

# 이동 에이전트 기반의 전자경매 시스템

김광수, 김인철  
경기대학교 전자계산학과  
e-mail : {suddol, kic}@kuic.kyonggi.ac.kr

## Electronic Auction System Based on Mobile Agents

Kwang-Su Kim, In-Cheol Kim  
Dept. of Computer Science, Kyonggi University

### 요약

컴퓨터 및 인터넷의 대중화가 급속도로 이루어지면서 전자상거래의 한 부분인 전자경매에 대한 관심과 개발이 상당부분 이루어지고 있다. 웹을 통한 전자경매의 발달로 직접 경매장에 가지 않고서도 적은 비용으로 다양한 상품들을 구매 및 판매할 수 있으므로 소비자와 공급자 모두에게 편의성과 이익을 제공해 주고 있다. 한편 에이전트에 대한 관심이 높아지면서 전자경매 시스템에 이를 적용하여 경매참여자의 편의성과 효율성을 증대시키려는 노력들도 활발히 이루어지고 있다. 대표적인 에이전트 기반 전자경매 시스템으로는 미국 워싱턴대학의 eMediator, 미시건대학의 AuctionBot, 스웨덴의 FishMarket 등이 있다. 이들 시스템들은 지원하는 경매유형, 입찰가격정책, 상품유형, 경매대상, 에이전트의 역할 및 기능등에서 상당한 차이를 보이고 있다. 본 논문에서 제안하는 전자경매 시스템은 특히 경매참여자인 사용자의 편의성과 경매관리 및 운영의 안정성을 목표로 하고 있다. 에이전트 기반의 전자경매 시스템을 개발함으로써 기본적으로 사용자 인터페이스를 통한 경매참여와 사용자가 직접 에이전트를 생성하여 이동시킴으로써 경매에 참여하는 방법을 제공한다. 두 방법 모두 참여자가 직접 경매에 참여하는 것이 아니고 에이전트를 통해 언제 어디서나 경매에 참여하여 원하는 가격대로 상품을 구매하거나 팔 수 있도록 시스템을 개발하는데 목적이 있다.

### 1. 서론

컴퓨터의 대중화와 인터넷의 발달로 전자상거래가 보편화됨에 따라 판매자나 구매자 모두 웹을 통한 상품거래가 가능해지면서 선택의 폭이 매우 넓어지고, 시간과 공간적 한계의 극복이 가능해지고 있다[1].

따라서 전자상거래의 한 부분인 전자경매도 상당한 발전을 가져오고 있다. 그러나 무수히 늘어나는 상품의 정보, 실시간으로 갱신되는 경매현황, 판매자나 구매자의 만족할 만한 경매결과 등이 전자경매를 시작함으로써 생겨나는 새로운 문제점이고 사람의 능력으로 해결하는데 한계가 있다. 이러한 문제점을 극복하기 위하여 에이전트 기반의 전자경매 시스템의 관심과 개발이 늘고 있다.

에이전트란 사람의 능력을 대신하고 지능을 가지는 소프트웨어(프로그램)이다[2].

에이전트의 종류는 여러 가지가 있으나 본 시스템에서 적용되는 이동 에이전트는 시스템의 효율성, 내고장성(fault tolerance), 이동 컴퓨팅 환경(mobile

computing environment) 등의 장점을 가질 수 있도록 개발되었다[3].

대표적인 에이전트 기반의 전자경매 시스템으로는 미국에서 개발한 워싱턴대학의 eMediator, 미시건대학의 AuctionBot, 그리고 스웨덴의 FishMarket 등이 있다. 그러나 아직 국내에서는 에이전트 기반의 시스템이 연구단계에 있다. 현재 나와있는 대부분의 전자경매 시스템은 데이터베이스를 기반으로 경매참여자의 질의(query)를 가지고 경매에 참여하고, 계속적으로 입찰을 시도할 때는 경매사이트에 지속적으로 접속해서 경매 진행 상황을 확인한 후 다시 질의를 보내야 한다.

따라서 본 논문에서는 사용자가 한 번의 입찰을 제시하면 그에 해당하는 에이전트가 자동적으로 경매에 참여하여 결과를 알려주는 에이전트 기반 전자경매 시스템을 개발하는데 목적이 있다.

2 절에서는 위에서 언급한 대표적인 에이전트 기반의 시스템을 소개하고, 3 절에서는 전자경매를 위한 경매의 기본 개념을 알아보고, 4 절에서는 본 시스템

의 설계와 구성을 소개하고, 5 절에서는 구현에 관한 설명으로 데이터베이스 스키마(schema)와 각각의 에이전트를 정의하고 그에 해당되는 pseudo code 를 보여준다. 마지막으로 6 절에서는 결론과 향후 연구과제를 소개하고자 한다.

## 2. 관련연구

국내에서 에이전트 기반 전자경매 시스템에 관한 연구는 아직 초보적인 수준에 머물러 있지만 공개된 시스템으로는 숭실대학교의 AAAS(Autonomous Auction Agent System)가 있다. 이 시스템은 미국 미시건대학의 AuctionBot 을 기반으로 개발되었으며 가격입찰에서 일반적인 입찰과 적극적인 입찰을 경매참여자가 선택할 수 있는 새로운 정책을 제시했다[4].

본 논문의 기반이 되는 eMediator 는 미국 워싱턴 대학에서 약 2 년 여에 걸쳐 개발한 시스템으로 Java 언어로 구현되었으며 세가지 component 로 구성되어 있다.

먼저, 다양한 경매유형(auction type)과 입찰방식(bid type)을 결정하고, 경매참가자가 5 가지의 이동 에이전트 중 하나를 선택할 수 있는 eAuctionHouse 가 있다. 두 번째는 인공지능 기법과 게임이론(game theory)을 이용하여 최적의 낙찰가격을 결정하는 Leveled commitment contract optimizer 가 있고, 마지막으로 낙찰가격이 결정된 후 안전하게 상품교환이 이루어지도록 하는 eExchangeHouse 가 있다.[5].

미시건대학에서 개발한 AuctionBot 은 경매참가자를 위한 web interface 부분과 에이전트를 위한 TCP/IP interface 로 나누어진다. 데이터베이스를 모니터링하는 scheduler 가 있고, 경매가 이루어지는 auctioneer 가 있다. C 언어로 구현되었으며 다양한 경매유형을 제공하고 있다[6].

마지막으로 스웨덴에서 개발한 FishMarket 은 Java 언어로 구현되었으며, round-robin 방식을 기반으로 downward-bidding protocol 을 이용하여 가격결정을 한다. 에이전트간 통신언어로 KQML (Knowledge Query Manipulation Language)을 채용하고, 에이전트의 증가로 인한 네트워크 부하를 줄이기 위해 PVM(Parallel Virtual Machine)을 사용한다. 그리고 사용자의 신용상태를 파악하여 경매참여를 결정한다[7].

## 3. 경매

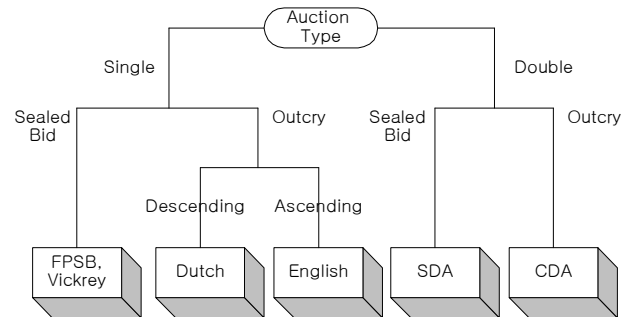
1987년 McAfee 와 McMillan 가 **경매**를 ‘경매는 입찰(bid)을 기반으로 결정되는 가격과, 자원의 배당(allocation)이 결정되는 규칙들로 이루어진 대체 시장이다.’ 라고 정의하였다[8].

**경매의 유형**은 크게 판매자와 구매자가 1:m, m:1, m:m 으로 구분되고[1], 입찰가격은 봉인되어 비공개적인 Sealed Bid 와 공개적인 outcry 가 있고[4], 가격 결정 순서에 따라 오름차순인 ascending 과 내림차순인 descending 이 있다.

**경매의 종류**로는 비공개 입찰(Sealed Bid)방식으로 가장 높은 가격을 결정하는 FPSB(First Price Sealed

Bid)와 비공개 입찰 방식인데 가격결정은 두번째로 높게 제시한 가격으로 결정하는 Vickrey(Second Price Sealed Bid)가 있다[8]. 그리고 가장 높은 가격에 낙찰되는 English 와 높은 가격에 판매자가 가격을 제시하여 구매자가 구매의사를 밝힐 때 까지 가격을 낮춰가는 Dutch 방식이 있고, 판매자와 구매자간에 원하는 가격을 동시에 비공개적으로 제시하는 Call Market(Sealed Double Auction)이 있고, 증권회사와 같이 판매자와 구매자가 계속적으로 가격을 관찰하면서 실시간으로 가격을 제시하고 경매를 하는 CDA(Continuous Double Auction)가 있다[1].

[그림 1] 경매의 유형



## 4. 시스템 설계

### 4.1 설계 목표

본 논문의 전자경매 시스템은 미국 워싱턴대학에서 개발한 eMediator 를 모델로 개발되었다. 많은 경매유형 중 English 방식과 Dutch 방식을 사용하고 입찰취소 규칙(bid cancellation rule)과 경매취소 규칙(Decommitment rule), 그리고 최종 낙찰가격이 동일한 경우에 대한 처리(tie breaking rule)에 관한 사항을 경매참여자가 선택(option)할 수 있도록 하였다. 그리고 경매의 시작과 끝은 선택되어진 시간에 의해 결정되도록 하였다.

### 4.2 시스템 구성

경매참여자는 두 가지 방법으로 접근할 수 있다. 먼저 사용자 인터페이스를 작성하여 HTTP Server 를 거쳐 CGI 로 경매 서버에 접근하면 자동으로 에이전트가 생성되어 경매에 참여하고, 다른 방법은 이동 컴퓨팅 환경에서도 가능하도록 먼저 이동에이전트를 생성하여 ATP(Agent Transfer Protocol)를 통해 경매 서버에 보내면 경매참여자는 네트워크 상태를 offline 으로 해도 이동에이전트에 의해 경매는 계속 되어진다.

사용자등록과 인증과정은 데이터베이스에 저장된 데이터를 모니터링하는 SupervisorAgent 가 인증여부를 결정한다. 인증과정이 끝나고 경매에 참여하는 사용자가 생기면 SupervisorAgent 가 사용자에게 관련 에이전트를 생성하고 AuctionHouse 에서 경매가 이루어진다.

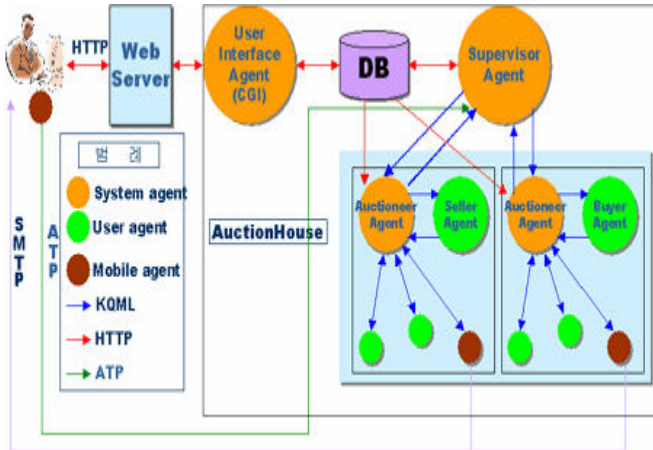
경매의 시작과 끝은 SupervisorAgent 가 관리하고, 경매는 AuctioneerAgent 가 책임진다. 여기에서 모든

정보는 DataBase 에 저장되고 낙찰가격 또는 진행상황은 에이전트에 의해 다시 사용자에게 e-mail 로 결과를 알려준다.

이동 에이전트 개발환경으로 플랫폼 독립적인 Java 언어로 쓰여진 IBM Aglets WorkBench[9]가 사용된다.[3][9].

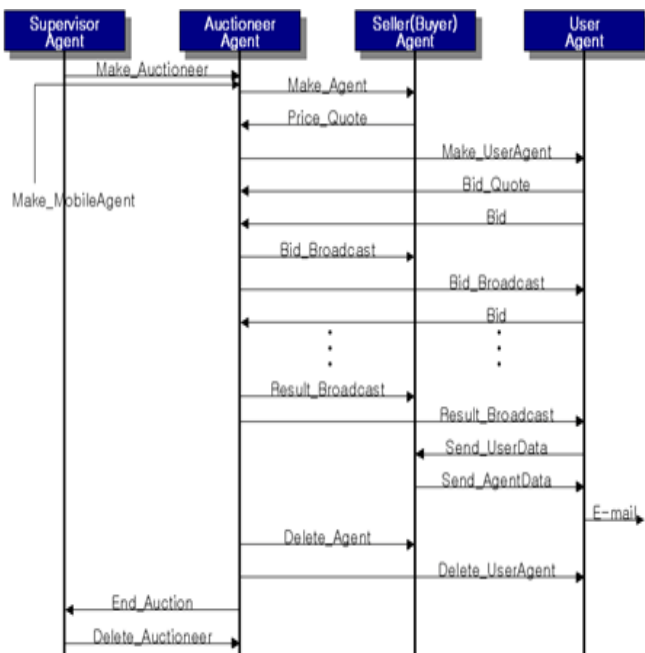
AuctionHouse 내에서 에이전트간의 통신이 이루어지는데 이때 사용되는 프로토콜은 KQML(Knowledge Query Manipulation Language)이고 에이전트가 사용자에게 경매에 관한 결과를 e-mail 로 보내기 위한 프로토콜로써 SMTP(Send Mail Transfer Protocol)를 사용한다.

AuctionHouse 에서 AuctioneerAgent 가 경매를 책임지고 관리하면서 경매가 이루어지고 가격이 결정되면 Seller(Buyer)Agent 와 가격을 제시한 사용자 에이전트(User Agent)에게 서로 결과를 알려주고, 두 에이전트간에 통신을 통하여 거래가 이루어지도록 한다.



[그림 2] 시스템 구조

### 4.3 에이전트간 협상 프로토콜



[그림 3] 에이전트간 협상 프로토콜

AuctionHouse 를 중심으로 외부에 있는 SuperVisorAgent 와 내부에 있는 AuctioneerAgent, Seller(Buyer)Agent, UserAgent 는 서로간에 에이전트간 통신언어인 KQML 를 채용하고 있다.

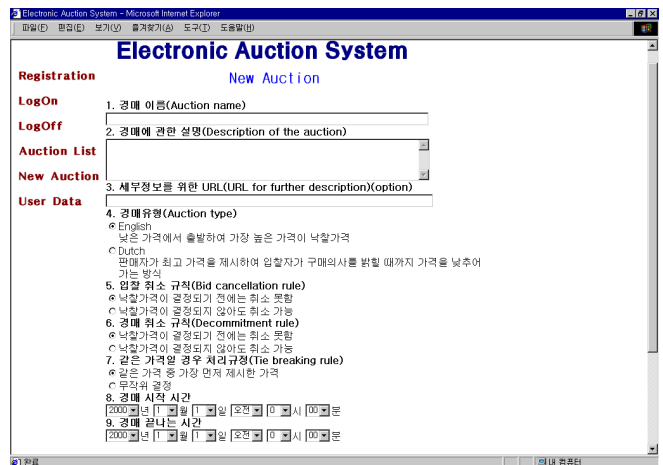
새로운 경매를 생성할 경우 사용자 인터페이스에서 기본적으로 제공하는 각각의 규칙은 입찰취소와 경매취소를 하지 못하는 경우에 [그림 3]과 같은 메시지 전달에 의해 경매가 이루어진다.

## 5. 구현

본 시스템은 Window NT server4.0 을 사용하고 있으며 Java 언어(jdk1.1.7)로 구현하였으며, 데이터베이스는 Oracle server 8.0 을 사용하고 이동 에이전트 개발환경인 IBM AWB(Aglets WorkBench)1.1beta2 를 사용하고 있다.

### 5.1 사용자 인터페이스

여러 가지 사용자 인터페이스 중 경매참여자가 새로운 경매를 생성 할 경우의 CGI 인터페이스를 [그림 4]에서 보여주고 있다. 해당 폼(form)을 완료하면 경매번호(auction ID)는 자동으로 DataBase 에서 부여하게 된다.



[그림 4] 사용자 인터페이스

### 5.2 데이터베이스

상용 데이터베이스인 Oracle server 8.0 을 사용하고 있는 본 시스템은 4 개의 테이블로 구성되어 있다. 각 테이블의 데이터베이스 스키마(schema)는 다음과 같다.

User = [User\_ID, User\_Password, User\_Name, E-mail, URL]

Auction = [Auction\_ID, Auction\_Name, Description, Auction\_Type, Bid\_Cancel, Decommitment, Tie\_Break, StartTime, End\_Time]

Bid = [User\_ID, Auction\_ID, Price\_Quote, Bid\_Quote]

Good = [User\_ID, Auction\_ID]

### 5.3 감독자 에이전트(SupervisorAgent)

감독자 에이전트(SupervisorAgent)는 인증(authentication)을 확인하고, 경매의 시작과 끝을 담당한다. 즉, AuctioneerAgent 를 생성함으로써 경매가 시작되고, 최종적으로 AuctioneerAgent 를 소멸함으로써 경매를 닫는다.

```

If Db_Monitor(User_Table) //인증
  If (UserTableForm=User_Table)
    print( "등록되었습니다.")
  then
    print( "다시 입력하세요" )

If Db_Monitor(User_Table, Auction_Table, Good_
  Table, Bid_Table)
  If (AuctionForm= User_Table, Auction_Table,
    Good_Table, Bid_Table)
    Send(AuctioneerAgent, "Make_Auctioneer" )
    .
    .
    Receive(AuctioneerAgent, "End_Auction")
    Send(AuctioneerAgent, "Delete_Auctioneer
      ")
  then print( "각 양식을 확인 후 다시 입력하
    세요" )

```

### 5.4 경매인 에이전트(AuctioneerAgent)

하나의 경매가 시작되면 생성되는 에이전트으로써 실질적으로 경매를 관리하고, Seller(Buyer)Agent 를 생성하고 UserAgent 의 입찰(bid)을 바탕으로 낙찰가격을 결정한 후 경매를 닫기 위하여 Seller(Buyer)Agent 와 UserAgent 를 삭제한다.

```

Generate(SupervisorAgent, "Make_Auctioneer")
Begin
  Send(Seller(Buyer)Agent, "Make_Agent" )
  Receive(Seller(Buyer)Agent, "Price_Quote"
    )
  Send(UserAgent, "Make_UserAgent" )
  Receive(Seller(Buyer)Agent, "Bid_Quote" )
  Send(Seller(Buyer)Agent, "Bid_Broadcast" )
  Broadcast(UserAgent, "Bid_Broadcast" )
  .
  .
  Broadcast(Seller(Buyer)Agent, "Result_Bro-
    adcast" )
  Broadcast(UserAgent, "Result_Broadcast" )
  Delete(Seller(Buyer)Agent, "Delete_Agent"
    )
  Delete(UserAgent, "Delete_UserAgent" )
End
Delete(SupervisorAgent, "Delete_Auctioneer")

```

### 5.5 정보 및 입찰 에이전트(UserAgent)

경매에 참여하여 입찰(bid)을 하는 에이전트으로써 하나의 경매에는 AuctioneerAgent 가 생성하는 에이전

트가 있고, 사용자가 이동에이전트를 보내서 생성되는 에이전트가 있다. 그러나 AuctionHouse 내에서의 역할은 같다. 낙찰가격이 결정되면 Seller(Buyer)Agent 와 실질적으로 상품거래를 위한 통신을 하고 사용자에게 e-mail 로 알려준다.

```

Generate(AuctioneerAgent, "Make_UserAgent" )
Begin
  Send(AuctioneerAgent, "Bid_Quote" )
  Send(AuctioneerAgent, "Bid" )
  Receive(AuctioneerAgent, "Bid_Broadcast
    ")
  .
  .
  Receive(AuctioneerAgent, "Result_Broadc-
    ast" )
  Send(Seller(Buyer)Agent, "Sent_Userdata")
  Receive(Seller(Buyer)Agent, "Sen_Agentd-
    ata")
  Mail_Transfer(E-mail)
End
Delete(AuctioneerAgent, "Delete_UserAgent")

```

## 6. 결론 및 향후과제

본 시스템의 가장 큰 특징은 일반적인 CGI 인터페이스를 통한 경매서버의 접근뿐만 아니라 사용자가 직접 하나의 에이전트를 생성하여 경매에 참여할 수 있도록 설계되었다. 이로 인하여 사용자는 한번의 경매참여로 offline 상태에서도 경매에 참여할 수 있고 실질적으로 경매에는 에이전트가 참여하게 되어 경매 참여자에게 편의성과 효율성을 제공하게 되었다.

향후과제로는 현재는 경매유형과 입찰유형이 제한되어 있으나 앞으로 다양한 경매와 입찰유형을 제공하고, 경매참여자의 관계도 현재는 1:1 로 한정되어 있지만 이후에는 m:m 까지 확장하고 이동에이전트에 대한 더욱 안전한 보안(security)체계를 유지하는데 있다.

## 참고문헌

- [1]. 이재규, 최형림, 김현수, 이경진 "전자상거래원론" 제 5 장 사이버경매, pp.158-181. 1999
- [2]. 이경진 "전자상거래 소프트웨어 에이전트". 정보처리학회지 제 6 권 제 1 호, pp.54-62. 1999
- [3]. 석황희, 김인철 "이동에이전트의 개념과 응용" 멀티미디어학회지
- [4]. 이종희, 이용준, 김태석, 오해석 "전자상거래 인터넷 경매를 위한 지능적 경매 에이전트 시스템 구현". 정보과학회 가을 학술발표논문집 Vol. 26, No.2. pp98-99. 1999
- [5]. Tuomas Sandholm "eMediator : A Next Generation Electronic Commerce Server". AAAI Workshop on AI in Electronic Commerce, Orlando, FL, July. AAAI Workshop Technical Report pp.46-55. 1999
- [6]. Peter R. Wurman, Michael P. Wellman and William E. Walsh. "The Michigan Internet AuctionBot : A Configurable Auction Server for Human and Software Agents". In Second

International Conference on Autonomous Agents, pp.301-308,  
Minneapolis. 1998

[7]. <http://www.iiia.csic.es/Projects/fishmarket/foundation.html>

[8]. <http://www.agorics.com/auction/auction8.html>

[9]. <http://www.trl.ibm.co.jp/aglets/JAAPI-whitepaper.html>