵기 때문에 시간과 비용이 낭비되는 문제점을 야기한다.

이런 경우, 데이터베이스에 저장되어 있는 데이터를 XSL, XPath[7], XQuery[8]를 사용하여 추출하여 XML 형태로 생성하고, 생성된 XML 문서의 엘리먼트간의 관계를 서로 매핑시켜 주는 미디어터(Mediator)를 통하여 원하는 XML 문서 형태로 변환한다. 변환된 XML 문서는 데이터베이스에 저장하여 서버간에 정보 공유를 할 수 있게 된다.

또한, 사용자는 여러 서버에 접속할 필요없이 하나의 서버에 접속하여 원하는 정보를 받아볼 수 있다. 그림 5는 서로 상이한 XML 구조에서 미디어터를 통해 동일한 콘텐츠 관계를 매핑시켜 주는 개략도이다.

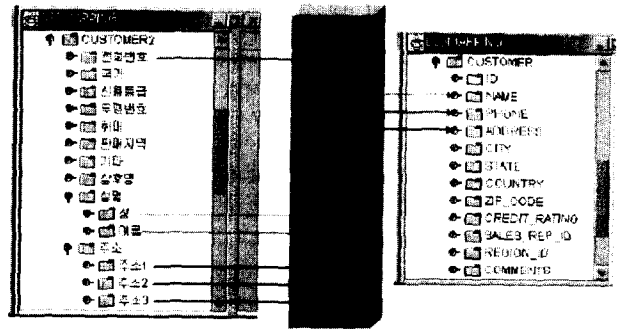


그림 5 Mapping Mediator를 통한 XML 변환

5. 결론 및 향후 과제

본 논문에서는 XForm, XML Query, XSU 등을 이용하여 웹상에서 XML 문서를 데이터베이스에 저장, 삭제, 갱신, 검색하고 저장된 데이터를 XML 문서로 생성할 수 있으며, 애플리케이션을 통해 사용자는 신속하고 정확하게 원하는 정보를 받아 볼 수 있는 시스템을 제안하였다. 특히 제안된 시스템은 이질 스키마를 사용하는 서버간에도 미디어터를 통해 각 XML 스키마를 매치시켜 정보를 공유하고 작업 효율성을 높일 수 있다. 이 시스템은 웹을 통해 원하는 정보를 가져올 수 있으므로 방화벽이 설치된 곳에서도 접근이 가능하며, 애플리케이션간에 주고 받는 메시지가 XML 형태이기 때문에 가독성과 확장성면에서 그 이용도가 높다. 또한 클라이언트의 종류에 따라 스타 일시트를 다르게 적용하기 때문에 화면의 크기가 제한된 PDA에서도 그 내용을 볼 수 있고, 비슷한 콘텐츠를 제공하는 서버간에 서로 정보를 공유할 수 있어서 클라이언트가 각 서버를 접근할 필요없이 하나의 서버에 접근하여 다른 서버에 있는 정보를 받아볼 수 있다. 이것은 B2B(Business to Business) 분야에 유용하게 사용될 것으로 본다.

향후 연구 과제는 접속한 클라이언트의 수가 증가했을 경우 발생하는 문제점을 분석하고 상이한 DTD를 매치시켜 줄 경우 중복되는 데이터를 줄여 효율적으로 서비스 할 수 있도록 시스템을 보완할 예정이다.

참고문헌

- [1] World Wide Web Consortium: eXtensible Markup Language (XML) 1.0, Feb.10.1998
- [2] Aders Kristensen : Formsheets and the XML Forms Language, in proceedings of WWW9, Toronto, Canada, May 1999.
- [3] M. Petropoulos et al : XML Query Forms(XQForms) : Declarative Specification of XML Query Interfaces, in proceedings of WWW10, May 2001.
- [4] XForm 1.0, W3C Working Draft 08 June 2001. <http://www.w3.org/TR/xforms>
- [5] Oracle Technology Network, http://technet.oracle.com/tech/xml/xdk_java
- [6] XSL Transformations (XSLT) Version 1.0 W3C Recommendation 16 November 1999. <http://www.w3.org/TR/xslt>
- [7] XML Path Language (XPath) Version 1.0 W3C Recommendation 16 November 1999, <http://www.w3.org/TR/xpath>
- [8] XQuery 1.0, W3C Working Draft 07 June 2001. <http://www.w3.org/TR/xquery>