아바타 생성 마크업 언어 연구

o
염창근, 박경환
동아대학교 컴퓨터공학과

A study of Avatar Markup Language

Chang-Gun Yum, Kyung-Hwan Park
Dept. of Computer Engineering, Dong-A University*

요 약

XML의 응용 예로 3DML은 VRML의 차기 사양인 X3D와 비교될만한 언어이다. XML에 기반하여 3차원 공간 데이터를 마치 미로를 만들 듯 기호화 된 문자들로 하나의 평면(Level)을 구성하면 그에 해당되는 물체들로 이루어 진 가상 공간이 만들어진다. 하지만, 고정된 가상공간만을 저작할 수 있으며 다중 사용자 환경을 지원하지 않는다. 본 논문은 아바타를 통해서 가상공간을 탐험할 수 있도록 XML의 한 응용된 언어로써 AML을 설계하고 AML을 인식할 수 있는 뷰어를 통해 아바타의 애니메이션을 수행 방법을 기술하였다.

1. 서론

XML[1][2]의 등장 이후로 월드와이드웹은 많은 변화를 일으키고있다. 단순히 정해진 틀 안에서 인터넷이 발전해왔다고 한다면 이제는 자유로운 형식 안에서 무한한 확장성을 가진 것이다. HTML(Hyper Text Markup Language)이 웹에서 표준화된 양식으로 문서를 표현하는 언어라면 XML(extensible Markup Language)은 HTML이 나타낼 수 없었던 다양한 형식의 언어를 생성하여 표현할 수 있는 메타 언어이다.

SGML(Standard Generalized Markup Language)은 비록 SGML이 가졌던 확장성, 구조성, 검증성과 같은 장점에도 불구하고 너무 방대한 사양 때문에 복잡하고 사용하기에 어려웠다. 이에 SGML이 가진 장점을 수용하면서 동시에 사용하기 쉬운 언어를 만든 것이 바로 SGML의 서브셋인 XML이다.

XML은 메타언어이기 때문에 SGML의 응용인 HTML과 같이 XML의 응용을 생성할 필요가 있다. XML의 응용 예로 XMLNews, SMIL, CDF, VML & SVG, 3DML, MathML 등등 다양하다. 다양한 응용 중에서 특히 3DML(3-D Modeling Language)은 기존의 Wed3D 위원회에서 주관해오던 VRML(Virtual Reality Modeling Language)[3]과 비교되는 XML 응용이다. 3DML은 가상공간을 쉽게 저작 가능 하도록 마치 미로를 구성하는 것처럼 문서를 기술한다. 하지만, 3DML은 가상공간만을 저작하며 단지 단일 사용자만이 가상공간을 탐험할 수 있도록 설계되었다. 따라서, 사용자 자신 외에는 가상공간에서 또 다른 사용자를 접할 수 없으며 진정한 가상공간 응용이라고는 볼 수 없다.

그러므로, 본 논문에서는 XML의 한 응용으로 가상공간에서 다중 사용자를 지원하며 상호작용이 가능하도록 사용자를 대신해서 활동하는 아바타를 저작하는 언어를 설계하고자 한다.

2. 3DML

3DML[4]은 XML 형식을 따르므로 well-formed 이긴 하지만 DTD를 요구하지 않으므로 valid 한 문서는 아니다. 3DML 이외의 다른 응용들

과는 달리 가상공간을 저작 함을 목적으로 하기 때문에 객체의 외양만을 기술할 뿐 객체간의 상호 관계는 명시하지 않는 언어이기 때문이다. 즉, 가상공간을 저작 하는 방식이 가상의 3차원 배열을 만들고 배열 안에 들어갈 정보로써 객체를 삽입하는 식으로 문서를 작성하게 된다. 이것은 마치 건물을 지을 때처럼 1층은 어떠한 자재로써 방을 만들고 문을 만들며 2층은 또 다른 형식으로 구성하는 것과 유사하다. 이렇게 하면 쉽고 빠르게 건물을 지을 수는 있지만, 객체 간의 상호 관계가 명시되지 않았기 때문에 논리적인 가상공간을 저작할 수는 없다. 사실 XML은 엘리먼트 하나하나가 의미있는 정보를 갖도록 권하지만 DTD를 가지지 않는 XML 응용들은 이런 것이 필요하지 않다. 결국 3DML은 XML의 문법만을 따왔을 뿐 의미론은 무시한 것이다.

XML 문서를 파서가 아닌 사람이 보았을 때 그 문서는 가독성이 있어야 하지만 3DML과 같은 문서는 헤더 정보를 제외한 실제 공간 정보를 나타내는 엘리먼트들은 문서만을 보아서는 무엇을 의미하는지 알 수 없다. 유지 보수의 측면에서만 보아서도 타당한 방법은 아닐 것이다. 단적인 예로 3DML은 가상공간을 저작할 때 <LEVEL> 이라는 태그를 이용하는데 이것이 가지는 속성으로 NUMBER가 있으며 NUMBER가 지시하는 숫자 값에 따라서 저작 대상이 구분된다.

예)

<LEVEL NUMBER="1">

.. 20

.. 20

.. .. .. 2x

.. .. .. 2x

</LEVEL>

위 코드만을 보고서 얻을 수 있는 정보로는 현재 LEVEL "1"을 저작하고 있다는 것과 크기가 4 X 4인 2차원 배열, 그리고 오른쪽 벽이 문자열이 나타내는 어떤 객체로 채워져 있다 라는 것 뿐이다. 문자열의 조합으로 수많은 객체를 지정할 수는 있지만 반대로 새로이 심볼을 만든다는 것은 그 만큼 문서의 의존성을 낮추는 것이다.

3. DTD와 W3C schema

DTD(Document Type Definition)는 비록 문법이 간단하지만 기능의 제약을 가지고 있다. DTD의 대안으로 등장한 것이 schema로서 현재 두 가지가 있는데 하나는 MS사의 XML Schema[5]와 W3C에서 추진 중인 W3C schema[6]이다. 여기서는 3DML과 연동하여 삽입 가능한 아바타를 생성하고자 하므로 표준을 따르는 W3C schema를 기술한다. W3C schema는 아직 완전한 사양이 아니기 때문에 차후에 변경될 가능성이 있으나 본래의 취지는 변함이 없으므로 적절히 적용이 가능할 것이다.

4. AML(Avatar Markup Language)

사용자를 대신하는 아바타를 생성하는 XML 응용으로 단일 가상공간이 아닌 다중 사용자 환경을 지원하기 위해서 설계된 언어이다. 3DML에서 아바타를 지원함을 목적으로 하며 3DML 가상공간과 상호작용이 가능하다. 아바타는 애니메이션이 가능하며 이런 정보들 역시 다른 아바타에 의해서 인식될 수 있다. 애니메이션을 수행하는 아바타를 저작하기 위해서는 AML 문서가 제약을 가져야 하고 이것은 곧 AML이 3DML과는 달리 스키마를 필요로 하는 이유이다. 아바타의 대표적인 형태로는 바로 인간형(Humanoid) 아바타로써 사람이 갖춘 외양을 스키마로 명시하였다. 스키마는 아바타가 갖추어야 할 골격들을 명시하는데 논리적인 정보를 포함하므로 순서관계와 포함관계가 중요하다. 스키마가 가지는 정보를 아래 표로 나타내었다.

[표 1] AML 스키마 구성 요소

마디(체절)	링크 이름	이동 형태
골반	Pelvis	아바타 전체
머리	Head	아바타 전체
목	Neck	아바타 전체
척추	Spine	상체의 위치 결정
엉덩이	Hip	무릎의 움직임에 따라
어깨	RArm1,LArm	어깨의 움직임에 따라
팔꿈치	RArm2,LArm	손의 움직임에 따라 IK
손	RHand,LHand	손의 움직임에 따라 IK
손가락	Fingers	손가락의 위치 결정
무릎	RLeg,LLeg	발의 움직임에 따라 IK
발	RFoot,LFoot	발의 움직임에 따라 IK
발가락	Toes	발가락의 위치 결정

아바타가 가지는 스키마 정보에는 대칭성을 가지는 체절이 있는데 손이나 발과 같은 요소가 이에 해당된다. 똑 같은 정보이지만 대칭적이므로 태그의 이름으로 구분하지 않았고 엘리먼트의 속성으로 분리시켰다. 한 예로 사람의 팔을 왼팔과 오른팔로 구분하기 위해서

<Arm direction="left"/>

<Arm direction="right"/>

로 기술하고 스키마 사양대로 하부 체절을 나열할 수 있다. 본 논문에서 제안한 AML 스키마의 일부분만을 요약하여 [그림 1]에 보였다.

```
<xsd:schema
xmlns:xsd=http://www.w3.org/1999/XMLSchema>
<xsd:element name="Avatar" type="AvatarType"/>
<xsd:complexType name="AvatarType">
<xsd:element name="Pelvis" type="xsd:string"/>
<xsd:element name="UpperHalf" type="UpperHalfType"/>
<xsd:element name="LowerHalf" type="LowerHalfType"/>
<xsd:attribute name="sex" type="xsd:string"/>
<xsd:attribute name="age" type="xsd:positive-integer"/>
<xsd:complexType name="UpperHalfType">
<xsd:element name="Head" type="HeadType"/>
<xsd:element name="Neck" type="xsd:string"/>
<xsd:element name="Chest" type="xsd:string"/>
<xsd:element name="Abdomen" type="xsd:string"/>
<xsd:element name="Arm" type="ArmType"/>
<xsd:complexType name="ArmType">
<xsd:element name="UpperArm" type="xsd:string"/>
...
```

[그림 1] Avatar Markup Language DTD

본 스키마는 가장 큰 골격인 골반(Pelvis), 상반신(UpperHalf), 하반신(LowerHalf)으로 나누었고 각각의 하부 체절들을 기술한 것이다. AML 스키마에 따라서 AML 문서를 작성하면 아바타의 생김새를 어느 정도 상상이 가능하기 때문에 문서의 가독성을 높이게 된다. 그러나 실제로 그 아바타를 볼 수는 없다. 아바타를 보려면 AML을 파싱하고 출력할 수 있는 뷰어가 필요하다.

5. 아바타의 애니메이션과 뷰어

아바타는 고정된 공간처럼 가만히 멈춰있는 객체가 아니다. 아바타는 항시 사용자를 대신해서 가상공간을 탐험하며 때로는 다른 아바타와 상호작용을 필요로 한다. 3DML은 단지 가상공간만을 저작하는 언어므로 어떤 객체의 애니메이션 정보를 필요로 하진 않지만 부분적으로 움직이는 객체를 기술하고는 있다.

AML이 애니메이션을 지원하기 위해서는 편법적으로 우회할 필요가 있다. 사실 아바타가 수행하는 애니메이션에는 행동 범위가 제한되어야 한다. 인간의 관절이 필요 이상으로 꺾이거나 접혀서도 안되며 크기가 변해서도 안 된다. 즉, 아바타가 가상공간을 탐험하는데 필요한 최소한의 행동을 명시해야 하며 걷기, 뛰기, 인사하기 등과 같은 애니메이션만을 수행한다.

애니메이션 정보는 URI를 통해서 인식되며 아바타의 애니메이션을 수행하는 프로그램은 바로 뷰어가 된다. 뷰어는 AML 문서를 파싱하고 AML 스키마에 명시된 엘리먼트를 DOM(Document Object Model)[7] 기술을 이용하여 특정 요소를 탐색하게 된다. 만약 애니메이션 정보에 따라 움직이게 될 태그를 DOM에 의해 검출하게 되면 그 태그는 하부 태그까지 포함하여 움직이게 되는데 이것이 바로 운동학(Kinematics)이다. 뷰어는 운동학과 동시에 역운동학(Inverse Kinematics)을 지원한다.

팔을 움직이는 애니메이션이 수행되려면 먼저 최상위 노드인 Avatar를 검색하고 이어서 UpperHalf -> Arm을 찾게된다. 만약 Arm을 AML 문서에서 찾을 수 없다면 비록 뷰어가 애니메이션 정보를 갖고 있더라도 뷰어의 아바타 인터페이스에는 팔과 관련된 모든 애니메이션 창들은 비활성화 하게 된다.

6. 아바타 모델 정보

3DML이 사용한 심볼을 이용하여 모델 외형을 불러오는 방법으로 뷰어를 통해서 출력한다. 3DML은 모든 모델 정보를 자사의 웹사이트에 상주시키고 새로운 연결이 생길 때 마다 클라이언트의 캐쉬로 정보를 전송하고 네트웍의 부하를 줄였다. 이와 유사하게

AML은 기본적으로 아바타를 구현함에 있어서 필요한 원형 모델을 심볼로써 참조하도록 서비스한다.
부가적으로 AML의 설계 원칙은 VRML의 차기 버전인 X3D를 인식하고자 한다. 즉, AML의 외형 정보를 X3D 문서를 파싱한 결과를 가지고서 출력하도록 한다. 이것이 가능하려면 AML의 스키마가 가지는 모든 엘리먼트의 속성들이 URI 속성을 가지도록 수정하면 되고 AML 뷰어가 각 엘리먼트를 파싱할 때 URI 속성을 찾으면 그 엘리먼트의 외형 정보는 명시된 URI를 통해서 가져와서 덧붙인다.

7. 결론

XML은 사용자가 자신만의 언어를 설계할 수 있도록 해 주는 메타언어로서 지금까지 다양한 XML 응용들을 볼 수 있었다. 3DML이 시도한 가상공간 저작 기법은 기존의 VRML을 포함한 X3D의 그것과는 사뭇 다르다. 심볼을 이용하여 미로를 꾸미듯 배열 원소를 기술하면 특정 3DML 뷰어를 통해서만 파싱된다. 하지만, 3DML은 고정된 공간이라는 제약과 단일 사용자만을 지원하는 한계 때문에 개선의 여지가 있다. 이에 본 논문에서는 다중 사용자 환경을 지원할 수 있도록 사용자를 대신하는 아바타를 생성하여 기존의 언어와 호환하는 XML 응용인 AML을 설계하고 이 언어를 이해하며 아바타의 애니메이션을 수행하는 뷰어의 설계 기법을 소개하였다. 향후 과제로는 AML 뷰어를 실제로 구현하는 것과 뷰어에서 표현되는 아바타 간의 네트웍 동기화를 묶어주어야 한다.

[참고문헌]

[1] Michael Morrison, et al., XML Unleashed, SAMS, 1999. 12.
[2] W3C, Extensible Markup Language(XML) 1.0, W3C Recommendation, http://www.w3.org/TR/REC-xml, 1998.
[3] Web3D Consortium. http://www.web3d.org/
[4] 3DML flatland site http://www.flatland.com/
[5] Microsoft XML site. http://msdn.microsoft.com/xml/default.asp
[6] XML Schema, W3C Working Draft, 25 February 2000. http://www.w3.org/TR/xmlschema-0/
[7] W3C DOM site. http://www.w3.org/TR/REC-DOM-level-1.