

# MAFNS를 이용한 서버 판매 에이전트 시스템 개발

강무홍<sup>a</sup> · 최형림<sup>b</sup> · 김현수<sup>c</sup> · 홍순구<sup>d</sup> · 박영재<sup>e</sup> · 박용성<sup>f</sup>

동아대학교 경영정보학과

<sup>a</sup>mongy@daunet.donga.ac.kr, <sup>b</sup>hrchoi@daunet.donga.ac.kr, <sup>c</sup>hskim@daunet.donga.ac.kr,  
<sup>d</sup>shong@daunet.donga.ac.kr, <sup>e</sup>yjpark@daunet.donga.ac.kr, <sup>f</sup>ys1126@daunet.donga.ac.kr

## Abstract

멀티 에이전트 기술은 다양한 환경변화와 복잡한 문제 등에 보다 빠르고 유연하게 대처하기 위한 새로운 패러다임으로 부각되고 있다. 따라서 많은 분야에서 동적인 문제를 해결하기 위해 멀티 에이전트 시스템을 구축하고자 연구가 진행중이다. 그러나, 기존의 멀티 에이전트 시스템을 개발하기 위한 Framework에 관한 연구들은 사용자를 대신하여 자율적으로 작업을 수행하는 일반적인 에이전트간의 정보 교환과 공유, 에이전트의 재활용, 이형질의 에이전트를 통합하기 위해 서로 각기 다른 에이전트간 작업 협력 구조를 제시하는 연구, 즉 멀티 에이전트가 가지는 기본적인 기능만을 제시한 Framework가 대부분이었다. 이러한 기존 멀티 에이전트 Framework의 한계점을 해결하고자 최형림 외 5명은 기본적인 멀티 에이전트 Framework의 기능뿐만 아니라 협상이라는 문제를 해결하기 위해 필요한 기능을 가지고 있는 MAFNS(Multi-Agent Framework for Negotiation Systems)를 설계하였다. 본 연구에서는 MAFNS의 구조를 이용하여 협상 메시지에 대한 평가, 협상 메시지의 관리, 협상 에이전트간의 빠르고 안정된 메시지 교환 지원 등과 같은 협상에 필요한 필수적인 요구 기능을 에이전트로 개발하고 각 에이전트 간의 유기적인 협력을 통해 자동으로 서버 판매 협상 문제를 해결할 수 있는 에이전트 시스템을 개발하였다.

## 1. 필요성 및 목적

컴퓨터와 인터넷의 보급으로 기존 오프라인에서 행해지고 있는 거래의 온라인화가 양적인 측면에서뿐만 아니라 질적인 측면에서도 빠르게 진행되고 있다. 이렇게 온라인화가 진행중인 거래 유형으로는 가격만으로 거래가 가능한 일반적인 물품에 대한 거래, 즉 경매나 고정가격거래가 대부분이고 아직까지 가격외의 다속성을 지닌 물품에 대한 협상 거래에 대한 온라인화는 그리 많지 않다. 또한 기존에 연구 및 개발되었던 협상시스템들도 자동으로 협상을 수행하는 자동협상시스템이 아니라, 사람의 협상을 지원하는 협상지원시스템(NSS; Negotiation Support Systems)이 대부분이었다. 이는 협상이 다속성의 거래 항목을 가지고 있고 이 협상 항목들간의 평가가 어려워 자동 협상 시스템을 개발하기에는 많은 어려움이 따르기 때문이다. 하지만, 자동협상시스템은 전자상거래 환경에서의 다양한 환경변화와 복잡한 문제 등에 보다 빠르고 유연하게 대처하고, 많은 협상을 일관성 있고 효율적으로 수행하기 위해서 필요하다. 전자상거래에서 협상이 활성화되면 많은 협상이 동시다발적으로 발생하게 될 것인데, 이를 모두가 사람이 수행할 수 없고 또한, 사람이 협상이 발생할 때

까지 한정 없이 기다리고 있을 수 없다. 이러한 문제를 해결하기 위해서는 협상을 자동으로 수행하는 시스템, 즉 자동협상시스템이 필요하다.

에이전트란 누군가를 위하여 무엇인가를 대신 해주는 컴퓨터 프로그램을 통칭한다. 좀더 구체적으로 정의하면 에이전트는 지속적으로 환경에서 지각된 것과 내부 지식을 바탕으로 추론하며 그 결과 행동하여 환경에 영향을 미치고 또한 사용자를 포함한 다른 에이전트와 의사소통하는 지속적으로 존재하는 소프트웨어 요소라고 할 수 있다[1]. 이러한 에이전트의 상호작용과 다양한 환경변화, 그리고 복잡한 문제 등에 보다 빠르고 유연하게 대처하기 위해 멀티 에이전트 기술이 새로운 패러다임으로 부각되고 있다. 많은 분야에서 멀티 에이전트 기술을 적용하여 여러 에이전트들이 상호작용하며 문제를 해결해 나가는 멀티 에이전트 시스템의 개발이 진행되고 있을 뿐만 아니라 이러한 멀티 에이전트 시스템을 개발하기 위한 개발 환경을 제공하는 멀티 에이전트 Framework들도 많이 제안되고 있다. 이러한 멀티 에이전트 Framework들은 개발자가 멀티 에이전트 시스템 내부의 에이전트를 개발하고 이를 통제하는 기능을 사용할 수 있도록 Framework의 API(Application Programming Interface)를 제공하고 있으며 또한 개발자와의 Feedback을 통해 문제점을 해결하고 계속 발전하

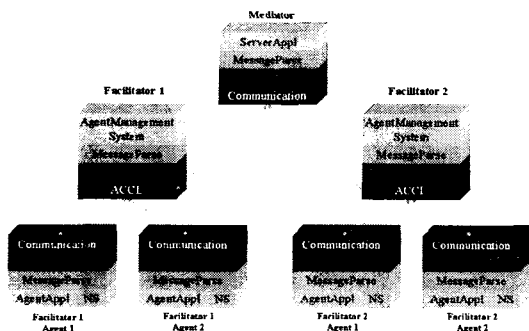
고 있다.

하지만 이러한 Framework들은 일반적인 멀티 에이전트 시스템의 개념인 에이전트간 통신, 관리만을 지원하고 있고 협상 시스템과 같이 특정 분야의 특성이 고려한 기능에 대해서는 지원을 하지 않고 있다. 이러한 협상을 지원하는 시스템에서는 협상의 특성을 고려한 기능이 추가되어야 한다. 현재까지의 멀티 에이전트 Framework들은 이러한 협상 시스템의 기능에 대한 개발지원이 없었다.

본 연구에서는 협상이라는 해결하기 어려운 문제를 원활히 수행할 수 있도록 협상안 평가, 협상안 작성, 협상전략 수립 등과 같은 기능을 제공하고 있는 MAFNS(Multi-Agent Framework for Negotiation Systems, 최형림 외 5명)[9]를 기반으로 하여 에이전트를 개발 및 구성하고 자동으로 협상을 수행할 수 있는 에이전트 시스템을 개발하였다.

## II. MAFNS (Multi-Agent Framework for Negotiation Systems)

MAFNS는 국제 표준안으로 채택된 FIPA의 AP 구조와 FIPA의 표준안을 따르는 JAFMAS를 기반으로 하고 앞에서 제기된 협상 시스템 개발시 기존 멀티 에이전트 Framework들의 문제점을 해결하고 이를 보완하여 개발자들이 쉽게 협상 시스템을 구현할 수 있도록 지원해 주는 멀티 에이전트 Framework이다. MAFNS는 기존의 멀티 에이전트 Framework와 마찬가지로 각 에이전트들을 관리하고 메시지를 중개해 주는 Facilitator인 AMS(Agent Management System)를 포함하고 있고 내부 에이전트가 아닌 다른 Facilitator의 외부 에이전트 간에도 메시지를 교환할 수 있도록 Facilitator의 위치 정보를 제공하는 Mediator를 가



<그림 1> MAFNS의 구조

지고 있다. 이러한 기능 외에도 MAFNS는 협상을 수행할 수 있도록 협상안 평가, 협상전략 수립, 협상안 작성의 기능을 지원하는 NS (Negotiation Strategy)를 가지고 있다.

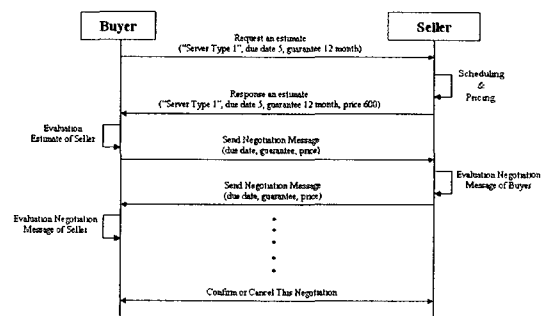
본 연구에서는 MAFNS에서 제공하는 기본적인 Mediator, Facilitator, 에이전트 개발 Framework 과 협상 수행을 지원하기 위한 NS를 이용하여 자동으로 협상을 수행하는 에이전트 시스템을 개발하였다.

## III. SSAS(Server Sales Agent System)

### 1. 적용 도메인 분석

본 연구에서는 에이전트 시스템을 개발하기 위해 적용 대상으로 조립형 서버 판매업을 선정하였다. 조립형 서버 판매업의 경우 사양(CPU, RAM 등)에 따라 다양한 종류를 가지고 있고 주문에 따라 소량으로 생산하는 것이 특징이다. 그렇기 때문에 일반적으로 판매자와 주문자간의 협상에 따라 계약의 체결이 주로 이루어지고 있다. 이에 본 연구에서는 조립형 서버 판매업을 적용 대상으로 선정하였고 거래가 이루어지는 과정을 분석하여 자동협상시스템을 개발하였다.

조립형 서버 판매업에서의 거래는 다음 그림과 같은 과정을 거쳐 이루어진다.



<그림 2> 서버 판매업의 협상 과정

먼저 구매자는 판매자에게 주문할 서버의 사양을 판매자에게 요청하고 견적을 의뢰한다. 판매자는 구매자가 요청한 내역에 따른 가격을 산출하여 견적 결과를 구매자에게 제시한다. 견적이 완료되고 나면 본격적으로 협상이 진행되게 된다. 판매자와 구매자는 상대방의 협상안을 평가하며 납기일, 가격, 무상수리기간을 조정하며 상대방에게 자신의 협상안을 전달한다. 이러한 협상과정에서 판

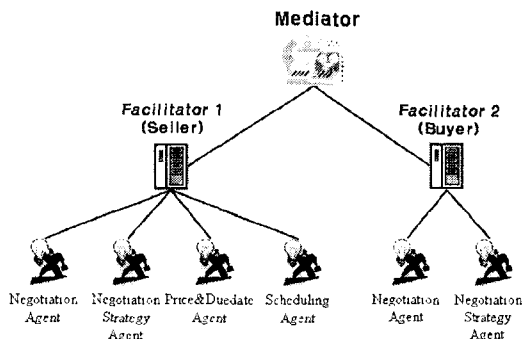
매자 또는 구매자는 상대방이 제시한 협상안에 만족할 경우에는 그 협상안을 채택하여 협상을 종료할 수 있고 반대로 몇 번의 협상을 수행하였으나 상대방과 협상이 이루어질 수 없다고 판단될 때에는 협상을 취소하게 된다.

## 2. SSAS 개발

### 2.1 SSAS의 구조

앞 절에서 분석한 조립형 서버 판매업의 협상 과정을 보면 크게 견적 단계, 협상 수행 단계, 협상 완료 단계로 나누어질 수 있다. 먼저 견적 단계의 경우 구매자가 판매자에게 견적의뢰를 요청하면 판매자는 공정계획 및 가격 산출을 수행하여 구매자의 견적에 응답하게 된다. 또한 협상 수행 및 완료 단계에서는 상대방이 제시하는 협상안을 평가하고 자신의 협상 전략을 수립하여 상대방에게 협상안을 전달하는 과정을 가진다. 이러한 과정을 지원하기 위해 본 연구에서는 먼저 판매자의 공정계획 및 가격 산출을 대신하여 수행하는 SCAgent(Scheduling Agent)와 PDAgent(Price & Due Date Agent)를 구현하였다. 또한 협상을 수행하기 위해 필요한 협상 전략 수립, 협상안 평가를 지원하기 위해 MAFNS에서 제시하고 있는 NS를 이용하여 NSAgent(Negotiation Strategy Agent)를 구현하였다. 그리고 이러한 협상 지원 에이전트들에게 결과값을 받아 실제 견적 및 협상 과정을 수행하는 NegoAgent(Negotiation Agent)를 구현하였다.

본 연구에서 개발한 SSAS 다음과 같은 구조를 가진다.



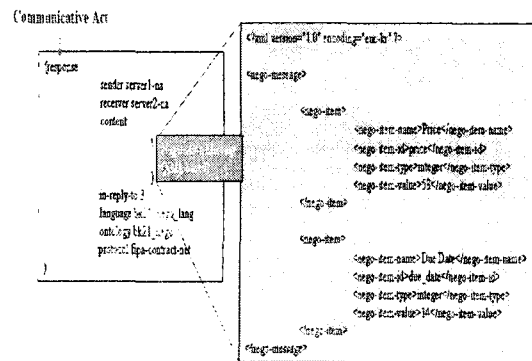
<그림 3> SSAS의 에이전트 구조

### 2.2 메시지 구조

각 에이전트 간의 협업을 위해서는 공통된 메시지를 통해 서로 작업에 대한 요청을 하고 결과를

전송해야 한다. 일반적으로 멀티에이전트 환경에서 에이전트 간에 전송되는 메시지는 KQML(Knowledge Query and Manipulation Language)을 기반으로 하고 있다. 본 연구에서는 KQML과 유사한 FIPA-OS의 ACL(Agent Communication Language)을 기반으로 에이전트 간 메시지를 작성하였다. 또한 판매자와 구매자가 주고받는 협상안은 다속성의 협상 항목을 표현할 수 있고 현재 교환 메시지로 많이 사용되고 있는 국제적인 표준인 XML을 기반으로 하여 작성하였으며 ACL의 content에 포함되어 전달된다.

본 연구에서 작성한 에이전트 간 메시지인 ACL과 XML기반의 협상안은 다음과 같다.



<그림 4> 메시지 구조

ACL의 Communication Act는 메시지의 타입을 나타내고 있으며 request(요청), response(응답), inform(통지)의 3가지가 있다. 또한 sender는 전송자, receiver는 수신자, content는 내용, in-reply-to는 몇 번째 메시지가 전송되었는지를 나타낸다. 본 논문에서는 in-reply-to를 통해 Communication Act에서 표현하지 못하는 세부적인 메시지 타입을 나타내고 있다.

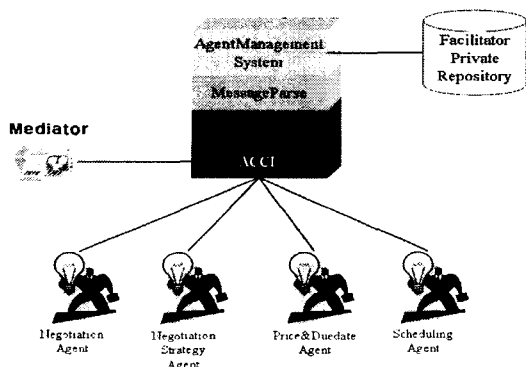
이러한 ACL을 통해 각 에이전트들이 누구를 대상으로 어떠한 Action을 할 수 있으며 이때 사용되는 Communicative Act와 in-reply-to의 타입은 다음 <표 1>과 같다.

<표 1> 각 에이전트와 Facilitator, Mediator의 기능

에이전트	Action	대상 에이전트	Communicative act	in-reply-to
NegoAgent	스케줄링 요청	내부 SCAgent	request	req_sc
	협상전략 요청	내부 NSAgent		req_ns
	협상안평가요청	내부 PDAgent		req_valuation
	납기일에 대한 가격 요청			req_price
	가격에 대한 납기일 요청			req_due
	견적 요청	외부 Nego Agent	response	req_estimate
	견적 전달			res_estimate
	최초협상안 전달			1
	카운터 협상안 전달			Round수
	협상취소			inf_cancel
협상완료		inform	inf_confirm	
SCAgent	스케줄링 결과 응답	내부 Nego Agent	response	res_sc
NSAgent	협상전략 결과 응답			res_ns
	협상안평가 결과 응답			res_valuation
PDAgent	납기일에 대한 가격 산출 결과 응답			res_price
	가격에 대한 납기일 산출 결과 응답			res_due

### 2.3 Mediator 및 Facilitator

Facilitator는 에이전트의 접속여부를 확인하여 SSAS 내의 에이전트간 메시지 중개와 에이전트 관리를 담당하고 있다. 이러한 기능을 담당하기 위해 ACCL(Agent Communication Channel Layer)를 제공하고 있으며 메시지의 작성을 위한 Message Layer, 메시지의 파싱을 위한 MessageParse Layer, 그리고 마지막으로 Facilitator의 주요 기능인 AMS(Agent Management System)과 DataBinder 등의 기능을 가지고 있는 ServerApplication Layer로 구성된다. Facilitator의 구조는 다음 <그림 5>와 같다.

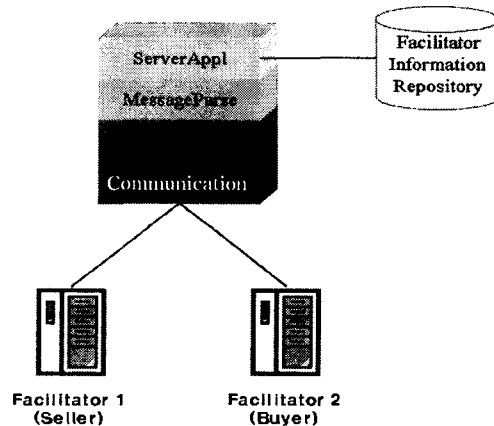


<그림 5> Facilitator의 구조

Mediator는 Facilitator와 Task 에이전트들로 이

루어진 하나의 멀티에이전트 시스템의 중개를 위해 각 Facilitator들의 위치정보를 보관하여 이를 요청시 제공하는 기능을 담당한다. 이렇게 부여 받은 Facilitator의 위치정보를 통해 Facilitator들은 서로 메시지를 교환할 수 있게 된다.

Mediator의 기본구조는 다음 <그림 6>과 같다.



<그림 6> Mediator의 구조

### 2.4 Task 에이전트

기본적으로 협상 문제 해결을 위해 각각의 업무를 수행하는 에이전트로서 특정 작업을 수행하기 위해 필요한 알고리즘 또는 비즈니스 로직을 포함하고 있다. 에이전트 시스템의 관리를 담당하고 있는 Facilitator와의 메시지 통신을 위한 Communication Layer가 있고 메시지의 작성을 위한 Message Layer, 그리고 MessageParse Layer가 있으며 각각의 업무 처리 알고리즘 또는 로직을 가지고 있는 Application Layer가 있다.

본 연구에서는 협상이라는 문제를 해결하기 위해 협상안 평가 및 협상전략 생성, 협상안 작성, 스케줄링, 그리고 가격 및 Due date 계산을 담당하는 4가지 에이전트를 구현하였다.

#### 2.4.1 NegoAgent

상대 에이전트 시스템과 협상안을 교환하며 협상 문제를 해결해 나가는 에이전트로 협상에 필요한 협상안 평가, 협상전략 수립, 스케줄링에 대해 각 에이전트에게 요청한다. 또한 협상을 수행하기 전 서버 사양에 대한 견적을 상대 에이전트 시스템에게 요청할 수 있으며 반대로 상대의 요청에 견적 결과를 전달할 수도 있다.

#### 2.4.2 SCAgent

NegoAgent의 요청에 의해 현재 공장 내의 일정 계획을 수립하여 최소의 납기가능 시간을 계산하

여 이 결과를 NegoAgent에게 전달한다. 본 연구에서는 이러한 스케줄링 문제를 해결하기 위해 유전알고리즘[12]을 사용하였다. 이러한 유전알고리즘을 이용한 스케줄링을 수행하기 위해 우리는 서버의 조립을 위한 공정을 조사하였고 다음과 같은 공정계획을 수립할 수 있었다.

J1 : 메인보드 조립	RAM → CPU → IDE → SCSI
J2 : 케이스 조립	HDD → FDD → CD-ROM → Secondary Storage
J3 : 전체 부품 취합	메인보드 → 케이스 → 키보드 → 마우스 → 모니터

각 공정들에 필요한 부품들의 조립순서는 바뀔 수 있으며 메인보드(J1)와 케이스의 조립(J2)이 완료되어야만 전체 부품 취합(J3)을 수행할 수 있다.

이러한 서버 조립 과정에서 스케줄링 문제가 발생하는 이유는 서버 내의 부품들이 가격이 비싸고 이로 인해 재고를 충분히 가지고 있지 않기 때문이다. 서버의 조립 의뢰가 들어오면 판매자는 각 부품들의 재고를 확인하고 없을 경우 부품 업체들에게 이를 요청하여 재고를 확보하게 된다. 이러한 주문에 따른 배송 시간이 걸리므로 서버 판매업에서의 스케줄링이 필요하게 되었다.

#### 2.4.3 PDAgent

견적에 대한 가격을 산출하는 에이전트이다. 이러한 가격을 산출하기 위해 본 연구에서 개발한 PDAgent는 다음과 같은 기본적인 식을 사용한다.

$$\text{제품의 원가}(C) = \{(T_a \times P_a M) + (T_b \times P_b M)\} \times D \quad \text{<식 1>}$$

- $T_a$  : 기본 작업 시간
- $T_b$  : 초과 작업 시간
- $P_a M$  : 기본 작업 시 비용
- $P_b M$  : 초과 작업 시 비용
- $D$  : 작업일 수

위의 <식 1>을 통해 제품의 원가를 계산하고 반대로 가격에 따른 납기일을 계산하여 NegoAgent에게 전달하게 된다.

#### 2.4.4 NSAgent

본 연구에서는 협상 문제를 해결하기 위해 협상안 평가, 협상전략 수립의 기능을 수행하는 NSAgent를 구현하였다. 이 NSAgent는 MAFNS에서 제공하고 있는 NS를 이용하여 구현되었으며

NegoAgent의 요청에 따라 상대방의 협상안 평가 및 협상전략을 수립하게 된다.

NSAgent의 협상안 평가는 기본적으로 MADM(Multi-Attribute Decision Making, 다속성 의사결정 방법)을 사용한다. MADM 방법은 기준이 다른 척도를 가진 협상 항목들을 동일한 기준으로 정형화하여 각 협상안들을 서로 비교할 수 있도록 각 대안들에 대해 평가값을 보여주며 또한 각 항목들 간에 대해 주관적인 가중치를 부여할 수 있어 최적의 대안을 찾을 수 있도록 지원한다. MADM 방법을 본 논문에 적용하기 위해 다음과 같이 기호를 정의한다.

- $n$  : 전체 속성의 수 ( $n=3$ )
- $m$  : 전체 협상안의 수( $m=2$ )
- $A_i$  :  $i$ 번째 협상안,  $i=1, 2$ (1:상대방 협상안(구매자), 2:자신의 협상안(판매자))
- $C_j$  :  $j$ 번째 속성,  $j=1$  납기일,  $j=2$  가격,  $j=3$  무상수리기간
- $x_{ij}$  : 협상안  $A_i$ 의 속성  $C_j$ 에 대한 값,  $i$ 번째 협상안에서 가격과 납기일

$$D = \begin{matrix} & C_1 & C_2 & C_3 \\ \begin{matrix} A_1 \\ A_2 \end{matrix} & \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \end{bmatrix} \end{matrix}$$

$P_{ij}$  : 평가치  $x_{ij}$ 를 폐구간[0, 1] 상에서 속성별로 표준 정규화한 값,  $i=1,2, j=1,2,3$

$E_j$  : 속성  $C_j$ 에 대한 정규화값  $P_{ij}$ 의 엔트로피 값,

$$0 \leq E_j \leq 1, j=1, 2, 3$$

$d_j$  : 속성  $C_j$ 의 평가치에 의해 제공되는 정보에 대한 다양함의 정도,  $d_j=1-E_j, j=1,2,3$

$s_j$  : 의사결정자가 속성을 고려하여 설정한 주관적 가중치,  $0 \leq s_j \leq 1, j=1,2,3$

$w_j$  : 다양함의 정도  $d_j$ 에 의해 구해지는 정규화된 가중치,  $0 \leq w_j \leq 1, j=1,2,3$

$W_j^*$  : 엔트로피 척도에 의해 구해진 속성별 가중치,  $0 \leq W_j^* \leq 1, j=1,2,3$

$S_i$  :  $A_i$ 에 대한  $C_j$ 의  $P_{ij}$ 와 엔트로피 척도에 의한  $W_j^*$ 를 곱한 값들의 합,  $i=1,2, j=1,2,3$

MADM 방법에서는 다음과 같은 공식으로 가중치 변환과 엔트로피값을 구하게 된다.

$$p_{ij} = \frac{x_{ij}}{\sum_{i=1}^m x_{ij}} \quad \text{<식 2>}$$

$$E_j = -k \sum_{i=1}^m p_{ij} \ln p_{ij} \quad \text{<식 3>}$$

(k= 상수, 1/(ln m))

$$d_j = 1 - E_j \quad \text{<식 4>}$$

$$w_j = \frac{d_j}{\sum_{j=1}^n d_j} \quad \text{<식 5>}$$

$$W_j^* = \frac{s_j w_j}{\sum_{i=1}^n s_i w_i} \quad \text{<식 6>}$$

판매자가 구매자의 협상안을 <표 2>과 같이 받았을 경우 자신의 협상안과 구매자의 협상안을 MADM 방법으로 비교하는 절차는 아래와 같다.

<표 2> 협상안 비교

협상안	속성	협상안		
		C <sub>1</sub> (납기일)	C <sub>2</sub> (가격)	C <sub>3</sub> (무상수리기간)
A <sub>1</sub> (구매자의 협상안)	x <sub>11</sub> (20일)	x <sub>12</sub> (20000원)	x <sub>13</sub> (12개월)	
A <sub>2</sub> (판매자의 협상안)	x <sub>21</sub> (23일)	x <sub>22</sub> (18000원)	x <sub>23</sub> (11개월)	

<표 2>에서 A<sub>1</sub>은 구매자의 협상안으로써 납기일 20과 가격 20000, 그리고 무상수리기간 12개월의 값을 가지고 있다. 그리고 A<sub>2</sub>는 판매자의 협상안으로써 납기일 23과 가격 18000, 무상수리기간 11개월의 값을 가지고 있다. 납기일과 가격이 서로 상이한 기준을 가지고 있기 때문에 이 협상안들을 비교하기 위해서는 평가지표를 맞추기 위해 엔트로피법을 이용하여 동일한 기준으로 변환시켜야 한다. 엔트로피법을 이용하려면 먼저 표준 정규화값(p<sub>ij</sub>)을 구해야 된다. D를 <식 2>에 대입하여 x<sub>ij</sub>의 표준 정규화값 {p<sub>ij</sub>}를 구하면 <식 7>과 같이 나타난다.

$$P = \begin{pmatrix} 0.465 & 0.526 & 0.522 \\ 0.535 & 0.474 & 0.478 \end{pmatrix} \quad \text{<식 7>}$$

이 행렬 P를 <식 2, 3, 4, 5>를 이용하여 풀면 가격과 납기일에 대한 구매자의 E<sub>j</sub>, d<sub>j</sub>, w<sub>j</sub>, s<sub>j</sub>, W<sub>j</sub><sup>\*</sup>를 구할 수 있다.

<표 3> MADM 계산 결과

구분	C <sub>1</sub> (납기일)	C <sub>2</sub> (가격)	C <sub>3</sub> (무상보증기간)
E <sub>j</sub>	0.9964	0.9980	0.9986
d <sub>j</sub>	0.0036	0.002	0.0014
w <sub>j</sub>	0.5109	0.2906	0.1985
s <sub>j</sub>	0.30	0.50	0.20
W <sub>j</sub> <sup>*</sup>	0.4531	0.4296	0.1173

이렇게 구해진 엔트로피 값과 <식 7>의 단순가중합법(SAW; Simple Additive Weighting Method)을 이용하여 구매자의 협상안(A<sub>1</sub>)과 판매자의 협상안(A<sub>2</sub>)을 비교평가하게 된다.

$$A^* = \left\{ A_i \mid \max_i \left( \sum_{j=1}^n W_j^* x_{ij} / \sum_{j=1}^n W_j^* \right) \right\} \quad \text{<식 7>}$$

(A\*는 대안들 중에서의 최적안)

위의 <식 7>에 의해서 구해진 구매자의 협상안 평가 결과(S<sub>1</sub>)는 0.54688이고 판매자가 준비한 협상안의 값(S<sub>2</sub>) 0.45311에 비해 크므로 판매자는 구매자의 협상안에 대해서 수락하게 된다.

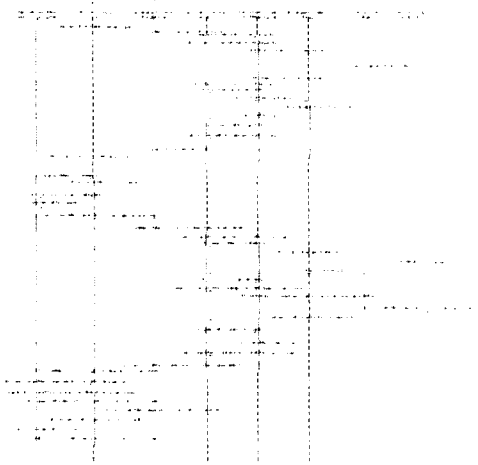
NSAgent는 위에서 본 것처럼 협상안들을 MADM 방법으로 평가하여 에이전트에게 그 결과를 제시하게 된다. 이러한 결과를 받은 NegoAgent는 상대방의 협상안을 받아들일 것인지를 판단하며 협상을 진행하게 된다.

또한 NSAgent는 협상안 평가 외에도 협상전략을 수립하는 기능을 가지고 있다. 총 몇 라운드에 걸쳐 협상을 진행할지 그리고 가격, 납기일, 무상보증기간을 각 라운드별로 어떻게 제시할지에 대해 전략을 세운다. 가격과 납기일은 PDAgent를 통해 최소납기가능 시간을 계산하고 하루 8시간을 기본 작업 수행 시간이라 가정 후 납기일을 계산한다. 그리고 납기일이 작아지면서 작업해야 하는 초과 작업시간을 계산하고 <식 1>을 이용하여 그에 따른 가격을 계산한다. 이때 1일 최대 작업 시간은 16시간으로 한정한다. 예를 들어 최소납기가능 시간이 100시간이고 기본 작업 비용이 100, 초과 작업 비용이 150이라 하면 납기일은 12.5일이 되고 가격은 80,000원이 된다. 또한 무상보증기간은 임의로 12개월 또는 6개월 2가지의 경우를 선택하여 제시하게 NSAgent를 구현하였다.

### 3. 사례 적용

본 장에서는 개발된 SSAS의 유효성 및 사용 가

능성을 판단하기 위해 실제 서버 판매 협상 문제를 적용시켜 보았다. 실제 Facilitator와 각 Task 에이전트들이 협상문제를 해결하기 위해 서로 메시지를 주고받는 프로세스는 다음의 그림과 같다.



<그림 7> SSAS의 협상 프로세스

먼저 구매자는 Server Type 1의 사양에 대해 견적을 요청한다. 견적 요청을 받은 판매자 NegoAgent는 SCAgent에게 스케줄링을 요청하여 최소 납기가능 시간을 받아오고 이 납기가능 시간을 PDAgent에게 보내어 그에 따른 가격을 받아온다. NegoAgent는 납기가능일과 가격을 포함한 견적 결과를 구매자 NegoAgent에게 전달하게 되고 구매자 NegoAgent는 NSAgent에게 협상전략을 요청하고 본격적으로 협상을 수행할 준비를 한다.

구매자가 에이전트 시스템을 통해 세운 협상전략은 다음 <표 4>와 같다.

<표 4> 구매자의 협상전략

R	가격 (0.5)	납기일 (0.3)	무상수리기간 (0.2)	R	가격 (0.5)	납기일 (0.3)	무상수리기간 (0.2)
1	18000	10	12개월	7	16800	13	6개월
2	17800	10.5	12개월	8	16600	13.5	6개월
3	17600	11	12개월	9	16400	14	6개월
4	17400	11.5	12개월	10	16200	14.5	6개월
5	17200	12	12개월	11	16000	15	6개월
6	17000	12.5	12개월				

이렇게 세워진 협상전략에 의해서 구매자 에이전트 시스템은 판매자 에이전트 시스템에게 첫 번째 협상안으로 가격 18,000원, 납기일 10일, 무상수

리기간 12개월을 작성하여 전달하게 된다. 판매자 에이전트 시스템은 구매자 에이전트로 받은 최초 협상안을 받고 협상전략을 수립하게 된다. 판매자 에이전트 시스템은 다음과 같이 협상전략을 수립하였다.

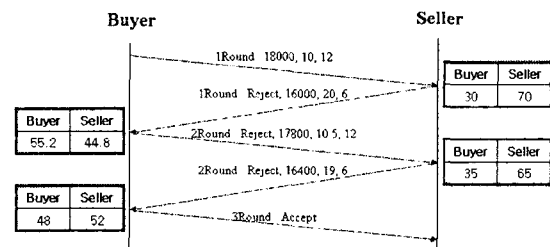
<표 5> 판매자의 협상전략

R	가격 (0.3)	납기일 (0.5)	무상수리기간 (0.2)	R	가격 (0.3)	납기일 (0.5)	무상수리기간 (0.2)
1	16000	20	6개월	7	18400	14	12개월
2	16400	19	6개월	8	18800	13	12개월
3	16800	18	6개월	9	18200	12	12개월
4	17200	17	6개월	10	18600	11	12개월
5	17600	16	6개월	11	20000	10	12개월
6	18000	15	6개월				

판매자 에이전트 시스템은 구매자의 첫 번째 협상안인 18,000원, 납기일 10일, 무상수리기간 12개월을 자신의 1라운드 협상안인 16,000원, 20일, 6개월과 비교하게 된다. NSAgent는 상대방의 협상안과 자신의 협상안을 비교해 본 결과 상대방 협상안은 30점, 자신의 협상안은 70점이 나타나는 것으로 판단하고 상대방의 협상안을 거절할 것을 NegoAgent에게 요청한다. 이런 거절요청을 받은 NegoAgent는 첫 번째 협상안인 16,000원, 20, 6개월을 구매자 에이전트 시스템에게 전달하게 된다.

구매자 에이전트 시스템은 판매자 에이전트 시스템의 협상안과 자신의 2번째 협상안을 NSAgent를 통해 평가하게 된다. 구매자 에이전트 시스템의 NSAgent는 판매자의 협상안을 44.8점, 자신의 협상안을 55.2점으로 평가하고 NegoAgent에게 거절할 것을 알린다.

이와 같은 프로세스로 구매자와 판매자 에이전트 시스템은 자동으로 협상을 수행하게 되며 다음과 같은 결과를 얻게 된다.



<그림 8> 협상 과정 및 결과

구매자 에이전트 시스템은 판매자의 2번째 협상

안인 16,400원, 19일, 6개월의 협상안을 자신의 3번째 협상안인 17,600원, 11, 12개월보다 좋다고 판단하여 협상을 완료하게 된다.

## V. 결론

협상이라는 거래 프로세스 자체가 환경변화에 민감하고 또한 고려해야 할 사항들이 많아 연구에 많은 어려움이 따랐다. 그렇기 때문에 이렇게 많은 어려움이 따르는 협상 문제를 다룰 수 있는 시스템의 개발은 거의 없었다. 이미 연구되었던 협상 시스템들도 대부분 협상을 지원하기 위한 협상지원 시스템이었으며 기존의 Kasbah나 Tete-a-Tete와 같은 자동 협상 시스템들 역시도 이러한 자동 협상 프로세스를 완벽하게 지원하고 있지는 않다. 본 연구에서는 협상전략 수립 및 협상안 평가, 협상안 생성을 자동으로 수행하여 사람의 개입 없이 협상을 수행할 수 있는 자동 협상 시스템을 개발할 수 있도록 지원해 주고 있는 MAFNS를 이용하여 SSAS를 개발하였다. 그리고, SSAS를 실제 서버 판매 도메인에서 발생하는 협상문제를 적용하여 그 적절성과 효용성을 확인하였다. 본 연구는 자동 협상 시스템 개발의 가능성을 제시하였고, 더욱 진보된 자동 협상 시스템 개발의 기반을 마련하였다.

향후 우리는 MAFNS를 이용하여 다른 도메인에서의 협상 문제를 해결할 수 있는 SSAS와 같은 몇몇 자동협상시스템을 개발하고 그 사용 가능성 및 적용 가능성을 한번더 확인할 것이다. 또한 현재 MAFNS에서 제공하고 있는 NS의 기능을 확장시켜 MADM 외에 다른 평가 방법론을 적용해 볼 것이며 협상전략 수립 역시 여러 가지 방법을 고안해 적용할 것이다. 이렇게 여러 가지 방법론이 적용되면 이러한 방법론들을 이용한 협상의 결과를 비교하여 적용 도메인에 따른 적절한 평가 및 전략 수립 방법론을 선택할 수 있도록 제시할 것이다.

## 관련 문헌

[1] Beam, C. and Segev, A. "Automated Negotiations: A Survey of the State of the Art," *CMIT Working Paper 97-WP-1022*, 1997.

[2] Chavez, Anthony, and Maes, P. "Kasbah: An Agent Marketplace for Buying and Selling Goods," *MIT Media Lab*, July 1996.

[3] Lennon, J., Liu, H. "HWONS: A Hyperwave Online Negotiation System," *Proceedings of E-Learn '02*, Montreal, AACE, 2002, pp. 573-577.

[4] Suarga, Y.Y., Rose, J.B. and Archer, N. "A Web-based Negotiation Support System," *International Journal of Electronic Markets* (8:3), 1998.

[5] Maes, P., Guttman, R.H. and Moukas, A.G. "Agent that Buy and Sell: Transforming Commerce as we Know It," *MIT Media Lab*, March 1998.

[6] Auction Bot, <http://auction.eecs.umich.edu>

[7] Tete-a-Tete, <http://ecommerce.media.mit.edu/tete-a-tete>

[8] Paula, E. G., Ramos, F. S. and Ramalho, G. L. "Bilateral Negotiation Model for Agent-Mediated Electronic Commerce," *Agent-Mediated Electronic Commerce III*, Springer-Verlag, 2001.

[9] Choi, H.R., Kim, H.S., Hong, S.G., Park, Y.J., Park, Y.S., Kang, M.H. "A Design of Multi-Agent Framework to Develop Negotiation Systems," *Journal of Intelligent Information Systems* (9:2), 2003, pp. 155-169.

[10] Faratin, P., Sierra, C., Jennings, N.R. "Using similarity criteria to make issue trade-offs in automated negotiations," *Artificial Intelligence 142*, ELSEVIER, 2002, pp. 205-237.

[11] Fatima, S.S., Michael, W., Jennings, N.R. "An agenda-based framework for multi-issue negotiation," *Artificial Intelligence 152*, ELSEVIER, 2004, pp. 1-45.

[12] Park, B.J., H.R., Choi and H.S., Kim, "A hybrid genetic algorithms for job shop scheduling problems", *Computers and Industrial Engineering*, Vol.45(2003), pp.597-613.