

# 仮想世界データベースシステム：VWDBにおける 共有型作業環境のためのトランザクション概念の導入

渡辺 知恵美<sup>†</sup> 大杉 あゆみ<sup>†</sup>  
佐藤 こず恵<sup>†</sup> 増 永 良 文<sup>††</sup>

我々は、VRシステム群のバックエンドにデータベースサーバを運動させたシステム、仮想世界データベースシステム(Virtual World Database system: VWDB)の設計と実装を進めている。VWDBでは、仮想世界で行われる操作をすべてVWDBトランザクションとして処理し、リアルタイムに仮想世界を管理する。このようなVWDBの機能は、1つの仮想世界を複数ユーザでアクセスする共有型作業環境で最大限に活用することができる。そこで、本論文ではVWDB上に構築した共有型作業環境のためのトランザクション概念について特に論じる。共有型作業環境では、オブジェクトの移動操作のように、(1)操作の途中経過を他のすべてのクライアントに公開する必要がある、(2)仮想世界における時間の流れが不可逆であり、操作の途中で失敗しても元に戻すことができない、(3)操作中のオブジェクトどうしが互いの操作に影響を与え合う場合がある、という共有型作業環境独特の特性がある。我々はこれらの特性を考慮に入れVWDBトランザクションという新しい概念を導入した。VWDBトランザクションはマルチレベルトランザクションを基に承認可能終了状態という概念を加えて定義されており、緩和されたACID特性を保証する。また、リアルタイム性を損ねることなくトランザクションを実行するために、バックエンドデータベースへの適切な間隔でのサブトランザクション発行予測方式を導入している。実験では、サブトランザクションが適切な間隔で発行されていることを確認したのち、VWDBトランザクションが適用可能な共有型作業環境の規模について実験を基に考察を行い、VWDBが数十人の小規模な共同設計環境で有効であることを明らかにしている。

## Introduction of a Transaction Concept for a Shared Work Environment of the VWDB: The Virtual World Database System

CHIEMI WATANABE,<sup>†</sup> AYUMI OSUGI,<sup>†</sup> KOZUE SATO<sup>†</sup>  
and YOSHIFUMI MASUNAGA<sup>††</sup>

The Virtual World DataBase system (VWDB) is currently under development at Ochanomizu University. This system is constructed by integrating a set of virtual reality systems and an object-oriented database system so that it supports database functions for cyberspace applications. Each operation issued by each VR client is managed as a new type of transaction called a VWDB transaction. The VWDB transaction is different from the traditional transaction concept in that it relaxed the ACID properties. The VWDB can support a shared work environment in which several users access the same virtual world. In this paper, we report on the VWDB transaction concept in detail for a shared work environment of the VWDB. We first formulated and analyzed operations in the shared work environment, and found their characteristics as follows: (1) It needs to exposure the uncommitted data to other users before the operation finishes, (2) the passage of the time in the virtual world is irreversible as it is so in the real world, and therefore it cannot be undone even if it fails, and (3) the objects under operations may affect each other. The VWDB transactions are introduced by taking into account these characteristics. The VWDB transaction is based on the multi-level transaction and is added the concept of the "acceptable termination state". In order to perform a transaction without spoiling real-time property, we introduced an algorithm which predicts an appropriate time interval between adjacent sub-transactions, and we verify that it works effectively. We evaluated the scale of the shared work environment which VWDB can support, and we found that the VWDB can support small-scale environment shared by dozens of users.

<sup>†</sup> お茶の水女子大学大学院人間文化研究科  
Graduate School of Humanities and Sciences, Ochanomizu University

<sup>††</sup> お茶の水女子大学理学部  
Faculty of Science, Ochanomizu University

### 1. はじめに

近年、フライトシミュレーションや仮想美術館など様々な分野でのバーチャルリアリティ(VR)システムの実用化にともない、VRシステムにおけるデー

データベース機能の必要性が認識されつつある。実際に、ウォークスルーにともなう検索機能<sup>8)</sup>や仮想世界オブジェクトの永続的管理機能<sup>5)</sup>など、VRシステムの一部の機能をデータベース機能によって管理したり機能拡張したりするシステムも開発されてきている。これらのシステムはアプリケーションの用途に合わせてデータベース機能の一部を利用しているが、データベースシステムはVRシステムをサポートするために必要と考えられる有用な機能を数多く備えており、これらのデータベース機能を完全に備え持つVRシステムを開発することは大変意義あることと考えられる(我々はこれを「バーチャルリアリティをデータベースシステムにする(Making Virtual Reality a Database System)」と呼ぶ)。我々はこのようなバーチャルリアリティをデータベースにしたシステムの実現を目指し、VRシステムとDBシステムをシステム連携したVirtual World Database System(VWDB)の開発を進めてきた<sup>10),11),18)</sup>。

VWDBは当初1台のVRシステムと1台のデータベースシステムをシステム連携してプロトタイプングが行われたが、最新のバージョンでは複数台のVRシステムと1台のデータベースシステムを統合してプロトタイプが構築されている。そのような構成のVWDBは、共有型作業環境(Shared Virtual Environment)の支援に有効に働くと考えられる。ここに、共有型作業環境とは、同じ世界に没入したユーザが、共有化された仮想世界オブジェクトを共同で編集できる環境であり、複数のVRシステムをネットワークで接続したネットワークVR(NVR)システム<sup>12)</sup>によって実現される。NVRシステムは軍事シミュレーションシステムであるSIMNET<sup>2)</sup>を先駆けとし、主に大規模シミュレーションやチャットシステムが数多く開発されてきた。しかし、近年ではNVRシステムによる共有型作業環境の研究が注目を集めており、たとえば香港大学とドイツのWeimar大学とで行われたVirtual Design Studioが非同期ではあるがVRシステムを用いたヘリポートの共同設計を実験的に行った<sup>15)</sup>ほか、Virtual Round Table<sup>1)</sup>、M3Dシステム<sup>4)</sup>などの建築物設計・レイアウトシステムなども開発されている。これらは、従来のシステムに比べて規模は小さく(2~数十人)、共有化された仮想世界オブジェクトの管理が重要となる。しかし、共有型作業環境におけるオブジェクト管理機能の研究はまだまだ初期段階にあり、加えて多数の作業の同時実行や障害時回復機能の実現を考えると、共有型作業環境の実現技術はまだ確立されているとはいえない。

そこで、我々は、VWDB上に、データベース機能を根底にして、共有している仮想世界の一貫性を保障し、かつシステムのリアルタイム性を維持した共有型作業環境を実現することを目指す。特に、本論文では、共有型作業環境のために新たに導入したVWDBトランザクションについて論じる。共有型作業環境における操作は、たとえばオブジェクトの移動操作のように、(1)操作の途中経過を他のすべてのクライアントに公開する必要がある、(2)仮想世界における時間の流れが不可逆であり、操作の途中で失敗しても元に戻すことができない、(3)操作中のオブジェクトどうしが互いの操作に影響を与え合う場合がある、という特徴を持つ。VWDBトランザクションはこれらの特徴を考慮に入れることによって導入された概念であり、マルチレベルトランザクションを基に承認可能終了状態という概念を加え、緩和されたACID特性を保障する。また、共有型作業環境でのリアルタイム性をできるだけ損ねることなくトランザクションを実行するために、適切な間隔でのサブトランザクション発行間隔予測方式を導入している。実験ではサブトランザクションが適切な間隔で発行されていることを確認したのち、VWDBトランザクションが適用可能な共有型作業環境の規模について実験を基に考察を行い、VWDBが2~数十人の規模の共有型作業環境で有効であることを明らかにしている。

## 2. 仮想世界データベースシステム：VWDB

VWDBの目的は「バーチャルリアリティをデータベースにしたシステムの実現であり、VWDBデータモデルの設計とプロトタイプシステムの実装を進めている。VWDBデータモデルでは、実世界および実世界のオブジェクトをVRシステムで表示される仮想世界および仮想世界オブジェクトとしてモデリングし、実世界のユーザを仮想世界に没入するアパタとして表現し、そして、ユーザのオブジェクトに対する行動や振舞いを仮想世界オブジェクトに対するメッセージととらえている。つまり、仮想世界および仮想世界オブジェクトのモデリングは基本的にオブジェクト指向分析設計手法に基づいて設計されている<sup>18)</sup>。

一方、プロトタイプシステムは、バックエンドのオブジェクト指向データベースシステムと複数台のVRクライアントとのサーバクライアント形式の構成で開発している(図1)。VRクライアントはJAVA言語用の仮想世界構築APIであるJAVA3Dを用いて実装し、バックエンドDBは商用のオブジェクト指向DBであるエクセロン社のObjectStoreを利用し、JAVA

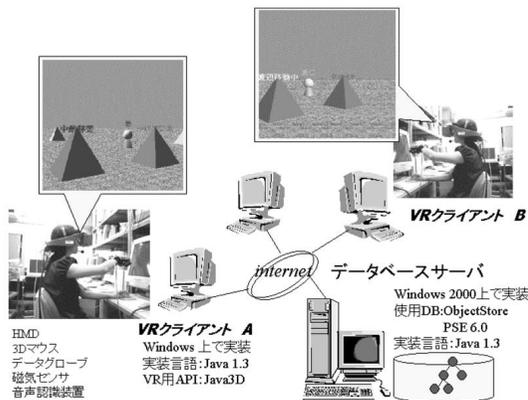


図 1 システム構成

Fig. 1 System architecture.

言語で実装している。

ユーザが VR クライアントを起動すると、VR クライアントはデータベースサーバにアクセスし、データベースサーバに格納された仮想世界を読み出して VR クライアントの画面に仮想世界を作り出す。ユーザは、HMD を被り、3D マウス、データグローブ、音声認識装置、磁気センサを身につけることで仮想世界の中に没入することができ、仮想世界の中をまるで実世界にいるかのようにウォークスルーしたり、オブジェクトを視認したりすることができる。また、ユーザは仮想世界のオブジェクトを手で動かしたり、声で指示を出してオブジェクトの操作を行ったりすることができる。このとき、仮想世界で行われた操作は直接データベースのトランザクションとしてデータベースへ送られ、データベースの更新が行われる。VWDB におけるデータベース操作言語はユーザの声や手によるマルチモーダルデータ操作言語であり、我々はこれを提案し、一部実装している<sup>11)</sup>。

VWDB はシステム構成として、データベースサーバ 1 台によるクライアント-サーバ構成を用いているが、この構成を用いる場合、サーバへのアクセス集中により全体のパフォーマンスを容易に低下させる場合があり、大規模 NVR アプリケーションへの適用は難しいといわれている<sup>9)</sup>。しかし、本論文で対象とする共有型作業環境は、(1) 同時利用人数が比較的少人数であること、(2) 大規模なシステムに比べて仮想世界オブジェクトの一貫性がより重視されるため比較的一貫性が確保しやすい構成が望まれること、(3) VR システムのためのトランザクション概念を導入するにあたり、フロントエンドの VR クライアントとバックエンド DB サーバというシンプルな構成が最も分かりやすいこと、などの理由から共有型作業環境を実現する

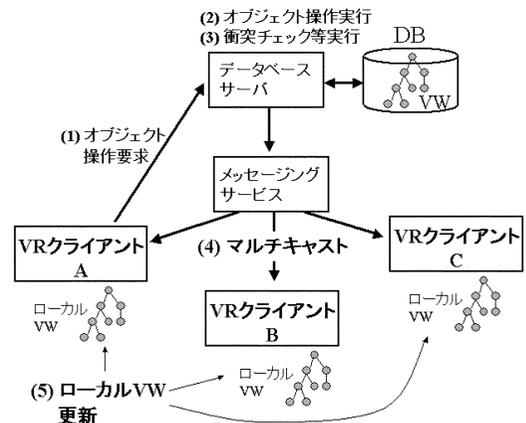


図 2 マルチキャストによる共有型作業環境の実現

Fig. 2 Realization of the shared work environment by multicasting.

ために最も適切な構成であると判断したうえでクライアント-サーバ構成を採用している。

### 3. VWDB における共有型作業環境の実現

VWDB では複数のクライアントが同じデータベースサーバに接続することによって共有型作業環境を実現する。これを実現するためのネットワークポロジとして、我々はデータベースサーバを中心としたマルチキャストによる集中管理型接続方式<sup>3)</sup>を採用した。クライアントとサーバ間の通信プログラムは HORB<sup>7)</sup>を用い、マルチキャストは HORB extension メッセージングサービスを用いた。

VWDB で VR クライアントのアプリケーションを起動すると、VR クライアントはデータベース中にある仮想世界のコピーをローカルに作成し、VR システム上に表示する。そして、各 VR クライアントの持つ仮想世界とデータベース中に格納されている仮想世界との同期をとることですべてのクライアントが「同じ仮想世界を共有している」感覚で VR システムを利用することができる。これが共有型作業環境を実現する仕組みである。

図 2 に VR クライアント A が更新を行ったときの同期処理の流れを示す。VR クライアント A が仮想世界オブジェクトの操作を実行すると、同じ操作がデータベースサーバに送られ (図 2(1))、データベースサーバはこれを実行する (図 2(2))。このとき仮想世界におけるオブジェクトどうしの衝突チェックや物理法則などを実行し (図 2(3))、メッセージングサービスによって他 VR クライアントにマルチキャストする (図 2(4))。マルチキャストされたデータは、各

VR クライアントがローカルに持つ仮想世界を更新する(図 2(5)). これにより, データベース中にある仮想世界と各 VR クライアントの仮想世界との同期をとることができる.

#### 4. 共有型作業環境における操作の体系化

次章で共有型作業環境にトランザクション概念を導入するために, 本章で共有型作業環境における操作の体系化を行う.

##### 4.1 共有型作業環境における操作とは

VR クライアントで行われる操作は, 以下の 2 種類に分類することができる.

- (1) VR クライアントのみで完結する操作
- (2) VR クライアントで実行した操作がデータベースに反映され, それがすべての VR クライアントにキャストされる操作

(1) に該当する操作は仮想世界内のウォークスルーや, オブジェクトの一時的な表示変更(自分の触っているオブジェクトの色を黄色く強調表示する)など, オブジェクトそのものを変更しない操作であり, (2) に該当する操作は, オブジェクトの移動や生成, 消去などオブジェクトを直接変更する操作である.

本章で扱う「共有型作業環境における操作」とは, このうちの (2) に該当する操作である. これは, 3 章で述べたようにすべての VR クライアントおよびデータベースが同じ状態の仮想世界を持つことによって仮想世界を共有しているように見せているため, 操作を実行する場合もすべてのクライアントとデータベースサーバで同じように実行することによって「共有型作業環境で操作を行った」ように見せているからである.

##### 4.2 操作体系

本節では共有型作業環境における操作の体系化を試み, 以下のようにまとめた. 例として, A という箱の上に新たな箱を 1 つ生成する操作(図 3(a))と, 箱をつかんで移動する操作(図 3(b))で説明する.

###### ● 基本操作 (primitive operation)

仮想世界における原始的な操作を基本操作と呼ぶ. これには, 仮想世界オブジェクトの生成, 消去, 属性の更新, 問合せ, 仮想世界オブジェクトの振舞いの実行などがある.

###### ● グループ操作 (group operation)

複数の基本操作の順序列をグループ操作としてまとめることができる. 図 3(a) では, 「名前が “A” である箱の上に新たな箱を生成する」という操作を「名前が “A” である箱は?」という問合せ操作 ( $p_1$ ), 「箱を生成」という生成操作 ( $p_2$ ), 「A の上

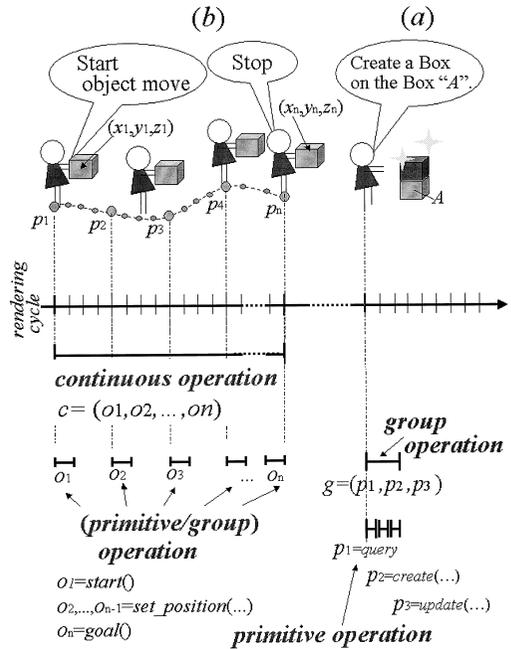


図 3 共有型作業環境における操作

Fig. 3 Operations in the shared work environment.

に置く」( $p_3$ )という位置更新操作によるグループ操作 ( $g = (p_1, p_2, p_3)$ ) によって実現している. なお, 共有型作業環境における 1 回の操作を「VR クライアントで実行した操作がデータベースに反映されるまで」としているため, 基本操作およびグループ操作はどちらも 1 回の操作である. なぜならば, 図 3(a) のグループ操作は  $p_1, p_2, p_3$  をすべてデータベースで実行してから, その結果を他クライアントにマルチキャストするからである.

###### ● 連続操作 (continuous operation)

基本操作とグループ操作は, どちらも 1 回の操作である. これに対して, 図 3(b) のような箱をつかんで移動するという操作は, 位置更新 ( $o_i = set\_position(x_i, y_i, z_i) (1 \leq i \leq n)$ ) という基本操作をある間隔で複数回実行することによって実現される. このように複数回の基本操作またはグループ操作を実行することによってたとえば移動のような意味のある一連の操作が実現されるとき, これを連続操作 (continuous operation) としてまとめることができる. 連続操作には, 移動や回転, 大きさの変更などがある. 連続操作では, その構成要素となる基本操作またはグループ操作がどれくらいの間隔で実行されるかが重要となる. VR クライアントがローカルに持つ仮想

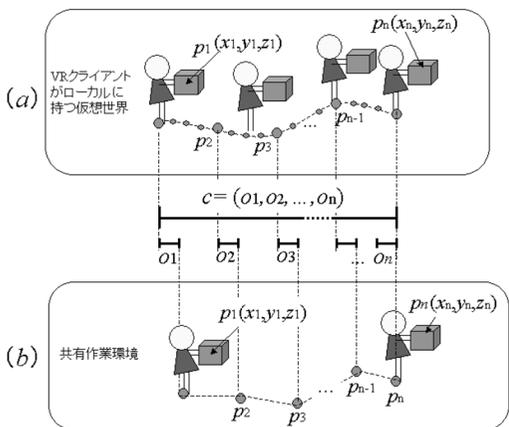


図4 VRクライアントがローカルに持つ仮想世界と共有型作業環境の違い

Fig. 4 Difference between a local virtual world in a VR client and shared work environment.

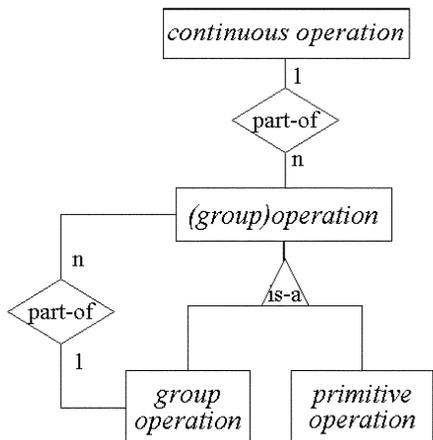


図5 操作の階層間の関係

Fig. 5 Relationship among primitive/group/continuous operation.

世界を図4(a)に、共有型作業環境を図4(b)に示す。VRクライアントがローカルに持つ仮想世界では、約30分の1秒の描画サイクル単位で位置更新が行われスムーズな移動が可能になっている。しかし、これをすべて共有型作業環境で実現するのは難しい(1回の基本操作にかかる時間が描画サイクルよりも長くなる場合が多い)ため、図4(b)のように適当な間隔(図では4回に1回)で基本操作またはグループ操作を実行し、VRクライアントで行われた移動操作の過程を近似する。この間隔が短いほど共有型作業環境の精度が高くなる。

これらの操作における関係を図5に示す。

表1 各階層における操作の特性  
Table 1 Characteristics of the primitive/group/continuous operation.

操作レベル	原子性	分離性
連続操作	No	No
グループ操作	Yes	Yes
基本操作	Yes	Yes

### 4.3 操作の特性

VWDBのトランザクション概念は、次節で見るように前節で体系化した基本操作、グループ操作および連続操作を基に定義される。そこで、基本操作、グループ操作および連続操作の各々がトランザクションで知られているACID特性、特に原子性と分離性を保証する必要があるかについて考察し、表1にまとめた。

基本操作およびグループ操作は原子性と分離性を保証する必要があるが、連続操作はこれらを保証する必要はない。これは仮想世界に流れる時間の概念や共有型作業環境の特徴による。以下、連続操作における原子性と分離性に関する特性について述べる。

- 連続操作における原子性

連続操作が途中でアボートされたとき、厳密な原子性を持つ場合には連続操作の始まる直前の状態まで引き戻されるが、仮想世界の場合このような現象は極力避けるべきである。なぜならば、仮想世界に流れている時間は実世界と同様に不可逆であり、仮想世界で行われている操作は瞬間瞬間にコミットされ引き戻すことができないからである。そこで、連続操作の途中でアボートされた場合、その時点までコミットされた基本操作またはグループ操作までを有効とし、その時点まで連続操作を終了したと見なす方法が望ましい。たとえば、図3の移動操作  $c$  が  $o_3$  の途中でアボートされた場合は、 $c$  は  $o_3$  をロールバックしたのち、 $o_2$  までで完了したと見なして終了する。

- 連続操作における分離性

VWDBでは、共有型作業環境を実現するために、3章で述べたような同期方法によって基本操作およびグループ操作のコミット時すべてのVRクライアントにその結果をマルチキャストする。そのため、各基本操作およびグループ操作をコミットした時点で、途中経過が他のトランザクションに公表される。このような場合の連続操作における分離性について述べるために、VRクライアントAとVRクライアントBでユーザがそれぞれオブジェクト  $O_A$  とオブジェクト  $O_B$  を同時に移動させている場合を考える。このとき、各々の

ユーザが動かすオブジェクトが連続操作中に仮想世界にあるすべてのオブジェクトに影響を与えない(または与えられない)限りは分離性は保証される。しかし、仮想環境ではオブジェクトどうしの衝突が起こったり、同じオブジェクトを複数のユーザが同時に押したりするというような操作が行われる。このような操作の場合、互いの連続操作の途中経過が影響を及ぼしあうため分離性は保証できない。

### 5. VWDB トランザクション

#### 5.1 VWDB トランザクションの定義

一般にトランザクションとはアプリケーションレベルの仕事の単位である。VWDBでは4章、図5操作の階層間の関係で規定した基本操作、グループ操作、連続操作を共有型作業環境における意味のある「仕事の単位(a unit of work)」であると考え、トランザクションとして定義する。つまり、VWDB トランザクションには以下の3種類のトランザクションがある。

- (1) 基本操作トランザクション
- (2) グループ操作トランザクション
- (3) 連続操作トランザクション

まず、基本操作トランザクションおよびグループ操作トランザクションは、ACID 特性の保証を要求することと操作が終了するまでの時間が短いことから、フラットトランザクションが適用される。しかし、連続操作トランザクションは表1および4.2節で論じた操作特性や、操作が終了するまでの長時間を必要とすること、さらに図5に示す階層を持つことから、フラットトランザクションを適用することはできない。

そこで我々はVWDB トランザクションのための新たなトランザクションモデルを導入した。これはマルチレベルトランザクション<sup>19)</sup>を基本モデルとしたものであるが、前章で述べた連続操作の原子性を考慮して、承認可能終了状態という概念を取り入れた結果、従来にない新しいトランザクションの定義となっている。ここに、承認可能終了状態は、ワークフローのためのトランザクション拡張の1つとして導入された<sup>14)</sup>概念である。そこでは、トランザクションの状態と状態遷移を考え、承認可能終了状態を定めておく。承認可能終了状態には、コミット承認可能終了状態(committed acceptable termination state)とアボート承認可能終了状態(aborted acceptable termination state)があり、トランザクションの実行途中でアボートされた場合には必ずアボート承認可能終了状態でトランザクションを終了する。

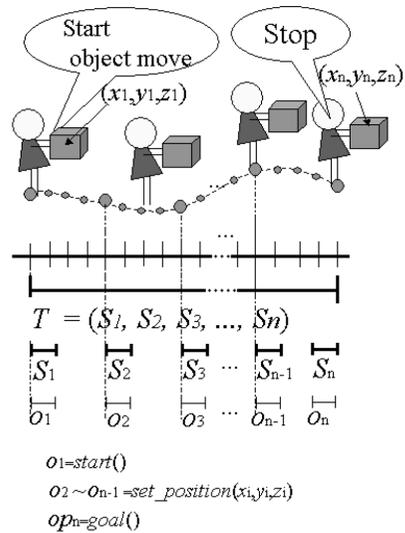


図6 移動トランザクション  
Fig. 6 Object move transaction.

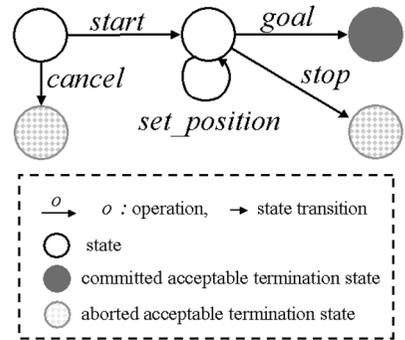


図7 移動トランザクションにおける状態遷移  
Fig. 7 State transition of object move.

図6は図3(b)の移動トランザクションである。移動の開始から終了までをトップトランザクション  $T$  とし、 $T$  を構成する基本操作およびグループ操作トランザクションによるサブトランザクションの系列を  $T = (S_1, \dots, S_n)$  とする。

まず移動開始時にサブトランザクション  $S_1$  で  $start()$  が実行されたのち、サブトランザクション  $S_i$  ( $2 \leq i \leq n - 1$ ) で  $o_i = set\_position(x_i, y_i, z_i)$  が実行される。そしてユーザがオブジェクトを目的地に運びオブジェクトを手離すと、サブトランザクション  $S_n$  で  $o_n = goal()$  を実行し  $T$  をコミットする。

この移動トランザクションにおける状態遷移を図7に示す。図6のように無事に目的地にオブジェクトを移動できた場合は、トランザクションはコミット承認可能終了状態となる。しかし途中でアボートされたと

表 2 階層を持つ各トランザクションにおける分離性と原子性の自由度

Table 2 Isolation and atomicity for the nested/multi-level transaction and VWDB transaction.

トランザクションモデル	分離性	原子性
入れ子型トランザクション	厳密	厳密
マルチレベルトランザクション	緩和	厳密
VWDBトランザクション	緩和	緩和

きは *stop()* 操作によってその場にストップさせてアポート承認可能終了状態とし、連続操作を終了する。

従来、VWDBトランザクションのように階層を持ち、長時間の操作に対して適用できるトランザクションには、入れ子型トランザクション<sup>13)</sup>とマルチレベルトランザクションが提案されている。そこで、入れ子型トランザクション、マルチレベルトランザクションとVWDBトランザクションが要求する原子性および分離性の厳密性と緩和性(=自由度)の比較を行い、その結果を表2に示す。

入れ子型トランザクションはトランザクションを入れ子構造にすることによってサブトランザクションの並列処理を実現する。しかし、サブトランザクションをコミットしても、トップトランザクションがコミットされるまでは外部に公表されない。また、トップトランザクションがアポートされると、サブトランザクションのコミット/アポートにかかわらずすべてのサブトランザクションがロールバックされる。

この点、マルチレベルトランザクションはサブトランザクションの早期コミットを許しており、サブトランザクションのコミット後、外部に公表することができるという点で分離性が緩和されている。VWDBトランザクションでは、仮想世界の同期をとるためにサブトランザクションの結果を公表する必要があり、マルチレベルトランザクションはこの要求を満たす。しかし、マルチレベルトランザクションもネストトランザクションと同様原子性は厳密に守られており、トップトランザクションがアポートされると、これまでのサブトランザクションがしたことと意味的に逆の処理を行うことのできる補償トランザクションを実行することによって、すべての更新を無効にしてしまう。VWDBトランザクションは補償トランザクションの実行を要求しないので、その意味で緩和されている。

5.2 VWDBトランザクションにおける同時実行制御

連続操作トランザクションでは、トップトランザク

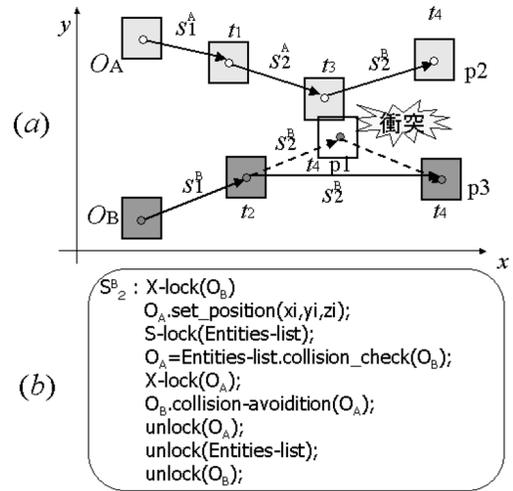


図 8 (a) 移動中の2オブジェクト  $O_A, O_B$  が衝突する例 (b) サブトランザクション  $S_2^B$

Fig. 8 (a) Example of collision two objects which are on moving, (b) Sub-transaction  $S_2^B$ .

ションとサブトランザクションによる2段階の階層を持つ。サブトランザクションは従来のトランザクションと同程度の分離性を保障する必要がある。しかし、トップトランザクションに関しては、4.2節でも述べたように、操作中のオブジェクトどうしが衝突する場合など、お互いの操作に影響を及ぼしあう場合が多いため、分離性を緩和する必要がある。

図8(a)はVRクライアントAのユーザが移動トランザクション  $T_A$  を、VRクライアントBのユーザが移動トランザクション  $T_B$  を同時に実行している様子を示している。 $T_A$  は時刻  $t_1$  と  $t_3$  ( $t_1 < t_3$ ) にサブトランザクション  $S_1^A, S_2^A$  を実行し、 $T_B$  は時刻  $t_2$  と  $t_4$  ( $t_2 < t_4$ ) にサブトランザクション  $S_1^B, S_2^B$  を実行し、その結果、時刻  $t_4$  に  $S_2^B$  を実行した際、地点  $p_1$  にて衝突が起こっている。このとき、 $S_2^B$  は衝突の結果としてオブジェクト  $O_B$  を地点  $p_3$  に移動させると同時に  $O_A$  を地点  $p_2$  まで動かさなければならないが、 $T_A$  によって  $O_A$  に専用ロックがかけられている場合、衝突の結果  $O_A$  を移動させることができない。

そこで、トップトランザクションでは2相ロックを行わず、各サブトランザクションをコミットすることにサブトランザクションでかけられたロックを解除することとした。つまり、たとえば  $T_A$  における  $S_i^A$  コミット時から  $S_{i+1}^A$  開始時の間はオブジェクト  $O_A$  を含めたすべてのロックが解除され、 $S_2^B$  が  $S_2^A$  終了時と  $S_3^A$  開始時の間に要求されたならば、 $O_A, O_B$  衝突時に  $O_A$  を移動させることができる。

また、移動操作トランザクション  $T$  におけるサブトランザクションの  $S_2^B$  では図 8(b) に示されるように、衝突チェックのために  $Entities - list.collusion\_check()$  を実行するが、この衝突チェックにおいて現在の実装では  $Entities - list$  に登録されているすべての仮想世界オブジェクトに対してその可能性を検証している．そのために、共有ロック  $S - lock$  ( $Entities - list$ ) が実行される．一方、サブトランザクション実行のためには、必要なオブジェクト（この例では  $O_B$ ）に専有ロックがかけられているので、結果として複数ユーザが同時に移動トランザクションを発行している場合には、複数のサブトランザクションが同時に実行されることはなく、各異動トランザクションのサブトランザクションが発行された順番に逐次処理されていく．次節ではこの点を考慮したうえで最小のサブトランザクション発行間隔を求める．

5.3 サブトランザクションの発行間隔

連続操作のトランザクションの場合、ある間隔でサブトランザクションを発行し基本操作またはグループ操作を実行する．このサブトランザクションの間隔をどれだけとるかを考える必要がある．

ここで、トランザクション  $T = (S_1, \dots, S_n)$  におけるサブトランザクション  $S_i$  と  $S_{i+1}$  の発行間隔  $d(S_i, S_{i+1})$  を、 $S_i$  がコミットしてから  $S_{i+1}$  の開始が要求されるまでの間隔という意味とする．たとえば図 9 の移動トランザクション  $T_A$  でサブトランザクション  $S_i^A$  を開始してから  $S_{i+1}^A$  を開始するまでの時間を  $l(S_i^A, S_{i+1}^A)$  とし、 $S_i^A$  の処理時間を  $l(S_i^A)$  とする．このときのサブトランザクション発行間隔  $d(S_i^A, S_{i+1}^A)$  は

$$d(S_i^A, S_{i+1}^A) = l(S_i^A, S_{i+1}^A) - l(S_i^A)$$

である． $l(S_i^A, S_{i+1}^A)$  の間隔が短いほど共有型作業環境の精度が高くなるが、 $l(S_i^A)$  よりも短くしても結局  $S_i^A$  がリソースを解放するまで待機しなければならないため意味がない．そこで、 $d(S_i^A, S_{i+1}^A)$  の間隔をできるだけ狭くする必要はある．共有型作業環境では、図 9 の  $T_B$  のサブトランザクション  $S_1^B$  と  $S_2^B$  の間隔  $d(S_1^B, S_2^B)$  のように間隔を短くとりすぎると、他のクライアントにより同時に実行されているトランザクションのサブトランザクション（この例の場合は  $S_2^A$ ）によってかけられているロックの解除を待機している状態（ロック待ち状態）が発生し、かえって同期精度が悪くなってしまふ．逆に  $S_2^A$  と  $S_3^A$  の間隔  $d(S_2^A, S_3^A)$  のように間隔を長くとりすぎると、ロックが解除されているにもかかわらずサブトランザクションが実行されない状態（過剰待機状態）が発生してし

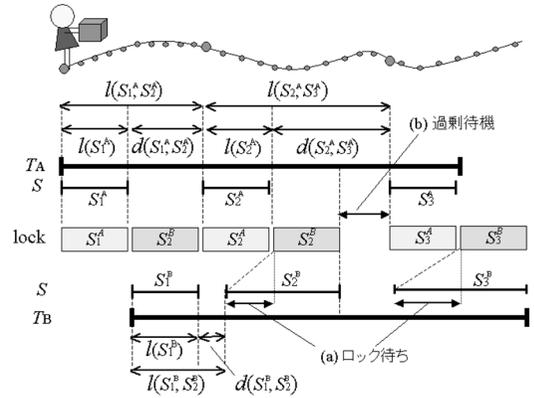


図 9 サブトランザクション間隔

Fig. 9 Interval between adjacent sub-transactions.

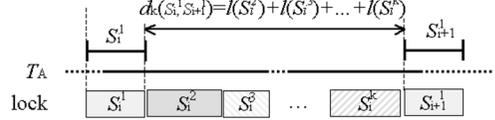


図 10  $S_i^j$  から  $S_{i+1}^j$  までのサブトランザクション間隔

Fig. 10 Interval between  $S_i^j$  and  $S_{i+1}^j$ .

まい、これも効率 (= 精度) が悪い．

そこで VWDB トランザクションにおいてロック待ちおよび過剰待機時間を最小限に抑えるためのサブトランザクションの発行間隔の予測方法を導入する．

前節で導入したロック方式から、たとえば  $k$  個のトランザクション  $T_1, T_2, \dots, T_k$  並行に実行されている場合、各々のサブトランザクションが  $S_1^i, S_2^i, \dots, S_k^i$  ( $1 \leq i \leq n$ ) が順番に実行されることになると考えられる（図 10）．そこで、 $S_i^i$  を終了してから  $S_{i+1}^i$  を開始するまでの適切な間隔  $d(S_i^i, S_{i+1}^i)$  を、並行に実行されているトランザクション  $T_2, \dots, T_k$  におけるサブトランザクションの平均処理時間の合計値によって予測することとした．

つまり、トランザクション  $T_i$  におけるサブトランザクション  $S_j^i$  がコミットされた後の  $S_j^i$  から  $S_{j+1}^i$  までの平均予測間隔  $\tilde{d}_k(S_j^i, S_{j+1}^i)$  を以下の式により求める．

$$\tilde{d}(S_j^i, S_{j+1}^i) = \sum_{m=1}^{i-1} Avg(T_m) + \sum_{m=i+1}^k Avg(T_m)$$

ここに、 $k$  は並行して実行されているトランザクションの総数で、 $Avg(T_m)$  は  $S_j^i$  がコミットされるまでの  $T_m$  のサブトランザクション  $T^m = (S_1^m, \dots, S_n^m)$  の平均処理時間であり、次の式で求められる．

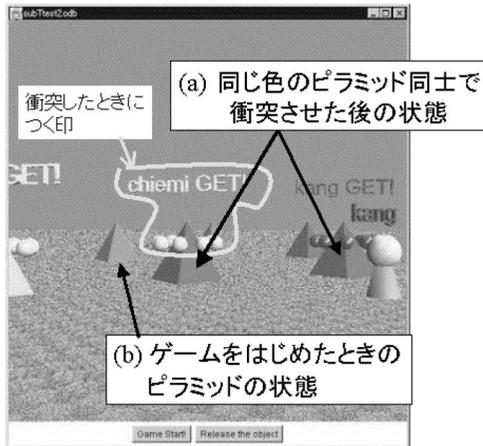


図 11 アプリケーションのスクリーンイメージ

Fig. 11 Screen image of the VWDB application.

$$Avg(T_m) = \frac{\sum_{s=1}^n l(S_s^m)}{n}$$

ここに、 $n$  はトプトランザクション  $T_m$  を開始してから  $S_j^i$  がコミットされるまでの間に  $T_m$  でコミットされたサブトランザクションの総数である。

#### 5.4 サブトランザクション発行間隔の上限

前節で求めたサブトランザクションの発行間隔は、同時に実行されているトランザクション数  $k$  に依存する。そのため、同時に作業を行っている人数が増えるほどサブトランザクション発行間隔が広がり、リアルタイム性が失われることがある。そこで、VWDBではサブトランザクション発行間隔の長さに上限を設け、トランザクション  $T$  を要求したときに、サブトランザクション発行間隔が上限を超える場合はそのトランザクションをアボートすることとした。なお、この方法をとることによってVWDBにおける共有型作業環境の同時使用人数に制限が出てくる。そこで次章では、VWDBのプロトタイプシステム上にアプリケーションを実装し、今回導入したトランザクション概念でどの規模の共有型作業環境が実現できるかを検証する。

## 6. VWDBトランザクションの評価

### 6.1 評価用アプリケーション

VWDBトランザクションを導入した場合のVWDBの共有型作業環境支援能力を評価するために、これまで構築してきたVWDBプロトタイプシステム上に次のアプリケーションを考案し、実装した。

図 11 はこのアプリケーションのスクリーンイメージである。仮想世界には図 12 に示すように 4 タイプのピラミッド（色で区別している）が 2 個ずつ計 8 個、

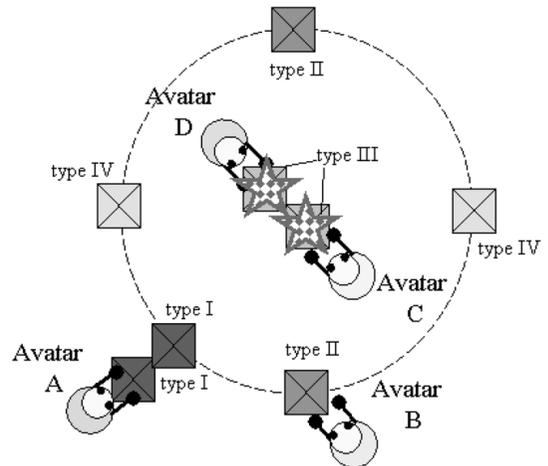


図 12 アプリケーション利用の様子を上から見た図

Fig. 12 Bird-eye's view of the virtual world of the application.

円状に並べられている。ユーザはこれらのピラミッドをつかんで移動し、図 12 のアバタ C と D のように、同じタイプのピラミッドを衝突させると、ピラミッドに図 11 (a) のような印がつけられる。ただし、衝突させる際オブジェクトは互いに誰かによって移動中ではなければならない。図 12 のアバタ A のように移動中でないオブジェクトにぶつかった場合、印はつけられず互いに跳ね返されるだけとなる。また、違うタイプのピラミッドをぶつかった場合も同様である。このように同じタイプのピラミッドを衝突させてすべてのピラミッドに印がつけられたら作業完了となる。このアプリケーションは、共有型作業環境にいるユーザが互いにどのオブジェクトを移動しているかをリアルタイムに知ることができなければ成立しない。そのため、移動トランザクションにおけるサブトランザクション発行間隔が重要となる。

### 6.2 評価

このアプリケーションを用いて、今回提案したVWDBトランザクションに対する以下の 2 つの評価を行った。

- サブトランザクション発行間隔予測の正確性  
5.3 節で提案したサブトランザクション発行間隔予測方法によって、実際に待ち時間および過剰待機時間を最小限に抑えることができていないか評価する。
- VWDB で実現可能な共有型作業環境の規模  
5.4 節で述べたように、VWDBトランザクションの性質上VWDBにおける共有型作業環境での同時使用人数に上限がある。そこで、このアプリケーションにおける最大同時使用人数を求め、こ

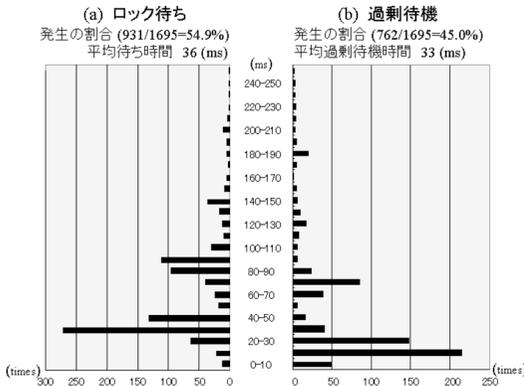


図 13 ロック待ち時間および過剰待機時間の分布

Fig. 13 Distribution of lock-waiting time and over-waiting time.

れを基に VWDB で実現可能な共有型作業環境の規模について考察する。

なお、これらの評価では、データベースサーバに PC ( PentiumIII 1,690 MHz, Windows2000 ) を、各クライアントには PC ( PentiumIII 700 MHz, Windows 2000 : すべて同スペック ) を利用し、Ethernet ( 100 BASE-T ) で接続した。

評価 1 : サブランザクション発行間隔予測の正確性  
5.3 節で定義したサブランザクション発行間隔予測の正確性を検証するため、クライアント 4 台でアプリケーションを 4 分間実行し、移動ランザクションにおける各サブランザクションで発生したロック待ち時間 ( 図 9 (a) ) および過剰待機時間 ( 図 9 (b) ) を求めた。

図 13 (a) はロック待ち時間、図 13 (b) は過剰待機時間の分布である。4 台のクライアントによる作業で実行されたサブランザクションは 1965 回、うちロック待ち状態が発生した回数は 931 回 ( 54.9% )、過剰待機状態が発生した回数は 762 回 ( 45.0% )、前のサブランザクションコミット直後に次のサブランザクションが実行された回数は 2 回 ( 0.1% ) であった。ロック待ち時間の平均は 36 ミリ秒、過剰待機時間の平均は 33 ミリ秒で、全体の約 7 割が 80 ミリ秒以内でロック待ちまたは過剰待機を終了させており、サブランザクション発行間隔の予測によってロック待ちおよび過剰待機を最小限に抑えることができたといえる。

評価 2 : VWDB で実現可能な共有型作業環境の規模

VWDB で実現可能な共有型作業環境の規模について考察する目安として、今回作成したアプリケーションの最大同時使用人数を求め、これを基に実現可能な

共有型作業環境について考察を行う。

アプリケーションの最大同時使用人数を求めるには、実際に何十台ものクライアントマシンを用いて実験すればよいが、今回の実験に用いることができるクライアントマシンは最大 4 台しかなく、このような実験を行うことができなかった。そこで、5.3 節で定義したサブランザクション発行間隔予測を基に最大同時使用人数を求める。まず、このアプリケーションで利用される連続操作は移動操作しかないことから、すべてのランザクション  $T_m$  ( $1 \leq m \leq k$ ) におけるサブランザクションの処理時間はほぼ同じであるとした。このことから、サブランザクション最大間隔を  $d(k_{max})$ 、このときの使用人数を  $k_{max}$ 、すべてのサブランザクションの平均処理時間を  $Avg'(l(S))$  とすると、以下の関係式が求められる。

$$d(k_{max}) = Avg'(l(S)) \times (k_{max} - 1) \quad (1)$$

この式を基に、以下の手順で最大同時使用人数  $k_{max}$  を求めることとした。

- Step1 : 5.1 節で紹介したアプリケーションを実行し、サブランザクション処理時間平均  $Avg'(l(S))$  を求める。
- Step2 : サブランザクション発行間隔を一定の値  $d$  に固定してアプリケーションを実行する。 $d$  の値を徐々に長くして何回か実験を行い、このアプリケーションが続行不可能となった時点での値  $d(k_{max})$  を求める。
- Step3 : 式 (1) より最大使用人数  $k_{max}$  を求める。

Step 1 : サブランザクション処理時間平均同時使用人数 4 人で 5.1 節で述べたアプリケーションを 4 分間実行し、そのときのサブランザクション平均処理時間  $Avg'(l(S))$  を求めたところ 114 ミリ秒となった。

Step 2 : サブランザクション最長間隔

サブランザクションの間隔を一定にして、同時使用人数 4 人で作業を行い、作業が完了するまでの時間を計測した。サブランザクションの間隔は 0.5 秒、1 秒、3 秒、5 秒、10 秒、20 秒とした。また参加者 4 人に「他のピラミッドとうまく衝突させることができたか」について表 3 に示す 4 段階評価でアンケートをとった。そして、これらの結果を基に「アプリケーションが成立しない」と判断する基準を「サブランザクションの間隔を 0.5 秒としたときの作業時間を基準にして、作業時間がその 5 倍を超えたとき、あるいは、各参加者による 4 段階評価の結果の平均が 2 未満になった場合」であるとした。

なお、作業時間には慣れの問題が影響すると予想さ

表3 主観評価基準  
Table 3 Subjectivity evaluation basis.

4	容易にピラミッドをぶつけることができた
3	多少戸惑うことはあったが、比較的容易にピラミッドをぶつけることができた
2	ぶつけるのが難しいと感じた
1	ぶつけるのが大変難しく、ゲームを進めることができなかった

表4 各サブトランザクション発行間隔における作業時間と参加者の評価

Table 4 Working time and evaluation of each intervals between adjacent sub-transactions.

サブトランザクション 発行間隔(秒)	0.5	1.0	3.0	5.0	10.0	20.0
作業時間(秒)	60	70	210	285	255	285
参加者Aの評価	4	4	3	2	2	2
参加者Bの評価	3	4	4	4	2	3
参加者Cの評価	3	4	4	4	3	3
参加者Dの評価	3	4	4	3	2	2
評価平均	3.3	4	3.8	3.5	2.3	2.5

れたため、参加者4人ともに3回ほど練習をしたのうち、サブトランザクションの間隔をあらかじめ告げずに実験を行った。

この実験の結果を表4にまとめる。

作業時間で最も速かったのは0.5秒間隔であり、最も遅かったのは5秒間隔と20秒間隔であった。また、1.0秒間隔での作業時間と3.0秒間隔での作業時間、3.0秒間隔での作業時間と5.0秒間隔での作業時間に大きな開きが見られた。しかし、5.0秒以上は作業時間に余り違いが見られなかった。参加者の評価では1.0秒間隔が最も評価が高く、10.0秒、20.0秒の評価が低かった。

今回の実験では、「アプリケーションが成立しない」と判断される結果は得られなかったが、5.0秒以上間隔がある場合は作業時間が250秒以上と基準のほぼ5倍近くかかっており、参加者の評価も2点台となっていることから、このアプリケーションのサブトランザクションの最長間隔は5.0秒であるとした。

なお、サブトランザクション発行間隔が0.5秒に参加者の評価が下がっているのは、クライアントにおける表示の更新にかかる計算の負荷が大きくなり、逆にリアルタイム性が損なわれるという現象が起きたためである。これは、実験で用いられたプロトタイプシステムで、各クライアントにおけるサブトランザクションの結果を4人のクライアントすべてにマルチキャストしたために、各クライアントは約0.125秒ごとにマルチキャストされたデータを受け取りオブジェクトを再描画することから、データベースサーバよりも負荷

がかかった結果起きたものと思われる。今後クライアントアプリケーションに対して、トランザクションの結果を、仮想世界の近隣にいるユーザにのみキャストするなどの改良を検討している。

### Step 3: 最大同時使用人数の導出と考察

Step1で得られた平均処理時間  $Avg'(l(S)) = 114$  (ミリ秒)とStep2で得られたサブトランザクション最長間隔  $d(k_{max}) = 5.0$  (秒)を式(1)に代入すると、

$$5000 = 114 \times (k_{max} - 1)$$

となり、このアプリケーションにおける最大同時使用人数  $k_{max} = 42.9$  (人)となった。今回の実験で用いられたアプリケーションは単純な操作のみであるため、実際に運用されるアプリケーションを考えるとVWDBで実現できる共有型作業環境は2人~30人規模であるといえる。これは小規模なグループによる共同設計システムを実現するのに適切な人数であるといえる。また、文献6)などのように100~1,000人以上の大規模な環境が必要なアプリケーションもあるが、100人を超える人数で共同で作業を行う場合、やはり30人から50人規模のグループに分散するであろう。この場合、グループごとの作業環境を用意しデータベースを分散させる方法によってVWDBを適用することを検討している。

## 7. おわりに

本論文では、仮想世界データベース上で構築した共有型作業環境と、共有型作業環境のためのトランザクション概念について述べた。まず、共有型作業環境における操作を体系化し、それに基づいてトランザクション概念を導入した。移動処理などの長時間にわたって実行される操作のためのトランザクションを定義するために、早期コミットを許すマルチレベルトランザクションを基本とし、これに対して移動中にアポートされたときに移動操作全体がアポートされないようにするために容認可能終了状態という概念を加えた。さらに、仮想世界でのオブジェクトの衝突や複数ユーザによる共同操作を実現するためのロック方式を提案し、この方式が適用できる共有型作業環境の規模について検証を行った。

今回は共有型作業環境を対象としたVWDBトランザクションを定義したが、この共有型作業環境を拡張し、ユーザ同士のコミュニケーション機能を充実させたものが協調作業環境であるといえる。今後は今回構築したVWDBを拡張し、協調作業環境支援システムとしての機能を充実させることを検討している。

## 参考文献

- 1) Broll, W., Meier, E. and Schardt, T.: The Virtual Round Table—A Collaborative Multi-User Environment, *Proc. ACM Collaborative Virtual Environments 2000 (CVE2000)*, pp.39–46 (2000).
- 2) Calvin, J., Dicken, A., Gaines, B., Metzger, P. and Miller, D.: The SIMNET Virtual World Architecture, *Proc. IEEE Virtual Reality Annual Internal Symposium (VRAIS'93)*, pp.450–455 (1993).
- 3) Funkhouser, T.A.: Network Topologies for Scalable Multi-User Virtual Environment, *Proc. IEEE Virtual Reality Annual Internal Symposium (VRAIS'96)*, pp.222–228 (1996).
- 4) Galli, R. and Luo Y.: Mu3D: A Causal Consistency Protocol for a Collaborative VRML Editor, *Proc. 5th Symposium on the Virtual Reality Modeling Language (VRML2000)*, pp.53–62 (2000).
- 5) Gausemeier, J., Krumm, H., Molt, T., Gbbesmeyer, P. and Gehrman, P.: A Database Driven Server for an Internet Based Plant Layout Presentation, *Proc. 5th Symposium on the Virtual Reality Modeling Language (VRML2000)*, pp.17–22 (2000).
- 6) Greenhalgh, C. and Benford, S: MASSIVE: A Distributed Virtual Reality System Incorporating Spatial Trading, *Proc. IEEE 15th International Conference on Distributed Computing Systems (DCS'95)*, pp.27–34 (1995).
- 7) HORB ホームページ . <http://www.horb.org>
- 8) Kamiura, M., Oiso, H., Tajima, K. and Tanaka, K.: Spatial Views and LOD-based Access Control in VRML-object Database, *Worldwide Computing and Its Application*, pp.210–225, LNCS 1274, Springer (1997).
- 9) 清川, 竹村, 片山, 岩佐, 横矢: 共有仮想空間を用いた協調型仮想物体モデラ VLEGOII, *電子情報通信学会論文誌*, Vol.80-A, 9, pp.1517–1526 (1997).
- 10) Masunaga, Y. and Watanabe, C.: The Virtual World Database System—Its Concept, Design and Prototyping, *Advances in Multimedia and Databases for the New Century—A Swiss/Japanese Perspective*, pp.61–70, World Scientific (2000).
- 11) Masunaga, Y. and Watanabe, C.: Design and Implementation of a Multi-modal User Interface of the Virtual World Database System (VWDB), *Proc. 7th International Conference on Database Systems for Advanced Application (DASFAA'01)* (2001).
- 12) Meehan, M.: Survey of Multi-User Distributed Virtual Environment, *Course Notes: Developing Shared Virtual Environments*, ACM Press (1999).
- 13) Moss, J.E.B.: Nested Transactions and Reliable Distributed Computing, *Proc. Symposium on Reliability in Distributed Software and Database Systems* (1982).
- 14) Rusinkiewicz, M. and Sheth, A.: Specification and Execution of Transactional Workflows, *Modern Database Systems*, Kim, W. (Ed.), pp.592–620 (1995).
- 15) Schnabel, M.A. and Kvan, T.: Implementing The First Virtual Environment Design Studio: Architectural Education for the Asian Century, *Proc. 1st ACAE Conference on Architectural Education*, pp.157–166 (2001).
- 16) Tomii, T., Salev, K., Imai, S. and Arisawa, H.: Human Modelling and Design of Spatio-Temporal Queries on 3D Video Database, *4th Working Conference on Visual Database System 4 (VDB4)*, pp.317–336 (1998).
- 17) Takemura, H. and Kishino, F.: Cooperating Work Environment Using Virtual Workspace, *Proc. Conference on Computer-Supported Cooperative Work (CSCW'92)*, pp.226–232 (1992).
- 18) 渡辺知恵美, 大杉あゆみ, 佐藤かず恵, 増永良文: 仮想世界データベースシステムにおける VWDB スキーマ・ドメイン定義言語の設計と実装, *電子情報通信学会第 12 回データ工学ワークショップ 1B-3* (2001).
- 19) Weikum, G.: Principles and Realization Strategies of Multilevel Transaction Management, *ACM Trans. Database Syst.*, Vol.16, No.1 (1991).

(平成 14 年 3 月 20 日受付)

(平成 14 年 7 月 5 日採録)

(担当編集委員 安達 淳)



渡辺知恵美 (学生会員)

昭和 50 年生・平成 12 年お茶の水女子大学大学院人間文化研究科数理情報科学専攻修了。現在同大学院同研究科博士後期課程在学中。VR システムにおけるデータベース支援について、特に、トランザクション管理、協調作業支援、マルチモーダルインタラクションに興味を持つ。



大杉あゆみ(学生会員)

昭和53年生。平成13年お茶の水女子大学理学部情報科学科卒業。現在同大学大学院人間文化研究科博士前期課程在学中。VRシステムにおけるデータベース支援について、特に、仮想世界における情報フィルタリング技術に興味を持つ。



佐藤こず恵(学生会員)

昭和53年生。平成13年お茶の水女子大学理学部情報科学科卒業。現在同大学大学院人間文化研究科博士前期課程在学中。VRシステムにおけるデータベース支援について、特に、問合せインタラクションに興味を持つ。



増永 良文(正会員)

昭和45年東北大学大学院工学研究科博士課程電気及通信工学専攻修了，工学博士。東北大学電気通信研究所助手，IIASA 研究員，IBM San Jose 研究所客員研究員，図書館情報大学助教授，同教授，を経て平成11年2月よりお茶の水女子大学理学部情報科学科教授，現在に至る。その間，情報処理学会データベースシステム研究会主査，ACM SIGMOD 日本支部長，情報処理学会監事等を歴任。情報処理学会フェロー。現在，日本データベース学会副会長。専門はデータベース。著書に「リレーショナルデータベース入門」(サイエンス社)、「リレーショナルデータベースの基礎—データモデル編」(オーム社)等。電子情報通信学会，日本パーチャルリアリティ学会，日本データベース学会，ACM，IEEE-CS 各会員。