社団法人 情報処理学会　研究報告
IPSJ SIG Technical Report

2003－DPS－114　(20)
2003－EIP－ 20　(20)
2003／8／29

# マルチエージェントに基づく QoS アウェアな３次元共有空間の設計と実装

大前 良介 †　菅沼 拓夫 †　白鳥 則郎 †

**あらまし**　近年，３次元共有空間を用いたアプリケーションが普及しつつある．３次元共有空間は，一般に高速な画像処理能力と広帯域ネットワークが必要であり，刻一刻と変化する計算機・ネットワークの資源状況下で，サービスを快適な品質で享受するためは，利用者自身が適切な設定を行わなくてはならない．これは不慣れな利用者にとっては大きな負担となっている．そこで本研究では，３次元共有空間の QoS 調整の自動化によりシステムの利便性を向上することを目的とする．本稿ではマルチエージェント指向コンピューティングに基づく QoS アウェアな３次元共有空間の設計と実装，および評価について述べる．

**キーワード**　３次元共有空間，マルチエージェント，QoS 制御

# Design and Implementation of Multi-User 3D-Virtural Space with QoS Awareness Based on Multi-Agent Framework

**Ryosuke Ohmae†, Takuo Suganuma†, Norio Shiratori†**

**Abstract**　Recently, applications using 3D-virtual space technology become widely used. Generally, the 3D-virtual space applications need powerful image processing capability of computer platform and broadband network environment. Therefore, user must perform appropriate setting of the system to receive the service from the virtual space with comfortable QoS, under the situation that computer and network resource are changing in every moment. This task is a big hurdle for novice users. This work is aiming for the improvement in convenience of user of 3D-virtual space system by the automatic mechanism of QoS control. In this paper, we explain the design, implementation of the 3D-virtual space with QoS awareness based on multiagent framework. Furthermore, we evaluate the effectiveness of our proposal with some experiment results of the prototype system.

**key words**　3D-virtual space, Multiagent framework, QoS control

## 1　はじめに

近年，計算機技術とインターネットの発展に伴い，より高度なインターフェイス・コミュニケーションツールとして，３次元共有空間を用いたアプリケーションが普及しつつある［1–4］．３次元共有空間とは，利用者端末でコンピュータグラフィックスを用い表現された仮想的な３次元空間を，ネットワークを介して利用者間で共有する技術である．３次元共有空間を用いることで，地理的に離れた利用者同士が，あたかも同じ場所にいるかのような感覚でコミュニケーションや作業を行うことができる．これに基づいて，3D チャットシステムや協調設計支援システムなど，さまざまな共有空間がネットワーク上に構築されつつある．

その一方で，この技術の実現には，提供されるサービスに見合った高い処理能力と高速なネットワークが必要となり，刻一刻と変化する計算機・ネットワークの資源状況 (CPU 利用率，帯域使用率など) 下においてサービスを快適な品質で享受するためには，

† 東北大学電気通信研究所/情報科学研究科 / Research Institute of Electrical Communication/Graduate School of Information Science, Tohoku University

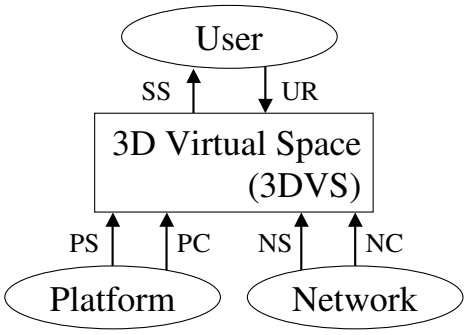


図 1: QoS アウェアな３次元共有空間の動作状況表現

利用者自身が適切な設定を行わなくてはならない．この作業は，計算機技術やネットワークに対する知識をほとんど持たない初心者にとっては非常に煩わしく，大きな負担となっている．

そこで本研究では，３次元共有空間の利用者の負荷を軽減し，その利便性を向上するシステムの実現を目指している．具体的には，利用者の要求，計算機・ネットワークの状況を自律的に獲得し，それらの情報を元に自動的に共有空間の提供する QoS を制御する，QoS アウェアな３次元共有空間の実現を目的としている．

本稿では，上記の QoS アウェアな３次元共有空間を構成するための技術として，マルチエージェント技術に着目し，これに基づく提案システムの設計および実装について述べる．提案システムのプロトタイプを用いた実験により，本システムの評価を行ったところ，利用者要求，計算機資源，ネットワーク資源の変動に対しこれまでにない柔軟な制御が行えることを確認した．

## 2 ３次元共有空間の QoS 制御

### 2.1 QoS アウェアな３次元共有空間のモデル

これまで，３次元共有空間の QoS 制御に関しては，いくつかの先行研究がなされてきた [5, 6]．これらは主に，システムに対する影響を考慮したものであり，利用者の視点での利便性までを考慮した制御方式については十分検討されていない．そこで，本研究では従来にない以下のような特徴をもつ，QoS アウェアな３次元共有空間を実現する．

- 利用者の設定作業の負荷軽減
- 使用中のリソース変化への動的な対応
- 利用者要求に基づく QoS 制御

QoS アウェアな３次元共有空間の動作状況を図 1 に示すように表現する．

利用者を表現する $User$ は，３次元共有空間 $3DVS$ に対し共有空間サービス品質に関する利用者要求 $UR$ を持つ．また $3DVS$ は $User$ に対しある品質 $SS$ にてサービスを提供する．ここで，利用者要求に対し十分な提供サービス品質が与えられているときは，利用者の要求が満たされている状況であり，望ましい状態である．一方，利用者要求に対し十分な提供サービス品質が与えられていないときは利用者要求が満たされていない状況であり，望ましくない状態である．

同様に，計算機資源を表現する $Platform$ とネットワーク資源を表現する $Network$ も，共有空間サービス品質 $SS$ の状態に影響を与える．$Platform$ と $Network$ は $3DVS$ に対し，資源利用に関する制約条件 $(PC,\ NC)$ を示す．また $Platform$ や $Network$ は $3DVS$ に対し計算機資源の利用状況 $(PS,\ NS)$ を伝える．ここで，計算機資源利用に関する制約条件の範囲内に利用状況が収まっているときは，制約が充足されている状況であり，望ましい状態である．一方制約条件の範囲を利用状況が超えているときは，制約が充足されていない状況であり，望ましくない状態である．

$UR,\ PS,\ NS$ は時間経過とともに変動する．当初の $NS$ に対する変化分を $\Delta NS$ と表記すれば，$NS$ は $NS \pm \Delta NS$ に変化する．同様に，$UR$ や $PS$ の変化分もそれぞれ $\Delta UR,\ \Delta PS$ として表記する．提案システムでは，$UR$，$PS$，$NS$ を自ら検出し，これを軽減・回避するために，３次元共有空間の各種パラメータの変更・調整などの操作を自動的に発動する．これらの操作を，変化に対する逆操作，すなわちカウンターアクション $\delta 3DVS$ と呼ぶ．これらの動作によってサービス品質の劣化や変動を軽減・抑制することにより，システムにとって望ましい状態を維持するように自律的に動作する共有空間が，QoS アウェアな３次元共有空間であるとする．

### 2.2 技術的課題

このようなシステムを実現する上で以下のような技術的課題がある．

(P1) **３次元共有空間に対する利用者要求獲得手法**
利用者の要求に基づいた制御を行うには，利用者がいつ，どのように制御して欲しいかという詳細な利用者要求を獲得することが必要である．

(P2) **３次元共有空間の効果的な QoS 制御法**
３次元共有空間は多種多様かつ多数のオブジェクトにより構成されるため，制御対象が非常に多くなり，きめ細かな制御を行うため

には効果的な制御法が必要となる．

これに対し，本研究では以下のような解決法を与える．

(S1) **利用者エージェントによる利用者要求獲得** (P1 の解決)

(S2) **オブジェクトをエージェント化し，エージェント間協調による QoS 制御** (P2 の解決)

上記 (S1)，(S2) について，以下の 2 つの節で詳細に述べる．

### 2.3 利用者エージェントによる利用者要求獲得

大きく分けて 2 つの手法により，利用者エージェントを用いて利用者要求獲得を行う．

一つは，3 次元共有空間における QoS の監視である．利用者は QoS が低下すれば必然的にそれを向上させたいという要求を持つと考えられる．そこでエージェントに 3 次元共有空間の QoS 定義，すなわち $SS$ の表現形を知識として与え，これに基づいて提供サービスの状況を監視する．この知識に基づいて $UR$ も表現されるので，$SS$ と $UR$ の差分によって，いつ，どのように制御を行ってほしいかという詳細な利用者要求を獲得することが可能となる．なお，簡単化のため，ここでは 3 次元共有空間の QoS を表示品質（ポリゴンの精密さ等）と動作品質（fps の高さ等）から定義するものとする．3 次元共有空間の QoS 定義についてはさらに詳細な検討が必要であるが，他稿に譲ることとする．

もう一つは，利用者の 3 次元空間内での行動の監視である．移動中であれば，動いているものを追いかけたりしている状況であるから，動作品質を優先するであろうし，静止中であれば，何かをじっくり見ている状況であるから，表示品質を優先すると予想される．以上から表示品質・動作品質要求を取得する．また，オブジェクトと利用者とのインタラクションを監視することにより，利用者にとって重要なオブジェクトを知ることができ，オブジェクトの重要度要求を獲得することができる．以上の手法により，$UR$ を自動的に獲得する．

### 2.4 オブジェクトのエージェント化

やわらかいビデオ会議システム [7] では，2.1 節と同様な動作表現に基づきビデオ会議サービスの QoS 制御を行っている．3 次元共有空間の QoS 制御を行う上で，2 次元のビデオ会議システム等と大きく異なる点は，ビデオ会議システムでは一つの画面単位での fps・解像度などの品質の調整のみを考慮すればいいのに対し，3 次元共有空間では内部のデータ構造的にそれぞれのオブジェクトが独立して扱われるため，一つの画面内に映し出されるオブジェクト単位での制御が必要となることである．これによって，制御する対象が非常に多くなるため，個々のオブジェクトをいかに管理・制御するかが問題となる．

本研究では，オブジェクトに対して，利用者との関係から得られる重要度や履歴を管理する機能，及び，自身の品質を決定する機能を与える．この操作をエージェント化と呼ぶ．オブジェクトと利用者との関係は，それぞれの利用者・オブジェクトごとに異なっているため，各オブジェクトをエージェント化することで，利用者にパーソナライズされた知的で柔軟な制御が行える．

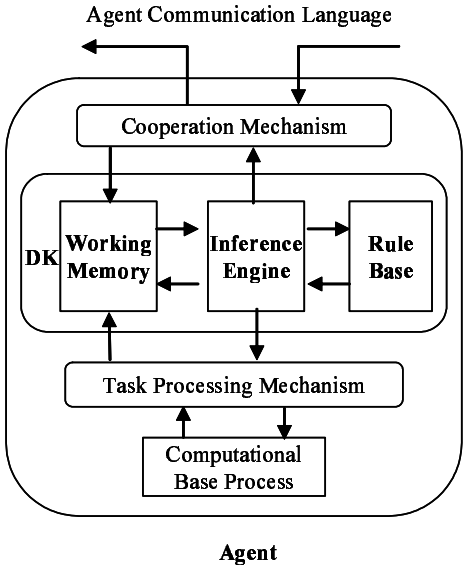


図 2: エージェントアーキテクチャ

## 3 設計

### 3.1 エージェントアーキテクチャ

本システムで用いるエージェントのアーキテクチャを図 2 に示す．Computational Base Process (BP) は，エージェントの保持する計算プロセスであり，本システムにおける 3 次元共有空間内のオブジェクトなどである．BP をエージェントとして動作可能とすることがすなわちエージェント化である．Cooperation Mechanism (CM) は，他のエージェントとのメッセージ交換を行うためのメカニズムである．エージェント間では，エージェント間通信言語によって規定された形式により，メッセージ交換を行う．Domain Knowledge (DK) は，そのエージェントの持つ BP に関する様々な領域知識を保持し，その知識に基づき BP の監視・制御を行ったり，他のエージェントに対して働きかけを行うた

めの知識ベースシステムである．Task Processing Mechanism (TPM) は，BP を直接的に制御するためのインタフェースである．BP からのイベント，たとえば例外通知などを DK に通知し，また DK の指示により，BP のシステムパラメータを直接変更する．

DK はさらに，Working Memory，Inference Engine，Rule Base によって構成される．Working Memory 内には Fact の集合が，また Rule Base には Rule の集合がそれぞれ保持され，Inference Engine がそれらを参照することによって，プロダクションシステムとしての動作を行う．これにより，CM を介した他のエージェントとのインタラクションや，BP に対する制御・監視などのアクションを実行する．

### 3.2 機能設計

本研究の提案システムに必要な機能は以下の通りである．

(F1) **QoS 監視機能**
QoS を監視することによって，いつ，どのように制御を行うかを決定する．

(F2) **リソース監視機能**
QoS に直接関係してくるリソースの状態を監視することで，実際に QoS の低下などが発生する前に制御を行う．

(F3) **利用者要求獲得機能**
３次元空間内の利用者の行動を監視し，利用者の制御に対する要求を推測する．

(F4) **オブジェクト重要度管理・品質決定機能**
オブジェクトごとに利用者との関係などを管理しており，調整が必要となったとき，その情報を元にオブジェクトの品質を決定する．

(F5) **解決方針決定機能**
様々なアプリケーションの制御を行う場合，その制御方針は対象とするアプリケーションの特性によって異なってくる．この機能はそれらの特性を元に制御方針を決定しアプリケーションごとの差異を吸収する．

(F6) **端末間交渉機能**
共有空間では自分本位の制御だけでは調整しきれない要因がある．そこで端末間で交渉を行うことにより，全体としてよりよい制御を行う．

### 3.3 エージェント構成設計

本研究ではこれらの機能をマルチエージェントを用いたミドルウェアとして実装する．そして，そのエージェントミドルウェアが３次元共有空間アプリケーションの制御パラメータを設定することで QoS 制御を行う．

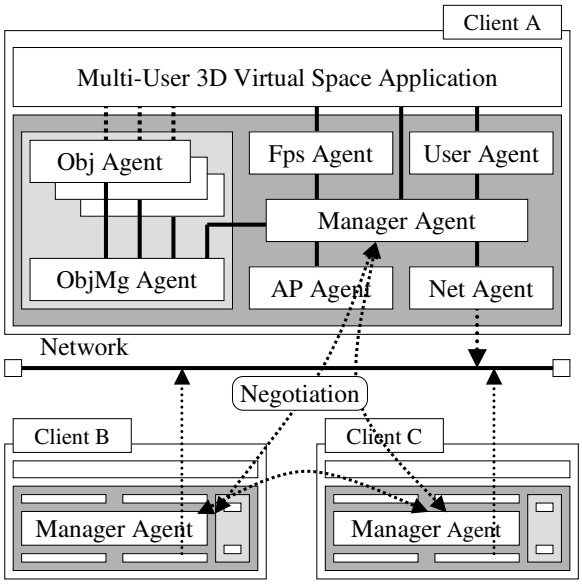


図 3: エージェント構成設計

マルチエージェントを用いたミドルウェアとして実装を行う理由は，ミドルウェアとして実装することで，既存の技術を生かしつつ新しい制御法を提供できるからである．また，エージェントの持つ自律性・協調性がこれらの機能の実現に有効であり，ミドルウェアを設計する上で高い拡張性を確保できるからである．

図 3 にエージェントの構成を示す．それぞれのエージェントは以下のような機能を持つ．

**Manager Ag**
エージェント全体を管理しており，各監視エージェントから状況報告を受け，調整を指示する．また，必要に応じて他の利用者端末のエージェントと交渉を行う．

**Obj Ag**
エージェント化したオブジェクト．自身と利用者の関係 (重要度) を管理しており，Manager の指示を受け自身の品質を決定する．

**ObjMg Ag**
Obj Agent を管理し，空間全体の品質を保持する．閾値を超えて品質が変化した場合 AP Agent へ通知する．

**User Ag**
利用者の空間内の移動から利用者要求を獲得し，Manager Agent へ通知する．

**Fps Ag**
QoS パラメータの一つである fps の監視を行い，閾値を一定時間超えると Manager Agent に警告メッセージを送信する．

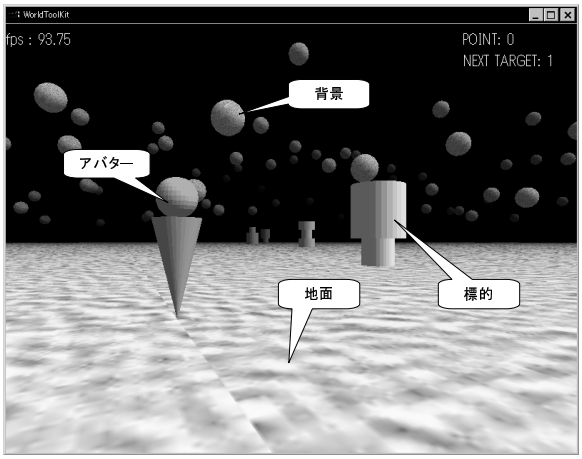


図 4: 実装したアプリケーション

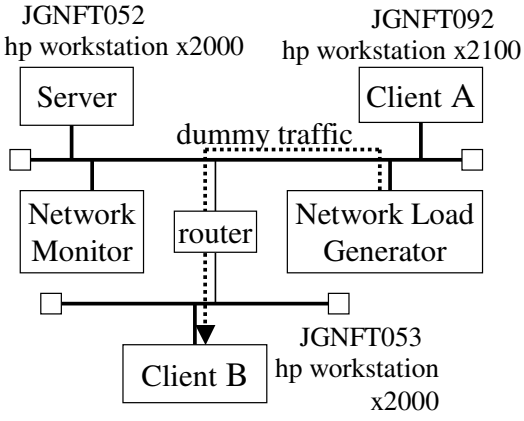


図 5: 実験環境

**Net Ag**

ネットワークトラフィックの監視を行い，閾値を一定時間超えると Manager Agent に警告メッセージを送信する．

**AP Ag**

アプリケーションの特徴に応じて動作品質と表示品質の妥協点を決定する．

## 4 実装

3 次元空間アプリケーションの実装には C 言語ベースの 3 次元空間構築用ライブラリ World Tool Kit (Windows OpenGL 版) を用い，3 次元空間共有にはサーバ-クライアント型空間共有システム World to World を用いた．

エージェントの実装にはルールベースのマルチエージェントフレームワークである DASH-1.1d (DASH: Distributed Agent System based on Hybrid architecture) [8]を用いた．

DASH は Java ベースのフレームワークであり，World Tool Kit はネイティブコードで実行される．本研究はミドルウェアとしての制御機構を実現することを目指している．そのためアプリケーションがどのようなコードで実装されても，この機能を利用できるように，アプリケーションとエージェントとの接続は，TCP 通信を用いた接続方法を採用した．

図 4 に実装した制御対象アプリケーションを示す．これは，利用者同士がコミュニケーションを行いながら標的を追いかけるという簡単なゲームを想定している．

このアプリケーションにはアバター，標的，背景，地面といったオブジェクトが存在している．これらの重要度の初期値は，アバター/高，標的/中，背景・地面/低と設定した．

今回の実装で，処理量の調整に用いるパラメータは LOD(Level of Detail) の切り替え位置である．LOD は視点から離れたオブジェクトの表示品質を下げ，全体の処理量を減らす技術である．この切り替え位置を小さな値にすると全体で表示品質の低いオブジェクトが増え，処理量を減らすことができる．また，通信量の調整に用いるパラメータは各オブジェクトの送信レートである．実装環境には送信レートについて，共有プロパティ単位，受信レートについてコネクション単位で調整する機能が備わっているが，今回は簡単化のため送信レートのみを用いた．これら調整パラメータは簡単化のため，0〜5 の 6 段階の値を指定するようにした．

## 5 実験と評価

### 5.1 実験の概要

実装したプロトタイプシステムを用いて実験を行った．実験では計算機・ネットワーク資源状況および利用者要求の変化に対するシステムの動作を観測した．実験において fps を計測した計算機は hp workstation x2000(Pentium4 1.7GHz, 1024MB, FireGL2, WindowsNT4.0) である．ネットワークに関する実験環境を図 5 に示す．計算機は fps の計測に用いた hp workstation x2000 と，hp workstation x2100(Pentium4 2.4GHz, 1024MB, Quadro4 900XGL, Windows 2000) を用いた．これらの計算機が Giga bit LAN で接続されたサーバ-クライアントネットワークに，1 台のクライアントを帯域制限を設けたルータを介して 10Mbps の LAN で接続した環境を用いた．

### 5.2 計算機・ネットワーク資源変化への対応

まず，アプリケーションの動作中にバックグラウンドタスクが発生したことを想定し，実行中に故

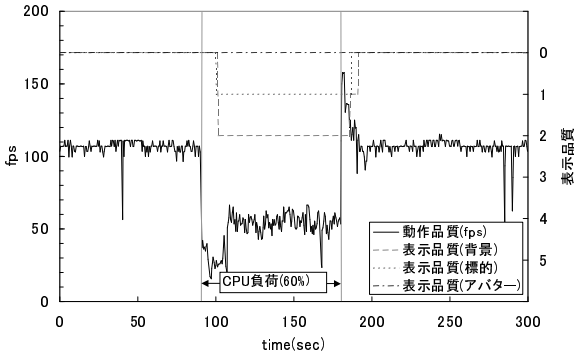

図 6: CPU 負荷の変動に対する制御

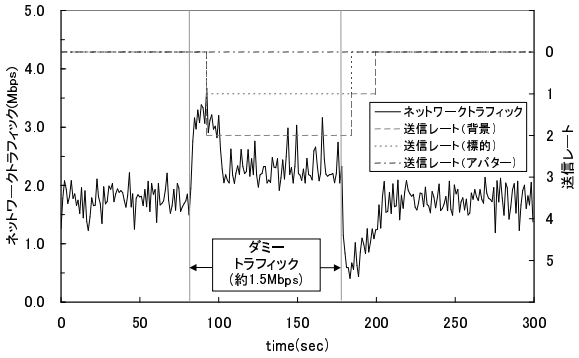

図 7: ネットワークトラフィックの変動に対する制御

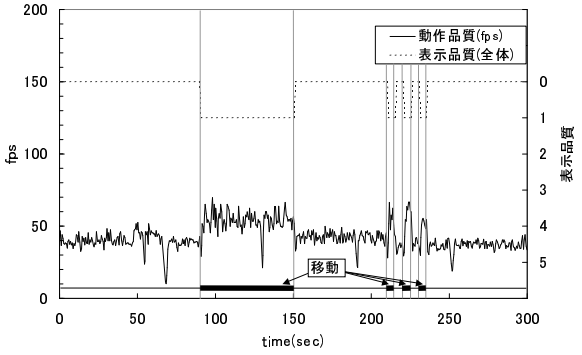

図 8: 移動の監視に基づく制御

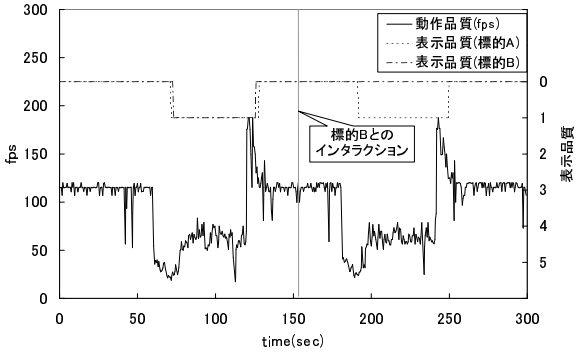

図 9: インタラクションに基づく制御

意に CPU に負荷をかけ，CPU 資源を減少させた ($\Delta PS$)．そのときのシステムの挙動を観測した．結果を図 6 に示す．CPU に負荷をかけると，処理能力が低下し，fps が劣化している ($\Delta SS$)．これに対しこの $\Delta SS$ を検出し，それぞれのオブジェクトの重要度に応じて表示品質を下げ，fps をある程度回復させる動作 ($\delta 3DVS$) を行っていることがグラフから読みとれる．

次に，使用中にネットワークトラフィックが増大した状況を想定し，実行中にダミートラフィックを発生させた，ネットワーク資源を減少させた ($\Delta NS$) ときのシステムの挙動を観測した．結果を図 7 に示す．先ほどと同様に，トラフィックの増大を検出し，それぞれのオブジェクトの重要度に応じてデータの送信レートを下げる $\delta 3DVS$ を起動することで全体のトラフィックを低く抑えていることがグラフから読みとれる．

### 5.3 利用者要求変化への対応

次に，利用者の空間内の動作を監視することによる制御が行えることを検証するため．以下のような実験を行った．

まず，利用者の移動を監視することによる制御を確認するため，移動と静止を繰り返させた．結果を図 8 に示す．利用者の移動を検出すると，利用者要求の変化 ($\Delta UR$) が発生したとみなし，表示品質を一段下げ，動作品質を上昇させる $\delta 3DVS$ を発動している．ここで，利用者の移動と静止は短時間で変動することが考えられるため，制御はどのオブジェクトの品質を下げるかということを考慮せず，一律で下げている．このことで，図 8 の 200 秒から 250 秒の間にみられる，短時間で変動する要求にも素早く対応できている．

次に，オブジェクトと利用者のインタラクションを監視することによる制御が行えることを検証した．結果を図 9 に示す．一度目の制御では，標的 A・B ともに同様の表示品質の制御が行われている．次に標的 B と利用者のインタラクションが発生した後の制御では，標的 B の重要度が高くなったとの $\Delta UR$ を検出し，標的 B の表示品質を維持している．

### 5.4 評価

以上のように，今回実装したシステムは，計算機資源・ネットワークトラフィックの変動を自動的に検出し，柔軟な制御が行えた．また，利用者の行動を監視することで自動的に利用者要求の変化を検出し，それに基づいた効果的な制御が行えることが確認できた．これらは，利用者エージェントによる利用者要求獲得とエージェント化したオブジェクトの自動制御による効果が現れているためであると考えられ

表 1: 関連研究と本研究の比較

| | [5] | [6] | 本研究 |
|---|---|---|---|
| 計算機負荷の軽減 | ○ | ○ | △ |
| 端末能力にあわせた制御 | × | ○ | ○ |
| 利用者負荷の軽減 | - | × | ○ |
| 利用者要求に基づく制御 | × | × | ○ |
| リソース変化への対応 | × | × | ○ |

る．これらの結果から，本提案のシステムは本研究の目的を達成するのに有効であることが示された．

### 5.5 関連研究

３次元共有空間は非常に計算機・ネットワーク負荷が大きいため，その QoS 制御のための様々な研究がなされてきている．

３次元空間のネットワーク共有を対象にしてはいないが，３次元処理の負荷低減を目的とした研究に［5］がある．この研究は，巨大な３次元空間に対しても全体のクオリティを下げることなく十分な速度で実時間描画することを目的とし，利用者の視点からの距離に応じて重要度を算出し，重要度の低いオブジェクトの品質を下げる，という手法が提案されている．

また，多人数参加型３次元共有空間を実現するための研究に［6］がある．この中で，３次元共有空間における利用者端末の差異の吸収を目的として，共有要求時に利用者端末の情報も同時にサーバに伝達し，サーバはその情報に基づき，送信する情報をコントロールする，という手法を提案している．

これらの研究と本研究を比較した結果を表 1 に示す．表 1 から，(1) 利用者が設定を行わなければならない負担の軽減，(2) 利用者ごとに異なる要求を反映した制御，(3) 動的にリソースが変化する環境への対応，という点に対し，従来の研究は十分に考慮しているとは言い難い．本研究では，これらの従来にない利点を持たせることができたといえる．

## 6 おわりに

本研究では，３次元共有空間アプリケーションの利用時の利用者の負荷軽減を目的とし，(1) 利用者エージェントによる利用者要求獲得，(2) オブジェクトをエージェント化し，エージェント間協調により制御を行う，といった方法により技術的課題を解決する手法を提案した．また，それらを実現するためのシステムを設計・実装した．実装したプロトタイプシステムを用い実験を行い，評価した結果，これまでにない柔軟な制御が行えることが確認され，本研究の目的の達成に非常に有効であることが示された．

現状では多数の利用者が参加したときのスケーラビリティやエージェント間の柔軟なネゴシエーション手法についての考慮が不足している．今後の課題としてこれらの問題を解決する手法を模索していく予定である．また，３次元共有空間における利用者要求の達成度を定量的に評価するための手法についても検討していく．

## 参考文献


[1] 箕浦大祐 山名岳志 正木茂樹 之瀬進 “ 多人数参加型３次元仮想空間における大規模人数表示方法 ” 電子情報通信学会論文誌 ’98/5 Vol.J81-D-II No.5

[2] 箕浦大祐 石橋聡 “ 千人規模の利用者のための３次元仮想空間コミュニケーション環境 ” 情報処理学会論文誌 Vol.42 No.11 Nov.2001

[3] Maja Matijasevic, Denis Gracanin, Kimon P. Valavanis, and Ignac Lovrek “A Framework for Maultiuser Distributed Virtual Environments” IEEE Transactions on systems, man, and cybanetics–Part B: cybanetics, Vol. 32, No. 4 August 2002

[4] 井芹威 堀真人 藤川和利 下條真司 宮原秀夫 “ 多人数参加型ネットワークアプリケーションの広域インターネット環境における利用実験 ” 信学技報 IN2000-121,MVE2000-91(2000-11)

[5] 玉田隆史　中村泰明 “ 多次元データ構造に基づく３次元仮想都市空間の管理と高速描画 ” 電子情報通信学会論文誌 ’95/8 Vol.J78-D-II No.8

[6] 中村暢達　里田浩三　平池龍一　根本啓次 “ インターネット対応の 3D マルチユーザシステム Ladakh ” 電子情報通信学会論文誌’98/5 Vol.J81-D-II No.5

[7] Takuo Suganuma, Shintaro Imai, Tetsuo Kinoshita, and Norio Shiratori, “A QoS Control Mechanism Using Knowledge-based Multiagent Framework”, エージェント合同シンポジウム (JAWS2002) 講演論文集, pp.38-48, 2002.11

[8] DASH - Distributed Agent System based on Hybrid architecture ! - http://www.agent-town.com/dash/