

# VRML と他言語を使用した PlayStation の ゲーム開発システムの提案

村吉 政登      河野 真治  
Masato Murayoshi      Shinji KONO

琉球大学大学院理工学研究科情報工学専攻  
Specialty of Information Engineering, University of the Ryukyus.

## 概要

VRML は、簡単な記述で 3 次元物体を表現することができる。本研究室で作成した PlayStation の同期型タプル通信システム、VRML パーサー、TMD 生成表示プログラムを応用し、簡単なオブジェクトデータのコードを VRML で記述できるようにした。この環境から、ゲームの処理を分散し、多様なプログラム言語で 3 次元のゲームオブジェクトを取り扱うことが可能な開発システムを提案し、インタラクティブオブジェクトの取り扱いを検討する。

## 1 はじめに

PlayStation では「ネットやろうぜ」というシステムにより、計算機と接続することにより、ユーザーが自分で PlayStation のアプリケーションの開発が行える環境が提供されている。これまではこの開発環境を中心に通信機能を実装し、3 次元用のマークアップ言語である VRML を表示する機能の実装を行ってきた。VRML は 2.0 から応答性やアニメーションといった動的な表現も可能になっており、この VRML コードを、ゲーム本体に送る物体情報として利用することで、ゲームプログラム内のオブジェクトを簡単に利用できるのではないかと考えた。

本稿ではこの環境を利用して、インターネット上からの 3 次元データ、または自分で作成した 3 次元データを PlayStation のオブジェクトとして利用でき、多様なプログラム言語でゲームオブジェクトを取り扱うことが可能な開発システムを提案し、新たな 3 次元物体・仮想空間の閲覧・体験と、ゲームオブジェクトをインタラクティブに利用できるゲームプログラミングの環境を検討する。

viewer for PlayStation の 3 つの処理系と、通信システムである PSX Linda、外部から入力可能な VRML コード、PlayStation の総合グラフィックサービスで扱うモデルデータのフォーマット (バイナリ形式) である TMD ファイルの 2 つのファイルから構成される。システムの構成を図 1 に示す。

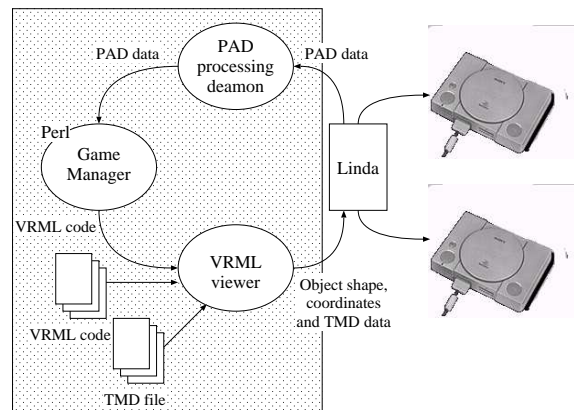


図 1: システム構成

## 2 システム構成

本開発システムの構成を図 1 に示す。本システムは PAD processing daemon, Game Manager, VRML

### 2.1 VRML2.0 文章の構成

VRML (Virtual Reality Modeling Language: 仮想現実実現設計言語) はインターネット上で 3 次元空間や 3 次元物体をリアルタイムに表示するための言

語である。VRMLには、プラットフォームに依存しない、拡張性がある、バンド幅が狭い接続でも快適に動作するといった特徴がある。VRMLの文には、物体(オブジェクト)の形状や座標を定義するノードと、ノードの細かな性質を定義するフィールドがあり、VRMLファイルの内容はノードの組合せによって記述される。図2にVRMLファイルの例を示す。

```
#VRML V2.0 utf8
```

```
Shape {
  geometry Box {size 2 3 5}
}
```

図 2: VRML ファイルの例

オブジェクトを区別して使用する時は、ノードのユーザ定義を使用して名前を定義する。本システムでは、ノード一部を拡張していて、TMDファイルが存在し、直接使用する場合に、新たに追加した拡張ノードであるTmdノードを使用することで、フィールド内にTMDファイル名を指定できる。

## 2.2 PSX LindaによるPlayStationのネットワークプログラム環境

ゲームプログラミングはリアルタイム処理で行われ、ネットワークプログラムは並列分散処理で行われる。この異なる2つのプログラムを組み合わせた、ネットワークゲームを実現するのが、本研究室で開発したPSX Lindaである(図3)。PSX LindaはPlayStationのネットワークゲームを同期型タプル通信によるリアルタイム処理で実現する。

PSX Lindaでは、描画計算後の残りCPU時間を通信に割り当て、ゲーム画面での遅延をなくす。PSX Lindaを実装したプログラムではLinda libraryの呼び出しを行い、呼び出したLinda libraryによってパケット形式に変換され、read/writeのブロッキングを行うルーチンを呼び出し、そのルーチンが物理的にread/writeを行う。read/writeはシリアルを通り、UNIX側にデータを転送する。UNIX側では、uxcomm(PlayStation-UnixCommunicator)がsocketを通してldserv(Linda Server)へ転送する。Linda Serverは各uxcommから送られてきたデータ

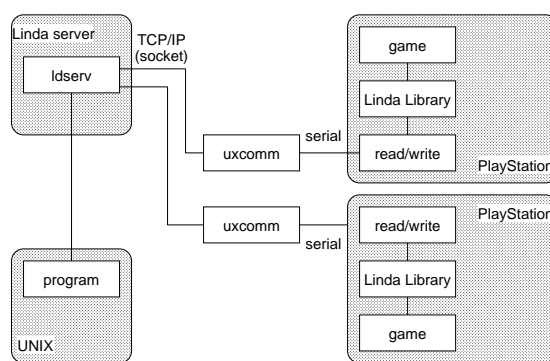


図 3: 同期型タプル通信 PSX Linda

をバッチ処理し、それぞれのuxcommに送り返す。ldservからデータを受け取った各uxcommはシリアルを通してPlayStation側に転送し、Linda libraryがデータを変換してプログラム上で反映させる。

### 2.2.1 Soccer ServerのPlayStation上での表示

PSX Lindaを用いたSoccer ServerのPlayStation上での表示について説明する。

Soccer Serverは、様々なタイプのプログラミング言語で書かれたプログラムどうしが、サッカーの試合を行うことが可能なシステムである。このシステムでコーチクライアントと呼ばれるクライアントは使用者が直接制御できる。このクライアントの操作と、サッカーフィールドとフィールド上の全ての物体を3次元でPlayStation上に表現する機能を本研究室で実装した。

サッカーフィールドとフィールド上の全ての物体のTMDファイルは、最初でPlayStationに読み込まれ、PSX LindaでUNIX上のSoccer Serverと接続される。PlayStationからのパッド入力はSoccer Serverに対応するキーボードデータに変換され、Lindaを通してコーチクライアントに送られる。Soccer Serverからは各クライアントとボールの座標がLindaを通してPlayStationに送られ、各TMDファイルとリンクされ、表示される。画面を図4に示す。

## 2.3 Game Manager

Game Managerの構造の例を図5に示す。

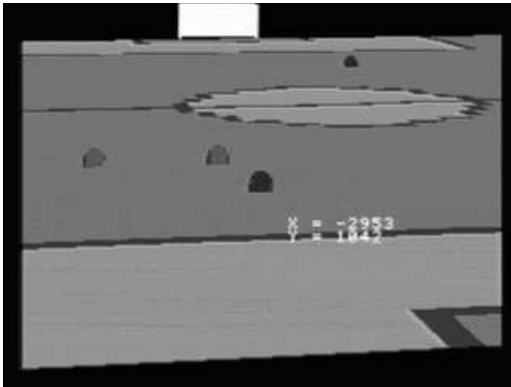


図 4: Soccer Server の PlayStation 上での表示

```
while(<>){
  PAD processing deamonからの入力データ受け取り
  for each ゲームオブジェクト{
    ゲーム上のオブジェクトの状態計算
  }
  現状態の VRML コードの出力
}
```

図 5: Game Manager のプログラム構造

プログラムの構造は、PAD processing deamonからの入力データ物体の座標や当たり判定などの状態計算を行い、ゲーム上でのオブジェクトの状態を把握する。そして、現状態を表す VRML コードを生成し、標準入出力で VRML viewer for PlayStationへ入力する。プログラマは基本的に Game Managerを中心にプログラミングする。GameManagerの構造に従って記述を行うことで、様々な言語で記述できる。

## 2.4 VRML viewer for PlayStation

VRML viewer for PlayStation は、UNIX 側の VRML のパーサ処理と PlayStation 側のディスプレイ処理に分かれている。

Game Manager から標準入出力で送られた VRML コードの構文解析を UNIX 側の VRML パーサで行い、オブジェクトの名前、形状、座標を判別し、データをまとめる。そして、データは指定されたパケットへ格納され、Linda を通して PlayStation に転送する。PlayStation 環境では総合グラフィックサービスでモデルフォーマットとして TMD が用意されている

ので、VRML コードを直接利用するより一度変換した方が最適に使用できる。ただし、TMD ファイルが存在している場合は、VRML コードで TMD ファイル名を指定して、TMD ファイルを直接 PlayStation に分割して転送する。PlayStation 側のプログラムは VRML viewer から送られた物体の情報を元に TMD データを生成し、オブジェクトの作成・描画処理を行う。

## 3 システムの評価

このシステムの利点は、メインメモリなどの環境は考慮する必要があるが、3次元オブジェクトを VRML コードで扱うことで、PlayStation の描画の詳細を知る必要がなくなり、ゲーム本体のプログラムを抽象化することができるようになることである。また、他の PlayStation との通信も比較的容易に実現できる。

ゲーム途中でも、外部からの VRML コードや TMD データを PlayStation に転送することができるが、VRML コードは形状ノードによって TMD データへの変換の時間が増大し、それによりゲーム画面の遅延または停止が発生する。この時間は生成する TMD ファイルのポリゴン数や座標計算の回数と、PlayStation 側の TMD 生成プログラムの処理性能によって決定される。同様に、TMD データの転送処理時間は TMD データサイズに比例して大きくなり、ゲームの進行に影響を与える。表 1 に、TMD データ転送にかかる時間を示す。この時間は PSX Linda と転送プログラムの処理性能に影響されている。これらの時間は、Game Manager のプログラム時に、ロード時間として考慮しなくてはならない。

表 1: TMD ファイル転送にかかる時間

ファイルサイズ (KByte)	5.4	24.2	63.3
時間 (sec)	1.35	6.34	16.24

## 4 システムの拡張性と課題

ここでは今後のシステムの拡張性と、いくつかの問題点について考察する。

現システムは Game Manager で VRML コードを書き出すことで、ゲームを管理する形になっている。

Game Managerはネットワーク上で分散して実行し、物体の情報を送ることが可能ではあるが、各 Game Managerが同期をとらないと、オブジェクトの表示のタイミングが考慮されない。

並列オブジェクト指向的に、VRMLコードとそのイベント処理を行うスクリプトを一組みとして各オブジェクトを記述する方法を考えてみた。この方法では、オブジェクトの状態変更や新規追加が具体的に把握でき、スクリプトでオブジェクトどうしの通信を行うことで、タイミングの制約が可能になり、オブジェクトの依存関係の記述が容易になるものと思われる。

#### 4.1 アクティブオブジェクトの新しい記述法

オブジェクトは PlayStation 側に対応するために TMD フォーマットに変換するのに必要なコードのみを書き出して転送している。この方法ではアクティブオブジェクトを表すには不十分である。

VRMLでは ROUTE 文と呼ばれるノード間でフィールドに値を伝播する機能があり、これとセンサーノードによってノード間のイベント発信と受信を行うことで物体の応答性やアニメーションを実現している。

これらの機能を PlayStation 環境に適応させるには、VRMLをそのまま使用するよりも、DXFやRSDなどの他のフォーマットを直接使用してポリゴン定義を行う方が、内部の変換処理としては速くなるが、現システムでは実装されていないので、VRMLのイベント処理を PlayStation に最適化した新しい記述法を考えている。

#### 4.2 通信機能の定義

PSX Lindaは PlayStation・UNIX間の通信の簡単な使用を重点に考えたプロトコルであり、オブジェクト間での同期を計るのに適した通信プロトコルであるわけではない。PlayStation環境にそれほど影響を受けない UNIX・ネットワーク上では別で用意したほうがよい。衝突の検出や、メッセージパッシングをうまく記述できる方法の設計と実装を検討中である。

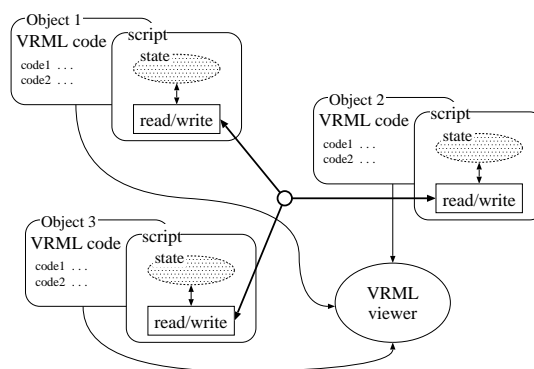


図 6: オブジェクトどうしでの通信プロトコルを用いたシステム

### 謝辞

本稿作成にあたって、大変お忙しい中、本システムの実装に多大な御協力と、貴重な助言を頂きました。新垣将史、神里健司、謝花蔵ら本研究室の院生の皆さんに深く感謝致します。

### 参考文献

- [1] 仲宗根雅臣、河野真治、“同期型タプル通信を用いたマルチユーザ PlayStation ゲームシステム” May 1998 IPSJ OS, CPSY, Okinawa Workshop (in Japanese)
- [2] “PlayStation”、  
<http://www.cong.ac.jp/maeda/playst/playst.htm>
- [3] “Soccer Server System”、  
<http://ci.etl.go.jp/noda/soccer/server/index.html>
- [4] “www.vrml.org Home of the VRML Consortium”、  
<http://www.vrml.org/>
- [5] “VRML2.0 TUTORIAL”、  
<http://www.nikonet.or.jp/spring/sanae/VRML2/Vrml2.htm>
- [6] John R. Levine/Tony Mason/Doug Brown 著、村上 列訳 “lex&yacc プログラミング” (ASCII、1994 年)