저장 공간과 검색 효율을 위한 XML 문서의 RDB 스키마 모델

RDB Schema Model of XML Document for Storage Capacity and Searching Efficiency

권 훈, 김정희, 곽호영
제주대학교 통신컴퓨터공학부 컴퓨터공학과

Hoon Kwon(dreamerz@cheju.ac.kr), Jeong-Hee Kim(carina@cheju.ac.kr),
Ho-Young Kwak(kwak@cheju.ac.kr)

요약

정보 교환을 위한 XML 인스턴스는 일반적으로 legacy한 관계형 데이터베이스에 저장되어 있기 때문에 효율적인 XML 응용을 위한 데이터베이스와의 연동이 요구 되었으며, 이러한 요구를 지원하기 위하여 인스턴스와 스키마 구조를 분리하여 관계형 데이터베이스에 저장하는 가상 분할 저장 또는 분할 저장 방식의 인스턴스 저장 모델들이 연구되어지고 있다. 그러나 이러한 저장 방식은 인스턴스 구조와 인스턴스간의 계층 정보가 불일치하여 검색 시 질의 처리를 어렵게 하고 있고, 또한 분리 저장에 따른 중복 데이터들의 존재로 저장 시 오버헤드가 높아진다. 따라서 본 논문에서는 XML 문서를 저장할 때, 기존 연구의 데이터베이스 스키마 구조에 Eltype이라는 필드를 추가하여 인스턴스와 스키마의 상이한 계층 정보를 동일화하고, 저장 구조 각각의 필드를 관계형 데이터베이스 스키마 필드와 일치되는 저장 구조를 제안하였다. 그 결과 XML 인스턴스와 스키마 구조간의 저장이 가능하게 되어 중복 저장에 따른 오버헤드 및 저장 공간 감소, 그리고 동일화된 저장 계층 구조로 인해 검색 질의 처리가 쉬워졌다.

■ 중심어 : | XML | 스키마 | 저장 효율 |

Abstract

XML instances for purpose of information exchange are normally stored in the legacy relational database. Therefore, integrations with relational database are required for effective XML applications. To support these requirements, virtual decomposition storage or decomposition storage methods which save separates structures of instances to relational database have researched. However, these storage methods contain different information of schema structure and layers which has caused difficulties to process query during search operation as well as increased overheads due to duplicate savings for separate storages. Therefore, in this research, additional field of "Eltype" has introduced to previous database schema structure to instance and schema structure, provide consistent level information and propose storage structure to map each field to schema field of relational database. As results, XML instance and structures can be stored together to minimize overheads and required storage-space. Also, synchronized storage layer structure provides easier processing of search query.

■ keyword : | XML | Schema | Storage Capacity |

접수번호 : #060113-001
접수일자 : 2006년 01월 13일
심사완료일 : 2006년 03월 21일
교신저자 : 권 훈, e-mail : dreamerz@cheju.ac.kr

I. 서 론

인터넷을 통한 정보 습득의 양이 증가함에 따라 웹기반의 정보들을 보다 효과적으로 이용하고자 하는 연구가 활발히 이루어지고 있다[1]. 그러나 HTML (HyperText Markup Language)은 정보의 구조보다는 표현에 중점을 두고 있어서 특정 응용 분야의 정보 재활용에는 기능이 부족하다는 단점을 갖고 있다[1-3]. 이에 W3C(World Wide Web Consortium)에서는 HTML의 편리성과 SGML(Standard Generalized Markup Language)의 확장성 등의 장점을 취합하여 이기종간의 시스템에서 문서가 작성되더라도 상호교환과 다양한 문서형식들의 구조에 따른 일관성을 유지할 수 있는 차세대 웹 문서의 표준인 XML(*eXtensible Markup* language)을 1997년에 제안하였다[2-4]. 기존의 EDI로 문서가 만들어지면 문서 포맷에 대한 수정이 쉽지 않고 또한 기업들은 물품 거래 시 거래할 품목과 데이터를 서로 맞춰야 하는 불편이 있다. 하지만 XML은 확장 언어이기 때문에 태그를 바꾸거나 매핑하는 등의 다른 방법을 사용하여 일정하지 않은 문서 포맷으로도 데이터를 주고받을 수 있는 장점을 지니고 있다. 따라서 전사적 자원 관리(ERP, Enterprise Resource Planning)와 조달 시스템 그리고 온라인 주문 관리 시스템간의 데이터 교환, 그룹웨어 등에 XML 기반 애플리케이션 활용이 확대되고 있으며, 특히 최근에는 전자상거래 분야에서 XML을 활용한 문서 교환이 증가하고 있다[4][5]. 하지만 현재 EDI로 이루어진 문서 교환을 처리하기 위해서는 저장구조로써 eXoelon 및 tamino와 같은 XML 전용 데이터베이스 또는 legacy 데이터베이스를 이용하여 저장과 검색이 이루어지고 있다. 이러한 저장 구조로써 legacy 데이터베이스 측면을 살펴보면, XML 저장구조와 legacy 저장구조가 서로 상이하여 문서의 재활용에 많은 어려움이 존재하고 있어서, XML 문서를 legacy 데이터베이스 기반에서 저장, 검색할 수 있는 구조로 변경해주는 모델링 연구가 요구되고 있다[6][7].

따라서 본 논문은 XML 인스턴스와 데이터베이스 스키마 구조를 동일화하는 XML 기반의 계층적 RDB 스키마 모델을 제안하여 XML 저장에 따른 저장 공간과 검색 효율성을 보장할 수 있도록 하였다. 제안하는 스키마 구조는 인스턴스 및 인스턴스 구조의 효율적인 저장과 검색을 위하여 XML 스키마에 종속적인 구조를 가진다. 그리고 기존에 분리되어 저장되던 방식을 하나로 통합하기 위하여 기존 스키마 모델에 "SIDCHK" 필드를 추가하여, 스키마와 DTD, 인스턴스를 구분하고, "Eltype" 필드를 통해 스키마 정의 시 반복되는 엘리먼트의 형식인 ComplexType과 SimpleType를 축약하여 CLOB(Character Large OBject)의 형태로 저장이 가능토록 하였다. 또한 엘리먼트와 속성에 따른 구조 표현 및 인라인 기법을 통한 저장 방식을 사용하였다. 따라서 기존 연구들이 처리하지 못한 인라인 저장 방식[8]에 따른 레벨 정보를 동일하게 저장하고, 반복되는 엘리먼트를 "Eltype" 필드를 이용하여 처리함으로써, 저장 공간과 검색에서의 효율성을 가져오게 된다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구, 3장에서는 제안하는 스키마 모델, 4장에서는 구현 그리고 5장에서는 결론 및 향후 연구를 기술한다.

II. 관련 연구

1. XML 저장을 위한 데이터베이스

XML 문서를 저장, 관리, 검색하려면 XML 특성에 맞는 문서 저장/관리 시스템이 필요하다. XML 데이터를 위한 데이터베이스 구현 기술은 크게 4가지로 나눌 수 있다[9].

1.1 파일 관리형 데이터베이스

파일 관리형 데이터베이스는 각 XML 데이터를 파일 시스템 내부의 파일로 보관하며, XML 파일을 디렉토리로 관리한다. 이는 구조정보를 가지지 않은 정형식 XML 문서관리에 적합하며, 인덱스를 통한 빠른 검색이 가능하다. 그러나 XML 문서 추출시마다 XML 파싱(Parsing)이 필요하다는 단점이 있다.

1.2 관계형 데이터베이스

관계형 데이터베이스(RDB)는 XML 문서를 매핑해서

저장 관리하는 방식이다. 기존의 관계 데이터베이스를 재사용할 수 있는 장점이 있고, 조작이 간편하고 SQL문 등을 사용할 수 있어 가장 일반적으로 구현하는 방식이다.

XML의 계층구조를 관계형 데이터베이스의 관계(relation)로 표현하며, XML 각 요소(element) 값과 속성(attribute)을 테이블 필드로 표현하기 때문에, 구조를 정의하고 있는 문서를 적용하는데 좋다.

1.3 객체지향형 데이터베이스

객체지향형(Object Oriented) 데이터베이스는 XML 데이터의 계층 구조를 객체지향의 클래스 계층에 매핑하여 구현하는 방식이다. 객체지향 개념을 이용할 수 있기 때문에 상속과 같은 객체지향 특성을 이용할 수 있으며, 엘리먼트 간의 전후종속 관계를 클래스에 기반한 객체들간의 링크로 나타낼 수 있다. 관계 데이터베이스보다 데이터 모델이 XML과 유사하여 문서 정의 타입으로부터 객체 스키마를 생성하는 문제와 XML 질의를 OQL(Object Query Language)로 변환하는 과정이 쉽고 가변 필드의 사용에 제한이 없어서 대용량 데이터 형식(built-in data type)으로 지원되기 때문에 XML의 순서정보 표현이 간단하다.

1.4 네이티브 XML 데이터베이스

네이티브(native) XML 데이터베이스는 XML의 계층 구조를 계층형 데이터베이스 구조로 관리하며 항목의 데이터나 속성을 어디까지나 XML 데이터로 관리하므로 문서 관리에 적합하다고 볼 수 있다. 또한, 구조를 정의하고 있는 XML 데이터에 독자적인 인덱스를 부여, 고속의 트랜잭션 처리를 지원한다. 이러한 네이티브 XML 데이터베이스는 데이터 모델 변환이나 데이터 저장에서 부담이 적지만, 데이터 검색을 위하여 여러 가지 인덱스를 참조해야 하므로 대규모 XML 데이터 처리에서 성능이 낮아진다는 단점이 있다.

XML 저장을 위한 데이터베이스를 [표 1]과 같이 비교 분석하였다. 이를 통해, 본 논문에서는 현재 가장 사용 빈도가 높은 기존의 관계형 데이터베이스를 활용하여 XML을 효율적으로 저장하고자 한다.

표 1. XML 저장 데이터베이스의 비교

	파일관리	관계	객체지향	네이티브
저장방식	File	테이블 매핑	클래스 매핑	독자적 인덱스
사용빈도	저	고	저	중
장점	정형식 문서 관리 적합	데이터 재이용 가능 조작 간편	데이터모델 다양	쉬운 인터페이스 고속 트랜잭션 처리
단점	문서 추출시 파싱 필요	조인 연산 필요	사용빈도와 정보 재이용율 낮음	검색시 인덱스의 빈번한 참조

2. XML 저장 모델

XML 저장관리 시스템을 개발하기 위해 선행되어야 하는 것이 데이터 모델링이다. 데이터 모델링이란 데이터와 데이터들 간의 관계를 기술하는 개념적 도구로서, 데이터베이스의 논리적 구조를 명시하는 과정이다.

이러한 데이터 모델링은 DBMS 활용과 문서 정의 형식에 따라 저장하는 방법들이 연구되어지고 있다. 이러한 저장 방식 중 일반적으로 계층적 XML 문서 구조정보를 관계형 데이터베이스에 저장하는 모델로써, 분할 저장 모델과 비분할 저장 모델을 들 수 있다[10].

2.1 분할 저장 모델

XML 문서에 대해서, 엘리먼트 단위로 나누어서 저장하는 방식으로 문서의 실제 내용을 가지고 있는 각각의 단말 엘리먼트 안에 문서의 내용이 나뉘어 저장되고, 검색 시 구조 정보를 참조하여 해당 엘리먼트나 하위 엘리먼트의 조합을 생성하여 처리하는 방식이다. 따라서 문서의 구조적 정보나 일부 내용들이 수정되었을 때 관계있는 엘리먼트만 수정하면 되므로 문서의 버전 관리가 쉽고, 동일한 내용을 갖는 노드들을 공유할 수 있다는 장점이 있다. 그러나 문서를 검색하여 내용을 추출해야 하는 경우에는 각 엘리먼트의 내용을 조합하여 결과를 구성해야 하므로 검색 과정이 복잡하고 검색 시간이 오래 걸린다는 단점과 문서 타입 정의에 의존적인 데이터 모델을 사용하므로 DTD나 스키마 구조가 갱신되어지면, 이와 관련된 모든 데이터들이 재구성 되어야 한다는 문제점이 발생한다.

2.2 비분할 저장 모델

XML 문서 전체를 저장하는 방식이다. XML 저장 관리 시스템에서는 전체문서뿐만 아니라, 엘리먼트 단위의 검색까지 지원해야 한다. 이때 엘리먼트들은 문서 안에서 계층상의 구조 정보를 가지고 있고 이로 인해서 모든 레벨에서 수평적 관계 및 수직적인 관계의 표현이 가능하다. 각 엘리먼트에 대한 구조 정보 및 위치 정보를 유지하기 위해서 별도의 테이블을 만들고, 엘리먼트가 문서 안에서 나타난 시작 오프셋과 끝 오프셋에 관한 정보를 엘리먼트 구분자와 함께 기록한다. 트리상의 각각의 노드는 실제 문서의 논리적인 구조만을 기술하고 내용부분을 다른 영역에서 관리하여 각각의 노드가 이 영역에 대한 위치정보와 길이 등을 가지고 내용을 표시하는 비분할 방법이다. 문서 전체는 한꺼번에 저장하므로, 전체 문서에 대해서는 빠른 검색이 가능하며, 각 엘리먼트에 대한 정보를 유지하므로 엘리먼트 단위의 검색도 지원한다.

분할 저장 모델과 비분할 저장 모델을 [표 2]에 비교하여 나타내었다.

표 2. 분할 저장 모델과 비분할 저장 모델의 비교

	분할 저장 모델	비분할 저장 모델
저장 방식	문서를 나누어 저장	전문(Full-Text) 전체 저장
검색 방법	해당 노드를 하나씩 직접 처리	오프셋, 길이 이용 문서 검색
테이블 구성	하나의 독립 테이블 구성	엘리먼트와 정보테이블로 구성
검색 시 취합여부	모든 리프 노드를 순회/취합	필요 없음

3. 관계형 데이터베이스를 위한 XML 저장 방법

계층적 구조를 가진 XML 문서를 평평한 구조의 관계형 데이터베이스에 저장하는 작업은 쉽지 않다. 무엇보다도 관계형 테이블로는 계층 구조를 표현하기 어렵기 때문이다.

대표적으로 기존의 준 구조적(semi-structured) 데이터를 저장하기 위해 많이 사용되던 간선저장 방식[11]과 일정한 속성을 공유하는 엘리먼트들을 하나의 테이블에 "인라인" 시키는 방식[8]으로 연구되어지고 있다.

3.1 간선(Edge) 저장 방식

XML 문서를 RDBMS를 사용하여 저장하고자 할 때, 가장 단순한 방식이라 할 수 있는 간선 저장 방식은 기본적으로 XML 문서를 표현한 그래프 상에서의 모든 간선들에 대한 정보를 간선(Edge)이라 불리는 하나의 테이블에 저장한다. 이러한 간선 테이블은 그래프 상에서의 각 간선의 소스(source)와 타겟(target)노드의 노드 식별자 값을 저장하기 위한 source 애트리뷰트와 target 애트리뷰트, 간선의 이름을 저장하기 위한 name 애트리뷰트, 간선이 가리키는 노드가 내부 노드(internal node) 인지 리프 노드인지를 표현하기 위한 flag 애트리뷰트, 간선의 순서정보를 저장하기 위한 ordinal 애트리뷰트로 구성된다.

3.2 인라인 저장 방식

인라인 저장 방식[8]은 일정한 속성을 공유하는 엘리먼트들을 하나의 테이블에 "인라인" 시키는 방법으로, DTD를 관계형 스키마로 표현하기 위하여 세 가지 관계형-변형 알고리즘인 Basic, Shared, Hybrid기법을 이용하고 있다. Basic 기법은 엘리먼트 그래프를 깊이 우선 순회하면서 방문하여 엘리먼트들을 하나의 테이블에 인라인시키는 방법을 사용하였다. Shared 기법은 DTD 그래프를 순회하면서 루트 노드에 해당하는 엘리먼트에서만 테이블을 생성한다. 자식 엘리먼트 노드들을 방문하면서 엘리먼트들을 인라인 시키는 것은 Basic과 같지만, 추가적으로 다른 테이블과 공유되어질 수 있는 엘리먼트들을 만나게 되면 독립적인 테이블로 처리하는 기법이다. Hybrid 기법은 앞의 두 기법을 혼합한 것으로, 가장 좋은 성능 평가를 나타낸다. 이 기법은 DTD를 이용하여 가능한 적은 수의 테이블로 XML 데이터를 사상시킨다.

III. RDB 스키마 모델과 저장 및 검색 시스템

1. RDB 스키마 모델

기존 데이터베이스 스키마[12]를 사용하여 XML을

기존 RDB에 저장하게 되면, 스키마와 인스턴스에 따르는 두 개의 정적 테이블이 만들어지고 모든 XML 구조를 인라인 방식으로 해석하여, 엘리먼트 단위로 저장함으로써, 계층정보를 표현하는 ref, ref_level, ref_step간의 정보가 스키마와 인스턴스 간에 서로 불일치하게 된다. 이로 인한 검색에 대한 질의가 복잡하게 된다. 기존 데이터베이스 스키마 모델은 [그림 1]과 같고, 각 필드별 세부 내용은 [표 3]에 나타내었다.

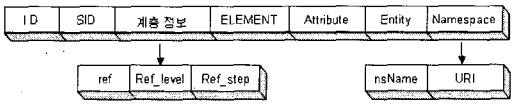

그림 1. 기존 스키마 모델

표 3. 기존 데이터베이스 스키마 필드의 세부 내용

필드명	의 미
ID	엘리먼트를 구분하기 위한 식별자
SID	XML 스키마와 인스턴스의 적용 구분자
ref	상위 엘리먼트
ref_level	엘리먼트 등급
ref_step	엘리먼트의 순서
element	엘리먼트 이름
attribute	속성 이름과 값
entity	엘리먼트의 값
Namespace	네임스페이스를 처리하기 위해 접두사 nsname과 경로 URI 로 표현

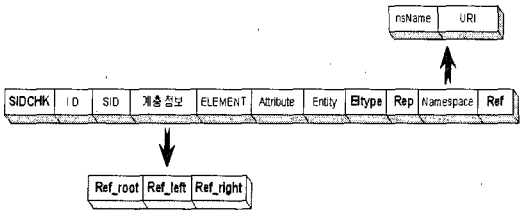

그림 2. 제안 스키마 모델

따라서 본 논문에서 제안하는 데이터베이스 스키마에서는 [그림 2]에 나타낸 것처럼 SIDCHK, Eltype, Rep, Ref 의 4개의 필드를 새로 추가하고, 계층정보를 나타내는 기존의 방법을 보안하여, XML 트리 구조의 효율적인 탐색을 위한 Ref_root, Ref_left, Ref_right로 바꾸었고, 그 필드의 세부 내용은 [표 4]와 같다.

표 4. 제안 데이터베이스 스키마 필드의 세부 내용

필드명		의 미
SIDCHK		XML 스키마와 인스턴스의 식별자 (스키마 : S , DTD : D , 인스턴스 : I)
Level Info	ref_root	자신의 상위 엘리먼트의 위치
	ref_left	자신의 왼쪽 형제 엘리먼트의 위치
	ref_right	자신의 오른쪽 형제 엘리먼트의 위치
Eltype		스키마에서의 element 타입을 정의 Complex Sequence : CS, Complex All : CA Complex Choise : CC Group Sequence : GS, SimpleType : S
Rep		엘리먼트의 재사용 횟수를 기억
Ref		Eltype의 재사용을 위한 참조 값을 기억

[표 4]의 필드들 중 LevelInfo 필드는 3부분으로 나뉘어, 실제 Element에 대한 위치정보를 나타내게 된다. 이 정보는 자신의 상위를 나타내는 Root, 왼쪽 형제를 나타내는 Left, 오른쪽 형제를 나타내는 Right로 세분화 되어 있다. Eltype 필드는 Element 정의 타입형식으로 스키마에서 주어지는 형식을 CLOB(Charactor Large OBject)기법을 이용하여 축약시켜 가상 분할 저장하였다. Rep 필드는 엘리먼트의 반복 횟수를 지정할 수 있도록 하였으며, 마지막으로 Ref 필드는 문서내의 Element 정의형식에 따른 참조를 나타내는 값으로 구성된다.

제안하는 RDB 스키마 모델을 이용하면, 스키마와 인스턴스간의 계층 정보가 동일하게 되어, 검색시 질의 복잡성을 줄이게 되고, XML 스키마에서 주어지는 형식을 축약시켜 가상 분할 저장함으로써, 저장 공간의 감소를 나타낸다.

2. 저장 및 검색 시스템

저장 및 검색 시스템은 문서의 입력을 담당하는 부분과 Mapping Layer 그리고, 사용자 인터페이스 부분으로 처리 된다. XML 문서가 들어오면 Mapping Layer에서 XML 문서가 정형화 되었는가를 분석을 한 후, 문서를 순차적으로 읽어 내려가면서 요소들의 연관 관계를 추출한다. 분류되어진 요소들은 매핑 테이블로 옮겨지고 데이터베이스 스키마를 생성한다. 생성된 스키마는 인덱스 관리자로 이동하고 각각의 엘리먼트, 속성, 콘텐츠에 인덱스를 부여한다. 요소들과 인덱스들은 데

이터베이스에 저장이 된다. 구조 정보를 검색하기 위해서는 사용자 인터페이스 통해 질의를 하고, 질의 처리기는 사용자가 질의한 키워드를 기반으로 구조적인 정보를 수집하여 원하는 구조 정보를 검색하여 사용자에게 다시 전달을 한다. [그림 3]은 저장 및 검색 시스템 구조를 나타낸다.

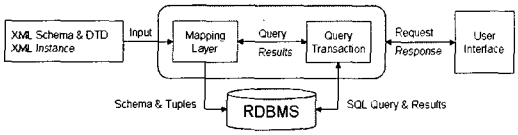

그림 3. 저장 및 검색 시스템 구조

저장 및 검색 시스템의 환경은 [표 5]와 같다.

표 5. 저장 및 검색 시스템 환경

항 목	내 용
운 영 체 제	Windows XP, Pentium 3 800MHz
데이터베이스	MySQL 3.23.58
개발 Tool	VisualBasic 6.0 Sp2
기타 Tool	MySQL-Front, MSXML Parser 4.0 SP2

IV. 구현 및 실험 결과

1. 사용자 인터페이스

사용자 인터페이스는 [그림 4]와 같이 크게, XML Document Input Part, RDB Contents View Part, RDB Contents Search Part로 구분된다.

XML Document Input Part는 스키마와 DTD 그리고 XML Document를 확장자를 통해 구분하여 입력을 받는다. Mapping Layer의 분석기는 입력 받은 파일을 분석하여 그 결과를 RDB에 저장을 하게 된다. 만약, 문서가 저장된 문서형식에 유효하지 않은 문서가 입력되었을 때에는 RDB에 저장을 하지 않고 걸러내는 역할을 수행한다.

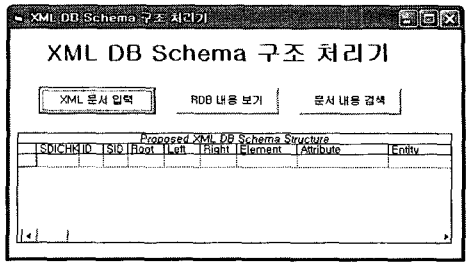

그림 4. 사용자 인터페이스

RDB Contents View Part는 ODBC를 통하여 데이터베이스 스키마의 내용을 출력하는 기능을 수행한다. [그림 5]는 저장 및 검색 시스템의 처리 결과를 RDB View Part가 받아서 보여주고 있다.

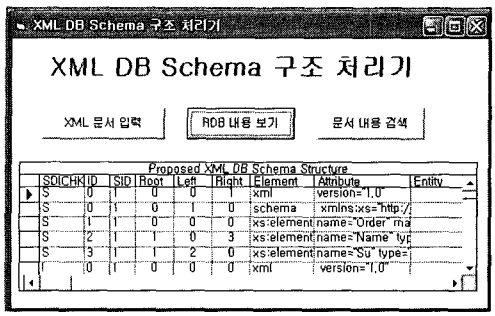

그림 5. 제안 모델에 의해 RDB에 저장된 문서 정보

RDB Contents Search Part는 RDB에 저장된 데이터를 검색하는 기능을 담당한다. 데이터를 검색하는 방법은 콘텐츠 값에 의한 데이터 검색, 계층 정보에 의한 데이터 검색, ID 값에 의한 데이터 검색 등 다음 3가지로 나뉜다.

- 콘텐츠 값에 의한 데이터 검색

데이터베이스 안에 저장된 ENTITY 필드를 기준으로 실제 XML 문서의 콘텐츠 값을 직접 검색하는 방법이다. 이는 단어위주의 데이터 검색 및 "*"를 사용하여 모르는 값에 대한 검색범위를 지원하고 있다.

◦ 계층 정보에 의한 데이터 검색

계층 정보에 의한 데이터 검색은 XML 문서에 따른 구조를 이용한 검색 방법이다. 이는 데이터간의 구조를 이용하여 종속적 관계 및 수평적 관계를 계층 구조를 통해 해당 데이터를 유추할 수 있는 장점이 있다.

◦ ID 값에 의한 데이터 검색

해당 데이터의 엘리먼트 ID에 의한 검색방법이다. 이는 엘리먼트 단위의 순차적인 ID 값을 이용하여, 해당 데이터의 엘리먼트 위치를 검색할 수 있다.

2. 저장 및 검색 시스템

3장에서 기술된 바와 같이 저장 및 검색 시스템은 문서의 입력을 담당하는 부분과 Mapping Layer 그리고, 사용자 인터페이스 부분으로 처리 된다. Mapping Layer에서는 제안된 데이터베이스 스키마에 따라 입력받은 XML 문서에 대해 [그림 6]과 같은 처리 순서도에 따른 처리과정이 이루어진다.

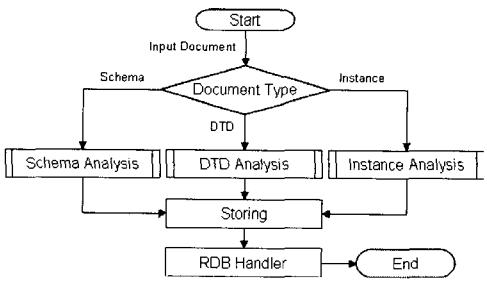

그림 6. Mapping Layer의 처리 순서도

처리과정은 입력받은 XML 문서를 스키마, DTD , 인스턴스로 구분하게 되며, 각각의 해당되는 경우의 분석(Analysis)과정을 거치게 된다. 문서 구조에 대한 분석 과정이 끝나면, 제안된 데이터베이스 스키마에 매핑 테이블을 구성한다. 그 후, RDB Handler를 통해 기존 RDB에 저장하는 단계로 이루어진다.

입력받은 XML 문서에 따른 분석과정은 다음과 같이 세분화된다.

◦ Schema Analysis

스키마의 입력이 들어오게 되면, 인라인 방식으로 각각의 엘리먼트를 읽어온다. 스키마는 엘리먼트명을 정의한 다음에는 반드시 엘리먼트 문서타입을 정의하게 되어있다. 따라서 엘리먼트 문서타입부분을 먼저 읽어 계층정보에 바로 저장하지 않고, "Eltype" 필드에 CLOB의 형식으로 약속된 문자열로 축약하여 저장하는 과정이 필요하며, 이에 따른 계층정보 역시 수정되어져야 한다. 또한, 엘리먼트 선언에 따른 제약사항을 분류하여, 각각의 정의된 필드에 분류하는 작업이 수행되어진다.

◦ DTD Analysis

DTD의 입력이 들어오면, Schema Analysis와 같이 인라인 방식으로 엘리먼트 정의를 읽어오게 되며, 이때, 엘리먼트의 포함관계에 따라 계층정보를 나뉘어 저장하게 된다. 이 과정에서는 Schema Analysis와 달리 DTD는 문서타입정의에 따른 저장방법은 필요하지 않다.

◦ Instance Analysis

인스턴스의 입력이 들어오면, 인라인 방식으로 엘리먼트를 순서대로 읽어오며, 읽어온 구조를 통해, 계층정보를 부여하고, 이때, 모든 인스턴스의 정보를 읽게 되면, 기존에 저장되어진 RDB 상의 스키마구조와 비교하여 해당 스키마를 찾아 검증(Validation)을 수행하게 된다. 이때, 검증된 인스턴스 문서에 대해서 데이터베이스 스키마에 저장되게 된다.

3. 실험 결과

성능 평가를 위한 검증데이터는 두 그룹으로 나뉘어 각각 10회 실시하였으며, 이에 따른 저장 공간 및 저장에 따른 시간, 그리고 검색에 따른 시간을 평가하였다.

이력서는 구조상 XML 문서로 표현하였을 때, 동일한 엘리먼트 구조가 반복되고, 계층구조가 명확하게 구분되며, XML 스키마에 따른 인스턴스 형식이 동일하다는 특징을 지니고 있다.

따라서 본 논문에서는 제안 시스템 검증을 위한 데이터로 동일 엘리먼트 구조가 반복적으로 존재하고, 계층

구조가 명확한 학생들에 대한 이력서를 사용하였다.

학생들에 대한 이력데이터는 1명의 이력만 가지고 있는 데이터(A)와 40명의 이력을 가지고 있는 데이터(B)로 나뉘어 구성하였다.

3.1 RDB 저장 공간 및 저장에 따른 시간 측정

데이터(A)와 데이터(B)에 따른 RDB 저장 공간 및 저장에 따른 시간은 기존 시스템과 제안시스템에서 각각 [표 5]와 [표 6]과 같은 결과를 얻을 수 있었다.

표 5. 기존 시스템의 저장 결과

	데이터 (A)		데이터 (B)	
	스키마	인스턴스	스키마	인스턴스
저장공간	21 Rec	20 Rec	24 Rec	800 Rec
저장시간	0.923 Sec	0.922 Sec	0.923 Sec	1.682 Sec

표 6. 제안 시스템의 저장 결과

	데이터 (A)		데이터 (B)	
	스키마	인스턴스	스키마	인스턴스
저장공간	14 Rec	20 Rec	15 Rec	781 Rec
저장시간	0.906 Sec	0.922 Sec	0.907 Sec	1.641 Sec

3.2 검색에 따른 시간 측정

검색에 따른 시간 측정은 데이터(A)와 데이터(B)에 각각 [표 7]의 검색 유형과 검색 조건을 가지고 측정하였다.

표 7. 검색 유형과 조건

검 색 유 형	검 색 조 건
콘텐츠 값에 의한 검색	1) 김* or 홍* 2) 홍길동
계층 정보에 의한 검색	1) 2/0/4
ID에 의한 검색	1) 9* 2) 9

검색 조건에 따른 측정을 10회 반복하여 나온 결과수치에 대한 평균을 기존 시스템과 제안 시스템에 따라 [표 8]과 [표 9]와 같은 결과를 얻을 수 있었다.

검색에 의한 산출 시간은 데이터 콘텐츠에 의한 검색시간보다 ID 검색에 따른 검색 시간이 상대적으로 빠른 결과를 나타내었으며, 이러한 결과는 이미 정렬되어진 ID 값을 이용한 탐색시간의 차이에 따른 것이라 할 수 있다.

기존 시스템보다 모든 검색유형에서 조금 향상된 결과를 나타내었다. 이러한 결과는 인스턴스 저장 공간의 감소로 인한 레코드 대비 탐색시간의 축소, 그리고 여러 테이블에 걸친 조인 연산이 아닌 하나의 독립 테이블에서의 질의 처리로 인한 조인 연산의 감소에 의한 것이라 할 수 있다.

표 8. 기존 시스템의 검색 결과

검색 유형	데이터 (A)	데이터 (B)
콘텐츠 검색	0.104 Sec	0.127 Sec
	0.098 Sec	0.119 Sec
계층정보 검색	-	-
ID 검색	0.084 Sec	0.085 Sec
	0.092 Sec	0.098 Sec

표 9. 제안 시스템의 검색 결과

검색 유형	데이터 (A)	데이터 (B)
콘텐츠 검색	0.102 Sec	0.116 Sec
	0.095 Sec	0.109 Sec
계층정보 검색	0.086 Sec	0.088 Sec
ID 검색	0.084 Sec	0.084 Sec
	0.091 Sec	0.096 Sec

V. 결 론

본 논문에서는 저장 공간과 검색 효율을 고려한 XML 문서의 RDB 스키마 모델을 제안하였다. 그럼으로써 기존의 스키마 모델에 의한 상이한 계층정보를 동일화 하였고, 기존의 분리 저장방식을 통합할 수 있도록 RDB 스키마 구조를 개선시켰으며, XML 정의형식인 DTD의 제한사항을 극복한 XML 스키마가 처리되도록 하였다. 그 결과 해당 엘리먼트 정보의 위치를 계층적으로 표현할 수 있게 되었으며, 기존 관계형 데이터베이스 스키마 구조의 개선을 통해 XML 스키마와 인스턴스간 동일한 계층정보를 유지하게 되었고, 저장 공간 활용 및 비율에 따른 효율성이 높아졌다. 또한 상이한 계층구조에 따른 손상을 최소한으로 유지하게 되어 별도의 연산시간 없이 문서간의 검색질의가 수월하게 되었다.

향후 연구 과제로는 이미 저장되어 있는 관계형 데이

터베이스 내의 데이터 갱신에 따른 처리에 대한 연구가 필요하다.

참 고 문 헌

[1] http://w3c.org/TR/2004/REC-xml-20040204

[2] 연제원, 조정수, 이강찬, 이규철, "XML 문서 구조검색을 위한 저장 시스템 설계", 한국정보과학회 학술발표논문집(B), 제26권, 제1호, pp.3-5, 1999.

[3] D. W. Shin, "BUS:An Effective Indexing and Retrieval Schema in Structured Documents," in Proc. Digitial Libraries, 1998.

[4] B. Lowe, J. Zobel, and R. Sacks-Davis, "A Formal model for Databases of Structured Text," Proceedings of the Fourth International Conference on Database Systems for Advanced Applications(DASFAA '95), pp.449-456, 1995.

[5] 박종관 외, "XML 문서의 효율적인 구조 검색을 위한 색인 모델", 한국정보과학회 논문지, 제8-D권, pp.451-460, 2001.

[6] D. Florescu and D. Kossmann, "Storing and Querying XML Data using an RDBMS," IEEE Data Engineering Bulletin, Vol.22, No.3, pp. 27-34, 1999.

[7] E. Damiani, S. Vimercati, S. Paraboachk, and P. Samarati, "XML Access Control System: A Component-Based Approach," In Proc. IFIP WG11.3 Working Conference on Database security, The Netherlands, Aug., 2000.

[8] J. Shanmugasundaram, H. Gang, K. Tufte, C. Zhang, D. DeWitt, and J. Naughton, "Relational databases for Querying XML Documents : Limitations and Opportunities," 25th VLDB Conference, Edinburgh, Scotland, 1999.

[9] 민준기 외 3인, "다양한 저장소에서의 효율적인 XML 저장 기법에 대한 연구", 정보과학회 데이터베이스 연구회지, 제19권, 제1호, pp.1-14, 2003.

[10] 김 훈 외 2명, "XML 문서 저장 시스템", 정보과학회 데이터베이스 연구회지, 제16권, 제2호, pp.29-35, 2000.

[11] S. Abitebousi, P. Bunneman, and D. Suciu, Data on the Web : from Relations to Semistructured Data and XML, Morgan Kaufmannm, 1999.

[12] 홍석건 외 2명, "검색과 저장의 효율성을 위한 정적 테이블 기반의 XML 저장구조 설계", 한국정보과학회 학술발표논문집(B), 제30권, 제2호, pp.205-207, 2003.

저 자 소 개

권 훈(Hoon Kwon) 준회원

- 2003년 8월 : 제주대학교 해양생물공학과(이학사)
- 2005년 8월 : 제주대학교 대학원 컴퓨터공학과(공학석사)
- 2006년 3월~현재 : 제주대학교 대학원 컴퓨터공학과(박사과정)
- 2004년 9월~현재 : 제주한라대학교 시간강사

<관심분야> : XML, 유비쿼터스, 센서 네트워크

김 정 희(Jeong-Hee Kim) 정회원

- 1994년 2월 : 제주대학교 정보공학과(공학사)
- 1997년 2월 : 제주대학교 대학원 정보공학과(공학석사)
- 2005년 2월 : 제주대학교 대학원 정보공학과(공학박사)
- 2002년 2월~현재 : 제주대학교 시간강사

<관심분야> : XML, 시멘틱 웹, 센서 네트워크

곽 호 영(Ho-Young Kwak) 정회원

- 1983년 2월 : 홍익대학교 전자계산학과(이학사)
- 1985년 2월 : 홍익대학교 대학원 전자계산학과(이학석사)
- 1991년 2월 : 홍익대학교 대학원 전자계산학과(이학박사)
- 1990년 3월~현재 : 제주대학교 통신컴퓨터공학부 교수

<관심분야> : 객체지향 프로그래밍, Web 응용