

# 자바기반 리포팅 툴에서의 OLAP 사용

김정호\*, 최지웅\*, 김명호\*

\*숭실대학교 컴퓨터학과

e-mail:jhkim@ss.ssu.ac.kr, jwchoi@ss.ssu.ac.kr,

kmh@computing.ssu.ac.kr

## Using OLAP in Java-based Reporting Tool

Jeong-Ho Kim\*, Ji-Woong Choi\*, Myung-Ho Kim\*

\*Dept of Computer Science, Soong-Sil University

### 요 약

리포팅 툴에서의 자료의 분석기능에 대한 요구가 커지고 있는 상황에서 기존의 리포팅 툴은 이러한 요구를 Crosstab을 이용해서 해결하려고 하고 있다. 하지만 데이터베이스에서의 자료 분석은 분석할 자료에 관련된 테이블 데이터를 모두 읽어서 처리를 해야 하기 때문에 리포팅 툴에서 Crosstab을 이용해서 데이터베이스의 자료를 읽어와 분석하는 방식은 데이터베이스에 부하를 많이 줄 뿐만 아니라 성능상에 문제가 많다.

이에 본 논문은 자료의 분석기능을 전문적으로 해주는 OLAP을 이용하여 OLAP에서 나오는 결과를 리포팅 툴의 테이블로 맵핑하여 리포팅 툴에서 OLAP의 분석기능을 지원 할 수 있도록 하였다.

### 1. 서론

리포팅 툴은 관계형 데이터베이스의 값을 이용하여 문서 내용과 문서 툴의 변환이 가능한 문서를 만들어 주는 툴이다. 리포팅 툴은 쉽게 문서를 배포할 수 있게 해준다.[1]

관계형 데이터베이스는 기업에서 매일 이루어지는 작업을 수행할 때 발생하는 기업의 업무 트랜잭션을 기록하므로, 이러한 시스템을 온라인 트랜잭션 처리 시스템(OLTP)이라고 한다.

OLAP(Online Analytical Processing)은 데이터 웨어하우스에 저장된 데이터를 다차원적으로 분석하여 그 결과를 온라인으로 사용자에게 제시하는 제반 기술이다.[2]

데이터 웨어하우스의 목적은 데이터를 분석하여 사용할 수 있도록 통합하고 정제하는 것이다. OLTP 데이터는 업무 트랜잭션을 처리하기에 적합한 구조로 되어 있으나 분석 처리가 어렵고 오래 걸리는 방식으로 되어 있다. OLTP 데이터가 데이터 웨어하우스로 이동될 때는 분석을 더욱 잘 지원하는 모양으로 변환된다. OLTP 데이터를 데이터 웨어하우스로 만드는 과정은 관계형 테이블에 저장된 OLTP 데이

터를 다차원 큐브에 저장된 OLAP 데이터로 구성하는 작업이다.

리포팅 툴이 사용하는 주요 표현도구중에는 Pivot Table 모양으로 자료를 표현하는 도구가 있다.

이 도구를 리포팅 툴에서 Crosstab[3][4]이라고 한다.

데이터 분석용 Crosstab의 문제는 OLTP를 사용하기 때문에 대용량의 분석 처리시 속도가 좋지 않고 데이터베이스에 부하를 많이 주게 된다.

본 논문에서는 자바기반 리포팅 툴을 바탕으로 기존 Crosstab의 기능을 OLAP을 이용하여 대체한다. 그래서 Crosstab을 이용하여 자료를 분석했을 때 나타나는 성능상의 문제와 데이터베이스에 생기는 부하를 OLAP을 이용함으로써 개선시킨다.

### 2. 관련연구

관련연구에서는 리포팅 툴에 대해 정리를 하고 리포팅 툴에 연결되어 OLAP의 기능을 할 수 있는 데이터베이스를 알아본다.

리포팅 툴은 국산으로 OZ와 Ubi Report를 외국산으

로 Style Report와 Crystal Clear를 조사했다.

OZ[3]

자바기반의 리포팅 툴로 Crosstab의 자체 내장 함수를 이용하여 결과 값을 변환할 수 있고 설정에 따라 결과값을 리스트 형태로 출력할 수 있다. OLAP을 이용한 분석기능은 없다.

Ubi Report[4]

자바기반의 리포팅 툴로 Crosstab의 자체 내장 함수를 이용하여 결과 값을 변환할 수 있고 OLAP 연산 중 1단계의 drill up/down을 지원하여 결과 값과 결과 값을 만드는데 쓰인 값을 바꿔가며 볼 수 있다. OLAP을 이용한 분석기능은 없다.

Style Report[5]

자바기반의 리포팅 툴로 배포 문서에서 지정된 표현도구를 사용하지 않고 문서에 사용되는 표현도구를 바꿔가며 문서를 보는 방식으로 만들어 졌다.

OLAP의 기능을 사용하는 소규모의 리포팅 툴과 많은 사용자에게 배포용 문서를 만드는 리포팅 툴로 제품을 나누어서 제공한다.

Crystal-Clear[6]

자바기반의 리포팅 툴로 Crystal Report의 문서를 읽고 쓸 수 있는 호환성을 가지고 있다.

JOLAP을 이용한 OLAP service를 제공할 계획이나 JOLAP이 아직 스펙이 아직 확정되지 않았다.

데이터베이스 기반의 OLAP tool로 MicroSoft SQL Server 2000과 Oracle 9i R2를 조사했다.

MicroSoft SQL Server 2000[7]

RDB와 OLAP엔진을 제공한다. OLE DB for OLAP interface와 구성 요소 개체 모델(COM) 이용해서 OLAP 데이터를 이용할 수 있다.

Oracle 9i R2[8]

RDB와 ROLAP엔진을 제공한다. BIBean이라는 컴포넌트와 JAVA OLAP API를 이용해서 OLAP 데이터를 제공한다.

기존의 연구를 바탕으로 요구사항을 정리하면 다음과 같다.

1. 리포팅 툴의 배포 문서에서 OLAP 분석의 결과를 나타낼 수 있어야 한다.

2. OLAP은 리포팅 툴에서 접속가능한 API가 있어야 한다.

본 논문에서는 기존의 리포팅 툴에 자바 OLAP API를 제공하는 OLAP을 연결하여 요구사항을 만족하는 리포팅 툴을 구현한다.

3. 시스템의 설계

3.1 전체 시스템의 설계

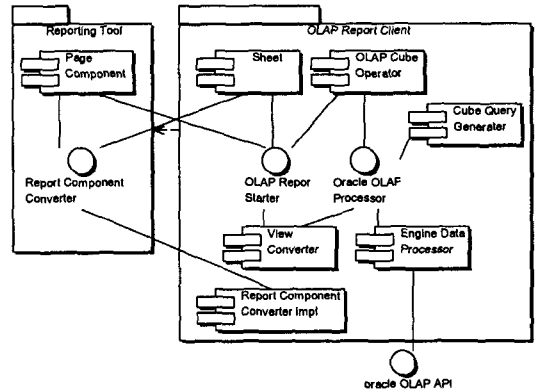


그림 2 전체 시스템의 Component Diagram

OLAP Report Client는 그림 1과 같이 구성되어 있다.

1. OLAP Client Starter

리포팅 툴에서 OLAP Report Client의 시작을 나타내는 인터페이스로 처음에 OLAP Cube Operator의 UI를 연결하여 OLAP Cube에 대한 결정을 하도록 해주고 View Converter에서 받은 Cube 결과값을 Sheet에 넘겨주는 역할을 한다.

2. OLAP Cube Operator

OLAP Cube 모양에 관련된 처리하는 컴포넌트로 Cube의 Pivot연산과 관련해서 앞으로 Cube의 결과값이 어떻게 보여질 것인가를 설정하고 자료의 Drill up/down 연산과 관련해서 원하는 Cube의 값을 얻기 위해 Dimension의 값을 선택한다.

3. Oracle OLAP Processor

Oracle과 관련된 처리를 하는 인터페이스로 Oracle과 관련된 컴포넌트를 조율하는 역할을 하고 있다. Cube Operator에서 결정된 Cube의 모양을 Cube Query Generator 보내주고 그것을 기반으로 만든 Oracle Query를 받는다. Query를 Engine Data Processor에 넘겨주고 Engine Data Processor에서 넘어온 Cube의 결과값을 View Converter에 넘겨주는 역할을 한다.

4. Cube Query Generator

OLAP Cube Operator에서 나온 Cube의 모양 정보를 바탕으로 Cube의 Drill up/down 기능에

해당하는 특정 자료를 선택하는 과정을 거쳐 Oracle에서 처리가능한 형태의 소스를 만드는 역할을 한다. (소스는 Oracle OLAP API에서 사용하는 쿼리 object이다.)

5. Engine Data Processor

Oracle OLAP processor에서 받은 Query 소스를 OLAP 엔진에 보내서 그곳에 있는 메타데이터나 Query의 결과값을 받는 역할을 한다. 결과값은 여러개의 list의 형태로 나오는데 이 list를 헤더가 한쪽에만 있는 Crosstab 모양으로 변환해 주어서 차후에 그 모양을 가공할 수 있게 한다.

6. View Converter

Engine Data Processor에서 나온 값은 헤더가 한쪽에만 있는 Crosstab 모양을 하고 있기 때문에 OLAP Cube Operator에서 결정한 형태로 Pivoting을 한후 Sheet에서 표현할 수 있는 형태로 바꿔주는 역할을 한다.

7. Sheet

설정대로 변환된 Cube의 값을 Sheet형식으로 화면에 표시해 주어 Cube의 값이 원하는 대로 나왔는지 확인을 시켜주는 역할을 한다. Cube의 모양과 관련하여 Sheet에서는 cell의 크기를 바꿀 수 있다. 그리고 Sheet에서 설정한 Cell의 모양 데이터와 cell의 값을 Report Component Converter에 자료로 들어가서 리포팅 툴의 component로 맵핑될때 그대로 적용이 된다.

8. Report Component Converter

Sheet에 표시된 값을 리포팅 툴에 추가할 수 있도록 리포팅 툴 Component를 만들어 준다. Component를 만든 후에는 OLAP Report Client를 접근하지 않고 Cube의 결과값을 볼 수 있다.

3.2 시스템의 세부설명

전체 시스템의 설계는 8가지의 컴포넌트와 인터페이스로 되어 있다. 이것들의 세부설명은 4개의 주제를 가지고 하도록 하겠다.

Oracle OLAP Source Process

Oracle의 OLAP API는 source라고 하는 쿼리 object 인스턴스를 사용한다. source는 일종의 데이터 셋으로 메타데이터에서 얻을 수 있는 object에 관해서 source를 추출할 수 있다. 추출된 source들을 가지고 서로 join, sorting, 함수등의 연산을 해서 원하는 자료로 구성된 source를 만들고 이것의 값을 데이터베이스에 접근해서 가지고 올 수 있다.

Cube의 값을 가지고 있는 measure와 사용자가 선택한 값을 가지고 있는 dimension을 join을 위해서는 각각의 source가 있어야 한다. dimension source는 나타낼 값이 정해져야 하는데 사용자가 선택한 값과 dimension source를 join 해서 source의 값을

만든다.

Crosstab form generation

Engine Data Processor에서 받은 결과 Crosstab은 한쪽 축에만 헤더가 있는 형태인데 OLAP Cube Operator에서 설정된 형태로 헤더에 있는 dimension들을 여러 축으로 나누어야 한다. 축들을 나누게 되면 cube의 값이 들어 있는 cell의 위치도 이에 따라 나누어지게 된다. 그림 2와 같이 cell을 리스트 형태로 나타냈을때 T가 P보다 정렬이 잘 되어 있다. cell의 위치를 옮길때에는 정렬이 잘된 값을 먼저 보내는 것이 알고리즘의 복잡도를 줄일 수 있다.[9]

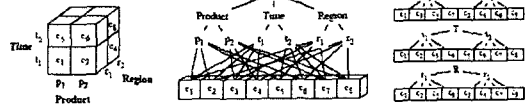


그림 2 셀의 형태에 따른 Dimension의 모양

Sheet

Sheet는 행, 열 헤더와 data cell로 구성되어 있다. 헤더에는 인덱스값이 들어가고 Cube의 값은 모두 data cell에 표시된다.

헤더를 이용해서 data cell에 위치와 크기를 조절할 수 있고 cell을 마우스로 클릭해서 선택할 수 있다. cell을 선택하면 cell의 대한 자료는 별도로 속성창에 표시가 된다.

Sheet를 끝내게 되면 Sheet에 있는 값은 Report Component Converter에 입력으로 들어가서 리포팅 툴에서 볼 수 있게 한다.

Report Component Converter

리포팅 툴에서 사용할 수 있는 컴포넌트로 바꿔주는 역할을 한다. Sheet에서 입력값을 받은 것을 리포팅 툴의 테이블과 같은 구조로 바꾼 새로운 컴포넌트를 만든다. 리포팅 툴에서 사용하는 테이블 저장방식은 최대한 살려서 한다.

4. 시스템의 구현

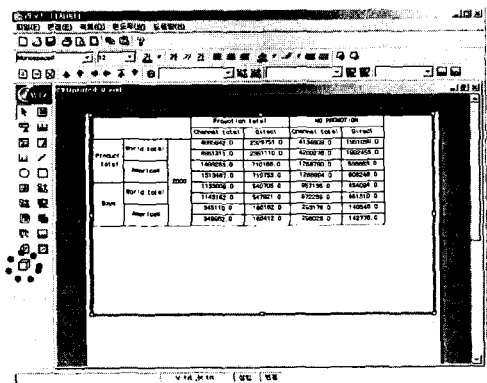


그림 3 리포팅 툴에 OLAP 결과값 나타낸 모습

그림 3은 시스템의 최종 결과값으로 리포팅 툴에

서 Sheet에 나타난 값을 표현한 것이다.

그림 3에서 나타난 아래 점선으로 된 부분의 아이콘을 클릭해서 Report OLAP Client를 실행 시키고 Cube의 모든 설정이 끝나면 Sheet에 보여지는 모양 그대로를 리포팅 틀에서 볼 수 있게 된다.

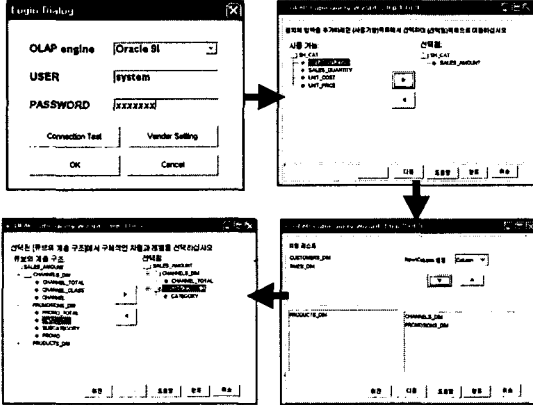


그림 4 Report OLAP Client의 설정 Dialog

그림 4는 Report OLAP Client를 실행시켰을 때 Dialog를 순서대로 표시한 것이다.

처음에 나오는 것은 OLAP Server Connection Dialog로 ID와 Password를 입력하고 Vender Setting에서 Oracle에서 사용하는 JDBC 설정을 한후 Connection Test에서 설정이 제대로 되었는지 확인을 한다. Vender setting과 Connection Test를 하지 않으면 OK를 눌러도 OLAP에 연결되지 않는다.

다음에 나오는 3개는 OLAP Cube Operator에서 쓰는 Dialog로 처음에 큐브를 선택하고 ▶를 누르면 선택된 큐브가 오른쪽에 표시된다. 다음을 눌러서 진행시킨다. 다음은 Dimension의 Layout을 설정하는 창으로 차원 리스트에서 차원을 설정하고 차원이 갈 방향(Row/ Column)을 설정한 후 ▼를 눌러서 차원을 이동 시킨다. 차원의 설정이 끝난후 다음, 마지막으로 Dimension의 Level을 설정하는 창이다 사용방법은 큐브선택과 같다. 모든 설정이 끝나면 Sheet가 나타난다.

Product	Region	Product Total	Production Total	NO PROMOTION	With Promotion	
			Channel Total	Direct	Channel Total	Direct
Product Total	North America	4996442.0	2327761.0	4134809.0	1961006.0	
	Asia Pacific	4861311.0	2261110.0	4200770.0	1962455.0	
Boys	North America	1480293.0	710166.0	1268760.0	508663.0	
	Asia Pacific	1519467.0	719753.0	1268904.0	608248.0	
Girls	North America	1123009.0	540706.0	967196.0	454094.0	
	Asia Pacific	1146622.0	579287.0	972289.0	461910.0	
Total	North America	3451110.0	1641822.0	2931716.0	1402549.0	
	Asia Pacific	3496822.0	1694112.0	2902925.0	1427796.0	

그림 5 Sheet에서 cell을 선택했을때의 모습

큐브에 대한 설정이 끝나면 그림 5와 같이 그 결과값을 표현해주는 Sheet가 나타난다. 이 Sheet에서는 헤더를 이용하여 셀의 크기를 바꿀 수 있고 셀을

선택할 경우 속성창에 셀에 대한 정보를 받아서 볼 수 있다. 이러한 모든 과정이 끝나면 처음에 보여진 그림에서처럼 Sheet에서 보여진 OLAP 결과값을 리포팅 틀에서도 볼 수 있다.

### 5. 결 론 및 향후과제

리포팅 틀에서 OLAP의 분석기능을 이용할 수 있도록 하였다.

향후에는 Sheet의 기능을 보강하여 Pivoting과 Drill up/down 의 기능을 추가하고 차트등의 GUI의 부분을 강화하여 cell을 숫자만 표현하는 것이 아니라 그림으로 표현하는 UI를 구성하도록 하겠다.[10]

### 참고문헌

- [1] 자바기반의 보고서 생성 시스템의 설계 및 구현, 최지용, 숭실대학교 대학원 석사학위 연구논문
- [2] 효율적인 ROLAP 큐브 생성 방법, 김명 송지숙, 한국정보과학회 논문지 29권 2호 99-109
- [3] <http://www.forcs.com/file/GettingStart.pdf>
- [4] [http://www.ubireport.com/files/UbiReport\\_kr.pdf](http://www.ubireport.com/files/UbiReport_kr.pdf)  
Ubi Report 소개 자료(2004.2.1)
- [5] <http://www.inetsoft.com/inetsoft/products/prodoverview.html>
- [6] <http://www.inetsoftware.de/English/Produkte/CrystalClear/JavaReportEngine/Default.htm>
- [7] MS SQL Server 2000 온라인 설명서 (업데이트-2004)
- [8] ORACLE 9i OLAP Release2 : A Relational - Multidimensional Database for Business Intelligence
- [9] A Visual Interface Technique for Exploring OLAP Data with Coordinated Dimension Hierarchies, CIKM'03, p.532-535
- [10] Stolet C., Tang D. and Hanrahan P. Query, Analysis and Visualization of Hierarchically Structured Data using Polaris In Procs. ACM SIGKDD, July 2002.