ISSN 1598-0170 (Print)
ISSN 2287-1136 (Online)
http://www.jksii.or.kr

게임 환경에서의 지능형 다중 에이전트 개발☆

Development of Intelligent Multi-Agent in the Game Environment

김 동 민[1] 최 진 우[1] 우 종 우**
DongMin Kim JinWoo Choi ChongWoo Woo

요 약

다중 에이전트 시스템에 대한 연구는 최근 다양한 분야에서 활성화 되고 있으며, 복잡한 시스템의 제어 및 최적화에 관한 연구가 진행되어 왔다. 본 논문에서는 게임 환경에서의 NPC(Non-Player Character) 시뮬레이션을 위한 다중 에이전트 시스템을 개발한다. 시스템 개발의 목적은 동적 이산사건 영역의 상황을 추론하여 신속하고 정확한 판단을 제공하고 에이전트 시스템의 최적화 과정을 보다 손쉽게 도와주는데 있다. 이를 위한 에이전트 시스템의 기본 모델은 페트리넷을 활용하여 구조를 단순화 하고 퍼지 추론엔진을 사용하여 다양한 상황을 결정할 수 있도록 하였다. 본 연구 시스템의 실험은 NPC간의 가상 전장 상황을 묘사하며, 퍼지 규칙이 적용된 에이전트와 유한 상태 기계로 구현된 NPC를 시뮬레이션 하여 에이전트의 승률과 생존율을 산출하였다. 실험 결과 퍼지 규칙 기반 에이전트의 승률과 생존율이 유한 상태 기계로 구현된 NPC보다 더 높은 것으로 나타났다.

☞ 주제어 : 지능형 다중 에이전트, 퍼지추론, 페트리넷, 시뮬레이션, 게임 환경

ABSTRACT

Recently, research on the multi-agent system is developed actively in the various fields, especially on the control of complex system and optimization. In this study, we develop a multi-agent system for NPC simulation in game environment. The purpose of the development is to support quick and precise decision by inferencing the situation of the dynamic discrete domain, and to support an optimization process of the agent system. Our approach employed Petri-net as a basic agent model to simplify structure of the system, and used fuzzy inference engine to support decision making in various situation. Our experimentation describes situation of the virtual battlefield between the NPCs, which are divided two groups, such as fuzzy rule based agent and automata based agent. We calculate the percentage of winning and survival rate from the several simulations, and the result describes that the fuzzy rule based agent showed better performance than the automata based agent.

☞ keyword : Intelligent Multi-Agent, Fuzzy reasoning, Petri-Net, Simulation, Game Environment

1. 서 론

현재 게임분야에서의 인공지능 캐릭터들은 대부분 유한상태 기계의 이론을 중심으로 발전되어 왔다. 유한상태 기계의 이론에서는 게임에서의 인공지능 캐릭터인 NPC (Non-Player Character)의 행동을 유한개의 상태로 정의하고, 상태를 전환시키며, 각 상태에 정의된 행동을 하도록 정의한다. 이러한 정의는 해당 상태에 대한 행동만 묘사하면 되기 때문에 단순한 행동에 대해서는 구현이 쉽고 빠르지만, 복잡한 상황에 놓여진 NPC에 적절한 행동을 묘사하기 위해서는 NPC의 행동이 부 자연스럽기 때문에 무수히 많은 상태가 고려되어야 한다 [1]. 예를 들어, NPC 간 통신을 통해 전투 시 아군에게 도움을 요청한다고 가정 할 경우, 정의된 상태 전환 규칙에 따라 상태가 전환되고 그 상태에 정의된 행동만을 수행되기 때문에, 발생 가능한 추가적인 동적 상황에 대응하지 못하는 결과를 주게 된다. 이를 해결하기 위해 보다 많은 상세한 상태를 고려하여 설계할 경우에는, 시스템의 구현 시 복잡도가 매우 높아지고 유지 보수도 쉽지 않다. 본 연구에서는 이러한 문제점을 극복하기 위하여, 에이전트의 이론을 기반으로 페트리넷 모델과 퍼지규칙을 함께 적용할 수 있는 시스템을 개발하고자 한다.

다중 에이전트 시스템(Multi-agent)은 기본적으로 지능성, 자동성, 이동성 등의 특성을 가지며, 복잡한 시스템의 제어 및 최적화를 비롯한 다양한 분야에서 연구되어 왔

[1] School of Computer Science, Kookmin University, Seoul, 136-702, Korea.
* Corresponding author (cwwoo@kookmin.ac..kr)
[Received 27 August 2015, Reviewed 11 September 2015, Accepted 28 October 2015]
☆ 본 연구는 국민대학교 교내연구비의 지원으로 수행되었음

다[2][3]. 특히, 다중 에이전트 시스템은 복잡하고 대규모 문제를 동적으로 해결하기 위하여 다수의 에이전트들이 상호작용을 하게 된다. 또한, 다중에이전트 시스템은 비동기적, 분산적인 특성으로 이산사건 시스템(Discrete Event Dynamic System:DES)으로도 간주되며, 페트리넷을 포함한 다수의 분석 방법으로 연구되고 있다 [4]. 페트리넷은 동적 이산사건 시스템을 분석에 잘 정의된 수학적 구조를 가지고 있기 때문에 다수의 분산 및 병렬시스템의 모델링과 분석에 성공적으로 사용되었다 [5][6]. 반면 다중 에이전트 시스템은 동적 이산사건 시스템(DES)의 복잡성이 증가되고, 신뢰도와 지속성이 문제점을 보완해 줄 수 있는 페트리넷의 적용이 연구되고 있다 [7][8]. 기본적인 페트리넷은 시간적 지식표현이 부족하기 때문에 Timed 페트리넷[9]이 연구되었으며, 객체지향 개념이 적용된 객체지향 페트리넷(OOPN)을 비롯한 다양한 개념의 페트리넷이 연구되고 있다[10]. 그러나 페트리넷은 동적 시스템에서의 학습능력을 모델링하지 못하는 문제점이 있으며, 이를 보완하기 위하여 퍼지로직의 기능이 추가된 Fuzzy timed petri net(FTPN)[11] 이 연구되고 있으며, 퍼지추론의 기능을 확장하여 다중 에이전트 시스템에서 적용되는 연구가 진행되고 있다[12]. 예를 들면, Yifan은 탱크 전투 게임에 퍼지 로직을 도입하여 에이전트의 행동을 퍼지 규칙을 이용해 외부에서 제어할 수 있게 하였고[13], Nugroho et al은 퍼지 조정자를 이용해 게임 에이전트의 그룹 미션수행 과정을 보다 사실적으로 표현하였다[14].

본 연구에서는 주어진 상황을 인지하여 시뮬레이션을 하는 다중 에이전트 시스템을 개발하고자 한다. 시스템 개발의 목적은 동적 이산사건 영역의 상황을 인지하여 추론한 후, 신속하고 정확한 판단을 제공하는데 있다. 이를 위하여 시스템의 기본 모델은 페트리넷을 활용함으로서 모델의 구조적 단순화를 지향하고, 퍼지 추론엔진을 사용하여 다양한 상황판단이 가능하도록 설계하고자 한다. 본 연구의 실험을 위해서는 적군과 아군이 자율적으로 교전하는 전투 게임의 영역을 선정하였고, 지속적인 시뮬레이션을 통해 산출된 전투 상황의 데이터를 분석하여 시스템의 설계에 반영하였다. 실험은 적군과 아군으로 구분되는 NPC간의 가상 전장 상황을 실험하기 위하여 퍼지 규칙이 적용된 에이전트와 유한 상태 기계로 구현된 NPC를 시뮬레이션 하여 에이전트의 승률과 생존율을 산출하여 평가한다.

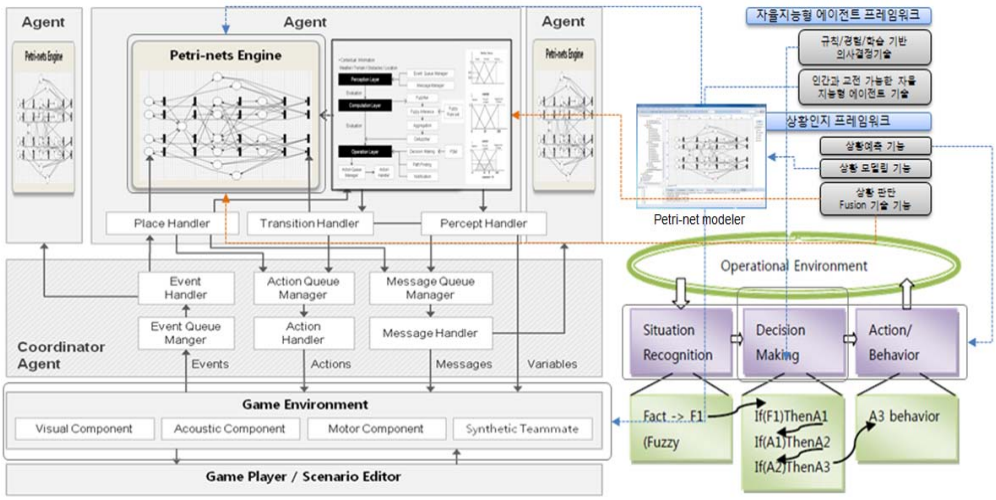
2. 관련 연구

2.1 지능형 에이전트

에이전트는 대규모의 복잡한 시스템에서 자동성(autonomous), 협업(cooperation), 통신(communication), 상호작용(interaction) 등의 특성을 가지는 소프트웨어 컴포넌트로 간주되고 있다 [15]. 다중 에이전트 시스템은 이러한 다수의 에이전트들로 구성된 구조를 가지며, 에이전트들 간의 상호작용을 통해 주어진 문제를 해결 하게 된다. 본 연구의 에이전트는 이러한 기본적인 에이전트 개념을 기반으로 페트리넷과 퍼지이론을 적용하는 에이전트로 설계 및 구현한다. 이러한 다중 에이전트는 최근 지속적으로 다수의 연구가 수행되고 있으며, 그중 시뮬레이션에 관한 대표적인 몇 가지 연구를 보면 다음과 같다. Sesam[16]은 에이전트 기반 시뮬레이션을 위해 모델링 환경을 제공한다. 시스템의 주요 특징은 상황변수들과 UML과 유사한 모델링을 위한 다이어그램을 제공하기 때문에 프로그래밍의 기술 없이 손쉽게 그래픽 기반으로 시뮬레이션 제작이 가능한 장점이 있다. SimWalk[17]는 보행자를 위한 시뮬레이션 시스템으로 사람들이 집중되는 공공의 장소에서 개인의 상황을 모델링하며, 주로 보행자에 관련된 상황을 시뮬레이션 할 수 있다. Brahms[18]는 Java 가상머신에서 구동되는 다중 에이전트 기반 도구로서 컴파일러, 그래픽 인터페이스, 뷰어 등 시뮬레이션을 용이하게 하는 사용자 환경을 제공한다. 본 연구에서는 이러한 다중에이전트에 관한 연구를 기반으로 에이전트 모델링에 관한 연구를 우선적으로 접근하고자 한다.

2.2 페트리넷

페트리넷은 병렬적, 분산적, 그리고 비동기적인 시스템을 연구하기 위한 가시적이며 수학적인 모델링 도구이다. 또한 페트리넷은 초기 상태를 가지는 방향그래프이며, 플레이스(Place) 와 트랜지션 (Transition)을 가지는 이분 가중 (weighted bipartite) 그래프 이다[19]. 이러한 특성은 문제영역에 따라 해석될 수 있는데, 분산 이산사건 시스템에서는 트랜지션이 사건(event)으로 간주되고, 플레이스는 사건이 발생될 수 있는 조건(condition)으로 간주된다. 시스템의 구조는 외부 환경에 따라 동적으로 변화될 수 있기 때문에, 이산사건 시스템으로 간주된다. 따라서 에이전트를 위한 일반적인 페트리넷에서 플레이스는 각 에이전트로 표시되며, 외부에서 주어지는 메시지에 따



(그림 1) 전체 시스템 구조
(Figure 1) The Structure of the System

라 에이전트의 상태는 트랜지션을 통하여 이동하게 된다. 페트리넷기반 모델링에 관한 대표적 연구는 다음과 같다. Murata et al[20]는 페트리넷을 로봇 연산에 사용하였는데, 페트리넷은 로봇의 동기적 상황을 계획하는데 사용되었고, 로봇 액션은 트랜지션의 활성화로 간주 하였다. Xu et al.[21]은 정적 계획기법에 사용하였는데, 에이전트의 액션을 트랜지션으로 모델링 하고 계획의 실행을 위한 알고리즘을 제시하였다. Ahn et al [22]은 수요와 공급의 분산 협업시스템을 위한 다중 에이전트 구조를 제시하였고, Reisig[23]는 인터넷상에서의 게임을 위한 가상 마켓을 위한 모델을 제시하였다. Gazdare [24]는 Colored Petri nets (CPN)을 운송시스템을 위한 모델정립에 적용하였고, 그 외 다수의 CPN모델을 위한 연구가 제시되고 있다. 본 연구에서는 이러한 연구에 기반하여, 게임환경에서 복잡한 다중 에이전트의 행위들을 보다 단순하게 모델링할 수 있는 페트리넷 기반으로 접근하고자 한다.

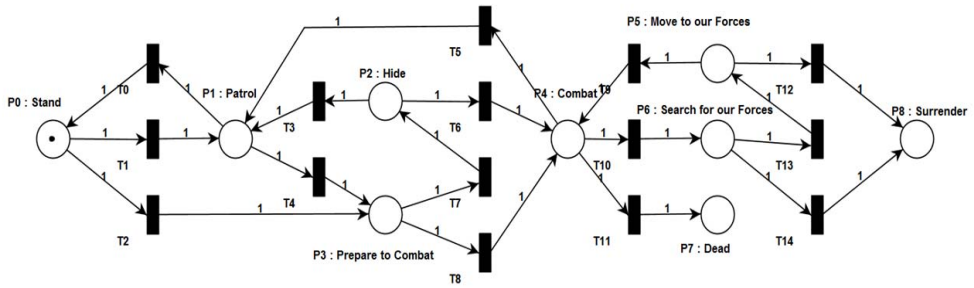
3. 시스템 설계

3.1 시스템 구조

본 연구의 시스템은 퍼지 추론 엔진이 탑재된 다수의 지능형 자율 에이전트로 구성된 다중 에이전트 시스템이다. 연구의 최종 목표는 실험 영역인 자율 지능형 교전정보 처리를 위한 상황인지 시뮬레이션 시스템의 개발이며, 시스템은 상황모델의 검증 및 퍼지 엔진의 성능 개선을 위한 시뮬레이션 환경으로 활용될 수 있다. 전체 시스템은 그림 1과 같이 다중 에이전트 구조로 설계 및 구현하였고 시스템의 운영흐름을 구조도상에 명시하였다. 시스템은 조정자 에이전트(Coordinator Agent)를 중심으로 다수의 단일 에이전트들로 구성되며, 각 단일 에이전트는 페트리넷 엔진과 퍼지 엔진을 포함하고 있다. 시스템의 간략한 흐름은 다음과 같다. 우선 실험 영역인 게임 환경으로부터 얻어진 기후, 지형, 장애물, 위치, 이벤트, 메시지와 같은 상황정보들이 조정자 에이전트의 Event Handler를 통해 각 에이전트의 페트리넷 엔진으로 전달되고, 전달된 정보는 페트리넷 모델에서 운행되며, 퍼지엔진을 통해 추론된 퍼지변수 값을 이용해 행동이 결정된다. 추후 결정된 행동은 조정자 에이전트의 Action Queue로 전달되고, 이후 뷰어를 통해 표현된다. 본 연구의 핵심 부분은 다중 에이전트 시스템, 페트리넷 엔진, 그리고 퍼지 엔진 이며, 각 모듈의 세부적 상세 설계 및 구조는 다음과 같다.

3.2 페트리넷 기반 상황모델링

이산 사건 동적 시스템은 근본적으로 비동기 시스템이며, 사건의 발생에 따라서 전개되는 동적 시스템으로 에이전트 또한 이와 같은 이산 사건 시스템이라 할 수 있다. 특히 지능형 자율 에이전트의 이벤트-행위에 대한 이산



(그림 2) 에이전트의 페트리넷 기반 이산사건 동적 시스템 모델
(Figure 2) The Petri-net based Dynamic Discrete System Model of the Agent

사건 동적 모델은 에이전트 행위의 동시 발생성과 병렬 처리를 표현하여야 하며, 페트리넷 모델링 기술은 이러한 에이전트의 해석 및 설계에 유용한 장점이 있다.

본 연구의 페트리넷 모델 시스템(페트리넷 모델러)의 특징점은 다음과 같다. 첫째, 페트리넷 모델러는 그래픽 도구로서 에이전트의 상황해석에 대하여 시각적 정보매체로 사용한다. 둘째, 에이전트의 동적이고 동시 발생적인 상황과 행위 자율성에 대한 시뮬레이션을 위하여 정보 흐름을 나타내는 토큰(token)을 사용한다. 셋째, 설계된 페트리넷 모델에 대하여 이산사건 시스템의 명확한 규정, 명확한 확인, 그리고 성능평가에 대한 통일된 환경을 제공한다.

본 연구의 페트리넷은 일반적인 페트리넷 모델의 정의와 같이 다음과 같이 5개의 원소로 정의하였다.

(정의 1) 페트리넷

$$N = < P, T, I, O, M_0 >$$

- P = {P1, P2, ..., Pm}: 유한개의 플레이스 집합
- T = {t1, t2, ..., tn}: 유한개의 트랜지션 집합
- P : 플레이스(place): 조건, 자원의 사용 가능성 또는 공정 상태를 나타내며, 원(circle)으로 표기
- T : 트랜지션(transition): 사건 또는 공정의 시작과 끝을 의미하며, 막대(bar)로 표기
- I : 입력 함수(input function): 플레이스로부터 트랜지션으로의 방향성 화살표
- O : 출력 함수(output function) - 트랜지션으로부터 플레이스로의 방향성 화살표
- M : 마킹(marking) - 각 플레이스에 있는 토큰의 개수를 의미하며, 초기 마킹을 M0로 표기

아래 그림 2는 본 연구의 실험을 위하여, 에이전트 이산사건 동적 시스템을 정의 1에 따라 설계한 것이며, 주제영역으로는 전투훈련을 위한 시뮬레이션으로 선정하여, 우선적으로 총 9개의 플레이스와 총 12개의 트랜지션으로 구성하였다. 이는 초기 모델에서 지속적으로 확대, 실험하여 얻어진 최종 모델이다. 이들 플레이스와 트랜지션에 대한 설명은 표1 및 표2와 같다. 이러한 페트리넷 모델구현의 중요점은 다음과 같다. 첫째, 에이전트의 행위에 대한 정보에 기반 하여 기본적인 상황모델의 설계를 그래픽 도구를 사용하여 수립 가능토록 구현하고 이를 내부 워크플로우에서 관리하도록 구현한다. 둘째, 에이전트의 이벤트-행위 기반의 이산사건 동적 시스템은 시스템을 구성하는 에이전트의 상태를 확인 가능토록 구현한다. 셋째, 에이전트 모델을 페트리넷 기반의 모델링 기술을 적용하여 설계의 단순화를 구한다. 넷째, 실제 엔진으로의 구현하기 이전에 모델의 안전성 및 실행 가능성에 대한 검증이 가능토록 구현한다.

(표 1) 그림 2의 플레이스 리스트
(Table 1) Place List of the Fugure 2

플레이스	설 명
P0	다음 행동을 취하기 전 잠시 대기 상태
P1	대상군을 발견하기 위한 순찰 상태
P2	대상군의 시야에 발각되지 않기 위한 엄폐 상태
P3	대상군을 발견/공격당했을 때 교전돌입 직전 상태
P4	대상군과의 교전 상태
P5	탄약 보급을 위한 이동 상태
P6	도움을 요청하기 위해 주변 아군을 탐색하는 상태
P7	사망 상태
P8	항복 상태

(표 2) 그림 2의 트랜지션 리스트
(Table 2) Transition List of the Figure 2

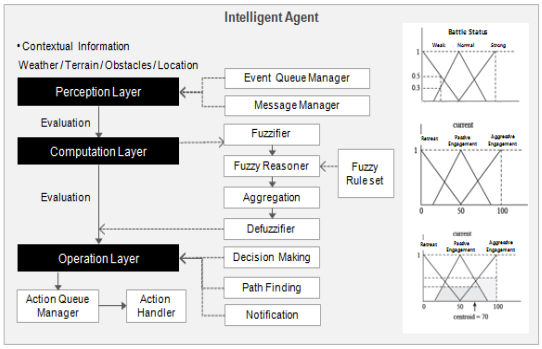
트랜지션	설 명
T0	시야에 대상군이 발견되지 않음 & 순찰 목적지에 도착함
T1	시야에 대상군이 발견되지 않음 & 다음 행동 주기 도달함
T2	시야에 대상군이 발견되지 않음 & 대상군에게 공격당함
T3	엄폐 상태 & 대상군이 시야에서 사라짐
T4	순찰 상태에서 대상군이 시야 내에 들어옴
T5	교전 중 대상군이 사망하여 전투에서 승리함
T6	엄폐 상태에서 전세 상황이 아군이 유리해짐
T7	전투대기 상태에서 전세가 불리함
T8	전투대기 상태에서 탄약이 부족함
T9	주변에 탄약을 소지한 아군이 있음
T10	교전 중 탄약을 모두 소진함
T11	교전 중 체력을 모두 소진함
T12	보급 가능 탄약이 없음
T13	주위 아군을 발견함
T14	주위 아군을 발견하지 못함

3.3 지능형 자율에이전트

본 연구의 에이전트 시스템은 주어진 상황을 추론할 수 있는 기능을 가지는 자율 지능형 에이전트이며, 시스템 설계의 주요관점은 다음과 같다.

- 첫째, 단일 에이전트와 조정 에이전트 간 낮은 결합력을 위해 이벤트 관리가 가능토록 설계/구현한다.
- 둘째, 일반적인 다중 에이전트 시스템에서 에이전트 고유의 단순 규칙을 사용하는 생성 시스템을 지능형 자율 에이전트 구조로 확장한다.
- 셋째, 에이전트의 의사 결정 및 상세한 행동 기술을 위해 퍼지추론을 사용한다.

이러한 관점으로 에이전트 시스템은 아래 그림 3의 3계층으로 정의되며, 각 계층의 상세 기능은 다음 표 3과 같다. 인지 계층에서는 외부 정보를 수신하며, 이를 연산 계층에서 퍼지규칙을 이용하여 상황 추론을 한 후, 추론된 결과를 운영 계층으로 전달하여 수행토록 한다. 이러한 설계의 장점은 페트리넷의 컴포넌트인 플레이스(Place)/트랜지션(Transition) 모듈을 핸들러에 의해 관리토록 구현함으로써 퍼지 컨트롤러의 추가가 용이하게 개발될 수 있다.



(그림 3) 지능형 자율 에이전트
(Figure 3) Intelligent Autonomous Agent

(표 3) 3계층 자율 지능형 에이전트
(Table 3) 3-Layer Autonomous Intelligent Agent

계 층	설 명
인지 계층 (Perception Layer)	상황 모델로의 입력정보를 수신 및 획득하는 계층으로 날씨/주야/지형 속성 등과 다른 에이전트로부터의 메시지를 수신하는 계층
연산 계층 (Computation Layer)	상황 모델을 기반으로 해석된 정보와 현 상황의 추론을 위해 퍼지 규칙을 사용하는 추론기관이 존재하는 계층
운영 계층 (Operation Layer)	현재 상황의 추론 결과를 의사결정에 반영하고, 이에 따른 목적지로의 경로 재설정 및 다른 에이전트로의 메시지 전파를 결정하는 계층

3.4 퍼지추론

에이전트의 상황 추론부는 상황 변수(환경 변수/이벤트 변수 등)와 같이 수치적 이해가 필요한 추론을 위하여 퍼지 추론(Fuzzy Inference) 엔진을 아래와 같이 주안점을 두어 설계한다.

- 퍼지단일 단일 규칙은 “IF A THEN B”으로 표현되며, 조건부들 간의 문장 연결어 “AND”, 그리고 시스템 출력 변수와 값으로 정의한다.
- 퍼지 규칙 기반의 계산과정은 예를 들어 그림 4에서와 같이, 소속 함수와 퍼지 규칙 테이블로 구성된다.

또한 각 입력 값에 대한 퍼지화, 퍼지추론 및 역퍼지화 과정을 거쳐 출력값을 산출하며, 이 과정에 대하여 기술하면 아래와 같다.

① 그림 3 에이전트 구조의 인지 계층으로부터 입력 (X1=60, X2=80)

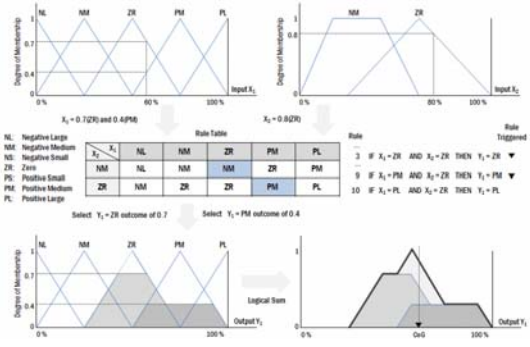
② 정의된 소속 함수에 의해 각 입력에 대한 소속정도(DoM: Degree of Membership)은 μ(X1)={0.4PM, 0.7ZR)}, μ(X1)={0.8ZR}

③ 퍼지규칙 테이블을 참조하여 규칙3, 9가 점화된다. 규칙3 "IF X1 = ZR AND X2 = ZR THEN Y1 = ZR"에서의 두 조건부의 AND 결합으로 인해 출력값은 최솟값 0.7을, 같은 이유에서 규칙 4 또한 0.4를 선택한다.

④ 출력을 위한 소속함수의 소속정도에 의해 결정되는 도형들(ZRout, PMout)의 논리합에 의하여 최종 그래프를 생성한다.

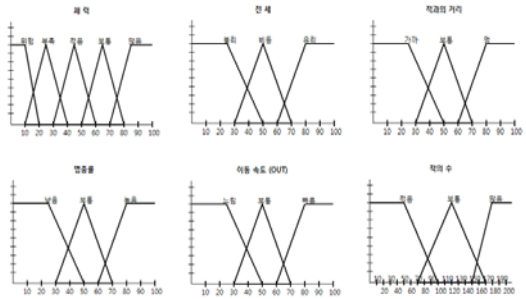
⑤ 역퍼지화 단계에서 최종 출력값을 산출하기 위해, 가장 대중적인 무게중심방법(CoG: Center of Gravity, 식 1.1)을 사용하여 획득한다.

$$CoG = \frac{\int_a^b \mu_A(x)x\,dx}{\int_a^b \mu_A(x)\,dx} \qquad \text{(식 1.1)}$$



(그림 4) 퍼지 규칙의 계산과정
(Figure 4) Computation process for the Fuzzy Rule

그림 5는 본 시스템에서 사용되는 실제 퍼지 소속 함수를 그래프로 나타낸 것이며, 체력, 전세, 적과의 거리, 명중률, 이동속도, 적의 수 등으로 구성된다. 적용된 퍼지 규칙 계산과정의 예를 들면, "체력이 위험하고, 탄약이 보통이고, 대상군이 적을 경우 전세가 불리하다"일 경우 실제 체력 수치가 10, 탄약의 수가 60, 대상군과 아군의 비율이 40인 경우 퍼지 추론이 수행 된 후의 결과 값은 전세가 20으로 낮게 나온다. 이는 탄약이 많고 대상군과 아군의 비율이 비등해도 자신의 체력이 낮기 때문에 전세가 낮게 나오는 것이다. 만일 탄약수와 대상군과 아군의 비율이 동일하고 체력 수치가 40정도로 증가될 경우 전세는 85의 수치를 나타내며 유리한 결과를 나타내게 된다.



(그림 5) 퍼지 논리를 위한 소속 함수
(Figure 5) Membership function for Fuzzy logic

표 4는 최종 자율 지능형 교전정보 처리를 위해 94개의 규칙들을 정의하고, 위에서 명시한 소속 함수들을 고려하여 전세상황, 이동속도, 명중률 계산을 위해 사용한다. 각각의 결과 값은 적과 조우, 혹은 교전 중 실시간으로 반영되며 전세의 경우 불리함, 비등함, 유리함으로 나타난 값을 이용해 후퇴, 우회, 엄폐, 소극적 전투, 적극적 전투 등의 행동으로 표현된다.

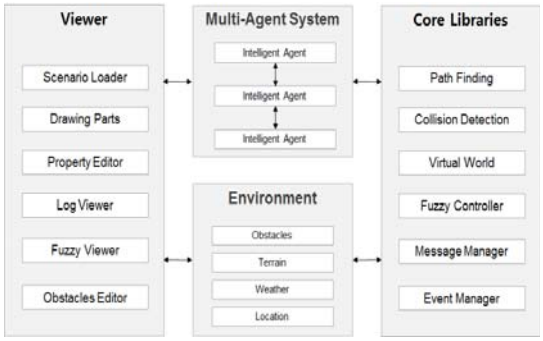
(표 4) 자율 지능형 교전정보 처리를 위한 퍼지 규칙
(Table 4) Fuzzy Rule for Autonomous Intelligent Combat Information Processing

1	체력_적음	탄약_적음	적의 수_적음	전세_비등
2	체력_적음	탄약_적음	적의 수_보통	전세_불리
3	체력_적음	탄약_보통	적의 수_적음	전세_비등
4	체력_보통	탄약_많음	적의 수_적음	전세_유리
5	체력_많음	탄약_적음	적의 수_적음	전세_유리
6	체력_보통	속도_느림	거리_가까움	명중_높음
...	...	...	...	...
94	체력_적음	주변 장애물_많음		속도_느림

4. 시스템 구현

본 연구에서는 다수 대 다수의 모의전투 상황을 묘사한 게임 환경을 대상영역으로 선정하였고 실험환경을 포함하는 구조도는 다음 그림 6과 같다. 시스템은 뷰어 및 환경정보, 멀티 에이전트 시스템 그리고 코어 라이브러리들로 구성되며, 각각의 기능은 다음과 같다.

우선, 코어 라이브러리는 게임 환경에서의 에이전트들이 사용할 필수 기능들을 포함하고 있다. 이는 길 찾기 알고리즘, 각 에이전트와 장애물 간의 충돌감지, 그리고 실험 영역인 가상세계의 속성, 퍼지 컨트롤러, 메시지, 이벤트를 관리한다. 환경정보는 장애물, 지형, 날씨, 장소 등을 관리 하며 멀티 에이전트 시스템은 에이전트의 의사결정, 환경인지, 행동, 그리고 에이전트간의 통신을 담당한다. 뷰어는 게임 환경과 에이전트의 모습을 2D로 표현하고, 시나리오 생성, 객체 표현, 속성 편집, 로그 뷰어, 퍼지 뷰어, 장애물 편집과 같은 세부 기능들을 제공한다.



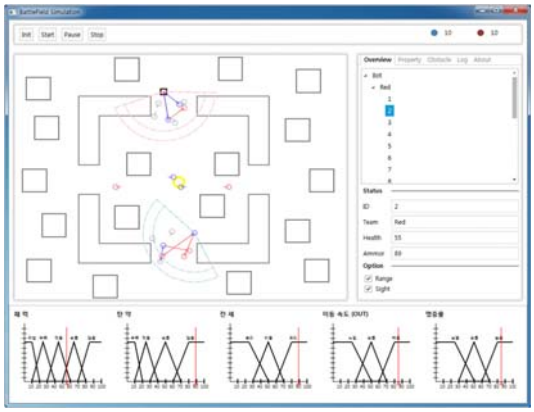
(그림 6) 실험을 위한 전체 시스템 구조도
(Figure 6) System Structure for the Experimentation

4.1 시스템 개발환경

실험을 위한 시스템의 구현을 위해서, 사용자 인터페이스는 C# 언어로, 퍼지 추론을 위한 라이브러리와 코어 라이브러리, 멀티 에이전트 시스템은 C++을 사용하였다. 그림 7은 레드 팀(10인) vs 블루 팀(10인)의 모의 전장 상황을 테스트 할 수 있는 실행 화면이다. 화면의 세부 기능은 다음과 같다.

- 화면 상단에 있는 메뉴는 시뮬레이션의 시작, 정지 등을 제어한다.
- 화면 중앙의 메인 창에서는 시뮬레이션 대상인 개체들의 움직임을 표시한다.
- 화면 우측은 각 에이전트의 속성 값을 표시하며 편집할 수 있다
- 화면 하단에서는 퍼지 소속 함수의 범위 및 출력 값을 표시 한다.

테스트베드의 실행은 먼저, 미리 정의한 시나리오를 Init버튼을 통해 로드하고, Start버튼을 통해 시뮬레이션을 시작 한다. 시나리오에 기술된 목표를 만족할 때 까지 시뮬레이션이 수행되며, 수행 중 퍼지 추론 과정은 하단 패널에 실시간으로 반영된다.



(그림 7) 자율 지능형 교전 정보처리 모듈의 테스트베드
(Figure 7) Testbed for Autonomous Intelligent Combat Information Processing Module

4.2 실험

실험은 두 가지 시나리오에 대해 각각 승률과 생존율을 산출하여 성능을 평가하였다. 레드 팀은 본 논문에서 제안한 퍼지 규칙 기반 에이전트로 구현하며, 블루 팀은 유한 상태 기계로 행동을 구현 한다.

시나리오 1: 레드 팀(10인)과 블루 팀(10인)의 전투모의
시나리오 2: 레드 팀(10인)이 블루 팀(10인)을 상대로 목표 지점에 먼저 도달하여 목표지점을 점령하는 미션을 수행한다.

시나리오 1은 장애물이 있는 환경에서 보다 최적화된 전투를 수행하는 레드 팀의 의 승률을 산출 한다. 시나리

오 2는 레드 팀의 미션 수행을 위해 최대한 전투를 지양하고 목표 지점을 우선 점령하며, 점령 시 살아남은 레드 에이전트의 수를 산출 한다. 각 실험은 20회 반복 수행하였으며 실험 결과 시나리오 별 레드 팀의 승률과 생존율의 평균은 표 5와 같다. 에이전트의 현실감 있는 행동 묘사를 위해서는 지속적으로 퍼지 규칙을 정제하는 과정이 필요하다. 이를 위해서 반복적인 시뮬레이션을 통한 퍼지 규칙과 퍼지 소속 함수의 범위를 정제하여 실험을 수행하였다.

반복적인 실험 결과, 시나리오 1에 대한 레드 팀의 승률은 70%, 시나리오 2에 대한 에이전트의 생존율은 82%로 두 시나리오 모두 퍼지 규칙 기반 에이전트가 유한 상태 기계보다 좋은 성능을 보였다.

(표 5) 실험 결과
(Table 5) Result of the Experimentation

시나리오 1		시나리오 2	
승	패	생존	사망
14	6	164	36

5. 결론 및 향후연구

본 연구에서는 상황인지 시뮬레이션을 위한 다중에이전트 시스템을 설계 및 구현하였으며, 연구의 주요 관점은 다음과 같다. 첫째, 본 연구의 시스템은 상황 인지 시스템 개발 시 모델링 및 구축이 가능하다. 둘째, 실제보다 적은 시간과 노력으로 목표시스템의 실증적 모의를 가능하게 하는 게임 시뮬레이션 환경을 제공하며, 이를 위해 자율 에이전트로의 이식이 가능한 다양한 컴포넌트들로 구성된다. 셋째, 모의 영역에 대하여 기술 가능한 상황 모의 엔진은 행위-이벤트 기반의 페트리넷 으로 모델링되어 에이전트 행위의 동시 발생성과 병렬처리를 위한 엔진으로 구성된다. 넷째, 모의 대상의 의사결정 논리는 퍼지규칙을 사용하는 추론엔진으로 구성하여 다양성을 제공한다. 다섯째, 에이전트의 행위에 대한 정보에 기반하여 기본적인 상황모델의 설계를 그래픽 도구를 사용하여 손쉽게 구현할 수 있으며, 이를 내부 워크플로우에서 관리 할 수 있다. 여섯째, 에이전트 모델은 페트리넷 기반 모델링 기술을 적용함으로 설계의 단순화를 얻을 수 있다. 일곱째, 실제 엔진으로 구현되기 전에 모델의 안전성 및 실행 가능성에 대한 검증이 가능하다. 마지막으로, 실험을 통해서 상황추론 향상율의 계산을 위한 로그 데이터를 생성하여 분석함으로 상황모델과 퍼지 추론의 검증이 가능하다.

현재 구현된 시스템에서 각 에이전트는 독립적인 퍼지 추론으로 의사 결정을 함으로써 팀 미션에 부적절한 행동을 보이는 경우가 있으며, 페트리넷 모델러와 에이전트 시스템이 분리되어있어 모델링 후에 실제 에이전트에 적용 후 수정이 용이치 않는 문제가 있다. 향후 연구에서는 퍼지 추론을 에이전트 조정자에 위임하여 전체 에이전트의 퍼지 추론을 관장하고, 페트리넷 모델러와 에이전트 시스템을 결합하여 모델의 실증적인 검증을 보다 용이하도록 할 계획이다.

참 고 문 헌 (References)

[1] S. Yildirim, and S. Stene, "A survey on the need and use of AI in game agents." Proceedings of the 2008 Spring simulation multiconference, pp124-131, 2008. http://dx.doi.org/10.5772/8968

[2] R. Khosla and T. Dillon, "Engineering intelligent hybrid multi-agent systems", Kluwer Academic Publishers, 1998. http://dx.doi.org/10.1007/978-1-4615-6223-8_4

[3] G. Weiss, Ed., "Multiagent systems: a modern approach to distributed artificial intelligence", MIT Press, 1999. http://dx.doi.org/10.1142/s1469026801000159

[4] A. Mahfoudhi, B. Marzougui, and M. Abid,"Agent Petri Nets: Theory and Application", International journal of Sciences and Techniques of Automatic control & Computer engineering, vol4, No2, pp1402-1419, 2010. http://dx.doi.org/10.1007/3-540-63139-9_31

[5] W. Reisig, "Elements of distributed algorithms: modeling and analysis with Petri nets", Springer-Verlag, New York, 1998. http://dx.doi.org/10.1049/ip-sen:19990509

[6] A. Desrochers and R. Y. Al-Jaar, "Applications of Petri Nets in Manufacturing Systems: Modeling, Control and Performance", IEEE, 1995 http://dx.doi.org/10.1109/mcs.1995.476396

[7] B. Marzougui, K. Hassine, and K. Barkaoui, "A New Formalism for Modeling a Multi Agent Systems: Agent Petri Nets", Journal of Software Engineering & Applications, vol.3, pp1118-1124, 2010. http://dx.doi.org/10.4236/jsea.2010.312130

[8] S. Pujari, and S. Mukhopadhyay,“Petri Net: A Tool for Modeling and Analyze Multi-agent Oriented Systems”, International J. of Intelligent Systems and Applications, vol.10, pp103-112, 2012.
http://dx.doi.org/10.5815/ijisa.2012.10.11

[9] Y. Yao, “A Petri net model for temporal knowledge representation and reasoning,” IEEE Transactions on Systems, Man and Cybernetics, Vol. 24, pp1374 – 1382, 1994.
http://dx.doi.org/10.4304/jcp.4.10.981-996

[10] R. Bastide, “Approaches in unifying Petri nets and the object-oriented approach,” Proceeding of the International Workshop on Object-Oriented Programming and Models of Concurrency, Turin, Italy, June, 1995.
http://dx.doi.org/10.5772/7507

[11] W. Pedrycz and H. Camargo, “Fuzzy timed Petri nets, fuzzy sets and systems,” vol.140, pp301 – 330, 2003.
http://dx.doi.org/10.1016/s0165-0114(02)00524-9

[12] X. Huang, H. Xu, and P. Jia, “Fuzzy Timed Agent Based Petri Nets for Modeling Cooperative Multi-Robot Systems”, International Journal of Communications, Network and System Sciences, vol.2, pp827-835, 2009.
http://dx.doi.org/10.4236/ijcns.2009.29096

[13] L. Yifan, P. Musilek, and L. Wyard-Scott. ″Fuzzy logic in agent-based game design“, Proceedings of the North American fuzzy information Processing society. vol.2, pp734-739, 2004.
http://dx.doi.org/10.1109/nafips.2004.1337393

[14] S. Nugroho, I. Widiastuti, M, Hariadi, and M, Purnomo, ″Fuzzy coordinator based intelligent agents for team coordination behavior in close combat games“, Journal of Theoretical & Applied Information Technology, vol.51, No.2, pp317-323, 2013.
http://www.jatit.org/volumes/Vol51No2/22Vol51No2.pdf

[15] M. J. Wooldridge, “Introduction to Multiagent Systems”, John Wiley & Sons, Inc., 2001.
http://www.cs.ox.ac.uk/people/michael.wooldridge/pubs/imas/IMAS2e.html

[16] SeSam, http://www.sesam-web.org/

[17] SimWalk, http://www.simwalk.com/

[18] W. Clancey, P. Sachs, M. Sierhuis, and R. Hoof, “Brahms: simulating practice for work systems design”, International Journal of Human-Computer Studies, vol.49, pp831-865. 1998.
http://dx.doi.org/10.1006/ijhc.1998.0229

[19] P. Upadhyay, and A. Dutta, “A Conceptual Graph Petri Net Model based Multi-agent System”, International journal of Computer Applications, vol.45, No.12, pp10-17, 2012.
http://research.ijcaonline.org/volume45/number12/pxc3879390.pdf

[20] T. Murata, “Petri nets: Properties, analysis and applications”, Proceedings of the IEEE, vol. 77, no. 4, pp. 541 – 580, 1989.
http://dx.doi.org/10.1109/5.24143

[21] D. Xu, R. Volz, T. Ioerger, and J. Yen, “Modeling and verifying multi-agent behaviors using predicate/transition nets”, Proceedings of the 14th international conference on Software engineering and knowledge engineering. ACM Press, pp193 – 200, 2002.
http://dx.doi.org/10.1145/568760.568794

[22] H. J. Ahn and S. J. Park, “Modeling of a multi-agent system for coordination of supply chains with complexity and uncertainty”, Lecture Notes in Computer Science, vol. 2891, pp13 – 24, 2003
http://dx.doi.org/10.1007/978-3-540-39896-7_2

[23] W. Reisig, “Elements of distributed algorithms: modeling and analysis with Petri nets”, Springer-Verlag, 1998.
http://dx.doi.org/10.1049/ip-sen:19990509

[24] M. K. Gazdare, “An Automatic Petri-net Generator for Modeling Multi-agent Systems”, The Proceedings of the International Conference on Software Engineering Advances, pp128-133, 2013.
http://dx.doi.org/10.1109/icsmc.1994.399917

◐ 저 자 소 개 ◑

김 동 민 (DongMin Kim)
2014년 경북대학교 컴퓨터공학부 졸업(학사)
2014년~현재 국민대학교 대학원 컴퓨터공학과 석사과정
관심분야 : 인공지능, 시뮬레이션, 빅데이터, 웹 서비스.
E-mail : kdm1171@kookmin.ac.kr

최 진 우 (JinWoo Choi)
1998년 한성대학교 전산학과 졸업(학사)
2000년 국민대학교 대학원 전산과학과 졸업(석사)
2004년 국민대학교 대학원 전산과학과 졸업(박사)
2009년~현재 국민대학교 컴퓨터공학부 겸임교수
관심분야 : 인공지능, ITS, 지능형 에이전트, 정보 보호
E-mail : jnwochoi@kookmin.ac.kr

우 종 우 (ChongWoo Woo)
1991년 Illinois Institute of technology 전산학과 졸업(박사)
1994년~현재 국민대학교 컴퓨터공학부 교수
관심분야 : 지능형 에이전트, 상황인식, Tutoring System, Modeling&Simulation
E-mail : cwwoo@kookmin..ac.kr