

아바타 행위 시나리오 생성을 위한 계층적 객체 및 스크립트 언어 모델

김재경[†], 손원성[‡], 임순범^{***}, 최윤철^{****}

요 약

텍스트 형식의 스크립트를 이용한 아바타 제어 기법은 사용자가 아바타의 행위 시나리오를 작성하기가 용이하고 특정 구현 환경으로부터 독립적으로 설계할 수 있다는 장점을 가지고 있다. 그러나 다양한 가상객체와의 행위 상호작용을 고려한 아바타 스크립트 언어 연구와 용이한 스크립트 작성을 위한 직관적인 인터페이스 연구가 미흡한 실정이다. 본 논문에서는 가상환경에서 아바타-객체 행위를 표현하는 컨텍스트 기반 객체 모델과 계층적 행위 스크립트를 제안한다. 객체 모델에서는 추상적 행위 명령시 발생할 수 있는 모호성을 제거하기 위해 객체의 상태를 표현하는 컨텍스트 정보를 정의하여 가상환경에서 사용자에게 직관적인 객체 기반의 아바타 행위제어 인터페이스를 제공한다. 또한 추상화된 인터페이스와 스크립트의 재사용성을 위해 제안 행위 스크립트는 계층적으로 정의되어 최상위 사용자 인터페이스 수준부터 구현환경에서 구동되는 스크립트까지 단계적으로 아바타의 행위 및 동작을 표현한다. 사용자는 제안 기법을 통하여 가상환경에서 다양한 객체들과 상호작용하는 아바타 행위 시나리오 스크립트를 빠르고 간편하게 생성한다.

Layered Object and Script Language Model for Avatar Behavior Scenario Generation

Jae-Kyung Kim[†], Won-Sung Sohn[‡], Soon-Bum Lim^{***}, Yoon-Chul Choy^{****}

ABSTRACT

A script language, which represents and controls avatar behaviors in a natural language style, is especially remarkable, because it can provide a fast and easy way to develop an animation scenario script. However, the studies that consider avatar behavior interactions with various virtual objects and intuitive interface techniques to design scenario script have been lack. Therefore, we proposed a context-based avatar-object behavior model and layered script language. The model defines context-based elements to solve ambiguity problems that occur in abstract behavior interface and it provides user interface to control avatar in the object-based approach. Also, the proposed avatar behavior script language consisted of a layered structure that represents domain user interface, motion sequence, and implement environment information at each level. Using the proposed methods, the user can conveniently and quickly design an avatar-object behavior scenario script.

Key words: Behavior(행위), Object Model(객체 모델), Context(컨텍스트), Script(스크립트)

* 교신저자(Corresponding Author) : 손원성, 주소 : 서울시 서초구 잠원동 잠원웨미리아파트 1동 1003호(137-796), 전화 : 02)393-7663, FAX : 02)393-7663,
E-mail : sohnws@gmail.com

접수일 : 2007년 3월 28일, 완료일 : 2007년 11월 2일

[†] 준회원, School of Information Science, Univ. of Pittsburgh
(E-mail : ki187cm@gmail.com)

[‡] 준회원, 경인교육대학교 컴퓨터과학과

^{***} 종신회원, 숙명여자대학교 멀티미디어과학과
(E-mail : sohnws@gmail.com)

^{****} 종신회원, 연세대학교 컴퓨터과학과
(E-mail : ycchoy@rainbow.yonsei.ac.kr)

* 본 연구는 한국과학재단 목적기초연구(R01-2004-000-10117-0 (2004))지원 및 2006년 한국학술진흥재단의 지원을 받아 수행된 연구임(KRF-2006-331-D00492)

1. 서 론

가상환경에서의 아바타 행위 제어는 중요 연구 분야로서 많은 연구들이 이루어지고 있다. 예를 들면, 아바타의 감정을 이용한 표정 변화[1], 모션캡쳐 데이터를 처리하여 보다 사실적인 움직임 제어[2], 인공지능을 이용한 자동화된 아바타 제어[3], 그리고 스크립트 언어[4-7]를 이용한 상위레벨의 동작 제어 등과 같이 아바타 제어의 목적과 종류에 따라 다양한 연구가 진행되어 왔다.

이중 아바타 스크립트 언어는 텍스트 형식의 명령 어로 아바타의 동작을 표현 및 제어하는 기능을 가진다. 즉, 스크립트 언어는 구현 환경의 아바타 엔진과 사용자의 자연 언어 명령 체계를 연결해주는 인터페이스 역할을 한다. 최근 스크립트 언어는 XML 기반으로 정의되고 있으며 사용자는 스크립트 언어의 정해진 문법을 이용하여 사용자 정의의 행위 시나리오를 작성하여 아바타 애니메이션을 재생할 수 있다. 이러한 스크립트 언어 정의에서 중요한 것은 사용자가 시나리오 스크립트를 용이하고 신속하게 디자인하기 위한 사용자 인터페이스를 제공하는 것이다. 이를 위해 스크립트 언어는 동작을 제어하기 위한 위치 좌표나 각도등과 같은 복잡한 저수준 개념(Low-level concept)으로부터 추상적이어야 한다[8]. 또한 스크립트 작성이 텍스트 편집기와 같은 비직관적인 인터페이스를 통해 이루어지기보다는, 3차원 사용자 인터페이스에서 아바타를 실시간으로 직접 제어하며 시나리오 스크립트를 생성하는 'Scripting by Demonstration'[9]이 총족되어야 한다.

이와 같은 추상화된 행위는 동작의 구체적인 수행을 지시하지 않으므로 모호한 명령이 될 수 있다. 따라서 아바타가 속한 환경의 정보를 이용하여 모호성을 해결해 주는 것이 필요하다. 그런데 가상환경에서 아바타 동작의 종류와 특성은 아바타의 사용 목적에 따라 제각기 다르다. 그래서 여러 종류의 환경에서 모든 경우의 아바타 행위를 제어하고 표현하기 위한 추상적인 스크립트 언어를 설계한다는 것은 매우 힘든 일이며, 그러한 스크립트 언어를 정의 하여도 기능이나 문법 등이 너무 방대해고 모든 경우에 대해 모호성을 해결하기가 어렵다. 그러나 특정 활용 도메인 내에서 사용되는 행위의 종류는 거의 정해져 있으며 추상화된 간결한 스크립트 언어의 정의

가 가능하다.

또한 기존의 아바타 스크립트 연구에서는 주로 아바타 동작 자체의 표현에 초점을 맞추어왔으며 객체와의 인터랙션에 대해서 연구가 부족한 실정이다. 이에 본 연구에서는 도메인 환경에서 추상적 행위 인터페이스를 지원하는 아바타-객체간의 행위 인터랙션을 컨텍스트 기반 객체 모델로 표현하고 이에 기반한 계층적 아바타 행위 스크립트를 정의하였다. 사용자는 제안 모델에 의해 가상환경에서 3차원 사용자 인터페이스를 통하여 아바타를 제어하고, 제안 스크립트에 의해 실시간으로 행위 시나리오를 생성한다.

2장의 관련연구에서는 기존 스크립트 연구와 아바타-객체 상호작용 관련 기법에 대해 언급하고 3장에서는 제안 컨텍스트 기반 객체 모델과 4장에서는 계층적 스크립트 행위 언어에 대해서 설명한다. 5장에서는 시스템 구현 결과와 5장에서 실험 및 토의, 마지막으로 7장에서는 결론 및 향후 연구에 대해서 알아본다.

2. 관련 연구

아바타 행위 스크립트는 아바타 행위 표현 연구 분야에서 중요한 이슈로 연구되어 왔다. 스크립트 언어들은 모두 아바타의 행위 표현 및 제어를 목표로 하지만 그 특성이 다양하다. 아바타의 행위를 스크립트로 표현하기 위한 연구인 Improv[10]에서는 저자의 스크립트 작성을 고려하여 상위레벨의 스크립트 모델을 제시하였다. 이를 기반으로 최근에는 웹 표준 언어인 XML을 기반으로 하여 재사용성과 표준화를 고려한 다양한 스크립트 언어들이 제안되었다.

CML[6]의 경우 아바타 신체를 상단, 중간, 하단으로 구분하여 각 부위에 대한 행위를 기술하였다. AML[4]은 행위간의 시간 동기화, 강도, 음성 출력과 같은 다양한 인자로 애니메이션을 호출하여 제어한다. TVML[11]의 경우 방송 스튜디오의 구성 및 방송에서 필요한 명령어들로 이루어진 스크립트로, 방송 프로그램을 스크립터가 손쉽게 작성할 수 있도록 하고 있다. 또한 CPSL[7]의 경우 사이버 교육 도우미를 이용한 시나리오 스크립트를 작성할 수 있는 언어이다. 이 외에도 얼굴표정, 몸동작, 음성, 감정, 제스처 등의 아바타 행위를 복합적으로 표현하고 있는 VHML[12], 아바타의 감정에 따른 음성 표현에 중점

을 둔 MPML 등[13]이 있다. 영문 스타일의 문법을 가진 스크립트를 통하여 3차원 가상환경에서 아바타 애니메이션을 제어하거나[14], 다이나믹 로직 프로그래밍 기법을 이용하여 복잡한 행위를 조합하여 표현하기도 한다[5]. 기존 스크립트 언어들은 대부분 상위레벨에서 사용자와 애니메이션 엔진간의 인터페이스 역할을 고려하여 디자인 되었다. 그러나 구현 계층의 기하 정보를 담고 있거나 복잡한 인자로 표현하고 있어 쉽고 빠른 스크립트 작성 인터페이스를 위해서는 개선이 필요하다. 또한 기존의 스크립트 언어들은 대부분 아바타 자체의 행위만을 표현하고 있어 객체와의 인터랙션을 표현하기 위해서는 아바타-객체의 행위 표현 모델의 정의가 필요하다.

아바타, 즉 에이전트와 객체와의 인터랙션을 표현하기 위해 많은 연구가 현재까지 다양한 분야에서 연구되고 있다. 초기에는 사용자-객체 인터랙션에 중점을 두고 객체 자체의 조작 기법에 대한 연구가 이루어졌으며 최근에는 몰입형 가상현실 시스템이나 증감 현실 환경으로 발전되어 왔다. 여기에 사용자가 아닌 가상 인간 개념을 도입한 에이전트-객체 인터랙션 기법은 OSR[15] 모듈에서 처음으로 제안되었으며 그림 1과 같이 기본적인 'grasping' 작업에 대해 주로 초점을 두고 있다. 스마트 오브젝트[16] 기법은 에이전트와 객체간의 행위를 오토마타, 동작, 부분 등으로 표현 및 정의하여 객체기반의 아바타 제어 기법을 제안하였다. 그러나 스마트 오브젝트는 아바타와 객체의 단일 인터랙션만 지원하여 다수의 객체와의 작업 수행을 위해서는 개선이 필요하다.

앞서 언급한 스크립트 언어들을 이용하여 시나리오를 작성하기 위한 도구들의 인터페이스 기법도 중요한 요소로서 여러 가지 방법들이 제시되어 왔다.

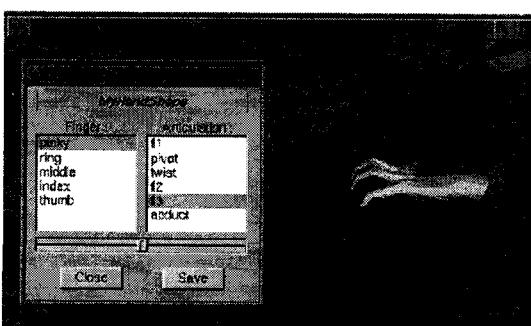
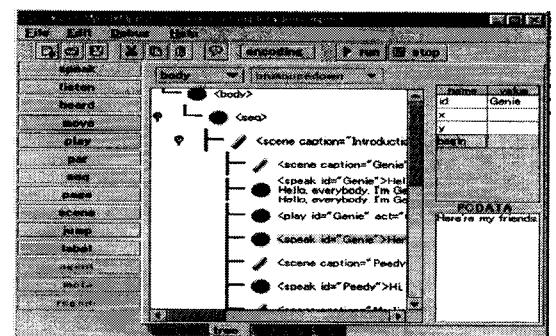
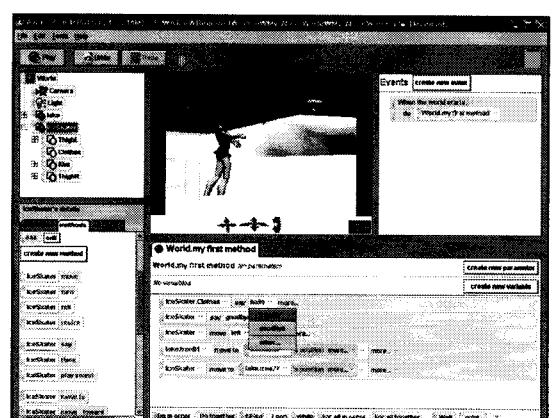


그림 1. Smart Object의 객체-아바타간의 손동작 정의

가장 기본적인 방식으로는 메뉴나 컨트롤 버튼을 이용한 WIMP(Windows-Icon-Menu-Pointer) 방식이다(그림 2(b)). 이는 사용자가 직접 XML과 같은 언어를 이용하여 작성하기 힘들기 때문에 텍스트 에디터와 같은 환경에서 직접 스크립트를 타이핑하는 것을 보조하는 역할을 한다[12,13]. 본 논문에서 제안하는 인터페이스와 유사한 환경을 제공하는 Alice[17]는 삼차원 가상환경에서 객체들을 직접 조작하고 그 과정을 순차적인 스크립트로 저장하고 재생할 수 있다. 그림 2(b)에서 보듯이 GUI 도구가 객체들을 렌더링하고 회전, 이동, 색상 변화, 확대등과 같은 세부 동작들을 리스트로 사용자에게 제공하는 방식이다. 유사한 인터페이스로 3DMax나 Maya와 같은 프로그램이 있다. 이 방식은 객체의 세밀한 동작을 제어할 수 있는 장점이 있으나 시나리오를 작성할 때 저수준 단계에서 모든 동작들을 제어해야 하므로 많은 시간과 노력이 듦다.



(a)



(b)

그림 2. MPML(a)과 Alice(b)의 스크립트 작성 인터페이스

다른 연구의 경우 스크립트 자체를 영문 스타일의 자연어로 정의하여 사용자가 직접 문장을 타이핑하여 시나리오를 작성하는 기법도 제안되었다[14]. 혹은 직접 3D 화면에서 마우스나 펜 입력의 제스처를 이용하여 아바타 동작을 제어하고 시나리오를 작성하는 비주얼 스크립팅(Visual Scripting) 방식도 연구되었다[18,19]. 가상공간에 선이나 미리 정의해 놓은 간단한 도형 등을 그리면 아바타는 선을 따라서 이동하거나 도형의 종류에 따라 미리 정의된 동작을 수행하는 것이다. 이와 같은 방법은 키보드 입력이 불편한 모바일 환경에서 응용될 수 있을 것이다. 그러나 기존의 방식들은 아직까지 텍스트 위주의 입력 방식이거나 간단한 동작을 명령하는 수준에 그치고 있어, 활용 도메인 환경에서 필요한 작업들을 수행하고 이를 시나리오로 작성하기 위한 직관적인 인터페이스 기법이 요구된다.

3. 컨텍스트 기반 아바타-객체 행위 모델

본 논문에서는 사용자에게 쉽고 신속한 스크립트 작성 인터페이스를 제공하고, 작성된 스크립트의 재사용성을 높이는데 초점을 두고 있다. 제안기법에서 사용자는 객체기반의 행위 명령 인터페이스를 통하여 아바타의 행위를 제어하며 사용자의 명령 과정은 스크립트로 생성되어 재생이 된다. 이를 위해 컨텍스트 기반 아바타-객체 행위 모델과 계층적 아바타 행위 스크립트 언어를 정의하였다.

3.1 컨텍스트 기반 객체 모델

본 논문에서 컨텍스트는 객체의 상태, 행위 정보 등을 뜻한다. 컨텍스트 기반 객체 모델에서는 사용자에게 현재 객체 상태에 따라 실행 가능한 컨텍스트 행위 메뉴를 제공하기 위한 정보를 제공함과 동시에, 사용자가 선택한 행위를 실제적인 동작 시퀀스로 변환하기 위한 변환 템플릿을 제공한다. 이와 같은 객체 모델이 제공하는 행위를 사용자가 가상환경에서 실행하면 이들은 추상적인 작업 레벨 행위 스크립트로 저장된다. 생성된 스크립트는 객체 모델에 따라 상위레벨 동작 스크립트로 변환되며 이것은 구현 계층에서 각 응용 프로그램이 가지고 있는 해석기에 의해 해당 프로그램에 종속된 기본 동작 스크립트 형식으로 변환된다. 결과적으로 사용자는 추상적인

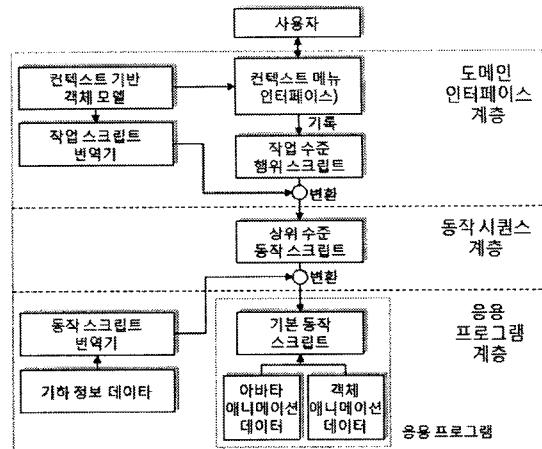


그림 3. 컨텍스트 기반 아바타-객체 행위 모델의 구조

행위 인터페이스를 이용하여 기하정보와 같은 저수준 표현으로부터 독립적인 행위제어 스크립트를 쉽고 빠르게 작성하고, 이는 각 계층을 거쳐 다양한 응용 프로그램 환경의 물리적인 정보에 맞도록 변환되어 재사용 될 수 있다(그림 3).

사용자에게 객체의 상태에 따라 실행 가능한 작업 레벨의 아바타-객체 행위 인터페이스를 제공할 때 해결해야 할 모호성이 발생한다. 첫 번째는 공간적인 모호성으로 작업레벨 행위는 아바타와 객체 간에 행위를 수행하기 위한 위치나 방향등과 같은 공간적인 정보를 표현하지 않는다. 예를 들어 ‘문을 열어라’의 경우 아바타가 문의 어느 위치에서 어느 방향으로 서서 여는 동작을 수행 할 것인지에 대한 정보가 없다. 따라서 실제적인 동작 수행을 위해서는 공간적인 참조 정보가 요구된다.

다음으로 행위 선택의 문제이다. 아바타가 임의의 객체에 대해 수행 할 수 있는 작업은 여러 가지가 될 수 있다. 그러나 객체의 상태나 아바타 제어의 주체 (스크립트 생성 혹은 재생자) 등과 같은 문맥적인 요소에 따라 현재 객체가 제공할 수 있는 행위 인터페이스는 달라질 수 있다. 객체와 아바타간의 모든 사용가능한 행위들을 사용자에게 나열 할 것이 아니라 현 문맥에서 실행 가능한 행위를 선택적으로 제공하는 것이 효과적이다. 또한 작업레벨 행위는 아바타가 객체에 무엇(What)을 할 것인지는 명시하지만, 실제로 객체와 어떻게(How) 동작을 수행 할 것인지는 지정하지 않는다. 따라서 실행된 행위에 대해 현재 객체와 아바타의 상태에 따라 어떤 동작을 수행해

야 할 것인지를 판단해야 한다.

제안 기법에서는 이와 같은 모호성을 해결하기 위해 컨텍스트 기반 객체 모델을 정의하였다. 객체 모델은 호환성 및 확장성을 고려하여 XML DTD 형식으로 정의하였다. 먼저 객체 리스트는 도메인을 구성하는 객체들로 구성되어 있는데 각 객체는 컨텍스트, 행위리스트, 동작리스트의 구조로 되어있다. 다음 절에서 각 요소들에 대해 자세히 알아보도록 하자.

가) 컨텍스트 요소

컨텍스트 요소는 행위 인터페이스와 동작 시퀀스를 제어하는 객체의 속성이다. 이것은 상태, 제어점, 접근그룹 및 도메인 종류 요소로 이루어져 있다(그림 4).

1) 상태 요소

상태(State) 요소는 객체의 내부 상태를 표현하는 변수명(Variable)과 변수값(Value)의 리스트로 구성된다. 각 변수는 영향 동작(Affective Motion)과 현재 상태 함수(State Function)를 가진다. 영향 동작이라는 것은 객체에 어떤 동작이 행해 졌을 때 임의의 변수의 값을 변화 시킬 수 있는 동작이며, 상태 함수(State Function)는 함수 내부에 정의된 특정 변수의 값이 현재 변수의 값과 일치하는 가의 여부를 Boolean 값으로 반환한다. 이 함수는 표준적인 처리를 위해 XSLT 스크립트로 이루어져 있다. 그림 5와 같이 임의의 객체는 자신의 상태를 표현할 수 있는 다수의 변수들을 가진다. 객체의 종류와 그 상태를 표현하는 변수명은 디자인 단계에서 설계된다. 객체의 상태는 동작의 이벤트 방식에 의해 변경되며, 변경된 객체의 상태는 구현 환경의 객체 처리 모듈에서 XSLT 스크립트 인터페이스를 통하여 접근할 수 있다. 객체 변수는 다음 절에서 설명할 행위 및 동작 구성을 결정하는 기준이 된다.

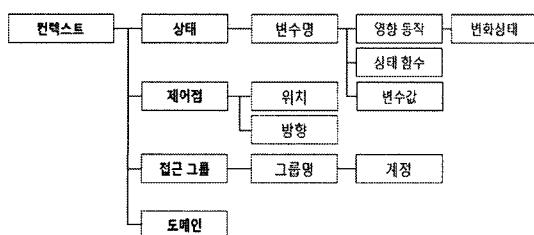


그림 4. 컨텍스트 요소의 DTD 구조

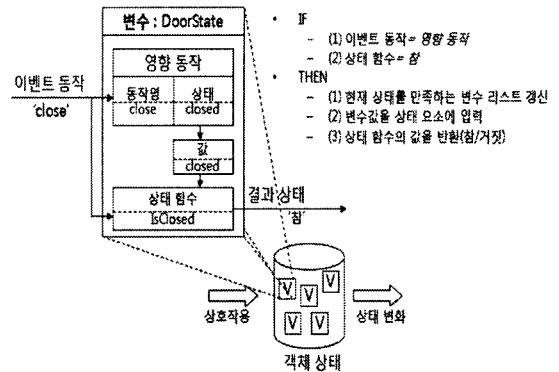


그림 5. 동작이벤트와 객체 상태 모델

2) 제어점 요소

제어점 요소는 아바타가 객체 주변에 위치하여 객체와 인터랙션을 하기 위한 공간적인 참조 지점이다. 이 제어점은 아바타의 위치와 방향 요소로 이루어져 있으며, 물리적인 3차원 좌표가 아닌 자연언어와 같은 추상적 표현 방식을 사용한다. 즉, 공간적인 정보를 상위레벨에서 표현하여 사람이 모델을 디자인하기 쉽게 한다. 또한 특정한 객체의 물리적인 정보에서 독립적으로 존재하여 의미적으로 동일하나 형태가 다른 여러 객체들에 적용이 가능하다.

제안 모델에서는 이와 같은 관점에서 제어점 요소를 다음 그림 6과 같이 정의하였다. 먼저 위치 요소는 각 객체에 대해서 Front, Behind, Left, Right, Center의 5개의 상대적인 기본 위치를 객체 중심 및 주변에 정의하고, 기본 위치를 AND 조합하여 4개의 확장 위치를 지정하였다. 각 위치점은 객체가 놓이는 xz-평면에 존재하며 z 축 방향에 있는 점을 기준 위치인 Front로 정의하였다. 다음으로 방향요소는 각 위치점에서 객체를 기준으로 아바타의 4개의 기본 방향인 Forward, Backward, Left 및 Right와 이들 기본 방향을 조합한 4개의 추가 방향을 합쳐 총 8방위를 표현한다. 각 위치에서 객체를 향하는 방향으로 아바타가 서있을 때 Forward가 되며 이를 기준으로 나머지 방향을 지정한다. 마지막으로 접촉점(Contact)은 아바타가 손을 이용하여 객체와 접촉을 하거나 가리키는 등의 동작을 할 때 객체의 특정 부분을 지정할 수 있도록 한다. 3차원 객체의 바운딩박스의 각 꼭짓점과 면의 중심점을 지정할 수 있다. 즉, 조작점은 기본 상하좌우, 전후 및 중심점 지점과 이들을 조합한 27개점으로 구성되어 있다.

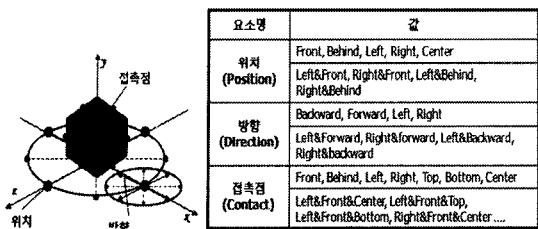


그림 6. 공간 제어점 요소 모델

3) 접근권한 그룹 요소

제안 기법에서 객체의 행위는 사용 목적에 따라 다른 사용자 혹은 시스템에 의해 호출될 수 있다. 즉, 사용 주체에 따라 객체의 행위를 호출하는 목적과 사용법이 다를 수 있다. 예를 들어 객체의 행위 인터페이스를 통하여 시나리오 스크립트를 작성하는 저자의 경우 디스플레이에 제공되는 모든 행위 메뉴에 대해 접근이 가능하다. 그러나 객체 내부에서 사용되는 세부 행위에 대해서는 접근이 필요하지 않다. 제안 기법에서는 기본적으로 스크립트 작성자, 스크립트 재생자 및 시스템 내부 호출 그룹으로 객체 행위 접근 권한 모델을 정의하였다. 각 그룹은 사용자 혹은 객체 아이디 리스트를 가지며 그룹에 속한 아이디들은 종속된 그룹의 권한을 가지게 된다. 그룹은 다음 절에서 설명할 행위리스트에 인자로 적용되어 해당 행위가 어떤 그룹에 의해서 접근이 가능한지 결정된다.

4) 도메인 종류 요소

본 논문에서 도메인은 아바타가 객체와 수행할 수 있는 작업의 종류와 특성의 범위를 말한다. 예를 들어 발표/강의 도메인은 스크린과 컴퓨터 객체 등으로 이루어진 공간에서 point 혹은 highlight text 등의 발표 관련의 아바타 작업으로 이루어진 가상환경이다. 도메인이 틀려질 경우 같은 객체라도 행위가 달라질 수 있다. 예를 들어 가상 일기예보 방송 도메인의 경우 가상도 출력을 위해 스크린 객체를 사용한다고 할 때, 사용자에게 제공하는 행위는 change map 등과 같은 기상관련 행위가 될 것이다. 도메인 종류 요소는 객체가 사용 될 수 있는 도메인들을 지정하는 것으로 객체의 특성에 따라 단일 도메인이나 복수의 도메인을 지정하여 여러 환경에서 사용될 수 있도록 한다. 도메인은 Domain 요소의 Type에 도메인 명을 지정함으로써 사용자 정의로 추가 될 수 있다.

나) 실행가능 행위 요소

실행가능 행위 요소는 디스플레이에서 사용자에게 행위 메뉴 인터페이스를 구성하기 위한 행위들의 집합이다. 제안 기법에서는 행위 요소의 모든 행위를 사용자에게 제공하는 것이 아니라 앞서 설명한 컨텍스트 요소에 따라 현재 객체가 실행 가능한 행위들을 선택하여 제공한다. 즉 컨텍스트 요소가 인자로 행위리스트의 행위 요소에 전달되고, 컨텍스트 조건을 만족하는 행위는 실행 가능한 행위로 선택되어 화면에 출력된다.

사용자가 임의의 행위를 실행 시켰을 경우 실제적인 동작 시퀀스를 생성하기 위해 행위 분할과정을 거쳐야 한다. 제안 모델에서는 하나의 행위는 서브 행위들로 구성되어 있는 행위 트리(Behavior Tree)로 구성되어 있다. 이 트리의 단말 노드는 더 이상 분할되지 않는 기본 행위이며 이것은 동작으로 표현되어 동작 시퀀스로 변환되게 된다.

행위트리의 각 서브 행위는 앞서 언급한 실행가능 행위와 같이 컨디션에 의해 선택적으로 실행된다. 예를 들어 사용자가 강의실의 컴퓨터 객체의 인터페이스 메뉴를 통하여 컴퓨터를 조작하는 행위를 명령했을 때, 아바타가 강의실 내에 있는지 혹은 문 밖에 있는지에 따라 문을 열고 들어오거나 혹은 바로 이동하여 컴퓨터를 조작한다. 또한 아바타가 강의실 밖에 있는 경우에도 강의실 문의 개폐 상태에 따라서 서브 행위가 다시 달라진다. 이와 같이 제안 모델은 객체 지향적인 행위 분할 및 호출 접근 방식을 가지고 있기 때문에 아바타와 다른 객체간의 인터랙션이 용이하다. 또한 도메인을 구성하는 객체의 변경이나 작업의 변화에도 유연한 작업 행위 수행이 가능하다. 왜냐하면 행위 수행에 연관된 객체들의 행위를 연결시켜 호출하고 대상 객체의 조건에 따라 상황에 맞게 아바타의 동작 시퀀스가 생성되기 된다. 따라서 객체의 구성이나 행위 구성의 수정시, 전체 동작 혹은 서브 행위들을 재구성 하는 것이 아니라 변경된 객체의 내부 구성만을 수정한다. 따라서 다른 객체에는 인터페이스만을 제공하므로 수정과 확장이 용이하다.

그림 7은 제안 모델에 따라서 컴퓨터 객체모델을 작성한 예이다. 방안의 컴퓨터의 'Next Page' 행위를 통하여 아바타가 스크린에 출력되는 페이지를 넘기는 작업을 수행하도록 구성되어 있다.

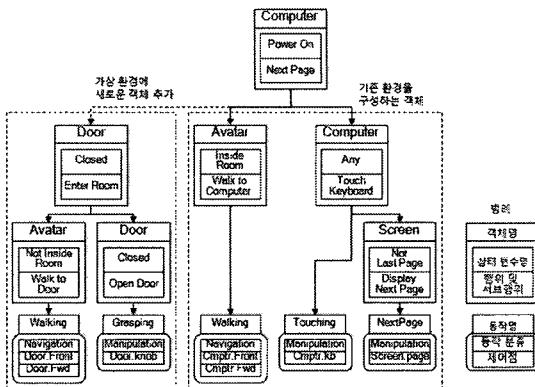


그림 7. 객체의 구성 변화의 예

이때 강의실에 문이 추가되어 강의실 밖에서도 컴퓨터 객체의 조작이 가능하게 하기 위해서 컴퓨터 객체의 행위리스트에 그림 7의 좌측의 점선으로 묶인 Door 객체와 'Enter Room' 행위 인터페이스를 추가함으로서 문을 열고 컴퓨터를 조작하는 아바타의 작업을 간단히 재구성 할 수 있다.

다) 동작 리스트 요소

행위 트리의 단말 서브 행위들은 더 이상 다른 서브 행위로 분할되거나 다른 객체의 행위를 호출하지 않는다. 행위가 추상적인 작업을 의미하였다면 동작은 구체적인 아바타의 움직임을 제어하기 위한 요소들로 이루어진다. 단말 서브 행위는 객체 모델에서 동작으로 변환되어 다양한 동작 제어 인자에 의해 상위레벨에서 동작을 표현한다. 동작 리스트 요소도 행위 리스트 요소와 마찬가지로 객체 모델 디자인 단계에서 작성된다.

동작 리스트 요소는 단말 서브 행위명과 이에 대응하는 동작들로 구성된다. 각 동작은 아바타의 동작명, 제어점 및 동작의 카테고리로 이루어져 있다. 아바타의 객체에 대한 동작의 제어를 위해 객체모델의 컨택스트 요소에서 정의한 아바타의 방향, 위치 및 접촉점 정보를 인자로 지정한다. 그리고 카테고리에서는 해당 동작이 걷기와 같은 이동 부류의 동작, 혹은 객체를 조작하는 조작 동작, 그리고 말하기 동작과 같은 제스처 동작인지를 지정한다.

4. 계층적 아바타 행위 스크립트 언어

제안 언어는 3계층으로 구성되어 있으며 도메인

계층의 작업레벨 행위 스크립트 언어, 동작 시퀀스 계층의 상위레벨 동작 스크립트 언어, 마지막으로 구현 환경 계층의 기본 동작 스크립트 언어로 구성되어 있다. 각 계층은 사용 목적에 따라 서로 다른 특성을 가지고 있는데 이를 알아보도록 한다.

4.1 작업레벨 행위 스크립트 언어

작업레벨 행위 스크립트 언어는 가장 상위 계층의 언어로써, 지정된 도메인 환경에서 사용자가 제공된 객체 행위 인터페이스를 이용하여 시나리오 스크립트를 기록하는 역할을 한다. 제안 언어는 가장 상위 레벨에서 사용자 인터페이스 역할을 하는 언어로서 다음과 같은 개념을 충족시키도록 디자인 되었다.

첫째로 기존 연구의 행위 스크립트에서 미리 정의된 고정된 행위를 이용할 경우, 도메인에서 사용되는 행위가 변경 될 때마다 스크립트 언어의 전체 행위셋을 수정하여야 한다. 그러나 변화하는 도메인 환경에 유연하게 대처하기 위해서는 스크립트 언어의 수정 없이 도메인에서 필요한 행위가 재구성이 가능해야 한다.

둘째로 도메인 환경 내에서 사용자가 요구하는 모든 행위를 충족시키도록 추상화된 행위를 구성하는 것은 한계가 있을 수 있다. 왜냐하면 도메인에서 사용되는 추상적인 작업의 경우, 그 추상화의 레벨에 따라 여러 작업을 묶어 하나의 작업으로 표현 할 수도 있으며 반대로 가장 기본적이고 단순한 작업들로 표현할 수도 있다. 추상화의 레벨이 높을 경우 스크립트 작성이 용이해질 수 있으나 사용자의 자유도는 떨어지고, 그렇지 않은 경우 사용자의 자유도는 높아지나 스크립트 작성이 길어질 수 있다(그림 8). 따라서 사용자의 의도에 따라 복잡한 작업 행위나 기본적인 작업 행위를 모두 제공할 수 있어야 한다.

마지막으로 대부분의 아바타 동작 스크립트 연구의 목적은 작성된 스크립트를 일회성 정보로 버리지 않고 재사용하는 것이다[5,6]. 때문에 기존 연구에서도 스크립트 정의를 위해 XML 언어 형식을 택하여 구현환경과 독립적인 특성을 이용하는 것이다. 따라서

기하 조작 수준
아바타 제어를 위한
높은 자유도, 시간과 비용

추상화 제어 수준
아바타 제어를 위한
자동화 및 낮은 시간과 비용

그림 8. 저수준 및 추상화 레벨의 아바타 제어 특성

서 작성된 스크립트가 사용자 지식정보로서 다른 환경에서도 재사용이 가능하여야 한다.

먼저 유연한 도메인 종속 행위 셋 표현을 위해서 제안 언어는 자체적으로 문법적으로 행위 셋을 정의하고 있지 않으며, 객체와 객체의 행위 정보, 그리고 추가적으로 아바타의 음성 출력을 위한 텍스트 입력 정보 담기 위한 템플릿으로 스크립트를 정의하고 있다(그림 9). 기존의 스크립트 연구의 경우 스크립트 DTD에서 미리 정의된 유한한 개수의 행위를 이용하였으나, 제안 기법은 도메인을 구성하는 객체의 추가나 수정으로 스크립트에서 사용하는 행위의 구성이 가능하므로 스크립트 언어의 DTD 수정 없이 도메인에서 필요한 행위가 유연하게 재구성이 가능하다.

$$\text{Task-level Behavior} = (\text{Object}, \text{Executable-Behavior}, \text{Narration})$$

다음으로 행위 제어의 높은 자유도 및 추상화 유지를 위해서 사용자에 의해 작성된 스크립트는 사용자 정의의 패키지 개념으로 새로운 행위로 추가되어 도메인 행위로 구성된다. 객체가 제공하는 행위는 사용자에게 기본적인 객체 단위의 작업레벨 행위를 제공하여 주므로, 스크립트 작성자에게 사용자 정의의

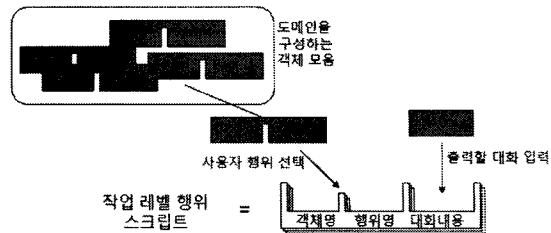


그림 9. 템플릿 구조의 작업레벨 행위 스크립트

시나리오 작성의 자유도를 보장하여 준다. 이때 작성된 스크립트는 패키지 라이브러리에 등록되어 다른 스크립트에서 하나의 행위로서 재호출 될 수 있다. 즉, 패키지 행위는 추상화 레벨이 높아 여러 작업을 하나의 작업 명령으로 실행 할 수 있는 반면, 자유도는 상대적으로 낮다. 이와 같은 템플릿과 패키지 개념에 의해 작업레벨 행위 스크립트는 다음과 같이 정의되었다.

$$\begin{aligned} \text{Task-level Behavior Script} &= (\text{Task-level Behavior} \mid \text{Package})^+ \\ \text{Package} &= (\text{Task-level Behavior} \mid \text{Package})^+ \end{aligned}$$

다음 그림 10은 이와 같은 개념들에 의해 작성된 작업레벨 행위 스크립트와 패키지 사용 예이다.

시나리오 스크립트 파일의 예 <pre><?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE Task-level_Behavior_Script SYSTEM "task.dtd"> <Task-level_Behavior_Script domain="presentation"> ... <Task-level_Behavior object="computer" executable_behavior="display next page" narration="Let me show the next page"/> <Task-level_Behavior object="paragraph" executable_behavior="point" narration="I'll show you a demonstration about push/pop operation"/> <Task-level_Behavior object="box" executable_behavior="push"> <Package name="Push Operation"></Package> ... </Task-level_Behavior_Script></pre>
Push 행위에 대한 패키지 파일의 예 <pre><Registered_Package name="Push Operation"> <Task-level_Behavior object="box1" executable_behavior="push" narration="Let's push a box item on the empty floor"/> <Task-level_Behavior object="box1" executable_behavior="point" narration="Now, stack data structure has an item"/> <Task-level_Behavior object="box2" executable_behavior="push" narration="I'm going to push the next item"/> ... </Registered_Package></pre>

그림 10. 작업레벨 스크립트의 예

4.2 상위레벨 동작 스크립트 언어

상위레벨 동작 스크립트는 그림 11과 같이 객체 모델에의 객체 트리에서 생성된 실행이 가능한 서브 행위 정보를 기반으로 아바타의 동작을 인자로 표현하는 것이다. 실행이 가능한 서브 행위는 앞서 설명하였듯이 행위명과 조건으로 표현된다. 상위레벨 동작에서는 기본적으로 동작명, 객체명, 카테고리 요소로 구성된다. 카테고리의 종류에 따라 이동 동작일 경우 상태(Condition)의 제어점(Control Point) 정보를 해석한 아바타의 방향과 위치를 추가적으로 인자로 표현하며 객체 조작 동작일 경우 대상 객체의 조작위치를 인자 요소로 가진다. 상위레벨 동작은 구현 환경에 독립적인 계층으로 각 요소는 객체 모델의 추상적인 표현 양식을 따른다. 따라서 위치나 방향 등의 인자 값은 제어점 요소 모델의 공간 참조 표현으로 이루어져 있다. 상위레벨 동작은 도메인 계층과 독립적으로 행위 수행에 필요한 동작 시퀀스로, 각 동작 실행에 필요한 추상적 정보만을 표현한다.

4.3 기본 동작 스크립트 언어

기본 동작 스크립트는 제안 계층 구조에서 가장 하위에 위치한 구현 계층에 속한다. 이것은 각 구현 환경에서 동작을 제어하기 위해 필요한 물리적인 수치 등을 담고 있다. 따라서 상위레벨 동작 스크립트와는 다르게, 생성된 스크립트는 특정 구현 환경에서만 사용될 수 있다. 기본 동작 스크립트는 각 아바타 소프트웨어에서 구동되는 것이므로 문법과 구조가 상이할 수 있다. 따라서 각 구현 환경은 해당 환경에서 사용되는 기본 동작 스크립트를 생성할 수 있는 해석기를 가지고 있다. 해석기는 상위 동작 시퀀스 계층의 상위레벨 동작 스크립트를 해석하여 해당 환경에 적절한 기본 동작 스크립트로 변환한다. 제안 시스템에서는 구현 계층에서 필요한 기본 동작의 인자를 표현하기 위한 템플릿으로 다음 표 1과 같은 요소들을 정의하였다.

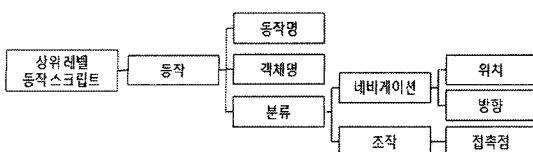


그림 11. 상위레벨 스크립트의 DTD 구조

표 1. 기본동작 스크립트 언어의 구성요소

요소	하위 요소들	기능
아바타	애니메이션 데이터	저수준 객체 애니메이션 데이터를 호출 인터페이스 정보
	애니메이션 데이터	저수준 아바타 애니메이션 데이터를 호출 인터페이스 정보
	조작	조작 동작을 위한 역운동학의 종말점(End effector)
	네비게이션	현재 아바타의 위치, 목적지, 방향 정보
	나레이션	음성출력 혹은 텍스트 출력 정보
일반 매개변수	반복	반복 동작의 횟수 정보
	프레임	해당 동작의 프레임수

상위레벨 동작의 자연어적인 정보를 이와 같은 구현 환경의 수치 정보로 변환하기 위해서는 매핑 정보가 필요하다. 제안 시스템의 구현 환경에서는 이러한 정보들을 관리하기 위해서 환경 정보 파일을 가진다. 환경 정보 파일은 상위레벨 동작명과 애니메이션 데이터 파일의 매핑 정보, 가상공간에서 각 객체의 중심위치 좌표와 이를 기준으로 상대적인 전후 및 좌우 지점의 좌표 등을 담고 있다. 구현환경에 속한 번역기는 이들 정보를 이용하여 기본 동작 스크립트의 정보를 생성한다.

5. 구현 결과 및 적용

제안 기법의 적용을 위하여 본 논문에서는 대표적인 아바타 활용 환경인 발표/강의 도메인을 선정하였다. 표 2와 같은 객체로 구성된 강의실에서 객체들과 상호작용을 하며 다양한 행위와 함께 정보를 전달한다. 즉 아바타는 행위를 이용하여 강의 자료를 가리키거나 줄을 긋거나 혹은 시연을 하며 발표를 진행한다. 객체 모델은 기본적으로 스크린, 컴퓨터 등과 강의에 보조가 되는 여러 가지 객체들이 될 수 있다. 각 객체는 객체 모델에 따라 정의되어 고유한 실행가능 행위들을 가지고 있으며 이들은 해당 객체의 컨텍스트에 따라 디스플레이에서 사용자에게 제공된다.

표 2. 도메인 구성 객체리스트와 실행가능 행위

도메인 객체	실행 가능 행위들
책(Book)	Start_Lecture
컴퓨터(Computer)	Next_Page, Prev_Page, Goto_Page
문단(Paragraph)	Explain, Point, Read, Highlight, Unhighlight, Ask_Questriion
그림(Picture)	Point_Picture
노트(Note)	Write_Note, Erase_Note
문(Door)	Enter_Room, Open_Door, Close_Door

도메인을 구성하는 각 객체들의 컨텍스트 요소의 제어점 정보에 따라 그림 12와 같이 각각 아바타와 상호작용을 위한 위치, 방향 및 접촉점을 지정하고 상태 요소들을 디자인 한다.

다음 그림 13은 가상 환경에서 사용자가 객체 인터페이스를 통하여 스크립트를 작성하고 있는 예이다. 스크립트 작성자는 각 객체를 클릭하여 표시되는 행위 메뉴를 선택하고, 아바타는 선택된 행위를 수행한다. 행위 메뉴는 객체의 상태에 따라 달라진다. 예를 들어 스크린의 텍스트에 강조(highlight) 마크 표시여부에 따라 사용자가 선택할 수 있는 행위가 변경된다. 사용자의 행위 선택 과정은 제안된 작업레벨 행위 스크립트로 생성되어 최종적으로 시나리오 스크립트로 저장된다. 사용자가 생성한 스크립트를 작업 패키지 라이브러리에 등록하게 되면 해당 시나리오 스크립트는 하나의 작업 행위로서 재사용이 가능하다. (b)는 STACK 연산 중 'pop' 및 'push' 연산을 시연으로 설명하는 시나리오를 작성하여 패키지에 등록 한 후, 이를 다시 하나의 작업 행위로 호출하여 사용하는 예이다.

그림 14는 가상공간을 구성하는 객체의 수정 시에 제안 객체 모델이 어떻게 적용하는지 예를 보여준다. (a)문 객체의 Open Door 행위를 실행하였을 때, (b)는 아바타가 문을 열고 들어가는 간단한 스크립트의 실행 예이다. (c)는 같은 스크립트를 실행한 결과이나 같은 실행가능 행위를 제공하는 다른 문 객체 모델로 환경을 변경한 경우이다. 이 경우 변경된 객체 내부의 컨텍스트와 동작 구성에 따라 아바타는 손으로 문을 잡아당기며 들어가던 동작에서 문을 옆으로

미는 동작을 하게 되며, 객체 또한 이에 대응되는 동작을 재생한다.

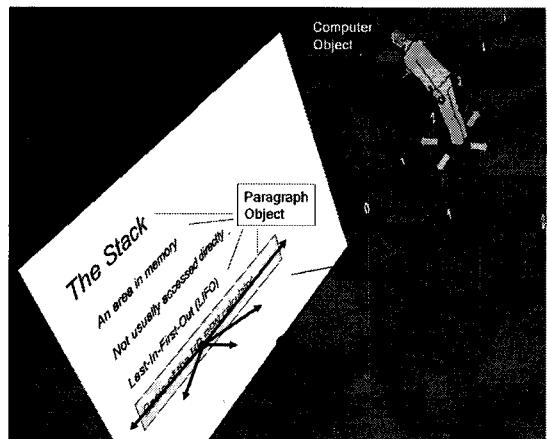


그림 12. 강의실 객체들의 제어점 요소 정의

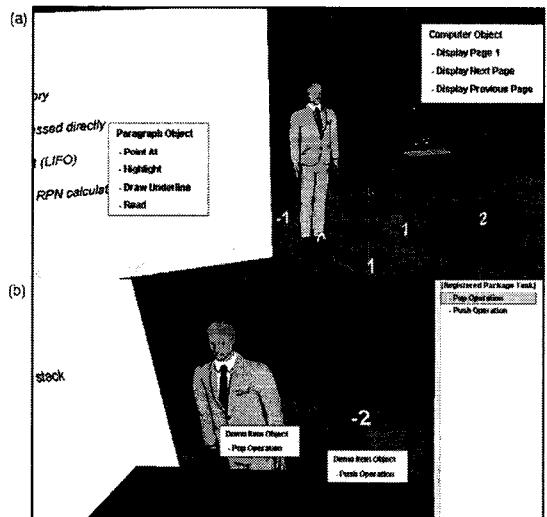


그림 13. 컨텍스트 기반 객체 인터페이스와 패키지 행위의 재사용

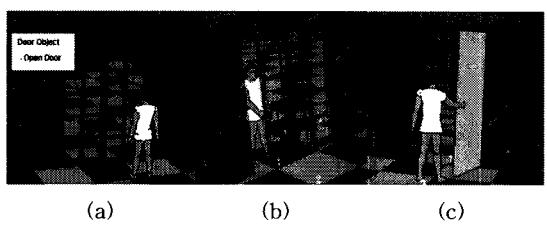


그림 14. (a)동일 행위를 서로 다른 문 객체(b)(c)에 적용한 결과

6. 실험 및 토의

본 논문에서는 제안된 방법에 대해서 정량적 및 정성적으로 시스템을 평가하였다. 평가자들은 39명의 대학원생(21명의 남학생 및 18명의 여학생)으로 구성되었으며 이들은 3차원 애니메이션 프로그램에 익숙한 20명의 전문가 그룹(G1)과 그렇지 않은 19명의 비전문가 그룹(G2)으로 분류하였다. 평가자들에게 제안 시스템과 비교 대상 시스템을 30분의 설명을 거쳐 사용법을 숙지시킨 후 테스트를 수행하였다.

실험 방법은 3차원 환경에서 추상적인 객체기반 행위 스크립트 작성 인터페이스를 제공하는 제안 시스템 환경(S1)과, 제안 시스템에서 컨텍스트 기반의 실행가능 행위 기법을 제거하고 단순한 서브행위 인터페이스를 제공한 경우(S2), 그리고 제안 시스템과 유사하나 상대적으로 저수준의 객체기반 행위 작성 인터페이스를 제공하는 Alice 시스템(S3)을 이용하여 간단한 강의 시나리오 예제를 각 그룹(G1, G2)의 실험자들이 작성하도록 하였다. 객관적인 실험 환경을 위해 각 시스템의 객체 구성 및 공간적인 환경을 최대한 동일하게 구성하였다. 평가자들은 미리 받은 자연어로 작성된 강의 시나리오 인쇄물을 참고하여

두 환경에서 해당 시나리오 스크립트를 생성한다. 각 그룹은 동일한 객체들로 이루어진 제안 시스템과 비교 시스템에서 아바타 제어 인터페이스(그림 15)를 이용하여 미리 받은 인쇄물을 보고 시나리오 스크립트를 생성한다. 시나리오는 30개의 작업(Goals)로 이루어져 있으며 강의 진행에 필요한 일반적인 동작들로 이루어져 있다(표 3).

각 시스템은 사용자의 아바타 제어 입력에 따라 시나리오 스크립트를 생성한다. 평가는 실험자들에게 제시된 예제 시나리오와 실제 생성된 스크립트에 의한 아바타 동작을 비교하여 작업 달성을 측정하고 각 시스템에서 스크립트를 작성하는데 소요된 시간을 비교하였다.

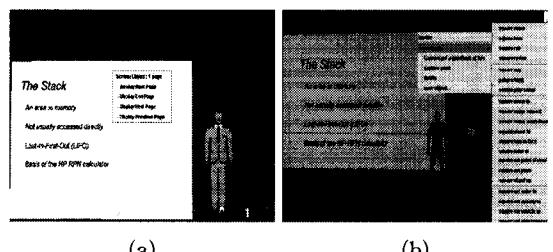


그림15. 실험을 위한 제안 시스템(a)과 Alice 시스템(b)의 인터페이스

표 3. 강의 진행을 위한 예제 시나리오

순번	작업 행위	순번	작업 행위
1	강사 인사(introduce)	16	상자 B를 탁자위로 이동
2	스크린으로 이동	17	상자 C를 탁자위로 이동
3	제목 가리킴	18	상자 A를 탁자위로 이동
4	슬라이드 넘김	19	탁자와 상자를 치움
5	첫 번째 문단 가리킴	20	슬라이드 넘김
6	두 번째 문단 가리킴	21	첫 번째 문단에 밑줄 그림
7	두 번째 문단 강조색 칠함	22	학생에게 질문 동작
8	마지막 문단 가리킴	23	학생에게 질문 내용 입력
9	두 번째 문단 강조색 지움	24	첫 번째 단락의 밑줄 지움
10	슬라이드 넘김	25	스크린의 원쪽으로 이동
11	탁자와 상자 객체 생성	26	학생에게 강의 내용 입력
12	상자 A를 이동	27	강의실 가운데로 이동
13	상자 C를 A위로 이동	28	마지막 슬라이드로 이동
14	상자 B를 C위로 이동	29	스크린을 끈다
15	첫 번째 문단 가리킴	30	강의를 종료한다

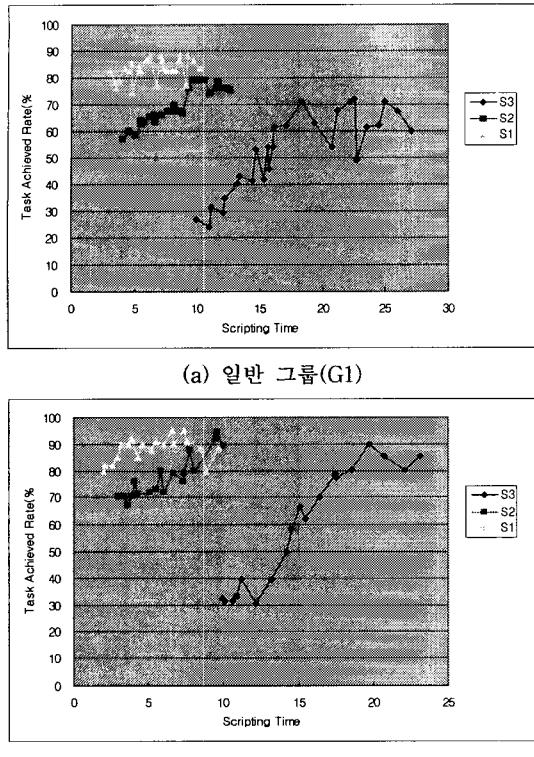


그림 16. 작업 달성을 및 소요 시간 평가 결과

두 그룹의 전체적인 결과는 그림 16와 같으며 반복측정분산분석(Repeated-measured ANOVA)을 이용하여 결과의 유효성을 검증하였다. 대다수의 사용자들은 제안 시스템을 통하여 스크립트 작성은 하였을 때 보다 평균적으로 적은 시간으로 정확도 높은 작업을 할 수 있었다($F(0.05,2,52)$, $F=41.51>4.02$). 그룹별로 살펴보면 G1의 경우에는 적은 시간을 사용한 실험자들은 S1, S2에서 상대적으로 좋은 결과가 나왔으며 실험 시간이 길어질수록 S3의 작업달성 정확도는 증가하였으나 증가폭이 미미하였다. 실험자들은 S1에서 S2에서 더 적은 시간으로 유사한 작업 달성을 보였다. G2의 경우에는 적은 시간에 실험을 끝낸 사용자의 경우 G1과 유사한 결과를 보이나 장시간 실험을 수행한 사용자들은 높은 작업 달성을 보였다.

즉, G1과 G2 그룹은 Alice 시스템과 제안 시스템에서 추상적 인터페이스를 제거한 경우 실험자의 숙련도와 소요 시간의 여부에 따라 작업 달성도가 변화함을 알 수 있었다. ($F(0.05,2,52)$, $F=23.34>4.11$). G2의 경우 시간과 정확도에 있어 서로 반비례(tradeoff)

관계가 있음을 알 수 있었다. 즉 일부전문가는 저수준 인터페이스를 이용하여 시간비용이 증가하는 대신 세밀한 작업을 통해 달성을 높였다. 그러나 제안 기법(S1)의 경우에는 두 그룹 모두 동일한 시간을 소비했을 때 작업 달성도에서 큰 차이를 보이지 않았다. 제안 시스템은 두 그룹의 대부분의 실험자들이 평균적으로 적은 시간을 이용하여 평균이상의 작업 달성을 보장 받을 수 있으며, 결과적으로 제안 시스템은 3차원 아바타 제어가 익숙하지 않은 사용자들에게 시나리오 작성을 위한 빠른 작업 인터페이스를 제공함을 알 수 있었다.

다음으로 정성적인 설문 결과는 그림 17과 같다. 이 평가는 실험이 끝난 후 각 실험자에게 표 4의 항목에 대한 설문 조사를 실시한 것이다. 각 항목의 평균은 높을 수 긍정적인 평가를 내린 것이다. 평가자들의 제안된 인터페이스에 대한 만족도는 평균 4.01로 전체적으로 비교적 높게 측정되었다. 2, 3, 5번 문항이 비교적 만족도가 낮게 측정되었으며 3번 문항이 긍정을 답한 비율이 가장 낮았다. 1, 4, 6, 7번 문항은 평균 4이상의 높은 만족도를 보이고 있다. 문항 4, 6의 경우 컨텍스트에 따라 현 상황에 적절한 행위 인터페이스를 제공하므로 메뉴가 간단하고 잘못 선택할 가능성이 적다는 응답이 많았다. 또한 7번 항목의 경우 행위 선택 시 아바타의 애니메이션 피드백으로 실험자가 작성하고 있는 스크립트가 어떻게 반영되는지 여부가 명확하게 전달되는 것으로 분석되었다. 2, 3, 5번 문항의 경우 긍정을 답한 비율이 상대적으로 낮았다. 이는 기존의 3차원 아바타 애니메이션 제어 경험자와 애니메이션 프로그램을 처음 사용해 본 평가자 사이에 의견이 다소 다르게 나왔기 때문에 분석되었다. 이는 경험자의 경우 추상화된 인터페이스를 사용하기 때문에 구체적인 아바타의 동작을 일일이 제어하지 못했다는 의견과 작업을 실행했을 때 구체적인 아바타의 동작 및 실행시간 등을 예측하지 못했다는 의견이 있었다. 그러나 비경험자의 경우에는 별다른 불만 의견이 없었다.

전체적으로는 높은 만족도를 보였기 때문에 본 실험 결과 기존 방법으로 작성하기 힘든 아바타 스크립트 작성은 보다 효율적으로 할 수 있는 방법이라는 가능성을 얻었다. 특히 아바타 애니메이션 제어 경험이 없는 초심자도 간단한 인터페이스를 통하여 완성도 있는 시나리오 스크립트 생성이 가능함을 알 수 있었다.

표 4. 설문 항목

번호	질문	점수
1	조작방법을 익힐 때 추가적인 기억의 부담이 없는가?	아니다 ①②③④⑤ 그렇다
2	조작기능을 수행하는 과정은 일관성이 있는가?	아니다 ①②③④⑤ 그렇다
3	원하는 명령을 자유롭게 지시할 수 있었는가?	아니다 ①②③④⑤ 그렇다
4	작업의 명령 과정이 충분히 간단하였는가?	아니다 ①②③④⑤ 그렇다
5	작업 명령어의 선택시 결과가 예상가능 하였는가?	아니다 ①②③④⑤ 그렇다
6	잘못 조작할 가능성이 적다고 생각되는가?	아니다 ①②③④⑤ 그렇다
7	작업 명령마다 적절한 행위 피드백이 제공되었는가?	아니다 ①②③④⑤ 그렇다

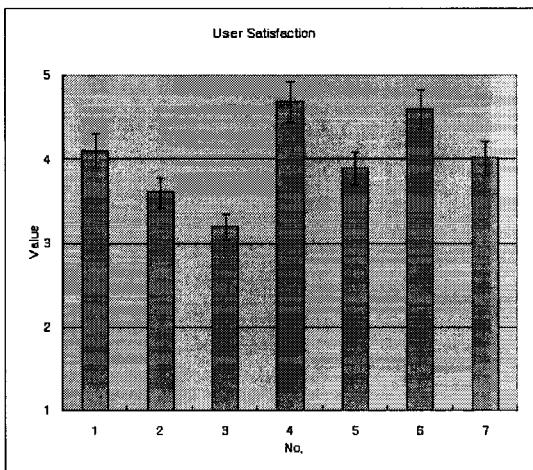


그림 17. 사용자 평과 결과

7. 결론 및 향후연구

본 논문에서는 도메인 환경에서 아바타와 객체간의 행위를 표현하고 스크립트 작성 인터페이스를 제공하기 위하여 계층적인 아바타 스크립트 언어와 컨텍스트 기반 객체 모델을 정의하였다. 제안 기법은 아바타를 제어하여 애니메이션 시나리오를 작성하는데 있어서 저수준 행위 인터페이스를 제공하는 스크립트 작성 방식보다 많은 시간과 노력을 줄일 수 있었다. 사용자 평가에서 제안 기법을 적용한 인터페이스는 저수준 행위 제어 인터페이스를 사용하였을 때 보다 약 60%에 가까운 시간을 단축하였으며 행위 생성의 정확도도 1.6배 정도 높았다. 이것은 어려운 3차원 아바타 제어에 익숙하지 않은 일반 사용자에게 작업 단위의 제어 인터페이스를 통하여 용이하게

스크립트를 작성하도록 하였기 때문이다. 다만 3차원 아바타 제어에 익숙한 일부 전문가 사용자는 세밀한 제어가 가능한 기존 저수준 인터페이스를 선호하는 경향이 있었다. 이는 향후 사용자의 수준 따라 차별적인 인터페이스를 제공하는 방향으로 개선하는 것이 필요하다. 따라서 본 기법은 일반 사용자가 3차원 아바타 콘텐츠를 쉽고 빠르게 제작하는데 효율적임을 알 수 있었다.

참 고 문 헌

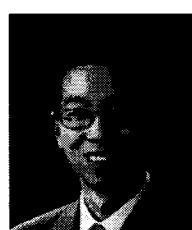
- [1] B. D. Carolis, F. Rosis, and I. Poggi, "Embodied Contextual Agent in Information Delivering Application," *In Proceedings of Autonomous Agents and Multiagent Systems*, 2002.
- [2] A. Safanova and J. K. Hodgins, "Analyzing the Physical Correctness of Interpolated Human Motion," *In Proceedings of the 2005 ACM Siggraph/Eurographics Symposium on Computer Animation*, 2005.
- [3] M. Gillies and D. Ballin, "Integrating Autonomous Behavior and User Control for Believable Agents," *In Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, Vol.1, pp. 336-343, 2004.
- [4] S. Kshirsagar, N. M. Thalmann, and Anthony Guye-Vuillème, "Avatar Markup Language," *Proceeding of the workshop on Virtual environments*, pp. 169-177, 2002.

- [5] Z. Huang, A. Eliëns, and C. Visser, "Implementation of a Scripting Language for VRML/X3D-based Embodied Agents," *Proceeding of web technology*, pp. 91-100, 2003.
- [6] Y. Arafa and A. Mamdani, "Scripting Embodied Agents Behaviour with CML," *Proceeding of Intelligent User Interfaces*, pp. 313-315, 2003.
- [7] Y. Shindo and H. Matsuda, "Design and Implementation of Scenario Language for Cyber Teaching Assistant," *International conference on Computers in Education*, 2001.
- [8] D. Thalmann, "Autonomy and Task-Level Control for Virtual Actors," *Programming and Computer Software*, No.4, 1995.
- [9] B. A. Myers, "Scripting Graphical Applications by Demonstration," *In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 534-541, 1998.
- [10] K. Perlin and A. Goldberg, "Improv: A System for Scripting Interactive Actors in Virtual Worlds," *Proceeding of Siggraph96*, pp. 205-216, 1996.
- [11] M. Hayashi, "TVML," *ACM SIGGRAPH 98 Conference on applications*, pp. 292-297, 2003.
- [12] A. Marriott and J. Stallo, "VHML- Uncertainties and Problems. A Discussion," *Proc. AAMAS*, 2002.
- [13] S. Saeyor, H. Binda, and M. Ishizuka, "Visual Authoring Tool for Presentation Agent based on Multimodal Presentation Markup Language," *Proceedings of Information Visualisation*, pp. 563-567, 2001.
- [14] X. Yangl, D. C. Petriu, and T. E. Whalen, "Script Language for Avatar Animation in 3D Virtual Environments," *IEEE International Symposium on Human-Computer Interfaces and Measurement Systems*, VECIMS '03, pp. 101-106, 2003.
- [15] L. Levison and N. I. Badler, "How Animated Agents Perform Tasks: Connecting Planning and Manipulation through Object Specific Reasoning," *In Toward Physical Interaction and Manipulation, AAAI Spring Symposium Series*, 1994.
- [16] M. Kallmann and D. Thalmann, "Modeling Behaviors of Interactive Objects for Virtual Reality Applications," *Journal of Visual Languages and Computing*, Vol.13, pp. 177-195, 2002.
- [17] W. Dann and R. Pausch, *Learning to Program with Alice*, Prentice Hall, 2006.
- [18] J. Lee, J. Chai, P. Reitsma, J. K. Hodgins, and Nancy Pollard, "Interactive Control of Avatars Animated with Human Motion Data," *ACM Transactions on Graphics, SIGGRAPH 2002*, Vol.21, No.3, pp. 491-500, 2002.
- [19] M. Kim, "A Visual Interface for Scripting Virtual Behaviors," *Proceedings of Computer Human Interaction*, pp. 165-168, 1998



김 재 경

2000년 단국대학교 화학/전산
통계 학사
2002년 연세대학교 컴퓨터과학
석사
2007년 연세대학교 컴퓨터과학
과 박사
2007년 연세대학교 컴퓨터과학
과 BK21 박사후연구원
2007년 Visiting Scholar, School of Information
Sciences, University of Pittsburgh
관심분야 : 아바타 행위 제어, Annotation 생성 및 응용



손 원 성

1998년 동국대학교 컴퓨터공학
학사
2000년 동국대학교 컴퓨터공학
석사
2004년 연세대학교 컴퓨터과학
과 박사
2004년 ~ 2006년 Carnegie Mellon
University, Associate Researcher
2006년 ~ 현재 경인교육대학교 컴퓨터교육학과 조교수
관심분야 : 웹 Annotation 생성 및 응용



임 순 범

1982년 서울대학교 계산통계학
학사
1983년 한국과학기술원 전산학
석사
1992년 한국과학기술원 전산학
박사
1989년~1992년 (주)휴먼컴퓨터

이사/연구소장

1992년~1997년 (주)삼보컴퓨터 부장
1997년~2001년 건국대학교 컴퓨터과학과 조교수
2003년~현재 숙명여자대학교 멀티미디어학과 조교수
관심분야 : 컴퓨터 그래픽스, 멀티미디어 응용, 전자출판
(폰트, 전자책, 사이버교재)



최 윤 철

1973년 서울대학교 학사
1975년 Univ. of Pittsburgh 석
사
1976년 Univ. of California,
Berkeley 석사
1979년 Univ. of California,
Berkeley 박사
1979년~1982년 Lockheed 사 및 Rockwell International
사 연구원
1990년~1991년 University of Massachusetts 교환교수
2002년~2003년 일본 게이오대학 교환 교수
1984년~현재 연세대학교 컴퓨터과학과 교수
관심분야 : 멀티미디어와 웹, 멀티미디어 문서처리, 3D
스케치 인터페이스, 아바타 인터페이스, 컴퓨
터그래픽스, eLearning 및 CyberClass