

《小特集》

Virtual Reality Modeling Language

金江和広*・松田晃一*

ABSTRACT VRML (Virtual Reality Modeling Language) is rapidly developing as the standard of the file format for the description of 3D virtual worlds on the Internet. VRML2.0, which supports behaviors of objects in a 3D scene and multimedia data like sound and video, is already standardized and now proposed for the ISO/IEC committee. The process of standardize VRML3.0, which will support multi-user environments, is now under way. This paper is an introduction to VRML world for researchers in various fields. We first review the VRML history to allow readers to understand it's concept and backgrounds. Next, we briefly explain the current standard VRML2.0 and display it's usefulness in various research fields. Last, we discuss the current status and future plans of the VRML.

1. はじめに

近年、インターネットは、Webの登場により急速な発展を見せている。これは、HTML (Hyper Text Markup Language) でホームページを作成することにより、誰もが簡単に情報を世界に発信することが可能になったことが大きな要因である。しかし、これまでのWebが扱う情報の中心は、テキストや静止画像などであり、3次元情報を扱うことはできなかった。そこで登場したのがVRML (Virtual Reality Modeling Language) である。VRMLは、インターネット上で仮想3次元空間を構築するために開発された言語であり、現在、デファクトスタンダードとなっている。VRMLは、バージョン2.0の発表後(1996年8月)、多くの研究機関やベンダーがそれに絡んだ製品を出し、活発な動きを見せている。本稿では、2章でその歴史を概観し、3章でその概要を示し、4章でVRMLの現状について述べる。

2. VRMLの歴史

VRMLは、歴史は浅いが急激な成長を見せている。その歴史を概観することは、VRMLの背景やコンセプトを理解するために、重要である。本章では、その歴史について述べる。表1にVRMLの年表を示す。

Virtual Reality Modeling Language. By Kazuhiro Kanee and Kouichi Matsuda (Sony Corporation, Software Laboratory, Architecture Laboratories).

*ソニー(株)アーキテクチャ研究所ソフトウェアラボラトリー

2.1 VRMLの誕生

1994年2月、Mark Pesce氏は、Tony Parisi氏の協力を得て、Web上で仮想3次元空間を実現する「Labyrinth」というWebブラウザのプロトタイプを完成させた。この成果は、1994年5月の第1回WWW国際会議で発表され、Webの更なる発展機能として、仮想3次元空間へのアプローチの重要性が認識された。そこで、次のステップとして、Mark Pesce氏らは、仮想3次元空間を記述するための標準言語の策定に乗り出した。これが、VRMLの歴史の始まりであった¹⁾。

VRMLの仕様の決定は、その作業をすべてメーリングリスト²⁾で行うというインターネットを利用した特有のアプローチがとられた。そこには、VRMLに興味のある人がすべて自由に参加でき、仕様に対する要求などが議論された。

そこでの議論を経て、VRMLの仕様の土台として、米Silicon Graphics社の提案したOpen InventorのASCIIファイルフォーマットが採用されることとなった。その理由は、

- すでに十分な仮想3次元空間の記述力とファイルフォーマットを持っている
- 3次元オブジェクトの形状や色、テキストなどの定義機能を備えており、拡張可能である
- グラフィックスのプロにより広く利用されており、十分な実績がある
- 米Silicon Graphics社がそれをパブリックドメイン

(注1) VRMLは、www-vrml@vag.vrml.orgにおいて、現在も議論されている。

表1 VRMLの年表

年月	内容
1994年5月	第1回 WWW 国際会議で、Mark Pesce 氏が VRML の構想を発表
1994年6月	VRML メーリングリストが開設
1995年5月	VRML1.0 の正式な仕様を発表
1996年1月	VRML2.0 の仕様候補の募集開始
1996年3月	投票の結果、Moving Worlds を VRML2.0 のベースに決定
1996年4月	VRML2.0 ドラフト#1 発表
1996年5月	VRML2.0 ドラフト#2 発表
1996年6月	VRML2.0 ドラフト#2b 発表
1996年7月	VRML2.0 ドラフト#3 および最終ドラフト発表
1996年8月	SIGGRAPH'96 にて、VRML2.0 の最終仕様発表

としたことなどであった。しかし、Open Inventor には、まだ、Web 上で仮想3次元空間を扱うためのネットワーク機能は存在せず、また、Web 上で扱うには、負担の掛かる複雑な3次元形状の定義などが含まれていた。そこで更に、メーリングリスト上で、Open Inventor の機能拡張、および、縮小などの議論が行われた。このときに、従来の機能に加え、Web 上での利用を考慮して、以下のようなネットワーク機能が付加された。

- リンク機能
- URL による外部ファイルの指定機能

そして、1995年5月、最初の仕様として、VRML1.0 が正式に発表された²⁾。この仕様は、メーリングリストを利用したことにより、着想からわずか1年足らずというスピードで開発され、成功を収めることとなった。

2.2 VRML2.0 の誕生

VRML1.0 では、静的なモデルを記述するための機能しか持たず、3次元オブジェクトの動作記述やマルチメディア再生などの機能は規定されなかった。このため、多くの研究機関によって、このような機能に関する VRML1.0 の独自拡張が提案されることとなった。筆者の所属するソニーアーキテクチャ研究所が、Telecom95 で発表した SE-VRML (Sony Enhanced-VRML) もその一つである。

このような動きにより、次の VRML の仕様化の気運が高まり、VRML 標準化を推進する組織である VAG (VRML Architecture Group)³⁾ は、1996年1月に、インターネット上での VRML2.0 の仕様案の募集を行った。この募集では、米 Microsoft 社の「Active VRML」、米 Sun Microsystems 社の「HoloWeb」や米 Apple Computer 社の「Out of This Worlds」など

合計6つの仕様案が提出された。そして、投票の結果、1996年4月、ソニー、米 Silicon Graphics 社、米 World Maker 社が共同提案した「Moving Worlds」を基に VRML2.0 の仕様を策定することが決定した。これは、いくつかのドラフトを経過し、1996年8月、VRML2.0 の最終仕様が SIGGRAPH '96 の場で発表された⁴⁾。VRML2.0 は、現在、ISO (International Standards Organization, 国際標準化機構) に規格提案されている。

VRML1.0 から VRML2.0 に追加された主な機能は、以下の機能であった⁵⁾。

- 動き (Behavior) の記述機能
- スクリプト言語 (Java や JavaScript) との連携機能
- マルチメディア再生機能
- プロトタイピング機能
- 衝突検出機能

3. VRML の概要

本章では、最新のバージョンである VRML2.0 を基に VRML について説明する。

3.1 Web 上での位置付け

Web は、サーバーとブラウザという二つのソフトウェアによって構成される。ブラウザは、ユーザの操作に応じてサーバーに情報の問い合わせを行い、サーバーは、ブラウザからの問い合わせ要求を受け取り、問い合わせを解釈して要求される情報を返す。そして、要求した情報を得たブラウザは、その情報をユーザが理解できる形態にしてコンピュータディスプレイ上に表示する。例えば、その情報が HTML である場合、ブラウザは、その情報を清書し、2次元の文書 (ホームページ) としてユーザに示す。HTML を表示するブラウザは HTML ブラウザと呼ばれる。

VRML もまた、Web で扱える情報の一つである。

VRMLで書かれた情報をHTMLブラウザが受け取ると、それはVRMLブラウザに渡される。VRMLブラウザは、VRMLで書かれた情報（VRMLファイル）を受け取ると、そのファイルを解釈し、仮想3次元空間を画面に表示する。図1にVRMLファイルをWeb上で公開する仕組みを示す。

実際に、VRMLファイルをWeb上の情報として扱う最も簡単な方法は、それをHTMLからのリンクで公開する方法である。例えば、“redCone.wrl”というVRMLファイルをWeb上で公開するには、そのファイルにリンクを張った図2のHTMLファイルをWebサーバ上で公開することとなる。

3.2 ファイルフォーマット

VRMLファイルは、ノードとフィールドから構成される。ノードとは、仮想3次元空間のオブジェクトを記述するものである。それには、3次元の形状（円錐、球、立方体など）、色、テクスチャといったデータばかりでなく動画や音といったマルチメディア情報などを持たせることができる。フィールドとは、それぞれのノードが扱うオブジェクトが備えている属性に対して記述するために定義されたものである。例えば、球を表すSphereノードには、“radius”と名付けられた半径を表すフィールドがある。フィールドは、それ自身の名前、型およびそれが保持する値および値の型を持っている^(注2)。これらの一般的な書式は、

ノード名 {

 フィールド名1 フィールド値1

 フィールド名2 フィールド値2

.....

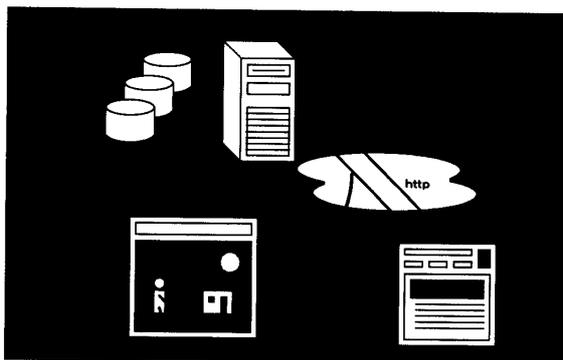


図1 VRMLファイルをWeb上で公開する仕組み

(注2) フィールドの値の型としてノードを持つことができる。VRMLでは、これにより階層構造（シーングラフ）を記述して、実際の仮想3次元空間を実現している。

- 1 <HTML>
- 2 これは赤い円錐です。
- 3 <HTML>

図2 VRMLファイル（redCone.wrl）を公開するHTMLファイル

```

1 #VRML V2.0 utf8
2 Shape {
3   appearance Appearance {
4     material Material {diffuseColor 1.0 0.0 0.0}
5   }
6   geometry Cone {height 3.0}
7 }

```

図3 赤い円錐を表示するVRMLファイル（redCone.wrl）

である^(注3)。ここで、フィールド名1と2は各フィールドの名称を、フィールド値1と2はそれぞれのフィールドに対応した値を意味している。

例として、高さ3mの赤い円錐を一つ表示するVRML2.0ファイルを図3に示す^(注3)。これをVRMLブラウザに読み込ませて表示させた結果が図4である。図4の場合は、redCone.wrlというファイル名とした^(注4)。

VRMLファイルでは、#の後の行末までの文字はコメントとして扱われる。図3の1行目は、VRMLファイルであることを示すヘッダーであり、すべてのVRMLファイルに存在する特別なコメントである。これは、VRMLのバージョンが2.0であることと、VRMLで使用される文字コードがUTF-8であることを示している。2行目以降が、実際に仮想3次元空間を記述した部分である。

2行目のShapeノードは、3次元オブジェクトの形状や見え方について記述するノードである。Shapeノードで形状について指定するフィールドがgeometryフィールドであり、ここでは、6行目で円錐の形状を示すノードであるConeノードを指定している。そこには、円錐の高さを指定するheightフィールドが記述されており、その値は3.0で、円錐の高さが3mで

(注3) 図3中の太文字はノードを示す。以下同様。また、VRMLでは、長さの単位で1.0は1.0mとして解釈される。

(注4) VRMLファイルの拡張子は、.wrlである。



図4 赤い円錐を表示したVRMLブラウザ

あることを示している。これ以外にも円錐の底面の半径を指定する `bottomRadius` というフィールドも存在するが、ここでは記述していない。VRMLでは、フィールドの記述を省略すると、仕様で決められたデフォルト値がその値として設定される^(注5)。また、VRMLでは、円錐以外にも様々な形状を示すノードが存在する。それらを表2に示す。

また、見え方について指定するフィールドが `appearance` フィールドである。そこでは、色やテクスチャといった見え方の種類について指定する `Appearance` ノード(3行目)、材質について指定する `Material` ノード(4行目)を通して、色を指定する `diffuseColor` フィールドを赤に設定することで、その円錐を赤色にしている。

表2 VRMLの形状ノード

ノード名	形状
Box	立方体
Cone	円錐
Cylinder	円柱
ElevationGrid	幾何地面
Extrusion	汎用円柱
IndexedFaceSet	ポリゴン
IndexedLineSet	線
PointSet	点
Sphere	球
Text	3次元文字

(注5) ここでは、`bottomRadius` フィールドのデフォルト値は、1.0であるので、円錐の半径は、1メートルである。

```

1 #VRML V2.0 utf8
2 Shape {
3   appearance Appearance {
4     texture ImageTexture {
5       url " http://www.shool.edu/vrml/someTexture.jpg"
6     }
7   }
8   geometry Cone {height 3.0}
9 }

```

図5 別のサイト上のテクスチャを円錐に張ったVRMLファイル

3.3 Web用に拡張した機能

前に述べたように、VRMLは、単にコンピュータディスプレイ上に仮想3次元空間を表示するだけの言語ではない。その特徴の一つに、Web上の情報資源を再利用する機能があげられる。その機能が、「リンク機能」と「URLによる外部ファイルの指定機能」である。前者は、HTMLで文字列に他のWebページなどへのリンクが張れるのと同様に、VRMLの仮想3次元空間内にある3次元オブジェクトにリンクが張れる機能である。後者は、一つの仮想3次元空間を構成する様々なオブジェクトを、別のWebサーバ上に置くことを可能にする機能である。

これらの機能により、ある屋敷をVRMLで作ろうとしたとき、リンク機能で、それぞれの部屋が違うWebサーバ上にあたり、また、URLによる外部ファイルの指定機能で、その部屋にある家具なども実は違うWebサーバ上にあるといった分散環境が構築できるのである。

図5は、表面にテクスチャを張った円錐を表示するVRMLファイルである。ここでは、5行目の `ImageTexture` ノードの `url` フィールドで指定したWeb上のイメージファイルをテクスチャとして利用している。

3.4 動作記述の枠組み

VRML2.0における最大の特徴は、動作を記述することができるということである。本節では、その枠組みについて述べる。

VRMLにおける動作とは、あるノードのフィールド値を変化させることによって、VRMLブラウザ上に表示されている仮想3次元空間のオブジェクトに何らかの変化を与えることである。これにより、色のフィールド値を変えることで表示される色を変えたり、

表3 VRMLのセンサーノード

ノード名	センサー機能
TouchSensor	マウスのクリックを検知し、それに対応してイベントを発生させる
TimeSensor	時間の経過を検知し、それに対応してイベントを発生させる
ProximitySensor	定義された領域に対する視点の挙動を検知し、それに対応してイベントを発生させる
VisibilitySensor	オブジェクトの可視性を検知し、それに対応してイベントを発生させる
CylinderSensor	マウスのドラッグを円柱のY軸回りの回転として検知し、それに対応してイベントを発生させる
PlaneSensor	マウスのドラッグをXY平面の平行移動として検知し、それに対応してイベントを発生させる
SphereSensor	マウスのドラッグを原点を中心とした回転として検知し、それに対応してイベントを発生させる

位置のフィールド値を変えることで物を動かしたりできる。また、その応用として、ある時刻がきたら音を鳴らしたり、映像を再生し始めたりすることも可能である。

この動作の記述を実現するために、VRML2.0は、イベント、センサーノード、ルーティング機能をサポートしている。イベントとは、あるノードのフィールド値の変化である。センサーノードとは、ユーザの挙動（マウスをクリックした、ある領域に入ったなど）や時間の経過などを感知し、VRMLシーン内にイベントを発生させるものである。これには、表3に示す7種類のセンサーが存在する。ルーティングとは、発生したイベントを別のノードに伝播させる機構である。

基本動作記述の例として、図3を改良して、円錐をクリックすると照明が点くという例を図6に示す⁹⁾。

ここで、Transformノードは、childrenフィールドで複数のノードを取り囲むことにより、それらを一つのオブジェクトとして扱うことを可能にするグループノードである^(注6)。10行目のDirectionalLightノードは、空間に照明を与えるノードであり、onフィールドの値がTRUEならば照明が点き、FALSEならば消える。DEFは、ノードに名前をつけるキーワードであり、DEFでつけた名前の後にドットをつけてフィールド名を指定するとそのノードのフィールドを参照することができる。例えば、11行目のDL.onは、10行目のDirectionalLightノードのonフィールドを参照する。

図6におけるセンサーノードは、4行目のTouchSensorノードである。これは、マウスのクリックにより、そのisActiveフィールド^(注7)の値をTRUEにして、イベントを発生させる。このセンサ

```

1 #VRML V2.0 utf8
2 Transform{
3   children[
4     DEF TS TouchSensor{}
5     Shape{
6       geometry Cone{height 3.0}
7     }
8   ]
9 }
10 DEF DL DirectionalLight{on FALSE}
11 ROUTE TS.isActive TO DL.on

```

図6 円錐をクリックするとライトが点く VRML ファイル

ーの対象となるオブジェクトは、そのノードの兄弟かそれらの子供である。この場合は、5行目の円錐の形状を記述するShapeノード（円錐オブジェクトを示すノード）がそうである。ルーティングの記述は、11行目であり、ここでは、TouchSensorノードのisActiveフィールドの値が変わると、それがDirectionalLightのonフィールドにその値が反映されることを記述している^(注8)。

これらにより、円錐をマウスでクリックするとTouchSensorノードのisActiveフィールドの値がTRUEになり、ルーティングによりイベントの伝播が行なわれ、DirectionalLightのonの値もTRUEになり、照明が点くことになる。

3.5 Scriptノードとスクリプト言語

前節で述べたセンサーノードとルーティング機構だけでは、単純な動作は記述できるが、円錐をクリックすることで、その円錐の色を変化させる動作を記述す

(注6) 本来は、新しい座標空間について定義するノードである。

(注7) 実際には、図6に記述されていないが、TouchSensorノードのデフォルトのフィールドである。

(注8) ルーティングで参照される2つのフィールドの型は、同じでなければならない。この場合は、TS.isActiveとDL.onであり、両方ともブール型である。

ることはできない。円錐をクリックする操作は、ブール型であり、その結果、変更される色という型と違うからである。ルーティングで参照される2つのフィールドの値の型は同じでなければならないという制約上、この動作は前節の範囲内では記述不可能なのである。

そこで、それを可能にするものとして、VRMLでは、Scriptノードをサポートしている。Scriptノードは、そこで定義されたフィールドにイベントが伝播されると、指定されたプログラムを起動し、そこで何らかの処理をする。さらに、そのノードで定義された他のフィールドからイベントを発生させ、動作に繋げるものである^(注9)。ここでは、その例として、図3を改良して、スクリプト言語にJavaを用い、「赤い円錐をクリックすると色が青に変わる」という例を図7、図8に示す。

このプログラムの仕組みを以下に示す。図7において、

1. 円錐をクリックすると、TSと名付けられたTouchSensorノード（4行目）がそれを検出し、TouchSensorのisActiveフィールドの値が変わり、イベントが発生する。
2. そのイベントは、図7の18行目のルーティングにより、SCと名付けられたScriptノード中で定義されたtouchフィールド（14行目）に伝播する。
3. Scriptノードは、touchフィールドにイベントを受け取るとurlフィールドで指定したプログラムを起動する。それは、図8のJavaプログラムである。
4. 起動後、このプログラムは計算を行い、図7のScriptノード中で定義されたcolorフィールド（15行目）の値を変更する。
5. その変更によるcolorフィールドのイベント発生により、20行目のルーティングで、MTと名付けられたMaterialノード（7行目）にイベントが伝播される。
6. この結果、そのノードのdiffuseColorフィールドの値が変更され、円錐の色が青に変化する。

これらの一連の流れを説明したものを図9に示す^(注10)。

(注9) VRML2.0の仕様では、VRMLブラウザは、プログラムのスクリプト言語として、JavaかJavaScriptの何れかをサポートすればよいことになっている。

(注10) 値が変化するフィールドを「eventOut」と呼び、イベントを受けて値が動的に変化するフィールドのことを「eventIn」と呼ぶ。

```

1 #VRML V2.0 utf8
2 Transform{
3   children [
4     DEF TS TouchSensor{
5       Shape{
6         appearance Appearance{
7           material DEF MT Material{diffuseColor 1 0 0}
8         }
9         geometry Cone{height 3}
10      }
11   ]
12 }
13 DEF SC Script{
14   eventIn SFBool touch
15   eventOut SFColor color
16   url "ChangeColor.class"
17 }
18 ROUTE TS.isActive TO SC.touch
19 ROUTE SC.color TO MT.diffuseColor

```

図7 赤い円錐をクリックすると青い円錐に変わるVRMLファイル^(注10)

```

1 import vml.*; import vml.field.*; import vml.node.*;
2 public class ChangeColor extends Script{
3   private SFColor color;
4   public void initialize(){
5     color = (SFColor) getEventOut("color");
6   }
7   public void processEvent(Event e){
8     if (e.getName().equals("touch")){
9       color.setValue(0.0f, 0.0f, 1.0f);
10    }
11  }
12 }

```

図8 図7用のJavaプログラム (ChangeColor.java)

本稿では紙面の都合上詳しくは述べられないが、ここで、注意すべきは、VRMLファイルとJavaプログラムとの連携である。つまり、Javaプログラム上でVRMLファイルに記述されたあるノードのあるフィールドについてその値を変化させる仕組みである。これは、それらの間にAPI (Application Programming Interface)⁷⁾を設けることにより、実現している。

この例では、色を変更しているが、位置情報やテキストなどの様々な属性を変化させることができる。この機構を用いると、更に、「決まった時間に株価データベースにアクセスを行い、得られた情報を3次元のグラフで表示する」という様な複雑なプログラミングを必要とする動作記述も可能である⁹⁾。

3.6 その他の機能

最後に、これまでに説明しなかった機能で重要と思われる、プロトタイプ機能とマルチメディア再生機能について述べる。

・プロトタイプ機能

VRMLでは、同じ形状を記述する場合、DEFとUSEというキーワードを用いて、一度記述したオブジェクトにDEFで名前をつけ、その名前をUSEを用いて利用することができる。複数個USEを用いることができ、これにより一種のコピー機能を実現している。しかし、それでは、全く同じオブジェクトを再利用することはできないが、色やテキストといった一部だけが違うといった類似オブジェクトの再利用はできない。これを可能にするのは、プロトタイプ機能である。これは、オブジェクトの雛形を作り、再利用する機能であり、オブジェクト指向言語のクラスとインスタンスの機能と同様のものである。これにより、オブジェクトの再利用がより簡単に行える。

・マルチメディア再生機能

VRMLでは、動画や音を再生する機能を持っている。動画は、MovieTextureノードによってサポートされ、これにより、動画を任意の3次元オブジェクトに貼って再生し、それを制御することができる。また、音は、AudioClipノードとSoundノードによってサポートされる。AudioClipノードは、どのような音

を再生するかを指定するノードであり、Soundノードは、音源や音場を指定することができる。これらの動画や音源も3次元オブジェクトと同様に扱うことができるので、動作記述を用いて、チャンネルスイッチを押すと流れる映像が変わるテレビの実現や、走っている自動車に音源を配置することが可能である。図10にテレビの例を示す。この例では、テレビのスイッチを入れると、動画が流れ、チャンネルを変えると、違う番組を見ることができる。また、音量ボタンを操作すると、そのボリュームをコントロールすることができる。

4. VRMLの現在と今後

VRMLは、メーリングリストなどを通じて、現在もその進化を続けている。本章では、現在、VRMLコミュニティで議論されている新機能、VRML3.0へと繋がるマルチユーザ機能の動向およびVRMLコンソーシアムについて述べる。

4.1 VRML2.0への機能追加

近い将来、VRML2.0の仕様に追加されるであろう機能として、バイナリフォーマットとブラウザへの外部APIがある。

・バイナリフォーマット

VRMLファイルは、3次元空間を記述するという性質上、座標などの数値データの記述が多く、複雑な空間を構築しようとするとその容量は膨大になる。これはネットワークの遅延に繋がる。そこで、それを効率的に圧縮する技術の登場が期待されている。現在、米IBM社、米ParaGraph International社、米Apple

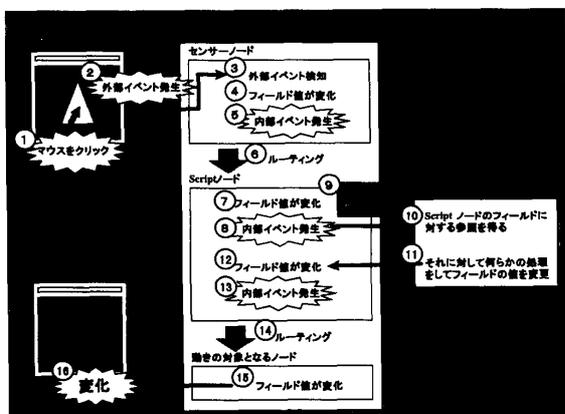


図9 スクリプト言語を用いた動作記述の流れ

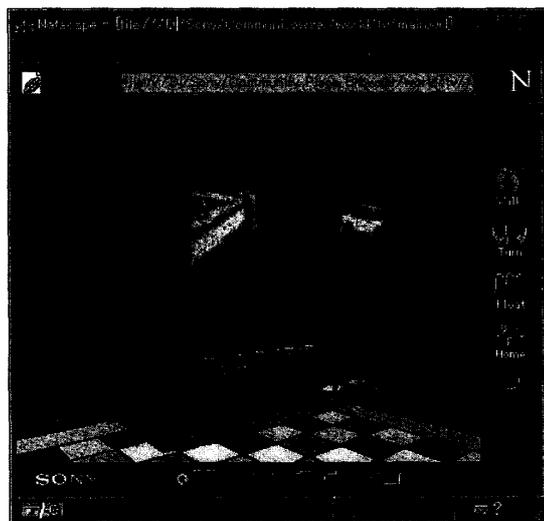


図10 VRMLで記述されたテレビ

Computer社が新しいフォーマット（バイナリフォーマット）⁹⁾を共同提案している。それは、VRML2.0に最適化して効率的に圧縮を行うファイル圧縮フォーマットであり、その仕様の枠組みは、米IBM社のジオメトリ圧縮技術と、米Apple Computer社のノードレベル圧縮技術である。米ParaGraph International社は、その仕様に基づいて実装を行っている。

• ブラウザへの外部 API

外部 API とは、VRML ブラウザを外部から操作可能にする API である。ここで外部とは、Java のアプレットや米 Microsoft 社の COM (Common Object Model) と ActiveX および OMG の CORBA などの分散オブジェクトアーキテクチャを言う。外部 API の仕様化に関しては、VAG が仕様案の募集を行ない、米 Silicon Graphics 社の「External Authoring Interface」¹⁰⁾と米 Dimension X 社の「The Dimension X External VRML API」¹¹⁾が提案された。両方とも Java の API について規定している。これらについて、投票を行った結果、1996年11月、米 Silicon Graphics 社の提案する「External Authoring Interface」が支持を集めた。

4.2 VRML3.0 に向けて：マルチユーザ機能

VRML におけるマルチユーザ機能とは、仮想 3 次元空間の中で複数人間がお互いにコミュニケーションを行うことを可能にする機能である。そのコミュニケーションは、仮想 3 次元空間上の 3 次元オブジェクトやユーザの分身として表示されるアバタおよびそれらの動作などをお互いに認識した（共有した）上で実現される。コミュニケーションの種類は、テキストチャット、音声チャット、アバタのアクション（動作による感情表現）、アバタや共有オブジェクトの移動および接触などである¹²⁾。

しかし、VRML2.0 では、マルチユーザ機能に関する仕様の取り決めは行われていない。そこで、現在、VRML 関連の研究機関や企業から、このマルチユーザ機能に関する様々なシステムや仕様が VRML2.0 の拡張として提案されている。これらのシステムや仕様は、まだまだ発展段階であり、これらの成果を踏まえて、VRML3.0 の仕様の決定が行われるであろう。

4.2.1 Community Place システム

例えば、筆者の所属するソニー(株)アーキテクチャ研究所では、マルチユーザ機能をサポートした「Community Place」システム（旧 Cyber Passage）¹³⁾を提供している。筆者らは、本システムを用いて、1996年2月よりインターネット上でマルチユーザ機能の公開実

験を行っている。図11に「Community Place」を用いて、VRML2.0 および Java で実現したマルチユーザ仮想 3 次元空間であるサーカスパーク 2 の例を示す。ここでは、同じシーンを違う角度から見たユーザの VRML ブラウザを 2 つ示している。そのシーンには、台の上に乗ったアシカがおり、それはボールで遊ぶ動作をしている。同じ仮想 3 次元空間上にいるユーザは、アバタの目を通してその動作を確認することができ、「アシカがボールで遊ぶ」シーンを共有している。実際のインターネットの環境では、それぞれのブラウザはインターネットに繋がった別のコンピュータで動いている¹⁴⁾。

また、ソニー以外にも、米 Black Sun Interactive 社では、マルチユーザ機能をサポートした「Cyber Gate」というシステムを開発し、商品化している。

4.2.2 Living Worlds

マルチユーザ機能の標準化に向けて、現在、ソニー、米 Black Sun Interactive 社、米 ParaGraph International 社が中心となり、「Living Worlds」という仕様を VRML コミュニティに対して提案している¹⁵⁾。これは、VRML ファイルレベルでのマルチユーザ機能の記述に関して標準化しようとする提案であり、以下の項目が掲げられている。

• VRML ノードレベルでの標準化

- アバタを含む共有オブジェクトの統一的な記述手段の提供
 - ネットワークシステムへの統一的なアクセス手段の提供
 - アバタを通じたコミュニケーションの拡張性の提供
- この規格の特徴は、ネットワークアーキテクチャを規定していないことである。つまり、サーバ・クライアント方式でも完全分散方式でも適用可能であり、パケットフォーマットやプロトコルについては規定しておらず、今後のネットワーク技術の発展を取り込みやすい形になっている。

4.2.3 Open Community

また、Mitsubishi Electric Research Labs、米 Chaco Communications 社および米 Worlds 社を中心として、「Open Community」という仕様が提案されている¹⁶⁾。これは、VRML2.0 と Living Worlds 規格に基づき、Java の API 群を規定している。

4.3 VRML コンソーシアム

VAG は、VRML2.0 の最終仕様を発表することにより、その役目を終えた。そこで、昨年末、新たに、大学やコンピュータ関連の企業 41 機関が中心となり、

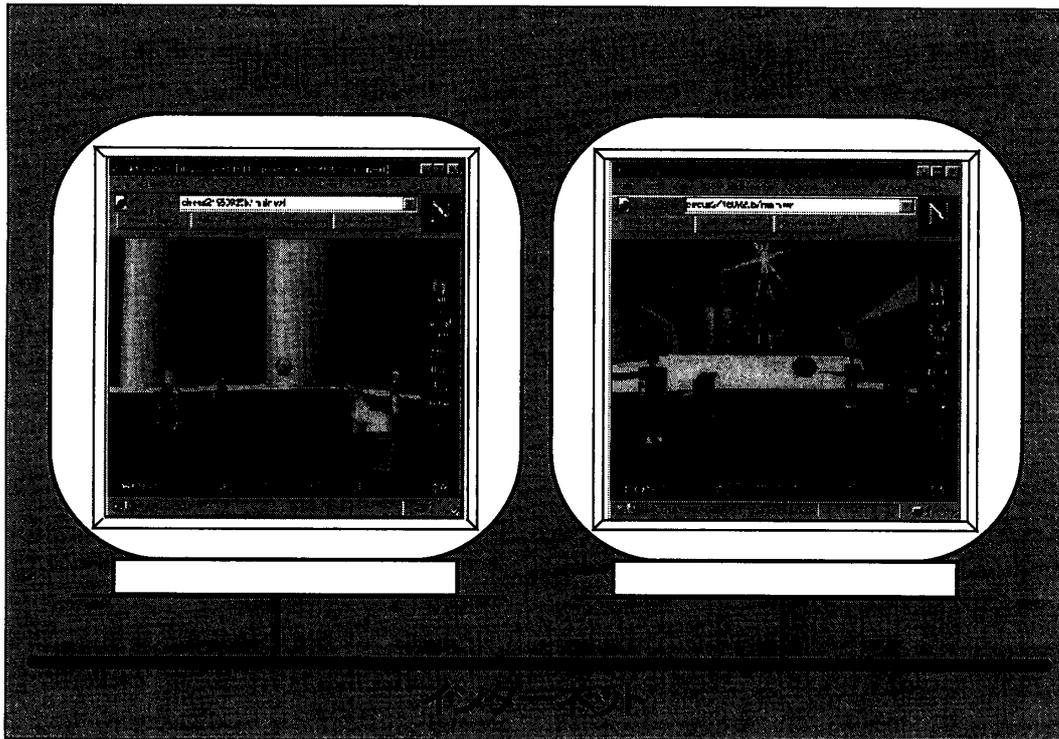


図11 Community Place のマルチユーザ機能

VRML の標準化と開発を行う「VRML コンソーシアム」¹⁷⁾が結成された。そこでは、今後の VRML 関連の技術について、標準化の必要性の話し合いを行う。その技術について標準化が必要であれば、インターネットにその候補の提案を呼びかけ、投票により標準化する技術を決定するといったことが行われる。ソニー、米 Microsoft 社、米 Netscape Communications 社、米 Silicon Graphics 社などがコンソーシアムのボードメンバーになっている。

5. おわりに

現在、VRML コミュニティ上では、コンピュータ言語技術、オブジェクト指向技術、ネットワーク技術、3DCG 技術、ユーザインターフェース技術、マルチユーザ技術、ひいては、人工生命、仮想社会現象の研究まで、様々な分野の議論がこれからの VRML の在り方に関連して行われている。これらの多くは、現在最も注目を集めている研究である。このような状況に対して、筆者は、「アポロ計画が様々な技術のインキュベータになったのと同様に、VRML がそれらの技術の進歩のインキュベータとなるのではないか」と考えている。つまり、VRML は、それらの研究の応用に対するインフラを与えることができるのである。今

後、VRML は、これらの技術との相互作用により、より良いもの、簡単かつ楽しいものへと発展していくものと思われる。VRML は、現在もその進化は続いており、多くの人が VRML に興味を抱くことを期待する。

参考文献

- 1) マーク・ベッジ著、松田晃一他訳：VRMLを知る、プレントリスホール（1996）
- 2) 「VRML1.0仕様書」：<http://vag.vrml.org/vrml10c.html>
- 3) VAGのホームページ：<http://vag.vrml.org>
- 4) 「VRML2.0仕様書」：<http://vag.vrml.org/VRML2.0/FINAL>
- 5) 松田晃一：VRML2.0の概要とCyberPassageが実現する共有仮想空間、Computer Today, 9, 20/30 (1996)
- 6) 本田康晃：仮想空間をより豊かに演出するVRML2.0、日経データプロ・マルチメディアコミュニケーションサービス, 12 (1996)
- 7) 「The Virtual Reality Modeling Language Appendix C. Java Scripting Referene」：<http://vag.vrml.org/VRML2.0/FINAL/spec/part1/java.html>
- 8) Rodger Lea, Kouichi Matsuda, Ken Miyashita: JAVA for 3D and VRML Worlds, New Riders (1996)
- 9) バイナリフォーマットのページ：<http://www.rs6000.ibm.com/vrml/binary>
- 10) External Authoring Interfaceのページ：<http://vrml.sgi.com/moving-worlds/spec/ExternalInterface.html>
- 11) The Dimension X External VRML APIのページ：<http://>

- /www.dimensionx.com/products/lr/docs/extapi.html
- 12) 松田晃一他：チュートリアル，VRMLと分散環境，日本ソフトウェア科学会 講習会 資料シリーズ，2 (1996)
 - 13) Community Place のホームページ：http://vs.sony.co.jp
 - 14) K. Matsuda, R. Lea, Y. Honda: Virtual Society: Multi-user interactive shared space on WWW, ICAT '96 (1996)
 - 15) 「Living Worlds 仕様書ドラフト#1」：http://www.livingworlds.com
 - 16) 「Open Community 提案書」：http://www.merl.com/opencom/opencom.htm
 - 17) VRML コンソーシアムのページ：http://vag.vrml.org//consort
 - 18) Y. Honda, K. Matsuda, J. Rekimoto, R. Lea: Virtual society, Procs. of VRML '95, 109/116, ACM press (1995)
 - 19) R. Lea, Y. Honda, K. Matsuda, S. Matsuda: Community Place: Architecture and Performance, Procs. of VRML '97, 41/49, ACM press (1997)
-