

# CVE 支援のための VR 空間連続アーカイバの提案と構築

小笠原 弘樹<sup>†</sup> 柴田 義孝<sup>†</sup>

<sup>†</sup> 岩手県立大学ソフトウェア情報学部ソフトウェア情報学科

## 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空間や資源、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、CVE における協調作業では作業履歴が記録されないため、途中からの参加者は以前の作業内容を把握できなかったり、途中からの編集作業を行うことができないといった問題が挙げられる。

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、Frame と Revision Tree の概念を新たに導入し、VR 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、CVE 支援のための VR 空間連続アーカイバシステムであり、図 1 に示すように、ユーザは各端末からネットワークに接続された CVE に参加し、他のユーザとコミュニケーションや協調作業を行う過程を記録する。その記録情報を元に Revision Tree と呼ばれるツリー構造を構築し、必要に応じて空間の再生や復元の際に参照する。

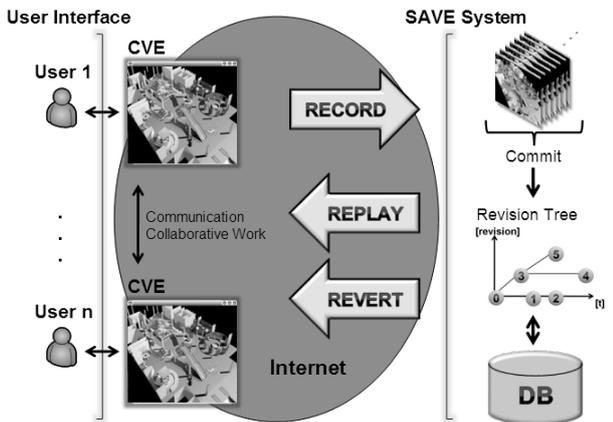


図 1: システム概要図

## 3 システムアーキテクチャ

提案するシステムのアーキテクチャを図 2 に示す。本システムはトランスポート層の上位層に位置し、3 階

層で構成され、CVE 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 Control Layer では、CVE 内のイベントを記録し、空間の再生や再現のためのインターフェイスを提供する。Revision Control Layer では Revision Tree を構築し、記録された空間情報を時間軸と Revision 番号で管理する。Repository Control Layer では Revision Tree を記録する。サーバおよびクライアント間では、仮想空間内のオブジェクトやイベントなどのメッセージが交換され、Repository を分散させることで非同期協調作業が可能となる。

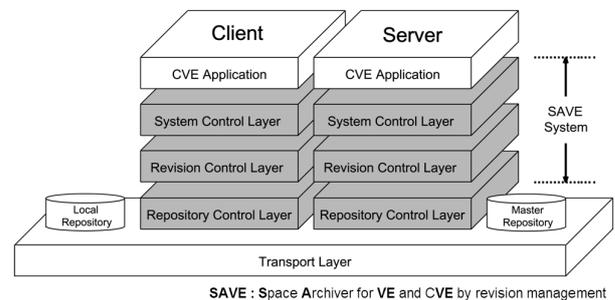


図 2: システムアーキテクチャ

## 4 提案手法

提案する手法には、アーカイバとしての機能とエディタとしての機能がある。仮想空間の表現法、Frame、Revision Tree、各機能の詳細を以下に示す。

### 4.1 仮想空間の表現法

仮想空間は階層構造を持ったオブジェクトの集合として表現される。オブジェクトの持つデータには、図 3 に示すようにオブジェクト ID、3 次元座標値、色情報、テクスチャ ID、サイズ、角度などがある。

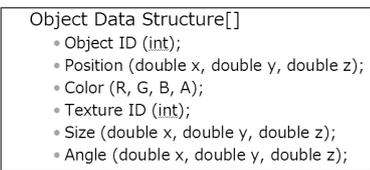


図 3: オブジェクトデータ構造

### 4.2 Frame

VR 空間の記録は Frame の連続記録として記録される。Frame とはある瞬間の VR 空間の状態のことであり、VR 空間は Frame の更新により描画され、描画の度に Frame 番号が増加する。図 4 に示すように Frame が持つ情報は Revision 番号、Frame 番号、全オブジェクトの属性である。本提案手法における記録対象は各 Frame に存在するオブジェクトであり、記録するかしないかは利用者がオブジェクト毎に設定可能である。

Continuous VR Space Archiver for CVE Support System

<sup>†</sup> Hiroki Ogasawara (g031b025@edu.soft.iwate-pu.ac.jp)

<sup>†</sup> Yoshitaka Shibata (shibata@iwate-pu.ac.jp)

Faculty Of Software and Information Science, Iwate Prefectural University (†)

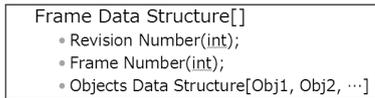


図 4: フレームデータ構造

### 4.3 Revision Tree

Revision Tree は全ての Revision を木構造で管理し、空間を再生、再現するために参照される。3次元仮想空間上で可視化することで空間の移動や視点移動、Revision Tree の選択、拡大、縮小を可能にする。

### 4.4 Record 機能

図 5 に示すように、VR 空間の記録方法には、Commit による記録と Recording による記録の 2 種類がある。Commit による記録は、参加者全員の同意を得ることで行うことができ、Frame 情報に Commit 情報を付加して記録される。Commit 情報の内容は、記録日時、Commit を要請したユーザ名、参加者名、コメント、直前の Revision 番号であり、Commit により Frame 番号はリセットされ、Revision 番号は増加する。この記録は空間を再現するために使用される。

一方、Recording による記録では、行動やオブジェクトインタラクションといったイベント発生中の Revision 間の全 Frame 情報が記録される。この記録は空間を再生するために使用され、途中の Frame が抜けているところは前の Frame 情報を参照することで再生することができる。

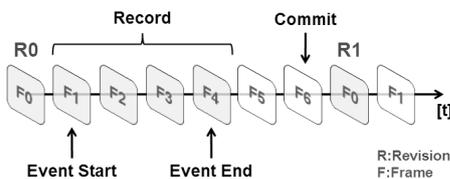


図 5: Frame と記録の関係

### 4.5 Replay 機能

空間の再生は Revision 単位で行われ、Revision Tree で管理される Recording の内容を Frame 番号順に再生し、記録内容の無い Frame は以前の Frame 情報を参照する。このフレーム番号を操作することで再生速度を変えたり、巻き戻すことができる。また、再生する必要が無いオブジェクトは LOD 制御 [3] によって表示レベルを下げる事が可能である。

### 4.6 Revert 機能

空間の再現を行うには、現在の作業空間を Commit する必要がある。その後、Revision 番号を指定し、作業空間の置き換えを行う。これにより任意の Revision からのやり直しや分岐作業を行うことが可能である。Revert 機能による分岐は Revision Tree に反映される。

### 4.7 Edit 機能

Edit 機能では、オブジェクトに対するインタラクションを提供する。生成、削除、置き換え、グループ化、移動、回転、拡大、縮小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ことで開発者の負担を減らすことが可能となる。

### 4.8 Clone 機能

Clone 機能により、個人空間におけるオフライン作業が可能となる。共有空間を Commit し、Server の Master Repository を個人の Repository にコピーすることで、実験的な作業を個人空間で試すことができ、非同期にも協調作業を行うことができる。

### 4.9 Merge 機能

Merge 機能により、個人空間におけるオフライン作業の結果を共有空間に取り込むことが可能となる。個人空間のオブジェクトを選択し、共有空間のオブジェクトとして新たに追加する。これにより、個人の成果物を無駄にすること無く共有空間に反映させることができる。

## 5 プロトタイプシステム

本研究ではプロトタイプシステムとして、本研究の提案手法を用い、バージョン管理による作業支援や柔軟なインタラクション、ユーザ間のコミュニケーションを実現する。仮想空間を構成するオブジェクトの分類と記録対象を、図 6 に示す。

本システムの開発には OS に Linux, Ruby 言語と OpenGL を用いて 3D 空間の描画を行っている。また、ネットワーク通信と DB には Ruby 言語標準添付ライブラリである dRuby と PStore を利用している。現在は、Record, Replay, Revert の機能の実装ができており、空間の再生と再現ができることを確認している。

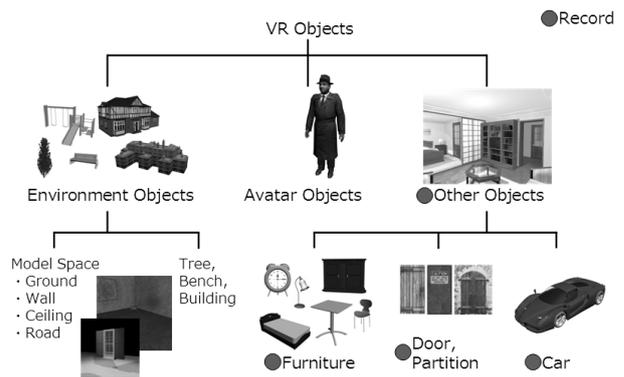


図 6: オブジェクトの分類と記録対象

## 6 まとめ

本稿では CVE 支援のための VR 空間連続アーカイバの提案と構築について Frame と Revision Tree という概念を新たに導入する方法について提案した。提案手法により CVE における非同期協調作業支援と生産性の向上が期待できる。現在は提案手法の評価を行うためにプロトタイプシステムの構築を行っている。

### 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*, 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.