2007－DPS－131 (18)
2007／6／7

Revision Tree を用いた CVE 連続アーカイブシステム

小笠原弘樹 柴田義孝
岩手県立大学大学院 ソフトウェア情報学研究科

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 Collaborative Virtual Environment (CVE) に関する研究が多数行われている. その CVE において協調作業を行う場合の事前連絡やスケジュールの考慮といった同期作業における問題を解決するために, 本研究では VR 空間そのものを連続的にアーカイブ化し, 利用者の目的や都合に応じて過去の作業空間を再生したり, 任意の時点の空間を再現できるシステムについて提案する. 空間内の全てのオブジェクトをアーカイブの対象とすることで選択的, かつ柔軟な空間の再生や再現が可能となり, 作業目標に向かって利用者同士が効率的かつ非同期にも協調作業が行える CVE 支援システムを構築できる.

Continuous CVE Archive System Using Revision Tree Presentation Method

Hiroki Ogasawara Yoshitaka Shibata

Graduate School of Software and Information Science
Iwate Prefectual University

In this paper, we introduce a new idea of revision tree which can provide a flexible mechanism to store the current cllaborative virtual environments (CVE) space, replay the past recorded CVE space and replace the current CVE by other past CVE space. The revision tree with hierarchical 3D structure is visualized in space and operated by users to refer the past space information using replay, revert, clone and merge functions. Those functions provide effective and flexible cllaborative work in CVE to support communication among many users over computer network. Using this method, the users as group members can progress cllaborative work and confirm the past CVE without prior communication even if some of the group members do not participate in the space. Thus, asynchronous collaborative work can be attained and applied to efficiently more creative CVE applications.

1 はじめに

近年のマルチメディアやネットワーク技術の発達により, CVE (Collaborative Virtual Environment) と呼ばれるアプリケーションが提案されている [1][2][3][4][5][6]. これは CG (Computer Graphics) による三次元仮想環境をネットワーク上に構築し, 複数人が同じ空間を共有して協調作業を行うアプリケーションでり, 教育やエンターテイメントなどの分野で利用されている. CVE の主な機能として挙げられるのは, 仮想オブジェクトへのインタラクションとユーザ間のコミュニケーションである.

このような CVE において協調作業を行う場合, 前提となっているのは空間や資源, 時間の共有である. ユーザは同じ空間を共有し, 同時に作業を進める必要がある. しかし, その様な前提で遠隔地のユーザ同士が協調作業を行うには, 事前連絡や相手のスケジュールを考慮する必要があることから, 特にユーザ数が増加するにつれて同期的な協調作業を行うことが困難な場合が多い. また, 通常 CVE における協調作業では作業履歴の記録が考慮されておらず, 途中参加のユーザは以前の作業内容が分からず, 途中からの編集作業ができないといった問題も挙げられる.

このような背景から, 非同期作業の支援と協調作業記録の必要性が考えられる. 本研究では新たに Frame と Revision Tree の概念を導入し, VR (Virtual Reality)

空間を連続的にアーカイブ化するシステムについて提案する．これにより過去の作業空間を再生して確認したり，任意の時点の作業空間を再現して作業のやり直しを行うことが可能となる．

2 関連研究

本研究の関連研究としては，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．

このシステムでは同ノッティンガム大学が開発した MASSIVE-3 [3] と呼ばれるシステムを利用している．MASSIVE-3 システムではサーバ・クライアント型の分散 CVE を対象として，3D グラフィックスやオーディオ・ビデオストリームを提供している．部屋や建物などの空間はロケールと呼ばれる時間的構造体として表現され，行動やオブジェクトインタラクション，ユーザの会話などをロケール単位で記録することが可能である．これにより過去の空間を再生することは可能であるが，過去の任意の時点に遡って作業をやり直すことはできない．また，別空間内で作成された成果物の取り込みといった非同期協調作業についても考慮されていない．

3 システム概要

本研究では，三次元仮想空間上での協調作業を支援するために，仮想空間を連続的にアーカイブ化することで過去の空間を再生したり，任意の時点の空間を再現し，ユーザーに対して必要なインタラクション機能を提供するシステムの提案と実現を行う．

これによりユーザは協調作業に必要な事前連絡や相手のスケジュールを意識せずに，その空間で何が行われたのかを把握でき，必要に応じてその作業空間を過去の任意の時点のものと置き換えることが可能となる．また，Revision Tree の概念を導入することで同期的な協調作業だけではなく，試験的な作業をローカル作業として行うことができ，その作業の成果物を共同作業空間に取り込むということも可能となる．

本研究の適応分野としては，複数人で空間構築したりデザインする CVE システムを想定しており，複数ユーザが作業空間の履歴を参照でき，作業空間のプレゼンテーションや他の参加者とのコミュニケーションが可能な空間を構築することができるアプリケーションを想定している．

3.1 ネットワーク構成

提案するシステムは図 1 に示すように複数クライアントとサーバによって構成される．各ユーザはクライアント端末から高速ネットワークにより接続されたサーバにログインすることで 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．その記録情報を元に Revision Tree を構築し，機能利用のために必要に応じて参照する．

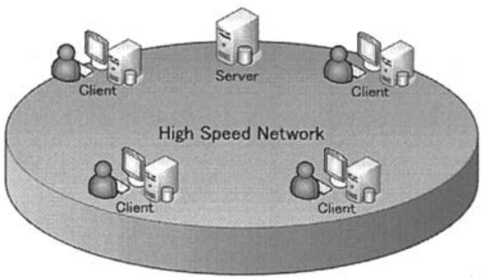

Figure 1: ネットワーク構成図

3.2 システム構成

提案するシステムは，機能的にクライアントとサーバに分けることができる．システム構成を図 2 に示す．

クライアントでは，ローカル作業空間である VR と協調作業空間である CVE に対する 2 つのインターフェースを持つ．CVE に参加する場合は，サーバに接続要求を出し参加する．

サーバでは，クライアントの接続をセッションとして管理し，CVE におけるオブジェクトインタラクションや作業記録を Frame と呼ばれる単位で保存する．保存方法には Commit と Record の 2 種類があり，それらの保存情報から Revision Tree を構築する．

クライアントでは，ローカル作業や協調作業を実行するため Replay, Revert, Clone, Merge の機能を使うことができる．例えば Clone によりサーバの Revision Tree をクライアントであるローカルホストにコピーすることで実験的なローカル作業を行うことができる．その作業内容は Merge により再びサーバの Revision Tree にマージすることも可能であり，非同期協調作業を行うことができる．

4 システムアーキテクチャ

提案するシステム全体のアーキテクチャを図 3 に示す．トランスポート層の上位層に 3 層から成る SAVE システムとその上位層に CVE アプリケーション層が位置し，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．SAVE システムはサーバとクライアントの両方の機能を備えており，System Control Layer では，CVE 内のイベントを記録し，空間の再生

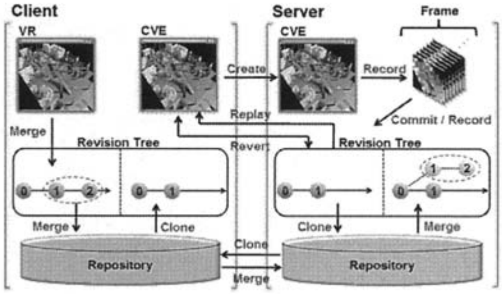

Figure 2: システム構成図

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 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, Revision Tree を Clone, Merge することで非同期協調作業支援が可能となる.

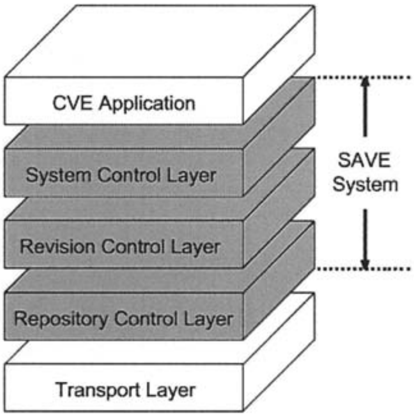

Figure 3: SAVE System

4.1 CVE Application

想定するアプリケーションでは Server に接続された複数の Client が空間を共有し, 自由に視点移動やウォークスルーができる. また, オブジェクトに対して移動や置き換え, 生成や削除といったインタラクションを行うことができ, その中で行われるインタラクションを以下で述べる SAVE により記録する. これらの機能はサーバにより管理され排他制御される. 例えばインテリアデザインを行うアプリケーションの場合, 家具の位置を移動させたり, 建物の内装を変えたりといったインタラクションを SAVE で記録しておき, 後で Revision Tree を参照することで空間の再生や再現を行うことが可能となる.

4.2 System Control Layer

上位アプリケーション層に対してシステムの機能を提供するインターフェースとなり, アプリケーション層における空間遷移をイベントとして連続的に記録することで Revision Tree の構築を行う. この記録を行うのはサーバの機能であり, 記録された空間情報は下位 Revision Control Layer によって Revision Tree の構築に使われる.

4.3 Revision Control Layer

上位 System Control Layer によって記録された空間情報から Revision Tree を構築する. Revision Tree はサーバによって管理されるが, クライアントは Clone 機能によりローカルホストに Revision Tree をコピーすることも可能である. 他にも Replay, Revert, Merge 機能を利用する際には System Control Layer より呼び出され Revision Tree の情報が参照される.

4.4 Repository Control Layer

上位 Revision Control Layer により構築された Revision Tree を Repository と呼ばれる DB に保存するためのインターフェースを提供する.

5 SAVE のデータ構造定義

本研究で提案する SAVE (Space Archiver for VE and CVE by revision management) システムには, 空間のアーカイバと空間エディタの両方の機能がある. 前者は空間を連続的にアーカイブすることで過去の空間を再生, 再現できる機能であり, 後者は上位アプリケーション層に対して CVE における基本的インタラクション機能を提供する. これらの機能を提供するために使われる Frame と Revision Tree のデータ構造の詳細を以下に示す.

5.1 Frame

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 の連続記録として記録される. VR 空間はアプリケーションにより定義される Frame の更新により描画され, 描画の度に Frame 番号が増加する. System Control Layer ではその Frame 間の差分を記録する. Frame の持つデータ構造は図 4 のように定義される.

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．

```
Frame Data Structure[]
  * Revision Number(int);
  * Frame Number(int);
  * Objects Data Structure[Obj1, Obj2, …]
```

Figure 4: Frame のデータ構造

5.2 Revision Tree

Revision とは Record 機能により Commit されたある瞬間の空間情報である．Revision Control Layer により構築，管理される Revision Tree は図 5 に示すように，全ての Revision を木構造で管理し [7], 3 次元仮想空間上で可視化することで Revision Tree を表示する空間の移動や，選択，拡大，縮小を可能にし，空間を再生，再現するために参照される．また，作業空間の遷移は Revision Frow として表現され，親 Revision を選択することで分岐する全ての子 Revision を表示できる．

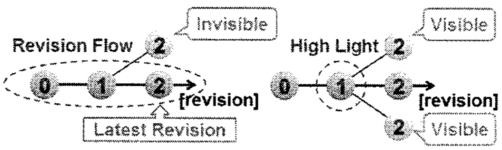

Figure 5: Revision Tree

6 SAVE の編集及びアーカイブ機能

CVE における協調作業では作業履歴の記録が考慮されていないため，途中参加のユーザは以前の作業内容が分からず，途中からの編集作業ができないといった問題があり，そのような問題を解決するために本システムでは 6 つの機能を提供する．各機能の詳細を以下に示す．

6.1 Create

Create は協調作業空間内の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，縮小といったインタラクションをシステムで提供することでアプリケーション開発者の負荷を減らすことが可能となる．

6.2 Record

VR 空間の記録方法には，Commit による記録と Record 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する．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 Revert 機能で使用される．

一方，Record 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，空間を再生するために Replay 機能で使用される．

Record 機能による Commit, Record, Frame の関係を図 6 に示す．

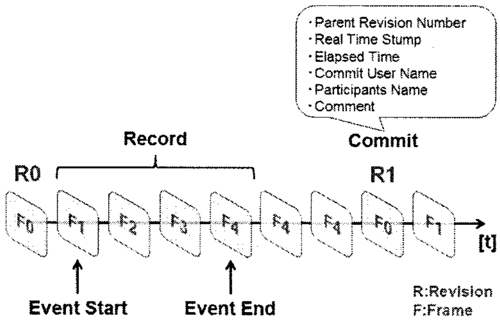

Figure 6: Record

6.3 Replay

空間の再生は図 7 に示すように Revision 単位で行われる．Revision Tree で管理される Record 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 [8] によって表示レベルを下げることが可能である．

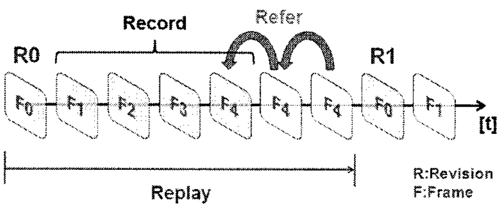

Figure 7: Replay

6.4 Revert

図8に示すRevert機能では現在の作業空間をCommitして, 任意のRevision番号の作業空間との置き換えを行う. これによりやり直しや分岐作業を行うことが可能である. Revertによる分岐はRevision Treeに反映される.

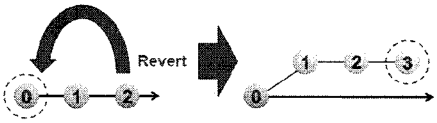

Figure 8: Revert

6.5 Clone

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. 図9のようにServerのRepositoryをclient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.

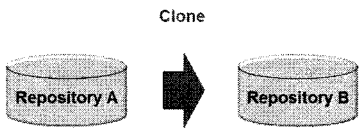

Figure 9: Clone

6.6 Merge

図10に示すように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Revision Treeを選択し, 共有空間のRevision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.

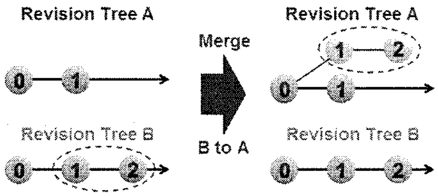

Figure 10: Merge

7 プロトタイプシステム

本システムのプロトタイシステムとして, 協調インテリアデザイン支援システムを構築する. 消費者, インテリアデザイナ, セールスマン, メーカが協調しながら消費者にとって最も相応しい部屋空間を実現する. 想定する構成は図11, 12に示すように100MのEthernet上に, 全体のメッセージや記録データをRevition Treeとして一元管理するServerと, 利用者が3Dインターフェースとして利用するClientを複数配置し, 利用者に空間デザインを行うことができる協調作業空間を提供する.

本提案手法を反映させたシステムを用いることで利用者は空間内での作業履歴を記録をしておくことができ, Create機能により自由に空間を構築することができる. その空間構築の過程はRecord機能により記録され, 途中から協調作業に参加した利用者はReplay機能によりどの様な変化があったのか知ることができる. また, 他の利用者とのコミュニケーションを行うことで構築してきた空間を作り直すことになった場合, Revert機能により最初から作り直すことなく過去の空間を再現できる. さらに, 協調作業をしているうちに実験的な作業を行いたくなった場合, Clone機能によりClientのローカルRepositoryにServerのRepositoryをコピーすることで個人作業を行うことができる. その個人作業の結果はMerge機能により再びServerのRepositoryに反映させることができる.

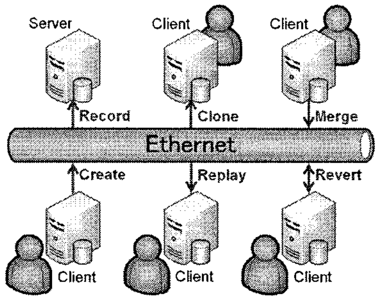

Figure 11: プロトタイプシステムネットワーク環境

7.1 仮想空間のモデリング

仮想空間を構成する家具などのオブジェクトは, オブジェクトID, グループID, 3次元座標値, 色情報, テクスチャID, サイズ, 角度のデータを持つ. アバタのインタラクションによりオブジェクトに対して変更があった場合はこれらのデータも変更される. 例えば家具の移動では3次元座標値や角度が変更され, 色やテクスチャの変更では色情報やテクスチャIDが変更される.

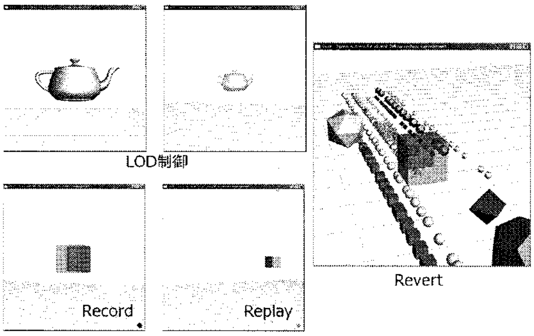

Figure 12: プロトタイプシステムの機能

また, オブジェクトは図 13 に示すように, 空間オブジェクト, 景観オブジェクト, インテリアオブジェクト, アバタオブジェクト等に分類され, グループ化することでモデル空間として DB に保存できる. これらのオブジェクトは Create 機能により作成, 削除, 置き換え, グループ化, 移動, 回転, 拡大, 縮小といったインタラクションが可能である.

例えば部屋空間を構築し, その部屋の中にいくつかの家具があった場合, それらの家具を一つのグループとしてグループ化することでそれを一つのオブジェクトとして扱うことができるようになる. 部屋空間自体もグループ化して Model Space として DB に保存しておくことでより柔軟な空間の構築を行うことが可能である.

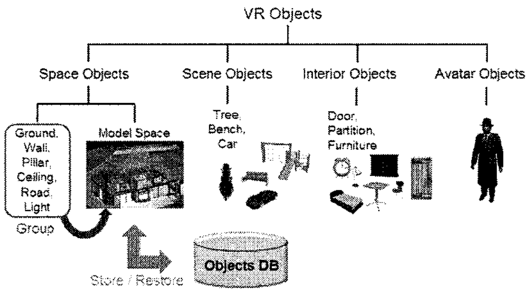

Figure 13: オブジェクトの分類

7.2 実装

プロトタイプシステムの実装はスクリプト言語 Ruby によって行った. Client のユーザインターフェースには OpenGL (Ruby/OpenGL) による 3D インターフェースを利用した. 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. これらを Linux OS で実装した. 現在は, Record, Replay, Revert と Create の一部の機能実装ができている.

Ethernet 上で Server と Client マシンを接続し, 空間を共有した上でオブジェクトを配置し, そのオブジェクトをマウス操作により移動させるインタラクションを行った. そのインタラクションを Record により記録し, Replay によりそのインタラクションを再生できることを確認した. また, Commit によりある瞬間の空間状態を記録しておき, Revert を選択することで Revert により復元できることも確認した.

8 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. また, 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.

References

[1] Chris Greenhalgh, Martin Flintham, Jim Purbrick, and Steve Benford. Applications of temporal links: Recording and replaying virtual environments. *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. *ACM Multimedia 2000*, 2000.

[3] Chris Greenhalgh, Jim Purbrick, and Dave Snowdon. Inside massive-3: Flexible support for data consistency and world structuring. *Proc. Third International Conference on Collaborative Virtual Environments (CVE2000)*, 2000.

[4] Ekaterina Prasolova-Forland. Supporting social awareness among university students in 3d cves: Benefits and limitations. *Third International Conference on Cyberworlds (CW'04)*, 2004.

[5] Theodor G Wyeld. Role play in 3d virtual environments: a pedagogic case study. *Proceedings of the 6th ACM SIGCHI New Zealand chapter's international conference on Computer-human interaction*, 2005.

[6] Yushi Nakai, Naoki Kamon, and Yoshitaka Shibata. Scalable collaborative virtual environment considering user ' s interest information. *Proc. 20th International Conference on Advanced Information Networking and Applications - Volume 2 (AINA '06)*, 2006.

[7] Kai Pan, E. James Whitehead, and Guozheng Ge. Textual and behavioral views of function changes. *Proceedings of the 3rd international workshop on Traceability in emerging forms of software engineering*, 2005.

[8] Dawei Ding and Miaoling Zhu. A model of dynamic interest management: interaction analysis in collaborative virtual environment. *VRST '03: Proceedings of the ACM symposium on Virtual reality software and technology*, 2003.