

관계형 DBMS를 이용한 XML 스키마 기반의 XML DBMS 설계

이상태*, 임중선**, 주경수***

Design a XML DBMS using RDBMS Based on XML Schema

Sang-Tae Lee*, Jong-Seon Lim**, Kyung-Soo Joo***

요약

XML DTD는 XML 문서에 대한 단순한 형태의 표현을 정의하고 있어서 XML 문서의 구조를 정의하는데 많은 어려움이 있다. 그래서 XML DTD의 문제점을 해결하기 위하여 W3C에서는 2001년 XML 스키마를 표준안으로 채택하여 사용을 권고하고 있다. 본 논문에서는 기존 XML DTD를 이용한 XML DBMS의 단점을 극복하기 위하여, W3C의 표준으로 채택된 XML 스키마를 기반으로 관계형 데이터베이스를 이용하여 XML DBMS를 설계 하였다.

Abstract

XML DTD is defining expression of simple form for XML document, there is much difficulties to define structure of XML document. Solving XML DTD's problem, W3C was select XML schema by standard in 2001. This paper proposes a design to XML DBMS using Relational Database based on XML schema to be selected by standard from W3C to overcome XML DBMS's shortcoming using XML DTD.

▶ Keyword : XML DTD, XML DBMS, XML Schema

-
- 제1저자 : 이상태
 - 접수일 : 2004.09.27, 심사완료일 : 2004.11.13
 - * 신성대학 컴퓨터정보계열 교수
 - ** 순천향대학교 대학원 전산학과 박사과정
 - *** 순천향대학교 정보기술공학부 교수

I. 서론

XML이 현재 전자상거래 등 기업간 정보유통에 주로 적용되고 있으나 웹서비스가 본격 구현되면 기업 기간시스템도 XML 기반으로 점차 전환될 것으로 보고 많은 업체들이 XML DBMS 개발에 경쟁적으로 나서고 있다. 그러나 기존의 XML DBMS 연구들에서는 XML 문서의 구조를 표현하기 위하여 XML DTD를 사용하였다. 이러한 XML DTD는 단순한 형태의 표현을 정의하고 있어서 XML 문서의 구조를 정의하는데 많은 어려움이 있다. XML DTD의 문제점을 해결하기 위하여 W3C에서는 2001년 XML 스키마를 표준안으로 채택하여 사용을 권고하고 있다. XML 스키마는 XML DTD에서 지원하지 못하였던 XML 문법의 준수, 네임스페이스의 지원, 다양한 형태의 빌트인 타입이나 사용자 지정 데이터 타입, 반복 횟수의 다양한 지원 등을 지원하도록 하였다[4].

본 논문에서는 기존 XML DTD를 이용한 XML DBMS의 단점을 극복하기 위하여, W3C의 표준으로 채택된 XML 스키마를 기반으로 데이터의 콘텐츠 저장에 일반적으로 사용되고 있는 관계형 데이터베이스를 이용하여 XML DBMS를 개발할 예정이다. 또한 XML 데이터를 저장하기 위하여 XML 스키마를 이용한 관계형 데이터베이스 스키마의 자동 변환 방법을 제안하고, 이를 토대로 설계한 EJB 컴퍼넌트를 이용하여 XML DBMS를 구현할 것이다.

II. 관련 연구

XML이 현재 전자상거래 등 기업간 정보유통에 주로 적용되고 있으나 웹서비스가 본격 구현되면, 기업 기간시스템도 XML 기반으로 점차 전환될 것으로 보고 많은 업체들이 XML DBMS 개발에 경쟁적으로 나서고 있다. 그러나 기존의 XML DBMS 연구들에서는 XML 문서의 구조를 표현하

기 위하여 XML DTD를 사용하였다. 이러한 XML DTD는 XML 문서에 대한 단순한 형태의 표현을 정의하고 있어서 XML 문서의 구조를 정의하는데 많은 어려움이 있다. XML DTD의 문제점을 해결하기 위하여 W3C에서는 2001년 XML 스키마를 표준안으로 채택하여 사용을 권고하고 있다. XML 스키마는 XML DTD에서 지원하지 못하였던 XML 문법의 준수, 네임스페이스의 지원, 다양한 형태의 빌트인 타입이나 사용자 지정 데이터 타입, 반복 횟수의 다양한 지원 등을 지원하도록 하였다[4].

XML 데이터를 저장할 수 있는 DBMS는 XML 데이터를 관계형 데이터베이스 형태로 변환해 저장하는 방법인 XML-enabled DBMS 방식과 XML 데이터를 XML 전용 DBMS에 저장하는 방법을 사용하는 순수 XML DBMS의 두가지 형태를 가지고 있다. 전자는 관계형 데이터베이스를 이용해야 하는 만큼 데이터 처리능력이 떨어질 수밖에 없고, XML 데이터의 100% 복원여부도 불투명하다. 또한 후자는 XML 데이터의 저장에 효과적이지만, 적용사례가 적고 DBMS를 새로 구축해야 한다는 부담이 있다. 그러나 현실적으로 많은 사용자가 관계형 데이터베이스를 사용하고 있기 때문에, XML-enabled DBMS 방식을 선호하고 있어 현재는 XML-enabled DBMS가 대세로 부각되고 있다.

2.1 국내외 동향

확장성표기언어(XML)가 기업내 및 기업간 데이터 교환을 위한 표준으로 부상하면서 XML전용 DBMS 수요가 가능성을 보여주었다. 여기에 XML 전용 DBMS는 기존 관계형 DBMS를 이용하는 것보다 웹 애플리케이션 속도를 증가시킨다는 점도 부각되었다. 이에 따라 테이택의 엑셀론, 펜타시스템 테크놀로지 등 외국계 제품과 인컴아이엔씨와 메타벌드가 각각 개발한 엑시마 서플라이, 비즈스토어 엑서스(Bizstore Exus) 등 XML 전용 DBMS가 선을 보였다. 또한 국외 기업인 오라클 사의 오라클 9i R2 등 XML지원 기능이 보강된 기존 DBMS들이 속속 시장에 등장하면서 XML 전용 DBMS에 대한 선호도 역시 기존 오라클, IBM 등의 인지도 틀을 벗어나지 못하고 있다[3].

국내의 동향은 신생 업체를 중심으로 XML DBMS의 개발에 노력하고 있는 추세이다. 주로 개발되는 XML DBMS는 Native XML DBMS로서 OODBMS(Object Oriented DBMS)를 위주로 사용되고 있으며, XML 문서의 논리적 모델을 정의하고, XML 문서를 저장소의 기능적인 부분처럼 사용하고 있다. 또한 물리적 저장 모델을 제공하고 있지 않다. 이와 같은 시스템을 사용하는 국외 DBMS

로는 Software AG사의 Tamino XML Server, eXcelon Corp.의 XIS 등이 있다. 또한 이러한 XML DBMS의 응용 분야로는 문서 중심의 XML 문서를 관리하는데 중점을 맞추고 있다[3][9].

2.2 EJB

J2EE(The Java 2 Platform, Enterprise Edition)는 다계층 엔터프라이즈 애플리케이션을 개발하기 위한 표준을 말한다. 즉 J2EE는 표준화되고 모듈화 된 컴포넌트상에 다 계층 엔터프라이즈 애플리케이션을 기본으로 하고, 해당 컴포넌트에 완벽한 서비스를 제공하며, 복잡한 프로그래밍 없이 자동으로 애플리케이션을 처리함으로써, 엔터프라이즈 애플리케이션을 단순하게 만든다. J2EE는 단순성, 이동성, 확장성, 통합성 등의 특징을 지원하는 엔터프라이즈 솔루션을 위한 플랫폼이다. 또한 J2EE는 JSP, Servlet, Java Bean, EJB등의 집합이라고 말하며, 핵심적인 기술로는 EJB가있다.

EJB는 컴포넌트 기반 분산 객체 기술로서 엔터프라이즈급 애플리케이션 개발에 있어 추상 데이터와 비즈니스 로직에 대한 부분을 담당하는 매우 중요한 핵심 기술을 가지고 있다. 또한 개발자를 도와주는 EJB 컨테이너는 자동으로 엔터프라이즈 빈의 생명주기 관리, 상태 정보관리, 보안, 트랜잭션 처리, 영속성 처리 등을 포함한 수많은 내재된 서비스를 제공해 주기 때문에 개발자는 비즈니스 로직만 담당하면 된다.

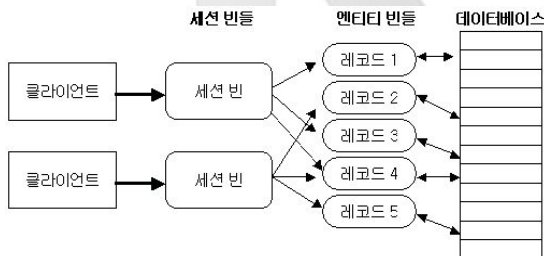


그림 1. 세션빈과 엔티빈의 관계

엔터프라이즈 빈(Enterprise Bean)이란 클라이언트가 호출하여 사용할 수 있는 EJB 컴포넌트를 의미한다. 이런 엔터프라이즈 빈은 세션 빈(Session Bean)과 엔티티 빈(Entity Bean)으로 구분되어 개발할 수 있다. (그림 1)은 세션 빈과 엔티티 빈의 관계를 보여준다[4].

III. XML DBMS 설계

XML DBMS의 부류는 기존의 상용 DBMS 업체에서는 XML-enabled DBMS라 부르는 관계형 데이터베이스 상에 XML을 처리할 수 있는 시스템을 구축하고 있으며, 신생업체나 OODBMS 업체에서는 기존의 관계형 데이터베이스를 사용하지 않고 XML 전용 DBMS를 개발하여 지원하게 하는 Native XML DBMS를 구축하고 있는 실정이다.

국내의 XML DBMS 업체는 신생업체가 대부분을 차지하고 있기 때문에, XML 전용 DBMS나 객체-지향 DBMS가 객체-관계형 DBMS를 이용하여 XML DBMS를 구축하고 있다[1]. 또한, 대부분의 업체에서 XML 구조 표현의 표준인 XML 스키마를 사용하지 않고, XML DTD를 사용함으로써 문서 표현의 능력을 감소시키고 있다[2]. 본 연구에서는 이러한 XML DTD의 단점을 극복하고자 XML 스키마로 표현된 XML 문서를 이용하여 저장 관리할 수 있는 XML DBMS를 개발하고자 하였다. 이를 위하여 (그림 2)와 같이 방법론 및 시스템 설계 단계를 적용하여 보다 효과적으로 XML DBMS를 개발할 것이다.

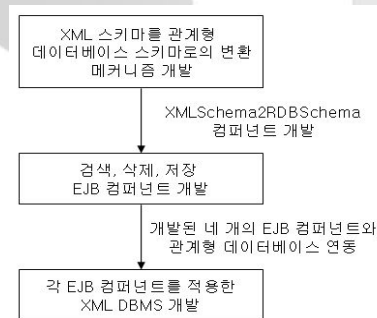


그림 2. XML DBMS 개발 단계

3.1 XML 스키마의 관계형 데이터베이스 스키마로의 매핑 방법론

XML 문서의 구조를 정의한 스키마를 관계형 데이터베이스 스키마로 변환하기 위하여 (그림 3)에서 보여주는 것과 같이 변환방법이 필요하다. 이는 XML 문서를 저장하기

나 검색할 때 변환해 주는 미들웨어를 통하여 진행된다 [7][8].



그림 3 객체 기반 변환방법

본 논문에서 사용하고 있는 변환방법은 객체를 이용한 변환방법으로서 복잡한 XML 문서를 관계형 데이터베이스에 저장하려고 할 때 사용하는 변환방법이며, XML DTD를 기반으로 한 변환방법에 비하여 다양하게 변환이 이루어지며, 스키마를 객체로 변환한 후, 그 변환된 객체를 관계형 데이터베이스 스키마로 변환함으로써 기존의 RDBMS의 활용영역을 확장하는데 기여한다.

3.2 XML 스키마를 관계형 데이터베이스 스키마로의 변환 메커니즘

XML 스키마를 관계형 스키마로 변환하는 과정은 (그림 3)에서 보였듯이 3단계를 이용하여 변환한다. 우선, XML 스키마를 객체 스키마로 변환한 후, 객체 스키마를 관계형 데이터베이스 스키마로 변환을 하고자 하였다.

3.2.1 XML 스키마를 객체 스키마로의 변환 메커니즘.

① 스키마 매핑

<표 1>은 자바의 패키지과 같은 관계된 프로그래밍 구조의 집합으로 스키마를 매핑한다.

② 어트리뷰트 매핑

어트리뷰트 노드(다중 값이 가능한)를 스칼라 타입으로 매핑하고, 끝지점의 어트리뷰트 노드를 properties로 매핑한다. <표 2>는 테이블이 어떻게 어트리뷰트 노드를 type으로 매핑하는지를 보여준다.

표 1. 스키마의 매핑

특성	객체 지향 구조로 매핑
모든 특성들	대응되는 구조는 없다. 스키마 컴포넌트의 특성들은 다른 컴포넌트의 리스트나 클래스의 리스트와 관련될 것이다. 이것은 확장 이외에는 다른 의미가 없다.

표 2. 어트리뷰트의 매핑

특성	객체지향 구조로 매핑
name, target namespace	타입 name, 다중 어트리뷰트 노드는 동일한 타입으로 매핑된다.
type definition	스칼라 타입의 집합을 각 노드로 매핑되게 결정한다.
scope(global)	scope는 모든 전역 타입을 포함한다.
scope (complex type)	클래스를 각각의 complex type으로 매핑된다. type은 local scope. scope는 모든 global type을 포함한다.
annotation (주석)	코드내의 코멘트

③ 엘리먼트 타입의 매핑

간단한 엘리먼트 타입 노드는 일반적으로 scalar data type으로 매핑되고, complex type 노드는 일반적으로 클래스로 매핑된다. 엘리먼트 타입 노드의 edges pointing은 property로 매핑된다. <표 3>은 어떻게 엘리먼트 타입 노드를 scalar type이나 클래스로 매핑하는 것을 보여준다.

④ 단일 타입의 매핑

단일 타입의 매핑은 <표 4>에서 어떻게 매핑되는지를 나타낸다. 기본 타입의 정의와 variety의 옵션에 의하여 결정하며, 일반적인 주석문장은 코드내에서 작성된다.

⑤ Complex Type의 매핑

complex type, complex type과 엘리먼트 type사이의 관계를 매핑하는 동안에 일어나는 문제를 고려하였으며, 이를 적용한 (표 6)은 어떻게 complex type 정의가 매핑되는지를 나타낸다. complex type은 스키마 그래프에서 엘리먼트 type node와 어트리뷰트 group과 particle node사이에 나타난다. 이런 경우, 엘리먼트 타입에서의 edge pointing은 엘리먼트 타입과 complex type사이의 관계에서 선택되어 complex type으로 매핑될 것이다.

complex type node는 class로 매핑될 것이며, node로부터 퍼져나가는 edge는 class의 property들로 매핑될 것이다.

3.2.2 객체 스키마에서 관계형 데이터베이스 스키마로의 변환 메커니즘.

객체 스키마에서 관계형 데이터베이스 스키마로의 변환 방법은 (표 5)와 같다. XML 스키마로부터 객체 스키마로 변환될 때는 객체 구조에 따라 변환되고, 객체 스키마에서 관계형 데이터베이스 스키마로 변환될 때는 내용에 따라서 매핑이 이루어진다.

표 3. 엘리먼트 타입의 매핑

특성	객체-지향 구조로의 매핑
name, target namespace	class 이름은 그들의 범위에서 유일해야 한다. 그러나 scalar type 이름은 유일해야 된다는 것이 필요하지 않다. 이것은, 다중 엘리먼트 타입 노드들이 동일한 scalar type으로 매핑될 수 있기 때문이다.
type definition (simple)	어떤 노드가 scalar type의 집합으로 매핑될 것인지를 결정한다.
type definition (complex)	클래스의 집합에 어떤 노드가 매핑될 것인지를 결정한다. 그리고 자식 노드를 종속되는 클래스내의 property로 매핑한다.
scope (global)	모든 전역 클래스를 포함하는 범위이다.
scope (complex type)	클래스는 각각의 complex type으로 매핑된다. 클래스는 local 범위를 가진다. 모든 전역 클래스를 포함하는 범위를 가진다.
annotation	코드 내의 코멘트이다.

표 4. 단일 타입으로의 매핑

특성	객체-지향 구조로의 매핑
name, target namespace	이름은 unique하지 않아도 된다. 이것은 다중 simple type이 같은 데이터 타입으로 매핑이 가능하다는 뜻이다.
기본 타입 정의	이것은 객체지향언어 데이터 타입으로 직접 매핑 될수 있는 타입으로 도달 할때 까지 타입 정의 계층으로 거꾸로 따라 간다.
facets	어떤 simple type으로 매핑될 것인지에 대한 데이터 타입을 결정해 준다. 이 매핑에서 사용되는 type은 property의 집합을 사용하는 mutator code로 매핑되어야 한다.
variety (atomic)	single-valued 타입이다.
variety (list)	multi-valued 타입이다.
variety (union)	union
annotation	코드의 코멘트이다.

표 5. 관계형 데이터베이스 스키마로의 매핑

객체-지향 구조	관계형 데이터베이스 구조
추상 클래스	추상 클래스는 inheritance 매핑을 제외하고 매핑되지 않는다.
class	class table로 알려져 있으며, object는 클래스 테이블의 행으로 나타난다.
상속성 (inheritance)	superclass 와 subclass는 unique key/foreign key로 join 되어 테이블로 나눈다. unique key는 superclass 테이블이며, object는 각 테이블의 행으로 나타난다. superclass property들은 subclass 테이블에 저장된다.
scalar 데이터 타입의 single-valued property	클래스 테이블의 열은 property column이며, 데이터 타입은 열의 가능한 데이터타입의 set으로 결정된다. (void 나 object 포인터의 데이터 타입은 BLOB으로 매핑된다. property는 property column의 값으로 나타나며, 분리된 테이블의 property column은 property table이다. property table은 unique key / foreign key의 관계로 class table에 조인된다. unique key는 class table에 있다.
scalar 데이터 타입의 Multi-valued(collection) property	클래스 테이블의 다중 property column은 collection의 특정 property 열에 매핑된다. 이것은 오직 collection이 제한된 최대 숫자의 값처럼 알려져 있을 때만 가능하다. property 테이블의 property 열은 collection의 값이 하나의 행만이 있음을 뜻한다. property 테이블은 unique key와 foreign key의 관계로 클래스 테이블로 조인된다. unique key는 클래스 테이블에 있다.
Property nullability	열 nullability.
Property finality	열 writeability는 최종 property가 read-only 열로 매핑 되고, non-final property는 read-write 열로 매핑된다.

표 6. Complex Type의 매핑 방법

특성	객체-지향 구조로의 매핑
name, target namespace	class name과 target namespace의 조합은 name으로 매핑될 것이다. complex type이 name을 가지지 않는다면, class로 매핑할 경우 이름을 주어야 한다.
base type definition (simple)	complex type이 매핑되는 class의 superclass는 명시된 simple type의 단일 property를 포함하는 클래스이다.
base type definition (complex)	class의 superclass는 complex type으로 매핑된다.
derivation method(extension)	상속
derivation method(restriction)	객체지향 analog에서는 직접적으로 나타나지 않는다. 이것은 역상속의 정렬이다.
final (extension)	Final class.
final (restriction)	객체지향 analog에서 직접적으로 나타나지 않는다.
abstract	어떤 경우에 클래스가 abstract인지 아닌지에 대한 경우는 매핑되지 않는다.
attribute use pairs	edge pointing에서 쌍을 사용하는 어트리뷰트는 complex type의 클래스내의 속성으로 매핑된다. 만약 불린 쌍이 true 이면 속성은 non-nullable이다. 만약 boolean이 false이면 property는 nullable이다.
content type (empty)	어떤 구성도 매핑되지 않는다. empty는 단지 맵에서 엘리먼트 타입 선언을 포함하여 정의하는 complex type에 대한 플래그이다.
content type (simple type definition)	non-nullable, scalar-값을 가진 속성은 클래스로 추가된다.
content type (content model, element-only)	content model의 범위는 element-only 값으로 매핑시 사용하지 않는다.
content type (content model, mixed)	content model의 범위는 추가로 다중값 속성이 Java의 String[] 처럼 클래스에 추가된다.
annotations	코드 내의 주석

3.3 XML DBMS와 EJB 컴퍼넌트 관계

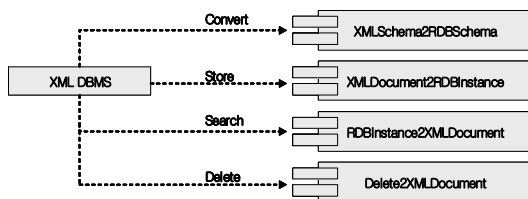


그림 4. XML DBMS의 컴퍼넌트 조직도

XML 저장관리 시스템을 구현하기 위해 사용하는 컴퍼넌트들은 (그림 4)와 같다.

변환을 담당하는 'XMLSchema2RDBSchema' EJB 컴퍼넌트, 저장을 담당하는 'XMLDocument 2RDBInstance' EJB 컴퍼넌트, 검색을 담당하는 'RDBInstance2XML Document' EJB 컴퍼넌트, 삭제를 담당하는 'Delete2

XMLDocument' 컴퍼넌트들을 조합하여 XML DBMS를 구현하며, 변환기능, 저장기능, 검색기능, 삭제기능을 갖는다.

3.4 XML DBMS의 기능

XML 저장관리 시스템 기능은 다음과 같다.

- 변환 기능 : XML 스키마를 관계형 데이터베이스 스키마로 변환하기 위하여, XML 스키마를 Object 스키마로 변환하고, Object 스키마를 토대로 RDB 스키마로 변환하는 자동 변환장치를 설계 및 구현할 것이다.
- 저장 기능 : XML 문서를 Map 형태에 맞게 관계형 데이터베이스에 저장하는 기능을 제공한다.
- 검색 기능 : 검색은 세 가지 방법이 가능하다. 첫째는 루트 테이블에 있는 데이터를 이용하여 전체 테이블들

검색한 후, Map 문서에 따라 XML 문서를 생성하는 방법, 둘째는 하나의 테이블에 있는 데이터를 모두 검색한 후, Map 문서에 따라 XML 문서를 생성하는 방법이 있다. 마지막으로 직접 사용자가 SELECT문을 이용하여 데이터를 검색한 후, Map 문서에 따라 XML 문서를 생성하는 방법을 제공한다.

- 삭제기능 : XML 문서가 저장되어 있는 XML DBMS에서 특정 XML 문서를 지우기 위하여 제공되는 기능으로써, SQL을 이용하여 문서를 삭제할 수 있도록 제공한다.

3.5 XML 저장관리 시스템 구조

XML 저장관리 시스템의 구조는 (그림 5)와 같이 변환 모듈, 저장모듈, 검색모듈, 삭제모듈로 나누어진다. 변환모듈은 XML 스키마를 입력받아 관계형 데이터베이스 스키마로 변환해 준다. 저장모듈은 XML 문서를 관계형 데이터베이스에 저장하는 부분이고, 검색모듈은 관계형 데이터베이스에서 키, 테이블, SQL을 이용하여 검색한 후, 검색한 데이터를 XML 문서화한다. 또한 삭제 모듈은 SQL을 이용하여 XML 문서나 스키마를 삭제하는 기능을 가지고 있다. XML 저장관리 시스템 구조는 XML DBMS에서 제공하는 기능들을 EJB 컴퍼넌트화 하고 각각의 EJB 컴퍼넌트를 조립하여 시스템을 구현하는 방식으로 설계하였으며, 구현될 XML DBMS는 기존의 XML DBMS에서 제공하지 못했던 XML 스키마 기반의 DBMS를 이용함으로써 e-business의 활성화, 기업간 또는 기업내 문서전달이 용이해질 것이다.

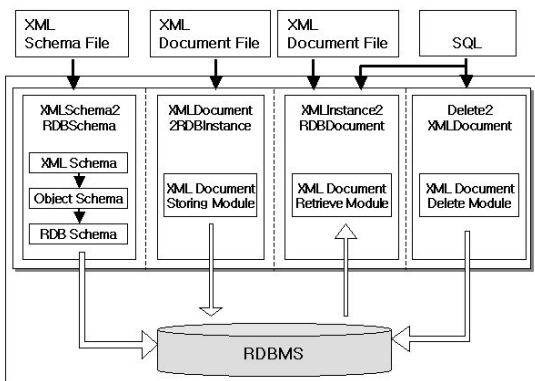


그림 5. XML DBMS의 전체 구조도

IV. 결론

XML 데이터를 저장할 수 있는 DBMS는 XML 데이터를 관계형 데이터베이스 형태로 변환해 저장하는 방법인 XML-enabled DBMS 방식과 XML 데이터를 XML 전용 DBMS에 저장하는 방법을 사용하는 순수 XML DBMS의 두가지 형태를 가지고 있다. 전자는 관계형 데이터베이스를 이용해야 하는 만큼 데이터 처리성능이 떨어질 수밖에 없고, XML 데이터의 100% 복원여부도 불투명하다. 또한 후자는 XML 데이터의 저장에 효과적이지만, 적용사례가 적고 DBMS를 새로 구축해야 한다는 부담이 있다. 그러나 현실적으로 많은 사용자가 관계형 데이터베이스를 사용하고 있기 때문에, XML-enabled DBMS 방식을 선호하고 있어 현재는 XML-enabled DBMS가 대세로 부각되고 있다.

본 연구에서는 기존 XML DTD를 이용한 XML DBMS의 단점을 극복하기 위하여, W3C의 표준으로 채택된 XML 스키마를 기반으로 데이터의 컨텐츠 저장에 일반적으로 사용되고 있는 관계형 데이터베이스를 이용하여 XML DBMS를 개발하기 위하여 XML 스키마를 이용한 관계형 데이터베이스 스키마의 변환 방법을 제안하고, 이를 토대로 설계한 EJB 컴퍼넌트를 이용하여 관계형 데이터베이스 스키마로의 자동 변환 컴퍼넌트를 개발하였다. 또한, XML 문서의 관계형 데이터베이스로의 저장을 위한 EJB 컴퍼넌트, 데이터베이스를 검색하여 XML 문서로 변환키 위한 EJB 컴퍼넌트, 저장된 XML 문서들을 삭제할 수 있는 EJB 컴퍼넌트 들을 개발하였다. 이를 통하여 기존의 XML DBMS에서 제공하지 못했던 XML 스키마 기반의 DBMS를 이용함으로써 e-business의 활성화, 기업간 또는 기업내 문서전달이 용이해질 것이다. 그리고 효율적이며 안정적으로 XML 컨텐츠를 저장, 관리할 수 있을 것이다.

참고문헌

- [1] 김경수, 최문영, 주경수, “객체지향 데이터베이스 기반의 XML응용을 위한 통합 설계방법론”, 한국컴퓨터정보학회지 Vol 7. No.1, 2002
- [2] 박준범, 박경우, 오수열, ODMG 객체기반의 XML문서 저장 관리 시스템에 관한 연구, 한국컴퓨터정보학회지 8권 2호, 2003
- [3] 오암석, “XML 전용 DBMS인 eXcelon의 소개”, 한국멀티미디어학회지, 6권, 4호, 2002
- [4] 용환승, 이월영, “객체-관계형 데이터베이스를 이용한 XML 문서 저장 기법”, 정보처리학회논문지D, 11권, 2호, 2004
- [5] 원재강, 김학성, “워크플로우를 위한 저장소 관리시스템”, 정보처리학회 춘계학술대회 논문집, 8권, 1호, 2001
- [6] 홍정선, 류재광, 김상배, “EJB 기반의 워크플로우 RuntimeDB Agent의 설계”,
- [7] Chengfei Liu, Hui Li, and Maria E Orlowska. Object-Oriented Design of Repository for Enterprise Workflows. CRC for Distributed System Technology and Computer Science Department, The University of Queensland, 1996. eai/concept/eai_term4.html ZDNet Korea, ZDNet Korea
- [8] Mark Graves, Designing XML Databases, Prentice Hall PTR, 2002
- [9] Workflow Management Coalition. Workflow Standards-Interoperability Wf-XML Binding. Document Number WFMC-TC-1023, pp.8-28, May 2000.

저자 소개



이상태

1987년 독일 DORTMUND대학교 전산학과 졸업(공학사)
 1994년 독일 DORTMUND대학교 대학원 전산학과 졸업(공학석사)
 2002년 순천향대학교 대학원 전산학과 졸업(공학박사)
 1996년 ~ 현재
 신성대학 컴퓨터응용계열 교수
 <관심분야> XML, 웹데이터베이스, 전자상거래, 인터넷 응용



임종선

1997년 청운대학교 전산학과 졸업(공학사)
 1999년 순천향대학교 대학원 전산학과 졸업(공학석사)
 2002년 ~ 현재
 순천향대학교 전산학과 박사과정
 <관심분야> XML DBMS, XQuery



주경수

1980년 고려대학교 이과대학 수학과 졸업(이학사)
 1982년 고려대학교 일반대학원 전산학과 졸업(이학석사)
 1986년 고려대학교 일반대학원 전산학과 졸업(이학박사)
 1998년
 University of North Carolina
 1999년 Visiting Professor
 1986년 ~ 현재
 순천향대학교 정보기술공학부 교수
 <관심분야> Database Systems, Semi-structured Data and XML, System Integration, Object-oriented Systems