

DCOM と OpenGL による仮想空間の共有化の検討

小野 行史[†] 浅谷 耕一

工学院大学大学院電気・電子工学専攻
〒163-8677 東京都新宿区西新宿 1-24-2
Tel: 03-3342-1211

[†]E-Mail: cm99009@ccs.kogakuin.ac.jp

あらまし

本論文では、分散処理による 1 サーバあたりの処理の軽減と、サーバの故障による仮想空間共有システムへの影響を最小限に抑える事を目的とする。まず、仮想空間共有システムにおけるサーバ/クライアントのアーキテクチャについて検討する。そして、その中から本研究の目的を実現するために最適なアーキテクチャを選択し、それに基づいた仮想空間共有システムの作成と評価をおこなう。さらに作成したシステムの課題を挙げ、それに対する検討をおこなう。本研究では仮想空間共有システムの作成には、Windows 環境におけるシステム開発の容易性を考慮し、DCOM (Distributed Component Object Model) を用いてサーバにおける各機能の分散処理化をおこなった。また仮想空間の描画には OpenGL を用いた。

キーワード 仮想空間共有, DCOM, OpenGL

A Study on Shared Virtual Space by Using DCOM and OpenGL

ONO Yukihito[†] ASATANI Koichi

Graduate School of Electrical and Electronic Engineering, Kogakuin University
1-24-2 Nishi-Shinjuku, Shinjuku-ku, Tokyo, JAPAN
Tel: +81-3-3342-1211

[†]E-Mail: cm99009@ccs.kogakuin.ac.jp

Abstract

This paper describes a shared virtual space system architecture to reduce required processing power of a server by adapting distributed servers. The architecture also enables each server functions with no disturbances from possible another server's failure. It describes the outline and evaluations of the test system results. It uses DCOM(Distributed Component Object Model) which is for the development in Windows environment. OpenGL is used for construction of the virtual space.

key words shared virtual space, DCOM, OpenGL

1. はじめに

現在、インターネットの普及、PC の処理能力の向上に伴い、ネットワークを介し 3 次元仮想空間内で通信するシステムの研究・開発が盛んにおこなわれている。これらのシステムは、複数の利用者が端末-端末間で直接通信するものや、一つまたは複数のサーバを介し、一つの仮想空間内にアバタ（利用者の仮の姿）として存在することで、互いの存在を確認しながらコミュニケーションするものである。

しかしこれらのシステムは、利用者が増加するにたが、端末またはサーバにおける計算量が増加することにより遅延が生じ、利用者間での情報の共有が困難になるといった問題が生じている。

本研究ではサーバにおける処理の負荷を軽減させるため、サーバにおける処理の分散化をおこなった。システムの作成時には、Windows 環境におけるシステム開発の容易性を考慮し、DCOM (Distributed Component Object Model) [1][2]を用いた。また 3 次元仮想空間の描画を OpenGL[3]によりおこなった。

2. サーバクライアントのアーキテクチャ

仮想空間共有システムにおけるサーバ/クライアントのアーキテクチャについて、大きく分けて以下の 4 つの方式が挙げられる[4]。

(1) ピア・ツー・ピア方式

ピア・ツー・ピア方式はサーバを置かない状態で、小規模なシステム構成に適しているが、ネットワークの伝送速度や接続対地数、クライアントにおける受信処理の負荷などの点で LAN 以外のネットワーク構成には適さない。

この方式を用いたシステムの例としては DIVE[6]がある。このシステムは、サーバが存在せずマルチキャストを用いて通信をおこなうシステムであるが、クライアント間での情報の整合性を保証するために参加人数が制限されている。

(2) 機能集中型サーバ方式

機能集中型サーバ方式は、一つのサーバを置く方式であり、ピア・ツー・ピア方式に比べ接続対地数、クライアントにおける処理の負荷が少なくなるが、サーバにおける負荷が集中する。また、システムに新しい機能を追加する場合や、サーバの故障時に与える影響を制限しにくいという問題も存在する。

この方式を用いたシステムの例としては[6]がある。

(3) 一部ピア・ツー・ピア方式

一部ピア・ツー・ピア方式は、ピア・ツー・ピア方式と機能集中型サーバ方式を組み合わせたものであるが、ピア・ツー・ピア方式と同様の問題が存在する。

この方式を用いたシステムの例としては[7]がある。ただし、この例では仮想空間内を複数の領域に分割し、その領域ごとにサーバを設けることによりサーバの負荷を軽減している。しかしながら、ピア・ツー・ピア方式と同様の問題が存在している。

(4) 機能分散型サーバ方式

機能分散型サーバ方式はサーバを機能ごとに分ける方式で、システムに新しい機能を追加する場合や、送受信処理の分散に柔軟に対応できる。また、一つのサーバに故障が発生した場合でも他のサーバに影響を与えないというメリットもある。

この方式を用いたシステムの例としては PAW (Personal Agent World) [8][9]がある。これはサーバの各機能が AO (Application Object) という単位で分けられており、各 AO を分散させることができる。ただし各 AO 同士も通信を行なう場合があり、その場合各 AO を分散させると通信時間の影響によ

りサーバ全体としての処理時間が増加する場合がある。

本研究ではサーバにおける処理の分散化と、今後システムに新しい機能を追加する場合を考慮し、機能分散型サーバ方式によりシステムの作成をおこなう。

3. 作成したシステムの概要

3.1 システムの目的

本研究では以下を目的として仮想空間共有システムの開発をおこなった。

- (1) サーバにおける処理の負荷を分散させるために、サーバ/クライアントのアーキテクチャには機能分散型サーバ方式を用いる。
- (2) サーバ間の通信はおこなわないものとし、一つのサーバが故障した場合でも他のサーバに影響を与えないようにする。
- (3) 複数の利用者間においてテキストによる会話（テキストチャット）を可能にする。

なお、本システムは Windows NT 4.0 において C++ 言語を用いて作成をおこなった。また、Windows 環境におけるシステム開発の容易性を考慮し、DCOM を用いた。また仮想空間の描画には OpenGL を用いた。

3.2 システムの構成

本システムの構成を図 3.1 に示す。

図 3.1 に示したシステムは機能分散型サーバ方式であり、それぞれ一つのチャットサーバと仮想空間サーバに全クライアントが接続し、利用者は仮想空間内にアバタとして存在することで、他の複数の利用者との仮想空間の共有とテキストによる会話を実現するものである。

図 3.1 において、クライアントは、仮想空間サーバに利用者の名前とアバタの状態（位置、向き）、チャットサーバに利用者の名前と会話を送信する機能を持つ。さらにクライアントは二つの COM オブジェクト (VirtualSpaceClient, ChatClient) を持ち、VirtualSpaceClient はサーバから送信された利用者の名前とアバタの状態を受信し、その情報を基に仮想空間を描画する機能、ChatClient はサーバから送信された利用者の名前と会話文を受信し表示する機能を持つ。

一方、仮想空間サーバは、利用者の名前とアバタの状態を全利用者に受信するよう要求する機能を持つ。また仮想空間サーバは一つの COM オブジェクト (VirtualSpaceServer) を持ち、VirtualSpaceServer は利用者の名前をリストとして保存しておく機能と、クライアントから送信された利用者の名前とアバタの状態を受信する機能を持つ。

チャットサーバは、利用者の名前をリストとして保存しておく機能と、利用者の名前と会話文を全クライアントに受信するよう要求する機能を持つ。またチャットサーバは一つの COM オブジェクト (ChatServer) を持ち、ChatServer はクライアントから送信された利用者の名前と会話文を受信する機能を持つ。

クライアントから仮想空間サーバに利用者の名前とアバタの状態を送信する場合は、VirtualSpaceServer が公開するインターフェイス (ITransactionState) を介し、VirtualSpaceServer が持つメソッド (put_State0) を実行することにより、仮想空間サーバに利用者の名前とアバタの状態を送信することができる。

仮想空間サーバから全クライアントに利用者の名

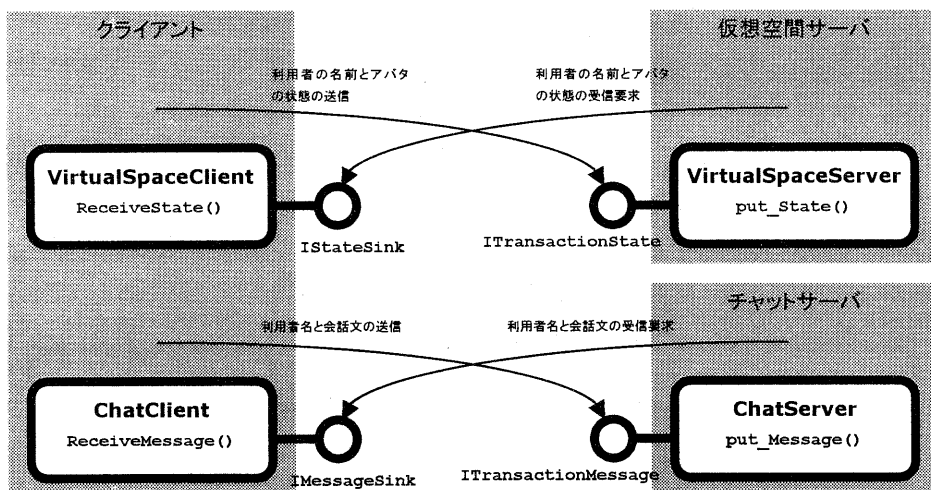


図 3.1 システムの全体構成

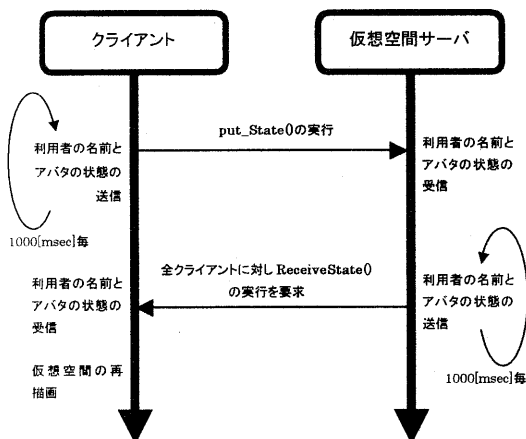


図 3.3 クライアントー仮想空間サーバ間の処理過程

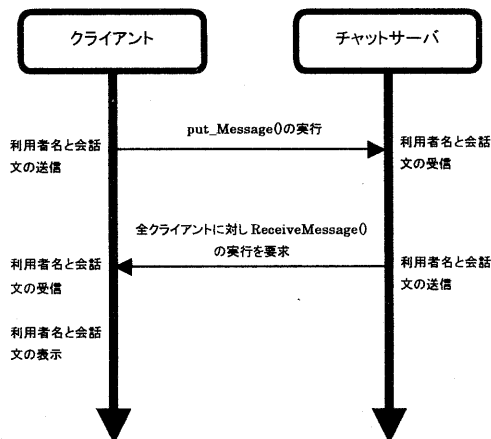


図 3.2 クライアントーチャットサーバ間の処理過程

前とアバタの状態を受信するよう要求する場合は、VirtualSpaceClient が公開するインターフェイス (IStateSink) を介し、ChatClient が持つメソッド (ReceiveState0) を実行するよう VirtualSpaceClient に要求する。そして VirtualSpaceClient は ReceiveState0 を実行することで仮想空間サーバから利用者の名前とアバタの状態を受信することができる。

クライアントからチャットサーバに利用者の名前と会話を送信する場合は、ChatServer が公開するインターフェイス (ITransactionMessage) を介し、ChatServer が持つメソッド (put_Message0) を実行することにより、サーバに利用者の名前と会話を送信することができる。

チャットサーバから全クライアントに利用者の名前と会話を受信するよう要求する場合は、ChatClient が公開するインターフェイス

(IMessageSink) を介し、ChatClient が持つメソッド (ReceiveMessage0) を実行するよう ChatClient に要求する。そして ChatClient は ReceiveMessage0 を実行することでサーバから利用者の名前と会話を受信することができる。

3.3 システムの処理過程

クライアントが利用者の名前とアバタの状態を送受信し、仮想空間を描画するまでの処理過程を図 3.2 に示す。

図 3.2 において、アバタが仮想空間内において移動、向きを変えるなどの行動をおこなった場合、VirtualSpaceClient はアバタの最新の状態を保存しておき、1000[msec]ごとに put_State0 を実行し、仮想空間サーバに利用者の名前とアバタの最新の状態を送信する。

仮想空間サーバでは、各クライアントから送信さ

れた各利用者の名前とアバタの最新の状態を保存しておき、1000[ms]ごとに全クライアントにReceiveState0を実行するよう要求する。そしてVirtualSpaceClientは仮想空間サーバから各利用者の名前とアバタの状態を受信し、その情報を基に仮想空間を描画する。

クライアントが利用者の名前と会話文を受信し、表示するまでの処理過程を図3.3に示す。

図3.3において、利用者から名前と会話文の送信要求が発生した場合、put_Message0を実行し利用者名と会話文を送信する。

チャットサーバでは、クライアントから利用者名と会話文を受信すると、直ちに全クライアントにReceiveMessage0を実行するよう要求する。そしてChatClientはチャットサーバから利用者名と会話文を受信し、表示する。

3.4 システムの特徴

本システムの特徴として以下のものが挙げられる。

(1) サーバにおける処理の分散が可能

本システムはDCOMにより作成したことにより、仮想空間サーバとチャットサーバを同一のマシン上に存在させる必要がない。したがって各サーバの処理を分散させることができる。

(2) サーバが故障した場合でもその被害を最小限に抑えることが可能

仮想空間サーバ、チャットサーバ間では通信をおこなっていないため、一つのサーバが故障した場合でも他のサーバに影響を与えることはなく、被害を最小限に抑えることができる。

(3) 新しい機能の追加に柔軟に対応できる

本システムに新しい機能を追加する場合は、既存のサーバをそのまま再利用でき、利用者は新しい機能に対応したクライアントプログラムによってその機能を利用できる。また既存の機能だけを利用するのであれば、既存のクライアントプログラムをそのまま利用することも可能である。

(4) サーバのセキュリティレベルを管理者レベルでカスタマイズできる

Windows NT においては、DCOM により作成されたサーバのセキュリティレベルを変更できる機能が標準で装備されており、本システムのサーバもこの機能を利用することにより、サーバのセキュリティレベルを再コンパイルすることなく、管理者レベルでカスタマイズできる。

(5) 複数の利用者でのテキストチャットが可能

文字情報をすべての利用者間で送受信することによりテキストによる会話ができる。

4. 結果と評価

4.1 ハードウェア構成

本システムの実験時の構成図を図4.1に示す。図4.1に示すように、仮想空間サーバ、チャットサーバ、クライアントともWindows NT 4.0 上において利用者数3名によりおこない、ネットワークは100Base-TXのEthernetを用いた。

4.2 実行結果と評価

本研究で作成したシステムにおけるクライアントのGUI部分を図4.2に示す。

図4.2では、一人の利用者(ユーザA)が他の二人の利用者と3次元仮想空間を共有し、テキストチャットをおこなっている様子を示しており、仮想空間の表示部分には他の二人のユーザのアバタが表示されているのが確認できる。また、ユーザAが他の

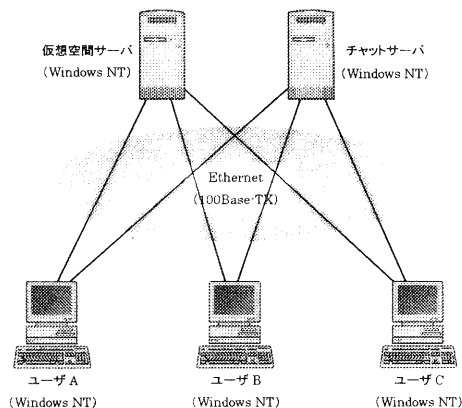


図4.1 ハードウェア構成

二人のユーザと会話をおこなえていることが確認できる。

本システムにおいて、利用者はマウスの操作により仮想空間内を移動したり見回したりすることができる。

テキストチャットをおこなう場合、利用者は“Chat:”というラベルのついたテキストボックスに会話文を入力して“Send”ボタンもしくはリターンキーを押すことによりサーバを介し全クライアントに会話文を送信することができる。クライアントが受信した会話文は“Chat Log:”というラベルのついたテキストボックスの最下行に利用者名と共に追加され、これにより他の利用者とのテキストによる会話をおこなうことができる。

また、本システムの実行中にチャットサーバのみを終了させたが、テキストチャット以外の機能は継続して利用することができた。逆に仮想空間サーバのみを終了させたが、仮想空間共有以外の機能は継続して利用することができた。よってサーバが故障した場合でも、その被害を最小限に抑えることができる。

5. 問題点と解決方法に関する検討

本システムにおける問題点としては、クライアントプログラムは特定のOS(Windows NT 4.0)に依存しているという問題点が挙げられる。

仮想空間サーバ、チャットサーバについてはDCOMにより作成したため、他のUNIX系のOSやMacOSなどのOSからでも利用可能であるが、クライアントプログラムはC++言語により作成しているため、利用環境は特定のOS(Windows NT 4.0)に限定されてしまっており、異機種間での通信を実現するためにはOSごとにクライアントプログラムを作成する必要がある。

この問題の解決策としては、クライアントプログラムをJava言語により作成する方法が考えられる。Java言語を用いれば、プラットフォーム独立のクライアントプログラムを作成でき、OSごとにクライアントプログラムを作成する負担が軽減できる。Java言語はDCOMをサポートしているので、この解決策は実現可能と思われる。

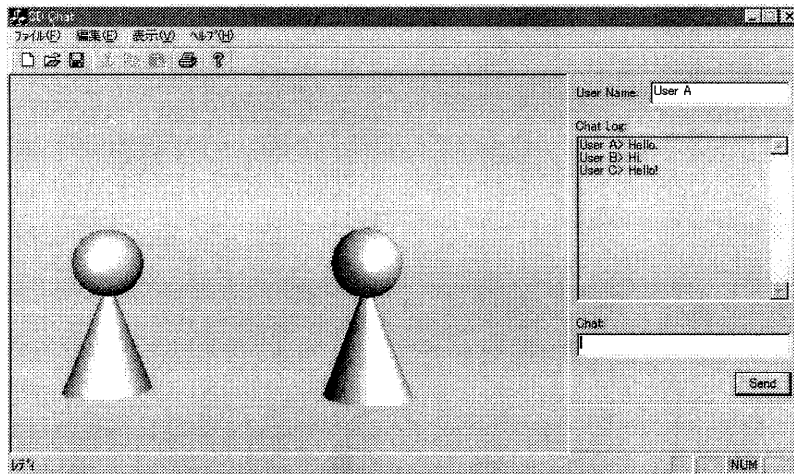


図 4.2 クライアント GUI

6. おわりに

本論文では、サーバ/クライアントのアーキテクチャについて検討し、機能分散型サーバ方式の仮想空間システムの作成と評価をおこなった。本研究により、機能分散型サーバ方式に基づき、仮想空間共有システムを作成し、各サーバにおける処理の軽減、さらにサーバの故障によるシステムへの影響を最小限に抑えることができた。

さらに、作成したシステムの問題点を挙げ、その解決法に関する検討をおこなった。

今後は本システムにおける問題点を解決するとともに、新しい機能を利用者が多人数になった場合の各サーバにおける負荷の軽減等に関しても検討をおこなっていく予定である。

参考文献

- [1] <http://www.microsoft.com/japan/com/>
- [2] リチャード・グリムス, アレックス・ストックトン, ジョージ・ライリー, ジュリアン・テンブルマン著, “ATL COM プログラミング”, 翔泳社, 1999.
- [3] <http://www.opengl.org/>
- [4] 清末悌之, 菅原昌平, 加藤洋一, 山名岳志, “インターネット空間のアーキテクチャ”, NTT R&D Vol.47 No.4, 1998.
- [5] <http://www.sics.se/diver/>
- [6] 川上義雄, 横井茂樹, 安田孝美, “VRML に基づく三次元仮想空間の共有化に関する検討”, 信学技報 MVE98-74 pp.7-14, 1998.
- [7] 賀戸大輔, 藤川和利, 下條真司, 宮原秀夫, “共有型仮想空間におけるリアルタイム性・信頼性を考慮したデータ配送機構の実現”, 信学技報 MVE98-89 pp.25-32, 1999.
- [8] 松田晃一, 上野比呂至, 三宅貴浩, “パーソナルエージェント指向の仮想社会「PAW」の評価”, 電子情報通信学会論文誌, Vol.J82-D-II No.10 pp.1675-1683, 1999.
- [9] <http://www.so-net.ne.jp/paw/>