

컨텍스트 기반의 아바타-객체 행위제어 스크립트의 정의

김재경^o 최승혁 *임순범 최윤철

연세대학교 컴퓨터과학과

*숙명여자대학교 멀티미디어과학과

{ki187cm^o, alienart, ycchoy}@rainbow.yonsei.ac.kr

sblim@sookmyung.ac.kr

Definition of Context-based Script for Avatar-Object Behavior Control

Jae-Kyung Kim^o, Seung-Hyuk Choi, *Soon-Bum Lim, Yoon-Chul Choy

Dept. of Computer Science, Yonsei University

*Dept. of Multimedia Science, Sookmyung Women's University

요약

아바타는 최근 각광 받고 있는 기술로서 다양한 분야에서 많은 활용 및 발전이 기대된다. 현재 아바타 행위 표현 및 제어를 위해 다양한 기법들이 연구되고 있으나, 대부분 객체와의 상호작용이 결여된 아바타 자체의 동작을 대상으로 하고 있다. 그러나 가상환경에서 아바타를 활용하기 위해서는 여러 객체와의 상호작용을 통한 행위가 필수적으로 요구되며, 이에 대한 아바타-객체간의 행위 모델 및 스크립트에 대한 연구가 이루어져야 한다. 본 논문에서는 3D 가상환경에서 아바타의 제어를 보다 용이하게 하기 위해서 아바타-객체 행위 표현을 위해 객체기반 아바타 행위 표현 및 제어 스크립트를 정의하였다. 제안 모델에서는 단순히 객체가 모든 행위를 사용자에게 나열하는 것이 아니라, 객체 상태에 따라 사용자가 적합한 행위를 사용자에게 제공하기 위해서 컨텍스트에 기반한 객체-아바타 행위 모델을 제안하고 있다. 제안 기법을 실제 3D 환경의 프로토타입 시스템으로 구현해본 결과, 사용자는 제안 객체 모델의 컨텍스트에 따라 변화하는 객체의 행위 인터페이스를 통하여 보다 용이하게 아바타-객체간의 행위를 제어할 수 있었다.

1. 서론

사이버 캐릭터(Cyber Character) 혹은 가상 인간(Virtual Human) 등으로도 불리는 아바타(Avatar)는 최근 사용자들의 많은 관심을 불러 일으키며 포털, 게임, 교육 및 방송 등의 다양한 분야에서 활용되고 또한 관련된 많은 연구가 진행되어 왔다. 그러나 게임, 영화 및 일부 방송 산업 분야를 제외하면 아바타는 활용은 다소 제한적이며 몇몇 상업적인 시도들도 큰 성공을 거두지 못했다. 아바타 활용이 부진한 이유중의 하나는, 아바타의 행위제어가 일반 사용자에게 너무 어렵다는 것이다.

아바타 행위 제어의 가장 기본적인 방법은 Low-level 애니메이션이다. 그러나 Low-level 애니메이션 편집툴[9,10]들은 아바타의 신체 각 부분을 조작하여 동작을 생성하는 것인데 세밀한 동작의 생성이 가능하지만 이는 전문 애니메이터가 아니면 제어하기 힘든 단점이 있다.

웹 환경에서 3D 어플리케이션의 활용이 증가하면서 아바타의 행위를 일종의 명령어들로 이루어진 적은 용량의 스크립트로 제어하는 연구가 활발하다. 이들 연구는 크게 스크립트 언어를 이용한 아바타 제어와 객체 기반 아바타 행위제어 기법으로 구분할 수 있다. 스크립트 언어를 이용한 아바타 제어는 기정의된 동작 명령어와 패러미터를 아바타에게 입력하게 된다. 아바타는 명령어에 따른 저수준 동작 데이터를 호출하여 패러미터에 맞게 애니메이션을 재구성하여 사용자에게 재생하는 방법이다. 사용자는 복잡한 저수준 동작 데이터를 신경쓰지 않고 자연어 수준의 명령을 내리기 때문에 보다 쉽게 아바타를 제어할 수 있다. 그러나 대부분의 스크립트 언어가 아바타 자체의 동작만을 표현하고 제어하고 있기 때문에 다양한 객체들과의 행위 상호작용을 제어하기에는 한계가 있다.

객체기반 행위 제어는 아바타가 특정 객체에 행할 수 있는 행위들을 객체 내부에 기록하여 객체가 아바타를 제어하는 기법이다. 아바타 행

위를 객체가 제어하기 때문에 많고 복잡한 객체들로 이루어진 환경에서는 매우 효과적이다. 그러나 현재 아바타-객체와의 행위 표현을 위한 표준형식을 이용한 스크립트 언어에 대한 연구가 거의 이루어지지 않고 있으며, 아바타-객체의 복잡하고 다양한 행위들을 사용자에게 보다 쉽게 전달할 것인가에 대한 연구가 부족할 실정이다.

본 논문에서는 이와 같은 문제점들을 해결하기 위해 컨텍스트에 기반한 아바타-객체 행위 모델을 정의하고 이에 따른 스크립트를 정의하여 3D 가상공간에서 객체기반 아바타 행위제어 시스템을 구축하였다.

2. 관련연구

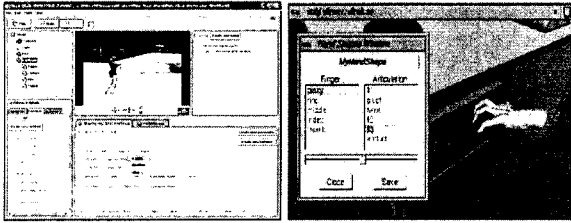
아바타 제어에는 크게 스크립트를 이용한 아바타 동작 제어 방식과 객체기반의 제어방식이 있다. 아바타 스크립트 언어는 아바타의 행위를 명령어와 패러미터 형식의 구문으로 표현 및 제어하는 것이다. 저수준 동작 데이터와 달리 아바타 관절의 DOF값 등을 고려하지 않고 walk, open 등과 같이 상위레벨의 행위명령을 통하여 아바타를 제어하기 때문에 사용자 및 기계가 이해 및 작성하기가 보다 용이하다. 또한 표준 포맷을 사용한 스크립트는 특정 구현 환경에 종속되지 않고 여러 응용프로그램에서 재사용될 수도 있다. 이와 같은 스크립트 언어에는 CML[3], AML[2], STEP[4], TVML[1] 등이 있다.

그러나 대부분의 아바타 스크립트 언어들은 객체와의 인터랙션을 전혀 고려하지 않거나 제한적으로 표현하고 있어 수많은 객체로 이루어진 가상 환경에서 아바타를 효과적으로 이용하기에 한계가 있다.

객체기반 아바타 제어 기법은 객체를 이용하여 아바타 및 객체 자신의 행위를 표현하고 재생하는 것이다. 아바타의 행위는 대상이 되는 객체에 따라 그 종류와 숫자가 매우 늘어날 수 있다. 따라서 모든 동작을 하나의 아바타 행위 모델로 표현 하는 것은 매우 힘든 일이기 때문에 객체 내부에 아바타가 해당 객체와 상호작용 시 필요한 행위들을 표현해 두는 것이다. 따라서 아바타는 걸기나 말하기등과 같은 기본적인 행위 모델만을 가지고서 어떤 객체와 상호작용을 할때는 해당 객체의 행

본 연구는 한국과학재단 목적기초연구 (R01-2004-000-10117-0(2004)) 지원으로 수행되었음

위 모델을 이용하여 여러 가지 행위를 수행할 수 있는 장점이 있다. 대표적인 연구로 Alice[5], 스마트 오브젝트[6,7], OSR[8] 등이 있다.



[그림1] 객체기반 아바타 행위 제어기법의 예

그러나 기존 연구들은 행위모델 표현에 있어 표준 형식에 기반하지 않고 자체 포맷을 사용함으로써 다양한 환경에 적용이 어렵다. 또한 객체는 상황에 따라 아바타에게 여러 가지 다양한 행위를 제공해 줄 수 있는데 예를 들어 문(Door) 객체의 경우 닫혀있을때와 열려있을때 사용할 수 있는 행위가 다르다. 그러나 기존연구는 항상 고정된 행위 인터페이스만을 제공하기 때문에 사용자에게 환경 정보(Context)에 따른 행위 인터페이스 모델을 제공해주는 것이 바람직하다.

3. 컨텍스트기반 객체 행위 모델 및 스크립트

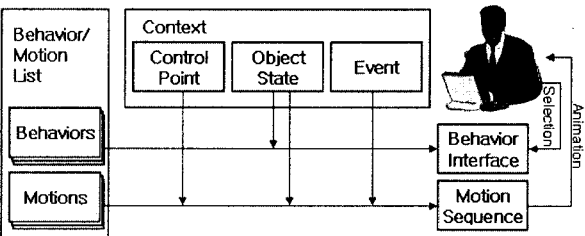
3.1 컨텍스트 기반 객체 모델

상위 레벨에서 아바타의 행위를 제어할 때 발생하는 문제점은 모호성이다. 본 연구에서는 공간, 행위 및 시간 모호성을 정의하였다.

공간 모호성은 아바타가 객체와 상호작용을 할 때 아바타의 구체적인 위치를 명시하지 않음으로써 발생된다. 예를 들어 "go to table"이란 스크립트 명령을 아바타에게 주었을 때, 아바타가 정확히 table의 어느 위치로 갈 것인지 결정해야 한다. 즉 table 중심점, 앞 혹은 옆쪽 등 구체적인 위치가 지정되지 않기 때문이다.

행위 모호성은 같은 행위라도 객체의 종류나 상태에 따라 실제 조작 동작이 달라지는데서 발생한다. 예를 들어 "enter the door"의 경우, 문의 개폐 상태에 따라 문을 열고 들어가거나 그냥 걸어 들어갈 수 있는데, 상황에 따라 어떤 동작을 취할 것인지 객체 상태에 따라 동작을 지정해줘야 한다.

시간 모호성은 아바타가 다양한 객체와 여러 동작을 수행함에 따라 객체 자체의 동작과 아바타의 동작의 동기화가 이루어져야 자연스러운 애니메이션을 사용자에게 보여줄 수 있는데, 상위 레벨에서는 단지 "open the door"와 같은 행위의 목적만을 주어야지 아바타-객체간의 행위 동기화를 어떻게 할 것인지에 대한 정보가 없으므로 동작의 동기화 시간의 모호성이 발생한다.



[그림2] 객체 모델의 전제구조 및 작동 과정

본 논문에서는 이와 같은 모호성을 제거하기 위해 컨텍스트를 정의하고 이를 이용하여 아바타-객체간의 명확한 행위 인터랙션이 가능하도록 한다. 컨텍스트는 공간, 객체 상태, 동기화 컨텍스트로 크게 3가

지로 정의하였다.

이는 다음 [그림2]에 나타나 있으며 사용자는 컨텍스트에 따른 행위 인터페이스에서 원하는 행위를 선택하고, 이는 다시 컨텍스트에 따라 현재 객체 상태에 적절한 동작 시퀀스로 구성되어 사용자에게 객체-아바타 애니메이션을 재생하여 보여준다

● 객체 컨텍스트

① 공간 컨텍스트(Spatial Context)

공간 컨텍스트는 객체 주변의 임의의 제어점들로 정의되며 아바타가 위치하여 객체를 조작하게는 지점이다. 객체가 가지는 행위와 구조에 따라서 제어점은 다수가 될 수 있으며 각 제어점은 3가지 요소(Element)를 가진다. 각 특성은 아바타의 위치 및 방향, 그리고 객체에서 아바타의 조작이 행해지는 부분이다.

• 아바타의 위치(Position) 요소

객체주변의 전후좌우 및 중심 5개 지점과 사용자 정의의 지점을 추가로 정의할 수 있다.

• 아바타의 방향(Direction)

전후방향 및 좌우, 그리고 객체 방향의 5가지 기본 방향과 사용자 정의의 아바타 방향을 지정한다.

• 조작 부분(Part)

아바타의 객체 조작시, 대상이 되는 객체의 부분을 지정한다.

Element	Values
Position	front, behind, left, right, center, user_defined _n
Direction	object, backward, forward, left, right
Part	part ₁ , ..., part _n

[그림2] 공간 컨텍스트의 구성요소

② 객체 상태 컨텍스트(Object State Context)

객체 및 객체가 속한 환경 정보를 나타내는 것으로 현재 상태에 따라 사용자에게 적절한 행위 인터페이스를 제공한다. 또한 행위 명령을 수행하기 위한 동작들을 조합할 때 객체 상태에 따라 동작 구성에 영향을 미친다. 제안 모델에서는 <state> 태그내에 <variables> 요소를 두어 사용자가 다양한 객체의 상태를 표현 할 수 있도록 상태명과 상태값을 자유롭게 정의할 수 있도록 하였다.

③ 동기화 컨텍스트(Synchronization Context)

아바타와 객체간의 동작을 동기화하기 위해서 제안 모델에서는 이벤트 방식을 이용한다. 임의의 아바타-객체의 행위는 복수의 아바타 및 객체의 동작 시퀀스로 구성된다. 이때 아바타의 특정 동작이 발생되면 이벤트 메시지가 객체에 전달되어 대응되는 객체의 동작이 실행된다. 이를 위해 <event> 태그에서 아바타의 이벤트 발생 동작과 이벤트 수신 객체 동작을 지정할 수 있도록 하였다.

● 행위 모델

제안 행위 모델에서는 사용자에게 행위 인터페이스를 제공하기 위해 객체는 사용가능한 행위 리스트를 가지고 있다. 또한 각 행위는 아바타와 객체의 동작 시퀀스로 구성되어 하나의 행위를 수행하기 위한 동작들을 생성한다.

① 행위 인터페이스(Behavior Interface)

행위 인터페이스는 사용자가 아바타 행위 제어를 위해 객체에 접근할 때 보여지는 행위들로 구성된다. 행위 인터페이스는 현재 객체 상태 컨텍스트(C₀)를 패러미터로 가지는 행위명으로 구성되며 이중 현재 컨텍스트와 일치하는 패러미터를 가진 행위를 사용자에게 선택적으로 제공한다.

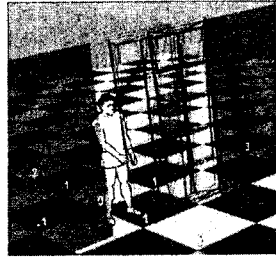
② 동작 시퀀스(Motion Sequence)

행위는 다수의 동작으로 구성된다. 예를 들어 닫혀있는 문에 'enter the door'의 행위를 지시하면 아바타는 'walk, grab knob, pull knob,

walk, turn, grab knob, push knob'의 동작 시퀀스로 움직이게 된다. 그러나 문이 열려있는 경우 아바타는 문을 여는 동작을 생략하고 걸어서 들어가게 되는데, 제안 모델에서는 앞서 정의한 객체 상태 컨텍스트에 따라 이를 판단할 수 있다. 또한 객체를 조작하기 위한 제어점에서 아바타가 동작하도록 공간 컨텍스트(C_s) 정보를 제공한다.

다음 [그림5]는 사용자가 'open' 명령을 내렸을때 아바타와 문의 상호작용 결과를 보여주는 예이다. 아바타의 손동작과 문이 열리는 동작이 동기화되어 아바타-객체간의 자연스러운 동작을 재생한다.

4. 구현 결과



[그림 5] 문을 열고 들어가는 아바타-객체 인터랙션

4.1 컨텍스트 기반의 객체 행위 스크립트

제안 객체 모델은 XML 언어를 이용하여 스크립트로 정의하였으며 [그림3]은 Door 객체에 대한 스크립트 예이다. <Context> 태그에서는 앞서 언급한 컨텍스트 요소들을 표현하고 있으며 <Interface> 및 <Sequence>에서는 컨텍스트 요소들을 인자로 받아 행위 인터페이스 표현 및 동작들을 스크립트로 표현하고 있다.

```
<?xml version="1.0" encoding="UTF-8"?>
<Object id="e1" name="Door" resource="door.wrl">
  <Context>
    <ControlPoint name="p1">
      <Position value="front"/>
      <Direction value="object"/>
      <Part value="part1"/>
    </ControlPoint>
    <State>
      <Variable id="v1" name="door_closed" value="true"/>
      <Variable id="v2" name="door_closed" value="false"/>
    </State>
    <Event>
      <motion object="object_open" avatar="open"/>
      <motion object="object_close" avatar="close"/>
    </Event>
  </Context>
  <Interface>
    <Behavior name="enter_door" state="ANY"/>
    <Behavior name="open_door" state="v1"/>
  </Interface>
  <Sequence name="enter_door" state="v3">
    <motion name="walk" cpoint="p1" state="ANY"/>
    <motion name="open" cpoint="p1" state="v1"/>
  </Sequence>
</Object>
```

[그림3] Door 객체의 스크립트 예

5. 결론 및 향후 방향

객체와 아바타간의 행위 인터랙션을 표현 및 제어하기 위해서 본 논문에서는 컨텍스트 기반의 아바타-객체 모델 행위 모델을 제시하였다. 컨텍스트는 공간, 객체 상태 및 동기화 요소로 이루어져 있으며 이들은 상위레벨의 스크립트 명령을 명확하게 처리하기 위해서 객체에 내장되어 아바타-객체행위를 제어하는데 사용된다. 또한 객체에 정의된 행위들은 이들 컨텍스트에 따라 상황에 맞는 동작을 사용자에게 제공함으로써 보다 용이하게 객체를 이용하여 아바타의 행위를 제어할 수 있도록 하였다.

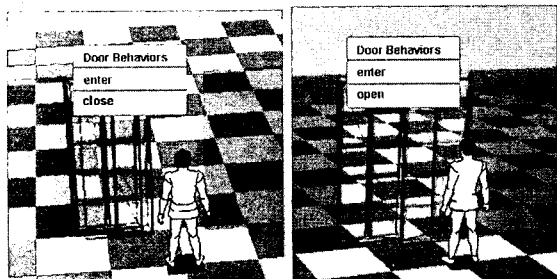
현재 본 연구는 개발 초기 단계로서 다양한 객체와의 행위 인터랙션을 지원할 수 있는 스크립트 구조 정의와 검증이 필요하겠다. 향후 연구로는 XML 기반으로 정의된 스크립트의 재사용성을 높이기 위해서 계층적인 스크립트 구조를 정의하여 제안 스크립트를 다양한 구현 환경에 적용하는 것이다.

참고 문헌

1. Masaki Hayashi, TVML (TV program making language), ACM SIGGRAPH 98, (1998), pp.292-297
2. S. Kshirsagar, et al, E. Mamdani, Avatar Markup Language, Proc. Eurographics, EGVE (2002), pp.169-177
3. Yasmine Arafa, Abe Mamdani, Scripting embodied agents behaviour with CML, Proc. IUI, (2003), pp.313-316
4. Zhisheng Huang, et al, Implementation of a scripting language for VRML/X3D-based embodied agents, Proc web technology, (2003), pp.91-100
5. Wanda Dann , Toby Dragon , Stephen Cooper , Kevin Dietzler , Kathleen Ryan , Randy Pausch, Objects: visualization of behavior and state, ACM SIGCSE Bulletin, v.35 n.3, September 2003
6. M. Kallmann, D. Thalmann, Direct 3D Interaction with Smart Object, Proc. ACM VRST 99, London
7. M. Kallmann and D. Thalmann, Modeling Behaviors of Interactive Objects for VR Applications, JVL, Vol. 13, pp.177-195, 2002
8. L. Levinson, "Connecting Planning and Acting: Towards an Architecture for Object-Specific Reasoning", PhD thesis, University of Pennsylvania, 1996.
9. 3DS Max, <http://www4.discreet.com/3dsmax>
10. Poser5, <http://www.curiouslabs.com/>

4.2 시스템 구현

시스템 구현은 Java3D API를 이용하여 구현하였으며 현재 프로토타입 단계로서 아바타와 Door 객체로 구성되어 있다. [그림4]에서 보듯이 문의 개폐 상태에 따라 사용자가 문을 선택하였을때 보여지는 컨텍스트 메뉴의 구성이 틀려진다. 문의 개폐여부에 상관없이 아바타는 항상 문에 대하여 'enter' 행위를 할 수 있으며 'open' 및 'close'는 문의 상태에 따라 사용자에게 선택적으로 표시된다.



[그림4] 문의 상태에 따라 가변하는 컨텍스트 행위 메뉴