

# マルチユーザによる非同期協調作業を考慮した CVE 支援システム

小笠原 弘樹

柴田 義孝

岩手県立大学ソフトウェア情報学研究科

## 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環境を実現する CVE(Collaborative Virtual Environment)と呼ばれるアプリケーションの提案が行われている。その CVE で前提となっているのは仮想空間や計算機およびネットワーク資源、時間の共有であり、協調作業を行うには事前連絡やスケジュールを考慮する必要があることから、同期的な協調作業を行うことは困難である。また、一般に CVE では作業履歴が記録されないため、過去の作業をやり直して編集できないといった問題も挙げられる。

CVE における空間記録の関連研究として、Temporal Links を用いた記録と再生システム [1][2] が研究されており、現在と過去の空間にリンク情報を持たせることで、現在の仮想空間上に過去の空間をオーバーレイさせ同時に再生することを実現している。

本研究では、新たに Revision Tree の概念を導入し、仮想空間を連続的にアーカイブ化し、過去の作業空間を再生するだけでなく、任意の時点の空間を再構築できるシステムの提案と構築法について述べる。

## 2 システム概要

本稿で提案するのは、マルチユーザによる非同期協調作業を考慮した CVE 支援システム (SAVE : Space Archiver for VE and CVE by revision management) であり、作業空間の記録を行うことで Revision Tree を構築し、過去の作業空間の再生や再構築を実現する。

### 2.1 仮想空間のモデリング

仮想空間はオブジェクトの階層構造として表現され、サーバおよびクライアントのメッセージ通信によって同期がとられ更新、共有される。オブジェクトは 3D モデルデータを読み込むことで描画され、そのデータ構造は Meta Data Format により定義される。各クライアントが所持していない 3D モデルデータおよび Meta Data Format は CVE 参加時にサーバよりダウンロードを行うことで参加者全員が同じ CVE 空間を共有する。

CVE Support System for Asynchronous Collaborative Work by Multiple Users

Hiroki Ogasawara (g031f003@edu.soft.iwate-pu.ac.jp)

Yoshitaka Shibata (shibata@iwate-pu.ac.jp)

Graduate School of Software and Information Science,  
Iwate Prefectural University

## 2.2 Meta Data Format

3D モデルデータはその種類によってデータ構造が異なるため、一般に上位アプリケーションから利用する際はフォーマット毎に異なるインターフェースを用意する必要がある。そこで図 1 に示す Meta Data Format を定義することで抽象化し、上位アプリケーションから統一した操作が行えるようにする。Meta Data Format はオブジェクトに一对一対応しており、仮想空間におけるオブジェクトの属性を決める。

```
Object ID (int); [S]
Group ID (int group_id, int parent_id);
Position (double x, double y, double z);
Angle (double x, double y, double z);
Size (double);
Texture Name (string);
Color (R, G, B, A);
Access Control Flag (bool); [S]
LOD Level (double);
Record Flag (bool);
Selected Flag (bool);
*[S]: System Only
```

図 1 : Meta Data Format

## 2.3 Revision Tree

図 2 に示す Revision Tree は空間遷移を表すデータ構造であり、ノード(Revision)とリンクによって構成される。3次元で可視化することでシステムの機能を利用する時に操作、参照する。

構成要素であるノードはある瞬間の空間のスナップショットであり、全ての記録対象オブジェクトの Meta Data Format を記録する。一方、リンクはオブジェクトインタラクションにタイムスタンプを付加したもので、その空間での作業経過を記録したものである。

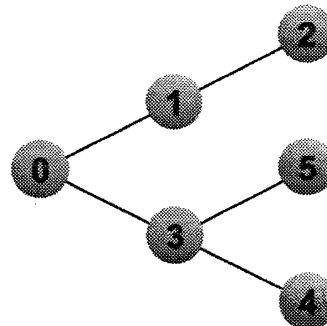


図 2 : Revision Tree

### 3 SAVE システム

提案するシステムのアーキテクチャを図3に示す。本システムはトランスポート層の上位層、CVEアプリケーションの下位層に3階層で構成され、アプリケーションに対して効率的な協調作業支援を提供する。System Interface LayerではI/OイベントのモニタリングとRevision Treeの可視化、System Function Layerでは後述するシステム機能の制御、Revision Control Layerではイベント記録やRevision Treeの管理を行う。

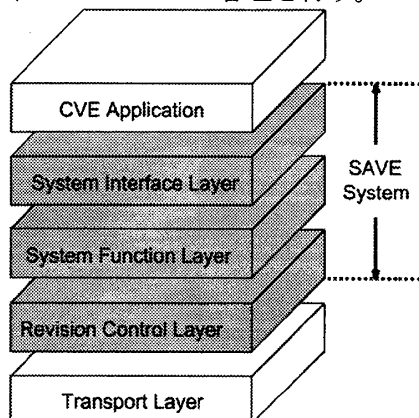


図3：システムアーキテクチャ

#### 3.1 Edit 機能

Edit機能はオブジェクトに対するインタラクションを提供する。配置、移動、回転、選択、削除、グループ化といった協調作業に必要なインタラクションをシステムで提供する。

#### 3.2 Record 機能

Record機能は仮想空間をCommitとRecordの2種類の方法で記録する。Commitによる記録では現在の仮想空間の状態を記録し、Revision TreeのノードであるRevisionを生成する。

一方、Recordによる記録ではEdit機能を利用したインタラクションにタイムスタンプを付加した連続記録を行い、Revision Treeのリンクを生成する。

#### 3.3 Replay 機能

Replay機能はRevision Treeを操作、参照して任意のリンクを選択することで過去の作業空間を再生する。様々な角度からオブジェクトを見ることができ、オブジェクトの選択非表示やLOD制御[3]ができる点がビデオ再生と異なる。

#### 3.4 Revert 機能

Revert機能はRevision Treeを操作、参照して任意のノードを選択することで過去の作業空間を再現する。CVEに参加する全てのユーザは選択されたRevisionの空間状態に戻り、そこから分岐作業や作業のやり直しを行うことができる。

#### 3.5 Clone 機能

Clone機能はサーバからローカルへRevision Treeのコピーを行う。以後ユーザはローカルサーバへ接続することで非同期なローカル作業を行うことができる。

#### 3.6 Merge 機能

Merge機能はC/S間のRevision Treeの差分をサーバ側へマージする。これによりサーバ側のRevision Treeには差分ノードが分岐として追加され、Clone機能による作業成果を取り込むことができる。

### 4. プロトタイプシステム

現在はプロトタイプシステムとしてSAVEシステムの機能を反映した、都市や部屋空間の構築を行うアプリケーション構築を行っている。



図4：プロトタイプシステム

### 5. まとめ

本稿ではマルチユーザによる非同期協調作業を考慮したCVE支援を実現するためのSAVEシステムについて、新たにRevision Treeという概念を導入した記録手法について提案した。これにより、非同期協調作業支援と生産性の向上が期待できる。

#### 参考文献

- [1] Chris Greenhalgh, Martin Flintham, Jim Purbrick, and Steve Benford. Applications of temporal links: Recording and replaying virtual environments. In Virtual Reality, 2002. Proceedings. IEEE, 2002.
- [2] Chris Greenhalgh, Jim Purbrick, Steve Benford, Mike Craven, Adam Drozd, and Ian Taylor. Temporal links: recording and replaying virtual environments. In ACM Multimedia 2000.
- [3] Dawei Ding and Miaoling Zhu. A model of dynamic interest management: interaction analysis in collaborative virtual environment. In VRST '03: Proceedings of the ACM symposium on Virtual reality software and technology, pages 223–230, New York, NY, USA, 2003. ACM Press.