

계층적 접근 방식의 아바타 행위 표현 및 제어 스크립트

김재경^o 최승혁 오재균 *임순범 최윤철

연세대학교 컴퓨터과학과

*숙명여자대학교 멀티미디어학과

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

sblim@sookmyung.ac.kr

Avatar Behavior Representation and Control : A Layered Scripting Language Approach

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

Dept. of Computer Science, Yonsei University

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

요 약

아바타는 최근 각광 받고 있는 기술로서 다양한 분야에서 많은 활용 및 발전이 기대된다. 그러나 현재 아바타 행위 표현 및 제어에는 체계적이고 표준적인 접근 방식이 결여되어 있어 아바타 행위 표현의 어려움이 있는 실정이다. 본 논문에서는 다양한 도메인 환경에서 아바타의 제어를 보다 용이하게 하기 위해서 아바타 행위 표현의 계층적 접근 방식을 제안하였다. 이를 위해 다양한 도메인 환경에서 사용되는 작업레벨 행위 모델을 정의하여 사용자가 특정 도메인에서 아바타에게 필요한 작업레벨 행위들을 할당하도록 하고, 이것은 아바타의 동작을 특정 도메인이나 구현환경과 독립적 및 추상적으로 설계한 상위레벨 동작으로 표현된다. 마지막으로 상위레벨 동작은 아바타 엔진이 제어할 수 있는 기본 동작으로 표현되어 다양한 구현 환경에 적용된다. 이와 같은 접근 방식을 통하여 사용자 측면에서 보다 쉽게 아바타의 행위를 제어할 수 있으며, 계층적인 표현 및 제어방식으로 스크립트의 호환성, 확장성 및 재사용성을 높였다.

1. 서 론

최근 아바타 활용과 관련된 많은 연구가 진행되었음에도 불구하고, 게임, 영화 및 일부 방송 산업 분야를 제외하면 아바타는 활용은 다소 제한적이며 몇몇 상업적인 시도들도 큰 성공을 거두지 못했다. 아바타 활용이 부진한 이유를 몇가지로 요약하면, 먼저 아바타의 행위제어가 일반 사용자에게 너무 어렵다는 것이다.

아바타 행위 제어의 가장 기본적인 방법은 Low-level 애니메이션이다. 3DS Max나 Poser와 같은 편집툴에서 아바타의 신체 각 부분을 조작하여 동작을 생성하는 것인데 세밀한 동작의 생성이 가능하지만 이는 전문 애니메이터가 아니면 제어하기 힘든 단점이 있다.

최근에는 웹기반 프로그램이 활성화되면서 아바타의 행위를 일종의 명령어들로 이루어진 스크립트로 제어하는 연구가 활발하다. 그러나 이들 스크립트가 현재 아바타가 속해있는 도메인의 컨텍스트를 인식하지 않으며, 소프트웨어 툴에 따라 그 사용법이 상이하기 때문에 스크립트 작성에 어려움을 겪는다. 또한 작성된 스크립트의 아바타 행위나 주변 환경에 대한 정보들이 특정 소프트웨어의 물리적인 환경에 종속적이기 때문에 ऐसे 작성한 아바타 행위제어 스크립트가 1회성 정보로 끝나버리는 문제점이 있다.

이와 같은 문제점들을 해결하고자 많은 연구들이 진행되었는데 크게 작업 기반 아바타 제어와 상위레벨 스크립트 언어 및 객체 기반 행위제어로 구분할 수 있다. 작업기반 아바타 제어는 특정 작업 혹은 목표를 아바타에게 할당하면 자동적으로 행위를 조합하여 작업을 수행하기 때문에 아바타 제어가 간단하고 용이하다. 그러나 작업을 수행하기 위한 아바타의 행위가 시스템에 의해 대부분 자동적으로 생성되기 때문에 사용자의 의도대로 아바타를 제어할 수 있는 여지가 적고, 특정 소프트웨어의 아바타 엔진에 의존하고 있어 재사용성이 저하된다.

상위레벨 스크립트 언어는 소프트웨어에 독립적인 아바타 행위 표현으로 확장성이나 재사용성에 중점을 두고 있다. 그러나 아바타의 행

위 표현을 위한 복잡하고 다양한 파라미터와 기능들 때문에 실제로 스크립터가 스크립트를 작성하여 아바타를 제어하기는 다소 까다롭다.

객체기반 행위 제어는 아바타가 특정 객체에 행할 수 있는 행위들을 객체 내부에 기록하여 객체가 아바타를 제어하는 기법이다. 아바타 행위를 객체가 제어하기 때문에 많고 복잡한 객체들로 이루어진 환경에서는 매우 효과적이거나 객체와 인터랙션이 없는 아바타 자체 행위 제어가 어렵고 한 객체에 대한 모든 아바타의 행위가 동일하여 단순로운 결과를 보여준다.

[표 1] 아바타 행위 표현 및 제어 기법

접근 방식	제어 기법	특징	한계점
Low-level Animation [11]	아바타의 신체 부위들에 대해 이동, 회전, 크기를 직접 지정	세밀하고 자연스러운 동작 제어가 가능	전문 애니메이터가 아니면 제어가 힘들며 고정된 동작만 지원
High-level scripting language [1,2,3,4,5]	파라미터를 통하여 행위들의 동기화, 강도 등을 조절	아바타 엔진과 애니메이션 틀간의 독립적인 아바타 행위 표현	복잡한 파라미터 및 싱크로 인해 스크립트 작성성이 어려움
Intelligent object [10]	아바타의 행위를 객체에 저장하여 아바타를 제어	아바타 행위의 재사용성 및 자동화	객체에 대한 아바타 행위만 지원하며 한 객체는 모든 아바타에 동일한 행위만 제공함
Task-oriented system [6,7,8,9]	작업 혹은 목표를 할당하면 아바타가 자동적으로 행위 수행	간단하고 용이한 사용자 인터페이스 제공	작업을 위한 행위의 자동화로 다양한 행위 스크립팅이 어렵고 소프트웨어에 종속적

본 논문에서는 가상환경에서 계층적인 접근방식을 통한 아바타 행위 제어 및 표현 기법을 제안한다. 사용자에게는 작업레벨 행위 수준의

본 논문은 2003년도 산업자원부의 지원에 의하여 연구되었음

스크립팅 인터페이스를 제공하여 아바타 행위 제어를 용이하게 하고, 작성된 아바타 행위는 아바타 엔진과 독립적인 상위레벨 동작으로 표현되어 다양한 소프트웨어에 적용이 가능하다. 상위레벨 동작 스크립트는 마지막 단계인 기본 동작 스크립트로 변환되고 이는 아바타 엔진이 지원하는 Low-level 애니메이션을 호출하여 아바타의 애니메이션이 수행된다.

이와 같은 방식을 통하여 아바타 행위 제어 스크립트 작성을 용이하게 하고 다양한 구현 환경에 적용이 가능하도록 하였으며, 아바타 행위의 재사용성 및 확장성을 추구하였다.

2. 계층적 스크립트 언어

본 논문에서는 계층적인 아바타 행위 제어를 위해 행위 표현 및 제어언어를 정의 하여 아바타 행위 제어의 높은 확장성, 호환성 및 재사용성을 목적으로 한다. 행위 표현 및 제어 언어는 특정 도메인 환경에서 사용자가 작성하는 아바타 작업레벨 행위 언어, 아바타 애니메이션 소프트웨어 혹은 엔진이 지원하는 기본 동작을 표현하는 기본 모션 언어, 그리고 이들 도메인이나 소프트웨어 환경에 독립적으로 아바타 동작을 표현하는 상위레벨 언어로 구성된다[표2].

[표 2] 제안 언어의 특성

계층별 언어명	특성
Task-Level Behavior	사이버교육, 쇼핑물 등과 같은 특정 도메인에서 필요한 작업행위들로 구성. 도메인에 종속적 및 아바타 엔진에 독립적.
High-Level Motion	일반 아바타의 동작을 추상적 패러미터로 표현 추상적인 표현 및 제어 언어. 도메인에 독립적 및 소프트웨어에 독립적
Primitive Motion	특정 소프트웨어의 아바타 엔진 혹은 동작 라이브러리가 지원하는 기본 동작. 물리적인 표현 및 제어 언어. 도메인에 독립적 및 소프트웨어에 종속적

2.1 작업레벨 행위 스크립트 언어

제안 시스템은 아바타의 행위 제어를 계층적으로 표현하고 있다. 이중 가장 상위 계층은 작업레벨 행위 스크립트 계층으로 사용자가 아바타를 제어하기 위한 스크립트를 작성하는 인터페이스를 제공한다. 제안 기법에서 용이한 스크립트 작성 인터페이스를 작성하기 위한 두가지 요소는 다음과 같다.

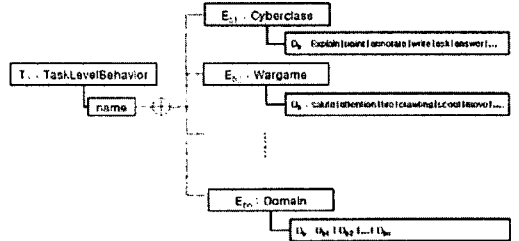
- Abstraction: 구현물 환경의 세부적인 속성들을 일일이 고려하지 않아야 한다
- Domain Dependency: 스크립트가 도메인 컨텍스트에 강하게 종속되어 있어야 한다.
- Extensibility: 다양한 도메인에 적용 가능 및 사용자 정의의 행위가 지원되어야 한다.
- Reusability: 스크립트는 여러 환경에서 재사용이 되어야 한다

먼저 추상적인 스크립트일수록 작성이 용이해진다. 예를 들어 아바타에게 '탁자위의 책을 집어라'라는 행위 명령은 탁자까지 몇 걸음을 몇도의 방향으로 걸어가서 탁자 주변 어느 좌표에 멈추어 서서 팔을 어느 위치까지 들어올리고 하는 식의 구현물에 종속된 행위 명령보다 스크립터에게 친숙한 형태의 명령이다.

다음으로 일반적인 아바타 행위 및 상호작용요소들은 그 종류가 많고 다양하기 때문에 현재 도메인 컨텍스트에 종속적인 것만을 사용자에게 제공해주는 것이 효율적이다. 또한 스크립트 자체는 도메인에 종속적이라도 스크립트 정의의 구현 구조는 다양한 도메인에서 사용될 수 있도록 일반적이어야 하며 작성된 스크립트는 현재 도메인내의

여러 구현환경에서 사용이 가능하여야 한다. 이와 같은 디자인 개념에 따라 작업레벨 행위 스크립트 언어를 XML로 정의하였는데 XML은 재사용성 및 확장성, 그리고 계층적이고 구조적인 아바타의 행위 표현에 적절하다. 작업레벨 스크립트의 구조 및 특성은 다음과 같다.

먼저 작업레벨 행위는 도메인마다 그 종류가 다르다. 제안 시스템에서는 특정 도메인의 행위의 집합을 엔터티로 모델화하여 제공하고 이는 <TaskLevelBehavior> 태그를 통하여 호출된다. 따라서 사용자는 현재 도메인에 해당되는 행위 집합 엔터티를 불러 사용한다. 즉 작업이 T, 행위가 B, Entity를 E라고 할 때 $T_b = E_{b1}|E_{b2}|...|E_{bn}$ 이며 각 E는 도메인 종속 행위인 $D_{b1}|D_{b2}|...|D_{bn}$ 으로 구성된다[표1].



[그림 1] 도메인별 작업레벨 행위 스크립트의 구조

작업은 어떤 객체에 대한 행위나 목적이 있는 행위로 정의된다. 즉 행위가 현 도메인의 컨텍스트와 인터랙션을 일으키는 것이 작업이다. 행위와 마찬가지로 도메인 컨텍스트 역시 엔터티 모듈로 제공되며 외부 XML 구조로 정의된다. 도메인 컨텍스트 $C = E_{c1}|E_{c2}|...|E_{cn}$ 이며 $E_c = XPath(external_XML)$ 로 접근된다.

또한 행위의 동기화를 위해 SMIL의 <Seq> 및 <Par> 태그를 사용하여 $\langle Seq \rangle T_{b1}|T_{b2}|...|T_{bn} \langle /Seq \rangle$ 및 $\langle Par \rangle T_{b1}|T_{b2}|...|T_{bn} \langle /Par \rangle$ 와 같이 각각 $T_{b1..n}$ 을 순차적 혹은 병렬로 수행한다.

그리고 행위의 다양한 제어를 위해 속도, 강도, 지연과 같은 패러미터 엘리먼트(P)들을 적용할 수 있다. 제안 언어에서는 XML의 구조적인 특성을 이용하여 패러미터 태그의 하위 노드들은 모두 상위 패러미터의 속성에 영향을 받는다. 즉 P_{scope} 은 P를 루트로 하는 DFS(P)를 통하여 결정된다.

작업은 다수의 행위의 집합으로 이루어 질 수 있다. 엔터티에서 제공되는 행위 외에 사용자 정의의 행위를 조합하거나 자주 수행하게 되는 작업을 정의하여 사용할 필요가 있다. 이 경우 <DEF> 태그를 이용하여 다수의 행위 조합으로 이루어진 사용자 정의의 행위를 생성하고 컨텍스트를 패러미터로 전달하여 정의된 행위를 재사용한다. 즉 $U_i = DEF(T_b|C_1|...|C_n|P|U_i)^*$ 로 표현된다.

다음 예는 앞서 언급한 요소들을 이용하여 작성한 강의 스크립트의 초기부분이다.

```
<TaskLevelBehavior name="introduce" target="students"/>
<TaskLevelBehavior name="point" target="screen/page/title"/>
<TaskLevelBehavior name="talk_gesture">Today we are going to learn about %screen/page[1]/title[1];</TaskLevelBehavior>
<DEF name="introduce" target="c1">
<Seq>
<Intensity degree="strong">
<TaskLevelBehavior name="greet"/>
</Intensity>
<Par>
<TaskLevelBehavior name="walk" target="c1"/>
<TaskLevelBehavior name="talk_gesture">Nice to meet you ! My name is Avatar</TaskLevelBehavior>
</Par>
</Seq>
</DEF>
```

다음 절에서는 도메인에 독립적인 상위레벨 동작 스크립트의 구조와 도메인 컨텍스트에 종속된 작업레벨 행위 스크립트를 상위레벨 스크립트로 변환하는 기법에 대해서 논한다.

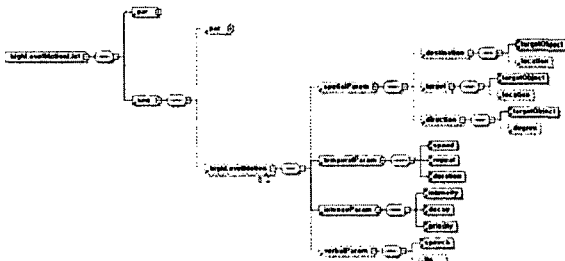
2.2 상위레벨 동작 스크립트 언어

아바타 상위레벨 모션은 특정한 도메인이나 구현물 환경에 종속되지 않고 일반적으로 아바타가 가질 수 있는 추상적인 표현 방식의 동작이다. 상위레벨 모션은 모션의 속도, 강도, 방향 등과 같은 인자(Parameter)를 이용하여 아바타의 모션을 표현 및 제어한다.

상위레벨 모션은 먼저 아바타의 기본정보인 명칭, 역할, 성별 등을 표현하고 하나이상의 동작 리스트로 구성된다. 동작 리스트를 구성하고 있는 동작 요소는 동작의 명칭과 이에 대한 패러미터들로 이루어지는데, 패러미터는 크게 공간 요소, 시간 요소, 강도 요소 및 청각 요소로 구성된다.

먼저 공간요소는 아바타의 이동 목적지를 나타내는 Destination 속성, 손 및 발 제스처의 목표 지점 등을 지정하는 Target 속성, 아바타의 방향을 표현하는 Direction 속성으로 구성된다. 시간 요소는 아바타의 속도(Speed), 동작의 지속 시간(Duration), 동작의 반복 횟수(Repeat), 그리고 순차 및 병렬(Seq/Par) 수행 속성으로 구성되어 있다. 다음으로 강도 요소는 아바타 동작의 강조 혹은 변조를 위한 Intensity 속성이며, 마지막으로 청각요소에서는 아바타의 음성 및 효과음 출력을 위한 스피치(Speech) 정보를 표현한다[그림2].

이와 같은 요소들은 아바타의 모션을 작업 도메인(Task Domain)이나 구현물 환경(Implementation Environment)과 독립적 및 추상적으로 표현하는 것으로 하위 계층인 기본 모션에서 구현물에 종속적인 물리적 표현 방식으로 변환되어진다.



[그림 2] 상위레벨상위레벨 동작 언어의 구성

2.3 기본 동작

기본 모션은 아바타 엔진 혹은 모션 라이브러리가 가지고 있는 아바타 모션들을 표현한다. 상위레벨이 일반적이고 추상적인 개념으로 아바타의 모션을 표현하는 반면, 기본 모션은 특정 아바타 엔진 및 모션 라이브러리에 따라 그 종류와 표현이 달라지며, 표현 방식도 구현 환경에 종속되어 물리적인 형태가 된다. 예를 들어 상위레벨 모션에서는 <go to="table">과 같은 모션이 기본 모션 단계에서는 <walk_to x="100" y="12" z="-2">와 같은 형식으로 표현된다.

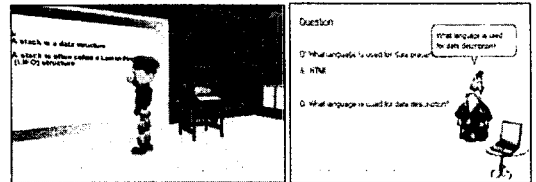
본 과제에서는 OpenGL로 구현된 3D 아바타 모션엔진과 MS Agent 기반의 2D 아바타가 가지는 모션 라이브러리를 표현하기위한 각 기본 모션 스크립트를 XML 형식으로 정의하였다.

3. 구현 결과

구현 결과에서는 위와 같은 기법을 적용하여 도메인 환경에서 사용자가 아바타 제어 시나리오를 작성하고 이는 다양한 가상환경 구조 및 구현 환경에 적용될 수 있는 아바타의 계층적 행위 제어 시스템을 구축하였다. 이는 기존의 아바타 제어 언어들이 특정한 가상환경 혹은 시스템 환경에 종속되어 아바타 행위 언어 스크립트의 재사용성 및 확장성이 떨어지던 단점을 보완하였고, 작업레벨에서 아바타 행위를 표현하여 사용자의 아바타 제어를 보다 용이하게 하였다.

[그림3]은 앞서 언급한 스크립트들에 의해 사이버클래스 환경에서

아바타가 강의를 진행하고 있는 예이다. 사용자에 의해 작성된 작업레벨 행위 스크립트가 하나의 추상적 표현 방식의 상위레벨 모션 스크립트로 표현된다. 다시 이것이 각 구현물의 아바타 엔진이 가지는 기본 모션으로 번역되고, 또한 추상적 패러미터들이 해당 환경에 맞는 물리적 패러미터로 변환되어 강의 애니메이션을 생성하여 보여주고 있다



(a) 3D 가상공간 환경(OpenGL) (b) 웹 환경 (MSAgent)

[그림 3] 사이버교육 환경에서 스크립트 실행 예

4. 결론 및 향후 연구

본 논문에서는 다양한 도메인 및 구현환경에서 아바타의 용이한 제어를 위해, 행위의 표준적인 표현방식을 위해 작업레벨 행위, 상위레벨 모션 및 기본 모션의 3가지 계층의 스크립트 표현을 통한 아바타 행위 표현 및 제어 기법을 제안하였다.

이와 같은 접근 방식을 통하여 사용자 측면에서 보다 작업레벨에서 보다 용이하게 아바타의 행위를 제어할 수 있으며, 상위레벨 및 기본 모션에서는 추상적 및 물리적인 표현 방식을 통하여 구현환경에 종속되지 않고 재사용성 및 확장성이 뛰어난 아바타 모션 제어 및 표현 방식이 가능하게 하였다.

향후에는 제안된 기법을 기반으로 하여 아바타의 작업 스크립트 입력을 위한 그래픽 사용자 인터페이스 개발 및 사용자 작업의 녹화 및 아바타와 사이버 환경내의 객체간의 상호작용을 통한 아바타 제어 기법등이 요구된다.

관련 연구

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. Marriott, A. & Stallo, J., VHML- Uncertainties and Problems ... A discussion." Proc AAMAS, Bologna, Italy. (2002)
5. Zhisheng Huang, et al, Implementation of a scripting language for VRML/X3D-based embodied agents, Proc web technology, (2003), pp.91-100
6. James C.Lester, et al, Explanatory Lifelike Avatars, Autonomous Agents, (1999)
7. Jeff Rickett, et al, Task-Oriented Collaboration with Embodied Agents in Virtual Worlds, Embodied Conversational Agents, MIT Press, (2000)
8. Elisabeth Andre, et al, WebPersona : A Life-like Presentation Agent for the WWW, International Multimedia Conference, (1998) pp.108121
9. D.Thalmann, Autonomy and Task-Level Control for Virtual Actors, Programming and Computer Software, No4, (1995)
10. M. Kallmann and D. Thalmann, Modeling Behaviors of Interactive Objects for VR Applications, JVLC, Vol. 13, pp.177-195, 2002
11. Humanoid Working Group, http://www.h-anim.org