韓國 OA學會 論文誌
第6卷 第4號, 2001. 12.

2001-6-4-3-3

스키마 기반의 XML 문서 관리 시스템 설계

조 윤 기*, 김 영 란**

Design of XML Document Management System based on Schema

Yoon-Ki Cho*, Young-Ran Kim**

요 약

정보화 사회로의 진행이 가속화됨에 따라 정보 양이 급격히 증가하면서 XML을 이용하여 정보를 효율적으로 저장하고 검색하기 위한 많은 연구들이 진행되고 있다. 그러나 기존의 방법은 특정 엘리먼트의 부모, 자식, 형제에 대한 다양한 구조 검색을 효율적으로 지원하지 못한다. 따라서 이 논문에서는 XML 문서의 효율적인 관리와 구조검색을 위해 OETID를 이용한 구조 정보 표현과 색인 기법을 제안한다. 또한 XML 문서의 구조 정보를 저장하기 위한 저장 기법과 검색 결과에 대한 문서 통합 과정을 제안한다. 제안한 방법을 이용하여 XML 문서의 구조 정보를 효율적으로 표현할 수 있을 뿐 아니라 간단한 연산으로 특정 엘리먼트에 직접적인 접근과, 다양한 질의 처리가 가능하다. 따라서 보다 효율적이고 빠른 검색을 지원할 수 있다.

Abstract

As progressing rapidly to the information society and increasing greatly the amount of information, many researchers have been made utilizing XML to store and retrieval the information effectively. But, many other existing method could not support various structured retrieval method for specific parent, children and sibling nodes. In this paper, we propose (1)an effective method of representation for structured information and of indexing mechanism using OETID(Ordered Element Type ID) for effective management and structured retrieval of the XML documents. Also it contains another proposal that is (2) a documents integration mechanism for retrieval result and storing technique to store structural information of the XML documents. With our methods, we could effectively represent structural information of XML documents, and could directly access the specific elements and process various queries by simple operations.

* 충북대학교 전자계산학과 박사과정수료
** 충청대학 컴퓨터학부 부교수

Ⅰ. 서론

정보화 사회로의 진행이 가속화됨에 따라 컴퓨터를 이용한 문서 처리와 관리의 중요성이 강조되고, 인터넷 상의 정보를 효율적으로 사용하기 위한 연구들이 활발하게 진행되고 있다. 이를 위해 인터넷 상의 정보를 구조화된 문서로 표현함으로써 보다 효율적으로 검색 관리하기 위한 표준이 요구 되었다. 구조화된 문서를 표현하기 위한 표준으로 1996년 W3C(World Wide Web Consortium)에서 제안한 XML(eXtensible Mark- up Language)은 고정된 태그만을 사용하여 정보를 표현하는 HTML (HyperText Markup Language)의 한계를 극복하고, SGML(Standard Generalized Markup Language)의 복잡함을 단순화함으로써 현재 웹 문서 뿐만 아니라 전자상거래, 전자도서관 및 EDI(Elect- ronic Data Interchange)를 포함한 다양한 분야에서 사용되고 있다[1][2][3].

현재 대부분의 XML 저장 관리 시스템은 DTD로 작성된 XML 문서를 대상으로 검색 및 저장 관리 기능을 수행한다. 그러나 2001년 5월 W3C에서 DTD의 단점을 보완하기 위해 XML 스키마의 권고안이 발표됨에 따라 DTD기반의 문서뿐만 아니라 XML 스키마 기반으로 작성된 XML 문서의 처리가 필요하다. 스키마를 기반으로 작성된 XML 문서는 기존의 문서와 달리 하나의 문서에 내용 정보와 구조 정보를 함께 가지고 있다. 따라서, 기존의 문서에서 제공하던 키워드를 기반으로 한 내용 정보에 대한 검색뿐만 아니라 문서의 논리적인 구조 정보에 대한 검색 기능이 요구된다. 이를 위해 XML 문서의 구조 정보를 표현하기 위한 여러 가지 방법들이 제안되고 있다. 그러나 일부 방법들은 조상, 자손, 형제 관계의 엘리먼트에 접근하기 위해 복잡한 연산을 수행하거나 형제 관계 노드를 알 수 없다[1][4]. 또한 XML 문서 관리 시스템은 질의를 통해 문서의 일부 혹은 전체를 검색 결과로 하는 문제점을 갖고 있다. 이와 같은 문제점을 해결하기 위해서는 스키마 구조에 따라 분할 기법으로 저장된 문서의 요소들을 통합하기 위한 과정이 필요하다.

이 논문에서는 XML 문서의 구조 정보를 효율적으로 검색하기 위한 XML 문서의 구조 정보 표현 방법과 색인 구조 및 XML 저장 관리 시스템의 검색 요소 통합 방법을 제안하고자 한다.

Ⅱ. 관련연구

XML로 표현된 구조화된 문서를 검색하기 위해서는 키워드에 의한 문서 단위의 내용 검색뿐만 아니라 엘리먼트를 기본 단위로 하는 구조 검색 및 애트리뷰트 검색이 지원되어야 한다[5][8]. 이를 위해 구조 정보를 효율적으로 표현하기 위한 연구가 선행되어야 한다. 구조 정보를 표현하기 위한 기존의 모델로는 Subtree 모델, SCL 모델, K-ary 완전 트리 모델, 그리고 ETID 모델 등이 있다.

Subtree 모델은 검색 효율을 향상시키기 위한 5가지 질의 명세를 제시하였다. 질의 분류에 대한 검색을 지원하기 위한 인덱스 구조로서 XML 문서를 subtree 형태로 표현한 후 여기에 나타나는 모든 단위에 대해서 중복 색인을 하고 색인어가 위치하는 단위를 기록한다. 반면에, 이 방법은 추출된 색인어가 나타난 계층의 모든 상위 계층에 대해서도 색인을 하므로 공간상의 중복이 일어난다는 단점이 있다.

SCL 모델은 텍스트와 마크업에 대해 색인 넘버를 부여한 후, 불용어를 제외한 텍스트 어휘들을 텍스트 인덱스에 색인 넘버로 저장하고, 마크업은 시작 태그와 종료 태그의 쌍으로 마크업 인덱스에 저장한다. 따라서, 구조 문서의 계층적 관계보다는 포함 관계를 이용한 표현 방법으로서 SC-list(Simple Concordance list)라는 데이터 타입을 통해 중첩된 정보를 허용하므로 리스트에 대한 리스트와 같은 순환 구조를 다룰 수 있다는 장점이 있다. 그러나 SCL 구조는 트리의 깊이를 표현할 수 없으므로 조상이나 형제 엘리먼트를 검색할 수 없다는 단점이 있다[4][7].

K-ary 완전 트리 모델은 문서에 대한 트리로부터 이들 노드 중 가장 큰 차수 K를 구하여 K-ary 완전 트리로 재구성한 후, 문서 트리를 매핑하여 각 노드에 노든 번호를 부여한다. 이 모델은 문서 구조 사이의 계층 관계

를 간단한 공식을 통해 쉽게 구할 수 있다는 장점이 있는 반면에, 매핑 과정에서 Null 노드가 많아질 수 있고 노드의 깊이가 깊어질수록 노드 변화가 커진다는 단점이 있다[6].

ETID 모델은 엘리먼트들 간의 계층 정보와 동일 부모 엘리먼트를 갖는 자식 엘리먼트들의 순서 정보, 그리고 동일한 부모 엘리먼트를 갖는 자식들 중 동일한 타입의 엘리먼트들에 대한 순서 정보를 통해 구조 문서를 표현한다. 이 방법은 기준 엘리먼트로부터 특정 엘리먼트에 대한 계층 정보와 순서 정보를 간단한 문자열 조작만으로 쉽게 구할 수 있다는 장점이 있는데 반해 트리의 깊이가 깊어질수록 각 노드를 표현하기 위한 공간이 무한대로 늘어난다는 단점이 있다[9][10].

Ⅲ. 다양한 구조검색을 지원하기 위한 색인 모델

XML 문서에 대한 다양한 구조 검색을 처리하기 위해 먼저, XML 문서의 계층 구조를 효율적으로 표현할 수 있도록 OETID(Ordered Element Type IDentifier)를 이용한 구조 정보 표현 방법과 이를 기반으로 설계한 색인 생성기를 통한 색인 모델을 제안한다.

3.1 구조 정보 표현

3.1.1 OETID

XML 문서는 내용 정보 뿐만 아니라 구조 정보를 함께 표현한다. 따라서 XML 문서의 부모 자식 관계뿐만 아니라 해당 엘리먼트에 대한 형제 관계, 그리고 동일한 타입의 엘리먼트에 대한 순서 정보를 표현하기 위해 (표현1)과 같이 OETID를 이용함으로써 사용자는 간단한 연산으로 질의에 대한 결과를 검색할 수 있다.

X1X2X3X4 ..... Xn-1Xn (표현1)

X1: 첫 번째 계층에 있는 엘리먼트의 SORD
X2: 첫 번째 계층에 있는 엘리먼트의 SSORD
X3: 두 번째 계층에 있는 엘리먼트의 SORD
X4: 두 번째 계층에 있는 엘리먼트의 SSORD
...
Xn-1: n-1 번째 계층에 있는 엘리먼트의 SORD
Xn: n-1 번째 계층에 있는 엘리먼트의 SSORD

(표현 1)에서 보는 바와 같이 해당 계층의 엘리먼트는 두 바이트(X1X2, X3X4, ... Xn-1Xn)로 표현되며 각 바이트는 SORD(Sibling ORDer)와 SSORD(Same SORD)로 표현된다. 이때 각 바이트는 ASCII 코드의 순서를 따르는 '0'→'9'→'A'→'Z'→'a'→'z'순으로 된 62개의 문자를 사용한다. 이때, X1X2을 부모 엘리먼트로 갖는 자식 엘리먼트 X1X2X3X4을 나타낸 것이다. 즉, 하위 계층에 있는 엘리먼트는 부모 엘리먼트에 대한 정보에 자신의 위치 정보를 추가하여 표현된다. 이때 X1X2, X1X2X3X4를 엘리먼트의 구조 정보를 식별하기 위한 식별자 OETID라 한다. 또한, XML 문서에서 반복적으로 사용되는 엘리먼트에 대해서 SORD와 SSORD로 표현함으로써 반복 구조를 해결할 수 있다. 따라서 간단한 문자열 조작으로 기준 엘리먼트로부터의 부모 엘리먼트나 자식 엘리먼트에 대한 정보를 쉽게 구할 수 있다.

그림 1은 논문을 주제로 한 스키마를 기반으로 작성한 간단한 XML 문서이고, 제안한 구조 정보 표현 방법에 따라 트리 구조로 표현하면 그림 2와 같다.

```
<?xml version="1.0" encoding="euc-kr">
<!DOCTYPE paper SYSTEM "paper.xsd">
<paper>
 <head>
  <title>
   <kor title>레지스트리를 이용한 다양한 스키마
       기반의 XML 문서 관리 시스템</kor title>
   <eng title>XML Repository System based on
       Various Schema using Registry</eng titile>
  </title>
  <author>
    <name>홍길동</name>
    <department>전자계산학과</department>
  </author>
  <abstract>이 논문은 레지스트리를 이용하여 다양한
    스키마를 기반으로 작성된 ...</abstract>
 </head>
 <body>
   <chapter>1.서론
     <section>
      <para>웹의 발전으로 인터넷상의 정보의
        양이 급증하면서 ...</para>
    </section>
  </chapter>
  <chapter>2. 본론
    <section>2.1 구조 정보 표현
          <para>다양한 검색을 위해 ...</para>
```

```
    </section>
    <section>2.2 색인 모델
        <para>내용 색인</para>
        <para>구조 색인</para>
        <para>애트리뷰트 색인</para>
    </section>
  </chapter>
  <chapter>3. 결론</chapter>
</body>
  <ref>참고 문헌</ref>
</paper>
```

그림 1. 스키마를 기반으로 한 XML 문서

3.1.2 계층 정보 및 순서 정보 표현

엘리먼트들 간의 계층 정보에 대한 표현은 부모 엘리먼트의 정보를 유지함으로써 가능하다. 예를 들어, 그림 2에서 루트 엘리먼트의 OETID를 "11"이라 할 때, <body>의 OETID는 부모 엘리먼트의 OETID에 <body>의 정보를 추가해 "1121"로 표현할 수 있다. 따라서 <body>의 부모 엘리먼트는 현재 계층에서 추가된 뒤의 두 자리를 제거한 OETID 값 "11"을 갖는 <paper>가 됨을 알 수 있다. 또한 <body>의 자식 엘리먼트는 OETID 값으로 "1121"을 앞 네자리로 갖는 엘리먼트인 <chapter>임을 알 수 있다.

앞에서 설명한 <body>의 OETID 중 뒤의 두 자리 "21"은 부모 엘리먼트에 대한 자식 엘리먼트의 순서 정보(SORD)와 동일한 부모의 자식 엘리먼트 중에서 동일한 이름을 갖는 엘리먼트의 순서 정보(SSORD)의 결합으로 표현된다.

즉 <body>는 <paper>의 두 번째 자식이며, <body>란 이름의 엘리먼트는 하나만 존재하므로 "21"로 표현된다. 또한 <body>의 자식 엘리먼트인 <chapter>의 경우 동일한 이름의 3개 엘리먼트가 존재하므로 SORD와 SSORD는 순서에 따라 11,22,33이 부여된다. 따라서 <chapter>의 OETID는 문서에 나타난 순서에 따라 "112111", "112122", "112133"으로 표현된다. 따라서 동일한 이름의 엘리먼트가 반복적으로 사용될 경우 SSORD를 이용하여 반복 구조의 표현이 가능하다.

3.1.3 구조 정보 추출 과정

XML 문서로부터 구조 정보를 추출하여 각 엘리먼트에 대한 구조 정보 결과물을 생성한다. 결과물을 생성하기 위해 다음과 같은 과정을 수행한다.

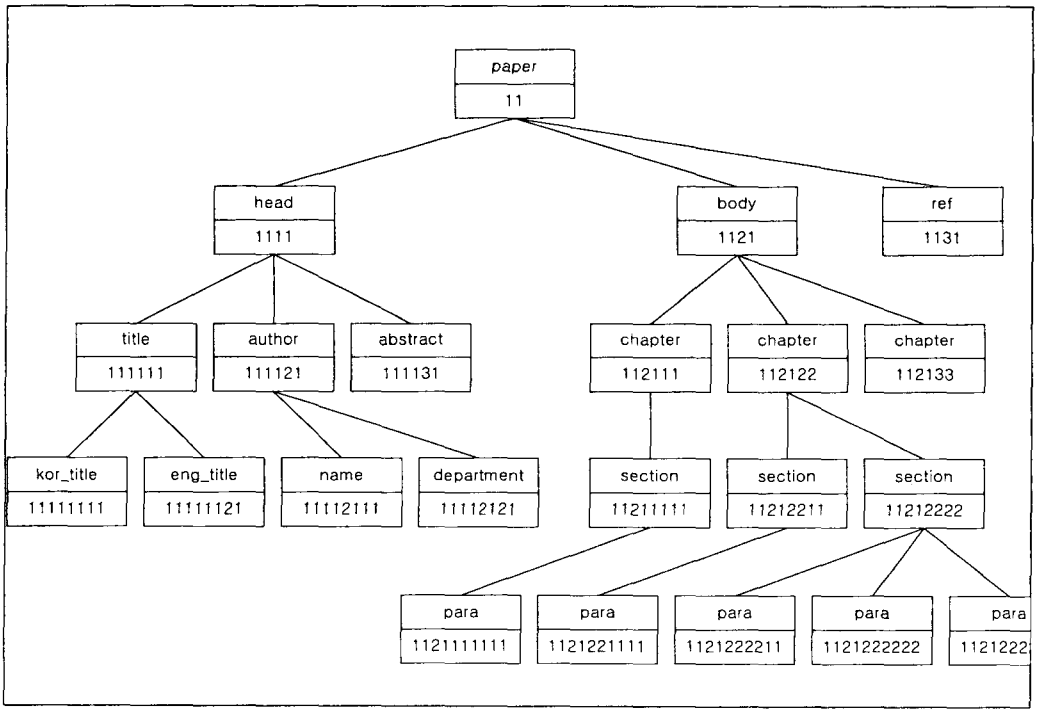


그림 2. XML 문서에 대한 구조 정보 표현의 예

전달받은 문서를 파싱하면서 시작 태그를 발견하면 이를 스택에 저장하고 구조 정보 결과물에 추가한 후, SORD_cnt[n]과 SSORD_cnt[n]을 1씩 증가한다. 이때 SORD_cnt[n]과 SSORD_cnt[n]은 각각 SORD와 SSORD에 대한 순서 정보를 유지하기 위한 변수이고, n은 현재 스캔한 엘리먼트가 속해있는 계층(level)을 의미한다. 이렇게 구한 값은 현재 엘리먼트에 대한 순서값으로 이 두 값을 결합한 뒤 부모 엘리먼트의 OETID 값에 추가하면 현재 엘리먼트에 대한 고유의 식별값 OETID를 구할 수 있다. 또한 각 엘리먼트의 데이터 타입은 엘리먼트 데이터 타입 매핑 테이블을 참조하여 구한다. content에는 현재 파싱한 시작 태그에 대한 종료 태그를 만날 때까지의 내용을 저장한다. 만일 자식 엘리먼트가 있을 경우 현재 태그의 종료 태그를 추가한다. 또한 루트 태그일 경우엔 선언구문과 처리구문을 추가한다. SORD_cnt[n] 값은 부모 엘리먼트의 종류 태그를 만나기 전에 동일한 부모에 대한 자식 엘리먼트가 발견될 때마다 1씩 증가하며, SSORD_cnt[n] 값은 동일한 부모에 대한 자식 엘리먼트 중에 동일한 이름을 갖는 엘리먼트를 발견할 때마다 1씩 증가한다. 현재 엘리먼트에 대한 종료 태그를 만나면 스택으로부터 시작 태그를 제거(pop)하고 다음에 읽은 값이 부모 엘리먼트에 대한 종료 태그일 경우 현재 계층 n에 속한 SORD_cnt[n]와 SSORD_cnt[n]의 값을 0으로 초기화 한 뒤 계층 n의 값을 1씩 감소한다. 만일 시작 태그에 애트리뷰트가 존재할 경우 att_list에 애트리뷰트 이름과 값을 추가한다. 이때 문서의 파싱 과정은 깊이 우선 탐색 방법으로 처리한다.

XML 문서로부터 구조 정보 결과물을 추출하는 과정은 그림 3과 같다.

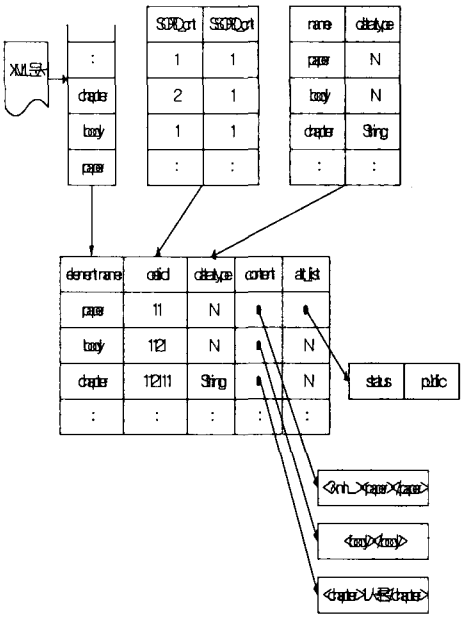


그림 3. 구조 정보 추출 과정

3.2 색인 구조

XML 문서로부터 추출한 구조 정보 결과물은 실제 문서의 구조 정보를 추출하기 위한 색인 파일을 생성한다. 이때, XML 문서에 대한 다양한 검색을 지원하기 위해 기존의 역파일 방식을 이용한 내용 색인, 구조 색인, 엘리먼트 색인으로 각각에 대한 색인 구조는 그림 4와 같다.

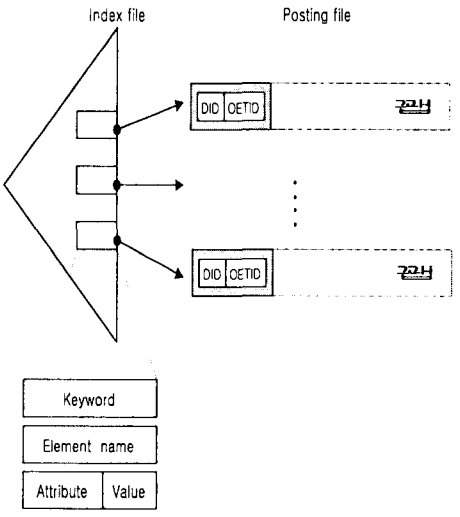


그림 4. 색인 구조

3.2.1 내용 색인

내용 색인은 기존의 키워드 검색에서와 마찬가지로 문서 내에 있는 내용을 기반으로 한 검색으로 키워드에 대한 정보를 포함한 색인 파일과 색인 파일이 참조할 포스팅 파일로 구성된다. 포스팅 파일은 키워드를 포함하고 있는 문서 정보 DID와 문서 내의 엘리먼트 정보 OETID로 구성된다. 따라서 내용 색인을 통해 문서 단위의 검색뿐 아니라 엘리먼트 단위의 검색이 가능하다.

3.2.2 엘리먼트 색인

엘리먼트 색인은 엘리먼트 기반의 구조 검색을 지원하기 위한 것으로 색인 파일은 엘리먼트 정보로 구성되며, 포스팅 파일은 내용 색인에서와 마찬가지로 DID와 OETID로 구성된다.

3.2.3 애트리뷰트 색인

애트리뷰트 색인은 엘리먼트에 포함된 속성과 속성에 대한 값을 기반으로 한 검색으로 색인 파일은 애트리뷰트 이름과 값으로 구성되며 이와 관련된 DID와 OETID로 포스팅 파일을 구성한다.

Ⅳ. XML 문서 관리 시스템 설계 및 질의 처리

4.1 시스템 구성도

XML 문서 관리 시스템은 그림 5에서 보는 바와 같이 크게 질의 처리 및 결과 생성과 저장 및 추출로 나눌 수 있다. 질의 처리 및 결과 생성 부분에서는 사용자 인터페이스로부터 질의를 입력받아 검색을 수행하고 그 결과를 다시 사용자 인터페이스로 돌려주는 것으로 질의 처리기와 질의 결과 생성기로 구성된다. 질의 처리기는 사용자 인터페이스로부터 입력받은 자료를 이용하여 질의를 형성하고 구조 검색의 종류에 따라 질의를 분류하여 해당 색인 파일의 검색을 수행한다. 질의 결과 생성기는 해당 색인 파일을 통해 검색된 정보를 이용하여 저장 관리기로부터 구조 정보 결과 파일에 있는 자료를 추출하여 사용자가 볼 수 있도록 문서를 통합한다.

저장 및 추출부는 기본 스키마를 기반으로 작성된 문서 또는 구조 정보 변환기를 통해 변환된 문서로부터 구조 정보를 추출하고 추출된 정보를 저장 관리하는 부분으로 구조 정보 추출기, 스키마 분석기, 저장 관리기, 그리고 색인 관리기로 구성된다.

구조 정보 추출기는 구조 정보의 저장, 검색, 그리고 검색 결과의 통합을 위해 XML 문서로부터 엘리먼트, OETID, content, 그리고 애트리뷰트 정보를 추출하여 구조 정보 결과 파일에 저장한다. 스키마 분석기는 스키마로부터 엘리먼트의 데이터 타입을 추출하여 엘리먼트 데이터 타입 매핑 테이블에 저장한다. 저장 관리기는 생성된 정보를 데이터베이스에 저장하기 위한 DB 스키마를 생성하는 스키마 생성기, XML 문서를 저정하기 위한 XML인스턴스 저장기로 구성된다. 또한 색인 관리기는 색인 파일을 생성하여 데이터베이스에 저장하고 관리하는 일을 담당하는 것으로 내용 색인기, 엘리먼트 색인기, 애트리뷰트 색인기로 구성된다.

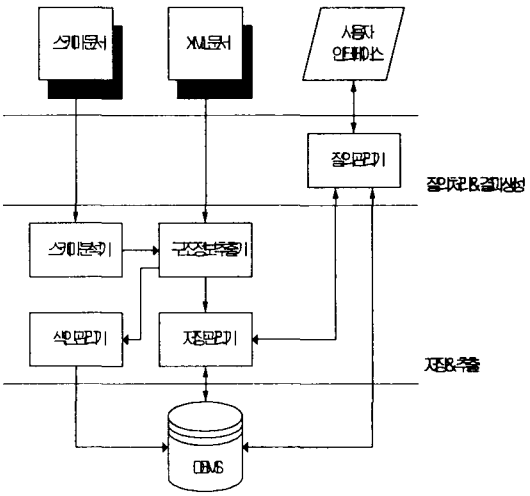


그림 5. XML 문서 관리 시스템 구성도

4.2 질의 처리와 결과 생성

질의 처리는 사용자 인터페이스로부터 전달받은 검색 요소들의 집합으로부터 해당 정보를 검색하기 위해 질의를 형성하여 이를 서버로 전달한다. 서버 측에서는 전달받은 질의를 분석하여 해당 검색을 수행한다. 이때 수행되는 검색은 내용 검색, 구조 검색, 애트리뷰트 검 그리고 혼합 검색으로 분류되며 검색의 종류에 따라 색인 테이블로부터 해당 엘리먼트의 OETID 값을 검색한 후, 구

조 정보 결과 테이블로부터 검색한 OETID 값을 포함하는 엘리먼트 정보에 대한 집합을 추출한다.

질의 결과 생성기는 요청한 질의에 대한 검색 결과를 생성하는 것이다. 검색의 결과는 문서 전체 또는 문서의 일부분이 될 수 있다. 이 논문에서는 문서 저장 방식으로 혼합 모델을 사용하므로 문서 전체를 검색 결과로 할 경우엔 문서 저장 테이블로부터 해당 문서를 검색하여 사용자 인터페이스를 통해 보여준다. 그러나 문서의 일부분을 검색 결과로 할 경우엔 추출한 엘리먼트 정보를 통합하는 과정이 필요하다. 우선 질의 처리기를 통해 추출한 엘리먼트 정보에 대한 집합으로부터 element name, oetid, content, 그리고 end_tag 정보만을 추출하여 query_result 테이블을 생성한다. 이렇게 생성된 query_result 테이블은 OETID를 기준으로 오름차순 정렬을 수행한다. 그림 6은 오름차순으로 정렬된 query_result 테이블이다.

element name	oetid	content	end_tag
paper	11	<?xml version="1.0" encoding="euc-kr"> <!DOCTYPE paper SYSTEM "paper.xsd"> <paper status="public">	0
head	1111	<head>	0
title	111111	<title>	0
kor_title	11111111	<kor_title>레지스트리를 이용한..	1
eng_title	11111121	<eng_title>XML Repository System...	1
author	111121	<author>	0
name	11112111	<name>홍길동	1
department	11112121	<department>전자계산학과	1
abstract	111131	<abstract>이 논문은 레지스트리를 ...	1
body	1121	<body>	0
:	:	:	:

그림 6. query_result 테이블

문서의 통합을 위해 정렬된 query_result 테이블로부터 테이블의 크기만큼 반복적으로 하나의 레코드를 읽어들여 다음과 같은 처리를 수행한다.

먼저, 읽어들인 레코드가 첫 번째일 경우 스택에 시작 태그를 푸시한 후, 문서에 content를 추가한다. 만일 첫 번째 레코드가 아닐 경우엔 바로 전에 읽어들인 레코드의 oetid 값의 길이와 방금 읽어들인 레코드의 oetid 값의 길이를 비교하여 나중에 읽어들인 레코드의 oetid 값의 길이가 길거나 같다면 자식 엘리먼트가 존재하는 경우이므로 시작 태그를 스택에 푸시하고 문서에 content를 추가한 후 다음 레코드를 읽는다. 나중에 읽어들인 레코드의 oetid 값의 길이가 짧으면 end_tag의 값을 조사하여 end_tag의 값이 1(종료 태그 존재)이면 스택으로부터 마지막에 푸시한 시작 태그를 팝하여 이와 쌍을 이루는 종료 태그를 문서에 추가하고, 0(종료 태그 부재)이면 다음 레코드를 읽는다. 그림 7은 query_result 테이블로부터 질의 결과를 생성하기 위한 문서 통합 과정이다.

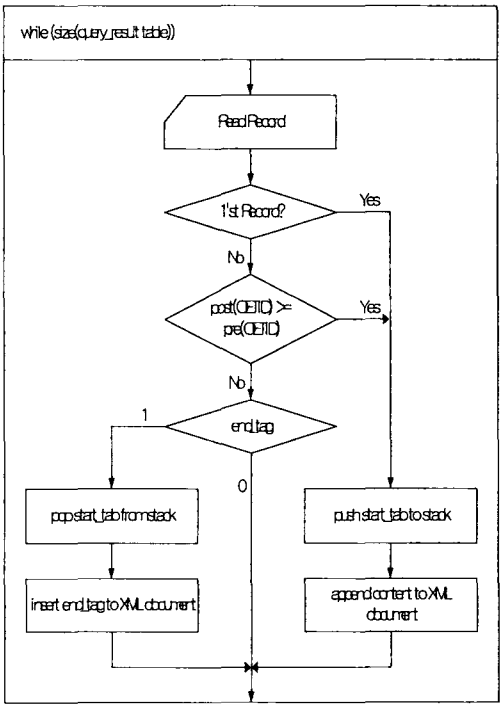


그림 7. 문서 통합 과정

4.3 실험 및 평가

XML 문서 검색을 위해 제안한 색인 모델의 비교 평가를 위해 기존 K-ary 완전 트리 구조 방법과 ETID에 의한 색인 모델을 비교 대상으로 하였다. 비교 항목으로는 구조 정보 표현 방법, 디스크 검색 회수, 그리고 각 엘리먼트 정보 크기로 하였다. 검색 성능 평가를 위해 디스크 접근 회수를 측정할 경우, K-ary 완전 트리 구조 방법은 K값을 구하기 위해 각 색인 파일마다 구조 색인

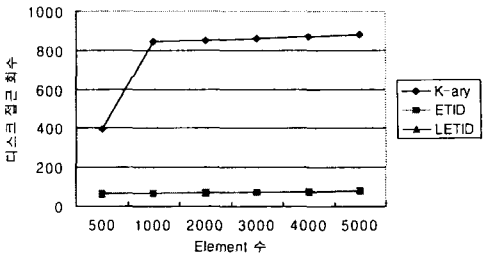
에 접근한다. 또한 문서 파싱을 통해 구한 형제나 자식 엘리먼트가 실제 문서에 존재하는 노드인지 가상 노드인지를 판별하기 위해 검색 결과 수만큼 다시 구조 색인에 접근하여야 한다. 그러나 제안한 색인 모델의 OETID 방법을 이용할 경우, ETID 방법을 이용할 때 추가되는 SORD와 SSORD를 위한 추가 정보 없이 하나의 정보로 부모, 자식, 형제 노드의 정보를 모두 표현하므로 가상 노드 여부를 판별하기 위한 접근 시간이 줄어든다. 각 방법에 대한 비교 평가 결과는 표 1과 같다.

표 1. 각 방법의 비교

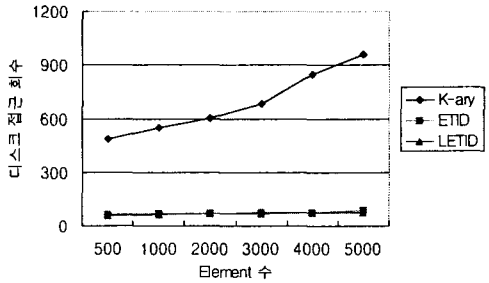
	K-ary	ETID	LETID
구조 정보 표현 방법	K-ary 완전 트리 방식으로 기존 트리를 완전 트리로 변환하여 노드 값을 구함	ETID, SORD, SSORD를 이용하여 엘리먼트 정보 표현	고정 크기의 LETID만으로 엘리먼트 정보 표현
디스크 검색 회수	$I_n+R_n*P_n+S_n*I_n$(부모검색) $I_n+R_n*P_n+S_n*I_n+R_n$(자식검색)	$2I_n+R_n*P_n$(부모/자식 검색)	$I_n+R_n*P_n$(부모/자식 검색)
엘리먼트 정보 크기	2바이트(정수형으로 노드 순서 표현)	ETID(depth * 2바이트) SORD,SSORD (depth * 1바이트)	OETID(depth * 2바이트) SORD, SSORD(추가 공간이 필요 없음)
장점	간단한 수식을 통해 부모/자식 관계의 엘리먼트를 구함	부모/자식/동일한 타입의 형제 엘리먼트의 순서 정보를 구할 수 있음	부모/자식/동일한 타입의 형제 엘리먼트의 순서 정보를 구할 수 있음
단점	노드의 깊이에 따른 노드변화가 크고, 사용하지 않는 노드가 많아지므로 데이터 양이 커짐	노드의 깊이가 깊어질수록 엘리먼트 정보를 표현하는 ETID의 크기가 계속적으로 증가함	노드의 깊이가 깊어질수록 엘리먼트 정보를 표현하기 위한 OETID의 크기가 계속 증가함

I_n : 각 색인 파일 접근 회수
R_n : 검색 결과
P_n : 포스팅 엔트리 크기
S_n : 구조 색인 접근 회수
SORD(Sibling ORDer)
SSORD(Same Sibling ORDer)

부모와 자식에 대한 구조 검색 시, 엘리먼트 수의 증가에 따른 디스크 접근 회수의 변화를 측정한 결과는 그림 8과 같다.



(a) 부모 검색



(b) 자식 검색

그림 8 구조 검색

V. 결론

이 논문에서는 스키마 기반의 XML 문서에 대한 저장 관리 시스템을 설계하였다. 이를 위해 XML 문서에 대한 구조 정보를 표현하는 방법과 구조 검색을 지원하기 위한 색인 구조를 제안하였다. 또한 시스템의 구조를 설계하고, XML 문서의 구조 정보를 저장하는 방법과 저장된 정보로부터 질의가 주어졌을 때 해당 정보를 추출하여 통합하는 과정을 제안하였다.

구조 정보 표현 방법은 스키마에 나타나는 각 엘리먼트에 대한 계층 정보, SORD, 그리고 SSORD를 표현한 OETID를 부여하여 고유값을 줌으로써 간단한 문자열 조작만으로 원하는 정보를 검색할 수 있다. 제안한 색인 구조는 내용 검색을 지원하는 내용 색인, 구조 검색을 지원하는 엘리먼트 색인, 애트리뷰트 검색을 지원하는 애트리뷰트 색인으로 구성된다. 또한 저장 관리 시스템의 저장 방식으로는 문서 전체를 검색할 경우 비분할 모델을 적용

하였고, 문서의 일부를 검색할 경우엔 분할 모델을 이용함으로써 검색 속도를 향상시켰다. 이와 같은 구조정보 표현과 색인을 이용하여 특정 엘리먼트에 직접적인 접근이 가능하고, 구조화 된 문서를 효율적으로 관리할 수 있으며, 다양한 질의처리가 가능하다. 향후 연구과제로는 제안한 방법을 기반으로 한 구조 문서 검색 시스템 환경에서 문서를 갱신할 때 요구되는색인 모델을 적용하기 위한 연구가 필요하다.

참고문헌

[1] 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

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

[3] W3C, "Extensible Markup Language(XML) 1.0, "http://www.w3c.org/TR/1998/REC-xml-19982 10.html, 1998.

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

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

[6] Sung-Geun Han, Jeong-Han Son, Jae-Woo Chang Zong-Cheol Zhoo "Design and Implementation of a Structured Information Tetrieval System for SGML Documents", IEEE, pp. 81-88, 1999.

[7] T. Dao, R.Sacks-Davis and J.A.Thom "An indexing scheme for structured documents and its implementation", In Proceedings of the 5th International Conference on Database Systems for Advanced Applications, pp 125-134, Melbourne, Australia, Aptial 1997.

[8] 이종설 외 7, "구조 정보 검색을 위한 XML 저장 관리시스템 설계 및 구현",

[9] 박종관, 강형일, 손충범, 유재수 "XML 문서에 대한 효율적인 구조 기반 검색을 위한 색인 모델", '2000 추계 학술발표논문집, 한국정보과학회, pp, 18-20, 2000.

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

저 자 소 개

조 윤 기

1994년 충북대학교 컴퓨터 과학과 졸업(학사)

1996년 충북대학교 대학원 전자계산학과졸업 (이학석사)

1998년 - 현재 충북대학교 대학원 전자계산학과 박사과정 수료

관심분야

저장관리시스템, 정보검색, 소프트웨어 테스팅

김 영 란

1988년 충북대학교 전산통계학과 졸업 (이학사)

1991년 충북대학교 대학원 전자계산학과 (이학석사)

1997년 충북대학교 대학원 전자계산학과 졸업 (이학박사)

1994 - 현재 충청대학 컴퓨터학부 부교수

관심분야

객체지향 소프트웨어 개발 기법, 정보검색, 인덱스 기법