

# EJB를 이용한 온라인 주문 시스템 구현

신용녀, 최진영  
고려대학교 컴퓨터학과 정형기법연구소

## Implementation Of On-Line Ordering System Using EJB

Yong Nyuo Shin, Jin Young Choi

Formal Methods Lab. Dept. CSE Korea University

### 요 약

복잡한 시스템을 구현하는데 컴포넌트의 사용이 유용하다는 것은 이미 누구나 인지하고 있는 사실이다. 엔터프라이즈 자바 빈즈(Enterprise JavaBeans, EJB)는 컴포넌트 트랜잭션 모니터를 위한 표준 서버측 컴포넌트 모델이다. 엔터프라이즈 자바 빈즈 아키텍처를 이용해 만들어진 어플리케이션은 확장성이 있고, 트랜잭션을 보장할 수 있으며, 다수 사용자 환경에서도 안전하다. 기존의 객체지향 방법론으로 어플리케이션을 개발하더라도 진정한 소프트웨어 재사용성을 보장할 수 없었다. 그러나 EJB 아키텍처를 사용하면 소프트웨어 재사용성을 보장하고 보다 용이하게 새로운 어플리케이션을 개발할 수 있다. 본 논문은 정확한 분석과 설계를 위하여, Rational rose의 UML을 이용하여 시스템을 모델링하고, 템플릿 제너레이트인 인라인(inline)과 비주얼 카페를 연동하여 온라인 주문 시스템을 구현하였다.

### 1. 서론

EJB는 사용자가 어플리케이션을 구현함에 있어서 오직 비즈니스 로직에만 전념할 수 있도록 하여 컴포넌트 재사용성을 높인 서버측 컴포넌트 모델이다. 엔터프라이즈 빈은 EJB 서버에서 수행되며, 3-tier의 클라이언트-서버 시스템에서 미들-tier의 역할을 한다.[1] EJB는 기본적으로 컴포넌트의 조합으로, 어플리케이션 구축을 위해 제작되었기 때문에 컴포넌트 중심으로 엔터프라이즈 어플리케이션 설계와 개발, 조합(assembly)과 배치(deployment)과정을 거쳐 분산 컴퓨팅 환경 구축이 가능하다. EJB의 컴포넌트 아키텍처는 컴포넌트의 표준적인 구축에서부터 관리 및 유지보수까지의 전반적인 컴포넌트 개발하기 위한 불과 배치된 컴포넌트를 관리하기 위한 배치된 컴포넌트를 관리하기 위한 컨테이너, 그리고 컴포넌트를 배치시키고 유지 관리하는 불로 이루어진다. 엔터프라이즈 자바 빈즈는 자바의 객체 지향적 특성을 모두 상속받았으며 부가적으로 컴포넌트에 대한 속성도 가지고 있다. 따라서 EJB로 프로그래밍하기에 앞서 구현하려는 시스템에 대한 정확한 분석, 설계 작업이 이루어지지 않으면 나중에 프로그래밍 스펙을 변경할 때 많은 어려움을 겪게 된다. 그렇기 때문에, EJB로 프로그래

밍하기에 앞서 구현하려는 시스템에 대한 정확한 분석/설계작업이 선행되어야 한다.

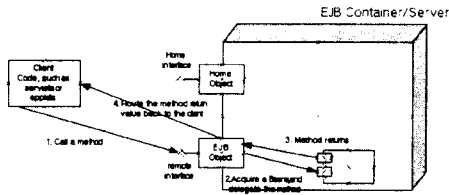
본 논문은 정확한 분석과 설계를 위하여, Rational rose의 UML을 이용하여 시스템을 모델링하고, 템플릿 제너레이트인 인라인(inline)과 비주얼 카페(visual cafe)를 연동하여 온라인 주문 시스템을 구현하였다. 본 논문은 2장 EJB를 구조와 엔터티빈과 세션빈을 기술하고, 3장에서는 웹주문 시스템을 모델링, 4장 구현 및 배치, 5장 결론 순으로 구성하였다.

### 2. EJB(Enterprise JavaBeans)

#### 2.1 EJB의 구조

<그림 1>에서는 EJB 객체의 생성과 실행순서를 도식적으로 보여준다. 클라이언트가 홈 인터페이스(Home Interface)를 이용해 새로운 EJB객체를 생성하고, EJB객체의 레퍼런스를 클라이언트에게 반환한다.

클라이언트가 리모트 인터페이스를 통해 엔터프라이즈 빈의 메소드를 호출하면 EJB 오브젝트는 빈을 얻고 메소드를 빈에게 위임한다. 엔터프라이즈 빈은 EJB 오브젝트에게 메소드를 반환하고 EJB 오브젝트는 메소드를 수행한 후, 그 결과 값을 클라이언트에게 반환한다.[4]



<그림 1> EJB object 객체의 생성과 실행 순서

2.2 세션 빈과 엔티티 빈

모든 엔터프라이즈 자바빈은 빈의 생성과 관련된 기능을 가지고 있는 홈(Home)과 비즈니스 로직을 가지고 있는 리모트(remote) 인터페이스들을 가지며, 클래스가 지원하는 기능과 상태, 자료의 특징에 따라 세션(session)빈과 엔티티(entity)빈으로 나누어진다. 따라서 EJB를 작성할 때는 세션빈과 엔티티빈을 나누는 작업이 선행되어야 한다. 엔티티 빈은 '영구 저장되는 객체'를 나타낸다. 엔티티 빈은 데이터베이스에 실제로 저장되는 데이터를 표현하고 있으며 엔티티 빈과 데이터베이스와는 매핑이 이루어져 있어서 엔티티 빈의 상태 값이 변경되면 곧바로 데이터베이스에 변경 내용이 반영된다. EJB의 사용자는 복잡한 JDBC-ODBC의 연동은 신경 쓰지 않고 어플리케이션을 구현할 수 있다. 전자 상거래를 구현함에 있어서도 각 빈을 배치할 때, 사용하는 데이터베이스 만 명시해 줌으로써 개발자가 신경써야할 모든 데이터베이스 연동을 위한 작업이 끝나게 된다. 세션 빈은 시스템에서 주로 비즈니스 로직에 해당하는 업무를 담당한다. 세션 빈은 엔티티 빈을 사용하거나 다른 세션 빈을 사용해 비즈니스 영역의 작업을 수행한다. EJB에서 엔티티 빈과 세션 빈을 구분한 가장 큰 이유 중 하나가 바로 데이터베이스 연동을 EJB 컨테이너에서 하고 나머지 비즈니스 로직만 개발자에게 맡기기 위한 것이다. 이렇게 각기 역할에 따라 빈을 구분하는 것은 사용의 유연성을 주기 위해서이다. 엔터프라이즈 자바빈즈를 일반적으로 구분하는 방법은 엔티티 빈은 지속적인(persistent) 상태를 가지는 반면, 세션 빈은 상호 작용을 수행하지만 지속적이지 않는다는 점이다.

3. 웹 주문 시스템(on-line ordering system)

집이나 사무실에서 원하는 음식을 배달시켜 맛보는 것은 우리 주위에서도 흔히 볼 수 있다. 그만큼 많은 사업장에서 이용할 수 있는 어플리케이션을 EJB 아키텍처를 이용하여 빈으로 만들어 놓는다면 유사한 시스템에 충분히 다시 사용할 수 있을 것이다. 고객의 주문에 대하여 레스토랑에 위치한 서버에서 계산서를 바로 발급하여 웹상에 보여주고 고객의 주문 사항에 관련한 정보

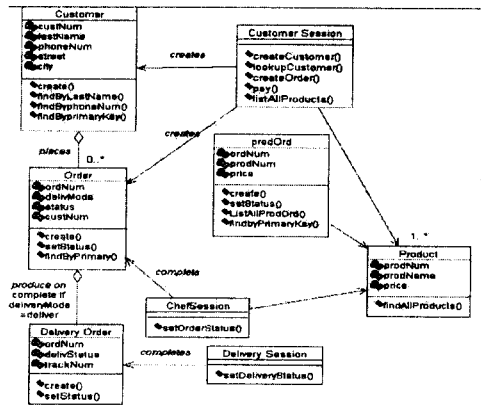
를 다시 배달회사의 웹상으로 전달하는 시스템을 모델링하고 구현한다.

3.1 요구사항

- 가. 고객은 인터넷상으로 주문 목록을 보고 자신이 원하는 메뉴와 해당 아이템 개수를 주문한다.
- 나. 고객의 선택 사항들은 서버로 전달되어진다.
- 다. 서버는 아이템에 대한 가격을 계산하고 고객에게 그 가격을 다시 확인시켜준다.
- 라. 고객은 그 가격을 받아들일 것인지 거절할 것인지를 결정한 후, 고객이 주문을 원할 경우 주문한 아이템과 개수가 레스토랑의 서버에 전달되어 프린트된다.
- 마. 시스템은 배달되어질 주문에 대한 정보를 배달 서비스 회사 모니터에 알람과 함께 보여준다.

3.2 UML 모델링

EJB 프로그램의 분석/설계 방법은 UML에서 제공되는 다이어그램과 EJB의 개념과 대부분 일치하기 때문에 UML로 EJB를 아주 훌륭하게 분석/설계할 수 있다.[2] 단순히 하나의 엔티티 빈 혹은 세션 빈으로 컴포넌트를 작성한다면 UML의 클래스 다이어그램(Class Diagram)으로 나타낼 수 있으며, 빈들을 복합적으로 사용하여 컴포넌트를 구성한다면 UML의 컴포넌트 다이어그램(Component Diagram)으로 나타낼 수 있다. 또한 이들을 패키지화 해서 관리한다면 UML에 있는 패키지 다이어그램(Package Diagram)으로, 컴포넌트를 배치하기 위해서는 UML에 있는 배치 다이어그램(Deployment Diagram)을 작성해 각 컴포넌트가 배치되는 위치를 도식적으로 나타낼 수 있다. <그림2>는 웹 주문시스템을 클래스 다이어그램으로 나타낸 것이다.

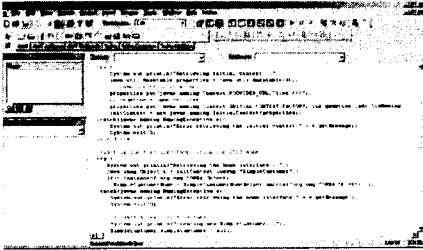


<그림 2> on-line ordering system의 class diagram

4. ODS(online ordering system) 구현

#### 4.1 구현 환경

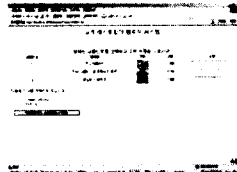
로즈를 사용해 분석/설계된 결과를 모델링하고, 온라인을 통해 모델링 클래스들에 대한 EJB 템플릿(Template) 코드를 생성했다. 또한 생성된 템플릿을 이용해 비주얼 카페(visual cafe)3.x(1.0)에서 코딩 작업을 하고, EJB 상용 서버인 잼스톤/J(Gemstone/J)에 배치해 실행했다.[3]



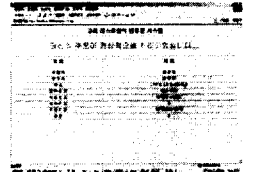
<그림 3. 비주얼 카페에서의 ODS 코딩 >

#### 4.2 잼스톤/J EJB 프로그램 작성과 배치

인라인 소프트웨어(inline software)를 이용하게 되면 <그림3>에서 보여주는 것처럼 비주얼 카페 메뉴바에 inline이라는 항목이 추가된다. 새로운 프로젝트를 만들게 되면 클래스 다이어그램이 코드로 생성된다. 잼스톤(gemstone)의 DDGen(Deployment Descriptor)을 이용하여 배치기술서(.ser)를 생성하고 잼스톤에 배치하기 위해 jar를 파일을 생성한다. 잼스톤/J를 구동시키고 로그인하여, 컴포넌트를 배치한다. 클라이언트가 웹상에서 입력한 값은 서블릿으로 전송되고 서블릿에서 세션 빈에 값을 넘겨주게 되어 주문에 대한 가격을 계산하게 된다. 클라이언트가 가격에 대한 주문을 받아들이면 값을 엔터티 빈에 저장하고 데이터 베이스에도 저장하게 된다. 서블릿에서 JNDI(Java Naming Directory Interface)를 이용하여 엔터티 빈의 값을 호출하면 엔터티 빈은 데이터 베이스에 값을 요청하게 된다. 요청된 값이 엔터티 빈에 저장되면 이 값을 서블릿이 참조하게 된다. <그림 4><그림5>는 웹상에서 이 시스템이 주문에 대해 처리하고 서블릿으로 엔터티 빈의 값을 참조하는 예를 보여 준다.



<그림 4> 클라이언트의 주문 인터페이스



<그림 5> 주문처리 결과 인터페이스

#### 5. 결론

본 논문에서는 EJB의 전반적인 구조를 소개하고, 세션 빈과 엔터티빈을 비교 설명하였으며, 온라인 주문 시스템을 UML로 모델링하여 비주얼 카페에서 코드를 생성, 구현한 경험을 기술하였다. 생성된 코드를 배치 기술서와 함께 압축하여 상용 프로그램인 잼스톤에 배치하고 클라이언트가 서버에 배치된 빈을 자유자재로 사용하게 되는 과정을 설명하였다. EJB로 소프트웨어의 재 사용성을 최적으로 실현하기 위해서는 구현 전에 반드시 정확한 설계와 분석이 이루어져야 한다. 시스템의 빈을 재 사용할 때, 빈의 최적화를 위해서 스펙을 변경하게 될 경우, 정확한 설계와 분석 후에 코드를 생성해야만 코드 수정을 최소화 할 수 있을 것이다.

향후 연구과제로는 EJB 버전 1.1 이후 새로이 추가된 기능들에 대해 정형명세(formal specification)를 하고자 한다. EJB가 시스템을 구축하는데 유용한 반면, 사용하기 위해서는 전반적인 구조에 대해 명확하게 이해하고 있어야 된다는 제약이 있는데[5] 이를 위해서 정형명세를 하는 것은 아주 요원한 일이라고 생각한다.

#### 6. 참고문헌

- [1]http://www.javasoft.com/ejb, "EJB 1.1 specification"
- [2]Henri Jubin, Jalapeno Team, Enterprise JavaBeans by Example, 1999
- [3]http://www.javasuccess.com/cookbook/create\_deploy\_desc\_ddgen.htm, "Creating a Deployment Descriptor with DDGen"
- [4]John Willy & Sons,E. Roman, Mastering Enterprise Java Beans, 1999
- [5] Joao Pedro Sousa and David Garlan, Formal Modeling of the Enterprise JavaBeans Component integration Framework, 1999