SGML 문서 검색시스템의 설계 및 구현

고 승규o, 조 승기, 백 승욱, 이 경호, 최 윤철

연세대학교 컴퓨터과학과
멀티미디어/그래픽스 연구실

Design and Implementaion of a Retrieval System for SGML Documents

SeungKyu Ko, SeungKi Cho, KyongHo Lee, SeungUk Baek, YoonChul Choy

Department of Computer Science, Yonsei University

요 약

문서의 논리적 구조정보를 표현하는 SGML 문서는 CALS, 디지털 도서관(Digital Library), 인터넷 분야에서 많이 사용되고 있다. 이렇게 SGML 문서들이 널리 사용됨에 따라 문서들의 효율적인 관리가 필요하게 되었고, 사용자가 원하는 문서를 정확하고 신속하게 검색해 줄 수 있는 검색 시스템의 개발이 필요하게 되었다. 좀더 정확한 문서를 검색해 내기 위해서는 SGML 문서의 특징인 구조 정보를 이용한 검색이 필수적이다. 이에 본 연구에서는 효율적인 SGML 문서의 검색을 위한 구조적 기반의 질의어로 eXQL 을 정의하고, 이를 지원하는 검색시스템을 개발하였다. 특히 질의어에 경로 연산자를 지원하여 원하는 정보를 정확하게 찾을 수 있도록 한다. 또한 본 시스템은 구조적인 검색을 효율적으로 지원하기 위하여 구조정보를 DSSSL의 Grove에 기반한 구조로 저장한다.

1. 서 론

SGML(Standard Generalized Markup Language)[3]은 이기종간의 호환이 가능하며 구조정보를 포함하고 있다는 장점 때문에 CALS(Commerce At Light Speed), EC(Electronic Commerce), EDI(Electronic Data Interchange) 등의 문서처리 표준 포맷으로 자리를 잡았다. 그리고 점차 많은 SGML 문서들이 사용되기 시작하면서 이러한 문서들의 효율적인 관리와 검색이 필요하게 되었다. 이에 본 연구에서는 먼저 SGML 문서의 구조 및 속성 정보를 효과적으로 표현할 수 있는 eXQL(EXtensible Query Language)을 정의하였으며 이에 기반한 검색시스템을 개발하였다.

본 논문의 구성은 다음과 같다. 2절에서는 SGML 문서의 효율적인 저장을 위하여 제안된 데이터 모델을 기술하고, 3절에서는 eXQL에 대해 자세히 설명한다. 4절에서는 제안된 검색시스템의 설계 및 구현을 소개하고, 마지막으로 5절에서는 결론 및 향후 연구 방향을 기술한다.

2. 데이터 모델

기존의 데이터베이스를 이용하면서 SGML문서의 특징인 구조 정보를 표현할 수 있는 데이터 모델은 관계형 모델, 확장 관계형 모델 그리고 객체 기반형 모델로 나눌 수 있다[1]. 관계형 모델에서는 구조정보를 표현하기 위해 테이블(table)이나 튜플(tuple)의 수가 많이 필요하게 된다. 또한 SGML 문서의 특징인 내포(nested)나 참조(reference) 등을 지원하기 힘들다. 이러한 내포나 참조 등을 지원하기 위해서는 관계형 모델을 수정한 확장 관계형 도델이나 SGML 문서와 자연스럽게 대응가능한 객체 기반형 모델을 이용할 수 있다.

확장 관계형 모델이나 객체 기반형 모델을 이용할 때, DTD(Document Type Definition)를 어떻게 모델링 하느냐에 따라 두 가지 방법으로 나눌 수 있다[1]. 첫번째 방법은 스키마(schema)를 생성하는 부분이 존재하여 새로운 종류(DTD)의 문서에 대해 스키마 생성기가 스키마를 생성해 주는 방법이다. 그래서 문서의 DI(Document Instance)부분은 생성된 스키마 형식으로 저장 된다. 두번째 방법은 모든 DTD를 표현할 수 있는 메타 스키마가 존재하여 이 메타 스키마를 이용하여 문서를 저장하게 된다. 전자는 특정 스키마를 생성해줌으로써 특정 문서 종류에 대해 성능이 좋을 수 있으나, 새로운 DTD에 대해서 새로운 스키마를 생성시켜주어야 한다. 후자는 일반성을 지녀 여러 문서 종류에 대해 사용할 수 있으나, 전자에 비해 성능이 약간 떨어질 수 있다.

본 시스템에서는 메타 스키마를 이용하여 SGML문서를 모델링하고, 저장 시스템은 객체 지향 데이터베이스를 사용하였다. 특히 메타 스키마는 효율적인 구조 기반의 내용 및 속성 검색을 지원하기 위하여 DSSSL(Document Style Semantics and Specification Language)의 Grove(Graph Representation Of property ValuEs)[4]에 기반한 구조를 이용하였다.

3. eXQL

SGML 검색시스템이 지원해주어야 될 검색의 종류는 일반적인 텍스트 검색, 우선 순위를 부여하는 검색, 질의의 대상을 제한하는 검색, 질의의 결과를 제한하는 검색, 문서의 구조에 대한 검색, 여러 문서 종류에 대한 검색, 마크업(markup) 자체에 대한 검색 등으로 나눌 수 있다[1].

본 연구에서는 구조정보를 이용한 내용검색과 속성 검색을 효과적으로 지원해주기 위한 질의어로 eXQL을 정의하였다. 특히 eXQL은 앞에서 언급한 검색들 중 전문 검색과 질의의 대상을 제한하는 검색, 여러 문서 종류에 대한 검색, 속성에 대한 검색 등을 지원하며 향후 계속 확장될 것이다. 위에서 언급한 특징 이외에 엘리먼트(element)를 지정할 때 경로를 이용할 수 있어서 좀 더 구체적으로 검색 대상을 제한 할 수 있다[2]. <그림 1>은 eXQL의 문법을 BNF(Backus-Naur Form)형식으로 표현한 것이다.

```
<query_expression> ::=
      FROM [<DTD_list>] CONTENT [<content_search >] |
      FROM[<DTD_list>] ATTRIBUTE [<attribute_search>]|
      FROM [<DTD_list>] CONTENT [<content_search >]
        ATTRIBUTE [<attribute_search>]
<DTD_list> ::=
        <DTD> | <DTD> <DTD_list>
<content_search> ::=
        <query_terms> % <element_names> |
        (<content_search >) |
         <content_search> <op> <content_search>
<query_terms> ::= "query_term" | ( <query_terms> ) |
      <query_terms> <op> <query_terms> |
      ! "query_term" | ! "query_term" <op> <query_terms>
<op>::= & | '|' | and | or
<element_names> ::=
      <element_name> | ( <element_names> ) |
      <element_names> <op> <element_names>
<element_name> ::=
        tag_name | tag_name <path_op><element_name>
<path_op> ::= . | ..
<attribute_search> ::=
      <attribute_name> = <query_attribute> |
      ( <attribute_search> ) |
      < attribute_search> <op> < attribute_search>
<query_attribute > ::= "query_attribute"
<attribute_name> ::=
      @ attribute_name | <element_names> @ attribute_name
```

<그림 1> eXQL의 BNF 형태

eXQL에서 지원 가능한 검색은 크게 구조기반의 내용검색과 구조기반의 속성 검색으로 나눌 수 있으며 이에 대한 자세한 설명은 다음과 같다.

3.1 구조기반의 내용 검색

구조기반의 내용 검색은 CONTENT절로 표현되며, 특정 문자열을 포함하는 엘리먼트를 갖고있는 문서를 찾는다. 이때 포함 관계를 표현하기 위하여 "%" 연산자를 사용한다. 검색 대상 엘리먼트를 지정할 경우, 하나 이상의 엘리먼트를 논리연산자(AND, OR)를 통하여 표현할 수 있다.

한편 "."와 ".." 등 경로 연산자를 제공하여 엘리먼트간의 계층적인 위상 관계를 표현할 수 있도록 하였다. "."는 계층적으로 바로 아래에 위치하는 엘리먼트를 표현하며, ".." 는 엘리먼트 간의 연결 관계만을 표현한다. 예를 들어 "Book.title"은 Book 엘리먼트의 바로 아래에 위치하는 title 엘리먼트를, "Book..title"은 Book 엘리먼트 아래에 위치하는 title 엘리먼트를 표현한다. 검색 문자열을 표현하는 "query_term"은 논리 연산자를 이용하여 연결할 수 있다. 특히 query_term에 대해서 "!"연산자를 제공하여 해당 문자열을 포함하지 않는 문서를 검색할 수 있도록 하였다. 그리고 CONTENT 절 내에는 여러 질의식(content_search)을 논리 연산자를 통하여 연결하여 표현할 수 있다.

3.2 구조 기반의 속성 검색

속성 검색은 엘리먼트의 속성에 대한 검색으로 속성을 표현하기 위해 "@"라는 연산자를 사용하였다. 예를 들어 title 엘리먼트의 id 속성은 "titile@id"로 표현할 수 있다. 또한 구조를 표현하기 위하여 엘리먼트의 이름에 경로 연산자를 이용할 수 있다. 그리고 어떤 엘리먼트에 속하는지 모르거나, 모든 엘리먼트에 나타날 수 있는 속성을 표현하기 위하여 엘리먼트 이름을 생략하고 "@" 연산자 다음에 속성 이름만을 기술할 수 있다.

3.3 질의문의 예

eXQL을 이용하여 사용 가능한 검색식의 예는 다음과 같다.

- 구조기반의 내용 검색

```
FROM [paper]
CONTENT [" sgml"  and " dsssl"  % book.title and
" Martin"  or " Goldfarb"  % book.author]
```

DTD가 paper인 문서 종류에서 book의 title이 " sgml" 과 " dsssl" 을 포함하고 있고, book의 author가 " Martin" 이나 " Goldfarb" 인 문서 검색.

- 구조 기반의 속성 검색

```
FROM [paper]
ATTRIBUTE[book.chapter@num=" 3" and @ID=" 길동" ]
```

DTD가 paper인 문서 종류에서 book 아래 chapter의 속성 num이 3이며, 속성 id의 값이 " 길동" 인 문서 검색.

- 속성 + 구조 기반의 내용검색

```
FROM [paper journal]
CONTENT [" xml"  or " sgml"  % chapter.title and
" 검색"  or " retrieval"  or " search"  %
chapter.paragraph]
ATTRIBUTE [chapter@num=" 10"  and reference@num =
" 10" ]
```

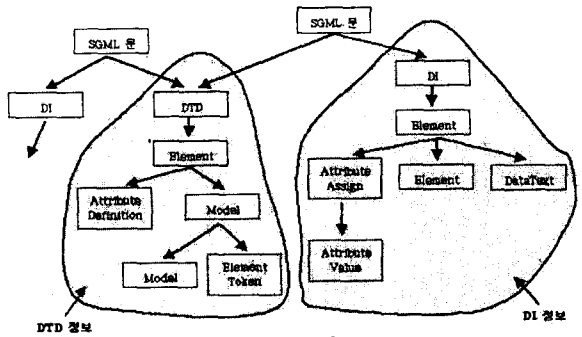
DTD가 paper나 journal인 문서 종류에서 chapter의 title이 " sgml" 또는 " xml" 을 포함하고, chapter 안에 있는 paragraph 가 " 검색" 이나 " retrieval" 또는" search" 를 포함하는 문서 중에서 chapter의 속성 num의 값이 " 10" 이고 reference의 속성 num의 값이 " 10" 인 문서를 찾아라(즉 속성 검색의 조건은 chapter와 reference의 개수가 10 이상인 조건으로 이해할 수도 있다.)

4. 설계 및 구현

4.1 저장 시스템

SGML 파서로는 공개용 파서인 SP를 사용하였고[5], SGML 문서는 Grove의 구조에 기반한 형태로 변환되어 저장 된다. 저장은 객체지향 데이터베이스인 오브젝트 스토어(ObjectStore)를 사용하였다. SGML 문서는 DTD와 DI로 구별 되는데 DI는 문서마다 다르나, DTD는 같은 종류의 SGML 문서에서는 같다. 그래서 DTD 구조를 공유

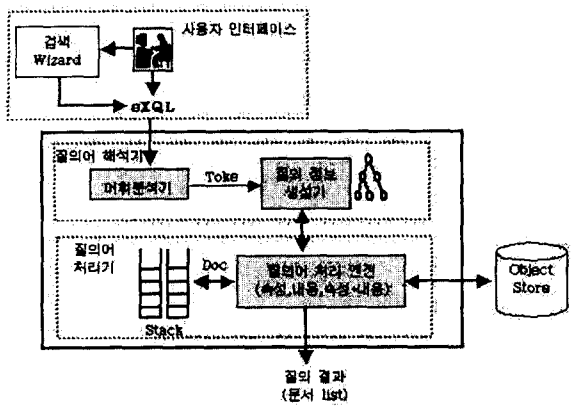
시킴으로써 저장 효율을 향상시켰다. <그림 2>는 개략적인 저장 구조를 나타내고 있다.



<그림 2> 저장 구조

4.2 구조 기반 검색시스템

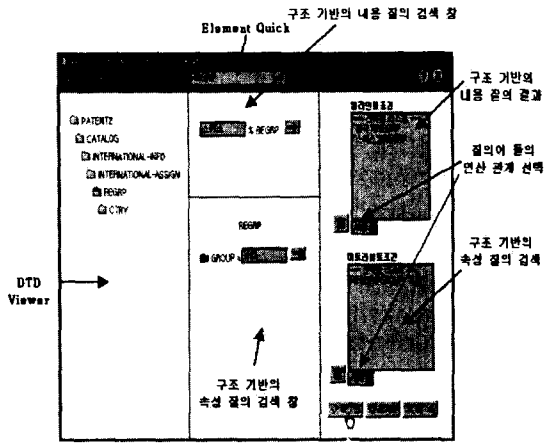
구조 기반 검색시스템은 크게 사용자가 질의를 입력하는 부분인 사용자 인터페이스 부분과 입력된 질의어를 검증하고, 후위식(postfix)형태로 바꾸어 저장하는 질의어 해석기 그리고 질의어 해석기의 정보를 이용하여 실제로 질의를 처리하는 질의어 처리기의 세 부분으로 나뉜다. 제안된 검색시스템의 구성은 <그림3>과 같다.



<그림 3> 검색 시스템의 구성

사용자 인터페이스에서는 검색식을 만드는 두가지 방법을 제공한다. 숙달된 사용자를 위해 eXQL을 직접 입력하는 방법과, 초보자를 위해서 질의어 생성 마법사를 이용하여 입력하는 방법으로 나눌 수 있다. 질의어 생성 마법사는 초보자를 위하여 검색식을 자동으로 생성시켜 준다. 특히 문서의 구조를 보여주는 DTD Viewer를 제공하여 초보자가 원하는 검색식을 만들도록 도움을 준다. <그림 4>는 질의어 생성 마법사의 사용자 인터페이스이다.

질의어 해석기는 어휘분석기와 구문분석기를 이용하여 질의문이 eXQL의 문법에 맞는지를 확인하고, 내부적으로 질의문을 후위식(postfix)으로 변환하고 이를 질의어 처리기에 제공한다. 질의어 처리기는 질의어 해석기의 결과를 이용하여 실제 검색을 수행한다. 질의문을 질의어 해석기가 처리한 결과는 후위식 형태이다. 이에 질의어 처리기는 이를 단위별로 처리하여 이를 스택에 저장하고, 논리 연산자(AND 또는 OR)를 만나면 이를 스택의 두 검색 결과에 적용한다. 이러한 과정을 반복적으로 수행하면 해당 질의문에 대한 최종 검색 결과를 얻을 수 있다.



<그림 4> 질의어 생성 마법사

5. 결론 및 향후 연구 방향

본 연구에서는 효율적인 구조 검색을 위해 SGML문서를 Grove에 기반한 구조로 바꾸어 객체지향 데이터베이스에 저장하는 저장시스템을 구현하였다. 또한 효율적인 구조 기반의 내용과 속성 검색을 지원하는 eXQL을 정의하고, eXQL을 지원하는 검색시스템을 설계 및 구현하였다.

본 연구에서 제시한 eXQL의 기능을 확장하여 검색 결과에 대해 제한을 둘 수 있게 하여, 그 검색 결과를 다른 곳에서 사용 가능하게 처리하면 SGML 문서의 생산성이 높아지게 될 것이다. 또한 엘리먼트 위치에 대한 검색, 속성과는 별개로 링크에 대한 검색 등도 지원되어야 될 것이다. 향후 검색시스템에 대한 성능 평가와 효율적인 검색을 위한 구조 색인 등에 대한 연구가 진행되어야 될 것이다.

6. 참고 문헌

[1] R. Sacks-Davis, T. Arnold-Moore and J. Zobel, "Database Systems for Structured Documents," IEICE TRANS. INF. & SYST., Vol.E78 D. No. 11, pp.1335~1341,1995

[2] S. Abiteboul. S. Cluet. V. Christophides. T. Milo. G. Moerkotte. J. Simeon, "Querying Document in Object Databases", Journal of VLDB, 1997

[3] ISO 8879, Information Processing - Text and office system - Standard Generalized Markup Language (SGML), ISO, 1986

[4] ISO/IEC 10179, Information Technology - Text and office system - Document Style Semantics and Specification Language (DSSSL), ISO/IEC, 1996

[5] SP : http://www.jclark.com/sp/index.htm