웹 정보원 통합을 위한 XML 기반의 랩퍼 시스템

배종민* · 박은경** · 정채영**

An XML-based Wrapper System for Integrating Web Information Sources

Jong-Min Bae* · Eun-Koung Park** · Chai-Young Jung**

이 논문은 한국과학재단(No. R01-2001-000-00151-0)과 한국전자통신연구원(0301-2006-0017)의 지원으로 수행되었음.

요 약

최근 웹 정보원에서 제공하는 정보가 정보서비스의 주류를 이루면서 웹 정보원 랩퍼 개발의 중요성이 크게 부각되었다. 본 논문은 웹 정보원을 통합하기 위한 미들웨어로서의 웹 랩퍼를 설계, 구현한 결과를 제시한다. 특히 HTML 문서로부터 XML 스키마로 변환하는 방법을 제시하고, XQuery 질의어에 대한 파서와 XQJ 기반의 질의처리 과정을 제시한다. 그리고 개발된 랩퍼 API의 사용 예를 통하여 그 유용성을 보인다.

ABSTRACT

It became important to develop a wrapper for web information sources due to prevalence of information services through web information sources. We present a wrapper prototype that is a middleware to integrate web information sources. We present the derivation strategy of XML Schema from HTML documents and the query processing method based on XQJ. The usage example of wrapper API will show the usefulness of our prototype system.

키워드

Database Integration, XML, XML Schema, XML View, XQJ, Wrapper

Ⅰ. 서 론

많은 정보원으로부터 사용자에게 양질의 정보를 쉽게 제공할 수 있는 방법 중의 하나는 물리적 혹은 가상적인 방법으로 정보원들을 통합하는 것이다. 전통적인 정보원 통합의 대상은 주로 관계형, 혹은 객체관계형 데이터베이스 등이었으나 최근에는 웹 정보원의 중요성이 부각되면서 웹 정보원을 통합하기 위한 랩퍼에 대한 많은 연구가 진행되고 있다. 웹 랩퍼에서의 주요 연구대상은 웹 정보원에서 제공하는 HTML 문서로부터 XML 스키마를 유도하는 방법에 관한 연구와 HTML 문서로부터 새로운 XML 뷰를 생성하는 방법에 관한 연구인데, 이 문제를 정보추출(Information Extraction: IE)문제라고 부른다. 이러한 정보추출과정을 자동으로 혹은 반자동으로 구현하는 기술에 관하여 연구가 이루어지고 있다[1,2,3].

본 연구에서는 웹 정보원 랩퍼 미들웨어 개발에 관하

* 경상대학교 컴퓨터과학부/컴퓨터정보통신연구소 교수
** 경상대학교 컴퓨터과학과

접수일자 : 2006. 8. 18

여 논한다. 이때 웹 정보원만을 위한 시스템을 개발하는 것이 아니라, 본 연구진에서 이미 설계 구현한 관계형, 객체관계형 등의 데이터베이스를 통합하기 위한 범용랩퍼의 구조와 연동하는 웹 정보원 랩퍼의 설계개념을 제시한다. 즉, 이미 설계 구현된 범용랩퍼 API(Application Program Interface)를 기반으로 웹 랩퍼를 개발한 결과를 제시한다.

설계 구현된 웹 랩퍼의 주요 특징은 다음과 같다. 첫째, 웹 정보원이 제공하는 HTML 문서에 대하여 의미있는 XML 문서로 변환하여 XML 스키마를 정의한다. 랩퍼의 기본 역할은 각 정보원이 관리하는 데이터에 대한 스키마를 추출하는 것이다. 그런데 관계형 모델과는 달리 웹 정보원은 명시적으로 스키마를 제공해 주지 않는다. 본 논문에서는 검색결과문서인 HTML 문서의 구조가 바로 웹 정보원이 관리하고 있는 데이터의 스키마로 간주하고 이를 기반으로 XML 스키마를 정의한다. 이때 HTML 문서의 태그(tag)는 주로 브라우징을 위한 것이기 때문에 XQuery[4] 등의 언어로써 질의하는 것이 거의 무의미하다. 따라서 사용자가 웹 정보원에 대하여 XQuery 등의 언어로써 의미있는 질의를 하기 위해서는 HTML 문서로부터 의미있는 XML 문서로 변환한다. 개발된 랩퍼에서는 그 변환규칙을 XQuery로써 표현한다.

둘째, 웹 정보원에 대한 XML 스키마를 정의할 때 검색결과에 대한 XML 스키마와 검색질의에 대한 XML 스키마를 함께 구성하여 하나의 XML 스키마로 정의한다. 검색엔진이 제공하는 HTML FORM 기반의 인터페이스를 통하여 검색결과를 얻는다면 입력 FORM에 관련된 정보 또한 스키마에 포함시킴으로써, XQuery로써 웹 정보원에 대한 질의 작성을 가능하게 하고 XQuery로부터 URL 질의스트링으로 변환하는데 활용한다.

셋째, 사용자가 XQuery를 사용해서 질의를 작성하고 결과문서를 얻는데 사용되는 API로는 관계형 데이터베이스에서 JDBC에 대응되는 개념인 XQJ[5]를 구현하여 질의를 처리한다.

기존의 많은 데이터베이스 통합시스템은 미디에이터-랩퍼 모델에 기반을 두고 있다.[13,14] 이들은 본 논문에서 제시하는 모델처럼 범용을 지향하고 있다. 이들 시스템과 본 시스템 사이에의 가장 큰 차이는 사용자정의 XML뷰 기능을 지원한다는 점과 XQJ를 지원하는 점, 그리고 강력한 웹 랩퍼기능이 있다는 것이다. 한편, 웹 전용 랩퍼시스템에 관한 연구도 많이 있는데[2,3,12] 이들은 주로 HTML문서로부터 스키마추출방법과 랩퍼소스를 자동으로 생성하는 방향에 그 초점이 있어서, 미디에이터에게 제공되는 강력한 랩퍼API를 기반으로 한 본 시스템과는 설계개념상의 차이가 있다.

본 논문의 구성은 다음과 같다. 2장에서는 설계된 웹 랩퍼 시스템의 전체적인 구조를 제시한다. 3장에서는 HTML 문서로부터 XML 스키마를 생성하는 시스템에 대하여 논한다. 4장에서는 XQuery로 작성된 사용자질의를 처리하기 위한 XQuery 파서(parser)의 동작원리 및 XQJ 구현에 대하여 논한다. 5장에서는 웹 랩퍼를 구현한 결과 및 API 구조를 간략히 설명하며, 마지막으로 결론 및 고찰에 대하여 논한다.

II. 시스템 구조

본 논문에서 제시하는 웹 랩퍼 시스템의 구조는 하나의 웹 정보원당 하나의 랩퍼가 존재하는 전통적인 구조와는 달리, 통합하고자 하는 모든 웹 정보원에 대해서 하나의 랩퍼가 존재하며 미디에이터의 한 컴포넌트로서 수행한다. 이는 랩퍼는 정보원에 접근해야 하기 때문에 기본적으로 정보원이 운영되는 시스템에 있어야하지만, 웹의 경우에는 정보원이 로컬시스템이 아니므로 랩퍼는 원격시스템에 존재해야 한다. 이때 랩퍼를 미디에이터와 함께 두고, 하나의 랩퍼가 여러 정보원에 대한 각각의 스키마를 관리하는 것이 개념적으로 훨씬 간단하기 때문이다.

그림 1은 본 논문에서 제시하는 웹 랩퍼 시스템의 구조를 나타낸다. 랩퍼는 크게 3개의 모듈로 구성되는데 웹 정보원에 대한 XML 뷰를 정의하고 그에 대한 XML 스키마

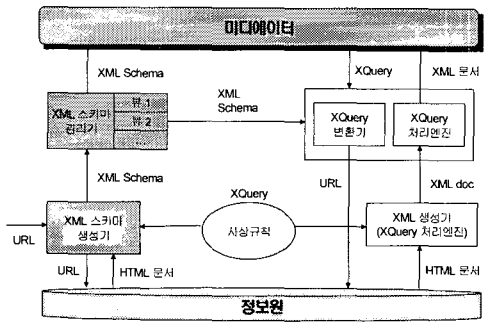

그림 1. 웹 랩퍼 시스템 구조
Fig. 1. Architecture of the Web Wrapper System

를 생성하는 XML 스키마 생성기(XML Schema Generator), 다수의 웹 정보원에 대한 스키마를 관리하는 XML 스키마 관리기(XML Schema Manager), 그리고 질의 변환 및 XML 결과 문서를 생성하는 질의 처리기로 구성된다.

XML 스키마 생성기에서는 각 웹 정보원에 대한 XML 뷰(view)를 생성하고 이를 기반으로 XML Schema[6,7,8]를 추출하여 XML 스키마 관리기에게 전달한다.

웹 정보원에 대한 스키마를 추출하기 위하여 먼저 샘플 URL을 생성하여 검색결과로서 HTML 문서를 받으면 이를 XHTML로 변환한 다음, XHTML 문서로부터 XML 스키마를 추출하는 알고리즘을 적용하여 XML Schema를 생성한다. 그런데 HTML 태그는 문서의 구조를 나타내는 것이 아니라 문서를 브라우징하는 방법을 표현한 것에 더 가깝다. 따라서 이 HTML 문서구조에 대한 스키마를 대상으로 XML 질의를 생성하는 것은 의미가 없다. 따라서 HTML 문서로부터 의미를 가지는 XML 문서로 재정의할 필요가 있는데, 이때 HTML 문서로부터 새로운 XML 문서를 생성하기 위한 사상규칙이 필요하다. 본 논문에서는 이 사상규칙을 표현하는 도구로 XQuery를 이용한다. XQuery에 의해 표현된 사상 규칙은 가상의 XML 문서를 생성하는 것으로 간주될 수 있다. 이 규칙은 추후에 사용자질의에 대한 결과로서의 HTML 문서를 XML 문서로 변환할 때도 사용된다.

XML 스키마 관리기에서는 웹 정보원에 대한 XML 스키마를 정의, 등록, 삭제 등의 기능을 담당한다. 본 논문에서 제시하는 웹 랩퍼는 다수의 웹 정보원을 하나의 랩퍼에서 관리하기 때문에 다수의 XML 스키마가 생성된다.

질의처리기는 질의 변환 및 XML 결과 문서를 생성하는 기능을 수행한다. 사용자로부터 XQuery로 표현된 질의를 받으면 XQuery 변환기(XQuery Translator)는 웹 정보원에대한 URL로 변환하여 검색질의를 수행한다. 웹 정보원으로부터 받은 결과는 HTML 문서이다. XML 생성기(XML Generator)에서는 HTML 문서를 XHTML 문서로 정제한 후, XQuery로 표현된 사상규칙을 적용시켜 의미기반의 XML 문서로 변환한다. 질의 처리기에서는 사상규칙을 적용시킨 가상의 XML 문서를 대상으로 사용자 질의를 수행하여 그 결과로서 XML 문서를 생성한다. 본 연구에서는 XQuery를 수행하는 처리 엔진으로 SAXONICA의 Saxon-B[9]를 사용하였으며 질의처리는 XQJ 기반으로 구현하였다.

Ⅲ. XML 스키마 생성

본 장에서는 웹 정보원이 제공하는 HTML 문서로부터 XML 스키마를 유도하는 방법에 대하여 논한다. 제시하는 XML 스키마는 그림 2와 같이 입력스키마(Input) 부분과 출력스키마(Output) 부분으로 구성된다.

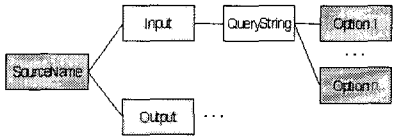

그림 2. XML 스키마 구성
Fig. 2. Structure of XML Schema

3.1. 입력스키마

제시하는 랩퍼에서는 동적 HTML 문서를 지원한다. 이때 HTML 폼(form) 기반의 입력자료는 URL의 질의 부분을 생성하는데 사용되어야 하는데, 이를 위하여 본 연구에서는 HTML 폼 역시 웹 정보원이 관리하는 데이터로 간주하여 이를 XML 스키마로 변환하는데 이 스키마를 입력스키마(Input Schema)라 한다. 입력스키마의 구성은 그림 2의 <Input> 엘리먼트 부분과 같다. HTML 문서의 <form>태그 내에 있는 <input>태그의 name속성들이 엘리먼트로 생성되어 <Input> 엘리먼트 하위에 오게되고, 이들은 하나 이상의 엘리먼트로 생성되기 때문에 <QueryString> 엘리먼트로 묶는다. 이때 생성되는 엘리먼트의 형은 해당 원소의 type 속성에 따르는데, 예를 들어, radio, select, checkbox 등은 XML 스키마에서 열거형으로 표현한다.

```
<TR><!--query box-->
  <TD>
    <TABLE id=searchbar cellSpacing=0 cellPadding=1 width="100%"
    bgColor=#cccccc border=0>
      <TBODY>
      <TR>
        <TD class=SMALL1 noWrap> Search <SMALL>
          <SELECT onchange=DbChange(this); name=db>
            <OPTION value=pubmed selected>PubMed</OPTION>
            <OPTION value=protein>Protein</OPTION>
              ...
            <OPTION value=unists>UniSTS</OPTION>
          </SELECT>
          <INPUT type=hidden value=pubmed name=orig_db></SMALL>
        for <INPUT id=term size=45 name=term>
          <INPUT onclick="Go('search');" type=submit value=Go>
            ...
          <INPUT type=hidden value=Search name=cmd>
</TD></TR></TBODY></TABLE></TD></TR>
```

그림 3. NCBI Pubmed의 질의인터페이스 문서
Fig. 3. Query Interface of NCBI's Pubmed

그림 3은 NCBI의 PubMed에서 제공하는 질의인터페이스 문서의 일부이고, 그림 4는 이를 XML 스키마로 표현한 것이다. 그림 3의 <SELECT> 태그부분은 그림 4의 9번째 줄과 같이 표현된다. <SELECT>태그가 아래에 나오는 <OPTION>태그 중의 하나를 선택하게 되므로, XML 스키마 표현 시 'enumeration' 으로 표현된다. 그림 3의 <SELECT>태그 아래에 나오는 <INPUT> 태그이름은 'orig_db', 'term', 'cmd' 엘리먼트로 각각 표현된다.

3.2. 출력스키마

웹 정보원에 대한 스키마를 정의할 때, 임의의 검색요구에 의한 검색결과(HTML 문서)를 XML 문서로 변환하여, 이 XML 문서에 대한 스키마를 웹 정보원이 관리하는 데이터의 스키마로 간주하는데 이는 XML 스키마의 출력스키마(Output Schema)를 구성한다. 출력스키마를 생성하는 구조는 그림 5와 같다.

```
01 :<xsd:element name="Input">
02 :   .....
03 :   <xsd:element name="QueryString" >
04 :     <xsd:complexType>
05 :       <xsd:all>
06 :         .....
07 :           <xsd:element name="db" type="xsd:string">
08 :             <xsd:simpleType>
09 :               <xsd:restriction base="xsd:string">
10 :                 <xsd:enumeration value="pubmed" />
11 :                 .....
12 :               </xsd:restriction>
13 :             </xsd:simpleType>
14 :           </xsd:element>
15 :           <xsd:element name="orig_db" type="xsd:string" fixed="pubmed" />
16 :           <xsd:element name="term" type="xsd:string" />
17 :           <xsd:element name="cmd" type="xsd:string" fixed="Search" />
18 :       </xsd:all>
19 :     </xsd:complexType>
20 :   </xsd:element>
21 :   .....
22 :</xsd:element>
```

그림 4. NCBI Pubmed의 입력 스키마
Fig. 4. Input Schema of NCBI's Pubmed

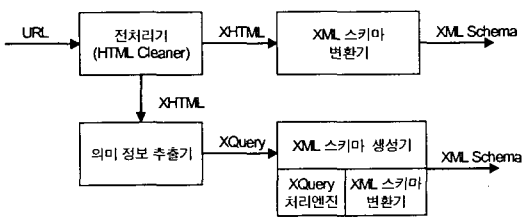

그림 5. 출력 스키마 생성기의 구조
Fig. 5. Architecture of the Output Schema Generator

웹정보원에 대한 주소가 주어졌을 때 검색 에이전트에 의해서 얻어진 HTML 문서에 대하여 전처리기는 정형(well-formed)의 HTML 문서인 XHTML 문서로 변환한다. 이 문서를 XML 문서로 간주할 수도 있기 때문에 XML 스키마 변환기는 이 문서로부터 XML 스키마를 자동으로 생성한다.

한편, XHTML 문서에서 정보를 추출하여 새로운 XML 문서를 생성한 경우, 나중에 질의처리를 위하여 두 문서를 사상하는 규칙이 필요하다. 본 연구에서는 정보추출과정을 데이터베이스의 사용자 뷰 정의의 개념으로 간주한다. 뷰 정의언어는 XQuery를 사용하기 때문에 정보추출 규칙을 표현하기 위해서 XQuery를 이용한다. XQuery로 표현된 사상규칙은 XHTML 문서에 대한 질의로 간주하여 XQuery 엔진을 수행시켜서 원하는 XML 문서를 생성한다. 그리고 XML 스키마 변환기는 생성된 XML 문서를 XML 스키마로 변환한다.

그림 6은 PubMed가 제공하는 검색결과문서에 대하여 사상 규칙을 정의한 예이다.

```
01: <medline>
02:     {
03:     for $x in doc("pubmed")/html/body/form/table[2]/tr[2]/td[3]/div/table
04:     where $x/@cellspacing="5"
05:     return
06:         <book id = "{substring-after($x/tr[2]/td[1]/a/img/@id, "_")}">
07:             <authors>
08:             {
09:             for $y in tokenize(substring-before
10:                         ($x/tr[1]/td[2]/a/font/text(), "."), ",")
11:             return
12:                 <author> {normalize-space($y)} </author>
13:             }
14:             </authors>
15:             <title> {substring-before
16:                         (string($x/tr[2]/td[2]), ".")} </title>
17:             <journal> {substring-before
18:                     ($x/tr[2]/td[2]/font/span/text(), ".")} </journal>
19:             <date> {substring-after(substring-before
20:                 (string($x/tr[2]/td[2]/font), ";"), ". ")} </date>
21:             <status> {substring-before(substring-after
22:         (string($x/tr[2]/td[2]/font), "[PubMed - "), "]")} </status>
23:             </book>
24:     }
25: </medline>
```

그림 6. 사상규칙
Fig. 6. The Mapping Rule Written in XQuery

Ⅳ. 질의 처리

4.1. 질의 처리 개요

랩퍼의 질의 처리 과정은 그림 7과 같다. XQuery로 표현된 사용자 질의는 질의 변환기를 통해 URL로 변환한 후 웹 정보원으로 전달한다. 웹 정보원에서 수행된 결과로서 HTML 문서를 받으면 XQuery로 표현된 사상 규칙을 적용시켜서 가상의 XML 문서를 생성한다. 여기서 XQuery로 표현된 사상 규칙을 적용시킨다 함은 XQuery 엔진으로 이 사상규칙을 실행시킨다는 의미이다. 사용자는 웹 정보원에 대한 검색결과가 이 가상의 XML 문서인

것으로 간주한 결과가 된다.

사용자는 XML 스키마를 통해서 질의를 한다. 그리고 XML 스키마는 입력스키마와 출력스키마로 구성되어 있다고 하였다. 그런데 입력스키마는 URL로 변환하기 위하여 사용되는 것이지 실제 정보원이 관리하는 데이터에 대한 질의가 아니기 때문에, 사용자질의로부터 URL을 생성한 이후에는 사용자질의에서 입력스키마에 관련된 질의 부분을 제거해야 한다. 이는 질의재작성기(Query Rewriter)가 그 역할을 담당한다. 재작성 된 사용자 질의는 XQJ 처리기에 의해서 질의결과객체를 생성하여 사용자에게 전달한다.

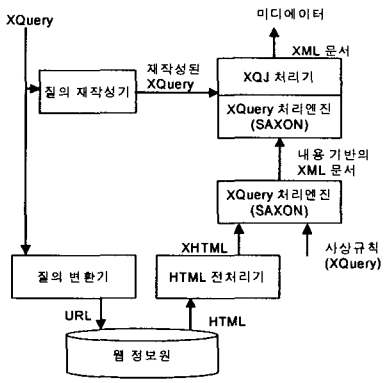

그림 7. 웹 랩퍼에서 질의 처리 과정
Fig. 7. Concept of Query Processing

4.2. XQuery로부터 URL로의 변환

앞 절에서 작성된 스키마를 기반으로 작성된 사용자 질의의 처리 과정을 예를 들어 설명한다. 사용자는 virus와 관련된 문헌을 검색하고자 하는데 검색된 논문이 "Cancer" 저널에 발표된 논문을 검색하고자 한다. XQuery로 작성된 사용자 질의의 예는 그림 8과 같다.

```
<query>
        {
        for $in_in document("refinepubmed")/refinepubmed/input,
            $out in document("refinepubmed")/refinepubmed/output/medline/book
        where $in//db = "pubmed"
        and $in//orig_db = "pubmed"
        and $in//term = "virus"
        and $in//cmd = "Search"
        and $out//journal = "Cancer"
        return
                <book title=$out/title/text()>
                        {$out/authors}
                        <date> {$out/date/text()} </date>
                </book>
        }
</query>
```

그림 8. 사용자 질의
Fig. 8. User Query

URL 변환에 필요한 부분을 사용자 질의로부터 추출하기 위해서는 XQuery로 작성된 사용자 질의를 분해한다. 본 연구진에서 개발된 XQuery 파서는 사용자 질의를 분석하여 XQuery를 구성하고 있는 각각의 문법적 요소를 대응되는 객체로 표현한다. 이때 만들어지는 객체의 예로는 StartElement, EndElement, Variable, Predicate, Conjunction 등이 있다. StartElement와 EndElement는 XQuery에 표현되어 있는 열린 엘리먼트와 닫힌 엘리먼트의 정보를 가지고 있는 객체이다. 주로 XQuery의 생성자 부분인 Return절에 구성되어 있다. Variable 객체는 XQuery의 For절에 바인딩된 변수의 정보를 저장하는데 변수의 이름과 이 변수에 바인딩된 XPath 경로를 저장한다. Prediacate 객체는 Where 절의 조건 정보와 관련이 있는데 Where절에 나타나는 조건식의 연산자 좌우에 존재하는 경로식이나 값 등이 저장된다. Conjunction 객체는 이 조건절들 사이에 존재하는 "and" 혹은 "or"의 값을 저장하고 있는 객체이다.

그림 9는 그림 8의 사용자 질의를 XQuery 파서로 파싱하여 XQuery의 문법적인 요소를 대응되는 객체로 변환하여 리스트에 저장한 결과이다. 이렇게 XQuery 파서에 의해서 XQuery의 문법적 요소를 객체화시켜 리스트에 저장해 두고 입력 스키마와 관련된 부분을 찾아서 URL로 변환하는 작업을 수행한다. 그림 10은 파싱된 객체를 이용하여 URL을 생성하는 알고리즘이다. 계속해서 사용자 질의에서 입력 스키마 관련부분을 제거하여 재작성된 사용자 질의를 생성한다. 이는 그림 9에서 생성된 객체를 조립하여 생성되는데 그 알고리즘은 여기서는 생략한다.

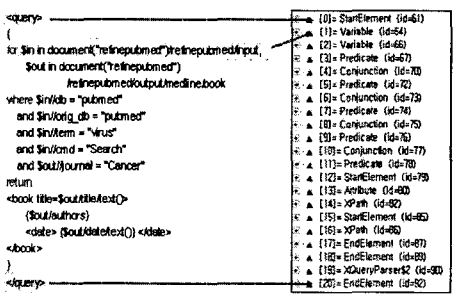

그림 9. 파싱된 사용자 질의
Fig. 9. Parsed User Query

```
//Predicate 객체를 이용하여 URL 변환하는 메소드
Translate(Predicate p) {
    StringBuffer url = new StringBuffer();

    if(VariableName = "Input") {
        String baseURL = generateBaseURL();
        //메타정보DB를 검색하여 기본 URL을 생성한다.

        url.append(baseURL + "?");

        //XQuery의 조건절이 저장되어 있는 Predicate객체를 이용하여
        질의 스트링 생성을 시작한다.
        List args = predicate.getArgs();

        XPath c = (XPath)args.get(0);
        //조건절의 왼쪽에 있는 XPath 경로를 추출
        Token t = (Token)args.get(1);
        //조건절의 오른쪽에 있는 값 추출

        //질의 스트링의 이름에 해당되는 부분이 조건절에 명시되어 있는
          XPath 경로의 마지막 엘리먼트명이기 때문에 그 부분을 추출
        String queryString = c.getLastStep();
        if(queryString.equalsIgnoreCase("text()") {
            String fullStep = c.getStep();
            String[] step = fullStep.split("/");
            queryString = step[step.length - 2];  }

        //질의스트링의 "이름=값" 형태 생성
        url.append(queryString + "=");
        url.append(t.getValue() + "&");
}}
```

그림 10. URL 변환 알고리즘
Fig. 10. Algorithm for URL Conversion

4.3. XQJ의 구현

XQJ는 자바 커뮤니티 프로세서(JCP)에서 제안한 XQuery API for Java의 약자로서 현재 초기 제안 리뷰(Early Draft Review) 상태이다. XQJ는 XML 문서 뿐 아니라 XML을 지원하는 관계형 데이터베이스를 대상으로 XQuery로 질의를 하고 그 결과를 얻기 위한 API를 제공한다. XQJ 구현의 핵심 중의 하나는 관계형 데이터베이스의 커서와 같은 개념을 적용하여 결과 문서의 일부분을 차례로 전달하는 방법을 제공하는 것이다.

본 연구에서는 전체 XQJ 명세에서 웹 정보원 질의처리에 필요한 내용을 WXQJ라는 이름을 구현하였다. WXQJ에서 클래스 WXQExression은 XQuery를 실행하고 그 결과를 반환하기 위한 클래스이다. 이를 구현하기 위하여, Saxson-B를 이용하여 XQuery를 실행시키고 반환되는 결과 객체를 분석하여 커서 개념을 지원할 수 있도록 WXQItem 객체로 재조립한다. WXQItem 클래스는 WXQJ 처리 결과를 커서 개념을 지원할 수 있도록 하는 최소 단위이다. 이 클래스에는 getInt, getString, getNode등 다양한 메소드들을 지원한다.

V. 구 현

5.1. 랩퍼 미들웨어 구성

랩퍼 미들웨어는 총 18개의 패키지로 구성된다. 최상위 wrapper 패키지에는 사용자가 랩퍼를 사용하기 위한 인터페이스를 정의한다. 웹 랩퍼의 세부적인 기능들은 web 패키지 아래에 존재한다. manager 패키지는 웹 정보원에 대한 스키마를 관리 및 이를 위한 메타 정보를 데이터베이스에 저장, 삭제, 갱신한다. wxqjimpl 패키지는 XQJ API를 구현한 패키지이다. util 패키지는 입력 스키마를 생성하는 f2xsd 패키지와 HTML을 XHTML 문서로 변환하는 tidy[10] 패키지, 그리고 XHTML 혹은 XML 문서에서 출력 스키마를 생성하는 xml2xsd 패키지로 구성되어 있다. query 패키지는 입력 스키마와 출력 스키마를 통합하는 기능과 질의 변환 및 질의를 재작성하는 기능을 수행한다. xquery와 tools 패키지는 XQuery를 입력받아 파싱하고, 이를 구성하는 문법적인 구조를 가진 객체 리스트를 생성하는 역할을 한다.

5.2 랩퍼 API 활용

개발된 랩퍼 API에 대한 상세한 설명 대신, API 사용 예를 간단히 보인다. 그림 11에서 먼저 WWrapper형의 랩퍼 객체를 하나 생성시킨 후, 그것으로 WSchemaManager형의 스키마 관리 객체를 만들어서 스키마를 생성하고 등록한다. 사용자질의를 위하여 XQJ API 중의 하나인 WXQExpression 형의 객체를 통하여 질의를 수행하고 그 결과를 WXQSequence형의 객체가 받아서 데이터베이스 커서개념을 활용하여 화면에 출력한다.

```
//랩퍼 객체를 생성하여 스키마 정리자를 등록한 후, 웹자원에 대한 XML 스키마를 생성한다.
WWrapper wrapper = new WWrapperImpl();
WSchemaManager sm = wrapper.getSchemaMeneger();
sm.createSchema("pubmed", "http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=Pubmed",
   "http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=Pubmed&orig_db=pubmed&term=james&cmd=Search");

//사상규칙을 적용하여 의미기반의 XML 스키마를 생성한다.
...
sm.refineSchema("pubmed",rule.toString(), "refinepubmed");

//사용자 질의를 수행한다.
...
WXQExpression expr = wrapper.createExpression();
WXQSequence result = expr.executeQuery(queryBuf.toString());

while(result.next()) {
    System.out.println("queryResult : " + result.getString());
}
```

그림 11. 랩퍼 API 사용 예
Fig. 11. An Usage Example of the Wrapper API

5.3 평가

본 시스템은 모든 웹 정보원에 대하여 적용될 수 있는 범용 랩퍼이다. 따라서 개발된 시스템은 대부분의 시스템에서 원활히 동작해야 한다. Lycos, 아마존, 구글, 야후, PubMed, Yes24, 알라딘, GeneBank[11] 등 많은 공공 사이트에 대하여 테스트 하였다. 테스트 결과, HTML 문서가

정상적으로 작성된 경우는 모두 잘 수행되었으나 일부 사이트는 정상적으로 동작하지 못하였다. 그 원인을 분석한 결과, URL이 수시로 변경되는 사이트, 입력화면에서 폼(form) 태그가 두 개 이상 존재하여 어느 폼 태그가 내용검색태그인지 프로그램 상에서 판단하기 어려운 사이트, 접근거부가 발생하는 사이트 등에서 동작되지 않았으며, 그 외 HTML 문법에 맞는 않게 작성된 유형이 매우 다양하여 이를 모두 지능적으로 처리하기 어려운 점 등이 있었다. 향후 이러한 문제가 모두 해결되기는 어려울 것으로 보이지만 계속해서 연구해야 할 과제가 많이 남아 있음을 보여준다.

VI. 결 론

웹 정보원 통합과 같은 응용을 개발하기 위해서는 웹 정보원에 대하여 데이터베이스 질의와 같은 질의기능이 가능하도록 하는 랩퍼가 필요하다. 본 논문에서는 웹 정보원에 대한 지역 스키마 추출 및 관리, 지역 스키마를 기반으로 의미있는 XML 문서를 생성하기 위한 사상규칙정의, XQuery 질의를 URL로 변환 및 실행, 그리고 질의수행결과를 쉽게 조작 가능하도록 XQJ API 구현 등의 기능을 갖춘 범용 랩퍼를 개발하고, 이 랩퍼를 활용하여 다양한 응용프로그램 개발을 용이하도록 하기 위한 API를 정의, 구현한 결과를 제시하였다.

개발된 시스템은 랩퍼 미들웨어로서 거의 완전한 기능을 갖추고 있으나, HTML 문서의 불규칙성으로부터 의미있는 XML 스키마를 효과적으로 추출하는 문제는 랩퍼의 성능을 높이는 핵심요소 중의 하나이기 때문에 향후 이에 대한 연구가 더 필요하다.

참고문헌

[1] Alberto H. F. Laender, Berthier A. Ribeiro-Neto, Altigran S. da Silva, Juliana S. Teixeira, "A brief survey of Web Data extraction tools." ACM Sigmod Record. 31(2):84-93, June 2002.

[2] Ling Liu, Calton Pu, Wei Han, "XWRAP: An XML-enabled Wrapper Construction System for Web Information Sources", In ICDE(San Diego CA), pages 611-621, 2000, February 2000

[3] Arnaud Sahuguet, Fabien Azavant, "Building intelligent Web application using lightweight wrappers", Data and Knowledge Engineering, 36(3):283-316, 2001

[4] World-Wide Web Consortium, "XQuery 1.0: An XML Query Language", [Online]. Available : http://www.w3.org/TR/xquery/, W3C Candidate Recommendation 8 June 2006

[5] Andrew Eisenberg, Jim Melton, "An early look at XQuery API for JavaTM(XQJ)", ACM Sigmod Record. 33(2):105-111, June 2004

[6] World-Wide Web Consortium, "XML Schema Part 0: Primer Second Edition", [Online]. Available :http://www.w3.org/TR/xmlschema-0/, W3C Recommendation 28 October 2004

[7] World-Wide Web Consortium, "XML Schema Part 1: Structures Second Edition", [Online]. Available : http://www.w3.org/TR/xmlschema-1/, W3C Recommendation 28 October 2004

[8] World-Wide Web Consortium, "XML Schema Part 2: Datatypes Second Edition", [Online]. Available : http://www.w3.org/TR/xmlschema-2/, W3C Recommendation 28 October 2004

[9] XSLT and XQuery Processing, [Online]. Available: http://www.saxonica.com/

[10] HTML TIDY, [Online]. Available: http://tidy.sourceforge.net/

[11] NCBI [Online]. Available: http://www.ncbi.nlm.nih.gov/

[12] Baumgartner, R., Flesga, S., Gottlob, G., "Visual Web information extraction with Lixto", In Proceedings of the 26th International Conference on Very Large Data Bases, pp119-128, 2001

[13] C.Baru, A.Gupta, B.Ludascher, R.MarchianoYannis, Y.Papakonstantinou, P.Velikhov, and V.Chu, "XML-based Information Mediation with MIX", In Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data, pp597-599, 1999

[14] Hammer, J., Garcia-molina, H., Nestorov, S., Yerneni, R., Breunig, M., and Vassalos, V., "Template-based wrapper in the TSIMMIS system", In Proceedings of the ACM SIGMOD International Conference on Management of data, pp532-535, 1997

저자소개

배 종 민(Jong-Min Bae)

1980년 서울대학교 수학교육과(학사)
1983년 서울대학교 대학원 계산통계학과 (석사)
1995년 서울대학교 대학원 계산통계학과(박사)
1982년-1984년 한국전자통신연구소 연구원
1997년-1998년 Virginia Tech. 객원연구원
1984년-현재 경상대학교 컴퓨터과학부 교수
※관심분야 : XML, 정보통합, 웹 정보기술

박 은 경(Eun-Koung Park)

1999년 경상대학교 컴퓨터과학과(학사)
2002년 경상대학교 컴퓨터과학과(석사)
현재 경상대학교 컴퓨터과학과 박사수료
※관심분야 : XML 데이터베이스 통합, 생물정보 통합

정 채 영(Chai-Young Jung)

1997년 경상대학교 전자계산학과(학사)
2001년 경상대학교 컴퓨터과학과(석사)
2004년 경상대학교 컴퓨터과학과(박사)
현재 경상대학교 컴퓨터과학과 시간강사
※관심분야 : XML, 웹 프로그래밍, 정보통합, 데이터베이스