

JDBC를 이용한 웹기반의 통합 객체 관리 모델 설계

선수균*, 송영재**

*동원 대학 사무자동화과

**경희대학교 전자계산공학과

Design of Web-based Integration Object Management Model by Using JDBC

Su-Kyun Sun* , Yong-Jea Song**

*Dept. of Office Automation, DongWon Collage, Korea

**Dept. of Computer Engineering, KyungHee University, Korea

요 약

최근 전산 환경은 웹 환경에 적합한 개방형 시스템으로 변모하고 있다. 서로 다른 플랫폼을 기반으로 한 클라이언트들과의 연동을 위해서, 각 시스템에 따른 어플리케이션이 개발되어야 했다. 이러한 문제점을 극복하기 위해 JDBC를 이용한 이기종간의 시스템을 통합할 수 있는 통합 시스템 필요하다. 따라서, 본 논문에서는 인터넷 환경하에서 생성되는 산출물을 객체 형태로 통합 관리하고, 데이터베이스와 연결하여 객체들을 효율적으로 관리해 주는 웹기반 통합 객체 관리 모델을 설계한다. 이 모델로 기존의 시스템을 재사용하고 급변하는 소프트웨어 산업에 능동적으로 대체와 소프트웨어 개발에 시간을 단축함으로써 현존하는 다양한 데이터베이스군들을 최소한의 코드 수정을 통하여 구동할 수 있게 함으로써 경제성을 높이는 것이 본 논문의 목적이다.

1 서론

객체 지향 패러다임의 확산으로 인하여 소프트웨어 개발을 위한 객체 모델의 사용이 일반화되고 있다[1]. 또한 웹 환경의 시스템이 확산되고 있는 추세이다.

기존의 ODBC를 기반으로 구현된 데이터베이스 연동 어플리케이션의 경우, 데이터베이스와 어플리케이션에서 부수적인 문제점들이 발견되고, 서로 다른 플랫폼을 기반으로 한 클라이언트들과의 연동을 위해서, 각 플랫폼에 따른 어플리케이션이 개발되어야 했다. 또한 21세기 환경에 능동적으로 적응 할 수가 없으며 웹 환경하의 시스템을 개발할 수가 없었다. 이러한 문제점을 극복하기 위해 데이터 베이스의 효율적인 관리와 이 기종간의 시스템을 통합할 수 있고, 웹 환경에 적합한 통합 시스템이 필요하게 되었다[2,3,4,8,9].

따라서 본 논문에서는 웹환경에서 통합 관리하고, 데이터를 수집하고 검색하여 최적의 데이터로 저장할 수 있는 저장소를 설계하고, 그런 산출물을 객체 형태로 통합 관리하는 웹기반의 통합 객체 관리 모델이다. 이것은 분산 데이터 처리를 위한 표준화 작업을 이루어 현존하는 다양한 플랫폼 및 응용 시스템을 그대로 살리면서 통합환경에서 웹 환경을 보완하여 설계한 것이 통합 객체 관리 모델(Integration Object Management Model)이다.

본 논문의 목적은 통합객체 관리 모델을 중심으로

기존 시스템을 통합에 필요한 환경을 만들어 효율적으로 관리함으로써 소프트웨어 재사용성을 향상시켜 생산성을 극대화시키는 것이 목적이다. 또한 객체지향 소프트웨어 공학 프로세스에서 발생하는 다양한 산출물을 데이터 베이스화하여 필요한 산출물을 관리하고, 정보 저장소를 오라클 데이터 베이스로 관리함으로써 클라이언트와 서버를 연결하는 기능을 JDBC를 이용하여 웹 환경에 쉽게 적용할 수 있도록 설계했다. 이것을 기초로 생성기를 생성하여 개발환경에 적용할 수 있는 새로운 코드를 만들어 궁극적으로 객체지향 소프트웨어 개발의 생산성을 대폭 개선시키는 데 있다.

자바 언어의 특성을 구현의 융통성도 높다고 할 수 있는 JDBC(Java Database Connectivity)를 이용하여 웹 기반의 통합 객체 관리 모델을 설계함으로써 소프트웨어 개발한다면 장점은 다음과 같다. 장점은 platform에 구애받지 않는 실행 환경 제공과 코드의 수정 없이 다양한 DBMS와 연동 가능한 점을 볼 때, 재사용성(Reusability)가 뛰어나다[4]. 또한 JFC(Java Foundation Classes : Swing)을 이용하여 platform에 관계없이 동일한 UI(User Interface)의 제공이 가능하다.

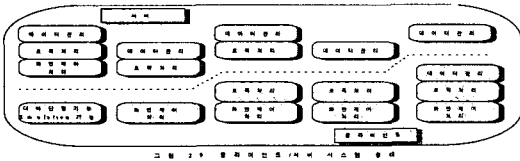
따라서 본 논문에서 설계한 시스템은 인터넷에서 데이터를 수집하고, 정리하여 관리하고, JDBC Thin driver를 이용하여 Oracle DB와 연동하고, JDBC를 이용하여 데이터 베이스와 원격지 사용자간의 보다 효과적인 이용이 가능하게 하였다. 또한 기존 AWT가

platform의 peer 객체에 따라서 서로 다른 UI를 제공하는 문제점의 해결을 위해, JFC를 사용하였고, 적용 설계로 시시각각 변화하는 날씨 데이터를 웹 상에서 데이터를 수집하여 데이터베이스에 관리함으로써 정보를 쉽게 제공할 수 있도록 설계했다.

2. 관련 연구

2.1 클라이언트/서버 시스템의 형태

클라이언트/서버 시스템의 형태를 하나의 그림으로 정리하면 다음과 같다.

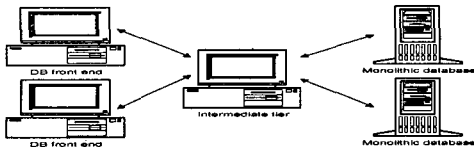


2.2 DBMS 모델

DBMS 모델은 일반적으로 데이터베이스(DB)와 사용자(Client)간의 관계로 정의할 수 있으며, 일반적으로 크게 세 가지로 분리되나, 오늘날 가장 많이 쓰이는 방식으로는 Two-Tier 모델과 Three-tier 모델 두 가지가 있다.

Two-Tier 모델의 경우, DB front end와 DB engine으로 분리되고 다른 방식으로는 지역데이터와 원격 데이터 분리할 수 있다. Two-Tier 모델의 경우 UI를 분리할 수 있으므로 개발이 용이하나, mirroring, caching, proxy services, secure transactions 등의 문제가 있다.

Three-Tier 모델은 Two-Tier 모델에서 발생하는 mirroring, caching, proxy services, secure transactions등이 해결되었고, 실제적 구조는 그림 3과 같다.[7]



(그림2) 3 - Tier Database

2.3 JDBC

JDBC의 특징은 다음과 같다. JDBC란 SQL을 실행하기 위한 자바 API라고 할 수 있으며 다음과 같은 특징이 있다. 표준 SQL 데이터베이스 접근 인터페이스이다. RDB에 대해 일관된 인터페이스 제공을 한다. 고수준의 도구나 인터페이스를 작성하기 위한 일반적인 표준이 될 수 있다.

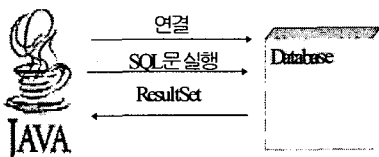


그림 3. JDBC의 연결

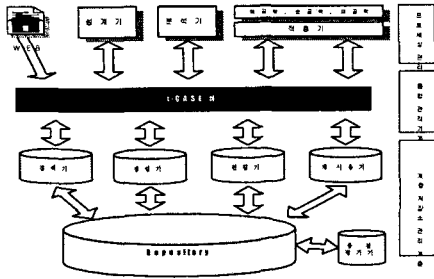
본 논문에서는 windows NT용 Oracle 데이터베이스(DB)를 사용하고 JDBC를 이용하여 3-tier의 형태로 구성하여 설계한다.

3. 통합 객체 관리 모델

[9] 단점인 이 기중간의 통합불능과 서비스 품질 향상을 위한 제어 기능 부재로 효과적인 관리를 할 수가 없었고 웹과 연동 할 수 있는 환경이 매우 미약했다. 따라서 본 논문에서는 이 기중간의 시스템을 통합할 수 있는 Middleware의 선정이 필요하게 되었고 보다 효율적으로 관리해 주고 급변하는 소프트웨어 산업에 능동적으로 대처하고, 분산객체 환경에서 프레임워크 객체 모델로 현존하는 다양한 플랫폼 및 응용 시스템을 그대로 살리고 웹 환경을 추가하여 클라이언트와 서버간 JDBC를 이용해서 효과적인 관리를 할 수 있는 모델을 설계한다. 또한 객체저장소의 효율적인 관리와 검색을 위한 연결관계 관련성을 부여함으로써 객체 모델을 바탕으로 라이브러리에 효과적인 저장이 이루어지도록 설계한다. 웹기반 통합 객체 관리 모델(Integrated object Management Model)은 세 계층으로 나누어지는데 분류계층과 통합 관리기 계층, 저장계층이다. 분류를 View 객체라 하는데 사용자에게 정보를 디스플레이 하는 객체이다. 통합 관리기를 제어객체라 하는데 사용자가 입력을 처리하는 객체이다. 저장을 모델 객체라 하는데 데이터를 갖는 모델 객체이다. 또한 이 객체들은 소프트웨어 라이프사이클의 효율적인 관리를 하며, 분산객체 환경에 적합한 구조를 만드는 데 중점을 두고 소프트웨어 개발에서 생성된 산출물들을 관리 할 수 있는 것이 객체저장소까지 포함하고 있다. 따라서 [9]에서 제안한 모델을 웹기반으로 확장시켜 웹기반 통합 객체 관리 모델(Integrated object Management Model)을 설계한다. 개발기간을 단축시켜 소프트웨어 산출물들을 효율적으로 유지보수 하고 저장 관리하여 생산성을 극대화시키는 것이 본 논문에서 제안한 목적이다. (그림 4)은 통합 객체 관리 모델 전체 구성도를 나타낸 것으로 세 가지 계층으로 구성되는데 프로세싱 관리계층, 통합 관리기 계층(I-CASE Manager), 마지막으로 객체 저장소 관리 계층이다.[9,10]

3.1 프로세싱 관리 계층

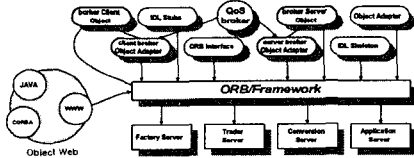
이 설계단계에서는 소프트웨어 라이프사이클 관리를 하기 위한 생명주기 서비스를 제공하고, 객체지향 소프트웨어 개발과정 중에는 문서, 다이어그램, 원시코드, 방법론 정보, 설계 패턴 등의 프레임워크, 인터넷 정보를 파싱하는 단계, 사용자 인터페이스 객체, 데이터 베이스 스키마 객체 등의 다양한 산출물이 있는데 이런 산출물들을 분류하는 방법을 제시하는 것이 프로세싱 관리 계층 설계이다. 특히 인터넷 정보를 Html 파일을 파싱하는 단계가 설계한다.



(그림 4) 통합 객체 관리 모델 전체 구성도

3.2 통합 관리기 계층

제어 메커니즘이 없고, 분산 객체 시스템에서 클라이언트와 서버에 각각 프로세스와 데이터의 분형과 불균형으로 통합에 필요한 적절한 아키텍처가 결실하게 필요하게 되었다. 통합 관리기는 이러한 단점을 보완하기 위하여 제어기능을 더 보완 처리한 것이다. 프레임워크 기반의 통합은 실제 통합에 필요한 설계 구현 비용을 최소화한다. 통합 관리기는 이러한 단점을 보완하고 프레임워크 기반의 통합은 실제 통합에 필요한 설계 구현 비용을 최소화 할 수 있다. 통합 관리기(I-CASE Manager)의 구조를 나타낸 것이 (그림 5)과 같다.



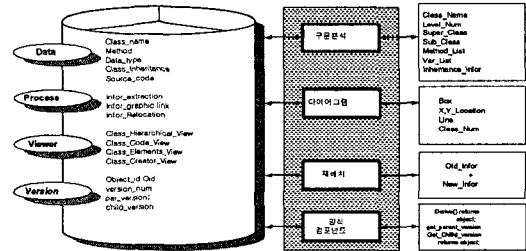
(그림 5) 통합 객체 관리기 구성도

구성은 프레임워크/ORB, 객체들의 위치 정보를 제공하는 트레이딩 서버(trading server), 데이터들의 변환을 담당하는 변환 서버(conversion server), 각종 데이터 객체들의 생성·복사·삭제를 담당하는 팩토리 서버(factory server)[9], 소프트웨어 QoS서버, ObjectWeb로 구성된다. 변환 서버는 서로 다른 데이터 표현을 사용하는 응용들간의 자료 변환에 필요한 정보 및 변환 서비스를 지원하며, 트레이더 서버는 객체들간의 위치 투명성을 보장하기 위해 적절한 서비스를 동적으로 선택할 수 있게 하며, 통합 객체 관리기(I-CASE M)는 ObjectWeb이란 소프트웨어 아키텍처를 사용하고 WWW와 CORBA로 연결하여 통합 관리하는데 필요한 구조를 만들어 통합 관리하는데 용이하게 했다. CORBA는 분산 컴퓨팅 환경과 이기종 분산 환경의 시스템통합을 위한 표준이다. CORBA는 효과적인 시스템통합을 위해 기술적인 이익을 제공하며 이기종의 시스템들의 분산 의사소통 환경을 위한 하부구조를 제공한다. Broker객체는 서비스 품질의 향상을 위한 제어 메커니즘으로 클라이언트 응용 프로그램과 클라이언트가 요구하는 서비스를 제공하는 서버들을 중재하는 제어기법이다. SQoS 서버의 네 가지 형태 중에서 하나가 되기 위한 객체의 모델 변환, 제어과정에 적용되는 것으로 Static QoS와 Dynamic QoS로 구별

된다. Broker 객체는 서비스 품질의 향상 제어 기능을 수행하기 위해서는 기존의 ORB에 BOA (Broker Object Adapter)를 추가하게 된다. BOA는 클라이언트 BOA와 서버 BOA로 구성된다. 클라이언트 BOA와 서버 BOA가 존재하는 이유는 BOA가 제공하는 QoS 제어 메커니즘은 분할 레벨에 따라 다음과 같이 네가지로 나눌 수 있는데 첫째는 Process Data Server QoS 제어 메커니즘과 둘째는 Process Balanced QoS 제어 메커니즘, 셋째는 Data Balanced QoS 제어 메커니즘과 끝으로 Process/data Balanced QoS 제어 메커니즘으로 이루어 지는데 본 논문에서는 Process/data Balanced QoS 제어 메커니즘으로 적용을 했다.

3.3 객체 저장소 관리 계층

앞 계층에서 분류가 결정되면 효율적인 검색과 재사용 할 수 있도록 객체저장소에 저장해야한다. 이런 객체 저장소의 주요기능으로는 많은 산출물을 관리해야 하는데 저장소 관리 계층은 이런 관리를 하는 계층이다. 또한 다양한 도구와 사용자의 동시 공유를 위한 병행 처리, 주요 산출물 객체에 대한 등급별 접근 처리와 보안처리, 시스템 장애 발생에 대한 회복, 객체의 올바른 상태를 유지하기 위한 무결성 처리 기능이 제공되어야 한다.



(그림 6) 객체 저장소 모델

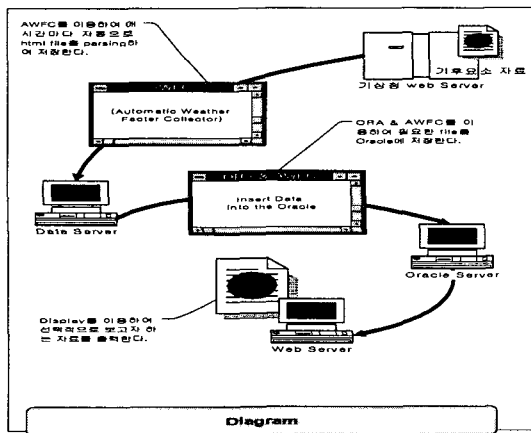
이러한 기능은 객체 저장소를 파일 시스템을 기반으로 하지 않고, DBMS를 기반으로 개발함으로써 제공받을 수 있다. 또한 도메인에서 발견되는 영구적인 정보나 데이터를 저장 및 관리하는 것인데, 객체지향의 저장소 관리 계층은 객체들로 구성되어 있다. 또한 정보 저장소는 객체저장소(Object Repository)라 하고 부품을 클래스별로 저장한다. 이런 객체 저장소를 관리할 수 있는 기능이 객체 저장소 관리 기능이다.

웹기반 통합 객체 관리 모델 설계를 “기상청 정보”에 적용사례를 들어 적용해본다. 오늘날 날씨는 우리의 생활과 밀접한 관련이 있는데 비해서, 웹상에서 실시간으로 날씨정보를 제공하는 사이트는 기상청 사이트 단 한 곳 뿐이다. 그러나 기상청은 73개 관측소에서 제공되는 그날의 자료만을 제공하고 있다. 이에 본 프로그램은 기상청에서 제공하는 매시간 기상자료를 파일로 자동 저장하는 프로그램과 선별된 파일을 Oracle DB에 저장하는 프로그램, Oracle DB에 저장된 자료를 웹상으로 제공하는 프로그램을 사용하여, 사용자가 원하는 날짜의 기상관측자료를 웹상으로 제공하는 시스템을 설계하려고 한다. 본 적용사례는 사용 목적에 따라 크게 다음과 같이 세 가지로 구분되며, 그 내용은 다음

과 같다.

첫째는 Automatic Weather Factor Saver이다.

기상청에서 제공하고 있는 기상정보관련 html 파일을 parsing하여 필요한 정보만을 텍스트 파일로 저장하는 프로그램으로 한 시간을 주기로 프로그램이 실행된다. 관측소 지점, 풍향 같은 문자들은 미리 정의된 숫자로 자동 변환되어 저장되고, 파일의 이름은 월, 일, 시간이 결합된 문자를 파일의 이름으로 하여 저장한다.



[그림 7] 모델 제시도

둘째는 JDBC & Oracle Saver이다. AWFS에서 저장된 파일을 선별적으로 선택하여 이를 Oracle server 에 저장한다. table의 이름은 파일의 이름으로 하고, 7개의 기상요소는 숫자로 변환된 그대로 저장 된다.

셋째는 Weather Data on Demand이다.

저장된 Oracle Server에 데이터를 사용자가 원하는 날짜, 시간을 선택하면 동적으로 html을 생성, 서비스하는 프로그램이다. Servlet환경을 위해서 Java Web Server가 사용되었으며, Oracle Windows Version 7.7.3이 데이터베이스로 사용되었다.

JDBC thin driver을 이용한 설계부분은 다음과 같다.

```

static final String driver_class = "oracle.jdbc.driver.Oracle
Driver";
static final String connect_string = "jdbc:oracle:thin:system
/manager@negis.kyunghee.ac.kr:1526:ORCL";
public Office(String setTitle)
{
    super(setTitle);
    setLayout(new BorderLayout());
    try
    {
        Class.forName(driver_class);
    }
    catch(java.lang.ClassNotFoundException e)
    {
        System.err.println("class not found");
        return;
    }
    try
    {
        myConnection=DriverManager.getConnection(connect_string);
        System.out.println("connected !!");
    }
    catch(SQLException e)
    {
        System.err.println("sql exception occurs" + e.getMessage());
        return;
    }
}
    
```

4. 결론 및 향후 연구과제

본 논문에서는 웹기반 통합 관리 모델을 설계 제시 한다. 이 모델을 적용하여 실시간으로 제공되는 기상청 정보를 인터넷으로 이용하여 제공받는다. 기상청에서 매시간 홈페이지를 통해서 제공하는 기후요소를 parsing하여 자동적으로 file형태로 저장한다. (http://www.kma.go.kr/weather/curr/k_curr_weath.htm) 자동적으로 저장된 텍스트 파일중에서 필요한 파일만을 선별하여 Oracle Database에 저장한다. 이 저장된 곳은 객체 저장소로 이 데이터베이스와 JDBC를 연결하여 필요한 정보를 파싱하여 JFC(Swing)을 이용하여 RDBMS와의 연동을 기반으로 한 application 개발 설계에 초점을 두었으며 이를 통해서 보다 빠르고 안정적으로 원격지 상에서의 사용자 사용환경에 중점을 두었다. 향후 연구과제로서 프로세서 계층에서 분류하는 알고리즘을 제시해야하고, 산출물 객체를 효율적으로 관리하는 객체 저장소 관한 연구를 할 것이고 이를 기반으로 하여 Java servlet을 이용한 WEB상에서의 가상 견적시스템등의 개발이 가능하며, 이 기종 DBMS을 사용할 경우, CORBA와 Java를 연동하여, 이기종 DBMS를 기반으로 다중 platform에서 작동하는 application의 구현에 중점을 둘 것이다.

참 고 문 헌

- [1] Jon Meyer & Troy Downing "JAVA Virtual Machine", O'REILLY, 1997
- [2] Mary Campione, Kathy Walrath, "The Java Tutorial", Addison-Wesily, 1996
- [3] Elliotte Rusty Harold, "Java Networking Programming", O'Reilly & Associates, Inc., 1996
- [4] Prashant Sridharan, "Advanced Java Networking", Prentice-Hall, Inc., 1997
- [5] Orfali, Harkey and Edwards, "The Essential Distributed Objects Survival Guide", Wiley Press, 1996
- [6] Jim Waldo, Geoff Wyant, Ann Wolrath and Sam Kenedall, "A Note on Distributed Computing", Sun Microsystems, 1994
- [7] "JDBC specification", http://splash.javasoft.com/jdbc
- [8] Emerson, Darnovsky, and Bowman, "The Practical SQL Handbook", Addison-Wesley, 1989
- [9] 선수균, 송영재, "통합객체지향 관리기 중점들 둔 F77/J++ 생성기 설계", 99년 추계 학술발표집 한국정보과학회, 1999.
- [10] 선수균, 송영재, "통합 객체 관리 모델을 위한 F77/J++ 생성기에 관한 연구", 정보처리 학회 논문지 제 7권 10호, 한국정보처리학회, pp. 3064-3074 2000.11