

작업 수준의 행위 표현 언어를 이용한 사이버강의용 아바타 시스템

김 정 희* · 임 순 범**

요 약

최근 웹이나 가상환경에서는 아바타의 활용이 증가하고 있으나, 아바타의 행위를 사용자가 직접 제어하는 수준의 서비스는 이루어지지 못하고 있다. 또한 기존에 제공되었던 언어들에는 제어에 필요한 복잡한 세부 정보들이 많이 포함되어 있어서 사용자가 작성하기에 까다로움이 많았으며, 작성한 언어를 다른 작업 도메인에 적용하고자 할 때에도 언어를 수정하거나 재 작성해야하는 번거로움이 있었다. 따라서 본 논문에서는 아바타의 행위를 간단히 정의하기 위해, 작업수준별 아바타의 행위를 작성 할 수 있는 “작업 수준의 행위 표현 언어”와 동작에 관련된 세부적인 데이터를 작성 할 수 있는 “동작 표현 언어”로 각각 정의하였다. 또한 시스템 내에 “인터프리터”를 두어 동작 표현 언어를 자동으로 생성 할 수 있도록 함으로써, 사용자가 행위 표현 언어만으로 아바타의 행위를 쉽게 제어 할 수 있는 시스템을 구성하였다. 이러한 내용을 사이버 강의에 적용해 보고, 정의한 작업 수준의 행위 표현 언어를 기존 언어들과 비교하여 아바타의 행위를 얼마나 간단히 표현할 수 있는지 검토하였다.

CyberClass Avatar System using Task-Level Behavior Description Language

Jung-Hee Kim* · Soon-Bum Lim**

ABSTRACT

In spite of recent increase in the use of avatar systems in Web and Virtual Reality, there has not been a service that allows users to control directly the avatar behaviors. In addition, the conventional behavior control languages required a lot of complicated information for controlling the behaviors, so that users had difficulty using them. To apply written languages to a different task domain, moreover, it was necessary to modify or rewrite the languages. In this paper, for the avatar behavior control more simply define, “Task-Level Behavior Description Language,” which allows description the avatar behaviors in each task domain and “Motion Representation Language,” which enables writing detailed data for motion control. The system, developed in this paper, “included an Interpreter,” which automatically creates the Motion Representation Language, allowing users to easily control the avatar behaviors simply with the Behavior Description Language. The system was also applied to cyber classes, and the Task-level Behavior Description Language was compared with conventional languages to see how it was more effective in behavior description.

키워드 : 아바타 동작(Avatar Behavior), 작업수준(Task-Level), 행위 표현 언어(Behavior Description Language)

1. 서 론

최근에 웹이나 가상환경에서는 사용자와의 상호작용을 효율적으로 할 수 있고, 시스템에 대한 몰입감을 높이기 위한 목적으로 아바타의 활용이 증가하고 있는데, 예를 들면 웹 파사나, 싸이월드에서의 단순 애니메이션 작업을 통한 2D 아바타나, Microsoft사의 MS Agent를 이용한 아바타, 또는 게임이나 3차원 가상환경에서 사용되는 3D 아바타 등이 있다[1-3]. 그러나 이러한 아바타의 활용을 보면 단순히 이미지를 변경하거나 미리 작업된 애니메이션 데이터를 보여주는 정도이고, 실질적으로 아바타의 동작을 사용자가 원하는

대로 제어하는 수준의 서비스는 이루어지지 못하고 있다.

또한 기존에 아바타 행위제어를 위한 연구들이 다양하게 이루어졌으나, 작성방식이 하위의 모델링 언어에 가까워서 일반 사용자가 작성하기에는 까다로움이 있었고, 작성한 내용을 다른 작업 도메인에 적용하고자 할 때는 해당 작업 도메인에 맞게 언어를 재 작성해야하는 번거로움이 있었다.

따라서 사용자는 복잡한 하위 모델링 데이터에 신경 쓰지 않고 간단하게 아바타의 동작을 제어 할 수 있는 상위레벨의 행위 표현 언어에 대한 연구가 필요하다.

본 논문에서는 사용자가 쉽게 작성할 수 있는 “작업 수준의 행위 표현 언어”와 시스템 수준의 “동작 표현 언어”를 각각 정의하고 이를 사이버 강의용 아바타 시스템에 적용하였다.

* 준 회원 : 숙명여자대학교 대학원 멀티미디어학과
 ** 종신회원 : 숙명여자대학교 멀티미디어학과 교수
 논문접수 : 2004년 5월 15일, 심사완료 : 2004년 7월 23일

2. 관련 연구

아바타 행위제어를 위한 기존연구에는 AML[4, 5], CPSL[6, 7], TVML[8, 9], CML[10], STEP[11], VHML[12], HumanML [13] 등 다양하나 작업수준이 정해진 상위레벨 언어들을 살펴보면 다음과 같다.

2.1 AML(Avatar Markup Language)

스위스의 MIRA Lab과 London Imperial College에서 연구된 언어로, 애니메이션 프로그램들이나 가상 환경 내에서 아바타의 행위 제어를 쉽게 사용할 수 있도록 하기 위해 XML로 작성된 아바타 행위 제어 언어이다.

MPEG-4를 기반으로 한 아바타 애니메이션을 지원하며, 음성을 지원하는 TTS(Text To Speech)엔진이 포함된다.

아바타의 몸동작의 정의를 위한 ABML(Avatar Body Markup Language)과 얼굴 표정(Facial Expression)을 위한 AFML(Avatar Face Markup Language)로 나누어 정의된다. ABML에는 <BodyAnimation Track> 태그를 통해 상위레벨의 행위 언어를 작성할 수 있으며 "Facing", "Pointing", "Walking", "Waiting", "Resetting"의 기본 행위가 정의 되어 있다(그림 1).

```
<BodyAnimationTrack name = "TrackName" >
  <FacingActing >
    <startTime >
      {mm : ss : mmm/autosynch/autoafter}
    </startTime >
    <XCoor > target's X coordinate in meters </XCoor >
    <YCoor > target's Ycoordinate in meters </YCoor >
    <ZCoor > target's Ycoordinate in meters </ZCoor >
    <Speed > {normal/slow/fast} </Speed >
    <Priority > 0 to n </priority >
  </FacingActing >
</BodyAnimationTrack >
```

(그림 1) ABML의 예

2.2 CPSL(Cyber Person Scenario Language)

일본 Institute of Technology에서 강의 보조 목적으로 개발한 Cyber Person의 행위 언어이다. 학생이 원격에 있는 강사에게 강의내용에 대한 질문을 요청을 하면 강사는 Cyber Person의 시나리오를 학생에게 전송하면 학생 컴퓨터에 있는 아바타가 자동으로 시나리오 DB를 검색하여 시나리오 대로 강의내용을 설명하도록 하는 시스템이다.

(그림 2) CPSL의 Body Action구조

```
< MOTION > BOWING </ MOTION >
< MOTION person = "Emi" Repeat = "10" >
  WORKING </ MOTION >
< ACTION Gain = "80, 30" >
  Lift up R.arm in Front </ ACTION >
< ACTION TTC = "0, 10, 100, 5, 100, 0" >
  Bend R.knee </ ACTION >
< ACTION Notify = "SYNC" Timing = "eTransit" >
  Bend Ankles Outward </ ACTION >
```

(그림 3) CPSL의 예

XML로 정의되었으며, 음성지원과 얼굴표정에 대한 정의도 포함된다. 아바타의 행위는 <Action>과 <Motion> 태그 내에 작성되는 행위 이름(Action Word)으로 "Action Word DataBase"와 "Body Motion Database"를 각각 검색하여 학생 컴퓨터에 내장된 "CPSL Engine"에 의해 아바타가 동작하게 된다. 기본 행위로는 "Walk", "Jump", "Slide", "Rotation"이 있다(그림 2), (그림 3)).

2.3 TVML(TV Program Making Language)

일본 NHK사에서 개발되었고, TV프로그램의 제작을 목적으로 개발되었다. TV프로그램 저작자가 Editor를 통해 아바타와 TVML을 작성하여 제공하고 사용자는 Player를 통해 아바타가 진행하는 TV프로그램을 볼 수 있다.

언어에는 아바타 행위와 관련된 정의 이외에도 TV 프로그램 제작에 필요한 여러 명령어(CG Character Command, Studio Set, Motion Picture, Title, Superimposing, Sound)들이 포함되어 있다. 기본 행위에는 "Walk", "Stop", "Sit", "Stand", "Turn", "Bow", "Look", "Gaze", "Shake"가 있다 (그림 4).

```
sound : playfile (filename = muddyhoochie.aiff)
super : on (type = infilehtml, tagname = open.script)
character : walk (name = BOB, x = -0.1, y = 0.0, z = -0.3, d = 200
  wait=no)
character : walk (name = Mary, x = -0.1, y = 0.0, z = -0.3, d = 160)
camera : twoshot (name1 = BOB, name2 = Mary)
character : sit (name = BOB, name2 = Mary)
character : sit (name = Mary)
super : on (type = text, text = "Mary BoB")
character : bow (name = BOB, wait = no)
character : bow (name = Mary)
character : talk (name = BOB, text = "Hello everybody.")
character : talk (name = Mary, text = "Thank you for tuning in.")
```

(그림 4) TVML의 예

2.4 기존 연구의 분석

기존에 제공된 언어들은 아바타의 행위에 대한 표현보다는 아바타의 얼굴표정이나 몸동작의 제어와 관련된 복잡한 정보들을 많이 포함하고 있으며, 정의된 속성 값이나 좌표 값 등에 대한 이해도 어려워 전문사용자가 아닌 일반 사용자는 접근하기가 어려웠다. 따라서 본 논문에서 행위 제어를 위한 언어에 수준을 두어, 작업수준에 대한 행위를 정의하는 작업수준(Task-Level)과 단순한 움직임만을 정의하

는 동작수준(Motion-Level)으로 나누었으며, 작업수준에서는 사용자가 이해하기 쉬운 “행위 표현 언어”를, 동작수준에서는 동작에 필요한 상세한 데이터들을 포함하는 “동작 표현 언어를 각각” 정의하였다. 또한 수준별 언어의 차이를 보충하기 위해 시스템 내에 인터프리터를 두었다.

3. 아바타 행위 및 동작 표현 언어의 설계

3.1 시스템 개요

모든 작업수준에 맞는 아바타의 행위 표현 언어를 정의하는 일은 매우 광범위한 작업이므로 본 논문은 작업 도메인을 사이버강의로 하는 2차원 환경을 기준으로 아바타의 행위를 정의하였다.

전체 시스템은 작업 도메인의 행위를 정의하는 작업수준(Task-Level)과 정의된 행위의 세부 동작들을 정의하는 동작수준(Motion-Level) 그리고 모션 라이브러리나 렌더링 엔진이 포함되는 기초수준(Primitive-Level)으로 나누어지며, 작업수준에서는 작업 도메인에 맞도록 정의된 행위 표현 언어(Task-Level Behavior Description Language)를, 동작수준에서는 상세한 세부 동작과 객체 정보 데이터들을 포함한 동작 표현 언어(Motion Representation Language)를 각각 정의한다. 또한 행위 표현 언어를 통해 객체 정보를 찾거나 계산하고 행위 템플릿을 매칭시켜 동작표현 언어로 변환시키는 작업을 위해 “인터프리터”를 구현하였으며, 동작 표현 언어로 정의되는 동작들에 대한 애니메이션 데이터인 “모션 라이브러리”를 구축하고, 렌더링엔진을 이용하여 아바타의 행위를 보여준다(그림 5).

(그림 5) 시스템 개념도

3.2 아바타 행위 표현 언어의 설계

일반적으로 아바타의 “행위”는 어떤 대상(Object)에 대한 목적을 갖는 행동이라고 할 수 있으며, 손을 들어 올린다거나 고개를 숙이는 등의 단순한 움직임만을 말하는 “동작”과는 차별 된다고 할 수 있다.

따라서 본 논문에서의 “행위”는 동작을 제어하기 위해 사용되는 복잡한 데이터를 배제한, 사용자가 이해하기 쉬운 상위레벨의 행위들에 대한 정의를 의미하며 작성된 “행위 표현 언어”는 사용자가 이해하기 쉽고 간결하며 하나의 행위로 여러 세부 동작들(Motion Set)을 표현할 수 있는 것이어야 한다. 그러므로 행위 표현 언어로 정의된 행위는 여러

동작의 집합형태로 나타낼 수 있는 것으로 정의되는데, 예를 들어 “설명하다”라는 행위를 동작으로 표현하면, “이동하다”와 “손을 뻗다”, 그리고 “가리키는 것을 보다”로 구성되는 동작의 집합 형태로 나타낼 수 있다.

- ① ~ (가리키는 곳)으로 이동한다.
- ② (왼/오른)손을 (위/아래/왼쪽/오른쪽)으로 뻗는다.
- ③ 가리키는 곳을 본다.

또한 아바타의 행위를 분석해 보면, 주로 행위를 하는 “주체”와 행위를 할 대상 “객체”, 그리고 “행위”가 있음을 알 수 있었으며, 이에 따라 행위 표현 언어의 문법은 행위의 주체인 아바타를 “주어”로, 아바타의 행위를 “동사”로, 그리고 행위를 위한 대상객체를 “목적어”로 하는 문법적인 개념을 이용하여 구분하였다.

따라서 행위 표현 언어에서는 <behavior>태그로 아바타의 행위를 표현 하도록 하였으며, 속성으로 행위이름을 나타내는 “name”과 행위주체를 나타내는 “avatar”, 그리 대상객체를 나타내는 “target”으로 정의하였다. 또한 하위에 <ment>태그를 주어 아바타의 대사를 표현 할 수 있도록 하였다(그림 6).

```

< behavior name = "EnterLeftTop" avatar = "snow" />
< behavior name = "Introduce" avatar = "snow" >
  < ment > 안녕하세요.
  이번 강의를 맡은 감사 스노우입니다.</ ment >
</ behavior >
< behavior name = "LectureRightTop" avatar = "snow"
  target = "Line(3)" >
  < ment > 오늘은 XML을 살펴보겠습니다. </ ment >
</ behavior >
    
```

(그림 6) behavior와 ment태그의 예

사이버 강의를 위한 행위 표현 언어의 DTD에는 아바타의 이름과 모션 라이브러리 파일명을 지정할 수 있는 <lecturer> 태그와 강의노트의 이름과 해당 파일을 정의할 수 있는 <lecturenote>태그가 포함되어 있다(그림 7).

(그림 7) 사이버강의 행위 표현 언어 DTD

작업 도메인인 사이버강의에서 주로 사용되는 아바타의 기본 행위는 정면을 보고 손을 앞으로 모으는 “ViewFront”와, 화면으로 입장하는 “Enter”, 소개하는 “Introduce”, 해당 위치를 가리키며 강의하는 “Lecture”, 그리고 화면에서 사라지는 “Exit”로 정의하였고, “EnterLeftTop”과 같이 파라미터명을 추가하여 행위 명을 변경함으로써 기본행위를 여러 형태로 표현할 수 있도록 하였다<표 1>. 이러한 기본 행위들은 인

터프리터에 의해 번역되어 동작 표현 언어로 작성된다.

〈표 1〉 사이버 강의 행위 정의

행 위 명	세부 행위명		설 명
	Parameter1	Parameter2	
ViewFront	-		정면을 본다. 손을 앞으로 모은다. 선다.
Enter	Left Right Middle Magic	Top Down Center	화면의 해당 위치에 나타난다. 해당위치로 이동한다. 선다.
Introduce	-		정면을 본다. 인사한다. 소개 한다.
Lecture	Lecture Right	Top Down Center	해당 위치를 손으로 가리킨다. 강의 한다.
Exit	Exit Magic ExitBye		정면을 본다. 인사한다. 사라진다.

3.3 아바타 동작 표현 언어의 설계

“동작”이라는 것은 “행위”로 표현된 것을 실제로 보여주기 위해 필요한 세부적인 움직임들을 말하며, “동작 표현 언어”에는 아바타가 각각의 동작을 하는 데에 필요한 모든 데이터들이 포함되어 있고 시스템 내에서 작업수준과 분리된 동작수준으로서 독립적으로 존재하게 된다. 사이버강의의 동작 표현 언어는 MS Agent[14, 15]의 아바타 동작함수들을 이용하여 정의하였다(그림 8).

따라서 동작표현 언어를 하위의 모션라이브러리나 렌더링 엔진의 표준이 되는 언어로 정의하게 되면, 여러 작업수준의 행위 표현 언어와 다양한 하위의 렌더링 엔진에 종속적이지 않은 시스템을 구성할 수 있다.

이러한 동작 표현 언어는 사용자가 직접 작성하지 않고, 시스템에 포함되어 있는 인터프리터에 의해 행위의 매칭에 필요한 템플릿과 파라미터들이 입력되어 동작 표현 언어 파일로 저장된다.

```

Agent1.Characters.Load "snow", "snow.acs"
Lecture.LanguageID = &H412
Lecture.MoveTo 600, 100
Lecture.Show
Lecture.Play "show_magic"
Lecture.Play "greet"
Lecture.Speak strMent
Lecture.Play "attention"
Lecture.MoveTo 450, 300
Lecture.Play "Gesture_Left_Top"
Lecture.Speak strMent
    
```

(그림 8) 동작 표현 언어의 예

사이버강의에서는 행위표현 언어에서 정의된 여러 행위들에 대해 14가지의 동작 집합으로 나타낼 수 있다<표 2>.

〈표 2〉 CyberClass 동작 정의

동 작 명	설 명
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

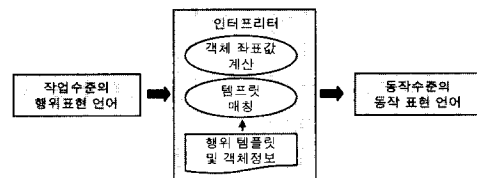
4. 사이버강의 아바타 시스템 구현

4.1 인터프리터의 구현

사용자가 쉽게 작성할 수 있도록 정의된 행위 표현 언어는 아바타를 실제로 조작하기 위해 필요한 정보들을 충분히 가지고 있지 못하다. 따라서 행위 표현 언어를 그대로 시스템에 적용 할 수 없으므로 시스템에 필요한 모든 정보를 가지고 있는 동작 표현 언어가 필요한데, 사용자가 이를 정확하게 작성하는 것은 어려우므로 이를 인터프리터가 대신 할 수 있도록 하였다.

이러한 인터프리터의 역할은 다음과 같다(그림 9).

- 행위 표현 언어를 동작 표현 언어로 변환 : 변환 시 정의된 각각의 행위들은 템플릿 매칭 방식을 이용해 여러 동작들로 변환된다.
- 동작 표현 언어에 필요한 파라미터 값 계산 : 설명할 Line이나 그림 등의 객체 데이터 정보를 이용하여, 아바타의 동작에 필요한 위치좌표 값이나 거리 등을 자동적으로 계산하여 동작 표현 언어를 작성 한다
- 아바타 행위의 제어 : 아바타가 동작 할 때 필요한 연결동작이나, 사용자로부터의 이벤트가 없을 때에 필요한 동작 등을 제어한다.



(그림 9) 인터프리터 구성도

동작표현 언어로의 변환을 위한 템플릿은 인터프리터 내에 저장되어 있으며 각각의 행위에 대한 동작의 집합에 관련된 상세한 정의가 포함되어 있다<표 3>.

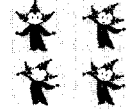
〈표 3〉 행위 표현 언어의 템플릿

행위명	설명
ViewFront	① attention(정면을 본다) ② fold(손을 앞으로 모은다)
Enter	① show(나타난다) ② move x, y(왼쪽/오른쪽위/가운데/아래)로 이동 ③ attention(정면을 본다) ④ fold(손을 모은다)
Introduce	① attention(정면을 본다) ② greet(고개를 숙인다) ③ greet_return(고개를 든다) ④ fold(손을 모은다)
Lecture	① move x, y(~가리키는 곳으로 이동한다) ② gesture(left/right)(top/down) (원/오른)손을 (위로/아래로/왼쪽/오른쪽)으로 뻗는다.
Exit	① 정면을 본다(attention) ② 손을 흔든다(wave), 뭉개구름(Magic) ③ 사라진다.(hide)

4.2 모션라이브러리 구축 및 렌더링 엔진의 적용

동작 표현 언어에 의해 표현 되는 동작들의 아바타 애니메이션 데이터인 모션라이브러리는 MS Agent를 이용하여 작성하였고, 동작 표현 언어에서 사용되는 동작명과모션라이브러리의 동작명이 동일하게 매칭 되도록 구성하였다(표 4).

〈표 4〉 모션라이브러리의 "gesture_left_top"의 예

동작명	모션 라이브러리의 애니메이션 명	애니메이션데이터 (Frame 1~Frame 4)
gesture_left_top	gesture_left_top	

4.3 구현 결과 및 분석

아바타 동작 제어를 위한 기존 언어의 문제점을 해결하기 위해 시스템을 작업수준과 동작 수준 그리고 기초수준으로 분리하여 정의하는 시스템을 제안하고 이를 사이버강의에 적용하였다. 사이버 강의의 행위 표현 언어는 XML을 이용하여 정의하였으며 동작 표현 언어는 MS Agent 함수로 정의하였다. 인터프리터는 XML로 작성된 행위 표현 언어를 파서로 분석한 후 필요한 동작 집합과 객체 정보들을 포함하는 동작 표현 언어로 변환하며 변환된 동작 표현 언어는 해당 모션 라이브러리와 렌더링 엔진을 통해 아바타의 애니메이션을 실행한다(그림 10), (그림 11)).

(그림 10) 사이버강의 시스템 구성도

(그림 11) 구현 결과 화면

정의된 행위 표현 언어를 기존 언어와 비교해 보았으며, 비교행위는 “해당 위치로 걸어가서 테이블을 가리키며 설명”하는 것이다(그림 12), (그림 13), (그림 14), (그림 15)).

```

< AML >
< FA >< TTS start_time = "00 : 03 : 000" >
  < Text > This is a Table </ Text ></ TTS >
</ FA >
< BA start_time = "00 : 01 : 00" >
< BodyAnimationTrack name = "Walk" >
  < WalkingAction >
  < StartTime > 00 : 01 : 000 </ StartTime >
  < XCoor > 35 </ XCoor >
  < ZCoor > 10 </ ZCoor >
  < Speed > normal </ Speed >
  < Priority > 1 </ Priority >
  </ WalkingAction ></ BodyAnimationTrack >
  < BodyAnimationTrack name = "FacedPoint" >
  < FacingAction > ... (중략) </ FacingAction >
  < PointingAction > ... (중략) </ PointingAction >
  </ BodyAnimationTrack ></ BA >
</ AML >
    
```

(그림 12) AML의 행위 적용 예

```

< WALK DISTANCE = "50" REPEAT = "100" FRAMES = "50" >
  WALKING </ WALK >
< ACTION Gain = "50, 50" > Rotate Body to the Left </ ACTION >
< ACTION Gain = "85, 85" > Raise Left Arm to the Front </ ACTION >
< ACTION > Rotate Face to the Right </ ACTION >
< ACTION Gain = "35, 35" > Raise Left arm to the Front </ ACTION >
< SPEECH NOTIFY = "SYNC" > This is a Table </ SPEECH >
    
```

(그림 13) CPSL의 행위 적용 예

```

character : walk(name = TATSUYA, x = 1.0, y = 0.0, z = 0.0, d = 0)
character : turn(name = TATSUYA, d = -30)
developer : do(command = TkKyeframeOpen, a0 = TATSUYA,
  a1 = point, a2 = /Tatsuya/point2.dcanime)
developer : do(command = TkKyeframe, a0 = TATSUYA,
  a1 = point, a2 = 1.0, wait = yes)
character : talk(name = TATSUYA, text = "This is a table",
  wait = no)
    
```

(그림 14) TVML의 행위 적용 예

```

< behavior name = "walk" avatar = "snow" target = "table" />
< behavior name = "lecture" avatar = "snow" target = "table" >
  < ment > this is atable </ ment >
</ behavior >
    
```

(그림 15) 작업수준 행위 표현 언어의 행위 적용 예

작업수준 행위 표현 언어는 작업수준에 필요한 행위를 정

의하며, 사이버 강의에서는 작업수준 행위로 "Lecture"가 정의되는 것을 알 수 있다. 반면에 시스템을 위한 동작표현 언어는 작업수준에는 무관하게 동작에 필요한 모든 정보들을 포함한 언어로 정의된다. 인터프리터에는 행위 템플릿과 객체 정보 값이 저장되어 필요시에 계산되어 동작 표현 언어를 생성하며 생성된 동작 표현 언어의 내용에 따라 모션라이브리의 내용이 호출되어 렌더링엔진을 통해 아바타의 강의 내용을 보여주게 된다.

이렇게 정의된 행위 표현 언어를 기존 언어들과 같은 행위로 비교해 보았을 때, 작업수준의 행위 표현 언어가 기존 언어들보다 아바타 동작에 필요한 모든 내용을 언어에 포함시킨 것과는 달리 언어를 수준별로 분리함으로써 좀 더 간단하게 아바타의 행위를 정의할 수 있음을 알 수 있다.

5. 결론 및 향후 연구방향

본 논문에서는 기존의 아바타 행위제어 언어의 복잡한 내용들을 작업수준의 행위 표현 언어를 이용하여 간단히 표현하고자 하였으며, 이를 위해 언어에 수준을 두어 작업수준에 관련된 행위를 정의하는 "행위 표현 언어"와 실제 동작에 필요한 상세정보를 갖는 동작 수준의 "동작 표현 언어"로 구분하였으며, 두 언어의 차이를 보충하고자 인터프리터를 두었다. 인터프리터는 작업수준의 행위 표현 언어를 적절한 동작 표현 언어로 변환시킴으로써 사용자가 복잡한 모델링 데이터를 신경 쓰지 않도록 해주고 동작 표현 언어에 필요한 상세 정보를 자동 생성해 줌으로써 상위 레벨인 행위 표현 언어가 좀 더 간단하게 정의될 수 있게 한다. 또한 기초수준에는 모션라이브리와 렌더링 엔진을 두어 아바타의 애니메이션을 보여줄 수 있도록 하였다.

이러한 개념을 이용해 작업수준을 사이버강의로 하는 행위 표현 언어와 동작에 필요한 객체정보를 포함하는 동작 표현 언어를 각각 정의하고, 인터프리터와 모션라이브리를 구축하여 사이버강의 아바타 시스템을 구현 하였다.

향후에는 다양한 작업 도메인별 행위 표현 언어의 정의와 작업수준(Task-Level)과 기초수준(Primitive-Level)에 독립적일 수 있는 동작 표현 언어의 표준에 관한 연구가 필요하다.

참고 문헌

[1] 김재경, 오재균, 임순범, 최윤철, "아바타 행위에 대한 작업레벨 표현 및 제어기법", 한국정보과학회 2004년도 춘계학술발표논문집, pp.520-522, April, 2004.
 [2] 엠파스 아바타볼, <http://avatar.empas.com/home/>.
 [3] 임순범, 송두현, "다양한 상호작용을 이용한 사이버교재의 구현", 한국 컴퓨터산업교육학회논문지, 제2권 제5호, pp.647-660, May, 2001.
 [4] Y. Arafa, B. Kamyab, E. Mamdani, S. Kshirsagar, N. Magnenat-Thalmann, A. Guye-Vuilleme, D. Thalmann, "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
 [5] S. Kshirsagar, A. Guye-Vuilleme, K. Kamyab, N. Magnenat-Thalmann, D. Thalmann, E. Mamdani, "Avatar Markup Language," Proceedings of the Eurographics Workshop on Virtual Environments(EGVE '02), pp.169-177, Spain, 2002.
 [6] 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.
 [7] Y. Shindo, H. Matsuda, "Prototype of Cyber Teaching Assistant," Proceedings of IEEE International Conference on Advanced Learning Technology (ICALT2001), IEEE, pp.70-73, USA, 2001.
 [8] M. Hayashi, H. Ueda, T. Kurihara, M. Yasumura, "TVML (TVProgram Making Language)-Automatic TV program Generation from Text-based Script," proceedings of Imagina '99, Monaco, 1999.
 [9] TVML, <http://www.nhk.or.jp/str/tvml/english/what>.
 [10] Y. Arafa, E. Mamdani, C. Visser, "Scripting Embodied Agents Behaviour with CML : Character Markup Language," Proceedings of the 8th international conference on Intelligent User Interface(UI), pp.313-316, USA, 2003.
 [11] Z. Huang, A.eliens, C. Visser, "Implementation of a Scripting Language for VRML/X3D-based Embodied Agents," Proceedings of the eighth international conference on 3D Web technology, ACM, France, pp.91-100, 2003.
 [12] VHML, <http://www.vhml.org>.
 [13] HumanML, <http://www.humanmarkup.org>.
 [14] Microsoft, "Microsoft Agent Software Development Kit," Microsoft press, 1999.
 [15] MS Agent, <http://www.microsoft.com/msagent>.

김정희

e-mail : preeway@sookmyung.ac.kr

2000년 건국대학교 전산과학과(학사)

2004년 숙명여자대학교 멀티미디어학과

(석사)

관심분야 : 아바타 행위제어, 웹 멀티미디어

(X3D), 전자책 보안(XML, DRM)

임순범

e-mail : sblim@sookmyung.ac.kr

1982년 서울대학교 계산통계학과(학사)

1983년 한국과학기술원 전산학과(석사)

1992년 한국과학기술원 전산학과(박사)

1989년~1992년 (주)휴먼컴퓨터 창업/연구

소장

1992년~1997년 (주)삼보컴퓨터 프린터개발부 부장

1997년~2001년 건국대학교 컴퓨터학과 교수

2001년~현재 숙명여자대학교 멀티미디어학과 교수

관심분야 : 컴퓨터 그래픽스, 웹 멀티미디어 응용, 전자출판(폰트,

XML, 전자책, e-Learning)