

내장형 XML 데이터베이스에 관한 연구

곽길신*, 최문영*, 주경수*

*순천향대학교 전산학부

e-mail:phnewma@hotmail.com, griffin@hyejeon.ac.kr, gsoojoo@sch.ac.kr

A Study on Embedded XML Database

Kwak Kil-Sin*, Choi Mun-Young*, Joo Kyung-Soo*

*Dept. of Computer Science and Engineering, College of Engineering
SoonChunHyang University, Asan 336-745, Korea

요 약

XML 데이터베이스와 관련된 연구는 세계의 여러 곳에서 활발하게 이루어지고 있다. 현재 이 분야의 연구들은 주로 기존 데이터베이스 기술을 XML 관리에 활용하는 것들이다. 이러한 추세들이 결합하여 PDA 등 소규모 정보 단말기에서도 독자적으로 XML 문서를 관리하고, 경우에 따라 서버에서 관리되는 데이터를 가져와 처리하려는 요구가 커지고 있다. 따라서, 이러한 요구에 적절히 대응하기 위해서는 내장형 XML 데이터베이스 시스템의 개발이 필요하다. 그러므로 본 논문에서는 내장형 XML 데이터베이스를 조사하고 분석하였다.

1. 서론

컴퓨터와 무선 통신 기술의 발전으로 이동 컴퓨팅 애플리케이션들이 주목을 받고 있다. 컴퓨터와 무선 통신 기술이 효과적으로 연계되어 이동 환경에서 정보 교환 및 처리할 수 있게 된 것이다. 최근 이러한 이동 환경에서 활용되고 있는 정보 단말기들의 성능이 크게 향상됨에 따라 다양한 무선 인터넷 애플리케이션들이 가능해졌다. 이에 따라 정보 단말기 자체에도 데이터베이스 시스템이 탑재되고 있다[1].

한편, 인터넷 환경에서 XML 문서의 사용이 크게 늘어나면서, XML 문서를 데이터베이스에 저장하는 연구가 활발하게 진행되고 있다[2,3,4]. 이 때 XML 문서는 주로 관계형 데이터베이스, 객체지향 데이터베이스, 전용 파일시스템 등을 이용하여 관리된다[5,6,7].

최근 이런 추세에 따라 PDA와 같은 소규모 정보 단말기에서도 독자적으로 XML 문서를 관리하며, 경우에 따라 서버에서 관리되는 데이터를 가져와 처리하려는 요구가 커지고 있다. 이러한 요구에 적절히 대응하기 위해서는 내장형 XML 데이터베이스 시스템의 개발이 필요하다.

내장형 시스템을 무선 환경에서 안정적으로 사용하기 위해서는 몇 가지 해결해야 할 문제가 있다. 먼저,

무선 통신 환경은 유선 통신 환경보다 전송 속도가 느리고 접속이 불안하여 안정적인 데이터 전송이 보장되지 않는다. 또한, 소형, 경량, 저전력, 소용량 메모리와 저성능의 처리 장치에서 소프트웨어를 구동해야 한다. 이러한 제약을 갖는 내장형 시스템에서 내장형 XML 데이터베이스 시스템이 사용되기 위해서 데이터베이스 시스템은 높은 이식성, 적은 시스템 리소스 사용, 유연한 확장성, XML 문서 관리에 용이한 구조의 지원 등이 필요하다. 이에 따라 본 논문에서는 내장형 XML 데이터베이스인 Birdstep RDM XML과 eXist 그리고 Berkeley DB XML를 분석하여 서술하였다.

2. 내장형 시스템과 운영체제

2.1 내장형 시스템

지난 몇 년 동안 컴퓨터 시장에서는 하드웨어보다는 소프트웨어 쪽으로 많은 발전을 해왔다. 앞으로의 정보가전 산업 역시 단순한 하드웨어적인 발전보다 우수하고 다양한 서비스를 제공하는 소프트웨어의 개발에 따라 제품의 경쟁력이 생길 것이다.

내장형 시스템은 제한된 자원을 갖고 특정한 목적을 갖는 작업을 처리하기 위한 시스템이다. 이러한 내장형 시스템은 산업분야, 가전분야, 사무분야, 군사용

등 다양한 응용분야가 있다.

내장형 시스템은 시간이 흐를수록 기능이 다양해지고 요구되는 시스템의 규모가 커짐에 따라 이를 효과적으로 운용하기 위해서 적절히 통제할 운영체제가 필요하게 되었다.

2.2 내장형 운영체제

내장형 시스템에서의 운영체제는 시스템의 규모가 커짐에 따라서 Multi Tasking과 같은 복잡한 기능을 요구하고 있으며, TCP/IP, GUI, Audio, Video 등과 같은 네트워크나 멀티미디어는 시스템의 기본으로 자리잡고 있다. 그러나 단순히 순차적으로 운영되는 프로그램만으로 시스템을 운영하기 어렵다. 따라서 내장형 시스템에도 운영체제의 개념이 필요하게 되었다.

pSOS, VxWorks, VRTX 와 같은 운영체제는 실시간 처리 기능을 제고하는데 목적을 두고 신뢰성에 중점을 두며, 범용성이 아닌 주로 한 가지 특수한 목적에 최적화된 시스템 운영체제이다. 그러나 최근 내장형 시스템이 네트워크 접속, 멀티미디어 처리 등의 기능이 요구됨에 따라 운영체제도 점차 범용 운영체제의 기능에 가까워지고 있으며, 그에 따른 효율적인 내장형 데이터베이스가 필요하게 되었다.

3. 내장형 데이터베이스

내장형 시스템은 제한된 자원을 갖기 때문에 데이터베이스를 설계할 때 다음과 같은 사항을 고려해야 한다.

- Small footprint (작은 메모리 사용량)
- Robustness (견고성)
- Easy maintenance (쉬운 관리)

내장형 데이터베이스는 내장형 시스템에 이식되어야 하기 때문에 세 가지 요구 사항 중에서 쉬운 관리와 견고성이 가장 큰 관심사이다.

내장형 데이터베이스는 사용할 애플리케이션 또는 시스템에서 직접적으로 실행된다. 모든 논리적인 데이터 관리가 애플리케이션의 주소에 링크되기 때문에 각각의 데이터베이스 서버를 필요로 하지 않는다. 그러나 기존 데이터베이스 시스템이 갖는 데이터 관리 서비스와 트랜잭션, 회복, 병행성의 기능을 사용하기 위한 API를 제공하므로 개발자는 이를 사용하여 신뢰성 있는 데이터베이스를 구축 할 수 있다. 또한, 사용자는 데이터베이스에 직접 접근할 필요가 거의 없기 때문에 사용자 질의 틀은 고려하지 않는다. 따라서

개발자는 사용자의 필요에 맞게 인터페이스를 고려하여 애플리케이션을 디자인하고 개발해야 한다. 또한, 데이터베이스의 오퍼레이션이 애플리케이션 프로세스 안에서 실행되기 때문에 클라이언트/서버 시스템보다 처리 속도가 빠르며 분산된 서버가 없기 때문에 사용자는 백업이나 오류 복구에서 자유롭다.

개발자는 클라이언트/서버 데이터베이스보다 좀 더 많은 코드를 작성해야 하고 애플리케이션을 디자인할 때 저장, 검색, 처리해야 할 데이터를 좀더 주의 깊게 고려해야 한다.

내장형 데이터베이스는 많은 양의 데이터 처리, 동시 접근, 시스템 오류에 대한 안전성, 빠른 데이터 처리에 관한 부분은 독립형 데이터베이스에서와 같이 잘 동작한다.

4. 내장형 XML 데이터베이스

내장형이나 클라이언트/서버 시스템의 데이터베이스는 모두 구조적인 것이다. 따라서 내장형 시스템도 클라이언트/서버 시스템과 같이 XML 데이터를 데이터베이스로 사용할 수 있다.

XML은 서로 다른 시스템에서 정보 처리 상호 운용을 가능하게 한다. 따라서 유선뿐 아니라 무선 환경에서도 정보처리 상호 운용을 쉽게 하기 위한 서비스를 제공하기 위해서 XML 데이터베이스를 사용하여 신용성과 안정성을 해결할 수 있다.

개발자들이 새로운 코드를 작성하고, 사용자가 XML 저장소에 직접 접근할 필요가 없기 때문에 애플리케이션 안에 XML 데이터베이스를 사용하면 새로운 소프트웨어의 인스톨과 관리를 쉽게 해주고, 성능을 향상시킬 수도 있다.

클라이언트/서버 혹은 내장형에서 XML 데이터베이스를 사용하면 XML 데이터의 관리, 애플리케이션의 데이터 모델과 데이터베이스의 엔진사이의 근접한 일치, 추가적인 XML 스키마의 실행에서 유리하다. 클라이언트/서버제품에 비교하여 내장형 XML 데이터베이스를 사용하면 수행 능력의 향상과 사용자에 의한 쉬운 관리, 애플리케이션에 맞는 인스톨과 관리, 높은 신뢰성, 작은 메모리 사용에서 유리한 모습을 보인다.

4.1 Birdstep RDM XML

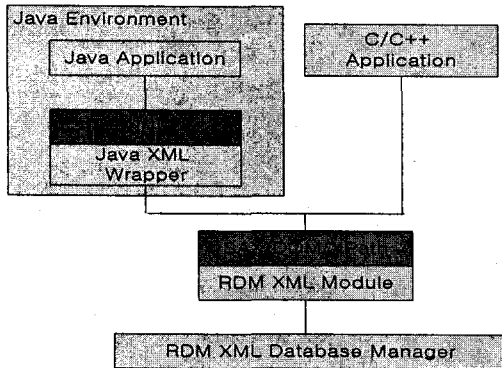
RDM XML은 PDA나 핸드폰과 같은 소형화된 모바일 애플리케이션이 목적으로 하는 요소들을 만족 시키며, 진보된 객체 지향과 효과적인 저장, 검색 모두 만족시키는 계층적인 데이터베이스를 지향하고 있다.

XML 도큐먼트는 신속한 검색을 위해 계층적인 노드 구조로 저장된다. 모든 Well-formed XML 도큐먼트는 데이터베이스에 저장될 수 있다. 또한, RDM XML 은 DTD와 XML 네임스페이스를 지원한다.

RDM XML은 산업 표준의 DOM, SAX, XPath 애플리케이션 프로그램 인터페이스를 제품 생산과 개발 프로세스의 편리함을 위하여 제공한다.

<표 1> Birdstep RDM XML의 특징

특징	비고
계층구조 데이터 표현	어떤 XML 문서도 저장가능
SAX 인터페이스 지원	XML 데이터의 import/export 가능
DOM 인터페이스 지원	데이터베이스내의 XML노드를 검색, 관리
XPATH 지원	XML path language에 특정한 query 지원
트랜잭션 처리	ACID를 모두 지원
다중 운영체제 지원	WinCE, PalmOS를 포함한 대부분의 wireless OS 지원
Java 지원	Java 애플리케이션을 이용하여 크기, 성능, 속도를 최적화



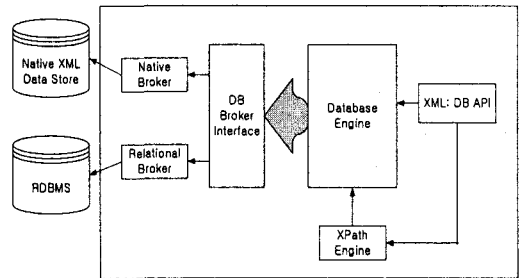
(그림 1) Birdstep RDM XML의 구조

4.2 eXist

eXist는 XPath 쿼리 진행, 키워드 검색의 확장, XUpdate 지원과 간결한 XML 개발 툴이 포함된 효과적인 공개 native XML 데이터베이스 시스템이다. 데이터베이스는 가법고 자바로 구성되었으며 다른 애플리케이션에 직접 내장되는 서블릿 엔진에서 독립적으로 실행되는 다른 방법보다 개발하기 쉽다. 여기서 native XML 데이터베이스란, XML로 저장되는 데이터베이스이다. 즉 XML 문서를 위한 DOM이나 SAX 같은 논리적 모델을 정의한 후, 그 모델에 따라 문서를 저장하고 추출하는 시스템을 말한다.

eXist에서 사용하는 저장시스템은 크게 두 종류가 있다. 첫 번째는 XML 문서만을 위해 특화된 시스템으로 Java로 짜여 있고, 기존의 관계형 데이터베이스를 이용하지 않으므로, 추가적인 데이터베이스 설치가 필요하지 않다. 두 번째는 관계형 데이터베이스 저장시스템으로 MySQL, PostgreSQL, Oracle을 지원한다.

그림 2는 eXist의 구조이다. 여기서 XPath Engine은 사용자로부터 넘겨받은 XPath 질의를 저장시스템에 맞게 변환 해주는 역할을 한다. 따라서 사용자는 저장시스템으로 native XML 데이터 저장시스템을 사용했는지 혹은, RDBMS를 사용했는지에 관계없이 명확하게 질의를 할 수 있다. DB Broker Interface는 시스템이 Native XML 저장장치를 쓸지, RDBMS를 쓸 것인지에 대한 인터페이스를 제공한다.



(그림 2) eXist의 구조

eXist의 특징에 대해 살펴보면, 우선 eXist는 문서를 collection 단위로 처리할 수 있게 한다. 즉 eXist는 문서를 관리할 때, UNIX 시스템에서의 디렉터리와 같이 문서들 간의 계층 구조를 고려하여 하나의 collection을 만들어 관리한다. 이때, 한 collection에는 서로 다른 스키마나 문서 타입을 가진 문서가 포함될 수 있다. 다음 특징으로는 인덱스 기반의 질의 처리를 들 수 있다. 이와 같은 처리로 인해, 부모와 자식 혹은 조상과 자식과 같은 관계형 구조를 쉽게 판명할 수 있다. 또한, eXist는 키워드 검색 시, 텍스트뿐만 아니라 문서 내의 속성도 검색할 수 있게 한다.

4.3 버클리 DB XML

버클리 DB 시스템은 오픈 소스 내장형 DB 엔진으로 Window NT, Unix, Linux 등 대부분의 OS에 이식이 가능하고, 작고 빠르며 사용자가 이해하고 사용하기 쉬운 신뢰성 있는 시스템이다. 이 시스템은 키/데이터 값을 한 쌍으로 고성능의 병행 저장과 검색을 요구하는 애플리케이션에 사용할 수 있도록 응용 프

로그래밍과 직접 연결될 수 있는 라이브러리를 제공하고 있다. 현재 C, C++, Perl, Tcl, Java 등의 다양한 프로그램 언어들에서 사용될 수 있도록 해당 인터페이스 접속 장치를 제공하고 있다.

버클리 DB 시스템은 처음에 새로운 해쉬(Hash) 접근 방법을 구현할 목적으로 1990년에 Switzer와 Yigit가 Hash 라는 패키지로서 만들었다. 이 이후 중복데이터 항목을 다룰 수 있는 중요한 확장과 함께 병행 접근과 트랜잭션을 지원하고 롤백(Rollback) 기능도 지원하게 되었다. 그리고 이 시스템은 Small Footprint를 가지며, 새로운 레코드를 삽입하는데 불과 몇 줄의 코드만으로 수행할 수 있게 하고, 많은 병행 사용들에 대해서도 정확하게 동작한다. 심지어 하드디스크의 손실과 같은 심각한 오류가 발생하더라도 올바르게 동작하도록 설계되었다.

버클리 DB 시스템에서 모든 데이터는 '파일명.확장자'와 같은 파일로 관리된다. 이때 모든 기본적인 파일 구조는 고정 및 가변 길이 레코드 형식을 가지며, 이를 B+ 트리와 해쉬 구조로 사용할 수도 있다. 무엇보다 버클리 DB 시스템은 데이터를 저장하기 위해서 단순히 키/데이터의 쌍으로 구성되어 있는 레코드를 자료 구조로 사용하고 있어서 관계형 데이터베이스에서의 테이블, 객체지향 데이터베이스에서의 클래스와 같은 중간 데이터 모델이 없는 단순한 자료 구조를 가진다.

여기서 키(Key) 필드는 레코드를 유일하게 식별할 수 있는 값으로 시스템이 관리하게 되며, 데이터(Data) 필드는 저장하고자 하는 값으로 길이에 제한이 없으며 어떠한 내용도 가능하다. 따라서 이러한 버클리 DB 시스템을 이용하여 XML 데이터를 저장하고 관리하기 위해서는 사용자가 적합한 자료 구조를 단순 키/데이터의 쌍으로 이루어진 레코드 구조 위에 직접 개발하여 사용해야 한다.

버클리 DB XML은 버클리 DB위에 native XML 데이터를 관리하기 위해 특수한 용도로 구축된 데이터 관리 엔진으로 빠른 속도, 신뢰성, 저렴한 비용의 효율적인 기억 장치, native XML 데이터, 세미 구조 데이터에 대한 검색을 제공한다. 버클리 DB XML은 컬렉션에 XML 문서를 저장하기 때문에 애플리케이션에서 많은 컬렉션이 동작하고, 서로 다른 컬렉션과 쉽게 데이터가 결합된다.

버클리 DB XML은 기억 장치와 규격화된 XML 문서의 검색에 대하여 명확하게 설계된 내장형 데이터베이스이다. 또한, XPath를 사용한 많은 XML 문서의 능률적인 변환에 대한 기능도 있다. XPath는 XML

문서의 조사와 부분적인 검색에 대하여 디자인된 질의 언어이다.

5. 결론

최근 인터넷 환경에서 XML 문서의 사용이 크게 늘어나면서, XML 문서 데이터베이스에 대한 연구가 활발하다. 이때 주로 XML 문서는 관계형 데이터베이스, 객체지향 데이터베이스, 전용파일 시스템 등을 이용하여 관리되는데, 이런 추세들이 결합되어 PDA 등 소규모 정보 단말기에서도 독자적으로 XML 문서를 관리하고, 경우에 따라 서버에서 관리되는 데이터를 가져와 처리하려는 요구가 커지고 있다. 따라서, 이러한 요구에 적절히 대응하기 위해서는 내장형 XML 데이터베이스 시스템의 개발이 필요하다. 그러므로 본 논문에서는 내장형 XML 데이터베이스를 조사하고 분석하여 기술하였다.

참고문헌

- [1] Mizi Research, <http://www.mizi.com>.
- [2] M. Olson, K. Bostic, and M. Seltzer, "Berkeley DB," Proc. of the 1999 Summer Usenix Technical Conference, Monterey, California, June 1999.
- [3] G. Bex, "simpleDB: a simple embedded database", <http://alpha.luc.ac.be/~gjb/simpleDB/>, January 2001.
- [4] J. McHugh, S. Abiteboul, R. Goldman, D. Quass, and J. Widom, "Lore : A Database Management System for Semistructured Data," Technical Report, Stanford University, Database Group, February 1997.
- [5] D. Florescu and D. Kossman, "Storing and Querying XML Data using as RDBMS," Bulletin of the Technical Committee on Data Engineering, Vol. 22, No.3, 1999.
- [6] J. Shanmugasundaram, K. Tuftte, C. Zhang, G. He, D. DeWitt, and J. Naughon, "Relational Database for Querying XML Documents : Limitations and Opportunities," Proc. of 25th Int'l Conf. on VLDB, Edinburgh, Scotland, UK, 1999.
- [7] R. Bourret, "Mapping DTDs to Databases," <http://www.xml.com/pub/a/2001/05/09/dtdtodbs.html>.