

인터랙티브 지식베이스 기반의 계획시스템

전형배[○] 한은지* 엄기현** 조경은***

동국대학교 멀티미디어공학과^{○*}, 동국대학교 게임멀티미디어공학과** ***

hyoungbae.jeon@samsung.com, unzz86@dongguk.edu, {khum, cke}@dongguk.edu

An Interactive Knowledge-based Planning System

Hyoungbae Jeon[○] Eunji Han* Kyhyun Um** Kyungeun Cho***

Dept. of Multimedia Engineering, Dongguk University^{○*}

Dept. of Game&Multimedia Engineering, Dongguk University** ***

요 약

본 논문에서는 가상 에이전트의 행동 계획을 위한 인터랙티브 지식베이스 구축과 인터랙티브 지식베이스를 바탕으로 하는 계획시스템에 관한 방법을 제안한다. 고정적인 지식베이스는 고정적인 계획 수립만 가능하기 때문에 환경의 변화에 잘 대처하지 못한다. 그래서 이 논문에서는 다양한 환경에서 적용이 가능한 인터랙티브한 지식베이스의 구축과 인터랙티브 지식베이스를 활용할 수 있는 인공지능 계획시스템을 제안한다. 본 연구에서 제안한 인터랙티브 지식베이스는 동기, 행동, 사물, 실행의 4가지로 이루어지며 지식베이스의 입력과 지식베이스들 사이의 연관관계는 개발된 자동화 툴을 사용하여 설정한다. 이 툴을 사용하여 사용자는 쉽게 지식베이스에 구성요소들을 추가 또는 수정할 수 있다. 이 지식베이스를 바탕으로 캐릭터는 행동가능한 모든 항목들을 계획을 세우게 되며 이 중 한 가지를 선택하여 행동을 하게 된다. 후에 캐릭터의 환경이 변하게 되더라도 지식베이스의 업데이트를 통해 새로운 행동을 적용시킬 수가 있기 때문에 가상현실 콘텐츠제작자의 입장에서는 상당히 유용하다. 본 논문에서는 확장성이 있는 인터랙티브 지식베이스 구성요소와 구성요소들 사이의 관계설정 그리고 이를 쉽게 입력할 수 있는 툴과 인터랙티브 지식베이스에 적합한 계획시스템의 알고리즘을 제안하여 가상도서관이라는 가상환경에서 실험을 통해 결과를 검증하였다.

ABSTRACT

This paper attempts to investigate the establishment of an interactive knowledge base for action planning by virtual agents and an interactive knowledge-based planning system. A fixed knowledge base is unable to properly handle a change in circumstances because fixed planning is only available under a fixed knowledge base. Therefore, this paper proposes the establishment of an interactive knowledge base which is applicable to diverse environments and an artificial intelligence planning system in which an interactive knowledge base is available. The interactive knowledge base proposed in this paper consists of motivation, behavior, object and action. The association relationship between knowledge base and its input is set using an automation tool. With this tool, a user can easily add to or amend the components of the knowledge base. With this knowledge base, a character plans all action items and chooses one of them to take an action. Since a new action can be applicable by updating the knowledge base even when the character environment changes, it is very useful for virtual reality content developers. This paper has established a relationship between scalable interactive knowledge base components and other components and proposes a convenient input tool and a planning system algorithm effective for an interactive knowledge base. The results of this study have been verified through testing in a virtual environment ('virtual library').

Keyword : interactive knowledge-based planning system, virtual agents, autonomous character

접수일자 : 2009년 06월 03일

심사완료 : 2009년 06월 17일

* 교신저자(Corresponding Author) : 조경은, 주소: 서울시 중구 필동 3가 26 동국대학교(100-715), 전화: 02)2260-3834,

E-mail : cke@dongguk.edu

1. 서론

가상현실 기술의 발달로 사람이 살고 있는 환경과 비슷한 환경을 만들 수 있게 되었다. 또한 사람과 매우 비슷한 에이전트를 만들 수 있게 되었다. 이런 기술들을 바탕으로 사실의 환경을 가상 세계에 구현을 하는 인공지능 시뮬레이션이 매우 발전하고 있고 게임 속 에이전트의 모습도 상당히 많이 발전을 하게 되었다.

가상현실 시뮬레이션과 게임 내에서 공통적으로 중요하게 여겨지는 것은 가상현실 내 존재하는 에이전트가 에이전트에게 주어진 환경에 맞게 자율적으로 행동을 계획해야 한다는 것이다. 게임속의 에이전트가 자율적으로 행동을 해야 게임의 이용자는 지루하지 않을 것이고 가상현실 시뮬레이션은 사실적으로 보여질 것이다.

가상현실의 에이전트가 자율적으로 행동을 하기 위해서는 계획시스템이 필요하다. 계획시스템이란 가상 에이전트가 해야 하는 행동을 계획하고 제어하는 시스템을 말한다. 이 계획시스템으로 인해서 가상현실 내에서 에이전트들은 자율적으로 행동을 할 수 있는 것이다. 예를 들면 가상 에이전트가 목이 마르게 되면 물을 마셔야 할 것이며 졸리다면 침대로 가서 자는 행동을 해야 할 것이다. 이 같은 행동들을 계획시스템에서 계획을 세워주는 것이다. 계획시스템내의 계획지식베이스는 에이전트들이 할 수 있는 행동들을 정리를 한 데이터베이스이다. 계획시스템은 에이전트의 두뇌에 해당하는 지식베이스를 바탕으로 가상에이전트의 계획을 세우게 되는 것이다.

계획 시스템을 구축함에 있어서 먼저 지식베이스를 구축하는 것은 상당히 중요한 부분을 차지한다. 지식베이스를 어떻게 구축하느냐에 따라서 계획시스템이 많은 영향을 받기 때문이다. 하지만 지금까지 연구된 계획시스템들은 지식베이스가 시뮬레이션하는 가상 환경에 따라 고정되어 있었기 때문에 다양한 환경에 계획시스템을 적용시키기 어려우며, 가상 에이전트에게 원하는 계획을 세우기 위

해서는 매번 지식베이스 전체를 수정해야 하는 불편함이 있다. 즉, 지금까지 연구되어진 계획시스템으로는 확장성이 부족하다.

따라서 본 논문에서는 가상환경의 변화에도 잘 대처를 할 수 있고 각각 다른 환경에서 적용되어질 수 있는 인터랙티브한 지식베이스를 제안하며 동시에 인터랙티브한 지식베이스에 적합한 계획시스템을 제안한다.

논문의 구성은 다음과 같다. 2장에서는 지금까지 연구 개발되어진 계획시스템에 대한 관련 연구들을 기술하며 3장에서는 계획시스템의 기본개념을 설명한다. 4장에서는 인터랙티브 지식베이스에 대한 구성과 관계설정에 대해 기술한다. 5장에서는 본 연구에서 제안하는 인터랙티브 지식베이스 계획시스템의 설계에 대해 설명하며 6장에서는 실험환경 및 구현결과를 보이며 마지막으로 7장에서는 결론을 맺는다.

2. 관련 연구

1970년대 초반부터 시작된 인공지능의 계획시스템에 관한 연구는 지능형 에이전트의 이론과 구조에 가장 밀접히 연관된 분야이다. 일반적으로 계획시스템은 에이전트가 주어진 목표(goal)를 달성할 수 있는 일련의 구체적인 행동(action)들과 그들의 실행 순서(order)를 자동으로 구해준다[1].

국내의 경우를 먼저 살펴보면 김인철은 계획시스템(planning system) 또는 계획기(planner)를 핵심 모듈로 하는 계획 기반 에이전트 시스템을 소개하고 있으며 계획 기반 에이전트 시스템을 반응성과 사회적 속성을 중심으로 분류하고 특징을 간략하게 소개하고 있다[1].

또한 계획시스템은 국가 재난발생시 즉각적인 계획 발생을 위한 분야에서도 이용되기도 한다. 박인숙, 이태경은 재난 관리 시스템에 환경변화를 스스로 인지하고 그에 맞는 행동을 취하며 지식을 바탕으로 추론과 학습 기능이 있는 에이전트를 도입하였다[2]. 즉 재난 발생시에는 인지요소에서 새

로운 계획을 생성하여 수행하는 것이 아니라 저장되어 있는 계획들에서 발생한 재난에 알맞은 계획을 추론을 통해 생성하는 것이다.

환경기반의 시스템에도 계획 시스템이 응용되어 졌는데, 민현정, 김경중, 조성배가 제안한 환경기반의 시스템은 환경의 자극에 즉각적으로 반응하여 불확실한 상황에도 효과적으로 대응할 수 있다고 한다. 하지만 복잡한 문제에 적용하기에는 부적합하고 초기화 시에 수동으로 설계해야 하는 제약이 있었기 때문에 행동들을 연결하여 생성할 수 있는 행동 네트워크를 연구하였다[3]. 행동 네트워크상에서 행동들은 기본적인 행동들로 고유의 활성화값을 가지고 있고 활성화값이 가장 큰 행동을 다음 행동으로 선택한다.

에이전트의 이동에서도 계획시스템이 사용되었다. 대부분의 기존 이동 에이전트 시스템들은 고정 스크립트에 기술된 대로 행동을 하기 때문에 스스로 자율적으로 최종 목표와 동적 상황에 맞는 계획을 수립하고 실행해나가는 것이 불가능하다. 따라서 석황희, 김인철은 제한점들을 극복하기 위해 반응형 계획 에이전트 구조인 JAM을 기초로 에이전트의 이동 모델과 에이전트간의 통신 모델을 설계하였다[4].

다음 국외의 경우를 살펴본다. T. Conde와 D. Thalmann은 점차적으로 가상 환경을 인식하여 자동적으로 학습과 행동을 할 수 있고 실시간으로 변하는 환경에 대처할 수 있는 계획 시스템을 개발하였다[5]. 처음에는 자율적인 가상 에이전트 학습을 위해 Q-learning을 사용하였다. Q-learning은 고정된 결과를 얻기 위해서는 Q-factor Table의 정확성이 높아야한 반면 일정한 수치가 없다면 Q-learning은 수행하기가 어렵다. 또한 애니메이터가 모든 상태와 행동을 처리해야하는 불편함으로 수행시간과 성능에 대한 문제점이 발생하게 되었다. 그래서 강화학습 기술을 이용해서 에이전트의 행동을 선택하는 새로운 기술을 개발하게 되었다. 이 방법은 강화학습을 이용해서 에이전트의 행동을 선택하게 하는 것이다. 강화학습이란 시행착오와

그로 인해 발생하는 피드백을 통해 정책을 조절해 나가는 학습 방식으로 특정한 액션에 대한 보상이 주어지게 되어 게임 속 NPC가 시행착오를 통해 최고의 전술을 학습하는데 이용된다.

또한 T. Conde와 D. Thalmann은 AVA (Autonomous Virtual Agent) Learning을 제안하였다[6]. 지금까지의 연구는 학습 능력이 없고, 행동에 제한이 있었으며 인식은 하지 못하고 오직 행동만 수행한다는 문제점이 있었는데 이 논문에서는 강화 학습에 기반을 둔 새로운 방법을 제안하고 있다.

T. Abaci, J. Ciger 그리고 D. Thalmann은 에이전트와 가상 오브젝트의 상호 작용을 위해 'Smart Object'의 개념을 도입하였다[7]. 하지만 Smart Object는 다른 대안들보다는 유동적이지만, 상호작용하는 계획들이 고정되어 있어 엄격한 규제가 요구되며 새로운 상황에 적용시킬 수 있는 가능성과 더 복잡하고 동적인 문제를 해결할 수 있는 능력이 저하된다는 단점이 있다.

마지막으로 E. de Sevin와 D. Thalmann은 가상 에이전트가 계획을 세우고 행동을 하기 위해 동기를 부여하고 계획을 실행하는 방법을 제안했다[8]. 이 계획 시스템은 환경(Environment)으로부터 반응 (Reactive)이 발생하게 되면 목표 지향적인 행동(Goal-Oriented Behaviors)을 발생시켜 실행을 하게 되는 시스템이다.

현재까지의 국내와 국외의 다양한 논문과 연구 결과를 살펴보면 고정적인 지식베이스를 두었다는 점에서 문제점이 생길 수 있다. 고정적인 지식베이스는 계획시스템을 시뮬레이션하는 환경에 최적화시켰기 때문에 다른 환경에서는 다르게 설정해야 한다는 확장성에서 문제점이 발생할 수 있다. 최근에는 다양한 곳에서 계획 시스템이 사용되고 있으며 사용자들마다 테스트하는 가상환경이 다양하므로 사용자들이 원하는 지식베이스를 만들기 위해서는 인터랙티브한 지식베이스가 필요한 것이다.

계획 시스템에서 고정적인 지식베이스의 사용은 사용자가 구성하기 원하는 가상현실을 만들지 못할 뿐만 아니라 변하는 가상현실에 제대로 대처하기가

어렵다. 또한, 지식베이스를 구축한 개발자만 이용하기 쉽게 설계되어 있어 다른 사람들이 이용하기가 어렵다. 따라서 인터랙티브한 지식베이스를 만드는 것이 중요하다.

또한 개발자가 아닌 일반 사용자들이 지식베이스의 입력과 연관관계 설정을 쉽게 사용하기 위해서는 지식베이스 입력을 자동화하는 툴이 필요하다. 이 자동화툴을 사용하여 사용자는 쉽게 지식베이스에 사물을 추가할 수 있으며 추가가 되는 동시에 관련된 지식베이스들도 업데이트가 된다. 이런 방법으로 지식베이스에 데이터들을 추가하게 된다면 가상환경에 다른 사물들이 추가가 되거나 삭제가 된다하더라도 지식베이스를 쉽게 업데이트시킬 수 있을 것이다. 이는 지식베이스를 처음에 모두 정의를 하지 않아도 지속적으로 확장하면서 계획 시스템을 사용할 수 있다는 것을 의미한다.

본 연구에서 제안하는 인터랙티브 지식베이스와 거기에 기반을 둔 계획시스템의 설계 및 구현은 가상환경이 변하게 되더라도 변화되는 환경을 인터랙티브 지식베이스에 의해 쉽게 확장시켜 새로운 계획을 가상환경 시뮬레이션에 적용시킬 수 있도록 하기 위함이다. 사용자는 사용자가 원하는 가상현실의 지식베이스를 자동화툴을 사용하여 구성할 수 있을 것이며 그에 따라서 가상에이전트의 행동 계획 또한 자동적으로 적용된다.

3. 계획 시스템의 기본개념

계획 시스템은 가상 에이전트의 움직임을 계획하고 행동하는 계획 알고리즘과 가상 에이전트의 행동을 계획하는데 쓰이는 지식베이스로 크게 구분할 수 있다.

지식베이스는 에이전트가 행동 가능한 액션들로 계획을 세울 수 있게 해준다. 그래서 지식베이스가 변경되면 그것으로 계획을 세우는 에이전트의 행동에도 변화가 있게 된다. 이전의 고정적인 지식베이스들은 가상환경의 변화에 잘 대처하지 못하여 인터랙티브 지식베이스를 구축할 필요가 있다. 인

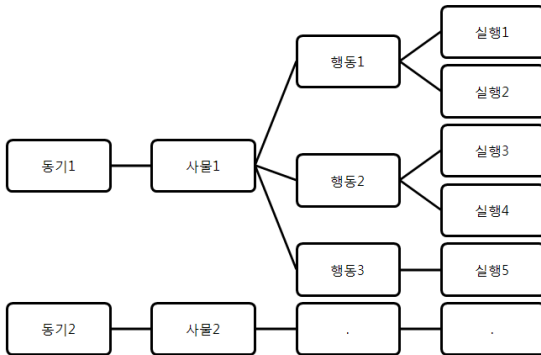
터랙티브 지식베이스를 통해 사용자가 구축하려고 하는 가상현실의 변화가 생겨 새로운 동기와 행동들을 입력하게 되더라도 그것을 즉시 시뮬레이션 환경에서 테스트를 해볼 수 있게 된다. 그러나 이 같은 입력을 프로그램 코드상으로 해야 한다면 사용자가 프로그래밍 코드를 이해를 해야 한다는 불편함이 야기된다. 이것을 해결하기 위해 인터랙티브 지식베이스 입력툴이 필요하다. 이 툴을 사용하여 사용자는 쉽게 데이터들을 입력 수정할 수 있게 된다.

계획 알고리즘은 지식베이스를 바탕으로 에이전트가 직접 행동할 수 있는 액션들의 목록을 만들게 한다. 이 알고리즘 또한 인터랙티브 지식베이스에 적합한 구조로 만들어야 한다. 인터랙티브 지식베이스의 구조상 가상환경에 들어가는 사물들이 존재하게 된다. 각 사물들마다 동기와 연결시키는 구조가 필요하다. 예를 들면 빵이라는 사물을 가지고 ‘먹고 싶다.’라는 동기를 수행할 수 있게 되는 것이고 음료수라는 사물을 가지고 ‘마시고 싶다.’라는 동기를 수행할 수 있는 것처럼 사물마다 동기를 두는 인터랙티브 지식베이스 구조를 사용한다.

이를 바탕으로 알고리즘의 구성은 크게 3단계로 나누어지게 된다. 지식베이스를 읽어오는 단계, 이를 저장하는 단계, 에이전트가 실행하는 단계로 나누어진다. 각각의 알고리즘에 대한 설명은 다음 장에서 자세하게 한다.

4. 인터랙티브 지식베이스

앞장에서 살펴보았던 것과 같이 고정적인 지식베이스는 한 가지 특정한 상황을 배경으로 하였기 때문에 다양한 가상환경을 표현하기에는 부족한 점이 있다. 하지만 인터랙티브 지식베이스는 4가지의 서로 다른 역할을 하게 되는 구성요소를 두어 다른 가상 환경에서도 적용시킬 수 있게 하였다.



[그림 1] 인터랙티브 지식베이스의 관계

[그림 1]은 인터랙티브 지식베이스의 구성을 보여주는 것이다. 동기는 사물과 결합하고, 사물은 행동과, 행동은 실행과 결합을 하게 된다. 이런 결합을 가지게 되는 이유는 다음과 같다. 가상현실에는 여러 가지 사물들이 존재를 하게 된다. 가상 에이전트가 행동을 하게 될 때는 가상 환경에 존재하는 사물들을 가지고 행동을 하게 되는 이론을 기반으로 계획시스템을 설계하였다. 에이전트가 행동하는 욕구들은 사물과 많은 관련이 있을 것이며, 행동하고 싶은 사물이 결정이 되어 지고 나면 이를 가지고 '걸어간다. '먹는다.', '본다.' 등의 행동을 하게 된다. 하나의 사물로 할 수 있는 행동의 연결은 아래와 같은 방법으로 이루어질 것이다.

책장으로 간다. > 책을 찾는다. > 책상으로 간다. > 의자에 앉는다. > 책을 읽는다.

이와 같이 사물들은 행동으로 연결이 될 것이고 이는 다시 행동 가능한 실행들로 연결이 되어 계획시스템에서 이용된다.

4.1 인터랙티브 지식베이스의 구성

인터랙티브 지식베이스는 동기, 사물, 행동, 실행으로 4가지로 구성이 된다. 동기 안에는 상황이 존재를 하게 되며 동기의 상황은 사용자가 Tool 내에서 설정하여 줌으로써 후에 동기와 상황, 두 조

건에 상응하는 사물을 선택하는데 영향을 준다. 또한 사용자는 동기에 따른 사물을 추가하고 그 사물에 따른 실행을 설정할 수 있고 그에 따른 행동들과 각 행동의 작은 단위인 실행들은 사용자의 입력을 통하여 지식베이스에 추가된다. 즉, 인터랙티브 지식베이스는 사용자가 새로운 사물을 추가, 삭제하는 작업 등을 편리하고 간편하게 할 수 있다는 특징이 있다.

에이전트의 동기는 내부적인 동기와 사물로부터 또는 환경의 영향으로 인해 발생하는 외부적인 동기로 나뉘게 된다. 예를 들어 내부 동기로는 목마름에 의한 갈증(thirsty), 배고픔(hungry), 손이 더러울 경우 씻고 싶어 하는 청결함(clean), 화장실에 용무가 급할 경우에는 화장실(toilet), 체력이 부족해서 휴식이 필요한 경우 휴식(rest) 등이 내부적으로 발생될 수 있는 내부 동기가 된다.

외부 동기로는 놀이(play), 대화(talk), 빌리기(borrow)등과 같이 컴퓨터나 TV, 책 등과 같이 사물로 무엇을 할 수 있는 동기나 환경에 의해 생겨나는 동기들이 외부 동기가 된다. 동기는 사물을 입력하기 전에 먼저 설정을 해주어야 하는 부분이며, 사용자가 Tool로 설정할 수 있도록 한다. 동기를 정리하면 표1과 같이 요약할 수 있을 것이다.

동기의 상황은 사용자가 Tool을 사용하여 입력할 수 있다. 입력된 상황은 가상 환경 내에서의 장소 또는 사물과의 거리 등을 의미한다. 또한 상황은 동기에 따라 선택할 수 있는 사물들 중 가장 적합한 사물을 선택하는데 영향을 끼친다. 하나의 동기는 다양한 상황들을 반영할 수 있도록 동기와 상황간의 관계를 사용자가 설정한다. 그리고 지식 표현 방법은 동사(명사)로 표기한다.

[표 1] 동기의 분류

Index	동기분류	설명	동기
M001	내부	갈증	thirsty
M002	내부	배고픔	hungry
M003	내부	청결함	clean
M004	내부	화장실	toilet
M005	내부	휴식	rest
M006	외부	놀이	play
M007	외부	대화	talk
M008	외부	반납	return
M009	외부	빌리기	borrow
M010	외부	읽기	read
M011	외부	호기심	curious

[표 2] 상황의 분류

no.	설명	상황
1	열람실에 있다	locate(bookroom)
2	독서실에 있다	locate(readingroom)
3	화장실에 있다	locate(toilet)
4	컴퓨터 근처에 있다	near(computer)
5	게임기 근처에 있다	near(game)
6	TV 근처에 있다	near(tv)
7	대출기 근처에 있다	near(lendmachine)
8	음료수-자판기 근처에 있다	near(vendingmachine1)
9	과자-자판기 근처에 있다	near(vendingmachine2)
⋮		

사물은 사용자에게 의해서 필요한 사물이 입력되어 추가 또는 수정할 수 있고 필요치 않은 오브젝트는 삭제할 수 있다. 기존 Thalmann 연구실의 시스템에서는 Goal이라는 이름을 사용하였지만 본 논문에서는 사물을 자유롭게 추가, 수정, 삭제를 할 수 있기 때문에 'Goal' 보다는 '사물'이라는 용어를 사용한다[7]. 사물은 환경내에 존재하는 사물 또는 장소를 의미한다고 할 수 있다.

실행은 행동과는 다른 의미로 애니메이션 행동의 작은 단위를 의미한다. 예를 들어 'oo를 가다'라는 행동에 따른 실행은 walk가 된다. 실행은 사물

을 추가할 때 사물로 할 수 있는 실행을 입력하는 단계와 행동들을 설정할 때 각각의 행동에 맞는 각 실행을 입력하는 단계가 있다. 실행은 애니메이션 행동의 단위와 같기 때문에 예를 들어 주위를 둘러보는 것과 게시판을 보는 것은 모두 실행 'see'이므로 이를 구분하기 위해서 'see1', 'see2'로 구분하여 표기한다.

[표 3] 사물의 분류

Index	설명	사물
O001	소설책	novel
O002	전공책	major book
O003	컴퓨터	computer
O004	게임기	game
O005	TV	TV
O006	음료수-자판기	vending machine1
O007	과자-자판기	vending machine2
O008	열람실-책장	bookcase
O009	대출기	lend machine
⋮		

[표 4] 실행의 분류

Index	실행	액션	설명
A001	stand	서기	서있는 동작
A002	walk	걷다	걸기
A003	touch	갸다 대다	대출기에 카드 찍고 책 가져다 대기
A004	put1	넣다	서재 책 넣기
A005	draw1	꺼내다	서재 책 꺼내기
A006	sit	앉다	의자에 앉기(의자 공통)
A007	stamp	찍다	출입구 카드 찍기
A008	throw	버리다	휴지통에 쓰레기 버리기
A009	see1	보다	주위를 둘러보기
A010	see2	보다	서재, 게시판 보기(공통)
A011	turnon1	켜다	TV 켜기
A012	turnoff1	끄다	TV 끄기
A013	read	읽다	책 읽기
A014	borrow	대출하다	대여실에 대출하기
A015	return	반납하다	대여실에 반납하기
⋮			

행동은 추가된 사물로 가능한 행동들이다. 사용자가 입력한 사물로 실행을 수행할 때 가능한 모든 행동들을 나타낸다. 즉, 에이전트가 책을 읽는다(read).는 실행을 하게 되면 그에 따른 행동들은 먼저 책장에 가서 책을 가지고 그것을 읽는 것으로 표현을 할 수가 있게 되는데 이것을 표현하는 방법으로는 다음과 같은 방법을 쓴다. 어디로 가는 것은 go, 무엇을 이용하는 것은 use와 같이 동사를 사용하며 동사(명사)로 표현한다.

[표 5] 행동의 분류

Index	설명	행동
B001	출입기를 찾으러 가다	go(entrance)
B002	출입기에서 학생증카드를 찍다	stamp(entrance)
B003	출입기를 들어가다	enter(entrance)
B004	컴퓨터를 찾으러 가다	go(computer)
B005	컴퓨터 의자에 앉다	sit(comchair)
B006	컴퓨터 모니터를 키다	turnon(monitor)
B007	컴퓨터 모니터를 끄다	turnoff(monitor)
B008	컴퓨터 마우스를 잡다	pickup(mouse)
B009	컴퓨터를 이용하여 책을 찾다	use(computer)
B010	컴퓨터 의자에서 일어나다	standup(comchair)
B011	열람실 서재를 찾으러 가다	go(bookcase)
B012	서재에서 책을 꺼내다	pickup(book)
B013	대출기를 찾으러 가다	go(lendmachine)
B014	대출기에 학생증카드를 찍다	stamp(lendmachine)
B015	대출기에 책을 가져다 대다	stamp(book)
B016	독서실 책상을 찾으러 가다	go(readdesk)
B017	독서실 의자에 앉다	sit(readchair)
B018	독서실에서 책을 읽다	read(book)
B019	휴게실을 찾으러 가다	go(restroom)
B020	휴게실의 자판기를 찾으러 가다	go(vendingmachine1)
B021	자판기에 돈을 넣다	push(money)
B022	자판기에서 메뉴 선택 한다	push(button)
	:	

4.2 인터랙티브 지식베이스의 구성관계

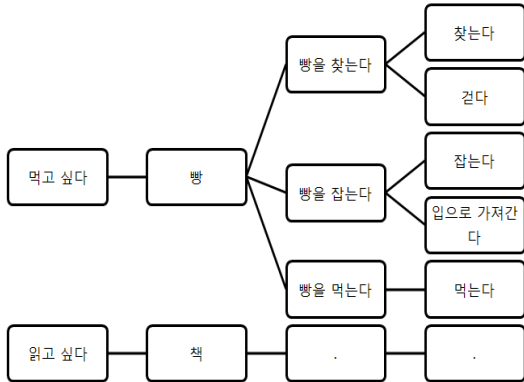
4장의 도입부에서 인터랙티브 지식베이스 구성요소들 사이의 관계에 대해서 살펴보았다. 본 논문에서의 지식베이스는 고정적인 지식베이스와는 다르게 가상환경의 변화에 따라 확장 가능한 인터랙티브 지식베이스이다. 이는 사용자가 직접 사물을 추가할 수 있다는 것을 말한다. 왜냐하면 사물의

추가는 가상환경의 변화를 의미하기 때문이다. 사물의 입력과 더불어 이와 관련이 있는 동기, 행동, 실행을 입력하여 지식베이스를 업데이트시킨다.

먼저 동기와 사물들의 관계를 살펴보자. 사물들은 하나의 동기를 가지는 것이 아니라 다양한 동기들을 가지게 된다. 예를 들어 물이라는 사물은 ‘목이 마르다.’ 라는 동기와 동시에 물을 가지고 요리를 할 수도 있기 때문에 ‘요리를 한다.’ 라는 동기를 가지게 될 것이다. 자동차라는 사물은 ‘드라이브를 한다.’라는 동기와 ‘정비를 한다.’라는 동기를 가질 것이다. 이처럼 한가지의 사물은 여러 가지 동기들을 가지기 때문에 사물의 기능이 추가가 되더라도 그 변화에 잘 대처할 수가 있을 것이다.

다음으로 사물과 행동, 실행의 관계를 살펴보자. 계획 시스템에서 동기가 결정이 되고 사물을 선택을 하고 나면 다음의 할 일은 바로 행동을 선택하여 실행을 해야하는 것이다. 배가 고파서 빵이라는 사물을 선택을 하게 되었다면 빵을 찾아야 할 것이고 빵을 가지러 가야 할 것이며 그 빵을 먹는 행동을 하여야 할 것이다. 이와 같이 행동할 수 있는 것이 행동이며 선택되어지는 사물과 관계가 되어있다. 행동단계들이 선택이 된 후에는 직접적인 실행이 가능한 실행들이 선택이 되어져야 한다.

그래서 행동들은 실행들과 관계가 되어 있다. 이는 실행의 집합이 행동이 된다는 것을 의미한다. 실행이라는 하위 항목을 두게 된 배경은 ‘빵을 먹는다.’라는 행동을 선택을 하더라도 그 안에는 더 많은 행동들이 존재를 하게 된다. ‘빵을 잡는다.’, ‘빵을 입으로 가져간다.’, ‘빵을 입에 넣는다.’, ‘빵을 먹는다.’등으로 세분화 할 수 있다. 이를 모두 행동으로 나타내게 된다면 행동의 효율이 상당히 낮고 그 관리 또한 매우 어려워질 것이다. 따라서 하위 항목인 실행을 두어 공통된 실행들의 연속을 묶어 행동으로 하여 인터랙티브 지식베이스의 효율을 높였다. [그림 2]는 인터랙티브 지식베이스의 구성관계를 예로 나타낸 것이다. 각각의 사물들의 선택은 상황에 맞게 선택된다. 사물에는 행동들이 연결되어 있고 행동에는 실행들이 연결되어 있다.

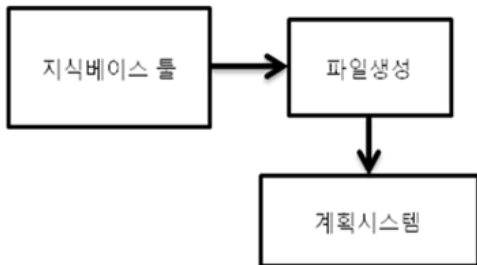


[그림 2] 구성관계의 예

5. 계획시스템의 설계

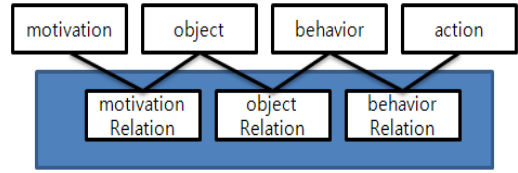
5.1 인터랙티브 지식베이스의 적재

지식베이스 자동화 입력 틀을 이용하여 지식베이스 라이브러리를 파일로 생성을 한 후에 그 지식 베이스 라이브러리를 계획 시스템에서 이용하게 된다. 지식베이스는 동기, 사물, 행동, 실행 총 4개의 요소로 이루어져있는데 이들의 데이터는 사용자가 지식베이스 틀을 이용하여 직접 입력을 할 수가 있으며 또한 이들 사이의 관계들에 대해서도 입력이 가능하다.



[그림 3] 개략적인 흐름

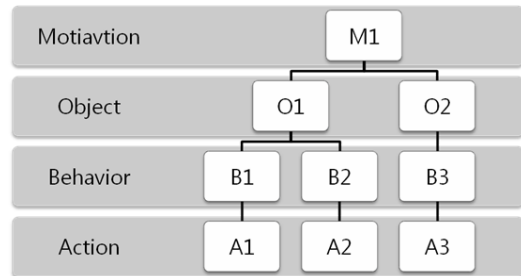
모든 입력이 끝나게 되면 그림 4와 같은 연관관계가 생성이 되며 지식베이스 라이브러리가 생성된다. 계획 시스템에서는 이 지식베이스 라이브러리를 로드한 뒤 계획 시스템에서 사용하게 된다.



[그림 4] 지식베이스 내부 구조

5.2 행동 가능한 계획의 설계

생성되는 지식베이스들의 관계는 그림 5처럼 이루어진다. 하나의 동기에 여러 사물들이 결합이 되고 여기에 여러 행동들이 결합되어 이루어지는 형태이다. 이중 직접 실행 가능한 실행들을 메모리에 저장을 하여 계획시스템에서 사용을 하게 되는 것이다.



[그림 5] 지식베이스의 관계

KnowledgebaseLoad() 함수를 호출하면 파일로 생성되어 있는 지식베이스를 메모리로 저장을 하는데 추후에 파일의 변경이 일어나게 되면 다시 위 함수를 호출하여 새로운 내용을 메모리에 적재한다. 이런 점에서 다른 고정적인 지식베이스와 다르게 인터랙티브하다고 할 수 있다. 행동 계획들을 저장할 때에는 Vector Array를 이중으로 구성해서 사용하였다. 아래는 지식베이스의 적재와 계획의 설계가 같이 들어가 있는 알고리즘의 슈도코드를 나타낸 것이다.


```

1. void KnowledgeBaseLoad()
2. {
3.   variable initialization;
4.   file open(filename);
5.   if(file open != null)
6.   {
7.     for(;;)
8.     {
9.       if (fgets()==NULL)
10.    {
11.      break;
12.    }
13.    for (end of the sentence)
14.    {
15.      save motivation
16.      find ("!");
17.      break;
18.    }
19.    find object(save variable, other file);
20.    find behavior(save variable, other file);
21.    find action(save variable, other file);
22.  }
23.  file close();
24. }
25. }

```

03번에서 프로그램 수행에 관한 변수들을 선언하고 초기화를 시켜준다. 04번에서 12번까지는 지식베이스 파일을 읽어온다. 읽어온 정보에서 동기정보를 따로 찾아 저장할 시키게 되고 사물을 연결시켜주는 "!"를 찾아서 그 위치를 저장하고 그 뒤에 있는 연결되어 있는 사물 번호를 저장하게 된다. 19번에서 저장된 사물 번호를 가지고 사물파일에서 검색을 한 다음 Vector 배열에 저장을 하게 된다. 사물파일과 연결되어있는 행동 번호를 찾게 되고 20번에서 그 정보를 바탕으로 다시 정보를 검색하게 되는 방식이다. 실행까지의 저장이 끝나게 되면 23번에서 파일을 close시키고 끝나게 된다.

5.3 계획의 실행

다음은 계획의 실행방법을 설명한다. 그림 6은 READ라는 읽고 싶다는 동기에 연결되는 지식베이스의 예를 나타낸 것이다. Major Book이라는 사물이 하나 연결되어 있고 거기에 여러 행동들이 결합되어 있는 것이다.



[그림 6] 지식베이스의 예

따라서 에이전트가 전공 책을 읽고 싶다는 행동을 하게 된다면 제일 왼쪽에 있는 걸기, 둘러보기, 서재 둘러보기 순으로 진행하여 마지막으로 책읽기라는 행동으로 끝을 낼 것이다. 위와 같은 행동들을 에이전트가 수행하기 위해서는 runaction()이라는 함수를 호출하게 된다. 이 함수는 수행하고 싶은 동기와 상황 정보를 받아 직접 실행 가능한 실행을 수행하게 된다. 이때 미리 저장되어 있는 지식베이스를 검색하여 그 정보를 찾아내어 실행하게 된다. 위 예에서 보면 가상 에이전트가 서재를 둘러보는 행동을 매번 실행하지 못하는 경우가 발생할 수도 있을 것이다. 이것을 대비해 에이전트가 실행을 수행할 때에는 Array를 제어 할 수 있는 iterator를 두어 에이전트가 행동할 때 발생할 수 있는 인터럽트를 대처할 수 있게 만들었다. 아래 슈도코드는 계획의 실행에 관한 것이다.

```

1. void frameMove()
2. {
3.   if(temp != 목록의 끝)
4.   {
5.     string temp;
6.     char* tmptoken;
7.     tmptoken = (char*)temp.c_str();
8.     SetCurrentAnimation(tmptoken);
9.   }
10. }

```

현재 읽어온 실행의 내용을 진행 중인 에이전트의 애니메이션으로 변경시켜주게 되면 에이전트의 움직임이 변하게 되는 것이다.

6. 실험환경 및 구현

계획시스템을 사용하기 이전에 먼저 지식베이스의 설정을 해야 한다. 지식베이스의 설정은 인터랙티브 지식베이스를 구현함에 있어 가장 중요한 부분이다. 인터랙티브 지식베이스의 구성관계에서 설명했듯이 하나의 사물에 여러 동기, 여러 행동들이 설정이 될 수 있기 때문에 이것을 지정해 주는 것이 중요하다. 하지만 이런 관계들을 코드로 구현을 하게 된다면 사용자의 입장에서는 지식베이스를 구축하기가 어렵게 되고 또한 이해하기가 복잡해진다. 따라서 이것을 위한 지식베이스 자동화 툴을 구현하였다.

[그림 7]은 자동화 툴의 동기를 입력하는 입력창이다. 여기서 사용자들은 지식베이스에 추가하는 동기를 입력하게 된다. 나머지 동기, 사물 데이터들도 각각의 입력창을 가지게 된다.



[그림 7] 동기 데이터의 추가



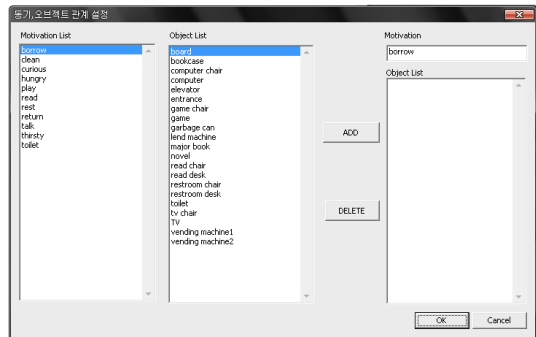
[그림 8] 사물 데이터의 추가



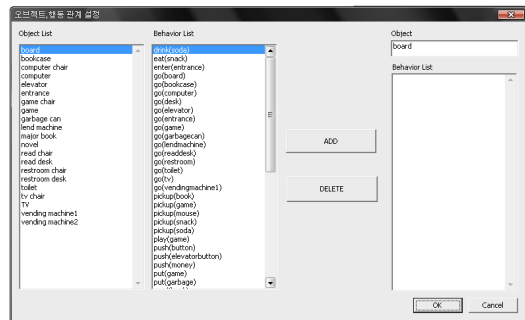
[그림 9] 행동 데이터의 추가

[그림 10]은 동기, 사물 관계설정의 화면이다. 이 메뉴는 동기와 사물의 관계설정에서 하나의 사물에 다양한 동기들을 설정하는 역할을 하게 된다. ADD버튼을 누르게 되면 현재 설정된 동기과 관련된 사물들이 입력이 된다. 반대로 DELETE를 눌러 설정한 부분들을 수정할 수도 있다. 예를 들어

“배가 고프다.”라는 동기가 존재를 하게 된다면 사과, 빵, 밥 등 먹을 수 있는 사물들이 존재를 하게 될 것이다. 위 메뉴는 이와 같은 설정을 할 수 있게 만들어준다. 다른 메뉴들의 인터페이스들도 위와 비슷하며 위와 같은 방법으로 관계들을 설정해 주게 된다. [그림 11]은 사물, 행동의 관계 설정 화면이다.



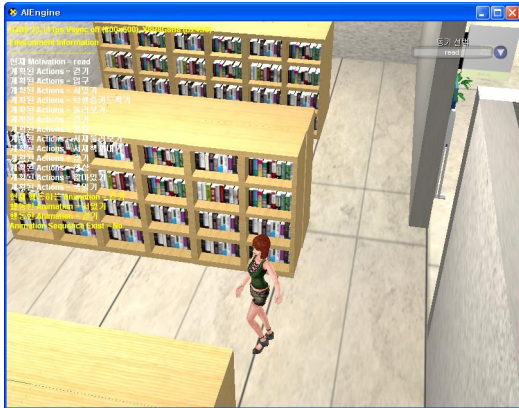
[그림 10]동기, 사물의 관계설정 메뉴화면



[그림 11] 사물, 행동의 관계설정 메뉴화면

지식베이스 자동화 입력 툴을 이용하여 지식베이스를 구축한 다음 도서관을 배경으로 에이전트가 행동을 하는 시뮬레이션을 만들어 보았다. [그림 12]에서 에이전트는 현재 책을 읽고 싶다는 동기를 가지고 책장으로 향하는 중이다. 책장으로 향하기 전에 입구에서 출입기에 학생증을 찍고 안으로 들어가게 된다. 안으로 들어간 다음 에이전트는 책장으로 향해서 책을 집어든 다음 책상으로 가서 책을 읽게 된다. 그 이후 ‘인터넷을 하고 싶다.’, ‘TV

를 보고 싶다.’의 동기를 수행한다. 이후 다른 에이전트와 대화를 끝낸 다음 다시 책상으로 돌아가서 책을 읽는 시뮬레이션이 수행되는 장면이다.



[그림 12] 책장으로 향하는 에이전트



[그림 13] 전체적인 시뮬레이션

[그림 13]에서는 전체적인 시뮬레이션의 모습을 보여주고 있다. 오른쪽 상단에는 현재 인터랙티브 지식베이스에 저장되어 있는 동기들의 목록들이 보여 지게 된다. 사용자가 지식베이스에 동기가 추가 되면 콤보박스에 새로운 동기가 추가되며 그에 따라 할 수 있는 행동들도 같이 추가되는 것이다. 그리고 왼쪽에는 현재의 동기와 계획된 실행 그리고 현재 진행 중인 실행들의 목록이 나오게 되는데 이를 통해 현재 가상 에이전트가 무엇을 하고 있는지 알 수 있다.

7. 결 론

지금까지 인터랙티브 지식베이스의 구성과 구성들 사이의 관계들을 설정하여 지식베이스를 구축하는 것과 이를 이용하는 계획 시스템에 대해서 알아보았다. 이것을 통해 기존의 고정적이었던 지식베이스의 단점을 보완하여, 가상현실 세계의 가변 환경에 맞는 지식베이스를 직접 구성을 하고 추가하고 싶은 사물들이 생기게 되면 언제든지 수정 입력 가능하며, 그에 맞게 가상 캐릭터가 계획을 세울 수 있는 인터랙티브 지식베이스 및 계획 기법을 제안하였다.

이는 사용자가 직접 원하는 지식베이스를 만들 수 있다는 점에서 인터랙티브 지식베이스를 구축한다는 점에서 의미가 있다. 이를 이용하여 도서관에서 간단한 행동들을 수행할 수 있는 시뮬레이션을 만들어보았다. 지금의 시뮬레이션은 사용자가 입력한 동기들 중에서 선택한 동기에 의해서만 에이전트가 행동을 하게 된다. 따라서 이후에는 더 나아가 내부 동기 시스템과 성격시스템 등을 도입한 에이전트 연구가 더 이루어져야 할 것이다.

참고 문헌

- [1] 김인철, “계획 기반 에이전트 시스템”, 한국 정보 처리학회 학회지 특집, pp.13-26, 1997
- [2] 박인숙, 이태경, “상황 기반 계획 생성 에이전트 설계”, 한국 정보 처리학회 추계 학술 발표논문집 제5권 제2호, pp.591-594, 1998
- [3] 민현정, 김경중, 조성배, “에이전트의 최적 행동 생성을 위한 행동 선택 네트워크의 계획 기능”, 한국 정보 과학회 춘계학술대회, 제 30권 제 1호, pp.473-475, 2003
- [4] 석황희, 김인철, “계획 기능을 가진 지능형 이동 에이전트 시스템”, 한국 정보 처리학회 논문지, 제 7권 제 11호, pp.3417-3426, 2000
- [5] T. Conde and D. Thalmann, “Learnable Behavioural Model for Autonomous Virtual Agents : Low-Level Learning”, Proceedings of Fifth International Conference on Autonomous Age

- nts and Multiagent Systems 2006, pp.89-96, 2006
- [6] T. Conde and D. Thalmann, "Autonomous virtual agents learning a cognitive model and evolving" Proceedings, IVA, pp.88-98, 2005
- [7] T. Abaci, J. Ciger and D. Thalmann, "Planning with Smart Objects", International Conferences in Central Europe on Computer Graphics, Visualization and Computer Vision, pp.25-28, 2005
- [8] E. de Sevin and D. Thalmann "A motivational Model of Action Selection for Virtual Humans", 2005 Proceedings of the Computer Graphics International, pp.213-220, 2005



전형배(Hyoung-bae Jeon)

2009년 2월 동국대학교 멀티미디어공학 학사
2009년 3월~현재 : 삼성 SDS

관심분야 : 게임인공지능, 계획시스템



한은지(Eun-ji Han)

2005년 3월~현재 동국대학교 멀티미디어공학과

관심분야 : 게임인공지능, 계획시스템



엄기현(Ky-Hyun Um)

1975년 2월 서울대학교 공과대학 응용수학과 공학사
 1977년 2월 한국과학기술원 전산학과 이학석사
 1994년 2월 서울대학교대학원 컴퓨터공학과 박사
 1978년 3월~2007년 6월 동국대학교 컴퓨터멀티미디어공학 교수
 2007년 7월~현재 동국대학교 영상미디어대학 게임멀티미디어공학과 교수
 1995년 3월~1999년 2월 동국대학교 정보관리처장역임
 2001년 3월~2003년 2월 동국대학교 정보산업대학 학장 역임
 2005년 3월~현재 한국 게임학회 자문위원
 1998년 12월~2001년 12월 한국 멀티미디어학회 부회장, 자문위원, 수석부회장 역임
 2007년 1월~2007년 12월 한국멀티미디어학회 회장

관심분야 : 게임시스템 및 구조 설계, 멀티미디어응용시스템, 멀티미디어데이터베이스



조경은(Kyung-Eun Cho)

1993년 2월 동국대학교, 전자계산학과(공학사)
 1995년 2월 동국대학교, 컴퓨터공학과 대학원(공학석사)
 2001년 8월 동국대학교, 컴퓨터공학과 대학원(공학박사)
 2003년 9월~2005년 8월 동국대학교 정보산업대학 컴퓨터멀티미디어공학과 전임강사
 2005년 9월~현재 동국대학교 영상미디어대학 게임멀티미디어공학과 조교수

관심분야 : 컴퓨터 게임 알고리즘, 게임 인공지능