

EJB 기반 전자상거래 시스템 설계 기법

⁰윤희윤, 김수동
송실대학교 컴퓨터학과

hyoon@selab.soongsil.ac.kr, sdkim@computing.soongsil.ac.kr

Design Techniques for Developing Electronic Commerce with EJB

⁰Hee Yoon Yoon, Soo Dong Kim
Dept. of Computer Science, Soongsil University

요 약

인터넷 기반의 전자상거래 시스템 개발은 안정적 트랜잭션 처리, 복잡한 데이터베이스 구조, 멀티미디어 데이터 처리, 입체적 사용자 인터페이스 등의 여러 기술들이 적용되는, 복잡한 과제로 인식되고 있다. 웹 서버 기반의 객체지향 및 컴포넌트 표준으로 알려진 Enterprise JavaBeans(EJB)은 복잡한 시스템 개발 기간을 단축하고 소프트웨어 품질을 높이며 재사용성이 높은 컴포넌트 소프트웨어 개발을 가능하게 한다. 본 논문에서는 전자상거래 개발을 위한 EJB 적용 지침을 제안한다.

1. 서론

인터넷의 급속한 보급은 전자상거래 라는 새로운 유형의 유통 문화를 형성하였다. 고객들은 직접 회사나 매장을 방문하지 않고, 전자상거래 웹 사이트에 접속하여 상품 정보를 검색하고, 원하는 물건을 주문하며, 구매 대금을 신용카드나 전자 화폐로 결제한 후, 주문한 물건을 배달 받게 된다. 전자상거래 거래 대상도 개인 고객을 대상으로 한 B2C 에서 기업 고객을 대상으로 하는 B2B 로 발전하고 있다.

인터넷 기반의 전자상거래 시스템 개발은 안정적 트랜잭션 처리, 복잡한 데이터베이스 구조, 멀티미디어 데이터 처리, 입체적 사용자 인터페이스 등의 여러 기술들이 적용된다. 따라서, 전자상거래 시스템 개발은 복잡한 과제로 인식되고 있다.

웹 서버 기반의 객체지향 및 컴포넌트 표준으로 알려진 Enterprise JavaBeans(EJB)은 복잡한 시스템 개발 기간을 단축하고 소프트웨어 품질을 높이며 재사용성이 높은 컴포넌트 소프트웨어 개발을 가능하게 한다.

본 논문에서는 전자상거래 개발을 위한 EJB 적용 지침을 제안한다. 제 2 장에서는 관련 연구로서 기존의 EJB 의 특징과 전

자상거래의 특징을 간략하게 설명하고 3 장에서는 EJB 기반의 전자상거래의 시스템 구축전략을 기술하고, 4 장에서는 결론에 관하여 고찰한다.

2. Enterprise Java Beans 개요

Enterprise JavaBeans(EJB)는 Java 기반의 컴포넌트 소프트웨어를 구현 할 수 있는 아키텍처로 각광을 받고 있다. 클래스 다이어그램의 클래스는 EJB 의 빈(Beans) 단위로 구현되며, 구현된 빈은 EJB 컨테이너(Container)라는 미들웨어를 통하여 관리되어 진다[1].

개발된 EJB 빈들은 컨테이너가 제공하는 여러 시스템 서비스를 제공받게 되는데, 대표적인 서비스가 원격 객체를 찾아 주는 Naming & Directory, 트랜잭션 처리, 데이터베이스 연결, 보안 등이다. 이러한 서비스를 이용하여 객체지향 소프트웨어를 개발하게 되면, 개발 시간을 크게 단축하고, 생산성을 향상시킬 수 있다. 또한, 서로 다른 하드웨어 플랫폼 및 운영체제들 간에도 EJB 컨테이너를 통하여, 상호 연동이 가능하여 개방형 시스템 개발을 유도하고 있다[2][3].

EJB 빈은 Session bean 과 Entity Bean 의 두 종류로 분류 할 수 있다. 정의된 각 클래스가 영구적 데이터를 가지고 있으면 Entity 빈으로 구현되며, Session 빈은 Entity 빈의 정보를 처리 하는 업무 로직을 구현하고 있다. 3-tier 클라이언트 서버 계층 과 비교하며, EJB 는 Session 빈들로 이루어 지는 업무 로직 계 층과 Entity 빈으로 구성되는 데이터 계층을 제공한다[4].

3. 전자상거래 시스템 특징

3.1. 전자지불 등 안정적 Transaction 처리

전자상거래는 제품 검색으로 시작하여 제품을 주문하고 구 매 금액을 지불한 후, 배달을 받는 일반적인 과정을 가지고 있다. 이때 주문, 결제 등은 트랜잭션 단위로 수행되는데, ACID 조건을 만족하는 트랜잭션이 제공되어야 한다. 즉, 중요한 전자상거래 기능 수행 중 어떤 문제가 발생하여도, Binary Effect 를 제공해야 한다[5].

트랜잭션을 구현하는 전통적인 방법은 2 Phase Commit 프로 토콜을 직접 구현하거나, Tuxedo 등의 TP Monitor 미들웨어를 사용하는 것이다. 전자의 경우는 개발 난이도가 높고 후자의 경우는 미들웨어의 시스템적 부하와 비용적 단점을 가지고 있다.

3.2. 복잡하고 대규모의 데이터베이스

전자상거래에서는 대규모의 제품 정보, 고객 정보, 구매 정 보 등의 시스템에서 사용되는 많은 정보들을 가지고 있으며 사용되는 정보들을 데이터베이스화 하고 있다. 타 시스템에 비해 비교적 복잡하고 대규모인 데이터베이스를 가지고 있으 며 데이터베이스 사용도 많이 일어난다. 또한 새로운 제품의 추가나 품질된 제품의 삭제로 인하여 잦은 데이터 베이스 스 키마의 변화를 요구하고 있다

3.3. 그래픽, 동영상 등 입체적 UI 제공

인터넷이 발달하면서 웹 사이트가 점점 복잡해지고 있다. 단순한 텍스트 위주였던 것이 몇 년 사이에 동적이고 이미지 형태로 변화하였다. 여기에서 전자상거래도 예외일 수 없다. 전자상거래 역시 화려한 그래픽과 동영상으로 고객의 눈길을 끌잡으려 하고 있다. 전자상거래는 더 이상 콘텐츠 만 가지고 경쟁할 수 있는 것이 아니라 질 높은 서비스와 다양한 가능성을 제시하게 된다. 이렇게 유저 인터페이스는 더욱 중요하게 인식되었고 더 큰 규모의 그래픽과 동영상의 UI 를 가진 시스 템을 요구하게 된다.

3.4. 확장성 높은 N-tier 아키텍처

전자상거래가 활발해지면서 고객의 정보나, 상품의 정보 등 의 데이터베이스, 시스템의 증가와 다수의 고객의 사용으로 인하여 서버의 부하가 생기게 된다. 서버의 부하의 따른 시스 템 정지나 데이터의 손실을 방지하기 위하여 서버의 역할을 분산할 필요가 있다. 이때 분산형 n-tier 아키텍처를 이용하여 시스템을 계층별로 분할하고 각 계층에 따라 확장하고 분 산을 할 수 있다. 이런 분산형 시스템을 통하여 향후 시스템 의 확장에 보장한다.

4. EJB 기반 전자상거래 시스템 구축전략

4.1. EJB 의 안정적인 Transaction

전자상거래 트랜잭션은 EJB 트랜잭션 서비스를 통하여, 효 윅적이고 안정적으로 구현될 수 있다. 전자상거래 시스템의 각 트랜잭션 특성을 분석하여, Bean-Managed Transaction(BMT)나 Container-Managed Transaction(CMT)를 적용하면 된다.

BMT 은 빈의 코드 안에 개발자가 직접 구현을 해서 트랜잭 션을 관리하는 방법이다. 그렇기 때문에 CMT 에 비해 구현하기 힘들다는 단점이 있지만 사용자가 원하는 대로 트랜잭션을 구 현할 수 있다. 전자상거래에서 많은 연산을 처리하는 클래스 일 경우 BMT 를 이용하여 트랜잭션을 프로그램 하거나 트랜잭 션을 컨트롤 할 경우에 사용할 수 있다.

CMT 은 개발자가 직접 코드를 구현할 필요가 없이 트랜잭션 에 관한 내용을 빈 설치자가 Deployment Descriptor 에 트랜잭션의 속성을 서술하여 트랜잭션을 설정한다. BMT 에 비해 구 현하기 간편하고 개발자의 부담을 줄여준다. CMT 에서 Deployment Descriptor 에 작성하는 트랜잭션의 속성은 NotSupported, Required, Supports, RequiresNew, Mandatory, Never 가지가 있다.

시스템에서 트랜잭션을 강하게 요구해야 하는 클래스일 경 우에는 Mandatory, RequiresNew, Required 순으로 사용하여 구현다. 또한 트랜잭션을 처리하지 않기를 요구하는 클래스의 경우 Never 의 속성을 이용하여 처리한다. 전자상거래 시스템 에서는 EJB 가 지원해주는 6 가지 트랜잭션을 이용하여 보다 안정적인 트랜잭션을 구현할 수 있다.

4.2. 복잡하고 대규모의 Database 사용

EJB 에서는 Entity bean 을 이용하여 Database 의 정보를 접근할 수가 있다. Entity bean 를 이용하여 bean 을 생산하면 재사용성을 높여주고 개발 비용을 줄임으로써 개발 생산성을 높여준다. Entity bean 중의 CMP 를 이용한 bean 은 개발자 로 하여금 별도의 지속성 관리를 위한 코딩을 해주지 않고 개 발자가 비즈니스로직의 작성에만 전념할 수 있다. 그렇기 때 문에 빈의 개발이 쉽고 개발 시간을 줄일 수 있다.

전자상거래에서 사용되는 데이터베이스를 사용하는 class 는 고객 정보, 제품 정보, 구매 정보 등의 대규모의 정보를 입력, 삭제, 불러오기, 갱신, 등의 기능을 한다. 이때 CMP 를 이용한 bean 의 개발은 아주 효과적이라 할 수 있다.

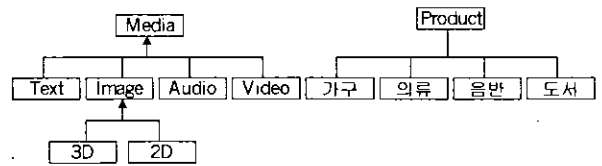


그림 1. 상속을 이용한 제품 클래스 계층

또한 전자상거래에서는 데이터베이스의 스키마가 자주 변화
 게 된다. 어떤 상품이 추가되며 그 상품에 대한 데이터베이
 스의 스키마가 변하게 되고 그에 따른 개발이 요구된다. 그림
 1에서 Audio 라는 새로운 상품이 추가되었을 때 Audio 는 자신
 을 포함할 수 있는 상위 상품의 서브클래스의 위치에 놓이게
 된다. Audio 는 Media 라는 슈퍼클래스에 대한 속성을 상속 받
 으며 Audio 의 특성에 대한 속성만을 추가하게 된다. 그러므
 로 자주 변화하는 DB Schema 에도 개발이 쉽고 개발 시간이 적
 어 유리하다.

대규모의 데이터베이스는 서버의 부하에 따른 데이터의 분
 산을 요구하게 된다. 이런 경우 데이터베이스를 사용하는 엔티
 티빈을 컨테이너에 의해 관리한다. 엔티티빈이 여러 서버에 분
 산되어 있어도 컨테이너의 JNDI(네이밍/디렉토리 서비스)를 이
 용하여 사용자가 같은 서버에 있는 것 같이 느낌을 주어 데
 이터 분산에 있어 투명하게 처리 할 수 있다.

4.3. 그래픽, 동영상 등 입체적 UI 제공

서블릿(Servlet)과 애플릿(Applet)은 J2EE 환경에서의 웹 인
 터페이스 부분을 맡는다. J2EE 환경을 고려하지 않는다면 서
 블릿과 애플릿은 자바빈즈를 사용해서 비즈니스 로직을 처리
 하지만 J2EE 환경에서는 자바빈즈 대신에 EJB 를 사용해서 비
 즈니스로직을 처리한다. 즉 개발자는 애플릿과 서블릿을 이용
 하여 사용자에게 있어서 입체적 유저 인터페이스를 제공할 수
 있다.

서블릿의 경우 클라이언트의 있는 정보가 서버로 이동하여
 처리된 후 다시 클라이언트로 보여지게 된다. 그렇기 때문에
 서버에 있는 데이터베이스를 많이 사용하거나 보안에 관련된
 클래스에서 효과적이다.

애플릿의 경우 웹 서버쪽에 있는 자바 애플릿 코드가 클라
 이언트로 넘어간 후 브라우저 내의 특정 영역에서 실행하게
 됩니다[6]. 즉 클라이언트에서 실행되기 때문에 서버의 부하를
 줄일 수 있으며 실행을 서버까지 데이터를 이동하지 않고도
 클라이언트 자체에서 실행하므로 실행 결과 값이 빠르다. 그렇
 기 때문에 애플릿의 경우 그래픽이나 동영상과 같이 규모가
 커서 서버에 많은 부하를 주는 부분에 대하여 클라이언트에서
 실행되어 서버의 부하를 줄여주는 유저 인터페이스를 제공해
 줄 수 있다. 이러한 클라이언트에서의 자체 실행으로 입체적
 인 UI를 제공하게 된다.

4.4. 확장성 높은 N-tier 아키텍처

전자상거래가 발전함에 따라 시스템이 대형화 되어가고 있
 다. 이러한 시스템을 확장 관리하기 위해서는 분산객체 기술
 에 기반을 둔 EJB 를 이용하는 것이 효과적이다.

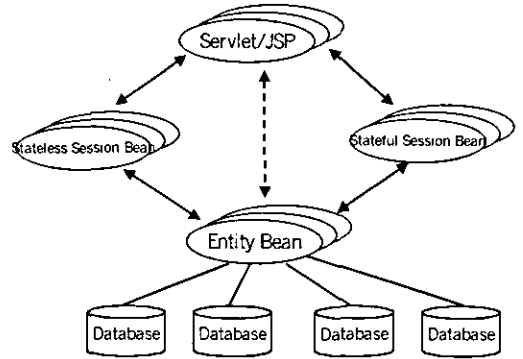


그림 2. EJB N-tier 아키텍처

그림 2 는 EJB 의 N-tier 아키텍처를 보여주고 있다. 여러
 개로 분할된 엔티티빈은 다수의 노드에서 분산하여 시스템의
 효율성을 증가 시킨다. 또한 Stateless Session Bean 과
 Stateful Session Bean 에서는 여러 노드에 분산하여 부하조절
 (Load Balancing) 함으로 어떤 노드에 이상이 생겼거나 부하
 로 인한 문제를 막을 수 있다.

전자상거래에서는 이런 EJB 아키텍처를 이용하여 시스템의
 새로운 노드의 확장이 수월하다. 각 빈에 따라서 새로운 노드
 로 시스템을 확장함으로써 서버의 부하를 줄이고 노드의 효율성
 을 증가시킨다.

5. 결론

전자상거래 시스템 개발에 있어서 안정적 트랜잭션 처리,
 복잡한 데이터 구조, 멀티미디어 데이터 처리, 입체적 사용자
 인터페이스 등의 복잡한 과제로 인식되었다. 본 논문에서는 이
 러한 복잡한 부분을 EJB 를 사용하여 개발 기간을 단축하고
 품질을 높이는 개발을 제시하였다. 시스템의 안정적인 트랜잭
 션을 위한 EJB 에서의 지원 속성을 결정하고, 대규모의 데이터
 베이스의 CMP 사용과 유동적인 데이터스키마를 위한 상속을
 사용한다. 또한 입체적 UI 를 위한 서블릿과 애플릿을 설명하
 고 확장성을 높이기 위해서 EJB 아키텍처를 제안하였다.

참 고 문 헌

[1] Enterprise JavaBeans Specification Version 2.0 2000
 [2] 김수동, "Enterprise JavaBeans(EJB) 기반의 컴포넌트 프로그
 램밍", 정보처리학회, 2000.
 [3] 민현기, 김수동, "효율적인 EJB 트랜잭션 설계 기법", 소프
 트웨어 공학회, 2000
 [4] Ed Roman, *Mastering Enterprise JavaBeans™ and the Java™2
 Platform, Enterprise Edition*, pp. 263-302, WILEY, 1999.
 [5] Philip A. Bernstein, *Principles of Transaction Processing*, pp. 8-
 14, Morgan Kaufmann Publishers inc, 1997
 [6] Ari Halberstadt, *Professional Java Server Programming*, pp. 33-41,
 Wrox, 1999