

# 에이전트 기반의 연속다중경매 시스템

김광수<sup>0</sup> 김인철  
경기대학교 전자계산학과  
{suddol, kic}@kuic.kyonggi.ac.kr

## Agent - Based Continuous Double Auction System

Kwang-Su Kim<sup>0</sup> In-Cheol Kim  
Dept. of Computer Science, Kyonggi University

### 요 약

현재 대부분의 전자경매 시스템들은 기존의 경매소와 경매인들에 의해 이루어졌던 경매진행과 관리기능을 하나의 서버프로그램으로 통합하여 제공하고 있다. 하지만 이러한 시스템들은 기존 경매소에서 이루어졌던 기능들의 독립성과 분산성 그리고 병렬성을 충분히 구현하지 못하고 있다. 또한 기존의 이러한 시스템들은 단지 경매관리와 진행만을 온라인상에서 자동화 하였을 뿐 경매는 여전히 경매참여자들이 원격접속을 통해 직접 입찰하는 것을 요구하고 있다. 따라서 경매참여자는 수시로 변하는 경매 진행상황을 파악하고 적절한 입찰가격을 결정하여 입찰하는 반복적인 작업을 계속해야 한다. 그리고 기존 시스템들이 갖는 또 다른 제한으로는 다양한 경매 유형을 지원하지 못한다는 것이다. 현재 대부분의 전자경매 시스템에서 지원하고 있는 경매유형은 영국식 또는 네델란드식을 포함해 단일경매가 대부분이며, 속도와 효율성 그리고 경매참여자의 이익면에서 우수한 다중경매를 지원하지 못하고 있다. 이러한 문제점들을 해결하기 위하여 본 논문에서는 다중 에이전트 기반의 연속다중경매 시스템인 CoDABot을 설계하고 구현하였다. 이 시스템에서는 연속다중경매를 지원하고 경매참여자인 사용자를 대신해 입찰할 수 있는 다양한 유형의 입찰 에이전트를 제공한다. 따라서 경매관리와 진행의 효율성을 위해 사용자 인터페이스 에이전트, 경매인 에이전트 그리고 다수의 입찰 에이전트들로 구성된 다중 에이전트 시스템으로 전체 시스템을 구현하였다.

### 1. 서론

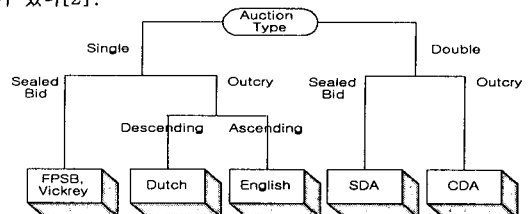
컴퓨터의 대중화와 인터넷의 발달로 전자상거래가 보편화됨에 따라 판매자나 구매자 모두 인터넷을 통한 온라인 상품거래가 가능해지면서 시간과 공간의 제한을 받지 않고 자유로운 거래가 가능해 졌으며 상품선택의 폭이 매우 넓어졌다. 따라서 전자상거래의 한 형태인 전자경매 역시 경매참여자가 직접 경매소에 가지 않고 적은 비용으로 다양한 상품들을 구매하고 판매할 수 있어 경제성과 효율성이 높고 온라인상에서 보다 편리한 부가 사용자 서비스를 제공할 수 있다는 장점이 있다. 그러나 현재 대부분의 전자경매 시스템들은 기존의 경매소와 경매인들에 의해 이루어졌던 경매진행과 관리기능을 하나의 서버프로그램(server program)으로 통합하여 제공하고 있다. 하지만 이러한 시스템들은 기존 경매소에서 이루어졌던 기능들의 독립성과 분산성 그리고 병렬성을 충분히 구현하지 못하고 있다. 또한 기존의 이러한 시스템들은 단지 경매관리와 진행만을 온라인상에서 자동화 하였을 뿐 경매는 여전히 경매참여자들이 원격접속을 통해 직접 입찰하는 것을 요구하고 있다. 따라서 경매참여자는 수시로 변하는 경매 진행상황을 파악하고 적절한 입찰가격을 결정하여 입찰하는 반복적인 작업을 계속해야 한다. 그리고 기존 전자경매 시스템들이 갖는 또 다른 제한으로는 다양한 경매유형을 지원하지 못한다는 것이다. 현재 대부분의 전자경매 시스템에서 지원하고 있는 경매유형은 영국식 또는 네델란드식을 포함해 단일경매가 대부분이며, 속도와 효율성 그리고 경매참여자의 이익면에서 우수한 다중경매를 지원하지 못하고 있다.

이러한 문제점들을 해결하기 위하여 본 논문에서는 다중 에이전트 기반의 연속다중경매 시스템인 CoDABot(Continuous

Double AuctionBot)을 설계하고 구현하였다. 이 시스템에서는 연속다중경매를 지원하고 경매참여자인 사용자를 대신해 입찰할 수 있는 다양한 유형의 입찰 에이전트를 제공할 뿐 아니라 경매관리와 진행의 효율성을 위해 사용자 인터페이스 에이전트, 경매인 에이전트 그리고 다수의 입찰 에이전트들로 구성된 다중 에이전트 시스템으로 전체 시스템을 구현하였다.

### 2. 관련연구

일반적으로 경매유형은 [그림 1]과 같은 크게 단일경매(single auction)와 다중경매(double auction)로 나뉘어 지고, 입찰방식에 따라 비공개입찰(sealed bid)방식과 공개입찰(outcry)방식이 있다. 그리고 입찰가격이 올라가느냐 내려가느냐에 따라 오름차순(ascending)방식과 내림차순(descending)방식이 있다[2].



[그림 1] 경매의 유형

그리고 대표적인 에이전트 기반 전자경매 시스템으로는 AuctionBot, eMediator 그리고 FishMarket 이 있다. 먼저, 미시간대학의 AuctionBot은 세가지 에이전트 유형을 제공하고 Mth

가격결정 알고리즘을 사용한다. 경매참여자를 위한 웹 인터페이스와 에이전트를 위한 TCP/IP 인터페이스로 나누어진다. 데이터베이스를 모니터링하는 스케줄러가 있고, 경매를 관리하는 경매인(auctioneer)이 있다. 비공개 입찰방식으로 주기적 다중경매(Periodic Double Auction)와 영국식 경매 그리고 Vickrey 경매를 지원하고 있다[2]. 두 번째로 워싱턴 대학의 eMediator는 여러 개의 상품을 묶어서 날개 또는 조합으로 거래가 이루어지는 조합 경매(combinatorial auction)를 지원하고 세 가지 구성요소가 있다. 다양한 경매유형과 입찰방식을 결정하고, 다섯 가지 에이전트 중 하나를 선택할 수 있는 eAuctionhouse, 인공지능 기법과 게임이론을 이용하여 최적의 낙찰가격을 결정하는 Leveled commitment contract optimizer, 그리고 안전하게 상품교환이 이루어지도록 하는 eExchange-House가 있다[4]. 마지막으로 스웨덴 인공지능연구소에서 개발한 FishMarket은 round-robin방식으로 downward bidding protocol을 이용하여 가격결정을 한다. 즉, 판매자의 상품을 다수의 구매자가 입찰을 하므로 초기 가격보다 가격이 내려가게 된다. 에이전트간 통신언어로 KQML(Knowledge Query Manipulation Language)을 사용하고, 에이전트의 증가로 인한 네트워크 부하를 줄이기 위해 PVM(Parallel Virtual Machine)을 사용한다[5].

3. 시스템 설계

3.1 연속다중경매(Continuous Double Auction)

단일경매는 한 명의 판매자가 제시한 하나의 상품과 최소 요구가격(minimum ask price)에 대해 다수의 구매자들이 경쟁 입찰을 통해 한 명의 최종 낙찰자와 낙찰가가 결정된다. 판매자와 구매자의 구성이 반대인 경우도 단일경매에 속한다. 이러한 단일경매에서는 다수의 경쟁 입찰자들 보다는 주로 한 명의 판매자나 구매자의 이익이 중심이 되며, 한 번에 하나의 상품에 대한 한 명의 낙찰자만을 결정하는 방식으로 경매 진행 속도와 효율성이 낮고 경쟁이 치열하다.

주기적 다중경매(PDA)는 다수의 판매자와 구매자들이 함께 참여하는 비공개 경쟁입찰 방식이다. 이 경매유형은 일정 시간동안 판매자와 구매자 모두에게서 입찰을 받아서 한번에 다수의 낙찰자들을 결정한다. 낙찰자와 낙찰가격 결정을 위해 Mth 가격 알고리즘을 사용하고 있다. 여기서 M은 판매자들의 수이다. 따라서 모든 입찰을 한 번에 받아서 M번째로 높은 가격이 낙찰가격으로 결정된다[2].

연속다중경매는 주기적 다중경매와는 달리 모든 판매자와 구매자의 입찰가격이 즉시 공개되는 공개입찰방식이며 최저 요구가격을 제시한 판매자와 최고 제한가격을 제시한 구매자간에 낙찰이 이루어지고, 이들을 제외한 나머지 경매참여자들간의 경쟁입찰은 계속되어 지속적으로 후속 낙찰자들을 결정해나가는 방식이다. 따라서 연속다중경매는 단일경매나 주기적 다중경매에 비해 진행속도와 효율성면에서 더 우수하고 판매자와 구매자 모두의 이익을 보다 공정하게 보호할 수 있는 장점이 있다.[2][3]. 연속다중경매의 낙찰가격은 시세(quote)를 기준으로 결정되고 판매자는 가격을 점차적으로 낮게 입찰(ask)하고 구매자는 높게 입찰(bid)하여 결정된다. 항상 판매자의 시세가 구매자의 시세보다 큰 상태를 유지하고 아래 [식 1]과 같이 가격이 서로 겹치는(overlap)되는 경우 낙찰가격이 생성된다[3].

- $Q^s$  : 판매자의 시세
- $Q^b$  : 구매자의 시세

$$Q^b \geq Q^s \quad [식 1]$$

3.2 시스템 구조(System Architecture)

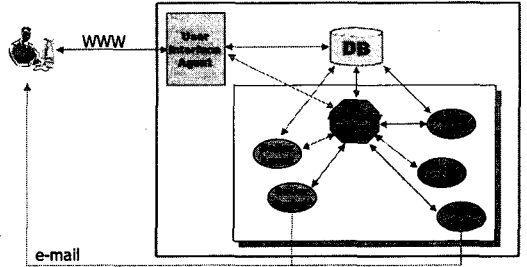
본 시스템인 CoDABot은 모든 구성요소가 에이전트로 구성되어 있고 메시지(message)를 기반으로 통신을 한다. 경매참여자는 인터넷을 통하여 사용자 인터페이스 에이전트

(user interface agent)로부터 정보를 주고 받는다.

사용자 인터페이스 에이전트는 데이터베이스에 자료를 저장하고 경매인 에이전트(auctioneer agent)를 생성하고 경매의 시작을 메시지로 알려준다.

경매인 에이전트는 경매참여자를 대신하는 입찰 에이전트(bidding agent)를 생성하고 경매를 관리한다. 그리고 입찰 에이전트의 입찰을 받아 낙찰가격과 낙찰자를 결정한다.

입찰 에이전트(bidding agent)는 입찰을 하기 위하여 제한된 범위 내에서 데이터베이스를 검색하고 경매인 에이전트에게 입찰을 제시한다. 경매인 에이전트에게 결과를 통보 받아 경매참여자에게 e-mail로 결과를 알린다.



[그림 2] 시스템 구조

3.3 입찰 에이전트 유형(Bidding Agent Type)

CoDABot에서는 서로 다른 입찰전략을 가진 다섯 가지 유형의 에이전트를 제공하며 경매참여자는 이들 중 하나를 선택함으로써 경매에 참여한다.

1. 단순 입찰 에이전트(Simple Bidding Agent) : 에이전트가 어떠한 정보도 이용하지 않고 경매참여자가 직접 입찰가를 제시하는 에이전트. 웹 인터페이스를 통해 경매참여자가 제시하는 입찰가를 대신 경매인 에이전트에게 전달하는 에이전트

2. 증가/감소 입찰 에이전트(Incremental/Decremental Bidding Agent) : 현재 시세(quote)를 기준으로 자신의 제한 가격(limit price)까지 일정 양 만큼 입찰가격을 올리거나 내려서 결정하는 에이전트

- $P_{limit}$  : 제한가격(limit price)
- $Q^b$  : 구매자 현재시세, □  $Q^s$  : 판매자 현재시세,
- $\alpha$  : 경매참여자 정의 백분율(%)

$$P_b = \min \left( Q^b + \frac{P_{limit} - Q^b}{100} \alpha, P_{limit} \right) \quad (\text{구매자})$$

$$P_b = \max \left( Q^s - \frac{P_{limit} - Q^s}{100} \alpha, P_{limit} \right) \quad (\text{판매자}) \quad [식 2]$$

3. 목표위주 입찰 에이전트(Target-Oriented Bidding Agent) : 경매에 올라온 상품들 중 선택된 특정 상품의 판매 혹은 구매를 목표로 다른 에이전트의 입찰가는 고려하지 않고 그 상품을 제시한 에이전트의 입찰가격을 기준으로 자신의 입찰가를 결정하는 에이전트

- $P_b$  : 입찰 예상 가격, □  $P_b'$  : 자신의 최근 입찰가격
- $P_{target}$  : 목표가 되는 에이전트의 입찰 가격

$$P_b = \min \left( P_b' + \frac{P_{target} - P_b'}{100} \alpha, P_{limit} \right) \quad (\text{구매자})$$

$$P_b = \max \left( P_b' - \frac{P_{target} - P_b'}{100} \alpha, P_{limit} \right) \quad (\text{판매자}) \quad [식 3]$$

4. 낙찰가 모델링 입찰 에이전트(Clearing-Price Modeling Bidding Agent) : 기존 낙찰가격의 평균값으로 낙찰가격을 모델링하여 현재 시세보다 일정 양 만큼 올리거나 내려 자신의 입찰가격을 결정하는 에이전트

□  $CP_{average}$  : 평균 낙찰가격

$$P_b = \min \left( Q^b + \frac{CP_{average} - Q^b}{100} \alpha, P_{limit} \right) \quad (\text{구매자}) \quad [식 4]$$

$$P_b = \max \left( Q^s - \frac{CP_{average} - Q^s}{100} \alpha, P_{limit} \right) \quad (\text{판매자})$$

5. 시간 제한 입찰 에이전트(Time-Bound Bidding Agent) : 낙찰가를 모델링할 뿐 아니라 에이전트의 유효시간에 따라 입찰 가격이 변하는 에이전트. 유효시간이 임박했음에도 낙찰이 결정되지 않았을 경우 입찰가격 폭이 커지는 에이전트

□  $V_t$  : 유효시간

□  $P_t$  : 현재시간

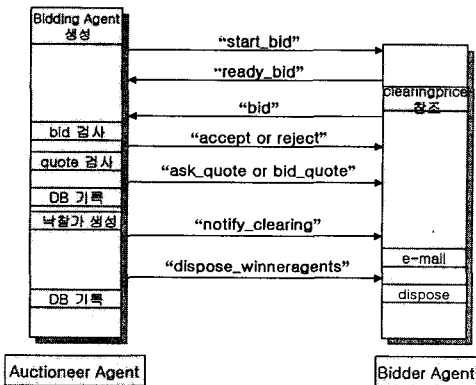
$$P_b = \min \left( Q^b + \frac{CP_{average} - Q^b}{100(V_t - P_t)} \alpha, P_{limit} \right) \quad (\text{구매자})$$

$$P_b = \max \left( Q^s - \frac{CP_{average} - Q^s}{100(V_t - P_t)} \alpha, P_{limit} \right) \quad (\text{판매자}) \quad [식 5]$$

### 3.4 에이전트 상호작용(Interaction between Agents)

사용자가 웹 인터페이스를 통해 요청하는 사용자등록, 로그인, 경매생성, 경매참여 및 탈퇴, 입찰 에이전트 생성, 입찰 현황 모니터링, 과거 낙찰가 조회등 다양한 서비스들이 시스템 내부의 사용자 인터페이스 에이전트, 경매인 에이전트 그리고 다수의 입찰 에이전트들간의 메시지 통신을 이용한 상호작용의 결과로 제공된다.

[그림 3]은 경매가 시작되어 한 번의 낙찰이 이루어질 때까지 경매인 에이전트와 입찰 에이전트들간의 상호작용만 일부 보여주고 있다. 먼저 경매인 에이전트가 입찰 에이전트들에게 경매시작을 알리는 start\_bid를 보내면 입찰 에이전트는 회신으로 ready\_bid를 보내고 입찰가격인 bid를 보낸다. bid가 규칙에 맞으면 accept 메시지를 보내고 그렇지 않으면 reject 메시지를 보낸다. 낙찰이 결정되면 경매인 에이전트는 모든 에이전트에게 notify\_clearing를 보낸다. 여기에는 낙찰된 에이전트와 낙찰가격이 포함되어 있다. 그리고 낙찰된 에이전트를 소멸하기위해 dispose\_winneragents 메시지를 보내면 이 메시지를 받은 낙찰 에이전트는 스스로 소멸된다.



[그림 3] 입찰을 위한 에이전트간의 상호작용

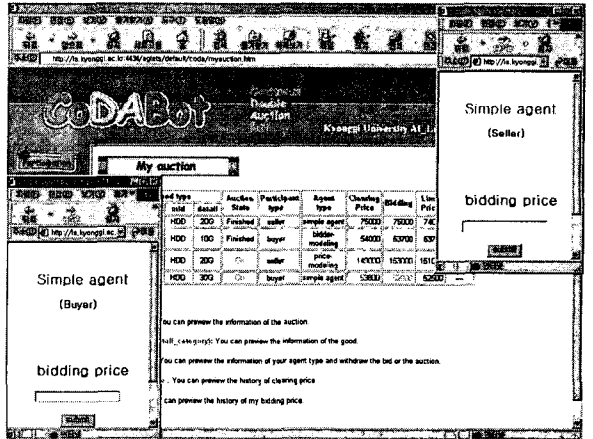
### 4. 구현

CoDABot은 Unix HP machine의 HP\_UX 10.20을 사용하여 Java 언어(jdk1.1.7)로 구현하고, 데이터베이스는 Oracle server 8.0을 사용하고 에이전트 개발환경인 IBM AWB ( Aglets Work-Bench)1.1beta3을 사용하였다. 데이터베이스는 사용자, 경매 참여자, 상품유형, 상품(판매자), 상품요구(구매자), 에이전트,

경매, 입찰가, 낙찰 테이블 등 전체 9 개의 테이블로 구성되었

다. 특히 경매참여자, 경매, 에이전트, 상품(판매자), 상품요구(구매자) 입찰가 테이블에는 활성과 여부를 확인하는 active\_flag 필드가 있다. 이 필드의 역할은 CoDABot 이 연속다중경매를 지원하고 입찰가격을 에이전트가 결정하므로 경매참여자와 에이전트유형에 따라 과거 낙찰가격, 입찰가격, 시세가격 등 필요한 자료들을 제공하기 위한 방법이다.

[그림 4]는 사용자가 동시에 서로 다른 두 경매에서 각각 판매자와 구매자로써 단순 입찰 에이전트를 이용할 때 사용자 인터페이스와 단순 입찰 에이전트들의 인터페이스를 보여주고 있다. 사용자 인터페이스에서는 경매 참여자의 현재 경매 상황 뿐만 아니라 과거 참여했던 경매의 모든 정보를 보여주고 있다.



[그림 4] CoDABot의 실행화면

### 5. 결론 및 향후 연구과제

본 논문에서는 에이전트 기반의 연속다중경매 시스템인 CoDABot을 설계하고 구현하였다. 이 시스템은 연속다중경매를 지원하며 경매참여자를 대신해 경매에 참여하는 다양한 입찰 에이전트를 제공하고 있다. 또한 경매관리와 진행의 효율성을 위해 사용자 인터페이스 에이전트, 경매인 에이전트, 다수의 입찰 에이전트 등 다중 에이전트 시스템으로 구현하였다.

향후과제는 에이전트 유형에 인공지능 기법을 도입하여 더욱 정형화되고 경매참여자의 성향을 정확히 반영할 수 있는 에이전트를 개발하고 낙찰이 이루어진 후 안정적인 상품교환을 가능하도록 하는데 있다

### 6. 참고문헌

- [1] 이경전 " 전자상거래 소프트웨어 에이전트 ", 정보처리회지 제6권 제1호, pp.54-62. 1999
- [2] 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
- [3] Peter R. Wurman, William E. Walsh and Michael P. Wellman, " Flexible Double Auctions for Electronic Commerce: Theory and Implementation ", submitted for publication
- [4] 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
- [5] <http://www.iiia.cs.ic.es/Projects/fishmarket/foundation.html>