

RDBMS로부터 XML 스키마 자동 생성에 관한 연구

박영수^o 장덕철^o
광운대학교 컴퓨터과학과
{yspark^o, dcjang}@kw.ac.kr

A Study on Automatic Generation of XML Schema from RDBMS

Young-Soo, Park^o Duk-Chul, Jang^o
Dept. of Computer Science, Kwangwoon University

요 약

웹 기술의 발달은 인터넷 사용자의 증가를 초래하였고, 같은 서비스에 대해 각자 다른 목적을 가지고 다른 환경 하에서 자신이 원하는 형태로의 서비스를 받고자 하는 경향으로 발전해 가고 있다. 또한 잦은 트렌드의 변화로 웹 문서의 개발 주기 또한 빨라지고 있는 상황이다. 따라서 이런 문제를 해결할 수 있는 방안으로 데이터의 구조적 표현이 가능하고, 동적인 요구에 대응하기 쉬운 XML기반의 웹 서비스에 관한 연구가 활발히 진행되고 있다. 본 논문에서는 DTD나 XML 스키마가 없는 상황에서 관계형 데이터베이스로부터 XML 스키마를 생성하고, 이를 바탕으로 XML문서를 생성할 수 있는 방안을 제시하였다. 이것은 XML문서의 개발 시간을 단축시킬 뿐만 아니라, 서비스가 요청되면 동적으로 클라이언트 환경에 맞추어 사용자가 원하는 형태로 서비스를 제공할 수 있을 것으로 기대된다.

1. 서 론

최근 웹 기술의 발달로 인해 유무선 인터넷 사용자가 급증하고 있다. 따라서 기업은 보다 새로운 정보를 다양한 형태의 채널을 통해 소비자에게 제공하려 하고, 소비자는 각기 자신이 원하는 채널을 통해 원하는 서비스를 받기를 원한다. 그러나 서버 측에서 보면 각기 다른 목적을 갖는 집단(소비자, 공급자, 판매자, 관리자, 유통업자 등)이 각기 다른 매체(데스크톱, PDA, 휴대폰, 전화, FAX 등)를 통해 각기 다른 환경(OS, 브라우저, 프로토콜 등)에서 각기 다른 서비스(상품정보, 가격정보, 주문정보, 판매정보, 고객정보 등)를 제공해야 한다.

또한, 소비자의 잦은 기호 및 트렌드 변화는 각종 컨텐츠의 life cycle을 짧게 하고, 잦은 구조 변경을 필요로 하게 되며, 동적인 변화에 대한 빠른 대응을 요구하게 된다. 이는 e-Commerce를 넘어 e-Business 시대에 해결해야 될 애플리케이션 통합과 직결되며, 이런 문제점을 해결할 수 있는 방법 중 하나가 W3C에서 표준으로 정한 XML(eXtensible Markup Language)[1]을 기반으로 한 웹 서비스 일 것이다.

본 논문에서는 XML문서를 보다 쉽고 빠르게 생성하기 위한 방법의 하나로, XML문서에 대한 DTD(Data Type Definition)나 XML 스키마의 정의가 없는 상태에서, 관계형 데이터베이스로부터 계층적 구조를 갖는 XML 스키마 및 문서를 생성할 수 있는 방법을 제시하고 설계하고자 한다. 2장에서는 관련연구를 기술하고, 3장에서는 XML 서버 구조 및 XML스키마를 생성하기 위한 6단계 정책을 제시한다. 4장에서는 앞서 제시한 정책을 기반으로 설계하고, 끝으로 5장에서 결론을 맺는다.

2. 관련 연구

2.1. XML

XML은 SGML(Standard Generalized Markup Language)의 복잡성을 제거하고, HTML(Hyper Text Markup Language)의 한계를 뛰어넘어, 웹 상에서 구조화된 문서를 전송 가능하도록 설계된 표준화된 텍스트 형식이며, 문서 그 자체가 데이터로서의 역할뿐만 아니라 XSL(eXtensible Stylesheet Language)과 동적으로 결합하여 최종 문서로서의 역할을 갖는다[1,2]. 따라서 기존의 XML 문서 생성 과정을 보면, 우선 데이터베이스가 구축된 상태에서, DTD를 정의하고, 이를 기반으로 XML문서를 작성한 후, XSLT(XSL Transformations)에 의해 XSL과 결합하여 최종적인 서비스를 제공하게 된다. 그러나 이런 일련의 작업들은 많은 시간과 노력을 요하게 되며, 특히 복잡한 구조를 갖는 문서에 대한 DTD를 정확히 정의하기는 쉽지 않다. 또한, DTD는 XML문서와 다른 문법을 사용하기 때문에 별도의 처리 과정이 필요하며, XML을 위해 개발된 DOM (Document Object Model), XPath, XLink(XML Linking Language) 등 다른 XML 표준들과의 협동 작업에 많은 제한을 갖고 있다[2,3]. 따라서 본 논문에서는 DTD 대신에 XML Schema를 쉽고 빠르게 생성할 수 있는 방법을 제시하고자 한다.

2.2 RDB와 XML DB의 비교

익시아소프트의 TextML, 소프트웨어 AG의 Tamino, XYZFind의 XYZFind 등 최근 XML을 위해 개발된 XML DB는 다른 형식의 문서도 쉽게 저장할 수 있고, 원본 문서의 정확한 재생 및 계층적 구조를 갖는 질의어에 빠른 응답이 가능하며, XML 기반의 질의어 및 프로그램이 가능한 XML 포맷을 지원함으로써 XML을 위한 DB의 최적화를 추구하고 있다. 그러나 기존 RDB가 제공하는 다양

한 관리기능, 상호 운영성, 프로그램 기능이 부족하며, 공급업체 고유의 질의어 및 프로그램 인터페이스를 사용해야 하는 단점과 기존 RDB와의 호환성 문제 등 아직은 많은 문제점을 드러내고 있다[4,5]. 그리고 최근 MS-SQL, Oracle 등 기존 RDB에서도 일부 XML 포맷을 지원하는 등 XML에 대한 기능을 점차 강화하고 있는 추세이다[3,4]. 따라서 본 논문에서는 MS-SQL Server 2000을 기준으로 설계하고자 한다.

2.3 DTD와 XML Schema의 비교

DTD와 XML Schema의 공통점은 XML 문서 내의 데이터에 대한 구조를 표현하는 규칙들의 집합으로 문서의 유효성을 판단하는 기준이 된다. 그러나 이들 간의 차이점은 <표 1>과 같다[3,5].

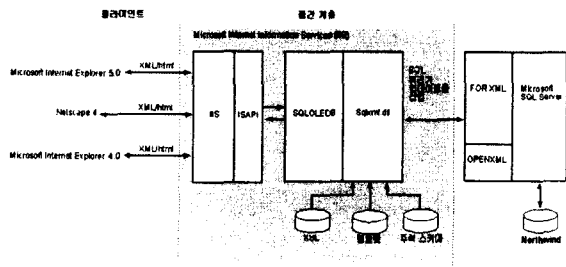
<표 1> DTD와 XML Schema의 차이점

| | DTD | XML Schema |
|----------------------|-------------------------------|--|
| 문법 | XML문서와 다른 문법 사용 | XML과 같은 문법 사용, 다른 XML 표준들과 함께 사용 가능 |
| 데이터 타입 | 지원 가능한 데이터 타입 제한 | 프로그래밍 언어에서 사용하고 있는 데이터 타입 제공 |
| 내용 모델(Content Model) | 요소(element)의 순서나 출현횟수 지정이 제한적 | 출현횟수의 지정 가능, 그룹개념 도입, 구조적/비구조적 텍스트의 혼용이 가능 |
| 확장성 | 한 문서에 오직 한 DTD만 적용 가능 | XML내에 네임스페이스를 이용하여 여러 스키마 참조 가능 |
| 동적 스키마 | 런 타임 상에서 DTD 변경 불가능 | 런 타임 시 원하는 스키마 선택 가능 |

3. XML 서버 구조 및 스키마 생성을 위한 정책

3.1 XML 서버의 구조

본 논문에서 사용되는 XML서버 구조는 <그림 1>과 같다. 이 서버를 사용하기 위해 우선 데이터베이스를 액세스하기 위한 가상 디렉터리가 설정되어 있어야 하고, 템플릿과 주석 스키마를 위한 가상경로도 함께 설정되어 있어야 한다. 우선 클라이언트에서 서비스 요청을 하면, IIS는 sqlisapi.dll에게 요청을 전달하고, ISAPI는 SQLOLEDB와 통신을 하게 된다. 실제적인 XML처리는 sqlxml.dll에서 처리하게 되며, FOR XML을 사용하여 데이터베이스로부터 질의에 의한 데이터를 가져오게 되고, OPENXML을 사용하여 XML 데이터를 데이터베이스로 가져오게 된다[4,5].



<그림 1> XML 서버 구조

3.2 계층구조를 갖는 XML 스키마 생성을 위한 정책
본 논문에서 제시하는 XML 스키마를 생성하기 위한 정책은 <표 2>와 같다.

<표 2> 계층구조를 갖는 XML 스키마 생성을 위한 정책

| 단계 | 내용 |
|-----|---|
| 1단계 | 데이터베이스 접속을 위한 URL입력 |
| 2단계 | 데이터베이스에서 Table List 2.1 사용할 Table 선택 |
| 3단계 | 선택된 Table of Column List 3.1 사용할 Column 선택 3.2 복수 Table 선택 시 Join 방법 설정 3.3 필요시 Column의 Alias 설정 |
| 4단계 | 선택된 Column 편집 4.1 요소(Element) 또는 속성(Attribute) 결정 4.2 필요시 가상 Column 생성(for Group) |
| 5단계 | Element의 세부 설정 5.1 상위 Level Element 결정 5.2 Element간의 순서 결정 |
| 6단계 | Attribute의 세부 설정 6.1 소속 Element 결정 |

<표 2>의 정책에 대한 설명은 다음과 같다. 1단계는 데이터베이스 접속을 위한 URL을 받아들인다. 2단계에서는 가상 경로에서 연결된 데이터베이스의 각 테이블을 리스트하고, 사용할 테이블을 선택하도록 한다. 3단계에서는 2단계에서 선택된 테이블에 대한 Column들을 리스트하고, 사용할 Column을 선택하도록 한다. 이때, 2단계에서 여러 개의 테이블이 선택되었을 경우 각 테이블의 Column들과 관계를 나타낼 수 있는 조건들을 기술할 수 있도록 하고, 필요하다면 각 Column들의 별칭(alias)을 사용할 수 있도록 한다.

4단계에서는 선택된 Column들만을 리스트하고, 각 Column들이 요소(Element) 또는 속성(Attribute) 중 어느 것으로 사용될 것인가를 결정하도록 한다. 필요할 경우 그룹(Group)을 위해 가상의 Column을 생성할 수 있도록 한다. 5단계에서는 4단계에서 요소로 선택된 Column들에 대한 상위 요소를 선택할 수 있도록 하고, 각 요소들 간의 순서(sequence)를 정할 수 있도록 한다. 끝으로 6 단계에서는 4단계에서 속성으로 선택된 Column들에 대한 상위 요소를 선택할 수 있도록 한다.

4. 질의어 생성 및 XML 스키마 생성을 위한 설계

4.1 XML 문서를 위한 질의어 생성 설계

<표 2>의 2단계에서 Table을 리스트하기 위한 XML문서의 코드는 다음과 같다.

```
<?xml version='1.0' encoding='EUC-KR'?>
<root xmlns:sql='urn:schemas-microsoft-com:xml-sql'>
  <sql:query >
    SELECT Table_Name //MS-SQL에서 제공되는 스키마
    FORM Information_Schema.Tables // 정보를 이용해서
    WHERE Table_Type = 'Base Table' //테이블을 리스트 함.
    FOR XML Auto, Elements
  </sql:query>
</root>
```

이때, 질의어와 "FOR XML, AUTO"가 포함된 XML문서를 템플릿을 통해 호출해 보면, 각 레코드 별 Column명이 속성 값으로 된 결과를 출력하게 되고, "Elements"를 추가하면 Column명이 요소 값으로 된 결과를 출력하게 된다 [4.6,7,8].

다음으로 3단계에서 선택된 테이블에서 Column들을 리스트하기 위한 XML문서의 코드는 다음과 같다.

```
<?xml version='1.0' encoding='EUC-KR'?>
<root xmlns:sql='urn:schemas-microsoft-com:xml-sql'>
  <sql:header>
    <sql:param name='TableName'>
      <Information_Schema.Tables/></sql:param>
    <sql:param name='OrdinalPosition'>
      <Information_Schema.Columns/></sql:param>
    </sql:header>
    <sql:query >
      SELECT Table_Name, Ordinal_Position, Column_Name,
             Data_Type, Is_Nullable
      FROM Information_Schema.Columns
      WHERE Table_Name=@TableName AND
            (Ordinal_Position in (@OrdinalPosition))
      // { UNION ALL
      // 테이블이 2개 이상 선택되었을 경우 위와 같은
      // SELECT ~ WHERE절이 반복됨.
      // }
      FOR XML AUTO, ELEMENTS
    </sql:query>
  </root>
```

위의 코드에서 "@TableName"은 2단계에서 선택된 테이블의 이름을 받아들이기 위한 매개변수로 header부분에 파라미터로 선언 되어 있고, "@OrdinalPosition"은 선택된 테이블 내에서 존재하는 모든 Column들의 순서를 나타내는 Ordinal_Position으로부터 선택된 Column들의 순서만을 받아들이기 위한 매개변수이다.

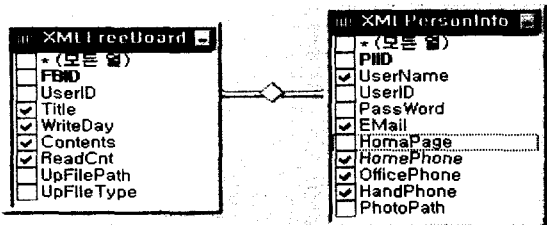
이상의 3단계를 거치면서 다음과 같은 XML문서 내에 포함될 질의어를 생성할 수 있다.

```
<?xml version='1.0' encoding='EUC-KR'?>
<root xmlns:sql='urn:schemas-microsoft-com:xml-sql'>
  <sql:query >
    SELECT TableName_(1..n).ColumnName_(1..n)
    FROM TableName_1, ... ,TableName_n
    WHERE TableName_1.UserID=TableName_n.UserID
    FOR XML Auto, Elements
  </sql:query>
</root>
```

위의 코드에서 SELECT문 다음에는 3단계에서 선택된 Column들이 나열되고, FROM절 다음에는 2단계에서 선택된 Table들이 나열된다. WHERE절 다음에는 3단계에서 설정한 테이블들 간의 관계를 나타내는 조건이 기술된다.

4.2 XML 스키마 생성을 위한 설계

<그림 2>와 같이 설계된 데이터베이스를 이용하여 XML스키마 생성을 위해 <표 2>의 4단계부터 6단계까지의 정책을 거치면서 <표 3>과 같은 결과를 얻게 된다.



<그림 2> ER-Diagram

<표 3> 6단계 정책에 의해 추출된 결과 테이블

| Table Name | Column Name | Element/Attribute | Parents Element | Level | Ordin | Alias | Data Type | Null |
|------------|-------------|-------------------|-----------------|-------|-------|-------|-----------|------|
| Board | Title | E | Table_N | 1 | 1 | 타이틀 | string | No |
| Board | Content | E | Title | 2 | 1 | 내용 | string | No |
| Board | W_Day | E | Table_N | 1 | 3 | 작성일 | datetime | No |
| Board | R_CNT | E | Table_N | 1 | 4 | 조회 | i4 | No |
| P_Info | User | E | Table_N | 1 | 2 | 작성자 | string | No |
| P_Info | E-Mail | E | User | 2 | 1 | 이메일 | string | Yes |
| | Phone | VE | User | 2 | 2 | 전화 | | No |
| P_Info | Hom_P | E | Phone | 3 | 1 | 집 | string | Yes |
| P_Info | Offi_P | E | Phone | 3 | 2 | 직장 | string | Yes |
| P_Info | Cell_P | E | Phone | 3 | 3 | 휴대폰 | string | Yes |

<표 3>에서 얻은 결과를 토대로 XML 스키마를 생성하게 된다. "Element/Attribute" 항목에는 E, A, VE, VA가 올 수 있는데, VE나 VA는 실제로 데이터베이스에는 없지만 <group>을 위해 생성된 것을 의미하며, Ordin 항목은 <complexType>의 <sequence>를 위해 사용된다. 또한, Null 항목의 "Yes"나 "No"는 minOccurs가 "0" 또는 "1"을 나타내기 위해 사용된다. 이렇게 생성된 XML 스키마는 XML문서의 구조를 갖게 되고, xpath-query mapping-schema를 통해 원하는 구조를 갖는 XML문서를 생성할 수 있게 된다.

5. 결 론

최근 웹 서비스에 대한 사용자의 요구가 다양하고, 이미 개발된 콘텐츠에 대한 Life Cycle이 짧아지고 있는 상황에서, XML기반의 웹서비스는 필수적일 것이다. 또한 사용자의 잦은 기호 및 트렌드의 변화는 기존 문서에 대한 구조 변경이 쉽고, 빠른 기간 안에 새로운 문서를 생성해 낼 수 있어야 한다.

본 논문에서는 XML문서나 XML 스키마가 없는 상태에서 관계형 데이터베이스를 이용하여 단계적인 절차에 의해 XML 문서 생성을 위한 질의어와 XML 스키마를 생성하고, 이를 통해 XML 문서를 생성할 수 있는 방법을 제시하였다.

향후 연구과제로는 XML 스키마를 기반으로 각 클라이언트에 관한 정보 테이블을 이용한 XML 스타일시트를 자동 생성할 수 있는 연구가 필요하다.

참고 문헌

- [1] <http://www.w3c.org/XML>
- [2] H.M. Deitel et al, "XML How To Program", Prentice Hall, 2001
- [3] Kevin Williams et al, "Professional XML Databases", Worx, 2001
- [4] msdn online(XML), <http://msdn.microsoft.com/xml>
- [5] <http://www.xmlgo.net>
- [6] <http://www.mssqllaboratory.pe.kr>
- [7] <http://myhome.naver.com/xmlmanager/>
- [8] <http://mie.paichai.ac.kr/users/Frog0214/workshop/>