입찰 추적을 지원하는 인터넷 경매 시스템 설계 및 구현

김 충 석†

요 약

WWW 등장이후 인터넷 응용 분야의 발전은 가히 상상할 수도 없을 만큼 빠른 속도로 급변하고 있다. 특히 전자 상거래와 관련된 응용 분야의 개발은 우리 생활 지체를 바꾸어 놓았다 해도 과언이 아니다. 본 논문은 인터넷의 응용 분야 중에서 가장 매력적인 서비스 분야로 인정되고 있는 인터넷 경매 시스템의 설계와 개발에 관한 논문이다. 인터넷 경매는 기존의 쇼핑몰과는 달리 모든 물품의 거래가 입찰과 낙찰과정을 거쳐 복잡하게 이루어진다. 본 논문에서는 복잡하게 이루어지는 경매의 전과정을 설계 구현하였으며, 특히 경매 시스템으로서는 최초로 사용자가 입찰과정을 추적할 수 있는 입찰 추적 시스템을 포함한 경매 시스템을 개발하였다. 입찰 추적 시스템은 웹에서의 호환성을 위해 HTTP만을 사용하여 구현하였으며, HTTP를 사용하여 양방향 통신을 시뮬레이션 하기 위해 자바의 블록킹 개념을 사용하였다. 본 논문은 NT 서버 상에서 JAVA와 ASP, 오라클 데이터 베이스를 사용하여 구현되었다.

The Design and Implementation of Internet Auction System Support Bidding Trace

Chung-Seok Kim†

ABSTRACT

Since the advance of WWW, internet related fields have been growing very rapidly. Especially, the development of electronic commerce areas have changed even our life style. This paper is about the design and implementation of an internet auction system, which is considered as one of the most attractive internet services. An internet auction system has more complex processes compared with other electronic commerce systems. In auction systems, bids and a successful bid are required in trading all goods. In this paper, I designed and implemented all processes of an internet auction system, in which a bid tracing system is included for the first time. The bid tracing system is implemented using only HTTP to maintain compatibility in WWW environment. I also used the JAVA blocking concept to simulate two-way communication. The proposed system is implemented on Windows NT environment using JAVA, ASP, and the Oracle database.

1. 서 론

최근 폭발적으로 증가하기 시작한 인터넷의 사용은 우리 생활의 많은 부분을 변화시키고 있다. 특히 상거래 분야에서의 인터넷 보급은 매우 빠른 속도로 진전되어 가고 있다. 초고속 인터넷의 보급과 인터넷 기술의 발달로 인하여 소비자, 기업, 정부 등 경제 주체간에 상품과 서비스를 교환하는 상거래에 있어서 인터넷을 이용하는 전자상거래가 급속히 확산되고 있다. 인터넷을 기반으로 하는 전자상거래는 시간과 장소의 제약을 탈피하고, 국경의 한계를 초월한 범 세계적인 속성을 가지고 있다.

† 종신회원 : 신라대학교 컴퓨터 정보공학부 교수
논문접수 : 2000년 3월 31일, 심사완료 : 2000년 5월 10일

현재 진보된 전자 상거래 시스템을 구축하기 위해 다양한 정보 기술이 활용되고 있으며, 새로운 기술들이 속속 등장하고 있다. 대부분의 전자 상거래 시스템은 조직의 사업 전략과 구현 대상에 따른 정보 기술의 선택, 사용 대상의 특성 등 모든 요소들이 결합되어 체계화된 시스템 구조로 구축된다. 일반적으로 정보 시스템은 지속적으로 발전하는 기술 변화를 수용할 수 있어야 하며, 특히 전자상거래와 같이 다수의 경제 주체가 참여하는 시스템의 경우에는 상호 운용성의 확보를 위해 포괄적인 기술구조와 세부 기술간 상관관계를 기반으로 구축되어야 한다.

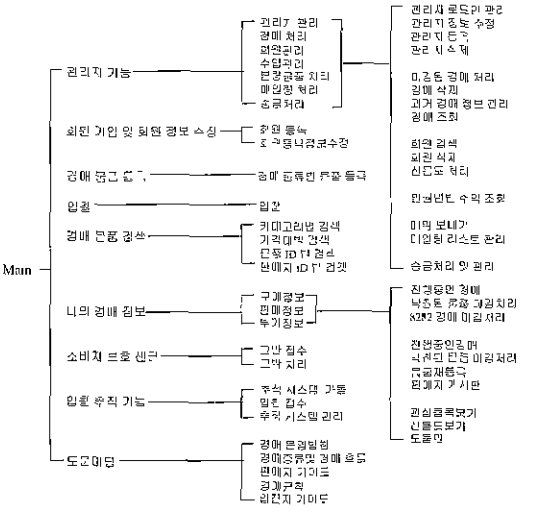
전자 상거래 시스템은 다양한 정보기술과 단위 시스템들이 통합되는 형태로 구성된다. 세부기술 및 단위 시스템간의 상관관계 측면에서의 전자 상거래 시스템은 최하위 계층인 통신 네트웍(Communication Network) 계층에서 최상위 계층인 시장 특화적 응용(Market Specific Application) 계층까지로 분류될 수 있으며 각 단계별로 새로운 기술과 응용들이 속속 개발되고 있다.

본 논문에서는 인터넷 전자 상거래 시스템 중 시장 특화적 응용의 범주에 속하는 시스템 개발에 관한 논문이다. 시장 특화적 응용 계층은 사용자가 직접 사용하는 계층으로서 사용자 인터페이스를 포함하며 전자 카탈로그, 전자 상점, 전자 경매, 전자 지불 시스템 등의 단위 시스템들이 조합되어 구성되는 계층이다. 본 논문에서 설계 구현한 인터넷 경매 시스템은 기존 판매 방식의 전자 상거래 시스템과는 달리 모든 거래가 입찰과 낙찰에 의해 이루어지고, 관심 품목에 대한 입찰 과정을 추적할 수 있는 기능을 포함하고 있는 시스템으로서 인터넷 분야에서 매우 활용성이 높은 전자상거래 시스템이다.

2. 경매 시스템의 일반기능 설계

본 논문에서 설계된 경매 시스템은 경매 기능 제공을 위한 다양한 기능을 제공하고 있다. 경매의 기능은 전자 상점 형태의 시스템과는 달리 모든 과정이 경쟁적인 입찰을 통해 물품의 거래가 이루어지고 있다. 이러한 기능들을 설계하기 위해 약 800개의 기능 프로세스와 40여개의 테이블이 설계되었다.

본 논문에서는 이러한 경매의 일반 기능 설계와 구현에 대해서는 지면상 자세하게 언급하지 않고 경매의 입찰 추적 시스템에 관해 자세하게 기술한다. 경매의 일반 기능에 관해서는 간략하게 언급하였고 시스템의 기능 나열과 테이블 설계도만 나타내었다. 본 논문에서 설계된 경매 시스템의 주요 기능을 기능/프로세스 분해도(PHD)로 (그림 1)에 나타내었다.

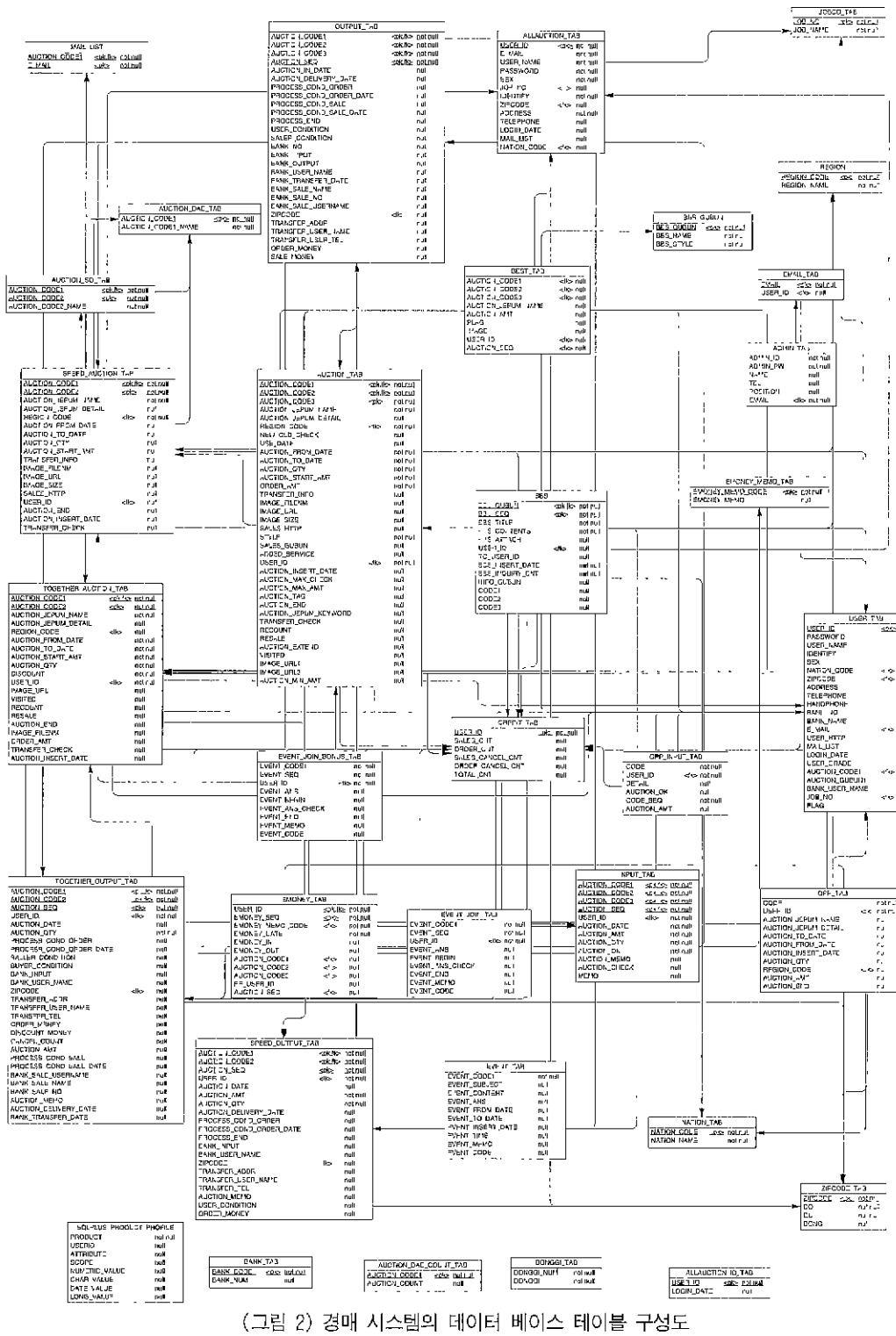


(그림 1) 경매 시스템의 주요 기능 프로세스

인터넷 경매 시스템은 모든 기능이 웹 상에서 수행된다. 특히 인터넷 경매는 일반 쇼핑몰과는 달리 사용자 사이의 물품을 경매로 중계하는 역할을 하고 있어 물품의 배송과 관련된 백엔드(Back-end) 부분을 직접 가질 수 없는 특징을 가지고 있다. 경매 시스템은 경매 사용자들이 물품을 등록하고, 등록된 물품을 입찰하고, 낙찰하고, 물품이 배송되고, 물품을 확인하고 대금이 지불되는 전 과정을 시스템에서 처리하여야 한다. 본 논문에서 개발된 시스템은 복잡한 경매를 지원하기 위해 약 800개의 단위 프로그램들로 구성되었다. (그림 2)에 개발된 경매 시스템에서 설계된 DB의 테이블 구조를 나타내었다.

3. 경매 시스템의 입찰 추적 시스템 설계

본 논문에서 개발된 경매 시스템은 입찰 추적 시스템을 가지고 있다. 경매는 특성상 치열한 입찰을 거치게 된다. 특히 인터넷 경매 시스템에서 사용하는 경매 방식은 실시간 동시 입찰 방식이 아닌 공개된 순차적 입찰 방식을 따르고 있다. 사용자는 자신이 입찰한 품목에 관심을 가지게 되고, 입찰한 품목에 대한 다른 사



(그림 2) 경매 시스템의 데이터 베이스 테이블 구성도

용자의 입찰을 쉽게 볼 수 있는 기능이 필요하게 된다.

본 논문에서 개발한 입찰 추적 시스템은 사용자가 관심 있는 품목의 입찰 추적 시스템을 가동하면 사용자의 컴퓨터 화면에 입찰 추적 시스템이 로드 되어 다른 사용자의 입찰이 발생할 경우 자동으로 통지하게 된다. 입찰추적 시스템의 클라이언트 부분은 자바 애플릿을 이용하여 설계하였다. 사용자가 관심 있는 품목에 대한 입찰 추적 시스템을 가동시키면 입찰 추적 애플릿이 서버부터 다운되어 애플릿은 서버 컴퓨터와 통신을 시작하게 된다.

3.1 애플릿과 서버와의 통신

애플릿과 서버와의 통신은 크게 2가지 형태로 구분될 수 있다. 애플릿과 서버상의 CGI 프로그램 사이에 HTTP 접속을 이용하는 방법과 서버 상에서 수행중인 비 HTTP 프로그램에 대한 원시 소켓 접속을 시도하는 방법이다. 비 HTTP 프로그램은 특정 포트를 이용하여 애플릿 프로그램과 통신하게 된다. <표 1>은 두 가지 방법의 장단점을 나타낸 것이다[3, 4, 9].

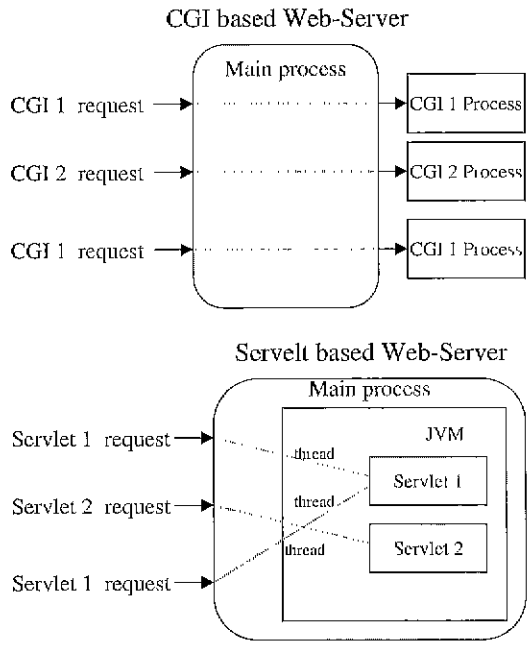
<표 1> HTTP와 비HTTP 통신의 장단점 비교

방법	장 점	단 점
HTTP	- 코드 작성이 쉽다 - 방화벽 상에서도 동작 - 다양한 서버측 프로그램 지원(Perl, C, C++ 등) - 보안 통신 허용 (HTTPS : HTTP SSL)	- 느리다(특히 CGI) - 상호 대화식 통신 불가능 - 요청이 정형화 되어야 한다 - 애플릿 만이 통신을 시작할 수 있다
비 HTTP	- 양방향 통신을 허용 - 효율성이 좋다	- 방화벽 상에서 동작 불능(원시소켓 접속 불가) - 코드 작성이 복잡하다 - 웹 검색기에 의해 접속될 수 없다

3.2 CGI와 서블릿

일반적으로 웹의 정적인 기능을 확장하기 위한 표준 방법으로 CGI(Common Gateway Interface)를 표준으로 사용하여 왔다. CGI가 다양한 언어로 작성될 수 있다는 장점이 있지만 서버에게는 많은 부담을 주는 형태로 동작한다. 서버는 클라이언트의 요청을 받으면 CGI 프로그램의 수행을 위해 각 요청별로 단독 프로세스를 생성시켜야 한다. 각 요청 클라이언트 별로 프로세스를 생성하는 것은 시간과 막대한 서버의 리소스를 요구하게 되어 서버가 동시에 처리할 수 있는 요청의 개수를 제한 할 수밖에 없게 된다. 또한 CGI는 일단 시작되면 독립적인 프로세스로서 동작하므로 웹 서버와 대화식으로 동작할 수 없다는 단점을 가지고 있다[3, 9].

자바로 작성되어 웹 서버 상에서 수행되는 서블릿은 CGI를 대체할 수 있는 방법으로 부상하고 있다. 프로세스의 생성을 요구하는 CGI와는 달리 서블릿은 웹 서버 안에서 독립적인 스레드(thread)에 의해 모든 작업이 처리된다. 서블릿은 웹 서버 내에서 동작하므로 CGI에서 불가능했던 서버와의 상호동작이 가능하고 자바 가상 머신내에서 수행되므로 이식성과 확장성이 뛰어나다. 다음 (그림 3)은 CGI와 서블릿의 라이프 사이클을 나타낸 것이다[3]



(그림 3) CGI와 서블릿의 라이프 사이클

3.3 입찰 추적을 위한 애플릿과 서블릿 설계

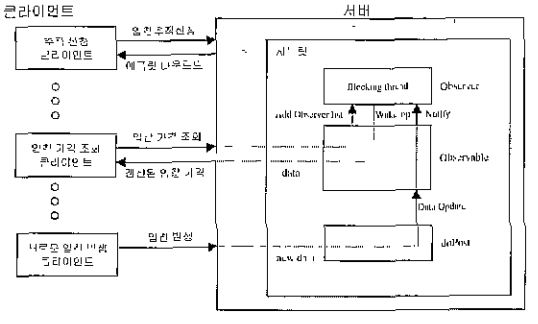
자바를 이용한 애플릿과 서블릿의 통신은 기존 CGI를 대체할 수 있는 대안이다. 서블릿은 앞에서 언급한 HTTP와 비 HTTP 통신을 지원함은 물론 서버에서 수행중인 자바 객체의 메소드를 호출할 수 있는 RMI(Remote Method Invocation) 기능을 지원한다. RMI는 애플릿과 서버가 객체지향적인 요소를 이용하여 통신할 수 있는 방법을 허용한다. 즉 필요한 객체를 서버에 저

장하고 애플릿에서 서버 객체의 메소드에 직접 접근하여 원하는 값을 얻을 수 있게 된다. 그러나 RMI는 코드의 작성이 복잡하고 지원하는 웹 검색기가 제한적이다[1, 3, 9].

본 연구에서는 인터넷 경매 시스템의 입찰 추적 시스템을 자바의 애플릿과 서블릿을 사용하여 설계하고자 한다. 앞에서 논의된 3가지의 통신방법인 HTTP, 비 HTTP, RMI 방법은 모두 서블릿에서 구현 가능하다. 그러나 RMI 방법과 비 HTTP 방법은 현재 웹의 검색기에서 지원되지 않고(넷스케이프 4.0만 지원) 방화벽을 통해서 수행될 수 없다는 단점 때문에 사용에 제한을 받고 있다. 본 연구에서는 입찰 추적 시스템을 웹 상에서 완전한 호환성을 가지고 수행될 수 있는 HTTP를 이용하여 설계하고자 한다.

HTTP를 이용한 애플릿과 서블릿의 통신으로 입찰 추적 시스템을 구현하는데는 한가지 문제점이 존재한다. 입찰 추적 애플릿은 서블릿으로부터 새로운 입찰 가격을 가져오기 위해 HTTP GET 메소드를 이용하게 된다. 그러나 입찰 애플릿은 언제 새로운 입찰이 발생하는지를 알 수 없게 되며 이러한 특성은 HTTP의 특성인 요청/응답 통신 방법의 문제점이기도 하다. 즉 항상 클라이언트가 먼저 통신을 시작해야 하고 양방향 통신을 지원하지 않는다는 문제이다[2, 3].

본 논문에서는 HTTP를 이용한 양방향 통신을 시뮬레이션 하기 위해 블록킹의 도입하여 설계하였다 본 논문에서 설계한 입찰 추적 시스템의 동작 방법을 (그림 4)에 나타내었다.

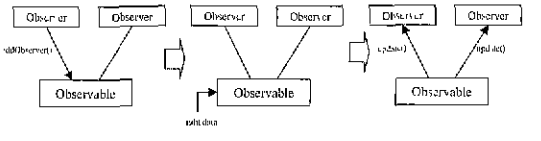


(그림 4) 입찰 추적 시스템의 동작 개요

(그림 4)의 동작 방법을 간단하게 기술한다. 추적 신청 클라이언트는 입찰 추적을 위해 서버에 입찰 추적을 신청하면, 입찰 추적 애플릿이 로드되어 클라이언트 컴퓨터에 상주하게 된다 일단 다운된 애플릿(입찰 가격조회 클라이언트)은 스레드의 기능을 가지고 무한 루프를 돌며 해당 항목의 입찰 최고 가격을 서블릿에 요청한다. 이때 서블릿은 해당 품목의 최고 가격을 데이터 베이스로부터 제공할 수 있지만 이 방법은 새로운 입찰이 발생하지 않은 경우에는 무의미하게 되며 무한 루프에서 요청되는 모든 요청을 서버가 처리해야 한다는 부담이 발생한다.

이러한 문제점을 해결하기 위해 본 논문에서는 클라이언트의 가격 요청 스레드를 블록킹하고 새로운 입찰이 발생했을 경우에 블록킹된 스레드를 깨워 갱신된 값을 반환하는 형태로 서블릿을 설계하였다. 블록킹된 스레드의 리스트와 블록킹된 스레드를 깨우기 위해 자바에서 제공하는 Observable 클래스와 Observer 클래스를 이용하였다[4, 8]. Observable 객체는 데이터를 가지는 객체이며, Observer 객체는 Observable 객체가 가지고 있는 데이터에 변화가 있는지를 모니터링 하는 객체이다. 이 두 개의 객체를 연관시켜 블록킹 문제를 해결하였다. 즉 품목 최고 가격을 가지고 있는 Observable 객체와 최고 가격 검색을 요청하는 다수개의 Observer객체를 연관시켜 Observable 객체의 값에 변화가 발생될 경우 블록킹된 스레드를 깨워 변화된 값을 반환하는 형태이다. (그림 5)는 Observable 객체와 Observer 객체가 동작하는 관계를 나타낸 것이다

이러한 형태의 설계는 서버에 많은 부담을 주지 않으면서 요청/응답의 형태로 동작하는 HTTP의 일반적인 동작을 양방향 통신이 가능한 형태로 시뮬레이션 한 것이다.



(그림 5) Observable 객체와 Observer 객체의 동작

3.3.1 애플릿 알고리즘

애플릿은 사용자의 추적 요청에 의해 클라이언트로 다운 로드된다. 애플릿이 로드될 때 애플릿 태그 매개변수를 이용하여 품목코드, 현재 가격, 입찰자 id, 입찰 화면 URL 등이 지정되어 애플릿 프로그램에서 사용하게 된다. 애플릿은 입찰 추적 GUI를 나타내며 스레드를 생성시켜 실행시킨다. 스레드는 무한 루프를 돌며

서버의 서블릿을 호출하여 지정된 품목의 최고 가격을 요청하고 서블릿에 의해 반환된 값을 애플릿에 나타내게 된다.

```
.....
class BiddingTraceApplet extends Applet implements Runnable {
 String auction_code;
 int bidding_price;
 String bidding_id;
 URL bidding_url;
 Thread r_thread;
 .....
 public void init() {
  label = new Label("입찰가격");
  text = new TextField();
  .....
 }
 public void start() {
  r_thread = new Thread(this);
  thread.start();
 }
 public void run() {
  while(true) {
    text.setText(getHighPrice(String auction_code));
  }
 }
 String getHighPrice(String args) {
  String nextHighPrice = null;
  while ( nextHighPrice == null) {
    try {
      URL url = new URL(getCodeBase(), "servlet/BiddingTrace
      Servlet" + "?" + toEncodedString(args);
      URLConnection con = url.openConnection();
      InputStream in = con.getInputStream();
      DataInputStream data = new DataInputStream (new Buffered
      InputStream(in));
      nextHighPrice = data.readLine()
        }
   catch {.....}
   return nextHighPrice;
   .....
  }
 }
}
```

3.3.2 서블릿 알고리즘

서블릿은 모두 3개의 클래스로 구성된다. 메인 클래스인 BiddingTraceServlet은 입찰가격을 요구하는 애플릿의 요청을 처리하기 위한 doGet() 메소드와 새로운 입찰이 발생했을 때 처리를 위한 doPost() 메소드로 구성된다. PriceSource 클래스는 새로운 입찰 가격을 요구하는 클라이언트가 소유하는 클래스로서 자바의 Observable 클래스로부터 상속되었다. PriceSink 클래스는 Observer 인터페이스를 포함하는 클래스로서 새로운 입찰 가격을 요구하는 클라이언트를 블록킹 시키고, 새로운 입찰이 발생하였을 때 블록킹된 스레드를 깨워 새로운 값을 반환시켜주는 기능을 제공한다. PriceSink 클래스에서 지정된 메소드는 동기(synchronized) 메소드로서 상호배제가 보장되는 메소드이다.

```
class BiddingTraceServlet extends HttpServlet {
  PriceSource p_source = new PriceSource();

  public void doGet(HttpServletRequest req, HttpServletResponse
  res) throws Exception {
   res.setContentType("text/plain");
   PrintWriter out = res.getWriter();
   out.println(getNextPrice());
    .....
  }
  public void doPost(HttpServletRequest req, HttpServletResponse
  res) throws Exception {
   String price = req.getParameter("highprice");
   if (price != null) sendPrice(price);
   .....
  }
  public String getNextPrice() {
   return new PriceSink().getNextPrice(p_source);
  }
  public void sendPrice(String price) {
   p_source.sendPrice(price);
  }
 }
 class PriceSource extends Observable {
  public void sendPrice (String price) {
   setChanged();
   notifyObservers(price);
  }
 }
 class PriceSink implements Observer {
  String price = null;
  synchronized public void update(Observable o, Object arg) {
    price = (String)arg;
    notify();
  }
  synchronized public String getNextPrice(PriceSource source) {
   source.addObserver(this);
   while (price == null) {
    try { wait(); }
    catch (Exception e) {} }
   source.deleteObserver(this);
   String highprice = price;
   price = null;
   return highprice;
  }
 }
```

4. 구현 및 활용

본 논문에서 설계된 경매 시스템은 NT서버 상에서 오라클 데이터 베이스를 이용하여 구현되었다. 경매의 일반 기능은 ASP를 사용하여 약 800본의 프로그램으로 구현되었고, 입찰 추적 시스템은 자바 애플릿과 서블릿을 이용하여 구현되었다 입찰 추적 시스템은 완전한 호환성을 제공하기 위해 HTTP 만을 사용하였고, HTTP의 일방향 통신을 양방향 통신으로 시뮬레이션하여 구현하였다.

현재 구현된 시스템은 두 곳의 인터넷 사이트(www.internet4989.co.kr, www.bankorea.com)에서 일반 물품 경매와 금융권의 금리 경매에 적용되어 운영되고 있으며 지속적으로 개선될 예정이다 본 논문에서 개발된 경매 시스템은 기존의 국내외의 많은 경매 사이트에서 운영중인 시스템과 기능 면에서 우월하다고 평가되고 있고, 특히 타 시스템에서는 제공되지 않는 입찰 추적 시스템을 최초로 제공하고 있다.

5. 결 론

본 연구에서는 인터넷의 전자상거래 분야에서 활용될 수 있는 인터넷 경매 시스템을 설계 구현하였다. 본 연구에서 개발된 경매 시스템은 경매의 일반적인 기능과 입찰추적 기능을 제공하는 시스템이다. 경매의 일반적인 기능을 위해 약 40여개의 테이블과 기능구조를 설계하였으며 약 800본의 ASP 코드로 구현하였다. 입찰 추적 기능을 위해 유용한 통신방법인 자바의 애플릿과 서블릿을 이용하였으며, 사용의 호환성을 위해 HTTP를 이용한 통신 방법을 선택하였으며, HTTP의 단절인 요청/응답의 구조를 양방향 통신 구조로 시뮬레이션하여 구현하였다.

향후 과제로는 RMI, 비HTTP 통신 방법을 사용하는 서블릿을 개발하고 사용자의 실행 환경을 자동으로 감지하여 적합한 애플릿을 제공하는 시스템의 개발과 3가지 통신모델에 대한 성능 평가가 이루어질 예정이다.

참 고 문 헌

[1] Merlin Hughes, Michael Shoffner, Derek Hamner, "Java Network Programming 2'nd Edition," MANNING, 1999.

[2] Jacob Harris, Vivek Sarkar, "Lightweight Object-Oriented Shared Variables for Distributed Applications on the Internet," OOPSLA, 1998.

[3] Alan R. Williamson, "Java Servlets by Example," MANNING, 1999

[4] Craig. Larman, Rhett. Guthrie, "Java2 performance and idiom guide," Prentice Hall PTR, 2000.

[5] Danny Ayers, Hans Bergster, "Professional Java Server Programming," Wrox Press, 1999.

[6] Elliot Rusty Harold, "Java Network Programming," O'REILLY, 1997.

[7] Alex Homer, "Professional ASP Techniques for Webmaster" WORX, 1999.

[8] Patrick Naughton, Herbert Schildt, "The Complete Referecne JAVA2," Osborn Mc Graw Hill, 1999.

[9] Jason Hunter, "Java Servelt Programming," O'REILLY. 1998.

[10] Jim Farley, "Java Distributed Computing," O'REILLY. 1999.

[11] Marry Campione, Kathy Walrath, "The Java Tutorial Second Edition Object-Oriented Programming for Internet," Addison-Wesley, 1998.

[12] David Arnow, Gerald Weiss, "Introduction to Programming Using Java An Object-Oriented Approach," Addison Wesley, 1998.

[13] John Lewis, William Loftus, "Java Software Solutions Foundations of Program Desing," Addison Wesley, 1998.

김 충 석

e-mail : cskim@silla.ac.kr

1986년 홍익대학교 전자계산학과 졸업(이학사)

1988년 홍익대학교 전자계산학과 졸업(이학석사)

1993년 홍익대학교 전자계산학과 졸업(이학박사)

1990년~현재 신라대학교(구 부산여자대학교) 컴퓨터정보공학부 부교수

관심분야 : Distributed Object Processing, Intranet Solution, OOP