

C#을 이용한 XML과 관계형 데이터베이스 매핑 모듈 개발

정승훈*, 고재진*, 권순덕*, 안형근*, 양상석*
*울산대학교 컴퓨터 정보통신 공학부
e-mail:noru37@mail.ulsan.ac.kr

Module Development for mapping between XML and Relational Database by using C#

Seung-hoon Jung*, Jaejin Koh*, Sundeok Kwon*,
Hyoungkeun Ahn*, Sangseok Yang*

*Dept of Computer Engineering & Information Technology,
University of Ulsan

요 약

인터넷의 사용 증가로 정보의 양은 기하급수적으로 증가하고 있으며 웹 데이터의 표준인 XML의 데이터 표현의 유연성으로 인해 EDMS(Electronic Document Management System), ebXML(e-business eXtensible Markup Language)등 웹 기반의 전자문서를 이용하는 XML를 문서 교환 및 표준 문서 형식으로 도입하고 있는 실정이다. 이런 XML문서들을 저장관리 하기 위해 XML전용 Database도 현재 많이 연구되고 있다. 하지만 이런 XML전용 Database들은 관계형 데이터베이스보다 실용성이 떨어지는 점들이 많아 XML과 관계형 데이터베이스 매핑에 대한 연구도 활발히 진행중이다. 본 논문에서는 C#을 이용하여 XML과 관계형 데이터베이스 간의 매핑을 위한 모듈을 개발하였으며, XML의 element와 관계형 데이터베이스의 필드를 상호 매핑시켜 줌으로 인해 XML문서의 element를 정확하게 저장, 관리 할 수 있다.

1. 서 론

인터넷의 발전이 진전되면서 이 기종간의 시스템에서 작성된 문서에 대한 데이터베이스의 구축과 검색 그리고 상호 교환의 중요성이 높아지고 있다. 이에 따라, 다양한 형식으로부터 원하는 정보를 효율적으로 관리, 공유하기 위해서는 문서를 일관성 있게 구조화하는 기술의 필요성이 대두되었고, 1996년 ISO(International Organization for Standardization)에서는 SGML(Standard Generalized Markup Language)이라는 문서의 논리 구조를 표현하는 표준안을 마련했다[1].

하지만 SGML은 다양한 기능에도 불구하고, 그 구성이 너무 복잡하다는 단점을 가지고 있었으며, 이를 해결하고자 HTML(Hypertext Markup Language)이 제기되었지만 이는 제한된 태그라는 한계를 가지고 있어 일관성있게 구조화하는 용도로는 부적당하였다. 이에 따라 W3C에서는 일반화된 마크업(Generalized Markup), 복합구조(Complex Structure), 검증(Validation)의 특성을 그대로 지닌

하는 한편 사용자에 의한 확장성(Extensibility)을 가지고 있는 XML(eXtensible Markup Language)을 제안하였다[2][3]. 그래서, 최근의 웹(Web) 또는 디지털 전자 도서관 시스템, 수확분야, 채널 기술의 CDF(Channel Definition Format), 이동 통신에서의 HDML(Handheld Device Markup Language)들과 같은 환경에서 많은 문서들이 XML 마크업 언어를 적극 활용하고 있으며 이러한 언어로 문서들을 표현함으로써 문서의 논리적인 구조를 표현할 수 있고, 그럼으로써 문서들을 데이터베이스에 저장하고 검색할 수 있는 필요성이 대두되고 있다[4].

XML 전용 Database의 경우 관계형 데이터베이스에 비하여 XML문서 안의 데이터들을 관리하기가 힘들며 Query를 할 경우 시간이 많이 소요되며, Database에서 XML문서를 재구성시에도 효율적이지 못하다.

본 논문에서는 먼저 XML 접근 방식에 따른 DBMS 유형과 C#을 이용한 ADO.NET에서의 Database 관리에 대하여 알아보고, C#을 이용한 XML과 관계형

데이터베이스사이 매핑기법을 제안하고, 향후 연구 과제를 제시하고자 한다.

2. 관련 연구

2.1. XML 문서의 분류

2.1.1. 데이터 중심 문서 (data-centric document)

XML 문서 중에 데이터 중심 문서 (data-centric document)는 조직적이고 정규화된 구조를 갖고 있으며, 혼합된 콘텐츠가 거의 없다. 기계적인 처리가 가능하며, 보통 관계형 데이터베이스에 저장한다. 데이터 중심 문서 (data-centric document)는 정형화된 데이터들을 포함하고 있는 문서로 상품 주문서, 주식 데이터, 과학 데이터 등이다.

데이터 중심 문서는 상대적으로 쉽게 데이터베이스에 저장 가능하며, 미들웨어(middleware)를 이용하여 자동적으로 저장될 수 있다. 데이터 중심 문서는 정형화된 데이터를 가지며, 데이터들 간의 순서가 중요하지 않다. 메타데이터가 없으며, 기계적으로 처리하기 쉬운 특징을 가지고 있다. XML 문서가 데이터베이스에 저장될 때는 문서의 메타 정보들은 모두 제거되며, 문서의 DTD 정보는 저장되지 않는다. 물리적 구조에 관한 정보들이 저장되지 않으며, 엔터티 참조와 일반 데이터는 데이터베이스에 저장될 때 구별이 없어진다. 데이터베이스에 저장시 엘리먼트와 속성의 차이가 없어지게 된다.

2.1.2. 문서중심문서(document-centric document)

문서 중심 문서 (document-centric document)는 데이터 중심 문서보다 좀 더 자유로운 형식의 텍스트가 자유롭게 표현된 문서이다. 데이터 중심 문서에 비해 비정형적이며, XML 전용 데이터베이스에 저장한다. 데이터 중심 문서가 기계적인 처리가 쉬운데 비해 문서 중심 문서는 보통 사람 중심적이고, 혼합된 콘텐츠를 충분히 사용하므로 정규화된 구조가 작다. 책, 전자메일, 광고 등이 문서 중심 문서라 할 수 있다.

2.2 XML 접근 방식에 따른 DBMS 유형

XML 데이터를 DBMS에 저장하는 가장 일반적인 접근 방법은 복잡한 프로그램 기술과 파서를 이용하여 XML의 엘리먼트들을 테이블과 칼럼의 형태로 매핑하는 것이다. XML 데이터를 데이터베이스에 저장할 때마다 DBMS는 XML 데이터를 분해해서 분해된 컴포넌트들을 각각 적절한 테이블에 저장하고, 질의할 때는 다시 분해된 컴포넌트들을 결합하여 XML 문서로 만들어 낸다. 이 과정을 통해 나타난 결과가 XML을 처리하는데 큰 문제가 없어 보이지만, XML을 분해, 변환하는 과정에서 XML문서의 원래 구조를 잃어버리는 등의 높은 신뢰성과 성능에 대해서는 보장할 수 없게 된다. 이 장에서는 C#을 이용한 상호매핑 모듈을 제안하기 앞서 구조적 스키마를 가진 XML의 변환을 위한 대표적인 방법으로 관계형 데이터베이스 매핑형, 객체지향형, Native XML형 데이터베이스의 특성에 대해 살펴보고 DBMS와 XML 사이

의 상호 매핑, 데이터 도큐먼트 비교하겠다.

2.2.1 관계형 데이터베이스 매핑형

관계형 데이터베이스는 역사가 오래 되었고 많은 응용프로그램이 개발되어 있어 기존의 많은 데이터가 관계형 데이터베이스 형태로 유지 관리되고 있다. 현재에는 대부분의 많은 관계형 데이터베이스 시스템들 역시 XML 문서를 지원하기 시작하였으며 저장 방법과 관련한 많은 연구가 이루어지고 있다. 그러나 관계형 데이터베이스로는 XML 문서를 표현하는 데에는 여러 가지 한계점이 따른다. 첫째로, 모든 구조를 테이블로 표현해야 하는 관계형 데이터베이스에서는 XML을 분해, 변환하는 과정에서 XML 문서의 원래 구조를 잃어버린다는 것이다. XML문서에서 중요한 의미를 지니는 엘리먼트의 순서라든가, 애트리뷰트의 정보라든가, 부모-자식 Node의 관계 등에 대한 정보를 테이블로서는 구현하기가 매우 까다로워서, 단순하게 엘리먼트나 애트리뷰트 값만 저장하게 된다. 따라서 저장된 데이터를 다시 읽어왔을 때 원래 문서의 구조가 그대로 복원되지 못할 가능성이 크다.

둘째는, 계층적인 구조의 XML 데이터를 여러 테이블과 칼럼에 나누어서 저장해야 하는데 오는 문제점인데, 먼저 XML 데이터에 대한 질의를 할 때에는 여러 테이블과 관련된 복잡한 조인이 필요하다. XML 데이터를 검색하기 위한 평범한 질의를 위해서도 관계형 데이터베이스는 여러 번의 조인 질의를 해야 한다.

셋째는, XML 데이터를 입력, 수정, 삭제하기 위해서는 해당 XML 문서를 다른 사용자가 접근하지 못하도록 하기 위해 관계된 모든 레코드에 락(Lock)을 걸어야 한다. 이런 이유로 관계형 데이터베이스에서는 사용자에 대한 접근 관리기능과 XML의 문서집합, XML 문서, 엘리먼트에 대한 접근 관리기능이 필요하지만 접근제어 단위(Granularity)가 작으면 작을 수록 관리 비용이 많이 들며, 접근제어의 단위가 클수록 실제의 트랜잭션 처리 시간이 길어지는 경향이 있다. 또한 다수의 사용자와 트랜잭션 환경에서 복잡한 XML 데이터를 다루어야 한다면 이런 조인과 락(Lock)은 데이터베이스의 성능을 현저히 떨어뜨릴 수도 있다.

결과적으로 XML의 계층 구조를 관계형 데이터베이스의 테이블로 표현하고, XML의 각 엘리먼트의 값과 애트리뷰트를 테이블의 필드로 표현하게 된다. 이때 XML 문서 구조는 인덱스를 가지고 있으나 저장 단위가 테이블이기 때문에 실제로 데이터를 매핑시키기 위해서 조인 연산이 필요하게 된다. 실제 상용 DBMS에서는 이를 위해 기존 관계형 DBMS 위에 이를 위한 미들웨어를 제공해 준다.

2.2.2 객체 지향형

XML 데이터의 계층 구조를 객체 지향의 클래스 계층에 매핑시키는 타입으로 초기의 관계형 데이터베이스에서는 데이터의 자료구조와 애플리케이션의 자

료구조가 다르므로 중간에 상이한 두 자료구조간의 데이터 변환을 애플리케이션에서 해주었지만, 객체 지향 데이터베이스는 애플리케이션과 데이터베이스의 자료구조가 동일하므로 데이터 변환 같은 부가적인 작업이 필요 없게 되었다. 객체 지향 데이터베이스에 XML 문서를 저장할 경우 문서의 엘리먼트를 객체로 생각하여 비교적 쉽게 모델링할 수 있고, XML 문서와 DTD의 복잡한 구조 역시 객체지향 데이터베이스에서는 자연스럽게 유지될 수 있다.

또한 W3C에서 응용프로그램에 활용하기 위한 XML API의 표준으로서 권고하고 있는 DOM이 객체를 기반으로 하고 있어서 객체지향 데이터베이스로의 구현은 더욱 간단해진다. 그러나 객체지향 데이터베이스는 대량의 문서가 저장될 경우, 즉 객체가 많아질 경우 저장과 검색 효율이 크게 떨어지는 단점을 갖고 있다. 대부분의 객체지향 데이터베이스는 높은 관심과 주목을 받았지만 상업적으로는 크게 성공을 거두지는 못하고 있는 실정이다. 하지만 객체 지향 데이터베이스의 개념들은 다음에 나오는 Native XML 데이터베이스의 모태가 되었다고 할 수 있다.

2.2.3 Native XML 형

Native XML 데이터베이스는 XML에 대한 논리적 구조 형태로 저장 및 검색이 이루어지는 구조인데 XML 데이터를 입력받아 저장하고, 질의 결과를 XML로 출력하기에는 가장 적합하게 설계되어 있다. XML 구조에 따른 검색 지원, 부분 범위 추출 등 XML의 특성에 맞는 검색이 이루어 질 수 있다. XML 데이터베이스에는 기존의 관계형 데이터베이스에 XML기능을 추가시킨 XML-Enabled Database들은 일반 데이터베이스에 XML 데이터를 테이블과 컬럼으로 매핑하여 저장하고, 질의 결과를 다시 XML로 바꾸어 리턴하는 컨버터 기능을 추가하여 XML을 지원하는 형태로 발전해 나가고 있다. 반면에, Native XML Database는 XML 데이터를 있는 그대로 문서형태로 저장하기 때문에 컨버터 등의 기능을 필요로 하지 않아 XML을 다루기에 적합하다. 일반적인 Native XML Database의 장점은 다음과 같다.

첫째, 데이터베이스 설계 및 프로그래밍 유지보수가 용이하다. XML 데이터를 관계형 데이터베이스의 테이블에 저장하기 위해서는 정규화 과정을 거쳐 Primary-key 와 Foreign-key로 복잡하게 연결된 여러 개의 테이블을 설계해야 한다. 계층적 구조의 관계를 관계형 데이터베이스에 매핑하기 위해서는 복잡한 구조의 테이블을 설계해야 한다. Native XML Database는 별도의 정규화 과정을 거치는 테이블 설계가 필요 없기 때문에 설계가 쉬우며 XML 문서의 구조가 바뀌어도 데이터베이스 스키마의 변경이 쉬워 유지보수가 용이하다.

둘째, XML 데이터의 저장 및 질의 성능을 향상시킬 수 있는데, XML-Enabled Database들은 계층적 구조의 XML 문서를 여러 개의 테이블과 칼럼에 걸쳐 쪼개어 저장하고, 질의할 때 다시 이를 XML로

재조합해야하므로 XML의 구조가 복잡할수록 복잡한 조인 질의를 해야 하기 때문에 검색의 속도가 떨어진다. 또 XML 데이터의 각 엘리먼트, 애트리뷰트 값을 테이블의 칼럼에 쪼개어 저장하기 때문에 XML 문서의 원래 구조를 잃어버리는 등의 단점이 있다. 반면 Native XML DBMS는 XML 데이터를 있는 그대로 저장하기 때문에 변환과정이 필요 없어 복잡한 구조의 XML 문서도 빠른 속도로 처리 가능하며, XML 문서의 구조를 잃지 않고 저장한다.

셋째, XML 구조에 맞는 검색이 이루어지는데 XML 구조에 맞는 검색이 이루어지는데 XML-Enabled Database는 XML을 질의하기 위해서는 복잡한 SQL 문장을 만들어야 한다. Native XML Database는 Xpath 표준을 이용하여 질의하기 때문에 XML 구조에 맞는 자유로운 검색을 간단한 질의 문장으로 수행할 수 있다[5][6][7].

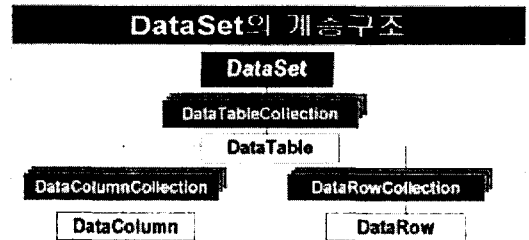
[표1] 관계형 데이터베이스와 Native XML 데이터베이스의 저장 비교

관계형 데이터베이스	Native XML 데이터베이스
·테이블의 정규화 과정이 필요	·변환 없이 저장하므로 저장시 부하가 없음
·XML 태그를 각 테이블의 원인으로 매핑	·XML 태그를 매핑하는 과정 불필요
·XML데이터가 테이블 형태로 분리되어 저장되므로 원래 형태의 복원이 어려움	
·저장하는 로직 구현이 복잡	

3. ADO.NET을 이용한 XML과 관계형 데이터베이스 상호 매핑 모듈 개발

ADO.NET은 닷넷에서 데이터베이스 조작에 관련된 닷넷 클래스들의 집합을 말하며, 그것을 통해서 다양한 방법으로 데이터베이스를 검색하고, 수정, 업데이트 등의 작업을 할 수 있습니다.

본 논문에서 사용할 DataSet의 계층구조는 그림2와 같다



[그림1] DataSet의 계층구조

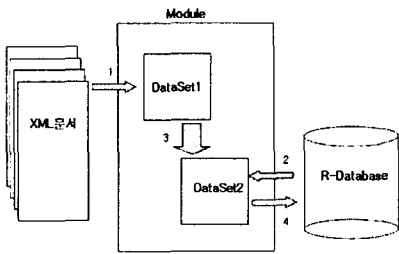
하나의 DataSet은 여러개의 DataTable을 가질 수 있으며, 각각의 DataTable은 여러개의 DataColumn과 DataRow을 가질 수 있는 Memory In Cash 형태 로 되어있다.

본 논문에서 제안하는 매핑방식은 XML 문서로 작성된 상품 주문서를 관계형 데이터베이스에 저장할 때 요구되는 매핑에 관련된 것이다.

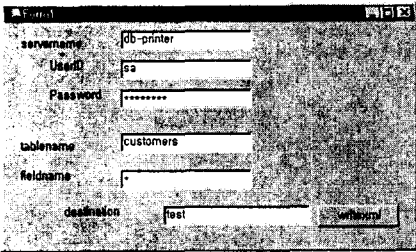
매핑 방식을 제안하기에 앞서 XML문서의 각 element들은 관계형 데이터베이스의 컬럼들과 이름과 같아야 된다는 제약을 가지고 제안을 하도록 하였다.

본 논문에서 제안하는 매핑은 아래와 같다.

- 1단계: XML문서를 DataSet1로 읽어들이는다.
- 2단계: DataSet1에 있는 DataColumn들을 기반으로 관계형 데이터베이스에서 같은 이름을 가지는 Column들을 가지고 DataSet2를 생성한다.
- 3단계: DataSet1에 있는 Data들을 DataSet2에 복사한다.
- 4단계: DataSet2의 table을 관계형 데이터베이스로 update한다.



[그림2] DataSet을 이용한 제안 모듈



[그림3] 모듈을 사용하여 C#으로 만든 Form

사용될 XML문서는 다음과 같다.

```
<customers>
<CustomerID>JUNG</CustomerID>
<CompanyName>University of Ulsan</CompanyName>
<ContactName>Mike</ContactName>
<ContactTitle>Sales Representative</ContactTitle>
<Address>Mosco Str. 57</Address>
<City>Ulsan</City>
<PostalCode>13209</PostalCode>
<Country>Korea</Country>
<Phone>001-0074321</Phone>
<Fax>030-00765</Fax>
</customers>
```

XML 데이터가 MS-SQL 2000에 저장된 모습은 다음과 같다.

[그림4] MS-SQL 2000에 저장된 XML 데이터

4. 결론 및 향후 과제

인터넷의 발전이 진전되면서 이 기종간의 시스템에서 작성된 문서에 대한 데이터베이스의 구축과 검색 그리고 상호 교환의 중요성이 높아지고 있다. 이에 따라 W3C에서는 일반화된 마크업, 복합구조, 검증의 특성을 그대로 지원하는 한편 사용자에 의한 확장성을 가지고 있는 XML(eXtensible Markup Language)을 제안하였다.

그래서, 최근의 웹(Web) 또는 디지털 전자 도서관 시스템, 수학분야, 채널 기술의 CDF(Channel Definition Format), 이동 통신에서의 HDML(Handheld Markup Language)들과 같은 환경에서 많은 문서들이 XML 마크업 언어를 적극 활용하고 있으며 이러한 언어로 문서들을 표현함으로써 문서의 논리적인 구조를 표현할 수 있고, 그럼으로써 문서들을 데이터베이스에 저장하고 검색할 수 있는 필요성이 대두되고 있다.

본 논문에서 제안한 모듈은 XML 문서 중에서 데이터 중심 문서를 관계형 데이터베이스에 저장할 때 ADO.NET의 DataSet을 2개 사용함으로써 보다 쉽게 매핑을 하도록 하였다.

향후 연구 과제로는 관계형 데이터베이스에서 중요한 Primary key, Foreign key등의 제약 조건을 고려하여 여러 개의 table이 조인되어 있는 데이터들을 XML문서에서 관계형 데이터베이스로 저장하는 모듈을 개발하는 것이다.

5. 참고 문헌

- [1] 손정환, 이희주, 장재우, 심부성, 주종철 "구조화된 문서를 위한 정보검색시스템의 설계 및 구현", '98 동계 데이터베이스 학술대회 논문집 제14권 1호 P102-106, 1998
- [2] 연재민, 장동준, 김용훈, 이광찬, 이규철. "효율적인 검색 지원 SGML 저장 관리기의 설계 및 구현", '99 한국 데이터베이스 학술대회 논문집 15권 1호, p136-143
- [3] Charles L. A. Clarke, Gordon V. Cormack, Forbes J. Burkowski "An Algebra for Structured Text Search and a Framework for its Implementation". The Computer Journal 38(1), p43-56, 1995
- [4] Francois, "Generalized SGML repositories: Requirements and Modeling", Computer Standards & Interfaces, 1996
- [5] 장희철 "관계형 스키마에서 XML DTD로의 변환에 관한 연구", '2003 연세대학교 석사 논문
- [6] Ronald Bourret <http://www.rpbourret.com/xml/6LandDatabases.htm>
- [7] Fachhochschule Wiesbaden, Fachbereich06Informatik, <http://www.informatik.th-wiesbaden.de/~turau/DB2XML/rules.html>