

행위 표현 언어를 이용한 CyberClass 아바타 시스템의 구현¹

김정희*, 김재경**, 최윤철**, 임순범*
*숙명여자대학교 멀티미디어학과
**연세대학교 컴퓨터학과

CyberClass Avatar System using Behavior Description Language

Jung-Hee Kim*, Jae-Kyung** Kim, Yoon-Chul Choy**, Soon-Bum Lim*
*Dept. of Multimedia Science, Sookmyung Women's University

**Dept. of Computer Science, Yonsei University

요 약

최근 웹과 가상환경에서는 사실감과 몰입감을 증대시키고 사용자와 상호작용 할 수 있는 아바타의 연구가 활발히 이루어지고 있으나, 아바타 동작의 제어를 위한 기존의 연구들은 하위 동작과 관련된 복잡한 데이터를 포함하고 있어서 사용자가 작성하기에는 난해한 점이 많았고, 다양한 작업 도메인에 적용할 때에는 해당 도메인에 맞게 언어를 재 작성해야 하는 문제점이 있었다.

따라서 본 연구에서는 사용자가 작성하기 쉬운 형태로 정의된 작업수준의 행위 표현 언어와 작성된 행위의 세부 동작들을 위한 동작 표현 언어를 각각 정의하고, 행위 표현 언어를 동작 표현 언어로 변환시키기 위해 Interpreter를 구현하였으며, 아바타의 모션라이브러리를 구축하여 CyberClass 아바타 시스템을 구현하였다. 또한 제안된 행위 표현 언어를 기존 연구와 비교하여, 행위 표현 언어가 아바타의 행위에 대해 좀 더 간단하게 작성 되는 것을 알 수 있었다.

1. 서론

웹 환경이나 가상환경(Virtual Reality)에서는 아바타의 연구가 활발히 이루어지고 있는데 가상환경에서 사실감과 몰입감을 증대시키고 사용자와의 상호작용을 효율적으로 할 수 있다는 장점이 있기 때문이라고 할 수 있다. 아바타의 활용이 늘어남에 따라 아바타 동작의 제어와 관련된 연구도 다양하게 이루어졌는데, 그중에서 시나리오를 통한 아바타 동작 제어방식은 간단한 방법으로 아바타를 제어할 수 있으므로 이에 대한 연구가 다양하게 있었다. 그러나 긴 시나리오 문장을 사용자가 직접 입력해야하고 제어 언어가 하위의 동작과 관련된 복잡한 데이터를 많이 포함하고 있어서 언어를 작성하고 수정하기에 까다로웠다. 또한 다른 작업 도메인에 시나리오를 적용하고자 할 때는 해당 작업 도메인에 맞게 아바타의 시나리오를 재 작성해야 하는 번거로움도 있었다. 따라서 사용자가 아바타의 행위를 쉽게 표현 할 수 있고, 다양한 작업 도메인에도 적용할 수 있는 행위 표현 언어에 관련된 연구가 필요하다.

본 연구에서는 사용자가 쉽게 작성할 수 있는 작업 수준의 행위 표현 언어를 정의하고, 정의된 언어를 모션 언어로 변환시켜 아바타의 동작을 렌더링 할 수 있도록 하는 CyberClass 아바타 시스템을 구현하여 정의한 행위 표현 언어를 적용해 보았다.

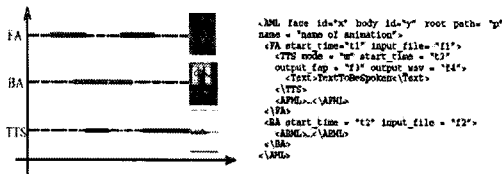
2. 관련연구

아바타 동작 제어와 관련된 연구들은 다양하게 있었으나 아바타를 위한 언어정의를 목적으로 하고, 작업 도메인이 정해져 있는 연구들로 정리하면 다음과 같다.

2.1.1 AML(Avatar Markup Language)[5][6]

XML을 기반으로 Animation제작자가 쉽게 이해하고 생성할 수 있는 언어 구현을 목표로 설계된 행위 언어이다. Mpeg4 기반의 Face Expression과 Body Animation 그리고 TTS(Text To Speech)를 지원한다.

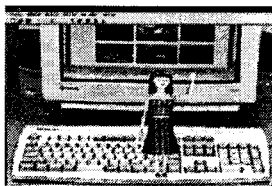
1) 본 논문은 2003년도 산업자원부의 지원에 의하여 연구되었음



[그림 1] AML

2.1.2 CPSL(Cyber Person Scenario Language)[7]

일본 Institute of Technology에서 강의 보조 목적으로 개발한 Cyber Person의 동작 표현 언어이다. 동작에 필요한 애니메이션 데이터는 모션 캡처 데이터를 이용하며 Voice효과, Facial Expression까지 포함하여 시나리오를 작성할 수 있도록 구성되어 있다.



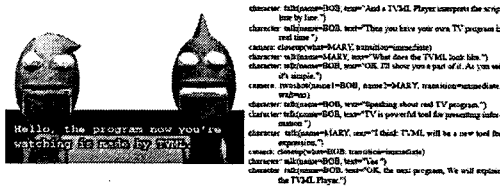
```

<FACE PERSON="Nasa" TRANSIT="50" TIMING="4"ANGLE<ANGLE
<SPEECH> Hello everyone. I am Cyber girl Nasa.
<FACE PERSON="Em" TRANSIT="50" <SMILE
<SPEECH> Hello everyone. Nice to meet you. I am Cyber girl Em. <SPEECH>
<FACE>
<SPEECH>
<FACE> ANGRY <FACE>
<SPEECH> I'm sorry now. <SPEECH>
<FACE>
  
```

[그림 2] CPSL

2.1.3 TVML(TVProgram Making Language)[1][2]

일본 NHK에서 개발한 TV프로그램을 제작하기 위한 시스템이다. 모델링 데이터는 Inventor를 이용하여 작성되고 사용을 위한 Editor와 Viewer가 개발되어 있다.



[그림 3] TVML

기존 연구들은 행위 정의 보다는 아바타의 제어와 관련된 정보들을 많이 포함되어 있다. 따라서 행위 표현 언어 보다는 동작 표현 언어에 가깝다고 할 수 있다.

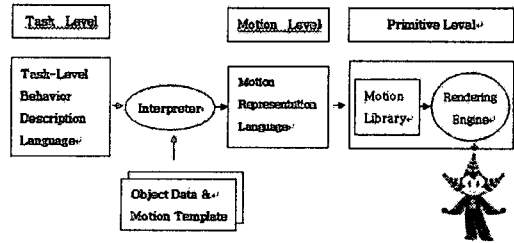
3. CyberClass에서의 아바타 시스템

3.1 시스템 개요

모든 작업 도메인의 아바타 행위 표현 언어를 정의하는 일은 매우 광범위한 작업이므로 본 연구에서는 작업 도메인을 2차원 환경의 CyberClass로 하여 행위 표현 언어를 정의하였다.

전체 시스템은 그림 4와 같은 계층 구조로 구성된다.

CyberClass의 행위를 분석하여 아바타의 작업수준 행위 표현 언어(Task-Level Behavior Description Language)와 동작 표현 언어(Motion Representation Language)를 정의하고, 행위 표현 언어를 동작 표현 언어로 변환시키거나 아바타 행위를 제어하기 위한 Interpreter를 구현한다. 또한 동작과 관련된 애니메이션 데이터가 들어있는 모션라이브러리를 구축하고 렌더링 엔진을 통해 아바타의 동작을 보여지게 된다.



[그림 4] 시스템 구성도

3.2 행위 표현 언어

본 연구에서는 아바타의 움직임을 “행위”와 “동작”으로 구분 하였는데, “행위”라는 것은 복잡한 아바타의 모델링 관련 정보를 배제한 사용자가 작성하거나 수정하기 쉬운 작업수준의 상위레벨 언어를 말한다.

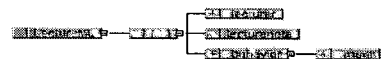
이러한 행위는 대상 객체를 위주로 한다고 할 수 있으며 따라서 행위 언어의 문법은 행위의 주체인 “주어”와 행위 자체를 의미하는 “동사” 그리고 행위를 위한 대상을 나타내는 “목적어”로 구분하였다.

이러한 문법에 따라 아바타의 행위를 나타내는 <behavior> 태그에는 속성 값으로 행위 명(동사)을 나타내는 “name”과 행위의 주체(주어)인 “avatar” 그리고 행위의 객체(목적어)가 되는 “target”의 속성 값을 가진다. 또한 하위 태그로 <ment>를 두어 아바타의 대사를 작성할 수 있도록 하였다.

```

<behavior name="lecturelefttop"
  avatar="snow" target="Line(3)" />
  <ment>XML에 대해 살펴보겠습니다.</ment>
</behavior>
  
```

[그림 5] <behavior>와 <ment>태그의 예



```

<!ELEMENT lectureML (lecture+, lecturenote, behavior+)>
<!ELEMENT lecture EMPTY>
<!ELEMENT lecturenote EMPTY>
<!ELEMENT behavior (ment*)>
<!ELEMENT ment (%PCDATA)>
<!ATTLIST lectureML name CDATA #REQUIRED
  file CDATA #REQUIRED>
<!ATTLIST lecturenote name CDATA #REQUIRED
  file CDATA #REQUIRED>
<!ATTLIST behavior name CDATA #REQUIRED
  avatar CDATA #IMPLIED
  target CDATA #IMPLIED>
  
```

[그림 6] 행위 표현 언어의 DTD

CyberClass에서 필요한 행위들은 표1과 같이 정의하였다.

행위명	상세 행위명	설명
ViewFront	-	정면을 보고 손을 앞으로 모은 상태로 선다.
Enter	EnterLeftTop EnterLeftCenter EnterLeftDown EnterRightTop EnterRightCenter EnterRightDown EnterMiddleTop EnterMiddleCenter EnterMiddleDown	화면의 해당 위치에 나타난 후 해당위치로 이동 한다.
Introduce	-	정면을 보고 인사하고 자기소개를 한다.
Lecture	LectureLeftTop LectureLeftCenter LectureLeftDown LectureRightTop LectureRightCenter LectureRightDown	해당 위치를 손으로 가리키며 강의한다.
Exit	ExitMagic ExitBye	인사하고 사라진다.

[표 1] CyberClass 행위 정의

3.3 동작 표현 언어

동작 표현 언어란 행위표현 언어에 의해 작성되는 세부 동작들을 정의하는 언어이며, 하나의 행위는 여러 동작의 집합(Motion Set)형태로 작성된다. 이러한 동작 표현 언어를 여러 모션Library와 렌더링 엔진의 표준이 되는 언어로 정의하게 되면, 작업레벨의 행위 표현 언어와 하위의 모션라이브러리나 렌더링 엔진들에 종속적이지 않은 시스템 구성이 가능하다.

이러한 동작표현 언어는 사용자가 직접 작성하지 않고, 시스템에 포함되어있는 Interpreter에 의해서 자동 생성 된다.

표2는 CyberClass에서 필요한 동작을 정의한 것이다.

동작명	설명
show	나타난다.
hide	사라진다.
move	이동한다.
greet	고개 숙이다.
greet_return	고개를 들다.
fold	손을 앞으로 모으다.
attention	정면보고 차렷 자세로 서다.
wave	손을 들고 흔들다.
gesture_left_up	왼손을 위로 뻗는다.
gesture_left_down	왼손을 아래로 내린다.
gesture_left_center	왼손을 가운데로 뻗는다.
gesture_right_up	오른손을 위로 뻗는다.
gesture_right_down	오른손을 아래로 내린다.
idle	이벤트가 없을 때 하는 동작

[표 2] CyberClass 동작 정의

3.4 시스템의 구현

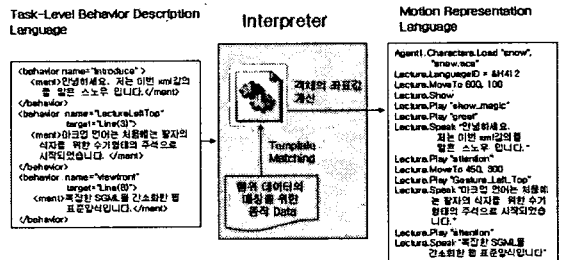
3.4.1 Interpreter의 구현

행위 표현 언어를 시스템에 필요한 언어인 동작 표현 언어로 자동으로 변환 되도록 하기 위해 Interpreter를 구현하였다.

시스템에서 Interpreter는 다음과 같은 역할을 하게 된다.

- 행위 표현 언어를 템플릿 매칭 방식으로 동작 표현 언어로 변환시킨다.
- 변환 시에 동작에 필요한 파라미터 값을 자동 계산하여 동작표현 언어를 작성한다.
- 아바타가 동작할 때 동작과 동작의 연결 및 대기 시간에 필요한 동작 등을 제어한다.

그림7은 Interpreter의 구성도이고, 그림8은 Interpreter에 의해 작성된 객체의 좌표 값을 보여준다.



[그림 7] Interpreter 구성도

```

<object name="Line (3)" x="400" y="200"/>
<object name="Line (4)" x="400" y="220"/>
<object name="Line (5)" x="380" y="240"/>
<object name="Line (6)" x="390" y="260"/>
<object name="Line (7)" x="410" y="300"/>
<object name="Line (8)" x="410" y="320"/>
<object name="Line (9)" x="450" y="240"/>
<object name="Image (1)" x="190" y="150" y="320"/>
<object name="Image (1).SGL" x="200" y="500"/>
<object name="Image (1).HTFL" x="150" y="380"/>
<object name="Image (1).XBL" x="200" y="360"/>
<object name="Image (2)" x="320" y="360"/>
<object name="Image (2).XBL" x="350" y="380"/>
<object name="Image (2).HTFL" x="380" y="420"/>
<object name="Link.next" x="360" y="490"/>
    
```

[그림 8] 객체의 좌표 값

3.4.2 Motion Library 구축

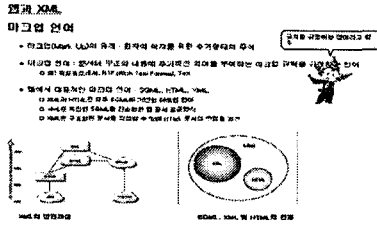
동작 표현 언어의 실제 아바타 애니메이션 데이터인 모션 라이브러리는 Ms-Agent[3][4]를 이용하여 작성 되었으며 동작 표현 언어의 동작 명과 모션라이브러리의 동작명은 1:1로 동일하게 매칭되며 이미지 데이터로 각각의 애니메이션 데이터를 구성하였다.

동작명	모션 라이브러리의 동작명	Animation Data (Frame1 ~ Frame6)
attention	attention	

[표 3] Motion Library의 "attention" 동작의 예

3.5 구현 결과 및 분석

시스템 구현을 위해 행위 표현 언어와 동작표현 언어는 XML로 작성되었으며, Interpreter는 Visual Basic으로 구현하였다. Interpreter는 XML파서로 행위 표현 언어를 분석하여 필요한 동작과 Object값을 계산하고 동작 표현 언어를 작성한다. 이를 Ms-Agent의 렌더링 엔진을 통해 아바타의 행위를 보여주게 된다.



[그림 9] 구현 결과 화면

기존의 언어들은 3차원 환경을 위주로 정의 되었다는 차이가 있으나 목표지점까지 "walk" 동작을 하면서 움직인 후 "I am walking" 이라고 말하는 시나리오를 비교해 보면 다음과 같다.

```
<BodyAnimationTrack name="Track name">
  <WalkingAction>
    <StartTime>
      {mm:ss:mmm/autosynch/autoafter}
    </StartTime>
    <ControlPoint>
      <XCoord> target's X coordinate</XCoord>
      <ZCoord> target's Z coordinate</ZCoord>
    </ControlPoint>
    <Speed>{normal/slow/fast}</Speed>
    <Priority>0 to n</Priority>
  </WalkingAction>
</BodyAnimationTrack>
```

[그림 10] AML의 "walk" 행위 정의내용

```
<WALK DISTANCE="1000" NOTIFY="ABORT"
  REPEAT="100" FRAMES="50">WALKING
  <SPEECH NOTIFY="SYNC">I am walking
  now</SPEECH>
</WALK>
```

[그림 11] CPSL의 "walk" 행위 정의내용

```
character: turn(name=BOB, d=-90)
character: walk(name=BOB, x=-1.5, y=0.0, z=1.0, d=-90)
```

[그림 12] TVML의 "walk" 행위 정의내용

```
<behavior name="walk" target="Line(3)" >
  <ment>I am walking now</ment>
</behavior>
```

[그림 13] Task-Level Behavior Description Language의 "Walk" 행위 정의내용

이와 같이 본 연구에서 정의한 행위 표현 언어(Task-Level Behavior Description Language)를 사용하면 하위 동작 제어에 필요한 상세 정보들은 Interpreter에 의해 시스템에서 자동으로 입력되어지므로 언어가 간단하게 정의 되어지는 것을 알 수 있다.

4. 결론 및 향후 연구방향

본 연구에서는 기존 연구에서의 작성과 수정이 까다로웠던 아바타 동작의 제어와 관련한 문제를 해결하기 위해 작업 도메인을 CyberClass로 하는 작업수준의 행위 표현 언어를 작성하여 사용자에게 제공하고, 시스템에 필요한 복잡한 동작 데이터는 동작 표현 언어를 정의하여 상위레벨 언어와 하위레벨 언어를 분리하였다. 또 행위표현 언어와 동작표현 언어의 자동 변환을 위해 Interpreter를 구현하였으며 이를 통해 사용자가 아바타의 동작을 제어할 때 행위 표현 언어만을 이용하면 되도록 하는 시스템을 구현해 보았다. 또한 기존 연구와의 비교를 통해 정의한 행위 표현 언어가 좀 더 간단히 아바타의 행위를 제어할 수 있음을 알 수 있었다.

이러한 행위 표현 언어를 CyberClass 뿐만 아니라 가상 쇼룸이나 게임 등의 다양한 작업도메인에 맞도록 정의함으로써 다양한 작업 환경에도 적용 될 수 있다.

향후에는 다양한 작업 도메인별 행위 표현 언어의 정의와 함께 하위레벨의 다양한 모션라이브러리들의 기준이 되는 아바타 동작 표현 언어의 표준에 관한 연구가 필요하다.

[참고문헌]

- [1] M. Hayashi, H. Ueda, T. Kurihara, M. Yasumura, "TVML(TV Program Making Language)-Automatic TV program Generation from Text-based Script", proceedings of Imagina'99, Monaco, 1999
- [2] M. Hayashi, Y.Gakumazawa, Y.Yamanouchi, "Automatic Generation of Talk Show from Dialog using TVML", Proceedings of Digital Convergence for Creative Divergence on ICCV'99, Vol.2, pp.325-332, Japan, 1999
- [3] Microsoft, "Microsoft Agent Software Development Kit", Microsoft press, 1999
- [4] MS Agent, "http://www.microsoft.com/msagent/default.asp"
- [5] S. Kshirsagar, A. Guye-Vuilleme, K. Kamyab, N. Magnenat-Thalman, D. Thalman, E. Mamdani, "Avatar Markup Language", Proceedings of the Eurographics Workshop on Virtual Environments(EGVE'02), pp.169-177, Spain, 2002
- [6] Y. Arafa, B. Kamyab, E. Mamdani, S. Kshirsagar, N. Magnenat-Thalman, A.Guye-Vuilleme, D. Thalman, "Two approaches to Scripting Character Animation", Workshop on 'Embodied Conversational Agents - let's specify and evaluate them!', Adaptive Agents and Multi-Agent Systems (AAMAS 2002), Italy, 2002
- [7] Y. Shindo, H. Matsuda, "Design and Implementation of Scenario Language for Cyber Teaching Assistant", Proceedings of International Conference on Computers in Education (ICCE2001), Vol.2, pp.643-650, Korea, 2001.