Design of Web Based Solid Modeler

Bo-yul Yoon*, Eung-kon Kim*

웹 기반 솔리드 모델러의 설계

윤보열*, 김응곤*

요 약

본 논문에서는 플랫폼에 구애받지 않으며, 특정한 3차원 그래픽스 소프트웨어 없이 웹 상에서 솔리드 모델을 설계할 수 있는 웹 기반 3차원 솔리드 모델러를 설계하였다. Java 3D를 이용하여 개발하기 위한 시스템 라이브러리의 설계 결과와 구현방법을 제시하였다. 클라이언트는 솔리드 모델러 서버에 접속하여 설계를 한 후 원하는 파일 형태로 저장할 수 있으며, 다른 CAD 시스템에서 기존에 설계된 데이터를 볼 수 있도록 하며, 렌더링 및 애니메이션과 같은 3차원 그래픽스 기능이 가능함으로써 정확한 설계를 지원하도록 한다.

Abstract

We designed a 3D solid modeler based on the web, which was independent from platforms, which could be executed without 3D graphics softwares. In this paper, we show the design of system libraries and how to implement in Java 3D. A client connects to the solid modeler server, design solid model, stores the data as the various file format, and displays the data from other CAD systems. This solid modeler can support the detail design as 3D graphics features such as viewing, rendering, animating are available.

I. Introduction

As the designs using computer have been widespread and generalized, the various kinds of CAD systems appear around us. However, the cost of commercial CAD system is very high and it is not easy for everyone to buy and use the system. The present CAD systems are depended on platform and they must be ported according to the environment of each platform.

* Department of Computer Science, Sunchon National University
접수일자 : 1999년 12월 4일

The porting is not a simple problem because the graphic interface of CAD application is very important and most of data must be displayed in graphic image. The developers who want to program in multi-platform must code properly in each platform, and they must correct and test the programming codes when the program need to be changed.

If a CAD system is developed in Java on the Web, these problems will be solved easily. Because Java codes are compiled to independent byte code, not to machine code, they run well on PC as well as SGI workstation and SPARC station. Among the Internet services, the Web service supports user friendly interface and environment of multimedia, so the users are explosively increasing and various services on the web appear in these days. This tendency of changing can be seen in environment of CAD engineering, too. For example, the design information can be shared on the Internet and collaborative project can be progressed sharing the data. This advantage of Internet has changed entirely the environment of CAD system, and Internet is changing for supplying this area of engineering[1].

The most important characteristic features of such applications are their time-, location-, and platform-independent availability, free of charge, no cost for production and installation, and little effort for updating the underlying database[2].

Commercial modeling systems have been available for many years. Yet they are still limited in functionality, difficult to use, and difficult to integrate into application systems. There are no easy-to-use tools for displaying and modifying even simple solid models as part of a web page. VRML(Virtual Reality Modeling Language) can be used to describe 3D shapes in virtual worlds, but has limited features in interactivity[3].

If a solid modeler is developed using Java 3D[4], some advantages are there. First, as the Java 3D is expanded Java API, we can get the same results with different systems, and we can develop it properly in each platform with less time and labor. Second, Java 3D includes the functions and conveniences of developed graphic libraries and the new techniques which do not use in other libraries Third, Java 3D can be executed on the web with Java entirely.

So far most of solid modelers can be executed only in a certain platform. Though they were developed in Java to be run in any platform, if without Java 3D, most of functions must be by coding. So the developing solid modeler need much time and have some limitation. In this paper, we will introduce the design of a 3D solid modeler based on the Web, which is independent from any platform, and executable without any 3D graphic software.

The organization of this paper follows. In chapter 2, we will mention about the related works on CAD system on the Web, in chapter 3 about the design of solid modeler proposing in this paper, in chapter 4 about the method of implementation of solid modeler, in chapter 5 about the conclusion and the researches in the future.

Ⅱ. Related Works

The CAD systems supporting Internet are developing remarkably in the area of automobile, home electronic equipment, and airplane. Advanced engineering CAD systems can search the modeling data in VRML stored in CAD system after loading on the Web Browser[1]. But these systems have the functions of searching the results of design in the Web and they don't have the ability of design.

In recent, the collaborative work using 3D visualization on the Web is being studied. We are satisfied with the study on the correspondence of sharing models, the area of control session. But the general methods for collaborative tools or design tools are poor at the real application. The developing the appropriate tools for each application is a general tendency[5]. Stephen developed the libraries for solid modeling on the Web[3]. It was only a class library developed for solid modeler in C and Java 2D, without 3D graphic functions such as rendering and animation.

In Korea, collaborative design/assembling evaluation system was developed using networks and virtual reality[5]. It constructed a library for parts which could be databased with part provider and part information being supplied information from provider. According to the library, after product designers search the information of parts and select the needed part, they can assemble them, and evaluate the design.

In considering the above studies, commercial system in Java for the design of solid model has not developed yet and we have found that there are many advantages of the developing in Java 3D on the Web.

Ⅲ. Design of Solid Modeler

A solid modeler consists of a set of Java classes to model and manipulate solids. The types of solid primitives are block, cylinder, cone, torus, pyramid, sphere, and so on. With these solid primitives, various solid models can be made by boolean operations such as INTERSECTION, DIFFERENCE, and UNION. And they cam be conversed by TRANSLATION, SCALING, REFLECTION, and ROTATION.

There are four different viewing frames for representing X-Y, Y-Z, X-Z, and parallel or perspective projection of solid model. For detail design, as the solid modeler is added to the function of rendering and animation, the solid models must be observed variously after designing. Controling view, rendering, and animating will be developed using a plenty of 3D graphic libraries in Java 3D.

1. The data structure of solid

The elementary components of a solid model are Solid, Face, Loop, Edge, Half edge, and Vertex in Fig.1. The core data structure was based on the half edge boundary-representation data structure proposed by M. Mantyla[6].

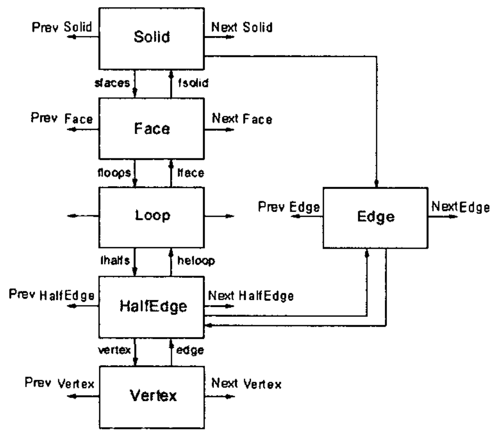

Fig. 1. Half edge data structure

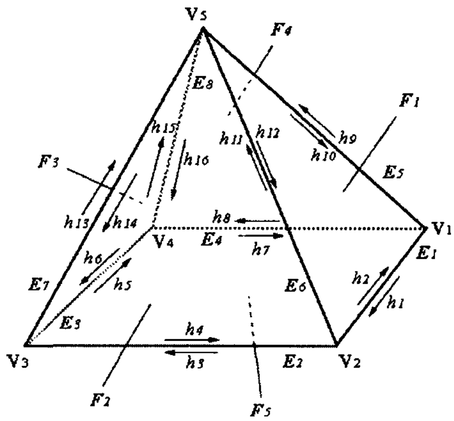

Fig. 2. Half edge representation of pyramid

Fig.2 shows the HalfEdge data structure of pyramid model[7]. Each face of a solid is a planar polygon whose shape is determined by the coordinates of the vertices of its edge loops. Each edge belongs to two edge-loops, each bounding a face. Each edge is hence broken down into two half edges, each oriented in the direction of one of the two edge loops. That is to say, one edge is divided into two, and each edge is used to each two faces sharing original edge. After each edge is divided into two half edges with opposite direction, half edge with consistent direction to each faces is stored as the structure of linked list. Fig. 3 is representing Java classes for data structures of solid model in this paper.

```
public class Edge extends Nodes
{
        HalfEdge  he1;
        HalfEdge  he2;
        Edge               nexte;
        Edge               preve;
}

public class Face extends Nodes
{       short              ID;
        Solid          fsolid;
        Loop               flout;
        Loop               floops;
        float              feq[4];
        Face               nextf;
        Face               prevf;
}
public class HalfEdge extends Nodes
{
        Edge               edge;
        Vertex             vertex;
        Loop               wloop;
        HalfEdge  nexthe;
        HalfEdge  prevhe;
}

public class Loop extends Nodes
{
        HalfEdge  ledg;
        Face               lface;
        Loop               nextl;
        Loop               prevl;
}
public class Solid extends Nodes
{
        short              ID;
        Face               sface;
        Edge               sedge;
        Vertex         svertex;
        Solid              nexts;
        Solid              prevs;
}
public class Vertex extends Nodes
{
        short              ID;
        HalfEdge  vedge;
        float              vcoord[4];
        Vertex             nextv;
        Vertex             prevv;
}
```

Fig. 3. Representing Java classes for the data

2. System Library

A solid modeler is developed in Java 3D, system library consists of many Java classes. System level classes are Workspace class, View class, ParallelView class, PerspectView class, and Solid class in Fig.4.

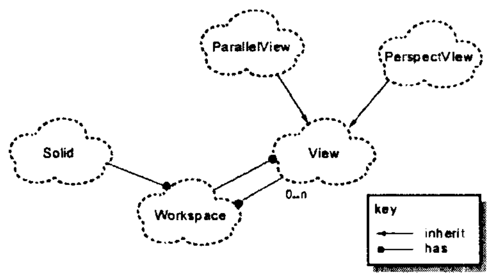

Fig. 4. Diagram of system level classes

The Workspace class provides functions which can be added some solids to workspace or removed some solids from workspace.

The View class is an abstract class, which provides functions for its children to display objects in workspace. The functions of hidden lines removal and clipping are the major responsibilities of this class. ParallelView and PerspectView inherit from the class View, they implement the parallel perspective transformation and the perspective transformation. Fig.5 shows the relation of Workspace class and multiple view.

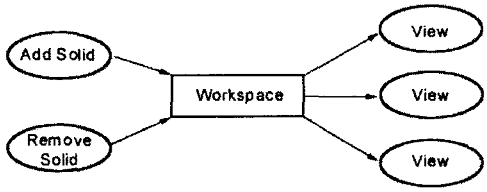

Fig. 5. Workspace class

The Solid primitive classes inherit from class Solid like Fig.6. They are related to make solid primitives such as Block, Cylinder, Sphere, Cone, Torus., and Pyramid.

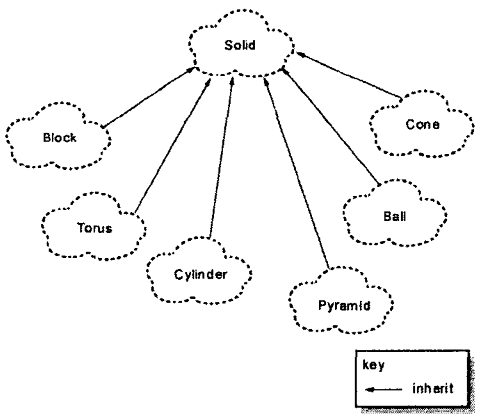

Fig. 6. Solid primitive class

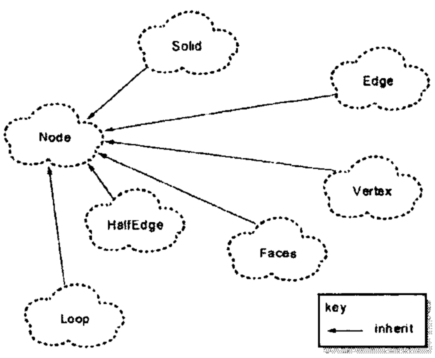

Fig. 7. Solid data structure class

The Solid data structure class consists like Fig.7, which is related to half edge data structure using in this solid modeler to represent solids. Class Node is a base class of the objects in the half edge data structure, Solid, Edge, Vertex, Face, HalfEdge, and Loop class inherit from class Node.

Ⅳ. IMPLEMENTATION

1. Scene Graph of Java 3D

The Java 3D API is an interface for writing programs to display and interact with 3D graphics. The API provides a collection of high-level constructs for creating and manipulating 3D geometry and structures for rendering that geometry. Java 3D provides the functions for creation of imagery, visualizations, animations, and interactive 3D graphics application programs[8].

Java 3D represents the solids through Scene Graph, tree structure. Fig.8 shows the Scene Graph, which may have Shape, Sound, Behavior, Light as terminal node.

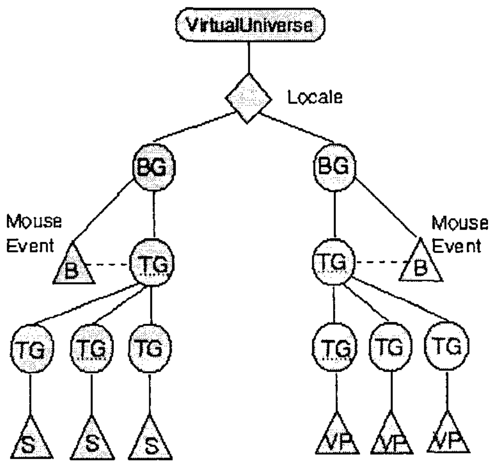

Fig. 8. Java 3D Scene Graph

The Geometry objects must be registered on Shape node. The translating and scaling of objects is available through Transform Group(TG), some TG's become a Branch Group(BG). Then BG loads to Locale, and it is active at the same time. If the objects are rotated by dragging mouse, BG duplicates TG node and Shape(S) node by Behavior(B), pastes upper TG. Fig.9 shows the process of Scene Graph creation.

```
create VirtualUniverse
create Locale from VirtualUniverse
create ViewPlatform(VP), View(V), Canvas3D(C)
attach ViewPlatform to Locale
create BranchGroup(BG)
create Shape3D(S), TransformGrop(TG) and
Behavior(B)
add to BranchGroup
attach BranchGroup to Locale
```

Fig. 9. The creation of Scene Graph

2. HTML for Java 3D

To run Java 3D on the web, the code must be applet format, and include tag <APPLET>, </APPLET> in HTML document. Because Java 3D is an expanded API, it is not execute on web browser. So browser support for Java 3D is available through the Java Plugin, and applet must be embedded as Plugin in HTML document like Fig.10[8].

```
for Netscape nevigator

<EMBED
Type="application/x-java-applet;version=1.2"
Width=760 Height=520 Align=BaseLine
Code="EVS.class"></EMBED>

for Internet Explorer

<OBJECT
ClassID="clsid:8AD9C840-044E-11D1-B3E9-008
05F499D93">
        <PARAM Name=code
Value="EVS.class">
        <PARAM Name=type
Value="application/x-java-applet;version=1.2">
</OBJECT>
```

Fig. 10. The example of HTML for Java Plugin

3. Creation of View and ViewPlatform

VirtualUniverse is a kind of studio offering virtual 3D space. The objects generating through Shape3D are stored in View film by camera, then they are shown on monitor through Canvas3D screen.

TransformGroup must be allowed to read and write in order to be modified TransformGroup by Behavior. PhysicalBody and PhysicalEnvironment set the environment for 3D objects. PhysicalBody is related to the positions of two eyes, two ears, and focus. PhysicalEnvironment is related to setting up the various environments.

```
vpRoot = new BranchGroup ();
trans3d = new Transform3D ();

trans3d.set (0.0f, 0.0f, 4.0f);

vpTrans = new TransformGroup (trans3d);
vpTrans.setCapability
   (TransformGroup.ALLOW_TRANSFORM_WRITE |
   TransformGroup.ALLOW_TRANSFORM_READ);

phBody = new PhysicalBody ();
phEnv = new PhysicalEnvironment ();
vp = new ViewPlatform ();
view = new View ();

view.addCanvas3D (canvas3D);
view.setPhysicalBody (phBody);
view.setPhysicalEnvironment (phEnv);
view.attachViewPlatform (vp);

vpTrans.addChild (vp);
vpRoot.addChild (vpTrans);
vpRoot.compile ();

locale.addBranchGraph (vpRoot);
```

Fig. 11. An example of creating View an ViewPlatform

4. Creation of Geometry Object

There are three major ways to create new geometric content[4]. One way uses the geometric utility classes for box, cone, cylinder, and sphere. Another way is for the programmer to specify the vertex coordinates for points, line segments, and/or polygon surfaces. A third way is to use a geometry loader. Fig.12 shows a class of Geometric Utility, Box. The Box geometric primitive creates 3D box visual objects. The defaults for length, width, and height are 2 meters, with the center at the origin, resulting in a cube with corners at (-1, -1, -1) and (1, 1, 1).

```
universe = new VirtualUniverse ();
locale = new Locale (universe);

objRoot = new BranchGroup ();
objTrans = new TransformGroup ();

objTrans.addChild (new Box (1.0f, 1.0f, 1.0f).getShape
());
objRoot.addChild (objTrans);
objRoot.compile ();

locale.addBranchGraph (objRoot);
```

Fig. 12. An example of creating Geometry Objec

5. Behavior Node

The variation of Object, the execution of event, the transformation of Scene Graph are available by Behavior Node. Fig.13 shows the example of rotating Behavior Node, which rotates the objects by dragging mouse.

```
import com.sun.j3d.utils.behaviors.mouse.*;

public class Rotate extends MouseRotate
{
        public Rotate (TransformGroup target) {
                super (target);
        }

        public void processMouseEvent (MouseEvent event) {
                super.processMouseEvent (event);
        }
}
```

Fig. 13. An example of rotating Behavior Node

6. The projection of solid object

The solid modeler proposed in this paper is implemented in Java 3D. It is possible to design solid models on connecting this solid modeler server on the web without any 3D software. Fig.14 shows the window of this solid modeler.

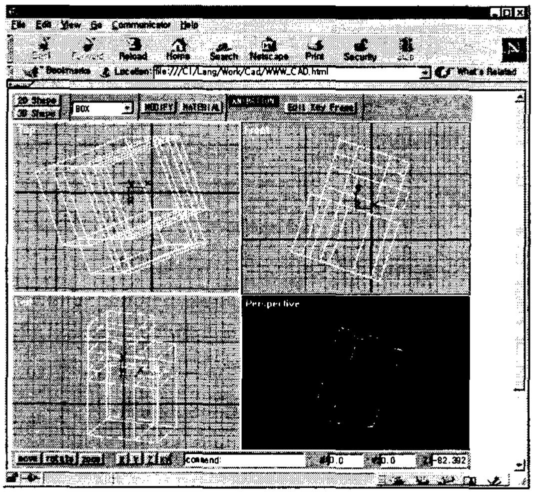

Fig. 14. The window of Solid modeler

There are four different viewing frames for representing X-Y, Y-Z, X-Z, and parallel or perspective projection of solid model. We can generate the primitive objects, such as box, cone, cylinder, and sphere on window. Then we can view the solid models variously, controling [move], [rotate], [zoom]. More over, the solid models can be animated and rendered.

V. Conclusion

In this paper, we designed the 3D solid modeler based on the Web, which was independent from platform, which could be executed without any 3D graphic software, and we proposed the way for implementing the modeler.

So far solid modelers have been executed in particular platforms. Though solid modelers are developed in Java, because they didn't use Java 3D, most of functions must be made by coding and the developer faced by the limitation of time and functions.

As this solid modeler is developed in Java 3D, it includes the functions and conveniences of developed graphic libraries and the new techniques which do not use in other libraries, for example, viewing, rendering, animation, and so on. Client connects to the modeler server, designs solid models, stores them as the various file format needed and the designed data can be searched by other CAD systems later. And this solid modeler can supply detail design adding to the 3D graphic features such as rendering and animation.

In the future works, this solid modeler will be completely implemented, collaborative works will be available sharing 3D CAD data on the web.

Acknowledgment

This work was supported by the RRC program of MOST and KOSEF through the Research and Development Center for Facility Automation and Information Systems(FAIS) at Yosu National University

References

1. Kwan-un Kim, "Extension of WWW and Change the Environment of Engineering", CAD&Graphics, Apr., pp114, 1997
2. Mathias Nousch, Bernhard Jung, "CAD on the World Wide Web : Virtual Assembly of Funiture with BEAVER", Faculty of Technplogy, University of Bielefeld, 1998
3. Stepen C.F. Chan, Vincent T.Y. Ng, Albert S.F. Au, "A solid modeling library for the World Wide Web", Computer Networks and ISDN Systems 30, 1998 pp.1853-1863, 1998
4. WWW, "Getting Started with the Java 3DTM API", http://java.sun.com/products/java-media/3D/collateral/#tutorial., 1999
5. Young-ho Kim, "Development of Collaborative Design/Assemble Evaluation System Using Network and Virtual Reality", Research and development report in Ministry of Information and Communication Jul., 1998
6. M. Mantyla, "An Introduction to Solid Modeling", Computer Science Press, Rockville, MD, 1988
7. Kun-woo Lee, "Computer Graphics and CAD", Young-ji Pub Co., 1994
8. WWW, "JAVATM Plug-in HTML Specification", http://java.sun.com/products/plugin/1.2/docs/tags.html., 1998

윤 보 열(Bo-yul Yoon)
1984년 전남대학교 학사
1988년 조선대학교 석사
1998년-현재 순천대학교 컴퓨터 과학과 박사과정
1984년-현재 순천매산고등학교 교사
*주관심분야 : 컴퓨터그래픽스, 멀티미디어, WBI

김 응 곤(Eung-Kon Kim)
1980년 2월 조선대학교 공학사
1987년 2월 한양대학교 공학석사
1992년 2월 조선대학교 공학박사
1993년 3월 -현재: 순천대학교 컴퓨터과학과 부교수
1997년 3월-1998년 2월 Univ. of California, Santa Cruz 객원교수
1987년 3월-1991년 2월 국방과학연구소 선임연구원
1984년 8월-1986년 8월 금성반도체(주) 연구소 연구원
*주관심분야 : 컴퓨터그래픽스, 영상처리, CAD