XML 큐브를 이용한 다차원 XML 문서 분석

박 병 권 (동아대학교 경영정보과학부)

I. 서 론

OLAP (Online Analytical Processing) 시스템은 의사결정 지원을 위한 강력한 데이터 분석 도구이다. 그것은 데이터 웨어하우스에 있는 방대한 양의 데이터를 여러 각도(또는 차원)에서의 분석을 제공한다. 일반적으로 데이터 웨어하우스는 하나의 큰 사실 테이블과 여러 개의 작은 차원 테이블들로 구성된다. 사실 테이블과 차원들은 대개 관계형 데이터베이스에 저장될 수 있는 구조화된 데이터이다.

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 구조를 가지고 있다. 뿐만 아니라, XML 문서는 텍스트와 같은 비구조화된 데이터를 포함하고 있다. 따라서 XML 문서에 대한 새로운 다차원 분석틀이 필요하다.

본 논문에서는 이러한 다차원 분석틀을 제안하고 이를 XML-OLAP이라 부른다. XML-OLAP은 모든 사실 데이터와 차원 데이터가 XML 문서로 저장되어 있는 XML 웨어하우스를 기반으로 한다. XML 큐브는 XML 웨어하우스로부터 만들어 진다. 기존의 데이터 큐브는 수치 데이터의 측정치를 가졌지만 XML 큐브는 수치 데이터와 텍스트 데이터 모두를 가진다. 본 논문에서는 XML 큐브에 대한 다차원 질의어를 제안하고 이를 XML-MDX라 부른다. 또한, 텍스트 데이터에 대한 aggregation을 위하여 summarization, classification, top keyword extraction 등과 같은 텍스트 마이닝 연산을 도입한다.

XML-OLAP의 유효함을 평가하기 위하여 미국 특허 웨어하우스에 적용한다. 미국특허 웨어하우스는 미국특허 웹사이트로부터 특허 정보를 추출하여 XML 문서로 바꾸고 이를 XML 데이터베이스에 저장하여 구축하였다. 그리고 XML-MDX 질의를 통하여 미국특허를 다차원적으로 분석하는데 XML-OLAP이 효과적임을 보인다.

본 논문의 공헌은 다음과 같다. (1) XML 문서의 다차원적 분석을 위하여 XML-OLAP이라는 새로운 틀을 개발하였다. XML 문서의 양이 점점 늘어감에 따라 이들을 효과적으로 분석하는 것이 필요하다. XML-OLAP은 이를 위한 최초의 분석틀이고 생각한다. (2) 새로운 다차원 질의어인 XML-MDX를 개발하였다. XML-MDX는 오늘날 OLAP 질의어의 산업계 표준으로 받아들여지고 있는 Microsoft MDX로부터 많은 영향을 받았으나 XML 문서의 계층적 트리 구조를 잘 반영할 수 있도록 발전시켰다. (3) 텍스트 마이닝 연산을 도입하여 XML 문서에 포함된 텍스트 데이터의 aggregation을 가능하게 하였다. 이는 텍스트 마이닝 기술이 OLAP과 결합할 수 있는

메카니즘을 제공한다.

본 논문의 구조는 다음과 같다. 제 2장에서는 XML 웨어하우스에 대하여 논한다. 특히, 사실 데이터와 차원 데이터를 XML 문서로 표현하는 방법에 대하여 논한다. 제 3장에서는 XML 웨어하우스로부터 XML 큐브를 생성하는 방법과 XML-MDX를 이용하여 XML 큐브를 질의하는 방법에 대하여 논한다. 제 4장에서는 제안한 XML-OLAP 틀을 미국특허 웨어하우스에 적용하여 그 효용성을 평가한다. 제 5장에서는 결론을 맺는다.

II. XML 웨어하우스

본 장에서는 XML 웨어하우스를 모델링하고 구축하는 것을 논한다. 제 2.1절에서는 XML 웨어하우스의 다차원 모델을 제시하고 이의 도출 방법을 논한다. 제 2.2절에서는 주어진 XML 문서 집합으로부터 XML 웨어하우스를 구축하는 방법을 제시한다.

2.1 XML 웨어하우스의 다차원 모델

본 논문에서는 XML 웨어하우스가 다차원 모델을 가진다고 가정한다. 그림 1은 XML-OLAP에서 가정하는 XML 웨어하우스의 다차원 모델을 보여주고 있다. 사실 데이터를 이루는 한 개의 XML 문서 집합이 존재하고, 각 차원마다 차원 데이터를 이루는 XML 문서 집합이 존재한다. 그림 1에서는 n 개의 차원이 존재하므로 n 개의 XML 문서 집합이 존재한다.

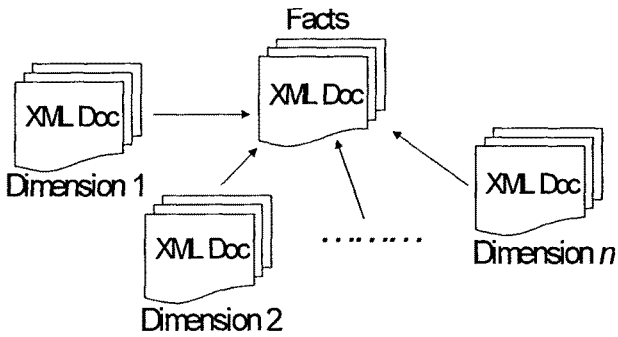

그림 1. XML 웨어하우스의 다차원 모델.

사실 문서 집합은 Nassis 등이 가정한 것과 같이[10] 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.

차원 데이터도 XML 문서로 기술되며 각 차원은 각각의 XML 문서 집합을 가진

다. 각 차원은 계층 구조를 가지므로 하나의 XML 문서는 최상위층 멤버를 루트로 하는 계층 구조의 한 인스턴스에 해당한다. 차원 데이터와 사실 데이터를 연관짓기 위하여 인덱스와 같은 보조 데이터 구조가 사용된다.

본 절의 다차원 모델은 다음과 같은 장점을 가진다: (1) 사실 데이터와 차원 데이터가 모두 XML 문서로 기술되므로 XML 웹어하우스를 쉽게 구축할 수 있다. (2) 사실 데이터와 차원 데이터를 XML 데이터베이스에 저장하고 관리할 수 있다. (3) XML 문서의 계층 구조를 이용하여 하나의 문서에 차원 계층 구조를 표현할 수 있다.

2.2 XML 웨어하우스의 구축

XML 웨어하우스를 구축하는 것은 사실 데이터를 이루는 XML 문서 집합과 차원 데이터를 이루는 XML 문서 집합을 구축하는 것이다. Rusu등은[15] 기존의 XML 문서를 처리하여 데이터 웨어하우스에 저장하기 위하여 데이터 정제의 문제를 다루었으나 본 논문에서는 사실 데이터를 이루는 XML 문서는 정제되어 있다고 가정하고 차원 데이터를 이루는 XML 문서의 생성에 초점을 맞춘다. 사실 데이터를 이루는 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[6] XML 데이터의 DTD를 이용하여 자동적으로 UML 클래스 다이어그램을 생성하는 알고리즘을 제안하였다. Lujan-Mora등은[9] UML을 확장하면 다차원 모델링 언어가 될 수 있음을 보였다. 본 논문에서도 그들의 방법을 도입하여 XML 문서의 개념적 모델로 UML 클래스 다이어그램을 사용한다.

사실 데이터를 이루는 XML 문서들의 개념적 모델을 통해 사실 데이터의 논리적 구조를 이해하고 분석을 위한 차원을 정한다. Nassis등은[10] 사용자의 요구사항을 분석하여 차원을 정하고 XML 뷰를 이용하여 차원을 표현할 것을 제안하였다. 그들은 모든 차원이 사실 데이터 속에 포함되어 있다고 가정하였다. 그러나, 본 논문에서는 어떤 차원은 사실 데이터 밖에서 주어질 수도 있으므로 각 차원에 대한 XML 문서를 구축하기로 한다.

III. XML 웨어하우스의 다차원 분석틀

본 장에서는 XML 웨어하우스 상에서의 다차원 분석을 위한 틀을 기술한다. 분석 틀은 크게 XML 큐브를 생성하는 것과 다차원 질의를 생성하는 것이다. 제 3.1절에서는 *XQ-Cube*라는 새로운 개념의 XML 큐브를 제시하고, 제 3.2절에서는 XQ-Cube에 대한 다차원 질의어로서 *XML-MDX*라는 새로운 언어를 제시한다. 그리고 제 3.3절에서는 XML-MDX로 표현된 질의 처리 방법을 제시한다.

3.1 XML 큐브

XML 웨어하우스는 측정치로서 XML 문서를 가지고 있으므로 XML 웨어하우스로부터 만들어진 XML 큐브의 셀 값은 XML 문서의 aggregation이다. XML 문서는 계층적인 구조를 가진 복합 객체이므로 XML 문서에 대한 aggregation은 정의하기가 어렵다. 그러나, XMl 문서의 일부인 숫자 데이터나 텍스트 데이터에 대한 aggregation은 정의하기가 쉽다.

본 논문에서는 XQuery[21] 표현식을 이용하여 측정치를 명시할 것을 제안한다. 그리고 XQuery 표현식에 의한 측정치를 가진 XML 큐브를 *XQ-Cube*라고 부른다. XQuery 표현식의 결과가 수치 데이터이면 XQ-Cube는 기존의 관계형 큐브와 같고, 텍스트 데이터이면 XQ-Cube의 aggregation 연산으로서 텍스트 마이닝 연산을 도입한다.

XQ-Cube는 다음과 같은 장점을 가진다. (1) XQuery 표현식을 이용하여 측정치를 지정하므로 다양한 종류의 큐브가 만들어 질 수 있다. (2) 측정치가 XML 문서의 일부이므로 데이터 타입에 따라 여러 가지 aggregation 연산을 적용할 수 있다. (3) XQ-Cube는 XQuery 표현식의 결과값에 따라 기존 관계형 큐브가 될 수도 있고 텍스트 큐브가 될 수도 있다.

3.2 다차원 질의 언어

큐브에 대한 질의를 하기 위해서는 다차원 질의어가 필요하다. 관계형 큐브를 위한 다차원 질의어로서 마이크로소프트가 제안한 MDX (Multidimensional Expression Language) 언어가 있다. 본 논문에서는 MDX의 영향을 받아 XQ-Cube에 적합한 다차원 질의어로서 *XML-MDX*를 제안한다. XML-MDX는 두 가지로 대별된다. XQ-Cube를 생성하기 위한 CREATE XQ-CUBE 문과 질의를 위한 SELECT 문이다.

CREATE XQ-CUBE: 그림 2는 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 3은 BNF 표기법에 따른 FROM 절의 정의를 보여 주고 있다. <XQ-Cube_specification>은 XQuery 표현식을 이용한 측정치를 명시한다. 이 때, 측정치의 데이터 타입에 따라 적절한 aggregation 연산자를 지정해 주어야 한다.

<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 aggregation 연산자를 다룬다. ADD, LIST, COUNT, SUMMARY, TOPIC, TOP KEYWORDS 그리고 CLUSTER이다. 이 중 ADD 연산자는 수치 데이터를 위한 것으로서 기존의 관계형 큐브에서와 같고, 나머지 연산자들은 모두 비수치 데이터를 위한 것이다. LIST 연산자는 측정치 집합을 모두 디스플레이하는 것이고, COUNT는 측정치 집합을 원소 개수를 구하는 것이며 나머지는 모두 텍스트 마이닝 연산자들이다. SUMMARY, TOPIC, TOP KEYWORDS는 각각 텍스트의 요약, 주제, 주요 키워드를 뽑는 것이고, CLUSTER는 aggregation 해야 할 전체 텍스트를 클러스터링하는 것이다.

WHERE 절은 선택적인데, slicer에 사용될 차원의 멤버를 선정한다. 즉, 선정된 차원의 멤버에 대해서 XQ-Cube를 slicing 하는 것이다. 그림 4는 BNF 표기법으로 명시한 WHERE 절의 정의이다. <slicer_specification>은 XQuery 표현식의 튜플로서 slicer를 명시한다. 각각의 XQuery 표현식은 slicing이 이루어질 차원의 멤버를 명시한다. All 멤버에 대해서는 특별한 XQuery 표현식이 사용된다 (그림 14 참조). <slicer_specification>에 명시되지 않은 차원은 생성될 XQ-Cube의 축을 이룬다.

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

그림 4. WHERE 절의 구조.

SELECT: 그림 5는 SELECT 문의 기본 구조를 보여 주고 있다. SELECT 문은 마이크로소프트 MDX의 SELECT 문과 같은 구조를 가진다. 즉, SELECT, FROM, WHERE 절을 가진다. FROM 절은 CREATE XQ-CUBE 문을 통해 생성된 XQ-Cube의 이름을 명시한다.

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

그림 5. XML-MDX 문의 기본 구조.

SELECT 절은 결과 큐브의 축을 명시한다. 그림 6은 BNF 표기법으로 명시한 SELECT 절의 정의를 보여 주고 있다. 각각의 <axis_specification>이 하나의 축을 명시한다. XML 웨어하우스의 차원의 개수가 축의 최대 개수이다. 하나의

<axis_specification>은 여러 개의 XQuery 표현식과 축의 이름으로 구성된다. XQuery 표현식의 결과값들은 그 축의 멤버를 구성한다. 즉, 한 축을 구성하는 각 멤버마다 하나의 XQuery 표현식이 존재한다. 각 차원은 하나의 XML 문서로 표현되어 있으므로 XQuery 표현식은 차원 계층 구조의 한 멤버를 명시한다. 축의 이름은 마이크로소프트 MDX[16]와 동일한 방법으로 정해진다. 각 축은 축 번호를 가진다. X-축은 0, Y-축은 1, Z-축은 2 등의 순이다. <index>는 축 번호를 가리킨다. 처음 5개의 축에 대해서는 COLUMNS, ROWS, PAGES, SECTIONS, 그리고 CHAPTERS 등이 AXIS(0), AXIS(1), AXIS(2), AXIS(3), 그리고 AXIS(4)의 별명으로 사용될 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 SELECT 절과 동일하다. <slicer_specification>은 FROM 절에 명시된 XQ-Cube를 필터링한다. 마이크로소프트 MDX에서와 같이, SELECT 절에 축으로 명시되지 않은 차원은 slicer 차원으로 간주한다. 그들은 자신의 디폴트 멤버 값으로 slicing 한다.

XML-MDX는 마이크로소프트 MDX에 비해 다음과 같은 장점을 가진다. (1) XQuery 표현식을 이용하여 측정치나 차원 멤버를 명시하므로 다차원 질의 구성과 처리가 쉽다. 즉, XML-MDX는 XQuery 외에 특별한 구문이 없으므로 배우기가 쉽고 그 처리도 기존의 XQuery 엔진을 이용하면 된다. (2) 축과 slicer를 명시할 때, XQuery 표현식을 이용하므로 어떤 조건을 만족하는 차원 멤버만 선택하도록 명시하는 것이 가능하다. 마이크로소프트 MDX는 단지 차원 계층 구조 상에서의 경로만 명시할 수 있다.

3.3 다차원 질의 처리

본 절에서는 XML-MDX로 기술된 다차원 질의의 처리 방법을 논한다. 그림 7은 XML-MDX 질의 처리기의 아키텍춰를 보여주고 있다. XML-MDX Parser는 사용자로부터 질의를 받아 들이고 질의 결과를 반환하는 역할을 한다. XQ-Cube Constructor는 XQuery Engine과 Text Mining Engine을 이용하여 XQ-Cube를 만드는 역할을 한다. XQuery Engine은 XML-MDX 질의에 명시된 XQuery를 처리하는 역할을 하며 Text Mining Engine은 텍스트 데이터에 대한 aggregation 연산을 처리하는 역할을 한다.

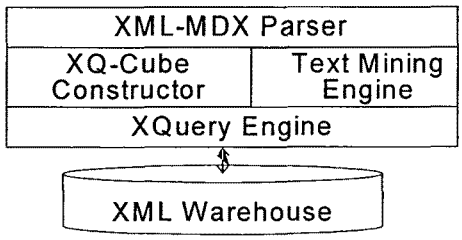

그림 7. XML-MDX 질의 처리기.

CREATE XQ-CUBE 문은 다음과 같은 순서로 처리한다. (1) XML-MDX Parser가 파싱을 하여 FROM 절과 WHERE 절로 나눈다. (2) XQ-Cube Constructor는 FROM 절에 명시된 측정치를 가지고 XQ-Cube를 생성한다. 이 때, XQuery Engine을 이용하여 XQuery 표현식을 처리한다. 생성된 XQ-Cube는 다음을 위해 저장된다. (3) 저장하기 전에, XQ-Cube Constructor는 XQ-Cube를 WHERE 절에 명시된 대로 slicing을 한다. (4) Slicing을 할 때, XQ-Cube Constructor는 FROM 절에 명시된 aggregation 연산자를 이용하여 측정치를 aggregation 한다. 측정치가 텍스트 데이터이면 aggregation을 위해 Text Mining Engine을 이용한다.

SELECT 문은 XML-MDX 질의이고 다음과 같은 순서로 처리한다. (1) 사용자가 XML-MDX 질의를 입력하면 XML-MDX Parser는 SELECT, FROM, WHERE 세 개의 절로 나눈다. (2) XQ-Cube Constructor가 FROM 절에 명시된 XQ-Cube를 로드한다. (3) XQ-Cube Constructor가 WHERE 절에 명시된 대로 XQ-Cube를 slicing을 한다. (4) Slicing을 할 때, XQ-Cube Constructor는 XQ-Cube에 정의된 aggregation 연산자를 이용하여 측정치를 aggregation 한다. (5) XQ-Cube Constructor가 SELECT 절에 명시된 대로 slicing 결과 큐브를 pivoting 한다.

IV. 미국특허 웨어하우스 다차원 분석

본 장에서는 XML 웨어하우스에 대한 다차원 분석틀을 미국특허 웨어하우스에 적용해 본다. 먼저, 미국특허에 관한 XML 문서 집합이 주어졌다고 가정한다. 이 문서 집합은 미국특허 XML 웨어하우스의 사실 데이터를 이룬다. 그림 8은 XML 문서로 표현된 미국특허의 한 예를 보여 주고 있다. 사실 데이터를 분석한 다음에는 그림 9와 같이 UML 클래스 다이어그램을 이용하여 개념적 모델을 수립한다. 개념적 모델을 통하여 다차원 분석에 사용될 차원을 결정한다.

그림 10은 미국특허 분석에 사용될 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.

```
<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. 미국특허에 관한 사실 데이터 예.

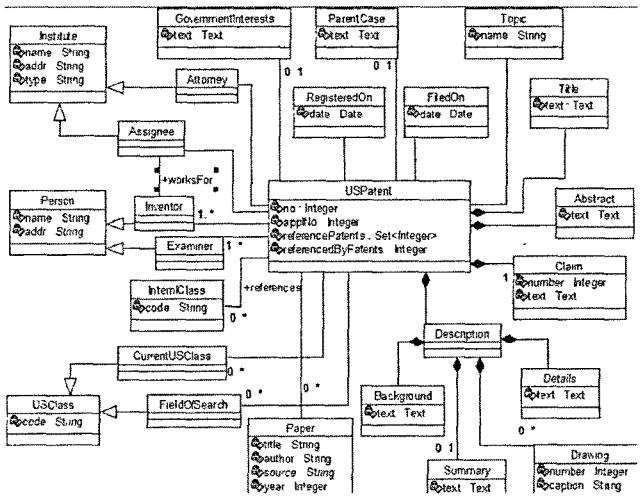

그림 9. 사실 데이터에 대한 개념 스키마.

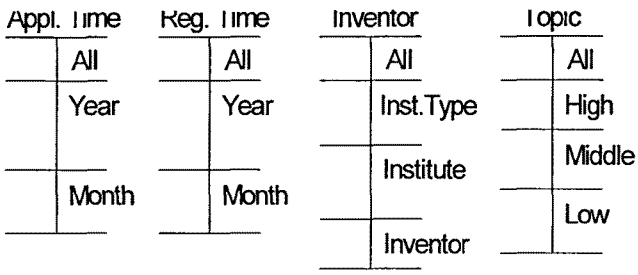

그림 10. 차원 계층 구조.

그림 11은 차원 'Appl.Time'에 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 12는 차원 'Inventor'에 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 13은 차원 'Topic'에 대한 XML 문서의 한 예를 보여 주고 있다. 최상위 수준의 분야는 'software'이고, 중간 수준의 분야는 'database'와 'AI'가 있다. 'database'에 대한 하위 수준의 분야는 'model'과 'language'가 있고, 'AI'에 대한 하위 수준의 분야는 'Vision'이 있다.

```
<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 14는 XQ-Cube를 만드는 XML-MDX 문의 예를 보여 주고 있다. 만드는 XQ-Cube의 이름은 XQ-Cube-1이다. FROM 절의 XQuery 식은 XQ-Cube-1의 측정치를 명시하고 있다. 즉, 사실 데이터를 나타내는 XML 문서들은 /cd/uspatent 라는 collection에 있고 여기서 //patent/no를 구한다. COUNT는 //patent/no 의 개수를 세는 연산이며 COUNT의 결과가 XQ-Cube-1의 측정치가 된다. WHERE 절은 slicer를 명시하고 있으며 Appl.Time 차원에 대해서는 All member만, Reg.Time 차원에 대해서는 2000보다 큰 year member만 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 15는 만들어진 XQ-Cube에 대한 XML-MDX 질의문의 예를 보여 주고 있다. 먼저 WHERE 절에 명시된 slicer 조건에 의해 XQ-Cube-1에서 RegTime이 2002보다 큰 year member만 선택되고 나머지는 버린다. 질의 결과로 반환될 큐브는 SELECT 절에 명시된 축을 가진다. COLUMNS 축은 'XML'과 'OLAP' 이라는 두 개의 'topic' 을 구성 멤버로 가지고, ROWS 축은 name이 'university' 와 'industry'인 두 개의 'instType'을 구성 멤버로 가진다.

```
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.

V. 결론

본 논문에서는 XML 웨어하우스에 대한 다차원 분석틀을 제안하였다. 본 논문에서 가정한 XML 웨어하우스는 모든 사실과 차원 데이터를 XML 문서로 표현한다. XML 웨어하우스로부터 XQ-Cube라 명명한 새로운 타입의 XML 큐브를 제안하였다. XQ-Cube는 XQuery 표현식에 의해 기술된 측정치를 이용하여 만들어지며 측정치가 텍스트 데이터인 경우 aggregation을 위해 텍스트 마이닝 연산자를 사용하였다. 그리고, XQ-Cube에 대한 다차원 질의어로서 XML-MDX를 제안하였다. 마지막으로, 미국 특허 XML 웨어하우스에 대한 다차원 분석 예를 통하여 XML-MDX의 효용성을 보였다. 본 논문에서 제안한 다차원 분석틀이 인터넷 상에 존재하는 방대한 양의 XML 문서들을 효과적으로 분석하는데 기여할 수 있으리라 믿는다.

참 고 문 헌

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

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

[3] 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, Atlanta, 2001.

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

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

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

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

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

[9] 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, Zaragoza, Spain, 2004.

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

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

[12] 12. T. Niemi, M. Niinimaki, J. Nummenmaa and P. Thanisch, "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] J. Pokorny, "Modelling Stars Using XML," In *Proc. The 4th ACM Intl Workshop on Dara Warehousing and OLAP (DOLAP01)*, pp. 24-31, Atlanta, 2001.

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

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

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

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

[18] United States Patent and Trademark Office, http://www.uspto.gov/

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

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

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

<Abstract>

Multidimensional Analysis of XML Documents using XML Cubes

Byung-Kwon Park
Dong-A University

Nowadays, large amounts of XML documents are available on the Internet. Thus, we need to analyze them multi-dimensionally in the same way as relational data. In this paper, we propose a new frame-work for multidimensional analysis of XML documents, which we call *XML-OLAP*. We base XML-OLAP on XML warehouses where every fact data as well as dimension data are stored as XML documents. We build XML cubes from XML warehouses. We propose a new multidimensional expression language for XML cubes, which we call *XML-MDX*. XML-MDX statements target XML cubes and use XQuery expressions to designate the measure data. They specify text mining operators for aggregating text constituting the measure data. We evaluate XML-OLAP by applying it to a U.S. patent XML warehouse.We use XML-MDX queries, which demonstrate that XML-OLAP is effective for multi-dimensionally analyzing the U.S. patents.