웹 기반의 OLAP 메타데이터 교환 시스템의 설계 및 구현

이 인 기[†] · 이 민 수[††] · 용 환 승[†††]

요 약

최근 지식경영의 중요성이 부각됨에 따라 데이터웨어하우징에 대한 관심이 집중되고 있다. 특히 온라인 분석 처리(On-Line Analytical Processing : OLAP) 시스템은 데이터 웨어하우스의 효과적인 활용 방안으로 많은 종류의 OLAP 제품들이 공급되어 왔다. 그러나 OLAP 기술은 실제로 아무런 표준 없이 매우 오랫동안 사용되어 왔으며, 이러한 다양성은 여러 OLAP 제품들 간에 데이터 교환과 인터페이스를 매우 어렵게 했다. 이에 본 논문에서는 서로 다른 OLAP 제품들간에 공통적으로 사용할 수 있는 OLAP 메타데이터 교환 모델을 설계하고, 그 메타데이터를 기반으로 생성된 큐브들을 교환할 수 있도록 하는 OLAP 메타데이터 교환 시스템을 구현하였다. OLAP 메타데이터 교환 모델의 설계는 XML을 사용하였고, 시스템의 사용자 인터페이스는 웹기반으로서 여러 OLAP 제품들과 메타데이터를 주고 받을 수 있는 환경을 제공하였다. 사용자는 OLAP 제품들의 복잡한 큐브 생성 과정을 특별히 습득할 필요가 없으며, 하나의 큐브로 여러 제품들이 제공하는 서로 다른 분석 환경을 경험할 수 있다. 본 연구를 확장하여 OLAP 제품들이 공통으로 사용될 수 있는 질의 언어를 설계하고 적용 할 수 있다면 OLAP 제품들간의 원활한 의사소통이 이루어질 수 있을 것으로 전망한다.

Design and implementation of a Web-based OLAP metadata interchange system

In-Gi Lee[†] · Minsoo Lee[††] · Hwan-Seung Yong[†††]

ABSTRACT

As the importance of knowledge management is being recognized, there is a significant amount of increase for interest in data warehousing. On-Line Analytical Processing (OLAP) systems can effectively make use of data warehouses. Although there are many commercial OLAP products, they have been developed without any kind of standard resulting in poor data exchange and difficulty in interfacing among the OLAP products. In this paper we propose an OLAP metadata interchange model that can be used among different OLAP products and have implemented an OLAP metadata interchange system that can interchange the cubes created from the metadata. XML is used for the OLAP metadata model and the user interface is Web-based, which makes it easier to interchange metadata among different OLAP products. Users can experience the different analysis environments of different products without the need to learn the complex cube creation process for each product. By extending this research to design a common query language that can be used among OLAP products, OLAP products should be able to more easily talk to one another.

키워드 : OLAP, 메타 데이터(Metadata), 교환(Interchange), XML

1. 서 론

최근 들어 데이터베이스의 합리적 관리와 신속한 의사결정 지원을 위한 각종방안이 심도 있게 논의되고 있는 가운데 특히 최종 사용자로 하여금 기업 내에 방만하게 흩어져 있는 데이터에 손쉽게 접근하고 활용하도록 하는 데이터 웨어하우스 개념이 전 세계적으로 주목을 받고 있다. 이러한 데이터 웨어하우스의 핵심은 온라인 분석 처리(On-Line Analytical Processing : OLAP) 시스템으로 최종 사용자가 다차원 정보에 직접 접근하여 대화식으로 정보를 분석하고 의사결정에 활용할 수 있는 기술을 제공하며, 데이터 마이닝과 함께 가장 대표적인 웨어하우스의 활용수단으로 받아들여지고 있다[1].

그러나 이러한 관심에도 불구하고 사용자들은 OLAP에 관한 체계적인 정보를 얻는데 어려움을 느끼고 있다. 현재 수많은 OLAP 제품들이 소개되고 있지만, 실제로 모든 OLAP 제품들이 공유할 수 있는 논리적 다차원 모델의 표준조차 없

† 준 회 원 : 이화여자대학교 과학기술대학원 컴퓨터학과
†† 정 회 원 : 이화여자대학교 컴퓨터학과 교수
††† 종신회원 : 이화여자대학교 컴퓨터학과 교수
논문접수 : 2002년 7월 24일, 심사완료 : 2002년 11월 27일

이 사용되고 있기 때문이다. 표준의 부재는 여러 OLAP 제품들간에 데이터 교환과 인터페이스를 매우 어렵게 했다. 이미 OLAP 기술을 접해 본 사용자라고 해도 실제 경험해 본 제품이 다를 경우 원활한 의사소통이 이루어지지 못한다는 것이다. 이러한 문제점을 해결하기 위해 OLAP에 관한 표준이 최근 API 수준에서 이루어지고 있다. 그 예로 OLAP 카운실의 MD-API와 마이크로소프트의 OLE DB FOR OLAP API가 표준으로 제안되었다[2]. 이 외에도 많은 벤더들이 OLAP 표준화를 위해 노력하고 있으나 다양하게 개발되어 온 OLAP 제품들을 통합하기에는 여러 가지 어려움들이 많다.

이에 본 논문에서는 서로 다른 OLAP 제품들간에 공통적으로 사용할 수 있는 OLAP 메타데이터 모델을 설명하고, 그 모델을 기반으로 XML을 사용하여 OLAP 메타데이터 교환 모델을 설계하였다. OLAP 메타데이터 교환 모델은 본 논문에서 구현된 OLAP 메타데이터 교환 시스템을 통해 여러 OLAP 제품들로부터 추출된 메타데이터를 웹을 통해 효과적으로 관리하고, 새로운 큐브를 생성하는데 발생하는 문제점들을 해결하는 방식으로 제안하고 있다. OLAP 메타데이터 교환 시스템은 서로 다른 OLAP 제품들간에 차원과 계층구조, 변수 등을 포함하는 큐브 데이타를 교환하기 위한 시스템으로 다양한 OLAP 서버로부터 메타데이터를 추출하여 저장하고, 필요할 경우 다른 OLAP 제품으로 큐브를 자동 생성하는 시스템이다. 이와 같은 시스템은 사용자가 여러 제품들의 복잡한 큐브 생성 과정을 거치지 않고도 다양한 인터페이스를 접할 수 있도록 하며, 각 큐브 생성의 목적에 맞는 성능과 질의를 가진 OLAP 제품들을 선택하고 사용할 수 있도록 해준다. 또한 데이터에 대한 지도 역할을 해주는 메타데이터를 웹 환경 하에서 저장, 관리할 수 있는 기능을 제공함으로써 시스템 개발자나 최종 사용자에게 대량의 OLAP 데이터 관리 및 유지보수를 용이하게 해준다[3, 4].

본 논문에서 제안하는 XML 기반의 OLAP 메타데이타 교환모델은 OLAP 제품들에서 가장 크게 문제가 되는 cube, attribute, dimension, hierarchy의 차이들을 수용하기 위하여 공통적으로 지원하는 메타 데이터 요소들을 식별하여 하나의 통일된 XML 태그로 정의할 뿐만 아니라 각 제품들의 특징들도 포괄적으로 수용할 수 있는 부가적인 태그들을 지원함으로써 OLAP 메타데이타 표준화에 있어서 기존의 제품들이 아무런 변경 없이도 OLAP 메타데이타 교환이 가능토록 해준다. 본 논문의 제안하는 기법은 그러므로 현재 제안되고 있는 MD-API나 OLE DB FOR OLAP과 같은 API 수준의 표준화 방안들보다 한층 포괄적인 표준으로서 사용이 가능하다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구로 OLAP에 관한 표준화 내용들을 살펴본다. 3장에서는 OLAP 메타데이터 모델을 설명하고 OLAP 메타데이터 교환 모델을 설계한다. 4장에서는 3장에서 제안된 OLAP 메타데이터 교환 모델을 기반으로 OLAP 메타데이터 교환 시스템을 구현하여 OLAP 제품들로부터 메타데이터를 추출하여 큐브로 자동 생성하는 모듈들을 보여준다. 마지막으로 5장에서는 본 논문의 결과를 정리하고 향후 연구방향을 제시한다.

2. 관련 연구

본 장에서는 OLAP 표준에 관한 연구들이 어떻게 이루어지고 있는지 살펴보고 이러한 연구들을 보완하기 위한 본 연구의 필요성을 알아본다.

2.1 OLAP 표준에 관한 연구

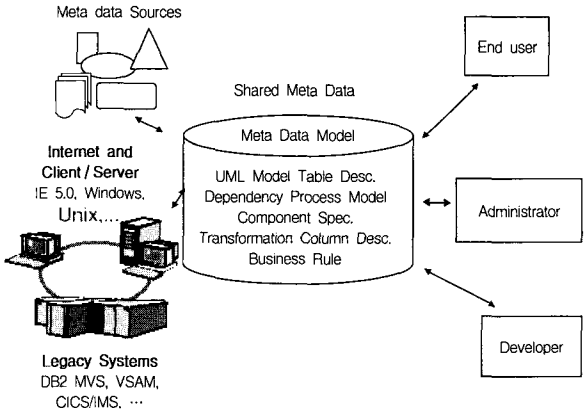
OLAP 제품들이 공유할 수 있는 논리적 다차원 모델의 표준이 없다는 것은 OLAP 제품들간의 데이터 교환을 어렵게 해왔다. 현재 사용되고 있는 40여개의 OLAP 제품들은 서로 다른 OLAP 메타데이터, 기반 기술, 동작 방식을 가지고 개발되었다. 따라서 사용자들은 자신이 경험해 본 OLAP 제품에만 의존하여 OLAP에 관한 편향된 지식을 습득함으로써 OLAP 기술에 전반적인 지식을 얻는데 어려움을 느껴왔다. 이러한 문제점을 해결하기 위해 OLAP에 관한 표준이 최근 API 수준에서 이루어지고 있다. OLAP 카운실의 MD-API와 마이크로소프트의 OLE DB FOR OLAP API가 표준으로 제안되었다[5, 6].

- MD_API (OLAP Council) : MD-API는 OLAP 카운실에서 제안한 표준 API로 다차원 데이터원본에 인터페이스 독립적인 객체 기반의 데이터베이스이다. 이 API는 클라이언트가 다차원 데이터 스키마를 선택하고 연결하여 메타데이터에 질의를 할 수 있도록 해준다. 또한 클라이언트가 서로 다른 벤더들에 의해 제공된 OLAP 제품들 사이에서 발생할 수 있는 다차원 데이터 스키마의 구조와 기능의 차이를 극복하도록 하기 위해 동일한 OLAP 모델을 사용하였다. 질의에 있어서는 질의를 수행하기 위한 특정 언어를 사용하지 않고 객체들의 집합으로써 질의를 표현하였다. 따라서 API에서 사용되거나, 지정된 특정 질의 언어는 존재하지 않는다. 모든 질의 결과 데이터는 큐브라고 불리는 객체로 출력되며, 사용자는 큐브로부터 셀 데이터를 가져올 수 있다[5].
- OLE DB FOR OLAP API(Microsoft) : OLE DB FOR OLAP API는 마이크로소프트에서 제안한 표준 API로 사용자가 다차원 데이터 원본에 접근 할 수 있도록 OLE DB를 확장하여 설계하였다. 이 API는 다차원 데이터의 제공자와 소비자 사이에 원활한 의사소통을 위해 설계된 COM 객체들과 인터페이스들의 집합이다. 또한 다차원 데이터 환경과 무관하게 다양한 OLAP 제품들이 용이하게

의사 소통할 수 있도록 만들어졌으며, OLAP 클라이언트와 서버를 연결하기 위해 MDX(Multidimensional Expressions)라는 다차원 질의 언어를 함께 제공하고 있다. 이 질의 언어는 관계형 모델의 SQL과 유사하며 매우 풍부한 기능을 갖추고 있다[6].

3. OLAP 메타데이터 교환 모델의 설계

본 논문에서는 다른 벤더들에 의해 지원되거나 서로 다른 하드웨어나 플랫폼에서 작동하는 이기종 OLAP 제품들간의 데이터 교환을 위해 표준화된 메타데이터 모델로 확장 가능한 OLAP 메타데이터 교환 모델을 설계하였다[7]. 표준화된 메타데이터 모델이란 동일한 방식으로 메타데이터를 공유하기 위해서 서로 다른 벤더들이 동의할 수 있는 방법으로 메타데이터의 구조와 의미들을 정의할 수 있는 모델을 말한다(그림 1). 세계적으로 다양한 표준 메타데이터가 존재하며, 이 중 메타데이터 연합회(MetaData Coalition)의 개방형 정보모델(Open Information Model)과 객체그룹(Object Management Group)의 공통 웨어하우스 메타모델(Common Warehouse Metamodel)등이 대표적이다[8-11].



(그림 1) 공유되는 메타데이터 모델

3.1 XML을 사용한 OLAP 메타데이터 교환 모델

3.1.1 모델 설계

개방형 정보모델을 기반으로 XML을 사용하여 OLAP 메타데이터 교환 모델을 정의한다. 메타데이터 교환 모델에서 XML을 사용한 이유는 다음과 같다[12, 13]. XML은 W3C(World Wide Web Consortium)에서 표준화되어 플랫폼, 벤더에 독립적으로 사용되며, 프로그래밍 언어 독립적이고 API에 독립적으로 작용하기 때문에 XML API를 포함하는 라이브러리에 의해 쉽게 사용될 수 있다. 또한 문서의 교환은 문서의 구조와 계층에 관계되는 것으로 XML 문서의 정의형을 구성하는 DTD, XML Schema에 의해서 표현할 수 있다.

OLAP 메타데이터 교환 모델의 기본적인 구조는 여러 제품들이 지원하는 모델 구조의 비교 및 분석에서 시작한다. 즉 OLAP 메타데이터 교환 모델은 여러 제품들에서 공통적으로 나타나는 모델 형태를 기본으로 하여 서로 다르게 표현되는 부분을 포함할 수 있는 형태로 설계하였다. 모델에는 여러 제품들이 공유할 수 있는 일반 메타데이터와 특정 제품에 의존적인 고유 메타데이터가 존재한다. 고유 메타데이터의 경우 일반 메타데이터와는 차별되나 여러 제품들에 의해 지원되는 메타데이터와 하나의 제품에서만 지원되는 특수한 메타데이터가 존재할 수 있다. 예를 들어 Attribute는 질의 결과를 보여주기 위해서 사용되거나 질의에서 멤버들의 집합을 선택하기 위한 속성으로 사용된다. 그러나 여러 OLAP 제품들이 모두 이 Attribute를 지원하지는 못하며, 지원을 하는 경우에도 각 제품의 특성에 맞게 다른 방식으로 지원한다. 또 각기 다른 지원 방식은 다음 장에서 설명 할 방법들을 통해 서로 다른 제품들에 적용될 수 있도록 설계하였다. 따라서 OLAP 메타데이터 교환 모델은 여러 제품들의 특성을 모두 포함 할 수 있고 서로 교환 될 수 있도록 설계하였다.

먼저 전체 메타데이터의 교환 모델을 살펴보면 기본 요소인 DATABASE, CUBE, DIMENSION, MEASURE, HIERARCHY, LEVEL, MEMBER, ATTRIBUTE를 중심으로 설계된다. (그림 2)는 교환 모델의 가장 상위 구조인 DATABASE의 DTD를 보여준다.

```
<!ELEMENT OLAPDATABASE ( PROPERTY, DATASOURCE *,
  CUBE+ ) >
<!ATTLIST OLAPDATABASE
          database_name        CDATA #REQUIRED
          created_on           CDATA #IMPLIED
          last_schema_update   CDATA #IMPLIED
          schema_updated_by    CDATA #IMPLIED
>
```

(그림 2) OLAP 메타데이터 교환 모델 데이터베이스

OLAPDATABASE는 PROPERTY, DATASOURCE와 CUBE로 이루어진다. OLAP 데이터베이스에서 데이터와 구조가 상호 독립적일 경우 구조적인 정보만을 가져올 수 있기 때문에 DATASOURCE는 필요가 없어서 제품에 따라 지원하지 않는 경우가 존재 할 수 있다.

OLAPDATABASE는 하나 이상의 CUBE를 갖는다. 각 제품마다 다양한 큐브방식을 지원하고 있기 때문에 본 모델에서는 멀티 큐브방식을 지원할 수 있도록 여러 개의 큐브를 허용한다. CUBE는 PROPERTY, MEASURE와 DIMENSION으로 이루어지며 STORE는 저장구조에 관한 정보를 갖는다(그림 3). CUBE의 PROPERTY는 가상 큐브인지를 확인하는 큐브의 타입과 큐브에 관한 설명, 그 밖의 선택 사항들을 갖는다.

```
<!ELEMENT CUBE ( PROPERTY, MEASURE+, DIMENSION*,
  STORE? ) >
<!ATTLIST CUBE
        cube_name           CDATA #REQUIRED
        created_on          CDATA #IMPLIED
        last_schema_update  CDATA #IMPLIED
        schema_updated_by   CDATA #IMPLIED
>
<!ELEMENT PROPERTY ( database_name+, cube_type, cube_desc *,
  cube_guid?, is_virtual, is_drillthrough_enabled?, is_linkable?,
  is_SQL_enabled? )>
```

(그림 3) OLAP 메타데이터 교환 모델 큐브

(그림 4)의 MEASURE는 변수라고도 불리며 분석을 위한 실제 데이터 값과 유형을 갖는다. 데이터 유형은 일반적으로 숫자의 형태를 가지며 numeric_precision, numeric_units, numeric_scale과 같은 속성들을 갖는다. MEASURE의 속성들로는 집계에 사용될 기본 연산을 의미하는 measure_aggregator와 caption, unique_name 등이 있다. MEASURE의 각각의 FIELD 요소들은 분석에 사용되는 DIMENSION들을 요소로 포함한다.

```
<!ELEMENT MEASURE ( PROPERTY, FIELD+, DATATYPE ) * >
<!ATTLIST MEASURE
measure_name        CDATA #REQUIRED
created_on          CDATA #IMPLIED
last_schema_update  CDATA #IMPLIED
schema_updated_by   CDATA #IMPLIED
>
<!ELEMENT PROPERTY ( database_name+, cube_name, measure_
  unique_name?, Measure_caption?, measure_guid?, measure_desc *,
  measure_aggregator )>
```

(그림 4) OLAP 메타데이터 교환 모델 변수

(그림 5)의 DIMENSION은 HIERARCHY와 DATATYPE, ATTRIBUTE를 요소로 갖는다. 차원의 HIERARCHY는 일반적으로 한 개이지만 제품에 따라 여러 개의 HIERARCHY를 지원하기도 한다. DATATYPE 역시 지원하지 않는 제품들이 많으며 ATTRIBUTE는 특정 제품에서는 차원에서, 다른 제품들은 레벨이나 멤버에서 지원하기도 한다. PROPERTY를 살펴보면 큐브의 이름은 필수 항목이며 만일 추출된 메타데이터가 하이퍼큐브 방식이 아닌 멀티큐브 방식을 지원하는 제품으로부터 추출된 것이라면 여러 개의 값을 가질 수 있다. dimension_ordinal은 큐브를 형성하는 차원들 집합 사이의 순서 번호이며, dimension_cardinality는 차원에 속한 멤버들의 숫자이다. Dimension_type은 시간 차원, 변수 차원, 일반 차원을 구분하는 것이다.

```
<!ELEMENT DIMENSION ( PROPERTY, HIERARCHY *,
  DATATYPE?, ATTRIBUTE * ) >
<!ATTLIST DIMENSION
        dimension_name      CDATA #REQUIRED
        created_on          CDATA #IMPLIED
        last_schema_update  CDATA #IMPLIED
        schema_updated_by   CDATA #IMPLIED
>
<!ELEMENT PROPERTY ( database_name+, cube_name *,
  dimension_unique_name?, dimension_desc *, dimension_ordinal,
  dimension_type, dimension_cardinality, default_hierarchy,
  is_virtual, is_drillthrough_enabled?, dimension_unique_settings?,
  is_SQL_enabled? ) >
```

(그림 5) OLAP 메타데이터 교환 모델 - 차원

```
<!ELEMENT HIERARCHY ( PROPERTY, LEVEL *, DATATYPE?,
  ATTRIBUTE * ) >
<!ATTLIST HIERARCHY
        hierarchy_name      CDATA #REQUIRED
        created_on          CDATA #IMPLIED
        last_schema_update  CDATA #IMPLIED
        schema_updated_by   CDATA #IMPLIED
>
```

(그림 6) OLAP 메타데이터 교환 모델 - 계층구조

이 밖의 HIERARCHY는 LEVEL과 ATTRIBUTE, DATA TYPE을 요소로 가지며 차원과 유사한 구조를 갖는다 (그림 6). 이러한 XML의 DTD는 OLAP 메타데이터 교환 모델의 기본 구조만을 보여주며 여러 제품들이 다른 메타데이타 이름 및 구조를 가지므로 실제로 제품별로 적용하기 위해서는 다음 장에서 설명될 모델 적용 방법들을 필요로 한다.

〈표 1〉 OLAP 제품 모델 분석

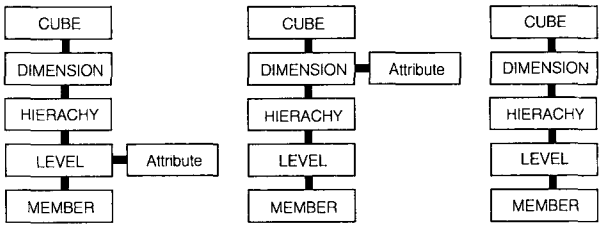
OLAP구성요소 / OLAP제품들	Storage	Cube	Attribute	Dimension
Cognos Powerplay	MOLAP	하이퍼 큐브	Not Supported	Multi level (Special category)
Hyperion Essbase	MOLAP	하이퍼 큐브	Dimension attribute	Single Hierarchy Multi level
Informix Metacube	ROLAP	블록 멀티 큐브	Supported	Single Hierarchy Multi level
MS OLAP Services	HOLAP	블록 멀티 큐브	Level Attribute	Single Hierarchy Multi level
Oracle Express	MOLAP(HOLAP)	시리즈 멀티 큐브	Not Supported	Multi Hierarchy Single level
Pilot DSS	MOLAP(HOLAP)	하이퍼 큐브	Dimension attribute	Single Hierarchy Multi level

3.1.2 모델 적용 방법

앞에서도 살펴 본 것처럼 각각의 제품들이 갖는 특수성을 고려하여 설계된 OLAP 메타데이터 교환 모델은 각 제품으로부터 추출된 메타데이터를 적용하고, 서로 다른 제품을 통해 큐브로 내보내기 위한 적용 방법들을 갖는다. 이러한 방법들은 여러 OLAP 표준들이 보이는 차이점을 극복하기 위한 OLAP 메타데이터 교환 시스템의 모듈로 구현되어졌다.

다양한 제품들이 보이는 모델의 차이는 추출된 메타데이터의 차이로 나타나며, OLAP 메타데이터 교환 모델이 이러한 모든 차이를 포함할 수 있도록 설계되어졌어도 다른 제품의 큐브로 생성될 때는 각 제품의 모델에 맞추어 변환되어야 한다.

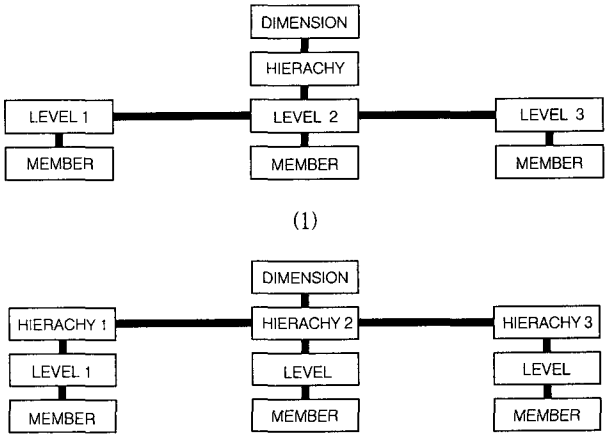
다음은 <표 1>에서 보이는 것과 같은 주요 제품 모델들의 차이를 극복하기 위한 변환 방법들을 설명한다.



(그림 7) OLAP 모델 - Attribute

• Attribute

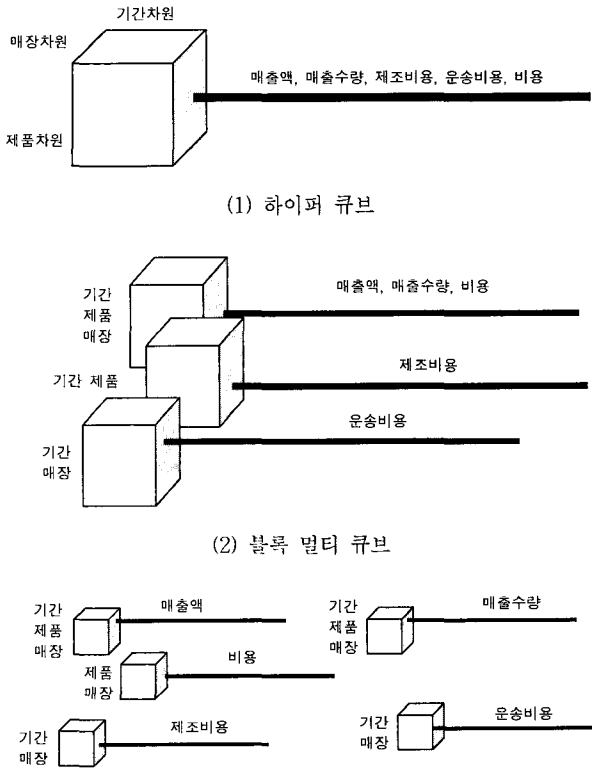
Attribute는 하나의 차원에 대해 차원을 구성하는 항목들의 특성을 나타내는 정보로 Property라고도 불리며, 질의 결과를 보여주기 위해서 사용되거나 질의에서 멤버들의 집합을 선택하기 위한 속성으로 사용된다. 그러나 (그림 7)에서 보이는 것과 같이 여러 OLAP 제품들이 모두 이 Attribute를 지원하지는 못하며 다른 이름으로 사용되기도 한다. Cognos의 Powerplay와 Oracle의 Express와 같은 제품들이 (그림 7)(3)과 같이 지원을 하지 못하고, 지원되고 있는 Informix의 Metacube나 MS의 OLAP Services, Pilot의 DSS는 서로 다른 모습을 보여준다. OLAP Services의 경우 (그림 7)(1)과 같이 차원의 레벨(Level)에서만 Attribute들을 가질 수 있다. 각각의 레벨은 자신의 Attribute집합을 가질 수 있고 따라서, 다른 레벨들은 동일한 이름으로 Attribute집합을 가질 수 있다. 또한 Attribute가 집계에 사용될 수 있도록 가상차원을 제공한다. 반면에 Pilot의 DSS는 (그림 7)(2)와 같이 차원을 설정할 때 차원이 Attribute를 갖도록 되어 있다. 이러한 다양성을 극복하기 위해서 먼저 Attribute를 지원하지 않는 제품들에 대해서는 새로운 차원을 설계하도록 한다. 또 차원 Attribute를 레벨 Attribute에 적용시키기 위해서는 가장 하위 레벨의 Attribute로 설정하도록 하며, 레벨 Attribute는 레벨이 포함되어 있는 차원의 Attribute로 표현될 수 있다.



(그림 8) OLAP 모델 - Dimension

• Dimension

차원은 일반적으로 계층구조와 레벨들을 갖는다. 계층구조는 레벨들간에 혹은 항목들 간에 존재할 수 있으며, 다차원 분석에서 매우 중요한 역할을 수행한다. 이러한 계층구조는 (그림 8)(1)과 같이 대부분 제품의 차원에서 하나의 계층 구조로 나타나며 다중 레벨을 갖는다. 그러나 Oracle



(그림 9) OLAP 모델 - Cube

의 Express의 경우는 (그림 8)(2)와 같이 기본적으로 하나의 차원에 다중 계층구조를 갖을 수 있으며 단일 레벨을 지원한다. 따라서 다중 계층구조는 다른 제품들에서 가상 차원과 같은 새로운 차원으로 변경될 수 있으며 다른 제품들이 지원하는 다중 레벨은 Express에서 사용되는 relation을 통해 정의 될 수 있다.

• Cube

다차원 데이터 구조를 논리적으로 표현하기 위한 차원들과 변수항목들의 연결 방식에 따라 하이퍼큐브(Hypercube) 방식과 멀티큐브(Multicube)방식으로 나눌 수 있다. (그림 9)에서 보는 것과 같이 하이퍼큐브방식은 하나의 모델을 하나의 큐브로 표현하기 때문에 각각의 차원들이 모든 변수 항목들에서 사용되게 된다. 그러나 멀티큐브 방식은 하나의 모델을 여러 개의 큐브로 표현하기 때문에 하나의 큐브에 의해 소유되는 것이 아니라 여러 큐브들에 의해 공유되기도 하며, 특정 큐브에 의해 사용되지 않을 수도 있다. 따라서 메타데이터 모델에서는 두 개의 큐브 형식을 모두 지원하기 때문에 차원의 속성들을 살펴보면, 큐브 이름은 null 값을 가질 수도 있으며, 여러 개의 값이 올 수도 있다. 이 중에서도 Oracle의 Express는 시리즈 멀티 큐브 방식을 사용한다. 이 방식은 모든 변수 항목들을 별개의 큐브로 다루기 때문에 각각의 변수 항목에 대해 개별적으로 차원이 설정된다. 따라서 이러한 모델의 메타데이터는 OLAP 메타데이터 교환 모델에 상세히 저장되나, 다른 제품을 통해 큐브를 생성할 때는 기본적으로 큐브가 생성될 제품의 큐브 방식으로 모델을 적용하도록 한다.

4. 시스템 구현

본 장에서는 서로 다른 OLAP 제품들간에 큐브 데이터를 교환할 수 있는 시스템을 구현하였다.

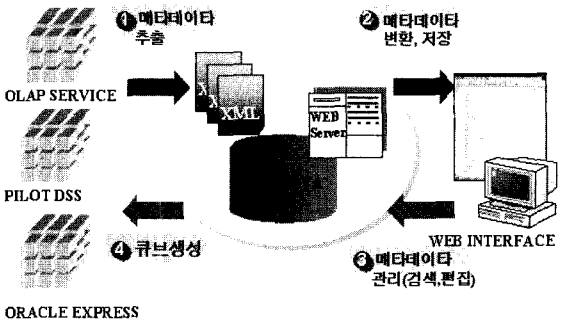
4.1. 시스템 구현 환경

본 논문에서 제안한 OLAP 메타데이터 교환 시스템의 구현 환경은 다음과 같다. 본 논문의 OLAP 메타데이터 교환 시스템은 서버측에는 Windows 2000 Server 환경을 기반으로 XML 데이터 서버인 Object Design 사의 엑셀론을 사용하였다. 엑셀론은 XML 데이터를 위하여 개발된 데이터 서버로써 XML 데이터의 동적인 구조를 저장하고 관리하는데 효율적으로 고안되었다. 존재하는 XML 데이터에 어떠한 변형 없이 그 자체로 저장이 가능하고, 쪼갬 없는 확장이 가능하여 구조화된 객체를 행과 열로 바꾸는 매핑 코드가 필요 없다. 이는 실행 도중 성능을 저해하는 조인이 필요 없음을 나타내며, 높은 성능의 향상을 보여준다.

메타데이터 추출에는 OLE DB FOR OLAP API의 COM 객체들과 인터페이스를 사용하였으며 프로그래밍 언어는 C++을 사용하였다. 사용자로부터 웹 브라우저 상에서 OLAP 메타데이터에 대한 검색 조건을 받아 이를 전송해 주는 웹 서버로는 Internet Information Server 4(IIS 4)를 사용하였으며, 웹 클라이언트로는 XML 구조를 지원하는 Microsoft의 Explorer 5.0 이상을 사용했다. 메타데이터의 검색과 편집에는 ASP, XQL, XSL을 사용하여 웹 환경을 구현하였다. 평가를 위한 OLAP 제품들로는 Microsoft의 OLAP Services, Pilot의 DSS(Decision Support Suit), Oracle의 Express를 사용하였다.

4.2 시스템 구성

시스템의 전체적인 구성은 OLAP 서버로부터 메타데이터를 추출하는 모듈과 추출된 메타데이터를 OLAP 메타데이터 교환 모델에 맞추어 XML 문서로 변환하고 XML 서버에 import 시키는 모듈, 그리고 XML 서버인 엑셀론으로부터 웹상에서 데이터를 검색, 관리할 수 있도록 하는 모듈, OLAP 서버로 큐브를 자동 생성하는 모듈로 구성된다. XML 서버와의 연결에서는 COM Client API를 사용하여 자동 Import 되도록 구현하였다. 구현한 시스템의 구성은 (그림 10)과 같이 표현될 수 있으며, 각 모듈은 그림에서 나타나는 것처럼 서로 연계되어 진행될 수 있다.



(그림 10) 시스템 구성

사용자가 메타데이터 추출 모듈의 인터페이스에서 자신이 사용한 OLAP 서버의 종류와 이름, 메타데이터를 추출하고자 하는 데이타베이스와 큐브 이름을 입력하여 메타데이터를 추출할 수 있다. 추출된 메타데이터를 XML로 변환할 것인지를 확인한 후 메타데이터는 자동으로 메타데이터 저장소인 엑셀론 서버에 저장된다. 이러한 과정을 거치면 사용자는 언제 어느 곳에서나 자신이 만든 큐브의 정보를 웹상에서 검색, 관리 할 수 있으며, 다른 OLAP 서버에서도 이러한 메타데이터를 사용할 수 있게 되는 것이다.

4.2.1 메타데이터 추출 모듈

이 모듈은 앞에서 설명된 것과 같이 OLAP 서버로부터

메타데이터를 추출하는 모듈이다. 프로그램은 C++을 사용하였으며, 다차원 데이터 저장구조에 접근하기 위해서 OLE DB for OLAP API에서 지원하는 객체들의 집합과 인터페이스를 사용했다. OLE DB for OLAP API는 OLAP의 저장구조에 따라 다른 접근을 지원하며, OLAP의 메타데이터라고 할 수 있는 다차원 구조 정보들을 IDBSchema Rowset이라는 객체에 의해 가져 올 수 있다. 여기서 Rowset이란 OLE DB 컴포넌트가 데이터를 조작할 수 있도록 한 중심 객체로 데이터의 컬럼들로 표현되는 열들의 집합이다. 이 객체로부터 3장에서 정의되어진 메타데이터를 추출하기 위해서는 Enumerator Object, Data Source Object, Session Object가 사용되며 결과로 CUBE rowset, DIMENSION rowset, HIERARCHIES rowset, LEVELS rowset, MEASURES rowset, MEMBERS rowset, PROPERTIES rowset을 얻을 수 있다.

4.2.2 메타데이터 변환 및 저장 모듈

본 논문의 앞 절에서 설명된 모듈을 통해 추출된 메타데이터를 XML 형태로 변환하는 모듈이다. XML의 스키마는 이미 정의해 놓은 OLAP 메타데이터 교환 모델의 스키마를 사용하였으며, 변환된 XML 문서는 엑셀론에서 지원하는 COM Client API를 통해 자동으로 서버에 Import되도록 하였다. (그림 11)은 변환된 XML 문서중에서 일부인 OLAP 메타데이터 교환 모델의 큐브와 차원 메타데이터가 XML로 저장된 결과를 보여주고 있다.

데이터는 비즈니스 데이터베이스의 Sales 큐브 메타데이터이며 큐브 타입은 가상 큐브가 아닌 일반 큐브 형태이다. 차원의 메타데이터를 살펴보면, 큐브의 이름은 필수 항목이며 만일 추출된 메타데이터가 하이퍼큐브 방식이 아닌 멀티큐브 방식을 지원하는 제품으로부터 추출된 것이라면 여러 개의 값을 가질 수 있다. dimension_ordinal은 큐브를 형성하는 차원들 집합 사이의 순서 번호이며, dimension_cardinality는 차원에 속한 멤버들의 숫자이다. 차원의 타입은 시간 차원, 변수 차원, 일반 차원을 구분하는 것으로 3은 일반 차원을 의미한다. 이러한 차원의 메타데이터는 계층 구조를 내부 항목으로 가지며, 계층 구조는 다시 레벨과 멤버를 내부 항목으로 갖는 구조를 XML 문서로 표현하고 있다.

```
< OLAPDATABASE database_name = "business" > …
< CUBE cube_name = "sales" created_on = "2001 04 27 오후 7 : 10 : 55" >
< PROPERTY >
< database_name > business < /database_name >
< cube_type > cube < /cube_type >
< cube_desc > business sales < /cube_desc >
< cube_guid > 7658 < /cube_guid >
< is_virtual> False < /is_virtual >
< is_drillthrough_enabled > False < /is_drillthrough_enabled >
< is_linkable> True < /is_linkable >
< is_sql_enabled > True < /is_sql_enabled >
< /PROPERTY > …
< DIMENSION dimension_name = "customer" created_on = "2001 05 01
  오후 5 : 20 : 15" >
< PROPERTY>
< database_name> business < /database_name >
< cube_name > sales < /cube_name >
< dimension_unique_name > [ customers ] < /dimension_unique_name >
< dimension_desc > customer information < /dimension_desc >
< dimension_ordinal > 6 < /dimension_ordinal >
< dimension_type > 3 < /dimension_type >
< dimension_cardinality > 10407 < /dimension_cardinality >
< default_hierarchy > region < /default_hierarchy >
< is_virtual > False < /is_virtual >
< is_drillthrough_enabled > False < /is_drillthrough_enabled >
< dimension_unique_settings > 0 < /dimension_unique_settings >
< is_sql_enabled > True < /is_sql_enabled >
< PROPERTY/ >
< HIERARCHY hierarchy_name = "region" >
… < /HIERARCHY >
… < /DIMENSION >
… < /CUBE >
< /OLAPDATABASE >
```

(그림 11) 추출된 메타데이터의 XML문서

4.2.3 메타데이터 관리 모듈

엑셀론에 저장된 메타데이터 문서를 웹을 통해 검색, 수정 가능하도록 하기 위해서 Active Server Page(ASP) 기술을 사용하였다. ASP 문서 내에서 엑셀론 서버와 연동하여 어플리케이션을 작성하기 위해서는 먼저 엑셀론 서버와 연결하여 세션을 생성한다. 생성된 세션을 통해 우리는 엑셀론 서버에 저장되어 있는 메타데이터의 XML 문서에 질의를 전달 할 수 있다[14]. 즉 사용자가 원하는 큐브의 메타데이터를 웹 문서를 통해 볼 수 있는 것이다.

엑셀론 서버에 저장되어 있는 XML 문서를 웹을 통해 보여 줄 때는 XSL 문서를 적용해야 한다. XSL 문서는 XML 문서의 스타일을 표현하기 위한 문서로 XML 문서와 함께 엑셀론 서버에 저장되어 있다. XML 문서 안에서 특정 메타데이터를 검색하기 위해서는 XQL 질의를 사용할 수 있다.

4.2.4 큐브 생성 모듈

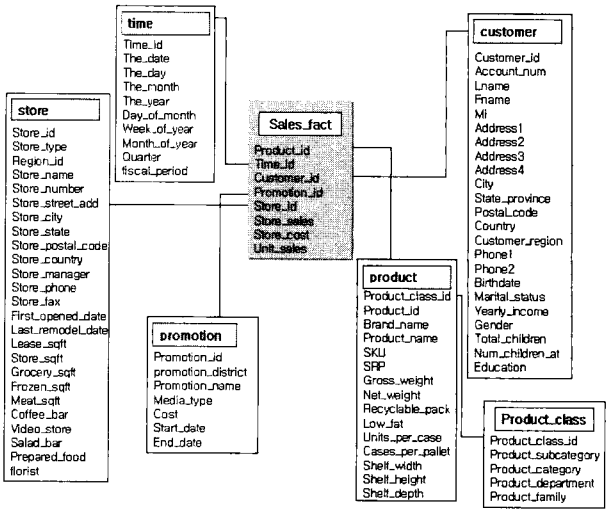
OLAP 제품들에서 사용자가 직접 수행해야 하는 큐브 생성 과정을 저장된 메타데이터를 이용해서 자동으로 처리하는 모듈이다. 각 제품들은 서로 다른 프로그램 환경을 제공하므로 각 제품의 특성에 맞는 큐브 생성 모듈을 구현하였다.

OLAP Services는 확장 가능하도록 DSO(Decision Support Objects) 라이브러리를 제공한다. DSO는 응용프로그램들이 분석 서버에 직접 연결하여 객체들을 생성할 수 있도록 한 객체 라이브러리이다. 본 시스템은 DSO 라이브러리를 사용하여 자동으로 큐브를 생성하는 모듈을 구현하였다. 큐브 생성에 필요한 변수들은 모두 저장되어 있는 메타데이터를 사용하여 만들어지며, 생성된 큐브는 Analysis Manager를 통해 사용자가 만든 큐브와 똑같이 사용 할 수 있다[15]. Pilot은 저장되어 있던 메타데이터를 사용하여 자동

으로 Model Template을 생성할 수 있도록 구현하였다. 자동 생성된 Model Template은 Pilot Model Builder를 통해 Import되어 다른 큐브들과 똑같이 사용될 수 있다[16]. Express는 저장되어 있던 메타데이터를 사용하여 ASCII 파일 형태로 큐브의 구조를 자동 생성 할 수 있도록 구현하였다. 자동 생성된 파일은 Express Administrator에 Import되어 다른 큐브들과 똑같이 사용할 수 있다[17].

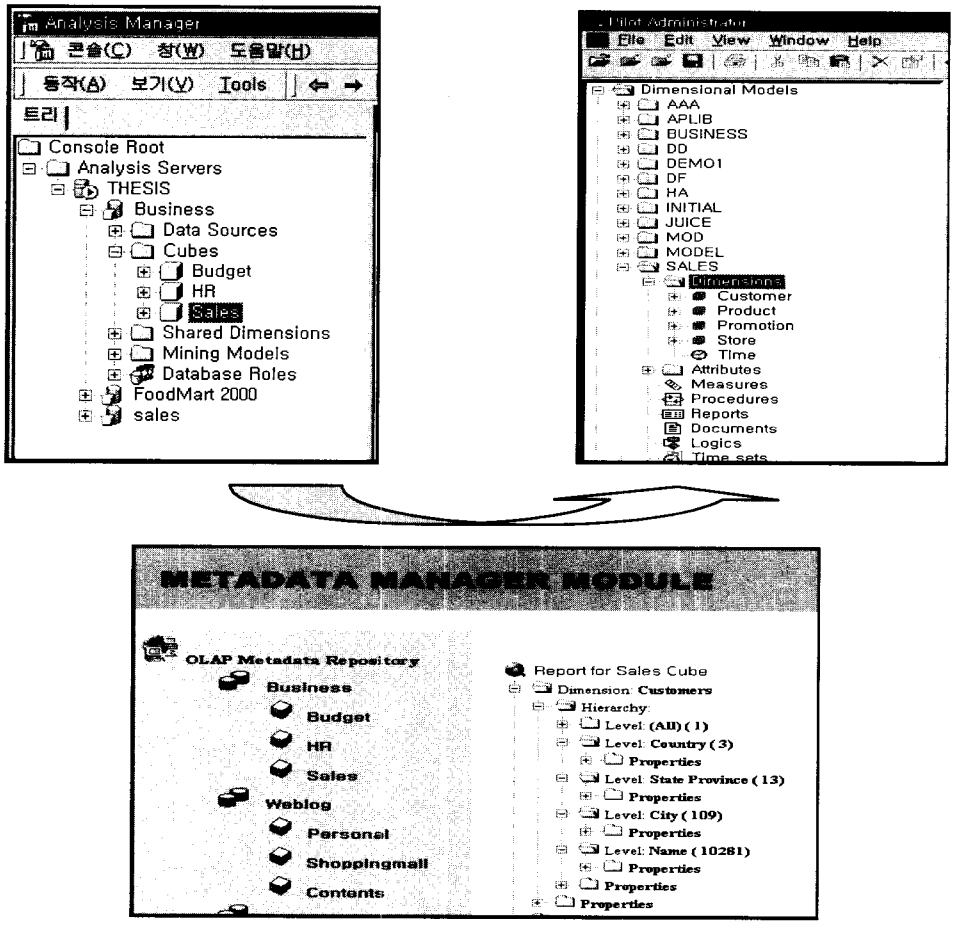
4.3. 예제를 통한 시스템 평가

구현된 시스템의 평가에는 마이크로소프트사의 OLAP Services, Pilot DSS, Oracle Express와 같은 OLAP 제품들을 사용하였다. 사용된 데이터는 기업에서 사용되는 비즈니스 데이터이다. 비즈니스 데이터베이스는 세 개의 큐브로 구성되어 있다. 세 개의 큐브는 예산에 관련된 Budget 큐브와 인사 관리에 관련된 HR 큐브, 판매와 관련된 Sales 큐브로 구성된다. 그 중에서도 Sales 큐브는 (그림 12)과 같은 스타 스키마를 갖는다. 5개의 차원테이블로 구성되었으며, Product 차원은 눈송이 스키마를 갖는다. 또한, Time과 Store 차원은 공유 차원이며 Store는 Store_type과 Store_sqft를 애트리뷰트로 갖는 레벨을 정의하였다.



(그림 12) Sales 큐브의 스키마

먼저 MS의 OLAP Services에서 큐브 마법사를 통해 Business 데이터웨어하우스로 큐브를 만든다. 데이터 원본은 SQL Server 2000에 저장되어 있는 테이블들을 선택하고 사실테이블을 정의한 후에 차원 마법사를 통해 다섯 개의 차



(그림 13) OLAP 데이터의 교환

원을 만들어 준다. 큐브 마법사를 완성하면 큐브 편집기를 통해 속성이나 식을 가진 멤버들을 정의해 줄 수 있다.

OLAP 메타데이터 교환 시스템은 인터넷을 기반으로 한 사용자 환경을 갖는다. 사용자는 첫 번째 항목을 클릭하면 자신이 추출하고자 하는 OLAP 큐브의 정보를 입력하는 화면을 볼 수 있으며, 두 번째 항목은 메타데이터가 OLAP 메타데이터 교환 모델에 맞추어 변환된 XML 문서를 볼 수 있다. 세 번째 항목은 사용자가 편리하게 메타데이터를 검색할 수 있는 화면을 제공한다.

OLAP Services에서 만들어진 Sales 큐브의 메타데이터는 메타데이터 추출 모듈을 통해 가져올 수 있다. Source는 메타데이터를 가져 올 OLAP 큐브를 설정하는 부분이고, Target은 메타데이터 저장소에 저장 될 위치를 지정하는 부분이다. 이렇게 추출된 메타데이터는 OLAP 메타데이터 교환 모델의 DTD에서 정의된 형태로 변환되어 OLAP 메타데이터 교환 시스템의 XML 서버인 엑셀론에 저장된다.

저장된 메타데이터는 검색할 수 있으며 원하는 메타데이터를 마우스로 클릭하면 보다 상세한 데이터들을 볼 수 있도록 하였다. 또 이 메타데이터는 다른 OLAP 시스템을 통해 큐브로 생성할 수 있다. (그림 13)은 OLAP Services로부터 추출된 비즈니스 데이터웨어하우스의 판매와 관련된 Sales 큐브의 메타데이터를 가지고 Pilot DSS에 큐브로 생성한 그림을 보여주고 있다.

5. 결론 및 향후과제

온라인 분석 처리(On-Line Analytical Processing : OLAP) 시스템은 데이터 웨어하우스의 효과적인 활용 방안으로 관심이 집중되고 있는 기술로서 40여개 이상의 OLAP 제품들이 공급되어 왔다. 그러나, 이들 OLAP 제품들은 서로 다른 OLAP 메타데이터, 기반 기술, 동작 방식을 가지고 개발되어 왔으며 이러한 다양성은 여러 OLAP 제품들 간에 데이터 교환과 인터페이스를 매우 어렵게 했다. 이에 본 논문에서는 서로 다른 OLAP 제품들간에 공통적으로 사용할 수 있는 OLAP 메타데이터 교환 모델을 설계하고, 그 메타데이터를 기반으로 생성된 큐브들을 교환할 수 있도록 하는 OLAP 메타데이터 교환 시스템을 구현하였다.

본 논문에서 제안한 OLAP 메타데이터 교환 시스템의 설계 및 구현은 기존의 다른 연구들에 비하여 다음과 같은 특징들을 갖는다. 첫째, 현재 OLAP 분야의 주요 과제인 표준화 과정에서 발생하는 OLAP 제품들의 차이점들을 분석하였다. 둘째, 현재 표준화된 메타데이터 모델로 확장 가능한 OLAP 메타데이터 교환 모델을 설계하여 여러 제품들이 사용하는 모델과 표준화 진행중인 모델들을 모두 포함할 수 있도록 했다. 셋째, XML기반의 메타데이터 교환 모델을 설계하여 웹 상에서 구조화된 문서를 전송 가능하도록 하고 여러 장소에서 생성된 메타데이터 정보의 교환, 통합이 용이 하도록 한다. 넷째, 사용자는 OLAP 제품들의 복잡한 큐브 생성 과정을 특별히 습득할 필요가 없으며 하나의 큐브로 여러 제품들이 제공하는 서로 다른 분석 환경을 경험할 수 있는 기회를 가질 수 있다.

현재 OLAP 분야에서 가장 많은 논란이 되는 부분으로 OLAP 기술과 관련된 표준의 정립을 들 수 있다. 본 논문의 연구를 확장하여 OLAP 제품들의 메타데이터나 인터페이스 뿐만 아니라 공통으로 사용될 수 있는 질의 언어를 설계하고 적용 할 수 있다면 OLAP 제품들간의 원활한 의사소통이 이루어질 수 있을 것으로 전망한다.

참 고 문 헌

[1] E. Thomsen, OLAP Solution, John Wiley &Sons, 1997.

[2] 조재희, 박성진, OLAP Technology, Sigma Consulting Group, 1999.

[3] Philip A. Bernstein, Brain Harry, Paul Sanders, "The Microsoft Repository," Proceeding of the 23rd VLDB Conference, Athens Greece, 1997.

[4] "Putting Metadata to Work in the Warehouse," http://www.cai.com/products/platinum/wp/wp_meta.htm.

[5] Microsoft, OLE DB for OLAP 2.0 Beta Specification, http://www.microsoft.com/data/oledb/olap/spec/.

[6] OLAP Council, MDAPI specification version 2.0, http://www.olapcouncil.org/.

[7] Cheng Hsu, Mhamed Bouziane, Laurie Rattner, Lester Yee, "Information Resources Management in Heterogeneous, Distributed Environments : Metadatabase Approach," IEEE Transaction on Software Engineering(Forthcoming), 1991.

[8] Tomas Stohr, Robert Muller, Erhard Rahm, "An Integrative and Uniform Model for Metadata Management in Data Warehouse," Proceeding of the International Workshop on Design and Management of Data Warehouses, 1999.

[9] David Marco, "Meta Data Moves Mainstream," Enterprise Warehousing Solutions, 1998.

[10] Meta Data Coalition, Open Information Model Version 1.0, http://www.mdcinfo.com/, August, 1999.

[11] Panos Vassiliadis, Timos Sellis, "A Survey on Logical Models for OLAP Databases," Technical Report, accepted for publication at SIGMOD.

[12] Daniela Florescu, Donald Kossmann, "A Performance Evaluation of Alternative Mapping Schemes for Storing XML Data in a Relational Database," INRIA Technical Report, INRIA, No.3680, May, 1999.

[13] J. C. Mamou, T. Milo, "XML repository and Active Views

Demonstration," Proceeding of the 25th VLDB Conference, Edinburgh, Scotland, pp.742-745, 1999.

[14] Vitorio Viarengo, "eXcelon XML Data Server Technical Overview," Object Exchange 98, Object Design User Conference, 1998.

[15] Developing Effective Decision Support Objects (DSO) Solutions with Microsoft SQL Server 2000 Analysis Services, http://msdn.microsoft.com/library/techart/dsosql.htm.

[16] Pilot Software, Introducing Pilot Desktop & Pilot Designer, 1998.

[17] Oracle, Oracle Express Server 6.3.2.1 Documentation, http://technet.oracle.com/software/products/exp_server/software_index.htm

이 인 기

e-mail : lig@ktf.com

1999년 이화여자대학교 컴퓨터학과 졸업 (학사)

2001년 이화여자대학원 컴퓨터학과(공학석사)

2002년~현재 KTF 정보시스템실 CRM 솔루션팀

관심분야 : 데이터웨어하우스, OLAP, XML 등

이 민 수

e-mail : mlee@ewha.ac.kr

1992년 서울대학교 컴퓨터공학과 학사

1995년 서울대 대학원 컴퓨터공학과 공학석사

1995년~1996년 LG전자 미디어통신연구소 연구원

2000년 University of Florida 컴퓨터공학과 공학박사

2000년~2002년 미국 Oracle Corporation, Senior Member of Technical Staff

2002년~현재 이화여자대학교 컴퓨터학과 전임강사

관심분야 : 데이터웨어하우스, 지식기반 시스템, 웹 데이타베이스

용 환 승

e-mail : hsyong@ewha.ac.kr

1983년 서울대학교 컴퓨터공학과 학사

1985년 서울대 대학원 컴퓨터공학과 공학석사

1985년~1989년 한국전자통신연구소 연구원

1994년 서울대 대학원 컴퓨터공학과 공학박사

1994년 서울대 컴퓨터신기술공동연구소 특별연구원

1995년~현재 이화여자대학교 컴퓨터학과 부교수

관심분야 : 멀티미디어 데이타베이스, 데이터 마이닝, 바이오정보학