

# X3D - Java 3D 번역기의 설계

○  
김 윤 기, 오 세 만  
동국대학교 컴퓨터공학과  
E-mail: {ripitup, smoh}@dgu.ac.kr

## Design of an X3D to Java 3D translator

○  
Yunkee Kim, Seman Oh  
Dept. of Computer Engineering  
Dongguk University

### 요 약

인터넷의 빠른 발전과 하드웨어 속도의 향상에 의해 가상현실이 점차 일반 사용자에게도 가깝게 다가서고 있다. 인터넷 가상현실 언어의 표준인 VRML이 XML의 개발과 발전에 의해 차기 버전을 XML의 장점인 용이한 확장성을 수용하는 X3D라 명명하고 표준화를 진행 중이다. VRML의 경우 전송 시 파일 크기의 문제를 안고 있고, X3D로 진보하면서 역시 같은 문제를 해결하기 위하여 보다 효율적인 전송 포맷에 대한 연구가 진행되고 있다. X3D의 효율적 전송을 위한 바이너리 스트림 개발의 일환으로 바이트코드를 제안하고, 강력한 Java 3D API와 플랫폼 독립적인 바이트코드의 장점을 살려 표준화 단계인 X3D를 별도의 브라우저 없이도 Java 애플릿을 이용하여 재생할 수 있도록 하고자 한다.

이를 위한 방법으로 X3D를 Java 3D로 변환하는 번역기가 필요하다. 따라서 본 논문에서는 X3D-Java 3D 번역기의 구조를 설계하고 구현 방법을 제시하고자 한다. XML 파서를 이용하여 X3D를 파싱하고 그의 출력인 AST를 순회하면서 Java 3D 파일을 생성한다. X3D DTD와 Java 3D 클래스의 구조, 계층 관계 정보를 독립된 자료로 작성하여 계속 변하게 될 두 언어의 변화에 유동적으로 대처하면서 소스 코드의 큰 변화 없이 이용할 수 있게 설계하고자 한다.

### 1. 서론

1994년에 VRML 1.0이 소개된 후, 인터넷의 대중화와 컴퓨터 그래픽 기술의 발달, VRML의 지속적인 발전에 힘입어 1997년 ISO/IEC 14772에서 VRML 2.0을 기초로 한 VRML97을 인터넷 가상현실의 표준언어로 채택하였다. XML의 발전으로 앞으로의 인터넷 문서가 XML로 변해가는 추세에 맞춰 XML의 장점인 용이한 확장성을 VRML에 도입 하고자 하는 목적으로 1999년 VRML의 차기 버전을 X3D로 명명하였다. 동년에 VRML200X-X3D initial draft가 발표되어 현재까지 Web3D 컨소시엄에서 지속적인 표준화 작업이 진행 중에 있다.

X3D - Java 3D 번역기는 X3D 파일을 Java 3D 파일로 번역하여 이것을 컴파일 함으로써 바이트코드를 생성할 수 있게 한다. 이렇게 생성된 바이트코드는 차 후 XML과 X3D가 대중적으로 활성화 될 시점에서 현재에도 거론되고 있는 전송량 축소를 위한 바

이너리 스트림의 하나로 사용될 수 있다. 또한 Java 3D를 기반으로 한 X3D 브라우저의 한 방법론적인 수단으로서의 의미도 가진다. XML의 특성을 이용하여 X3D의 DTD가 아직 확정되지 않은 지금의 시점에서 앞으로의 변화에도 XML 파서를 유동적으로 사용할 수 있도록 하여 X3D 파일을 효율적으로 Java 3D 코드로 변환 한다. 이를 통해 얻어지는 바이트코드를 X3D 파일의 전송에 적용하여 UTF-8 형식인 X3D 파일 크기를 줄이기 위해 압축하고, 브라우저 시에 또다시 해제하는 번거로운 과정을 생략할 수 있다. 별도의 재생기 없이 기존 인터넷 브라우저 환경에서도 X3D를 Java 애플릿 형식으로 브라우저 할 수 있게 되어 X3D의 개발 과정에도 응용할 수 있다.

본 논문에서는 배경 연구로 X3D와 Java 3D의 주요 노드들과 클래스에 대하여 2장에서 다루고, 3장에서는 실제 설계 내용을 설명하며, 4장에서는 결론과 향후 연구방향에 대해 기술한다

## 2. 배경 연구

### 2.1 X3D의 노드와 Java 3D 클래스

X3D는 3D 콘텐츠를 표현하는 널리 사용되고 있는 수단을 제공하고, Java 3D는 3D 프로그래밍에 강력한 API를 제공하고 있다. Java 3D는 플랫폼 독립적인 특성을 갖고 있어, 쉬운 3D 콘텐츠 표현의 장점을 갖는 X3D를 Java 3D 바이트코드로의 변환은 두 언어의 특장점을 결합 한다는 의미를 갖는다. 또한 두 언어는 Intel, SGI, Apple, Sun사의 공동 연구로 진행 되었기 때문에 각 노드에 대응되는 클래스와 장면 그래프(scene graph)에서 상호 공통되는 부분을 많이 갖고 있다. 이 점을 감안하여 두 언어의 의미적 대응 관계를 도출함으로써 번역에 용이함을 추구한다. 아래의 표는 X3D의 노드 중 대표적인 노드와 이에 대응되는 Java 3D의 클래스를 비교한 것이다.

X3D	Java 3D
ColorInterpolator	ColorInterpolator
OrientationInterpolator	RotationInterpolator
PositionInterpolator	PositionInterpolator
ScalarInterpolator	ScaleInterpolator
Appearance	Appearance
Background	Background
Group	Group
IndexedFaceSet	IndexedGeometryArray
IndexedLineSet	IndexedLineArray
ViewPoint	View

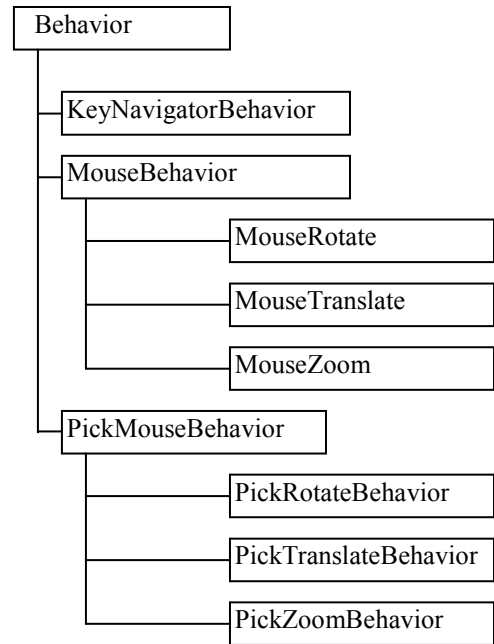
[표 1] X3D 노드와 Java 3D 클래스 비교

[표 1]에서 보듯이 두 언어의 노드와 클래스 사이에 유사 또는 동일하게 대응되는 것이 있고, 어느 한 언어에만 있는 노드들이나 클래스들이 있다. 성격이 동일한 노드와 클래스 사이의 변환 이외에 차이를 보이고 있는 변환에서는 Java 3D의 API를 이용하여 실제 X3D에서 의도하는 바를 충실히 표현한다. 이 수행 과정에서 후에 설명할 X3D to Java 3D match table의 정보를 이용해 서로 대응되는 클래스들을 선택한다.

### 2.2 네비게이션 관련 Java 3D 클래스

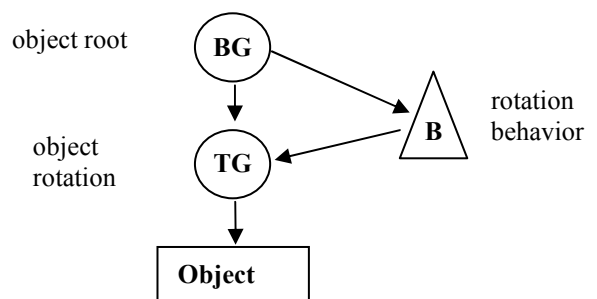
VRML과 X3D는 아바타라는 개념이 있다. 이는 가상현실 안에서 사용자의 분신 역할을 하며 가상현실

공간을 자유롭게 이동하는 기준이다. 이러한 기능을 수행할 수 있는 Java 3D 클래스는 다음과 같은 Behavior 클래스들로 구성된다. 이 클래스들을 이용하여 사용자의 네비게이션 상태의 역 변환을 그룹 노드의 코디네이터에 가해서 Java 3D로 이루어진 가상현실의 네비게이션을 수행한다.



[그림 1] 네비게이션 관련 Java 3D클래스 계층

네비게이션을 위한 [그림 1]의 Behavior 클래스들은 장면 그래프 내에서 각 Transform Group에 연결 되어있다. 사용자의 키보드와 마우스 입력은 Transform Group 내의 필드 값을 변경시키고, 연결 된 Transform Group의 상태에 변화를 가함으로써 가상현실 내에서의 자유로운 이동을 구현할 수 있다.



[그림 2] 네비게이션을 위한 Java 3D 클래스

### 2.3 애니메이션을 수행하는 주요 Java 3D 클래스

- ColorInterpolator : 색상 변화
- RotationInterpolator : 회전
- PositionInterpolator : 위치 이동
- ScaleInterpolator : 크기 변화
- Pathinterpolator : 경로 이동

### 2.4 인터넷 하이퍼링크 구현 클래스

X3D에서는 3D 객체에 연결된 Anchor 노드를 이용하여 인터넷 하이퍼링크처럼 동작하게 한다. Java 3D에서는 이를 위한 클래스가 안에 존재하지 않는다. Java 애플릿과 java.net.URL 클래스를 이용하여 Java 3D 객체에 하이퍼링크 속성을 주어 일반적인 하이퍼링크와 동일한 동작을 하게 한다.

## 3. 번역기 구현 모델

### 3.1 번역기의 구조

X3D-Java 3D 번역기는 [그림 3]에서 보는 바와 같이 X3D를 XML 파서를 이용하여 파싱한 결과인 AST를 입력으로 받는다. X3D의 각 노드와 Java 3D 클래스의 대응 관계 정보를 담고 있는 X3D to Java 3D match table과 생성할 Java 3D 클래스의 구조 정보를 담고 있는 Java 3D class descriptor를 내부적으로 이용한다. 프로그램의 효율성을 위해 X3D 노드의 성격에 따라 다섯 개의 프로세서로 나뉘어 해당 클래스를 전담하여 처리하며, 번역기의 출력으로 Java 3D 파일을 낸다. 이 Java 3D 파일의 컴파일은 javac를 이용한다.

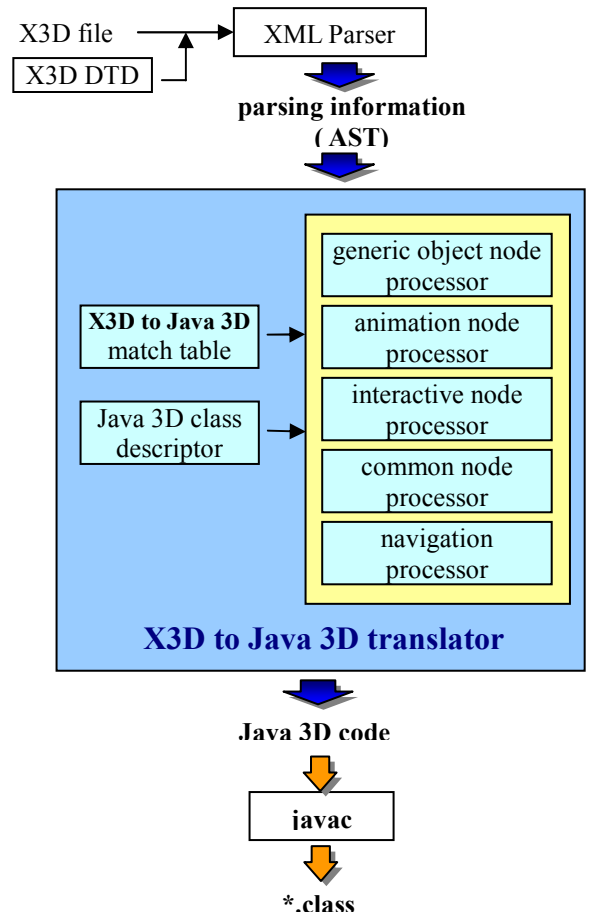
### 3.2 파서

본 논문에서 제안하는 파서는 기존의 XML 파서를 이용하여 X3D 파일을 파싱하며 AST 형태의 출력을 내는 파서를 선택해서 사용한다. 현재 공개되어있는 XML 파서는 대부분 소스도 공개되어있어 제작하고자 하는 프로그램에 수월한 통합이 가능하다. DTD는 현재 Web3D 컨소시엄에서 발표한 X3D\_draft.dtd(version 0.7)를 사용하며 추후에 계속되는 업데이트에 따라 그에 따르는 추가적인 사항과 변경 사항을 적용한다.

### 3.3 X3D to Java 3D match table

파서로부터 입력으로 받은 노드의 이름으로 본 테이블을 검색하여 그에 해당하는 Java 3D의 클래스를 선택 하게 된다. 테이블의 매치는 X3D와 Java 3D 양 쪽 모두 동일하게 존재할 경우 1:1 대응이 될 수 있으며, 노드의 특성상 1:N과 N:1 대응도 존재한다.

그리고 X3D 노드의 필드와 Java 3D의 클래스의 필드의 상관 관계도 이 테이블에 표현된다.



[그림 3] 번역기 구현 모델

### 3.4 Java 3D class Descriptor

선택된 Java 3D class의 구조 정보가 담겨 있는 자료이며 X3D to Java 3D match table에서 선택된 클래스의 슈퍼클래스, 필드구조, 메소드 등의 정보를 읽어 그 형태에 맞게 적절한 클래스를 생성하는데 사용된다. 이 정보를 따로 두고 실행 시간에 로드하여 적용함으로써 앞으로의 Java 3D의 변화에도 많은 코드의 변화 없이 사용할 수 있게 한다.

### 3.5 노드의 성격과 목적에 따른 전담 프로세서

X3D는 현재 VRML97의 노드들을 앞으로 X3D의 표준화가 진행 되면서 여러 가지 변화가 있겠지만 현재 발표된 문서에서는 기본적으로 같은 형태로 정의되고 있다. X3D의 노드는 그룹 노드, 특별 노드, 일반 노드, 센서 노드, 기하 노드, 속성 노드, 모양 노드, 애니메이션 노드, 부가 노드의 9가지 노드 그룹으로 이루어져 있다. 이를 크게 애니메이션 관련 노드, 객체의 형태와 속성 관련 노드(generic object

node), 사용자와의 상호작용과 인터넷 하이퍼링크를 위한 노드(interactive node), 사운드, 라이트, 스크립트 등의 관련노드(common node), 그리고 네비게이션을 위한 노드들로 나뉘어 각각의 클래스를 전달하여 생성하는 프로세서들로 이루어진다. 각 프로세서들의 담당은 다음과 같다.

**-Generic object node processor**

- Geometry
- Geometry properties
- Appearance

**-Animation node processor**

- Interpolators,

**-Interactive node processor**

- Sensors

**-Common node processor**

- Common nodes

**-Navigation processor**

- User action

각 프로세서들은 X3D to Java 3D match table과 Java 3D class descriptor의 정보를 이용하여 XML 파서의 출력인 AST를 순회 하면서 \*.java 파일을 생성하게 된다.

**3.6 바이트코드의 생성**

Java 3D 파일의 바이트코드로의 컴파일은 javac 컴파일러를 사용한다. 이 과정은 편의성을 위해 커맨드 라인에서 직접 입력을 하는 형식을 배제하고 일괄 작업으로 \*.class 파일을 출력한다.

**4. 결론 및 향후 연구 방향**

현재 Web3D 컨소시엄에서도 효율적인 전송과 그를 위한 바이너리 스트림 개발, 편리한 VRML과 풍부한 Java 3D의 API 활용성에 관하여 VRML-Java 3D working group이 활동하고 있다. 본 논문에서 제시하는 X3D - Java 3D 번역기는 이를 통해 VRML이 X3D로 발전하고 있는 과정에 효율적 전송 포맷의 하나로 인터넷 환경에서 높은 신뢰도와 넓은 활용성을 갖고 있으며 강력한 3D 프로그래밍 API를 제공하는 Java 3D의 바이트코드를 제안하고 그를 위한 방법을 제공한다. 또한 표준화 작업 중인 X3D의 편리한 브라우저로 Java의 애플릿을 사용할 수 있도록 하여 별도의 틀 제작없이 재생할 수 있게 한다. 이와 같이 X3D 표준화 단계에서의 손쉬운 X3D 브라우저 환경을 제공하여 앞으로의 발전성에 기여할 수 있으며, 인터넷 가상현실 분야에 있어서 바이트

코드의 활용성을 증대시킨다.

향후 연구 방향은 X3D-Java 3D 번역기를 구현하면서 편리한 번역을 위해 언어의 구조와 의미를 표현하는 자료의 효과적인 형식과, 같은 가상현실을 표현하는 gzipped UTF-8 형식과 Java 3D 바이트코드의 전송시간과 로딩타임을 분석하여 보다 효율적인 바이너리 스트림에 대하여 연구할 계획이다.

**참고문헌**

- [1] Java 3D™ API Collateral, <http://java.sun.com/products/java-media/3D/collateral/>, 1999.
- [2] Kirk Brown, Daniel Petersen, Ready-to-Run Java 3D, WILEY Press, 1999.
- [3] VRML97 specification, <http://www.web3d.org/Specifications/VRML97/>, 1999.
- [4] VRML200x-X3D initial draft, <http://www.web3d.org/TaskGroups/x3d/specification/>, 1999.
- [5] VRML-Java3d Working Group, <http://www.vrml.org/WorkingGroups/vrml-java3d/>.
- [6] Web3D Consortium - X3D Documents, <http://www.web3d.org/TaskGroups/x3d/>, 1999.
- [7] 오세만, 이용규, 김지인, 인터넷 프로그래밍, 생능출판사, 2000.