

CBD기법을 이용한 EJB기반의 주문관리 시스템의 설계 및 구현

이종수, 황두성, 김미성, 권오현
동명정보대학교 컴퓨터공학과

e-mail : {kama1435, dove3118, gugulove02@hotmail.com , ohkwon@tmic.tit.ac.kr

Design and Implementation of CBD Based Orders Management System under the EJB Environments

JongSoo Lee , DooSung Hwan , MiSung Kim, Oh Hyun Kwon

Dept. of Computer Engineering, TongMyong Universty of Information Technology

요 약

소프트웨어 위기 극복으로부터 시작한 소프트웨어 공학이 최근 가장 경쟁력 있는 소프트웨어 개발 기술로 지목하고 있는 컴포넌트 기반 개발(CBD; Component Based Development) 기술은 높은 생산성과 고품질의 소프트웨어의 개발을 가능하게 되었다.

소프트웨어 Component 를 조립해 새로운 Application 을 만들 수가 있어 개발기간을 단축할 수 있으며 기존의 Component 를 재사용할 수 있어 생산성과 경제성을 높일 수 있다. 본 논문에서는 특정물품의 주문 관리시스템을 대상으로 하여 컴포넌트기법을 활용 분석,설계, 구현을 하였다.

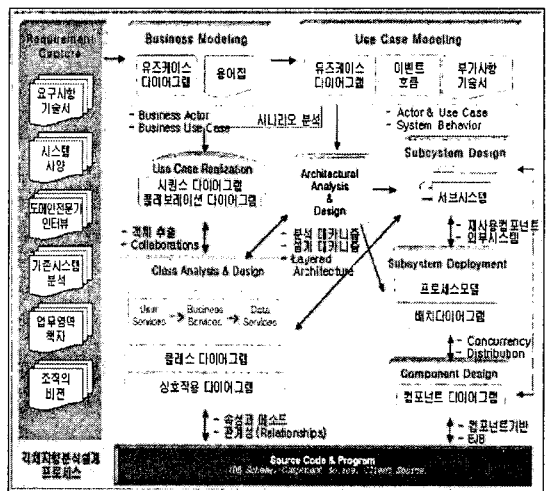
분석과 설계는 UML 을 활용하였으며, 구현은 EJB 를 이용하여 Component 를 생성하였다.

1. 서론

소프트웨어 위기 극복으로부터 시작한 소프트웨어 공학이 최근 가장 경쟁력 있는 소프트웨어 개발 기술로 지목하고 있는 Component 기반 개발(CBD; Component Based Development) 기술은 높은 생산성과 고품질의 소프트웨어의 개발을 가능하게 할 것으로 기대하고 있다.

본 논문에서는 특정품목의 주문관리를 위해 소요되는 컴포넌트들의 개발에 주안점을 두었으며, CBD개발의 분석 및 설계단계에서는 UML(Unified Modeling Language), 구현단계에서는 EJB(Enterprise java Beans)를 사용하였다.

UML을 이용한 분석 및 설계과정은 <그림 1>과 같다.

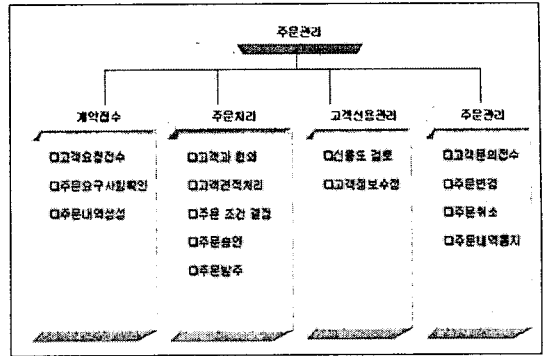


<그림 1> UML기반 분석/설계 프로세스

그림에서 보는것처럼 Usecase 다이어그램을 이용하여 요구분석 결과물을 산출하며, Collaboration 다이어그램, Class 다이어그램등을 통해 설정된 객체와 오퍼레이션을 튜닝한후 각 클래스간의 상속, 의존관계 및 인스턴스들의 수량등에 관해 논리적 표현이 되도록 설계된다.

이렇게 논리적으로 설계된 모델들은 컴포넌트 다이어그램에 의해 물리적으로 표현되게 되며, 실제 구현 환경으로 연결되게 된다.

EJB는 자바 분산환경에서 컴포넌트 어플리케이션개발을 위한 목적으로 자주 사용되는데, 트랜잭션처리의 안정성과, 보안성에 뛰어나며, 3-Tier를 지원 하는 관계로, 본 연구의 목적과 부합되어 이를 적용하였다.



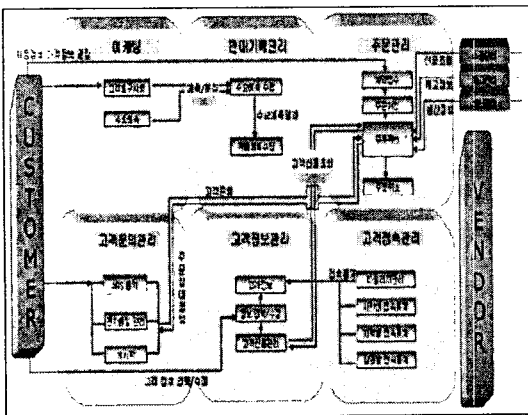
<그림 3> 주문관리 내부 프로세스

주문관리 내부 프로세스는 <그림 3>과 같다.

2. 주문관리처리 시스템 분석 및 설계

2.1 주문관리시스템의 분석

주문관리시스템은 고객과 관리자측면을 양대축으로 하여 마케팅과정의 일환으로 발생하는 업무체계이며 이들간의 상관관계를 그림으로 표시하면 <그림 2>과 같이 나타낼 수 있다



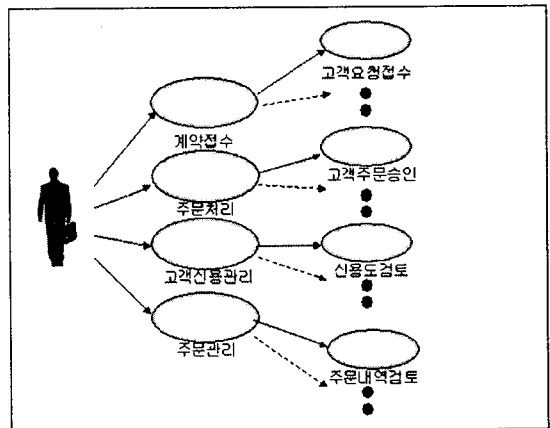
<그림 2> Customer와 Vendor의 상관관계도

2.2 시스템 설계

2.2.1 Use Case Diagram

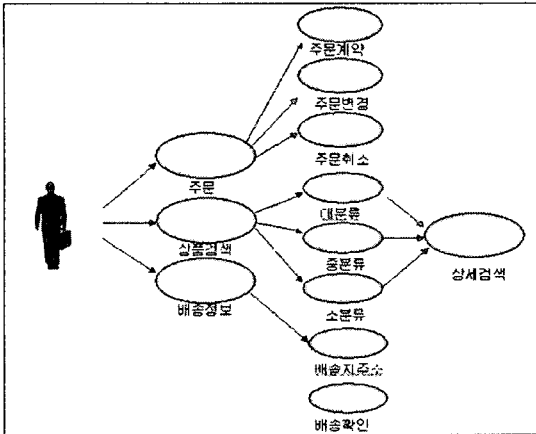
본 논문에서는 주문처리관리시스템의 예로 신발 E-SHOP을 예로 하여, 업무 프로세스를 분석하였다. CASE도구로 래셔널로즈를 사용하여 분석 및 설계를 진행하였다.

업무분석 단계는 크게 2가지로 나누었다. 첫번째로 관리자측면이다 관리자는 각종 상품(신발)을 등록하고, 수량을 추가할 수 있고, 주문리스트를 볼 수 있고, 배송현황 등을 조회 할 있다.



<그림 4> Vendor Use-Case Diagram

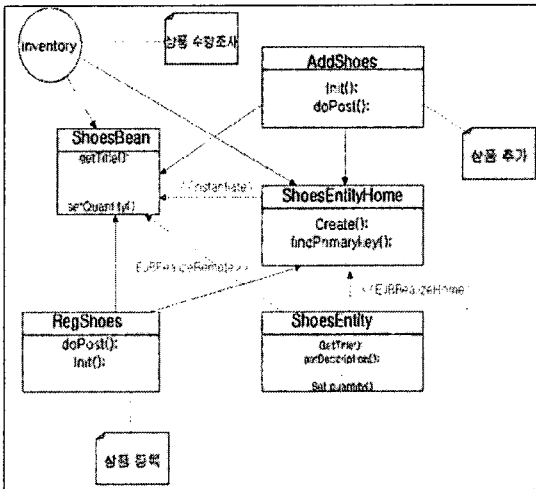
두번째로 고객측면이다. 고객은 상품목록을 볼 수 있고, 상품을 구입할수 있으며, 상세 상품내용을 볼 수 있고, 배송상태를 알 수가 있다.



<그림 5> Customer Use-Case Diagram

2.2.2 클래스 설계

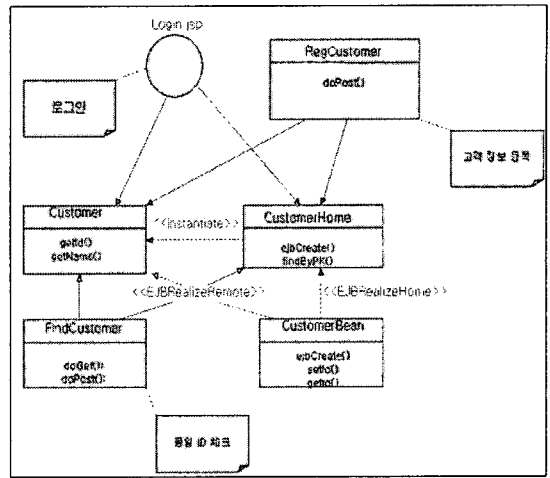
먼저 관리자 측면에서 본 클래스 유형은 ShoesBean 클래스, 상품정보를 입력하기위한 RegShoes 클래스, 등록된 모든 상품의 정보와 수량을 파악하기 위한 inventory.jsp, 등록된 상품의 수량을 증가 시키기 위한 AddShoes 클래스 등이 있으며 <그림 6>와 같이 설계되어 있다.



<그림 6> Vendor 관련 Class Diagram

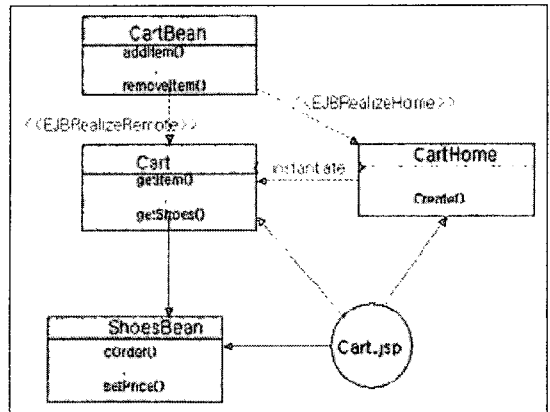
고객측면은 고객정보를 나타내는 CustomerBean과, 자신의 정보를 등록하기 위한 RegCustom클래스, ID사용 여부를 알아보기 위한 FindCustomer, 로그인 하기위한

login.jsp 등, <그림 7>와 같이 설계되어 있다.



<그림 7> Customer 관련 Class Diagram

그리고 고객이 선택한 상품은 CartBean class (세션빈) 을이용해서 표현되고, 상품 정보를 표현하는 ShopBean class등으로 <그림 8>과 같이 설계되어 있다



<그림 8> Cart 관련 Class Diagram

3. 구현

본 논문에서는 위와 같은 분석 및 설계 결과물을 토대로 하여 비즈니스 로직 부분과 인터페이스 부분을 분리하여 구현하였다.

3.1 주문처리관리 시스템 구성파일

시스템 구성파일을 살펴보면 크게 JSP파일, Beans 컴포넌트 파일, EJB컴포넌트 파일,HTML파일, 서블릿 클래스 파일, 공통으로 사용되는 클래스파일, 예외처리를 한 예외클래스 파일로 구성되어져 있다. Beans컴포넌트 파일에는 LoginBean,Rota,TorBean, ShoesBean, CartBean가있다.

EJB컴포넌트 파일로는 Shoes.java, ShoesHome.java, ShoesBean.java, Cart.java, CartHome.java, CartBean.java, ShopBean등이 있다.

3.2 주요 Bean(컴포넌트) 로직

주문처리 관리 시스템에서 핵심 부분인 CartBean 컴포넌트의 소스코드를 살펴보면, 아래와 같다.

```
public class CartBean extends SessionAdapter {
private ShoesBean finalItem;
private Hashtable items = new Hashtable();
private String param="";
.....
```

Hashtable 객체인 items 와 ShoesBean class객체인 finalItem, String 형변수인 param 등이 특정 사용자의 상태정보를 나타내는 멤버 변수들이다. Hashtable 객체인 items가 쇼핑카드 역할을 하고, ShoesBean객체 finalItem은 쇼핑카드에 저장된 모든 아이템을 확인하거나 특정 아이템을 추가한다. Param변수는 마지막으로 조회했던 title과 category정보를 저장하기 위해사용한다. Hashtable 객체는 모든 데이터 Key값과 Value값으로 저장한다. 동일한 Key값을 갖는 여러 개의 Value값을 가질 수가 없기 때문에, addItem()메소드 내에서는 쇼핑카드에 이미 동일한 Key값을 갖는 ShoesBean객체가 존재하는 경우에는 해당 객체의 Quantity(수량)만을 추가 하도록 하였다.

```
Public void addItem(ShoesBean shoes) throws InalidUI
Exception {
```

```
If(items.containsKey(shoes.getID( ))) {
이미 아이템이 쇼핑카드에 존재하는경우
ShoesBean item = (ShoesBean)items.get(shoes.getID( );
Long quantity = items.getQuantity() + shoes.getQuantity();
Items.setQuantity(quantity);
} else { 새로운 아이템을 쇼핑카드에 저장하는경우
finalItem = shoes;
item.put(shoes.getID( ), Shoes);
}
} else { 수량이 음수로 전달 되거나,입력오류인경우
throw new InalidUIException("");
} }
```

cOrder() 메소드에서는 updateDB()나 updateODB()메소드 수행도중 예외가 발생하는 경우 컨테이너가 트랜잭션을 콜백하도록 시스템 예외인 RemoteException을 발생시켰다.

updateDB() 메소드 내에서는 ShoesBean객체를 통해 사용자가 선택한 해당 상품 아이템 수량을 확인하고, Shoes엔터프라이즈 빈의 EJB객체 레퍼런스를 구하여 재고 수량(REMAIN)을 갱신한다.

```
Public void cOrder( ) throws RemoteException {
Try {
Enumeration shoes = cart.getShoes();
While(shoes.hasMoreElements()) {
ShoesBean shoes = (ShoesBean)shoes.nextElement ( );
UpdateDB(shoes);
UpdateOLDB(shoes);
}
catch(FinderException e) {
throw new RemoteException(e.getMessage());
} catch(SQLException e) {
throw new RemoteException(e.getMessage());
} }
```

4. 결론 및 향후 연구방향

본 논문에서는 CBD기법을 이용하여 EJB 기반의 컴포넌트를 분석 및 설계하여 구현해 으로서 소프트웨어 공학의 중요성을 인지하고, 기존의 소프트웨어 개발방법론에 비해 비즈니스로직 부분과 인터페이스 부분이 분리된 주문처리관리 프로그램의 재사용성과 효율성을 조금이나마 알아보았다..

향후 연구계획으로는 고객관리 시스템 전부분을(고객창출,마케팅,고객접속관리..) 컴포넌트화 하여 하나의 완전한 시스템을 구축 하여 CBD기법의 효율성 및 재사용성을 검증할 예정이다.

[참고문헌]

- [1] Budi Kurniawan. Java for the Web with Servlets, JSP, and EJB
- [2] Ed Roman. Mastering Enterprise JavaBeans (2nd Edition).
- [3] 지영수저. Rational Rose2000. 홍릉과학출판사. 2001
- [4] 김성호. Enterprise Java Beans. 대림. 2001
- [5] 조완수저. UML객체지향 분석설계. 홍릉과학출판사. 2000
- [6] John Daniels 외 저/김경주 외 공역.Uml Components. 인터비전. 2001
- [7] 문창수 외 공저. EJB and Weblogic. 가메 출판사 2002