

웹 기반 데이터웨어하우스 질의 도구의 구현

박유림
이화여자대학교 과학기술대학원 컴퓨터학과
youlim@ewha.ac.kr

Web-based Query Reporting Tool for Data Warehouses

Youlim Park

요 약

데이터베이스에 저장된 수많은 자료들을 사용자의 질의에 맞추어 웹상에서 표현해주는 것이 바로 리포팅 툴의 역할이다. 그에 관한 연구들로 인해서 EASY BASE, ReportShop 과 같은 여러 리포팅 툴들이 상용화 되고 있다. 하지만 사용자가 보다 쉽고 직관적으로 XML문서를 작성하고 관리할 수 있는 툴들은 미흡한게 사실이다. 이 논문에서는 리포팅 툴의 기본이 되는 쿼리 결과를 웹상에서 표현해주는 xml 을 설계하였다.

1. 서 론

비즈니스 환경은 급격하게 Intranet, Internet 환경으로 변해가고 있다. 고객 서비스와 업무를 브라우저를 통해 처리하게 되면서 웹에서 기존 C/S 환경에서와 같은 출력물을 미리 보고 프린트 할 수 있는 기능이 필수적으로 요구되고 있다.

즉 DB를 가치 있는 정보로 만들어 내는 기능을 하는 BI(Business Intelligence) 시장 도래 함으로서 인터넷상에서 다양한 양식의 보고서를 작성 및 출력할 수 있는 웹 리포팅의 필요를 충족하는 리포팅 솔루션 도입의 가속화되고 있다.

웹은 자체적인 특성상 화면에 정보를 보여주는 데는 대단한 능력을 가지고 있다. 그러나 그 정보를 사용자가 원하는 형태로 출력하기란 그리 쉬운 문제가 아니다. 브라우저에서는 애플릿이나 액티브X 등의 기술을 활용해 인쇄 기능을 일부 지원하고 있지만 보고서를 작성하기엔 그 기능이 크게 부족한 상태다. 특히 차트나 프레임, 바코드 등 특수 컴포넌트로 구현되어 있는 화면은 출력을 원하는 사용자에게 어려움을 가중시키고 있다.

리포팅 툴은 데이터베이스에 저장된 수많은 자료들을 표준 양식의 보고서로 만들어주는 소프트웨어로 기업이나 기관에서 요구되는 수많은 문서 작성 업무를 효율적으로 만들어주며 사용자가 리포트를 생성하기 위한 도구인 디자이너 프로그램과 개발한 리포트의 결과를 보는데 필요한 뷰어 프로그램을 제공한다.

따라서 이러한 리포팅 툴과 관련 쿼리에 관한 결과를 사용자들에게 효과적으로 전달하기 위해 쿼리 결과를 XML 기반의 문서로 효율적으로 변환하고 저작함으로써 양질의 콘텐츠를 제공하는 제반기술 및 연구에 대한 필요

성이 커지고 있다. 본 논문에서는 XML을 사용하여 쿼리 결과를 웹 환경에 적합한 테이블 및 문서로 변환해주는 것을 구현함으로써 기본 테이블 및 크로스 테이블 제공이 가능하도록 하였다.

본 논문에서는 먼저 리포팅 툴 관련 연구들을 설명한다. 다음으로 XML DTD와 XML 시스템 구조에 대한 근간이 된 XMLA를 소개한 후 구체적인 구현내용을 소개할 것이다. 마지막으로 결론과 향후과제를 제시하고자 한다.

2. 관련연구

2.1 EASY BASE

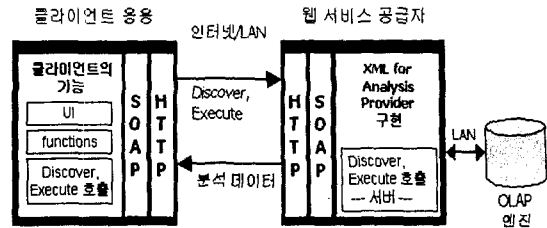
데이터베이스나 Sap R/3에 연결하여 데이터에 따라 변하는 정보집약형 보고서를 프로그래밍 없이 WYSIWYG 방식으로 빠르게 개발할 수 있는 EasyBase는 높은 개발 생산성과 고객에 대한 신속한 대응력을 제공한다.[1]

2.2 ReportShop

EnterpriseEnvironment(C/S,Web,Unix/Linux/Windows NT)에서의 복잡하고 다양한 보고서의 생성과 관리의 편의성을 위해 제공되어지고 항상 최신의 Data를 활용하여 Chart 분석 등의 Image 보고서 생성으로 사용자의 요구를 100% 충족시킬 수 있는 High Quality 보장을 제공한다. [2]

2.3 UbiReport

UbiReport는 JAVA환경에서 데이터베이스와 연동한 WYSIWYG(What You See is What You Get)을 지원하는 GUI Report 생성 Tool이다. 생성된 Report 파일을 UbiReport내에서 혹은 다른 Application이나 웹상에서 미리 볼 수 있고 또 출력할 수 있다.[3]



2.4 Oracle Discoverer

가장 직관적이며 사용이 쉬운 인터페이스를 통해 모든 유형의 데이터 애드혹 질의 및 분석을 실행할 수 있는 뛰어난 확장성 및 성능을 제공하고 더 많은 일반 사용자들은 사전 정의된 레포트와 그래프를 통해 데이터를 참조할 수 있다. Discoverer는 기초 데이터 구조의 복잡함을 감출 수 있는 비즈니스 뷰를 제공함으로써 사용자가 데이터 액세스 문제보다 비즈니스 문제를 해결하는데 좀 더 집중할 수 있도록 해준다.[4]

-단점

- 데이터와 서비스에 관한 구체적인 내용을 포함하는 xml 소스를 매번 생성해야 함
- 수행 후 반환된 결과 xml 문서의 데이터 요소에 접근하기 위해서는 xml 질의 언어를 사용해야 하는 불편함

3.XML/A

웹을 통해 데이터 provider와 Client application간의 Data Access 상호작용을 표준화 하기 위해 설계된 OLAP Api인 XML/A는 데이터의 검색만을 허용하며, 이를 수행하기 위해 Discover와 Execute 메소드를 제공한다. [5]

Discover 메소드는 메타 데이터 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.[5]

-Discoverer : 데이터 원본 또는 특정 데이터 원본에 대한 Cube, Dimension, value, level 등의 OKAP Object 들을 검색

-Execute : OLAP 명령문을 사용하여 Cube를 Slice, Dice, Drill-down, Roll-up하여 다차원 데이터를 검색

4. 구현

4.1 XML DTD

우선 테이블에 적용될 수 있는 Xml DTD를 구현한다. 크게 METADATA와 ROWDATA로 나누어지는데 전자부분에는 쿼리 결과에 관한 축들이 표현되어지고 후자 부분에는 각각의 값들이 표현되어진다. METADATA와 ROWDATA의 연결은 각각의 아이들을 부여하여 인식하게 하였다.

```
<?xml version="1.0" encoding="euc-kr" ?>
<ELEMENT Test <METADATA, ROWDATA>
<ELEMENT METADATA (ROWTABLE*, ColumTABLE*)>
<ELEMENT ROWTABLE (ROWID, ColumID*, attrName, FIELDTYPE)>
<ELEMENT ROWID (#PCDATA)>
<ELEMENT ColumID (#PCDATA)>
<ELEMENT attrName (#PCDATA|EMPTY)>
<ELEMENT FIELDTYPE (#PCDATA)>
<ELEMENT ColumTABLE (ColumID, ROWID, attrName, FIELDTYPE)>
<ELEMENT ColumID (#PCDATA)>
<ELEMENT ROWID (#PCDATA)>
<ELEMENT attrName (#PCDATA|EMPTY)>
<ELEMENT FIELDTYPE (#PCDATA)>
<ELEMENT ROWDATA (ROW*)>
<ELEMENT ROW (ROWID, ColumID, VALUE)>
<ELEMENT ROWID (#PCDATA)>
<ELEMENT ColumID (#PCDATA)>
<ELEMENT VALUE (#PCDATA)>
```

4.2 XML TABLE

```

<?xml version="1.0" encoding="euc-kr"?>
<!DOCTYPE Test SYSTEM "/Test.dtd">
<Test>
<METADATA>
<ROWTABLE>
<ROWID>1</ROWID>
<ColumnID>1</ColumnID>
<attrName> </attrName>
<FIELDTYPE></FIELDTYPE>
</ROWTABLE>
<ROWTABLE>
<ROWID>1</ROWID>
<ColumnID>2</ColumnID>
<attrName>Youlim</attrName>
<FIELDTYPE>string</FIELDTYPE>
</ROWTABLE>
:
:
<ColumTABLE>
<ColumnID>1</ColumnID>
<ROWID>1</ROWID>
<attrName></attrName>
<FIELDTYPE>string</FIELDTYPE>
</ColumTABLE>
<ColumTABLE>
<ColumnID>1</ColumnID>
<ROWID>2</ROWID>
<attrName>SALARY</attrName>
<FIELDTYPE>fixed</FIELDTYPE>
</ColumTABLE>
:
:
</METADATA>
<ROWDATA>
<ROW>
<ROWID>1</ROWID>
<ColumnID>2</ColumnID>
<VALUE>50000000</VALUE>
</ROW>
<ROW>
<ROWID>1</ROWID>
<ColumnID>3</ColumnID>
<VALUE>40000000</VALUE>
</ROW>
<ROW>
<ROWID>1</ROWID>
<ColumnID>4</ColumnID>
<VALUE>50000000</VALUE>
</ROW>
:
:
</ROWDATA>
</Test>
    
```

4.2.1 XML TABLE

	Youlim	HyuSook	HeeJung
SALARY	50000000	40000000	50000000
CITY	Seoul	Pusan	KwangJu
Univ.	SungShin	Ewha	DokSung

5. 결론

본 논문에서 구현한 XML DTD와 XML은 쿼리 결과를 XML 기반의 테이블로 효율적으로 변환하고 사용자가 보다 쉽고 직관적으로 XML문서를 작성하고 관리할 수 있도록 하였다는 점에서 그 의미가 크다 하겠다.

현재 이러한 기능을 제공하는 상용화된 제품은 점점 늘어나는 상황이다. 하지만 '기업이나 공공기관에 축적되어 있는 방대한 양의 DB는 어떻게 활용할 것인가?', '기업경영이나 경영성과분석', '고객관리', '마케팅 계획 수립 보고', '각종 현황 보고' 등 Quality 높은 보고서를 생성하기 위한 리포팅 툴의 존재는 미흡하다.

따라서 사용자의 질의 결과에 맞게 더 쉽고 편리하게 리포팅 해 줄 수 있는 시스템 아키텍처를 만드는 것은 앞으로 많은 연구가 필요한 분야이다.

참고문헌

- [1] EASYBASE <http://www.easybase.co.kr/>
- [2] ReportShop <http://www.h2osystech.com/>
- [3] UbiReport <http://www.ubireport.com/>
- [4] Oracle <http://otn.oracle.co.kr>
- [5] 배은주, 김명 A Web Services-based Client OLAP API and Its Application to OLAP Cube Browsing, 2002
- [6] Masuishi, T., Takahashi, N A Reporting Tool Using "Programming by Example" For Format Designation, 2000
- [7] Sérgio Mergen, Carlos Alberto Heuser, Eduardo Kroth AXIS a Java implementation of a data exchange architecture among heterogeneous data sources using XML documents , 2001