# EJB 기반 경매시스템 사례 연구

최시원, 김수동
숭 실 대 학 교 컴 퓨 터
swchoi@selab.soongsil.ac.kr

# A Case Study of the Auction System based on EJB

Si-Won choi, Soo Dong Kim
Dept. of Computer Science, Soongsil University

## 요 약

인터넷의 급속한 발전으로 인터넷 경매와 같은 e-business 가 활발해 지면서 사용자의 요구사항도 빠르게 변하고 있다. 급변하는 사용자들의 요구사항을 반영하고 소프트웨어의 경제성, 시장 경쟁력 확보를 위한 방법은 산업계 및 학계에서 많이 거론되고 있다. 최근 EJB (Enterprise JavaBeans) 기반의 컴포넌트 개발은 재사용성을 확보할 수 있는 가장 주목 받는 방안으로 제시되고 있다. 기존의 객체지향에 의한 소프트웨어 개발 방법은 이미 재사용성, 유지보수성, 무결성, 안정성 등의 많은 장점으로 인해 학계와 산업계에서 이미 많이 보편화 되어 있고 널리 이용되어지고 있다. 하지만 EJB 에 기반하여 컴포넌트 어플리케이션을 개발하는데 있어서는 그 적용 사례가 드물고 그 활요성 또한 검증된 바가 거의 없다. 따라서 본 논문에서는 인터넷 경매시스템에 EJB 를 적용해 봄으로써 시스템 개발시의 그 실무적인 유용성을 검증해 본다.

## 1. 서론

요즘 EJB 컴포넌트 기반 시스템 개발에 대한 관심이 집중되고 있다. 인터넷의 급속한 확산으로 인해 전자 상거래와 같은 e-business 가 활발해 지면서 이에 따라 사용자의 요구사항도 급변하고 있다. 과거의 구조적 개발방법으로 시스템을 개발하는데는 급변하는 사용자 요구를 수용하기에는 이미 많은 문제점이 대두 되었고, 그 후 객체지향 개발 방법에서는 이러한 문제점을 극복하고 보편화 되어 사용하고 있다. 그러나 객체지향 개발에서의 객체는 같은 언어로 짜여진 응용 프로그램에서만 사용가능하고 또한 고객 응용프로그램을 위해 다른 언어로의 확장이 쉽지 않다는 점이다. 이에 반해 EJB 컴포넌트는 플랫폼과 언어의 독립적으로 개발자로 하여금 유연성 있게 응용 프로그램의 코드를 다시 사용할 수 있어 기존의 방법보다 급변하는 요구사항을 능동적으로 대처할 수 있다.

이에 본 논문에서는 인터넷 경매 시스템을 EJB 기반의 컴포넌트화한 구현 사례를 제시 하려고 한다. 2 장에서는 관련 연구로서 UML 과 EJB 에 대해 기술하고, 3 장에서는 인터넷 경매 시스템의 핵심 기능을 분석과 설계하여 제시하고, 4, 5 장에서는 구현 및 결론을 제시한다. 아직까지 EJB 로 구현된 사례가 많지 않기 때문에 보다 더 많은 검증과 연구가 필요하다.

## 2. 관련연구

### 2.1 UML

UML 은 Booch, Rumbaugh, Jacobson 에 의해서 개발된 객체지향 분석 및 설계 방법으로 객체지향 개발을 위한 통일된 모델링 언어(Unified Modeling Language)이다. UML 은 소프트웨어를 시각화 및 명세화 하여 생성하고 문서화하기 위한 표준화된 언어이며 구현과는 상관없는 소프트웨어의 전 개발 과정에 걸쳐 프로세스에서 사용되어지는 방법이다.

### 2.2 EJB

엔터프라이즈자바빈(EnterpriseJavaBeans,EJB)은 서버 상의 자바 컴포넌트와 그 컴포넌트의 동작 환경에 대한 프레임워크를 의미한다. 엔터프라이즈빈은 세션빈과 엔티티빈으로 나뉘는데 세션빈은 일반 비즈니스 로직을 가지며, 데이터베이스 같은 지속적인 저장 장치에 연결되지 않은 빈이다. 엔티티빈은 데이터베이스에 연결되어 영구적이 자료관리를 할 수 있는 빈을 말한다. 또 엔티티빈은 컨테이너-관리 엔티티 빈(Container-Managed Entity Bean) 과 빈-관리 엔티티 빈(Bean-Managed Entity Bean) 으로 나뉘어지는데, 컨테이너-관리 엔티티 빈은 컨테이너에서 프라이머리 키 클래스를 이용

하여 테이블의 컬럼과 엔티티 빈의 필드 변수를 연결시켜주면 컨테이너가 자동적으로 내부 코드를 생성해 주게 된다. 빈-관리 엔티티 빈은 빈 개발자가 직접 소스코드상에서 데이터베이스를 접근하여 자료의 지속성을 보장하고, 빈 인스턴스와 데이터베이스 사이에서 상태를 관리함에 유연성을 줄 수 있다.

## 3. 분석 및 설계 단계

### 3.1 Use Case 다이어그램

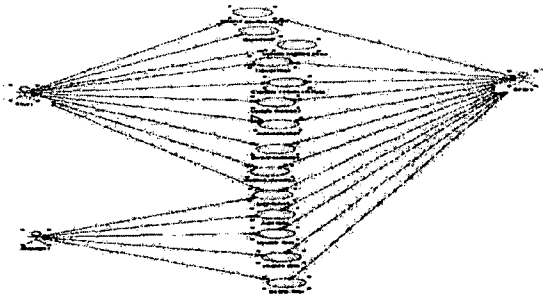

그림 1. Use Case 다이어그램

그림 1 은 다음과 같은 시나리오에 의해 작성된 핵심 유즈케이스 이다. 구매자는 자신이 사고싶은 물품이 현재 경매중인지를 찾아보고, 경매에 참여 한다. 그리고 계속적인 입찰을 하여 물품을 낙찰시킨다. 그러나 입찰도중에 구매의사가 없어질 경우 그 경매를 포기한다. 판매자는 판매할 물품을 경매에 붙인다. 판매자는 입찰내역을 확인하고, 물품이 낙찰 되었다 하더라도 판매자가 부당한 거래라고 판단될 때는 경매를 취소 할 수 있다.

### 3.2 Class 다이어그램

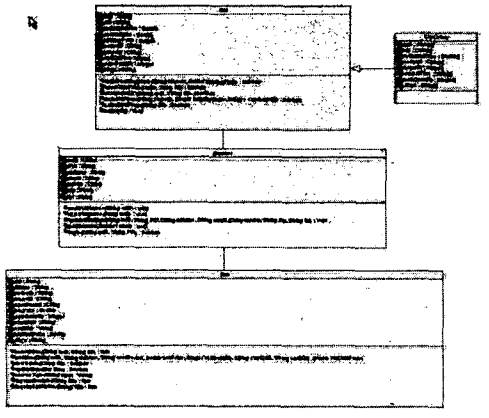

그림 2. 클래스 다이어그램

그림 2 는 유즈케이스다이어그램을 바탕으로한 클래스 다이어그램으로 3 개의 클래스 비드(Bid), 멤버, 아이템으로 구성되어있는데, 인터넷 경매의 중요 일부분이다. 우선 사용자와 판매자 모두 회원등록을 해서 인증을 받아야 경매를 통해 구매 와 판매를 할 수 있다. 멤버클래스에 회원등록(addMember(), 삭제(deleteMebmer()), 갱신(updateMember()), 검색(searchMemer())등의 오퍼레이션이 제공되고, 비드클래스는 인증받은 회원들이 경매에 입찰에 참가 할 수 있는 오퍼레이션들 입찰갱신(updateBiddingPrice()), 입찰검색(searchBidList()), 입찰등록(writeBidForm()), 입찰취소(cancelBid()) 과 판매자가 새로운 경매를 만들 때 사용하는 오퍼레이션 경매추가(make bid())등이 제공된다. 아이템 클래스에는 아이템 등록(addItem()), 삭제(deleteItem()), 갱신(updateItem()), 검색(searchItem()) 기능이 제공된다.

## 4. 구현

구현단계에서는 설계단계에서 만들어진 클래스 다이어그램을 가지고 엔터프라이즈 자바 빈 의 컨테이너를 이용하여 컨테이너-관리 빈으로 구현한다.

### 4.1 비드 빈

```
public void ejbRemove() throws RemoveException{
            log("BidBean.ejbRemove()");
            System.out.println("BidBean.ejbRemove()");
  }

public String ejbCreate(String bID, String iID, String initialBidPrice, String
curBidPrice, String biddingPrice, String sellerID, String buyerID, String curBidTime,
String saleQuantity,String iType,String bEndTime)
    throws CreateException
  {
        this.bID = bID;
        this.iID = iID;
        this.initialBidPrice = initialBidPrice;
        this.curBidPrice = curBidPrice;
        this.biddingPrice = biddingPrice;
        this.sellerID = sellerID;
        this.buyerID = buyerID;
        this.curBidTime = curBidTime;
        this.saleQuantity = saleQuantity;
        this.iType = iType;
        this.bEndTime =bEndTime;
        return null;
    return null;
  }

  public void ejbPostCreate(String bID, String iID, String initialBidPrice, String
curBidPrice, String biddingPrice, String sellerID, String buyerID, String curBidTime,
String saleQuantity,String iType,String bEntTime) {

        System.out.println("ejbPostCreate()");   }

public String updateBid(String bID, String iID, String initialBidPrice, String
curBidPrice, String biddingPrice,   String sellerID, String buyerID, String
curBidTime, String saleQuantity,String iType, String bEndTime)
  {
    this.bID = bID;
    this.iID = iID;
    this.initialBidPrice = initialBidPrice;
    this.curBidPrice = curBidPrice;
    this.biddingPrice = biddingPrice;
    this.sellerID = sellerID;
    this.buyerID = buyerID;
    this.curBidTime = curBidTime;
    this.saleQuantity = saleQuantity;
    this.iType = iType;
    this.bEndTime = bEndTime;
    setModified(true);
        return null;        }

  public String cancelBid(String mID, String iID){
                return null;              }

  public String searchBidList(String mID, String iID) {
              return null;            }

  public String writeBidForm(String buyerID, String initialBidPrice, String
saleQuantity) {
    this.bID = bID;
    this.iID = iID;
    this.initialBidPrice = initialBidPrice;

    return null;    }
```

위의 소스는 비드 빈의 중요코드이다. 입찰갱신(updateBiddingPrice()), 입찰검색(searchBidList()), 입찰

등록(writeBidForm()), 입찰취소(cancelBid()) 등이 모두 구현 되어 있다. 하지만 코드를 보면 파라메터와 변수의 선언말고는 별도의 코드를 써주지 않았다. 이것이 엔터프라이즈 자바 빈의 컨테이너-관리 빈의 가장 큰 장점이다. 사용하기 쉽다. 컨테이너가 기본적으로 트랜잭션 처리 및 빈 생성, 삭제 등을 관리 해 주기 때문에 별도의 코드를 추가 할 필요가 없다.

## 5. 결론

본 논문에서 인터넷 경매 시스템을 EJB 기반의 컴포넌트를 구현 했다. EJB 기반의 컴포넌트의 특징은 플랫폼과 언어에 독립적이라는 것이다. 기존의 객체지향방법으로 인터넷 경매 시스템을 개발할 경우, 하나의 서버에 모든 객체들이 다 들어 가는 것이 일반적이다. 경매의 참여자가 동시에 서버에 접속하게 될 경우 서버의 과부하가 걸려서 서버가 다운 될 위험이 많이 있고, 또 이렇게 대용량의 서버의 비용문제도 대두 되고 있다. 하지만 EJB 기반으로 경매 컴포넌트를 개발할 경우 플랫폼 독립적이고 분산 환경 시스템 환경에 맞게 설계되므로 이런 과부하와 서버 비용 문제를 해결하는 하나의 방법이 될 수 있다.

또 다른 특징으로 EJB 는 트랜잭션을 보장해 준다는 것이다. 그리하여 개발자가 별도로 트랜잭션 문제를 신경쓰지 않아도 되고, 복잡한 코드로 인한 런타임 오류발생 위험을 미연에 방지 할 수 있다.

아직 까지 산업계에서 EJB 기반 시스템 개발이 많지 않아서 보다 많은 검증이 필요하지만, 경매시스템의 경우에는 EJB 기반의 컴포넌트가 매우 유용하다는 것을 알 수 있다.

다음 연구에서는 여러종류의 입찰을 포함하는 재사용성이 높은 EJB 컴포넌트를 설계 하겠다.

## 참고문헌


[1] Keith Short, *CBD and Object Modeling, Sterling Software CBD White Paper version 1.0*, February, 1997.

[2] Martin Fowler, *UML Distilled*, Addison Wesley, 1997.

[3] Booch G., Rumbaugh J., and Jacobson I., *The Unified Modeling Language User Guide*, Addison-Wesley, 1999.

[4] 민현기, 김수동, "효율적인 EJB 트랜잭션 설계 기법", 한국정보과학회 '제 2 회 한국 소프트웨어공학 학술대회', 2000

[5] 이종국, 김수동, "효율적인 컴포넌트 설계 기법", '제 13 회 춘계 학술 발표논문집', 2000

[6] 김철진, 김수동, "컴포넌트 일반성 향상 기법", '제 13 회 춘계 학술 발표논문집', 2000

[7] 김수동, "실무자를 위한 소프트웨어 공학", 에드텍, 1999.