

VRML 対応多人数参加型システムアーキテクチャの提案

2 A A - 1

中村暢達 平池龍一

NEC ヒューマンメディア研究所

E-mail: {nakam,hiraike}@HML.CL.nec.co.jp

1 はじめに

近年、家庭向ゲーム機やPC用のビデオカードでは三次元グラフィックスが当然の機能となった。最近では、ほとんどのゲームソフトが三次元グラフィックスを使っているといっても過言ではない。一方で、インターネット上では三次元仮想空間を構築するための記述言語VRML [1] が標準となったが、アミューズメント分野へ応用するためには、より高度なリアルタイム性、多くのユーザの参加といった機能の強化を進める必要がある。

現在、いくつかのVRML対応のマルチユーザ参加型のアプリケーション [2] が発表されているが、それらは独自にVRMLの仕様の拡張を行い、独自のサーバプログラムを用意し、独自の通信プロトコルを採用している。そのため、セキュリティ、ログ管理といった機能を実現するためには、通常のWWWサーバとは別に組み込む必要があり、またプロキシサーバを利用するためには、別途設定が必要であるなどの問題がある。

また、VRMLの仕様策定WGでは、外部プログラム (Java アプレット) とのインタフェースが提案されている [3]。このインタフェースを利用することで、動くオブジェクトの制御やマルチユーザ参加の機能を実現することは可能である。しかしながら、これらの処理の負荷は重く、現状の計算機環境では応答性に問題がある。

本稿では、これら問題点を考慮し、より高度な3Dインターネット情報サービスを実用可能とするマルチユーザ参加3Dシステムのアーキテクチャについて述べる。

2 システムアーキテクチャ

本稿で提案するマルチユーザ参加3Dシステムのアーキテクチャを図1に示す。本方式の特徴を以下に説明する。

HTTP 互換プロトコル クライアント/サーバ間の通信にはHTTP (RFC 1945) を用いる。つまり、クライアントからサーバへの要求は、

```
<メソッド> <URI> <バージョン>
<ヘッダ>

<エンティティ ボディ>
```

A proposal of VRML multi-user system architecture
Nobutatsu NAKAMURA and Ryuichi HIRAIKE
Human Media Research Labs., NEC

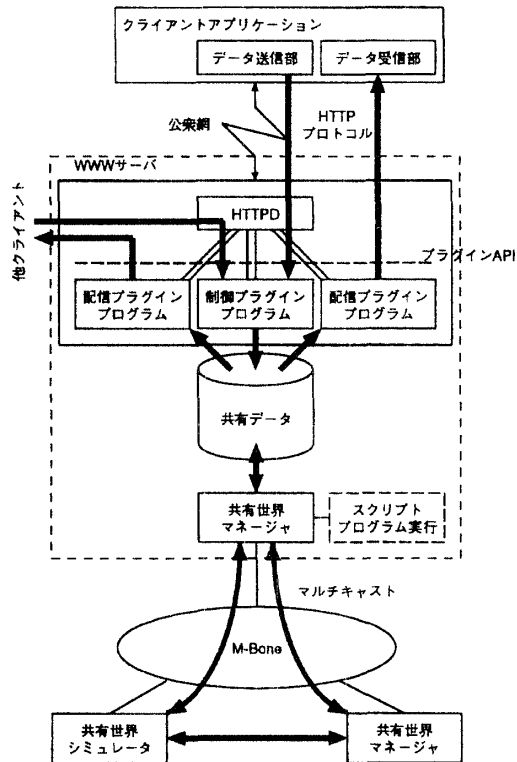


図1: マルチユーザ3Dシステムアーキテクチャ

のフォーマットで行う。WWWサーバ側では、HTTDPサーバが通常のHTTP処理と同じく、ログやセキュリティ管理などの処理を行う。その後、要求に対するサービスを実行する時点で、マルチユーザ機能のためのプラグインプログラム [4] が起動される。

URIにはユーザが共有したい仮想世界を記述したVRMLデータファイルが指定される。まず、最初の要求に対しては、配信プラグインと呼ばれるプログラムが応答し、要求されたファイルをクライアントに転送する。さらに、ファイルの転送後もコネクションを継続したままで、他のユーザの動作などの更新情報 (差分データ) が更新される都度、配信プラグインはそのコネクションを使って更新情報をクライアントに送信し続ける。

2回目以降の要求は、既に実行されている配信プラグインを制御するための情報や他のクライアントへ通信する自サイトにおける更新情報を含む。最初の要求と同じように、サーバへの送信はHTTPを使い、HTTDPサーバがログやセキュリ

ティ管理などの処理を行う。その後、今度は制御プラグインが実行され、クライアントからのデータに対応して、共有データを更新する。既に実行中の配信プラグインは共有データを監視しており、更新情報や制御情報を必要に応じて読み出す。

サーバ側での動きの管理 VRMLバージョン 2.0 対応のブラウザでは、Behavior(動き)の機能をクライアント側でスクリプトプログラムを実行することで実現している。しかし、クライアントの端末の種類はさまざまに処理能力も異なっているため、分散したクライアント間で常に仮想世界を同一に保つことは難しい。スクリプトプログラムの実行のために必要な計算機資源も多大であり、Behaviorの実現のため、ビューワの操作性は著しく損なわれている。一方、サーバは一般的に十分な処理能力が期待でき、またサーバで管理する共有世界を基準にすることで、クライアント間での共有世界の整合性を取りやすい。このような観点から、マルチユーザの場合、サーバ側でスクリプトプログラムを解釈する方がメリットが大きい。

サーバにおいて Behavior のためのスクリプトは解釈され、クライアントでは Behavior は無視するようにした。クライアントでイベントが発生すると、サーバがネットワークを介してイベント情報を受け取り、イベントを処理し、更新情報を生成して共有データを更新する。前述した配信プラグインによって、更新情報はクライアントに送信される。クライアントは、更新情報を使って、仮想世界データを更新し、表示するだけである。

サーバ間通信 1つのWWWサーバでシステムを構築し、多くのユーザの参加を可能とすると、応答性が損なわれる。HTTPはTCPを用いるため、どうしても応答性に劣ることもあるが、1つのサーバの処理能力では数百のクライアントと同時にある程度のデータ量を繰り返し通信することは不可能である。そのため、1つのWWWサーバに接続できるクライアント数を制限し、複数のWWWサーバを使うことにする。サーバ間では、マルチキャスト通信 [5] を用い、各サーバの共有世界マネージャ間で更新情報をまとめて通信する。共有世界マネージャは仮想共有世界の整合性も管理する。

また、接続するサーバとして、高速なリアルタイムシミュレーション計算機を使ったり、サーバでエージェントプログラムを実行することで、より高度なマルチユーザシステムが構築可能である。

3 更新情報

VRMLには、インスタンスと呼ばれる仕様がある。インスタンスには、DEFとUSEがあり、`DEF my_cone Cone { }` というように使うことで、ファイル中の形状、材質、光源、座標変換などの各要素(ノード)に名前を付けることができる。そ

して、`USE my_cone` というように使うことで、名前付けされた形状、材質、光源、座標変換などを再利用することができる。

VRMLの仕様では、ファイルを越えたインスタンスは無効であるが、本提案では、更新情報を扱うために、次のようにインスタンスを利用している。

更新 変更したいフィールドのみを通信する。もし、`my_cone` の高さのみを変更したい場合は、次のようなデータを転送する。

```
DEF my_cone Cone { height 0.5 }
記述されなかったフィールドの値は保存される。つまり、ここでは半径のフィールドである radius の値は変化しない。
```

消去 もし、`my_cone` を消去したければ、次のように空の Group ノードのデータを転送する。

```
DEF my_cone Group { }
```

追加 もし、`my_world` と定義された Group ノードに、`my_cone` を追加したければ、次のようなデータを転送する。

```
DEF my_world Group {
  children DEF my_cone Cone { } }
```

4 おわりに

以下の特徴を持つマルチユーザ3Dシステムアーキテクチャを提案した。

- HTTPを使うので、既存のWWWインフラの設定および機能をそのまま利用できる。
- Behaviorのスクリプトをクライアント側へ通信および実行しないため、クライアントへの負荷が小さく応答性に優れる。
- サーバ間の通信を用いて、より多くのユーザの参加などの高度な3Dインターネット情報サービスが可能である。
- 更新情報として、VRMLのインスタンスを利用しているので、既存のビューワからわずかな変更で実現できる。

今後、試作システムで性能評価実験を行う予定である。

参考文献

- [1] <http://vrm1.sgi.com/> など
- [2] 例えば、<http://vs.sony.co.jp/>
- [3] <http://vrm1.sgi.com/moving-worlds/spec/ExternalInterface.html>
- [4] http://home.netscape.com/newsref/std/server_api.html
- [5] <http://www.mbone.com/>