

EJB Architecture를 기반으로 하는 EC Merchant Server의 상품관리 컴포넌트의 설계 및 구현

손민정*, 박은주**, 이종호*
*공주대학교 전자상거래학과
**세림정보기술(주)
e-mail:mj0717ex@hotmail.com

A Design and Embodiment of Product Management Component of EC Merchant Server based on EJB Architecture

Min-Joung Son*, Eun-Ju Park**, Jong-Ho Lee***
*Dept of Electronic Commerce, Kong-Ju University
**Selim Technologies, Inc
***Dept of Business Information Education, Kong-Ju University

요 약

인터넷의 급격한 발달과 사용자의 증가로 인하여 클라이언트와 서버로 구성된 기존의 웹 환경에서는 EC(Electronic Commerce)서비스 제공이 원활하지 못하였다. 시스템의 안정성 및 신뢰성이 미약하게 되면서 이를 보완하기 위한 새로운 환경으로 분산처리 기술의 필요성이 부각되었다. 본 연구는 현재 활용되고 있는 EC Server에서 상품관리에 대한 안정성과 신뢰성 제고 등의 문제에 대한 해결방안으로, 핵심 비즈니스 프로세스를 컴포넌트로 변환 적용하는 내용을 담고 있다.

1. 서론

최근 컴퓨터와 통신의 급격한 발달로 인하여 인터넷 사용이 보다 편리해지고, 사용자들의 급격한 증가 추세로 인하여 관리자가 사용자 요구에 서비스를 제대로 제공하기가 점차 어려워지고 있다. 더불어 새로운 기술변화에 능동적으로 대응하기 위한 노력이 필요하고, 산업사회에 대한 신기술 적용을 통한 생산성 제고와 소프트웨어 재사용 기법의 현실적 모델의 대안으로서 새로운 소프트웨어 개발이 필요하게 되었다.

따라서 본 논문에서는 전자상거래 환경에서 관리자 측면의 상품관리 효율성을 증가시키기 위해 컴포넌트 소프트웨어 기법을 적용하였다. 컴포넌트 기법을 도입함으로써 재사용성과 모듈성의 극대화를 꾀

하고, 코드 최적화 및 개발 시간과 비용 절감을 지향하고자 하였다.

본 논문의 구성은 2장에서 컴포넌트 기술의 특성과 EJB(Enterprise JavaBeans)시스템에 대해 기술하였고, 3장에서는 본 논문이 제안하고자 하는 EJB 아키텍처를 적용한 상품관리 컴포넌트의 설계 및 구현에 대해 기술하였다. 그리고 4장에서는 결론 및 향후과제에 대하여 기술하였다.

2. 관련 연구

2.1 컴포넌트 기술

본 연구에서는 세션 관리나 상태 관리 등의 어려움을 가지고 있던 기존 Web 기반기술의 문제점을 컴

포넌트 방식으로써 해결하였다. Web Application Server를 돕으로써 세션이나 상태관리는 물론 트랜잭션 관리 기능도 보다 쉽도록 했고, 데이터베이스 커넥션 풀링의 향상도 추구했다. 2-Tier방식의 기존 S/W의 구조와 3-Tier방식의 제안모델 구조의 차이점은 (그림 1)과 같다.

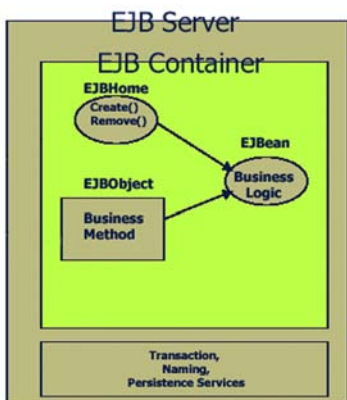


(그림 1) 2-Tier와 3-Tier의 차이점

2.2 EJB 시스템 개요

여러 컴포넌트 기술 중 본 연구에서는 플랫폼에 독립적이고 기존 시스템과 연동이 쉬운 EJB를 채택하여 사용하였다. 본 절에서는 EJB에 대한 자세한 설명을 기술한다.

EJB는 엔터프라이즈 환경에서 애플리케이션 개발에 적합한 J2EE(Java 2 platform Enterprise Edition) 서버 측면의 컴포넌트 기술이다. EJB 시스템은 미들웨어인 애플리케이션 서버, 컨테이너, 그리고 컴포넌트로 구성되어 있다. 서버는 보안, 지속성, 쓰레딩, 트랜잭션 관리 및 컨테이너를 관리하고 컨테이너는 컴포넌트를 관리한다. 그리고 새로운 프로그램 컴포넌트가 추가되거나 변경될 때마다 개별 컴퓨터를 갱신하지 않고도 서버에서 변화를 통제할 수 있는 이점을 제공한다. EJB 시스템의 구조는 (그림 2)와 같다.



(그림 2) EJB 시스템의 구조

EJB는 크게 두 종류로 나뉜다. 클라이언트 코드를 서버 상으로 확장하여 비즈니스 로직을 포함하고 있

는 Session Bean과 데이터베이스 내의 데이터를 표현하기 위해 사용하고 데이터 관련 로직을 포함하고 있는 Entity Bean이 있다. Session Bean은 클라이언트 레퍼런스가 없을 경우 지속적이지 못한 반면 Entity Bean은 한 클라이언트에 종속적이지 않고 영구적 저장장치에 지속성을 가지는 것이 특징이다. Session Bean은 또 상태에 따라 <표 1>과 같이 State Session Bean과 Stateless Session Bean으로 나누어진다.

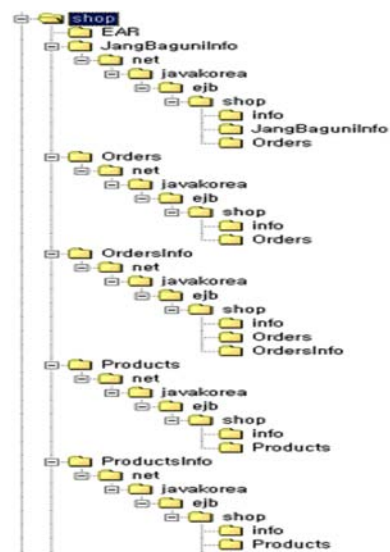
<표 1> 두 가지 Session Bean의 비교

| State Session Bean | Stateless Session Bean |
|--|---|
| <ul style="list-style-type: none"> · 세션 상태 저장 안 됨 · 확장성 · 클라이언트 컨텍스트 관리 안 됨 · 스와핑이 일어나지 않는다. | <ul style="list-style-type: none"> · 세션 상태가 저장 됨 · 클라이언트 수 - Bean Instance 수 · 클라이언트 컨텍스트 관리 됨 · 스와핑이 일어남 |

3. 시스템 설계 및 구현

3.1 시스템 설계

본 작품은 (그림 3)과 같이 다섯 개의 컴포넌트로 구성되어 있다.

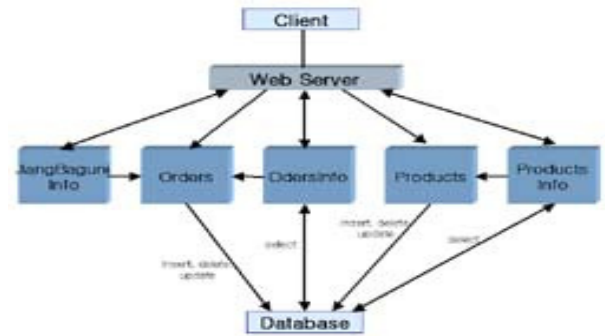


(그림 3) 각 기능별 컴포넌트 구성

각 컴포넌트가 가지는 기능은 다음과 같다.

- JangBaguniInfo(State Session Bean) : 소비자가 선택한 물품을 장바구니에 담고 수량 변경이나 물품 삭제, 장바구니를 비우는 등의 기능을 가진다. 이 컴포넌트는 상태유지 세션빈으로써 소비자가 원하는 물품을 선택하면 백터를 생성하여 해당 상품에 대한 정보를 담고 최종 주문이 이루어지기 전까지 지속적으로 여러 상품의 리스트를 해쉬테이블에 저장한다. 이렇게 함으로써 소비자에게 상품을 최종적으로 한꺼번에 구매할 수 있도록 지원한다.
- Orders(Entity Bean) : 소비자가 주문을 한 물품의 키값에 해당하는 빈 인스턴스 정보의 유무를 검사한 후 중복되지 않도록 빈 인스턴스 정보를 OrdersInfo 객체에 담고 주문정보를 데이터베이스에 넣는다.
- OrdersInfo(Stateless Session Bean) : Orders 모듈과 연결되어 매개변수로 전달받은 키를 가지고 데이터베이스를 검색하여 해당 주문번호의 주문서를 관리한다. 소비자가 주문한 물품에 대한 거래 정보와 배송상태를 업데이트하고 모든 주문 리스트를 가져와 관리자가 전체 주문목록을 볼 수 있게 한다.
- Products(Entity Bean) : 관리자가 카테고리별로 새로운 상품을 등록할 때 해당 카테고리별 키가 중복되지 않도록 테이블에 데이터가 있는지 검사하여 백터값으로 리턴, 데이터베이스에 삽입하는 역할을 하고 추후에 상품정보를 수정하는 기능을 제공한다.
- ProductsInfo(Stateless Session Bean) : 무상태 세션빈으로써 Products 모듈과 연결되어 상품정보를 관리한다. 데이터베이스로부터 상품정보를 가지고 온 후 백터에 저장하여 리턴해 주는 방식으로 소비자로부터 하여금 상품을 탐색할 수 있도록 한다.

이러한 컴포넌트들은 각각의 특성에 따라 서로 연관되어 동작된다. (그림 4)는 컴포넌트들이 웹서버와 데이터베이스 사이에서 어떻게 연관되어 동작하는지, 또 각 컴포넌트 사이에는 어떤 관계가 있는지 나타내 준다.

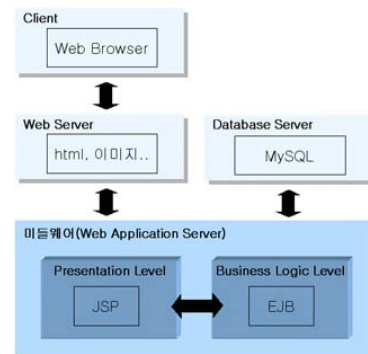


(그림 4) 컴포넌트 설계도

3.2 시스템 구현

본 시스템은 Window2000 Professional/Server 환경에서 제작되었다. 3-Tier 방식의 구조 중 클라이언트측의 웹과 서버측의 데이터베이스를 연결하는 미들웨어인 WAS(Web Application Server)로 J2EE를 사용했다. 데이터베이스로 MySQL을 채택하여 J2SDK1.4.1.01 버전의 JAVA 언어로 시스템을 구현하였고 GUI(Graphic User Interface)는 JSP와 HTML로 개발하였다.

시스템은 (그림 5)에서 보는 바와 같이 기능에 따라 크게 네 부분으로 구성되어 있다. 클라이언트의 요청은 데이터베이스로부터 원하는 정보를 얻기 위해서 웹서버와 미들웨어를 거치게 된다. 웹서버는 클라이언트가 요청한 정보를 받아 WAS에 전달해 주고 그 수행결과를 브라우저에 뿌려주는 역할을 한다. J2EE로 구현한 WAS는 사용자와의 인터페이스 기능을 담당하는 Presentation Level과 실질적인 기능을 담당하는 Business Logic Level로 나누어 볼 수 있다. Presentation Level에서 JSP는 클라이언트의 요청을 받아 EJB객체를 호출하게 되며, Business Logic Level에서 데이터베이스와의 상호작용으로 처리된 결과를 받아 웹서버로 돌려준다.



(그림 5) 시스템 구조

본 시스템이 가지고 있는 기능들을 사용자 측면과 관리자 측면으로 나누어 살펴보면 아래와 같다.

먼저 사용자가 사용하는 기능으로 관리자가 등록한 상품을 검색할 수 있는 기능과 원하는 상품을 구매할 수 있도록 지원하는 장바구니 기능, 주문 기능, 그리고 주문한 상품의 내역이나 배송상태를 확인할 수 있는 주문확인 기능이 있다. (그림 6)과 (그림 7)은 사용자 페이지 중 장바구니를 확인하고 주문서를 작성하는 페이지이다.

| 상품번호 | 상품명 | 제품가격 | 수량 | 수량변경 | 삭제 |
|------|-----|------|----|------|-----------------------------------|
| 1 | 책사 | 1200 | 1 | EA | <input type="button" value="확인"/> |
| 1 | 책스 | 1000 | 1 | EA | <input type="button" value="확인"/> |

총 구매금액 : 2200원

(그림 6) 장바구니를 확인하는 페이지

| 상품번호 | 상품명 | 제품가격 | 수량 |
|------|-----|------|-----|
| 1 | 책사 | 1200 | 1EA |
| 1 | 책스 | 1000 | 1EA |

총 구매금액 : 2200원

주문 정보

| | |
|-----------|-------------------|
| 받으시는 분 | 홍길동 |
| 주소 | 대전광역시 |
| 전화번호 | 111 - 1111 - 1111 |
| 배송시 요청사항 | 안전한 배송 바랍니다. |
| 주문자 성명 | 홍길동 |
| 주문자 이메일주소 | hong@mail.com |
| 주문자 전화번호 | 111 - 1111 - 1111 |

(그림 7) 주문서를 작성하는 페이지

관리자가 사용하는 기능은 크게 둘로 나누어 볼 수 있다. 먼저 상품관리 관련 기능으로 (그림 8)과 같이 각 상품을 대, 중, 소의 범위로 나누어 카테고리 별로 분류해 등록하고 삭제할 수 있도록 했다.

| 순번 | 상품번호 | 종류 | 상품명 | 가격 | 삭제 |
|----|------|----|-----|------|--------------------------|
| 1 | 1 | 지류 | 책사 | 1200 | <input type="checkbox"/> |
| 2 | 1 | 지류 | 책스 | 1000 | <input type="checkbox"/> |
| 3 | 1 | 지류 | 책지 | 1300 | <input type="checkbox"/> |

순번 4

상품번호 1

종류 지류

상품명

가격

사진 등록

(그림 8) 상품을 등록하고 삭제하는 페이지

주문관리 관련 기능으로는 사용자가 주문한 내역과 정보를 확인하고, 각 주문별 업데이트를 할 수 있도록 구현했다. (그림 9)는 주문서를 확인하는 관리자 페이지이다.

| 주문번호 | 주문일자 | 주문인 | 주문인 전화번호 | 판매사합 | 삭제 |
|----------|------------|-----|---------------|---------|--------------------------|
| 14613684 | 2003-09-02 | 손 | 1-1-1 | (주말미확인) | <input type="checkbox"/> |
| 17592819 | 2003-09-03 | 손 | 33-3-3 | (주말미확인) | <input type="checkbox"/> |
| 12504188 | 2003-09-05 | 김 | 1-2-3 | (주말미확인) | <input type="checkbox"/> |
| 6967903 | 2003-12-15 | 홍길동 | 111-1111-1111 | (주말미확인) | <input type="checkbox"/> |

(그림 9) 주문 리스트를 확인하는 관리자 페이지

4. 결론 및 향후 과제

본 논문은 기존에 사용되고 있던 EC Merchant Server에서 상품관리에 대한 기능을 보완하고, 컴포넌트 방식을 도입함으로써 비즈니스 프로세스 관리의 유연성을 확보하며 코드의 재사용성을 증가시키는 등 시스템의 유연성을 향상시키고자 하였다. 더불어 프로젝트 총 비용의 60%를 유지보수비가 차지하는 현 상황에서 업그레이드의 신속성과 향상된 관리기능 등으로 유지보수의 편의성과 비용 절감성을 추구하였다.

그러나 본 논문의 주제가 현장에서 적용되기 위해서는 기업 특성을 좀 더 반영하는 등 최적화 프로세스가 이루어져야 한다.

참고 문헌

- [1] 김경래, 하상호, “EJB기반 XML 상품 카탈로그의 구현 및 분석”, 정보처리학회논문지, 제9-D권 제3호, 2002.
- [2] 신동규 외 5, “EJB기반 모바일 전자상거래 컴포넌트의 설계 및 구현”, 정보처리학회논문지, 제9-D권 제4호, 2002.
- [3] 김세곤, “엔터프라이즈 자바빈즈 바이블”, 정보문화사, 2001.
- [4] 박동혁, “JSP 웹프로그래밍”, 가메출판사, 2000.
- [5] 박지훈, “엔터프라이즈 자바빈즈”, 대청미디어, 2001.
- [6] 박천수, 문창수, “EJB&WebLogic”, 가메출판사, 2002.