XML문서의 검색을 위한 효율적인 색인 기법과 질의 언어(TQL)의 설계

이계준, 신동욱, 권택근,
충남대학교 컴퓨터공학과 정보검색 연구실

Efficient Indexing Technique for Retrieval of an XML Document and Design of Query Language (TQL)

Kyejun Lee, Dongwook Shin, Taeckgeun Kwon
Department of Computer Engineering, Chungnam National University

요 약

현재 WWW(**World Wide Web**), 사무 자동화 시스템(Office Information System), 전자 도서관(Digital Library)등의 빠른 발전으로 인하여 정보가 기하급수적으로 증가하였다. 이러한 방대한 양의 정보를 처리하기 위하여 많은 인터넷 기반의 문서 표준들이 출현하였고, 대표적으로 XML(eXtensible Markup Language)이 차세대 인터넷 전자 문서의 표준으로 많은 곳에 응용되고 있다. 이에 따라 XML 문서의 정보들을 효율적이고 정확하게 저장하고 이용, 검색 할 수 있는 기능을 요구되어졌다. 현재 대부분의 연구들은 XML 문서에 대한 구조적인 정보만을 저장하고 검색하는 기능만을 지원 할뿐 검색된 결과에 대한 재사용이나 재구성에 대한 기능의 제공은 미흡한 실정이다. 본 논문에서는 현재 검색기들이 제공하는 XML문서에 대한 구조적인 검색 기능을 확장하여 XML문서를 보다 효율적으로 검색하기 위하여 새로운 색인 기법을 제안하고, 데이터베이스 내에 저장된 XML문서에 대해 구조적인 검색과 이것을 바탕으로 문서를 재구성하고 재사용 하는 기능을 수행할 수 있도록 새로운 질의어(TQL)를 설계하였다

1. 서론

현재 인터넷에는 수 많은 양의 정보들이 전자 문서의 형태로 존재하고 있고, 계속적으로 이러한 정보들이 급속도로 증가하고 있는 상황이다. 이에 따라 이러한 전자 문서들을 보다 효율적으로 저장하고 검색, 이용하는 연구가 활발히 진행 중에 있다. 이러한 연구는 다양한 형태의 전자 문서의 표준을 요구하게 되었고, 이 가운데 W3C에서 제안한 XML(eXtensible Markup Language)이 차세대 인터넷 전자 문서의 표준으로 많은 연구 분야에서 사용하고 있다.

XML은 웹상에서 구조화된 문서를 전송 가능하도록 설계된 표준 마크업(Markup) 언어이다. XML은 인터넷 상에서 가장 많이 사용하는 HTML의 단순함을 극복하고, 복잡한 SGML을 단순화 했으며, HTML과 SGML의 표현 사이에서 상호 운용성 및 용이한 구현 환경을 제공한다[5]. 다양한 XML 관련 연구 중에서 디지털도서관, 전자상거래, 데이터웨어하우징, EDI(Electronic Data Interchange)등과 같은 분야에서는 XML문서를 효율적으로 저장, 검색, 이용하기 위한 많은 연구를 하고 있으나 지금까지는 XML문서에 대한 재구성과 재사용이 없는 검색 기능만을 제공하는 경우가 많다.

따라서 본 논문에서는 XML문서를 보다 효율적으로 검색할 수 있는 색인 기법을 제안하고, 저장된 문서들에 대해 검색과 재구성 및 재사용을 수행할 수 있는 질의어(TQL : Transitive xml Query Language)을 설계하였다.

본 논문의 구성은 다음과 같다. 2장에서는 XML의 특징과 XML문서에 대한 색인 구조를 제시하고, 3장에서는 데이터베이스에 저장된 정보를 추출하고 재구성할 수 있는 질의어(TQL)를 설계하고 마지막으로 결론을 내리도록 한다.

2. XML의 특징과 효율적인 색인 기법

XML은 확장된 Markup 언어로써 문법체계를 만드는 문서형 정의부인 DTD와 이를 바탕으로 만들어지는 실제 문서인 인스턴스로 구성이 된다. 이러한 구성은 문서를 구조적으로 표현하고 구조 정보에 대해 다양한 검색을 지원할 수 있도록 한다.

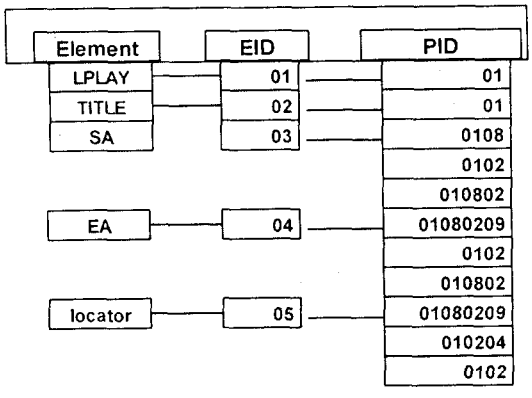
```
<!ELEMENT LPLAY     (TITLE, FM, PERSONAE, ACT+)>
<!ATTLIST LPLAY     ...     ID      #IMPLIED>
<!ELEMENT TITLE     (#PCDATA | %Anchor;)*>
<!ELEMENT SA        (#PCDATA)>
<!ELEMENT EA        (locator)+>
<!ELEMENT locator       (#PCDATA)>
                        . . .
```

[그림 1] DTD의 예

```
<?xml version="1.0" encoding="EUC-KR"?>
<!DOCTYPE LPLAY SYSTEM "Lplay.dtd">
<LPLAY id="1995">
<TITLE><EA xml:link="extended" inline="false" title="Title"
role="OtherPlay">
</EA></TITLE>
TITLE><SA xml:link="simple" title="Title" role="OtherPlay"
        The Comedy of Errors</SA></TITLE>
</LPLAY>
```

[그림 2] 인스턴스의 예

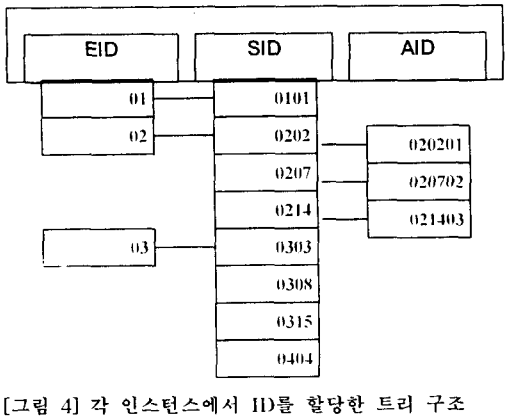
다음은 본 논문에서 제안하고 있는 DTD에 대한 색인 구조이다.



[그림 3] 각 DTD에서 ID를 할당한 트리 구조

DTD 색인에서는 DTD에서의 유일한 엘리먼트의 식별자인 EID(같은 이름을 가지는 엘리먼트는 같은 EID 갖게 됨)와 각 엘리먼트 사이의 포함 관계를 알기 위한 자신의 부모들에 대한 EID의 조합인 PID(자기 자신으로부터 부모가 되는 엘리먼트에 EID를 순서적으로 더해 줌)를 부여한다. [그림 3]은 한 문서가 가질 수 있는 같은 EID를 갖는 엘리먼트들을 나오는 순서대로 나열해 놓은 것이다.

다음은 인스턴스에 대한 색인 구조이다.



[그림 4] 각 인스턴스에서 ID를 할당한 트리 구조

인스턴스에서는 DTD에서 부여된 EID를 가지고서 같은 이름을 가진 엘리먼트를 구분하기 위한 SID(EID + Sequence number)와 엘리먼트가 가지는 어트리뷰트에 대한 AID(SID + Sequence number)를 부여한다. [그림 4]는 한 인스턴스 내에 여러 번 나오는 엘리먼트들을 나열해 놓은 것이다. 예를 들어 검색을 할 경우 LPLAY/TITLE/SA를 검색한다고 가정할 경우 DTD에 EID의 조합, 즉 LPLAY(01) + TITLE(02) + SA(03) = 010203(EID + PID와 동일)을 가지고 일단 필터링(Filtering)을 한 다음 실제 인스턴스에서의 SID를 통해서 검색이 이루어 진다. 이러한 검색은 DTD에서 필터링을 한다음 인스턴스를 검색하게 되므로 적은 범위의 데이터베이스 검색을 시도하므로 검색 효율이 높아 진다.

3. XML 문서 질의어

본 논문에서는 데이터베이스에 색인 되어 저장된 문서들에 대한 구조적인 검색과 검색 결과를 원하는 형태로 재구성 및 재사용할 수 있도록 하기 위한 보다 효과적이고 간결한 질의 언어를 제안하고 있다.

이러한 기능을 제공하는 것으로는 W3C에서 제안한 XML-QL[4]이 있는데 이것은 XML문서에 대한 구조를 알아야만 검색이 가능하며, 검색을 위해서 많은 데이터베이스의 접근을 필요로 한다. 또한 어트리뷰트에 의한 직접적인 검색이 제공되지 않는 비효율적인 면을 가지고 있다. 따라서, 여기서는 XML-QL이 가지는 단점을 해결하여 보다 효율적인 질의어(TQL)을 설계하였다. 다음은 TQL의 BNF 형태이다.

```
TQL ::= (Query) <EOF>
Query ::= Where Construct
Where ::= WHERE Condition (, Condition)*
Condition ::= Pattern BindingAs* IN DataSource | Predicate
|Select
Pattern ::= (Element (. Element)* = <VAR> | <STRING> (&&
            Pattern)*) | Attribute
Element ::= (<ID>|<VAR>) (/ Attribute)?
Attribute ::= (<ID> ( (<STRING> | <VAR>) )) | (= <VAR>)+
BindingAs ::= ELEMENT_AS | CONTENT_AS
DataSource ::= <DtdName> | <InstanceName>
Predicate ::= Expression OpRel Expression
Expression ::= <VAR> | <CONSTANT>
OpRel ::= < | <= | > | >= | = | !=
Select ::= <VAR> = <ID> (, <ID>)*
Construct ::= CONSTRUCT Tag? <VAR>+ | Element OrderedBy?
              Query?
Tag ::= < (<ID> | <VAR>)
OrderedBy ::= ORDERED-BY <VAR>
```

[그림 5] TQL의 BNF 형태

TQL은 XML문서에 대한 검색 결과에 대해서 재구성 및 재사용 기능을 제공하는 XML-QL을 기반하여 설계된 질의 언어이기 때문에 XML-QL이 가지고 있는 기능을 모두 제공한다. 또한 보다 간결하며 쉽게 사용할 수 있도록 설계 하였다. 다음은 다양한 검색을 TQL로 표현한 예이다.

1. Constructing XML Data

1) Bible.xml문서 내에서 Addison-Wesley라는 이름으로 Publisher된 책의 Title과 Author를 찾아서 <result>라는 엘리먼트로 묶어서 출력해라.

```
WHERE book.publisher.name = Addison-Wesley
      && title = $title && author = $author
      IN bible.xml
CONSTRUCT <result> $author $title
```

2.Grouping with Nested Queries.

1) Bible.xml문서 내에서 Addison-Wesley 라는 이름으로 Publisher된 책의 Title을 검색하고 출력은 같은 내용

을 가지는 Title은 한번만 <result> 엘리먼트 내에 author와 묶어서 출력해라.

```
WHERE book.publisher.name = Addison-Wesley
      && title = $title
      CONTENT_AS IN bible.xml
CONSTRUCT <result> $title
      Where author = $author IN bible.xml
      CONSTRUCT $author
```

2) Bible.xml문서 내에서 Addison-Wesley 라는 이름으로 Publisher된 책이나 author의 주소를 검색하고 출력은 같은 이름의 엘리먼트는 address는 한번만 <result> 엘리먼트내에 author와 묶어서 출력해라.

```
WHERE book.publisher.name = Addison-Wesley
      || author.address = $addr
      ELEMENT_AS IN bible.xml
CONSTRUCT <result> $addr
      Where author = $author IN bible.xml
      CONSTRUCT $author
```

3. Attribute에 의한 검색

: 엘리먼트에 포함된 속성으로서의 검색과 어트리뷰트의 차제만으로 이루어지는 검색이 있다.

1) 출판 년도가 1995인 책의 author의 firstname과 lastname을 출력해라.

```
WHERE book/year(1995).author.firstname = $firstname
      && $lastname IN bible.xml
CONSTRUCT <article> $author
```

2) year가 1995의 값을 가지는 것을 출력해라.

```
WHERE year(1995) = $year
CONSTRUCT $year
```

4. Integrating data from different XML sources

1) data.xml에서는 person의 name과 taxpayers.xml 에서는 taxpayer의 imcome을 엘리먼트의 중복없이 sorting해서 <result> 엘리먼트로 묶어서 출력해라.

```
WHERE person.name = $name ELEMANT_AS IN data.xml,
      taxpayer.income = $income ELEMENT_AS
      IN taxpayer.xml
CONSTRUCT <result> $name $income ORDERED_BY $name
```

2) data.dtd에서는 sorting 된 person의 name과 taxpayers.dtd에서는 taxpayer의 imcome을 엘리먼트의 중복없이 <result> 묶어서 출력해라.

```
WHERE person.name = $name ELEMANT_AS IN data.dtd,
      taxpayer.income = $income ELEMENT_AS
      IN taxpayer.dtd
CONSTRUCT <result> $name $income ORDERED_BY
$name
```

위의 질의 언어를 보면 Constructing XML Data은 하나의 XML문서 내에서의 엘리먼트 이름이나 값에 의한 질의, Grouping with Nested Queries은 CONTENT_AS나 ELEMENT_AS으로 검색 시에 서로 같은 내용을 포함하거나 엘리먼트의 이름이 같은 것이 존재할 경우에 중복을 피해서 한번만 출력할 수 있게 하는 질의, Attribute에 의한 검색은 엘리먼트에 포함된 속성으로서 보다 세밀한 검색을 하기위한 것과 직접적인 어트리뷰트 값에 의한 질의, Integrating data from different XML sources은 서로 다른 XML문서나 DTD 내에 있는 내용을 검색하는 질의이고 그 외에 XML-QL이 제공하는 기능을 제공한다. 이러한 질의는 기존에 내용이나 엘리먼트의 구조에 대한 질의만 되던 검색에서 검색된 결과를 가지고서 새롭게 재구성을 가능하게 함으로 문서의 이용가치를 높일 수 있으며 사용자가 보다 원하는 결과에 대한 질의를 여러 번 나누어 할 필요가 없고 문서를 편집하지 않고도 한번의 질의를 통해서 원하는 결과를 손쉽게 얻을 수 있게 한다. 또한 문서의 편집이 있을 경우에도 편집하고자 하는 부분을 질의를 통해 쉽게 변경할 수 있게 된다.

다음은 본 논문에서 제안하고 있는 XML 질의 언어인 TQL과 XML-QL과의 특성들을 비교한 것이다.

질의 언어	특 성
XML-QL	- XML문서 형태의 질의 문 - 어트리뷰트 검색 지원 안함 - 내용 검색, 구조 검색 지원 - 검색 결과에 대한 재구성 가능 - 문서의 구조를 알아야만 질의 가능
POET의 XQL	- UNIX의 디렉토리 형태의 구조 - Filter기능의 사용이 복잡 - 내용 검색, 구조 검색, 어트리뷰트 검색 지원 - 링크에 대한 검색 지원 안함 - 한 문서 내에서만 검색이 가능 - 검색 결과에 대한 재구성 불가
TQL	- XML-QL을 기반으로 하여 보다 효율적인 검색을 위해 간결하게 함으로써 사용자가 손쉽게 사용할 수 있는 질의 언어 - 내용 검색, 구조 검색, 어트리뷰트 검색 지원 - 검색 결과에 대한 재구성 가능 - XML문서 형태의 질의 문

4. 결론

XML은 많은 분야에서 이용가치를 인정 받았으며, 앞으로 더 많은 분야로 확장될 것이다. 그러므로, XML문서들을 관리하기 위한 다양한 도구들이 출현하고, XML문서들을 효율적으로 저장하고 관리하며 보다 정확한 검색이 중요하게 요구되어졌다. 하지만 이제는 검색된 결과를 가지고서 문서를 효과적으로 재편집하여 보다 많은 곳에 재편집된 문서를 사용 함으로 얻어지는 기대효과가 상당히 크다. 따라서, 많은 곳에서 이러한 연구가 진행 중이며 상당한 이슈가 되어 있다.

본 논문에서는 일반적인 XML 문서들을 효과적으로 검색하기 위해 꼭 해야 되는 색인 방법에 대해 효과적인 기법을 제시하였고, 이러한 색인 방법을 기반으로 저장된 정보들을 단순한 구조 검색뿐만 아니라 검색 결과를 재구성과 재사용할 수 있으며 보다 쉽게 사용할 수 있는 질의어(TQL)를 제시하였다.

참고문헌

[1]. 류은숙, 구조화된 멀티미디어 문서 모델링에 관한 연구, 박사학위 논문, 충남대학교, 1998.

[2]. 김용훈, 이강찬, 이규철, 링크 검색을 지원하는 XML문서 질의 언어(XQL)의 설계 Proceedings of The 25th KISS Fall Conference 1998

[3]. Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Extensible Markup Language (XML) 1.0, REC-xml-19980210,

[4]. Alin Deutsch, Mary Fernandez, Daniela Florescu, Alon Levy, Dan Suciu, XML-QL: A Query Language for XML, NOTE-xml-ql-19980819,W3C,

[5]. 김용훈, 다양한 구조 검색을 지원하는 XML 문서 검색기의 설계 및 구현, 석사 학위 논문, 충남대학교, 1999.