

시설계를 위한 FSM 도구의 설계

김성룡^o 윤성희 김명세 오삼권
호서대학교 벤처전문대학원^o, 호서대학교
songryon@mail.co.kr^o, bongustar@hotmail.com, bongustar@hotmail.com
ohsk@office.hoseo.ac.kr

Design the FSM Editor for AI

Seong-Ryong Kim^o, Sung-Hee Yoon, Myung-Se Kim, Sam Kweon Oh
Dept. of The Graduate School of Venture, Hoseo University^o
School of Computer Engineering, Hoseo University

요 약

가상현실에 생성되는 NPC(Non-Player Character)의 인공지능을 설계 하는 AI 디자이너가 NPC 행동 패턴을 효율적으로 모델링을 할 수 있게 도와주는 FSM(Finite-State Machines) 도구를 제시한다. 이 도구는 FSM 각 상태와 상태의 변이에 따른 여러 가지 행동 패턴을 쉽게 모델링 할 수 있으며, AI를 디자인하는 디자이너가 가상현실 속에 존재하는 NPC의 AI를 설계 하고 프로그래머가 AI를 구현하는데 필요한 시간을 줄여준다. 또한 행동 패턴의 FSM 구조가 일반화 되어 재사용성이 높아질 뿐만 아니라 AI를 하드코딩으로 구현 하는 것을 방지 할 수 있다.

1. 서 론

가상현실에서의 NPC 인공지능은 사용자 및 주변 환경변화에 따른 피드백으로 생겨난 데이터를 이용하여 수학적 계산을 하거나 조건을 통하여 여러 행동 패턴들 중 상황에 맞는 하나의 패턴으로 행동을 하도록 되어 있다. 이러한 인공지능을 구현하는 방법 중 FSM을 이용한 방법이 최근 급성장 하고 있으며, 응용분야로는 온라인게임 에서 많이 사용하고 있다. FSM은 NPC가 가질 수 있는 상태들과 각 상태의 전이를 위한 이벤트 또는 조건을 정의 하여 NPC의 상태변화를 구현하는 것으로서 다른 인공지능 기법들 보다는 상대적으로 구현이 간단하고, 다양한 형태의 NPC의 행동패턴을 정의할 수 있는 장점을 가지고 있다[1].

그 동안 가상현실 구현시 NPC 행동패턴 구현 은 AI 디자이너가 행동패턴을 설계하고 프로그래머는 설계된 패턴을 가지고 하드코딩으로 각각의 행동 패턴을 구현을 하였다. 이와 같은 개발방법은 행동 패턴이 단순하게 구현되고 재사용성과 확장성도 떨어지는 문제점이 있었다. 이러한 문제점을 해결하기 위하여 행동 패턴을 제어하는 여러 요인을 가상현실 디자이너가 외

부에서 제어 할 수 있도록 프로그램을 개발 하였다 [2]. 그러나 이는 프로그램에서 제어되는 모든 인자에 대한 이해를 디자이너에게 요구 해야 하는 문제가 있고, 디자이너와 프로그래머간의 많은 협조가 필요한 개발 방법이다.

본 논문은 이러한 개발 환경을 개선하기 위한 방법으로 디자이너와 프로그래머의 능률적인 역할 분담과 FSM으로 구현된 인공지능의 재사용성, 확장성 그리고 다양한 패턴을 쉽게 구현 할 수 있게 도와주는 FSM 모델링 에디터를 제안하고 설계한다. 본 논문의 구성은 2장에서 관련연구를 살펴보고, 3장에서 기존에 있는 FSM 도구를 설명한다. 4장에서 FSM 기반의 시설계를 위한 FSM 모델링 도구를 설계하고, 마지막으로 5장에서 결론과 향후 과제에 대해 기술한다.

2. 관련연구

FSM 모델링을 위한 에디터에 관한 연구는 인제대학교의 전학과의 게임개발연구실에서 연구한 "FSM의 직관적인 모델링을 위한 에디터 설계[2]"가 있다. 인제대학교의 연구내용은 모델링 에디터를 설계하는데

있어 상태와 상태 전이 이벤트정보를 일반화 시키고, 표 1 과 같이 다양한 형태의 행동패턴에 적용 시킬수 있는 NPC 행동 패턴 오브젝트의 데이터 저장 구조를 제시하고 있다.

표 1 데이터 구조

데이터형	설 명
unsigned nTransitions;	상태별 유한상태 개수
int InputID;	상태전이를 위한 이벤트 ID
int OutputID;	상태출력 ID
int Value;	각 상태를 전이 시키는 변수들의 설정값
int stateID;	상태의 고유 ID

3. 기존 FSM 도구

기존 FSM 도구로는 Aldec 사의 Active-HDL이라는 제품이 지원하는 FSM 도구[4]와 Gerwin Klein의 FSM 도구[5]가 있다. Aldec 사의 Active-HDL은 공정제어 설계하기 위한 도구로서 GUI를 지원하는 에디터이다. 이 도구는 단지 공정제어를 설계하는데 필요한 기능으로 FSM 도구를 지원하는 것으로 시를 모델링하는 도구로서 그 기능이 적합하지 않다. Gerwin Klein의 FSM 도구는 그래프 편집기를 만들기 위해 사용되는 자바도구의 예를 보여주기 위해 만든 것으로서 간단히 구현되어 있는 도구이며, 기능은 단순히 FSM을 표현하는 간단한 기능만을 제공하여 시를 모델링하는 도구로 적합하지 않다.

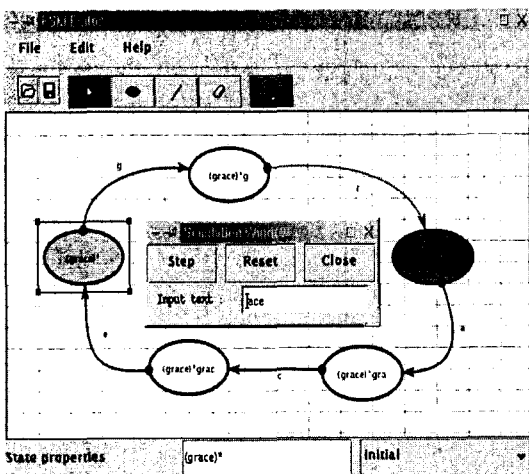


그림 1 Gerwin Klein의 FSM 에디터

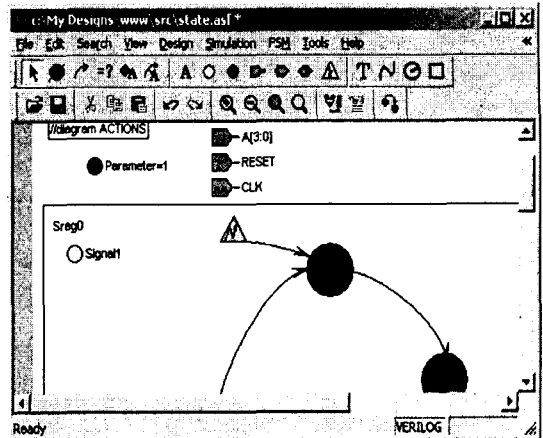


그림 2 Aldec 사 Active-HDL내 FSM 에디터

위 그림 1과 그림 2는 대표적인 기존 FSM도구의 실행 화면을 보여주고 있다.

4. FSM 모델링 도구의 설계

“FSM 기반 시설계를 위한 FSM 모델링 도구”는 시의 효율적인 모델링과 완성된 시 모델을 코드로 자동 변경하는 기능을 가져야 한다. 완성된 모델을 변경한 코드는 프로그래머가 바로 사용하여 시를 쉽게 구현 할 수 있게 도와준다. 그림 3은 본 논문에서 제시하는 FSM 기반 시 설계를 위한 FSM 모델링 도구의 구조를 간단하게 보여 준다.

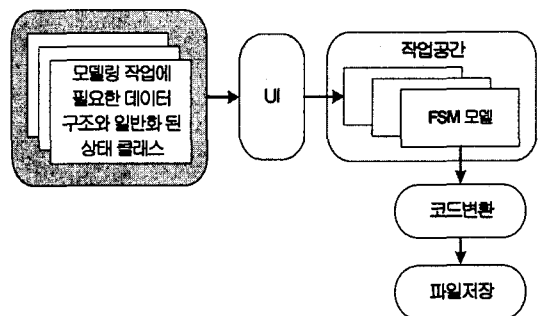


그림 3 FSM 도구의 구조

그림 3의 구조를 보면 UI를 이용하여 작업공간에 FSM 을 모델링하는데 있어서 데이터 구조와 상태를 클래스를 이용하여 작업하는 것을 볼 수 있다. 이는 FSM 모델링을 하는 동안 상태를 그래픽으로 표현하는데 필요한 객체 클래스이다.

객체 클래스들은 기능을 일반화 시켜 클래스로 정의

한 것으로서 이 객체 클래스를 통해 생성된 객체를 이용하여 모델링된 FSM은 기본적인 코드로 변환 시킬 수 있다. 그리고, 그 코드를 이용하여 시를 최종적으로 프로그래머가 개발한다. 이와 같은 형태의 FSM 도구는 시디자이너와 프로그래머 사이에서 일어나는 많은 의견 조율 과정을 줄여준다. 마지막으로 모델링 작업의 결과를 코드로 변환시키고 그것을 파일로 저장하는 것을 볼 수 있다.

4.1 요구 기능

표 2 요구기능

필요기능	설 명
상태표현 기능	모델링되는 상태를 표현하는 기능
전이조건 정의 기능	상태전이에 필요한 조건을 정의 하는 기능
코드변환 기능	모델링된 FSM을 코드로 변환하는 기능

“FSM 기반의 시설계를 위한 모델링 도구”에 필요한 요구 기능은 표2 와 같이 3가지의 기능을 가지고 있어야 한다.

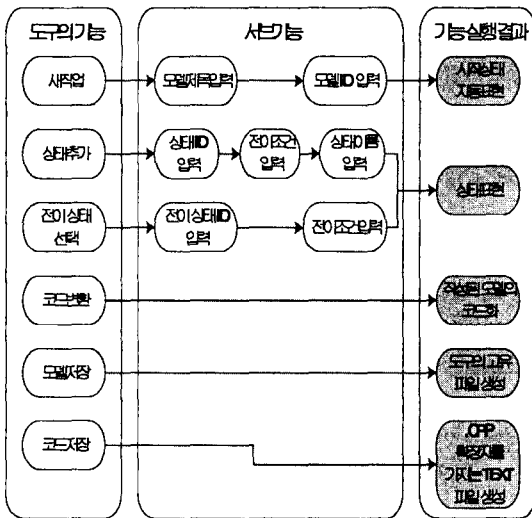


그림 4 도구의 기능과 실행 결과

표2 에서 제시한 3가지 기능을 기본으로 하여 구체적인 도구의 기능을 그림 4와 같이 제시 한다.

그림4에서 도구의 기능에 표현된 기능들중에 새작업과 상태추가 기능은 표2에 제시된 상태표현 기능에 속하고, 서브기능에서 전이 상태 ID와 전이조건입력기능

은 표2에 제시된 전이 조건 정의 기능에 해당된다. 도구기능에서 코드변환과 코드저장은 표2에서 제시한 코드변환 기능에 속한다. 그림2의 기능 실행결과는 도구의 기능이 실행되었을 경우 나오는 실행결과를 간단히 보여주고 있다.

4.2 사용자 인터페이스

시 디자이너가 “시설계를 위한 FSM 모델링 도구”를 이용하는데 필요한 사용자 인터페이스는 4.1에서 제시하고 있는 기능을 포함 하고 있어야 하며, 모델이 표현되는 작업 공간을 가지고 있어야 한다. 표3은 사용자 인터페이스를 정의 한 것으로서 그림2의 도구의 기능과 서브기능을 사용자 인터페이스로 사용한 것을 보여주고 있다.

표 3 사용자 인터페이스

인터페이스	설 명
새작업	새 모델을 설계시 새 작업을 이용하여 새로운 작업을 시작한다.
작업이름 입력	새로운 모델링 작업을 시작할 경우 작업이름을 입력.
상태추가	전이될 새로운 상태를 추가할 경우 사용한다.
전이 상태선택	이미 표현된 상태중에 전이할 상태가 있을 경우 이용한다.
코드변환	완성된 FSM 모델을 코드로 변환시킨다.
코드저장	변환된 코드를 파일로 저장한다.
모델저장	작업중이던 모델을 저장한다.
전이조건 입력	상태추가 또는 전이 상태선택을 이용하여 전이될 상태가 생성되면 전이될 조건 값을 입력
상태 ID 입력	새로운 상태가 생성될때 그 상태에 고유 ID를 입력
작업공간	그래픽 적으로 모델링 작업을 하는 공간

4.3 객체 클래스

도구에서 사용자 인터페이스를 통해 작업을 하는 동안 사용자 인터페이스는 모델링에 기본적으로 필요한 클래스를 이용하여 객체를 생성하게 된다. 이는 FSM 모델링을 작업공간에서 그래픽 적으로 표현하는데 필요하며, 또한 모델링된 모델들이 단지 모델로만 끝나지 않고 완성된 모델을 통해 가상현실 속의 NPC의 행

동 패턴의 구현에 필요한 기본적인 코드를 제공하기 위해서 필요하다.

“시설계를 위한 FSM 모델링 도구”에서 모델링 작업을 하는데 필요한 객체클래스의 종류는 표 4 와 같이 정의 할 수 있다.

표 4 객체 클래스의 정의

객체 클래스 이름	설명
FSM클래스	현재의 상태와 기본 상태 정보 그리고 입력과, 출력, 조건 및 상태 리스트를 가지고 있다.
상태클래스	상태를 표현하는 그래픽요소와 상태 이름을 기억하는 데이터베이스를 가지고 있다.
조건 데이터 구조체	전이 조건으로 만들어지는 데이터 구조체를 일반화 시켜서 구조체로 만든다.

4.4 코드변환

“시설계를 위한 FSM 모델링 도구”는 AI 디자이너가 FSM 모델을 효율적으로 모델링 하는 것 외에 완성된 모델을 보고 시를 구현하는 프로그래머에게 완성된 모델의 기본적인 코드를 제공한다.

프로그래머에게 제공되는 기본적인 코드는 행동 패턴에 따른 상태별로 구체적인 코드를 구현하기 쉽게 구분 되어있다. 4.3에서 정의한 클래스들을 통해 생성된 객체들은 모델링 작업 동안 FSM을 구현하는데 필요한 정보들을 가지고 있게 된다. 이 정보들을 이용하여 코드로 변환을 한다.

5. 결론 및 향후 과제

많은 가상현실에는 많은 종류의 NPC들이 존재한다. 또한 앞으로도 많은 가상현실과 NPC들이 개발될 것이다. 이 전부를 시디자이너가 그림 몇 개로 설계하고 이를 보고 프로그래머가 하드코딩으로 구현을 하는 것은 생산성과 재사용성 그리고 확장성이 떨어지게 된다. 이를 방지하고 시를 효율적으로 개발 하기위해 FSM을 효율적으로 디자인하고 일반화된 코드를 제공해 주는 도구가 필요하다. 본 논문은 FSM 기반의 시를 개발에 앞서 설계를 하는데 필요한 FSM 도구를 소개하고 설계를 하였다.

향후 과제로서 지금 까지 설계하고 제시한 FSM 도구를 구현하고, FSM 도구에서 사용된 객체 클래스들을 이용한 효율적인 코드 변환에 대한 연구가 더

필요하다.

참고문헌

- [1]이정훈 신성운 오상권 “XML 기반의 FSM 시스템에 관한 연구” 정보과학회 2004년 춘계학술대회 VOL. 31 NO 01 pp 0550~0552 2004. 4
- [2] 송병근 이현진 김상균 서재현 “FSM의 직관적인 모델링을 위한 에디터 설계” 정보과학회 2003년 춘계학술대회 VOL. 30 NO. 01 pp. 0000 ~ 0000 2003.04
- [3]이경록 김인철 이재호 “온라인 컴퓨터 게임에서 지능형 NPC 구현을 위한 제어구조” 정보과학회 2002년 춘계학술대회 VOL. 29 NO. 01 pp. 0277 ~ 0279 2002 . 04
- [4]http://www.aldec.com/support/application_notes/knowledgebase/an0003_design_entry.htm
- [5]<http://www.doclsf.de/grace/about.html>