

OO-VRML 을 위한 3D 브라우저의 구현

최석우 한태숙
과학기술원 전산학과
(swchoi.han)@pllab.kaist.ac.kr

Implementation of 3D Browser for OO-VRML

Seok-Woo Choi Taisook Han
Dept. of Computer Science, KAIST

요 약

VRML은 상호 작용을 하는 3D 객체와 세계를 기술하는 파일 형식이다. OO-VRML은 VRML을 객체 지향 언어로 확장하여 정보 은닉, 상속 그리고 동적 바인딩 등을 가능하게 한 언어이다. OO-VRML의 이런 특징들은 더 동적인 가상 세계를 섬세하게 조정할 수 있도록 해 준다. 이 논문에서는 OO-VRML의 표현 능력을 잘 활용하기 위해 OO-VRML언어를 사용하는 전용 브라우저를 설계하고 구현한다. 브라우저는 파싱 및 인스턴스화 모듈, 실행 모듈, 브라우저 모듈의 세 부분으로 나누어진다. 파싱 및 인스턴스화 모듈은 OO-VRML 형식의 가상 세계를 읽어들이어서 객체로 바꾸어주고 그 객체들을 인스턴스화하여 OO-VRML 장면 그래프로 바꾸어준다. 장면 그래프는 실행 모듈은 발생한 이벤트들을 처리하여 인스턴스의 필드 값을 바꾸어준다. 브라우저 모듈은 가상 세계를 화면에 나타내 주고 사용자 입력과 내비게이션을 처리한다.

1. 서론

객체 지향 언어에서 객체는(object)는 그 안에 객체의 속성(property)과 행위 코드(behavior code)를 포함한다. 객체간의 정보 전달은 메시지(message)를 통해 이루어진다. 객체는 상속(inheritance)을 통해서 부모 객체(parent object)의 속성과 행위 코드를 자식 객체(child object)에게 물려줄 수 있다. 자식 객체는 부모 객체의 특징을 변화시키며 속성과 행위 코드를 추가시킬 수 있다. 상속을 통해서 재사용(reuse)이 가능해진다[1].

VRML은 객체 지향적인 요소를 포함하고 있다[2]. VRML의 노드의 필드들은 객체 지향 언어의 속성을 나타낸다고 할 수 있다. VRML의 이벤트인(eventIn), 이벤트아웃(eventOut), 라우트(route) 그리고 스크립트 노드(script node)는 객체 지향 언어의 메시지 전달이나 행위 코드를 나타낸다고 할 수 있다. 노드 사이에 전달되는 이벤트(event)는 객체 지향 언어에서의 메시지에 해당하는 것이다. VRML의 노드들은 DEF, USE 문을 통해서 재사용될 수 있고, PROTOTYPE 문을 통해서 새로운 노드를 정의할 수 있다. 그러나 VRML은 노드 안에 속성과 코드가 함께 있지 않으므로 정보 은닉(information hiding)을 지원할 수 없다.

본 연구는 첨단정보기술 연구센터를 통하여 과학재단의 지원을 받았음.

OO-VRML은 VRML에 객체 지향성(object orientation)을 향상시켜 설계한 언어이다[3]. VRML의 노드는 OO-VRML의 객체에 해당한다. OO-VRML의 객체는 VRML의 노드와 ROUTE문을 포함하며 Script문에 의해서 분리되어 있던 이벤트 핸들러(event handler)를 포함하고 있다. 객체는 상속에 의해서 만들어진다. 클래스(class)는 객체의 타입을 정의하며, 클래스 자체는 하나의 객체로 취급된다. OO-VRML은 기본적으로 객체 기반(object based)의 언어이지만 타입의 처리에 있어서는 클래스 기반(class based) 언어의 특징을 포함하고 있다.

OO-VRML로 만들어진 가상 세계를 해석하여 화면에 보여 주고 사용자의 입력을 받기 위해서 브라우저(browser)가 필요하다. 본 논문에서는 OO-VRML을 해석하고 보여줄 수 있는 브라우저를 설계하고 구현하는 방법에 대해서 설명한다.

2. OO-VRML의 특징

2.1 객체 및 인스턴스

OO-VRML은 객체의 집합으로 구성된다. 객체는 객체는 속성들과 이벤트 핸들러들로 구성된다. 속성에는 field, exposedField, eventIn, eventOut이 올 수 있다. 객체 내에서는 속성들에 자유롭게 접근할 수 있으나 다른 객체에서는 속성들에 자유롭게 접근할 수 없다. 객체 외부에서 읽을 수 있는 속성은 exposedField이다. 객체 외부에서 이벤트가 전달될 수 있는 속성은 eventIn과 eventOut이다. Field는 객체 내부에서 사용하기 위한 상수이다.

ExposedField는 객체 외부에서 읽을 수 있는 변수이다. ExposedField P가 객체 내에 선언되면 내부적으로는 다음과 같이 변환된다.

```
eventIn <type> P.SET
field <type> P
eventOut <type> P.CHANGE
```

Event In P.SET 은 외부에서 이벤트를 받아들이고 그 이벤트의 값을 field P에 치환해주는 내부적으로 정의된 이벤트 핸들러를 호출한다. Field P 값이 바뀌면 eventOut P.CHANGE 가 새로운 이벤트들을 발생시킨다. Event In 은 객체에 들어오는 이벤트를 받아서 그 event In 에 해당하는 이벤트 핸들러를 호출해준다. Event Out 은 이벤트가 전달되는 패스(path)를 지정해 준다. 패스는 객체의 event In 이나 event Out 을 가리키는 참조자(reference)이다.

객체는 바로 렌더링(rendering)하는 데 사용될 수 없다. 객체는 인스턴스화(instantiation)의 과정을 거쳐 인스턴스(instance)로 만들어질 때 렌더링될 수 있다. 객체는 다른 객체에 의해서 참조될 때 인스턴스화된다.

2.2 상속

객체가 상속되었을 때, 객체의 속성들은 상속 모드에 따라서 다른 값을 가지게 된다. 상속 모드는 copy, inherit, local 의 세 가지이다. Copy 모드로 상속되었을 때는 새로운 객체에 원본 객체(source object)의 속성의 값이 복사된다. Inherit 모드로 상속되었을 때는 새로 만들어진 객체의 속성들은 원본 객체의 속성들을 가리키는 패스가 된다. 이렇게 상속이 되었을 때는 원본 객체의 속성들을 새로 만들어진 객체가 공유하게 된다. 상속 모드가 local 이라면 새로 만들어진 객체는 속성에 UNDEFINED 라는 값을 가지며, 인스턴스화될 때 타입에 따른 초기값, 예를 들면 0, [0 0 ...] 같은 것을 갖게 된다.

2.3 클래스

OO-VRML에서 클래스는 객체의 타입을 결정한다. 그러나 클래스도 객체와 같이 인스턴스화되기 때문에 객체의 기능을 갖고 있다.

클래스는 상위 클래스(superclass)를 상속해서 만들어진다. 클래스는 인터페이스(interface)를 구현할 수 있다. 클래스의 속성은 외부 속성(external attribute)과 내부 속성(internal attribute) 으로 나뉘어서 정의될 수 있다. 외부 속성은 객체 생성시 새로운 값을 지정할 수 있는 것들이다. 내부 속성은 고유의 값이므로 어떤 클래스를 상속해서 객체나 다른 클래스를 만들 때 변경할 수 없는 것이다. 외부 속성을 정의할 때는 상위 클래스의 내용을 그대로 받아들이거나 새로운 속성을 추가할 수 있다.

클래스를 정의할 때는 이벤트 핸들러 까지 함께 정의하게 된다. 이벤트 핸들러는 이벤트 인을 통해 이벤트를 받아서 해당되는 코드(code)를 정의한다. 코드는 자바스크립트(javascript)로 정의된다. 코드

안에서는 해당하는 객체의 외부 속성과 지역 변수를 사용할 수 있다.

3. 브라우저의 구조

브라우저가 하는 일은 OO-VRML 파일을 읽어서 해석하는 일, 해석한 가상 세계를 화면에 나타내는 일, 외부 입력을 받는 일의 세 가지이다[4]. 브라우저는 크게 세 부분으로 구성되는데 파일 분석기는 파일을 해석하는 일을 담당하며, 실행 모듈은 해석된 가상 세계를 변화시키는 일, 브라우저 모듈은 입력과 출력을 담당한다.

3.1 파일 해석기 및 인스턴스화 모듈

파일 해석기는 가상 세계를 그리기 위한 OO-VRML의 기본 노드들에 대한 자료와 OO-VRML 파일을 받아들이어서 그것을 해석해서 객체를 만들어준다. 파일 해석기는 파서 생성기(parser generator) JavaCC 를 사용해서 만들어졌다. OO-VRML의 문법은 문맥 자유(context free) 문법이 아니기 때문에 파서 생성기만을 이용하면 문법 검사를 완벽하게 할 수 없다. 따라서 파서 생성기를 수정하여 문맥에 따라 달라지는 의미를 분석하는 부분을 추가하였다.

객체는 다른 객체에 의해서 참조될 때 인스턴스로 바뀐다. 또 객체의 속성에 따라 여러 번 참조될 때 인스턴스가 여러 개 생길 수 있다. 이것을 인스턴스화 모듈이 담당한다. 객체가 인스턴스화되었을 때에 그 객체의 속성 중에서 패스는 객체를 가리키던 것에서 해당되는 인스턴스를 가리키는 것으로 변화된다. ExposedField도 인스턴스화 과정을 통해서 field, eventOut, eventIn으로 변환된다. 인스턴스들은 트리 형태로 구성되므로 이것을 장면 그래프(scene graph)라고 부른다

이벤트 핸들러가 자바스크립트 코드로 정의된 경우 자바스크립트 인터프리터(interpreter)를 통해 다음과 같은 과정을 통해 처리될 수 있다.

- 자바스크립트를 위한 컨텍스트(context)를 만들고 초기화한다.
- 인스턴스의 외부 속성과 지역 변수의 값을 컨텍스트에 저장한다.
- 자바스크립트코드를 실행한다.
- 변환된 속성의 값을 인스턴스에 기록한다.

3.2 실행 모듈

실행 모듈은 다음과 같은 세 단계가 반복으로 이루어진다.

- 인스턴스 처리
- 이벤트 처리
- field 값 계산

1) 인스턴스 처리

인스턴스 중에는 다른 인스턴스에 의해 전달된 이벤트에 의하지 않고 자체적으로 이벤트를 발생시키는 것이 있다. TimeSensor, ProximitySensor 와 같은 환경 센서(environmental sensor)가 이에 해당한다. 이 단계에서는 각 인스턴스가 자체적으로 이벤트를 발생할 것인지를 검사해서 그 인스턴스에 해당하는 이벤트를 발생시켜준다.

이벤트는 내용(content)와 타입(type)으로 구성되며 처리되지 않은 이벤트는 각 인스턴스의 eventIn과 eventOut에 저장되어 있게 된다.

2) 이벤트 처리

실행 루프의 전단계에서 발생하여 저장된 이벤트나 인스턴스의 처리과정에서 발생하여 저장된 이벤트들을 처리해준다. 개념적(conceptual)으로는 모든 이벤트들이 동시에 처리되어야 하지만 그렇게 구현할 수 없기 때문에 각 장면 그래프를 트래버스(traverse)하면서 각 인스턴스에 저장된 이벤트들을 차례로 처리해주게 된다. EventIn에 저장된 이벤트의 내용은 이벤트 핸들러의 인자(parameter)로 넘어가며 해당하는 이벤트 핸들러를 실행한다. 이벤트 핸들러는 실행 과정을 통해서 eventOut에 값을 쓰게 될 수 있다. EventOut에 저장된 내용은 eventIn으로 전달되게 된다. EventOut을 처리하게 되면 다른 인스턴스로 이벤트가 전달될 수 있다. 그렇게 되면 그 인스턴스에 대해 다시 이벤트 처리를 거치게 된다. 이 과정에서 순환 이벤트(recursive event)가 발생할 수가 있게 된다. 순환 이벤트가 발생하는 것을 방지하기 위해서 어떤 인스턴스가 처리중에 있다면 그것을 나타낼 수 있도록 해서 완전히 처리되지 않은 인스턴스에 또 다른 이벤트가 전달될 때는 그 인스턴스의 이벤트 처리는 다음 루프에서 하게 한다.

3) field 값 계산

필드값을 계산한다는 것은 렌더링을 위해서 패스가 가리키고 있는 값을 인스턴스에 저장한다는 것이다. 다른 객체의 속성을 참조하게 되면 순환 참조(circular reference)가 일어날 수가 있다. 이 경우는 field 타입에 따른 기본 값을 갖게 된다.

4. 브라우저 모듈

4.1 렌더링(rendering)

자바에서 OpenGL을 이용하기 위해서 iicm의 GE3D를 이용했다. GE3D는 JNI(java native interface)를 이용해서 OpenGL의 함수를 호출하며 VRML에 사용되는 함수를 라이브러리로 만든 것이다. GE3D를 사용해서 OpenGL 캔버스(canvas)를 사용할 수 있다. 렌더링은 장면 그래프를 트래버스하며 각 노드에 따라서 해당하는 GE3D 함수를 호출하는 것을 통해 이루어진다.

4.2 사용자 입력

사용자 입력을 통해서 가상 세계를 네비게이션(navigation)할 수 있다. 입력은 마우스와 키보드로 구성되며 Flip 모드와, Walk 모드로 구성된다. Flip 모드는 관찰자가 움직이지 않고 가상 세계를 회전, 확대, 축소할 수 있다. Walk 모드는 가상 세계에서 관찰자가 움직이게 한다.

센서 노드(sensor node)의 경우에는 마우스가 그 위에 있거나 클릭할 때 노드에 이벤트가 전달되어야 한다. 이 때 노드를 찾아내는 부분은 Vrwave 브라우저의 유틸리티를 수정하여 사용했다[5].

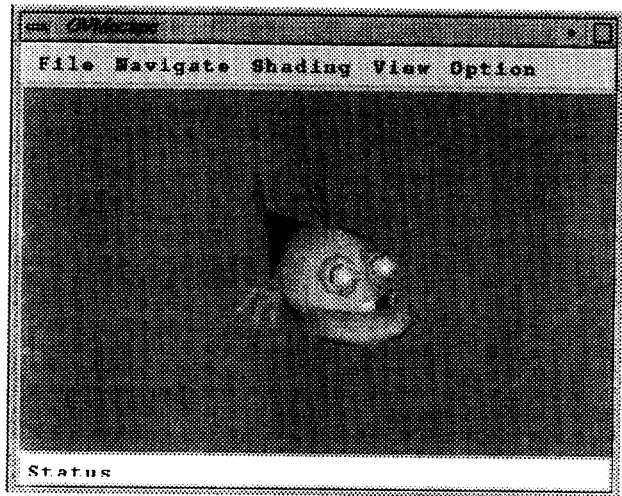


그림 1 OO -VRML로 만든 움직이는 물고기

5. 결론 및 향후 과제

OO-VRML의 이벤트 처리 방식, 동적 바인딩, 정보 은닉, 상속 등을 활용할 수 있는 브라우저를 설계하고 구현하였다. OO-VRML 브라우저는 자바와 OpenGL을 기반으로 IRIX6.4, Solaris 2.6에서 작동한다. 그림 1은 OO-VRML을 브라우저로 표현한 모습이다.

향후 과제는 다음과 같다. 현재 브라우저에서 렌더링을 C로 하고 있는 부분은 Java3D를 이용해서 자바로 바꿀 수 있다. 또, 자바스크립트가 브라우저를 제어한다든지 다른 노드를 제어하는 부분이 구현되어 있지 않다. 이것을 위해서 이벤트 핸들러와 다른 노드와의 연결에 대한 규정이 추가로 필요하다.

6. 참고 문헌

[1] Bertrand Meyer, Object -Oriented Software Construction 2nd edition, Prentice Hall, 1997
 [2] K.Matsuda, Y.Honda and R.Lea, "Sony's approach to behavior and scripting aspects of VRML: an Object-Oriented Perspective", <http://www.csl.sony.co.jp/project/Vs/proposal/behascr.html>
 [3] Sungwoo Park, Taisook Han, The Design Principles of Objected -Oriented VRML. In the Proceedings of the Workshop OO and VRML at VRML98, Monterey California, USA, 1998
 [4] The Virtual Reality Modeling Language, International Standard ISO/IEC 14772-1:1997
 [5] Keith Andrews, Andreas Pesendorfer, Michael Pichler, Karl Heinz Wagenbrunn, Josef Wolte. Looking Inside Vrwave : The Architecture and Interface of the Vrwave VRML97 Browser. In the Proceedings of the 3rd Symposium on the Virtual Reality Modeling Language, Monterey California, USA, 1998