DTD 여과 및 질의 확장에 의한 효율적인 XML 문서의 정보 검색

김명숙, 이경수, 공용해
순천향대학교 정보기술공학부
e-mail:krhkms@sch.ac.kr

Efficient XML Information Search through DTD Filtering and Query Expansion

Myoung Sook Kim, Kyeung Soo Lee, Yong Hae Kong
Division of Information Technology Engineering
Soonchunhyang University

요 약

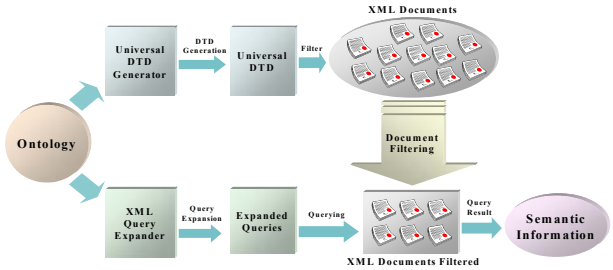
본 논문은 정보검색의 대상이 되는 XML 문서를 효율적으로 선별하기 위해 온톨로지를 기반으로 XML 문서를 여과하였으며, 여과된 XML 문서를 대상으로 문서에 내재한 정보를 효과적으로 검색하도록 XML 질의를 확장하였다. 이를 위해, 온톨로지로부터 포괄적 DTD를 생성하는 알고리즘을 개발하였고, XML 문서의 효과적인 정보 검색을 위해 온톨로지의 개념 구조와 연관 관계를 분석하여 XML 질의를 확장하는 알고리즘을 개발하였다. 제안한 문서 여과와 질의 확장 알고리즘의 효과를 샘플 XML 문서에 적용하였다.

1. 서론

동일한 구조의 XML 문서라 할지라도 다양한 구조로 표현될 수 있으므로, 내재된 정보에 대한 XML 질의는 크게 제한된다. 이러한 문제점을 온톨로지의 관점에서 해결하고자 하는 연구가 수행되고 있으며, 온톨로지를 Web-Ontology Working Group 등에서 획득하여 연구개발에 사용할 수 있다[1],[2].

본 논문은 온톨로지의 개념과 속성을 반영하는 알고리즘을 적용하여 DTD를 생성하였고, 온톨로지의 개념 구조와 연관 관계를 고려하여 질의를 확장하는 효과적인 XML 정보검색 알고리즘을 제안하였다. 우선, 온톨로지로부터 정보 검색 대상인 XML 문서를 선별하기 위해 포괄적 DTD 생성기를 구현하였으며, 포괄적 DTD에 의해 XML 문서를 사전에 여과하였다. 또한 여과된 XML 문서에 내재된 정보 검색을 위해 온톨로지의 개념 구조와 연관 관계를 고려하여 규칙 추출기를 구현하였고, 개념 계층 구조와 추론된 규칙을 기반으로 XML 질의를 확장하는 질의 확장기를 구현하였다. 그림 1에 DTD 여과 및 질의 확장 시스템을 나타내었다.

본 논문에서 제안된 알고리즘을 적용하여 포괄적 DTD에 의한 XML 문서 여과와 확장 질의에 의한 XML 문서의 의미 정보 검색이 효과적으로 수행됨을 보였다.



(그림 3) DTD 여과 및 질의확장 시스템

2. 포괄적 DTD에 의한 XML 문서 여과

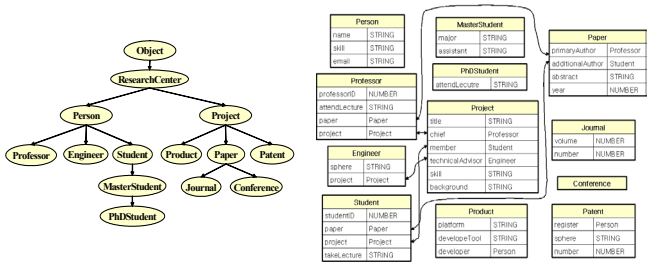
XML 정보 검색은 특정 영역의 XML 문서만을 대상으로 하기 때문에 해당 XML 문서를 선별하고 불필요한 문서는 여과해야 한다. 본 논문에서는 정보 검색의 대상이 되는 XML 문서를 선별하기 위해 온톨로지의 개념과 속성을 분석하여 생성된 포괄적 DTD로 XML 문서를 여과하였다.

2.1 온톨로지

온톨로지는 프로그램과 인간이 정보를 공유하기 위해 사용되는 개념화된 명세서이며, 이러한 온톨로지는 개념의 계층구조와 속성의 연관관계를 표현하는 부분으로 분류된다[3],[4]. 본 논문에서는 대학 연

본 논문은 정보통신부 정보통신연구진흥원에서 지원하고 있는 정보통신기초기술연구지원사업의 연구결과입니다.

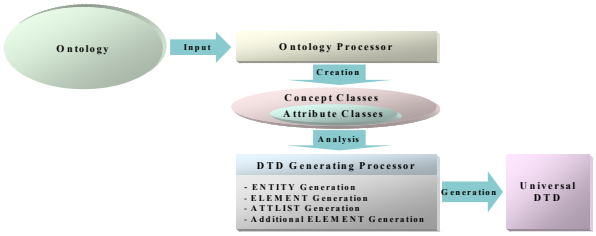
구센터 영역을 대상으로 하는 온톨로지의 개념 계층 구조와 속성의 연관관계를 그림 2에 보였다.



(그림2) 대학 연구센터 온톨로지

2.2 포괄적 DTD 생성

포괄적 DTD 생성기는 그림 3과 같이 온톨로지 프로세서와 DTD 생성 프로세서로 구성된다. 온톨로지 프로세서는 온톨로지의 개념과 속성을 파싱하여 개념 및 속성 클래스를 생성하고, DTD 생성 프로세서는 생성된 개념 및 속성 클래스에 표 1~4의 ENTITY 생성, ELEMENT 생성, ATTLIST 생성, 추가 ELEMENT 생성 알고리즘을 각각 적용하여 포괄적 DTD를 생성한다.



(그림 3) 포괄적 DTD 생성기의 구성

<표 1> ENTITY 생성 알고리즘

```
Until concept list is empty, Select a concept of concept list
    Declare ENTITY for the selected concept
    Until concept list is empty(from selected concept)
    Search a sub-concept of concept list
        If searched concept is sub-concept of the selected concept
            Add the sub-concept at ENTITY
```

<표 2> ELEMENT 생성 알고리즘

```
Until concept list is empty, Select a concept of concept list
    Declare ELEMENT for the selected concept
    Until concept list is empty(from the selected concept)
      Search a sub-concept of concept list
        If searched concept is sub-concept of the selected concept
            Add the sub-concept at ENTITY
      Search a super-concept of concept list
        If searched concept is super-concept of the selected concept
            Add the attributes of super-concept at ELEMENT
    Add the attributes of selected concept ELEMENT
```

<표 3> ATTLIST 생성 알고리즘

```
Until concept list is empty, Select a concept of concept list
    Declare ATTLIST for the selected concept
    Until concept list is empty Search a super-concept of concept list
        If searched concept is super-concept of the selected concept
            Add the attributes of super-concept at ATTLIST
    Add the attributes of selected concept ATTLIST
```

<표 4> 추가 ELEMENT 생성 알고리즘

```
Until concept list is empty, Select a concept of concept list
   Until attribute list of the selected concept is empty,
      Declare ELEMENT for attribute of the selected concept
      If type of attribute is STRING or NUMBER
          Define data type of ELEMENT as #PCDATA
      Else (if type of attribute is other concept)
          Add the concept at data type of ELEMENT
```

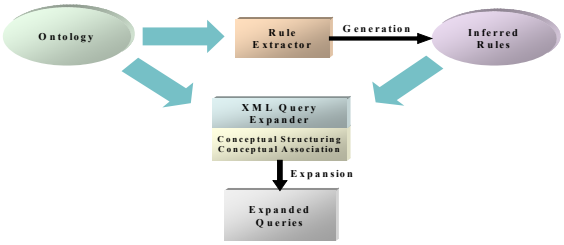
포괄적 DTD는 특정 영역의 온톨로지를 반영하는 DTD를 생성하며, 생성된 DTD는 해당 영역에 적합한 XML 문서를 여과할 수 있다. 표 5는 대학 연구센터 영역에 대하여 생성된 포괄적 DTD의 예이다.

<표 5> DTD 생성기에 의한 포괄적 DTD의 일부분

```
<!ENTITY%Person"Person|Student|MasterStudent|PhDStudent|Engineer|Professor">
<!ENTITY%Student"Student|MasterStudent| PhDStudent">
.....
<!ELEMENT Student (#PCDATA|MasterStudent|PhDStudent|name|email|skill|studentID|takeLecture|
paper|project)*>
<!ELEMENT MasterStudent (#PCDATA|PhDStudent|name|email|skill|student|takeLecture|paper|
project| major|assistant)*>
.....
<!ATTLIST Student
name CDATA #IMPLIED
email CDATA #IMPLIED
.....
<!ELEMENT name (#PCDATA) >
<!ELEMENT email (#PCDATA) >
.....
```

3. 온톨로지를 이용한 XML 질의 확장

XML 문서의 정보를 검색하기 위해 온톨로지의 개념 계층구조와 연관관계를 이용하여 질의를 확장하였다. 개념 계층 구조는 온톨로지에 표면적으로 나타나 있지만, 연관관계는 묵시적으로 표현되어 있다. 묵시적인 연관관계를 추출하기 위해 연관 관계를 추론하여 규칙의 형태로 추출하는 규칙 추출기를 구현하였으며, 개념 계층 구조와 추론된 규칙을 바탕으로 질의 확장을 수행한다. 그림 4에 온톨로지를 이용한 XML 질의 확장 과정을 보였다.



(그림 4) 온톨로지를 이용한 XML 질의 확장

3.1 개념 계층 구조에 의한 질의 확장

온톨로지는 특정 영역에 대한 개념과 속성을 정의한 것으로 개념간에는 계층 구조가 존재하며 이 계층 구조를 이용하여 질의를 확장하였다. 계층 구조에 의한 질의 확장 알고리즘은 표 6과 같다.

<표 6> 계층 구조에 의한 질의 확장 알고리즘

```
Analyze XML query, Select a adapted concept for inputted query Until concept list is empty
    Search sub-concepts of the adapted concept
    If searched concept is sub-concept of the adapted concept
        ADD the concept at XML query
```

3.2 연관 관계에 의한 질의 확장

개념 계층 구조 사이의 상속에서 추론될 수 있는 연관관계에 의해 온톨로지는 의미 정보를 내포하게 된다. 이러한 온톨로지의 연관 관계를 이용하여 XML 질의를 확장하면 XML 문서의 의미 정보 검색이 가능하다. 온톨로지에서 개념간의 연관 관계는 규칙으로 표현할 수 있고, 표 7에 이러한 규칙을 추출하는 알고리즘을 보였다.

<표 7> 규칙 추출 알고리즘

```
Until concept list is empty, Select a concept
  If type of selected concept's attribute is other concept
    Search a other concept(from concept list)
    If type of searched other concept's attribute is selected concept
      Established selected concept Rule with searched other concept
```

규칙 추출 알고리즘에 의해 그림 2의 대학 연구센터 온톨로지에서 개념간의 연관관계를 추론할 수 있다. 예를 들어 개념 Project의 속성인 member가 개념 Student와 서로 연관관계가 있으며, 온톨로지에 내포된 관계를 규칙으로 추출하면 표 8과 같다.

<표 8> 온톨로지상의 연관관계에 의한 규칙 추출

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

위와 같은 규칙을 이용한 연관관계에 의한 질의 확장 알고리즘을 표 9에 보였다.

<표 9> 연관 관계에 의한 질의 확장 알고리즘

```
Analyze XML query
  Select a adapted concept for inputted query Until concept list is empty
    If concept has RULE with the adapted concept
      ADD the concept at XML query
```

4. XML 문서 여과 및 질의 확장에 대한 실험

본 절에서는 구현한 DTD 생성기와 XML 질의 확장기에 의한 문서 여과 및 확장된 질의의 효과를 실험하였다.

4.1 포괄적 DTD에 의한 문서 여과

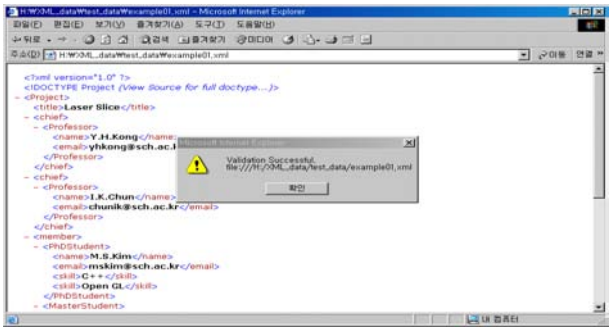
대학 연구센터와 관련된 영역의 XML 문서를 포괄할 수 있도록 설계된 DTD를 이용하여 XML 문서를 여과하였다.

표 10의 Example1.xml은 대학 연구센터의 프로젝트를 제목과 구성원 등으로 구성한 예로서 문서 구조를 정의한 Example1.dtd를 포함한다.

<표 10> Example1.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>
```

그림 5는 포괄적 DTD에 의해 Example1.xml 문서의 유효성을 검사한 결과로서, 유효한 문서로 판별된 예이다.



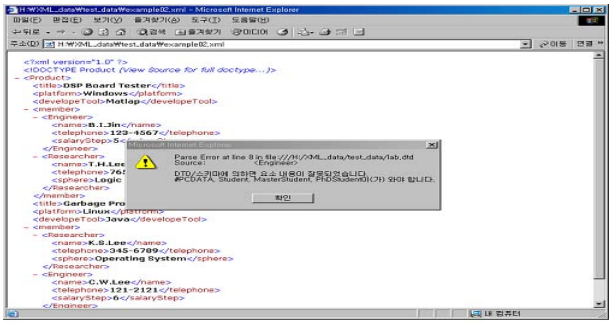
(그림 7) 포괄적 DTD에 의해 선택된 문서

표 11의 예제 Example2.xml은 업체의 제품 생산에 관련된 내용으로 제품의 제목, 사용 환경, 제작도구, 구성원에 관한 항목으로 구성되며, 이 문서도 구조를 정의한 DTD를 포함한다.

<표 11> Example2.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>
```

Example2.xml 문서는 앞의 Example1.xml 문서와 비슷한 구조와 항목을 갖는다. 그러나 이 문서에서 member 항목은 개발기술자(Engineer)를 의미하지만, 포괄적 DTD에서는 Student를 의미한다. 이는 포괄적 DTD는 member의 속성인 Student의 하위 개념으로 MasterStudent, PhDStudent를 포함하며, Engineer 항목은 member가 아닌 외부기술자문을 의미하게 되어, 의미적으로 서로 다른 개념이 된다. 따라서 Example2.xml은 그림 6과 같이 포괄적 DTD에 의한 유효성 검사에서 실패하게 된다.



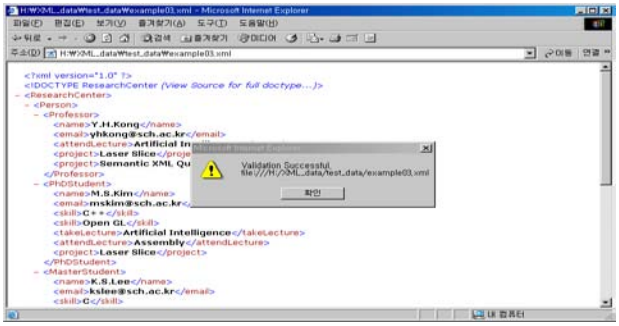
(그림 8) 포괄적 DTD에 의해 제거되는 문서

표 12의 예제 Example3.xml는 대학 연구센터에 소속된 구성원과 프로젝트에 관한 내용으로, 소속된 구성원은 교수, 박사, 석사로 분류되고, 프로젝트는 제목, 책임자, 구성원 등으로 분류되어 있다.

<표 12> Example3.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>
```

Example3.xml은 대학 연구센터와 관련된 문서이지만, 앞에서 보인 Example1.xml과는 구조적으로 다르다. 그러나 포괄적 DTD는 Example1.xml 문서의 구조와 Example3.xml 문서의 구조를 모두 포함하고 있으므로 포괄적 DTD에 의한 유효성 검사 결과는 그림 7과 같이 유효한 문서로 판별된다.

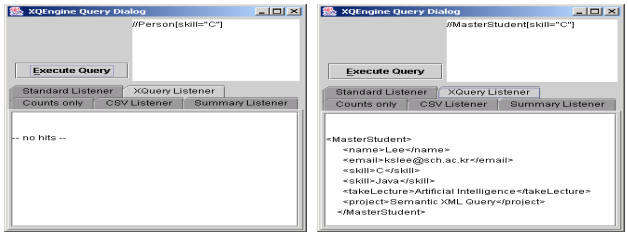


(그림 7) 포괄적 DTD에 의해 선택된 문서

결과적으로 포괄적 DTD에 의한 XML 문서들의 유효성 검사에 의해 정보 검색에 필요한 문서를 여과하였다.

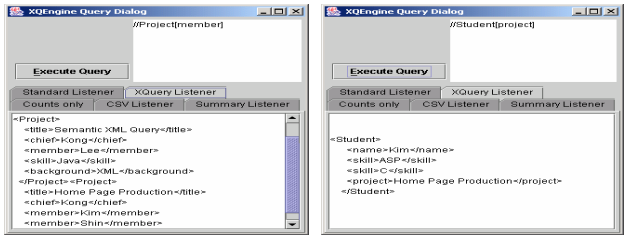
4.2 여과된 XML 문서에 대한 질의 확장

Example3.xml 문서로부터 “C”에 관한 skill을 갖는 사람을 검색하는 //Person[skill="C"]와 같은 단순 질의는 온톨로지의 계층 구조에 의해 해당 개념의 하위 개념으로 질의가 확장된다. 예제 온톨로지에서 Person에 대해 질의는 Person의 하위 개념인 Professor, Engineer, Student로 질의가 확장된다. 그림 8은 질의 확장 이전과 이후의 검색 결과이다. //Person[skill="C"]라는 단순 질의로는 검색되지 못했던 정보가, 확장된 질의에 의해 PhDStudent 내의 “C” 라는 정보를 검색하였다.



(그림 8) 계층 구조에 의한 질의 확장 이전과 이후의 검색 결과

온톨로지 연관관계에 의해 Project에 참여하는 member를 검색하는 //Project[member]라는 질의는 Project에 연관된 Student가 추가되고, Student의 하위 개념인 MasterStudent, PhDStudent에 대해서도 질의가 확장된다. 그림 9는 온톨로지의 상호 연관관계에 의한 질의 확장 이전과 이후의 검색 결과이다.



(그림 9) 연관 관계에 의한 질의 확장 이전과 이후의 검색 결과

결과적으로 문서 내에는 존재하지만 검색할 수 없었던 정보가 확장된 질의에 의해 추가적인 정보 검색이 가능하였다.

5. 결론

본 논문은 정보검색 대상이 되는 XML 문서를 여과하기 위해 온톨로지 기반의 포괄적 DTD 생성기를 구현하였고, 온톨로지에 내포되어 있는 구조와 연관 관계를 추론하여 XML의 의미 정보를 검색할 수 있는 XML를 질의 확장기를 구현하였다. 포괄적 DTD는 정보 검색의 대상이 되는 XML 문서를 사전에 선별할 수 있었으며, 선별된 XML 문서만을 대상으로 확장된 XML 질의는 문서에 내포되어 있음에도 불구하고 기존의 질의로는 검색할 수 없었던 의미 정보를 보다 효과적으로 검색할 수 있었다.

참고문헌

[1] Natalya F. Noy, Mark A. Musen, “The PROMPT suite:interactive tools for ontology merging and mapping”, Int. J. Human-Computer Studies 59, pp.983-1024, 2003.

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

[3] E. Mena, V. Kashyap, A. Illarramendi, A. Sheth, “Domain specific ontologies for semantic information brokering on the global information infrastructure”, N. Guarino, ed. Formal Ontology in Information Systems. IOS Press, 1998.

[4] T. R. Gruber, “A Translation Approach to Portable Ontology Specifications”, Knowledge Acquisition. Vol.6, No.2, pp.199- 221, 1993.