

# 감성정보관리시스템을 위한 데이터베이스 래퍼의 설계

구홍서, 박철순, 고한우\*  
청주대학교 컴퓨터정보공학과, 한국표준과학연구원\*  
e-mail: hskoo@chongju.ac.kr

## Design of Database Wrapper for Human Sensibility Information Management System

Heung-Seo Koo, Chul-Soon Park, Han-Wo Ko\*  
Dept. of Computer & Information Eng., Chongju Univ.,  
Korea Research Institute of Standards and Science\*

### 요 약

감성정보관리시스템은 주거·사무환경에 속한 6개 분야의 연구과제에서 생성된 피험자의 생리신호와 주관적 감성평가 데이터를 통합 관리하는 시스템이다. 본 논문에서는 여러 형태의 감성평가 실험데이터를 데이터베이스에 저장·처리하기 위해 감성정보 데이터베이스에 대한 접근을 하나의 컴포넌트에 집중시킨 데이터베이스 래퍼(wrapper)를 설계하였다. 이 접근방법은 감성정보 데이터베이스에 대한 단순화된 응용 프로그래밍 인터페이스(API: application programming interface)를 제공함으로써 접근코드를 단순화하여 코딩의 생산성을 높이고, 데이터베이스 접근을 일관성 있게 유지함으로써 프로그램의 오류발생 가능성을 감소시켰다. 또한 데이터베이스 접근을 캡슐화함으로써 내부 논리의 변경이나 소프트웨어 버전 향상으로 JDBC API가 변경되더라도 영향을 최소화할 수 있을 것이다.

Keyword : human sensibility ergonomics, information system, JDBC, wrapper

### 1. 서론

감성공학이 G7 과제의 하나로 선정되면서 시작된 감성기반기술개발 연구과제들은 지금까지 많은 감성측정 및 평가 방법들이 시도되었고, 그 결과로 여러 형태의 방대한 데이터가

생산되었다[1]. 그러나 효율적인 관리체계의 부재로 인해 많은 노력과 비용이 투자된 가치 있는 감성평가 데이터가 자료의 공유 및 활용이라는 측면에서 여러 가지 문제점을 내포하고 있는 실정이다.

청주대학교에서는 한국표준과학연구원과 협

력하여 주거/사무환경에서 인간의 감성을 지표화하는 6개 분야의 연구과제에서 생성된 피험자의 생리신호 및 주관적 감성평가 데이터를 통합 관리할 수 있는 감성정보관리시스템(HuSIMS : Human Sensibility Information Management System)을 개발하고 있다. 이 시스템은 기본적인 구조로 2계층 클라이언트/서버 구조를 사용하고 있고 데이터베이스 서버로는 효율적인 데이터 저장 및 처리를 위해 MS SQL Server 7.0을 사용하고 있다. 또한 시스템의 이식성을 높이기 위해 데이터베이스 미들웨어(middleware)인 JDBC(Java DataBase Connectivity)를 통하여 데이터베이스를 접근하고 있다.

JDBC API는 Java 프로그램이 관계형 데이터베이스관리시스템(이하 RDBMS라고 함)에 접근할 수 있는 강력하고 포괄적인 인터페이스이다. RDBMS는 사용자 데이터를 저장 관리하는데 필요한 모든 기능을 지원하는 강력하고 복잡한 시스템이기 때문에, JDBC도 역시 강력한 기능을 지원하는 인터페이스를 포함한다. 그러나, 강력한 기능을 지원하는 인터페이스는 응용 프로그래머들에게 불필요한 복잡함을 제공하여 프로그래밍의 생산성을 떨어뜨릴 뿐만 아니라 응용 프로그램들 간에 서로 다른 데이터 접근방법을 사용하여 데이터베이스 접근에 대한 일관성을 떨어뜨릴 수 있다.

그래서, 우리는 이러한 문제점을 해결하기 위해 JDBC API의 일부분만을 사용하는 단순한 추상계층(abstraction layer)인 데이터베이스 래퍼를 설계하였다. 이 계층은 JDBC 앞쪽에 위치하며, 감성정보관리시스템에서 발생하는 데이터 처리에 대한 요구사항을 기반으로 하여 감성정보 데이터베이스에 대한 접근을 하나의 컴포넌트에 집중시킨 것이다. 따라서 데이터베이스 접근 인터페이스의 불필요한 복잡함을 피할 수 있으며, 그 결과로 프로그래밍의 생산성을 향상시킬 수 있다. 또한 서로 다른 응용 프로그램에서 데이터베이스를 접근할 때 일관성을 유지할 수 있고, 데이터베이스 접근에 대한 내부 논리의 변경이나 미들웨어 버전의 향상으로 JDBC API가 변경되더라도 영

향을 최소화할 수 있다.

## 2. 관련 연구

### 2.1 클라이언트/서버 구조

90년대 중반이후 기업들은 2계층 구조를 갖는 클라이언트/서버 시스템으로 수많은 정보 시스템을 구축해 왔다. 2계층 구조는 클라이언트 계층에 표현논리(presentation logic)와 응용논리(application logic)를, 그리고 서버 계층에 데이터베이스 서버를 배치한다. 그러나 이와 같이 표현논리와 응용논리를 모두 클라이언트에 배치함으로써 "Fat" 클라이언트가 가지는 시스템 관리상의 여러 문제점을 발생시켰다.

2계층 구조가 가지는 문제점들을 해결하기 위해 최근에 많은 개발 그룹들이 3계층 구조로 전환하고 있다. 3계층 구조는 클라이언트 계층에 표현논리를, 중간 계층에 응용논리를, 서버 계층에 데이터베이스 서버를 배치하여 표현논리와 응용논리의 분리, 응용논리의 중앙 집중식 관리, 클라이언트/서버 간에 균형적인 작업분담(load balance), 그리고 확장성(scalability) 제공 등의 장점을 가진다[2,3].

그러나, HTML기반의 3계층 클라이언트/서버 구조는 정교한 GUI(Graphical User Interface)를 제공할 수 없다는 문제점을 포함한다. Java 개발환경에서 Java Applet을 사용하면 이러한 GUI 지원이 가능하지만, 실행할 때마다 서버로부터 다운로드 되어야 하기 때문에 추가적인 시간이 소요되는 또다른 문제점이 나타난다.

감성정보관리시스템의 경우는 일반적인 기업용 응용 시스템과 달리 예상되는 사용자 수가 적고, 감성평가 실험이 종료되어 관리 시스템의 개발이 완료되면 새로운 버전의 소프트웨어 배포문제 등이 발생할 가능성이 적을 것으로 예상된다. 그러므로, 사용자의 편의성을 높이기 위해 정교한 GUI를 제공할 수 있도록 2계층 클라이언트/서버 구조로 감성정보관리 시스템을 설계하였다.

## 2.2 래퍼

많은 소프트웨어 개발환경의 경우에 응용 프로그래밍 인터페이스가 매우 광범위하고 복잡한 구조를 갖기 때문에, 응용 프로그래머에게 불필요한 복잡함을 제공할 수 있다. 따라서 이러한 인터페이스에서 실제 필요한 부분만을 추출하여 보다 간략화된 인터페이스로 집중시킴으로써 프로그래밍의 생산성을 향상시킬 수 있다. 이러한 기술을 래퍼라고 하며 [4], 래퍼를 설계할 때는 일반적으로 Facade Design Pattern[5]을 사용한다. 이 기술은 복잡한 응용 프로그래밍 인터페이스에 대해 단순한 인터페이스를 제공하기 위한 경우와 클라이언트들과 구현 클래스들 간에 종속성이 많은 경우, 그리고 응용 프로그래밍 인터페이스를 계층화하고자 하는 경우에 주로 사용된다[6].

## 3. 데이터베이스 래퍼의 설계

### 3.1 감성정보관리시스템의 구조

본 논문에서 제시하는 HuSIMS(Human Sensibility Information Management System)은 다양한 감성평가 데이터의 효율적인 관리

와 정보 공유체계를 구축하기 위한 도구로써, 그림 1에 나타난 것과 같이 높은 수준의 GUI를 지원하는 사용자 인터페이스 부분과 응용논리가 클라이언트 쪽에 포함된 2계층 클라이언트/서버 구조로 설계되었다. 클라이언트와 데이터베이스 서버 간의 통신은 산업계 표준 인터페이스인 JDBC를 사용하고, 이진파일 전송은 소켓(socket)을 사용하므로 인터넷이 연결된 클라이언트에서는 위치에 관계없이 감성정보 데이터베이스에 접근할 수 있다. 그리고 감성정보 데이터베이스에 대한 접근을 단순화시키기 위해 래퍼인 EmJDBCWrapper를 설계하여 JDBC의 앞쪽에 위치시켰다. 구현 환경은 Windows NT 4.0과 MS SQL Server 7.0을 기본으로 하며, 개발언어는 Java를 사용하였다.

### 3.2 데이터베이스 래퍼의 클래스 구조

본 논문에서 제시한 데이터베이스 래퍼는 JDBC API에 대한 래퍼이며, EmJDBCWrapper라고 명명하였다. 그림 2는 UML(Unified Modeling Language)을 사용하여 감성정보관리시스템을 위한 EmJDBCWrapper의 클래스 다이어그램을 표현한 것이다. 클래스 다이어그램은 해당 시스템에 포함되는 클래스들의 정적인

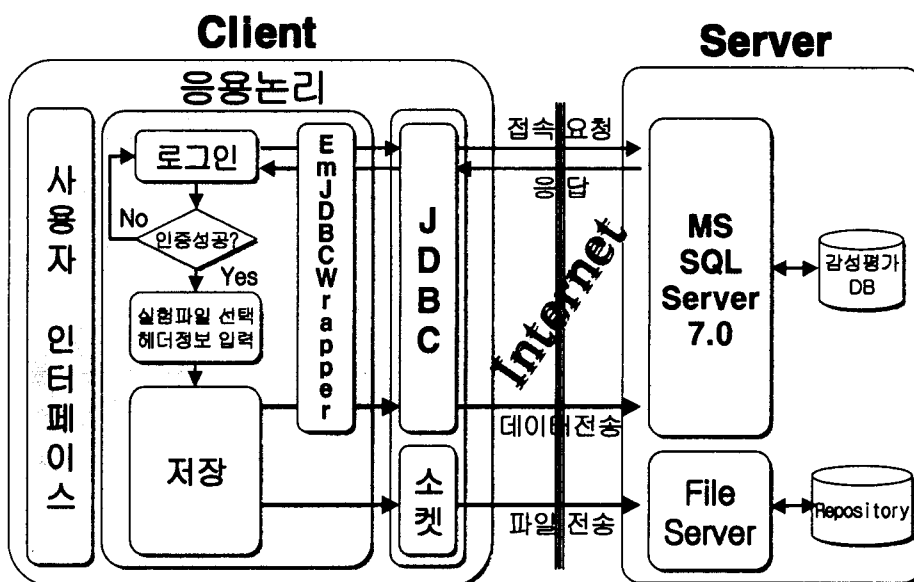


그림 1. 감성정보관리시스템의 기본 구조

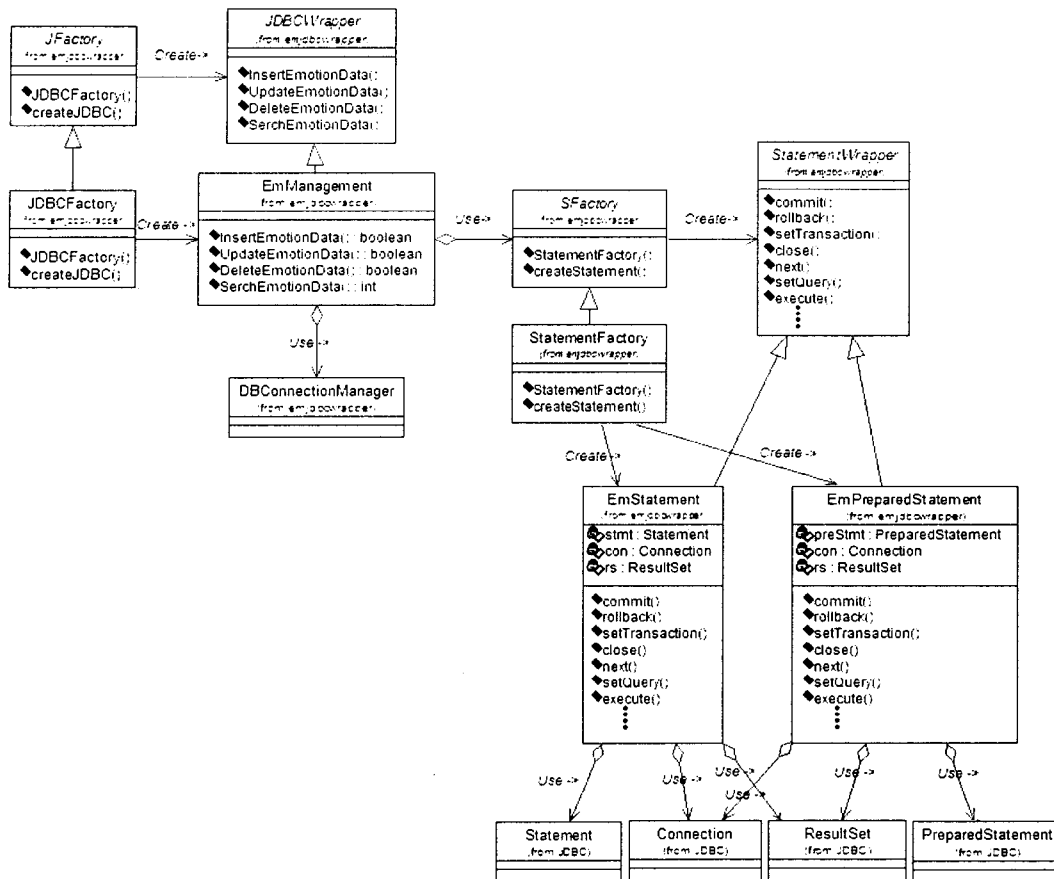


그림 2. EmJDBCWrapper의 클래스 다이어그램

구조를 나타내는 것으로서 클래스들 간의 연관 관계 등을 표현한다. UML은 OMG(Object Management group)에서 정한 객체지향 분석 및 설계를 위한 표준 방법론이다.

EmJDBCWrapper는 확장성을 고려하여 설계 패턴(design pattern) 중 하나인 Factory Pattern을 적용하였다. Factory Pattern은 하나의 객체를 생성할 수 있는 인터페이스를 정의하며, 이때 서브클래스는 어떤 클래스가 실체화(instantiation)될지를 결정한다. Factory Pattern은 하나의 클래스가 자신의 서브클래스로 실체화를 위임(defer)할 수 있도록 한다.

그림 2의 클래스 다이어그램은 4개의 추상 클래스인 JFactory, JDBCWrapper, SFactory, StatementWrapper와 6개의 일반 클래스인 JDBCFactory, EmManagement, DBConnection\_

Manager, StatementFactory, EmStatement, EmPreparedStatement를 포함한다. 이 중에서 JDBCFactory와 StatementFactory 클래스가 Factory Design Pattern을 적용하기 위해 설계된 것이다. 그리고 EmManagement 클래스에서는 데이터베이스 연결 저장소(database pool)와 관련된 응용 논리가 구현되었고, EmStatement, EmPreparedStatement 클래스는 데이터베이스 접근 논리가 구현된 클래스로서, 데이터베이스 트랜잭션 선언 등의 상세함이 캡슐화되어 있다.

### 3.3 데이터베이스 레퍼의 순차 다이어그램

그림 3은 EmJDBCWrapper에 대한 UML의 순차 다이어그램(sequence diagram)을 표현한 것이다. 순차 다이어그램은 여러 객체들 간에

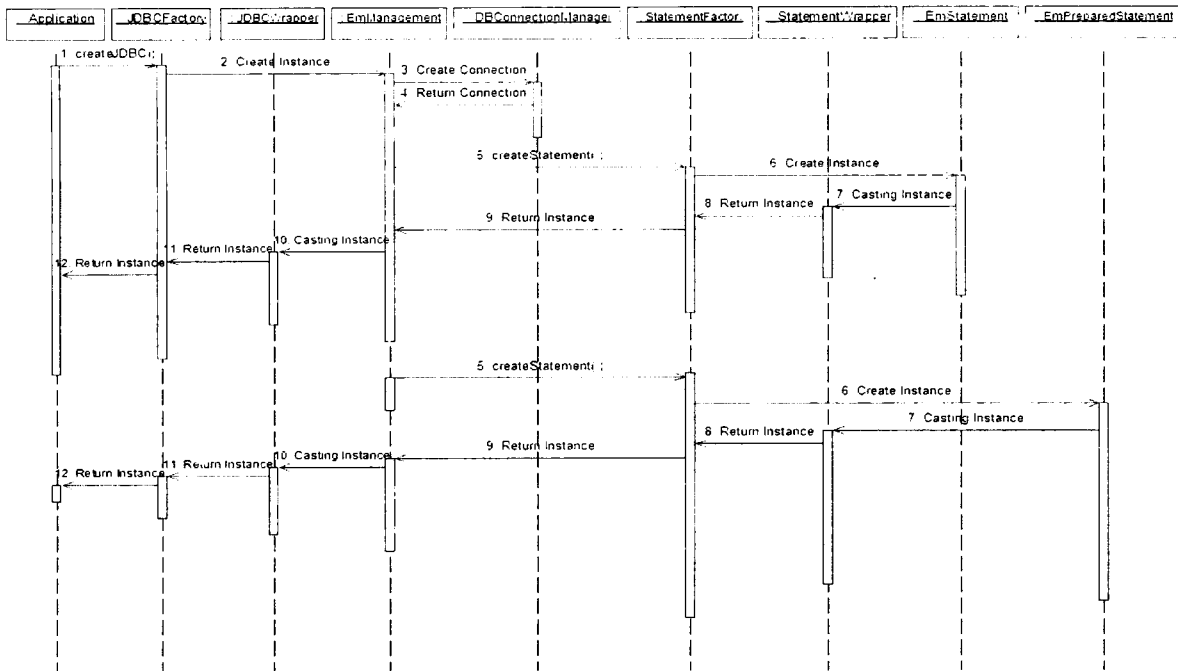


그림 3. EmJDBCWrapper의 순차 다이어그램

이루어지는 동적인 협력사항을 표현하기 위한 것으로 객체들 간에 메시지 전송순서를 명확하게 알 수 있다. 그림 3의 상단부에는 객체들을 나타낸 것이며, 시간의 흐름은 아래 방향으로 진행하고 시간의 흐름에 따라 번호 순서대로 객체들 간에 메시지의 전송을 나타낸 것이다.

#### 4. 결론

본 논문은 주거/사무환경에서 진행된 감성기반기술개발 연구과제에서 생성된 피험자의 생리신호 및 주관적 감성평가 데이터를 효율적으로 관리하기 위한 감성정보관리시스템(HuSIMS : Human Sensibility Information Management System)의 설계 중에서 데이터베이스 레퍼에 관한 연구결과이다. 감성정보관리시스템은 2계층의 클라이언트/서버 구조를 가지며, 시스템의 이식성과 확장성을 높이기 위해 데이터베이스 미들웨어로써 JDBC를 사

용하였다. 그러나 JDBC API는 데이터베이스 관리시스템의 강력하고 포괄적인 기능을 모두 포함하기 때문에, 우리는 감성정보관리시스템의 측면에서 불필요한 복잡함을 제거하기 위해 JDBC API의 일부분만을 포함하는 추상계층(abstraction layer)인 EmJDBCWrapper를 설계하였다. 이 계층은 JDBC 앞쪽(front-end)에 위치하며, 감성정보 데이터베이스 접근에 대한 요구사항을 모두 반영하였다. 따라서 감성정보 데이터베이스를 접근하기 위한 인터페이스의 불필요한 복잡함을 피할 수 있으며, 그 결과로 프로그래밍의 생산성을 향상시킬 수 있다. 또한 서로 다른 응용 프로그램에서 동일한 형태의 데이터를 접근할 때 데이터베이스 접근 논리(access logic)의 일관성을 유지할 수 있고, 데이터베이스 접근에 대한 내부 논리의 변경이나 미들웨어 버전의 향상으로 JDBC API가 변경되더라도 영향을 최소화할 수 있을 것이다.

※본 연구는 G-7 감성공학기반기술개발 사업에 의해 지원되었음(2000-J-ES-02-A-01).

## 참고문헌

- [1] 이윤희, 장현호, 김영윤, 고희동, 김현택, “감성 공학 데이터베이스 시스템의 설계 및 구현”, 한국감성과학회 2001 춘계 학술대회 논문집, 2001.5.
- [2] Marc A. Mnich, Multi Tier Architectures for Database Connectivity, JavaExchange.com, White Paper, Jan. 5, 1998.
- [3] 구홍서, “Java를 기반으로 한 웹 데이터베이스 응용을 위한 프레임워크”, 정보기술과 데이터베이스 저널, 한국데이터베이스학회, 제7권, 제2호, 2000.12.
- [4] 디자인 패턴의 적용, [http://www.jstorm.pe.kr/BBS/files/colJG/DesignPattern2000-09\\_10-forjstorm.pdf](http://www.jstorm.pe.kr/BBS/files/colJG/DesignPattern2000-09_10-forjstorm.pdf).
- [5] Erich Gamma and et al., Design Patterns Elements of Reusable Object-Oriented Software, Addison-Wesley, 1995.
- [6] 샌프란시스코 패턴, <http://www.sisait.co.kr/199907/solution/182.html>.
- [7] 구홍서, “WWW과 데이터베이스 연동기술의 조사분석”, 한국정보과학회지, 제18권, 제4호, 2000.4.
- [8] Fernando G. Guerrero, Carlos Eduardo Rojas, SQL Server 2000 Programming: By Example, QUE, 2001.
- [9] J. Rowe, Building Internet Database Servers with CGI, New Riders, 1996.
- [10] Ken North, “Building Web Databases: Tools and Techniques for Web Database Developers”, Web Techniques, Sep. 1996.