

실시간 다중 에이전트 환경에서 동적 역할 조합과 배정

박근수^o, 권기덕, 김인철

경기대학교 게임웨어학과^o, 전자계산학과

pksu21c@hanmail.net^o, {kdkwon, kic}@kyonggi.ac.kr

Dynamic Role Combination and Assignment in Real-time, Multi-Agent Environments

Gun-Soo Park^o, Ki-Duk Kwon, In-Cheol Kim

Dept. Gameware^o, Dept. Computer Science, Kyonggi University

요 약

현재의 일반적인 다중에이전트 시스템 환경은 실시간이며 복잡한 환경을 제공 한다. 또한 제한적인 통신 환경을 제공한다. 본 논문의 테스트 환경인 Unreal Tournament의 환경은 일반적인 다중 에이전트 시스템 환경을 제공한다. UT 게임의 GameBots 시스템에서 실시간 다중 에이전트 협상 시스템의 구현을 위한 ACL(Agent Communication Language)을 정의하였으며 그에 따른 다중 에이전트 협상 프로토콜을 정의하였다. 통신 환경은 단일 채널 환경이며 제한적인 통신을 제공한다. 에이전트들은 게임 시작과 동시에 인지 정보를 기반으로 맵을 작성 하게 된다. UT 게임은 환경이 실시간으로 급변하기 때문에 최단의 협상 과정을 가져야 한다. 협상 시스템의 구성은 협상 과정에서 모든 것을 정하기엔 시간이 부족하기 때문에 빠른 협상 과정을 유도하기 위하여 협상과정의 일부분을 사전에 정의함으로써 협상과정을 단순화 시켰으며, 나머지는 실시간 협상과정을 통하여 동적으로 역할 분담을 하였다. 협상 방법으로는 각 에이전트의 의견이 반영될 수 있는 투표(voting) 방법을 사용하였다.

1. 서 론

실시간 환경에서 에이전트 간의 협상을 바탕으로 작업하는 다중 에이전트 협상 시스템의 중요성이 점차 확대 되어 가고 있다. 이에 따른 실시간 다중 에이전트 환경에서의 동적 역할 조합 및 역할 배정에 대해서 연구하고자 한다. 이를 위해 실제 환경과 비슷한 가상 환경을 제공해 주는 UT(Unreal Tournament)게임을 테스트 환경으로 선택 하여 다중 에이전트 간의 협상 프로토콜을 정의하고 팀 메시지를 통해 이를 실험하였다. 또한 다중 에이전트간의 협상 과정에서 사전정의 작업을 통하여 협상 과정의 최소화를 실현 하였다. 불확실한 정보를 바탕으로 역할 배정을 해야 하므로 투표(Voting) 방법을 사용하였으며 팀원의 평가가 역할 배정에 반영하도록 협상 과정을 구현 하였다.

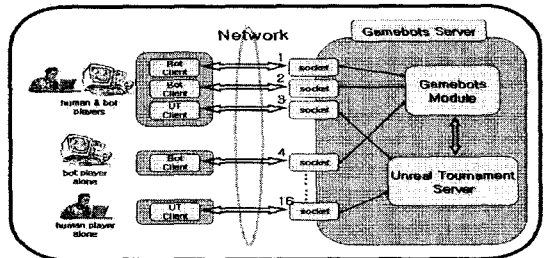
2. 관련 연구

팀버(Timber)는 실시간이며 동적인 환경에서의 팀워크 구성을 연구 하였다. 이 시스템의 환경은 여러 팀이 존재하고 각 에이전트들마다 고유의 역할 및 기능을 가지고 있으며 1000여 가지가 넘는 역할이 존재한다. 이러한 많은 역할을 줄이면서 위험 요소를 줄이는 데 중점을 두었다. 또한 실시간 환경에서 실행하다 보면 실행된 역할이 없어지는 경우가 발생하였는데 이를 자동적으로 처리해주는 루틴을 통해 해결한다. 역할을 실행 시키는 방법에는 단순 학습을 통한 방법과 휴리스틱을 이용하는 방법을 사용 하였다. 이 시스템에서의 팀 레벨 전략은 카테고리 방법을 통해 휴리스틱한 것을 선택하도록 하였으며, 에이전트의 각 행위는 학습의 결과를 이용하여 적용하였다. 다른 시스템과 구별 되는 것은 복구(Recovery) 알고리즘을 이용하여 행위에 대해서 보장한 것이다[3]. 피터 스톤(peter stone)이 제안한 로봇 축구 시스템은 팀 협상과 팀워크 구조 기반의 통신 페러다임을 지원한다. 로봇 축구 시스템은 팀 포메이션(Team Formation)과 역할 배정(Role Assignment)을 경기 전 Locker-Room Agreement에 의해서 미리 정해 놓는다. 경기 중 상황에 따라 Locker-Room Agreement에

서 미리 정해 놓은 포메이션과 역할에 의해 역할 배정을 한다. 게임 진행 중 시간과 점수에 따라 사용 가능한 포메이션을 동적으로 바꿔게 된다. 특히 포메이션에서 위치(position)의 변화로 팀 성능을 개량 할 수 있으며 Set-Play를 지원하여 좀 더 유연한 팀워크를 구성하였다[4].

역할 결정방법에는 중앙 집중적인 결정 방법과 비중앙적인 역할 결정 방법이 있다. 중앙 집중적인 결정 방법에는 적은 비용이 소요되며 리더에게 모든 정보가 존재하기 때문에 정확한 판단을 할 수 있다. 그러나 리더가 독단적인 판단을 하여 잘못 판단할 경우 큰 문제를 발생 시킬 수 있다. 비 중앙 집중적인 역할 결정 방법의 장점은 모든 에이전트가 동등한 입장에서 의사 결정에 모두 참여 할 수 있다는 것이다. 그러나 의사 결정에 많은 비용이 소요되는 단점도 있다. 본 논문에서는 실시간 다중 에이전트 협상 팀 구성을 위해 상태에 따른 동적 역할 조합을 통한 포메이션을 구성하였으며, 협상 과정을 통하여 역할 배정을 하였다.

3. 실시간 다중 에이전트 환경

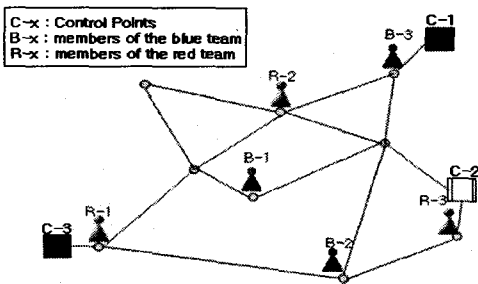


[그림 1] GameBots 시스템

실시간 다중 에이전트 시스템 환경은 실시간이며 복잡한 환경과 제한적인 통신 환경을 제공한다. 본 논문의 테스트 환경인 UT 게임은 다양한 맵과 다양한 개

입타입, 멀티 플레이, 제한적인 통신 환경을 제공하여 실제 환경과 비슷한 가상환경을 제공하게 된다. [그림 1]의 GameBots 시스템은 지능형 에이전트 및 다중 에이전트를 연구하기 위한 목적으로 USC의 ISI와 CMU가 공동으로 만든 테스트 환경이다. 특히 기존의 환경에서와는 달리 한 대의 컴퓨터에 동시에 Client Bot와 사용자가 접속하여 테스트 할 수 있고 다양한 분석 툴을 제공하고 있으며, GameBots의 모듈들과 UT 게임 서버는 분리 되어있다. 게임 정보는 기존에 게임에서의 NPC(Non Player Charter)처럼 게임의 모든 정보를 보내주는 것이 아니라 일반 사용자와 똑같은 정보가 에이전트들에게 주어진다.[1] 이를 기반으로 GameBots 환경에서 동작하는 KGBot이 구현 되었다.

현재 GameBots 시스템은 VTM(Virtual Team Message) 형태로써 팀 메시지를 지원해주고 있다. 이 메시지는 게임 내에서 채팅으로 이용하는 것인데 이를 이용하여 팀원 간에 의견이나 정보를 주고받게 된다. VTM 에는 몇 가지 제약 사항이 있다. 메시지의 길이가 256byte로 제한되어 가지고 있으며 1 초당 2~3회 정도만 메시지의 안정성이 보장된다. 이런 제약적인 환경에서 팀간의 메시지를 주고받기 때문에 얻어지는 정보의 완벽한 동기화는 불가능 하며 최대한 메시지를 줄이면서 게임내의 핵심 정보를 주고받아야만 한다.



[그림 2] 두 팀간의 도메이션 게임

[그림 2]는 본 연구에서 사용된 도메이션 게임에서의 한 가지 상황을 나타내고 있다. C-1, C-2, C-3은 점령해야할 지점(Control Point)을 나타내는 것이다. Control Point의 색은 각 팀이 점령했을 때 그 팀의 색으로 바뀌며 흰색은 아직 점령하지 못한 곳이다. R-1, R-2, R-3은 홍팀(red team), B-1, B-2, B-3은 청팀(blue team) 에이전트 각각을 나타내는 것이다. 실시간이며 다수의 에이전트가 동시에 경기 진행을 하고 여러 개의 목표지점이 존재하게 된다. 상황에 따라 팀 원간에 효과적으로 역할 분담을 하여 Control Point의 점령 및 수비를 하는가가 팀 승리에 중요한 역할을 한다.

4. 팀워크 구조 설계

효과적인 팀워크 구성을 위해 ACL을 정의 하였다. 현 상태정보와 각 에이전트의 상태 정보들을 고려하여 동적으로 역할을 조합하여 포메이션을 구성하였다. 또한 사전 정의 작업을 통해 협상 과정을 최소화시키기 위해 또한 협상 프로토콜은 투표 방법을 사용하였으며 중재 에이전트를 두어 실시간 협상 과정을 통해 동적으로 역할 배정을 하였다.

4.1 팀 구성원 에이전트

각 에이전트는 실시간 환경에서 반응을 할 수 있는 지능형 에이전트이며 월드 탐색과 Control Point에 대한

점령 및 수비, 적에 대한 공격 기능을 가지고 있다. 이런 공통적인 구조 및 기능을 가진 에이전트들로 이루어진 팀에 Control Point의 점령 수와 각 에이전트의 골, 팀 점수를 고려하여 팀 포메이션을 정한다. 정해진 팀 포메이션에 대해 팀 원간에 역할을 분담하고 효과적인 전투 수행을 하는 것이 목적이다. 역할 분담 시 게임 상황에 따른 팀 레벨의 골과 각 에이전트의 골을 조화 시키는 것이 중요하다.

4.2 상태

게임의 상태는 게임 서버에서 알려주는 정보를 기반으로 하여 월드 맵에 현재 게임 상태 정보 및 에이전트 상태 정보, 노드 맵 정보를 저장한다.

- Sc : 현재 팀 점수
- Sce : 현재 상대 팀 점수
- CP= {cp1 , cp2 , cp3} : Control Point 상태
- AS= {as1, as2 ..., asn} : n명의 각 에이전트 자신의 정보
- S= {AS, CP, Sc, Sce} 현재 월드 맵의 상태

예를 들어, 상태가
 에이전트 수 : 3개
 팀 점수 : 45
 상대 팀 점수 : 55
 Control Point 1 : 점령
 Control Point 2 : 미점령
 Control Point 3 : 미점령 일 경우
 $S = \{ as_1, as_2, as_3, \{ 0, 1, 1 \}, 45, 55 \}$ 로 표현 가능하다.

4.3 동적 역할 조합과 배정

- $A = \{ a_1, a_2, \dots, a_n \}$: n명의 에이전트들로 구성된 팀
- $R = \{ r_1, r_2, \dots, r_k \}$: k개의 서로 다른 역할들
- $r : R_{i-1} \times S \rightarrow R_i$: 역할 변이 함수(role transition function)

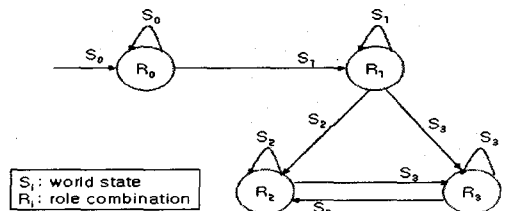
- $(R_{i-1} : \text{이전의 역할조합}, R_{i-1} \subset R,$
- $S : \text{환경상태},$
- $R_i : \text{새로운 역할조합}, R_i \subset R)$

$m : A \rightarrow R$: 역할 배정 함수(role assignment function)

$F_i = \langle A_i, R_i, m_i \rangle$: 팀 포메이션(team formation)

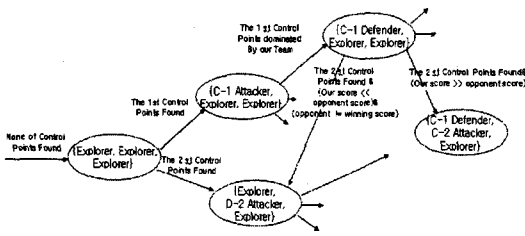
- A_i : 에이전트의 부분집합,
- R_i : 역할조합, 즉 역할들의 부분집합,
- $m_i : A_i \rightarrow R_i$: 역할 배정 함수)

- 역할 변이 함수 r과 역할 배정 함수 m의 구현방식 :
 - r은 사전에 에이전트간의 합의, [그림 3]과 같은 역할 변이도(role transition diagram)로 표현
 - m은 에이전트간의 실시간 협상에 의해 결정



[그림 3] 역할 변이도

[그림 4]는 상황에 따른 팀 포메이션의 변이도 일부를 나타내고 있다. 팀의 역할 집합은 F_i 는 에이전트 수와 사용 가능한 역할, 현재 상태를 고려하여 팀 포메이션을 정한다.



[그림 4] 도미네이션 게임을 위한 팀 포메이션의 변이도

4.4 역할 배정을 위한 에이전트 간 협상 프로토콜

다중 에이전트 시스템의 핵심은 에이전트 상호간의 협력 또는 경쟁 상황에서 공동의 목표를 위해 에이전트 간 조정 및 협상을 통해 최상의 역할 분담을 하는 것이다. 협상은 간단하고, 효과적, 안정적, 분산적, 균형적이어야 한다. 협상 프로토콜의 유형에는 불확실한 정보를 바탕으로 구성된 각 에이전트의 정보가 의사 결정에 반영이 가능한 투표(Voting)방법과 한 가지 사실에 대해 각 에이전트가 판단을 하여 하나의 에이전트 의견만이 반영되는 경매(Auction)방법, 그리고 하나의 사실에 대해 어떤 에이전트에게 얼마만큼 이익을 내는지가 중심이 되는 거래(Bargaining) 등의 프로토콜이 존재한다. 본 논문에서는 불확실한 정보를 바탕으로 역할 분담을 결정해야 한다. 그러므로 다중 에이전트 간의 협상을 위해 중재 에이전트를 통한 투표 방법을 사용하였다. 투표란 현안에 대해서 각 에이전트들이 자신의 의사를 밝히고 가장 많이 나온 의견을 따르는 방법을 말한다. 투표는 에이전트의 숫자가 많거나 에이전트 수를 알지 못하는 경우에 널리 사용된다. 장점은 많은 에이전트들이 있을 경우 의사 결정에 용의하며 그 결정 사항이 직접 반영이 된다. 또한 에이전트가 동등한 입장에서 의견을 표출하기 때문에 공평하고 분산적이다. 그러나 많은 사전 정의 작업과 중재 에이전트가 필요하다. 중재 에이전트의 역할 결정은 가장 최근에 Control Point를 점령했던 에이전트가 된다. Control Point 상황이 바뀌었을 때 중재 에이전트는 협상 과정의 시작 및 협상 과정을 진행하고 그 결과를 팀원들에게 통보 해준다. 본 협상 프로토콜에서 사용되는 중재 에이전트 역할은 기존의 중재 에이전트와 달리 중재 에이전트가 역할 분담에 의사 결정을 하는 것이 아니라 각 에이전트의 의견을 수렴하여 의견의 결과를 알려주는 통보 역할과 협상 과정이 이끌어 가는 중재역할을 가지고 있다. 물론 중재 에이전트도 다른 에이전트와 동등한 입장에서 의사 결정에 참여 한다. 단 협상 결과가 역할 배정의 결과로 부적당할 경우 중재 에이전트의 의견이 역할 배정에 반영된다.

4.5 에이전트 간의 통신

다중 에이전트 간의 통신환경은 UT 게임에서 제공하는 에이전트 간의 통신을 위해서 VTM 메시지를 이용하였다. VTM의 제약 조건을 고려하여 메시지를 설계 하였으며, 본 논문의 환경이 빠르고 복잡한 환경이기 때문에 그에 따른 효과적인 협상 과정을 필요로 한다.

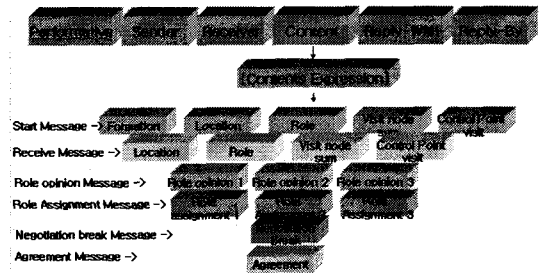
4.5.1 메시지 설계 시 고려사항

본 논문에서 사용하는 통신 환경은 단일 채널만을 지원한다. VTM 메시지는 1회에 256byte 길이의 메시

지를 보낼 수 있기 때문에 다음과 같은 ACL(Agent Communication Language) 구조 설계를 하였다. [그림 5]는 협상에 필요한 ACL을 구현한 일부분 나타내는 구성도이다. 그리고 게임의 상황이 실시간으로 바뀌기 때문에 협상 과정 중 상황이 변해 협상 과정이 중단되는 경우가 자주 발생하게 된다. 이를 효율적으로 보장하기 위해 다음의 제약 조건이 필요하다.

1. 1초에 2회의 메시지만이 보장되기 때문에 협상 메시지를 우선순위에서 가장 높게 배정하는 것이 필요하다.
2. 최단의 협상 과정이 필요하다.
3. 협상 단계는 최대 10단계를 넘지 않아야 한다.
4. 협상 과정에 중재 에이전트가 필요하다.

Control Point의 변화 발생시 가장 최근에 Control Point를 점령한 에이전트가 중재 에이전트가 되어 협상을 해야 하고, 메시지를 팀원 에이전트에게 전달한다.



[그림 5] 구현한 ACL의 구성도

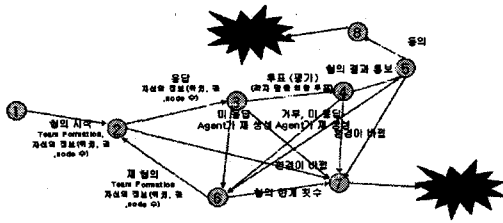
예를 들어, 상태가 사용할 포메이션 : 00 자신의 위치정보 : (432.1341, 1031.1342, 13.5361) 자신의 역할 : Control Point 1 점령 방문 노드 수 : 34개 Control Point 방문여부 : 방문하지 않을 경우 메시지 content 내용 : {'FMAT" 00, "LACA" 432.1341 1031.1342 13.5361, "ROLE" CPA1, "VTND" 34, "CPVT" 0}.

4.5.2 협상 방법

투표 방법을 통해서 팀원의 의견이 최대한 반영될 수 있도록 하였다. 역할 분담을 하지 못하는 결과 값이 나올 경우에는 중재 에이전트의 의견을 따르게 된다. 각각의 에이전트가 현재 상황에 대해 평가한다. 협상 단계 시 수신자를 팀원 전체로 하여 협상 단계를 최소화 및 단순화 한다.

4.5.3 협상 과정

Control Point의 점령 상태에 변화가 생기면 협상을 시작한다. Control Point를 최근에 점령한 에이전트가 협상을 시작하며 중재 에이전트의 역할을 맡게 된다. [그림 6]는 협상 과정을 보여준다. 협상을 시작하는 중재 에이전트는 협상의 시작과 자신의 상태 정보를 보내준다. 이에 각 에이전트들은 자신의 상태 정보를 각 팀원들에게 보내준다. 각 팀원들이 정보를 모두 받으면 각각의 에이전트들은 역할 분담에 대해 평가하여 자신의 의견(평가 값)을 중재 에이전트에게 보낸다. 중재 에이전트는 각 에이전트의 의견을 바탕으로 역할 배정을 하며 팀원들에게 역할 배정에 대해 통보를 하게 된다. 배정 받은 역할이 행동 가능하면 동의한다는 메시지를 중재 에이전트에게 보낸다. 모두 동의 한다면 협상은 성공한 것이다.



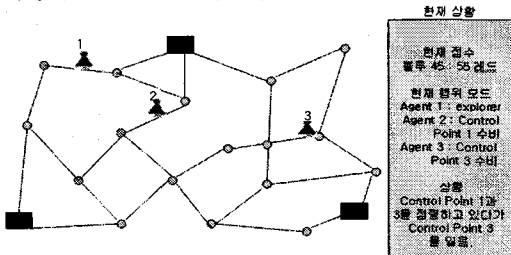
[그림 6] 협상 과정

4.5.4 협상 과정에서의 장애 회복

협상 과정에서는 애기치 않은 장애가 발생하게 된다. 따라서 이에 대한 대처가 필요하다. 또한 그에 따른 환경 상태 변화가 있을 수 있다. 본 논문의 환경에서는 메시지 전송의 지연을 허용 할 수 없다. 그래서 메시지의 전송 지연과 손실은 일정 시간 동안(2초) 협상 과정이 진행 되지 못할 경우 재협상 과정에 들어간다. 환경 변화에는 Control Point의 상태 변화와 에이전트가 소멸될 경우가 있다. Control Point의 상태 변화가 있을 경우에는 현재 협상 과정을 멈추고 바뀐 환경을 고려하여 새로운 협상 과정에 들어간다. 에이전트가 소멸되는 경우가 발생할 경우 에이전트 생성 시 위치가 바뀌기 때문에 협상 과정을 중단하고 재협상 과정에 들어간다. 단, 한계 협상 횟수를 두어 그 횟수를 넘는 경우에는 협상 과정을 중단하고 새로운 협상 과정에 들어간다.

5. 평가

UT게임은 실시간 다중 에이전트 환경이며, 실시간이고 물리법칙이 적용된 3차원 공간으로 제한적인 정보 습득을 한다. 따라서 팀 원간의 통신은 제한적이며 잡음이 발생 할 수도 있다. 각 에이전트는 탐색 과정에서 제한적인 정보 습득을 통하여 맵을 작성 하게 되는데, 맵 작성의 완성도가 승패에 큰 영향을 미치게 된다. 팀원 간 통신의 제약으로 인하여 각 에이전트는 맵 작성 시 맵의 공유는 불가능하다. 협상 과정에서 모든 것을 정하기엔 시간이 부족하기 때문에 빠른 협상 과정을 유도하기 위하여 협상과정의 일부분을 사전에 정의하여 협상과정을 단순화 시켰으며, 나머지는 실시간 협상과정을 통하여 동적으로 역할 분담을 하였다. 3:3 게임을 기본으로 팀 전략이 적용된 팀과 그렇지 않은 팀을 다양한 맵에서 시험해 보았다. 팀 전략 및 협상과정의 횟수와 성공 여부를 평가하여 협상 과정이 효과적으로 이루어졌는지 평가 하였다.



[그림 7] 협상과정의 상황 예제

예를 들어, [그림 7]과 같이 청팀의 상황은 Control Point 1을 점령하고 있고 Control Point 2는 홍팀이 점령, Control Point 3을 홍팀이 방금 점령하였다. 에이전트 1, 2, 3이 있고 현재 위치는 [그림 7]과 같다.

에이전트 3이 최근에 Control Point 3을 점령한 뒤 수비를 하고 있고 에이전트 1은 탐색을 하고 있으며 에이전트 2는 Control Point 1에 대한 수비를 하고 있다. 이 때 최근에 Control Point를 점령한 에이전트는 에이전트 3이다.

따라서, 에이전트 3은 중재 에이전트가 된다. 에이전트 3은 현재 상황에 사용 가능한 포메이션과 자신의 상태 정보를 팀원들에게 알려준다. 팀원들은 각각 자신의 상태 정보를 팀원들에게 알려준다. 팀원들의 정보를 모두 받은 후 에이전트 각각은 자신의 역할 분담에 대한 평가를 하여 중재 에이전트에게 평가 값을 보내준다. 그 평가 값이 [표 1]과 같다면 협상된 결과에 따라 팀원들에게 역할 배정을 통보 해준다. 이에 동의 한다면 동의의 메시지를 팀원들에게 보내고 협상은 성공적으로 끝마치게 된다.

	Control Point 1 수비	Control Point 2 점령	Control Point 3 점령
에이전트 1	에이전트 1	에이전트 2	에이전트 3
에이전트 2	에이전트 2	에이전트 1	에이전트 3
에이전트 3	에이전트 1	에이전트 2	에이전트 3
결론	에이전트 1	에이전트 2	에이전트 3

[표 1] 협상의 결과

6. 결론

다중에이전트의 일반적인 환경은 실시간이며 복잡하고 제한적인 통신 환경을 제공한다. 본 논문의 테스트 환경인 UT 게임에서 팀원 간에 통신이 가능하도록 제공되는 VTM 메시지를 이용하였으며 통신 환경의 제약을 극복하기 위해 효과적인 ACL정의의 하였다. 본 환경에서는 빠른 협상 과정이 필요하기 때문에 사전 정의 작업을 통하여 협상 과정을 단축하였다. 그러나 대부분은 실시간으로 협상 과정을 통해 정하도록 구현 하였다. 협상 프로토콜은 불확실한 정보에 바탕이 되는 환경에서 각 에이전트의 정보를 이용하여 의사 결정 반영이 가능한 투표(Voting)방법을 사용하였다. 협상 과정에 중재 에이전트를 두었으며 실시간 협상과정을 통해 동적 역할 배정을 구현 하였다. 협상 과정은 최근에 Control Point를 점령한 에이전트가 중재 에이전트가 되어 협상 과정을 이끌어 나가며 각 에이전트들의 의견을 모아 그 결과를 통보하는 역할을 한다. 단 협상한 결과 역할 배정에 대한 결론이 나오지 않을 경우에는 중재 에이전트의 의견을 존중하여 역할을 배정한다. 이로써 좀더 효율적인 다중 에이전트의 동적 역할 배정을 할 수 있었다.

참고문헌

- [1] Gal A. Kaminka, et al, "GameBots : A Flexible Test Bed for MultiAgent Team Research", Communications of ACM, Vol.45, No.1, pp.43-45, 2002.
- [2] Gerhard Weiss, MultiAgent Systems, MIT Press, 1999.
- [3] Milind Tambe, Teamwork in Real-World Dynamic Environments, ICMAS-96, pp.361-368, 1996.
- [4] Peter Stone, Layered Learning in MultiAgent Systems, MIT Press, 2000.
- [5] William van der Sterren, "Squad Tactics : Team AI and Emergent Maneuvers", In Steve Rabin, editor, AI Game Programming Wisdom, Charles River Media Inc, 2002.