

## 마켓 시스템에서 거래를 위한 브로커 기반 동기화 거래 알고리즘\*

강남오\*\*, 한상용\*\*

### A Broker Based Synchronous Transaction Algorithm For Virtual Market Place

Nam-O Kang, Sang-Yong Han

#### Abstract

Internet-based electronic trade has been growing fast. But most users are not yet familiar with the system and find it very difficult to purchase and sell the products in the cyber market place. To handle these problems, agent-based virtual market place system has been proposed where agents instead of individuals participate in trading of goods. Most of the proposed models have been in the two general categories. The first is the direct transaction among sellers and buyers, and the second is the agent-based transaction. However, the transaction is not fair and the best deal can't be guaranteed for both models.

In this paper, we propose a new broker based synchronous transaction algorithm which is fair to both parties and guarantees the best deal. Our algorithm is implemented using Visual C++ and the experimental results show that our method is better than the two traditional transaction models in every performance metrics. Number of transactions are increased up to 21% and price adjustment is up to 280% better for some transactions.

*Key Word : Agent, Transaction Model, The Direct Transaction, The Agent-based Brokered Transaction, Broker Based Synchronous Transaction*

---

\* 이 논문은 1999학년도 중앙대학교 학술 연구비 지원에 의한 것임.

\*\* 중앙대학교 컴퓨터 공학과

## 1. 서론

오늘날, 전자상거래는 국내외를 막론하고 이미 활발하게 진행되고 있으며 그 거래 규모 또한 날로 증가하고 있다. Forrester Research사에서는 2003년 인터넷 상거래(기업간 거래, 기업과 소비자간의 거래) 규모가 전세계 판매액의 5%인 3조 2,000억 달러에 이를 것으로 전망하고 있으며, IBM의 거스너 회장은 1998년 국내 전자 상거래가 2억 4,000만 불에서 2001년에는 50억불에 이를 것으로 전망하고 있다. 조사기관에 따라 전자 상거래 시장 규모에 대한 전망이 상이하기는 하지만, 21세기에는 전자 상거래가 매우 활성화 될 것으로 전망된다.[주재훈, 1998]

이러한 증가 예상은 인터넷을 이용한 거래가 저 비용, 고효율을 얻을 수 있기 때문으로 분석되고 있는데, 앞으로 초고속 통신망과 가상현실 기술이 보편화되면 인터넷을 통한 전자상거래는 더욱 그 범위를 넓혀갈 것으로 기대된다.[최중만, 1997]

하지만 현재의 인터넷 전자상거래는 사용자의 컴퓨터에 대한 비전문성, 실세계 상거래와의 큰 격차로 인한 이질감 및 거부감이라는 문제점을 내포하고 있다. 이를 해결하여 보다 친화성이 높은 상거래 환경을 제공하기 위해 에이전트 기술의 도입이 적극 추진되고 있다.[이은석, 이진구, 강제연, 1997]

에이전트란 사용자를 대신하여 사용자가 원하는 작업을 자동으로 해결해주는 소프트웨어라고 할 수 있다.[이승원, 1999] 에이전트를 도입한 시스템으로는 크게 종합 정보 시스템의 형태와 거래 대리 시스템의 형태

들로 나뉜다. 종합 정보 시스템의 형태는 에이전트가 사용자에게서 물건에 대한 요구를 받아들이면, 사용자를 대신하여 온라인 상점을 찾아 돌아다니고 정보를 획득하여 보여준다. 거래 대리 시스템의 형태에서 에이전트는 사용자로부터 구매 혹은 판매를 원하는 상품에 대한 정보를 입력받아서, 사용자를 대신하여 거래 대상을 찾고 협상 과정을 통하여 거래를 결정 짓는다.

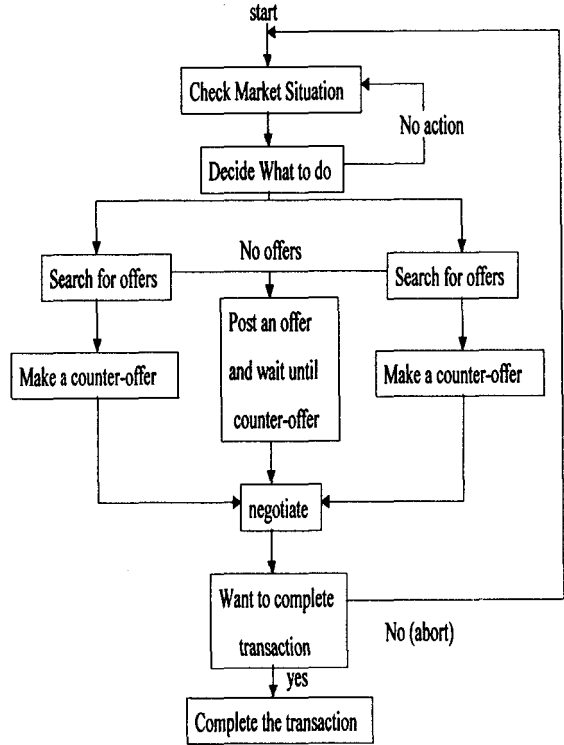
거래 대리 시스템의 형태에서 에이전트는 상품의 판매와 구매를 위하여 상거래(marketplace)에 참여하게된다. 에이전트는 거래를 행하는데 있어 참여한 마켓 시스템의 거래 알고리즘에 따라 조정을 받게된다. 대표적인 것으로는 판매자와 구매자 사이의 직접적인 거래와 브로커 에이전트를 통한 거래 등이 있다. 이들 알고리즘에서는 거래에 있어서의 공정성과 쌍방의 최선의 거래를 보장하기 어렵다는 단점이 있는데 본 논문에서는 이를 보완한 브로커 기반 동기화 거래(broker based synchronous transaction)를 제시한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 통하여 기존의 거래 알고리즘을 서술하고, 3장에서는 기존의 거래 알고리즘들의 단점에 대한 해결 방안으로 브로커 기반 동기화 거래를 제시한다. 4장에서는 기존의 거래 알고리즘을 적용한 시스템과 본 논문이 제안한 브로커 기반 동기화 거래를 적용한 시스템의 설계 및 구현을 설명하고, 이를 이용한 실험 및 결과 분석을 5장에서 기술하기로 한다. 마지막 6장으로 결론 및 향후 연구 방향을 언급한다.

## 2. 관련 연구

에이전트를 도입한 시스템 중 거래 대리 시스템의 형태에서 에이전트는 상품의 판매와 구매를 위해 상거래 공간에 참여하게 된다. 상거래 공간에 참여한 에이전트는 마켓 시스템의 거래 알고리즘에 따라 조정을 받으며 거래 대상을 찾고 협상 과정을 통하여 거래를 결정 짓는다. 이에 따라 효율적이고 통일된 방식의 거래 알고리즘 설계에 많은 연구가 되어졌다.

현재까지 제안되었던 거래 알고리즘의 대부분은 크게 두 가지 형태로 나뉘어질 수 있다. 첫째는 판매자와 구매자 사이의 직접적인 거래(Direct transactions among providers and consumers)이고 두 번째로는 브로커 에이전트를 통한 거래(brokered transactions among agents) 이다.[M. Tsvetovaty, M Gini, B Mobasher, Z Wieckowski, 1997]



<그림 1> Algorithm for direct transactions  
[M. Tsvetovaty, M Gini, B Mobasher, Z Wieckowski, 1997]

### 2.1 직접적인 거래 알고리즘

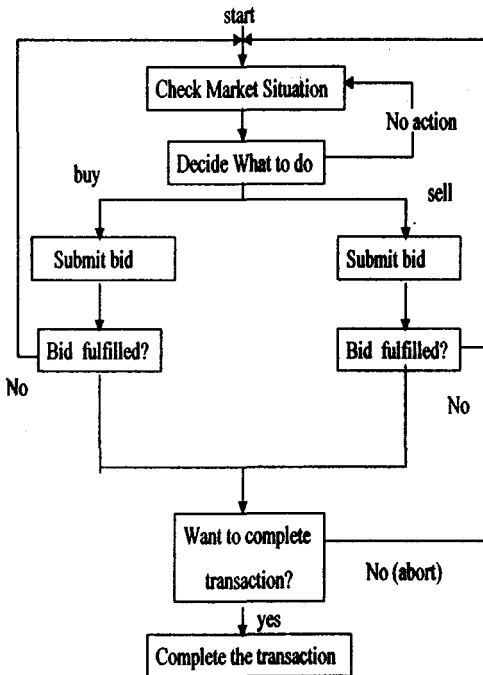
직접적인 거래에서 상거래 공간에 참여하는 에이전트는 시분할 방식으로 동작을 허가 받아 행동한다. 활동을 허가 받은 에이전트는 거래할 상대방을 직접 찾아서 협상을 통하여 거래를 한다. 거래가 성사된 에이전트는 마켓에서 제거되어지고 그렇지 못한 에이전트는 최종 협상시간 내에서 다시 동작 시간(processing time)을 받아서 거래에 참여한다.(그림 1)

이 알고리즘은 판매자와 구매자 사이의 개별적인 협상이 가능하다는 장점이 있는 반면 거래의 상대방을 찾기 위한 비용이 많이 들고, 거래에 참여하는 에이전트가 시분할 방식으로 동작이 허가되므로 때에 따라서는 구매자나 혹은 판매자에게 손해가 된다. 특히 이러한 현상은 에이전트들이 서로 다른 시점에 산출된 가격을 바탕으로 거래에 임하므로 최근에 산출한 희망 가격으로 거래에 참여하는 에이전트가 손해를 본다. [강남오,한상용,1999]

이러한 거래 모델을 채택하고 있는 시스템으로는 Kasbah가 있다.

## 2.2 브로커 에이전트를 통한 거래 알고리즘

브로커 에이전트를 통한 거래에서, 상거래 공간에 참여하는 에이전트는 동작을 허가 받아 브로커 에이전트에게 자신의 희망 가격을 알려주고, 브로커 에이전트는 최선의 상대방을 찾아준다. 거래가 성사되면 마켓에서 제거되어지고 그렇지 못한 에이전트는 최종 협상시간 내에서 다시 동작을 허가 받아서 거래에 참여한다. (그림 2)



<그림 2> Algorithm for brokered transactions  
[M. Tsvetovaty, M Gini, B Mobasher, Z Wieckowski, 1997]

이 알고리즘은 마켓 시스템에서 거래의 상대방을 찾기 위한 비용을 줄이지만, 에이전트들은 서로 다른 시점에서 산출된 가격

을 바탕으로 거래에 참여하므로 거래에 있어 공정성을 확보하고 있지는 못하고, 구매자와 판매자 쌍방에 있어 최선의 거래를 기대하기는 어렵다는 단점이 있다.[장남오,한상용,1999]

이러한 거래 모델을 채택하고 있는 시스템으로는 Magma가 있다.

## 3 브로커 기반 동기화 거래

본 연구에서는 브로커 에이전트를 통한 거래를 바탕으로 하여서, 거래의 생성 효율을 높이고, 거래에 참여하는 에이전트의 가격 산출에 있어서의 공정성을 확보하며 쌍방간에 최선의 거래를 이룰 수 있는 브로커 기반 동기화 거래를 제안한다. 이 알고리즘 설계의 주안점은 다음과 같다.

### 3.1 거래의 공정성

거래의 공정성이란 판매자와 구매자가 같은 시점에서 산출된 가격으로 거래에 임해야 함을 말한다. 직접적인 거래에서는 에이전트가 시분할 방식으로 처리 시간을 할당받아서 에이전트가 동작한다. 이때 에이전트는 자신의 희망 가격을 산출한다. 만약 에이전트가 자신의 희망 가격을 산출하고 거래에 임하게 되면 처리 시간을 할당받아서 거래를 주도하는 에이전트가 손해를 보게 되는 현상이 일어난다. 왜냐하면 거래에 임하는 상대방은 예전에 구해놓은 희망 가격을 가지고서 거래에 임하기 때문이다. 그러므로 처리 시간을 할당받은 에이전트는 거래에 임해서 거래가 이루어지지 않으면 자신

의 희망 가격을 바꾸는 정책을 쓴다. 하지만 이러한 경우 거래가 이루어졌을 때 처리 시간을 할당받아 동작하는 에이전트가 유리하다. 왜냐하면 자신은 상대방보다 더 과거의 시점에서 산출된 희망 가격으로 거래에 임하기 때문이다. 이런 현상은 브로커 에이전트를 통한 거래에서도 발생한다.

브로커 기반 동가화 거래에서는 이러한 문제점을 해결하기 위해 기준 시간이라는 개념을 도입했다. 기준 시간이란 물리적인 마켓 시간에서의 한 시점으로서 거래에 참여하는 모든 에이전트가 자신의 희망 가격을 산출하는데 기준이 되는 시간을 말한다. 기준 시간을 바탕으로 거래에 임하는 희망 가격을 산출함으로써 모든 에이전트가 거래에 임하는 가격은 같은 시점에 산출되어지고 이를 바탕으로 거래가 이루어지게 하여 거래의 공평성을 유지한다.

### 3.2 최선의 거래

최선의 거래란 판매자에게는 최고가, 구매자에게는 최저가를 보장해주는 것이다.[M. Tsvetovaty, M Gini, B Mobasher, Z Wieckowski, 1997] 이는 판매 가격과 구매 가격의 차이로써 구한다. 판매자 에이전트와 구매자 에이전트가 희망 가격을 산출하여 판매자(가) : 3000, 구매자(A) : 1000, 구매자(B) : 2000 와 같이 되었다면, 거래가 이루어지는 경우가 존재치 않는다. 왜냐하면 구매자 에이전트의 희망 가격보다 판매자의 희망 가격이 높기 때문이다.

하지만 판매자(가) : 1000 구매자(A) : 1500, 구매자(B) : 2000의 경우는 판매자

(가) : 1000 구매자(A) : 1500 과 판매자(가) : 1000 구매자(B) : 2000 두 경우 모두 거래가 이루어진다. 하지만 전자의 경우에 거래 가격의 차이는 500인 반면 후자의 경우는 1000이다. 즉 판매자 : 1000 구매자 : 2000이 최선의 거래가 된다.

동일한 에이전트 생성의 시나리오라 할지라도 에이전트간에 어떻게 맺어지는가에 따라서 거래 가격의 차이는 틀려진다. 마켓에 참여하는 에이전트가 <표 1>과 같다고 하자.

<표 1> 거래에 참여한 에이전트의 상태

판매자	희망가격	구매자	희망가격
가	157	A	147
나	176	B	185
다	150	C	153
라	183	D	162

위의 경우 최선의 거래를 맺어주는 거래의 쌍은 다-B, 가-D이고 이때의 에이전트간에 거래 금액 차의 평균은 20원이다. 직접적인 거래에서는 시분할 방식으로 처리 시간을 할당받아서 에이전트가 동작하므로 상황에 따라서 거래를 이루는 쌍이 정해지므로 최선의 거래를 기대하기 힘들다. 브로커 에이전트를 통한 거래에서는 브로커 에이전트에게 거래를 요청하는 에이전트에게 최선의 상대를 찾아준다. 하지만 요청하는 에이전트는 상황에 따라 결정된다. 즉 '다'가 브로커 에이전트에게 요청했을 경우에는 B를 찾아주지만, '라'가 요청했을 때에도 B를 찾아주어서 새로운 거래의 쌍이 이루어지고 이는 쌍방간의 최선의 거래를 만족시

켜주지는 못한다.

브로커 기반 동기화 거래에서는 seller 에이전트를 가격에 따라서 내림차순으로 정리하고 buyer 에이전트는 오름차순으로 정리한다. 그리고 seller의 최소 값과 buyer 에이전트의 최대 값을 거래가 이루어지는 동안 연결함으로써 최선의 거래를 보장하도록 하였다.

### 3.3 거래생성의 효율

시장에 참여하는 에이전트는 마감 시간이 존재한다. 즉 마감 시간 내에 거래를 이루지 못하면 에이전트는 시장에서 제거되어 진다. 그러므로 각 에이전트가 유휴 시간 없이 거래를 이루기 위해 동작하도록 하는 것이 무엇보다 중요하다. 직접적인 거래의 경우 에이전트가 할당받은 처리 시간이 남을 수도 있고, 처리 시간이 다된 경우 다른 에이전트에게 처리 시간을 넘겨야 하는데 이 경우 오버 헤드가 존재한다. 브로커 에이전트를 통한 거래의 경우 사용자 에이전트가 브로커 에이전트에게 거래할 상대방을 찾아주기를 요청한 경우 매번 브로커 에이전트는 상대방 에이전트를 검색해야하고 이는 중복된 작업에 시간을 소비하게 된다. 즉 앞의 두 경우 모두 거래할 상대방을 찾는 데 많은 비용이 들고있다.

브로커 기반 동기화 거래에서는 사용자 에이전트가 자신의 희망 가격만을 산출하게 하고, 거래할 상대방을 찾기 위한 어떠한 행위도 하지 못하게 하였다. 그리고 브로커 에이전트가 주도적으로 개입하여 판매 에이전트와 구매 에이전트를 각각 정렬시키고

이들의 거래를 맺어주도록 하였다. 거래할 상대방을 찾는 데 있어서, 브로커 에이전트를 통한 거래에서보다 중복을 줄임으로 해서 마켓에서 거래 생성의 효율을 높이도록 하였다.

### 3.4 브로커 기반 동기화 거래 알고리즘

브로커 기반 동기화 거래에서, 매니저 에이전트는 마켓 내의 사용자 에이전트들을 정리한다.(기한이 끝난 에이전트는 삭제하고, 참여를 바라는 에이전트는 등록을 한다). 거래에 참여하려는 사용자 에이전트의 희망가격을 산출하기 위한 기준 시간을 설정한 후 각 사용자 에이전트에게 각자의 정책에 따라 자신의 희망가격을 산출하도록 한다. 희망 가격의 산출이 끝나면 브로커 에이전트가 주도적으로 개입하여 거래를 맺어 준다.

사용자 에이전트들은 하나의 기준 시간에 맞추어서 가격을 산출하고, 거래에 임하기 때문에 거래 가격의 공정성을 확보한다. 그리고 브로커 에이전트가 모든 에이전트의 가격을 보고 거래를 맺어주므로 쌍방간의 최선의 거래를 이루도록 할 뿐 아니라, 구매자 에이전트와 판매자 에이전트는 자신의 협상 가격만 산출하도록 하고 브로커 에이전트가 주도적으로 계약을 성립시키도록 하여 상거래 시장의 거래 생성 효율을 높인다.[강남오,한상용,1999] 브로커 기반 동기화 알고리즘이 <표 2>에 나타나있다.

<표 2> 에이전트 기반 동기화 거래 알고리즘

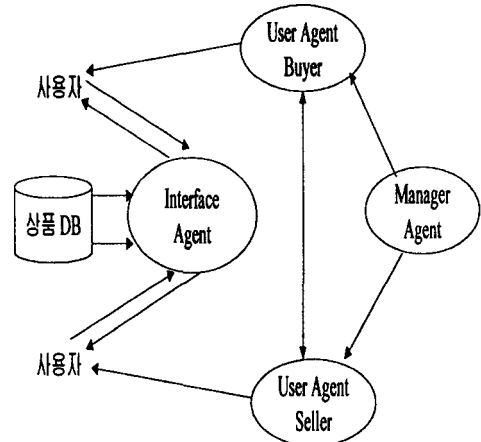
알고리즘 : Broker Based Synchronous Transaction
입력 : WBL, WSL // 시장에 참여하려는 판매, 구매 에이전트 리스트 출력 : 거래가 형성된 에이전트
MBL, MSL // 시장에 참여한 판매 구매 에이전트 리스트 Market_Time // 시장의 상태를 저장
<pre> WHILE(TRUE) {   IF(Market_Time == MANAGE)   {     IF((WBL == NULL) &amp;&amp; (WSL == NULL) &amp;&amp;       (MBL == NULL) &amp;&amp; (MSL == NULL))       수행을 끝낸다;     ELSE     {       WBL, WSL에서 에이전트의 생성시간이 된       에이전트들을 MBL, MSL로 옮겨서 등록시킨다.       거래가 맺어진 에이전트와 협상 시간을 넘긴       에이전트들을 마켓 시스템에서 제거한다.       Market_Time = TRANSACTION;     }   }   ELSE //Market_Time == TRANSACTION   {     기준 시간을 설정한다.     MBL, MSL에 있는 모든 에이전트들은 기준 시간에     따라서 자신의 거래 희망 가격을 산출한다.     거래 희망 가격에 따라 판매 에이전트는 내림차순     구매 에이전트는 오름차순으로 정렬한다.     높은 구매가의 구매 에이전트와 낮은 판매가의     판매 에이전트를 순서로 거래를 맺어준다.     Market_Time = MANAGE;   } }                 </pre>

### 4. 시스템 설계 및 구현

기존의 거래 알고리즘과 이 논문에서 제안하는 브로커 기반 동기화 거래의 성능을 비교하기 위하여 각각을 채택한 상거래 시스템을 구현하였다. 그리고 이러한 상거래 시스템들의 성능을 공정하게 실험하기 위하여 에이전트들을 생성하는 에이전트 생성기를 구현하였다.

#### 4.1 SMP(Simple Market Place)

SMP(Simple Market Place)는 직접적인 거래를 거래 알고리즘으로 채택한 상거래 시스템이다. 구성은 아래와 같다.(그림 3)



<그림 3> SMP

인터페이스 에이전트(Interface Agent)는 상품 데이터 베이스에 접근하여 전자 상거래 공간에 접근한 사용자가 구매하거나 혹

은 판매하려는 상품에 대하여 기입해야 할 항목을 요구하고, 이를 바탕으로 사용자 에이전트를 생성하는 역할을 한다.

사용자 에이전트(User Agent)는 사용자를 대신하여 전자 상거래 공간에서 거래를 행하고 거래가 이루어지면 사용자에게 이루어진 거래에 대한 사항을 연락하는 역할을 한다. 사용자 에이전트는 구매 에이전트와 판매 에이전트로 나뉘어지는데, 사용자가 전자 상거래 공간에 접근하여 사용자 에이전트 생성에 필요한 항목을 기입함으로써 생성된다. 생성된 사용자 에이전트는 일단 마켓에 참여하기 위하여 기다리고 있다가, 매니저 에이전트에 등록이 되어서 전자 상거래 공간에서 거래 활동을 행한다. 직접적인 거래에서는 시분할 방식으로 동작을 허용 받아 자신의 희망 가격을 산출하고 자신이 거래할 상대방을 직접 찾는 역할을 한다.

매니저 에이전트(Manager Agent)는 거래에 참여하는 사용자 에이전트가 동작할 수 있도록 시분할로 동작을 허가하고, 마켓에 참여한 에이전트 중 거래가 성사된 에이전트와 최종 협상시간을 넘긴 에이전트를 정리하며 마켓에 참여하기 위해 기다리는 새로 생성된 사용자 에이전트를 등록하여 시장에 참여시키는 역할을 한다.

상품 데이터 베이스는 전자 상거래 공간에서 다루는 상품에 대한 데이터가 저장되어 있었다.

상거래 공간에는 두 가지 형태의 시간이 존재하는데, 정리 시간(Manage Time)과 거래 시간(Transaction Time)이다.

정리 시간은 생성한 사용자 에이전트를

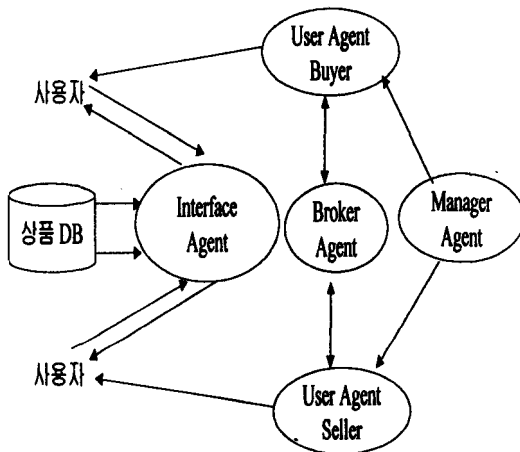
상거래 공간에 참여시키고 거래가 성사된 에이전트를 상거래 공간에서 제거하는 등 상거래 공간에 참여한 에이전트를 정리하는 시간이다. 거래 시간은 상거래 공간에 참여하는 사용자 에이전트가 자신의 희망 가격을 가격 변화 정책에 따라 변화시키고 거래를 성사시키도록 허용하는 시간이다.

사용자(구매자, 판매자)가 전자 상거래 공간에 접근하면 인터페이스 에이전트는 상거래 공간에서 다루는 상품을 사용자에게 알려준다. 사용자가 자신이 구매 혹은 판매하려는 상품을 선택하면 인터페이스 에이전트는 상품 데이터 베이스에 접근하여 해당 상품을 구매 혹은 판매하는 사용자 에이전트를 생성하기 위해서 기입해야 할 항목에 대한 정보를 가져온다. 사용자가 해당 항목에 대해 기입을 하면 인터페이스 에이전트는 사용자 에이전트를 생성시킨다. 생성된 사용자 에이전트는 매니저 에이전트에 등록함으로써 상거래 공간에 참여한다. 거래 시간 동안 매니저 에이전트에 등록된 사용자 에이전트는 시분할 방식으로 동작을 허가 받아 자신이 거래할 상대방을 찾고 거래에 임한다. 거래가 성사된 에이전트는 거래된 내용을 사용자에게 알리고 최종 승낙을 요구하는 메시지를 보낸다. 거래를 이루지 못한 에이전트는 자신의 희망 가격을 새로이 산출하고 다음 동작을 허가 받을 때까지 기다린다. 거래 시간이 끝나면 정리 시간으로 들어가는데 이때 매니저 에이전트는 거래가 성사된 에이전트와 최종 협상 시간을 넘긴 에이전트를 정리한다 그리고 시장에 참여하기 위해 기다리는 새로 생성된 에이전트를 등록시킨다.



### 4.2 BBMP

BBMP(Broker Based Market Place)는 브로커 에이전트를 통한 거래를 거래 알고리즘으로 채택한 상거래 시스템이다. 구성은 아래와 같다.(그림 4)



<그림 4> BBMP

인터페이스 에이전트와 상품 데이터 베이스는 SMP와 동일하다.

BBMP에서의 사용자 에이전트의 기본적인 동작은 SMP의 사용자 에이전트와 동일하나, 브로커 에이전트를 통한 거래에서의 사용자 에이전트는 자신이 직접 거래할 상대방을 찾지 못하고 브로커에게 적합한 거래자를 요청하여 브로커가 거래할 상대방을 찾아주면 거래를 행한다는 점이 다르다.

브로커 에이전트(Broker Agent)는 거래를 요청하는 사용자 에이전트에게 가장 좋은 가격을 제시하는 상대방을 찾아주는 역할을 한다.

매니저 에이전트는 거래에 참여하는 사

용자 에이전트와 브로커 에이전트의 행위를 조정하고, 마켓에 참여한 에이전트 중 거래가 성사된 에이전트와 최종 협상시간을 넘긴 에이전트를 정리하는 역할을 한다.

사용자(구매자, 판매자)가 전자 상거래 공간에 접근하면 인터페이스 에이전트는 상거래 공간에서 다루는 상품을 사용자에게 알려준다. 사용자가 자신이 구매 혹은 판매하려는 상품을 선택하면 인터페이스 에이전트는 상품 데이터 베이스에 접근하여 해당 상품을 구매 혹은 판매하는 사용자 에이전트를 생성하기 위해서 기입해야 할 항목에 대한 정보를 가져온다. 사용자가 해당 항목에 대해 기입을 하면 인터페이스 에이전트는 사용자 에이전트를 생성시킨다. 생성된 사용자 에이전트는 매니저 에이전트에 등록함으로써 상거래 공간에 참여한다. 거래 시간 동안 동작을 허가 받은 사용자 에이전트는 브로커 에이전트에게 거래를 행할 상대방을 찾아 주기를 요청하면 브로커 에이전트가 가장 좋은 가격을 제시하는 상대방을 찾아준다. 찾아준 상대방과 거래가 이루어지면 사용자에게 거래된 내용을 알리고 최종 승낙을 요구하는 메시지를 보낸다. 거래를 이루지 못한 에이전트는 자신의 희망 가격을 새로이 산출하고 다음 동작을 허가 받을 때까지 기다린다. 거래 시간이 끝나면 정리 시간으로 들어가는데 이때 매니저 에이전트는 거래에 성사된 에이전트와 최종 협상 시간을 넘긴 에이전트를 정리한다 그리고 시장에 참여하기 위해 기다리는 새로 생성된 에이전트를 등록시킨다.

### 4.3 SBBMP

SBBMP(Synchronous Broker Based Market Place)는 브로커 기반 동기화 거래를 거래 알고리즘으로 채택한 상거래 시스템이다. 구성은 BBMP와 동일하나 각각의 구성요소의 역할에서 차이를 가진다. 인터페이스 에이전트, 상품 데이터베이스, 매니저 에이전트는 BBMP와 동일하다.

SBBMP에서의 사용자 에이전트의 기본적인 동작은 SMP의 사용자 에이전트와 동일하나, 브로커 기반 동기화 거래에서의 사용자 에이전트는 마켓 시장에서 설정된 기준 시간에 맞추어서 자신의 희망 가격을 산출할 뿐, 직접적인 거래나 브로커를 통한 거래와는 달리 거래를 행할 상대방을 찾기 위한 어떠한 행위도 하지 않는다.

브로커 기반 동기화 거래에서의 브로커 에이전트는 희망가격이 산출된 사용자 에이전트들 중에서 최선의 거래가 되는 쌍을 찾아 주도적으로 거래를 맺어주는 역할을 한다.

사용자(구매자, 판매자)가 전자 상거래 공간에 접근하면 인터페이스 에이전트는 상거래 공간에서 다루는 상품을 사용자에게 알려준다. 사용자가 자신이 구매 혹은 판매하려는 상품을 선택하면 인터페이스 에이전트는 상품 데이터 베이스에 접근하여 해당 상품을 구매 혹은 판매하는 사용자 에이전트를 생성하기 위해서 기입해야할 항목에 대한 정보를 가져온다. 사용자가 해당 항목에 대해 기입을 하면 인터페이스 에이전트는 사용자 에이전트를 생성시킨다. 생성된 사용자 에이전트는 매니저 에이전트에 등록함으로써 상거래 공간에 참여한다. 거래 시

간이 되면 매니저 에이전트는 마켓의 기준 시간을 설정한다. 동작을 허가 받은 사용자 에이전트는 마켓의 기준 시간에 맞추어서 자신의 희망 가격을 산출한다. 희망 가격의 산출이 끝나면 브로커 에이전트는 이들 중 최선의 거래가 되게 거래를 맺어준다. 거래가 성사된 에이전트는 사용자에게 거래된 내용을 알리고 최종 승낙을 요구하는 메시지를 보낸다. 거래를 이루지 못한 에이전트는 다음 거래 시간까지 기다린다. 거래 시간이 끝나면 정리 시간으로 들어가는데 이때 매니저 에이전트는 거래에 성사된 에이전트와 최종 협상 시간을 넘긴 에이전트를 정리한다 그리고 시장에 참여하기 위해 기다리는 새로 생성된 에이전트를 등록시킨다.

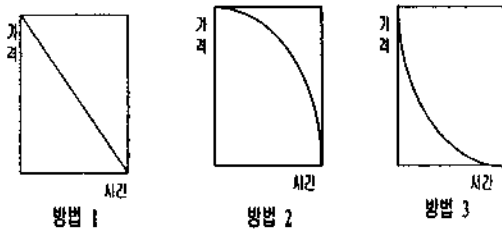
### 4.4 에이전트 생성기

사용자는 시장 상거래 공간에 에이전트를 생성하여 참여한다. 주어진 시간 내에 사용자가 생성한 에이전트를 시간 별로 정렬해 보면 이것은 하나의 에이전트 생성 시나리오가 될 것이다.

이 논문에서는 세 가지 형태의 거래 알고리즘을 소개하고 있다. 직접적인 거래, 브로커 에이전트를 통한 거래, 그리고 브로커 기반 동기화 거래이다. 이들 거래 알고리즘의 성능을 비교하기 위하여 이들을 채택한 시스템을 구현하였다. 이들의 성능은 하나의 에이전트 생성 시나리오에 대하여 비교하는 것이 실험의 공정성을 위하여 중요하다. 그래서 에이전트 생성기를 구현하였다. 에이전트 생성기는 주어진 시간 내에 일정한 에이전트를 생성해내는 역할을 한다.

판매자 에이전트는 생성 시점부터 에이전트의 소멸 시점까지 자신의 희망 가격을 변화시킨다. 생성 시점에 가장 높은 거래 가격을 제시하지만 시간이 흐름에 따라 자신의 희망 가격을 낮춘다. 구매자 에이전트는 이와는 반대의 현상이 일어난다.

본 시스템에서 생성하는 판매 에이전트의 희망 가격 변환 정책은 아래의 3가지가 있다. (그림 5)



<그림 5> 에이전트의 희망 가격 변환 정책

방법 1에서는 판매 에이전트가 시간에 대하여 선형적으로 자신의 가격을 낮춘다. 방법 2에서는 초기에는 완만하게 자신의 희망 가격을 낮추다가 마감 시간에 다가갈수록 급격히 자신의 희망 가격을 낮춘다. 방법 3에서는 초기에는 자신의 가격을 급격히 낮추다가 마감시간에 가까워질수록 완만히 낮춘다.

## 5. 실험 및 결과 분석

### 5.1 실험

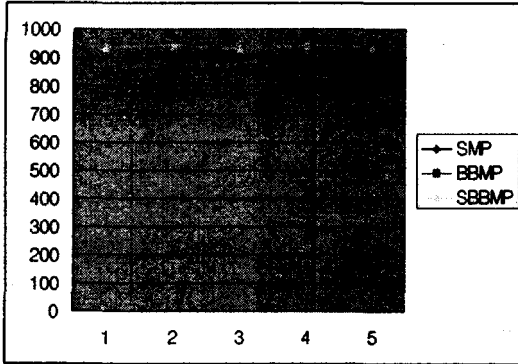
실험은 windows NT 상에서 Visual C++ 언어로 구현하였다. 에이전트 생성기를 통하여 구매자 에이전트 1000 개와 판매자

에이전트 1000개를 가지는 리스트를 각각 5 개씩 만들었다. 정리 시간과 거래 시간 사이에는 1 second의 차이를 두었다.

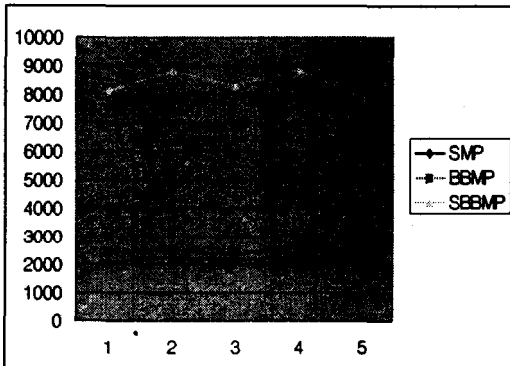
생성된 에이전트의 내용은 다음과 같다. 에이전트의 상품 항목은 10개가 있다고 보고 임의로 생성하였다. 에이전트의 생성 시점은 10분내에서, 소멸 시점은 20분내에서 임의로 정하도록 하였다. 에이전트의 시작 가격은 20000에서 40000사이에서 임의로 정하도록 하였다. 에이전트의 최종 가격은 50000에서 100000사이에서 임의로 정하도록 하였다. 희망 가격 변환 정책은 방법 1, 방법 2, 방법3 중에서 임의로 정하도록 하였다. 직접적인 거래를 채택한 SMP에서 에이전트의 처리 시간은 10 millisecond로 하였다.

실험의 중점은 직접적인 거래, 브로커 에이전트를 통한 거래, 그리고 브로커 기반 동기화 거래를 채택한 마켓의 성능 비교와 최선의 거래를 맺어주는 정도를 측정했다.

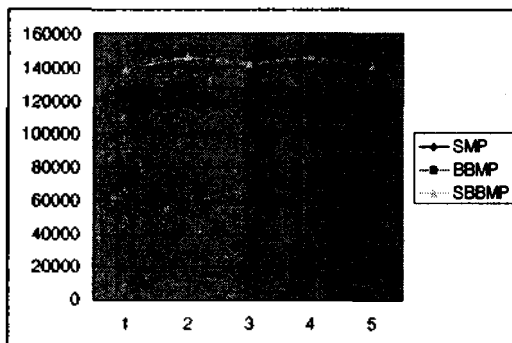
각 마켓 시스템을 비교하는 요소는 동일한 제한 시간 내에 얼마나 많은 거래를 맺어주는 가를 나타내는 생성된 거래의 개수, 얼마나 최선의 거래를 맺어주는 가를 나타내는 거래 차액의 평균, 그리고 에이전트가 생성되어 거래에 이르기까지 걸린 시간의 평균을 나타내는 에이전트가 거래에 이르기까지 걸린 평균 시간이다. 생성된 거래의 개수와 에이전트가 생성되어 거래에 이르기까지 걸린 시간의 평균은 마켓 시스템의 성능을 거래 차액의 평균은 맺어진 거래의 만족 정도를 측정하는 기준이다.



<그림 6> 생성된 거래의 개수에 대한 그래프



<그림 7> 거래 차액의 평균에 대한 그래프



<그림 8> 에이전트가 거래에 이르기까지 걸린 평균 시간에 대한 그래프

## 5.2 결과 분석

다섯 번의 실험 평균으로 에이전트간의 거래 성사 개수는 직접적인 거래가 773.8개 그리고 브로커 에이전트를 통한 거래가 853.4개 그리고 브로커 기반 동기화 거래가 937.6개이다. 브로커 에이전트를 통한 거래가 직접적인 거래보다는 10.3%정도 더 많은 거래 성사를 보였다. 브로커 기반 동기화 거래는 직접적인 거래보다는 21.2% 그리고 브로커를 통한 거래보다는 9.9% 많은 거래 성사를 보였다.

어느 정도 최선의 거래가 이루어졌는지는 거래 가격의 차이로서 나타낼 수 있다. 다섯 번의 실험 평균으로 직접적인 거래는 2209.2원, 브로커 에이전트를 통한 거래는 2474.2원, 그리고 브로커 기반 동기화 거래는 8395.8원이다. 브로커 에이전트를 통한 거래가 직접적인 거래보다는 12%향상을 보였다. 브로커 기반 동기화 거래는 직접적인 거래보다는 280% 그리고 브로커 에이전트를 통한 거래보다는 239%의 향상을 보였다.

에이전트가 생성되어서 거래에 이르기까지 걸린 시간으로 보면 직접적인 거래는 120초, 브로커 에이전트를 통한 거래는 126초, 그리고 브로커 기반 동기화 거래는 142초이다. 거래를 맺는데 있어서, 브로커 에이전트를 통한 거래가 직접적인 거래보다는 5%시간이 더 걸렸고, 브로커 기반 동기화 거래는 직접적인 거래보다는 18% 그리고 브로커 에이전트를 통한 거래보다는 12%정도 더 시간이 걸렸다.

## 6. 결론 및 향후 연구 방향

거래 대리 시스템에서 매매를 위해 상거래 공간에 참여한 에이전트는 마켓 시스템의 거래 알고리즘에 따라 조정을 받는다. 본 논문에서는 공정하고, 만족스러운 에이전트간의 거래가 이루어지게 브로커 기반 동기화 거래 알고리즘을 제시하였다. 그리고 실험을 통하여 기존의 거래 알고리즘과 비교하였다.

위의 실험 결과에서 보여주듯이 에이전트가 거래를 성사시키기 위해 마켓에 머무

르는 시간은 브로커 기반 동기화 거래, 브로커 에이전트를 통한 거래, 그리고 직접적인 거래 순서로 오래 걸렸으나, 거래 성사의 개수나 최선의 거래에 대한 수치에서는 브로커 기반 동기화 거래가 기존의 두 알고리즘보다 좋은 성능을 보였다.

에이전트 생성에서 거래가 이루어지기까지 마켓에 머무르는 시간은 적을수록 좋다. 본 논문에서의 브로커 기반 동기화 거래는 이 수치가 기존의 거래 알고리즘보다 떨어지는 결과를 보이고 있지만 브로커 에이전트를 지능화 하여 이를 보완할 예정이다.

## 참고 문헌

- [강남오, 한상용,1999] 강남오, 한상용, "전자 상거래 시장 공간에서 브로커 기반 동기화 거래 모델 알고리즘", 정보처리 학회지, 1999
- [이승원,1999] "전자 상거래 정책 추진 방향", 정보 처리 학회지, 1월 1999, pp.3-6
- [이은석, 1997] 이은석, "멀티에이전트 기술의 실세계 시스템으로의 응용", 정보과학회지, 3월 1997, pp.17-28
- [이은석, 이진구, 강재연,1997] 이은석, 이진구, 강재연 "인터넷 상에서의 전자 상거래를 위한 멀티에이전트 시스템", 정보처리 학회지, September 1997, pp.55-66
- [최중만,1997] "에이전트의 개요와 연구 방향", 정보과학회지, 3월 1997, pp.7-16
- [주재훈,1998] 주재훈, "지역 중소기업의 전자 상거래 활성화를 위한 정보화 전략", 한국 CALS/EC 학회지, 제3권 2호, 12월 1998, pp.1-15
- [A. Chavez, P. Maes,1996] A. Chavez, P. Maes, "Kasbah: An Agent MarketPlace for Buying and Selling Goods", PAAM96, pp.75-90, 1996
- [Joakim Eriksson, Niclas Finne, Sverker Janson, 1996] Joakim Eriksson, Niclas Finne, Sverker Janson, "Information and interaction in MarketSpace towards and open agent-based market infrastructure", USENIX Workshop, Nov 1996

- [M. Tsvetovaty, M Gini, B Mobasher, Z Wieckowski,1997] M. Tsvetovaty, M Gini, B Mobasher, Z Wieckowski, "MAGMA: An Agent-Based Virtual Market for Electronic Commerce", Journal of Applied Artificial Intelligence, September 1997
- [Maksim B. Tsvetovaty, Maria Gini, 1996] Maksim B. Tsvetovaty, Maria Gini, "Toward a Virtual Marketplace: Architectures and Strategies", PAAM96, pp.597-613, 1996
- [Robert H. Guttman, Pattie Maes, 1998] Robert H. Guttman, Pattie Maes, "Cooperative vs. Competitive Multi-Agent Negotiations in Retail Electronic Commerce", CIA'98 Paris, France, 5 July 1998

## 저자소개

**강남오** (e-mail : kang@archi.cse.cau.ac.kr)

중앙대학교 컴퓨터 공학과 학사

중앙대학교 컴퓨터 공학과 석사

관심분야 : 전자 상거래, 저전력 설계

**한상용** (e-Mail:han@archi.cse.cau.ac.kr)

B.S : Seoul National University, 1975

Ph.D: University of Minnesota, 1984

6/84~2/95 Researcher, IBM(Poughkeepsie and Waston Research)

9/89~6/90 Visiting Scholar, U.C. Berkeley

9/85~6/88 Adjunctive Professor, Rensellaer Polytechnique Institute

1/96~8/96 Consultant, Advanced Micro Device

3/95~present Professor, ChungAng University

관심분야 : Virtual Prototyping, VLSI CAD, EC(Electronic Commerce)