

# XMLA를 사용한 OLAP과 데이터 마이닝 분석이 가능한 리포팅 툴의 구현

## (Implementation of Reporting Tool Supporting OLAP and Data Mining Analysis Using XMLA)

최 지 응 <sup>†</sup>      김 명 호 <sup>††</sup>  
(Jeewoong Choe)      (Myungho Kim)

**요약** 기업 운영에서 발생하는 데이터의 수집과 통합에서부터 의사결정을 위한 정보의 분석 및 그 결과로의 접근을 제공하기 위한 BI 환경에서 최종 사용자들을 위한 프론트-엔드 툴로서는 데이터베이스 쿼리 및 리포팅 툴, OLAP 툴, 데이터 마이닝 툴이 대표적이다. 데이터베이스 쿼리 및 리포팅 툴은 SQL 쿼리 결과 셋을 반영하는 워드프로세서가 생성하는 문서 수준의 정교한 동적 문서의 생성과 웹 환경을 통한 문서 배포 능력이 장점이지만 데이터 소스가 RDBMS로 제한되어 있다. 반면, OLAP 툴과 데이터 마이닝 툴은 각기 고유한 방식으로 데이터를 분석할 수 있는 능력은 강력하지만 차트와 표 등의 제한적인 컴포넌트들만으로 분석 결과를 제공할 수 있다는 한계를 가지고 있다. 본 논문에서는 상호 보완적으로 사용될 수 있는 BI 환경을 위한 프론트-엔드 툴들을 통합하였다. 본 논문에서 제안하는 리포팅 툴은 RDBMS에서 데이터를 추출하기 위한 SQL 기반의 쿼리 편집기만을 내장한 기존의 리포팅 툴과 달리 OLAP과 데이터 마이닝을 위한 쿼리 편집기를 추가하여 OLAP과 데이터 마이닝 서버로부터도 데이터를 추출할 수 있다. 그리고 기존의 리포팅 툴은 동일한 문서를 다수의 사용자들이 조회하는 상황에서 반복된 문서 생성을 피하기 위하여 서버 측에서 문서를 생성하는 구조를 갖지만 이 시스템은 다수의 사용자들을 위한 문서배포 목적이 아닌 사용자들이 데이터 분석 목적으로 서로 다른 문서를 생성하는 상황에 적합하도록 서버 측에 비해 제한된 리소스 환경을 갖는 클라이언트 측에서 동작하는 리포트 뷰어에서 대량의 데이터를 포함하는 문서를 생성할 수 있는 구조와 처리방식을 갖고 있다. 또한 이 시스템에서 접근하는 세 가지 종류의 데이터 소스에서 추출한 데이터들을 연계하여 하나의 문서에서 통합할 수 있도록 하는 자료 구조를 갖추고 있다. 마지막으로 이 시스템은 특정 벤더의 OLAP과 데이터 마이닝 서버에 종속적으로 동작하지 않기 위하여 웹 서비스 기반의 XMLA를 이들 서버와의 통신 프로토콜로써 선택하였다.

키워드 : 리포팅 툴, OLAP, 데이터 마이닝, XMLA, 데이터 웨어하우스(DW), 비즈니스 인텔리전스(BI)

**Abstract** Database query and reporting tools, OLAP tools and data mining tools are typical front-end tools in Business Intelligence environment which is able to support gathering, consolidating and analyzing data produced from business operation activities and provide access to the result to enterprise's users. Traditional reporting tools have an advantage of creating sophisticated dynamic reports including SQL query result sets, which look like documents produced by word processors, and publishing the reports to the Web environment, but data source for the tools is limited to RDBMS. On the other hand, OLAP tools and data mining tools have an advantage of providing powerful information analysis functions on each own way, but built-in visualization components for analysis results are limited to tables or some charts. Thus, this paper presents a system that integrates three typical front-end tools to complement one another for BI environment. Traditional reporting tools only

· 본 연구는 숭실대학교 교내연구비 지원으로 이루어졌음

† 학생회원 : 숭실대학교 컴퓨터학과  
jwchoe@ss.ssu.ac.kr

†† 종신회원 : 숭실대학교 컴퓨터학과 교수  
kmh@ssu.ac.kr

논문접수 : 2008년 8월 26일

심사완료 : 2008년 12월 16일

Copyright©2009 한국정보과학회: 개인 목적이거나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 컴퓨팅의 실제 및 레터 제15권 제3호(2009.3)

have a query editor for generating SQL statements to bring data from RDBMS. However, the reporting tool presented by this paper can extract data also from OLAP and data mining servers, because editors for OLAP and data mining query requests are added into this tool. Traditional systems produce all documents in the server side. This structure enables reporting tools to avoid repetitive process to generate documents, when many clients intend to access the same dynamic document. But, because this system targets that a few users generate documents for data analysis, this tool generates documents at the client side. Therefore, the tool has a processing mechanism to deal with a number of data despite the limited memory capacity of the report viewer in the client side. Also, this reporting tool has data structure for integrating data from three kinds of data sources into one document. Finally, most of traditional front-end tools for BI are dependent on data source architecture from specific vendor. To overcome the problem, this system uses XMLA that is a protocol based on web service to access to data sources for OLAP and data mining services from various vendors.

**Key words** : Reporting Tools, OLAP, Data Mining, XMLA, Data Warehouse, Business Intelligence

## 1. 서론

오늘날 기업이 IT 기술을 활용하여 비즈니스 프로세스의 자동화를 위한 목적으로 OLTP(Online Transaction Processing) 시스템을 구축하고 그로부터 축적한 데이터를 바탕으로 보다 나은 의사 결정을 위한 정보를 획득하기 위하여 데이터 웨어하우스를 구축하는 모습은 이제 일반적이다. OLTP 시스템은 일반적으로 RDBMS(Relational Database Management System)를 데이터 저장소로써 사용하며 클라이언트 툴들은 SQL을 사용하여 쿼리를 요청하고 결과 셋을 응답받는 방식으로 통신한다. 데이터 웨어하우스는 OLAP(Online Analytical Processing)과 데이터 마이닝과 같은 정보 분석 및 추출 목적의 응용 툴들이 활용하기에 적합하도록 데이터들을 정제하여 구축한 저장소이다. 이와 같이 기업 운영에서 발생하는 데이터의 수집과 통합에서부터 의사결정을 위한 정보의 분석 및 지식의 추출 그리고 그 결과로의 용이한 접근제공 등의 일련의 과정을 지원하기 위한 IT 환경을 BI(Business Intelligence)라 하며 BI 환경에서 최종 사용자들을 위한 프론트-엔드 툴로서는 데이터베이스 쿼리 및 리포팅 툴, OLAP 툴, 데이터 마이닝 툴이 대표적이다[1].

데이터베이스 쿼리 및 리포팅 툴[2]은 일반적으로 관계형 데이터베이스를 데이터 소스로 하여 이들의 데이터를 반영하는 동적 문서를 생성할 수 있는 툴이다. 리포팅 툴이 동적 문서를 생성하는 방식은 웹 서버 측에서 동작하는 스크립트 기반의 웹 애플리케이션이 데이터베이스와 연동하여 동적 웹 페이지를 생성하는 일반적인 방식과 달리 워드프로세서와 유사한 GUI 환경의 편집기를 통해 디자인한 템플릿 문서를 기반으로 동적 문서를 생성한다. 이러한 방식은 템플릿 문서의 생성시 특별한 전산 지식을 요구하지 않으므로 전산 인력의 도

움 없이 웹 환경에서 신속한 동적 문서의 생성 및 배포를 가능하게 한다. 웹 문서의 표준 형식인 HTML은 스크린 상에 정보를 디스플레이하기 위하여 고안된 언어이므로 HTML 형식의 문서를 인쇄하고자 할 때 인쇄 용지 크기에 맞추어 정보를 포맷팅 하는데 어려움이 따른다. 그러나 리포팅 툴이 생성하는 문서는 워드프로세서를 통해 생성한 문서처럼 특정 인쇄 용지 크기 내에서 정교하게 포맷팅한 문서를 웹 환경으로 출력할 수 있다.

OLAP 툴은 데이터 웨어하우스 또는 데이터 마트에 접근하여 분석 주제에 적합한 데이터 큐브[3]를 선택하여 roll up, drill down, slice and dice, pivot 연산을 사용하여 큐브를 탐색하며 대화식으로 데이터를 분석할 수 있도록 한다. OLAP 툴의 데이터 분석을 위한 사용자 인터페이스는 피벗 테이블이다. 데이터 분석 결과는 테이블과 차트 형태로 제공할 수 있다.

데이터 마이닝 툴은 대량의 저장된 데이터로부터 기계 학습(Machine Learning) 알고리즘을 사용하여 정보, 지식, 규칙, 패턴, 특성 등을 추출할 수 있는 툴이다. 추출한 정보는 해석과 취합 과정을 거친 후 가시화 컴포넌트를 사용해 사용자에게 제공된다. 데이터 마이닝 기술의 BI를 위한 응용과 관련한 최근의 경향은 CRM(Customer Relationship Management), SCM(Supply Chain Management) 등의 기업 경영을 위한 개념이 도입되면서 기업 환경에서 발생하는 데이터를 기반으로 의사 결정을 지원하기 위한 의미 있는 정보 추출을 목적으로 그 활용이 증가하고 있다.

과거에는 RDBMS와 OLAP 그리고 데이터 마이닝을 위한 연산 서버들이 독립된 제품과 시장으로 분리되어 있었으나 최근에는 마이크로소프트, 오라클, IBM 등의 대형 데이터베이스 벤더들은 이들을 통합하기 위한 하나의 플랫폼을 형성한 후 각각이 플랫폼 위에서 동작하

는 서비스의 개념으로 추상화 하여 제공하고 있다. 이러한 통합된 서비스를 이용하기 위한 클라이언트 툴들은 타겟 플랫폼에서 제공하는 API를 사용하여 해당 플랫폼 내에서 일관성 있는 접근 방식과 각 서비스로부터 제공 받은 결과를 용이하게 통합할 수 있게 된다. 그러나 이러한 클라이언트 툴들은 벤더에 따른 플랫폼 사이의 표준이 없는 관계로 특정 벤더의 플랫폼에 종속적인 툴이 된다. 이러한 플랫폼별 폐쇄성을 극복하기 위하여 플랫폼 독립적인 통신 프로토콜과 메시지 포맷 그리고 쿼리 언어가 필요하다. XMLA(XML for Analysis)[4,5]는 이러한 목적을 위한 OLAP과 데이터 마이닝 툴을 위한 메시지 인터페이스이다. XMLA는 웹 표준 프로토콜인 HTTP를 사용하여 통신하며 메시지는 XML 포맷인 SOAP을 사용한다. 또한 쿼리 언어는 다양한 벤더에서 지원하고 있는 OLAP을 위한 MDX와 데이터 마이닝을 위한 DMX를 지원한다.

리포팅 툴은 데이터베이스 쿼리 편집기와 리포트 생성기(Report Writer)가 결합된 형태이다. 이러한 결합이 가져오는 장점은 데이터 분석 업무와 분석 결과에 대한 보고서 작성 및 배포 업무를 결합함으로써 업무 생산성을 향상시킬 수 있다는 점이다. 기존의 방식은 데이터베이스 쿼리 편집기를 사용하여 데이터베이스에 접근하여 데이터 분석을 수행한 후 그 결과인 테이블 결과 셋들을 워드프로세서 혹은 스프레드시트 프로그램으로 가져와서 가공한 후 보고서의 일부로 삽입하여 사용하는 과정을 수행해야 했다. 두 번째 장점은 동일 성격의 주기적으로 수행해야 하는 업무를 최소화 할 수 있다는 점이다. 이는 리포트 생성기의 동적 문서를 생성하는 방법에 기인한다. 리포트 생성기의 입력은 템플릿 파일과 사용자 파라미터이다. 템플릿 파일의 내용은 동적 문서를 생성하기 위한 룰(rule)에 해당한다. 사용자 파라미터는 동적 문서에서 반영할 데이터베이스 쿼리 결과 셋들의 범위를 결정한다. 따라서 주, 월, 분기 등의 주기로 동일 성격의 보고서를 작성할 때 사용자 파라미터의 값만을 달리할 경우 동일한 룰에 따라 보고서를 생성 및 배포할 수 있다.

OLAP 툴과 데이터 마이닝 툴 또한 리포팅 툴과 결합할 경우 분석 결과를 기존 툴의 내장된 표 혹은 차트 형식의 제한된 표현 컴포넌트로만 출력하던 한계를 극복하여 완전한 보고서 형태로 출력할 수 있다. 또한 리포팅 툴의 사용자 파라미터 특성을 활용하여 분석 데이터의 범위를 조정함으로써 동일 주제의 정보 분석과 추출 과정을 자동화할 수 있다.

그러나 BI 환경에서의 프론트-엔드 툴들의 연계 혹은 통합에 의한 이러한 장점들을 수렴하는 클라이언트 툴에 대한 기존의 연구는 대형 데이터베이스 벤더들의 서

로간의 호환성이 취약한 플랫폼에 종속적으로 이루어지고 있다. 또한 동일한 플랫폼을 지원하는 클라이언트 툴 또한 각각의 개별적인 목적의 툴들이 느슨한 결합 형태로 제공되고 있기 때문에 서로 다른 데이터 소스들(RDBMS, OLAP, 데이터 마이닝)로 부터 가져온 결과 데이터들을 클라이언트 측에서 독립적으로 통합하여 가공할 수 없다는 한계를 가지고 있다.

따라서 본 논문에서는 데이터베이스 쿼리 및 리포팅 툴의 장점인 템플릿 문서에 기반한 자동 문서 생성 기능과 OLAP 툴과 데이터 마이닝 툴의 정보 분석 기능을 상호 보완적으로 결합하여 다양한 데이터 소스를 기반으로 하는 데이터 분석을 하나의 통합된 환경에서 수행할 수 있으며 그 결과를 하나의 문서로 생성하고 웹 환경으로 배포할 수 있는 시스템을 제시한다. 또한 XMLA를 사용하여 OLAP과 데이터 마이닝 분석의 경우 특정 벤더의 데이터 저장소에 종속적으로 동작하지 않는 클라이언트의 구조도 제시한다.

본 논문의 구성은 다음과 같다. 제2장에서는 본 논문에서 구현한 시스템의 통합 대상인 리포팅 툴, OLAP, 데이터 마이닝, XMLA에 대해서 기술한다. 제3장에서는 구현 시스템을 위한 요구 사항을 도출한다. 제4장에서는 시스템의 구성 요소별 설계 사항을 기술한다. 제5장에서는 구현 내용을 기술하고 마지막으로 제6장에서는 결론 및 향후 연구에 대하여 기술한다.

## 2. 관련 연구

이 장에서는 본 논문에서 제안하는 시스템의 통합 대상인 리포팅 툴, OLAP, 데이터 마이닝 그리고 통신 프로토콜인 XMLA에 대해서 설명한다.

### 2.1 리포팅 툴

리포팅 툴은 그림 1과 같이 포맷 편집기(Format Editor), 리포트 생성기(Report Writer), 리포트 뷰어(Report Viewer)로 구성되어 있다[2]. 포맷 편집기는 데이터베이스 쿼리 편집기와 문서 편집기가 결합된 형태이다. 포맷 편집기는 결과 데이터 셋을 필드 단위로 문서의 필요한 곳에 맵핑할 수 있는 기능을 갖추고 있다. 리포트 생성기는 포맷 파일의 최종 문서 생성을 위한 규칙을 해석하여 데이터베이스로부터 데이터를 수집한 후 배치하여 동적 문서를 생성한다. 리포트 뷰어는 동적 문서가 HTML 형식일 경우 웹 브라우저가 그 역할을 담당하며 그 외의 형식은 별도의 뷰어가 ActiveX, Applet, Flash 형식으로 제작되어 브라우저 내에 삽입되어 동작하는 형식을 취한다.

리포팅 툴의 중요성과 그 활용이 증가하면서 최근에는 대형 DBMS 벤더들이 그들의 기존의 DBMS 제품을 하나의 플랫폼으로 확장하여 리포팅 툴을 그 위에서

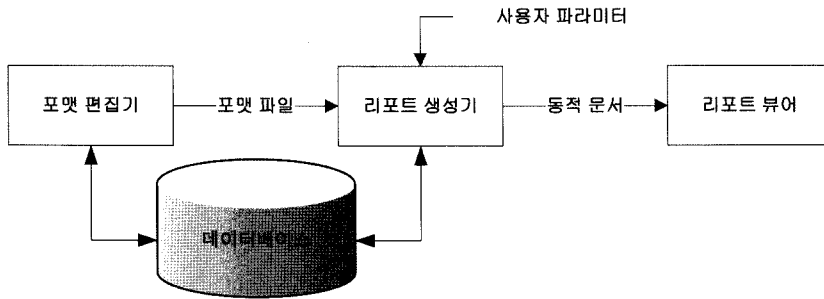


그림 1 리포팅 툴의 구조와 처리과정

동작하는 하나의 서비스 형태로 제공하고 있다. Microsoft는 자사의 DBMS인 Microsoft SQL Server 2005를 기반으로 [6]을 리포팅 툴로써 제공하고 있다. Oracle의 경우는 Oracle 10g의 제품군으로써 [7]이 리포팅 툴에 해당한다. [6,7]과 같은 리포팅 툴들은 특정 벤더의 DBMS에 종속적인 한계를 가지고 있지만 동일 플랫폼 위에서 동작하는 다른 서비스들과의 연계가 뛰어나다. 대표적인 리포팅 툴인 [8]의 경우 제조업체인 Business Objects가 대형 ERP 벤더인 SAP에 합병되면서 SAP의 ERP 관련 제품들과의 API 레벨에서의 연계성이 향상되었으며 특히 Microsoft의 DBMS와의 호환성이 우수하다. 제품의 동작 플랫폼 또한 Microsoft의 Windows 계열로 제한되어 있다.

국내의 리포팅 툴을 도입하고자 하는 기업들은 기존 오프라인 환경에서 사용하던 워드프로세서로 생성한 문서와 동일한 품질을 갖춘 동적 문서의 웹 브라우저를 통한 접근과 인쇄를 요구한다. 이러한 문서의 특징은 영·미권에서는 빈번히 사용하지 않는 특정 페이지 사이즈 내에서 복잡한 표 서식으로 이루어진 문서를 요구한다는 것이다. [6-8]과 같은 외산 리포팅 툴은 HTML 문서와 같이 페이지 사이즈를 고려하지 않은 스크린 상에서 웹 브라우저를 통한 정보의 출력력이 주요 목적이다. 이럴 경우 웹 브라우저의 인쇄 기능을 사용하여 인쇄 시 인쇄 품질이 만족스럽지 못하게 된다. 이러한 문화적 차이로 인하여 외산 리포팅 툴들은 국내 환경에 적용 시 어려움이 있다.

## 2.2 OLAP

OLAP은 데이터 웨어하우스의 다차원 모델에 접근하여 대화식으로 정보를 분석하는 과정 및 제반 시스템을 일컫는다. 데이터 웨어하우스는 변경과 삭제를 고려하지 않는 읽기 작업만 지원하는 정보 분석용 데이터베이스이다. 다차원 모델을 물리적으로 저장하는 방식에 따라서 ROLAP과 MOLAP으로 분류한다[1]. OLAP을 위한 다차원 모델인 데이터 큐브의 논리적 구조는 차원(dimension)으로 이루어져 있으며 각 차원은 계층 구조

표 1 OLAP의 기본 연산

rollup	상위 레벨로의 이동
drill-down	하위 레벨로의 이동
slice_and_dice	차원의 프로잭션(projection)
pivot	뷰(view)의 차원 회전

를 이루며 계층 구조 상의 각각의 집계 깊이(aggrea-tion depth)를 레벨이라고 한다. 본 논문의 관심사항은 사용자의 OLAP 연산을 지원하는 OLAP 클라이언트 툴이다. 일반적으로 OLAP 툴은 데이터 큐브를 탐색하기 위하여 표 1의 연산을 지원해야 한다.

과거에는 MOLAP 위주의 OLAP 시장은 관계형 DBMS 시장과 분리되어 있었으나 최근 들어 [9,10]처럼 대형 RDBMS 벤더들은 자신들의 데이터베이스 플랫폼에 ROLAP과 MOLAP이 혼합된 형태의 OLAP 서비스를 통합하여 제공하고 있다.

## 2.3 데이터 마이닝

데이터 마이닝은 기업이 축적하는 데이터 양이 폭발적으로 증가하면서 OLAP과 같이 인지적으로 정보를 분석 및 획득하는 방식에 대한 한계의 인식과 기계 학습(Machine Learning) 기술의 발전으로 인해 최근 들어 기업이 구축하는 CRM, SCM 등의 BI 시스템에서 활용이 증가하고 있다. 시장의 이러한 요구를 수용하기 위하여 대형 데이터베이스 벤더들은 [11,12]와 같이 그들의 데이터베이스 플랫폼에 데이터 마이닝 서비스를 추가하여 제공하기 시작했다. 표 2는 [11,12]에서 사용할 수 있는 데이터 마이닝 알고리즘 리스트이다.

[11,12]를 위한 알고리즘들은 하나의 관계형 테이블 혹은 뷰를 입력으로 한다. 데이터 마이닝 서비스에 대한 클라이언트 측에서의 연산 제어를 위하여 [12]은 DMX(Data Mining Extensions)라는 SQL과 유사한 별도의 쿼리 언어를 제공하며 [11]는 PL/SQL의 'DBMS\_DATA\_MINING' 패키지를 사용하면 된다.

## 2.4 XMLA

XMLA는 OLAP과 데이터 마이닝 클라이언트 툴을

표 2 데이터 마이닝 알고리즘

Oracle Data Mining	Microsoft Data Mining
Naïve Bayes	Decision Trees
Decision Trees	Association Rules
Adaptive Bayes Networks	Naïve Bayes
Support Vector Machines	Sequence Clustering
Enhanced k-Means Clustering	Time Series
Orthogonal Partitioning Clustering	Neural Nets
Anomaly Detection	Text Mining
Association Rules	
Nonnegative Matrix Factorization	
Minimum Description Length	
Basic Local Alignment Search Technique	

위한 HTTP, SOAP, XML 기술을 사용하는 웹 서비스 기반의 API이다. XMLA는 'Discover'와 'Execute' 두 가지 메소드를 제공한다. 'Discover'는 쿼리문 작성을 위해 필요한 데이터 소스 측의 스키마 정보와 같은 메타데이터를 요구하기 위한 메소드이며 'Execute' 메소드는 서버 측에 분석 결과 데이터를 요구하는 쿼리문을 매개변수로 사용한다. 클라이언트와 서버 사이의 메시지들은 모두 XML 메시지가 SOAP 형식으로 랩핑(wrapping)된 후 HTTP 프로토콜로 전송된다. XMLA 명세는 OLAP과 데이터 마이닝을 위한 쿼리 언어에 대한 표준의 부재로 인하여 원칙적으로는 벤더에 따른 쿼리 언어를 사용하도록 정의하고 있다. 그러나 현실적으로는 XMLA를 지원하는 벤더들은 쿼리 언어도 통일하여 사용하고 있다. XMLA는 OLAP을 위한 쿼리 언어로써 MDX(Multi-Dimensional Expression)를 사용하며 데이터 마이닝을 위한 쿼리 언어로써 DMX를 사용한다. XMLA를 지원하는 벤더는 현재 Hyperion, Microsoft, SAP, SAS가 있다.

### 3. 요구 사항 분석

이 장에서는 OLAP과 데이터 마이닝 분석과 그 결과를 반영한 동적 문서의 생성과 배포가 가능한 리포팅 툴의 구현을 위한 시스템 구성 요소별 요구 사항을 분석하여 설계 단계에서 고려해야 할 사항을 도출한다.

#### 3.1 데이터 소스와의 통신을 위한 메시지 인터페이스

이 시스템은 최종 보고서에 반영할 정보를 획득하기 위하여 리포팅 툴에서 관계형 데이터베이스, OLAP 서버, 데이터 마이닝 서버에 접근할 수 있는 메시지 인터페이스와 데이터 저장소에서 수행할 연산을 제어할 수 있는 쿼리 언어를 생성할 수 있어야 한다. 표 3은 이 시스템을 위한 각각의 데이터 저장소 타입에 따른 메시지 인터페이스와 쿼리 언어이다.

관계형 데이터베이스의 경우 SQL이라는 표준 쿼리

표 3 데이터 소스별 쿼리 언어와 메시지 인터페이스

	관계형 데이터베이스	OLAP	데이터 마이닝
쿼리 언어	SQL	MDX	DMX
메시지 인터페이스	JDBC	XMLA	XMLA

언어가 존재하며 대부분의 RDBMS는 JDBC 드라이버를 제공하고 있다. OLAP과 데이터 마이닝의 경우 XMLA를 사용함으로써 다양한 벤더의 OLAP 서버와 데이터 마이닝 서버에 HTTP 프로토콜로 접근할 수 있으며 쿼리 언어인 MDX와 DMX는 현재까지 표준은 아니지만 대부분의 대형 벤더에서 지원하는 쿼리 언어 사양이다.

#### 3.2 쿼리 편집기

이 시스템은 세 가지 종류의 데이터 분석을 지원하기 위해서 사용자가 쿼리문을 작성하고 결과를 조회할 수 있는 GUI 기반의 쿼리 편집기를 필요로 한다. 특히 OLAP의 경우 OLAP 서버와 인터랙티브한 방식으로 데이터 분석이 이루어지므로 분석 UI인 피벗 테이블 상에서 사용자의 커맨드에 해당하는 쿼리문을 자동 생성할 수 있어야 한다.

#### 3.3 쿼리 결과를 반영할 수 있는 리포팅 툴의 표현 컴포넌트

이 시스템은 동적 문서를 생성하기 위하여 일반적인 워드프로세서와 같이 텍스트, 도형, 이미지, 테이블, 리스트, 차트 등의 표현 컴포넌트 이외에 쿼리 결과를 반영할 수 있는 표현 컴포넌트를 요구한다. 데이터 마이닝 분석 결과는 관계형 데이터베이스에 대한 쿼리 결과 셋과 동일한 형태이다. 즉, 레코드들의 집합 형태이다. OLAP 연산 결과는 최종 피벗 테이블이다. 이들 분석 결과를 문서 내에 반영할 수 있는 표현 컴포넌트가 필요하다. 관계형 데이터베이스와 데이터 마이닝 결과는 레코드의 필드 단위로 선택하여 문서에 반영할 수 있어야 하며 OLAP의 경우 쿼리 편집기 내의 피벗 테이블과 동일한 형태의 피벗 테이블을 요구한다.

#### 3.4 다양한 분석 양태에 따른 동적 문서 생성을 위한 구조와 처리 방식

OLAP과 데이터 마이닝의 쿼리 결과는 대량의 데이터를 생성하지는 않는다. 그러나 관계형 데이터베이스에 대한 쿼리 결과는 레코드의 개수에 비례하여 이를 반영하는 동적문서 또한 페이지 개수가 증가한다는 특징이 있다. 따라서 문서 생성을 위하여 많은 양의 리소스와 처리 시간을 요구할 수 있다. 반면 OLAP의 경우는 동적 문서에서 OLAP 분석 컨텍스트를 불러와서 분석을 수행할 경우 OLAP 연산에 해당하는 사용자 커맨드마다 동적 문서의 잦은 생성을 요구할 수 있다. 따라서 다양한 방식의 분석 결과가 혼합된 동적 문서의 생성에

대한 요구 사항을 반영할 수 있는 구조와 처리 방식이 필요하다. 이를 위해 이 시스템은 문서 생성에 필요한 데이터를 가용 메모리 크기를 고려하여 효과적으로 관리할 수 있는 구조와 대량의 데이터로 인해서 긴 생성 시간을 가지면서 많은 페이지를 생성해야 하는 경우에도 문서 생성과 사용자 페이지 조화가 병행될 수 있는 문서 생성 방식을 취한다.

#### 4. 시스템 설계

##### 4.1 쿼리 컨텍스트의 설계

포맷 편집기에서 생성한 쿼리 컨텍스트는 포맷 파일에 저장되며 리포트 생성기에서 동적 문서를 생성하기 위하여 사용된다. 관계형 데이터베이스, OLAP, 데이터 마이닝에 대한 각각의 쿼리 컨텍스트를 설계하였다. 쿼리 컨텍스트의 내용은 연결 정보, 쿼리문, 쿼리 결과 데이터에 대한 메타데이터, 쿼리 결과 데이터로 구성되어 있다. 'DMXQuery'는 데이터 마이닝, 'MDXQuery'는 OLAP, 'SQLQuery'는 관계형 데이터베이스를 위한 쿼리 컨텍스트에 해당하는 클래스이다. 이 클래스들은 사용자가 포맷 편집기에 내장된 쿼리 편집기의 사용시 쿼리 편집기의 UI에 대한 모델로서 최초 생성된다. 'DMXQuery'와 'MDXQuery'는 쿼리문 요청을 위하여 'XMLConnector'를 사용하여 쿼리 메시지를 SOAP 메시지로 생성하여 전송하며 'SQLQuery'의 연결정보와 쿼리문은 'ServletConnector'를 사용하여 HTTP의 파라미터로 삽입되어 전송된다. 또한 이들 컨텍스트의 쿼리 결과는 동적 문서 생성시 쿼리 결과를 반영하는 표현컴포넌트에서 접근하여 사용한다.

##### 4.2 OLAP 쿼리 편집기의 설계

OLAP 쿼리 편집기는 대화식 정보 분석을 특징으로 하므로 사용자가 피벗 테이블 상에서 OLAP의 기본 연산 명령을 요청할 수 있는 사용자 인터페이스가 필수적으로 요구된다. 그림 3은 OLAP 쿼리 편집기를 위한 전체적인 클래스 다이어그램이다.

- OLAPClient : OLAP 쿼리 편집기의 전체적인 기능을 컨트롤하는 클래스이며 OLAP 기본 연산에 관한 인터페이스인 OLAPOperations를 구현하고 있다.
- ServerSideTree : 사용자에게 접속한 OLAP 서버의 큐브 리스트와 각 큐브의 하위 구조를 제공하며 'ServerSideMetadata' 인터페이스를 구현한다.
- OLAPResultTable : OLAP 연산 결과를 피벗 테이블 형태로 표현할 수 있는 클래스이며 사용자의 rollup, drill-down, pivot 연산 명령을 입력 받는 역할을 수행한다. 또한 'CubeMetadata' 인터페이스를 구현함으로써 현재 선택된 큐브의 커서 정보를 담고 있다 'OLAPClient'에서 서버측에 요구하는 모든 쿼리 요청은 'MDXQuery' 객체를 통해서 이루어진다.

그림 4는 사용자의 OLAP 연산 명령 입력과 그에 대한 결과값을 표현해 줄 수 있는 피벗 테이블의 상세 클래스 다이어그램이다. 피벗 테이블은 행 헤더 영역, 열 헤더 영역, 데이터 셀 영역의 세 영역으로 구성되어 있으며 각 영역은 'Sheet' 인터페이스를 구현함으로써 여러 셀을 컨트롤 할 수 있다. 각 셀은 'Cell' 인터페이스를 구현하고 있으며 해당 셀의 디스플레이 값과 테이블 내에서의 역할 그리고 현재 반영하고 있는 큐브와 연결된 속성을 내장하고 있다. 이러한 셀의 속성들은 사용자가 해당 셀을 기반으로 어떤 종류의 OLAP 연산 명령을 내릴 수 있는가에 대한 판별 기준으로 사용된다.

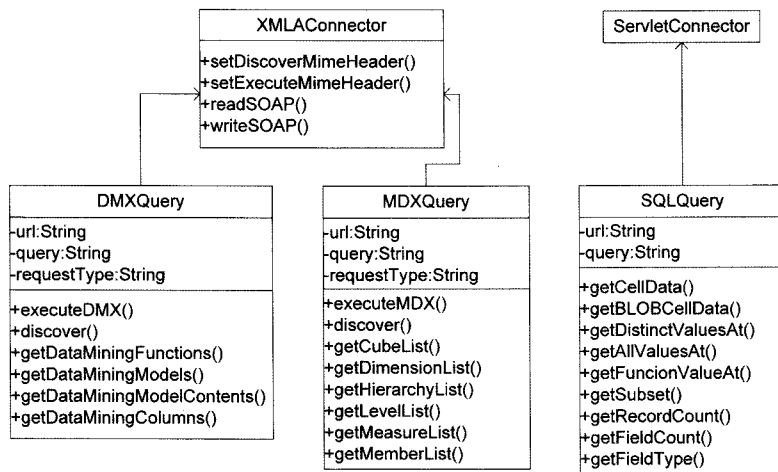


그림 2 쿼리 컨텍스트와 커넥터 클래스 다이어그램

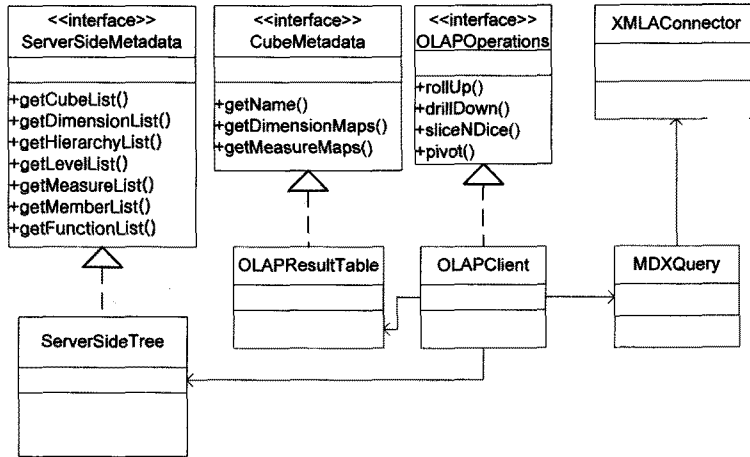


그림 3 OLAP 쿼리 편집기의 클래스 다이어그램

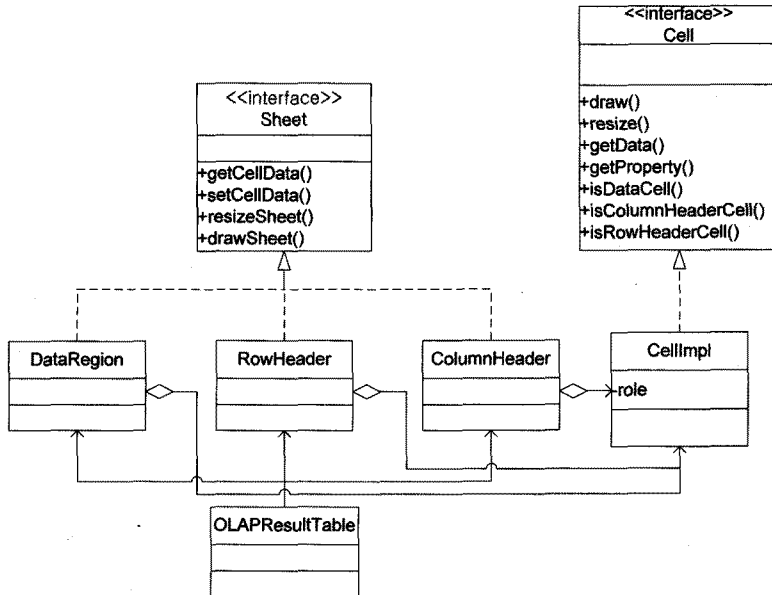


그림 4 피벗 테이블 클래스 다이어그램

### 4.3 정보 분석 결과를 반영하는 리포팅 틀의 표현 컴포넌트의 설계

동적 문서의 페이지 위에 구체적 형상을 가지고 디스플레이 되는 표현 컴포넌트는 정적 컴포넌트와 동적 컴포넌트로 구분할 수 있다[13]. 정적 컴포넌트는 포맷 편집기에서 디자인한 위치와 크기 그리고 텍스트가 동적 문서에 그대로 반영되는 컴포넌트이다. 동적 컴포넌트는 별도의 데이터 소스로부터 추출한 데이터에 영향을 받는 컴포넌트이다. 선, 원, 사각형 등의 기본 도형에 속하는 컴포넌트들은 정적 컴포넌트로서만 동작한다. 표, 텍스트 상자, 이미지 컴포넌트는 기본적으로 정적 컴포넌트

트로서 동작한다. 그러나 쿼리 결과 데이터가 이들 컴포넌트에 할당될 경우 동적 컴포넌트로서 동작한다. 이때에도 위치와 크기는 변화가 일어나지 않고 단순히 쿼리 결과 텍스트로 치환되는 방식으로 동작한다. 리스트 컴포넌트는 할당된 쿼리 결과 데이터의 레코드 개수에 비례하여 페이지 내에서의 영역이 확장된다. 피벗 테이블 컴포넌트는 OLAP뿐 아니라 관계형 데이터베이스를 대상으로 한 쿼리 결과로도 표현할 수 있으므로 각각을 위하여 내부적으로 두 가지 구현이 존재한다. OLAP을 대상으로 하는 피벗 테이블은 동적 문서 상에서 분석 컨텍스트를 불러올 수 있다. 데이터 마이닝 결과는 표

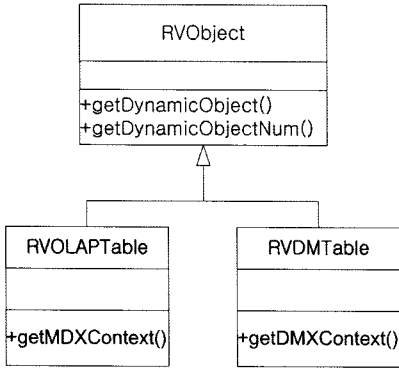


그림 5 OLAP과 데이터 마이닝 분석 결과를 반영하는 표현 컴포넌트

컴포넌트를 사용하여 동적 문서 상에서 표현하며 일반적인 표 컴포넌트와는 다른 별도의 구현이 존재한다. 이 경우도 동적 문서 상에서 데이터 마이닝 분석 컨텍스트를 재생할 수 있다.

그림 5의 'RVOLAPTable' 클래스는 OLAP 분석 결과를 표현하는 피벗 테이블의 클래스 다이어그램이며 'RVDTable'은 데이터 마이닝 분석 결과를 반영하는 표 컴포넌트의 클래스 다이어그램이다. 두 클래스 모두 페이지 컴포넌트가 표현 컴포넌트로서 인식하는 'RVOBJECT'를 상속하고 있다. 또한 각각의 종류에 알맞은 쿼리문 컨텍스트를 불러올 수 있는 메소드를 내장하고 있다.

4.4 리포트 뷰어에서 동적 문서 생성을 위한 구조와 처리 방식

이 절에서는 리포트 뷰어가 동적 문서를 생성하기 위한 구조와 절차를 기술한다[13]. 그림 6은 리포트 뷰어

의 구조를 설명하기 위한 클래스 다이어그램이다. 각각의 주요 클래스는 다음과 같은 역할을 수행한다.

- ReportViewer : 리포트 뷰어의 메인 클래스
- RVXML : 포맷 파일을 로딩하기 위한 클래스
- RVPPageGenerator : 동적 페이지 생성을 처리하는 쓰레드 클래스
- RVPPageCache : 동적 페이지 컨테이너 클래스
- RVPPage : 동적 페이지 클래스
- RVOBJECT : 표현 컴포넌트의 부모 클래스
- RVParameter : 쿼리문에 삽입될 사용자 파라미터 클래스

리포트 뷰어의 동적 문서의 각 페이지들을 생성하는 처리 과정은 다음과 같다. 첫 번째 단계는 포맷 파일로부터 포맷 에디터에서 디자인한 페이지와 같은 상태의 원본 페이지를 복원한다. 두 번째 단계는 첫 번째 단계에서 복원한 원본 페이지들이 각각 몇 페이지의 동적 페이지를 생성하게 되는지를 계산한다. 동적 페이지 개수는 각각의 원본 페이지가 자신이 포함하는 표현 컴포넌트들에게 자신들이 몇 개의 페이지를 필요로 하는지를 계산하게 함으로써 그 결과값들 중 최대값으로 얻게 된다. 이러한 절차를 수행하는 과정에서 각 표현 컴포넌트는 각 페이지 당 필요로 하는 동적 데이터의 범위에 대한 인덱스를 저장해 둬으로써 임의의 동적 페이지 생성 요청 시 빠르게 그 범위의 데이터만을 수집할 수 있다. 또한 이 단계는 별도의 쓰레드를 발생시켜 처리하는데 그 이유는 쿼리 결과 셋의 레코드 수와 표현 컴포넌트의 종류에 따라 처리 시간이 가변적이므로 경우에 따라 장시간의 대기 시간이 발생할 수 있다. 이러한 경우를 방지하고 이 단계가 완료되기 이전에도 현재까지 카운트된 동적 페이지 범위 안에서 사용자가 임의의 동적

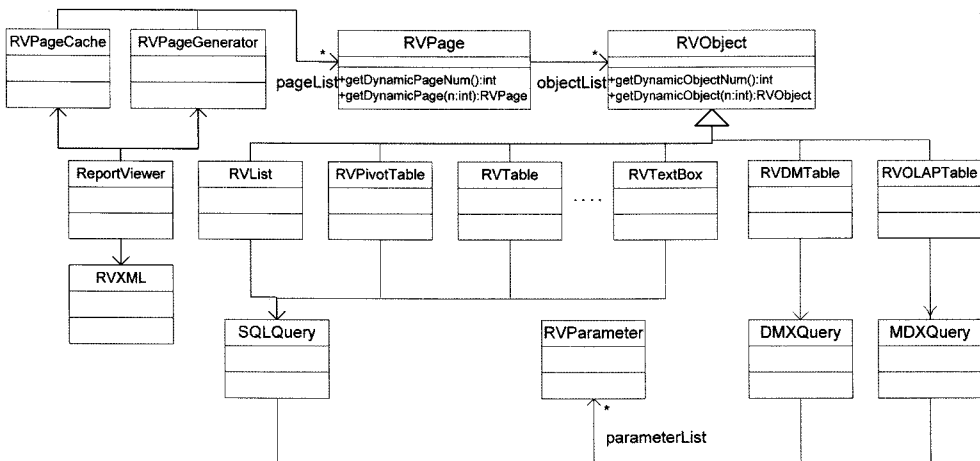


그림 6 리포트 뷰어의 클래스 다이어그램



페이지 생성을 요청하고 조회할 수 있도록 하기 위함이다. 마지막 단계는 사용자가 요청한 동적 페이지를 생성하는 단계이다. 동적 페이지 생성에 대한 이벤트가 발생하면 RVPageCache 객체에게 해당 페이지의 보관 여부를 확인하고 캐쉬 내에 존재하지 않을 경우에만 생성한다. 새로 생성한 페이지는 다시 RVPageCache 객체에 저장된다. 즉, 항상 일정한 개수의 페이지들만이 리포트 뷰어에 저장된다. 이러한 방식은 리포트 뷰어 측의 리소스가 부족할 경우를 대비하기 위함이다.

**5. 시스템 구현**

본 시스템은 모든 구성 요소의 구현을 위한 프로그래밍 언어로써 Java를 사용하여 개발되었다. 포맷 편집기는 독립적인 애플리케이션 형태로 동작한다. 포맷 편집기는 SQL, MDX, DMX 를 사용한 데이터 분석을 수행할 수 있는 쿼리 편집기들을 내장하고 있다. 포맷 편집기를 사용해 디자인한 포맷 파일은 XML 형태로 저장한다. 리포트 뷰어는 Applet 형태로 제작하였으므로 HTML 페이지 내에 삽입되어 실행된다. 또한 리포트 뷰어는 리포트 생성기 역할도 병행한다. 마지막으로 리포트 서버는 HTTP 프로토콜을 사용하여 접근할 수 있으며 XMLServlet과 QueryServlet 두 개의 웹 애플리케이션으로 구성되어 있다. XMLServlet은 리포트 뷰어가 포맷 파일을 다운로드하기 위해 사용하며 QueryServlet은 포맷 편집기와 리포트 뷰어에서 관계형 데이터베이스에 쿼리문을 전달하고 결과 셋을 응답받기 위해서 사용한다. QueryServlet은 포맷 편집기와 리포트 뷰어로부터 전달받은 쿼리문을 EJB의 Session Bean 형태로 제작한 QueryEJB에 전달한다. 이러한 처리 방식은 관계형

데이터베이스에 대한 쿼리 결과 셋의 레코드들을 리포트 뷰어에서 한 번에 메모리에 저장할 수 없는 상황을 고려한 것이다. XMLA를 지원하는 데이터 저장소들은 웹 서버를 내장하고 있으므로 OLAP과 데이터 마이닝 분석을 위한 요청은 이들 웹 서버로 직접 연결된다.

그림 7은 전체적인 시스템 구조이다. 그림 8, 9, 10은 세 가지 종류의 쿼리 편집기의 실행 화면이다. 그림 11, 12, 13은 세 가지 종류의 쿼리 결과를 리포팅 툴의 표현 컴포넌트를 통해 동적 문서에 반영한 실행 화면이다.

**6. 결론 및 향후 연구**

이 논문에서는 전통적으로 관계형 데이터베이스에 SQL을 사용한 쿼리 요청에 대한 결과 셋을 반영하여 동적 문서를 생성할 수 있는 리포팅 툴을 최근 기업에서의 활용이 활발해지고 있는 OLAP과 데이터 마이닝 분석을 함께 수행할 수 있는 시스템으로 확장하였다. 이 시스템은 기존 리포팅 툴이 취하는 thin client 구조 대신에 fat client 구조를 선택하였다. 그 이유는 기존 fat client 구조의 리포팅 툴은 동일한 문서를 서버 측에서 한 번 생성한 후 다수의 사용자가 조회하는 용도로 사용하기 때문에 이러한 구조가 동일한 문서에 대한 반복된 생성을 최소화 할 수 있기 때문이다. 그러나 본 논문에서 구현한 시스템은 사용자들이 각각의 개별적인 목적으로 OLAP과 데이터 마이닝 분석을 수행하는 과정에서 사용자에게 따라 상이한 데이터를 반영하는 동적 문서의 즉각적인 생성을 요구하기 때문에 서버의 처리량 증가에 따른 성능 저하 문제가 발생하므로 이를 해결하기 위하여 클라이언트에서 직접 동적 문서를 생성하는 구조를 선택하였다. 특히, 포맷 편집기에서는 세가지 데

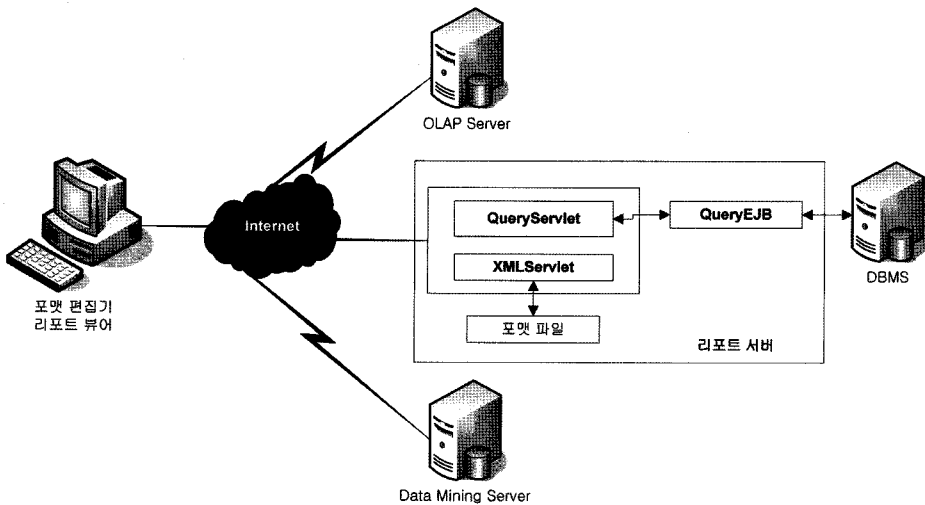


그림 7 전체 시스템 구성도

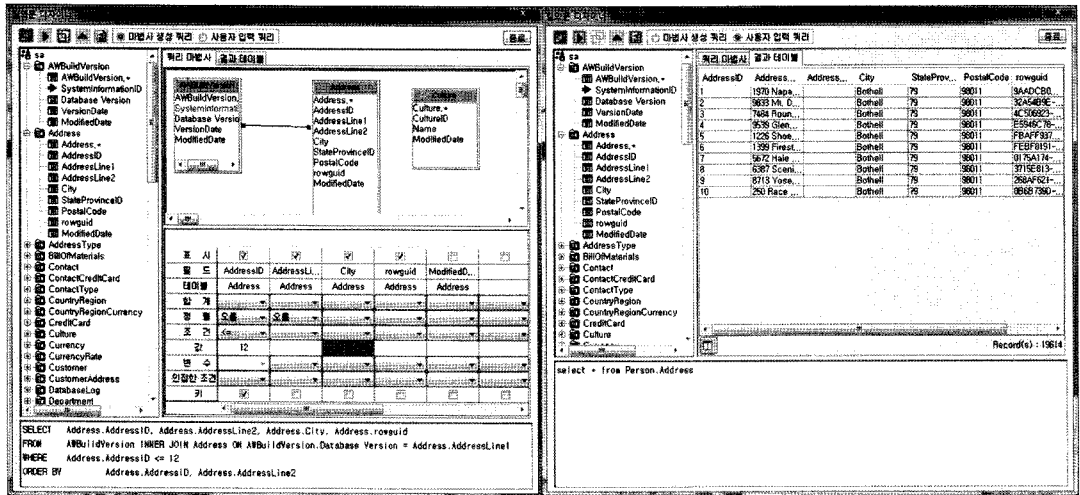


그림 8 SQL 쿼리 편집기

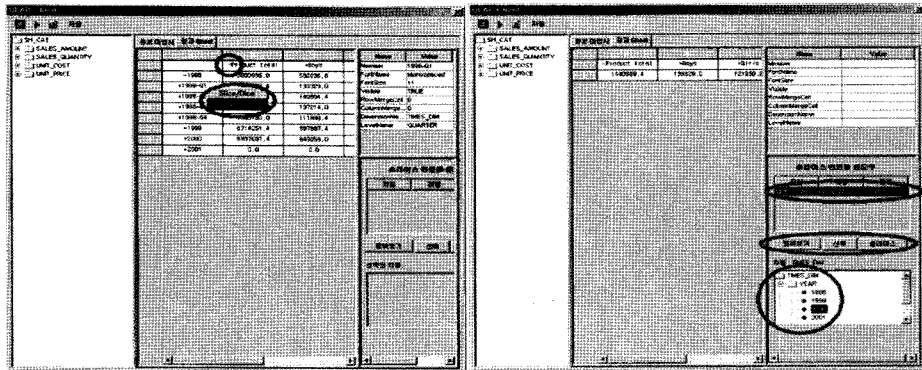


그림 9 OLAP 쿼리 편집기

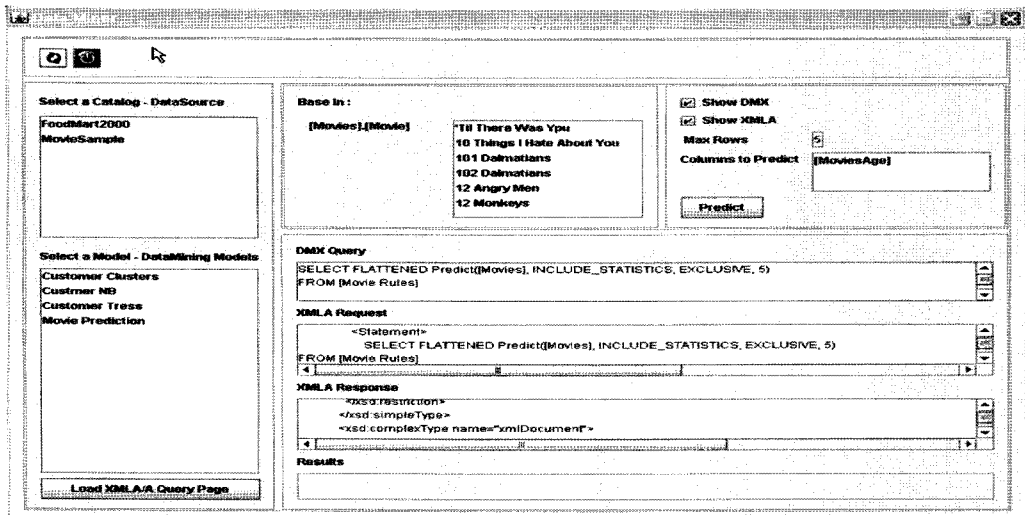


그림 10 데이터 마이닝 쿼리 편집기

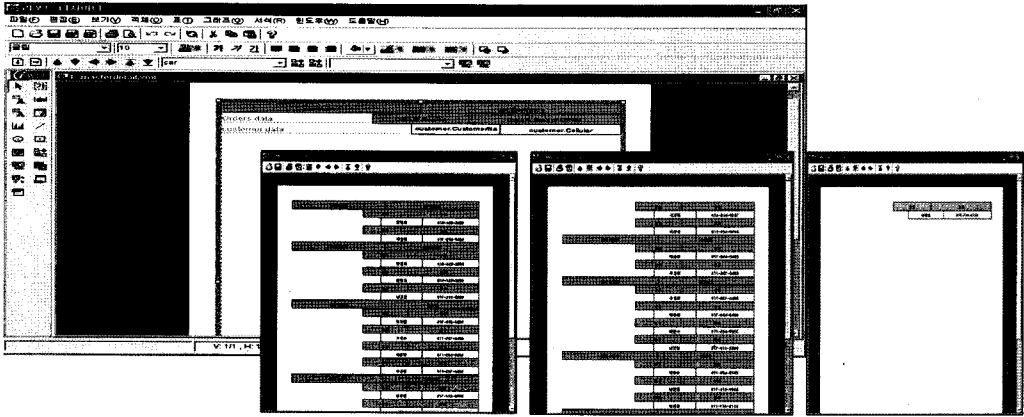


그림 11 SQL 분석 결과의 리포팅 툴로의 적용

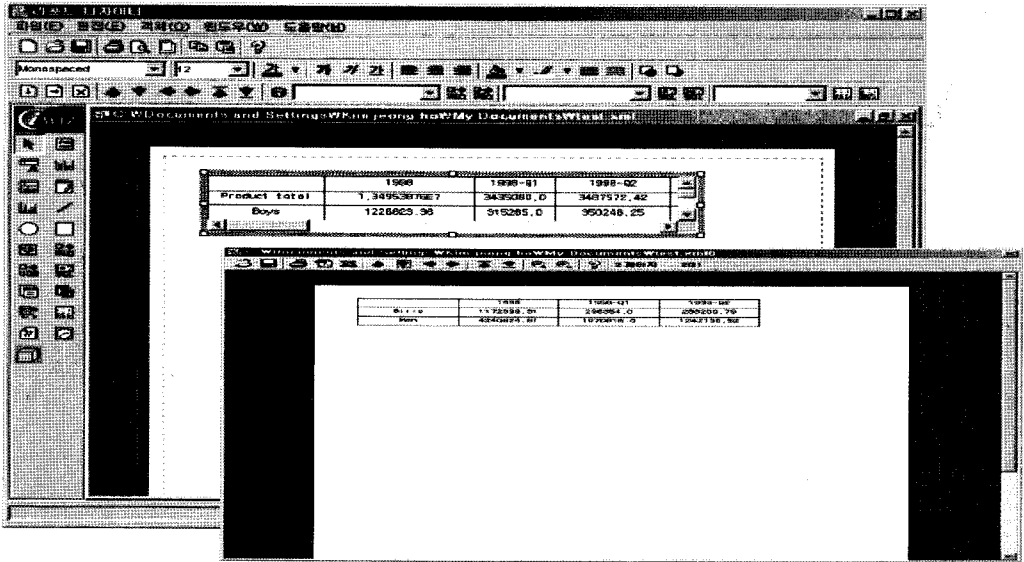


그림 12 OLAP 결과의 리포팅 툴로의 적용

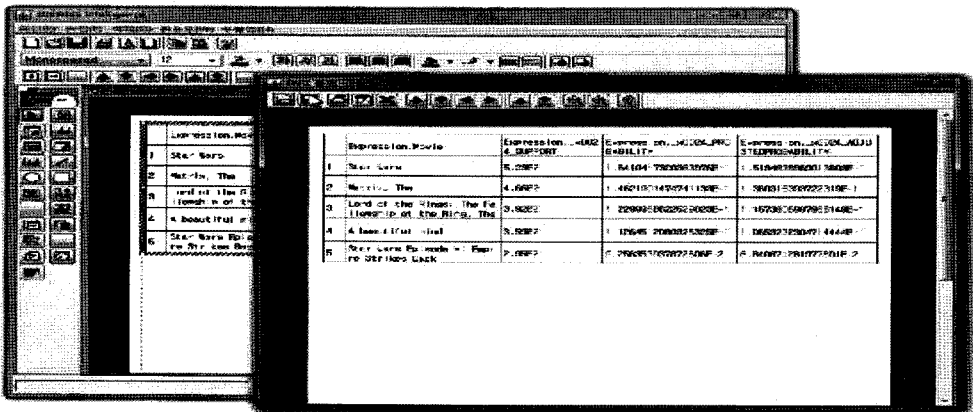


그림 13 데이터 마이닝 결과의 리포팅 툴로의 적용

이타 소스에 대하여 각각의 쿼리문 문법에 익숙하지 않은 사용자들을 고려하여 쿼리문을 생성할 수 있도록 하는 GUI 기반의 쿼리 편집기들을 구현하였으며 클라이언트의 제한된 실행 환경에서도 대량의 데이터를 가공 및 처리할 수 있는 구조와 처리 방식을 갖는 리포트 뷰어의 설계 및 구현 사항을 제시하였다.

리포트 뷰어는 이 시스템에서 지원하는 세가지 데이터 소스 즉, RDBMS, OLAP, 데이터 마이닝에 대한 각각의 쿼리 결과 데이터를 관리하는 쿼리 결과 컴포넌트들과 쿼리 결과 데이터를 실질적으로 사용하는 표현 컴포넌트들을 분리하여 설계 및 구현하였다. 이러한 구조는 각각의 표현 컴포넌트들이 그들이 필요로 하는 데이터를 직접 서버에 요청한 후 그 결과를 사용할 경우 클라이언트 측에 쿼리 결과 데이터의 중복이 발생하므로 리소스를 낭비하게 된다. 따라서 이러한 문제를 방지하기 위하여 각각의 표현 컴포넌트들이 필요로 하는 데이터를 메소드 호출 방식으로 해당 쿼리 결과 컴포넌트에 요청하며 쿼리 결과 셋 컴포넌트는 필요에 따라 서버에 데이터를 요청하는 방식을 취하도록 하였다. 이러한 방식은 하나의 표현 컴포넌트에서 각기 다른 데이터소스의 데이터를 용이한 방법으로 통합하여 반영할 수 있도록 한다. 그리고 각 표현 컴포넌트들과 쿼리 결과 셋 컴포넌트는 동일한 인터페이스를 가지고 통신하므로 인터페이스 규칙만 준수할 경우 필요에 따라 표현 컴포넌트들을 용이하게 추가할 수 있는 구조로 설계 하였다.

리포트 뷰어는 전처리 단계와 동적 페이지 생성 단계의 두 단계를 거쳐 동적 문서를 생성하도록 하였다. 전처리 단계는 동적 문서의 전체 페이지 수를 계산하면서 각 페이지에 대한 메타데이터를 생성한다. 동적 페이지 생성 단계는 사용자가 요청한 페이지 만큼 전처리 단계에서 생성한 메타데이터를 사용하여 생성한다. 전처리 단계와 동적 페이지 생성 단계는 동시에 처리된다. 즉, 전처리 단계가 완료하기 이전이라 하더라도 현재 전처리 단계에서 생성한 각 페이지에 대한 메타데이터 범위 안에서 사용자의 요청이 있을 경우 페이지를 생성할 수 있도록 하였다. 이러한 처리 방식은 대량의 데이터로 인한 대량의 페이지를 생성하게 되는 동적 문서라 하더라도 사용자에게 가능한 범위 안에서 빠르게 동적 문서를 제공하기 위함이며 요청이 있는 페이지 단위로 생성하는 것은 클라이언트의 제한된 메모리 환경을 고려한 것이다.

본 논문에서 구현한 시스템의 OLAP과 데이터 마이닝 분석을 위한 메시지 인터페이스는 기존의 벤더 종속적인 API를 사용하는 구조를 탈피하여 데이터 소스 벤더에 따른 통신 일관성을 유지하기 위하여 HTTP 프로토콜과 XML 기반의 메시지 포맷을 사용하는 XMLA를

사용함으로써 다양한 벤더의 데이터 소스에 동일한 방식으로 접근할 수 있도록 하였다.

리포트 뷰어는 Java 기반의 애플릿으로 구현하였으므로 웹 브라우저 안에서 실행 가능하며 자동으로 다운로드 되어 실행된다. 따라서 다양한 플랫폼에서 동일하게 동작하며 사용자로 하여금 별도의 설치 과정을 요구하지 않는다.

향후에는 OWL(Web Ontology Language)을 사용하여 구축한 온톨로지에 접근하여 SPARQL을 통해 정보 분석을 수행한 후 그 결과를 리포팅 툴에 반영할 수 있는 시스템으로 확장할 예정이다.

## 참 고 문 헌

- [1] Surajit Chaudhuri and Umeshwar Dayal, "An Overview of Data Warehousing and OLAP Technology," ACM:SIGMOD Record, 26:1, 1997.
- [2] Tetsuya, M. and Nobuo, T., "A Reporting Tool Using Programming by Example For Format Designation," Proceedings of 5th international conference on Intelligent user interfaces, ACM Press, 2000.
- [3] Gray J. et.al., "Data Cube: A Relational Aggregation Operator Generalizing Group-by, Cross-Tab and Sub Totals," Data Mining and Knowledge Discovery Journal, Vol.1, No.1, 1997.
- [4] Robert Chu, "Web Services Standards for Data Mining," KDD-2004 Workshop on Data Mining Standards, Services and Platforms(DM-SSP 04), Sunday, August 22, 2004.
- [5] XMLA, "http://www.xmlforanalysis.com"
- [6] Microsoft SQL Server 2005 Reporting Services, "http://www.microsoft.com/sql/technologies/reporting/default.aspx"
- [7] Oracle Reports, "http://www.oracle.com/technology/product/reports/index.html"
- [8] Crystal Reports, "http://www.businessobjects.com/product/catalog/crystalreports/"
- [9] Microsoft SQL Server 2005 Analysis Services, "http://www.microsoft.com/sql/technologies/analysis/default.aspx"
- [10] Oracle OLAP, "http://www.oracle.com/technology/product/bi/olap/olap.html"
- [11] Oracle Data Mining, "http://www.oracle.com/technology/product/bi/odm/index.html"
- [12] Microsoft SQL Server 2005 Data Mining, "http://www.microsoft.com/sql/technologies/dm/default.aspx"
- [13] 최지웅 외, "웹 기반의 Ad Hoc 리포팅을 위한 Fat Client를 갖는 리포팅 툴", 정보과학회 논문지 컴퓨팅의 실제, 제12권, 제4호, 2006.8.



최 지 응

2001년 숭실대학교 컴퓨터학부(학사). 2003년 숭실대학교 컴퓨터학과(석사). 2007년~2008년 고등기술연구원 연구원. 2003년~현재 숭실대학교 컴퓨터학과 박사과정. 관심분야는 분산/병렬 컴퓨팅, 시맨틱 웹, 웹서비스, BI, 보안, 유비쿼터스



김 명 호

1989년 숭실대학교 전자계산학과(학사) 1991년 포항공과대학교 전자계산학과(공학석사). 1995년 포항공과대학교 전자계산학과(공학박사). 1995년 한국전자통신연구소 선임연구원. 1998년~1999년 미국 테네시주립대 교환교수. 1995년~현재 숭실대학교 정보과학대학 컴퓨터학부 교수. 관심분야는 분산/병렬 컴퓨팅, 그리드, 웹서비스, BI, 보안