

## Spline アプリケーションによる 他プロセス所有オブジェクトの変更方式

佐藤 浩司 小塚 宏 高橋 克英 福岡 久雄 下間 芳樹

三菱電機(株) 情報技術総合研究所

### 1 はじめに

分散仮想環境構築基盤ソフトウェア Spline[1] は、アプリケーション構築用 API を提供し、仮想世界情報の管理とユーザ間インタラクションに必要な通信を支援する。

本論文では、Spline 上に構築したアプリケーション [2][3] において、他プロセスが所有する仮想世界内オブジェクトの変更を実現した二つの方式について議論する。

### 2 仮想空間内オブジェクトの共有

Spline では、共有される仮想空間情報は拡張が可能な様々なクラスのオブジェクトとして、各ユーザサイトが持つ仮想空間データの複製(世界モデル)内にコピーされ、共有される。複数アプリケーション間で仮想空間の情報を共有するための通信は、Spline によって自動的に行われる。

世界モデルを中心とした Spline の内部構造を図 1 に示す。

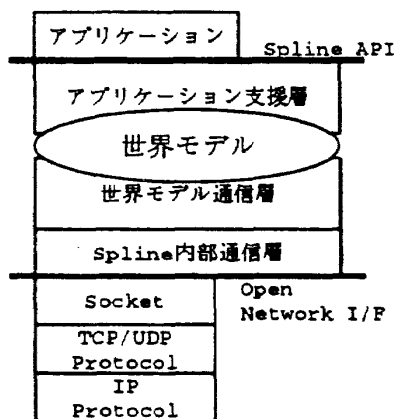


図 1: Spline の内部構造

Spline を利用するアプリケーションは、Spline が提供する API を用いて世界モデル内にオブジェクトを生成することができる。世界モデル内に生成されたオブジェクトは、そのクラス定義により共有されるオブジェクト

と共有されないオブジェクトに区別される。また、オブジェクトが内部に持つ様々な変数に対してもクラス定義により共有/非共有が指定される。

Spline は、仮想空間全体のうち自プロセスが注目している一部分に関して、その部分領域内に存在している「共有されるオブジェクト」の「共有される変数の値」のみを、同じ部分領域に注目している他プロセスと共有するための通信を行う。

### 3 他プロセス所有オブジェクトの変更

Spline 上のアプリケーションは、Spline の提供する API を用いて世界モデル内のオブジェクトにアクセスすることが可能である。世界モデル内の各オブジェクトはそのオブジェクトを所有するプロセスを識別するための値を変数として保持しており、通常その値はオブジェクトを生成したプロセスに割り当てられた値となっている。Spline API を用いるアプリケーションは、あるオブジェクトの持つ変数の値を参照することは自由にできるが、その一方、オブジェクトの持つ変数の値を変更することができるのは、そのオブジェクトを所有するプロセスのみとなっている。

このため、Spline は、他プロセスが仮想空間内に生成したオブジェクトを変更する場合に備えて、オブジェクトの所有権をプロセス間で譲渡するための仕組みを提供している。

次に、我々が構築したアプリケーション BMP[2] において用いた、所有権の委譲を利用した他プロセス所有オブジェクトの変更方式に関して述べる。

#### 3.1 所有権の委譲

Spline の提供する API を用いて世界モデル内のオブジェクトの所有権 (Ownership) を他プロセスに委譲する際には、所有権を受け取りたいプロセスから、所有権を持つプロセスに対してリクエストが出される。このリクエストも世界モデル内に生成されるオブジェクトであり、注目されているオブジェクトと同様に Spline によって両プロセス間で自動的に共有される。

自プロセスが所有するオブジェクトに対する、所有権委譲のリクエストを検知したプロセスは、アプリケーションの方針に従って所有権を委譲するかしないかを定める。

所有権を委譲されたプロセスは、オブジェクトに変更を加えた後、必要がなければ所有権を返却する。

この方式を、仮想空間における、より具体的な例で示すと以下ようになる。

ユーザのアバタ(化身)を仮想空間内に生成し、ユーザからの入力に応じてアバタの位置や動きを変更するプロセスAと、建物や部屋、テーブル、椅子といった様々なオブジェクトを仮想空間内に生成し、これらのオブジェクトの管理を行うプロセスBがある。今、アバタが部屋の中に入って、椅子を移動しその椅子に座ろうとしている。この時プロセスAはプロセスBに対して椅子オブジェクトの所有権の委譲を求めるリクエストを生成し、プロセスBはこれを許可し所有権をプロセスAに譲る。プロセスAは椅子の位置を変更し、プロセスBからの所有権委譲のリクエストに答えて所有権を戻した後、アバタの姿勢と位置を変えて椅子に座る動作を行う。

このように、所有権の委譲を用いる方式では、あるオブジェクトに対する変更の具体的な手順を、そのオブジェクトを変更したいプロセス(=そのオブジェクトを生成したプロセス以外のプロセス)が知っている必要がある。

次に、アプリケーションVTS[3]において用いた、所有権の委譲を使わないで上記椅子の例を実現する方式を述べる。

### 3.2 イベント通知による方式

所有権の委譲を行わずに、他プロセスが仮想空間内に生成したオブジェクトを変更する方式として、変更を加えたいオブジェクトを所有するプロセスに対して、その変更を表すイベントを通知し、元々そのオブジェクトを所有しているプロセスに、必要な変更を行わせる方式を考える。

各アプリケーションが世界モデル内に持つことができるオブジェクトのクラス定義は、アプリケーション開発者によって拡張が可能であり、必要なクラスを定義することができるようになっている。このような拡張クラスとして、他プロセスにイベントを通知するためのクラスを定義し、それぞれのイベントに対応して実行される処理を各オブジェクトを生成したプロセスが持つことで他プロセスが所有するオブジェクトを変更することが可能となる。

前節の椅子の例を本方式にあてはめると次のような動作となる。

プロセスAは椅子に近付いた後、椅子に座るというイベントを表すオブジェクトを世界モデル内に生成する。椅子オブジェクトを所有するプロセスBはこのオブジェクトを検知し、このイベントに対応する処理として椅子オブジェクトの位置を変更する。プロセスAはこの後、アバタの姿勢と位置を変更し椅子に座る動作を行う。

この方式を用いると、椅子オブジェクトを生成したプロセスのみが椅子オブジェクトに対する処理を知っていればよい。

### 3.3 2つの方式の比較

仮想空間が複数のプロセスにより構築される場合、椅子の例のように、あるオブジェクトに関する処理はそのオブジェクトを生成したプロセスが行うことが望ましい。この点からはオブジェクトの所有権を委譲する方式よりもイベントを通知する方式の方がすぐれているといえる。

しかし仮想空間内で、あるオブジェクトに対して近付いた時にアバタがとるべき行動(アバタオブジェクトに対する処理)が、非常に多岐に渡る場合、アバタを操作するプロセスは非常に負荷が高くなることが考えられる。また、アバタを操作するプログラムの実装時に、アバタがとるべき行動(例えば椅子に座るという行動)のすべてが明確ではない場合は、椅子の例とは逆に椅子オブジェクトを生成したプロセスがアバタオブジェクトの所有権を受け取り、操作する必要がある。

一般的に、仮想空間内においてユーザの化身であるアバタがとる行動は多種多様である。アバタプロセスの軽量化や、さまざまなイベントに対する柔軟な対応を考慮すると、アバタの基本的な動作はアバタプロセスに持たせ、特別な動作に関しては、その対象となるオブジェクトを生成したプロセスに持たせることが望ましいといえる。

## 4 おわりに

我々は、分散仮想環境構築基盤ソフトウェアSpline上に構築したアプリケーションにおいて、他プロセスが生成したオブジェクトを変更する方式として、所有権の委譲を利用する方式とイベントの通知を利用する方式の2つを用いた。これら2つの方式にはそれぞれ第3節で述べたような長所短所があり、分散仮想環境アプリケーションでは2つの方式をうまく併用する必要があるという結論に達した。

今後の課題として、これらの2つの方式を融合することを考えている。

## 参考文献

- [1] R. C. Waters 他: 分散仮想環境基盤ソフトウェア Spline, 三菱電機技報, Vol. 71, No. 2, pp. 26 - 29 (1997).
- [2] 高橋他: Spline上の仮想公園 - 自転車メディアパーク -, VR学研報, Vol. 2, No. 1, pp. 1 - 6 (1997).
- [3] 佐藤他: パーチャルトレードショー:Splineの一応用システム, VR学研報, Vol. 1, No. 1, pp. 31 - 36 (1997).