

## サイバースペース間でのオブジェクトの移動を実現するプロトコル

中尾 太郎 † 小川 剛史 ‡ 塚本 昌彦 † 西尾 章治郎 †

† 大阪大学大学院工学研究科情報システム工学専攻

‡ 大阪大学サイバーメディアセンター

E-mail: {taro,tuka,nishio}@ise.eng.osaka-u.ac.jp, ogawa@cmc.osaka-u.ac.jp

本稿では、複数の仮想空間アプリケーションを統合的に利用するためのオープンサイバースペースプロトコル (open cyberspace protocol, OCSP) を提案する。ひとつの仮想空間管理システムで複数の仮想空間を制御するのではなく、アバターや物体を仮想空間システムから分離してネットワーク上に分散配置し、OCSP を用いてそれらを実行時に統合することで仮想空間システムから別の仮想空間システムへアバターや物体を移動させる。OCSP によってこれまで互いに独立であった複数の仮想空間アプリケーションを同じアバターで継続的に利用し、物体を持ち運んだりできるようになる。

### A Protocol to Realize Successive Use of Objects among Cyberspaces

Taro NAKAO

Takefumi OGAWA

Masahiko TSUKAMOTO

Shojiro NISHIO

† Dept. of Information Systems Eng., Graduate School of Eng., Osaka University

‡ Cybermedia Center, Osaka University

In this paper, aiming at realizing the integrative use of several applications which use virtual space, we propose the open cyberspace protocol (OCSP), which realizes the open cyberspace environments where users can move an single avatar and possibly with some objects from a virtual space to another. The OCSP integrates virtual space with avatars and objects which are separated and distributed over the network on execution. It enhances the reusability of avatars and objects and, at the same time, the availability of virtual spaces.

## 1 はじめに

近年、VR(virtual reality)やAR(augmented reality)の分野では、現実の世界をコンピュータ上に再現し、人々のさまざまな活動を支援する仮想空間が実現されてきている。その例として、サイバーモールやサイバーミュージアム [11]、都市景観や建築デザインの評価や紹介 [9]、遠隔地にいる人々のコミュニケーション [1, 5, 8] や協調作業を支援するもの [3] などが挙げられる。

これらの仮想空間では、空間内に配置する物体や空間を利用する際にユーザが用いるアバターは、それぞれの仮想空間管理システムにあらかじめ組み込まれて用意されていた。そのため、複数の仮想空間を連続して利用できるように参照関係を示すハイパーリンクで空間同士を接続しても、ユーザが用いるアバターが仮想空間ごとに変更されるなどの問題が生じ

ていた。ここで、協調作業を支援する空間で作成した物体を仮想美術館で展示したり、サイバーモールで購入したものについて他のユーザと別の仮想空間上で意見交換するなど、複数の仮想空間にまたがる活動を実現するためには、ひとつのシステムで複数の仮想空間を制御することも考えられるが、管理のコストやスケーラビリティの点で現実的ではない。

本稿では、オープンサイバースペース環境 [6] を実現するオープンサイバースペースプロトコル (open cyberspace protocol, OCSP) を提案する。オープンサイバースペース環境では、複数の仮想空間をある一つのアバターで連続して利用できる。OCSP は HTTP に基づくプロトコルであり、仮想空間管理システムから分離してネットワーク上に分散配置したアバターや物体を実行時に仮想空間に統合するための通信手順を定めている。なお本稿では、仮想空間中で扱う

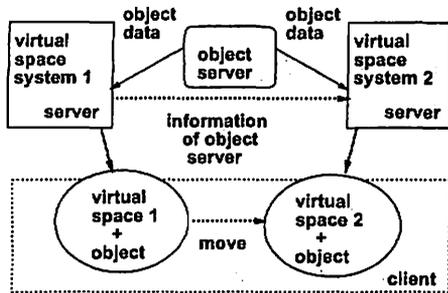


図 1: オープンサイバースペース環境におけるオブジェクトの移動

アバタや物体をオブジェクトと呼ぶ。また、仮想空間とそれを管理する仮想空間管理システムを総称してサイバースペースと呼ぶ。

以下 2 章でオープンサイバースペース環境について述べ、3 章で OCSP について述べる。4 章では、実装した仮想空間システムのプロトタイプについて説明する。5 章で議論を行った後、最後に 6 章で本稿のまとめと今後の課題について述べる。

## 2 オープンサイバースペース環境

オープンサイバースペース環境では、オブジェクトをサイバースペース (cyberspace, CS) から分離し、ネットワーク上のオブジェクトサーバ (Object Server, ObS) に分散して配置することでオブジェクトのサイバースペースに対する独立性を高めている [6, 7]。

ObS は、オブジェクトを提示するための画像データや形状データといったビジュアルデータを、外部からの要求に対して適切に提供する。一方、CS では外部から得たオブジェクトを仮想空間内に配置し、その動作を制御する。外部のオブジェクトを空間内に配置し、制御できる CS をオープンな CS と呼ぶ。

オープンサイバースペース環境でユーザが仮想空間アプリケーションを利用する場合、オープンな CS はユーザの操作に応じて ObS から必要なビジュアルデータを取得し、仮想空間にオブジェクトを挿入して提示する。ユーザが複数の CS を連続して利用する際は、CS 間で ObS への参照情報を伝播し、図 1 に示すように、空間から別の空間へのオブジェクトの移動を実現する。図 1 では、オブジェクトと仮想空間の組み合わせ方をユーザの利用状況に応じて動的に変更し、ユーザは CS1 から CS2 へ移動しても引き続き同じオブジェクトを利用している。

こうすることによって、例えばユーザはさまざまな仮想空間を全て自分専用のアバタで継続的に利用

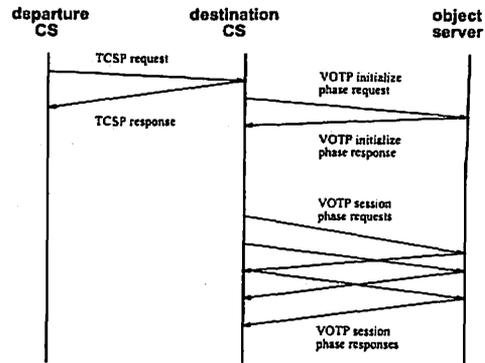


図 2: OCSP における通信手順の例

できるようになる。

## 3 オープンサイバースペースプロトコル

OCSP は、CS と CS の間や CS と ObS の間でそれぞれ必要となる通信を制御するアプリケーション層プロトコルの総称である。前者のプロトコルをサイバースペース間転送プロトコル (trans-cyberspace protocol, TCSP)、後者のプロトコルを仮想オブジェクト転送プロトコル (virtual-object transfer protocol, VOTP) と呼ぶ。なお、両者ともトランスポート層には信頼性のあるプロトコル (インターネットであれば TCP) を想定する。

図 2 に OCSP における通信手順の例を示す。CS 間でオブジェクトを移動させるために TCSP が用いられ、CS と ObS の間では、仮想空間にオブジェクトを表示するためのデータ取得に VOTP が用いられている。

### 3.1 オブジェクトリファレンス

TCSP ではオブジェクトリファレンス (Object Reference) と呼ぶ ObS を指すポインタでオブジェクトの情報を CS 間でやりとりする。各 CS はオブジェクトのビジュアルデータを Object Reference を用いた VOTP によって ObS から取得する。以下に Object Reference の記述方法を示す。なお、以下の本稿での記述文法は RFC 822[2] に基づいている。

```
object-reference = host [ ":" port ] "/" [ path ]
host              = <A legal Internet host domain
                  name of IP address (in dotted
                  decimal form), as defined by
                  Section 2.1 of RFC 1123>
port              = *DIGIT
path              = path_segments "/"
```

path\_segments = <Expressions defined by Section 3.3 of RFC 2396>

port 番号が省略された場合は 80 番 (HTTP と同じ) とする。Object Reference の記述は一意でなければならない。Object Reference の例を以下に二つ示す。

```
obj.example.com:5450/nakao/avatar/  
obj.example.com:5450/ogawa/avatar/
```

これらの Object Reference では、path の記述で区別することで、同一のホスト上で二つのオブジェクト (アバタ) のデータを独立して提供している。このように、ひとつの ObS が複数個のオブジェクトを提供する場合には、それぞれのオブジェクトを異なるルートフォルダからなる path で管理する。

### 3.2 サイバースペース間転送プロトコル

TCSP は HTTP/1.1 (RFC 2616[4]) の GET もしくは POST メソッドを利用したプロトコルであり、メッセージタイプ、メッセージヘッダ、メッセージボディ等の詳細は HTTP/1.1 に準じる。

TCSP においてはオブジェクトの移動元となる CS がクライアント、移動先の CS がサーバとなり、クライアントが移動リクエストを送信し、サーバがそれに対して移動レスポンスを送信する要求応答型の通信を行ってオブジェクトの CS から CS への移動処理を実現する。移動リクエストは、移動元 CS や移動するオブジェクトの Object Reference を引数にもつ GET もしくは POST メソッドによる HTTP リクエストであり、移動レスポンスは CS への入り口となるコンテンツをメッセージボディに含む HTTP レスポンスである。

なおこの通信では、実際のオブジェクトデータではなく、Object Reference のみを送信する。

[ 移動リクエスト ] 移動リクエストでは、リクエストメッセージ中の Request URI でオブジェクトの移動処理を行う CGI プログラム等のリソースを指定する。以下に TCSP リクエストに含める変数を GET メソッドの Query String を例として示す。

```
Query_String      = TCSP-from-CS  
                  1*(TCSP-avatar-reference)  
                  1*(TCSP-object-number)  
                  *(TCSP-object-reference)  
TCSP-from-CS     = "TCSP-from-CS=" http_URL
```

```
TCSP-avatar-reference = "&"  
                      "TCSP-avatar-reference"  
                      "=" object-reference  
TCSP-object-number   = "&"  
                      "TCSP-object-number" "="  
                      number-of-objects  
number-of-objects    = *DIGIT  
TCSP-object-reference = "&"  
                      "TCSP-object-reference"  
                      *DIGIT "="  
                      object-reference
```

アバタが複数の物体を持ち運んで CS から CS へと移動する場合などには物体毎に TCSP-object-reference の変数名に 1,2,3... と順次数字を付け加えて object-reference によって ObS を指定するとともに TCSP-object-number で移動する物体の総数を指定する。なお、TCSP-object-number にはアバタは含めない。

TCSP-avatar-reference や TCSP-object-reference によって移動するオブジェクトの ObS を通知された移動先 CS は、TCSP-from-CS で指定された http URL によって移動元 CS を特定し、移動してきたオブジェクトを仮想空間においてどの場所に配置するか決定する。

以下に TCSP リクエストの例を示す。

```
GET http://sampleCS.com/entrance.exe?TCSP-from-CS=http://gateway.space.com/main.html&TCSP-avatar-reference=obj.example.com/nakao/avatar/ HTTP/1.1
```

この場合、サイト http://sampleCS.com/ の CS において移動処理を行う entrance.exe に対し、http://gateway.space.com/main.html から Object Reference が obj.example.com/nakao/avatar/ で表わされるオブジェクト (アバタ) が移動してきたことが通知されている。

[ 移動レスポンス ] 移動レスポンスは、移動先 CS において移動処理を行った実行プログラムの出力をメッセージボディとする HTTP レスポンスである。TCSP リクエストに対する HTTP レスポンスの Status Code 200 OK は、移動先 CS でリクエストが正常に処理されたことを示す。TCSP によってオブジェクトの存在は移動先 CS に移る。

TCSP レスポンスのメッセージボディは移動先 CS の入り口を示す HTML コンテンツとする。こうすることでアバタが移動先 CS に移るとともにユーザも自動的に移動先 CS へ移る。TCSP-avatar-reference

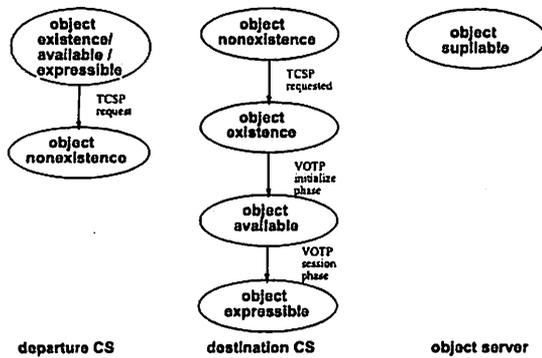


図 3: OCSP における移動元 CS, 移動先 CS, ObS の状態遷移図

を含まない TCSP リクエストの場合は, 移動元 CS が TCSP レスポンスのメッセージボディを無視する。

### 3.3 仮想オブジェクト転送プロトコル

VOTP は, CS が TCSP によってもち込まれたオブジェクトを仮想空間内に配置する際に, ObS からそのオブジェクトのさまざまなデータを取得するためのプロトコルである。VOTP も TCSP 同様, HTTP/1.1 を用いたプロトコルであり, VOTP のメッセージタイプ, メッセージヘッダ, メッセージボディ等の詳細は HTTP/1.1 に準じる。VOTP ではオブジェクトを利用する CS がクライアント, オブジェクトデータを提供する ObS がサーバとなり, 要求応答型の通信を行う。

#### 3.3.1 初期化フェーズ

VOTP は初期化フェーズとセッションフェーズの二つのフェーズをもち, これらを CS の状態に応じて使用する。図 3 に, OCSP における移動元 CS, 移動先 CS, ObS の状態遷移を示す。CS は, 外部から TCSP リクエストを受けると仮想空間内にそのオブジェクトが存在しないオブジェクト非存在状態からオブジェクト存在状態に遷移する。オブジェクト存在状態からオブジェクト利用可能状態へは, 初期化フェーズの VOTP でそのオブジェクトの ObS と通信することで遷移する。ここで初期化とは, ObS 上に用意されているオブジェクトの詳細情報を CS が知ることである。初期化フェーズの VOTP では, CS が設定リクエストを送信し, それに対して ObS が設定レスポンスを送信する。

[ 設定リクエスト ] 設定リクエストは, 以下の Request Line をもつ HTTP リクエストである。

```
Request-Line = "GET" SP
               init-file-URI SP
               HTTP-Version CRLF
init-file-URI = "http://" object-reference
               init-file-name
```

init-file-name には, オブジェクトの詳細情報が記述されたファイルの名前が使われる。このファイルをオブジェクトのイニシャルファイルと呼ぶ。イニシャルファイルが OCA-XML[7] で記述されている場合, init-file-name には "object.xml" が用いられる。

[ 設定レスポンス ] CS は, 取得したイニシャルファイルによって CS は ObS がどのようなビジュアルデータを提供できるのかを知る。こうして CS は随時必要なオブジェクトのビジュアルデータを取得できるオブジェクト利用可能状態となる。

#### 3.3.2 セッションフェーズ

オブジェクト利用可能状態となった CS は, セッションフェーズの VOTP で通信し, オブジェクトを仮想空間内に表示する。セッションフェーズでは CS がデータリクエストを ObS に送信し, オブジェクトを仮想空間に表示するためのデータをメッセージボディに含むデータレスポンスを受信する。

セッションフェーズの VOTP による通信を繰り返し行うことで, CS は仮想空間内にオブジェクトを提示する。随時データを要求できるため, CS サーバがオブジェクトの全データを ObS からあらかじめ取得・保持しておき, CS クライアントが CS サーバに接続してきたときにそのデータを全て転送する従来の方法以外に, CS クライアントが随時必要なデータだけ ObS から直接取得し, 利用する方法も可能となる。後者の方法では CS のサーバ・クライアント間でシステムの初期化時に発生する大量のトラフィックを低減できる。

[ データリクエスト ] イニシャルファイルにおいてビジュアルデータ生成プログラムの URI が指定されている場合は, データリクエストとしてオブジェクトの方向の運動状態を引数にもつ GET もしくは POST メソッドによる HTTP リクエストを用いる。Request

URI でビジュアルデータのジェネレータを指定する。以下に VOTP リクエストに含める変数を GET メソッドの Query String を例として示す。

```

Query_String      = VOTP-object-action "&"
                  VOTP-object-direction "&"
                  VOTP-object-action-sequence
VOTP-object-action = "VOTP-object-action" "="
                  object-action-name
VOTP-object-direction =
                  "VOTP-object-direction"
                  "=" object-direction
VOTP-object-action-sequence =
                  "VOTP-object-action-sequence"
                  "=" action-sequence
object-action-name = "default" | <name of the
                  action defined in the initial
                  file>
object-direction  = 1*directions
directions        = "l" | "r" | "f" | "r"
action-sequence   = *DIGIT

```

イニシャルファイルにビジュアルデータのジェネレータが指定されていない場合、CS は ObS に格納されたビジュアルデータを GET メソッドの HTTP リクエストで直接指定する。ObS を文献 [6] に示した方法で構築しておくことで、Request URI を用いたビジュアルデータの指定が行える。この場合のデータリクエストを以下に示す。

```

Request-Line      = "GET" SP
                  visual-data-URI SP
                  HTTP-Version CRLF
visual-data-URI   = "http://" object-reference
                  object-action-name "/"
                  object-direction
                  [action-sequence]
                  ".gif"

```

なお、この VOTP リクエストにおいて、VOTP-object-action と VOTP-object-direction の二つの変数は必須である。object-direction の記述は文献 [7] による。これらの変数で指定するパラメータはイニシャルファイルの記述に依存する。

[ データレスポンス ] ObS はデータリクエストに対して、データレスポンスとして Content-Type: image/gif をメッセージボディに含む HTTP レスポンスを送信する。オブジェクト以外の部分を透過ピクセルとした画像データは、ObS のビジュアルデータのジェネレータが CS からの要求に応じて動的に 3 次元モデルからレンダリングして生成してもよいし、あらかじめ用意しておいた画像データの中から適切なものを選択して提供してもよい。CS は取得した画像データを用いて仮想空間にオブジェクトを表示する。

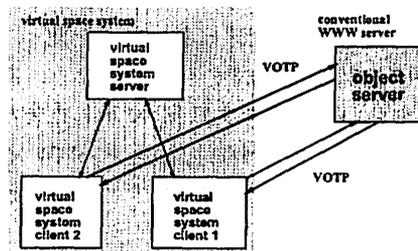
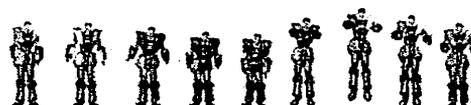
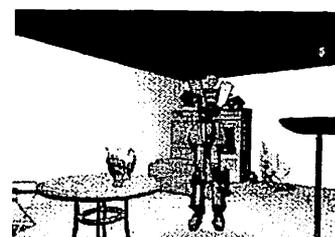


図 4: プロトタイプ CS の概要



(a)



(b)

図 5: ObS から提供される画像とその使用例

## 4 プロトタイプの実装

筆者らはこれまでに、オープンサイバースペース環境を実現するために、オープンな CS と、ObS、オブジェクトのプロトタイプを構築してきた。ここでは OSCP によって実現しているオープンサイバースペース環境を紹介し、その動作と利用の様子について述べる。

プロトタイプ CS では CS クライアントが随時 ObS から必要なデータを取得する方法を採用している。図 4 に CS の概要を示す。プロトタイプでは、CS の各クライアントはそれぞれが独自に VOTP によって ObS からデータを取得している。CS のサーバではオブジェクトのビジュアルデータを管理していない。

プロトタイプでは、例えば jump として定義されている動作を CS が実行する際は、CS がセッションフェイズの VOTP で VOTP-object-action-sequence の値を変化させながら、ObS に用意された図 5(a) に示すような一連の画像を順に取得することで、図 5(b)

のように仮想空間内で動作するオブジェクトを表現している。この ObS ではジェネレータを使ってビジュアルデータを動的に生成するのではなく、あらかじめ生成しておいた画像を文献 [6] に示した方法で格納しておき、それを提供する方法をとっている。インシャルファイルには OCA-XML [7] で記述されたものを用いている。

## 5 議論

現在 WWW 上に実現されている多くの仮想空間アプリケーションクライアントは、HTML ページ上のプラグイン [5] として、あるいは独自に拡張した HTTP クライアント [1, 8] として実現されている。それらのアプリケーションの多くでは初期リクエストを HTTP で行うため、既存の CS サーバの多くは HTTP サーバの機能をもっている。OCSP は HTTP を用いたプロトコルであり、通常の HTTP サーバとの通信が可能であるため、各 CS をオープンな CS にすることは容易である。この場合、既存の仮想空間システムで入り口部分に相当する HTML コンテンツを、移動してくるオブジェクトを処理するための CGI プログラム等に置換することで、オープンな CS にできるなど、既存の CS そのものを大きく拡張する必要はない。

そのため OCSP によって、多くの CS において互いの仮想空間同士でオブジェクトの自由な移動を可能にする接続関係が容易に実現できる。より多くの CS がオープンな CS となることで、これまで独立して存在していた仮想空間は、互いがより緊密に関係し合う仮想空間ネットワークを構成することになる。

現在、VOTP ではビジュアルデータの要求の際にカメラの被写角深度や光源などの細かなレンダリングパラメータを指定できない。また、要求応答型のプロトコルのために、本来連続的に表現されるべきオブジェクトの動作は離散的に表現されてしまう。そのため、OCSP は高精度な空間の再現とオブジェクトの精細な動作の実現を必要とするような 3 次元空間 CSCW などのアプリケーションには不向きであるといえる。前者に対しては、セッションフェイズの VOTP においてより多くの変数を指定できるようにすることが考えられる。後者に対しては、一連の動作を一括してリクエストできるようにすることで、より高いフレームレートを実現できるようにすることが考えられる。

## 6 まとめ

本稿ではオープンサイバースペース環境を実現するための HTTP ベースのプロトコル、OCSP について述べた。OCSP は HTTP との親和性が高く、既存の多くの CS を容易にオープンサイバースペースに対応させることができる。実現したプロトタイプでは、複数の CS の間を同一のアバタを操作しながら利用することができる。

今後の課題としては、既存のいくつかのシステムを実際に OCSP でリンクし、その動作を検証することを考えている。また、3 次元形状データを用いたより表現能力の高いデータ提供の可能な ObS の実現なども課題の一つである。

**謝辞** 本研究は、日本学術振興会未来開拓学術研究推進事業における研究プロジェクト「マルチメディア・コンテンツの高次処理の研究」(プロジェクト番号: JSPS-RFTF97P00501) によっている。ここに記して謝意を表す。

## 参考文献

- [1] こみゅー 3D: <http://www.commue.com/>
- [2] Crocker, D.: "Standard for The Format of ARPA Internet Text Messages," RFC 822 (1982).
- [3] Fahlen, L. E., Brown, C. G., Stahl, O., and Carlsson, C.: "A Space Based Model for User Interaction in Shared Synthetic Environments," Proc. of INTERCHI '93, ACM, pp.43-48 (1993).
- [4] Fielding, R., et al.: "Hypertext Transfer Protocol - HTTP/1.1," RFC 2616 (1999).
- [5] HelloPOP: <http://www.hellopop.com/>
- [6] 中尾 太郎, 小川 剛史, 塚本 昌彦, 西尾 章治郎: "WWW 上のオープンな仮想空間のアーキテクチャの構築について," 日本ソフトウェア科学会第 4 回プログラミングおよび応用のシステムに関するワークショップ (SPA2000), オンライン論文集 (<http://www.dcl.info.waseda.ac.jp/SPA2001/spa2001-program2.html>) (2001).
- [7] 中尾 太郎, 小川 剛史, 塚本 昌彦, 西尾 章治郎: "仮想空間の統合利用の基盤となるオブジェクト共有アーキテクチャについて," 分散, 協調とモバイルシンポジウム (DICOMO2001), pp.247-254 (2001).
- [8] ROOMANCER: <http://www.moove.com/>
- [9] 杉原健一, 松島桂樹: "GIS ベースの多目的 3 次元仮想都市空間の構築," 日本バーチャルリアリティ学会研究報告, Vol.4, No.1, ISSN 1343-0572, pp.7-12 (1999).
- [10] Tsukamoto, M: "Image Based Pseudo-3D Visualization of Real Space on WWW," Proc. of Kyoto Meeting on Digital Cities (1999).
- [11] Usaka, M. and Sakamura, K.: "A Design and Evaluation of the Multi-User Virtual Environment Server System for the Digital Museum," The 13th TRON International Symposium, IEEE (1996).