경영정보학연구
제15권 제4호
2005년 12월

XML 웨어하우스에 대한 다차원 분석 프레임워크*

박 병 권**, 이 종 학***

A Multidimensional Analysis Framework for XML Warehouses

Byung-Kwon Park, Jong-Hak Lee

Nowadays, large amounts of XML documents are available in the Internet. Thus, we need to analyze them multidimensionally in the same way as relational data. In this paper, we propose a new framework for multidimensional analysis of XML documents, which we call XML-OLAP. We base XML-OLAP on XML warehouses where all fact and dimension data are stored as XML documents. We build XML cubes from XML warehouses. We propose a new OLAP language for XML cubes, which we call XML-MDX. XML-MDX statements target XML cubes and use XQuery expressions to designate measure, axis and slicer. They incorporate text mining operations for aggregating text data. We apply XML-OLAP to the United States patent XML warehouse to demonstrate multidimensional analysis of XML documents.

Keywords : OLAP, Multidimensional Analysis, XML Warehouses, XML cube, XML-MDX

* 이 논문은 한국과학재단의 해외 Post-doc. 연수지원에 의하여 연구되었음.
** 동아대학교 경영정보과학부
*** 대구가톨릭대학교 컴퓨터정보통신공학부

Ⅰ. 서 론

OLAP(Online Analytical Processing) 시스템은 의사결정 지원을 위한 강력한 데이터 분석 도구이다[Spofford, 2001]. 그것은 데이터 웨어하우스(data warehouse)에 있는 방대한 양의 데이터를 여러 각도(또는 차원)에서 분석할 수 있도록 해 준다. 일반적으로 데이터 웨어하우스는 하나의 큰 사실 테이블(fact table)과 여러 개의 작은 차원 테이블들(dimension tables)로 구성된다. 사실 테이블과 차원 테이블들은 대개 관계형 데이터베이스에 저장될 수 있는 구조화된 데이터(structured data)이다.

오늘날에는 인터넷 상에 많은 양의 XML 문서들이 존재한다. 따라서 기존의 관계형 데이터에 대한 방법과 동일하게 XML 문서들을 다차원적으로 분석하는 것이 필요하다. 그러나, XML 문서의 데이터 모델은 관계형 데이터와 달리 트리(tree) 구조를 가지고 있다. 뿐만 아니라, XML 문서는 텍스트(text)와 같은 비구조화된 데이터를 포함하고 있다. 따라서 XML 문서에 대한 새로운 다차원 분석 프레임워크(framework)가 필요하다.

본 논문에서는 이러한 다차원 분석 프레임워크를 제안하고 이를 XML-OLAP이라 부른다. XML-OLAP은 XML 웨어하우스를 기반으로 하는데 여기에는 모든 사실 데이터와 차원 데이터가 XML 문서로 저장되어 있다. 이에 대한 다차원 분석을 위해서는 다차원 큐브를 만들고 질의할 수 있는 질의어가 필요하다. 기존의 OLAP 질의어로서 Microsoft MDX[Spofford, 2001]가 널리 사용되고 있으므로 본 논문에서는 이를 확장한 XML-MDX를 제안한다. XML-MDX는 Microsoft MDX와 같은 구문 구조를 가지며 XQuery[XQuery, 2005]와 텍스트 마이닝을 도입한 언어이다. XQuery는 XML 문서의 구조를 기술하는데 사용되며 텍스트 마이닝은 XML 문서에 포함된 텍스트 데이터의 '요약'(summarization), '분류'(classification), '주요 키워드 추출'(top keyword extraction) 등과 같은 통합연산(aggregation)을 기술하는데 사용된다.

마지막으로 XML-OLAP의 효용성을 보이기 위하여 XML-OLAP을 미국 특허 웨어하우스에 적용한다. 미국 특허 웨어하우스는 미국 특허 웹사이트[USPTO]로부터 특허 정보를 추출하여 XML 문서로 바꾸고 이를 XML 데이터베이스에 저장하여 구축한다. 이를 통하여 XML 문서를 다차원적으로 분석하는 예를 보인다.

본 논문의 구조는 다음과 같다. 제Ⅱ장에서는 관련 연구를 살펴본다. 제Ⅲ장에서는 XML 웨어하우스에 대하여 논한다. 특히, 사실 데이터와 차원 데이터를 XML 문서로 표현하는 방법에 대하여 논한다. 제Ⅳ장에서는 XML 웨어하우스로부터 XML 큐브를 생성하는 방법과 XML-MDX를 이용하여 XML 큐브를 질의하는 방법에 대하여 논한다. 제Ⅴ장에서는 미국 특허 웨어하우스를 통하여 XML 문서에 대한 다차원 분석 예를 보인다. 마지막으로 제Ⅵ장에서는 결론을 맺는다.

Ⅱ. 관련 연구

XML과 OLAP의 결합에 관한 연구는 다음 세 가지로 분류할 수 있다. 첫째, 기존의 ROLAP 도구를 그대로 사용할 수 있도록 XML 데이터를 변환하는 연구이다. 둘째, 서로 독립적으로 존재하는 ROLAP과 XML 데이터를 연동(federation)시키는 연구이다. 셋째, XML 웨어하우스의 개념적 모델링에 관한 연구이다.

첫 번째 부류에 속하는 연구로는 Jensen[2001], Niemi[2001, 2002, 2003], Hummer[2003] 등이 있다. Jensen등은 인터넷 상의 XML 데이터를 관계형 데이터로 변환하여 기존의 OLAP 도구를 그대로 사용하는 방안을 제안하였다. Niemi 등도 사용자의 OLAP 질의를 분석하여 필요한 데이터를 인터넷 상에 분산된 데이터 웨어하우스들로부터 XML 형태로 가져와 OLAP 큐브를

만드는 시스템을 개발하였다. Hummer[2003]등은 여러 개의 데이터 웨어하우스를 통합하여 하나의 가상적인 데이터 웨어하우스를 만드는 문제를 연구하였다. 그들은 각 데이터 웨어하우스의 구조 데이터, 사실 데이터, 그리고 차원 데이터를 기술할 수 있는 일군의 XML 문서 템플릿을 제안하였다.

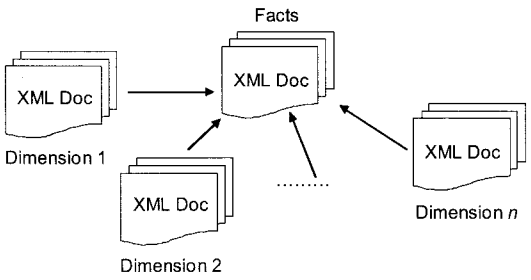
두 번째 부류에 속하는 연구로는 Pedersen[2002] 등의 연구가 대표적이다. Pedersen등은 OLAP 질의에 외부 XML 문서의 내용을 결합하여 OLAP 질의를 확대하는 문제를 연구하였다. 기존의 OLAP은 미리 정해진 차원 데이터를 통해서만 질의할 수 있으나 외부 XML 문서와 결합하면 보다 확장된 차원 데이터에 기반한 질의가 가능해 진다. 그러나, 첫 번째와 두 번째 부류의 연구들은 여전히 OLAP 질의의 대상이 XML 웨어하우스가 아닌 관계형 데이터 웨어하우스이다.

세 번째 부류에 속하는 연구로는 Pokorny[2001], Nassis[2004], Golfarelli[2001] 등이 있다. Pokorny 등은 사실 데이터와 차원 데이터가 모두 XML 문서로 기술된 XML 웨어하우스에서 차원 계층 간의 참조 무결성 제약조건에 대한 형식 모델을 제안하였다. Nassis등은 UML을 이용한 XML 웨어하우스의 개념적 모델 설계에 관하여 연구하였다. Golfarelli등은 XML 데이터로부터 개념 스키마를 자동적으로 찾는 문제를 연구하였다. 본 논문은 세 번째 부류의 연구에서 한 걸음 더 나아가 XML 웨어하우스에 대한 다차원 분석을 가능하게 하는 XML-OLAP에 대하여 연구한다.

Ⅲ. XML 웨어하우스

3.1 XML 웨어하우스 데이터 모델

본 논문에서 가정하는 XML 웨어하우스는 <그림 1>과 같은 다차원 모델을 가진다. 즉, 사실 데이터를 구성하는 하나의 XML 문서 집합이 존재하고, n개의 차원 데이터를 구성하는 n개의 XML 문서 집합이 존재한다.



<그림 1> XML 웨어하우스의 다차원 모델

사실 데이터를 구성하는 XML 문서 집합은 Niemi등이 가정한 것과 같이[Niemi, 2001] 한 개의 사실 데이터는 한 개의 XML 문서로 표현된다. 사실 데이터는 기존의 데이터 웨어하우스처럼 단순하지 않고 계층적 트리 구조를 가진다. 뿐만 아니라, 구조화된 데이터와 비구조화된 데이터를 모두 포함한다. 사실 데이터를 구성하는 XML 문서 집합은 분석을 원하는 XML 문서 집합을 그대로 사용하면 되므로 재구축할 필요가 없다.

하나의 차원 데이터를 구성하는 XML 문서 집합은 그 차원의 계층 구조를 반영하고 있다. 즉, 하나의 XML 문서는 최상위층 구성요소(member)를 루트로 하는 계층 구조의 한 인스턴스에 해당한다. 차원 데이터와 사실 데이터를 연관짓기 위하여 인덱스와 같은 보조 데이터 구조가 사용된다.

<그림 1>과 같은 다차원 모델은 다음과 같은 장점을 가진다: (1) 사실 데이터와 차원 데이터가 모두 XML 문서로 기술되므로 XML 웨어하우스를 쉽게 구축할 수 있다. 특히, 사실 데이터는 새로이 구축할 필요가 없다. (2) 사실 데이터와 차원 데이터를 XML 전용 데이터베이스(native XML database)에 저장하고 관리할 수 있다. (3) XML 문서의 계층 구조를 이용하여 차원 데이터의 계층 구조를 표현할 수 있다.

3.2 XML 웨어하우스 구축

본 논문에서는 주어진 기존의 XML 문서 집합을 사실 데이터로 사용하므로 차원 데이터를 구성하는 XML 문서의 생성에 초점을 맞춘다. 사실 데이터를 구성하는 XML 문서 집합이 주어지면 이를 분석하기 위한 차원을 결정하여야 한다. 이를 위해서는 주어진 XML 문서의 개념적 모델링이 필요하다.

UML을 이용하여 XML 데이터의 개념적 모델링을 한 연구가 많이 있다. Jensen등은[Jensen, 2001] XML 데이터의 DTD를 이용하여 자동적으로 UML 클래스 다이어그램을 생성하는 알고리즘을 제안하였다. Lujan-Mora등은 [Lujan-Mora, 2004] UML을 확장하면 다차원 모델링 언어가 될 수 있음을 보였다. 본 논문에서도 그들의 방법을 도입하여 XML 문서의 개념적 모델로 UML 클래스 다이어그램을 사용한다.

사실 데이터를 구성하는 XML 문서들의 개념적 모델을 통해 사실 데이터의 논리적 구조를 이해하고 분석을 위한 차원을 정한다. Nassis등은[Nassis, 2004] 사용자의 요구사항을 분석하여 차원을 정하고 XML 뷰를 이용하여 차원을 표현할 것을 제안하였다. 그들은 모든 차원이 사실 데이터 속에 포함되어 있다고 가정하였으나, 어떤 차원은 사실 데이터 밖에서 주어질 수도 있으므로 본 논문에서는 각 차원 데이터를 XML 뷰로 표현하지 않고 별도로 생성한다. 이때 각 차원 데이터와 사실 데이터의 연결은 색인을 통해 이루어진다고 가정한다.[1)]

Ⅳ. XML 웨어하우스의 다차원 분석 프레임워크

본 장에서는 XML 웨어하우스에 대한 다차원 분석 프레임워크를 기술한다. 분석 프레임워크는 크게 XML 큐브를 생성하는 것과 다차원 질의를 생성하는 것이다. 제4.1절에서는 *XQ-Cube*라는 새로운 개념의 XML 큐브를 제시하고, 제4.2절에서는 XQ-Cube에 대한 다차원 질의어로서 *XML-MDX*라는 질의어를 제시한다. 그리고 제4.3절에서는 XML-MDX로 표현된 질의의 처리 방법을 제시한다.

4.1 XQ-Cube

XML 웨어하우스는 사실 데이터가 XML 문서들이므로 XML 문서 전체를 측정치로 할 경우, XML 웨어하우스로부터 만들어지는 데이터 큐브는 XML 문서의 통합(aggregation)을 요구한다. 그런데, XML 문서는 계층 구조를 가진 복합 객체이므로 XML 문서 전체에 대한 통합은 정의하기가 어렵다. 하지만, 문서를 구성하는 일부 숫자 데이터나 텍스트 데이터에 대한 통합은 정의하기가 쉽다.

본 논문에서는 XML 웨어하우스로부터 데이터 큐브를 만들 때 문서의 일부 데이터를 측정치로 하고 XQuery[XQuery, 2005] 식을 이용하여 이를 기술한다. 그리고, 이러한 데이터 큐브를 *XQ-Cube*라 부른다. XQ-Cube에서 XQuery 식의 결과가 수치 데이터이면 XQ-Cube는 기존의 관계형 큐브와 같아진다. 그러나, XQuery 식의 결과가 텍스트 데이터이면 이에 대한 통합 연산이 필요하다. 본 논문에서는 이를 위해 텍스트 마이닝 연산을 도입한다.

XQ-Cube는 다음과 같은 특징을 가진다. (1) XQuery 식을 이용하여 측정치를 기술하므로 같은 XML 웨어하우스로부터 다양한 종류의 큐브를 만들 수 있다. (2) 측정치가 XML 문서의 일부이므로 데이터 타입에 따라 여러 가지 통합 연산을 적용할 수 있다. (3) XQ-Cube는 XQuery 식의 결과값에 따라 기존 관계형 큐브가 될 수도 있고 텍스트 큐브가 될 수도 있으므로 기존 데이터 큐브의 일반화된 모습이다.

1) 이때 사용되는 색인 구조는 향후 연구한다.

4.2 XML-MDX

데이터 큐브에 대한 질의를 하기 위해서는 다차원 질의어가 필요하다. 관계형 큐브를 위한 다차원 질의어로서 마이크로소프트가 제안한 MDX (Multidimensional Expression Language) 언어가 있다[Spofford, 2001]. 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 2>는 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.

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

<그림 2> CREATE XQ-CUBE 문의 구조

FROM 절은 XQ-Cube의 생성시 사용될 측정치를 명시한다. <그림 3>은 BNF 표기법에 따른 FROM 절의 정의를 보여 주고 있다. <XQ-Cube_specification>은 XQuery 식을 이용한 측정치를 명시한다. 이 때, 측정치의 데이터 타입에 따라 적절한 통합 연산자를 지정해 준다.

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

<그림 3> FROM 절의 구조

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 4>는 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.

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

<그림 4> WHERE 절의 구조

SELECT 문: <그림 5>는 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  <axis 0 specification>,
        <axis 1 specification>,
        ...
FROM  <XQ-Cube name>
[ WHERE  <slicer specification> ]
```

<그림 5> SELECT 문의 구조

SELECT 절은 SELECT 결과 큐브의 축을 명시한다. <그림 6>은 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.

```
<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>)
```

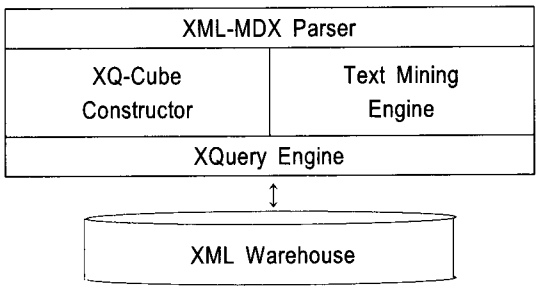
<그림 6> SELECT 절의 구조

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 모두 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 조건식을 사용할 수 있다. 마이크로소프트 MDX는 차원 계층구조의 경로식만 명시할 수 있다.

4.3 XML-MDX 질의 처리

본 절에서는 XML-MDX로 기술된 다차원 질의의 처리 방법을 논한다. <그림 7>은 XML- MDX 질의 처리기의 아키텍춰를 보여주고 있다. XML-MDX Parser는 사용자로부터 질의를 받아들이고 처리된 질의의 결과를 반환하는 역할을 한다. XQ-Cube 생성기는 XQuery 엔진과 텍스트 마이닝 엔진을 이용하여 XQ-Cube를 만드는 역할을 한다. XQuery 엔진은 XML-MDX 질의에 명시된 XQuery를 처리하는 역할을 하며 텍스트 마이닝 엔진은 텍스트 데이터에 대한 통합 연산을 처리하는 역할을 한다.



<그림 7> XML-MDX 질의 처리기

CREATE XQ-CUBE 문은 다음과 같은 순서로 처리된다. (1) XML-MDX 파서가 질의를 파싱하여 FROM 절과 WHERE 절로 나눈다. (2) XQ-Cube 생성기는 FROM 절에 명시된 측정치를 가지는 XQ-Cube를 생성한다. (3) WHERE 절에 명시된 대로 절단을 한다. (4) 측정치의 통합이 필요하면 FROM 절에 명시된 통합 연산자를 이용한다. 측정치가 텍스트 데이터이면 텍스트 마이닝 엔진을 이용한다.

SELECT 문은 다음과 같은 순서로 처리된다. (1)XML-MDX 파서가 질의를 파싱하여 SELECT, FROM, WHERE 세 개의 절로 나눈다. (2) XQ-Cube 생성기가 FROM 절에 명시된 XQ-Cube를 로드한다. (3) WHERE 절에 명시된 대로 XQ-Cube를 절단한다. (4) 결과 큐브를 SELECT 절에 명시된 축 순서대로 선회(pivoting)한다. (5) 결과 큐브를 사용자에게 반환한다.

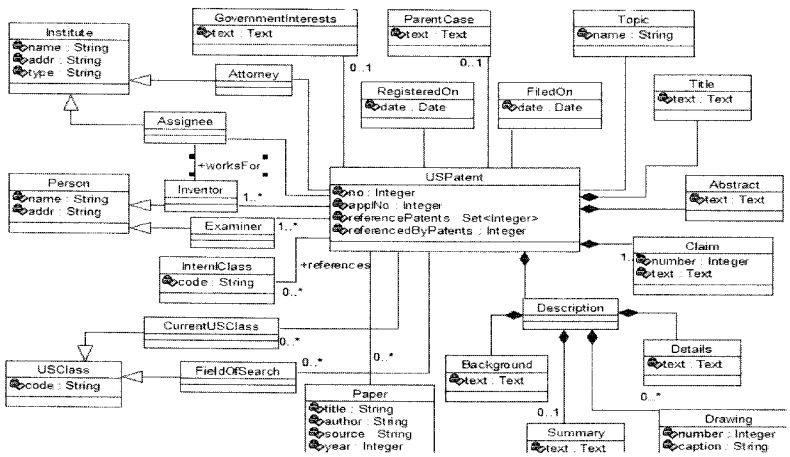
```
<uspatent>
    <title>
        <text>  Rule based database security system and method  </text>
    </title>
    <abstract>
        <text> A rule-based database security system and method are disclosed. </text>
    </abstract>
    <inventor>
        <name> Cook; William R. </name>
        <addr> Redwood 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 8> XML로 기술된 미국특허 문서 예

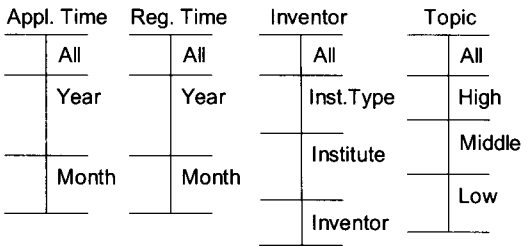
V. 미국특허 웨어하우스 다차원 분석

본 장에서는 XML 웨어하우스에 대한 다차원 분석 프레임워크를 미국특허 웨어하우스에 적용해 본다. 먼저, XML로 기술된 미국특허 문서들이 주어져 있다고 가정한다. <그림 8>은 XML 문서로 표현된 미국특허의 한 예를 보여 주고 있다. 미국특허 문서를 분석하여 <그림 9>와 같은 UML 클래스 다이어그램 기반 개념적 모델을 수립한다. 개념적 모델을 통하여 다차원 분석에 사용할 차원을 결정한다.

<그림 10>은 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 9> 미국특허 문서의 개념 스키마



<그림 10> 차원 계층 구조

<그림 11>은 'Appl.Time' 차원에 대한 XML 문서의 한 예를 보여 주고 있다. 출원년도가 1998년도에 관한 것이다. 년도의 하위 수준으로는 월이 있고 1998년도에는 출원월이 3월과 9월이 있다.

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

<그림 11> Appl.Time 차원 데이터 XML 문서

<그림 12>는 'Inventor' 차원에 대한 XML 문서의 한 예를 보여 주고 있다. 발명자 이름은 'Il-Yeol Song'이고 소속된 기관 이름은 'Drexel'이며 기관 타입은 'university'이다.

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

<그림 12> Inventor 차원 데이터 XML 문서

<그림 13>은 'Topic' 차원에 대한 XML 문서의 한 예를 보여 주고 있다. 최상위 수준의 분야는 'software'이고, 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.

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

<그림 13> Topic 차원 데이터 XML 문서

<그림 14>는 XQ-Cube를 생성하는 XML-MDX 문의 한 예를 보여 주고 있다. 생성할 XQ-Cube의 이름은 'XQ-Cube-1'이다. FROM 절의 XQuery 식은 'XQ-Cube-1'의 측정치를 명시하고 있다. 즉, '/cd/uspatent'라는 collection에 있는 XML 문서들의 '//patent/no'를 구한다. 통합 연산자 'COUNT'는 '//patent/no'의 개수를 세며 그 결과가 'XQ-Cube-1'의 측정치가 된다. WHERE 절은 절단자를 명시하고 있으며 'Appl.Time' 차원에 대해서는 'ALL', 'Reg.Time' 차원에 대해서는 '2000' 보다 큰 'year'만 선택하고 나머지는 버린다.

```
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 14> XQ-Cube 생성 예

<그림 15>는 만들어진 XQ-Cube에 대한 XML-MDX 질의문의 한 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 결과로 반환될 큐브는 SELECT 절에 명시된 축을 가진다. COLUMNS는 'XML'과 'OLAP'이라는 두 개의 'topic'을 가지고, ROWS는 이름이 'university'와 'industry'인 두 개의 'instType'을 가진다. <그림 16>은

<그림 15>의 질의에 대한 결과의 한 예를 보여 주고 있다.

```
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 15> XML-MDX 질의 예

	XML	OLAP
university	126	435
industry	267	672

<그림 16> 질의 결과 예

<그림 17>은 측정치가 텍스트 데이터인 XQ-Cube를 생성하는 XML-MDX 문의 한 예를 보여 주고 있다. 생성할 XQ-Cube의 이름은 'XQ-Cube-2'이고 측정치는 특허 제목의 주요 키워드이다. <그림 18>은 'XQ-Cube-2'에 대한 XML-MDX 질의문의 한 예를 보여 주고 있으며 <그림 19>는 그 질의에 대한 결과의 한 예를 보여 주고 있다.

```
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 17> 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 18> XML-MDX 질의 예

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

<그림 19> 질의 결과 예

Ⅵ. 결 론

본 논문에서는 XML 웨어하우스에 대한 다차원 분석 프레임워크를 제안하였다. 본 논문에서 가정한 XML 웨어하우스는 사실과 차원 데이터를 모두 XML 문서로 표현한다. XML 문서를 다차원적으로 분석하기 위해 XQ-Cube라는 새로운 타입의 XML 큐브를 제안하였다. XQ-Cube는 XQuery 식에 의해 기술된 측정치를 가지며 측정치가 텍스트 데이터인 경우 통합 시 텍스트 마이닝 연산자를 사용한다. 그리고, XQ-Cube에 대한 다차원 질의어로서 XML-MDX를 제안하고 미국특허 XML 웨어하우스를 통하여 XML-MDX의 사용 예를 보였다. 본 논문에서 제안한 다차원 분석 프레임워크는 인터넷 상에 존재하는 방대한 양의 XML 문서들을 효과적으로 분석하는데 기여할 수 있으리라 믿는다.

본 논문의 공헌은 다음과 같다. (1) XML 문서의 다차원 분석을 위하여 XML-OLAP이라는 새로운 프레임워크을 개발하였다. XML-OLAP은 XML 문서를 다차원적으로 분석할 수 있는 최초의 프레임워크이라고 생각한다. 특히, XML-MDX 질의어는 MDX에 XQuery를 결합하여 XML 문서의 계층적 트리 구조를 잘 반영할 수 있다. (2) XML 문서에 포함된 텍스트 데이터의 통합을 위해 텍스트 마이닝 연산을 도입 하였다. 이는 텍스트 마이닝 기술이 OLAP과 결합할 수 있는 메카니즘을 제공한다.

본 논문의 향후 연구는 다음과 같다. (1) XML-

OLAP은 하나의 프레임워크로서 아직 완전히 구현되지 못했다. 따라서, 구현이 완성되면 그 성능을 평가해 보는 것이 시급하다. (2) XML 웨어하우스 구축 시에 차원 데이터의 생성과 더불어 색인도 함께 구축된다. 이 색인은 차원과 사실 데이터를 연결하는 기능을 수행하는데 이에 적합한 색인 구조를 연구하여야 한다. (3) XML-OLAP은 텍스트 데이터 통합을 위하여 텍스트 마이닝 연산을 도입하였다. 향후, 외부의 텍스트 마이닝 연산을 XML-OLAP에 플러그인(plug-in) 할 수 있는 메카니즘을 제공할 계획이다.

〈참 고 문 헌〉

[1] Abello, A., Samos, J., and Saltor, F., "Understanding Facts in a Multidimensional Object-Oriented Model," In *Proc. The 4th ACM Intl Workshop on Data Ware-housing and OLAP (DOLAP01)*, Atlanta, 2001, pp. 32-39.

[2] Conallen, J., *Building Web Applications with UML*, Addison Wesley, 2000.

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

[4] Hummer, W., Bauer, A., and Harde, G., "XCube - XML For Data Warehouses," In *Proc. The 6th ACM Intl Workshop on Data Warehousing and OLAP (DOLAP03)*, New Orleans, Louisiana, 2003, pp. 33-40.

[5] Jensen, M.R., Mller, T.H., and Pedersen, T.B., "Specifying OLAP Cubes on XML Data," *Journal of Intelligent Information Systems*, Vol. 17, No. 2/3, 2001, pp. 255-280.

[6] Jensen, M.R., Mller, T.H., and Pedersen, T.B., "Converting XML Data To UML Diagrams For Conceptual Data Integration," In *Proc. The 1st Intl Workshop on Data Integration Over The Web*, 2001, pp. 17-31.

[7] Katz, H., *XQuery from the Experts - A Guide to the W3C XML Query Language*, Addison Wesley, 2004.

[8] Lujan-Mora, S., Trujillo, J., and Vassiliadis, P., "Advantages of UML for Multidimensional Modeling," In *Proc. the 6th Intl Conf. on Enterprise Information Systems (ICEIS 2004)*, ICEIS Press, Porto (Portugal), 2004, pp. 298-305.

[9] Nassis, V., Rajugan, R., Dillon, T.S., and Rahayu, W., "Conceptual Design of XML Document Warehouses," In *Proc. Data Warehousing and Knowledge Discovery, 6th International Conference, DaWaK 2004*, Zaragoza, Spain, 2004, pp. 1-14.

[10] Niemi, T., Nummenmaa, J., and Thanisch, P., "Constructing OLAP Cubes Based on Queries," In *Proc. The 4th ACM Intl Workshop on Dara Warehousing and OLAP (DOLAP01)*, Atlanta, 2001.

[11] Niemi, T., Niinimaki, M., Nummenmaa, J., and Thanisch, P., "Constructing an OLAP Cube from Distributed XML Data," In *Proc. The 5th ACM Intl Workshop on Data Warehousing and OLAP (DOLAP02)*, McLean, 2002, pp. 22-27.

[12] Niemi, T., Niinimaki, M., Nummenmaa, J., and Thanisch, P., "Applying grid technologies to XML based OLAP cube construction," In *Proc. The 5th Intl Workshop on Design AND Management Of Data Warehouses (DMDW03)*, Berlin, Germany, 2003.

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

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

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

[16] Rusu, L.I., Rahayu, W., and Taniar, D., "On Building XML Data Warehouses," In *Proc. Intelligent Data Engineering and Automated Learning - IDEAL 2004, 5th International Conference*, Exeter, UK, 2004, pp. 293-299.

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

[18] Sullivan, D., *Document Warehousing and Text Mining*, John Wiley & Sons, 2001.

[19] Theodoratos, D., "Exploiting Hierarchical Clustering in Evaluating Multidimen-sional Aggregation Queries," In *Proc. The 6th ACM Intl Workshop on Data Ware-housing and OLAP (DOLAP03)*, New Orleans, Louisiana, 2003, pp. 63-70.

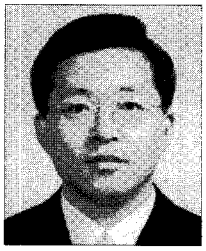
[20] USPTO (United States Patent and Trademark Office), http://www.uspto.gov/

[21] XML Path Language (XPath) 2.0, W3C Working Draft, Feb. 2005, http://www.w3.org/TR/xpath20/

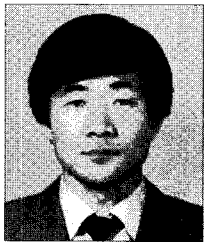
[22] XQuery 1.0: An XML Query Language, W3C Working Draft, Feb. 2005, http://www.w3.org/TR/xquery/

[23] Zhang, J., Ling, T.W., Bruckner, R.M., and Tjoa, A.M., "Building XML Data Warehouse Based on Frequent Patterns in User Queries," In *Proc. Data Ware-housing and Knowledge Discovery, 5th International Conference, DaWaK 2003*, Prague, Czech Republic, 2003, pp. 99-108.

◆ 저자소개 ◆



박병권 (Park, Byung-Kwon)
서울대학교 공과대학 산업공학과를 졸업하였고, KAIST 경영과학과에서 공학석사, KAIST 전산학과에서 공학박사를 취득하였다. 삼성전자(주) 컴퓨터 개발실 주임연구원과 중앙연구소 선임연구원으로 근무하였다. 현재 동아대학교 경영정보과학부 조교수로 재직 중이다. 주요 관심분야는 정보검색, XML 데이타베이스, XML OLAP, XML Stream, 비즈니스 인텔리전스, SOA 등이다.



이종학 (Lee, Jonghak)
경북대학교 전자공학과를 졸업하였고, 한국과학기술원 전산학과에서 공학석사, 박사를 취득하였다. 정보처리기술사와 금성통신(주) 부설연구소 주임연구원, 한국통신 연구개발본부 선임연구원으로 근무하였다. 현재 대구가톨릭대학교 컴퓨터정보통신공학부 교수로 재직 중이다. 주요 관심분야는 객체 데이타베이스, 다차원 파일구조, 물리적 데이터베이스 설계, 데이터 웨어하우스, 생물정보학 등이다.

◈ 이 논문은 2005년 5월 25일 접수하여 2차 심사를 거쳐 2005년 11월 30일 게재확정되었습니다.