효율적인 구조 정보 검색를 위한 색인 모델

고혜경[v] 조윤기 조정길 이병렬 구연설
충북대학교 전자계산학과
{ellefgt, ykcho}@selab.chungbuk..ac.kr
cho0530@chollian.net
inanet@splinux.co.kr
yskoo@cbucc.chungbuk.ac.kr

An Indexing Model for Efficient Structure Information Retrieval

Hye-Kyung KO[v] Yun-Kee Cho Jung-Gil Cho Byung-Real Lee Yeon-Seal Koo
Dept. of Computer Science, Chungbuk National University

요 약

본 논문에서는 XML 문서의 효율적인 관리와 구조검색을 위한 구조적 색인 방법을 제안한다. 기존의 방법은 특정 엘리먼트의 부모, 자식, 형제에 대한 다양한 구조검색을 효율적으로 지원하지 못하므로, XML 문서의 구조정보를 LETID(Labeled Element Type ID)로 표현하여 엘리먼트를 식별하고, 부모와 자식 엘리먼트간의 계층 정보와 동일한 부모 엘리먼트를 갖는 자식 엘리먼트들의 순서정보를 나타낸다. LETID는 고정된 크기로 하며, 엘리먼트에 고유 ID를 부여하는 방식을 통해서 DTD의 논리적 구조를 분석할 때 부모, 형제 노드를 직접적으로 찾을 수 있고 ID 값에 깊이정보가 포함되어 있기 때문에 고유번호만 보고 깊이를 알 수 가 있다. 이 구조정보를 이용하여 빠른 검색을 위한 내용 색인, 구조 색인, 애트리뷰트 색인을 설계하고, 설계된 색인을 통하여 질의를 처리하여 다양한 구조적 질의를 효율적으로 처리할 수 있다 .

1. 서론

최근 인터넷이 발전함에 따라 웹 문서의 중요성이 부각되면서 XML에 대한 다양한 연구가 활발히 진행되고 있다. 현재 인터넷상의 대부분 정보는 HTML 문서로 구성되어 있기 때문에 구조적인 문서를 표현하기에는 부족하다. 이의 대안으로 제시된 인터넷 전자 문서의 표준이 XML인데, XML은 HTML의 단점을 극복하고, 이기종간의 시스템에서 작성된 문서의 상호 교환과 다양한 형식의 문서들을 일관성 있게 구조화하기 위해 고안된 SGML을 웹 상에서 원활히 사용할 수 있도록 간략화 시킨 표준안이다.[2][6]. XML 문서는 하나의 문서에 내용 정보와 구조 정보를 가지고 있다. 따라서, 기존의 문서에서 제공하던 내용 정보에 대한 검색뿐만 아니라 논리적인 구조 정보에 대한 검색 기능도 필요하다. 대부분 XML 문서의 구조정보 표현 방법은 조상, 자손, 형제의 관계 노드를 쉽게 알 수 없고 엘리먼트에 접근하기 위해 복잡한 연산을 수행한다.[2][5].

이에 본 논문은 특정 엘리먼트를 직접적으로 탐색할 수 있고 *DTD*에 나타난 엘리먼트들과 XML 문서의 구조정보를 사용하여 엘리먼트간의 관계를 구한다. 따라서 복잡한 연산 없이 XML 문서의 검색이 가능하다. 이 정보들을 이용하여 효율적인 검색을 하기 위한 색인 구조를 구성하는 색인 모델을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 XML 문서를 검색하는 방법에 대한 기존 색인 모델들에 대한 연구들을 살펴보고 3장에서는 효율적인 구조정보 표현을 제안하고 구조 정보 추출 과정을 보여준다. 4장에서는 추출된 구조정보를 이용하여 색인을 구성하고 질의 처리 방법을 보여준다. 마지막으로 5장에서는 결론 및 향후 연구 방향을 제시한다.

2. 관련연구

구조화된 문서 관리 시스템은 문서 단위 검색과 엘리먼트 단위의 검색이 필요하며, 문서의 구조에 의한 구조검색과 엘리먼트의 특성 값에 대한 질의가 필요하다. 색인구조에 대한 기존의 연구는 다음과 같다.[6][7]. K-ary 완전트리를 이용하는 방법[4]은 계산에 의해서 엘리먼트를 빠르게 찾을 수 있으나, 부분 삽입과 부분 삭제 시에 노드 번호가 대부분 바뀌므로 연산 오버헤드가 크다. SCL 모델[5]은 정의된 문서 계층과 정의된 마크업 스키마로부터 독립적이나 색인어가 포함된 엘리먼트에 대해 트리 내의 깊이를 표현할 수 없다. Subtree 모델은 XML문서를 Subtree 형태로 표현한 후 여기에 나타나는 모든 단위에 대해 중복 색인을 하고 색인어가 위치하는 단위를 기록하는 방법을 제안하였는데, 추출된 색인어가 나타난 단위의 모든 상위 단위에 대해서도 색인을 하게 되어 공간상에 중복이 일어난다. 추상화에 기반 한 방법[4]은 추상화에 의해 색인의 크기를 줄일 수 있으나 추상화되지 않은 구조의 검색 시 문서 전체를 읽어야 하므로 오버헤드가 크다. 이러한 색인의 문제점을 줄이기 위해 엘리먼트에 ID를 부여하여 구조적 검색을 하는 방법이 제안되었는데 이 방법은 특정 엘리먼트에 바로 접근은 가능하지만 트리의 깊이가 깊어질수록 노드의 자리수가 2바이트씩 계속 증가되기 때문에 저장 공간이 커지고, 구조 문서 검색시간이 오래 걸리는 단점이 있다.[1]

3.구조정보 표현

본 논문에서는 XML 데이터 모델에서 색인정보나 데이터베이스의 접근을 최대한 줄이는 방안으로 객체의 ID에 문서의 구조 정보를 갖게 하는 방법을 사용했다. 엘리먼트에 고유ID를 부여하여 부모 엘리먼트, 형제 엘리먼트, 동일한 타입의 엘리먼트에 대한 순서정보를 표현했다. 이 ID는 각각의 엘리먼트에 대하여 모두 다른 값을 가지고 ID 자체내에 깊이를 포함하고 있기 때문에 특정 엘리먼트를 복잡한 연산

없이 직접 찾는다.

3.1 LETID

본 논문은 엘리먼트들 간의 계층 정보를 표현하기 위해 엘리먼트를 식별할 수 있는 LETID(Labeled Element Type ID)로 계층 정보를 표현하였다. LETID는 DTD에서 정의된 각 엘리먼트에 대한 고유값을 나타낸다. 엘리먼트에 고유 ID를 부여하는 방식을 통해 DTD의 구조를 분석할 때 부모, 형제 노드를 직접 찾을 수 있고 LETID에 깊이 정보가 있어서 고유번호만으로 깊이를 알 수 있다. LETID는 모든 엘리먼트의 노드를 고정된 8바이트로 표현하는데 각 바이트는 ASCII code의 순서를 따르는 '0'→'9'→'A'→'Z'→'a'→'z'순으로 구성된다. (그림 1)은 DTD에서 정의된 각각의 엘리먼트에 대해 LETID를 부여하고 LETID 매핑 테이블을 표현한 것이다.

Paper DTD

```
<!-- paper -->
<!Element paper(head,section+,reference+)>
<!AttList paper status(public | confidential) "public">
<!-- head -->
<!Element head (subject,author+,abstract)>
<!Element subject(#PCDATA)>
<!Element author (#PCDATA)>
<!Element abstract (%Pcdata)>
<!-- section -->
<!Element section ((title,para+)+,subsection*)>
<!Element title (%Pcdata)>
<!Element subsection (title?,para+)>
<!Element para (%Pcdata | table | img)*>
<!-- reference -->
<!Element reference (num,name*)>
<!AttList reference  refid ID #REQUIRED>
<!Element num (#PCDATA)>
<!Element name (#PCDATA)>
```

LETID mapping Table

Element Type	LETID
paper	00000100
head	01000211
section	01000221
reference	01000231
subject	02110311
author	02110321
abstract	02110331
title	02210311
para	02210321
subsection	02210331
num	02310311
name	02310321

(그림 1) LETID 부여 방법

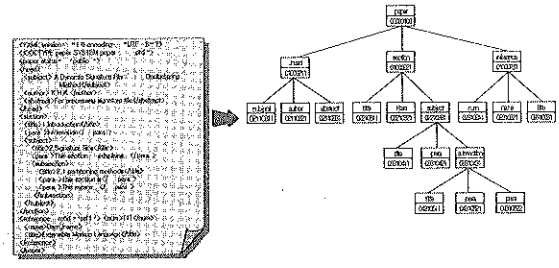
3.2 구조정보 표현 방법

XML 문서의 구조정보의 표현방법은 엘리먼트를 기반으로 표현한다. 본 논문에서는 LETID를 부여하여 특정 엘리먼트를 구별하고 엘리먼트의 위치정보를 알 수 있는 계층정보를 표현하였다.

자리수(8바이트)	나타내는 정보
12	부모 노드의 계층 정보
34	부모노드의 형제 노드에 대한 순서 정보 4번째 바이트는 동일한 형제노드에 대한 순서 정보
56	현재 노드의 계층 정보
78	현재 노드의 형제 노드에 대한 순서 정보 8번째 바이트는 동일한 형제노드에 대한 순서 정보

(그림 2) LETID 표현 정보

(그림 2)와 같이 LETID는 고정된 8바이트로 표현이 되며 각각 2바이트로 엘리먼트의 부모, 자식간의 계층정보와 형제간의 순서정보를 나타낸다. 앞의 4바이트(1234)는 부모노드의 계층정보와 부모노드의 형제 노드에 대한 순서 정보를 담고 있다. 이중 뒤에 2바이트(34)는 형제 노드에 대한 순서 정보와 동일한 타입의 형제 노드들 간의 순서 정보를 나타낸다. 마찬가지로 뒤의 4바이트(5678)는 현재 노드의 계층 정보와 현재 노드의 형제 노드에 대한 순서 정보를 나타내 준다. LETID는 각 엘리먼트들에 유일한 LETID 값을 부여하기 때문에 구조적 검색을 지원할 수 있고 반복적인 엘리먼트에 대해서도 서로 다른 LETID값을 부여하기 때문에 다양한 구조검색이 가능하다.

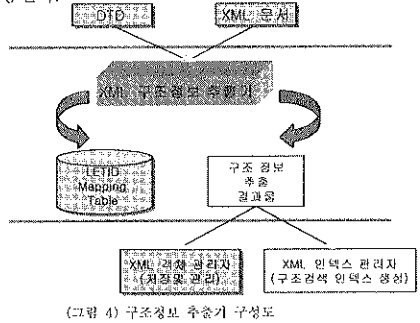


(그림 3) XML 문서의 구조 정보 표현 트리 구조

(그림 3)은 들어 온 XML 문서의 구조정보를 LETID를 이용하여 트리로 표현한 예이다. 위 트리는 엘리먼트들 간의 계층정보를 나타내며 LETID값이 문서의 구조상의 각 엘리먼트에 부여되는 고유의 값인 것을 알 수 있다. 계층 정보 표현 방법을 보면, 예를 들어 (그림 3)에서 루트 엘리먼트인 parer의 LETID는 "00000100"이고, head의 LETID는 "01000211"이다. (그림 2)에서 LETID를 표현한 방법을 보면 paper의 뒤 4바이트(5678)가 head에 상속되어 head의 앞 4바이트(1234)가 "0100"이 된다. 이렇게 각 엘리먼트들은 부모의 정보를 상속받아 LETID만 가지고 엘리먼트의 계층정보를 검색한다. 또한 XML 문서에서는 반복적인 엘리먼트 사용이 가능하기 때문에 동일한 엘리먼트들이 반복적으로 나타날 수 있다. 따라서 8바이트 중에 "4,8"번째 바이트에 동일한 형제 노드에 대한 순서 정보를 두어 구별한다. 예를 들면 (그림 3)에서 subsection의 자식노드를 보면 para가 두 번 반복된다. para의 값은 "04310521", "04310522"로 각각 표현된다. 뒤 4바이트(5678)는 "0521"과 "0522"로 다르게 표현된다. 여기서 "05"은 현재 노드의 계층 정보를 표현하고, "21"과 "22"는 현재 노드의 형제 노드에 대한 순서 정보를 나타내는데 8번째 바이트를 "1", "2"로 표현하여 동일한 형제 노드의 순서 정보를 표현한다. 결과적으로 모든 엘리먼트는 서로 다른 LETID값을 갖고, LETID만으로 특정 엘리먼트가 검색된다.

3.3 구조 정보 추출 과정

XML 문서의 구조정보는 구조정보 추출기에 의해서 얻어진다. 구조정보 추출기는 DTD와 XML 문서로부터 엘리먼트 Type을 추출하고 엘리먼트 이름과 엘리먼트의 ID를 연결시켜주는 LETID Mapping Table을 구성한다. 구조정보 추출기에 의해 추출된 정보는 XML문서의 색인 정보에 이용된다.



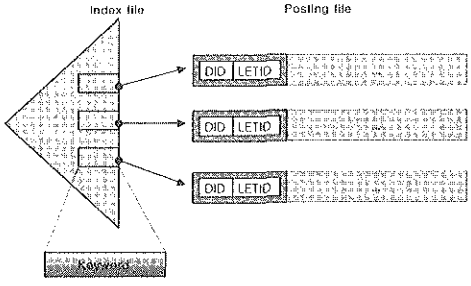
(그림 4) 구조정보 추출기 구성도

4. XML 색인 구성 및 질의 처리

XML 문서의 특성을 고려하여 다양한 질의를 효율적으로 처리할 수 있는 색인 구조를 설계한다. 색인은 내용 색인, 구조색인, 에트리뷰트 색인의 3가지로 구성되며 기존의 역파일 방식을 사용한다.

4.1 내용 색인

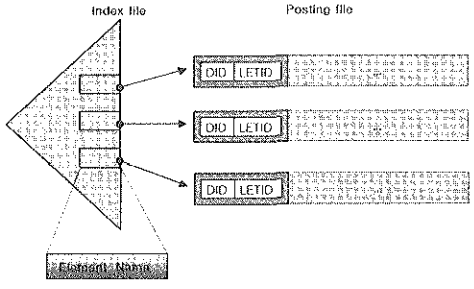
내용 색인은 내용 검색을 수행하는 색인으로 들어온 Keyword로 인덱스 파일을 구성하고 포스팅 파일에는 문서 인스턴스와 매핑할 수 있는 DID와 LETID를 둔다.



(그림 5) 내용 색인 구조

4.2 구조 색인

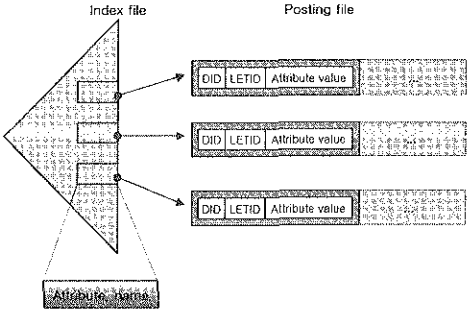
구조 색인은 엘리먼트가 중심이 되며 구조 색인의 포스팅 파일은 개개의 엘리먼트와 매핑할 수 있는 정보인 DID, LETID로 구성된다. 또한 엘리먼트 간의 계층관계 검색, 형제관계 검색을 지원한다.



(그림 6) 구조 색인 구조

4.3 에트리뷰트 색인

애트리뷰트 색인는 애트리뷰트 이름으로 인덱스가 구성되고, 해당 애트리뷰트가 포함되어 있는 엘리먼트와 매핑할 수 있도록 DID, LETID, Attribute Value로 구성된 포스팅 파일이 존재한다.



(그림 7) 에트리뷰트 색인 구조

4.4 질의 처리

XML 문서에 대한 검색은 내용 검색, 구조 검색, 애트리뷰트 검색, 혼합 검색으로 나눌 수가 있다. 내용 검색은 해당 Keyword를 갖는 문서의 DID list를 가져온 다음, DID를 이용하여 저장관리기로부터 문서의 내용을 가져와서 사용자 인터페이스로 넘겨준다. 구조 검색은 기준 엘리먼트를 식별할 수 있는 DID와 LETID를 구하고, 기준 엘리먼트로부터 관계에 의한 연산과 이를 통한 목적 엘리먼트를 찾아 엘리먼트의 내용을 전달받아 넘겨준다. 에트리뷰트 검색은 애트리뷰트 이름과 값을 이용하여 에트리뷰트 인덱스를 통해 해당 애트리뷰트가 정의되어 있는 엘리먼트를 식별할 수 있는 DID, LEID, Attirute Value를 구한다. 한 예로 subsection의 2단계 상위의 부모노드를 검색 시 subsection의 LETID값은 "03310431"로 level이 4인 것을 알 수 있다. 검색 시 앞 4바이트 "0331"을 뒤 바이트로 갖는 노드들을 검색하면 subject을 찾을 수 있다. 찾아진 subject의 LETID값은 "02210331"인데 다시 한번 "0221"을 뒤 바이트로 갖는 노드를 찾으면 "01000221"을 가지고 있는 section 찾는데, section이 상위 2단계 부모임을 알 수 있다.

6. 결론 및 향후 연구

본 논문에서는 XML 문서의 구조정보를 표현하는 방법을 제안하고 효율적인 검색을 지원하기 위해 색인 구조를 설계하였다. 구조정보표현 방법은 각 엘리먼트들에 대해 LETID를 부여하여 고유값을 주었다. LETID는 고정된 크기로 구성이 되며 각 바이트에 부모의 계층정보와 형제 노드의 순서정보를 알 수 있고 마지막 바이트로 동일한 형제 노드의 순서정보를 알 수 있다. 제안한 색인 구조는 내용 검색을 지원하는 내용 색인, 구조 검색을 지원하는 구조 색인, 애트리뷰트 검색을 지원하는 에트리뷰트 색인으로 구성되며 혼합 검색은 이들 색인을 결합하여 검색한다. 이와 같은 구조정보 표현과 색인을 이용하여 특정 엘리먼트에 직접적인 접근이 가능하고, 구조화 된 문서를 효율적으로 관리할 수 있으며, 다양한 질의처리가 가능하다. 따라서 보다 효율적이고 빠른 검색을 지원할 수 있다. 향후 연구과제로는 본 논문에서 제안한 구조정보 표현 방법을 이용한 색인 구조와 기존 구조 검색 방법들과의 성능평가가 필요하고 문서의 갱신 시 색인모델을 적용하기 위한 연구가 필요하다.

참고 문헌

[1] 박종관, " XML 문서에 대한 효율적인 구조 기반 검색을 위한 색인 모델", 충북대학교 석사학위논문, 2001

[2] Brian Lowe, Justin Zobel, Ron Sacks-Davis "A Formal Model for Databases of Structured Text", Proceedings of the Fourth International Conference on Database Systems for Advanced Applications (DASFAA '95), pp449-456,1995

[3] Chow, J.H., Cheng, J., Chang, D., Xu, J., "Index Design for Structured Documents Based on Abstraction", Proceedings of the 6th International Conference on Database Systems for Advanced Applications, pp, 89-96, 1999

[4] Lee, Y.K., Yoo, S.J., Yoon, K.R and Berra, P.B., "Index Structures for Structured Documents", Proc. Digital Library 96, pp. 91-99, 1996

[5] Toung Dao " An Indexing Model for Structured Documents to Support Queries on Content, Structure and Attributes", Proceedings of ADL'98, pp,88-97, 1998

[6] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, ExtensibleMarkupLanguage(XML)1.0,REC-xml-19980210

[7] V. Christophides. et al, "From Structured Documents to Novel Query Facilities," ACM SIGMOD, pp. 313-324, Minesota, USA, 1994