Edge-Labeled Graph에 기반 한 XML 인스턴스의 RDB 저장 모델

김정희[O] 김정필* 곽호영

제주대학교 통신컴퓨터공학부
중앙대학교 전기전자공학부*
{carina[O], kwak}@cheju.cheju.ac.kr
phill@cau.ac.kr

RDB Storage Model of XML Instance based on the Edge-Lageled Graph

Jeong-Hee Kim[O] Ho-Young Kwak
Dept. of Communication & Computer Engineering, Cheju National University
School of Electrical and Electronic Engineering, Chung-Aug University

요 약

본 논문에서는 Edge-Labeled Graph에 기반하여 XML 인스턴스들을 관계형 데이터베이스(RDB)로 저장하는 모델을 제안하고 구현한다. 저장되는 XML 인스턴스들은 Edge-Labeled Graph에 기반 한 Data Graph로 표현되고 이를 이용하여 데이터 경로(Data Path), 요소(Element), 속성(Attribute), 테이블 인덱스(Table Index) 테이블에 정의된 값들이 추출된 후 Mapper를 이용하여 데이터베이스 스키마를 정의하고 추출된 값들을 저장한다. 그리고, RDB 저장 모델은 질의를 지원하기 위해, XPATH를 따르는 질의 언어로 사용되는 XQL을 SQL로 변환하는 변환기를 제공하며, 또한 저장된 XML 인스턴스를 복원하는 DBtoXML 처리기를 갖도록 하였다. 구현 결과, XML 인스턴스들과 RDB 구조로의 저장 관계가 그래프(Graph) 기반의 경로(Path)를 이용한 표현으로 가능했으며, 동시에, 특정 요소(Element) 또는 속성(Attribute)들의 정보들을 쉽게 검색할 수 있는 가능성을 보였다.

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, 1986년 ISO(International Organization for Standardization)에서는 SGML(Standard Generalized Markup Language)이라는 문서의 논리구조를 표현하는 국제적인 표준안을 마련했다[1].

하지만 SGML은 다양한 기능에도 불구하고, 그 구성이 너무 복잡하다는 단점을 가지고 있고, 이를 해결하고자 HTML(Hypertext Markup Language)이 제기되었지만 이는 제한된 태그로 인해 한계를 가지고 있어서 사용이 부적당하였다. 이에 따라 W3C에서는 일반화된 마크업(Generalized Markup), 복합구조(Complex Structure), 검증(Validation)의 특성을 그대로 지원하는 한편 사용자에 의한 확장성(Extensibility)을 가지고 있는 XML(eXtensible Markup Language)을 제안하였다[2,3,4]. 그래서, 최근의 웹(Web) 또는 디지털 전자 도서관 시스템, CALS(Commerce At the Light Speed), 수학 분야, 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 검색 할 수 있는 필요성들이 대두되고 있다[5,6,7,8].

현재 관계형 데이터베이스상에 XML 문서를 저장하거나 추출하는 연구들이 진행 중[2]이며 이미 ADO(Active Data Object) 2.5와 SQL Server 2000에서는 각각 일차원적인 구조를 가진 레코드셋을 XML문서로 반환하거나 조인(Join)된 구조를 완벽하진 않지만 XML로 직접 추출해 내고 있다[9].

이에 본 논문에서는 XML 인스턴스들을 RDB에 저장하기 위한 모델을 제안하고 구현한다. 모델은 Edge-Labeled Graph에서 제공하는 Data Graph를 사용하여 요소(element)와 속성(attribute) 정보들을 추출한 후 데이터 경로(Data Path) 테이블과 요소와 속성 테이블을 생성하여 저장되도록 하였다. 또한 RDB에 대한 질의를 처리하기 위해, Query 변환, 그리고 RDB에서 XML 인스턴스로의 복원을 위해 DBtoXML 생성기를 갖도록 하였다.

본 논문의 구성은 다음과 같다. 2장에서는 제안하는 RDB 저장 모델을 설명하고, 3장에서는 구현 및 결과 그리고 4장에서는 결론 및 향후 연구 방향을 제시한다.

2. RDB 저장 모델

본 논문에서 제안한 시스템 모델은 하부 저장 시스템으로 ORACLE 9i를 사용하며, XML 인스턴스 분석기(Analyzer), Query 번역기, DBtoXML 생성기로 구성된다. 시스템의 전체 구조는 그림 1에서 보여준다.

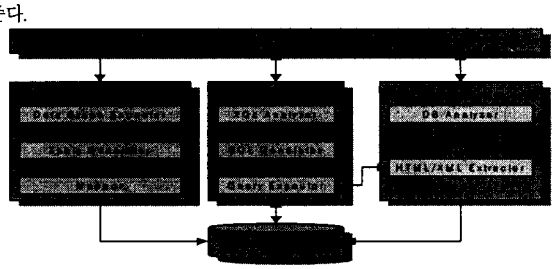

그림 1. 저장 시스템 구조

XML 인스턴스를 RDB로 저장하기 위한 본 논문의 제안 모델에서는 저장 시 필요한 물리적인 RDB의 구조를 정의하기 위해, 즉, 데이터베이스 스키마 정의를 위해 주어진 XML 인스턴스에서 다음과 같은 4가지의 주요 정보를 추출하여 데이터베이스 스키마 정의에 사용된다.

2.1 요소의 경로와 값

요소의 경로(Element Path)는 주어진 XML 인스턴스를

Edge-Labeled Graph에 데이터 그래프(Data Graph)화 한 다음 리프(leaf) 노드까지 깊이 우선 탐색 기법을 적용하여 모든 데이터가 단 한 번만 표현이 되도록 탐색하여 리프(Leaf) 노드까지의 전체 경로(Path)를 추출한다. 추출된 정보에는 리프(Leaf)까지의 엘리먼트 경로와 값이 포함되게 되고 이를 데이터 경로 테이블(Data Path Table)로 생성하도록 한다.

2.2 속성의 경로와 값

2.1절의 요소의 경로와 값을 구하는 방식으로 추출한다. 다만 속성은 필요에 의해 요소의 추가 정보를 지니면서 또한 요소 당 속성의 수는 가변적인 특징을 갖기 때문에 속성의 경로를 구할 때 이러한 점을 구분할 수 있도록 구별자를 갖도록 해야 한다. 값은 속성이 가지는 값이며 이 정보들은 데이터 경로 테이블(Data Path Table)에 포함된다.

2.3 식별자(Object Identifier)

식별자는 요소와 속성을 구별하기 위해 사용된다. Data Path Table상의 경로 값(Value)이 요소인지 속성인지를 구별하며, 또한 검색의 효율성을 위해 추가로 생성되는 요소 테이블(Element Table)과 속성 테이블(Attribute Table)을 참조할 때 참조키(Reference Key)로 사용된다. 요소는 E, 속성은 A를 선두 문자로 사용한다.

2.4 문서 식별자(Document Identifier)

문서 식별자는 여러 개의 XML 인스턴스를 구별하기 위한 식별자이다. 작업 영역 안에 처리할 XML 인스턴스가 여러 개 존재할 수 있기 때문에 이를 구별하기 위한 식별자이다. 문서 식별자는 XML 인스턴스의 URL 또는 URI를 사용하거나, 또는 이를 인덱스 하여 사용한다.

표 1. Document ID Index Table

D1	http://www.testcourse.com/ex.xml
D2	http://www.course.com/ex2.xml
…	…

2.5 데이터 경로 테이블(Data Path Table)

데이터 경로 테이블은 XML 인스턴스를 분석하여 특정 요소 또는 속성에 대해 문서 식별자(DI), 경로(Path), 식별자(OID)를 튜플로 가지는 테이블이며 XML 인스턴스를 분석한 데이터 그래프(Data Graph)의 전체 내용이 삽입되게 된다. 또한 데이터 경로 테이블은 검색의 효율을 위해 추가로 생성되는 요소 테이블과 속성 테이블에서 질의에 대한 경로를 추출할 때 참조될 테이블이기도 하다. 그림 2는 XML 인스턴스에 대한 데이터 그래프이다.

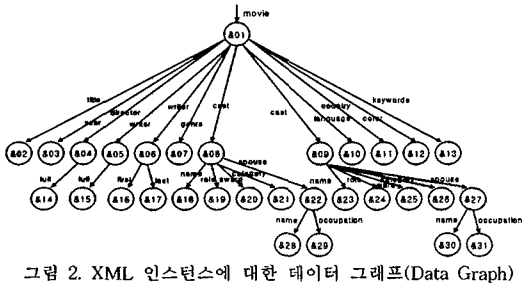

그림 2. XML 인스턴스에 대한 데이터 그래프(Data Graph)

표 2. Data Path Table

D1	E_1	movie/title
D1	E_2	movie/year
D1	E_3	movie/director/full
…	…	…

2.6 요소 & 속성 테이블(Element Table & Attribute Table)

이들은 데이터 경로 테이블(Data Path Table)에서 식별자가 요소값을 가지고 있으면 요소 테이블의 튜플로, 그리고 식별자가 속성값을 가지고 있으면 속성 테이블의 튜플로 경로 테이블을 요소와 속성에 따라 다시 재구성한 테이블이다. 요소, 속성 테이블을 별도로 두는 이유는 대부분의 XML 인스턴스에 대한 질의가 내용 검색과 구조 검색에 국한되기 때문에 질의 분석 결과에 따라 요소 또는 속성 테이블을 검색 대상으로 하는 것이 바람직하기 때문이다.

표 3. Element Table

D1	E_1	Citizen Kane
D1	E_2	1941
…	…	…

표 4. Attribute Table

D1	A_1	1
D1	A_2	2
…	…	…

2.7 테이블 인덱스 테이블(Table Index Table)

여러 개의 XML 인스턴스를 관리하기 위해서는 XML 인스턴스간 구별하기 위한 방법이 요구되어지는데, 2.4절에 기술된 문서 식별자는 하나의 XML 인스턴스와 1대1의 관계로 존재하게 된다. 그리고 이는 실제 데이터베이스내에 저장될 때 테이블들의 이름들과도 서로 연관된다. 따라서 여러 개의 XML 인스턴스를 서로 구별하기 위해서는 문서 식별자, 데이터베이스내의 테이블 이름과의 관계를 관리하는 테이블이 필요하게 된다. 그 구조는 문서 식별자(DID), Data Path Tabl(DPT)명, Element TableIET)명, Attribute Table(AT)명, Document ID Index Table(DIDIT)명을 표 5와 같이 생성한다.

표 5. Table Index Table

2.8 질의 분석

질의는 데이터베이스가 제공하는 기본 기능이된다. 따라서 본 논문에서도 저장된 XML 인스턴스에 대한 질의를 가능하도록 하였는데, 일반적으로 XML 인스턴스에 대한 검색은 크게 내용 검색, 구조 검색, 애트리뷰트 검색, 그리고 내용 + 구조 등의 혼합 검색으로 나눌 수 있다. 질의 분석 알고리즘은 그림 3과 같고, 관계 값과 엘리먼트간의 관계(Relationship)는 표 6과 같다.

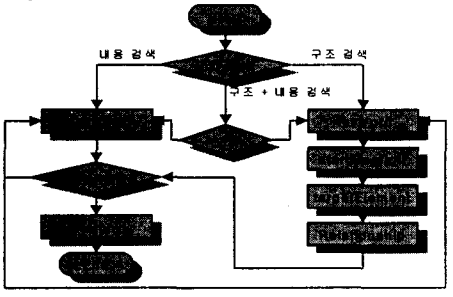

그림 3 질의 분석 알고리즘

표 6. 엘리먼트간의 관계

관계	관계값	관계 순위	관계	관계값	관계 순위
조상	조상	없음	바로 앞 형제	형제	-1
부모	조상	1	앞 형제	형제	-
자손	자손	없음	바로 뒤 형제	형제	+1
자식	자손	1	뒤 형제	형제	+

2.9 XML 인스턴스 생성기

이는 데이터베이스내에 저장된 XML 인스턴스를 원래의 XML 인스턴스로 추출한다. 2.8절의 질의 결과에 대한 내용을 HTML형식과 XML 형식으로 사용자에게 제공함과 동시에 데이터베이스에서 직접 추출하여 전체 XML 인스턴스 내용을 제공하는 방식을 지원하기 위해 필요하다.

3. 시스템 구현 및 결과

3.1 질의 인터페이스

그림 4는 위에서 설명한 내용 검색 및 속성 검색 그리고 구조 검색과 구조 + 내용 검색을 수행하기 위한 사용자 인터페이스 화면이다. 이는 「Citizen Kane를 포함하는 첫 번째 Title의 부모 엘리먼트를 찾아라」라는 구조 + 내용 질의 인터페이스를 보여주며, 그림 5는 그림 4의 질의 처리 결과이다.

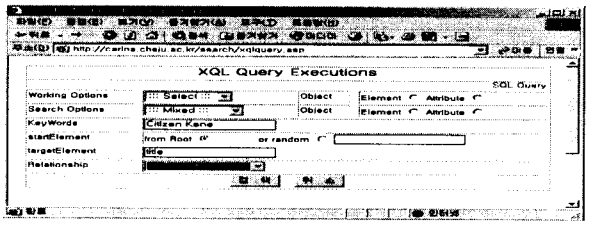

그림 4. XQL 질의 인터페이스

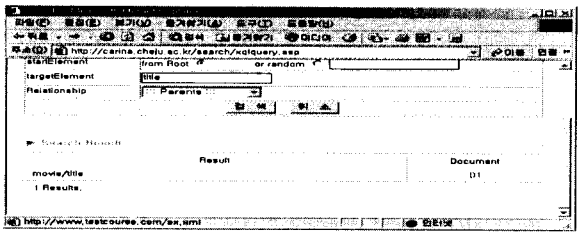

그림 5. 그림 4의 질의 처리 결과

3.2 DBtoXML Generator(DXG)

DXG는 검색 결과에 대한 XML 형식의 추출과 데이터베이스내에서의 직접 추출 형식을 지원하도록 하였다. XML 형식의 추출은 3.1절의 검색 결과를 HTML 형식과 XML 형식으로 구분하여 출력하는 과정에서 수행되며, 여기에서는 직접 추출 과정을 그림 6과 같은 사용자 인터페이스를 이용하여 추출할 데이터베이스를 선택하며, 선택된 값에 따라 현재 데이터베이스내의 테이블 인덱스 테이블(TIT)를 조회하고 해당되는 테이블의 Element, Attribute 테이블과 이들의 Path 테이블을 JOIN하여 그 결과를 그림 7과 같이 보였다.

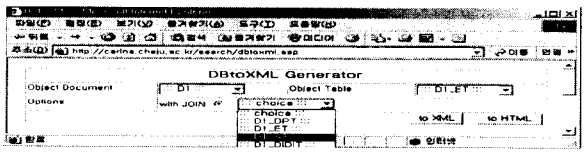

그림 6. DBtoXML 생성기

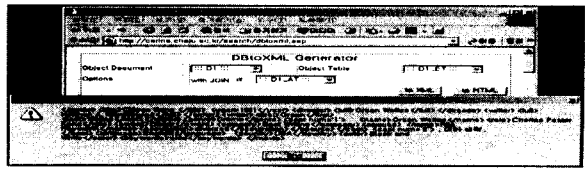

그림 7. XML 문서 생성

4. 결 론

본 논문에서는 대용량의 XML 인스턴스 관리 시스템을 개발하는데 있어서 기반 기술이 되고 있는 XML 저장 관리 시스템 모델을 제안하고 구현하였다. 이를 위해 Edge-Labeled Graph를 기반 한 Data Graph를 적용하였으며, 저장 모델에 필요한 각 프로세스의 세부적인 기능을 분석하고 정의하였다.

XML 인스턴스의 효율적인 저장과 관리를 위한 XML Document Analyzer는 XML 인스턴스를 분석하고 이를 데이터 경로 테이블(Data Path Table)과 Element, Attribute, Table Index Table을 생성하도록 구현 되였으며, 검색을 위한 Query에는 내용 검색(속성 검색 포함)과 구조 검색, 그리고 구조 + 내용 검색이 가능하도록 하였다. 또한 검색 결과에 대한 결과를 HTML과 XML 형식과 데이터베이스에서 직접 추출하는 형식을 지원하도록 검색 인터페이스를 구현하여 테스트하였다. 테스트 결과 Data Graph에 의한 경로(Path)기반으로 엘리먼트와 속성들의 구조 정보가 구축(계층 구조의 특성을 유지함)됨으로 인해 특정 엘리먼트를 쉽게 검색할 수 있었으며 다양한 XML 질의를 처리할 수 있는 가능성을 보였다. 특히, XML 인스턴스를 Data Graph의 경로(Path)를 기반으로 한 RDB내의 저장을 보였다. 향후 연구 과제로는 XML 구조는 트리 형태의 계층적 구조로 표현될 수 있으며 이러한 원리에 의한 Data Path Graph와 현재 중점적으로 연구되고 있는 객체지향 데이터베이스와의 연동에 있다.

참고문헌

[1] 손정한, 이희주, 장재우, 심부성, 주종철 "구조화된 문서를 위한 정보검색시스템의 설계 및 구현", '98 동계 데이터베이스 학술대회 논문집 제14권 1호, PP102-106, 1998

[2] 연제원, 장동준, 김용훈, 이강찬, 이규철. "효율적인 검색 지원 SGML 저장 관리기의 설계 및 구현", '99 한국 데이터베이스 학술대회 논문집 15권 1호, pp136-143. 1999

[3] 유재수외 8명, "전자도서관 표준문서관리를 위한 XML 저장관리기 기술 개발", 케이오텍 최종보고서, 1999.

[4] 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), pp43-56, 1995.

[5] Dongwook Shin, Hyuncheol Jang, and HongLan Jin "Bus : An Effective Indexing and Retrieval Schema in Structured Documents", ACM. pp. 235-243, 1998.

[6] Francois. "Generalized SGML repositories: Requirements and Modeling", Computer Standards & Interfaces, 1996.

[7] Tuong Dao, Ron Sacks-Davis, James A.Thom. "An indexing scheme for structured documents and its implementation", Proceedings of the 4th International Conference on DATABASE Systems for Advanced Applications, Melbourne, Australia. pp.125-135, 1997.

[8] 맹성형, 주종철. "문서 구조화와 정보 검색", 정보과학회지, 제16권, 제8호 1998. 8.

[9] Kevin Williams 외 9인 공저 "Professional XML Databases" Wrox, 2001.