다양한 상황을 시뮬레이션 하기 위한 멀티미디어 스크립트

*신 동 승, 박 정 용, 배 경 표, 박 종 희
경북대학교 전자공학과

A Multimedia Script for Simulation of Uncanned Situations

*Dong-Seung Shin, Jung-Yong Park, Kyung-Pyo Bae, Jong-Hee Park
Dept. of Electronic Engineering, Kyungpook National University

요 약 : 멀티미디어의 정보 전달 기능과 인상에 의한 교육 기능이 강력함에 따라 많은 멀티미디어 교육용 시스템이 개발되었으나 대부분의 경우 내용이 빈약하거나 시나리오가 고정되어 있어서 다양한 상황을 연출하지 못하고 단편적인 내용 전개에만 그치고 있으며 학생의 참여가 미흡한 일방적인 지식 전달에 머무르고 있는 실정이다. 이것은 멀티미디어 데이터는 많으나 데이터를 상연할 시나리오가 제작자의 의도대로만 전개되고 사용자의 행동에 따라 다양한 상황으로 전개할 수 있는 여건이 마련되지 않았기 때문이다. 따라서 본 논문에서는 데이터베이스와 멀티미디어 스크립트를 이용하여 필요한 데이터만 갖추어지면 제작자의 창의력에 따라 다양한 교육 내용을 저작할 수 있고, 실행 중에 사용자가 다른 상황으로 전개할 수 있는 스크립트와 시뮬레이터를 제안한다.

1. 서론

교육방법에 있어서 멀티미디어를 이용한 교육방법이 놀라운 효과가 있다는 것은 이미 증명되어 왔다[1][2]. 이에 따라 많은 멀티미디어 컨텐츠들이 제작되어 왔으나 대부분의 경우 단순한 구성으로 인해 사용자의 참여가 미흡한 아동용 교육 프로그램 또는 일방적인 지식 전달 수단에 그치고 있는 실정이며, 어떤 것들은 소방 훈련[3], 발전소 사고 대비[4][5] 등 특정 전문가 교육 분야에만 활용되고 있다. 이에 비해 언어 교육 시스템은 일상 생활에서 일어나는 거의 모든 상황을 시뮬레이션 해야 하는 경우가 많다 따라서 창의력을 가진 제작자가 다양한 교육 내용을 제작할 수 있도록 Dustin[1] 등에서는 저작도구를 개발하여 같이 제공하고 있다. 최근의 시뮬레이션 게임, 교육 시스템 등은 거의 기본적으로 저작도구를 제공하고 있고, 저작도구의 기능도 설계할 때 실제로 어떻게 표현될 것인지 미리 눈으로 보면서 작업하는 WYSIWYG의 기법을 사용한 것이 대부분이다. 그러나 시뮬레이션 진행 도중에 사용자들의 예측치 못한 행동으로 인해 발생할 수 있는 모든 상황을 설계 시에 미리 예측해서 만들어야 하는 것에는 특별한 대안이 없다. 본 논문에서는 스크립트와 미리 만들어진 Agent 객체들을 이용하여 장면 저작 및 시뮬레이션 도중에 사용자가 개입하여 미리 입력된 고정된 상황이 아닌 예기치 못한 상황으로 전개할 수 있는 방법을 제안한다. 2장에서는 기존 관련 연구를, 3장에서는 제안된 시뮬레이션 시스템의 동작 방법을 각각 알아보고 4장에서는 시스템의 구현 결과를 보여준다. 5장에서 결론 및 앞으로의 연구 방향을 제시한다.

2. 관련 연구

MADEUS[6]와 같은 시스템에서는 멀티미디어 상연을 위해 미디어의 시공간 배치를 고려하는 저작기능을 제공하고 있으나 본 논문에서는 시나리오를 사용자 의도대로 다양하게 전개하는 것이 목적이므로 관심사가 아니다

MIT에서 개발한 언어 교육 시스템인 Dustin[1]은 교육 내용의 다양화를 위해 MOPed라는 저작도구를 제공하고 있다. 이 저작도구의 초기 버전은 복잡한 문법을 사용하여 시나리오와 if-then 규칙을 저작할 수 있도록 하고 있다. 그러나 이러한 저작물이 저작단계에서 이해하기 어려운 이유로 비주얼한 방법으로 시나리오를 작성할 수 있도록 MOPed라는 도구가 개발되었다. MOPed는 스크립트(scripts)라고 부르는 이벤트의 순서를 표현하는 구조를 사용하고 있으며 스크립트에 기술된 사건을 순서대로 실행하면서 시나리오가 전개된다. 그러나 MOPed는 시나리오를 저작할 당시에만 다양한 상황을 만드는 것이 가능할 뿐이며 실행 단계에서 사용자의 예측치 못한 행동, 예를 들어 잘못된 문법 사용, 의미는 같으나 다른 어휘 구사, 주제와는 상관없는 이야기를 하는 등의 행동에 대해서 다른 상황으로 시나리오를 전개하기보다는 원래의 시나리오로 진개할 수 있도록 교육 보조자(Simulated agent)를 통해 사용자가 지정된 행위를 하도록 유도하고 있어 다양한 상황 전개를 하지 못하고 있다. 본 논문에서는 비주얼한 시나리오 저작 기능은 지원하지 않지만 스크립트를 이용하여 시뮬레이션 실행 중에 다른 상황으로 전개할 수 있는 시뮬레이터를 제안한다.

3. 장면 기반 시뮬레이터

3.1. 개요

장면 기반 시뮬레이터는 스크립트로 기술된 장면을 시나리오 전개의 기본 단위로 하여 실행한다. 여기서 장면이란 현재 시나리오를 보는 주시자의 시공간적 위치가 단절되지 않는 상태에서 상연되고 있는 시나리오를 말한다. 스크립트는 이벤트의 순서를 기술한 구조이다[7] 시나리오는 장면에 스크립트로 정의한 이벤트의 연속으로 기술되고 기술된 이벤트가 하나씩 실행됨으로써 시뮬레이션이 전개된다 본 논문에서의 스크립트는 기존의 스크립트[7]의 구조를 목적에 맞게 변형시켜 사용하였다. 특히 이벤트를 기술하는 부분에서 순서에 관계없이 트리거 조건만 맞으면 발생하는 이벤트를 기술하는 방법으로 사용하였다

3.2 장면 구성 내용

장면은 시나리오 전개의 기본으로 객체들의 상호작용과 기술된 이벤트를 통해 사건을 전개한다. 장면의 구성요소는 다음과 같다.

Description : 장면의 기본 줄거리 및 필요한 사항들을 간략히 설명한다.

Role & Prop : 장면 내에 등장하는 객체들에 대해 정의한다. 기존 스크립트에서 Role, Props로 각각 구분한 것을 하나로 통합시킨 것이다.

Pre-Condition : 장면이 실행되기 위해 만족되어야 하는 필요조건을 정의한다. 현재의 장면이 실행되기 위해서 반드시 만족되어야 하는 상태들이 기술된다.

Initial State : 장면이 실행되었을 때 필요한 초기화 상태를 기술한다. Role과 Props들의 물리적, 논리적 초기 상태를 정의한다.

Events : 장면에서 발생하는 다양한 이벤트들을 기술한다 이벤트는 순서대로 발생하는 것이 아니며 이벤트를 실행시킬 조건만 만족되면 실행된다.

End-Condition : 장면을 종료하기 위한 필요조건이다 고정된 시나리오를 전개하지 않도록 이벤트가 순서대로 실행되지 않으므로 언제 종료된다고 확실히 말할 수 없다. 따라서 여기서 어떤 상황일 때 종료시킬지를 정의한다.

Post-Condition : 장면이 실행되고 난 후의 결과와 상태를 나타낸다. End-Condition이 장면 내의 모든 상태가 아닌 일부 상태만 정의하는 반면 Post-Condition의 모든 상태를 나타낸다.

이러한 요소들은 다음과 같은 형태로 표현된다.

Description : 개를 무서워하는 사람이 도망치면 누군가가 니타니서 개를 쫓아내는 장면		
	Pre-Condition	**Initial-States**
Role & Props Worker1:THuman Worker2:THuman DonQ:TDonQ Dog1:TDog	**Events** evOnTime:Dog1 Follow Human1 evOnCondition Human1 FleeFrom Dog1;Human2 Kick Dog1	
	End-Condition	**Post-Condition**

표 1 장면 'On The Road'의 스크립트

장면에 Pre-Condition이 정의됨으로써 이전에 실행한 장면의 Post-Condition이 다른 장면의 Pre-Condition에 정의된 상태를 모두 가지고 있으면 해당 장면을 실행할 수 있다. 또한 이전에 실행되었던 장면도 다시 실행될 수 있다

3.3. 장면의 이벤트 처리 및 객체들의 규칙 처리 방법

본 논문에서의 이벤트란 장면에서 발생하는 모든 사건을 포괄적으로 지칭하는 말이다. 예를 들어 시간이 1초 흘러간 것, 개가 짖는 것, 짖는 소리를 들은 것 모두가 이벤트이다. 이벤트는 장면의 실행 중에 객체들의 상호작용에 의해 발생할 수 있고, 사용자가 직접 발생시킬 수 있다.

3.3.1. 객체들의 규칙 처리 방법

객체들은 장면 내에서 일어나는 상황을 스스로 판단해서 자기의 다음 행동을 결정한다. 이러한 객체들을 Agent라 하며[8] Agent를 중심으로 하는 시스템도 제안되었다[9] 객체들이 상황을 판단하기 위해서 다른 객체들의 상태를 참조하고 자신의 상태를 다른 객체들이 알 수 있도록 해야 하기 때문에 각 객체들은 자기의 상태를 메모리에 기록하고, 자기의 상태 및 메모리에 기록된 다른 객체들의 상태에 접근하여 자기의 규칙을 처리하고 다음에 취할 행동을 결정하게 된다. 예를 들어 개에게 쫓기고 있는 사람이 있을 때 취할 행동에 대한 처리 과정은 다음과 같다

1 장면 내의 객체들이 발생시킨 이벤트 검색
2. 누군가가 개에게 쫓기고 있는 이벤트가 있는지 검사
3 2의 조건이 만족되고 쫓기고 있는 사람과 개가 가시 범위에 있으면 개를 쫓아내러 간다

3.4. 사용자 정의 이벤트 처리 방법

사용자가 장면에서 발생하는 이벤트를 직접 처리할 수 있도록 하는 방법을 말한다. 이 방법을 이용하면 제작자의 의도와는 다르게 사용자가 의도하는 대로 시나리오를 전개하는 것이 가능하다. 사용자가 이벤트를 표현 문법에 맞춰 입력하여 시뮬레이터에 인식시켜 주면 시뮬레이터는 입력된 이벤트를 해석하여 실행 중에 이벤트를 검사하여 설정된 내용을 실행시키게 된다.

3.4.1. 이벤트 표현 방법

사용자가 의도하는 대로 사건 전개를 하려면 이벤트, 즉 장면에서 발생한 사건에 대한 지식을 사용자가 쉽게 다룰 수 있어야 한다 지식 표현 방법에는 declarative 방법과 procedural 방법이 있다[1] '밥을 먹는다'라는 행동을 표현하기 위해서 숟가락을 쥐고 밥을 떠서 입에 넣고 씹어서 삼키는 절차를 표현해야 한다(procedural 지식표현). 그리고 이러한 절차를 간단히 '밥을 먹는다'라고 간단히 표현하는 것이 declarative 지식 표현이다. 사용자가 지식을 단순한 방법으로 사용할 수 있도록 본 연구에서는 모든 이벤트를 declarative한 형식으로 표현하고 procedural 지식을 표현하기 위해서는 procedural 지식을 기초적인 declarative 지식의 조합으로 세분화하여 declarative 지식의 연속된 순서로 표현한다. 이를 위한 이벤트의 declarative한 표현 방법으로 '장면내의 어떤 객체가 무엇을 했다(Agent Action withParameter)'는 간단한 형식을 제안한다 이 형식은 Agent, Action, withParameter라는 세 가지 요소로 이루어져 있다 Agent는 행위의 주체가 되는 객체를 나타낸다 Action은 객체가 실행한 물리적, 정신적 행위를 포함한 모든 행위를 나타낸다 withParameter는 객체가 행위를 실행하기 위해 필요한 매개변수를 나타낸다 예를 들어 'Human1이 Dog1에게서 도망친다(FleeFrom)'는 사건이 발생했을 때 Agent는 Human1이 되고 Action은 FleeFrom, withParameter는 Dog1이 된다. 이것은 다음과 같이 표현된다.

Human1 FleeFrom Dog1

withParameter 요소는 단일 정보만이 아니라 여러 정보가

결합된 복합정보가 될 수 있다. 다음의 예는 x,y,z 좌표라는 복합 정보를 매개변수로 사용한 이벤트 표현이다.

Table1 Locate 150,0,50

3.4.2. 이벤트 트리거 및 실행 방법

이벤트가 발생할 트리거 조건과 조건이 만족되었을 때 실행할 내용으로 구성된다. 트리거 조건 표현은 이벤트 표현 방법과 같은 방법을 사용하며 발생한 이벤트가 트리거 조건과 같으면 기술된 실행할 내용이 실행한다. 예를 들어 트리거 조건으로 'Human1 FleeFrom Dog1'이라고 기술하면 장면 내에서 'Human1 FleeFrom Dog1'이라는 이벤트가 발생했을 때 트리거 조건이 만족된다. 트리거 조건과 실행할 내용은 세미콜론(;)으로 구분된다. 다음은 사용자 이벤트 선언의 한 예로써 Human1이 Dog1에게 쫓겨 도망치고 있을 때 DonQ가 Dog1을 걷어차는 이벤트 선언이다. evOnCondition은 다양한 이벤트를 종류별로 구분하기 위한 예약어이며 콜론(:)으로 구분한다.

evOnCondition:Human1 FleeFrom Dog1;DonQ Kick Dog1

이벤트 표현은 텍스트 정보로 기술되기 때문에 텍스트로 입력된 이벤트는 분석작업을 통해 시뮬레이터가 인식할 수 있는 코드로 변환되어 실행된다. 따라서 이벤트 표현 규칙만 알면 시뮬레이션 실행 중에 사용자가 이벤트를 발생시키는 것이 가능하다

위에서 보인 이벤트 선언에서 실행 내용 부분은 경우에 따라 여러 가지 다른 이벤트를 발생시킬 필요가 있다. 따라서 실행 부분에 if-then 규칙을 사용하여 더 세분화해야 할 필요가 있으며 차후의 연구과제로 남겨둔다.

4. 구현 및 실험 결과

본 연구에서 제안한 시스템을 Borland Delphi 3.0 Standard 버전으로 OpenGL을 이용하여 작성하였다 객체의 세밀한 표현보다는 장면 저작 기능, 객체들의 논리적 기능 표현 및 이벤트 처리 기능 구현에 중점을 두었다 다음의 예제는 건물 안으로 들어가려는 사람이 쫓아오는 개를 보고 무서워하여 도망치면 주위에 있던 누군가가 도와주는 줄거리를 스크립트로 작성하여 실행한 결과를 나타낸 것이다 그림 1은 저작도구로 저작한 화면이며 그림 2에서는 저작된 장면을 이용하여 멀티미디어로 재구성하고 개에게 쫓기고 있을 때 이벤트로 정의한 개를 쫓아내는 사람, 자기의 규칙대로 개를 쫓아내러 가는 Agent, 그리고 사용자가 직접 명령을 내린 Avatar들이 각각 개를 쫓아내러 가는 상황에 대한 시뮬레이션을 보여준다.

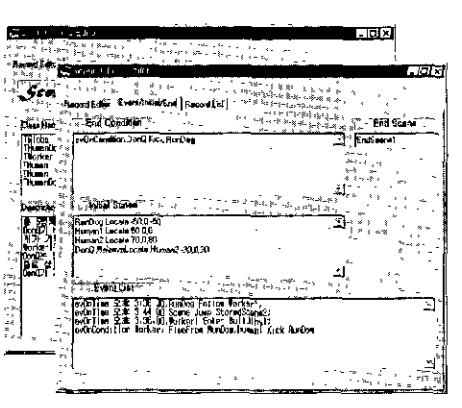

그림 1 장면 저작도구

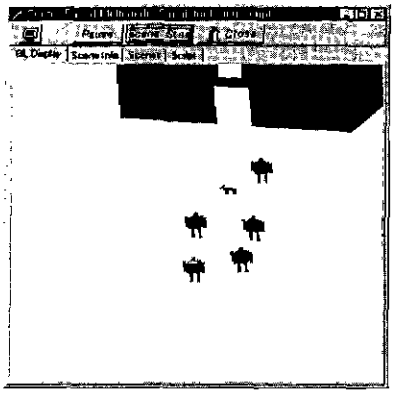

그림 2 장면 실행

시뮬레이터에서는 저작단계에서 정의한 이벤트를 보여주는 기능과 이벤트를 추가, 삭제, 변경하는 기능을 제공하므로 새로운 상황으로 전개할 수 있다

5. 결론

지금까지 멀티미디어 데이터베이스와 스크립트를 이용하여 다양한 상황을 만들 수 있는 시뮬레이터에 대해 살펴보았다. 본 시스템은 스크립트로 시나리오 제작이 가능하며 시뮬레이션 중에 새로운 이벤트 설정이 가능하기 때문에 사용자의 의도에 따라 다양한 상황으로 전개될 수 있다 특히 시스템에서 사용된 여러 객체와 장면들은 재사용이 가능하기 때문에 규칙과 이벤트를 다양하게 만들면 적은 데이터로도 얼마든지 다양한 상황을 만들어 낼 수 있으므로 성격에 따라 교육, 게임 등 여러 분야에 사용될 수 있다.

<참고문헌>

[1] E.Ohmaye, "Simulation-Based Language Learning An Architecture and a Multimedia Authoring Tool", *Northwestern Univ*, 1992

[2] J.H Murray and S.A.malone, "The Structure of Advanced Multimedia Learning Environments Reconfiguring Space, Time, Story, and Text," MIT

[3] *"Planning a drill inside MUSTER Environment,"* http'//wwwerg.casaccia.enea it/ing/tispi/emergency/muster/planning.html, 1997

[4] V.Vasandani, "Knowledge Organization in Intelligent Tutoring Systems for Diagnostic Problem Solving in Complex Dynamic Domains," IEEE Trans on Systems, Man, and Cybernetics, Vol.25, No.7, July, 1995

[5] Y.Matsubara, "Virtual Learning Environment for *Discovery Learning and Its Application on Operator* Training", IEICE Transactions on Information & Systems, V.E80-D, N.2, 1997

[6] M Jourdan, N Layaida, L Sabry-Ismail, "Authoring *Environment for* Interactive Multimedia Documents," Proceedings of the 13th conference on Computer Communication, November, 1997

[7] E.Rich "Artificial Intelligence," McGraw-Hill, pp.201-244, 1987

[8] S J.Russell, P.Norvig, "Artificial Intelligence A Modern Approach," Prentice Hall, pp 31-52, 1995

[9] M Badjonski, M.Ivanovic, Z.Budimac, "Intelligent Tutoring System as Multiagent System," IEEE International Conference on Intelligent Processing Systems. October, 1997