구조 및 의미적 유사성에 기반한 XML 문서들의 효율적인 저장을 위한 통합 기법

김연희*, 김병곤**, 이재호***, 임해철*
*홍익대학교 컴퓨터공학과
**부천대학 e-비즈니스과
***인천교육대학교 컴퓨터교육과
{kyh, lim}@cs.hongik.ac.kr, bgkim@bc.ac.kr, jhlee@mail.inue.ac.kr

The study of integration techniques for storing XML documents efficiently based on structures and semantics

Youn Hee Kim*, Byung Gon Kim**, Jaeho Lee***, Hae Chull Lim*
*Dept. of Computer Engineering, Hong Ik University
**Dept. of e-Business, Bucheon College
***Dept. of Computer Education, Inchon National University of Education

요 약

최근 XML이 웹 상의 데이터의 표현, 교환, 중재의 표준으로 각광받으면서 이러한 XML 문서를 효과적으로 저장, 접근 및 검색하기 위한 기법에 대한 연구가 많았으나, 기존의 연구들은 하나의 XML 문서를 저장 및 검색의 대상으로 하는 경우가 대부분이였다. 그러나 XML 문서를 데이터의 표현과 교환의 표준으로 이용하는 애플리케이션의 개발이 점차 활성화됨에 따라 저장해야하는 XML 문서의 수가 크게 증가하면서 의미나 구조적으로 많은 유사성을 지니는 XML 문서들을 함께 효율적으로 저장하고 검색하기 위한 기법의 연구가 요구된다. 따라서 본 논문에서는 의미 및 구조적으로 유사성을 가지는 여러 XML 문서들을 통합하는 기법을 제안한다. 제안된 통합 기법은 같은 DTD나 XML Schema를 가지는 경우와 다른 DTD나 XML Schema를 가지는 경우를 모두 고려한다. 또한 특별한 구조적 정보를 가지지 않는 XML 문서의 경우도 다른 DTD나 XML Schema를 가지는 경우와 마찬가지로 처리함으로써 다양한 XML 문서들에 대한 통합이 가능하도록 한다. 이러한 통합 기법은 중복되는 엘리먼트나 애트리뷰트에 대한 저장 공간의 낭비를 최소화한다. 또한 의미적으로 또는 구조적으로 관련성있는 여러 XML 문서의 부분들을 디스크 상의 페이지내에 서로 가까이 저장할 수 있기 때문에 사용자의 일반적인 질의에 대해 효율적이고 빠른 검색 결과를 유도할 수 있고, I/O 횟수를 줄임으로써 그에 따른 오버헤드를 줄일 수 있는 장점이 있다.

1. 서 론

HTML과는 달리 데이터의 내용과 구조를 모두 표현하는 자기 기술적(self-describing) 특징을 가지고 있는 XML(eXtensible Markup Language)은 최근 웹 기반 애플리케이션 개발 시 데이터의 표현 및 교환의 표준으로서 각광을 받고 있다. 이러한 변화에 따라 XML과 관련한 다양한 연구의 필요성이 증대되고 있는데, 특히 XML로 표현된 문서를 효과적으로 저장, 접근 및 검색하기 위한 많은 연구들이 활발히 진행되어 왔다[1,2].

XML 문서의 저장 및 검색에 관한 연구는 크게 상용 DBMS을 이용하는 방법과 XML 전용 저장 기법에 대한 연구로 나뉠 수 있다[3]. 초기에는 상용 DBMS를 기반으로 개발된 기존의 많은 응용 시스템들과의 손쉬운 연동이 가능하다는 장점때문에 RDBMS나 ODBMS/ORDBMS를 이용한 XML 저장 및 검색 기법에 대한 연구가 많이 이루어졌다[4]. 하지만 이러한 경우에는 XML 문서 자체가 내포하고 있는 의미와 구조적 정보를 완벽하게 표현하기 어렵고, 갱신 연산이나 질의 처리가 복잡하며 XML 문서의 재구성 비용이 추가로 발생하는 등의 단점이 있다. 따라서 최근에는 XML 문서 자체가 내포하고 있는 의미와 구조적 정보를 완벽히 지원할 수 있는 XML 전용의 저장 시스템에 대한 관심이 높아지고 있다[5,6,7]. 그러나 지금까지 연구된 XML 전용 저장 기법들은 하나의 XML 문서를 저장의 대상으로 하는 경우가 대부분이였다. 이러한 이유로 의미나 구조적으로 많은 유사성을 지니는 XML 문서들을 저장하는 경우에도 특정 데이터 모델에 따라 각각 독립적으로 저장함으로써 실제적인 저장 및 검색 측면에서 많은 비효율성을 초래하는 단점이 있다.

따라서 본 논문에서는 의미 및 구조적으로 유사한 여러 XML 문서들을 효율적으로 저장하고 검색하도록 하는 통합 기법을 제안한다. 본 논문에서 제안한 통합 기법은 각 XML 문서를 독립적으로 저장하는 대신, 의미 및 구조의 유사성에 기반하여 마치 하나의 문서처럼 통합하여 저장함으로써 중복되는 엘리먼트나 애트리뷰트로 인한 저장 공간의 낭비를 줄이고, 의미적으로 또는 구조적으로 관련성이 많은 각 XML 문서의 부분들을 저장 장치 내에 가깝게 위치시킴으로써 검색 시 빠르고 정확한 검색을 가능하게 한다.

2. 관련 연구

XML 전용 저장 기법에 대한 연구는 기존 상용 DBMS를 이용하는 XML 저장 기법이 지닌 문제점을 해결하기 위해 제안된 것이다. 즉, XML이 내포하고 있는 내용과 의미뿐만 아니라 그 구조를 완전하게 표현하기 위해 XML 문서 자체를 위한 데이터 모델을 논리적으로 정의하고, 이 모델에 따라 XML 문서를 저장하고 검색함으로써 XML 문서의 재구성 비용이나 추가적 질의 처리 비용을 줄이고 효과적으로 XML 문

본 연구는 한국과학재단 기초과학연구사업 (과제번호 : R01-2002-000-00080-0(2002)의 지원을 받아서 작성되었음

서를 저장하고 검색할 수 있도록 하는데 그 목적을 두고 있다. 이와 같은 XML 전용 저장 기법을 적용한 대표적인 저장 시스템으로 Lore 시스템과 NATIX 시스템 등을 들 수 있다.

Lore 시스템은 매우 간단하고 중첩이 가능하며 자기 기술 능력을 갖는 OEM(Object Exchange Model) 데이터 모델에 따라 각각의 XML 문서를 독립적으로 변환하여 디스크에 저장하고 새롭게 제안한 Lorel 질의어를 이용해 검색이 가능하도록 한다[5].

Latix 시스템은 XML 문서의 논리적 데이터 모델을 순서가 있는 트리 구조로 정의하고 이를 기반으로 XML 문서를 그 자체의 의미와 구조적 정보가 그대로 유지하도록 저장한다. 특히 검색 시 디스크 I/O 횟수를 줄이고 검색 시간의 향상을 위해 페이지 단위보다 큰 크기의 XML 문서를 물리적으로 저장할 때 의미적으로 관련있는 노드를 같은 페이지에 저장할 수 있도록 하는 클러스터링 알고리즘을 제안하였다[6].

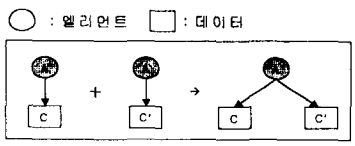
Lore 시스템과 Latix 시스템은 간단한 데이터 모델을 정의하고 이에 따라 XML 문서의 의미와 구조적 특성을 지원하는 전용 저장 기법을 제안한다. 하지만 독립적인 하나의 XML 문서를 저장과 검색의 대상으로 하기 때문에 의미와 구조적으로 유사함에도 불구하고 여러 XML 문서들을 독립적으로 디스크 페이지 내에 저장함으로써 저장 공간이 낭비되고 검색이 비효율적으로 이루어지는 단점이 있다. 따라서 의미와 구조적으로 유사한 여러 XML 문서를 효율적으로 함께 저장하고 검색하기 위해서는 하나의 XML 문서 내에 또는 여러 XML 문서간에 중복되는 엘리먼트나 애트리뷰트을 통합하고, 유사한 XML 문서의 부분들을 가까운 디스크 페이지내에 클러스터링하여 저장할 필요성이 있다.

3. 문서 통합 기법

이 논문에서는 유사한 의미와 구조를 가지는 여러 XML 문서들의 효율적인 저장과 검색을 위한 XML 전용의 통합 문서 저장 기법을 제안한다. 유사한 의미와 구조를 가지는 XML 문서들은 그 유사성의 정도에 따라 크게 두 개의 집합으로 분류할 수 있고, 각 집합에 따라 다음과 같은 통합 기법이 고려된다.

3.1 같은 DTD나 XML Schema를 가지는 XML 문서 집합

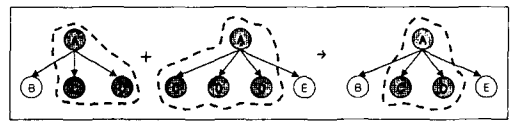
(1) 통합하려는 XML 문서간의 엘리먼트나 애트리뷰트의 데이터 값이 다른 경우는 공통된 엘리먼트나 애트리뷰트의 자식으로 모든 데이터 값을 저장한다. <그림 1>은 A라는 같은 이름의 엘리먼트를 중심으로한 데이터의 간단한 통합의 예를 보여준다.



<그림 1> 같은 이름의 엘리먼트에 대한 통합

(2) 통합하려는 XML 문서 간에 동일한 부모 엘리먼트를 가지는 자식 엘리먼트나 애트리뷰트의 출현 유무가 다른 경우에는 하나의 부모 엘리먼트가 모든 애트리뷰트나 자식 엘리먼트를 포함하도록 한다. <그림 2>에서 첫 번째 문서의 B 엘리먼트와 두 번째 문서의 E 엘리먼트가 통합 저장될 문서에는 모두 표현되어 있음을 확인할 수 있다.

(3) 통합하려는 XML 문서 간에 동일한 부모 엘리먼트를 가지는 자식 엘리먼트는 그 출현 횟수에 상관없이, 여러번 나타나는 자식 엘리먼트는 하나로 통합하여 저장한다. 이때, 각 문서 내에 자식 엘리먼트간의 순서 정보는 공통으로 나타나는 엘리먼트를 기준으로 원래 문서의 순서 정보를 그대로 유지하도록 한다. <그림 2>에서 A 엘리먼트의 자식 엘리먼트 C와 D는 두 문서에 모두 공통적으로 포함되어 있고, 특히 D 엘리먼트의 경우 그 출현 횟수에 차이가 있다. 따라서 실제로 통합하여 저장할 때는 <그림 2>의 오른쪽 트리 구조에서 처럼 동일한 엘리먼트는 하나의 엘리먼트로 통합하여 저장한다.

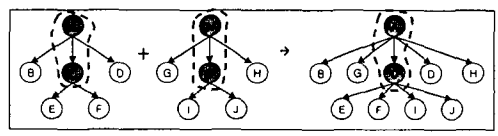


<그림 2> 자식 엘리먼트의 출현 유무 및 빈도에 따른 통합 예

3.2 다른 DTD나 XML Schema를 가지는 XML 문서 집합

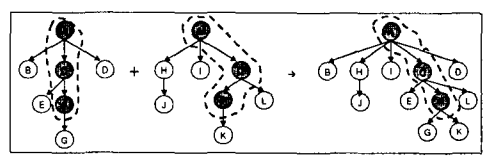
의미와 구조적으로 아주 유사하나, 다른 DTD나 XML Schema를 가지거나 또는 그러한 구조 정보를 따로 가지지 않는 여러 XML 문서에 경우에는 아래와 같은 통합 기법을 적용한다.

(1) 통합하려는 XML 문서간에 동일한 부모/자식 엘리먼트의 관계를 가지는 경우에는 그러한 관계를 중심으로 통합하여 저장한다. 이때, 같은 부모 엘리먼트를 가지는 서로 다른 문서 내의 모든 자식 엘리먼트는 하나의 부모 엘리먼트를 기준으로 통합된다. <그림 3>은 첫 번째 문서의 A 엘리먼트와 C 엘리먼트의 부모/자식 관계와 두 번째 문서의 A 엘리먼트와 C' 엘리먼트의 부모/자식 관계에 의한 구조적 유사성과 엘리먼트 이름의 유사성에 기반한 간단한 통합의 예를 보여준다. C와 C' 엘리먼트는 표면적인 이름의 차이는 있으나 표현하고자 하는 의미가 같으므로 같은 이름으로 처리하되, 통합 저장 문서 형태에는 대표 이름만을 사용한다.



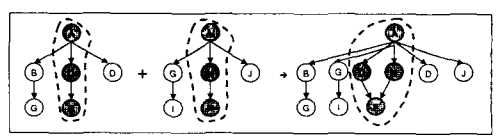
<그림 3> 부모/자식 관계의 유사성에 기반한 통합 예

(2) 통합하려는 XML 문서간에 동일한 다중 레벨의 경로 정보를 가지는 경우에는 공통의 경로 관계를 중심으로 통합하여 저장한다. 자식 엘리먼트의 처리나 이름 충돌의 경우에는 (1)과 마찬가지로 처리한다. <그림 4>는 첫 번째 문서의 A, C, F 엘리먼트의 경로 정보와 두 번째 문서의 A, C', F 엘리먼트의 경로 정보의 유사성에 기반하여 통합한 문서의 예를 보여준다.



<그림 4> 다중 경로의 유사성에 기반한 통합 예

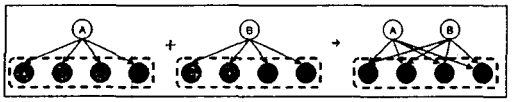
(3) 통합하려는 XML 문서간에 동일한 조상-자손 엘리먼트의 관계를 가지는 경우에도 공통의 관계를 중심으로 통합하여 저장한다. <그림 5>는 A 엘리먼트와 F 엘리먼트 간의 조상/자손 관계를 중심으로 두 XML 문서를 통합한 예이다. A 엘리먼트와 F 엘리먼트는 통합된 문서의 형태에서 단 한번만 나타나게 된다.



<그림 5> 조상/자손 관계의 유사성에 기반한 통합 예

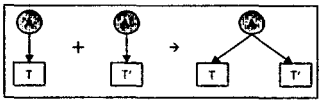
(4) 통합하려는 XML 문서간에 서로 다른 부모 엘리먼트를 중심으로 같은 형제 엘리먼트 관계를 가지는 경우에는 형제 엘리먼트는 통합하되, 두 개의 부모 엘리먼트를 그대로 유지하도록 저장한다. <그림 6>은

형제 관계를 이루는 C, D, E, F 엘리먼트의 유사성에 기반하여 두 문서를 통합한 예를 보여준다.



<그림 6> 형제 관계의 유사성에 기반한 통합 예

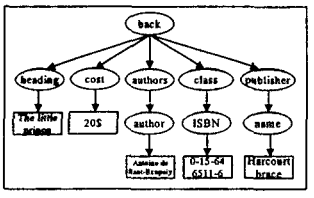
(5) 통합하려는 XML 문서간에 같은 엘리먼트(애트리뷰트)를 중심으로 서로 다른 엘리먼트(애트리뷰트) 데이터 값을 가지는 경우 동일 엘리먼트(애트리뷰트)를 중심으로 통합 저장한다. <그림 7>은 두 문서 내에 같은 의미를 가지는 A와 A' 엘리먼트를 중심으로 데이터를 통합한 예를 보여준다.



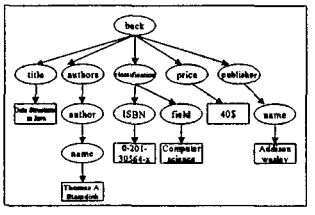
<그림 7> 엘리먼트 데이터에 대한 통합 예

4. 예제

<그림 8>와 <그림 9>는 본 논문에서 제안한 통합 기법을 적용하기 위한 대상 XML 문서를 간단히 트리 형태로 표현한 것이다. <그림 8>와 <그림 9>의 XML 문서는 책에 대한 정보를 표현한 것으로 저자, 가격, 출판사, 내용 소개, 분류 등의 내용을 담고 있다.

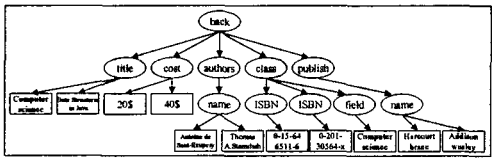


<그림 8>어린왕자에 대한 XML 문서 예



<그림 9>자료구조에 대한 XML 문서 예

<그림 10>은 <그림 8>와 <그림 9>의 XML 문서에 본 논문에서 제안한 통합 기법을 적용한 결과로 실제 두 문서를 디스크 페이지 내에 저장할 때 이와 같이 통합된 형태로 저장됨으로써 저장이나 검색의 측면에서 효율성을 얻게된다.



<그림 10> 두 XML 문서의 구조 및 의미적 통합 결과

5. 결 론

본 논문에서는 대량의 XML 문서들을 효율적으로 저장 및 검색하기 위하여 구조 및 의미에 기반한 XML 문서의 통합 기법을 제안하였다. 제안한 통합 기법은 의미 및 구조적으로 유사한 XML 문서들을 마치 하나의 XML 문서처럼 통합하여 저장한다. 의미 및 구조적으로 유사한 여러 XML 문서들을 저장할 때 우선, 동일한 엘리먼트나 애트리뷰트의 중복된 저장을 피할 수 있어 저장 공간의 낭비를 막을 수 있다. 또한 많은 관련성을 지닌 여러 XML 문서의 부분들을 디스크 상의 같은 페이지내에 저장할 수 있도록 클러스터링하기 때문에 사용자의 일반적인 질의에 대해 효율적이고 빠른 검색 결과를 유도할 수 있고, I/O 횟수를 줄임으로써 그에 따른 오버헤드를 줄일 수 있는 장점이 있다. 그러나 이러한 통합 기법의 장점을 유지하지 위해서는 XML 문서 내의 의미는 물론 부모/자식, 조상/자손, 형제 등의 구조적 경로 정보들을 그대로 표현하기 위한 저장 모델이 필요하다. 특히 여러 XML 문서들을 통합하여 저장하지만 각 문서에 대한 독립된 정보도 필요하기 때문에 각각의 문서 내용에 대한 구별과 출현 횟수, 각 XML 문서 내의 순서 정보등도 함께 유지할 필요가 있다. 또한 통합 시 발생할 수 있는 엘리먼트 이름의 충돌 문제를 해결하기 위해서는 같은 의미로 다른 이름이 쓰인 경우에, 대표로 사용되는 이름 외의 이름 정보도 저장 모델 내에 유지되어야 한다. 따라서 이러한 요구 조건을 만족하는 저장 모델에 대한 연구를 현재 진행 중에 있다.

참고 문헌

[1] Jennifer Widom, "Data Management for XML", IEEE Data Engineering Bulletin Special issue on XML, Vol. 22, No. 3, pp. 44-52, September 1999.

[2] Stefano Ceri, Piero fraternali and Stefano Paraboschi, "XML: Current Developments and Future Challenges for the Database Community", Proc. of the 7 th Int. Conf. on EDBT, pp. 3-17, March, 2000.

[3] Ronald Bourret, "XML and Databases", January, 2003.

[4] D. Florescu and D. Kossmann, "Storing and Querying XML Data using an RDBMS", Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, 1999.

[5] R. Goldman, J. McHugh, and J. Widom, "From Semistructured Data to XML: Migrating the Lore Data Model and Query Language", Proc. of 2nd International Workshop on the Web and Database, 1999.

[6] Carl-Christian Kanne, Guido Moerkotte, "Efficient storage of XML data", Technical Report, University of Mannheim, 1999.

[7] SungWan Kim, Youn Hee Kim, Jaeho Lee, and HaeChull Lim, "Developing a Native Storage Structure for XML Repository System in Main Memory", Proc. of the 5th IEEE International Conference on High Speed Networks and Multimedia Communications, pp. 96-100, July, 2002.

[8] Chantal Reynaud, Jean-Pierre Sirot, and Dan Vodislav, "Semantic Integration of XML Heterogeneous Data Sources", Proc. International Database Engineering & Applications Symposium(IDEAS '01) , July, 2001.

[9] Yasuo Yamane, Nobuyuki Igata, Isao Namba, "High-performance XML Storage/Retrieval System", FUJITSU SCIENTIFIC & TECHNICAL JOURNAL, vol 36-2.