

# UML 기반 최적화된 J2EE 디자인 패턴 모델링 (Optimized Modeling Design of J2EE Design attern Based on UML)

김종경(Jong-Gyung Kim)<sup>1)</sup>, 이돈양(Donn-Yang Lee)<sup>2)</sup>

## 요 약

최근에 널리 이용되고 있는 객체지향 분석 및 설계 방법에서 새롭게 EJB 기반의 컴포넌트 개발이 활성화되고 있다. 일반적으로 EJB 기반 빈 생성에서는 데이터베이스를 이용한 영속적인 데이터를 사용하는 경우가 많다. 본 논문에서는 이런 환경에서 다양한 DBMS가 이용되고 있어 각각의 엔티티 빈에 종속되어 운용되고 있는 DBMS 연결을 DAO 패턴을 적용하여 분리하였고, UML을 이용하여 객체지향 설계 및 분석에서 표준화되고 정형화된 클래스를 생성할 수 있도록 최적화된 모델링을 하였다.

## Abstract

Component development of EJB base is activated newly been using widely object oriented analysis and design method recently. Usually, case that use permanent data that use database in EJB base bean creation is much. Separated DBMS connection that various DBMS is applied been subordinate to each entity bean used in the environment in this paper applying DAO pattern, and is normalized object oriented design and analysis because using UML and standardization so that can create done class optimized modelling do .

---

1) 정회원 : 인천전문대학 컴퓨터정보과 겸임교수

2) 정회원 : 경인여자대학 전산정보과 겸임교수

## 1. 서론

정보통신산업의 발전과 더불어 소프트웨어 산업의 발전에도 많은 변화가 일어나고 있다. 특히 소프트웨어의 고비용에 대한 비용절감에 지속적인 연구가 이루어지고 있다[1]. 그리고 최근에 널리 이용되고 있는 객체지향 분석 및 설계 방법에서 새롭게 EJB 기반의 컴포넌트 개발이 활성화되고 있다. 일반적으로 EJB 기반 빈 생성에서는 데이터베이스를 이용한 영속적인 데이터를 사용하는 경우가 많다. 본 논문에서는 이런 환경에서 다양한 DBMS가 이용되고 있어 각각의 엔티티 빈에 종속되어 운용되고 있는 DBMS 연결을 DAO 패턴을 적용하여 분리하였다. 아울러 설계정보를 표준화하기 위해 OMG의 표준 모델링언어인 UML을 이용하여 최적의 J2EE 기반의 디자인패턴을 설계하였다.

## 2. 연구배경

### 2.1 UML(Unified Modeling Language)

UML은 객체지향 분석(analysis)과 설계(design)를 위한 모델링언어이다. UML은 Booch, Rumbaugh(OMT), Jacobson등의 객체지향 방법에 관한 전문가들이 내어 놓은 방법론의 통합이라고 볼 수 있다[2]. UML의 구성은 그림1 에서와 같이 9개의 다이어그램으로 되어있다.

UseCase Diagram
Class Diagram
Object Diagram
Sequence Diagram
Collaboration
Package Diagram
State Diagram
Activity Diagram
Deployment Diagram

그림 1. UML 다이어그램구성

그리고 본 논문에서는 UML의 클래스 다이어그램을 이용하여 DAO 패턴의 객체들을 표현하고 있다.

### 2.2 디자인 패턴

디자인 패턴은 객체지향 소프트웨어를 설계하는데 있어서 반복적으로 발생하는 설계상의 문제들을 패턴으로 등록하여, 설계자들이 어떤 한 설계 문제에 대해서 다시 해결책을 모색하는 과정을 거치지 않고, 바로 디자인 패턴을 선택하여 설계에 적용할 수 있도록 한 것이다 [3]. 1990년 초 Erich Gamma와 Richard Helm은 패턴을 여러 가지로 분류하여 그룹화하고, 약속된 일정한 형태로 표현하였다[4][5]. 당신은 많은 객체 지향 시스템에서 클래스들과의 견교환하는 객체들이 순환되는 패턴들을 찾을 것이다. 이들 패턴들은 특정한 설계문제들을 해결하며 보다 유연하고, 우아하고, 그리고 궁극적으로 재사용 가능한 객체지향 설계를 만들어준다[6]. 이들은 설계자가 이전의 경험에 근거해서 성공적인 설계를 재사용하는데 도움을 준다고 했다[1].

### 2.3 J2EE DAO 디자인 패턴

J2EE 패턴은 15개의 패턴들로 구성되어 있다. 이는 상위부분과 하위부분으로 나뉘어지고 있다. 본 논문에서 적용되고 있는 DAO 패턴은 하위부분에 해당하며 이 하위부분에 해당되는 패턴들은 Business Logic과 저장소 및 데이터 전달에 관한 것을 담당하고 있다[7]. 특히 이중 DAO 패턴은 Data Source로부터 데이터를 조회하거나 사용할 수 있도록하는 추상적인 방법으로 사용되기도 있다.

DAO 패턴의 사용자는 데이터가 Oracle database, Microsoft SQL Server, LDAP Server 또는 Web Server에서 조회되는 지를 알 수 없다. 어플리케이션이 Data Source에 직접적으로 접근하지 않는 사실에 근거해 DAO 패턴은 유동적으로 제공되고 있다. 대신 중간 정도 부분에서 Data Source에 접근하는데 있어서 DAO를 생성하고 사용할 수 있다

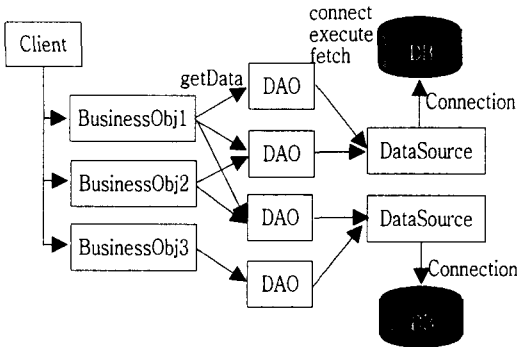


그림 140. DAO Pattern 구조

### 3. UML 기반 모델링

객체지향언어 프로그래밍을 작성하는데 가장 중요한 것중 하나는 객체지향의 분석과 설계에 해당된다. 지금 까지의 소프트웨어 개발 방법으로는 구조화 기법이 대표적으로 사용되어 기능적인 부분을 중요시 처리하는 방법으로 접근을 하였다. 그러나 소프트웨어가 대형화되고 복잡해지는 현재의 추세에서는 이를 적용하는데 많은 무리들이 발생되고 있다. 본 논문에서는 이러한 약점을 보완할 수 있는 방법으로 객체지향 분석과 설계방법을 적용하였다. 그리고 UML을 이용하여 객체지향 분석 및 설계를 작성하는 모델을 표현하는 notation을 설계하였다.

#### 3.1 DAO 디자인 패턴 모델링

일반적으로 DAO 패턴은 Data Source로부터 데이터를 이용하게하는 추상적인 방법으로 사용된다. 본 논문에서는 이 패턴을 이용하여 각각 다른 DBMS 환경에서 운용되고 있는 데이터를 클라이언트에서 직접적인 DB의 환경의 변화 없이 편리하게 이용할 수 있도록 하였다. 즉, 엔티티 빈에서 Data Source와 연결되는 부분을 DAO 패턴을 이용하여 Abstract 부분과 Concrete 부분으로 분리하고 있다. 이 방법으로 UML 이용되었다. 그림3은 쇼핑몰을 구축하는데 필요한 기본적인 클래스에 대하여 사례를 들어 전체적인 구성 클래스 다이어그램을 모델링하였다. 기본적으로 Customer, Product, JangBaguni, Order에 대한 엔티티 빈을 생성

하였다. 여기서는 DAOFactory를 Abstract 클래스로 하고 있고, 4개의 DBMS에 해당하는 CloudscapeDAOFactory, MSSqlDAOFactory, OracleDAOFactory, InformixDAOFactory를 Concrete 클래스로 분리하여 시스템을 설계하였다. 그리고 오퍼레이션들은 인터페이스 객체를 이용하여 구현되도록 하고 있다.

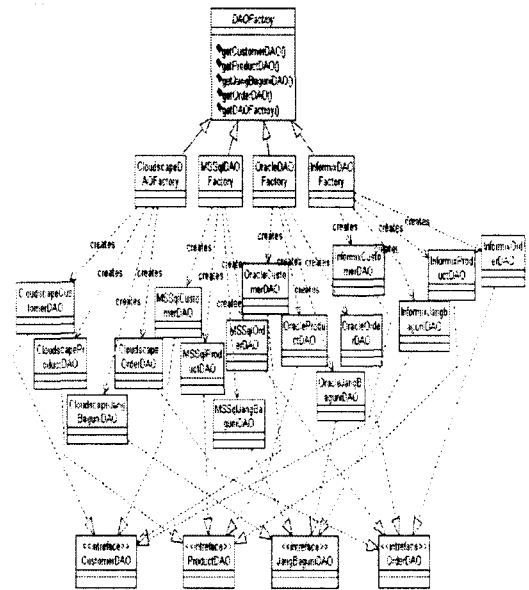


그림 141. 쇼핑몰 클래스 다이어그램

#### 3.2 빈 클래스 자동 생성

객체지향 방법론으로 사용되고 있는 UML의 모델링과 코드의 생성에 대한 방법으로 본 논문에서는 Rational Rose를 이용하였다. 그림3에서는 4개의 각기 다른 DBMS 환경을 분리하고 있으며 각각의 클래스를 상속하거나 인터페이스 하면서 object들을 공유하고 있다. 일반적으로 이 클래스에 대한 프로그램을 작성하는 것은 개발자가 직접하는 것이 일반화되어 있지만 본 논문에서는 클래스의 재사용과 정확한 모델링 방법의 적용, 개발기간의 단축을 최대한으로 갖기 위해 Rational Rose의 Generate Code를 이용하여 각 클래스에 해당하는 java 프로그램을 생성하였으며, 그 외의 DB Connect에 해당되는 부분들은 직접 프로그램

을 작성하였다. 그림4 에서는 MSSql DBMS에 해당되는 부분을 따로 분리하여 세부적으로 클래스 다이어그램을 설계하였다.

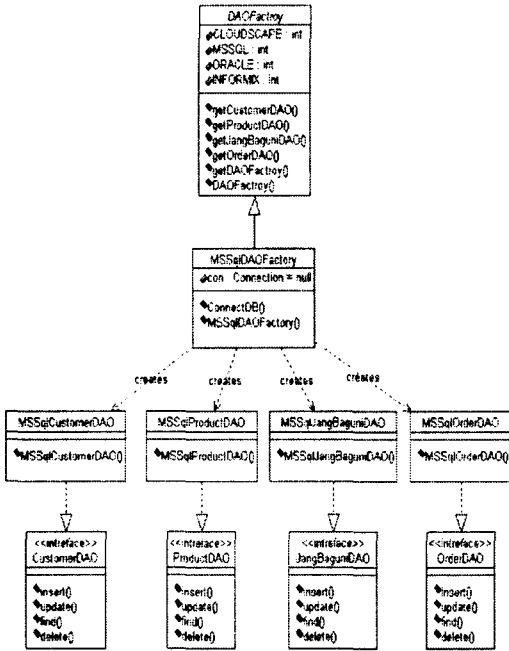


그림 142. MSSqlDAOFactory 클래스 다이어그램

그리고 그림5, 그림6은 DAOFactory 클래스와 MSSqlDAOFactory에 대한 자동생성된 Java 프로그램이다.

```
//Source file: C:\WWW\Jaguar CTS
3.5\www\java\www\classes\www\DAOFactory.java
public abstract class DAOFactory
{
    public int CLOUDSCAPE;
    public int MSSQL;
    public int ORACLE;
    public int INFORMIX;

    /**
     * @roseuid 3F4583B50231 */
    public DAOFactory()
    {
    }

    /**
     * @return CustomerDAO @roseuid 3F4432520140
     */
    public CustomerDAO getCustomerDAO()
    {
        return null;
    }

    /**
     * @return ProductDAO @roseuid
     3F44328502D4 */
    public ProductDAO getProductDAO()
    {
        return null;
    }

    /**
     * @return JangBaguniDAO @roseuid
     3F4432AD0246 */
    public JangBaguniDAO getJangBaguniDAO()
    {
        return null;
    }

    /**
     * @return OrderDAO @roseuid 3F4432CF0303
     */
    public OrderDAO getOrderDAO()
    {
        return null;
    }

    /**
     * @return DAOFactory @roseuid
     3F457FE601AD */
    public DAOFactory getDAOFactory()
    {
        return null;
    }
}
}
```

그림 143. DAOFactory.java 클래스 프로그램

```

//Source      file:      C:WWJaguar      CTS
3.5WWjavaWWclassesWWMSSqlDAOFactory.java
public class MSSqlDAOFactory extends
DAOFactory
{
    public Connection con = null;

    /** @roseuid 3F45877903BE */
    public MSSqlDAOFactory()
    {
    }

    /** @roseuid 3F4586D90309 */
    public void ConnectDB()
    {
    }

    /** @roseuid 3F4587DA00CE */
    public void getCustomerDAO()
    {
    }
}

```

그림 144. MSSqlDAOFactory.java 클래스 프로그래밍

#### 4. 결론

컴퓨터 산업의 발전과 더불어서 소프트웨어 분야도 매우 빠르게 변화되고 있다. 그러나 대형화되고 복잡화되어가는 소프트웨어 개발 분야에서는 새로운 개발방법들이 적용되어왔다. 객체지향기술은 객체와 객체간의 관계를 모델링하는 기술로서 메시지를 이용하여 객체의 동작을 표현하고 있다. 특히 객체지향 모델링의 이점으로는 인간지향 모델링과 부드러운 개발 공정간의 연계성, 재사용의 향상, 다른 기술과의 조화 및 통합이 가능하다. 본 논문에서 적용한 J2EE DAO 패턴에 대한 UML 기반 모델링은 EJB에서의 어플리케이션 개발의 복잡성과 어려움을 줄일 수 있도록하고 있다. 그리고 본 논문에서 적용하고 있는 엔티티 빈에서 Data Source를 이용한 프로그램 개발 방법은 영속적인 데이터의 사용이 많이 이용되고 있다. 일반적으로는 동일한 DBMS 환경에서 데이터베이스의 연동이 많이 이루어지고 있다. 그러나 Data Source의 환경이 DBMS의 환경의 종류에 따라 매번 바뀌어야만 하는 서버의 환경에서는 프로그램 작성의 번거로움과 시

스템의 퍼포먼스 및 유지 보수에 대한 저하가 발생할 수 있다. 이를 해결할 수 있는 방법으로 J2EE DAO 디자인 패턴을 적용하면서 본 논문에서는 정확하고 최적화된 방법으로 UML을 이용하여 시스템을 모델링하였다.

#### 참고 문헌

- [1] Software Engineering : A practitioner's approach, McGraw-Hill International Edition, 1997
- [2] Rational Software Corporation, "UML Nation Guide Version 1.3", 1997
- [3] Dirk Riehle, "Composite Design patterns", OOPSLA '97, 11.218-228
- [4] Mark Grand, "Patterns in Java", WILEY, 1998
- [5] E.Gamma, R.Helm, R.Johbson, and J.Vlissides, "Design Pattern : Elements of Reuseable Object-Oriented Software", Addison-Weley, 1995
- [6] Floyd Marinescu, EJB Design Pattern, Wiley, 2002
- [7]<http://java.sun.com/blueprints/Corej2eepatterns>

김종경



1980 .3 - 1984. 4 : 광운대학  
교 전자계산소 주임

1984. 4 - 1999. 7 : 경희대학  
교 정보처리처 팀장

1999. 3 - 2003. 7 : 인천대학  
교, 경기대학교 강사

1999. 12 - 현재 : (주) 원포텍 이사

2003 . 3 - 현재 : 시립 인천 전문대학 컴퓨  
터정보과 겸임교수

1990. 2 : 경희대학교 산업정보대학원 전자  
계산공학 (공학석사)

1999. 7 : 아주 대학교 일반대학원 컴퓨터공  
학과 (박사수료)

관심분야 : 멀티미디어 응용 및 시스템구조, 소  
프트웨어 시스템 디자인 운영체제

이돈양



1995 .3 - 2002. 8 : 대한상공회  
의소 교사

2002. 9 - 현재 : 경희대학교,  
인천전문대 강사

2003. 3 - 현재 : 경인여자대  
학 전산정보과 겸임교수

1987. 2 : 대구대학교 통계학과 졸업

1993. 2 : 경희대학교 산업정보대학원 전자계  
산공학(공학석사)

2002. 2 : 경희대학교 일반대학원 전자계산공  
학과(박사수료)

관심분야 : EJB, XML, 디자인 패턴, 저장소  
설계, OOD