

실시간 동적인 환경에서 다중 에이전트의 협동 기법

유한하⁰ 조경은* 엄기현**

동국대학교 대학원 컴퓨터공학과⁰, 동국대학교 영상미디어학부 게임멀티미디어공학과
o12hana@hotmail.com⁰, {cke*, khum**}@dongguk.edu

A Cooperation Strategy of Multi-agents in Real-Time Dynamic Environments

Han-ha Yoo⁰, Kyung-eun Cho*, Ky-hyun Um**

Dept. of Computer Engineering, Graduate School of Dongguk University⁰,

Dept. of Game & Multimedia Engineering, Dongguk University*

요 약

오늘날 스포츠, RTS, RPG 게임과 같이 멀티플레이어가 한 팀을 이루는 집단형 방식의 게임은 팀 인공지능 기술이 더욱 필요하다. 기존의 독립적인 지능형 에이전트는 스스로 문제를 해결하는 자율성 향상 연구에 치중하였으나, 이는 다른 에이전트간의 협동 및 상호작용 능력이 부족하다. 이를 위해 본 논문은 다중에이전트 시스템에서 효과적인 역할 분담과 자율성을 갖는 레벨통합 접근방식을 소개한다. 복잡한 목표를 성취하기 위해 에이전트의 역할 정보를 이용하여 각자의 목표를 할당하고 각 에이전트는 맡은 역할을 동적인 환경에서 스스로 판단하고 행동한다. 팀 전체의 목표는 상호 보완된 역할 분담의 전략적인 측면에서 조정한다. 각 에이전트는 데이터보드를 이용하여 서로의 상태 정보를 공유하여 상호 협동을 유도한다. 역할이 할당된 각 에이전트는 스스로 계획기능을 갖고 있어 동적인 환경에서 적합한 행동을 취한다. 이 협동과 상호작용 과정에서 충돌의 문제점이 발생하는데 이를 제어하는 조정 에이전트의 역할을 전략적 측면에서 접근한다. 본 논문에서 제시하는 레벨통합 접근방식이 기존의 중앙 집권적 접근방식, 분권적 접근방식과 비교 실험하여 기존 방식보다 성능이 향상됨을 보인다.

ABSTRACT

Games such as sports, RTS, RPG, which teams of players play, require advanced artificial intelligence technology for team management. The existing artificial intelligence enables an intelligent agent to have the autonomy solving problem by itself, but to lack interaction and cooperation between agents. This paper presents "Level Unified Approach Method" with effective role allocation and autonomy in multi-agent system. This method allots sub-goals to agents using role information to accomplish a global goal. Each agent makes a decision and takes actions by itself in dynamic environments. Global goal of Team coordinates to allocated role in tactics approach. Each agent leads interactive cooperation by sharing state information with another using Databoard. As each agent has planning capacity, an agent takes appropriate actions for playing allocated roles in dynamic environments. This cooperation and interactive operation between agents causes a collision problem, so it approaches at tactics side for controlling this problem. Our experimental result shows that "Level Unified Approach Method" has better performance than existing central approach method or de-centralized approach method

Keyword : AI engine, Multi-Agent System, Planner, HTN, DataBoard

1. 서론

최근 가상현실과 게임 환경에서는 멀티플레이어들이 팀을 이루어 공통의 목표를 달성하는 집단형 방식이 주를 이루고 있다. 이는 팀 전체 목표(Global Goal)를 성취하기 위한 팀(Team) 인공지능 기술의 필요성이 부각됨을 의미한다. 기존의 에이전트 기반 인공지능은 에이전트의 추론 및 학습 기법을 통해 자율성을 높이는 지능형 에이전트 기술로 발전하였다. 그러나 독립적인 지능형 에이전트는 다른 에이전트와 협동 및 상호작용에 대한 문제점을 안고 있다. 팀 인공지능 기반의 다중 에이전트 시스템은 자율성뿐만 아니라, 전술적 의미를 갖는 접근방식이 필요하다. 또한 다중 에이전트 시스템은 동적인 실시간 환경으로 사용범위가 넓어지고 있다. 다중 에이전트 시스템은 한 에이전트로는 성취하기 어려운 복잡한 목표를 해결하며, 독립적인 이형질의 에이전트사이 협동과 상호작용을 위한 팀의 전략적인 제어를 다루어야 한다. 이러한 문제를 대안으로 개별 에이전트 간 협력과 상호작용 방법의 연구가 활발히 진행되고 있으며, 다중 에이전트 시스템의 구조 설계 측면에서 성능 향상을 기대하는 방법이 활발히 연구되고 있다 [1]. 특히 RPG 게임이나 스포츠 게임과 같이 팀 단위의 인공지능이 요구되는 환경에서 하나의 에이전트로 달성하기 어려운 복잡한 목표를 처리하는 구조 설계가 연구되고 있다.

기존의 인공지능이 추구하는 하나의 독립적 에이전트는 지식 표현과 추론 등을 통해 문제를 해결하는 역할을 하지만, 제한된 환경에서 하나의 에이전트로 모든 기능을 다 갖추 수 없으며, 자신이 해결하지 못하는 부분은 그것을 해결할 수 있는 다른 에이전트와 협동해야 한다. 이러한 협동과 상호관계의 향상을 위해 팀 전략측면에서 에이전트 간 협동 계획의 필요성이 증대된다.

본 논문에서는 다중 에이전트 시스템의 환경과 동적인 실시간 게임 환경에서 고려사항과 필요한 요소를 분석하고 그에 따른 문제점과 해결 방안을 제시한다. 다중 에이전트의 협력 관계를 위한 팀 인공지능의 전술적 측면과 독립적인 지능형 에이전트를 위한 자율성 측면의 두 상반된 관계를 조율하고, 이 관계를 명확히 한다. 본 논문은 다중 에이전트 시스템에서 두 관계의 효과적인 구조 제안을 위해 “레벨 통합 접근방식”을 설정한다.

다중 에이전트 시스템의 고려사항은 다음과 같다. 첫째,

전술적 의미에서 하나의 에이전트로 성취될 수 없는 복잡한 목표를 다루기 위해 에이전트 사이의 상호 협력 불가피하다. 하나의 독립적인 에이전트는 추론 또는 학습을 통해 스스로 문제를 해결하지만, 하나의 에이전트로 모든 기능을 갖추 수 없는 불안정한 존재이다. 자신이 해결할 수 없는 문제는 다른 에이전트를 통해 대신 해결하도록 상호협력이 필요하다. 둘째, 독립적인 지능형 에이전트는 행동의 자율성과 전술적 구조의 상반된 관계를 갖는다. 에이전트의 자율성이 높으면 전술적 측면에서 효율성이 떨어지고, 전술적 요소를 강조하면 에이전트의 자율성이 떨어진다. 이를 위해 에이전트간의 요구사항 정보를 공유하는 통신방법이 필요하며, 각 에이전트의 기능과 역할을 관리하고 제어할 수 있는 조정 에이전트가 존재한다.

본 논문의 구성은 다음과 같다. 2절에서는 다중 에이전트 시스템 환경에서 팀 전술 접근방식과 관련된 기존 연구를 기술한다. 3절에서는 본 논문에서 제시하는 다중 에이전트 시스템 구조로, 자율성과 전술적 의미를 효과적으로 증대시킬 수 있는 “레벨 통합 접근방식”을 설명한다. 4절에서는 레벨 통합 접근방식 시스템을 설계 및 구현된 실험 결과를 제시하고, 5절에서는 결론을 맺는다.

2. 관련연구

이 절에서는 다중 에이전트 시스템 환경에서 중앙 집권적 접근방식과 분권적 접근방식 등의 팀 전술 접근방식과 관련된 기존 연구를 서술한다.

2.1 중앙 집권적 접근방식

팀 전체의 목표는 상위 에이전트의 능력, 지능적인 개별 에이전트의 구성, 각 에이전트간의 상호 관계가 중요하다. 시스템의 특성 및 환경에 따라 상위 에이전트와 하위 에이전트의 기능이 다르지만, 동적인 실시간 환경에서 다중 에이전트 시스템의 경우는 이 두 요소의 역할의 절충이 무엇보다 중요하다[15]. 특히, 팀 목표가 전체 목표에 많은 영향을 미치는 시스템의 경우에는 독립적인 지능형 에이전트의 능력보다 상위 에이전트의 능력이 전체 목표에 더욱 영향을 미친다. 상위 에이전트의 역할은 상황을 빠르게 분석하고 집단 후퇴와 집단 공격 또는 최적의 팀원 구성으로 공격

하는 등 여러 가지 전략 조건을 결정해야 한다.

중앙 집권적 접근방식은 상위 에이전트의 미리 계획된 전술을 개별 에이전트로 강압적으로 통보하는 방식이다. 상위 에이전트는 코치와 같은 역할을 하여 하위 에이전트로 과제를 부여한다. 이 과정을 거쳐 과제를 받은 하위 에이전트는 구체적인 문제 해결방안을 스스로 결정하고 실행한다. 명령 스타일은 옵션을 통해 보완될 수 있으나 중앙 집권적 접근방식을 기반으로 이루어진다.

중앙 집권적 접근방식은 전략을 계획하고 결정하는 상위 에이전트의 역할을 통해 전술 및 전략 성취도가 높은 특성을 보이지만, 개별 에이전트의 자율성은 팀 전체 목표에 영향력이 낮은 잠재적 특성을 지닌다.

2.2 분권적 접근방식

분권적 접근방식은 독립적인 지능형 에이전트로 구성된 방식으로서 각 에이전트 사이 협동과 상호 작용으로 팀 전체 목표를 달성하는 방식을 말한다. 상위 에이전트의 존재가 없으므로 전체 팀의 중재자 역할을 수행할 수 없다. 그러므로 스스로 문제 해결하고 최상의 계획을 수립하는 구조이고, 필요에 따라 팀 구성원 간의 요청과 의도의 교환으로 상호 문제 해결을 도모하는 방식이다. 정보는 개별 에이전트들에게 균등하게 분배되며 개별 에이전트는 각 역할과 과제를 스스로 해결한다. 개별 에이전트 간에는 자신의 의도, 자신의 관측 등을 메시지로 교환하는 방식을 취한다.

이는 단독 AI FPS 등 게임에 응용되고 있다[16]. 개별 에이전트의 수집 정보는 사선, 소유 자원, 목표 행위, 목표 위치, 현재 위치, 행동, 능력 등 개인정보이다. 분권적 접근방식은 독립적인 지능형 에이전트가 스스로 의사 결정하는 구조로서 개인행동에 대한 최상의 계획 및 실행을 통해 자율성이 높다. 그러나 에이전트 사이 최상의 경로 선택으로 다른 에이전트와 충돌이 야기된다. 예를 들어, 하나 이상의 에이전트가 동시에 같은 자원을 점유 하려는 계획을 수립 할 경우 행동 선택에 장애를 일으키는 문제점을 안고 있다. 이런 잠재적인 특성으로 인해 팀 전체의 전략적 효율성이 매우 낮음을 알 수 있다.

3. 레벨 통합 접근방식

다중 에이전트 시스템은 하나의 에이전트로 달성하기 어

려운 목표를 갖고며, 동적인 환경에서는 자연스러운 계획수립이 이루어져야 한다. 이러한 특성을 갖는 다중 에이전트 환경에서 팀 전체의 목표를 성취하기위해 에이전트간의 협력 및 상호작용을 이루어 접근하는 팀 전술 접근방식을 취한다. 또한 실시간 동적인 환경에서 계획수립 및 재수립이 용이한 구조를 제안한다.

본 논문은 실시간 동적인 환경에서 다중 에이전트 시스템의 특성에 맞는 필요한 항목을 3가지로 요약할 수 있다. 첫째, 하나 이상의 에이전트 계획수립에 있어서 메시지 통신 방법은 필요한 정보공유와 보완문제에 좋은 해결 방법이나 통신양이 많은 다중 에이전트 시스템에는 부적합하므로 동일한 효과와 시스템 성능에 영향을 미치지 않는 정보공유 방식을 모색한다. 둘째, 기존 계획 기법의 경우 계획을 실행하는 도중 환경이 변화하면 수립된 계획을 중단하고 다시 새로운 계획을 수립하는 방식을 취한다. 이는 에이전트가 계획을 수립하고 실행하는데 걸리는 시간과 비용으로 시스템 성능에 영향을 미친다. 이러한 문제점을 해결하기 위한 계획기법을 조사하고 본 논문에서 제시하는 시스템에 적용한다. 셋째, 다중에이전트 시스템에서 팀 전술 접근방식을 제안한다. 즉, 팀 전체의 목표를 달성하기 위해서는 상위 에이전트의 능력과 지능적인 개별 에이전트의 구성이 주요한 요인이 된다. 본 논문에서는 상위 에이전트의 역할과 기능을 정의하고 하위 에이전트간의 관계를 정의한다.

본 논문은 위 3가지 필요 항목에 초점을 맞추어 다중에이전트 시스템 구조를 설명한다. 조정 에이전트의 역할을 수행하는 상위 에이전트와 동적 환경에서 목표 지향적인 행위 계획을 수립하는 독립적인 지능형 에이전트로 구성된다. 또한 상위 에이전트와 하위 에이전트의 관계, 하위 에이전트간의 상호 작용 관계를 설명하기 위해 두 레벨로 구별하며 이를 통합함으로써 “레벨 통합 접근방식” 설명할 수 있다.

3.1 상하 레벨의 명령 위임 구조

상위 에이전트의 역할은 환경 상태를 감시하고 적합한 전술을 판단, 결정하여 팀 전체가 목표를 달성하기 위해 효율적이고 최상의 적합성을 갖는 전략을 판단하는 것이다. 상위 에이전트는 팀 전술을 이루고자 하는 전략적 배치, 대형(포메이션)의 형태 지정, 하위 에이전트의 역할 분담 등을 결정한다. 또한 하위 에이전트의 상태를 감시하고 동적인

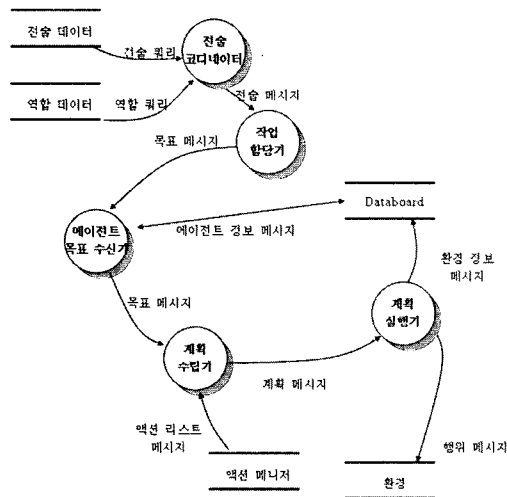
환경을 고려하여 최상의 전술을 검색하여 팀 전체의 목표를 성취한다.

3.2 등급 레벨의 정보 공유 구조

독립적인 지능형 에이전트의 행동 계획일지라도 에이전트 간 상호작용이 이루어지지 않으면 충돌이 야기될 수 있다. 예를 들어, 가상공간에서 한 에이전트 이동경로가 다른 에이전트에 의해 진로를 방해받거나 충돌이 발생하면 이동이 지연될 수 있다. 또는 하나 이상의 에이전트가 동시에 하나의 자원을 소유하는 경우 계획수립의 장애가 발생한다. 이러한 상황을 미리 예방하기 위해서 에이전트 간 상호작용이 필요하다. 개별 수집 정보로서 다른 에이전트의 사선과 소유 자원, 자신의 목표 행위, 목표 위치, 현재 위치, 행동, 능력, 소유 자원 등을 모두 공유하여 충돌 상황이 발생하는 것을 방지할 수 있다.

3.3 레벨 통합 접근방식의 시스템 구조

[그림3]은 본 논문에서 제시하는 “레벨 통합 접근방식” 시스템의 자료흐름도이다. 상하 레벨 명령 위임 구조를 통해 상위 에이전트에서 결정된 사항은 하위 에이전트로 전달되어 실행된다. 개별 에이전트는 목표 메시지를 통해 이벤트 처리되어 계획을 수립하고 실행한다. 상위 에이전트는 지식(전술과 역할 데이터 정보) 기반의 전술 코디네이터를 통해 현재 상태에 맞는 적합한 전술과 전략을 결정한다. 결정



[그림3] 레벨 통합 접근방식의 시스템 자료흐름도

된 전술은 작업 할당기를 통해 목표 메시지를 하위 에이전트의 목표 수신기로 전달한다.

개별 에이전트는 다른 에이전트의 수집 정보를 기반으로 계통적 계획 수립기를 통해 계획을 수립한다. 이 기법은 동적인 환경에서 수립된 계획을 재사용함으로써 계획수립의 실행속도와 유연성을 보인다. 액션에는 각 에이전트가 지니는 액션의 리스트를 관리한다. 이 정보를 기반으로 결정된 계획은 계획 실행 모듈을 거쳐 환경에 반영되며 동시에 Databoard에 자신의 행동과 자원 정보, 속성 등을 기록하여 다른 에이전트의 행동 계획에 정보로 사용된다. 에이전트 간 메시지 통신 방법을 이용할 경우 통신량, 에이전트 수, 메시지 량에 따라 성능이 평가된다는 점을 감안하면, 칠관 구조를 이용한 Databoard 구조는 적은 자원으로 통신 방법과 같은 효과를 주는 다중 에이전트 시스템에 적합한 정보 공유방식이다.

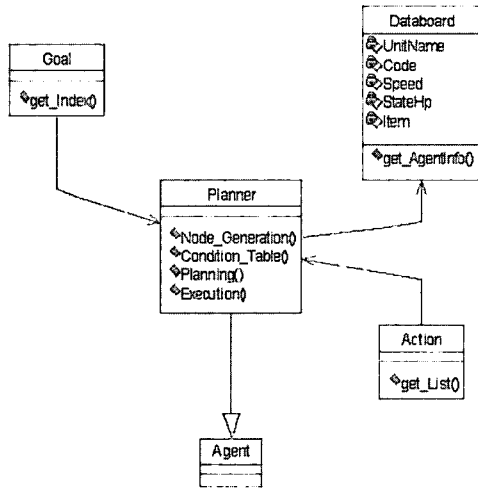
[그림4]는 계획 수립기를 구현하기 위한 클래스 다이어그램을 보여준다. 그림1에서 설명한 바와 같이 상위 에이전트로부터 받은 목표는 에이전트 목표 수신기로 전달된다. 이 목표는 Goal 클래스와 매치되며, 이는 에이전트의 입력 이벤트이다. 계획 수립기의 역할은 planner 클래스에서 담당한다. Planner클래스는 Agent의 하위 항목으로서 Agent 클래스에 포함된다. Planner에서는 다른 에이전트의 상태 정보와 점유 자원 정보를 공유하기 위해 Databoard 클래스와 상호 연관된다. Action 클래스는 자신의 행동 리스트를 보관하는 장소로서 각 캐릭터의 속성이 정의된 클래스이다. 이 지식정보를 바탕으로 Planner 클래스에서 계획을 수립한다. 계획 수립기의 동작 메소드로는 Node_Generation(), Condition_Table(), Planning(), Execution()으로 구성된다. 계통적 과제망(Hierarchical Transition Network, HTN) 계획 기법은 노드와 전제조건으로 구성된다. 이 동작을 하는 함수는 Node_Generation(), Condition_Table()이다. 계통적 과제망 계획수립 함수는 Planning()에서, 실행은 Execution()에서 담당한다.

3.4 계통적 과제망 계획 수립기

개별 에이전트는 목표 지향 행동 계획을 기반으로 하는 계통적 과제망 기법을 이용한다. 계획 수립 또는 실행 도중에 환경이 변화하는 경우 수립된 계획을 버리고 새로운 계획을 세우는 기존 기법과는 달리 수립된 계획을 유지하고

현재 변화된 환경을 고려하여 계획을 재사용하는 이점이 있어 동적인 환경에서 주로 사용되는 기법이다.

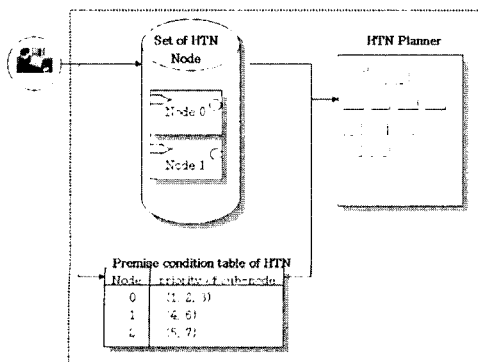
[그림5]은 개별 계획 수립 에이전트 플랫폼(Platform)을 설명한다. 계통적 과제망의 계획 수립기는 Node와 간선으로 구성된다. 이 시스템을 이용하게 될 개발 환경은 각 노드와



[그림4] 계획 수립기 클래스 다이어그램

간선의 정의에 따라 많은 차이를 보일 수 있다. 예를 들어, 게임에서의 장르에 따른 특성적 차이 또는 가상환경에서의 특별환경 상태, 로봇 시스템에서 예기치 못한 물리적인 환경 등 계획에 이용될 정보로서 노드와 간선의 정의는 고정적인 패턴이 될 수 없다. 이 같은 환경에서 계획수립에 필요한 노드와 간선관계를 설정하기 위해 사용자가 정의할 수 있도록 제공한다.

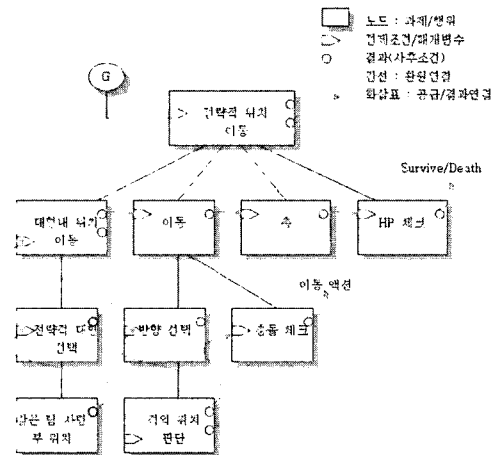
[그림6]은 이러한 설정을 토대로 한 계통적 과제망 계획



[그림5] 계통적 과제망의 계획 수립 에이전트 플랫폼

수립기의 계획 생성 과정을 보여주는 예이다. 노드와 간선의 정보를 통해 계통적 과제망 계획 수립기의 추상적인 다이어그램이 생성된다. 노드와 노드 사이의 관계를 전제조건 테이블을 통해 연결 관계를 구성한다. 노드는 과제와 행위로 구성되고 하위 노드의 전제 조건을 따라 가능한 액션의 시퀀스를 검색한다.

[그림7]은 계통적 과제망 계획 수립기에서 계획을 수립하는 워크 플로우를 보여 주는 것이다. 계획을 저장하는 PLAN 배열과 각 노드의 true/false 상태를 저장하는 FLAG



[그림6] 계통적 과제망의 계획 생성 과정

배열을 정의하고 노드 관리를 위해 일반 리스트 연결 방법으로 구현한 것이다.

계획수립은 초기 단계를 시작으로 목표 노드 0의 상태가 true가 될 때까지 전제조건을 검색한다. 단계1에서는 전제 조건 테이블을 검색하여 목표 노드 0의 하위 노드 1,2,3,4를 순서대로 삽입한다. 단계2에서는 하위 첫 번째 노드1의 전제 조건 검색하고, 검색 결과 전제조건이 존재하지 않으므로 true로 변환된다. 단계3은 노드 2의 하위 노드를 전제 조건 테이블을 통해 검색하여 노드 5,6,7을 순서대로 삽입한다. 추가된 각 노드 상태를 true로 만들기 위해 하위 노드를 검색하는 단계 4,5,6을 거친다. 같은 방법으로 노드3,4의 하위 노드를 검색하며, 모든 노드의 FLAG가 true가 되면 계획 수립을 종료한다.

[표1]은 계통적 과제망의 추상 데이터 타입을 표현한다. 초기 단계에서는 임시저장 장소인 P, F를 정의한다. F의 모든 노드상태가 true이면 계통적 과제망의 계획 수립을 종료

| | | | | | | | | | | |
|------|--|-----|-----|-----|-----|------|--|--|--|--|
| | | [0] | [1] | [2] | ... | [10] | | | | |
| PLAN | | | | | | | | | | |
| FLAG | | f | f | f | | | | | | |

| | | | | | | | | | | | | |
|------------|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|
| Step\array | | [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] | [10] |
| init | PLAN | 0 | | | | | | | | | | |
| | FLAG | F | | | | | | | | | | |
| 1 | PLAN | 1 | 2 | 3 | 4 | 0 | | | | | | |
| | FLAG | F | F | F | F | F | | | | | | |
| 2 | PLAN | 1 | 2 | 3 | 4 | 0 | | | | | | |
| | FLAG | T | F | F | F | F | | | | | | |
| 3 | PLAN | 1 | 5 | 6 | 7 | 2 | 3 | 4 | 0 | | | |
| | FLAG | T | F | F | F | F | F | F | F | | | |
| 4 | PLAN | 1 | 8 | 5 | 6 | 7 | 2 | 3 | 4 | 0 | | |
| | FLAG | T | T | T | F | F | F | F | F | F | | |
| 5 | PLAN | 1 | 8 | 5 | 9 | 6 | 7 | 2 | 3 | 4 | 0 | |
| | FLAG | T | T | T | T | T | F | F | F | F | F | |
| 6 | PLAN | 1 | 8 | 5 | 9 | 6 | 7 | 2 | 3 | 4 | 0 | |
| | FLAG | T | T | T | T | T | T | T | F | F | F | |
| 7 | PLAN | 1 | 8 | 5 | 9 | 6 | 7 | 2 | 3 | 4 | 0 | |
| | FLAG | T | T | T | T | T | T | T | T | F | F | |
| 8 | PLAN | 1 | 8 | 5 | 9 | 6 | 7 | 2 | 3 | 4 | 0 | |
| | FLAG | T | T | T | T | T | T | T | T | T | F | |
| exit | FLAG | T | T | T | T | T | T | T | T | T | T | |

[그림7] 계통적 과제망 계획 수립기의 워크플로우

한다. 그렇지 않을 때 F가 false인 조건에서 반복문을 수행하여 노드를 검색한다. 각 하위 노드의 상태를 판단하여 false이면 전제 조건 테이블을 검색하여 하위 노드를 추가하고 true 변환한다. 이 같은 반복을 통해 모든 하위 노드가 true이면 계획수립을 종료한다.

4. 구현 및 실험

```

HTNO
Char[10] L: //임시 계획 저장 공간
Char[10] F: //임시 상태 배열 저장 공간

IF(P == true) {exit;} //초기 결정 : 모든 상태 정보 true 이면 종료
Else {
  While (P == false) {
    if (P == true) { PLAN <- L: exit; } //모든 상태 정보가 true 일 때까지 반복
    else {
      if (P[sub_node_index] != false) { //각 sub_node에 대한 상태 결정
        x <- action(sub_node_index); //agent 탐색의 하위 노드 검색
        insertNode(L, 0, x); //임시 L 노드의 맨앞에 x를 삽입
        //sub_node가 true 인지 검사
        P[sub_node_index] = true;
      }
      sub_node_index ++; //next sub node
    }
  }
  if (P == true) { PLAN <- L: exit; } //모든 상태 정보가 true 일 때까지 반복
}
End HTNO
    
```

[표1] 계통적 과제망 알고리즘

본 논문은 다중 에이전트 시스템 특성에 맞는 구성원(에이전트)을 통해 계획수립과 팀간의 상호작용 문제를 해결하는 실험환경을 구축한다. 이 실험환경은 제안하는 계획수립기법과 기존 계획수립기법 간 대전 시물레이션을 통해 성능이 향상되는 지를 확인하는 실험이다.

제안하는 다중 에이전트 시스템 구조는 하나 이상의 에이전트로 구성되며, 하나의 에이전트로 성취될 수 없는 복잡

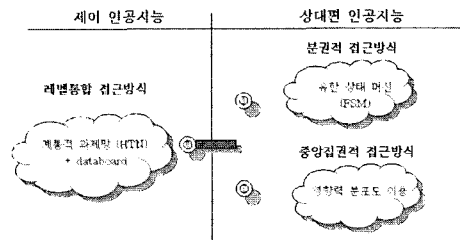
한 목표를 달성하기 위해 전략을 조정(Coordination)하는 상위 에이전트의 명령위임 구조를 따른다. 이 구조 설계를 위해 각 팀의 에이전트 수는 상위 에이전트를 포함하여 5유닛으로 구성되며, 상위 에이전트를 제거함으로써 승리하는 실시간 전략 시물레이션(Real Time Strategy Simulation, RTS)이다. 여기서 각 상위 에이전트는 배치 대형의 6가지 전술중 하나의 전술을 선택하여 하위 에이전트에게 명령을 내리고, 각 에이전트는 서로 다른 속성을 갖는 에이전트로 구성된다.

독립적인 지능형 에이전트는 할당된 목표를 수행하기 위해 각 액션 리스트 속성 정보를 기반으로 스스로 계획을 재수립하는 계통적 과제망 기법으로 구현한다. 특히, 다중 에이전트 간에 협력 및 협동을 통해 전략적 목표를 다른 에이전트 간 정보를 공유하기 위해 Databoard 구조로 구현한다. 이때, 에이전트 사이 충돌을 방지하기 위한 상호작용 문제 해결 대안에 주안점을 두어 실험한다.

4.1 실험 환경

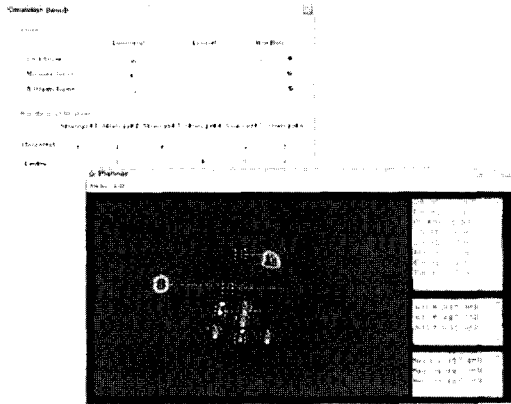
[그림8]은 본 논문의 제안 구조인 레벨 통합 접근방식의 성능 평가 및 실험 환경을 제공하기 위한 시스템 설계 구조도이다. 제안 구조인 계통적 과제망 계획 방식과 비교 대상인 상대 에이전트 계획방식으로 나뉜다. 비교대상은 사용자 정의에 의해 설정 가능하며, 선택할 수 있는 접근방식은 중앙 집권적 접근방식과 분권적 접근방식으로 구성한다. 팀 인공지능의 접근방식으로는 레벨 통합 접근방식으로 구현하고 팀간 대전 시물레이션을 통해 통계치를 분석한다. 중앙 집권적 접근방식은 영향력 분포도를 이용하고, 분권적 접근방식은 유한상태기계를 이용하여 구현한다.

[그림9]는 실험 환경 설계도를 JAVA를 이용해 자체적으로 구현한 게임의 스크린 샷 화면이다. 게임 장르는 RTS이고, 상대 사령부를 제거하면 승리를 거두는 시물레이션이다.



[그림8] 실험 환경 설계 구조도

테스트 베드 화면을 보면 시뮬레이션 화면, 메뉴 바, 세 개의 텍스트 창으로 구성된다. 우측 상단의 첫 번째 창은 팀 AI의 사령부 전략 정보를 보여 주고 두 번째 창은 팀 인공지능



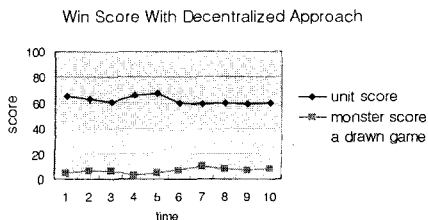
[그림9] 실험 환경 스크린 샷

능의 에이전트 상태 정보를 나타내고 세 번째 창은 상대방 에이전트의 상태 정보를 기록한다.

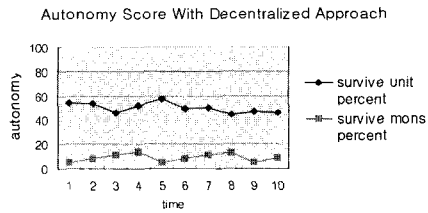
4.2 실험 내용

성능평가는 승리비율, 전술의 적합도, 팀 전략 성취도, 개별 에이전트의 자율적 행동 계획 성취도 4가지를 기준으로 삼는다. 첫 번째 승리비율은 전체 팀 전체 목표 달성 비율을 의미한다. 두 번째 전술의 적합성은 전술 선택의 능력에 있어 잘못된 판단으로 팀 전체 목표에 영향을 주므로 이를 판별하는 통계이다. 세 번째 팀 전략 성취도는 전략 선택과 적합성 여부를 판단한다. 네 번째 개별 에이전트의 자율적 행동 계획 성취도는 팀 전략에 맞게 주어진 목표를 개별 에이전트가 동적인 환경에서 스스로 계획을 수립하고 실행하여 자유도를 판별한다.

[표2]는 분권적 접근방식의 적 인공지능과 매 100회의 대전을 총 10회를 거쳐 진행한 결과를 보여준다. 제어 인공지능



[표2] 분권적 접근방식 대전 승리비율



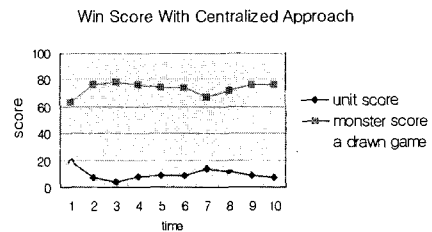
[표3] 분권적 접근방식 대전 자율성

능을 Unit Score로, 적 인공지능을 Monster Score로 표현했다. [표4]에서처럼 Unit Score가 60%이상 승리비율을 보이고 있다. [표3]은 대전후 살아남은 유닛의 수를 나타낸다. 이는 각 유닛의 자율성을 표현한다. [표3]에서 살아남은 제어 인공지능의 유닛은 전체 60%내외로 분권적 접근방식보다 높은 자율성을 보인다.

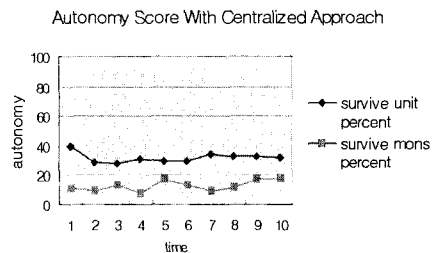
[표4]는 중앙집권적 접근 방식과 대전한 결과이며, Monster Score가 60%이상 승리 비율을 보이고 있다. 전략적 측면에서 효율적인 접근방식이다. 그러나 [표5]에서처럼 대전후 살아남은 유닛의 수는 제어 인공지능의 경우가 20%이상으로 자율성이 높음을 보인다.

4.3 실험 결과

분권적 접근방식과 대전 시뮬레이션을 실행한 결과, 분권적 접근방식은 전술적 의도가 매우 부족한 결과를 보인다. 이 방식은 개별 에이전트의 자율적 행동에는 효율적인 구



[표4] 중앙집권적 접근방식 대전 승리비율



[표5] 중앙집권적 접근방식 대전 자율성

조이나 전체목표를 성취하는데 효과적이지 못하다. 특히, 각 에이전트의 메시지 교환 방식을 통해 이루어 팀 전체 목표를 도달해야 하므로 다중 에이전트와 같이 통신량이 많은 경우 부적합하다. 중앙 집권적 접근방식과 대전 시뮬레이션 한 결과 중앙 집권적 접근방식은 전술을 지시하는 상위 에이전트의 명령에 의해 전술 성취가 높으나, 동적인 환경에서 예기치 못한 상황에서 개별 에이전트 스스로 판단하고 계획을 선택하는 자유도는 매우 낮다.

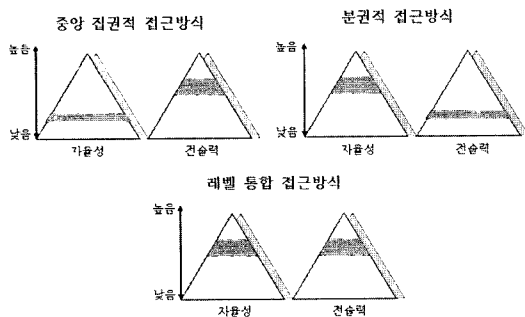
이 두 가지 접근방식과 대전 시뮬레이션 한 결과를 레벨 통합 접근방식의 경우 팀 전체 목표를 달성하는 승리비율은 높았으며, 전술 성취도가 [그림10]에서처럼 높았다. 또

| | 이 점 | 단 점 |
|------------|--|---|
| 분권적 접근방식 | 에이전트의 자율적 행동에 효율적 | 팀 전체 목표 달성에 비효과적임. 다중 에이전트 통신량에 따라 성능평가 |
| 중앙집권적 접근방식 | 전술적 가상환경에 적합 | 자율적인 행동 |
| 레벨통합 접근방식 | 동적인 환경에서 전술 적용력 좋음. 에이전트 개수와 상관없이 공개적 정보교환 | 전술과 자율성의 잠재적 조율 문제 발생 개발자 정의 필요 |

[표6] 팀 전술 접근방식의 특징 비교

한, 개별 에이전트는 실시간 동적인 환경에서 예상치 못한 상황에 계획을 재수립함으로써 스스로 행동을 계획하여 자율성이 높았다. 이런 관점에서 보면 중앙 집권적 접근방식과 분권적 접근방식의 서로 결점을 보완된 구조로써 다중 에이전트의 특징에 부합되는 구조임을 보인다.

성능 평가 기준으로는 승리비율, 전술의 적합성, 팀 전략 성취도, 개별 에이전트의 자율적 행동 계획 성취도 4 가지 통계치를 평가 기준으로 삼는다. 승리비율은 전술, 전략과



[그림10] 팀 전술 접근방식의 효과 비교

상관없이 결과만을 평가하기 위한 것이고, 전술의 적합성은 상위 에이전트의 전술 계획 능력을 평가하는 것이다. 팀 전략 성취도는 상위 에이전트에 의한 하위 에이전트로의 역할 분담 평가 기준이 되며 계획 성취도는 독립적인 에이전트의 자율성을 평가하는 기준이 된다.

5. 결론 및 향후과제

본 논문은 여러 에이전트로 구성된 다중에이전트 시스템에서 효과적인 역할 분담과 자율성을 갖는 레벨통합 접근 방식을 제안한 것이다. 레벨통합 접근방식은 여러 에이전트 사이에 상호 보완된 역할을 할당 조정하는 전략적 레벨과 스스로 환경에 대한 정보를 인식하여 주어진 문제를 해결하는 개별 에이전트 레벨로 구분된다.

가상 게임 환경과 같이 모든 상태를 미리 예측할 수 없는 동적인 실시간 환경에서 부분적인 계획 기능을 갖는 독립적인 지능형 에이전트를 설정한다. 팀 전체의 목표를 하나의 에이전트만으로 완전하게 갖출 수 없기 때문에 에이전트의 역할과 상태 정보를 통해 상호 보완되도록 전술적 접근을 유도한다.

에이전트의 상호 작용 문제와 통신동작은 데이터보드를 이용하여 서로의 상태 정보를 공유하여 상호 협동한다. 역할이 할당된 각 에이전트는 내부의 계통적 과제망을 바탕으로 계획을 수립하여 실시간 동적인 환경에서 행동 계획을 세우고 실행한다.

본 논문에서 제시하는 레벨통합 접근방식은 기존의 중앙 집권적 접근방식, 분권적 접근방식과 비교 실험에서 성능이 향상되었다. 중앙 집권적 접근방식은 전술적인 공동 목표 달성 비율이 좋으나 개별 문제해결 능력이 매우 낮다는 특징이 있다. 분권적 접근 방식은 자율성이 높은 에이전트로 구성되나 공동의 목표 달성 비율이 매우 낮다. 이에 반해 레벨통합 접근방식은 전략적 측면과 자율성의 상반된 관계를 조정하여 팀 목표와 자율성의 비율이 기존의 접근방식에 비해 상대적으로 향상되었음을 밝혔다.

본 연구에서 향후 과제는 실행 중 발생할 수 있는 외부 환경 변화에 신속히 대응하여 계획을 변경하거나 계획을 재수립할 수 있는 기능 체제 연구가 필요하다.

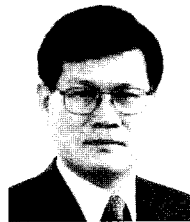
[참고 문헌]

- [1] "Computer aided multi-agent system engineering," Soe-Tsyr Yuan; Multi Agent Systems, 1998. Proceedings. International Conference on 3-7 Jul 1998.
- [2] "Ai interface standards committee", working group on goal-oriented action planning,
URL : <http://www.igda.org/ai/report-2004/report-2004.html>
- [3] "Blackboard architecture for intelligent control system," Linkens, D.A.; Abbod, M.F.; Browne, A.; Emerging Technologies and Factory Automation, 1999. Proceedings. ETFA '99. 1999 7th IEEE International Conference on Volume 2, 18-21 Oct, 1999 page(s):1185-1192 vol.2
- [4] "HTN planning : Complexity and expressivity." Erol, K., J. Hendler, D.S Nau, 1994 In Proceedings of the Twelfth National Conference on Artificial Intelligence, Menlo Park, CA: AAAI Press/MIT Press, pp. 1123-1128.
- [5] "Applying Goal-Oriented Action Planning to Games," Jeff Orkin; AI Game Programming Wisdom II, Charles River Media, 2004.
- [6] "Applications of SHOP and SHOP2," Nau, D.; Au, T.-C.; Ilghami, O.; Kuter, U.; Wu, D.; Yaman, F.; Munoz-Avila, H.; Murdock, J.W.; Intelligent Systems, IEEE [see also IEEE Intelligent Systems and Their Applications] Volume 20, Issue 2, March-April 2005.
- [7] "Localized Search for Controlling Automated Reasoning," Lansky, A.; Proceedings of the Ninth National Conference on Artificial Intelligence, 1991.
- [8] "Communication and Interaction in Multiagent Planning," Georgeff, M.P.; Proceedings of the Third National Conference on AI, pp. 125-129, 1983.
- [9] "Some problems of basic organization in problem-solving programs," A. Newell, Conf. on Self-Organizing Systems, G.T. Jacobi and G. D. Goldstein, Eds., Washington, D.C., 1962.
- [10] "A blackboard architecture for control," B. Hayes-Roth, Artificial Intelligence. Vol. 26, 1985.
- [11] "event-based blackboard architecture for multi-agent system," Dong, J.; Chen, S.; Jeng, J.-J.; Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on Volume 2, 4-6 April 2005 Page(s):379 - 384 Vol. 2 Digital Object Identifier 10.1109/ITCC.2005.149
- [12] "Blackboard systems formalized within a software architectural style," Stiger, P.R.; Gamble, R.F.; Systems, Man, and Cybernetics, 1997. 'Computational Cybernetics and Simulation', 1997 IEEE International Conference on Volume 2, 12-15 Oct, 1997 Page(s):1204 - 1209 vol.2
- [13] "A language and architecture for efficient blackboard systems," Hewett, M.; Hewett, R.; Artificial Intelligence for Applications, 1993. Proceedings., Ninth Conference on 1-5 March 1993 page(s):34-40
- [14] "Blackboard architecture and applications", Vinze, A.S.; Sen, A.; System Sciences, 1993, Proceeding of the Twenty-Sixth Hawaii International Conference on Volume iii, 5-8 Jan, 1993
- [15] "Suad Tactics : Planned Maneuvers," Sterren, William; AI Game Programming Wisdom, Charles River Media, 2002. Page(s):481-48
- [16] "Suad Tactics : Team AI and Emergent Maneuvers," Sterren, William; AI Game Programming Wisdom, Charles River Media, 2002.



조 경 은

1993.2 동국대학교, 전자계산학과(공학사)
 1995.2 동국대학교, 컴퓨터공학과 대학원(공학석사)
 2001.8 동국대학교, 컴퓨터공학과 대학원(공학박사)
 2003.9 ~ 2006.6 동국대학교, 정보산업대학 컴퓨터멀티미디어공학과
 전임강사
 2006.7~현재 동국대학교 영상미디어학부 게임멀티미디어공학과
 조교수
 관심분야: 컴퓨터 게임 알고리즘, 게임 인공지능, 컴퓨터 그래픽스 용
 용, 멀티미디어 정보처리



엄 기 현

1975.2 서울대학교 공과대학 응용수학과 공학사
 1977.2 한국과학기술원 전산학과 이학석사
 1994.2 서울대학교 대학원 컴퓨터공학과 공학박사
 1978.3 ~ 2006.6 동국대학교 컴퓨터멀티미디어공학과 정교수
 2006.7 ~ 현재 동국대학교 영상미디어학부 게임멀티미디어공학과
 정교수
 1995.3 ~ 1999.2 동국대학교 정보관리처장 역임
 2001.3 ~ 2003.2 동국대학교 정보산업대학 학장 역임
 1998.9 ~ 2000.8 한국정보과학회 데이터베이스연구회 운영위원장
 1999.4 ~ 2005.4 Int. Conf. on Database Systems for Advanced
 Applications Steering Committee 위원 역임
 2001.3 ~ 2002.2 한국정보과학회 논문지 편집부위원장
 (데이터베이스 담당)
 1998.12 ~ 현재 한국멀티미디어학회 부회장, 자문위원, 현재 수석
 부회장
 2004.1 ~ 현재 한국게임학회 부회장, 현재 자문위원
 관심분야: 게임 시스템 설계, 멀티미디어 시스템 설계, 멀티미디어
 정보처리, 멀티미디어 데이터베이스

논문투고일 - 2006년 5월 25일
 심사완료일 - 2006년 9월 13일



유 한 하

2000.2 삼척대학교 전자공학과 (공학사)
 1999.11 ~ 2004.2 청호정보통신 소프트웨어 팀장
 2004.3 ~ 2006.2 동국대학교 대학원 컴퓨터공학과 석사 졸업
 2006.3 ~ 현재 동국대학교 대학원 컴퓨터공학과 박사과정
 관심분야: 게임 엔진, 게임 인공지능, 모바일 게임, 입출력 시스템