Journal of Korea Multimedia Society Vol. 11, No. 5, May 2008(pp. 567-578)

XML을 이용한 웹 정보 추출 및 다차원 분석

박 병 권†

요 약

인터넷에 있는 방대한 양의 웹 페이지들을 분석하기 위해서는 웹 페이지에 내재된 정보를 추출하는 것이 필요하다. 본 논문에서는 웹 페이지로부터 정보를 추출하고 이를 XML 문서로 변환하여 다차원적으로 분석하는 방법을 제안한다. 웹 페이지로부터 정보를 추출하기 위하여 두 종류의 언어를 제안한다. 하나는 객체지향 모델에 의거하여 웹 정보 추출 규칙을 기술하기 위한 것이고, 다른 하나는 추출하고자 하는 정보를 찾기 위한 HTML 태그 패턴을 정규식으로 기술하기 위한 것이다. XML 문서에 대한 다차원 분석을 위하여 관계형 데이터에 대해 하는 것처럼 웨어하우스를 구축하고 이로부터 다양한 큐브를 생성하는 방법을 제안한다. 마지막으로 본 논문에서 제안한 방법을 미국특허 웹 페이지에 적용한 예를 통해 그 타당성을 보인다.

Web Information Extraction and Multidimensional Analysis Using XML

Byung-Kwon Park†

ABSTRACT

For analyzing a huge amount of web pages available in the Internet, we need to extract the encoded information in web pages. In this paper, we propose a method to extract and convert web information from web pages into XML documents for multidimensional analysis. For extracting information from web pages, we propose two languages: one for describing web information extraction rules based on the object-oriented model, and another for describing regular expressions of HTML tag patterns to search for target information. For multidimensional analysis on XML documents, we propose a method for constructing an XML warehouse and various XML cubes from it like the way we do for relational data. Finally, we show the validness of our method through the application to US patent web pages.

Key words: Web(웹), XML(XML), OLAP(온라인 분석처리), Data Cube(데이터 큐브), Data Warehouse (데이터 웨어하우스)

1. 서 론

오늘날에는 많은 조직들이 웹을 통하여 정보를 제공하므로 방대한 양의 정보가 웹에 존재한다. 웹 정보를 분석하기 위해서는 수많은 웹 페이지들을 웹 브라우저를 통해 모두 다 읽기보다는 질의하는 것이 보다 효과적이다. 질의를 위해서는 웹 페이지에 들어있는 정보를 추출하여 구조화된 데이터 (예, SQL 질의를 위한 관계형 데이터) 또는 반구조화된 데이터 (예. XQuery 질의를 위한 XML 데이터)로 변환하는 것이 필요하다. 웹 페이지가 정해진 스키마를 가진 구조화된 데이터 (주로 릴레이션 데이터)로 이루어져 있다면 자동 추출이 가능하다[1,2,3,4]. 그러나 많은 웹 페이지들은 비구조화된 데이터 (주로 텍스트 데이터)로 이루어져 있으므로 이러한 웹 페이지로부터 정보를 추출하기 위해서는 추출 방법을 명시해 놓은 추출 규칙이 필요하다.

본 논문에서는 HTML 태그 패턴(tag pattern)에

※ 교신저자(Corresponding Author) : 박병권, 주소 :부산광역시 사하구 하단 2동 840번지(604-714), 전화 : 051) 200-7480, FAX : 051)200-7481, E-mail : bpark@dau.ac.kr

접수일 : 2007년 6월 13일, 완료일 : 2008년 4월 2일
† 정회원, 동아대학교 경영정보과학부 부교수
※이 논문은 동아대학교 학술연구비 지원에 의하여 연구되었음

기반 한 추출 규칙을 사용한다. 즉, 웹 페이지 내에서 추출하고자 하는 정보를 찾을 수 있는 HTML 태그 패턴을 사용하여 추출 규칙을 명시한다. 이를 위해 TagRex (Tag Regular Expression)라는 언어를 제안하며, 추출 규칙을 기술하기 위해 xRule (eXtraction Rule)이라는 언어를 제안한다. 추출된 데이터는 XML 문서로 저장된다.

추출 규칙을 이용한 웹 정보 추출 연구로는 Hammer등의 연구[5]가 있다. 그러나 Hammer의 웹 정보 추출 규칙은 상호 의존적인 관계로 항상 처음부터 순차적으로 처리해야만 한다는 문제점이 있다. 그러나 본 논문의 추출 규칙은 서로 독립적이므로 병렬처리가 가능하다는 장점을 가진다.

HTML 태그 패턴을 이용한 웹 정보 추출 연구로는 IEPAD[2]가 있으나 IEPAD는 패턴 마이닝을 이용하여 규칙적이고 반복적인 패턴을 자동으로 찾는 문제를 다루었다. 패턴을 이용한 또 다른 연구로 W4F[6]가 있으나 W4F는 HTML 태그 계층 구조 경로를 정확히 기술해 주어야 하므로 사용하기가 어렵다. 반면에, 본 논문은 HTML 태그 정규식을 사용하므로 사용하기가 쉽다.

웹 정보 추출을 자동화하기 위한 연구들도 있다[3,7]. 그러나 구조화된 데이터를 가진 웹 페이지들만 대상으로 하였으며, 주로 웹 페이지 전체에 대한 페이지 문법을 찾는데 주력하였다. 반면에, 본 논문은 웹 페이지 중에서 사용자가 관심 있는 정보만 추출한다.

기존의 관계형 데이터처럼 XML 문서도 다차원적으로 분석하는 것이 필요하다. 그러나 XML 문서는 관계형 데이터와 달리 트리(tree) 구조를 가지고 있을 뿐만 아니라, 텍스트 데이터를 가지고 있다. 본 논문에서는 XML 문서에 대한 다차원 분석을 위하여 XML 웨어하우스와 XML 큐브를 만들어 다차원 질의를 할 수 있는 방법을 제안한다.

XML 문서의 다차원 분석에 관한 연구로는 Pedersen[8,9]등의 연구가 대표적이다. Pedersen등은 OLAP(Online Analytical Processing) 질의에 외부 XML 문서의 내용을 결합하여 OLAP 질의를 확대하는 문제를 연구하였다. 기존의 OLAP은 미리 정해져 있는 차원 데이터를 통해서만 질의할 수 있으나 외부 XML 문서와 결합하면 보다 확장된 차원 데이터에 기반한 질의가 가능해 진다. 그러나, 여전히 OLAP 질의의 대상이 XML 문서가 아닌 관계형 데이터이다.

한편 Golfarelli[10], Nassis[11], Pokorny[12] 등은 XML 문서에 대한 웨어하우징 개념을 제안하였다. Golfarelli[10]등은 XML 데이터로부터 개념 스키마를 자동적으로 찾는 문제를 연구하였고, Nassis[11]등은 UML을 이용한 XML 웨어하우스의 개념적 모델 설계에 관하여 연구하였다. Pokorny[12]는 사실 데이터와 차원 데이터가 모두 XML 문서로 기술된 XML 웨어하우스에서 차원 계층 간의 참조 무결성 제약조건에 대한 형식 모델을 제안하였다. 본 논문은 여기서 한 걸음 더 나아가 XML 큐브를 만들고 다차원 질의를 할 수 있는 분석 방법에 관하여 연구한다.

본 논문의 구조는 다음과 같다. 제 2 장에서는 웹 페이지로부터 데이터를 추출하기 위한 추출 규칙에 대하여 논한다. 제 3 장에서는 XML 문서의 다차원 분석 방법에 대하여 논한다. 제 4 장에서는 미국 특허 웹사이트로부터 특허 정보를 추출하고 이를 다차원적으로 분석하는 예를 보이고 한계점을 논한다. 마지막으로 제 5 장에서는 결론을 맺는다.

2. 추출 규칙

본 논문에서는 추출 규칙을 작성하기 위한 두 개의 언어를 제안한다. 하나는 추출 규칙 명세 언어인 xRule이고, 다른 하나는 HTML 태그 시퀀스에 대한 정규 언어인 TagRex이다. 본 논문에서는 웹 페이지를 하나의 HTML 태그 시퀀스로 간주하고 xRule 속의 TagRex 정규식과 매치되는 부분을 추출한다.

2.1 추출 규칙 명세 언어: xRule

xRule (eXtraction Rule)은 추출할 웹 정보를 객체지향 모델에 의거하여 기술한다. 한 xRule 문장은 어떤 객체의 한 애트리뷰트 값에 대한 추출 규칙을 나타내며, 그림 1과 같은 기본 형식을 가진다. 먼저 'CONTEXT'란 키워드 다음에 객체의 클래스 이름이 온다. 그리고 할당 연산자 '=' 의 왼편에는 애트리뷰트 이름이, 오른편에는 TagRex 표현식이 온다.

```
CONTEXT: 클래스이름
속성이름 = HTML 태그 시퀀스에 대한 TagRex 표현식
```

그림 1. xRule 문장의 기본 구조

TagRex 표현식의 결과 값을 애트리뷰트의 값으로 할당하라는 의미이다.

변수: xRule은 변수를 가질 수 있다. 그림 2는 하나의 변수를 정의하는 문장으로서 TagRex 표현식의 결과 값을 변수에 저장한다. xRule의 모든 변수는 전역변수로서 모든 xRule 문장에서 사용될 수 있다.

반복문: xRule은 두 종류의 반복문을 가질 수 있다. 하나는 Array 타입을 가진 애트리뷰트를 위한 것이고, 다른 하나는 같은 클래스의 여러 다중 인스턴스 객체를 위한 것이다. 그림 3은 하나의 애트리뷰트에 Array 값을 할당하는 반복문의 예이고, 그림 4는 여러 개의 다중 인스턴스 객체 각각의 애트리뷰트 값을 할당하는 반복문의 예이다. 또한 그림 5는 반복문 속에 또 반복문을 가지는 중포 루프 반복문의 예를 보여 주고 있다.

시작위치 지정: 시작위치 지정문 'set location'을 통해 매칭의 시작 위치를 변경할 수 있다. 시작 위치를 지정하지 않으면 웹 페이지의 처음부터 매칭이 항상 시작된다. 그림 6은 시작위치 지정문의 형식을 보여주고 있다. 키워드 'set location' 오른편의 TagRex 표현식과 매치되는 곳이 시작 위치이다. 시작위치 지정문이 실행되고 난 이후부터는 지정된 위치부터 매칭이 항상 시작된다.

본 절에서 기술한 xRule은 배우기 쉽고 간단할 뿐만 아니라 Array, 다중 인스턴스, 반복문 등의 고급 기능을 가지고 있으며 객체지향 모델과 잘 부합된다. 또한, xRule로 기술된 추출 규칙들은 서로 독립적이므로 병렬 매칭(parallel matching)이 가능하다는 장점이 있다.

```
변수이름 = HTML 태그 시퀀스에 대한 TagRex 표현식
```

그림 2. 변수의 정의

```
CONTEXT: 클래스이름
for 변수이름 repeat
    속성이름 = TagRex 표현식
end
```

그림 3. Array 타입 애트리뷰트를 위한 반복문

```
for 변수이름 repeat
    CONTEXT: 클래스이름 속성이름 = TagRex 표현식
end
```

그림 4. 여러 개의 다중 인스턴스 객체들을 위한 반복문

```
for 변수이름 repeat
    for 변수이름 repeat
        CONTEXT: 클래스이름 속성이름 = TagRex 표현식
    end
end
```

그림 5. 중포 루프 반복문

```
set location = TagRex 표현식
```

그림 6. 시작위치 지정문

2.2 HTML 태그 시퀀스에 대한 정규식 언어: TagRex

본 절에서는 HTML 태그 시퀀스에 대한 탐색 패턴을 정규식(regular expression)으로 기술할 수 있는 언어인 TagRex에 대하여 논한다. TagRex 정규식은 xRule 문에 포함되어 사용된다. 표 1은 TagRex 정규식의 종류를 보여주고 있다. TagRex 정규식은 HTML 태그 시퀀스를 대상으로 하므로 문자열을 대상으로 하는 문자열 정규식과는 다르다. 즉, TagRex 정규식에서는 매칭을 위한 기본 단위가 문자가 아니고 HTML 태그이다. HTML 문서 내의 텍스트 세그

표 1. TagRex 정규식

정규식	의미
#...#	TagRex 정규식의 구분 표시자
#<a>~<b>#	<a>로 시작하여 <b>로 끝나는 HTML 태그 시퀀스와 매치
#<a>~!<b>#	<a>로 시작하여 <b> 직전까지의 HTML 태그 시퀀스와 매치
#<a> ~!(<b>|<c>)#	<a>로 시작하여 <b> 또는 <c> 직전까지의 HTML 태그 시퀀스와 매치
#<a>%<b>#	<a>로 시작하여 <b>로 끝나는 HTML 태그 시퀀스 중에서 <a>와 <b> 사이의 HTML 태그 시퀀스를 반환
#<a>%!<b>#	<a>로 시작하여 <b> 직전까지의 HTML 태그 시퀀스 중에서 <a>와 <b> 사이의 HTML 태그 시퀀스를 반환
$...$	문자열 정규식
#<a>@href=$...$#	<a>의 애트리뷰트 href 값에 대한 문자열 정규식

먼트도 하나의 태그로 간주한다. 그러면 하나의 HTML 문서는 하나의 HTML 태그 시퀀스를 이룬다. TagRex 정규식은 식별자 '#'으로 구분하며, 문자열 정규식은 식별자 '$'로 구분한다.

임의 서브시퀀스: TagRex 정규식에서 기호 '~'는 임의의 HTML 태그 서브시퀀스와 매치될 수 있음을 가리킨다. 그림 7은 임의 서브시퀀스가 포함된 한 예이다. 여기에서 '<tr><td>~ </td><td><b>'는 '<tr><td>'로 시작하여 '</td><td><b>'로 끝나는 어떠한 HTML 태그 시퀀스와도 매치된다.

반환: TagRex 정규식에서 기호 '%'는 TagRex 정규식과 매치된 HTML 태그 시퀀스 중에서 반환할 부분을 가리킨다. 반환 결과는 xRule 문에서 한 객체의 애트리뷰트 값으로 할당된다. 그림 8은 두 개의 예를 보여주는데 하나는 숫자를 반환하는 예이고, 다른 하나는 텍스트를 반환하는 예이다.

논리합: TagRex 정규식에서 기호 '|'는 논리합을 의미한다. 즉, 같은 내용에 대하여 서로 다른 용어를 사용할 경우 TagRex 정규식에서는 이들을 논리합으로 표현한다. 그림 9의 예에서 '
($DESCRIPTION$|$EMBODIMENT$)
'의 의미는 '
' 다음에 'DESCRIPTION' 또는 'EMBODIMENT'가 포함된 단어가 오고 그 다음에 '
'이 오는 HTML 태그 시퀀스와 매치된다는 것이다.

문자열 정규식: 웹 페이지 내에는 많은 텍스트 세그먼트들이 존재한다. 이들을 위하여 TagRex는 문자열에 대한 기존의 정규식을 허용하며 TagRex 정규식과 구분하기 위하여 식별자로 기호 '$'를 사용한다. 그림 10의 예에서는 클레임 텍스트 내에서 클레임 번호를 추출하기 위하여 문자열 정규식을 사용하고 있다.

태그 애트리뷰트: HTML 태그는 애트리뷰트를 가질 수 있으며 애트리뷰트의 값은 문자열이다. TagRex 정규식에서는 기호 '@'를 사용하여 HTML 태그 애트리뷰트를 표시한다. 그림 11의 예에서는 태그 <a>의 애트리뷰트 'href'에 들어 있는 특허번호를 추출한다.

본 절에서는 HTML 태그 시퀀스를 위한 정규 언어 TagRex에 관하여 기술하였다. TagRex 정규식을 이용하면 문자열이 아닌 HTML 태그 시퀀스에 대한 탐색 패턴을 기술할 수 있다.

3. XML 문서에 대한 다차원 분석

3.1 XML 웨어하우스

웹에서 추출한 XML 문서들을 분석하기 위해 XML 웨어하우스를 구축한다. 본 논문에서는 XML 웨어하우스를 위한 다차원 데이터 모델로서 그림 12와 같은 스타 스키마를 가정하고 이를 ***XML 스타 스키마*** (***XML Star Schema***)라고 부른다. XML 스타 스키마에는 사실 데이터를 이루는 XML 문서 집합이 한 개, n 개의 차원 데이터를 이루는 XML 문서 집합이 n 개 존재한다. 사실 데이터와 차원 데이터가 모두 XML 문서이므로 XML을 지원하는 모든 DBMS를 이용하여 XML 웨어하우스를 구축할 수 있다.

기존의 관계형 데이터 웨어하우스에서 사용되는 스타 스키마에서는 분석의 대상이 되는 사실 데이터가 단순한 측정값들(simple measure values)이지만,

```
CONTEXT: RegisteredOn
date = #<tr><td>~</td><td><b>%</b></td></tr>#
```

그림 7. 임의 서브시퀀스 예

```
CONTEXT: Patent
no = #<td><b>%</b></td>#

CONTEXT: Abstract
text = #<center><b>Abstract</b></center><p>%</p>#
```

그림 8. 반환 예

```
CONTEXT: Details
text = #<br>($DESCRIPTION$|$EMBODIMENT$)<br>%!<br><center><b>#
```

그림 9. 논리합의 예

```
_claims = #<CENTER><B><I>$Claims$</B></I></CENTER><HR>%<HR>#

for _claims repeat
    CONTEXT: Claim number = $([0-9]+)\.$ text = $(?i)(.*?)(?=(?:<BR><BR>[0-9]+\.|\z))$
end
```

그림 10. 문자열 정규식 예

```
CONTEXT: Patent
referencedByPatents = #<center><b>References Cited<a>@href=$.*(?=ref/)ref/(.*)$#
```

그림 11. 태그 애트리뷰트 예

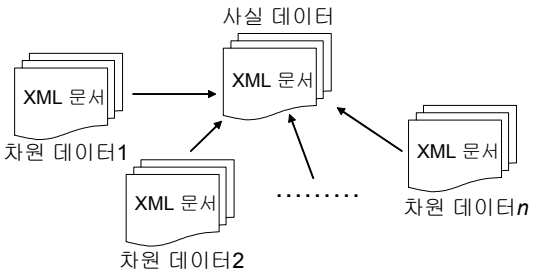


그림 12. XML 문서 분석을 위한 다차원 데이터 모델

XML 스타 스키마에서는 사실 데이터가 XML 문서들이다. 즉, 분석을 원하는 XML 문서 집합이 그대로 사실 데이터를 이룬다. XML 문서는 내용이 계층적 트리 구조를 가지고 있을 뿐만 아니라 텍스트 데이터를 포함하고 있다. 그리고 각 차원별로 그 차원 데이터를 이루는 XML 문서들이 존재한다. 하나의 XML 문서는 그 차원의 계층 구조의 한 인스턴스이다. 즉, 그 차원의 각 계층이 한 XML 요소(element)에 대응된다. 예를 들면 그림 13은 'regtime' 차원의 한 인스턴스이고 'year→month→day' 의 계층 구조를 가지고 있다. XML 문서의 최하위 요소에는 이 차원 데이터와 연관된 사실 데이터를 가리키는 포인터가 존재한다. 그림 13의 경우 이 차원 데이터와 연관된 사실 데이터는 '6,820,082.xml' 파일에 저장된 XML 문서이다.

이러한 차원 데이터는 차원 데이터 템플릿(template)에 의하여 사실 데이터로부터 자동 생성된다. 차원 데이터 템플릿은 그림 14와 같이 사용자가

```
<regtime>
   <year num=2004 >
      <month num=11 >
         <day num=16 >
            <factdocument> 6,820,082.xml </factdocument>
         </day>
      </month>
   </year>
</regtime>
```

그림 13. 차원 데이터 예

```
:factCollection = "/db/uspatent"
:date = #/uspatent/registeredon/date/text()#
<regtime>
   <year num=:regex(:date, ".*,\s+([0-9]+)") >
      <month num=:regex(:date, "([A-Za-z]+)") >
         <day num=:regex(:date, "([0-9]+),") >
            <factdocument> :factDocumentName() </factdocument>
         </day>
      </month>
   </year>
</regtime>
```

그림 14. 차원 데이터 템플릿 예

기술하는 것으로서 차원 계층 구조에 대응되는 각 요소마다 사용자 정의 함수를 명시하여 사실 데이터로부터 값을 추출한다. 사용자 정의 함수와 변수 이름은 구분을 위해 ':' 문자로 시작한다. '#' 문자로 둘러싸인 부분은 XPath 표현식을 나타내며, "" 문자로 둘러싸인 부분은 단순한 문자열 상수를 나타낸다. ':factCollection' 변수는 사실 데이터가 저장되어 있는 위치를 가리키며 사실 데이터를 이루는 XML 문서 각각에 대해 템플릿을 적용하여 차원 데이터를 생성한다.

3.2 XML 큐브

XML 웨어하우스의 사실 데이터는 XML 문서들이므로 XML 웨어하우스로부터 XML 큐브를 만들기 위해서는 XML 문서의 통합연산(aggregation)이 요구된다. 그런데, XML 문서는 계층 구조를 가진 복합 객체이므로 XML 문서에 대한 통합연산은 정의하기가 어렵다. 그러나 문서의 내용 중에 있는 숫자나 텍스트 데이터에 대한 통합연산은 정의하기가 쉽다.

본 논문에서는 XML 큐브를 만들 때 XML 문서의 한 부분을 XQuery 식을 이용하여 기술하고 이를 측정치로 하는 XML 큐브 개념을 제안한다. 본 논문에서는 이를 *XQ-Cube* 라 부른다. XQ-Cube에서 XQuery 식의 결과가 수치 데이터이면 XQ-Cube는 기존의 관계형 큐브가 되고, 텍스트 데이터이면 텍스트 큐브가 된다. 텍스트 큐브가 되면 텍스트 데이터에 대한 통합 연산이 필요하다. 본 논문에서는 이를 위해 텍스트 마이닝(text mining) 연산을 도입한다.

XQ-Cube는 다음과 같은 특징을 가진다. (1) XQuery 식을 이용하여 측정치를 기술하므로 다양한 종류의 XML 큐브를 만들 수 있다. (2) 측정치가 XML 문서의 일부이므로 데이터 타입에 따라 여러 가지 통합 연산을 적용할 수 있다. (3) XQ-Cube는 XQuery 식의 결과 값에 따라 기존의 수치 타입의 데이터 큐브가 될 수도 있고 텍스트 큐브가 될 수도 있으며 그 외 다른 타입의 데이터 큐브가 될 수도 있다.

3.3 XML 다차원 분석 질의어

데이터 큐브에 대한 질의를 하기 위해서는 다차원 질의어가 필요하다. 관계형 큐브를 위한 다차원 질의

어로서 마이크로소프트가 제안한 MDX (Multidimensional Expression Language) 언어가 있다[13]. 본 논문에서는 MDX를 확장한 다차원 질의어로서 *XML-MDX*를 제안한다. XML-MDX는 두 가지 명령문을 가진다. 하나는 XQ-Cube를 생성하기 위한 CREATE XQ-CUBE 문이고, 다른 하나는 질의를 하기 위한 SELECT 문이다.

CREATE XQ-CUBE 문: 그림 15는 CREATE XQ-CUBE 문의 기본 구조를 보여 주고 있다. <XQ-Cube name>은 생성할 XQ-Cube의 이름을 명시한다. CREATE XQ-CUBE 문은 FROM 절과 WHERE 절로 구성된다. 생성된 XQ-Cube는 나중의 사용을 위해 저장된다.

FROM 절은 XQ-Cube의 생성시 사용될 측정치를 명시한다. 그림 16은 BNF 표기법에 따른 FROM 절의 정의를 보여 주고 있다. <XQ-Cube_specification>은 XQuery 식을 이용한 측정치를 명시한다. 이때, 측정치의 데이터 타입에 따라 적절한 통합 연산자를 지정해 준다.

본 논문에서는 모두 7개의 통합 연산자를 다룬다. 즉, 'ADD', 'LIST', 'COUNT', 'SUMMARY', 'TOPIC', 'TOP KEYWORDS' 그리고 'CLUSTER'이다. 이 중 'ADD' 연산자는 수치 데이터를 위한 것이고, 나머지 연산자들은 모두 비수치 데이터를 위한 것이다. 'LIST' 연산자는 측정치를 모두 나열하라는 것이고, 'COUNT'는 측정치의 개수를 구하는 것이며 나머지는 모두 텍스트 마이닝 연산자들이다. 'SUMMARY', 'TOPIC', 'TOP KEYWORDS' 는 텍스트의 요약, 주제, 주요 키워드를 각각 뽑는 것이고, 'CLUSTER'는 전체 텍스트의 군집(cluster)을 구하는 것이다.

WHERE 절은 선택적인데, 절단(slicing)에 사용될 차원의 멤버(member)를 지정한다. 즉, 지정된 차원의 멤버에 대해서 XQ-Cube를 절단한다. 그림 17은 BNF 표기법으로 명시한 WHERE 절의 정의이다. <slicer_specification>은 절단자(slicer)를 명시하는데 XQuery 식의 튜플(tuple)로서 명시한다. 튜플 내의 각 XQuery 식은 차원의 멤버를 지정한다. <slicer_specification>에 명시되지 않은 나머지 차원들은 XQ-Cube의 축이 된다.

SELECT 문: 그림 18은 SELECT 문의 구조를 보여 주고 있다. SELECT 문은 MDX의 SELECT 문과 같이 SELECT, FROM, WHERE 절을 가진다. FROM 절은 CREATE XQ-CUBE 문을 통해 생성된 XQ-Cube의 이름을 가리킨다.

SELECT 절은 SELECT 결과 큐브의 축을 명시한다. 그림 19는 BNF 표기법으로 명시한 SELECT 절의 정의를 보여 주고 있다. 각각의 <axis_specification>이 하나의 축을 명시한다. XML 웨어하우스가 가진 차원의 개수가 축의 최대 개수이다. 하나의 <axis_specification>은 여러 개의 XQuery 식과 축의 이름으로 구성된다. XQuery 식의 결과값들은 그 축의 멤버를 이룬다. 즉, 한 축을 구성하는 각 멤버마다 하나의 XQuery 식이 존재한다. 각 축은 축 번호를 가지며 축의 이름은 MDX와 동일한 방법으로 정해진다. 즉, X-축은 0, Y-축은 1, Z-축은 2 등이다. <index>는 축 번호를 가리킨다. 처음 5개의 축(AXIS(0), AXIS(1), AXIS(2), AXIS(3), 그리고 AXIS(4))에 대해서는 COLUMNS, ROWS, PAGES, SECTIONS, CHAPTERS 등의 별명을 각각 사용할 수 있다.

```
CREATE XQ-CUBE  <XQ-cube name>
FROM  <XQ-cube specification>
[ WHERE  <slicer specification> ]
```

그림 15. CREATE XQ-CUBE 문의 구조

```
<FROM_clause> ::= FROM <XQ-cube_specification>
<XQ-cube_specification> ::= <XQuery_expression> : <aggregation_operator>
<aggregation_operator> ::= ADD | LIST | COUNT | SUMMARY | TOPIC |
                           TOP KEYWORD | CLUSTER
```

그림 16. FROM 절의 구조

```
<WHERE_clause> ::= WHERE <slicer_specification>
<slicer_specification> ::= "(" <XQuery_expression> { "," <XQuery_expression> } ")"
```

그림 17. WHERE 절의 구조

```
SELECT  <axis 0 specification>,
        <axis 1 specification>,
        ...
FROM  <XQ-Cube name>
[ WHERE  <slicer specification> ]
```

그림 18. SELECT 문의 구조

```
<SELECT_clause> ::= SELECT <axis_specification> { "," <axis_specification> }
<axis_specification> ::= <XQuery_expression_set> ON <axis_name>
<XQuery_expression_set> ::= "{" <XQuery_expression> { "," <XQuery_expression> } "}"
<axis_name> ::= COLUMNS | ROWS | PAGES | SECTIONS | CHAPTERS |
                AXIS(<index>)
```

그림 19. SELECT 절의 구조

SELECT 문의 WHERE 절의 정의는 CREATE XQ-CUBE 문의 WHERE 절과 동일하다. <slicer_specification>은 FROM 절에 명시된 XQ-Cube를 절단한다. 그리고 SELECT 절과 <slicer_specification>에 명시되지 않은 차원은 최상위 멤버인 'ALL' 값으로 절단한다.

XML-MDX는 마이크로소프트 MDX에 비해 다음과 같은 장점을 가진다. (1) XQuery 식만 사용하므로 배우기가 쉽고 그 처리도 기존의 XQuery 엔진을 그대로 이용할 수 있다. (2) 축과 절단자를 명시할 때 마이크로소프트 MDX는 차원 계층구조의 경로식만 명시할 수 있는 반면에 XML-MDX는 조건식을 사용할 수 있다.

4. 웹 정보 추출 및 다차원 분석 시스템

4.1 웹 정보 추출 서브시스템

웹 정보 추출 서브시스템의 아키텍춰는 그림 20과 같으며, RME(Rule Matching Engine)와 XDG(XML Document Generator)라는 두 개의 주요 모듈로 구성되어 있다.

RME는 사용자로부터 xRule 언어로 기술된 추출 규칙을 입력받아 컴파일한다[1]. HTML 파서는 웹 크롤러가 수집한 웹 페이지를 HTML 태그 시퀀스 구조로 바꾸어 RME로 넘겨준다. RME는 추출 규칙과 HTML 태그 시퀀스를 매치시켜 추출 규칙에 명시된 대로 웹 정보를 추출한다. RME는 추출된 웹 정보를 XDG에게 전달하며, XDG는 이를 XML 문서로 변환한다. 변환된 XML 문서는 XML 데이터베이스에 저장된다.

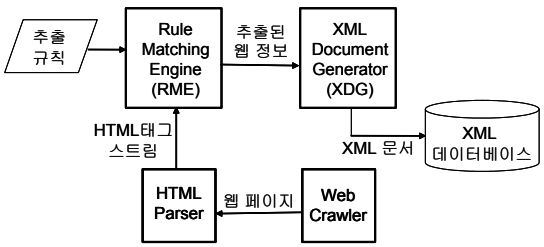


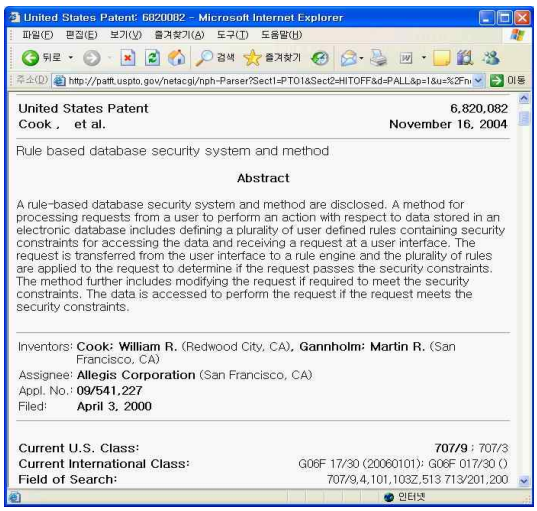
그림 20. 웹 정보 추출 서브시스템 아키텍춰

1) 추출 규칙은 텍스트 형태로 되어 있으므로 효율적인 처리를 위해 메모리 데이터 구조로 컴파일을 한다.

TagRex 정규식에 의존한 추출 규칙을 사용하여 웹 페이지에 내재된 정보를 추출하려면 각 웹 페이지의 구조가 같아야 한다. 이는 동일한 스키마를 가진 대량의 웹 페이지를 대상으로 할 경우 효과적인 반면, 웹 페이지의 스키마가 모두 다르거나 자주 바뀌는 경우 적용하기가 어렵다. 따라서 특허 웹 페이지, 인터넷 서적 정보 웹 페이지, 인터넷 쇼핑몰 상품 정보 웹 페이지 등 동일한 양식을 가진 전자문서 형태의 웹 페이지를 대량으로 제공하는 웹 사이트의 경우 적용이 효과적이다.

4.2 웹 정보 추출 예

본 절에서는 제 2 장에서 기술한 추출 규칙 언어를 이용하여 미국특허청, YES24 인터넷 서점, G마켓 인터넷 쇼핑몰 등 다양한 종류의 웹 페이지로부터 원하는 정보를 추출할 수 있음을 보인다. 그림 21은 미국특허 웹 페이지[14]의 한 예이며, 그림 22는 이로부터 추출된 특허정보를 XML 문서로 변환한 예를 보여주고 있다. 미국특허 웹 페이지는 특허번호[2] 등의 구조화된 데이터와 요약문(abstract)[3] 등의 비구조화된 데이터(텍스트 데이터)를 모두 가지고 있다.


United States Patent: 6820082 - Microsoft Internet Explorer

United States Patent 6,820,082
Cook , et al. November 16, 2004

Rule based database security system and method

Abstract

A rule-based database security system and method are disclosed. A method for processing requests from a user to perform an action with respect to data stored in an electronic database includes defining a plurality of user defined rules containing security constraints for accessing the data and receiving a request at a user interface. The request is transferred from the user interface to a rule engine and the plurality of rules are applied to the request to determine if the request passes the security constraints. The method further includes modifying the request if required to meet the security constraints. The data is accessed to perform the request if the request meets the security constraints.

Inventors: Cook; William R. (Redwood City, CA), Gannholm; Martin R. (San Francisco, CA)
Assignee: Allegis Corporation (San Francisco, CA)
Appl. No.: 09/541,227
Filed: April 3, 2000

Current U.S. Class: 707/9 ; 707/3
Current International Class: G06F 17/30 (20060101); G06F 017/30 ()
Field of Search: 707/9,4,101,103Z,513 713/201,200

그림 21. 미국특허 웹 페이지

2) 특허번호 추출 규칙: CONTEXT: Patent
no = #<td>%</td>#

3) 요약문 추출 규칙: CONTEXT: Abstract
text = # Abstract % #

그림 23은 국내 인터넷 서점 사이트인 'YES24'에서 판매되고 있는 한 도서의 정보를 담고 있는 웹 페이지의 예이며, 그림 24는 이로부터 추출된 도서정보를 XML 문서로 변환한 예를 보여주고 있다. 도서정보에는 도서명, 저자명, 출판사명, 출판일, 가격, 분류, 서문, 목차 등의 정보가 포함되어 있다.

그림 25는 국내 인터넷 쇼핑몰인 'G마켓'에서 판매되고 있는 한 상품의 정보를 담고 있는 웹 페이지의 예이며, 그림 26은 이로부터 추출된 상품정보를 XML 문서로 변환한 예를 보여주고 있다. 상품정보에는 상품번호, 상품명, 가격, 사은품, 제조사, 배송비, 상품상태, 판매자 등에 관한 정보가 포함되어 있다.

위에서 예로 든 미국 특허정보, YES24 도서정보, G마켓 상품정보 등의 웹 페이지들은 모두 같은 특징을 가지고 있다. 즉, 수많은 미국 특허정보 웹 페이지들은 그 구조가 모두 동일하다는 것이며, 이는 YES24 도서정보, G마켓 상품정보 웹 페이지들도 마

```
<uspatent>
  <title>
    <text> Rule based database security system and method </text>
  </title>
  <abstract>
    <text> A rule-based database security system and method are disclosed. A method ... </text>
  </abstract>
  <inventor>
    <name> Cook; William R. </name>
    <addr> Redwood City, CA </addr>
  </inventor>
  <patent>
    <no> 6,820,082 </no>
    <applNo> 541227 </applNo>
  </patent>
  <registeredOn> <date> November 16, 2004 </date> </RegisteredOn>
  <filedOn> <date> April 3, 2000 </date> </FiledOn>
  <claim>
    <number> 1 </number>
    <text> A method for processing requests from a user to perform an act ... </text>
  </claim>
</uspatent>
```

그림 22. 추출된 미국특허 XML 문서 예

그림 23. YES24 서적 정보 웹 페이지

```
<book>
  <title> 대한민국 웹2.0 트렌드 </title>
  <authorList> 김상범, 이희욱, 황치규, 도안구 등저 </authorList>
  <press> 행복한 나무 </press>
  <date> 2007년 9월 </date>
  <price>
    <retail> 10,000원 </retail>
    <selling> 9,000원 </selling>
  </price>
  <class> 국내도서
    <subclass> 비즈니스와 경제
      <subsubclass> 마케팅/세일즈
        <subsubsubclass> 인터넷 마케팅 </subsubsubclass></subsubclass></subclass>
  </class>
  <introduction>
    나이 30세인 오대리는00무역회사 2년차 대리이다. 인터넷 활용수준 중하 정도. 포탈사이트...
  </introduction>
  <authors>
    <author> <name> 김상범 </name>
      현 블로터닷넷 기자 4명이 공동 집필했다. 김상범 대표블로터를 포함해 ...
    </author>
  ...
</book>
```

그림 24. 추출된 YES24 서적 정보 XML 문서 예

그림 25. G마켓 상품 정보 웹 페이지.

```
<item>
  <number> 125339207 </number>
  <name> [디지털큐브 [가격비교보다싸다!! 359,000]] [단골찬스]아이스테이션 PMP M43 스탠다드(30GB)+추가베터리+가죽케이스+액정필름+이어폰홀더+클리너/DMB/ </name>
  <price>
    <retail> 428,000 </retail>
    <selling> 386,020원 </selling>
  </price>
  <gift> 추가베터리(블루블랙)+가죽케이스+액정필름+이어폰홀더+클리너 </gift>
  <maker> 디지털큐브/국내 </maker>
  <delivery> 무료 </delivery>
  <status> 새상품 / 터치스크린:터치스크린 가능 / DMB지원:DMB지원 가능 </status>
  <seller>
    <name> (와앱피나비/최정민 </name>
    <phone> ** - *** - **** </phone>
    <e-mail> webmaster@headphones.co.kr </e-mail>
    <license> 106-86-11930 </license>
    <call> 용산-00549 </call>
    <addr> 서울 영등포구 양평동3가 15-1 월드메르디앙 비즈센터 201호,202호 </addr>
  </seller>
  <category> 디카/MP3/게임/사전
    <subcategory> PMP
      <subsubcategory> 아이스테이션 </subsubcategory>
    </subcategory>
  </category>
</item>
```

그림 26. 추출된 G마켓 상품 정보 XML 문서 예

찬가지이다. 따라서 동일한 구조를 가진 웹 페이지들로부터 원하는 정보를 추출하는 경우에는 본 논문에서 제안하는 추출 규칙 언어가 안정적으로 동작함을 알 수 있다.

4.3 다차원 분석 서브시스템

그림 27은 XML 웨어하우스 기반 다차원 분석 서브시스템의 아키텍춰를 보여주고 있다. 이 시스템은 'eXist' 라는 XML DBMS를 이용하여 XML 웨어하우스를 저장 관리하며 '차원데이터생성기'와 'XML-MDX처리기' 라는 두 개의 주요 모듈로 구성되어 있다. 차원데이터생성기는 XML 웨어하우스에 저장되어 있는 XML 문서들의 분석에 필요한 차원 데이터를 생성하는 모듈이다. 제 3.1 절에서 언급한대로 차원 데이터 템플릿을 모든 사실 데이터에 적용시켜 차원 데이터를 생성한다. XML-MDX처리기는 제 3.3 절에서 설명한 XML-MDX 질의를 처리하는 모듈이다. XML-MDX 질의를 분석하여 XQ-Cube를 생성하거나 XQ-Cube를 처리한 결과를 보여 준다.

본 논문에서는 XML 문서를 다차원적으로 분석하기 위하여 먼저 XML 웨어하우스를 구축하는데 이때, 차원 데이터는 차원데이터 템플릿에 의해 사실 데이터로부터 자동 생성된다. 그러나 분석을 위한 차원 데이터에는 사실 데이터 이외의 외부 소스로부터 가져와야 하는 것도 있다 (예, 지역 코드, 제품 분류기준 등). 본 논문의 모형에서는 이러한 차원 데이터들을 포함하지 못한다는 한계점을 가진다. 그리고 XML 웨어하우스로부터 XQ-Cube를 생성하고 다차원 질의를 할 때 사용되는 XML-MDX는 XQuery 식을 기반으로 하므로 XQuery 사양에 의존적이다. XQuery 사양이 발전하여 OLAP 질의 기능을 지원하게 될 경우 XML-MDX 구문 구조도 따라서 바뀌어야 한다.

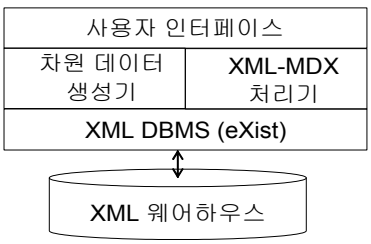


그림 27. XML 문서 다차원 분석 서브시스템 아키텍춰

4.4 XML 문서 다차원 분석 예

본 절에서는 미국특허 웹 사이트로부터 추출한 미국특허 문서들에 대한 다차원 분석의 예를 기술한다. 그림 28은 미국특허 분석에 사용할 네 개의 차원에 대한 계층 구조를 보여주고 있다. 모든 차원은 모두 최상위 멤버로서 'ALL' 을 가지고 있다. 차원 'Appl.Time' 과 'Reg.Time' 은 특허가 출원된 날짜와 등록된 날짜를 각각 나타낸다. 그들은 모두 'year' 와 'month' 라는 두 가지 수준을 가진다. 차원 'Inventor' 는 특허 발명자를 나타내며 'Institution Type', 'Institute', 그리고 'Inventor' 의 세 가지 수준을 가진다. 차원 'Topic' 은 특허의 주제를 나타내며 'High', 'Middle', 그리고 'Low' 의 세 가지 수준을 가진다.

그림 29는 'Appl.Time' 차원에 대한 XML 문서의 예를 보여 주고 있다. 출원 연도가 1998년도이고, 출원 월은 3월과 9월이다.

그림 30은 'Inventor′ 차원에 대한 XML 문서의 예를 보여 주고 있다. 발명자 이름은 'Il-Yeol Song′ 이고 소속된 기관 이름은 'Drexel′이며 기관 타입은 'university′ 이다.

그림 31은 'Topic′ 차원에 대한 XML 문서의 예를 보여 주고 있다. 최상위 수준의 분야는 'software' 이

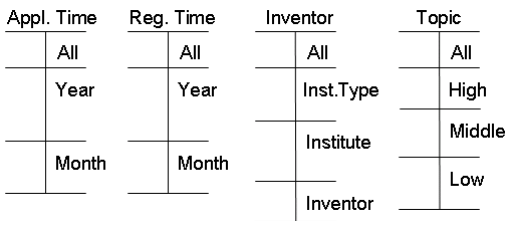


그림 28. 차원 계층 구조

```
<year num = "1998">
   <month num = "3" name = "Mar." />
   <month num = "9" name = "Sep." />
</year>
```

그림 29. Appl.Time 차원 데이터 XML 문서

```
<instType name = "university" code = "001">
   <institute name = "Drexel" addr = "Philadelphia, PA">
      <inventor name = Il-Yeol Song" addr = "Philadelphia, PA" />
   </institute>
</instType>
```

그림 30. Inventor 차원 데이터 XML 문서

```
<high area = "software">
   <middle area = "database">
     <low area = "model" />
     <low area = "language" />
   </middle>
   <middle area = "AI">
      <low area = "Vision" />
   </middle>
</high>
```

그림 31. Topic 차원 데이터 XML 문서

고, 중간 수준의 분야는 'database' 와 'AI' 이다. 'database' 에 대한 하위 수준의 분야는 'model' 과 'language' 이고, 'AI' 에 대한 하위 수준의 분야는 'Vision' 이다.

그림 32는 XQ-Cube를 생성하는 XML-MDX 문의 예를 보여 주고 있다. 생성할 XQ-Cube의 이름은 'XQ-Cube-1' 이다. FROM 절의 XQuery 식은 'XQ-Cube-1'의 측정치를 명시하고 있다. 즉, '/db/uspatent' 라는 collection에 있는 XML 문서들의 '//patent/no' 의 개수를 측정치로 한다. 통합 연산자 'COUNT' 는 '//patent/no' 의 개수를 통합한다. WHERE 절은 절단자를 명시하고 있으며 'Appl.Time' 차원에 대해서는 'ALL', 'Reg.Time' 차원에 대해서는 '2000' 보다 큰 'year' 만 선택하고 나머지는 버린다.

그림 33은 'XQ-Cube-1' 에 대한 XML-MDX 질의문의 예를 보여 주고 있다. 먼저 WHERE 절에 명시된 절단자에 의해 'XQ-Cube-1' 에서 'RegTime' 이 '2002' 보다 큰 'year' 만 선택되고 나머지는 버린다. 질의 처리 결과 반환될 큐브는 SELECT 절에 명시된 축을 가진다. COLUMNS는 'XML' 과 'OLAP' 이라는 두 개의 멤버를 가지고, ROWS는

```
CREATE XQ-CUBE  XQ-Cube-1
FROM   col('/db/uspatent')//patent/no : COUNT
WHERE  ( col('/db/applTime')/ALL,
           col('/db/regTime')//year[@num>2000] )
```

그림 32. XQ-Cube 생성 예.

```
SELECT { col('/db/topic')//high[@topic='XML'],
           col('/db/topic')//high[@topic='OLAP'] } ON COLUMNS
       { col('/db/inventor')//instType[@name='university'],
         col('/db/inventor')//instType[@name='industry'] } ON ROWS
FROM   XQ-Cube-1
WHERE  ( col('/db/regTime')//year[@num > 2002] )
```

그림 33. XML-MDX 질의 예

	XML	OLAP
university	126	435
industry	267	672

그림 34. 질의 결과 예

```
CREATE XQ-CUBE  XQ-Cube-2
FROM   col('/db/uspatent')//title/text : TOP KEYWORDS
WHERE  ( col('/db/applTime')/ALL,
           col('/db/regTime')//year[@num=2003],
           col('/db/regTime')//year[@num=2004] )
```

그림 35. XQ-Cube 생성 예

```
SELECT { col('/db/regTime')//year[@num=2003],
           col('/db/regTime')//year[@num=2004] } ON COLUMNS
       { col('/db/inventor')//instType[@name='university'],
         col('/db/inventor')//instType[@name='industry'] } ON ROWS
FROM   XQ-Cube-2
WHERE  ( col('/db/topic')//high[@area='AI'],
           col('/db/topic')//high[@area='database'] )
```

그림 36. XML-MDX 질의 예

	2003	2004
university	ML, Genome	XML, Sequence
industry	Robot, Vision	Grid, Stream

그림 37. 질의 결과 예

'university' 와 'industry' 라는 두 개의 멤버를 가진다. 그림 34는 그림 33의 질의를 처리한 결과 예를 보여 주고 있다.

그림 35는 측정치가 텍스트 데이터인 XQ-Cube를 생성하는 XML-MDX 문의 예를 보여 주고 있다. 생성할 XQ-Cube의 이름은 'XQ-Cube-2' 이고 측정치는 특허 제목의 주요 키워드이다. 그림 36은 'XQ-Cube-2' 에 대한 XML-MDX 질의문의 예를 보여 주고 있으며 그림 37은 질의 처리 결과를 보여 주고 있다.

5. 결 론

본 논문에서는 웹 페이지의 정보를 추출 규칙에 의해 XML 문서로 추출하고 XML 웨어하우스에 저장한 후 XML 큐브를 통해 다차원적으로 분석할 수

있는 방법을 제안하였다. 추출 규칙에 사용된 TagRex 정규식은 HTML 태그 시퀀스에 대하여 태그 단위의 정규식을 쉽게 기술할 수 있다는 장점을 가지고 있다. XML 웨어하우스 구축을 위해 사용된 차원데이터 템플릿은 사실 데이터로부터 차원 데이터를 자동 생성함으로써 웨어하우스 구축을 쉽게 해준다. XML 문서를 다차원적으로 분석하기 위해 XQ-Cube라는 새로운 타입의 XML 큐브를 정의하였고 XQ-Cube에 대한 다차원 질의어로서 XML-MDX를 제안하였다.

본 논문의 공헌은 인터넷 상의 방대한 웹 정보를 다차원적으로 분석할 수 있는 모형을 제시하였다는 것이다. 웹 브라우저를 통해 웹 서핑을 하는 것만으로는 방대한 웹 정보를 체계적으로 분석하거나 습득할 수 없다. 필요한 정보를 추출하여 데이터 웨어하우스를 구축하고 다차원적으로 분석할 수 있는 장치가 필요하다. 본 논문에서 제안한 XML 스타 스키마 모형과 XML-MDX 질의어는 XML 문서에 대한 다차원 분석의 새로운 형태를 제시하였다.

향후 연구로는 본 논문에서 제안한 웹 정보 추출 모형과 다차원 분석 모형의 성능을 평가하는 것이다. 웹 정보 추출은 성격상 배치(batch) 작업이므로 다차원 분석 모형에 대한 성능 평가에 초점을 맞출 것이다. 우선 본 논문에서 예로 보여준 미국특허 웹 사이트뿐만 아니라 인터넷 서점, 법률 및 판례, 전자 신문, 온라인 저널 등 다양한 분야의 웹 사이트에 적용하여 시스템의 안정성을 평가할 것이다. 그리고 XML 웨어하우스의 크기에 따라 XML 큐브 생성 시간과 XML-MDX 질의처리 시간을 측정하여 시스템의 성능을 평가할 것이다.

참 고 문 헌

[1] A. Arasu and H. Garcia-Molina, "Extracting Structured Data from Web Pages," In *Proc. Int'l Conf. on Management of Data*, ACM SIGMOD, pp. 337-348, 2003.

[2] C. Chang and S. Lui, "IEPAD: Information Extraction based on Pattern Discovery," In *Proc. Int'l Conf. on World Wide Web (WWW10)*, pp. 681-688, 2001.

[3] V. Crescenzi, G. Mecca, and P. Merialdo, "RoadRunner: Towards Automatic Data Extraction from Large Web Sites," In *Proc. Int'l Conf. on Very Large Data Bases*, pp. 109-118, 2001.

[4] D. W. Embley, D. M. Campbell, Y. S. Jiang, S. W. Liddle, D. W. Lonsdale, Y.-K. Ng, and R. D. Smith, "Conceptual-model-based data extraction from multiple-record Web pages," *Data & Knowledge Engineering*, Vol.31, No. 3, pp. 227-251, 1999.

[5] J. Hammer, H. Garcia-Molina, S. Nestorov, R. Yerneni, M. Breunig, and V. Vas-salos, "Template-Based Wrappers in the TSIMMIS System," In *Proc. Int'l Conf. on Management of Data*, ACM SIGMOD, pp. 532-535, 1997.

[6] A. Sahuguet and F. Azavant, "Looking at the-Web through XML glasses," In *Proc. IFCIS Intl Conf. on Cooperative Information Systems (CoopIS99)*, pp. 148-159, 1999.

[7] V. Crescenzi and G. Mecca, "Grammars Have Exceptions," *Information Systems*, Vol.23, No. 8, pp. 539-565, 1998.

[8] D. Pedersen, K. Riis, and T. B. Pedersen, "XML-Extended OLAP Querying," In *Proc. The 14th Intl Conference on Scientific and Statistical Database Management (SSDBM02)*, pp. 195-206, 2002.

[9] D. Pedersen, K. Riis, and T. B. Pedersen, "Query Optimization for OLAP-XML Federations," In *Proc. The 5th ACM Intl Workshop on Data Warehousing and OLAP (DOLAP02)*, pp. 57-64, 2002.

[10] M. Gofarelli, S. Rizzi, and B. Vrdoljak, "Data Warehouse Design from XML Sources," In *Proc. The 4th ACM Intl Workshop on Data Warehousing and OLAP (DOLAP01)*, pp. 40-47, 2001.

[11] V. Nassis, R. Rajugan, T. S. Dillon and W. Rahayu, "Conceptual Design of XML Document Warehouses," In *Proc. Data Warehousing and Knowledge Discovery, 6th International Conference, DaWaK 2004*, pp.

1-14, 2004.

[12] J. Pokorny, “Modelling Stars Using XML,” In *Proc. The 4th ACM Intl Workshop on Dara Warehousing and OLAP (DOLAP01)*, pp. 24-31, 2001.

[13] G. Spofford, *MDX Solutions with Microsoft SQL Server Analysis Services*, John Wiley & Sons, 2001.

[14] USPTO (United States Patent and Trademark Office), http://www.uspto.gov/

박 병 권

1986년 서울대학교 산업공학과 학사
1988년 KAIST 경영과학과 석사
1998년 KAIST 전산학과 박사
1998년~2000년 삼성전자 중앙연구소 선임연구원
2000년~현재 동아대학교 경영정보과학부 부교수
관심분야 : 데이터베이스, 데이터웨어하우스, OLAP, XML, 정보검색, e-비즈니스