

# XML 문서를 위한 효율적인 색인 기법

강형일<sup>°</sup>, 송석일<sup>°°</sup>

<sup>°</sup> 주성대학 멀티미디어정보통신공학과

<sup>°°</sup> 충주대학교 컴퓨터공학과

## An Efficient Index Method for XML Documents

Hyung Il Kang <sup>°</sup>, Seok Il Song <sup>°°</sup>

<sup>°</sup> Juseong College, Dept. of Multimedia Information Communication

<sup>°°</sup> Chungju National University, Dept. of Computer Engineering

### 요 약

이 논문에서는 XML 문서에 대한 경로질의를 효과적으로 처리할 수 있는 색인기법을 제안한다. 제안하는 색인 기법은 문서의 변경에 동적으로 대처하며 단순경로 질의뿐 아니라, 경로의 길이가 주어지지 않는 질의에도 효과적으로 동작한다. 이 논문에서는 제안하는 색인기법을 구현하고 실험을 통해서 경로질의를 처리하는 시간을 측정하여 제안하는 방법의 타당성을 보인다.

### 1. 서론

인터넷이 발전하면서 이기종 시스템간의 자료교환을 표준화 하기 위한 한 방법으로 XML이 제안되었다. 1996년 출현한 XML은 SGML 과 HTML의 장점을 적절히 수용하여 문서의 논리적인 구조를 표현하면서도 사용이 쉽도록 하였다. XML은 발전을 거듭하여 W3C에서 권고안으로 채택하기에 이른다. 현재 XML은 인터넷 표준 문서로 사용될 뿐 아니라 전자출판, 의학, 경영, 법률, 전자도서관, 전자상거래 등 매우 광범위한 분야에서 사용되고 있다.

이처럼 XML의 사용처가 증가하고 문서의 양이 커지면서 XML 문서를 효과적으로 저장하고 검색하기 위한 XML 데이터베이스 기술에 대한 연구가 수

행되었다. XML 문서에서는 기본 단위인 엘리먼트를 계층적으로 배열하여 트리 형태로 구조화한다. 따라서, XML 문서에 대한 검색은 트리의 특정 경로를 포함할 수 있으며 이들 경로에 대한 처리가 검색의 효율을 높이는데 중요한 역할을 한다.

경로질의를 처리하기 위한 여러 방법들이 제안되었다. 지금까지의 조사에 따르면 제안된 방법들은 모두 나름대로의 단점을 가지고 있다. 이 단점들을 정리해 보면, 먼저 문서의 구조가 변경될 때 경로질의 위한 색인구조가 대대적으로 재구성이 된다. 두번째, 경로의 길이가 2 이상이거나 (예, a/\_/b) 주어지지 않는 경우(예, a/b)의 질의처리가 어렵다. 마지막은, 조상과 자손간의 상하 관계가 아닌 형제관계를 포함하는 경로질의의 처리가 어렵다는 점이다.

이 논문에서는 위에서 지적한 문제들을 효과적으로 해결하는 방법을 제안한다. 또한, 제안하는 방법을 구현하고 간단한 실험을 통해 제안하는 방법의

이 논문은 2002년도 한국학술진흥재단의 지원에 의하여 연구되었음(KRF-2002-003-D00281)

타당성을 보인다. 이 논문의 구성은 다음과 같다. 2장에서는 기존에 제안된 방법들을 체계적으로 정리하고 분석한다. 3장에서는 제안하는 방법을 설명한 후 4장에서 실험을 통해 제안하는 방법의 타당성을 보인다. 그리고, 5장에서 결론을 맺는다.

## 2. 관련연구

기존에 제안된 방법들은 다음과 같은 범주로 나누어 볼 수 있다. 첫번째, 문서의 트리 구조를 중복 없이 요약하여 색인구조로 사용하는 기법이다. 이 유형의 색인구조로 대표적인 것이 DataGuide[1]이다. 이 방법은 단순 경로질의 처리는 효과적이지만 '/a/b', 'a/b', 'a/\_/\_/b'와 같이 경로가 불분명한 경우에는 많은 경로를 순회해야 하므로 비효율적이다.

다음으로는 문서의 각 엘리먼트에 상하관계를 파악할 수 있는 고유한 번호를 할당하고 부여된 번호를 기반으로 엘리먼트간에 조인을 수행해서 질의를 처리하는 방법이 있다. 여기에 가장 대표적인 것으로 XISS[2]를 들 수 있다. 각 엘리먼트에는 문서에서 해당 엘리먼트가 차지하는 범위를 수치화한 번호를 부여한다. 범위 값은 엘리먼트간의 상하관계를 파악할 수 있는 수단이 된다. 번호가 부여된 엘리먼트들을 B-트리와 같은 색인구조에 색인하고 질의 경로의 엘리먼트를 색인구조에서 찾아서 이들간에 조인을 하여 질의를 처리한다.

이 방법은 상대적으로 "a/b", "a/\_/b", "/a/b"와 같은 질의 처리에 효과적이지만 단순경로 질의 처리에 있어서는 조인을 여러 번 해야 하는 이유로 DataGuide에 비해서 비효율적이다. 또한, 엘리먼트가 문서에 추가되거나 삭제되면 각 엘리먼트에 부여한 범위값이 변경이 되게 된다. 이 범위 값이 변경되면 색인구조에 저장된 엘리먼트들중에서 해당되는 엘리먼트의 범위값을 모두 변경해야 하는 대단위 작업이 발생하게 된다. 이를 해결하기 위해서 이 방법에서는 범위값을 미래에 입력될 엘리먼트에 대비하여 넉넉한 값을 부여하는 방법을 이용하고 있다. 하지만 이 방법은 변경이 일어나는 시기를 늦추는 효과는 있지만 대량의 변경 작업을 필요로 하게 된다.

마지막으로 가장 최근에 제안된 방법 중 하나가 VIST[3]이다. 이 방법에서는 문서트리의 각 엘리먼트의 루트부터 엘리먼트까지의 경로를 인코딩하고 인코딩 된 경로를 색인구조에 저장하는 방식이다. 이 방법의 단점이라면, '/a/b'와 같은 질의를 처리하는 비용이 상대적으로 많이 든다는 것이다.

이 논문에서는 이상 언급한 색인구조가 않고 있는 문제점을 해소할 수 있는 색인기법을 제안한다. 제안하는 방법은 요약 트리 기법과 조인방법을 적절히 혼합하는 접근 방식을 사용한다. 이에 대한 자세한 설명은 3장에서 계속 하기로 한다.

## 3. 제안하는 색인기법

앞에서 간략히 언급한 대로 제안하는 방법은 요약 트리 기법과 조인방법을 혼용한다. 요약 트리 기법에 대해서 조금더 자세히 알아보도록 하자. 그림 1은 XML문서와 이에 대한 요약 트리를 보여주고 있다. XML 문서에는 총 6개의 엘리먼트가 있으며 이 엘리먼트에 고유한 번호를 붙인다. 문서 트리는 엘리먼트 이름대신에 이 엘리먼트 번호를 가지고 표현했다. 각 노드는 두개의 숫자 쌍으로 이루어져 있는데 앞의 숫자는 엘리먼트 번호이며 뒤의 숫자는 문서트리의 각 노드에 부여한 노드 번호이다. 노드 번호는 어떤 순서에 의해서 부여되는 것이 아니며 문서의 각 노드를 유일하게 구분할 수 있으면 된다.

이 문서 트리를 요약해 놓은 것이 바로 오른쪽의 요약 트리이다. 이 요약트리는 문서트리의 각 노드들 중 루트 노드로부터 현재 노드까지의 경로가 같은 노드들을 하나의 노드로 표현한다. 이 표현방법은 DataGuide에서와 같다. 각 노드역시 엘리먼트 이름대신에 엘리먼트 번호를 가지고 있고 노드 옆의 숫자는 해당 경로에 해당하는 노드들의 번호이다. Dataguide에서는 이 요약트리를 순회해서 질의를 처리했다. 예를 들어서, /personnel/person/email 이라는 질의가 있으면 이것은 다시 /1/2/4 로 표현될 수 있으며 요약 트리의 루트부터 순회하여 노드번호 5, 7을 획득할 수 있다. 노드번호를 이용해서 원하는 chief@foo.com, one@foo.com 을 검색할 수 있다.

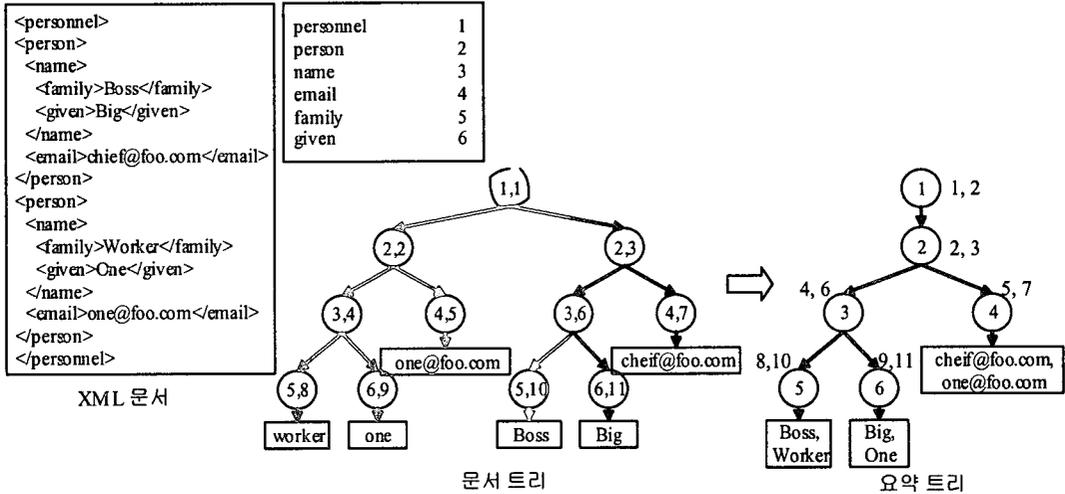


그림 0 XML 문서와 요약 트리

하지만, //family 나 person/email, person/\_/email 과 같은 질의를 처리하기 위해서는 매우 많은 양의 노드들을 순회해야 한다. 이 논문에서 제안하는 방법은 요약 트리의 노드들로부터 적절한 특징 값을 추출하고 이를 다차원 색인구조에 삽입하여 질의를 매우 빠르게 처리한다. 노드에 대한 특징값을 추출하는 것을 설명하기 위해서 몇 가지 예를 들어 보자.

‘person/email’ 이라는 질의가 있다고 하자. 이 질의의 최종목적은 person 엘리먼트를 조상으로 하는 모든 email 엘리먼트들을 찾아 내는 것이다. 즉, 이 질의를 처리하기 위해서는 email 엘리먼트의 조상중에 person 이 있다는 사실을 알고 있어야 한다. 이번엔 ‘person/\_/email’ 질의에 대해서 생각해 보자. 이 질의는 person 엘리먼트를 3대조로 하는 email 엘리먼트를 찾아야 한다. ‘person/email’과는 다르게 이 질의는 email 의 조상에 어떤 엘리먼트가 있는지와 각 엘리먼트가 email의 몇대 조상인지를 알고 있어야 할 것이다.

이 점에 착안하여 이 논문에서는 요약 트리의 각 노드마다 (EID, PID, DL)와 같은 특징을 추출해 낸다. EID란 현재 노드의 엘리먼트 번호이며 PID는 부모 엘리먼트의 엘리먼트번호, DL은 EID와 PID 간의 레벨 차이이다. 그림 1의 요약 트리의 각 노드의 특징을 추출하면 표 1과 같다.

표 1 추출된 특징

EID	특징	NID
1	(1, null, null)	1
2	(2, 1, 1)	2, 3
3	(3, 2, 1), (3, 1, 2)	4, 6
4	(4, 2, 1), (4, 1, 2)	5, 7
5	(5, 3, 1), (5, 2, 2), (5, 1, 3)	8, 10
6	(6, 3, 1), (6, 2, 2), (6, 1, 3)	9, 11

EID는 엘리먼트 번호를 이야기 하고 NID는 EID 가 문서 트리에서 나타날 때의 노드 번호를 의미한다. ‘person/email’ 질의를 위의 테이블을 이용해서 처리해 보자. person의 EID는 2이고 email의 EID는 4이다. 질의는 2//4로 다시 표현될 수 있다. 엘리먼트 2를 조상으로 갖고 2와 4 사이의 레벨차이는 관계가 없는 엘리먼트 4를 찾는 것이다. 이 질의를 (4, 2, \*) 로 표현해 보자. 여기서 \*는 DL은 어떤 값이든지 관계 없다는 뜻이다. 테이블에 있는 특징들 중에서 이를 만족하는 것들을 찾아보면 4번 엘리먼트의 (4, 2, 1)이 답이 될 것이다. 최종적으로는 문서상에서 이를 만족하는 노드를 찾아야 하는데 이 노드들은 5와 7이 될 것이다. ‘person/\_/email’ 은 (4, 2, 3) 로 표현될 수 있으며 이를 만족하는 특징은 없다.

두개 이상의 엘리먼트로 이루어진 질의의 경우

에는 보다 복잡하다. ‘personnel/person/email’ 이라는 질의가 있다면 이 질의는 ‘personnel/\_/email’ 와 ‘person/email’ 로 나누어질 수 있으며 두 질의의 교집합이 최종적인 결과가 될 것이다.

추출된 특징들은 차원이 3개 이다. 이렇게 다차원의 특징들을 갖는 데이터를 위한 색인구조로는 R-트리[4]를 비롯해 다수가 제안되었다. 이 논문에서 추출한 특징들을 이 다차원 색인구조에 색인하면 질의와 일치하는 특징들을 찾는 시간은 대폭 감소할 것이다. 이 논문에서는 R-트리를 이용하였다.

#### 4. 구현 및 고찰

이 논문에서는 제안하는 방법의 타당성을 보이기 위해 제안하는 방법을 구현하고 다양한 형태의 질의 처리 시간을 측정하였다. 2.4 GHz CPU, 512 Mbytes 메모리, 80Gbytes의 하드디스크에 리눅스를 운영체제로 하는 플랫폼에서 구현하였으며 사용한 컴파일러는 gcc 3.2.3 이다. 실험에 사용된 데이터는 XMARK[5]에서 제공하는 생성기로 생성한 문서를 이용하였다. 구현한 검색 시스템은 문서로부터 특징을 추출하는 특징 추출기와 R-트리 관리기로 구성된다. 표 2와 같이 3개 유형의 질의를 실험에 사용했다. 이 표는 3개의 질의와 각각을 EID로 변환하고 R-트리에 적용 가능하도록 변환한 질의를 보여주고 있다.

표 2 실험에 사용된 질의

질의	R-트리를 위한 질의
africa/_/quantity	(26, 18, 2)
autralia//listitem	(16, 20, *)
site/regions/africa/item	(24, 1, 3), (24, 2, 2), (24, 18, 1)

XMARK에서 제공하는 auction.dtd 의 총 엘리먼트 개수는 77개 였으며 순환을 제거한 요약 트리의 높이는 10 이었다. 요약 트리에서 추출한 특징의 개수는 총 2339개 였다. 2339개의 특징을 R-트리에 삽입하고 표 2의 질의를 수행시키면서 각각의 시간을 측정하였다. 측정한 시간에는 실제 문서에 있는 엘리먼트들을 디스크로부터 읽어오는 시간은 제외하였

다. 표 3은 각 질의의 수행시간을 보여주고 있다.

표 3 질의 수행시간

질의	시간(μsec)
africa/_/quantity	7
autralia//listitem	8
site/regions/africa/item	22

모든 질의의 처리시간이 모두 1ms 이하로 매우 빠른 것을 볼 수 있다. 마지막 질의의 경우 다른 질의에 비해서 시간이 3배 이상 소요되는 것을 볼 수 있다. 이유는 마지막 질의는 총 3회의 R-트리 검색이 이뤄져야 결과 값을 얻을 수 있기 때문이다.

#### 5. 결론

이 논문에서는 XML 문서에 대한 경로질의를 효과적으로 처리할 수 있는 방법을 제안하였다. 제안한 방법은 문서의 요약 트리를 만들고 요약 트리 각 노드의 특징을 추출하여 다차원 색인구조에 색인하는 방식이다. 실험 결과를 보면 제안하는 방식은 다양한 형태의 질의를 1ms 이내에 처리하는 성능을 보였다. 제안하는 방법의 전체적인 성능은 뛰어나지만 경로가 길어질수록 처리속도가 느려지는 단점이 있다. 향후 연구에서는 이를 보완하도록 한다.

#### [참고문헌]

- [1] R. Goldman, J. Widom, "DataGuides: Enabling Query Formulation and Optimization in Semistructured Databases," VLDB, 1997, pp. 436-445
- [2] P. J. Harding, Q. Li, B. Moon, "XISSL/R: XML Indexing and Storage System using RDBMS," VLDB, 2003, pp. 1073-1076
- [3] H. Wang, S. Park, W. Fan, P. S. Yu, "ViST: A Dynamic Index Method for Querying XML Data by Tree Structures," SIGMOD, 2003, pp. 110-121
- [4] A. Guttman, "R-Trees: A Dynamic Index Structure for Spatial Searching," SIGMOD, 1984, pp. 47-57
- [5] <http://monetdb.cwi.nl/xml/index.html>