이질적 분산 데이터베이스의 통합검색을 위한 질의어 XML DTD 설계에 관한 연구

A Study on the XML DTD Design of Query for Integrated Retrieval of Heterogeneous Distributed Databases

이성진, 충남대학교 대학원 문헌정보학과
이응봉, 충남대학교 문헌정보학과

Sung-Jin Lee, Eung-Bong Lee
Dept. of Library & Information Science, Chungnam National University

정보 저장 검색 기술의 발달로 여러 개의 데이터베이스를 한꺼번에 검색할 수 있는 분산통합검색 시스템에 관심이 증가하고 있다. 그러나 데이터베이스의 종류 및 검색방식이 다양하기 때문에 분산통합검색 시스템의 구축에는 통합검색의 확장성과 데이터베이스간의 독립 운영성이 떨어지는 문제가 있다. 따라서 본 고에서는 이러한 문제점 해결을 위해 기존 데이터베이스들의 질의 구조를 분석해서 질의 구조의 핵심 요소들과 관련 요소들을 추출한 후, XML을 사용하여 대부분의 데이터베이스의 질의어를 포괄할 수 있는 질의어 Meta Format을 설계한다. 이렇게 작성한 표준화된 XML 질의어 Meta Format(DTD)은 분산통합검색에 적용되어 분산통합검색 시스템과 지역 데이터베이스들간의 독립 운영성 및 확장성을 증대시킬 전망이다.

1. 서 론

21세기 지식기반사회는 정보의 홍수 시대로서 이용자는 정보 획득에 많은 노력과 시간을 소비하고 있다. 게다가 각각의 데이터베이스들은 서로 다른 질의어 입력 방식을 취하고 있기 때문에 이용자는 검색 대상 데이터베이스의 질의어 입력 방식을 모두 알아야만 적합한 결과를 얻을 수 있다. 예를 들어 "정보" 와 "검색" 이라는 단어가 모두 들어간 문서를 검색할 경우, Altavista는 "+정보 +검색" 이라고 질의하지만 Lycos는 "정보 & 검색"이라고 질의한다. 따라서 한 번의 질의어 입력으로 여러 데이터베이스의 질의 결과를 동시에 얻을 수 있는 방법을 찾게 되는데, 바로 분산통합검색이 이에 해당한다. 하지만 분산통합검색을 수행하기 위해서는 우선적으로 여러 데이터베이스에 동일하게 적용되는 질의어 메타 포맷이 요구된다.

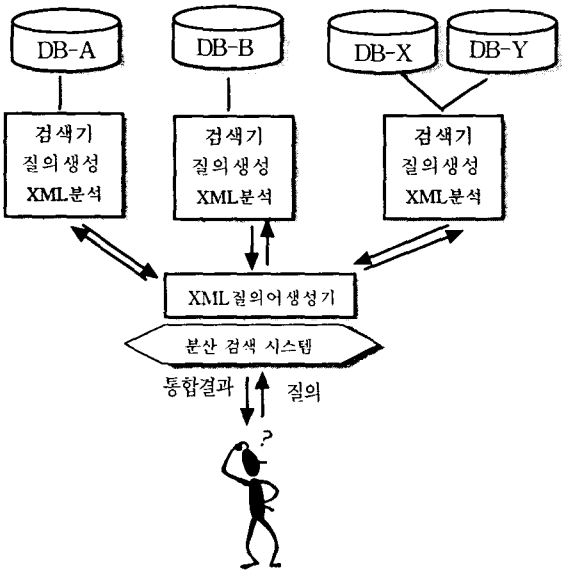
따라서 본 고에서는 분산통합검색에서 요구하는 동일 질의어 포맷을 작성하기 위해 확장성이 뛰어난 XML(extensible Markup Language)을 사용해서 질의어 DTD를 설계한다.

2. XML 및 분산통합검색

2.1 XML

XML은 SGML의 한 부분으로 1996년 W3C가 제안했으며 구조화된 문서의 웹 전송을 가능하게 하는 표준화된 텍스트 형식이다. 기존의 HTML과 SGML은 웹 사용에 있어서 제약사항이 많았지만 XML은 SGML에서 거의 사용하지 않는 사항들은 모두 없애고 꼭 필요한 기능만을 수용하여 기존의 HTML을 확장하고 보완함으로서 HTML보다 더 복잡한 문서 생성을 가능하게 하고 구조적 정보를 포함할 수 있도록 설계되었다.

따라서 이러한 특징의 XML을 이용해서 질의어 DTD(Document Type Definition)를 작성해 놓으면, 분산통합검색 시스템에 참여하는 여러 데이터베이스들은 동일한 XML 질의어 DTD를 사용함으로서 질의어가 데이터베이스에 일관성 있게 적용되도록 한다(그림 1).



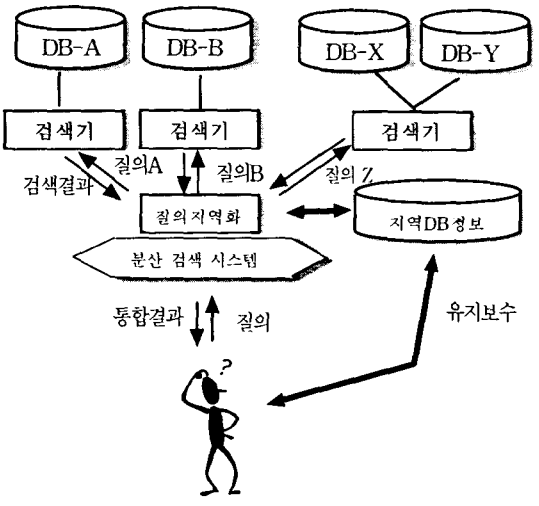
(그림 1) XML 질의어를 이용한 분산통합검색시스템

2.2 분산통합검색

인터넷의 정보의 양이 기하급수적으로 증가하면서 정보발견 시스템을 이용하지 않고는 원하는 정보를 찾을 수 없게 되었다. 여기서 대두되는 것이 분산통합검색 시스템(그림 2)이다. 분산통합검색 시스템은 자체적으로 데이터베이스를 가지고 있지 않을 뿐 아니라 정보검색 알고리즘도 수행하지 않는다. 다만 데이터 소스를 찾아다니며 정보 검색을 의뢰하고 적당한 정보를 찾아서 사용자에게 제공하는 역할을 한다. 분산통합검색 시스템의 특성은 다음과 같다.

- 현존하는 모든 정보원 혹은 원천 시스템을 통합할 수 있는 포용성을 갖춤.
- 기존 원천 시스템에 대한 자율성을 보장.
- 통합 환경에서의 상호운용성과 새로운 원천 시스템의 참여를 용이하게 하는 확장성을 갖춤.

분산통합검색 관련 연구로는 STARTS(Stanford Protocol Proposal for Internet Retrieval and Search), GILS, SDLIP(Simple Digital Library Interoperability Protocol), Dienst 시스템(분산 디지털 도서관 시스템을 구축할 수 있도록 해주는 프로토콜), Z39.50(네트웍 정보 검색용 표준 프로토콜)이 있다.



(그림 2) 분산통합검색시스템

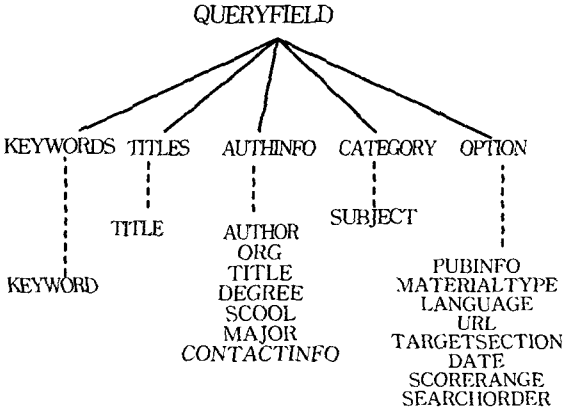
3. 질의어 XML DTD 설계

3.1 연구의 설계

본 연구는 XML DTD 설계를 위해서 인터넷 웹사이트 20개를 선정하여 각 사이트의 검색 질의 포맷 구조를 분석한다. 포털 사이트 4개, 신문 사이트 4개, 전자 도서관 사이트 8개, 대학 도서관 사이트 4개의 질의 포맷 구조를 분석한 후 핵심적인 상호관련 요소들과 각 사이트들마다 가지는 특유의 요소들을 추출하고, 이렇게 추출된 요소 모두는 XML DTD에 요소 및 속성으로 포함시킨다. 하지만 각각의 웹 사이트들은 질의 구조가 매우 다양하기 때문에 모든 요소를 질의 포맷에 포함시키는 데는 한계가 있다.

3.2 질의어 XML DTD 설계

웹 사이트 20 곳의 질의 포맷을 조사한 결과 추출한 요소들의 구조는 우선 키워드 부분(KEYWORDS), 서명 부분(TITLES), 저자 부분(AUTHINFO), 주제 부분(CATEGORY), 기타 부분(OPTION)으로 구성되며, 각각 하위요소들을 갖는다(그림 3).



(그림 3) 요소들의 계층화

(그림 4)는 XML 문서선언 및 키워드 DTD를 정의한 내용이다. XML 1.0 표준 규약에 따르면 모든 XML 문서는 '<?xml'로 시작하므로 DTD 맨 처음에 <?xml을 선언하고 버전 정보와 문자 인코딩 방식을 정의한다.

웹 사이트 20 곳의 질의어 포맷을 살펴보면 기본적으로 키워드 요소와 키워드 요소를 제외한 요소로 나뉜다. 따라서 질의어 DTD의 상위 요소로 'QUERYFIELD'를 정의하고 하위요소로서 키워드부문(KEYWORDS), 서명부분(TITLES), 저자부분(AUTHINFO), 주제부분(CATEGORY), 기타부분(OPTION)을 정의한다. QUERYFIELD 하위 요소들은 임의의 순서로 나오므로 연결자 '|'로 정의하고, 1번 이상 나타나므로 발생자 '+'를 부여한다. QUERYFIELD 속성인 OPERATOR는 QUERYFIELD 요소의 하위 요소간 연산을 의미한다. 즉, OPERATOR 값이 'AND'이고 자식 요소로서 TITLES와 AUTHINFO를 갖는다면 TITLES와 AUTHINFO 요소의 해당 질의어를 'AND' 연산한다.

(그림 5)는 QUERYFIELD 하위 요소인 KEYWORDS를 정의한다. KEYWORDS는 하위 요소로 KEYWORD를 갖으며, KEYWORD가 0개 또는 무한정 나올 수 있으므로 발생지시자 '*'로 정의한다. 속성인 OPERATOR는 하위 요소인 KEYWORD 사이의 연산을 의미한다.

(그림 6)은 QUERYFIELD의 하위 요소인 서명부분(TITLES), 저자부분(AUTHINFO), 주제부분(CATEGORY)을 설명한다.

TITLES 요소는 서명을 정의하며, 자식 요소로서 TITLE을 갖는다. 속성인 OPERATOR는 하위 요소인 TITLE 간의 연산을 의미한다.

AUTHINFO는 저자정보로서 저자명(AUTHOR), 저자소속기관(ORG), 저자직분(TITLE), 학위(DEGREE), 출신학교(SCHOOL), 전공(MAJOR), 저자 연락 정보(CONTACTINFO)를 하위요소로 정의한다. AUTHINFO 요소는 순서에 상관없이 나올 수 있기 때문에 연결자 '|'와 발생 지시자 '*'로 정의하였다. 'AUTH'는 'AUTHOR | ORG | TITLE | DEGREE | SCHOOL | MAJOR

```
<?xml version="1.0" encoding="euc-kr"?>
<!-- ======= Start Element Declaration ======= -->
<!ELEMENT QUERYFIELD ((KEYWORDS) | (TITLES | AUTHINFO | CATEGORY | OPTION ))+>
<!ATTLIST QUERYFIELD OPERATOR (AND | OR | NOT | NONE) #REQUIRED >
```

(그림 4) XML 문서 선언 및 QUERYFIELD DTD

```
<!ELEMENT KEYWORDS (KEYWORD)*>
<!ATTLIST KEYWORDS OPERATOR (AND | OR | NOT | NONE) #REQUIRED>
<!ELEMENT KEYWORD (#PCDATA)>
```

(그림 5) KEYWORDS DTD

```
<!ELEMENT TITLES (TITLE)*>
<!ATTLIST TITLES OPERATOR (AND | OR | NOT | NONE) #REQUIRED>
<!ELEMENT TITLE (#PCDATA)>
<!ELEMENT AUTHINFO (%AUTH;)*>
<!ENTITY % AUTH "AUTHOR | ORG | TITLE | DEGREE | SCHOOL | MAJOR | CONTACTINFO">
<!ATTLIST AUTHINFO OPERATOR (AND | OR | NOT | NONE) #REQUIRED>
<!ELEMENT AUTHOR (#PCDATA)>
<!ELEMENT ORG (#PCDATA)>
<!ELEMENT TITLE (#PCDATA)>
<!ELEMENT DEGREE (#PCDATA)>
<!ELEMENT SCHOOL(#PCDATA)>
<!ELEMENT MAJOR (#PCDATA)>
<!ELEMENT CONTACTINFO (EMAIL | HOMEPAGE | PHONE | ADDRESS)*>
<!ATTLIST CONTACTINFO OPERATOR (AND | OR | NOT | NONE) #REQUIRED>
<!ELEMENT EMAIL (#PCDATA)>
<!ELEMENT HONEPAGE (#PCDATA)>
<!ELEMENT PHONE (#PCDATA)>
<!ELEMENT ADDRESS (#PCDATA)>
<!ELEMENT CATEGORY (SUBJECT)*>
<!ATTLIST CATEGORY OPERATOR (AND | OR | NOT | NONE) #REQUIRED>
<!ELEMENT SUBJECT (#PCDATA)>
```

(그림 6) TITLES, AUTHINFO, CATEGORY DTD

| CONTACTINFO'를 파라미터 개체로 선언한 것이다.

AUTHINFO의 하위 요소인 CONTACTINFO는 EMAIL, HOEMPAGE, PHONE, ADDRESS 요소로 구성된다.

CATEGORY는 해당 주제와 관련된 부분으로 SUBJECT라는 자식요소를 갖으며, 발생지시자 '*'로 설계하고 SUBJECT 요소는 #PCDATA로 정의한다. 그리고 CATEGORY 속성인 OPERATOR는 하위요소인 SUBJECT간의 연산을 담당한다.

(그림 7)은 OPTION 요소를 정의한다. 이 요소는 QURYFIELD에서 KEYWORDS, TITLES, AUTHINFO, CATEGORY를 제외한 하위 요소이다. OPTION 요소는 출판 정보인 PUBINFO, 자료 유형인 MATERIALTYPE, 언어 LANGUAGE, 웹사이트 주소인 URL, 해당 부분을 지정하는 TARGETSECTION, 검색 기간을 지정하는 DATE, 검색 건수를 선택하는 SCORERANGE, 검색 결과 순서를 지정하는 SEARCHORDER를 하위 요소로 그룹화한다. 해당 요소들이 임의의 순서로 나타날 수 있으며, 또한 한 번도 나오지 않을 수 있으므로 연결자 '|'와 발생지시자 '*'로 정의한다. OPTION 요소는 속성으로 OPERATOR를 갖으며, 하위 요소들간의 연산을 처리한다.

또한 OPT는 OPTION의 하위요소인 'PUBINFO', 'MATERIALTYPE', 'LANGUAGE', 'URL', 'TARGETSECTION', 'DATE', 'SCORERANGE', 'SEARCHORDER'를 연결자 '|'로 그룹화하여 파라미터 개체로 선언한다.

```
<!ELEMENT OPTION (%OPT;)*>
<!ENTITY   %   OPT   "PUBINFO | MATERIALTYPE | LANGUAGE | URL | TARGETSECTION | DATE | SCORERANGE |
SEARCHORDER">
<!ATTLIST OPTION OPERATOR (AND | OR | NOT | NONE) #REQUIRED>
<!ELEMENT PUBINFO (PUBLISHER | PUBYEAR | PUBPLACE | FREQUENCY)*>
<!ATTLIST PUBINFO OPERATOR (AND | OR | NOT | NONE) #REQUIRED>
<!ELEMENT PUBLISHER (#PCDATA)>
<!ELEMENT PUBYEAR (#PCDATA)>
<!ELEMENT PUBPLACE (#PCDATA)>
<!ELEMENT FREQUENCY (#PCDATA)>
<!ELEMENT MATERIALTYPE (TYPE,ORIGINAL)*>
<!ATTLIST MATERIALTYPE OPERATOR (AND | OR | NOT | NONE) #REQUIRED>
<!ELEMENT TYPE (#PCDATA)>
<!ELEMENT ORIGINAL EMPTY>
<!ATTRIST ORIGINAL SUPPORTED (YES | NO) #REQUIRED>
<!ELEMENT LANGUAGE (#PCDATA)>
<!ELEMENT URL (#PCDATA)>
<!ELEMENT TARGETSECTION ((BODY,TITLE) | ABST | TOC | AUTHOR)*>
<!ELEMENT BODY EMPTY>
<!ATTRIST BODY SUPPORTED (YES | NO) #FIXED "YES" >
<!ELEMENT TITLE EMPTY>
<!ATTRIST TITLE SUPPORTED (YES | NO) #FIXED "YES" >
<!ELEMENT ABST EMPTY>
<!ATTRIST ATST SUPPORTED (YES | NO) #REQUIRED >
<!ELEMENT TOC EMPTY>
<!ATTRIST TOC SUPPORTED (YES | NO) #REQUIRED >
<!ELEMENT AUTHOR EMPTY>
<!ATTRIST AUTHOR SUPPORTED (YES | NO) #REQUIRED >
<!ELEMENT DATE (START?, END?)>
<!ELEMENT START (#PCDATA)>
<!ELEMENT END (#PCDATA)>
<!ELEMENT SCORENUM (MIN? | MAX?)>
<!ELEMENT MIN (#PCDATA)>
<!ELEMENT MAX (#PCDATA)>
<!ELEMENT SEARCHORDER (ASC? | DESC? | RELATE?)>
<!ELEMENT ASC (#PCDATA)>
<!ELEMENT DESC (#PCDATA)>
<!ELEMENT RELATE (#PCDATA)>
```

(그림 7) OPTION DTD

출판정보인 PUBINFO는 PUBLISHER(출판사), PUBYEAR(출판년), PUBPLACE(출판지), FREQUENCY(간행빈도)를 하위 요소로 그룹화한다. 속성으로 OPERATOR를 갖으며, 하위 요소들은 임의의 순서로 0번 이상 발생할 수 있기 때문에 연결자 '|'와 발생지시자 '*'로 정의한다.

자료유형 요소인 MATERIALTYPE은 형태를 정의한 TYPE 요소와 원문의 유무를 나타내는 ORIGINAL 요소를 하위요소로 둔다. TYPE 요소가 나온 후 ORIGINAL 요소가 나오기 때문에 연결자 ','로 정의하며, 이 요소들은 한번도 나오지 않을 수 있으므로 발생지시자 '*'로 설계한다. ORIGINAL은 속성 값으로 원문 유무를 선택하도록 한다.

LANGUAGE 요소와 URL 요소는 #PCDATA로 정의한다.

TARGETSECTION 요소는 속성으로 본문(BODY), 서명(TITLE), 초록(ABST), 목차(TOC), 저자(AUTHOR) 가운데 사용자가 선택할 수 있도록 하며 본문(BODT)과 서명(TITLE)은 디폴트 값으로 설계한다.

DATE 요소는 검색 기간을 정의하며 하위요소로서 어느 기간부터 시작해서 어느 기간까지

검색 대상으로 할 것인지 선택하도록 START 요소와 END 요소를 하위 요소로 설계한다. 이 요소들은 순서대로 나오므로 연산자 ','로 정의하며, 한 번 또는 한 번도 나오지 않을 수 있기 때문에 발생지시자 '?'로 나타낸다.

SCORENUM은 질의에 대한 검색 문헌의 개수를 사용자가 지정해 주는 것으로서 최소 검색 개수인 MIN과 최대 검색 개수인 MAX를 하위요소로 갖는다.

SEARCHORDER는 검색 결과 순서를 정의하는 것으로 ASC(오름차순), DESC(내림차순), RELATE(관련성) 요소를 자식 요소로서 두며, 한 번 또는 한 번도 나오지 않을 수 있으므로 발생지시자 '?'를 부여하고, 또한 임의의 순서대로 나오므로 연결자 '|'로 정의한다.

4. 결 론

여러 데이터베이스를 검색하고자 하는 정보이용자는 각각의 데이터베이스에 질의어를 작성하여서 만족스러운 결과를 입수할 때까지 수많은 데이터베이스를 검색해 봐야 한다.

하지만 하나의 질의어 입력 방식이 모든 데이터베이스의 질의 방식을 만족시키지는 못한다. 이때 제시되는 해결책으로 분산통합검색을 들 수 있으며, 이 방식을 사용하기 위해서는 동일한 질의어 Meta Format이 요구된다.

따라서 본 고에서는 이러한 질의어 Meta Format을 작성하기 위해 기존 데이터베이스들의 질의 구조를 분석한 후 핵심 요소들과 관련 요소들을 추출했다. 그런 다음 데이터베이스의 질의 구조 대부분을 포괄할 수 있는 하나의 질의어 Meta Format(DTD)을 설계했으며, 이 때 확장성이 뛰어난 XML을 사용했다.

이렇게 설계한 표준화된 XML 질의어 포맷은 분산통합검색 시스템에 적용되어서 분산통합검색 시스템과 지역데이터베이스들간의 독립운영성 및 확장성 증대를 도모하고, 또한 한 번의 검색으로 여러 데이터베이스들의 결과를 동시에 이용자에게 제공하여 검색 효율성도 높일 전망이다.

참 고 문 헌

김현희, 장혜원. 1999. 디지털도서관 문서 양식으로서의 XML과 HTML의 특성 및 검색 기능 비교 연구. 『정보관리학회지』. 16(2) : 105-134.

박기복. 2000. 차세대 인터넷 표준언어 XML. 『전자신문』. <http://www.etimesi.com/news/detail.html?id=200005090063>

안영선. 2000. 학위논문의 XML DTD 설계에 관한 연구. 『정보관리학회지』. 17(4): 113-129.

연구개발정보센터. 2001. 『분산통합기 개발 최종연구보고서』.

이민호. 2000 『분산 검색 환경에서의 효율적인 융합 방법』. 석사학위논문, 충남대학교 대학원.

Dienst, A Protocol for a Distributed Digital Document Library. <http://www.broadcatch.com/dienst.html>

Extensible Markup Language (XML) 1.0 (Second Edition), 2000. <http://www.w3c.org/TR/2000/REC-xml-20001006>

Global Information Locator Service. <http://www.gils.net>

STARTS : Stanford Protocol Proposal for Internet Retrieval and Search. <http://www-db.stanford.edu/~gravano/starts.html>

Taehee Kim. 1999. Integration of multiple heterogeneous databases using a client-side meta search agent. 『IAT'99 Proceedings of the 1st Asia-Pacific Conference on Intelligent Agent Technology』. 234-243.

The Open Archives Initiative Protocol for Metadata Harvesting. <http://www.openarchives.org/OAI/openarchivesprotocol.htm>

http://www.ucc.ie/xml

http://www.xml.com

http://www.xml.org