

## 멀티 에이전트 기반의 지능형 시뮬레이션 도구의 개발

우종우\*, 김대령\*\*

# A Development of Intelligent Simulation Tools based on Multi-agent

Chongwoo Woo\*, Daeryung Kim\*\*

### 요약

시뮬레이션이란 실세계의 다양한 객체들의 구조와 행위에 대한 자료를 수집하여 모델링하고 이를 컴퓨터 프로그램으로 모의 실험함을 말하며, 주요 기반기술들로서 DEVS (Discrete Event System Specification) 형식론을 비롯하여 페트리 넷이나 구조적 오토마타 등이 연구되고 있다. 그러나 이러한 시뮬레이션의 연구영역이 보다 다양화되고 복잡하게 발전됨에 따라, 최근에는 인공지능의 지능형 에이전트기법을 도입하여 해결하는 연구가 활성화 되고 있다. 본 연구에서는 시뮬레이션에 관한 모의실험을 보다 원활히 수행하기 위해서 지능형 멀티 에이전트기반의 시뮬레이션 도구를 개발하고자 한다. 이러한 도구의 특징은 첫째, 인공지능의 기능들을 라이브러리로 제공할 수 있고, 둘째, 유한상태기계(FSM)기반으로 에이전트 시스템을 설계하여, 시뮬레이터등의 보조툴들을 제공함으로써 사용자들이 보다 편리하게 시뮬레이션 시스템을 개발할 수 있는 프레임워크를 제공한다. 시스템의 구성은 코어 시뮬레이션 엔진, 유틸리티, 그리고 기타 보조툴들로 구성 하였으며, 현재까지 개발된 시스템으로 몇 가지 영역을 대상으로 실험을 하였고 그 결과를 기술하였다.

### Abstract

Simulation means modeling structures or behaviors of the various objects, and experimenting them on the computer system. And the major approaches are DEVS(Discrete Event Systems Specification), Petri-net or Automata and so on. But, the simulation problems are getting more complex or complicated thesedays, so that an intelligent agent-based is being studied. In this paper, we are describing an intelligent agent-based simulation tool, which can supports the

• 제1저자 : 우종우

• 접수일 : 2007.10.23, 심사일 : 2007.10.25, 심사완료일 : 2007.12.15.

\* 국민대학교 컴퓨터공학부교수 \*\* 국민대학교 컴퓨터공학부 대학원

※ 본 논문은 2007학년도 국민대학교 교내연구비 지원을 받아 수행되었음

simulation experiment more efficiently. The significances of our system can be described as follows. First, the system can provide some AI algorithms through the system libraries. Second, the system supports simple method of designing the simulation model, since it's been built under the Finite State Machine (FSM) structure. And finally, the system acts as a simulation framework by supporting user not only the simulation engine, but also user-friendly tools, such as modeler, scriptor and simulator. The system mainly consists of main simulation engine, utility tools, and some other assist tools, and it is tested and showed some efficient results in the three different problems.

▶ Keyword : 시뮬레이션 툴(Simulation Tools), 지능형 에이전트(Intelligent Agent), 인공지능(Artificial Intelligence)

## 1. 서론

시뮬레이션이란 실제계의 다양한 객체들의 구조와 행위에 대한 자료를 수집하여 모델링하고 이를 컴퓨터 프로그램으로 모의 실험함을 말한다. 주로 복잡한 시스템의 내부구조 및 객체들의 상호작용을 관찰하거나 상태전이를 예측 하는 경우, 또는 새로운 시스템을 설계하는 경우 등에 효과적이며, 전반적으로 연구용, 훈련용, 오락용 등으로 그 사용 목적 및 적용분야가 다양하다. 실제로 국내의 연구에서는 의료분야 [1,2], 정보보호분야 [3,4], 그리고 다수의 군 관련 분야에 [5,6] 활발히 연구되고 있다. 이러한 시뮬레이션 연구의 기반기술은 대부분의 경우 DEVS (Discrete Event Systems Specification) 형식론 [7] 을 비롯하여 petri-net이나 구조적 오토마타 등을 중심으로 연구되고 있다. 그러나 이러한 시뮬레이션의 연구영역이 보다 다양화되고 지능화됨에 따라, 최근에는 인공지능의 지능형 에이전트 기법을 도입하여 해결하는 연구가 활성화 되고 있다 [8,9].

본 논문에서는 이러한 지능적 시뮬레이션의 연구를 보다 활성화 하기 위해서, 지능형 멀티 에이전트 기반의 시뮬레이션 도구를 개발하고자 한다. 본 연구의 특징은 다음 몇 가지로 설명된다. 우선, 본 연구의 도구는 범용적인 에이전트 기반의 시뮬레이션 엔진구축을 목표로 한다. 따라서 시스템은 인공지능 라이브러리의 제공을 우선적인 목표로 하며, 이러한 라이브러리를 사용하게 되면, 필요시 지능적 기능들을 보다 편리하게 활용할 수 있는 장점을 제공할 수 있을 것이다. 둘째, 본 연구에서는 다양한 인공지능의 기법들 중에서 유한상태기계 (Finite State Machine: FSM)을 기반으로 기본적인 시뮬레이션 모델을 설계하였으며, 이를 기반으로 지능형 에이전트의 기능을 수행할 수 있게 구성하

였다. 이러한 유한상태기계 기반의 에이전트를 이용하게 되면 모델 설계가 용이하며, 시스템을 구성하고 있는 모든 에이전트들의 상태를 쉽게 확인 할 수 있고, 시뮬레이션 설계 모델을 단순화 할 수 있는 장점이 있다. 셋째, 시뮬레이션 엔진의 기능을 보다 확대하여, 추론 엔진뿐 만 아니라, 스크립트 및 모델링 제작 툴까지, 보다 사용자 친화적인 다양한 기능들을 모두 포함하는 지능적 시뮬레이션 시스템 개발을 위한 프레임워크를 제공할 수 있을 것이다.

본 연구의 시스템은 코어엔진, 유틸리티, 그리고 기타 보조틀 들로 구성하였다. 코어 엔진은 지능형 에이전트를 기본적인 구조로 하여, 효과적인 객체관리를 위한 유한상태기계 모듈, 퍼지 규칙생성을 위한 퍼지 모듈, 범용적인 검색을 위한 검색모듈들로 구성되어 있다. 또한, 유틸리티로는 윈도우 기반 응용시스템을 위한 GDI 엔진, 3D기반 응용 프로그램을 위한 DX 엔진들로 구성하였고, 기타 보조틀들은 시뮬레이션의 객체들을 쉽게 모델링하기위한 모델링툴, XML 스크립트 및 시뮬레이션의 결과를 보기위한 시뮬레이터 등으로 설계하였다. 본 연구에서는 현재까지의 결과를 엘리베이터 시뮬레이션, 퍼지 에이전트 시뮬레이션, 그리고 로봇 에이전트 길찾기 영역을 실험대상으로 하여 실험하였으며, 지속적으로 각 모듈의 기능을 확대하여 연구할 예정이다.

본 논문의 구성은 다음과 같다. 2장 관련 연구에서는 에이전트 기반 시뮬레이션 연구에 필수적인 이론적 기반 및 몇 가지 사례연구에 대하여 설명 하고, 3장에서는 시스템의 설계에 관하여 기술한다. 4장에서는 시뮬레이션 테스트 결과를 기술하고, 5장에서 결론을 맺는다.

## II. 관련연구

시뮬레이션을 위한 주요 기반 기술들로는 구조적 오토마타, Petri-Net, 에이전트, DEVS 형식론 등이 존재한다.

본 장에서는 이러한 기반 기술중에서, 이산사건 시스템의 모델링 시뮬레이션을 하는 DEVS 형식론과 에이전트 시스템에 대하여 기술한다.

## 2.1 DEVS 형식론

Zeigler에 의해 제안된 DEVS 형식론[7]은 이산사건 시스템을 모듈 별로 나누어서 계층적으로 모델링 할 수 있는 방법을 제공한다. 기본적으로 집합이론에 기반을 두고 이산사건 시스템을 수학적 공식에 의거하여 객체 지향적으로 모델링 할 수 있는 틀을 제공한다. 또한 입력사건/출력사건을 명시적으로 정의하고 전체시스템을 부 시스템들의 계층적 결합으로 모델링 한다. 전체 시스템을 부 시스템으로 나누어 가는 과정에서 부 시스템 객체를 결합모델(coupled model)로 정의하며 더 이상 나눌 수 없는 모델 객체를 원자모델(atomic model)로 정의한다. 이러한 원자모델은 시스템의 상태천이를 집합 기호를 사용한 수학적 공식에 의해 명세화하며 원자모델들 간에는 어떤 정보도 공유되지 않고, 필요한 정보교환은 오직 입/출력에 의해서만 이루어지게 하는 모듈(modular) 명세 기법을 사용하고 있다.

이러한 DEVS 모델은 이산사건 시스템 모델을 상태방정식으로 표현할 수 있는 수학적인 틀이며, 따라서 DEVS 모델 방정식은 대상 시스템의 논리 검증 및 성능해석에 통합적으로 적용할 수 있다. 이러한 관점에서 DEVS 형식론은 모델링의 간편성 및 객체 지향적 프로그래밍의 구현이 용이한 장점이 있으며, 추가적인 요소들이 지속적으로 연구되고 있다 [10][11].

## 2.2 에이전트 시스템

에이전트란 사용자를 대신하여 작업을 자동으로 해결해주는 소프트웨어라고 할 수 있다. 따라서 에이전트는 목적을 달성하기 위하여 자율적으로 행동하고 환경에 반응하게 된다. 또한 자율적으로 생성, 작동, 소멸하면서 다른 에이전트와의 협력을 통해 문제를 해결하고 목표를 달성해 나가며 기존의 소프트웨어와는 다르게 자동성, 목표지향성, 유연성, 자율성, 사회성, 적응성, 그리고 이동적 특성을 가질 수 있다 [12][13].

이러한 에이전트는 다양한 기술들로 개발이 가능하지만, 본 논문에서는 유한상태 기계를 기반으로 설계 및 개발하였다. 유한상태 기계의 알고리즘 자체는 간단하지만, 거의 모든 시스템들이 상태들로 표현이 될 수 있어 많이 사용되고 있다. 기본기능은 하나의 입력에 의해 현재 상태에서부터 다른 상태로 전이하게 된다. 이렇게 구성된 에이전트는 시물

레이션의 각 객체로 모델링을 하게 되며, 각각의 에이전트가 시뮬레이션 진행 도중, 주어진 상태로 변화하게 되며 그 변화에 따른 특정 작업을 수행하게 된다.

## 2.3 사례연구

에이전트기반 시뮬레이션 시스템은 최근 활발히 연구되고 있으며, 그중 대표적인 몇 가지 시스템의 특징은 다음과 같다.

### 2.3.1 SeSam

SeSam[14] 은 에이전트 기반 시뮬레이션을 위한 일반적인 모델링 환경을 제공 한다. 시스템의 주요 특징은 상황 변수들과 UML과 유사한 다이어그램을 제공 한다는 것이다. 또한 시스템은 Java기반으로 구현이 되어있으며, 사용자들은 기존의 프로그래밍기술이 없이 그래픽으로 시뮬레이션 제작이 가능한 것이 특징이다.

### 2.3.2 SimWalk

SimWalk[15] 은 보행자를 위한 에이전트 기반의 시뮬레이션 소프트웨어 이다. SimWalk를 사용하여 많은 사람들이 모이는 곳, 예를 들면, 쇼핑몰, 기차역, 다리, 공항등의 장소에서 개개인의 상황을 모델링하고 시뮬레이션을 할 수 있는데, 주로 보행자의 보행속도, 통행로의 복잡도, 통행인의 숫자 등을 시뮬레이션 할 수 있다.

### 2.3.3 Ps-i

Ps-i[16] 는 에이전트기반 시뮬레이션 시스템으로 cross-platform의 형태를 제공한다. 시스템의 모델은 Tcl/Tk 스크립트 언어를 사용하며 그래픽 인터페이스를 제공한다. 또한 모델의 변수를 수시로 변경할 수 있으며, 다양한 사용자 친화적인 기능들을 제공한다.

### 2.3.4 Brahms

Brahms[17]는 Java VM상에서 구동되는 멀티 에이전트 툴로서, 시뮬레이션모드와 실시간 모드를 같이 지원한다. Brahms의 특징은 멀티 에이전트 시스템 개발이 용이한 환경제공에 있으며, 컴파일러, 그래픽 인터페이스, 에이전트 뷰어 등 사용자에게 편리한 기능도 함께 제공한다.

## III 시스템 설계

본 연구의 시스템은 코어 시뮬레이션 엔진, 유틸리티, 그리고 보조 틀의 세부분으로 구성되어 있으며, 각 모듈의 세부 기능은 다음과 같다.

### 3.1 코어엔진

코어엔진은 크게 엔티티(Entity) 시스템과 에이전트 시스템으로 구분된다. 엔티티 시스템은 본 시스템의 전역에서 사용되는 기본 시스템이다 (그림 1). 엔티티 시스템은 엔티티 들의 관리와 정보관리를 편리하게하며, 시뮬레이션에서 사용되는 대부분의 객체들은 엔티티 클래스를 상속받아 만들어지게 된다. 시스템의 세부 기능은 다음과 같다.

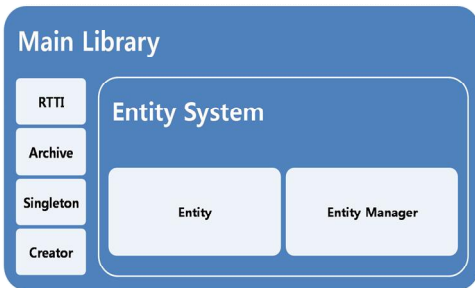


그림 1. 엔티티 시스템  
Fig 1. Entity System

#### 3.1.1 엔티티 시스템

- 1) 엔티티 관리자 (Entity Manager): 엔티티 객체 들은 ID와 name 으로 구분되고, 관리자에 의해 관리된다. 또한 Add, Remove, Get 등의 함수를 통해 엔티티 들이 추가, 제거 및 생성된다.
- 2) RTTI(Run-Time Type Information): 엔티티 객체들의 실시간 타입정보를 문자열로 관리하며, 상위 클래스들과의 관계에 대한 정보를 제공 해준다.
- 3) Archive: 엔티티 객체의 메모리 정보의 기록을 쉽게 관리해 준다. 또한 ADD함수를 통해 기록하기 원하는 메모리를 설정하게 되고, LOAD, SAVE 등의 함수를 통해 정보를 읽거나 쓰게 된다.
- 4) Singleton: 단일 패턴을 편리하게 사용할 수 있는 매크로를 제공해 준다.
- 5) Creator: 엔티티 객체의 생성을 편리하게 관리해주는 매크로를 제공해 준다.

#### 3.1.2 에이전트 시스템

에이전트 시스템은 범용적인 에이전트 시뮬레이션을 위한 인공지능 모듈들을 제공한다. 본 연구에서는 유한상태기계, 퍼지, 그리고 그래프 탐색 알고리즘들을 인공지능기법으로 적용하였고, 지속적으로 인공지능 기법들을 추가할 예정이다. 본 시스템의 구성은 엔티티 시스템을 이용한 에이전트 관리와 유한상태기계를 이용한 각 에이전트들의 상태관리 및 에

이전트 간 메시지 전달시스템으로 이루어진다 (그림 2).

#### 1) 유한상태기계 (Finite State Machine)

유한상태기계는 개념적으로 간단하고, 효율적이며, 확장이 쉬우면서도 다양한 상황처리를 할 수 있는 장점이 있다. 그림3은 유한상태기계를 이용한 에이전트시스템의 기본구조를 나타낸 그림이다. 에이전트 관리자는(agent manager) 에이전트들을 관리하게 된다. 각 에이전트 들은 유한상태기계를 가지고 있으며, 유한상태기계를 에이전트들의 각 상태를 가지게 된다. 메시지 관리자 (MessageManager)를 통해서 서로간의 메시지를 전달 하게되며, 메시지의 처리는 에이전트가 자체적으로 처리하거나, 각 상태에서 처리한다.

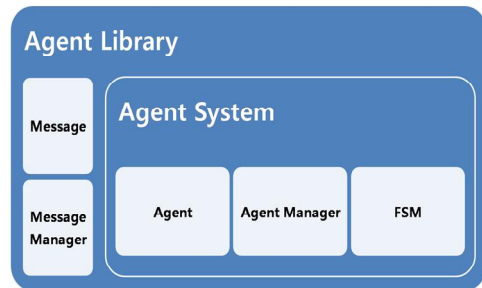


그림 2. 에이전트 시스템 구조  
Fig 2. Structure of the Agent System

#### 2) 퍼지 모듈

퍼지추론은 퍼지변수와 퍼지규칙들로 이루어진다. 퍼지 변수는 퍼지집합으로 이루어지고, 퍼지규칙은 And, Or 연산과 Term으로 이루어진다. 퍼지모듈 사용시 예는 먼저 퍼지변수들을 등록하고, 변수들에 관련된 퍼지규칙을 등록함으로써 사용할 수 있게 된다.

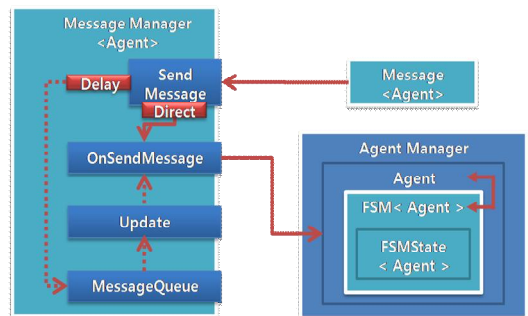


그림 3. 유한상태기계 기반의 에이전트 시스템 구조  
Fig 3. FSM-based Agent System

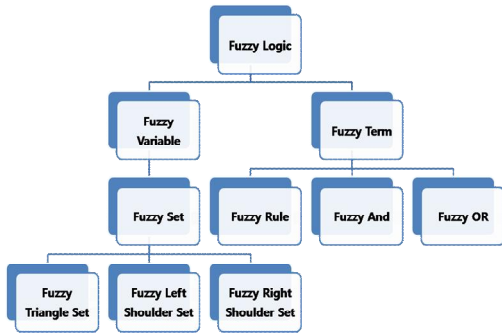


그림 4. 퍼지 모듈의 구조  
Fig 4. Structure of Fuzzy Module

### 3) 그래프 모듈

그래프 모듈은 기본적으로 그래프 기능을 제공하며, 추가적으로 탐색(search) 기능을 제공한다. 그래프는 Node 와 Edge 로 구성된다. Search 클래스는 그래프 클래스를 상속받아 구현되었으며, 기본적으로 Breadth First Search(BFS), Depth First Search(DFS), A\* 등의 탐색 알고리즘을 구현하였다.

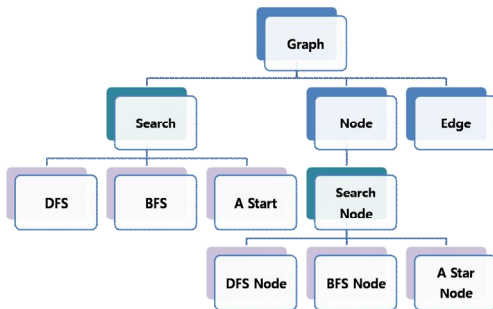


그림 5. 그래프 모듈의 구조  
Fig 5. Structure of Graph Module

## 3.2 유틸리티 라이브러리

유틸리티 라이브러리는 윈도우 기반 응용시스템을 위한 GDI 엔진, 3D기반 응용 프로그램을 위한 DX 엔진들로 구성하였다. 각각의 세부기능은 다음과 같다.

### 3.2.1 GDI 라이브러리

GDI 기반 어플리케이션에서 쉽게 GDI를 이용하여 agent 시뮬레이션의 결과를 보여주기 위한 라이브러리이다. LxGDI 클래스는 메모리 DC를 통해 비트맵 버퍼를 만들고 그를 이용하여 더블 버퍼링을 하게 해준다. 이를 이용하면 화면 렌더링시 화면의 깜빡임이 없어진다. LxGDIBitmap

클래스는 Bitmap 파일을 쉽게 읽고 화면에 그리게 해주는 클래스이다. LxSprite 클래스는 2D 렌더링을 위한 이미지 클래스이다. 투명키 처리와 알파 처리 등의 기능을 포함하고 있으며, 이미지 프레임 관리를 해준다.

### 3.2.2 DX 라이브러리

DX 라이브러리는 GDI 기반 어플리케이션의 렌더링 품질의 부족함과 3D 효과를 이용하기 위한 DX 라이브러리이다. 3D 렌더링, 리소스 관리, UI 처리 등의 기능을 포함하고 있다. 세부적인 구성은 다음과 같은 모듈들로 구성된다.

- 1) Renderer : D3D9 기반의 렌더러와 관련된 클래스들로 이루어진 모듈로서, Texture, VertexBuffer, IndexBuffer 등의 D3D9 장치에 의해 관리되는 리소스들에 대한 클래스들도 포함된다.
- 2) Resource : Image, Sound, Font, Geometry 등의 리소스와 그 리소스를 관리하는 클래스들로 이루어진 모듈이다. Resource 들은 문자열 아이디를 이용하여 구분되어지며, 이 부분은 스크립트 연동에서도 사용된다.
- 3) User Interface(UI) : DX 렌더러 기반에서 화면에 출력되는 UI 모듈로서 Control들과 Event 처리를 위한 클래스들로 구성되어지며, XML로 연동된다.
- 4) Script : 리소스 관리 및 시뮬레이션 전반적인 설정과 관련된 클래스들로 이루어진 모듈로서, 기본 텍스트 파일과 XML 파일 등을 다루게 된다.

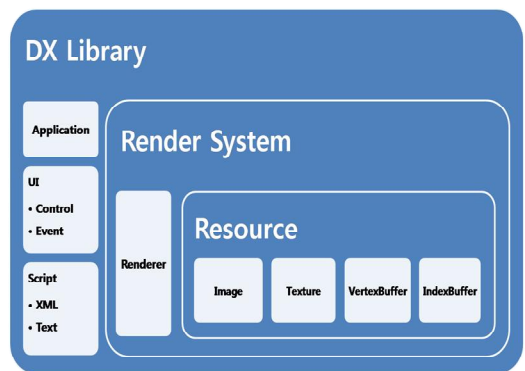


그림 6. DX 기반 라이브러리의 구조  
Fig 6. Structure of DX based Library

## 3.3 보조 틀

시스템의 기타 구성으로 보조틀들이 있다. 시뮬레이터는 MFC와 GDI 라이브러리를 이용하여 구현되며, 시뮬레이션 결과를 확인할 수 있다. 현재는 DX library를 포함한 가장

기본적인 기능들로만 구현된 상태로서, 시뮬레이션의 결과를 시작적으로 확인할 수 있으며, 콘솔창또는 파일로 출력되는 메시지를 확인할 수 있다.

그 외에는 시뮬레이션의 객체들을 쉽게 모델링하기 위한 모델링 툴이 추가될 것이며, 또한 이벤트 처리를 위해 Lua를 이용한 스크립터와 XML을 이용한 설정 스크립트 모듈 등이 추가될 예정이다.

### IV. 실험 및 고찰

본 연구에서 설계된 시스템을 사용하여 다음과 같은 세가지 영역으로 실험을 하였으며, 본 장에서는 그 실험결과에 관하여 기술한다.

#### 4.1 엘리베이터 시뮬레이션

엘리베이터 시뮬레이션은 임의의 시간에 사람이 생성되어, 임의의 층에 가서 일을 하고 돌아오는 시나리오이며, 세부설계는 다음과 같다.

##### 4.1.1 계층도

엘리베이터 시뮬레이션을 위한 각 객체 클래스들의 계층적 구조도는 다음 그림 7과 같다.

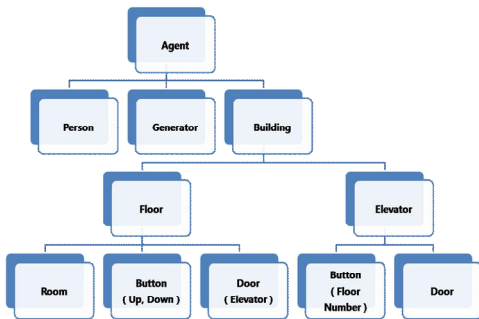


그림 7. 클래스 계층도  
Fig 7. Hierarchy of the Classes

##### 4.1.2 클래스

- 1) Person : 이동하는 사람을 나타내는 클래스
- 2) Generator : 사람을 생성하는 클래스
- 3) Building : 건물을 나타내는 클래스
- 4) Floor : 층을 나타내는 클래스
- 5) Button : 버튼을 나타내는 클래스
- 6) Door : 문을 나타내는 클래스

7) Room : 방을 나타내는 클래스

#### 4.1.3 State 다이어그램과 시나리오

엘리베이터 시뮬레이션의 기본적인 흐름도는 다음 그림 8의 State 다이어그램과 같다. 이를 이용한 운용 시나리오는 아래의 단계로 나타내었고, 최종 구현화면은 그림 9와 같다.

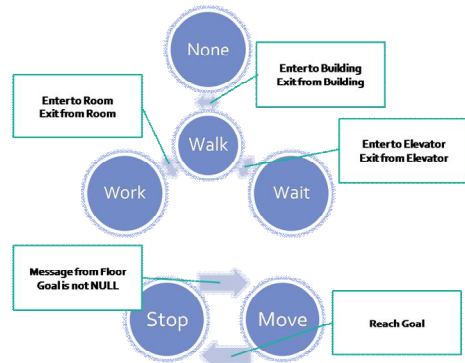


그림 8. State 다이어그램  
Fig 8. State Diagram

- (1) Create Person
- (2) Go to the Elevator
- (3) [ Go up in the Elevator( Go to the Floor that has the Room ) ]
- (4) Go to the Room
- (5) Enter to the Room
- (6) Work in the Room
- (7) Exit from the Room
- (8) Go to the Elevator
- (9) [ Go down in the Elevator( Go to 1st Floor ) ]
- (10) Go out of the Building
- (11) Destroy Person

\* 스텝 (3)과 (9)는 다음 세부단계를 가진다.

- (a) Wait for the Elevator
- (b) Enter to the Elevator
- (c) Wait for Reach
- (d) Exit from the Elevator

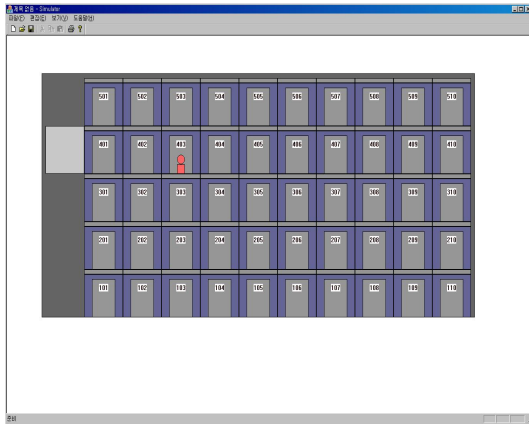


그림 9. 엘리베이터 시뮬레이션  
Fig 9. Elevator Simulation

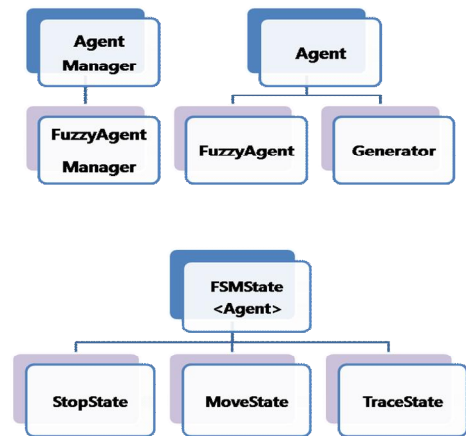


그림 10. 클래스 계층도  
Fig 10. Hierarchy of the Classes

## 4.2 퍼지에이전트 시뮬레이션

퍼지 모듈을 사용한 에이전트 시뮬레이션으로 임의의 위치에 에이전트들이 생성되며, 임의의 위치로 돌아다니게 된다. 운행중 다른 에이전트를 만나면 서로간의 에너지 정보를 퍼지 모듈의 입력 정보로 사용하여 공격과 도망을 결정하게 된다. 계속 공격을 당해 에너지가 모두 소멸되면 에이전트가 소멸되게 된다. 그림 11은 구현화면이다.

### 4.2.1 퍼지 에이전트 클래스

- 1) Generator: 퍼지 에이전트들을 임의의 위치에 생성한다.
- 2) 퍼지 에이전트(Fuzzy Agent): 퍼지모듈을 이용한 에이전트로 임의의 위치로 이동하며 다른 에이전트를 만나면 공격하고 에너지가 모두 소모 되면 소멸된다.
- 3) 퍼지 에이전트 관리자(Fuzzy Agent Manager): 퍼지 에이전트들을 관리 하는 클래스로 이다.

### 4.2.2 퍼지변수

퍼지변수는 다음과 같이 구간 설정을 하였다.

- 1) Energy ( 0 ~ 100 )
  - Empty ( 0 ~ 50 )
  - Normal ( 25 ~ 75 )
  - Full ( 50 ~ 100 )
- 2) Action ( 0 ~ 100 )
  - Defensive ( 0 ~ 50 )
  - Normal ( 25 ~ 75 )
  - Offensive ( 50 ~ 100 )

### 4.2.3 퍼지 규칙(Rule)

- 1) Empty & Empty => Defensive
- 2) Empty & Normal => Defensive
- 3) Empty & Full => Defensive
- 4) Normal & Empty => Offensive
- 5) Normal & Normal => Normal
- 6) Normal & Full => Defensive
- 7) Full & Empty => Offensive
- 8) Full & Normal => Offensive
- 9) Full & Full => Offensive

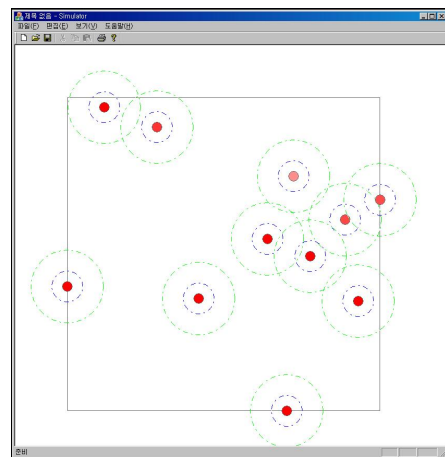


그림 11. 퍼지 에이전트 시뮬레이션  
Fig 11. Fuzzy Agent Simulation

### 4.3 로봇 에이전트 시뮬레이션

휴머노이드 로봇을 시뮬레이션하기 위한 가상의 에이전트를 생성하여 길찾기 시뮬레이션을 구현하였다 (그림 12). 또한 이러한 실험을 위해 다음을 가정하였다.

- 1) 로봇은 거리 센서와 네트워크 장비, 내부 메모리를 가지고 있으며, 그 장비들을 이용하여 길찾기를 하게 된다.
- 2) 로봇은 임의의 시작점에서 길찾기를 시작하며, 주변의 장애물 정보는 알지 못하고 목표점의 좌표만을 알고 이동한다.
- 3) 로봇은 이동 하면서 센서를 통해 주변 정보를 얻게 되고 메모리에 기록하면서, 목표점으로 가는데 가장 적합한 중간 목표점을 설정해 가면서 이동하게 된다.

본 실험은 시스템의 Graph 탐색module 을 이용하였으며, 에이전트의 설정은 다음과 같다.

- 로봇에이전트들이 미지의 공간을 탐색하며 목적지를 찾아가는 시뮬레이션이다.
- 에이전트들을 임의의 위치에 배치시킨다.
- 에이전트들은 센서를 통해 주변 공간을 감지할 수 있고, 감지된 공간은 메모리에 저장된다.
- 에이전트들은 서로간의 공간 정보를 네트워크를 통해 공유할 수 있다.

#### 4.3.1 클래스 정의

로봇 에이전트 시뮬레이션을 위한 클래스들은 위 그림 10과 유사한 계층도로 구성되며, 다음과 같은 클래스들로 구성된다.

- 1) Robot: Sensor, Network module, Memory를 가지고 있으며, 이동할 수 있다.
- 2) Map : 사각형 셀들로 구성되고, 셀은 이동 가능한 것과 불가능 한 것으로 나뉜다.
- 3) Sensor : 직선 방향으로 장애물과의 거리를 알려준다.
- 4) Network Server : Robot의 Network module들과 통신을 한다.

#### 4.3.2 알고리즘

로봇은 이동하면서 센서를 통해 주변 정보를 얻게 되고 메모리에 기록하면서, 목표점으로 가는데 가장 적합한 중간 목표점을 설정해 가면서 이동하게 된다. 알고리즘은 다음과 같다.

WHILE ( Goal is not founded )

```

{
  Move to best cell
  Result = OperateSensor( Position )
  IF ( Result ) THEN
  {
    Update memory map
    Find goal( or best cell )
  }
}

```

#### 4.3.3 최적경로 탐색

에이전트들은 이동하면서 주변 정보를 얻게 되고, 이 정보를 기반으로 최종 목적지로의 최단 거리 경로에 최대한 유사한 경로를 찾아 이동하게 된다. 이때, 현재 가지고 있는 주변 정보 중 최종 목적지로의 최단 거리 경로에 가장 적합한 셀이 최적노드가 된다. 최적노드를 찾는 데에는 여러 가지 알고리즘이 있지만, 여기서는 휴리스틱 값들과 이전 방문 여부를 가지고 계산하였다. 최적노드를 찾기 위한 우선순위 조건은 다음과 순서와 같고, 이러한 순서로 비교하게 된다. 비교값이 같을 경우 다음 조건들도 비교하게 된다.

- 1) Not Visited
- 2) Total Cost(cost from start + cost to goal)
- 3) Cost to goal

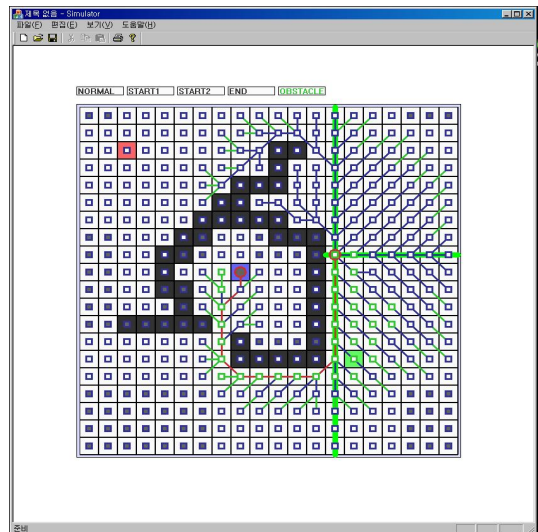


그림 12. 로봇 에이전트 시뮬레이션  
Fig 12. Robot Agent Simulation



#### 4.4 기존연구와 비교분석

본 연구의 시스템은 에이전트 시뮬레이션 기반의 응용 시스템제작을 위한 모듈 제공을 목적으로 하며, 추가적으로 모델러, 뷰어, 스크립터, 분석기 등의 툴을 제공한다. 기본적으로는 모듈 제공이 우선적인 목적이기 때문에, 전체적인 통합 툴이나 자체 언어는 제공되지 않는다. 그리고, 각 모듈들은 독립적으로 구현되며, 추후 조합하여 사용할 수 있게 될 예정이다. 기존 시스템과 다른 기능으로는 3D 렌더링 모듈이나 네트워크 모듈, 인공지능 모듈의 제공 등이다.

〈표 1〉 기존시스템과의 비교분석  
(Table 1) Comparative Studies from Previous Systems

	주요기능
SeSam	범용 시뮬레이션을 위한 통합 개발 툴(IDE)로서, 모델러, 뷰어(2D), 스크립터, 분석기 등을 제공한다.
SimWalk	보행 시뮬레이션을 위한 통합 개발 툴(IDE)로서, CAD 데이터와의 연동이 가능하며, 사-리오 설정 및 분석 기능등이 제공된다.
Ps-i	범용 시뮬레이션을 위한 통합 개발툴(IDE)로서, 지형 생성(2D) 및 모델들에 속성을 설정함으로써 시뮬레이션을 쉽게 제작할 수 있다.
Brahms	범용 시뮬레이션을 위한 통합 개발 툴(IDE)로서, Composer를 통해 시뮬레이션을 제작하고, VM 기반에서 실행된다.
본 연구의 시스템	범용 시뮬레이션 기반의 어플리케이션 개발을 위한 모듈 제공을 목적으로 한 라이브러리이며, 렌더러(2D, 3D), 네트워크 기능, 스크립트, 인공지능기능을 제공한다.

## VI. 결 론

본 연구에서는 시뮬레이션에 관한 모의실험을 보다 원활히 수행하기 위해서 지능형 멀티 에이전트기반의 시뮬레이션 도구를 설계하고 이를 사용하여 몇 가지 실험결과를 기술하였다.

본 연구의 특징은 다음 몇 가지로 설명된다. 우선, 본 연구의 시스템은 인공지능기법을 활용한 에이전트 기반의 시뮬레이션 엔진구축을 우선적인 목표로 하였다. 둘째, 유한 상태기계를 기반으로 지능형 에이전트의 기능을 수행할 수 있게 구성하였다. 이러한 시스템은 모델 설계가 용이하며, 시스템을 구성하고 있는 모든 에이전트들의 상태를 쉽게 확인할 수 있고, 시뮬레이션 설계 모델을 단순화 할 수 있는

장점이 있다. 셋째, 시뮬레이션 엔진의 기능을 보다 확대하여, 스크립트 및 모델링 제작 툴까지 제공함으로써, 사용자에게 시뮬레이션 시스템 개발을 위한 지능적 프레임워크를 제공할 수 있을 것이다.

본 연구에서는 현재까지의 결과를 엘리베이터 시뮬레이션, 퍼지 에이전트 시뮬레이션 그리고 로봇 에이전트 시뮬레이션의 세 가지 영역을 실험대상으로 하여 실험하였으며, 지속적으로 각 모듈의 기능을 확대하여 연구할 예정이다.

## 참고문헌

- [1] 김광년 외 5인 (2004), "DEVS 모델을 사용한 심근 활성화 과정의 시뮬레이션", 한국 시뮬레이션학회 논문지 제13권 제4호, pp1-16.
- [2] 이규원 외 4인 (2005), "비례제어밸브와 혼합 제어를 이용한 혈압 시뮬레이터의 구현", 한국 시뮬레이션학회 춘계학술대회 논문집, pp149-153.
- [3] 유용준, 이장세, 지승도(2004), "SIMVA를 이용한 시뮬레이션 기반의 네트워크 취약성 분석", 한국 시뮬레이션 학회 춘계학술대회 논문집, pp13-19.
- [4] 정정례 외 3인(2003), "사이버 공격 시뮬레이션을 위한 공격자 및 호스트 모델링", 한국 시뮬레이션학회 논문지 제12권 제2호, pp63-73.
- [5] 최상영(2003), "기갑 전투그룹 교전 시뮬레이션 모델", 한국 시뮬레이션학회 논문지 제12권 제1호, pp73-83.
- [6] 강정호 외 7인(2005), "DEVS 기반 모델링을 적용한 잠수함의 어뢰회피 성능 분석 시뮬레이션", 한국 시뮬레이션 학회 논문지 제14권 제2호, pp57-71.
- [7] Ziegler, B.P. and Kim, T.G., and Praehofer, H. (2000), "Theory of Modelling and Simulation".
- [8] Le Bars, M., Attonaty, J.M, Ferrand, N., and Pinson, S., (2005), "An Agent-based Simulation Testing the Impact of Water Allocation on Farmers' Collective Behaviors", Simulation, Vol.81, No.3, pp223-235.
- [9] Cioppa, T. and Lucas, T. (2004), "Military Applications of Agent-based Simulations", Proceedings of the 2004 Winter Simulation Conference, pp994-1000
- [10] Nutaro, J., Zeigler, B.P., "On the Stability

and Performance of Discrete Event Methods for Simulating Continuous Systems”, Submitted to Journal of Computational Physics, available at:  
[http://www.acims.arizona.edu/Pblications/PDF/qss\\_paper.pdf](http://www.acims.arizona.edu/Pblications/PDF/qss_paper.pdf)

[11] Palaniappan, S., and Sarjoughian, H. (2006), “Application of DEVS Framework in Construction Simulation”, Proceedings of the 2006 Simulation Conference,  
<http://www.acims.arizona.edu/PUBLICATIONS/PDF/construction-Sim.pdf>

[12] 김광중 외 3인 (2006), “이동 에이전트의 효율적 이주를 위한 최적경로 탐색”, 한국 컴퓨터정보학회 논문지 제 11권 제3호 p117-224.

[13] 하창승, 정이상 (2007), “내용기반 검색을 이용한 선박매매 정보추출 에이전트의 구현에 관한 연구”, 한국 컴퓨터정보학회 논문지 제12권 제1호, p43-50.

[14] SeSam <http://www.simsesam.de/>

[15] SimWalk <http://www.simwalk.ch/>

[16] Ps-i <http://ps-i.sf.net/>

[17] Brahms <http://www.agentisolutions.com/index.htm>

## 저 자 소 개



### 우 종우

1994~현재: 국민대학교  
전자정보통신대학  
컴퓨터공학부 교수  
관심분야: 인공지능, 지능형에이전트,  
정보보호



### 김 대령

국민대학교 자연과학대학 컴퓨터학부  
학사  
국민대학교 대학원 컴퓨터공학부 대학  
원 석사과정 재학중