

가상 에이전트 팀을 위한 동적 역할 배정

황중근 김인철
경기대학교 전자계산학과
(wargen, kic)@kyonggi.ac.kr

Dynamic Role Assignment for Virtual Agent Teams

Jong kun. Hwang, In-Cheol Kim
Department of Computer Science, Kyonggi University

요 약

본 논문에서는 컴퓨터 게임 환경에서 에이전트들의 효율적인 협력과 팀워크를 위한 동적 역할 배정 방법을 제시한다. 본 논문에서 제시하는 동적 역할 배정 방식은 설계단계 혹은 실행단계에서 역할 배정에 필요한 모든 사항을 결정하는 기존의 정적 역할 배정 방식이나 동적 역할 배정 방식과는 다른 새로운 역할 배정 방식이다. 이 역할 배정 방식에 따르면, 설계단계에서 가능한 역할집합들을 미리 결정하지만, 이 역할집합에 명시된 각각의 역할을 어느 에이전트가 맡아 수행할 것인가는 실행단계에 가서 에이전트들의 경매를 통해서 결정한다. 이 방법은 역할배정에 필요한 실행단계의 협상을 최소화할 수 있는 방법으로서, 실시간 멀티 에이전트 환경에 매우 효과적인 방법이다. 본 논문에서는 실험을 통해 새로운 동적 역할 배정 방식의 우수성을 증명해 보았다.

1. 서론

지금까지 멀티 에이전트 분야에서 시도되어 온 역할배정 방식은 주로 정적 역할배정 방식으로서, 각 에이전트에 대한 역할배정이 실행 이전에 시스템 설계자에 의해 이루어지고, 실행 중에는 역할변경이 이루어지지 않는 방식이다. 이 방식이 성공하기 위해서는 설계자가 사전에 필요한 모든 역할들과 역할 수행자를 파악할 수 있어야 한다. 하지만 동적으로 빠르게 변화하는 환경에서는 실행에 필요한 모든 역할과 역할 배정을 사전에 결정하기 매우 어렵다. 따라서 정적 역할배정 방식은 빠르게 변화하는 실시간 멀티 에이전트 환경에서는 그 효과를 기대하기 어렵다. 본 논문에서는 대표적인 멀티 에이전트 컴퓨터 게임 환경인 Unreal Tournament(UT) 환경에서 지능형 캐릭터 에이전트들의 효율적인 협력과 팀워크를 위하여, 새로운 동적 역할배정 방법을 설계, 구현하였다.

2. 멀티 에이전트 컴퓨터 게임 환경

Unreal Tournament 게임은 대표적인 3D 일인칭 액션게임으로서 이 게임은 온라인 상용게임이면서도 대부분의 소스가 공개되어 있다. UT게임은 견고한 3D 시뮬레이션 엔진, 다양한 게임 유형과 맵을 제공하고 있으며, 게임의 확장성을 위해 전용 스크립트 언어인 Unreal Script도 제공하고 있다. Gamebots 시스템은 지능형 에이전트 및 멀티 에이전트 연구를 목적으로 USC의 ISI연구소와 CMU에 의해 공동으로 개발된 UT게임 기반의 지능형 에이전트 연구용 테스트베드이다. Gamebots시스템은 네트워크를 통해 NPC를 제어하는 원격의 보트 클라이언트(bot client)들과 Gamebots 서버로 구성된다. Gamebots 서버모듈은 원격의 보트 클라이언트들에게 휴먼 플레이어와 동일한 센서정보를 제공하고 역으로 보트 클라이언트의 행동명령을 받아 실행하는 역할을 한다.

Gamebots 서버가 원격의 에이전트에게 전달하는 메시지들은 크게 초기화 메시지(initialization message), 동기

메시지(synchronous message), 비동기 메시지(asynchronous message)로 구성된다. 각 에이전트의 연결 초기 단계에서는 초기화 메시지의 교환을 통해 서버 측에서는 에이전트에게 게임 타입과 게임의 승리조건 등을 알려주며, 에이전트는 자신이 제어하는 캐릭터의 이름, 소속 팀 등을 서버 측에게 알려준다. 일단 연결이 되고 나면, 서버는 동기 메시지 모드와 비 동기 메시지 모드를 옮겨 다니며 캐릭터들의 지각정보(sensory information)를 해당 에이전트들에게 보내준다.

3. 에이전트 구조

본 연구에서는 Gambots 시스템 환경에서 자율적으로 동작하는 지능형 캐릭터 에이전트의 제어를 위해 범용 에이전트 구조의 하나인 CAA(Context-sensitive Agent Architecture)를 채용하였다. 본 연구를 위해 개발된 CAA는 에이전트의 모든 부분을 Java 언어로 구현할 수 있는 행위기반의 에이전트 구조이며, 복잡한 행위들을 효과적으로 표현하고 수행할 수 있는 기능을 제공한다. CAA는 크게 행위 라이브러리(behavior library), 월드 모델(world model), 내부 모델(internal model), 센서(sensor)와 실행기(effector), 그리고 인터프리터(interpreter) 등으로 구성된다. CAA의 행위 라이브러리(behavior library)는 실행 가능한 모든 행위 객체들을 저장, 관리한다. 각 행위는 행위 유형을 가진다. 행위의 유형들로는 내부의 의사결정행위인 내부 행위(internal behavior)와 실제실행기를 통해 환경을 변화시키는 외부 행위(external behavior), 그리고 다른 에이전트와의 통신을 위한 대화 행위(conversation)가 있다. CAA의 월드 모델은 외부 환경의 상태를, 반면에 내부 모델은 에이전트의 내부 상태 즉, 행위모드를 표현한다. CAA의 인터프리터(interpreter)는 에이전트의 행위에 관한 총체적인 제어를 담당한다. 본 연구에서는 이와 같은 동일한 기능을 가진 다수의 UTBot들로 하나의 가상 에이전트 팀을 구성하여 UT Domination과 같은 팀 별 게임을 수행토록 하였다.

4. 동적 역할 배정

정적인 역할 배정 방식은 시스템 설계자가 설계단계에서 각 구성원 에이전트가 맡을 역할을 미리 배정해 놓으면 실행단계에서는 변화 없이 그대로는 진행되는 방식이었다. 이러한 정적인 역할 분담의 방법의 경우 실행 중에 에이전트들 사이에 상호 협의를 할 필요가 없기 때문에 실시간 환경에서 매우 빠르게 역할이 결정된다는 장점이 있다. 하지만 최근의 인터랙티브 컴퓨터 게임에서는 환경의 변화가 매우 빠르며 변화의 결과 또한 비결정적으로 나타난다. 그러므로 설계자가 모든 변화를 예측하기 어려우며 기존의 정적인 역할 배정 방법으로는 에이전트들이 제 역할을 수행하기가 어려운 점이 있다. 순수한 동적 역할 배정 방식의 경우 비결정적인 상황에서 에이전트들의 역할이 결정되어 있지 않기 때문에 비결정적 상황에 에이전트들이 협의의 통해서 역할 분담을 할 수 있다. 하지만 에이전트들 간에 협의를 위한 많은 통신이 필요하기 때문에 지연시간을 늦추는 요인이 되어 실시간 환경에 적합하지 않다. 따라서 본 연구에서는 기존의 정적 역할 배정의 장점인 실시간 적용성과 동적 역할 배정의 유연성을 결합한 새로운 동적 역할 배정 방식을 제안한다. 특히 Unreal Tournament Game에서 지원하는 Domination 게임을 통해서 새로운 역할배정 방식을 적용시켜 보았다. 이 게임 방식은 2인 이상의 2~3개의 팀이 3곳의 점령 지역(Domination Point)을 일정한 시간 동안 점령해야 하며 점령된 지역은 점령 팀에게 3초 간격으로 점수를 1점씩 상승시켜 주게 되어 있다. 최종적으로 팀들 중에 목표점수에 가장 먼저 도달하게 되는 팀이 승리하는 게임이다. UT Domination 게임은 각 팀원들 간의 게임의 승리를 위한 역할이 나누어져 있기 때문에 새로운 방식을 적용하기에 적합하다. 본 연구에서는 초기단계에서 게임을 승리하기 위한 에이전트들의 역할 집단을 미리 설계하고, 실행단계에서 역할집단 내의 역할을 동적으로 에이전트에게 분할하는 실시간 동적 역할 배정 방식을 제안한다. 이 방식은 기존 역할 배정을 위한 협상을 최소화 시켜준다. 따라서 실시간 게임환경에 매우 적합하다.

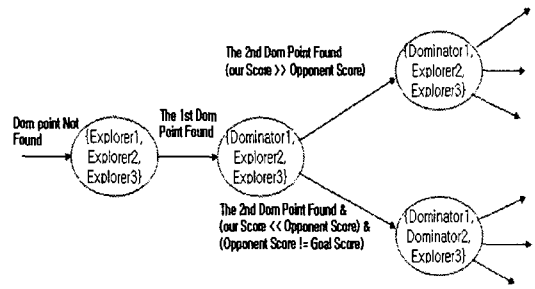
다음은 본 연구에서 제안하는 역할 배정 방식을 정리한 것이다.

- * $A = \{a_1, a_2, \dots, a_n\}$: n명의 에이전트들로 구성된 팀
- * $R = \{r_1, r_2, \dots, r_k\}$: k개의 서로 다른 역할들
- * $t_i: R_{i-1} \times S_i \rightarrow R_i$: 역할집합 변이함수(role-set transition function)
(R_{i-1} : 이전의 역할 집합, S_i : 환경상태, R_i : 새로운 역할 집합)
- * $m_i: R_i \rightarrow A$: 역할 배정 함수(role assignment function)
- * $F_i = \langle A, R_i, m_i \rangle$: 팀 포메이션(team formation)
(A : 에이전트 집합, R_i : 역할 집합, m_i : 역할 배정 함수)

여기서, 역할집합 변이함수 t 는 시스템 설계자에 의해 미리 정의되고, 역할 배정 함수 m_i 는 에이전트간의 실시간 협상에 의해 결정된다.

본 연구의 적용 대상이 되는 UT Domination 게임에서는 두 가지 역할이 필요하다. 하나는 현재 게임이 진행되고 있는 미지의 월드맵에서 월드모델의 구성과 점령목표지역을 발견하기 위한 Explorer 역할과 발견한 점령목표지역을 공격과 수비하기 위한 Dominator의 역할이다. 게임에 승리하기 위해서는 두 가지의 역할이 알맞게 같은 팀 내에 에이전트들에게 배분되어야만 효과적으로 게임에서 승리할 수 있다.

게임상황에 따른 역할 집합의 변이는 [그림 1] 와 같은 역할 집합 변이도(Role-set Transition Diagram)로 표현할 수 있다



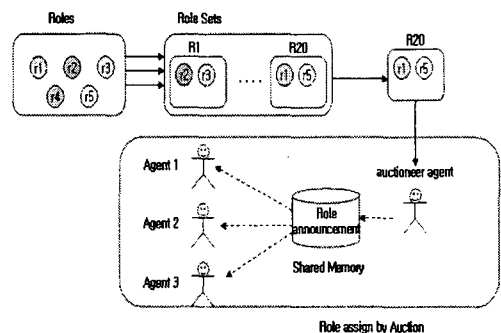
[그림 1] 역할 집합 변이도

역할 집합 변이도에서 각 노드는 필요한 역할들의 집합을 나타내고, 각 링크는 역할 집합의 변화를 나타낸다. [그림 1]의 역할 집합 변이도는 Team Domination 게임을 위한 전형적인 예를 나타내고 있다.

이 역할 집합 변이도는 게임 초반 Domination point들을 하나도 발견하지 못한 때에는 오직 Explorer 역할들만 요구하나, 점령 목표지들을 발견함에 따라 그곳을 점령하고 지키는 Dominator 역할들을 요구하는 역할집합 변이를 보여주고 있다.

이와 같이 역할 집합은 역할 집합 변이도에 따라 자동으로 결정되나, 실질적인 역할배정은 에이전트간에 협상을 통한 방식으로 결정된다.

환경이 실시간으로 급변하는 게임환경에서 지속적인 통신을 이용한 긴 협상 방법은 적합하지 않다. 본 연구에서는 실시간 동적 역할 배정을 위해서 에이전트들에게 실시간으로 경매(auction)형식을 이용한 역할 배정 방법을 적용하였다. 본 연구에서 사용할 경매방식은 같은 팀 에이전트들 중에서 하나의 에이전트에게 경매인 역할을 가지도록 설계하여 각각의 팀 에이전트들 중에서 가장 높은 입찰조건을 제시하는 에이전트에게 역할을 배정하는 것을 기본으로 한다. 경매인 에이전트는 에이전트간에 공유할 수 있는 메모리를 이용해 경매물품에 해당하는 배정 역할을 공고하고 다른 에이전트들은 자신의 입찰가를 공유 메모리에 입력한다. 경매인 에이전트는 이를 수집하여 가장 높은 입찰조건을 제시한 에이전트에게 역할을 낙찰하는 방식이다. [그림 2]은 경매인 에이전트를 통해서 미리 구성된 역할 집합 중 하나의 역할이 배정되는 과정을 나타낸다.



[그림 2] 역할 배정 과정

본 연구에서 에이전트가 게임을 진행하는 중에 경매인에게 제시하는 경매 입찰가에 대한 조건은 Explorer와 Dominator 들만을 고려하였다 Dominator 역할에 대한 각 에이전트의 비용함수로는 정해진 점령 목표지(domination point)까지의 도달거리를 사용하였고, Explorer 역할에 대한 비용함수로는 현재까지 작성된 월드 맵 경계선(frontier)까지의 도달거리를 사용하였다. 한편, 수입함수 R은 다음과 같이 정의할 수 있다.

$R: R_i \rightarrow R+$. 이때 역시 R_i 는 역할 집합을, $R+$ 는 양의 실수집합을 각각 나타낸다. 본 연구에서는 Dominator 역할에 대한 각 에이전트의 수입함수로는 목표지 점령을 통해 얻게 될 취득 예상 점수(score)를 사용하였고, Explorer 역할에 대한 수입함수로는 새로 발견할 노드와 에지의 수를 사용하였다.

역할 r에 대한 한 에이전트의 예상 이익은 그 역할을 수행함으로써 얻게 될 수입(revenue)과 소요될 비용(cost)의 차(difference)로 계산할 수 있다. 이렇게 역할 r에 대한 예상 이익이 계산되면, 각 에이전트는 자신의 예상 이익만큼을 입찰가로 제시한다. 따라서 결과적으로 해당 역할을 수행함으로써 가장 큰 이익을 얻을 수 있는 에이전트가 그 역할을 위한 낙찰자로 선정된다.

5. 실험

앞서 제시한 역할 배정방식의 효과를 확인해보기 위해서 동적인 역할배정 팀과 임의의 역할 배정 팀을 구성하여 UT Domination 게임을 수행해보았다. 각 팀원들은 동질의 에이전트들로 구성되어 있으며, 각 구성원 에이전트는 본 연구를 위해서 개발된 CAA UTBOT을 이용하였다. 동적 역할 배정 팀은 구성원 들간에 공유메모리를 사용하여 역할 배정을 동적인 상황에 맞춰 Dominator와 Explore로 나누어 역할배정을 하도록 구현하였으며, 상대팀은 구성원간에 역할 배정을 위한 협의가 전혀 이루어지지 않으며, 따라서 각 구성원 에이전트는 임의로 자신의 역할을 결정하고 수행하는 팀을 구현하였다.

실험을 위해서 노드수가 30개, 60개, 120개인 맵을 제작하여 실험을 하였고, 최고점수를 100점과 200점으로 설정하여 실험하였다.

<표1>과 <표 2>를 살펴보면 왼쪽 점수가 동적 역할 배정팀 오른쪽이 무작위 독립행위 팀이며, 동적 역할 배정팀의 점수가 압도적으로 높음을 알 수 있다. 또한 주목할 사항은 최고점이 100점일 때 <표1>에 나타난 실험결과보다 200점일 때의 <표2>에 나타난 실험결과가 동적 역할 배정 팀의 우수성이 더 강조 된다는 점이다. 이것은 게임의 지속시간이 길어 질수록 동적인 역할배정이 더욱 더 효과를 발휘한다는 결론을 얻을 수 있다. 또 하나의 사실은 실험 도중 동적 역할 배정 팀이 Domination Point를 빨리 찾을수록 그 효율성이 더욱 더 뚜렷하게 나타나며 탐색이 늦어 질수록 임의의 역할 배정팀과 차이가 줄어들어 든다는 것이다. 따라서 효율적인 탐색결과가 동적 역할 배정의 효율성과 정비례하게 작용한다는 것을 알 수 있다.

<표 1> 최고 점이 100점 일 때 양팀 실험결과

시도 \ 맵	Map1 (Node30)	Map2 (Node 90)	Map3 (Node 120)
1회	100 : 58	100 : 24	100 : 18
2회	100 : 46	100 : 34	100 : 22
3회	100 : 48	100 : 30	100 : 12
4회	100 : 65	100 : 38	100 : 30
5회	100 : 57	100 : 35	100 : 28

<표 2> 최고점이 200점 일 때 양팀 실험결과

시도 \ 맵	Map1 (Node30)	Map2 (Node 90)	Map3 (Node 120)
1회	200 : 109	200 : 55	200 : 23
2회	200 : 113	200 : 47	200 : 29
3회	200 : 99	200 : 51	200 : 41
4회	200 : 130	200 : 62	200 : 46
5회	200 : 101	200 : 58	200 : 62

6. 결론

본 논문에서는 멀티 에이전트 게임환경에서 지능형 에이전트들의 동적인 역할 배정 방법을 제안하였다. 본 논문에서 제안한 동적인 역할 배정 방법은 기존의 정적인 역할 배정 방식과 순수한 동적 역할 배정 방식과 달리 미리 설계 한 역할을 실시간으로 변화하는 비결정적 환경에 동적으로 경매 방식을 도입하여 역할을 분배 함으로 정적인 역할 배정 방식과 순수한 동적 역할 배정 방식의 장점을 모두 살린 새로운 역할 배정방식이다. 그리고 새로운 배정방식을 이용한 실험을 통하여 새로운 역할 배정 방법의 효율성을 증명하였다. 그러므로 새로운 역할 배정방식은 컴퓨터 게임환경과 같은 복잡하고 빠르게 변화하는 환경에 적합한 것으로 결론 지을 수 있다.

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] Ranjit Nair, Milind Tambe, Stacy Marsella, "Role Allocation and Reallocation in Multiagent Teams: Towards A Practical Analysis," 2003.
- [4] Sarit Kraus, "Negotiation and Cooperation in Multi-Agent Environments," Artificial Intelligence journal, Special Issue On Economic Principles of Multi-Agent System, Vol. 94, No.1-2, pp.79-98, 1997.
- [5] William van der Sterren, "Squad Tactics : Team AI and Emergent Maneuvers," In AI Game Programming Wisdom, Charles River Media Inc, 2002.