

## 連結された共同仮想空間における、オブジェクト管理方式

2 A A - 3

久松正和、森賀邦広、山本太郎

masa,moriga,taro@isl.ntt.co.jp

NTT 情報通信研究所

### 1 はじめに

近年のPCの急激なパフォーマンスの向上により、ユーザがクライアントとして用いてきたマシンをサーバーとしても利用できるようになってきている。我々は、このようなホモジニアスな環境を想定して、エンドユーザーがホームページのように手軽に立ちあげられる仮想空間を作成し、他ユーザーの仮想空間と相互に行き来できる共同仮想空間の実現を、目指している。

エンドユーザーが独自に作成する仮想空間は、膨大なコンテンツを擁するわけではないため、他の仮想空間と連結可能になることは大きな意味を持つ。現状実現されている仮想空間でも、URLでVRMLファイルを指定し、次々に空間を提示できる<sup>1)</sup>。しかし、アバターやオブジェクトを持って他空間との間を行き来する事はできない。また、大規模サーバーによるマッピングによって、仮想空間の連結をサポートするものもあるが<sup>2)</sup>、独自に作成されたものを任意に連結するには至らない。そのため、ユーザーが空間をまたがって協調作業する環境には不十分であると考えられる。

### 2 従来の仮想空間

仮想空間内のオブジェクトに関する情報（オブジェクトのID、形状や位置など）は、サーバーだけでなく、クライアント側でも保持している。クライアントは、これらの情報をもとに仮想空間を描画し、また、ユーザからの入力を受ける。（以下、図1）

①ユーザーによる操作が入力されると、その入

力は、ネットワークを通じて、②サーバーに問い合わせられ、他のユーザーとの競合、操作権利の有無などの有効性から判断され、③上記情報の更新が各クライアントに配信される。④各クライアントは配信された情報に基づき、描画内容を更新する。

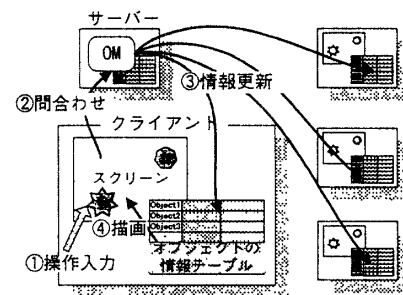


図 1:従来の仮想空間の構成

### 3 課題

連結された共同仮想空間の実現にあたり、本報告では以下の点に着目した。

- 1) オブジェクトの操作に対するレスポンス向上。
- 2) 複数サーバーで仮想空間を構成する事のオーバーヘッドの低減。

### 4 仮想空間の構成

#### 4.1 オブジェクトの分散管理

本報告で提案する共同仮想空間システムでは、クライアント側にもサーバーと同様のオブジェクト管理機構を設け（以下、オブジェクトマネージャー）、オブジェクトごとに管理をクライアントに分業させている。（図2）

この、オブジェクトの管理を「所有権」と呼び、全てのサーバー、クライアントが持つオブジェクトの情報管理テーブル（図3）には、操作時の問い合わせ先となる所有者ID（クライアントアドレス）を記す。これをサーバーが動的に更新する事で、所有権

Distributed Object Management for Connected Virtual Environments.

Masakazu Hisamatsu, Kunihiro Moriga, Taro Yamamoto  
NTT Information and Communication Labs.  
1-1, Hikarinooka, Yokosuka, Kanagawa, Japan

のやり取りが可能になる。

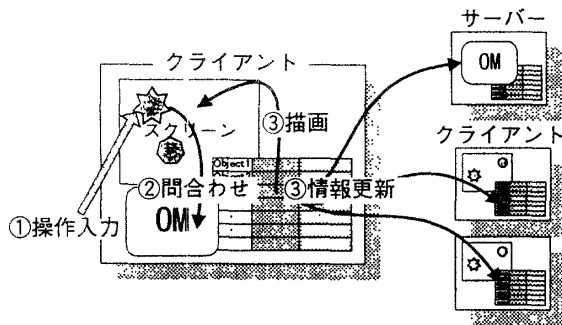


図 2: オブジェクトの分散管理

管理情報	Dynamic Data			Static Data		
所有者ID	Object ID	位置	状態	その他	形状	Script
Client A	2907843	0.1.2.0.4	None	None	File A	File B
Server	1992003	0.0.3.4	Open	None	File C	File D
新規追加				None	File E	None
				None	File F	

← 従来のテーブル

図 3: クライアントのオブジェクト情報テーブル

クライアントは、描画する空間にあるオブジェクトに関する情報、およびその空間に接続しているクライアントのアドレスを取得する。クライアントは、所有権を獲得したオブジェクトに対しては、サーバーに問い合わせる事なく主体的に操作を行える。

すなわち、図2のように、操作入力①に対し、自クライアント内のオブジェクトマネージャーに問い合わせ②をし、オブジェクトマネージャーの判断で、自シーンの描画と他クライアントおよび、サーバーに情報更新③を指示する。

一方、接続している他のクライアントは、そのオブジェクトの操作をする際、所有者となっているクライアントに問い合わせを行う。問い合わせを受けた所有者クライアントは、従来サーバーが行っていたのと同様に、判断、情報更新を行う。

サーバーの大きな役割としては、初期値でのオブジェクトの管理、接続されているクライアントの管理が残されている。

#### 4.2 単体の仮想空間内でのレスポンスの向上

仮想空間内のクライアントがオブジェクトを操作する場合、従来はサーバーとの通信のためのタイムラグは免れなかった。しかし、本システムでは、クライアントは、図2に示したように、操作対象の

オブジェクトについての所有権をサーバーから取得することで、サーバーへの問い合わせプロセスを経ることなく、描画できる。このため、操作をしたユーザーには早く、正確に描画できる。また、他人の操作を見ている他のクライアントにはある程度間引いて情報を送る等、システムの負荷を調節する事ができる。

#### 4.3 空間をまたがったオブジェクトの交換

オブジェクトの移動を操作するのはクライアントである事に注目し、オブジェクトの所有権を移動操作するクライアント渡して、移動処理を行う。即ち、移動処理を行うクライアントは移動元のサーバーから所有権を獲得した後、移動元空間からオブジェクトを削除する。クライアントは、移動先の空間上にオブジェクトを配置するとともに、その空間の管理サーバーに追加を指示する。このように、サーバー同士のインタラクションがない空間の間をサーバーに負荷をかける事なく、オブジェクトを移動させる事ができる。

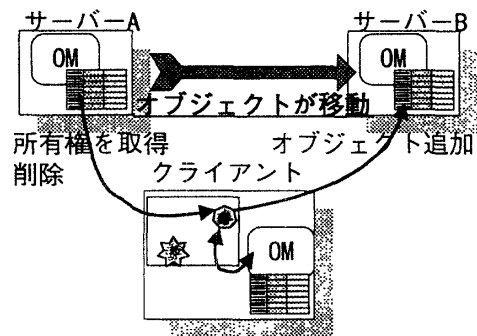


図 4: 仮想空間をまたがったオブジェクトの移動

### 5 まとめ

本稿では、複数サーバーで構成される仮想空間について、空間内に配置されるオブジェクトの管理を、そのオブジェクトを操作するクライアントに動的に設定する、分散オブジェクト管理方式を提案した。

<sup>1</sup> [http:// www.vrml.org/](http://www.vrml.org/)

<sup>2</sup> John W. Barrus et al. "Locales: Supporting Large Multiuser Virtual Environments", IEEE Computer Graphics & Applications. P50-57, Nov.1996