

웹 서비스 기반의 클라이언트 OLAP API와 큐브 브라우징에의 응용 사례

배 은 주[†] · 김 명^{††}

요 약

이중 플랫폼간의 데이터 교환과 통합을 위한 표준 기술로 XML과 웹 서비스 기술이 각광을 받고 있다. 이러한 기술을 지원하는 XML/A는 클라이언트 응용과 데이터 분석 엔진이 인터넷 상에서 데이터를 교환할 수 있도록 하는 SOAP 기반의 XML 통신 API이다. 데이터가 XML 문서 형태로 교환된다는 점에서 XML/A는 플랫폼 독립적이지만, 클라이언트 응용 개발자들은 서버에 데이터를 요청할 때마다 XML 문서를 생성해야 하고, 서버가 보내는 XML 데이터 검색을 위해 별도의 XML 쿼리 언어를 사용해야 하는 단점이 있다. 본 연구에서는 윈도우즈 환경의 클라이언트 응용 개발자들이 XML/A의 OLAP 관련 서비스를 편리하게 사용할 수 있도록 하는 상위 수준의 클라이언트 OLAP API인 XMLMD를 설계하고 구현하였다. XML/A와 XMLMD의 관계는 OLEDB for OLAP과 ADOMD의 관계와 유사한 것으로, XMLMD는 OLAP 응용 개발에 필수적인 속성들과 메소드로 구성되어 있다. 본 연구에서는 또한 XMLMD의 응용 사례로써 데이터의 다차원적 분석결과를 인터넷을 통해 제공하는 웹 OLAP 큐브 브라우저도 개발하였다. 이 브라우저는 전송된 데이터를 XML, HTML, 엑셀, 그래프 등의 다양한 형태로 제공한다.

A Web Services-based Client OLAP API and Its Application to Cube Browsing

Eunjo Bae[†] · Myung Kim^{††}

ABSTRACT

XML and Web Services draw a lot of attention as standard technologies for data exchange and integration among heterogeneous platforms. XML/A, which supports such technologies, is a SOAP based XML API that facilitates data exchange between a client application and a data analysis engine through the Internet. The fact that the XML format is used for data exchange makes XML/A to be platform-independent. However, client application developers have to go through a tedious job of creating the same type of XML documents for downloading data from the server. Also, an XML query language is needed for extracting data from the XML documents sent by the server. In this paper, we present a high level client OLAP API, called XMLMD, for the client application developers in the Windows environment to easily use the OLAP services of XML/A. XMLMD consists of properties and methods needed for OLAP application development. XMLMD is to XML/A what ADOMD is to OLEDB for OLAP. We also present a web OLAP cube browser that is developed using XMLMD. The browser displays data in various formats such as XML, HTML, Excel, and graph.

키워드 : 온라인 데이터 분석(OLAP), OLAP API, 큐브 브라우징(Cube Browsing), 웹 서비스(Web Services)

1. 서 론

OLAP(On-Line Analytical Processing)은 정보를 다차원적으로 분석하여 그 결과를 온라인으로 제공하는 기술로써, 기업 경영자나 분석가가 기업의 전략적 방향 등을 효과적

으로 결정하는데 사용하는 비즈니스 인텔리전스 기술 중의 하나이다[6]. OLAP 시스템의 기본 기능을 예로 들면, 어떤 유통업체에서 제품을 판매한 내역이 물품, 매장, 판매시기 별로 저장되어 있을 때, 각 매장별 판매 총액이나, 매장들이 속한 지역별 판매총액, 각 매장의 제품별 판매 총액 등과 같이 다차원적 집계 결과를 분석자에게 제공하는 것들을 수 있다. OLAP 시스템들은 또한 일반적인 데이터 웨어하우스와는 달리 '우유 원가가 10원/리터가 오르고 운송비

※ 본 연구는 한국과학재단 목적기초연구(R04-2001-000-00191-0) 지원으로 수행되었음.

† 준 회 원 : 이화여자대학교 대학원 컴퓨터학과

†† 정 회 원 : 이화여자대학교 컴퓨터학과 교수

논문접수 : 2002년 10월 9일, 심사완료 : 2002년 12월 21일

용이 100원/km가 내리는 경우, 그 결과가 요구르트 가격에는 어떠한 영향을 미칠까?와 같은 예측도 OLAP 응용 개발을 통해서 가능하게 한다[6].

OLAP 응용은 초기에 클라이언트/서버 환경에서 개발되었으나 OLAP 사용자가 다양한 계층으로 확장됨에 따라 점차 웹 브라우저 인터페이스를 사용하는 웹 응용으로 개발되고 있다[1]. 웹 OLAP 응용의 개발은 HTML 파일로 미리 분석될 데이터를 만드는 방법이나 CGI 형태로 사용자 쿼리에 의해 동적으로 HTML 파일을 만드는 방법을 통해 이루어지며, 이외에 자바애플릿, 액티브엑스 컨트롤을 이용하는 방법이 있다[1].

OLAP 응용을 개발하기 위해서는 OLAP API가 필요하다. OLAP API의 역할은 OLAP 서버와 클라이언트 사이의 연결 관리, OLAP 명령문 수행, OLAP 데이터 조회 등의 기능을 하는 메소드를 클라이언트 응용에게 제공하는 것이다. 대표적인 OLAP API의 종류로는 OLE DB for OLAP[7], XML for Analysis (XML/A)[10], JOLAP (Java-based OLAP)[2]이 있으며, 이 중에서 OLE DB for OLAP이 de facto 표준으로 자리잡음으로써[9] 많은 OLAP 제품들은 OLE DB for OLAP을 지원하고 있다. 본 연구에서 취급하는 OLAP API는 XML for Analysis (XML/A) [10]이며, 이는 OLE DB for OLAP의 디자인 개념을 재사용하고, SOAP(Simple Object Access Protocol)을 기반으로 하여 이기종간의 데이터를 인터넷 상에서 교환할 수 있으며, 웹에 최적화된 OLAP API이다[10].

XML/A는 데이터의 검색만을 허용하며, 이를 수행하기 위해 Discover와 Execute 메소드를 제공한다. Discover 메소드는 메타데이터 검색을 위한 것이고, Execute 메소드는 다차원 분석 데이터의 질의처리를 위한 것이다. 클라이언트 응용과 데이터 분석 엔진 사이의 데이터 교환은 XML 문서 형태로 이루어진다. 즉, XML/A의 메소드 수행을 위해서 클라이언트 응용은 XML 소스를 만들어 XML/A provider에 분석 데이터를 요청해야 하고, 요청 결과는 XML 문서로 받게 된다. XML이 사용된다는 점에서 XML/A는 다양한 플랫폼에서 운용되는 분산 OLAP 서버의 통합 등에 용이하다는 장점을 갖는다. 그러나 데이터 검색을 위해 클라이언트는 SOAP 헤더와 호출 메소드, 매개변수와 속성 등으로 이루어진 XML 소스를 생성하여 전송해야 하고, OLAP 서버에 질의하여 얻은 결과 XML 문서내의 특정 요소(element)에 접근하기 위해서는 별도의 XML 질의 언어를 사용해야 하는 불편함이 있다. 이는 플랫폼 독립성을 유지하기 위해 클라이언트 응용이 감수해야 하는 부분이다.

본 연구에서 제안하는 XMLMD(XML-Multidimensional)는 윈도우즈 환경의 클라이언트 응용 개발자들이 XML/A

가 제공하는 기능 중에서 OLAP 관련 서비스를 손쉽게 사용할 수 있도록 캡슐화한 상위 수준의 클라이언트용 OLAP API이다. ADOMD가 OLEDB for OLAP의 상위 수준 API인 것과 같이, XMLMD는 XML/A의 상위 수준 API로써 XMLMD의 전기능을 메소드로 제공하는 것은 아니다. XMLMD는 GetCubes(), GetDimensions()와 같은 상위 수준의 함수를 제공함으로써 응용 개발자가 수월하게 OLAP 데이터에 접근할 수 있도록 하려는 것이다. XMLMD는 XML/A의 Discover 메소드에 해당하는 Catalog 오브젝트와 Execute 메소드에 해당하는 MDDataset 오브젝트로 구성되어 있다. XMLMD는 또한 mdDiscover() 함수와 mdExecute() 함수를 제공하여 XML/A의 부수적인 기능을 처리할 수 있도록 한다. XMLMD가 제공하는 메소드들은 XML/A Provider에게 전송하는 XML 소스를 자동으로 생성하여 서비스를 요청하고, 수행 후 반환되는 결과를 XML 형태나 각 데이터에 접근하기 쉬운 변수 형태로 변환하여 돌려준다. XMLMD는 Visual C++, Visual Basic 등의 언어를 지원한다.

본 연구에서는 또한 XMLMD를 사용하여 웹 OLAP 큐브 브라우저인 WOCB (Web OLAP Cube Browser)를 응용 사례로 개발함으로써 XMLMD의 편리성을 검증하였다. WOCB는 OLAP 서버에 계산되어 저장되어 있는 다차원적 집계 결과(또는 OLAP 큐브)를 웹 상에서 브라우징하는 틀이다. WOCB는 큐브의 데이터를 HTML, XML, 엑셀과 그래프의 형태로 나타내 준다.

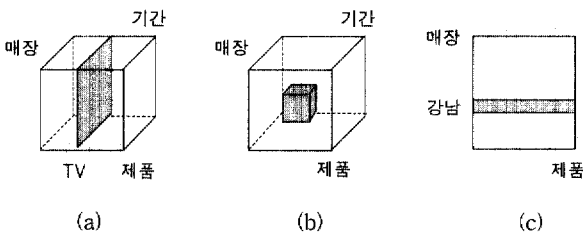
본 논문의 구성은 다음과 같다. 2절에서 OLAP의 기본 기능을 소개하고, 3절에서 OLAP과 웹 서비스를 설명한다. 4절에서 XMLMD의 근간이 되는 XML/A를 기술하고, 5절에서 XMLMD의 구조와 구현에 대해 설명한다. 6절에서 WOCB의 구현 내용, 사용 방법에 대해 기술하고, 결론 및 향후과제를 7절에서 설명하기로 한다.

2. OLAP의 기본 기능

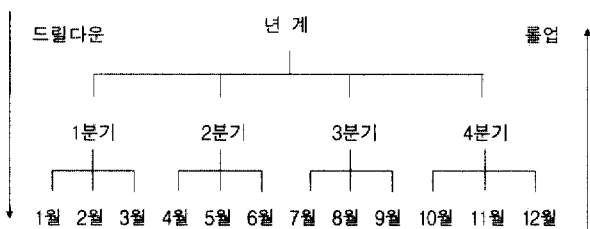
다차원 데이터는 다수의 차원 정보와 측정값으로 구성된다. 예를 들어, 유통업체의 판매 정보는 제품, 매장, 판매시기 등의 차원 정보와 판매액, 판매량 등의 측정값으로 구성될 수 있다. 각 차원은 또한 계층 구조로 설계될 수 있다. 예를 들어, 매장 차원은 매장 → 지역 → 국가의 구조를 가질 수 있고, 판매시기는 날짜 → 주 → 월 → 분기 → 년 구조를 가질 수 있다. OLAP 서버의 기본 역할은 이와 같은 데이터를 다차원적으로 집계 연산하여 그 결과를 큐브에 저장해 놓고, 큐브 데이터의 일부를 사용자에게 온라인으로 제공하는 것이다.

OLAP의 기본 연산을 예를 들어 살펴보기로 한다. OLAP의 주요 연산으로는 다차원 데이터의 슬라이스(slice), 다이스(dice), 드릴다운(drill-down), 롤업(roll-up) 연산이 있다 [8]. (그림 1)은 '판매' 데이터 상에서 이러한 연산 결과들을 나타낸다. '판매' 데이터는 '제품', '매장', '기간' 차원과 '판매액' 측정값을 갖는 3차원 데이터이다. 사용자가 TV 제품의 매장별, 기간별 판매액을 알고자 하는 경우는 (그림 1)(a)의 음영 부분에 해당하는 데이터를 슬라이스하여 볼 수 있다. 또한 각 차원별로 특정 범위에 속하는 데이터를 선택하는 다이스 연산 (그림 1)(b)을 통해 '강남' 지역 매장의 '가전' 제품들의 '2001년' 판매액을 분석할 수 있다. (그림 1)(c)는 '판매' 데이터를 기간 차원으로 집계한 2차원 집계 데이터를 보이며, 이러한 집계 데이터 상에서도 슬라이스와 다이스 연산을 할 수 있다.

각 차원에 계층구조를 둘 수 있다. (그림 2)는 '판매' 큐브의 기간 차원에 줄 수 있는 계층구조를 나타낸다. 사용자는 월별 데이터를 분석할 수도 있고, 분기별 데이터를 분석할 수도 있다. 월별 데이터를 분석하다가 요약된 형태의 분기별 데이터를 보고자 할 때는 롤업 연산을 사용하고, 반대로 분기별 데이터를 분석하다가 자세하게 월별 데이터를 보고자 할 때는 드릴 다운 연산을 사용한다. OLAP 응용은 다양한 인터페이스를 통해 큐브의 슬라이스, 다이스, 드릴다운, 롤업의 OLAP 연산 기능을 사용자가 수행할 수 있도록 개발되어야 한다. 이는 OLAP API에서 제공하는 메소드를 사용함으로써 쉽게 개발될 수 있다.



(그림 1) '판매' 큐브 상에서의 OLAP 기본 연산



(그림 2) 기간 차원의 계층구조와 드릴다운, 롤업 연산

3. OLAP과 웹 서비스

최근 들어 OLAP 사용자층이 확대되고, 조직 내의 IT 환

경이 점차 웹 환경으로 옮겨가면서 인트라넷 환경의 제한된 네트워크 안에서 클라이언트-서버 시스템으로 개발되던[1] OLAP 응용이 웹 기반으로 개발되는 추세이다. 웹 기반의 OLAP 응용은 웹 브라우저를 이용하여 OLAP의 기능을 수행하며, 다음과 같은 형태로 개발되고 있다. 첫째는 HTML 파일로 분석될 데이터를 미리 생성해 놓는 방법이다. 이는 정적인 문서를 나타내는 방법으로 큐브 데이터가 수정되면 다시 파일을 생성해야 하는 단점이 있다. 둘째는 사용자 쿼리에 의해 HTML 파일을 동적으로 생성하는 방법으로, OLAP 명령문을 이용하여 수정된 큐브의 데이터를 실시간으로 가져오는 장점이 있다. 최근 가장 많이 사용되는 방식은 자바 애플릿(Java Applet), 액티브엑스(ActiveX) 컨트롤 등을 이용하는 방법으로써, 웹 브라우저에 더욱 많은 기능을 부여하여 사용자와의 상호작용을 증대시킨다[1].

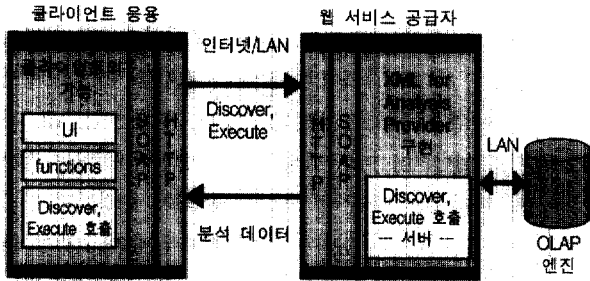
OLAP 제품들은 OLAP 응용 개발을 지원하기 위하여 OLAP API를 제공한다. OLAP API는 OLAP 연산을 수행하는 메소드를 OLAP 응용에게 제공하는데, 예를 들면, OLE DB for OLAP은 SQL과 유사한 MDX(Multidimensional Expressions) 언어를 제공한다. MDX는 Select 절과 From 절을 사용하여 분석할 큐브와 차원들을 선택할 수 있도록 하고, Where 절을 사용하여 큐브의 단면을 슬라이스할 수 있도록 한다[4]. 이러한 기능 이외에도 OLAP API는 OLAP 서버와 클라이언트의 연결을 관리하고, OLAP 메타데이터를 검색할 수 있는 기능의 메소드를 클라이언트 응용에게 제공한다. 대표적인 OLAP API들로는 OLE DB for OLAP[7], JOLAP[2]과 XML for Analysis(XML/A)[10] 등이 있다.

XML/A는 XML 웹 서비스[5]를 지원하는 API로써, SOAP, XML, HTTP 등의 개방형 인터넷 표준을 따른다[10]. 웹 서비스는 표준화된 XML 메시지를 통해 서로 다른 플랫폼 사이에 데이터를 공유하고 통신하기 위한 기술로써, 인터넷 상에서 분산된 서비스 모듈 형태로 운영된다[3]. 따라서, 기업 내의 다양한 플랫폼을 갖는 OLAP 서버간의 데이터 교환과 통합을 용이하게 한다. 본 연구에서는 XML/A의 웹 서비스 기능을 지원하면서, 보다 쉽게 사용할 수 있도록 캡슐화한 XMLMD OLAP API를 개발하였다.

4. XML/A

XML/A의 구조[10]는 (그림 3)과 같이 표현될 수 있다. 클라이언트 응용은 SOAP과 HTTP 프로토콜을 사용하여 서버의 웹 서비스 URL과 함께 Discover, Execute 메소드를 호출하는 XML 문서를 서버에 보낸다. 서버는 XML/A Provider 인스턴스를 만들어서, 데이터 원본과 통신하며 요청 받은 메소드를 처리하고, 결과 데이터를 XML로 패키징화

한 후에 클라이언트한테 보낸다.



(그림 3) XML/A의 구조

XML/A는 Discover와 Execute의 2개의 메소드로 구성된다. Discover는 메타데이터를 검색하는데 사용되는 메소드이다. Discover를 사용하여 OLAP 데이터 원본의 목록, OLAP 큐브 정보, 차원 정보, 차원 레벨 정보, 측정값 등의 메타데이터를 검색할 수 있다. Execute는 데이터 검색을 위한 메소드이다. 이는 OLAP 큐브 상에서 슬라이스, 다이, 드릴 다운, 롤업 등의 주요 연산을 처리하는데 사용된다.

Discover와 Execute 메소드의 구문은 (그림 4)와 같다. Discover 메소드는 3개의 입력 매개 변수를 갖는다. RequestType은 검색하고자 하는 OLAP 메타데이터의 종류를 나타내고, Restrictions는 검색 대상에 대한 특정 제한조건을 지정하는데 쓰이며, Properties는 데이터 원본의 정보나 디스플레이 형식 등의 속성을 지정할 때 쓰인다. 검색 결과는 Result로 받는다. Execute 메소드는 두 개의 입력 매개 변수를 갖는다. OLAP 큐브 상에서 수행될 연산은 MDX 문으로 표현되어 Command 매개 변수를 통해 전달된다. 데이터 소스 등과 같은 속성은 Properties를 통해 전달되며, 검색 결과 데이터는 OLAP 큐브내의 위치 정보, 측정 값과 포맷된 결과 값 형태로 Result 매개 변수를 통해 받는다.

Discover (Execute (
[in] RequestType As EnumString,	[in] Command As Command,
[in] Restrictions As Restrictions,	[in] Properties As Properties,
[in] Properties As Properties,	[out] Result As Resultset)
[out] Result As Rowset)	

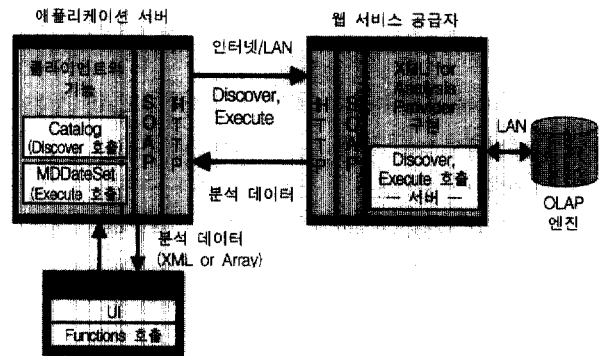
(그림 4) Discover, Execute 메소드 구문

5. XMLMD

XMLMD(XML-Multidimensional)는 윈도우즈 환경의 클라이언트 응용 개발자들이 XML/A에서 지원하는 OLAP 인터페이스를 보다 쉽게 사용할 수 있도록 캡슐화한 상위 수준의 클라이언트용 OLAP API이다. XML/A는 XML 웹 서비스를 기반으로 하기 때문에, 다양한 하드웨어 플랫폼과

언어를 지원하지만, 데이터와 서비스에 관한 구체적인 내용을 포함하는 XML 소스를 매번 생성해야 하고, 수행 후 반환된 결과 XML 문서의 데이터 요소에 접근하기 위해서는 XML 질의 언어를 사용해야 하는 불편함이 있다. XMLMD는 윈도우즈 플랫폼 환경에서 이러한 불편함을 최소화하고자 설계된 API로써, 클라이언트가 전송할 XML 코드를 조건에 맞게 자동으로 생성하고, 수행 결과를 XML 또는 데이터 요소에 쉽게 접근할 수 있는 배열 변수에 저장하여 반환하도록 설계되었다.

(그림 5)는 XMLMD의 구조이다. 클라이언트 응용에서 Visual Basic 형태의 XMLMD 함수를 호출하면, XMLMD는 해당 함수를 XML/A의 Discover, Execute 메소드 구문 형태로 변환하여 XML/A Provider에 전송하고, 결과로 받은 XML 데이터를 배열 형태로 변환하여 응용에게 반환한다. XMLMD를 사용하면 클라이언트 응용 개발자는 XML/A Provider를 통해 교환하는 XML 데이터 구조를 모르더라도 쉽게 OLAP 응용을 개발할 수 있다.



(그림 5) XMLMD 구조

XMLMD는 Catalog, MDDataSet 두 개의 오브젝트로 구성된다. Catalog 오브젝트는 XML/A의 Discover 메소드에 대응되는 오브젝트로서, 데이터베이스 카탈로그, 큐브, 차원, 측정값, 계층, 레벨, 멤버 등의 OLAP 메타데이터를 검색할 때 쓰인다. XML/A의 Discover와 XMLMD의 Catalog의 대응 관계는 <표 1>에 나타나 있다. XML/A의 Discover 메소드는 검색할 메타데이터의 종류를 RequestType 매개 변수를 통해 지정한다. RequestType 중에서 OLAP 관련 주요 메타 데이터는 <표 2>의 2열에 나타나 있는 것과 같이 6개 종류로 나뉜다. XMLMD의 Catalog 오브젝트는 이 각각의 RequestType에 대하여 <표 2>의 3열과 같은 메소드를 통하여 메타데이터의 종류를 지정하도록 하였다. (그림 6)에 Catalog 오브젝트가 지원하는 메소드의 리스트가 주어 져 있다. mdDiscover(), GetCatalogs(), GetFunctions()가 추가되어 있는 것을 알 수 있다. GetCatalogs()는 데이

터베이스 카타로그를 반환하는데 쓰이고, GetFunctions()는 MDX가 지원하는 함수 리스트를 반환하는 메소드이다. mdDiscover()는 XML/A의 Discover 메소드가 지원하는 모든 오브젝트를 검색할 수 있도록 하는 부가적인 메소드이다. 클라이언트 응용 개발자들이 RequestType과 Restrictions만을 지정해 주면 대응되는 XML/A의 Discover 메소드를 포함하는 완전한 XML 요청 문서를 생성해 주는 역할을 한다. XML/A의 Discover 메소드의 Restrictions 매개 변수는 RequestType에 따라 다르다. 이는 Catalog 오브젝트의 해당 메소드의 매개 변수로 처리하였고 Discover의 Properties는 <표 1>에서와 같이 DBServer, Format, Content 속성으로 선언하였다.

<표 1> Discover와 Catalog와의 대응 관계

Discover	Catalog
RequestType 예 : MDSHEMA_CUBES, MDSHEMA_DIMENSIONS, ...	함수 예 : GetCubes(), GetDimensions(), ...
Restrictions 예 : <RestrictionsList> <CATALOG_NAME> ... </CATALOG_NAME> </RestrictionsList>	매개변수 예 : GetCubes(ByRef CatalogName, ByRef Cubes, ByRef CubeType, ByRef CubeDesc, Catalog)
Properties 예 : DataSourceInfo, Format, Content	전역변수 예 : DBServer, Format, Content
Result 예 : <row> <CATALOG_NAME> ... </CATALOG_NAME> <CUBE_NAME> ... </CUBE_NAME> </row>	매개변수 (ByRef ...) 예 : GetCubes(ByRef CatalogName, ByRef Cubes, ByRef CubeType, ByRef CubeDesc, Catalog)

<표 2> XML/A와 XMLMD의 OLAP 메타데이터 검색 기능

OLAP 메타데이터	XML/A (Discover-RequestType)	XMLMD (Catalog)
큐브	MDSHEMA_CUBES	GetCubes()
차원	MDSHEMA_DIMENSIONS	GetDimensions()
측정값	MDSHEMA_MEASURES	GetMeasures()
계층	MDSHEMA_HIERARCHIES	GetHierarchies()
레벨	MDSHEMA_LEVELS	GetLevels()
멤버	MDSHEMA_MEMBERS	GetMembers()

Catalog : OLAP 오브젝트를 검색
DBServer : DB 서버의 위치 (default : local)
Format : 반환되는 결과집합의 형식 (default : Tabular)
Content : 결과로 반환되는 데이터의 형태 (default : SchmaData)
mdDiscover (RequestType, Restrictions) : XML/A의 Discover 메소드 수행하여 결과 XML을 반환

GetCatalogs (ByRef Catalogs, ByRef Desc, ByRef Roles, ByRef DataModified)
GetCubes (ByRef CatalogName, ByRef Cubes, ByRef CubeType, ByRef CubeDesc, Catalog)
GetDimensions (ByRef CatalogName, ByRef CubeName, ByRef Dimensions, ByRef DimUName, ByRef DimCaption, ByRef DimOrdinal, ByRef DimType, ByRef DimCardinality, Catalog, Cube)
GetMeasures (ByRef CatalogName, ByRef CubeName, ByRef Measures, ByRef MeaCaption, ByRef MeaAggregator, ByRef NumPrecision, ByRef NumScale, ByRef LevelsList, ByRef SQLCol, Catalog, Cube)
GetHierarchies (ByRef CatalogName, ByRef CubeName, ByRef DimUName, ByRef HName, ByRef HUName, ByRef HCaption, ByRef HCardinality, ByRef DefaultMember, ByRef AllMember, ByRef HOrdinal, ByRef DimensionIsShared, Catalog, Cube, Dimension)
GetLevels (ByRef CatalogName, ByRef CubeName, ByRef DimUName, ByRef HUName, ByRef Levels, ByRef LevelUName, ByRef LevelCaption, ByRef LevelNum, ByRef LevelCardinality, ByRef LevelType, Catalog, Cube, Dimension)
GetMembers (ByRef Members, ByRef LevelNumber, ByRef LevelUName, ByRef MemberUName, ByRef ParentUName, ByRef MemberCaption, Catalog, Cube, Dimension, Option LevelNum)
GetFunctions (ByRef FunctionNames, ByRef Description, ByRef ParameterList, ByRef ReturnType, ByRef InterfaceName, ByRef Caption)

(그림 6) Catalog 오브젝트의 속성과 메소드

XML/A의 Discover 메소드와 XMLMD의 Catalog 오브젝트의 사용 예를 비교해 보자. 그 예로, "FoodMart 2000" 카탈로그의 "Sales" 큐브에 속한 차원의 목록을 검색하려고 한다. XML/A는 (그림 7)과 같이 Discover 메소드를 사용하여 XML 소스를 작성한 후에 이를 XML/A Provider에 전송한다. 소스의 맨 위에 SOAP Envelope과 SOAP Body가 있다. OLAP 메타데이터인 "차원" 정보를 검색하기 위해서 Discover 메소드를 호출하는데, 이것이 박스 (1) 부분이다. 예제에 대한 매개변수 값을 살펴보면, RequestType 변수에 "MDSHEMA_DIMENSIONS"을 지정하는데, 이것이 박스 (2) 부분이다. 박스 (3)은 Restrictions 변수로써, <CATALOG_NAME>에 "FoodMart 2000", <CUBE_NAME>에 "Sales"를 지정한다. Properties 변수에는 데이터 원본 정보와 반환될 데이터의 형식을 지정한다. 이를 XML/A Provider에 요청을 하면, 서버는 "Sales" 큐브의 차원 정보 목록을 테이블 형식의 XML 문서로 클라이언트에게 반환한다.

```
<SOAP-ENV: Envelop xmlns :
  SOAP-ENV = "http://schemas.xmlsoap.org/soap/envelope"
  xmlns : xsi = "http://www.w3.org/2001/XMLSchema-instance"
```

```

xmlns : xsi = "http://www.w3.org/2001/XMLSchema">
<SOAP-ENV : Body>
(1) <Discover> <xmlns = "urn : schemas-microsoft-com : xml-analysis">
    SOAP-ENV : encodingStyle = "http://schemas.xmlsoap.org
        /soap/encoding">
(2) <RequestType> MDSHEMA_DIMENSIONS </RequestType>
    - <Restrictions>
(3) - <RestrictionList>
        <CATALOG_NAME> FoodMart 2000 </CATALOG_NAME>
        <CUBE_NAME> Sales </CUBE_NAME>
    </RestrictionList>
    </Restrictions>
    - <Properties>
    - <Properties>
    - <PropertyList>
    <DataSourceInfo> Provider = MSOLAP ; Data
        Source = local </DataSourceInfo>
    <Format> Tabular </Format>
    <Content> SchemaData </Content>
    </PropertyList>
    </Properties>
</Discover>
</SOAP-ENV ; Body>
</SOAP-ENV ; Envelope>
    
```

(그림 7) XML/A의 Discover 메소드 소스

(그림 8)은 이에 해당하는 XMLMD 코드이다. 차원 정보를 얻기 위해 Catalog 오브젝트의 GetDimensions() 함수가 호출되었고, 이 메소드는 내부적으로 XML/A Provider에게 보낼 XML 소스를 생성하여 원하는 데이터를 요청하고, 요청된 결과를 XML 형태와 배열 형태로 반환한다. 변수에 할당된 데이터는 클라이언트 응용에서 바로 접근하여 조작할 수 있다는 장점을 갖는다.

```

Set objCat = Server.CreateObject ("XMLMD.Catalog")
xmlResult = objCat.GetDimensions(arrCatalogName, arrCubeName,
    arrDimensions, arrDimUName, arrDimCaption,
    arrDimOrdinal, arrDimType, arrDimCardinality,
    "FoodMart 2000," "Sales")
    
```

(그림 8) XMLMD의 Catalog 오브젝트 소스

XMLMD의 MDDataset 오브젝트는 XML/A의 Execute 메소드에 대응하는 오브젝트로서, OLAP 명령문을 사용하여 큐브 데이터를 검색할 때 사용하는 오브젝트이다. XML/A의 Execute 와 XMLMD의 MDDataset의 대응 관계는 <표 3>에 주어져 있다. XML/A의 Execute는 Command와 Properties 2개의 입력 매개 변수를 갖는다. Command는 OLAP 큐브의 데이터를 검색하는데 쓰이는 MDX 문을 전달하는 데 쓰이고, Properties는 연산 수행에 관련된 속성을 전달하는데 쓰인다. 연산 실행 후 반환되는 큐브 데이터는 Result 매개 변수를 통해 전달되는데 이는 (그림 12)의 박스 (1)에

서와 같이 데이터의 큐브내의 상대 주소 (CellOrdinal)와 데이터의 값, 데이터의 포맷된 값 형태를 갖는다. XMLMD는 MDDataset 오브젝트의 GetDataSet 메소드를 통해서 MDX 문을 실행하도록 한다. 클라이언트 응용 개발자가 제공하는 MDX문은 GetDataSet의 Command 매개 변수를 통해서 전달된다. 연산 수행 결과 전달되는 큐브의 데이터는 데이터의 위치 정보 (차원 축 정보)와 함께 배열 형태로 전달되어 클라이언트 응용에서 직접 액세스하여 사용할 수 있도록 하였다. Execute의 속성들은 MDDataset의 DBServer, Format, Content 전역 변수를 통해 전달하도록 하였다. XMLMD의 Catalog 오브젝트와 마찬가지로 MDDataset 오브젝트 역시 XML/A의 Execute 메소드를 직접 사용할 수 있도록 mdExecute 메소드를 제공한다. 이를 사용하면 큐브 데이터를 XML 형태 그대로 전달 받을 수도 있도록 하였다. (그림 9)는 MDDataset 오브젝트의 속성과 메소드 목록이다.

<표 3> Execute와 MDDataset과의 관계

Execute	MDDataset
Command 예 : <statement> SELECT ... </statement>	매개변수 예 : mdExecute(Command, Catalog) :
Properties 예 : DataSourceInfo, Format, AxisFormat, Content	전역변수 예 : DBServer, Format, AxisFormat, Content
Result 예 : <olapinfo> ... </olapinfo> <Axes> ... </Axes> <CellData> ... </CellData>	매개변수 (ByRef ...) 예 : GetDataSet(Command, Catalog, ByRef AxisCaption, ByRef AxisLNum, ByRef CellValue, ByRef CellFmtValue)

```

MDDataset : OLAP 큐브 데이터를 검색
DBServer : DB 서버의 위치 (default : local)
Format : 반환되는 결과집합의 형식 (default : Multidimensional)
Content : 결과로 반환되는 데이터의 형태 (default : Data)
AxisFormat : 축의 형식 (default : TupleFormat)
mdExecute (Command, Catalog) : XML/A의 Execute 메소드 수
    행하여 결과 XML을 반환
GetDataSet (command, Catalog, ByRef AxisCaption ByRef
    AxisLNum, ByRef CellValue, ByRef CellFmtValue) : MDX
    문의 수행결과를 반환
    
```

(그림 9) MDDataset 오브젝트의 속성과 메소드

XML/A의 Execute 메소드와 XMLMD의 MDDataset 오브젝트를 사용하여 큐브의 데이터를 검색하는 코드를 비교해보자. (그림 10)은 XML/A의 Execute 메소드의 소스를 보여준다. 이는 '측정값'을 열 축으로 하고, 'Account' 차원을 행 축으로 하여 'Budget' 큐브의 데이터를 분석하는 예제로써 OLAP 명령문으로 MDX 언어를 사용한다. Execute 메소

드의 매개변수를 살펴보면, Command 변수에 OLAP 명령 문장을 지정하고, Properties 변수에는 데이터 원본 정보와 카탈로그 정보, 반환될 데이터의 형식 등을 지정한다. 이에 대응하는 XMLMD 소스는 (그림 11)과 같다. MDDataset 오브젝트를 생성한 후 GetDataSet() 함수를 호출하면, XMLMD는 축의 캡션 정보와 레벨 번호, 셀 데이터의 값과 셀의 포맷된 값을 변수에 저장하여 반환한다.

```

<SOAP-ENV:Envelope xmlns:
SOAP-ENV = "http://schemas.xmlsoap.org/soap/envelope"
xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd = "http://www.w3.org/2001/XMLSchema">
<SOAP-ENV:Body>
(1) <Execute> <xmlns = "urn:schemas-microsoft-com:xml-analysis"
SOAP-ENV:encodingStyle = "http://schemas.xmlsoap.org
/soap/encoding">
(2) <Command>
<Statement> SELECT ([Measures].members) ON COLUMNS,
{DrilldownLevelTop ([Account].[All].members, 3)}
ON ROWS FROM [Budget] WHERE ([Category].[All Category], [Store].[All Stores], [Time].[1997])
</Statement>
</Command>
<Properties>
<PropertyList>
<DataSourceInfo> Provider = MSOLAP ; Data
Source = local </DataSourceInfo>
<Catalog> Foodmart 2000 </Catalog>
<Format> Multidimensional </Format>
<AxisFormat> TupleFormat </AxisFormat>
<Content> Data </Content>
</PropertyList>
</Properties>
</Execute>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
    
```

(그림 10) XML/A의 Execute 메소드 소스

```

strCommand =
"SELECT ([Measures].members ) ON COLUMNS,
{DrilldownLevelTop ([Account].[All].members, 3)} ON ROWS
FROM [Budget]
WHERE ([Category].[All Category], [Store], [All Stores],
[Time].[19 97])"
Set objMD = Server.CreateObject ("XMLMD.MDDataset")
xmlResult = objMD.GETDataSet (strCommand, "FoodMart 2000,"
AxisCaption, AxisNum, CellValue, CellFmtValue)
    
```

(그림 11) XMLMD의 MDDataset 오브젝트 소스

(그림 12)는 (그림 10)의 소스를 수행하여 얻은 결과 XML이다. Execute 메소드의 결과는 "OlapInfo", "Axes", "CellData"의 세가지 요소로 구성되어 반환된다. "OlapInfo"는 축(Axis), 셀(Cell)과 같은 결과의 구조를 정의하고, "Axes"는 OlapInfo에 정의된 축의 데이터, "CellData"는 셀의 데이터를 포함

하고 있다. (그림 12)(1)은 CellData 요소의 데이터로써 위치정보와 값을 가지고 있는데, 클라이언트 응용에서 이들 값에 직접 접근하여 데이터를 조작하기 위해서는 별도의 XML 질의 언어를 사용해야 한다. 하지만, XMLMD는 축 데이터와 셀 데이터를 배열 변수에 저장하여 반환하기 때문에 클라이언트 응용에서 이들 데이터에 접근하기가 매우 쉽다.

XMLMD의 Catalog, MDDataset 오브젝트와 XML/A의 Discover, Execute 메소드를 사용 예를 들어 비교하여 보았다. XMLMD는 윈도우즈 환경의 클라이언트 응용 개발자들이 XML/A에서 지원하는 OLAP 인터페이스를 보다 쉽게 사용할 수 있도록 캡슐화한 상위 수준의 클라이언트용 OLAP API이다. XMLMD는 Discover 메소드에 대응하는 Catalog 오브젝트와 Execute 메소드에 대응하는 MDDataset 오브젝트로 구성되며, 기본적으로 XML/A의 메소드를 모두 수행할 수 있도록 mdDiscover() 함수와 mdExecute() 함수를 제공하고 있다. 또한 XMLMD는 보다 상위 수준의 함수를 제공함으로써 응용 개발자가 더욱 쉽게 OLAP 데이터에 접근할 수 있도록 하였다. XMLMD를 이용하면 XML/A Provider에게 보내는 XML 문서의 자동 생성과 함께 XML 쿼리 언어를 사용하지 않고도 결과 데이터에 접근할 수 있다. 이는 개발자들이 OLAP 응용을 개발할 때 XML 문서의 구조를 모르고도 개발할 수 있다는 장점과 함께 개발 시간을 훨씬 단축시킬 수 있다는 잇점이 있다.

```

<?xml version = "1.0">
<SOAP-ENV:Envelope xmlns:
SOAP-ENV = "http://schemas.xmlsoap.org/soap/envelope"
SOAP-ENV:
encodingStyle = "http://schemas.xmlsoap.org/soap/encoding/">
<SOAP-ENV:Body>
<m:ExecuteResponse xmlns:
m = "urn:schemas-microsoft-com:xml-analysis">
<return xsi:type = "xsd:string"
xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd = "http://www.w3.org/2001/XMLSchema">
<root xmlns = "urn:schemas-microsoft-com:xml-analysis:
mddataset"
xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd = "http://www.w3.org/2001/XMLSchema">
<OlapInfo>
+ <AxesInfo>
+ <cellInfo>
</OlapInfo>
+ <Axes>
(1) <CellData>
<Cell CellOrdinal = "0">
<Value xsi:type = "xsd:double"> 398755.69 </Value>
<FmtValue> ₩ 398.765 </FmtValue>
</Cell>
<Cell CellOrdinal = "3">
<Value xsi:type = "xsd:double"> 398755.69 </Value>
<FmtValue> ₩ 398.765 </FmtValue>
</Cell>
</CellData>
    
```

```

</root>
</return>
</m : ExecuteResponse>
</SOAP-ENV : Body>
</SOAP-ENV : Envelope>
    
```

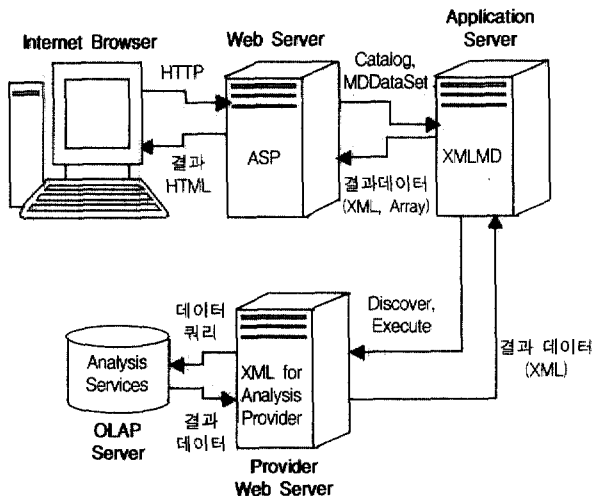
(그림 12) XML/A의 Execute 메소드 수행 결과 XML

6. WOCB

WOCB(Web OLAP Cube Browser)는 다차원 데이터 분석을 위하여 인터넷 상에서 OLAP 큐브의 데이터를 검색하는 툴이다. 이는 윈도우즈 플랫폼 환경에서 개발되었으며, 웹 서버는 IIS(Internet Information Services) 5.0, OLAP 서버는 SQL Server 2000 Analysis Services, XML 파서는 MS의 XML Parser 3.0을 사용하였다. 사용자 인터페이스의 작성은 ASP(Active Server Page)를 사용하였고, OLAP 서버와 데이터 통신을 하는 부분은 기본적으로 XML/A의 인터페이스를 사용하며, 클라이언트에서는 XMLMD OLAP API를 사용하였다. 이 절에서는 WOCB의 시스템 구조와 실행 방법, 구현 내용에 대하여 설명한다.

6.1 WOCB 시스템 구조

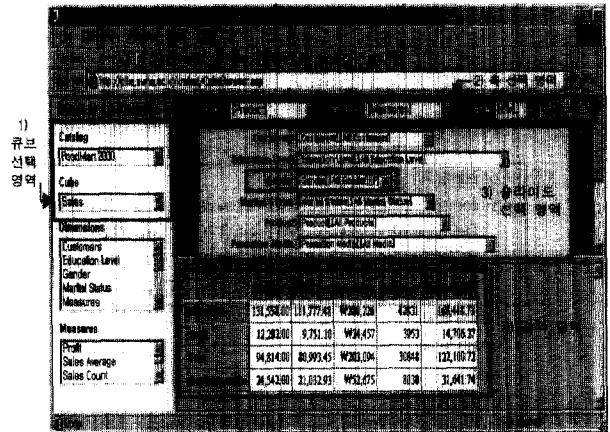
WOCB의 시스템 구조는 (그림 13)으로 표현될 수 있다. 사용자가 인터넷 브라우저를 통해 사이트에 접속하고 분석 조건을 입력한 후 실행하면, ASP에서는 XMLMD의 인스턴스를 만들어 요청한 결과를 수행하기 위한 메소드를 호출하게 된다. XMLMD는 XML 소스를 자동으로 생성하여 XML/A Provider에게 보내고, XML/A는 OLAP 서버와 통신하여 그 결과를 받아 XML 형태로 XMLMD에 보낸다. XMLMD는 XML 또는 배열 형태로 데이터를 변환하고, 이를 ASP에서 받아 사용자에게 결과 화면을 보여준다.



(그림 13) WOCB의 시스템 구조

6.2 WOCB의 실행과 구현

WOCB는 (그림 14)와 같이 1)큐브 선택, 2)축 선택, 3)슬라이스 선택, 4)결과의 4영역으로 나뉜다. 사용자는 1)에서 분석하고자 하는 데이터베이스 카탈로그와 큐브를 선택하고, 2)에서 데이터의 행, 열 축을 선택한 후, 3)에서 슬라이스, 드릴다운, 롤업 연산을 위한 차원 멤버들을 선택한다. 모든 조건을 선택한 후에, 결과 데이터의 디스플레이 유형을 선택하고 실행 버튼을 누르면 4)영역에 분석 데이터가 디스플레이 된다.



(그림 14) 웹 OLAP 큐브 브라우저의 전체 화면

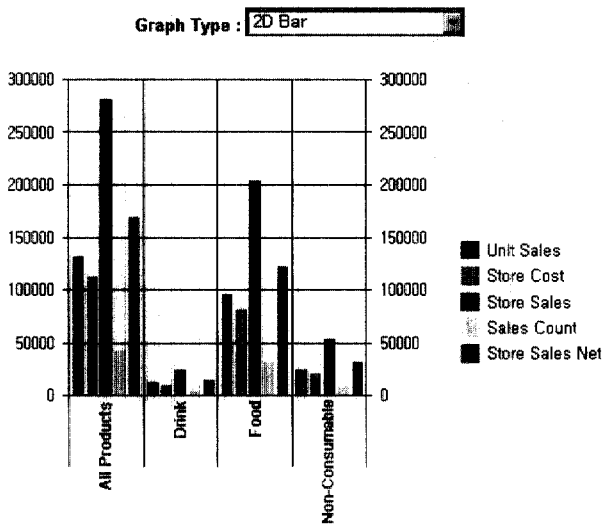
(그림 14)의 예제는 'FoodMart 2000' 카탈로그의 'Sales' 큐브를 분석하는 경우이다. 상품(Product)을 행, 측정값(Measures)을 열 축으로 선택하고, 성별(Gender)이 'F'(여자)인 것을 슬라이스하여 큐브 데이터를 브라우징하였다. 이 예제의 결과를 엑셀 형태로 보여주는 것이 (그림 15), 그래프 형태로 보여주는 것이 (그림 16)이다. WOCB는 그래프의 유형을 2D/3D 형태의 Bar, Line, Area, Step, Combination 등으로 다양하게 제공하고 있다.

A1					
	131,558.00	111,777.48	120,228	42831	168,448.73
	12,202.00	9,751.10	124,457	3953	14,706.27
	94,814.00	80,993.45	1203,094	30848	122,100.72
	24,542.00	21,032.93	152,675	8030	31,641.74

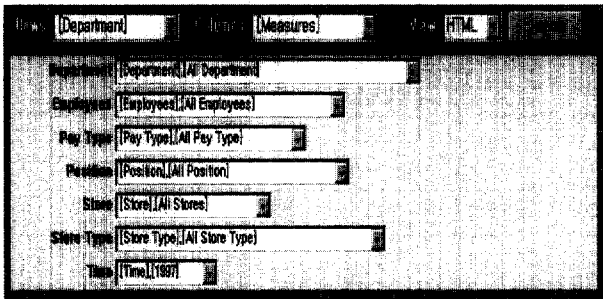
(그림 15) 엑셀 유형의 결과 화면

WOCB는 XML/A의 인터페이스를 간접적으로 사용하는 XMLMD를 사용하여 개발되었다. ASP 소스에서는 큐브 브라우징을 할 수 있는 화면을 그려줄 뿐 아니라 사용자가 선택한 행과 열, 슬라이스 연산 조건 등을 조합하여 MDX(Multidimensional Expressions) 명령문[4]을 자동으로 생성한다.

조건에 맞게 생성된 MDX 문장은 XMLMD의 Execute 또는 GetDataSet 메소드의 매개변수로 넘겨져서, 분석하고자 하는 큐브의 데이터를 얻게 된다. 사용자가 (그림 17)과 같이 선택하였을 경우, 생성되는 MDX 문장은 (그림 18), 이를 이용하여 XMLMD의 메소드를 호출하는 ASP 소스는 (그림 19)와 같다.



(그림 16) 그래프 유형의 결과 화면



(그림 17) 행, 열의 축과 슬라이스 조건 선택 영역

```
SELECT {[Measures].members } ON COLUMNS,
      (DrilldownLevelTop([Department].[All].members, 12) )
      ON ROWS
FROM [HR]
WHERE ([Employees].[All Employees], [Pay Type].[All Pay Type],
       [Monthly], [Position].[All Position], [Store].[All Stores],
       [Store Type].[All Store Type].[Deluxe Supermarket],
       [Time].[1997])
```

(그림 18) 자동 생성된 MDX 명령문

```
Set objCat = Server.CreateObject("XMLMD.MDDDataSet")
xmlResult = objCat.Execute(strMDXCommand, "FoodMart 2000")
```

(그림 19) XMLMD를 이용한 ASP 소스

7. 결론 및 향후 과제

본 논문에서는 웹 서비스 기반의 클라이언트 OLAP API인 "XMLMD"와 함께 웹 서비스를 통해 OLAP 큐브 데이터를 브라우저할 수 있는 웹 OLAP 큐브 브라우저인 "WOCB"를 설계하고 구현하였다. XMLMD는 윈도우즈 환경의 클라이언트 응용 개발자들이 XML/A가 제공하는 기능 중에서 OLAP 관련 서비스를 손쉽게 사용할 수 있도록 캡슐화한 상위 수준의 클라이언트용 OLAP API이다. ADOMD가 OLEDB for OLAP의 상위 수준 API인 것과 같이, XMLMD는 XML/A의 상위 수준 API이다. XMLMD를 이용하여 개발한 WOCB는 웹 상에서 다양한 형태로 다차원 데이터를 분석할 수 있도록 지원한다.

현재 본 연구팀은 WOCB의 성능 향상을 꾀하고, 다양한 기능을 부여하고자 .NET 환경으로 프로그램을 확장 중에 있고, 향후 연구 과제로는 큐브 데이터를 검색하기 위한 MDX 문을 자동으로 생성하는 메소드와 큐브, 차원 등의 OLAP 메타데이터를 생성, 수정, 삭제할 수 있는 메소드를 추가하여 XMLMD를 확장하는 것이다.

참 고 문 헌

- [1] W. W. Eckerson, "Web based Query Tools and Architectures," Journal of Data Warehousing, Vol.2, No.2, April, 1997.
- [2] Java OLAP Interface (JOLAP), <http://jcp.org/jsr/detail/069.jsp>.
- [3] H. Kreger, IBM Software Group, "Web Services Conceptual Architecture (WSCA1.0)," <http://www.ibm.com/software/solutions/webservices/pdf/WSCA.pdf>, May, 2001.
- [4] MDX (Multidimensional Expressions), http://msdn.microsoft.com/library/default.asp?url=/library/en-us/olapdmd/agmdxbasics_04qg.asp.
- [5] J. M. Myerson, "Web Service Architecture," <http://www.webservicesarchitect.com/content/articles/webservicesarchitectures.pdf>, Tect Ltd., 2002.
- [6] OLAP Council White Paper, <http://www.olapcouncil.org/research/whtpapply.htm>, 1997.
- [7] OLE DB for OLAP, http://msdn.microsoft.com/library/default.asp?url=/library/en-us/oledb/html/olapintroduction_to_ole_db_for_olap.asp, Microsoft Co.
- [8] Pilot Software, "An Introduction to OLAP: Multidimensional Terminology & Technology," http://www.pilotsw.com/news/olap_white.htm, White Paper, 1999.
- [9] The OLAP Report, <http://www.olapreport.com>.
- [10] XML for Analysis Specification, <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnxmlspec/html/xmlanalysis.asp>, Microsoft Co., 2001.



배 은 주

e-mail : ejbae@ewha.ac.kr

1995년 국민대학교 산림자원학과(학사)
1995년~1997년 (주)에스프리컨설팅 사원
1997년~1998년 (주)LG-EDS 시스템
사원
1999년~2000년 (주)삼성 SDS 전입

2000년~2001년 (주)웹이지 대리

2001년~현재 이화여자대학교 컴퓨터학과 석사과정

관심분야 : OLAP, 데이터웨어하우징, 웹 서비스 등



김 명

e-mail : mkim@ewha.ac.kr

1981년 이화여자대학교 수학과(학사)
1983년 서울대학교 계산통계학과(이학석사)
1990년 미네소타대학교 컴퓨터학과(공학
석사)

1993년 캘리포니아 주립대학교(산타바바라
소재) 컴퓨터학과(공학박사)

1993년~1994년 캘리포니아 주립대학교(산타바바라 소재)

Postdoc, 강사

1995년~1999년 이화여자대학교 컴퓨터학과 조교수

2000년~현재 이화여자대학교 컴퓨터학과 부교수

관심분야 : 지식공학, OLAP, 인터넷 기술, 고성능 컴퓨팅 등