

## CVE における空間バージョン管理とアクセス制御を用いた 非同期協調作業支援システムの研究

山田貴彦 柴田義孝  
岩手県立大学ソフトウェア情報学部

### 1. はじめに

近年マルチユーザによるコミュニケーションや協調作業を実現するための CVE (Collaborative Virtual Environment) に関する研究が多数行われ、様々な分野での利用が期待されている。

しかしながら既存の CVE には 2 つの問題点がある。まず、協調作業は参加者全員の同期作業を前提としており、事前のスケジュール調整が必須になる。また、その時作業に参加できなかったユーザは、空間で行われた過去の作業内容が分からないため現在の作業状況を把握できず、非同期作業が困難になる。

次にこれまでの CVE ではインタラクション制御が考慮されていない点である。そのため、不用意なインタラクションにより作業不参加者の作業範囲へ干渉し、不参加作業者の同意無しに作業状態が変更されてしまう等、非同期環境における作業の一貫性の維持が困難になる。

そこで本研究では、CVE における空間バージョン管理とアクセス制御を導入した非同期/分散協調作業を支援する SAVE (Space Archiver for Virtual Environment and CVE by revision management) システムの開発を行う。これにより非同期/分散作業における協調作業の効率化を実現する。

### 2. 関連研究

前章で述べた 2 つの問題点に対して、非同期作業支援に関しては、ユーザのインタラクションを記録して再生を行う文献<sup>1)</sup>が挙げられる。また、空間内におけるアクセス制御としては、ユーザの持つアクセス権に応じてオブジェクトの表示の詳細度を変更する文献<sup>2)</sup>が存在する。しかし、文献<sup>1)</sup>では空間の復元等は行えず非同期環境での協調作業は考慮していない。また文献<sup>2)</sup>ではオブジェクトの表示のみにとどまりインタラクションの制御は考慮しておらず、協調作業時の問題解決には至っていない。

本研究では空間バージョン管理とアクセス制御を導入することにより非同期協調作業を支援するシステムの開発を行った。

### 3. システム概要

#### 3.1 システム構成

本システムは図 1 に示すように複数の利用者が遠隔地から非同期に参加できる環境を想定しており、各参加者は IP ネットワークによりサーバに接続し、システムへログインを行う。一方サーバでは各クライアントに対して作業空間を提供し、空間記録を

Research on Asynchronous Collaborative Support System By in CVE  
Introducing Space's Version Management and Access Control  
Takahiko Yamada, Yoshitaka Shibata  
Faculty of Software and Information Science,  
Iwate Prefectural University

行う。記録されたデータは Revision Tree として表現され、サーバが Revision Tree を管理することでクライアントへの空間操作を提供する。また、サーバは同時にクライアントの一人として他のユーザ同様に作業に参加できる。

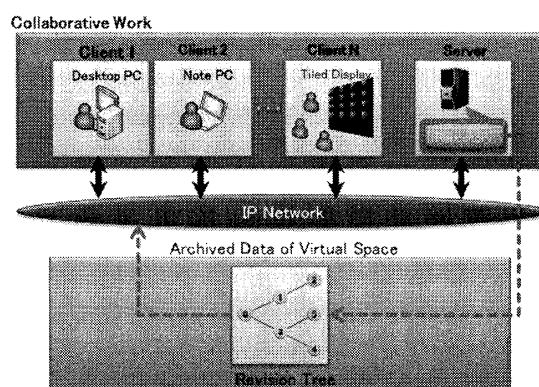


図 1 システム構成

#### 3.2 Revision Tree

本システムでは Revision Tree の概念を用いて 3 次元空間のバージョン管理を行う。図 2 に示すように Revision Tree はノードとリンクで構成され、ノードが保存された空間を、リンクが空間の間で行われたインタラクションの記録を表している。ノードの分岐は分岐元の過去のノードで異なる時間に異なる作業が行われたことを示しており、複数バージョンの空間と時系列関係が明確化される。ノード番号は生成順に付けられ、大きい番号のノードほど最近作業された空間であることを示す。これにより空間の状態遷移や過去の作業状況の視覚的な確認が容易になる。

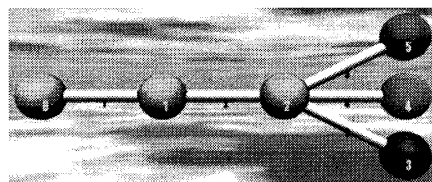


図 2 Revision Tree

### 4. システム機能

本章では本システムの持つ空間構築、空間記録、空間操作及びアクセス制御機能に関しての詳細を述べる。

#### 4.1 Commit

現在の作業空間を記録する。Commit の実行により Revision Tree に新たなノードが追加される。Commit は通常記録の Partial Commit と、それ以降の空間構築を Lock し、参照のみを可能にする Full Commit がある。Full Commit により一つの完成空間の保

存や分散作業時の完全整合の維持が可能になる。

#### 4.2 Record

作業中のインタラクションを現実時間のタイムスタンプ情報と共にサーバが自動的に記録する。Commit が実行されると同時にリンクとして出現する。

#### 4.3 Replay

Record により保存された情報を元に、過去のインタラクションを再現する。巻き戻しや早送りといった操作の他に、自由な視点での再生確認が可能になる。これにより非同期な参加者も当時の作業状況の確認が可能になる。

#### 4.4 Revert

記録された過去の空間を復元する。実行には参加者全員の合意が必要となるため、参加者は Revert 実行前に実行要求に対する投票を行う。要求が認められると空間の復元が行われ、ある段階からの作業のやり直しが可能になる。

#### 4.5 Clone

サーバの Revision Tree をクライアントのローカル環境へコピーする機能である。Clone により、分散環境での作業が可能になり、例えばローカル環境における実験的な作業や分散環境にて個々に作業を進めることが可能になる。

#### 4.6 Merge

Merge は、クライアントのローカル環境における特定のノードを指定サーバにアップロードする。対象ノードの共通の親をサーバから見つけ出し、その間の差分ノードも自動でアップロードしてサーバの Revision Tree へ統合する。これにより、分散作業を行った結果の異なる部分だけをサーバに統合し、Clone 以降の空間状態遷移を複数人で確認することが可能になる。

### 5. アクセス制御

本システムにおける作業空間へのアクセス制御では、セグメントドメインとセグメントドメインオブジェクトと呼ばれる 2 種類の制御対象をもち、アクセス制御権を有するユーザをスペースマネージャと呼ぶ。

#### 5.1 セグメントドメイン

セグメントドメインは、共有空間内の部分的な空間である。スペースに対するアクセス権を持っているユーザのみスペース内でのオブジェクトインタラクションが可能になる。スペースの情報は Clone, Merge 時にも引き継がれ、非同期/分散環境における作業範囲や作業状況等の確認が視覚的に可能になる。

#### 5.2 セグメントドメインオブジェクト

セグメントドメインオブジェクトは参加者単位のグループであり、例えば UserA が生成したオブジェクトは UserA のグループに属するため、UserB は UserA が生成したオブジェクトを扱うには UserA グループへのアクセス権が必要になる。これにより非同期/分散環境においても他人のオブジェクトへのアクセスを防ぐことで、作業状態の一貫性の維持が可能になる。

#### 5.3 スペースマネージャ

本システムではアクセス制御を実行するユーザを「スペースマ

ネージャ」と呼び、サーバとしての参加者が役割を務める。セグメントドメインの設置等もスペースマネージャが行う。

### 6. プロトタイプシステム

本システムの機能を評価するためのプロトタイプシステムとして、複数人で郊外型ショッピングモールをデザインするアプリケーションを開発した。開発は Windows Xp 上で行い、IDE に Visual Studio 2008、開発言語には C++、3D ライブラリとして G3D-7.01 を使用した。

参加するユーザとしてオーナーと複数のテナントが非同期的に作業する。オーナーはモール共通部分（通路や街頭等）や自身のコンテンツを、テナントは店舗周りのデザインを行う。これに本システムを適用することで、例えばテナントが密集している際には作業範囲を明確にすることで効率的な作業が可能になる。また、テナントが別々の日程で作業する際にもアクセス制御を用いることで、あるテナントが不参加の場合にも過去の作業の一貫性が保証され、非同期環境における協調作業の効率化を実現する。

今後はプロトタイプシステムを用いた本システムの機能評価として、アクセス制御機能評価と総合機能評価の 2 種類を行う。ユーザに空間構築の作業を行ってもらいアンケートによる定量評価を行う予定である。

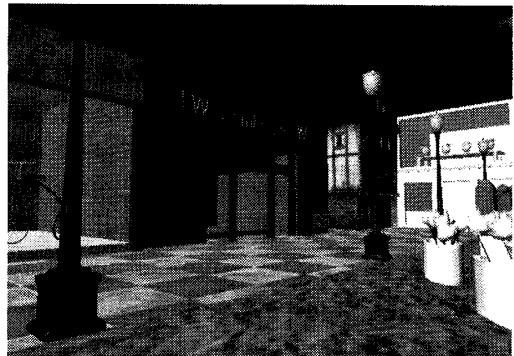


図3 プロトタイプシステムによる空間構築例

### 7. まとめ

本論文では、CVE における非同期/分散環境を考慮した空間バージョン管理とアクセス制御を行う SAVE システムを開発した。本システムを利用することでアクセス制御による作業一貫性の維持や効率的な作業の実現が可能になり、空間バージョン管理により非同期/分散環境における空間の前後関係の整合性を容易に保つことが可能になる。

### 参考文献

- 1) Chris G, Marti F, Jim P, and Steve B: Applications of temporal links: Recording and replaying environments, Proceedings of the IEEE Virtual Reality 2002
- 2) 田島敬史, 依田和也, 田中克己: 複数詳細度をもつ CG データのためのアクセス制御とアクセス権管理, 電子情報通信学会論文誌, Vol. 1, No. 1, pp. 193-200 (1999)