Journal of Korea Multimedia Society Vol. 8, No. 5, May 2005(pp. 596-605)

온톨로지 기반 문서여과 및 질의확장에 의한 XML 정보검색

김명숙[†], 공용해[††]

요 약

기존의 XML 질의 방법은 단순 키워드 정합이나 단순 구조적 확장 등에 국한되므로 문서에 내재된 정보를 검색하기에 불충분할 뿐만 아니라, 모든 문서에 질의를 인가함으로써 정보검색의 효율을 저하시킨다. 본 연구는 온톨로지로부터 생성한 포괄적 DTD에 의해 검색대상 문서를 사전에 미리 선별하는 문서여과 방법과 온톨로지의 개념구조와 개념 간 연관관계를 추론하여 질의를 확장하는 방법을 제안함으로써 XML 정보검색의 효과를 증대하고자 한다. 제안한 문서여과 및 질의확장 방법은 다양한 XML 문서를 대상으로 검색 효과를 실험하였다.

XML Information Retrieval by Document Filtering and Query Expansion Based on Ontology

Myung Sook Kim[†], Yong Hae Kong[††]

ABSTRACT

Conventional XML query methods such as simple keyword match or structural query expansion are not sufficient to catch the underlying information in the documents. Moreover, these methods inefficiently try to query all the documents. This paper proposes document filtering and query expansion methods that are based on ontology. Using ontology, we construct a universal DTD that can filter off unnecessary documents. Then, query expansion method is developed through the analysis of concept hierarchy and association among concepts. The proposed methods are applied on variety of sample XML documents to test the effectiveness.

Key words: Ontology(온톨로지), XML, DTD, Query Expansion(질의확장)

1. 서 론

최근 웹상의 표준 문서로서 대두된 XML(eXtensible Markup Language)에 대한 관심과 연구가 여러 분야에서 진행되고 있다. 그러나 동일한 정보를 가지는 XML 문서라도 구조와 형식이 매우 다양하게 표현될 수 있으므로 정보검색에 어려움이 따른다[1]. 이러한 XML 문서의 구조적 다양성에 기인한 정보검색의 어려움은 온톨로지(Ontology)를 통한 접근으로 해결될 수 있다. 온톨로지는 정보 교환용으로 합의된 어휘나 구조를 만들기 위해 사용되는 개념들의 집합을 의미한다. 현재 W3C 등에서 응용 영역에 대한 온톨로지 표준화 작업을 진행하고 있으므로, 온톨로지를 기반으로 XML 문서를 접근하려는 시도는 향후 보다 활발해지리라 예상한다[2].

XML 정보검색 연구는 대부분 XML 문서에 대한 키워드 정합 등의 단편적인 정보검색에 국한되고 있다[3,4]. 단순 키워드 검색을 보완하는 방법으로

※ 교신저자(Corresponding Author) : 김명숙, 주소 : 충남 아산시 신창면 읍내리(336-745), 전화 : 019-434-6137, FAX : 041)530-1548, E-mail : krhkms@sch.ac.kr
접수일 : 2004년 6월 25일, 완료일 : 2004년 12월 27일
[†] 준회원, 순천향대학교 정보기술공학부 박사과정
[††] 정회원, 순천향대학교 정보기술공학부 교수
(E-mail : yhkong@sch.ac.kr)
※ 본 논문은 정보통신부 정보통신연구진흥원에서 지원하고 있는 정보통신기초기술연구지원사업의 연구결과입니다.

Erdmann 등은 온톨로지를 이용하여 인가된 질의를 확장한 바 있으나 온톨로지의 개념을 하위개념으로 확장하는데 그쳤다[5]. XML 문서에 내포된 중요한 정보를 검색하기 위해서는 개념의 구조적 관계뿐만 아니라 개념간의 연관관계를 추론하는 방법이 필요하다.

본 논문은 XML 질의를 모든 문서에 인가함으로써 발생하는 검색의 비효율성을 해결하기 위하여 불필요한 문서를 선택적으로 사전에 여과함으로써 질의의 효율을 증대하고자 하였다. 또한 온톨로지의 개념구조 및 온톨로지에 묵시적으로 내포된 연관관계 추론규칙에 의해 질의를 확장함으로써 문서에 내재된 중요한 의미정보를 검색하고자 하였다. 그림 1은 온톨로지 기반의 문서여과 및 질의확장에 의한 정보검색 과정을 나타낸 것이며, 제안한 방법에 의해 한 번 생성된 포괄적 DTD와 연관관계 규칙은 동일 영역의 XML 문서를 선별하고 질의를 확장하는데 재사용될 수 있다.

2장에서 관련연구 및 본 논문의 접근방법을 살펴보고, 3, 4장에서 문서여과 및 질의확장의 세부 방법을 설명한다. 5장에서는 XML 문서여과 및 질의확장의 효과를 예제 XML 문서에 적용하였다.

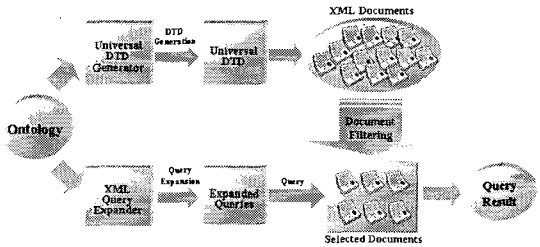


그림 1. 온톨로지 기반의 문서여과 및 질의확장 과정

2. XML 정보검색 배경연구

현재, XML의 구조적 표현 방식을 이용한 정보검색 관련연구가 여러 분야에서 진행되고 있다. 특히 XML 정보검색 관련연구 중 구조적 질의처리 및 질의확장은 데이터베이스 및 시맨틱(Semantic) 웹 기반 연구에서 중점적으로 다루어지고 있다. 본 장은 관련 내용으로 XML, RDF(Resource Description Framework), SDI(Selective Dissemination Information), 랩퍼(Wrapper), 메디에이터(Mediator), 온톨로지 등을 살펴보고, 이러한 배경연구를 바탕으로 본 연구의 필요성과 접근방법을 기술하고자 하였다.

XML은 태그에 의미정보를 부여함으로써 구조화된 문서를 표현할 수 있다. 그러나 XML 문서의 내용은 동일하지만 구조가 다양하게 표현될 수 있으므로 XML 문서를 일관된 프로그램으로 처리하기가 매우 어렵다. RDF는 이러한 XML의 문제점을 보완하고 시멘틱에 초점을 맞추기 위해 제시된 구조이다[6]. RDF는 XML 구조로 표현되나, 의미가 리소스와 특성 값으로 표현되어 같은 내용에 대한 해석이 하나로 귀결되므로 XML 문서에 비해 이해가 쉬우나, 태그 이름의 중첩성과 모호성 문제가 여전히 존재한다.

SDI는 최신의 정보를 정기적으로 수집하는 온라인 데이터베이스 서비스로서, 정보가 온라인 데이터베이스 시스템에 들어오는 즉시 자동으로 검색하여 배포해 주는 기능이다. XML 기반의 SDI는 중요한 데이터가 발생하면 이를 XML 구조로 변환하고, 변환된 XML 문서로부터 새로운 데이터 항목들을 지속적으로 수집한 후, 사용자 질의와 관련된 데이터를 검색하여 사용자에게 제공한다[7].

랩퍼는 각 로컬에 하나씩 존재하여 로컬 데이터베이스에 대한 내용을 미디에이터에게 전달함으로써 미디에이터가 통합 스키마를 구성할 수 있도록 정보를 제공하며, 미디에이터는 정보 통합 기능을 제공하는 핵심으로써 통합 스키마 관리 및 통합 질의처리 기능을 제공한다. XML을 통합 자료 모델로 사용하는 랩퍼 및 미디에이터에는 XML의 구조적인 한계를 극복할 수 있는 중재자가 필요하며, 로컬 질의처리 및 통합 검색결과 관리를 위한 연구가 중요하게 부각된다[8].

온톨로지는 단어와 관계들로 구성된 사전으로서, 특정 영역에 관련된 단어들에 대하여 상위개념과 하위개념, 그들 간의 관계를 계층구조로 표현하고 추가적으로 이를 확장할 수 있는 추론 규칙을 포함한다[9]. 온톨로지의 역할은 서로 다른 데이터베이스가 같은 개념에 대해서 서로 다른 단어나 식별자를 사용할 경우에 이를 동일 개념으로 처리하고, 웹 기반의 지식처리나 응용 프로그램 사이의 지식공유 및 재사용을 가능하게 한다[10]. 현재 응용영역에 대한 온톨로지 표준화 작업이 W3C 등에서 활발히 진행되고 있으므로, 가까운 미래에 문서의 다양한 구조에 기인한 문제를 극복할 수 있는 기반이 제공될 것이다.

XML 문서와 DTD를 구조적으로 파싱하여 질의

에 대한 키워드를 매칭 시키는 기존의 단순 질의검색 방법은 인가된 질의 항목이 검색 문서의 엘리먼트 및 속성과 일치할 때까지 대상 XML 문서나 DTD를 구조적으로 파싱하여 정보를 검색하므로, 인가된 질의 항목과 일치하는 단편적인 정보 이외에 의미있는 정보를 검색할 수 없다[3,4]. 또한 대상 문서에 대한 정보검색이 무작위로 수행되므로 질의에 대한 적중도가 낮으며, XML 태그표현에 대한 표준이 제공되지 않으므로 동일 정보를 검색하기 위해 의미가 같은 다수의 질의가 요구된다.

단순 질의검색 방법의 한계를 극복하기 위해 온톨로지를 기반으로 XML 질의를 구조적으로 확장하는 연구가 수행된 바 있다[5]. 이는 온톨로지를 이용하여 인가된 질의를 하위개념으로 확장하였다. 이러한 시도는 키워드 매칭에 의한 질의검색의 한계를 어느 정도 보완할 수 있지만 단순 구조적 질의확장에 불과하며, 문서 내에 숨겨진 의미있는 정보를 검색하기에는 불충분하다.

본 연구는 온톨로지에 의한 구조적인 질의확장을 포함하여 내부 연관관계에 의한 의미정보 검색이 가능하도록 하기 위하여 다음과 같은 세 가지 방법을 고려하였다. 첫째, 동일 의미의 질의에 대한 모호성과 다양한 XML 구조에 기인한 정보검색의 어려움을 극복하기 위해 온톨로지를 사용한다. 둘째, 검색에 적합한 XML 문서만을 사전에 선별하기 위해 포괄적 DTD를 생성하며, 포괄적 DTD를 이용하여 대상 XML 문서를 사전에 선별함으로써 정보검색 시간을 단축한다. 셋째, XML의 의미정보 검색을 위해 온톨로지로부터 개념구조 및 연관관계 규칙을 추론하여 질의를 의미적으로 확장한다.

3. 포괄적 DTD를 이용한 XML 문서여과

다수의 XML 문서를 대상으로 효과적인 정보검색을 수행하기 위해서는, 관심영역의 XML 문서들을 사전에 선별할 수 있어야 한다. 본 연구는 온톨로지로부터 동일 분야의 XML 문서에 포괄적으로 적용 가능한 DTD를 생성하였으며 이를 이용하여 불필요한 XML 문서를 사전에 여과하였다.

3.1 "대학연구센터" 온톨로지

온톨로지는 개념 구조 및 속성에 대한 연관관계로 표현된다[10]. 그림 2는 본 논문에서 문서여과 및 질의확장을 위해 설계한 "대학연구센터" 온톨로지로서 개념들의 계층구조, 개념의 속성, 개념 간 연관관계를 보여준다.

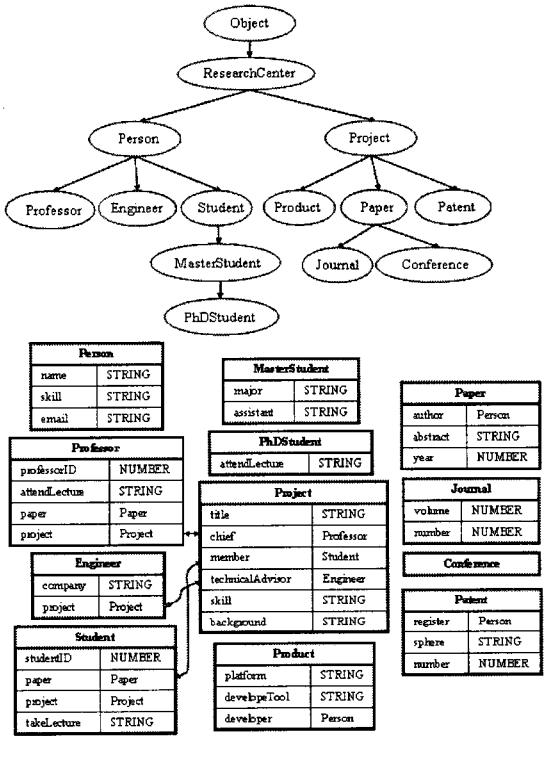


그림 2. "대학연구센터" 온톨로지

3.2 포괄적 DTD 생성 방법

포괄적 DTD 생성 방법은 그림 3과 같이 온톨로지 프로세서와 DTD 생성 프로세서로 구성된다. 온톨로지 프로세서는 온톨로지의 개념과 속성을 파싱하여 개념 클래스와 속성 클래스를 생성한다. 개념 클래스는 속성 클래스를 내포하며, 하나의 개념 클래스와 그 개념의 속성을 나타내는 클래스로 구성된다. DTD 생성 프로세서는 온톨로지 프로세서에서 생성된 개념 클래스와 속성 클래스를 바탕으로 ENTITY 생성 알고리즘, ELEMENT 생성 알고리즘, ATTLIST 생성 알고리즘, 추가 ELEMENT 생성 알고리즘을 적용하여 포괄적 DTD를 생성한다.

3.2.1 ENTITY 생성 알고리즘

온톨로지 개념들의 계층구조를 DTD에 표현하기

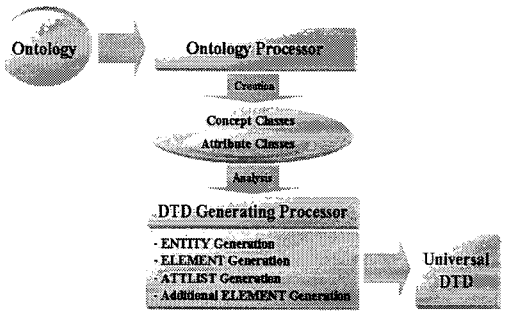


그림 3. 포괄적 DTD 생성 과정

위하여 하나의 개념을 선택하면, 선택한 개념에 대하여 ENTITY를 선언하고 자신의 하위개념들을 탐색하여 ENTITY의 원소로 추가한다. 알고리즘은 표 1과 같다.

표 1. ENTITY 생성 알고리즘

온톨로지의 모든 개념에 대하여,
1. 개념을 선택한다.
2. 선택한 개념을 ENTITY로 선언한다.
3. 선택한 개념의 하위개념들을 ENTITY의 원소로 추가한다.

3.2.2 ELEMENT 생성 알고리즘

온톨로지의 개념을 선택하여 ELEMENT로 선언하고 자신의 모든 상위개념으로부터 상속받은 속성들과 자신의 속성들을 ELEMENT의 원소로 추가한다. 알고리즘은 표 2와 같다.

표 2. ELEMENT 생성 알고리즘

온톨로지의 모든 개념에 대하여,
1. 개념을 선택한다.
2. 선택한 개념을 ELEMENT로 선언한다.
3. 선택한 개념의 모든 하위 개념을 검색하여 ELEMENT의 원소로 추가한다.
4. 선택된 개념의 모든 상위 개념에 대하여,
   ① 상위 개념을 순차적으로 선택한다.
   ② 상위 개념의 속성들을 검색하여 ELEMENT의 원소로 추가한다.
5. 선택한 개념 자신의 속성들을 ELEMENT의 원소로 추가한다.

3.2.3 ATTLIST 생성 알고리즘

온톨로지의 각 개념에 대한 속성은 ELEMENT와 함께 ATTLIST에도 정의된다. ATTLIST 생성 알고리즘 역시 개념의 속성을 정의하는 것이므로, ATTLIST 구문 표현법을 제외하면 표 2의 ELEMENT 생성 알고리즘과 유사하다. 알고리즘은 표 3과 같다.

표 3. ATTLIST 생성 알고리즘

온톨로지의 모든 개념에 대하여,
1. 개념을 선택한다.
2. 선택한 개념을 ATTLIST에 선언한다.
3. 선택한 개념의 모든 상위 개념에 대하여,
   ① 상위 개념을 순차적으로 선택한다.
   ② 선택된 상위개념의 속성명, 속성의 CDATA 타입, #IMPLIED 옵션 등을 ATTLIST의 항목으로 추가한다.
4. 선택한 개념 자신의 속성명, 속성의 CDATA 타입, #IMPLIED 옵션 등을 ATTLIST의 항목으로 추가한다.

3.2.4 추가 ELEMENT 생성 알고리즘

추가 ELEMENT 생성 알고리즘은 개념의 각 속성에 대한 ELEMENT를 정의한다. 개념의 속성 값은 기본적으로 STRING으로 정의되지만 다른 개념으로 정의될 수도 있다. A라는 개념의 속성 값이 개념 B로 정의되었다면, 개념 A의 ELEMENT에 매개변수 ENTITY 형태로 개념 B를 추가한다. 알고리즘은 표 4와 같다.

표 4. 추가 ELEMENT 생성 알고리즘

온톨로지의 모든 개념에 대하여,
1. 개념을 순차적으로 선택한다.
2. 선택한 개념의 모든 속성에 대하여,
   ① 속성을 순차적으로 선택한다.
   ② 선택한 속성을 추가 ELEMENT로 선언한다.
      i 만약, 선택한 속성의 값이 STRING 또는 NUMBER이면, 추가 ELEMENT의 원소로 #PCDATA를 추가한다.
      ii 만약, 속성 값이 다른 개념이면, 추가 ELEMENT의 원소로 #PCDATA와 다른 개념을 추가한다.

3.3 포괄적 DTD 예

포괄적 DTD는 앞서 설명한 알고리즘에 의해 온톨로지로부터 생성된다. 표 5는 "대학연구센터" 온톨로지로부터 생성된 포괄적 DTD의 일부이며, 동일 영역의 정보를 적절히 내포하고 있는 모든 문서들을 수용하게 된다.

표 5. "대학연구센터" 포괄적 DTD의 일부분

```
            .....
<!ENTITY % Person "Person|Student|MasterStudent|PhDStudent|En
gineer|Professor">
<!ENTITY % Student "Student|MasterStudent|PhDStudent">
            .....
<!ELEMENT Person (#PCDATA|Student|MasterStudent|PhDStudent
|Engineer|Professor|name|email|skill)* >
<!ELEMENT Student (#PCDATA|MasterStudent|PhDStudent|name|
email|skill|studentID|takeLecture|paper|project)* >
            .....
<!ATTLIST Person
            name      CDATA      #IMPLIED
            email     CDATA      #IMPLIED
            .....
<!ELEMENT attendLecture (#PCDATA) >
<!ELEMENT project (#PCDATA|%Project;)* >
            .....
```

4. 온톨로지 기반 XML 질의확장

질의확장 알고리즘은 XML 문서로부터 의미적으로 중요한 정보를 추론하기 위해 온톨로지의 계층구조와 연관관계를 이용하여 질의를 확장한다. 개념 계층구조는 온톨로지에 명시적으로 나타나 있지만, 연관관계는 묵시적으로 표현되어 있다. 그림 4는 온톨로지와 개념간의 묵시적 속성 관계를 분석하여 추론된 규칙을 바탕으로 XML 질의를 확장하는 과정을 나타낸 것이다.

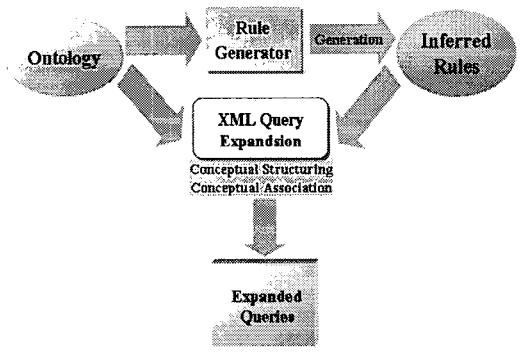


그림 4. 온톨로지를 이용한 XML 질의확장

4.1 개념구조에 의한 질의확장

온톨로지의 개념 계층구조를 이용하여 질의를 확장할 수 있다. 개념구조에 의한 질의확장 알고리즘은 표 6과 같다.

표 6. 개념 계층구조에 의한 질의확장 알고리즘

입력된 질의에 포함된 모든 개념에 대하여,
1. 하나의 개념을 선택한다.
2. 선택한 개념의 하위개념을 검색한다.
3. 만약 선택한 개념의 모든 하위개념들을 검색하여 질의에 하위개념을 추가한다.

예를 들어, "C"라는 "skill"을 보유한 사람을 검색할 경우, //Person[skill="C"]라는 질의를 하게 된다. 이러한 질의는 표 6의 질의확장 알고리즘에 의해 온톨로지의 계층구조를 바탕으로 "Person"의 하위개념으로 정의된 "Professor, Engineer, Student, MasterStudent, PhDStudent"로 확장된다. 구조적으로 확장된 질의는 엘리먼트 "Person"뿐 만 아니라, 확장된 하위개념을 포함하여 검색을 수행한다.

4.2 연관관계 추론에 의한 질의확장

온톨로지는 개념 계층구조, 속성 상속, 개념 간 연관관계에 의해 의미적으로 중요한 정보를 내포할 수 있다. 온톨로지의 연관관계는 표 7의 알고리즘에 의해 규칙으로 생성될 수 있으며, 추론된 규칙은 XML 문서에 내재한 정보검색에 유용하게 사용된다.

표 7. 규칙생성 알고리즘

온톨로지의 모든 개념에 대하여,
1. 개념 A를 선택한다.
2. 만약 개념 A의 속성 값에 개념 B가 존재하면,
   ① 온톨로지에서 개념 B를 검색한다.
   ② 개념 B의 속성 값에 개념 A가 존재하면,
   ③ 개념 A, B 사이의 규칙을 생성한다.

"대학연구센터" 온톨로지에 내포된 모든 연관관계를 규칙의 형태로 나타내면 표 8과 같다. 표에서 밑줄 된 규칙은 개념 "Project"의 "member"와 개념 "Student"의 "project" 사이에 존재하는 연관관계를 규칙으로 나타낸 것이다. 추론된 규칙들은 질의확장에 항상 참조된다.

예를 들어 "Project"에 참여하는 "member"를 검색할 경우, //Project[member]라는 단순질의가 인가된다. 그러나 온톨로지로부터 추론된 규칙에 의해 "Project"의 속성 "member"가 "Student"와 연관관계가 있음을 알 수 있다. 따라서 인가된 질의는

표 8. 온톨로지로부터 추론된 규칙

```
FORALL Stud1, Proj1
    Proj1 : Project[member ->> Stud1]
            <-> Stud1 : Student[project ->> Proj1].
FORALL Prof1, Proj1
    Proj1 : Project[chief ->> Prof1]
            <-> Prof1 : Professor[project ->> Proj1].
FORALL Eng1, Proj1
    Proj1 : Project[technicalAdvisor ->> Eng1]
            <-> Eng1 : Engineer[project ->> Proj1].
```

//(Project|Product|Patent|Journal|Conference) [member]와 //(Student|MasterStudent|PhDStudent) [project]로 확장된다. 결과적으로 "Project"라는 개념만으로는 검색할 수 없는 정보가 추론된 규칙에 의해 개념 "Student"와 이의 하위 개념에서 검색될 수 있다. 표 9는 연관관계 추론규칙을 이용한 질의확장 알고리즘이다.

표 9. 연관관계 추론에 의한 질의확장 알고리즘

1. 입력된 질의에 포함된 모든 개념에 대하여,
   ① 개념을 선택한다.
   ② 선택한 개념에 대하여 추론된 규칙이 존재하는지 검사한다.
   ③ 만약 규칙이 존재하면,
      i 규칙에서 연관되어 있는 다른 개념을 질의에 추가하고 다른 개념에 대해서 존재하는 규칙이 있는지 검사한다.
      ii 다른 개념에 대하여 규칙이 존재하면 단계 i 을 반복한다.
2. 수정된 질의에 대하여, 표 6의 구조적 질의확장 알고리즘을 적용하여 질의를 확장한다.

온톨로지로부터 추론된 연관관계 규칙들은 동일 영역 정보검색에 반복적으로 재사용이 가능하므로 전체적인 검색 시간을 단축시킬 수 있다.

5. XML 문서여과 및 질의확장 실험

3장과 4장에서 제안한 XML 문서여과와 질의확장의 효과를 다양한 구조의 XML 예제 문서들에 적용하였다.

5.1 포괄적 DTD에 의한 문서여과 실험

그림 2의 온톨로지로부터 생성된 포괄적 DTD가 "대학연구센터" 영역에 포함되어 있지 않는 불필요한 문서들을 여과할 수 있는지를 예제 XML 문서를 사용하여 실험하였다.

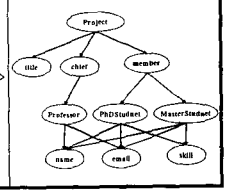
표 10의 XML 문서 example01.xml은 엘리먼트 "Project"의 "title, chief, member" 등으로 "대학연구센터"의 개념 "Project"와 관련된 내용으로 구성되며, 표 11과 같은 DTD 구조를 갖는다. 포괄적 DTD는 example01.dtd의 모든 엘리먼트들을 포함하므로 그림 5와 같이 example01.xml 문서를 검색 대상 XML 문서로 선별한다.

표 10. example01.xml

```
<Project>
  <title>Laser Slice</title>
  <chief>
    <Professor>
      <name>Y.H.Kong</name>
      <email>yhkong@sch.ac.kr</email>
    </Professor>
  </chief>
  <member>
    <PhDStudent>
      <name>M.S.Kim</name>
      <email>mskim@sch.ac.kr</email>
      <skill>C++</skill>
    </PhDStudent>
    <MasterStudent>
      <name>H.J.Lee</name>
      <skill>C++</skill>
    </MasterStudent>
  </member>
  <title>Semantic XML Query</title>
  <chief>
    <Professor>
      <name>Y.H.Kong</name>
      <email>yhkong@sch.ac.kr</email>
    </Professor>
  </chief>
  <member>
    <MasterStudent>
      <name>K.S.Lee</name>
      <email>kslee@sch.ac.kr</email>
      <skill>XML</skill>
      <skill>Java</skill>
    </MasterStudent>
  </member>
</Project>
```

표 11. example01.dtd와 구조

```
<!ELEMENT Project(title|chief|member)*>
<!ELEMENT Professor(name|email)*>
<!ELEMENT title(#PCDATA)>
<!ELEMENT chief(#PCDATA|Professor)*>
<!ELEMENT member(PhDStudent|MasterStudent)*>
<!ELEMENT PhDStudent(name|email|skill)*>
<!ELEMENT name (#PCDATA)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT skill (#PCDATA)>
<!ELEMENT MasterStudent(name|email|skill)*>
```



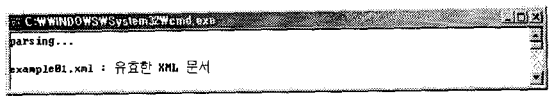


그림 5. 포괄적 DTD에 의해 선별된 example01.xml

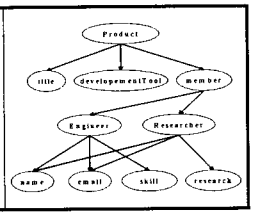
표 12의 예제 XML 문서 example02.xml은 "Product" 엘리먼트와 "title, developmentTool, member" 등의 하위 엘리먼트로 구성되며, 표 13과 같은 DTD 구조를 갖는다. Example02.xml 문서는 표 10의 example01.xml 문서와 비슷한 구조로 보이나, "Product"의 하위 요소들이 "대학연구센터" 영역에 정의된 "Product" 개념의 요소와 일치하지 않으므로 그림 6과 같이 검색에 불필요한 문서로 여과된다.

표 12. example02.xml

```
<Product>
  <title>DSP Board Tester</title>
  <developementTool>Matlab
  </developementTool>
  <member> <Engineer>
        <name>B.I.Jin</name>
        <email>bij@mail.test</email>
        <skill>DSP</skill>
      </Engineer>
      <Researcher>
        <name>T.H.Lee</name>
        <email>thl@mail.test</email>
        <research>Logic Circuit</research>
      </Researcher></member>
  <title>Garbage Process Eraser</title>
  <developementTool>Java</developementTool>
  <member> <Researcher>
        <name>K.S.Lee</name>
        <email>ksl@mail.test</email>
        <research>Operating System
        </research> </Researcher>
      <Engineer>
        <name>C.W.Lee</name>
        <email>cwl@mail.test</email>
        <skill>Unix Programming</skill>
      </Engineer>
  </member>
</Product>
```

표 13. example02.dtd와 구조

```
<!ELEMENT Product (title|developementTool|member)+>
<!ELEMENT title (#PCDATA)>
<!ELEMENT developementTool (#PCDATA)>
<!ELEMENT member (Engineer|Researcher)+>
<!ELEMENT Engineer (name|email|skill)+>
<!ELEMENT Researcher (name|email| research)+>
<!ELEMENT name (#PCDATA)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT skill (#PCDATA)>
<!ELEMENT research (#PCDATA)>
```



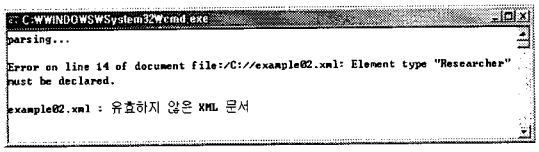


그림 6. 포괄적 DTD에 의해 제거된 example02.xml

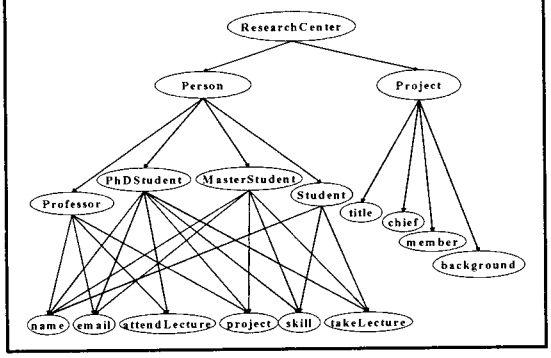
위의 두 실험과 동일한 방법으로 표 14의 XML 문서에 대한 문서여과를 실험하였다. 표 14(a), (b)의 XML 문서는 "대학연구센터" 온톨로지와 비교해 볼 때 매우 상이한 구조를 갖는다. 표 14(a)의 example03.xml은 "대학연구센터"의 개념 "Person"과 "Project"에 관한 내용으로 구성되어 있으며, 표 14(b)의 example04.xml은 개념 "Person"과 하위개념들로 구성되어 있다. 이 두 문서들은 온톨로지에 비해 복잡하거나 표면적으로 상이하게 보일 수 있으나 모두 "대학연구센터" 온톨로지에 포함되는 문서들로서 검색 대상으로 선택되었다. 반면에 표 14(c)의 example05.xml은 "대학연구센터" 온톨로지의 개념 "Person"에 포함되어 있는 "Professor, Student, MasterStudent, PhDStudent" 개념과 동일하다. 그러나 example05.xml의 엘리먼트의 속성은 "대학연구센터"에 무관하므로 여과되어 검색 대상에서 제거되었다.

예제 XML 문서들을 대상으로 포괄적 DTD로 여과한 결과, example02.xml과 example05.xml은 검색

표 14. 문서여과에 사용된 예제 XML 내용 및 구조

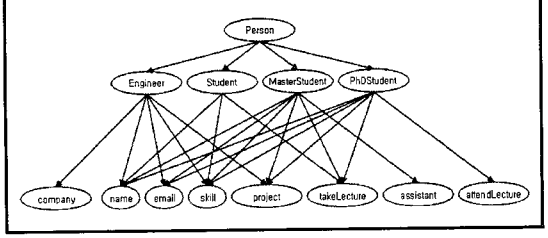
(a) example03.xml

```
<ResearchCenter>
 <Person>
  <Professor>
   <name>Y.H.Kong</name>
   <email>yhkong@sch.ac.kr</email>
   <attendLecture>Artificial intelligence
   </attendLecture>
   <project>Laser Slice</project>
   <project>Semantic XML Query </project>
  </Professor>
  <PhDStudent>
   <name>M.S.Kim</name>
   <skill>C++</skill>
   <takeLecture>Artificial Intelligence
   </takeLecture>
   <attendLecture>Assembly</attendLecture>
   <project>Laser Slice</project>
  </PhDStudent>
<Student>
    <name>J.K.Shin</name>
    <skill>Visual Basic</skill>
    <takeLecture>C</takeLecture>
   </Student>
  </Person>
<Project>
   <title>Semantic XML Query</title>
   <chief>Y.H.Kong</chief>
   <member>K.S.Lee</member>
   <skill>Java</skill>
   <background>XML</background>
  </Project>
  <Project>
   <title>Data Structure</title>
   <chief>Y.H.Kong</chief>
   <member>K.S.Lee</member>
   <member>J.K.Shin</member>
  </Project>
</ResearchCenter>
```

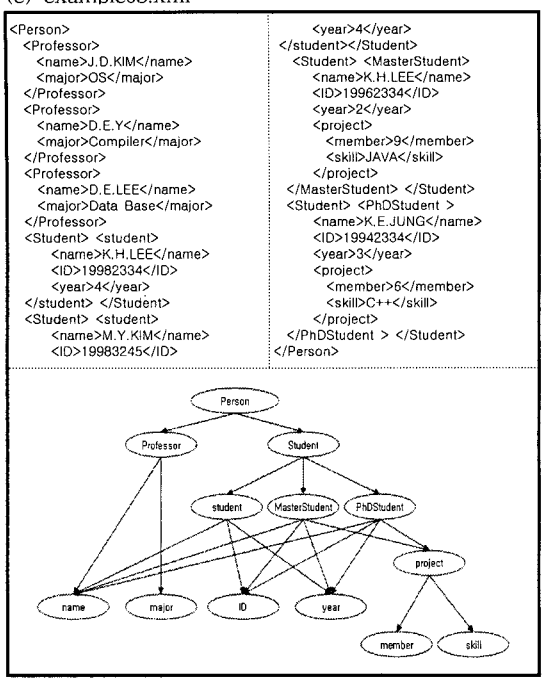


(b) example04.xml

```
<Person>
  <Engineer>
    <name>S.H.Yun</name>
    <email>shyun@sch.ac.kr</email>
    <company>AI.LAB</company>
    <skill>C</skill>
    <skill>Java</skill>
    <project>Semantic XML Query</project>
  </Engineer>
  <Engineer>
    <name>H.D.Kwon</name>
    <email>hdkwon@sch.ac.kr</email>
    <company>AI.LAB</company>
    <skill>C++</skill>
    <skill>C#</skill>
    <skill>Java</skill>
    <project>Semantic XML Query</project>
  </Engineer>
  <Student>
    <name>J.K.Shin</name>
    <skill>Visual Basic</skill>
    <takeLecture>C</takeLecture>
  </Student>
  <Student>
    <name>S.G.Min</name>
    <skill>C</skill>
    <takeLecture>Visual Basic</takeLecture>
  </Student>
  <MasterStudent>
    <name>K.S.Lee</name>
    <email>kslee@sch.ac.kr</email>
    <skill>C</skill>
    <skill>Java</skill>
    <takeLecture>Artificia Intelligebce
    </takeLecture>
    <project>Semantic XML Query</project>
    <assistant>S.H.Yun</assistant>
    <assistant>H.D.Kwon</assistant>
  </MasterStudent>
  <PhDStudent>
    <name>M.S.Kim</name>
    <email>mskim@sch.ac.kr</email>
    <skill>C++</skill>
    <skill>Open GL</skill>
    <takeLecture>Artificial Intelligence
    </takeLecture>
    <attendLecture>C</attendLecture>
    <project>Laser Slice</project>
  </PhDStudent>
</Person>
```



(c) example05.xml



대상에서 제외되고 나머지 3개의 문서는 검색에 적합한 문서로 선별되었다. 즉, 포괄적 DTD에 의해 부적합한 문서를 사전에 여과함으로써 불필요한 검색을 줄일 수 있다.

5.2 질의확장에 의한 정보검색 실험

5.1절의 문서여과 실험에서 선택된 세 개의 문서들을 대상으로, 질의확장에 의한 정보검색의 효과를 실험하였다. 질의확장 실험은 개념구조에 의한 질의확장과 연관관계에 의한 질의확장으로 분류하여 실험하였다.

5.2.1 개념구조에 의한 질의확장 실험

Example01.xml과 example03.xml 문서에서 속성 "skill"이 "C"인 사람을 검색할 경우, //Person [skill= "C"]라는 단순질의가 인가된다. 이를 온톨로지의 개념구조에 의해 확장하면 "Person"의 하위개념인 "Professor, Engineer, Student" 및 그 하위개념인 "MasterStudent, PhDStudent"로 질의가 확장된다. 예제 example01.xml과 example03.xml 문서에 확장된 질의를 인가하면, 구조적으로 확장된 //MasterStudent[skill= "C"] 또는 //PhDStudent[skill="C"]에 의하여 단순질의로는 검색될 수 없었던 정보가 그림 7, 8과 같이 검색되었다.

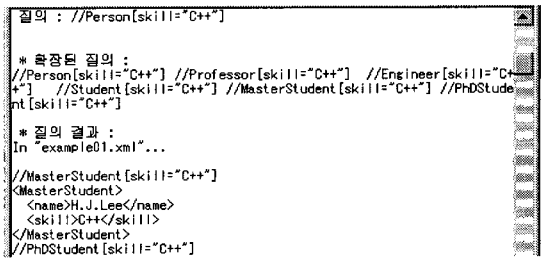


그림 7. 계층구조에 의한 example01.xml 질의 결과

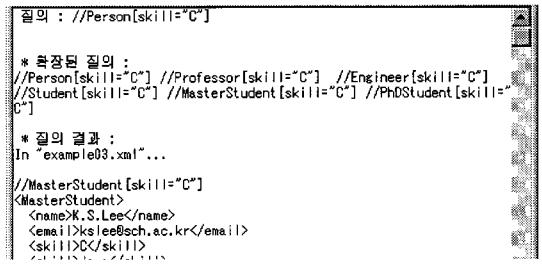


그림 8. 계층구조에 의한 example03.xml 질의 결과

그림 9는 example04.xml에 대하여 "skill"이 "Java"인 학생을 검색하는 //Student[skill="Java"]라는 질의는 온톨로지의 개념 계층구조에 의해 //(Student|MasterStudent|PhDStudent)[skill="Java"]로 확장되어 하위개념을 포함하는 폭넓은 정보검색을 가능하게 하였다.

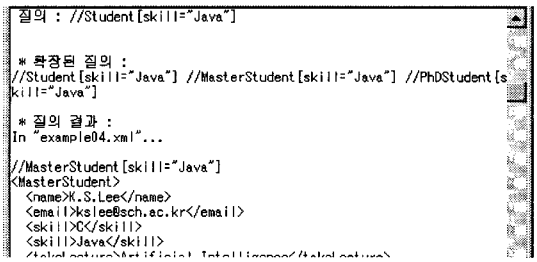


그림 9. 계층구조에 의한 example04.xml 질의 결과

5.2.2 연관관계에 의한 질의확장 실험

온톨로지에 의한 구조적 질의확장만으로는 의미적으로 중요한 정보를 검색하는데 한계가 있다. 따라서 온톨로지에 내포된 연관관계를 포함하는 질의의 확장이 의미정보 검색에 필요하다. 연관관계에 의한 질의확장은 표 8과 같이 "대학연구센터" 온톨로지로부터 추론된 규칙을 사용하며, 예제 XML 문서에 대

하여 입력된 단순질의는 추론규칙에 의해 표 15와 같이 연관된 개념에 대한 질의로 확장되었다.

표 15. 질의확장에 사용된 추론규칙

검색 대상 XML 문서	단순질의	추론규칙에 의한 질의
(a)example01.xml	//Professor[project]	//Project[chief]
(b)example03.xml	//Project[member]	//Student[project]
(c)example04.xml	//Project[technicalAdvisor]	//Engineer[project]

예를 들어, 표 14(a)의 example03.xml에서 "Project"에 참여하는 "member"를 검색하면, "Project" 뿐 아니라 "Project"와 연관된 "Student"에 대하여 질의가 확장되고, 또한 "Student"의 하위개념인 "MasterStudent, PhDStudent"에 대해서도 질의가 확장된다. 이와 같이 문서 내에는 존재하지만 //Project[member]라는 질의로는 검색할 수 없었던 정보가, 온톨로지의 연관관계에 의해 확장된 //Student [project] 질의로 검색되었다. 그림 10, 11 그리고 12는 이러한 연관관계에 의한 질의확장 결과이다.

```
질의 : //Professor[project]

 * 확장된 질의 :
//Professor[project] //Project[chief] //Patent[chief] //Paper[chief
]   //Conference[chief]     //Journal[chief]   //Product[chief]

 * 질의 결과 :
In "example01.xml"...

//Project[chief]
<Project>
  <title>Laser Slice</title>
  <chief>
    <Professor>
      <name>Y.H.Kong</name>
      <email>yhkong@sch.ac.kr</email>
    </Professor>
  </chief>
  <chief>
```

그림 10. 연관관계에 의한 example01.xml 질의 결과

```
질의 : //Project[member]

 * 확장된 질의 :
//Project[member] //Patent[member]  //Paper[member]   //Conference[m
ember]     //Journal[member]   //Product[member]  //Student[project]
 //MasterStudent[project] //PhDStudent[project]

 * 질의 결과 :
In "example03.xml"...

//Project[member]
<Project>
  <title>Semantic XML Query</title>
  <chief>Y.H.Kong</chief>
  <member>K.S.Lee</member>
  <skill>Java</skill>
  <background>XML</background>
</Project>
<Project>
  <title>Data Structure</title>
  <chief>Y.H.Kong</chief>
  <member>K.S.Lee</member>
```

그림 11. 연관관계에 의한 example03.xml 질의 결과

```
질의 : //Project[technicalAdvisor]

 * 확장된 질의 :
//Project[technicalAdvisor] //Patent[technicalAdvisor]  //Paper[tech
nicalAdvisor]   //Conference[technicalAdvisor]     //Journal[technic
alAdvisor]   //Product[technicalAdvisor]  //Engineer[project]

 * 질의 결과 :
In "example04.xml"...

//Engineer[project]
<Engineer>
  <name>S.H.Yun</name>
  <email>shyun@sch.ac.kr</email>
  <company>AI.LAB</company>
  <skill>C</skill>
  <skill>Java</skill>
  <project>Semantic XML Query</project>
```

그림 12. 연관관계에 의한 example04.xml 질의 결과

질의확장 실험 결과, 온톨로지 기반의 구조적 질의확장과 온톨로지에 내포되어 있는 개념 및 속성 간의 연관관계를 추론하는 질의확장이 XML 문서내의 중요한 의미적 정보검색을 가능하게 하였다.

6. 결 론

XML은 매우 다양한 구조와 형식으로 표현될 수 있기 때문에, 모든 XML 문서를 대상으로 질의를 인가하게 되면 불필요한 문서에 의한 검색의 비효율이 발생하며, XML 질의에 의한 단순 키워드 정합이나 구조적 질의확장만으로는 의미적으로 중요한 정보를 검색하는데 불충분하다. 본 논문은 XML 문서에 내재된 의미정보를 효율적으로 검색하기 위해 온톨로지를 기반으로 문서여과 및 질의확장 방법을 제안하였다.

문서여과 방법은 온톨로지로부터 구조가 다양한 XML 문서에 적용 가능한 포괄적 DTD를 생성하여 검색에 적합한 문서를 사전에 선별하였다. 질의확장 방법은 온톨로지의 개념구조 및 연관관계로부터 규칙을 추론하였으며, 추론된 규칙을 바탕으로 질의를 확장하여 XML 문서에 내재된 정보를 검색하였다. 문서여과 실험에서 영역에 포함되지 않는 다른 영역의 문서들이 모두 여과되었으며, 질의확장 실험에서 단순질의가 온톨로지의 계층구조 및 연관관계에 의해 확장되었다.

결과적으로, 온톨로지 기반의 문서여과 방법은 정보검색 대상 XML 문서를 사전에 선별함으로써 검색에 불필요한 문서를 효과적으로 제거하였으며, 질의확장 방법은 인가된 질의를 확장함으로써 단순질의로는 검색이 불가능하였던 XML 문서에 내재된 중요한 의미적 정보검색이 가능할 수 있다.

참 고 문 헌

[1] 최중민, "시맨틱 웹의 개요와 연구동향," *정보과학회지*, 제21권, 제3호, pp.4-10, 2003.

[2] D. L. McGuinness and F. V. Harmelen(eds.), "OWL Web Ontology Language Overview," *W3C Recommendation*, 10, February, 2004.

[3] Y. D. Chung, J. W. Kim, and M. H. Kim, "Efficient preprocessing of XML queries using structured signatures," *Information Processing Letters 87*, pp.257-264, 2003.

[4] C. Y. Chan, P. Felber, M. Garofalakis, and R. Rastogi, "Efficient Filtering of XML documents with XPath expression," *The VLDB Journal*, pp.354-379, 2002.

[5] M. Erdmann and R. Studer, "How to Structure and Access XML Document with Ontologies," *Data & Knowledge Engineering*, Vol.36, No.3, pp.317-335, 2001.

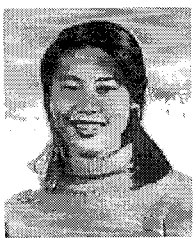
[6] F. Manola and M. Miller(eds.), RDF Primer, *W3C Recommendation*, 2004.

[7] Y. Diao, M. Altinel, M. J. Franklin, H. Zhang, and P. Fischer, "Path Sharing and Predicate Evaluation for High-Performance XML Filtering," *ACM Transactions on Database Systems (TODS)*, Vol.28, Issue4, pp.467-516, 2003.

[8] C. Baru, A. Gupta, B. Ludascher, R. Marciano, Y. Papakonstantinou, and P. Velikhov, "XML-Based Information Mediation with MIX," *Demonstrations Program of ACM SIGMOD Conf.*, 1999.

[9] M. Erdmann and R. Studer, "Ontologies as Conceptual Models for XML Document," *KAW '99-Proceedings of the 12th Workshop on Knowledge Acquisition*, Modeling and Management, 1999.

[10] T. R. Gruber, "A Translation Approach to Portable Ontologies Specifications," *Knowledge Acquisition*, Vol.5, No.2, pp.199-220, 1993.



김 명 숙

1995년 순천향대학교 전산학과 학사
1999년 순천향대학교 전산학과 석사
2002년~현재 순천향대학교 전산학과 박사과정

관심분야 : 영상처리, 신경회로망, 웹응용

공 용 해

1982년 연세대학교 전자공학과 학사
1986년 미국 Polytechnic Univ. 전산학과 석사
1991년 미국 Polytechnic Univ. 전산학과 박사
1982년 한진중공업 연구원
1983년 삼성전자 연구원
1991년~현재 순천향대학교 정보기술공학부 교수
관심분야 : 지능에이전트, 신경회로망, 멀티미디어응용