

ハイパーメディアに基づく共同文書作成環境 MuHyme

村 永 哲 郎[†] 守 安 隆[†]
友 田 一 郎[†] 水 谷 博 之[†]

共同文書作成作業における複数ユーザ間の情報の共有と、コミュニケーションを支援する環境を提供するために、マルチユーザ・ハイパーメディアシステム MuHyme (Multi-user Hypermedia) を開発した。MuHyme では基盤データベースとしてハイパーメディアを採用し、相互に関連するマルチメディア情報を長期間にわたって共有一元管理することを可能とした。またリンクづけ操作として、ノードへのコメントづけを導入することにより、協調作業における基本的なコミュニケーション手段を提供している。これにより複数のユーザが文字や図形、静止画、音声を含むマルチメディア文書を共有して、その文書に非同期的にコメントづけを行って議論することを可能とした。MuHyme の特徴は、(1)ハイパーメディアのノード更新に、チェックイン・チェックアウト操作に基づくデータ更新モデルを導入し、長時間セッションにおいて他のユーザの作業進行を妨げないようにした、(2)他のユーザの操作をメッセージにより通知し合うメカニズムを提供することにより、協調的なアクセスを可能とした、(3)文書のバージョン管理と議論の履歴管理とを統合し、議論の繰り返しを妨ぐようにしたことにある。MuHyme は実験システムの開発を終了し、利用評価を行っている。本論文では、MuHyme の特徴的なグループウェア機能の設計と実現、利用評価について述べる。

MuHyme: An Environment for Collaborative Writing Based on Multi-user Hypermedia

TETSURO MURANAGA,[†] TAKASHI MORIYASU,[†] ICHIRO TOMODA[†] and HIROYUKI MIZUTANI[†]

We have developed a **Multi-user Hypermedia** system (**MuHyme**) to support information sharing and communication between users in collaborative works such as group writing. MuHyme is built on top of a hypermedia database, which allows users to collaboratively share multimedia documents including text, image, drawing and sound. For coordinated work, MuHyme employs annotated nodes which provide asynchronous communication between multiple users. We focus on two issues in applying hypermedia technology to groupware: access management to shared information in long-term session, and mechanisms to promote coordination. We propose (1) shared data access based on check-in/check-out operations so as not to block other users' access in long-term session, (2) notification of user-operations via inter-session messages to promote coordination of access to shared information, and (3) version management of documents and their annotations to facilitate group discussions. In this paper, we discuss design and implementation of these features including an evaluation of a prototype of MuHyme.

1. はじめに

近年マルチメディア・ワークステーションの開発、計算機ネットワークの高速化を背景として、分散環境における人と人との協調作業を支援するグループウェアの研究開発が活発化している^{1),2)}。会社などの組織においてよく行われるグループ作業として、共同文書作成、ソフトウェアなどの共同設計・開発がある。これらのタスクは、複数のユーザが情報を比較的長期間

にわたって共有しながら、それぞれが独立性の高い個人作業を分担する形で進行する。その間、途中の成果物を共有して議論を重ねたり、助言をしあったりしながら、それぞれの作業を進めていく。これらは、個人作業を中心としてユーザ間のコミュニケーションが非同期的に行われる、いわば疎結合のグループ作業である。

例えば設計書、マニュアル、企画書といった文書の共同作成においては、全体の構成と分担が決まると、あとは個々の担当者がそれぞれ独立して文書作成を行う。そして下書きをグループ内でレビューして、相互にコメントづけを行いながら議論を重ね、それに基づき各著者が改訂を行う。コメントづけによる議論は、

[†] (株)東芝 研究開発センター 情報・通信システム研究所 第二研究所
Research Lab. II, Communication and Information Systems Research Laboratories, Research and Development Center, Toshiba Corporation

必ずしも全員が同時に行う必要はなく、非同期的に行われる。こういったプロセスが文書が完成するまで長期にわたって続けられる。

このような作業過程では、複数のユーザ間で、相互に関連したマルチメディア情報や知識を長期間共有することが要求される。共有される情報は最終的な成果ばかりではなく、途中過程で生じる各種の断片的な情報（例えば、作成中の文書、文書になる前のメモ情報、文書に付加されたコメントなど）が含まれる。設計書、企画書などには、図表や絵が数多く含まれ、マルチメディア情報の共有が必要となる。そして共有された情報をもとに、ユーザ同士がコミュニケーションを行う手段が求められる。

こういった共同文書作成作業を支援するための環境を提供することを目的として、われわれはマルチユーザ・ハイパーメディアシステム MuHyme (Multi-user Hypermedia) を開発した^{3),4)}。MuHyme では基盤データベースとしてハイパーメディアを採用し、相互に関連するマルチメディア情報を長期間にわたって共有一元管理することを可能とした。またリンクづけ操作の一種として、ノードへのコメントづけを組み込むことにより、協調作業における基本的なコミュニケーション手段を提供している。MuHyme はノードに文字や図形、静止画、音声といったマルチメディア情報を含むことができるハイパーメディアであり、複数のユーザがマルチメディア文書を共有して、その文書に非同期的にコメントづけを行って議論することが可能となる。コメントもノードの一種として扱うため、コメントづけには文字ばかりではなく音声や画像を含むことができる。

ハイパーメディアをグループウェア化する試みはいくつか存在し^{5),6)}、Berlin らはマルチユーザ化における諸問題を整理している⁷⁾。われわれは MuHyme の設計にあたり、次の二つの問題に焦点を当てた。

- 長時間のセッションにおける共有データへのアクセス
 - 協調作業を効率化し、促進するためのメカニズム
- 一つ目の問題は、長時間トランザクションと呼ばれている⁸⁾。共同文書作成では、共有データの更新にかかる時間が長い。例えば文書編集に要する時間は通常数十分から数時間、時には数日に及ぶこともある。ロックを使った並行制御方式によれば、長時間一人のユーザがデータを占有してその間他のユーザはそれを参照することができず、作業の進行が妨げられる。そ

こで、長時間にわたる編集を考慮した共有データの更新モデルが必要となる。

もう一つの問題は、競合的な操作を互いに避けたり、他のユーザの結果を見てから自分の仕事を進めたりといった、協調的な作業の促進である。他ユーザの行動を随時知ることにより競合する作業の調整を促し、協調的なアクセスが可能となる。またコメントのやりとりによる議論では、同じ議論が繰り返されることが多い。グループ作業の効率化のためには、無駄な議論の繰り返しを防ぐ必要がある。

これらの問題はそれぞれ密接に関連している。MuHyme ではこれらを以下のように解決した。

- ハイパーメディアのノード更新に、チェックイン・チェックアウト操作に基づくデータ更新モデルを導入した。
- 上記のデータ更新モデルにバージョン管理機能を導入し、議論の経過を文書のバージョン（下書き）と関連づけて保存して、議論の繰り返しを防ぐようにした。
- 他ユーザの操作をセッション間メッセージにより、随時知らせる通知メカニズムを提供した。

ハイパーメディアをグループウェア化する試みとして、Intermedia⁹⁾では、コメントづけ操作を少ないステップで行うためのリンク機構を導入している¹⁰⁾。またgIBIS¹¹⁾ではハイパーテキストを利用して議論のモデルを作り、効果的に議論を管理する試みを行っている。PilotCard¹²⁾では、グループ作業を支援するためのデータベースに、レイヤ構造とハイパーメディアのリンク機能を導入している。いずれもハイパーメディアデータベースにおける並行制御や、協調作業における通知の問題には焦点が置かれていない。また共同文書作成システムとしては、Quilt^{13),14)}が先駆的であるが、ハイパーメディアのような一般的なデータ管理の枠組を導入していない。

Hoopertext⁷⁾では、ハイパーテキストをマルチユーザ化するにあたっての諸問題として、並行制御や通知に言及している。MuHyme は文書作成という応用において、これらの問題の一つの解決策を示したとすることができる。MuHyme の特徴は、ビジネス分野における共同文書作成という応用領域において、必要な技術进行分析し、ハイパーメディアとして統合的にシステム化したことにある。

MuHyme は実験システムの開発を終了し、利用評価を行っている。本論文では、第2章にて MuHyme

のハイパーメディア機能について概観する。第3章で MuHyme の特徴的なグループウェア機能であるデータ更新モデルと、通知機能について述べる。第4章で、それらの機能が実験システムにおいてどのように実現されているか説明する。最後に第5章で MuHyme の利用評価についてまとめる。

2. MuHyme の概要

2.1 グループ・データベースとしてのハイパーメディア

われわれは、グループの共有情報を管理するデータベースとして、ハイパーメディアを採用した。MuHyme では、作成中のマルチメディア文書を章・節・図といった論理構造の構成要素を単位として、ハイパーメディアのノード群に分解し、リンクづけによりネットワーク状に組織化する。

そしてコメントづけが共同文書作成における中心的な非同期コミュニケーションの手法であると考え、コメント自体もハイパーメディアのノードとして、データモデルの中に組み込んでいる。コメントづけは文書ノードからコメント・ノードへのリンクづけ操作に相当する。通常のノードと同じく、文字・音声・静止画をコメント・ノードの中を含むことができる。ハイパーメディア技術を適用することにより、文書とコメントを同じ枠組の中で、相互に関連して共有一元管理することが可能となった。

2.2 基本データモデル

図1に MuHyme の基本データモデルを示す。

基本オブジェクトは、ノード、リンク、アンカーであり、扱えるデータ型はテキスト、静止画、音声である。アンカーはノードにおけるリンクの出発点となるオブジェクトであり、一つのノードに複数存在する。リンクには、ノードの参照 (refer) と、ノードへのコメントづけ (annotate)、文書の構成 (structure) が存在する。基本操作として、ノードの生成、消去、リンクづけ、リンクの消去、リンクをたどる航行 (navigation) があり、それらは各オブジェクトのメソッドとして定義される。

マルチメディア情報として、テキスト、静止画、音声を扱えるようにするために図1に示したようなクラス階層を、ノードとアンカーに持たせている。各メディアに依存しない共通の操作を、上位のクラスで定

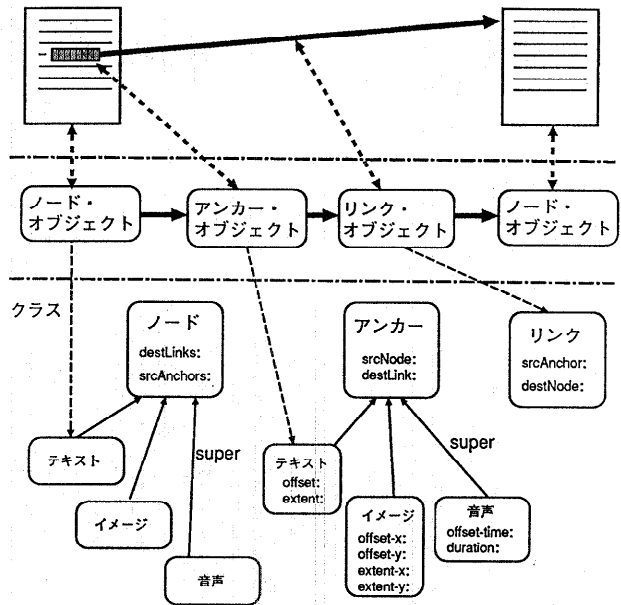


図1 基本データモデル
Fig. 1 Data model.

義すると同時に、各メディア操作のインターフェースが同一になるようなカプセル化を行った。

2.3 ユーザインターフェース

MuHyme の操作画面を図2に示す。各ノードにビューと呼ばれるウィンドウ・オブジェクトを対応させて、内容を表示する。ユーザはアンカー上でマウスボタンをクリックすることにより、次のノードへ航行することができる。各ビューにボタン、メニューが付随し、それをマウスで操作することによって第2.2節に述べたハイパーメディアの基本操作を実行する。またビューにはノード内容に対する編集機能が備わっている。

図2では右下にブラウザが表示されている。ブラウザはハイパーメディア・ネットワークの全体構造を表示する。ユーザはブラウザ上で適当なノードを指定することにより、そのノードにアクセスすることができる。

また図2では、音声によるコメントがつけられている。その内容は音声用のビューにて録音・再生できる。コメントのリンクづけ操作が終了すると、関連するユーザに通知が送信される。

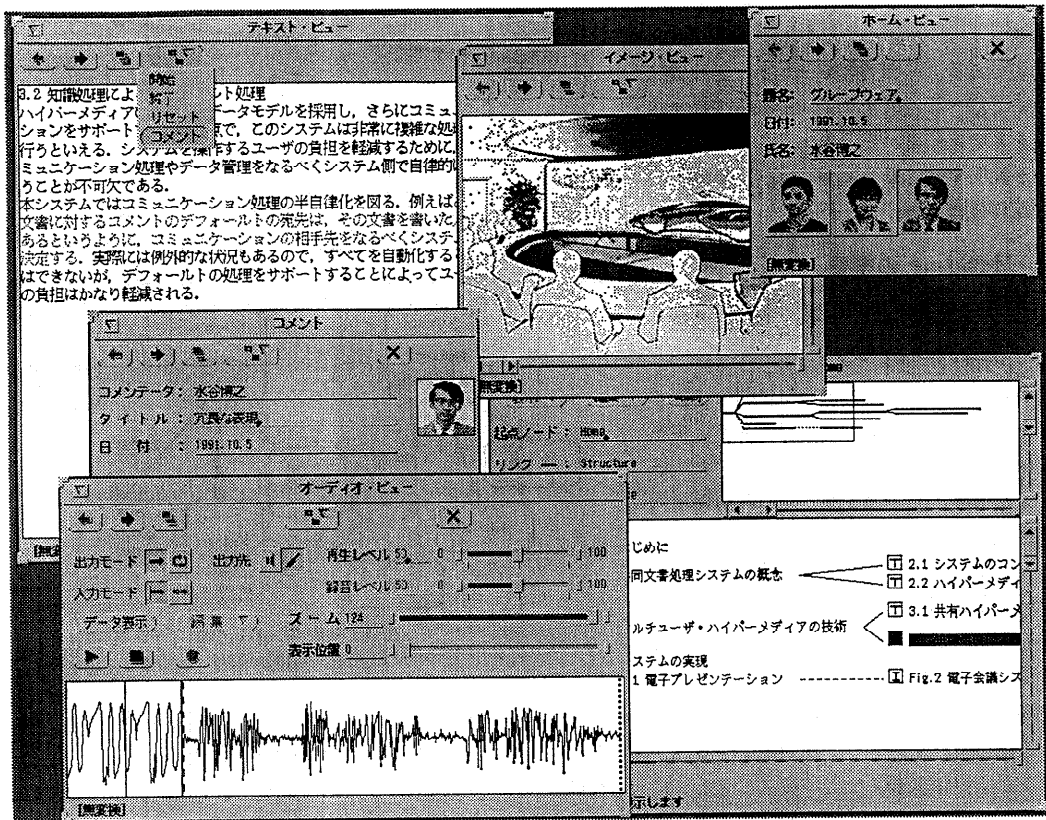


図 2 MuHyme の画面例
Fig. 2 User interface of MuHyme.

3. MuHyme のグループウェア機能

3.1 協調作業のための問題点

第 1 章で述べたように、ハイパーメディアを複数のユーザが協調的に利用できるグループウェアとするためには、以下の二つが問題となる。

- 長時間セッションにおける共有データへのアクセス：長時間編集時、他のユーザの作業進行を妨げないデータ更新モデルが必要である。
- 協調作業を効率化し、促進するためのメカニズム：他のユーザの行動を随時知ることによって協調的なアクセスを促進し、また議論の繰り返しを防いで作業の効率化を図る。

これらを MuHyme では以下の手段で解決した。

- ハイパーメディアのノード更新にチェックイン・チェックアウト操作に基づくデータ更新モデルを導入した。
- それとバージョン管理機能とを統合して、議論の経

過を作成中の下書きと関連づけて保存して、無駄な議論の繰り返しを防ぐようにした。

- 他ユーザの操作をセッション間メッセージにより、互いに通知し合うメカニズムを提供することにより協調的なアクセスを促している。

3.2 データ更新モデル

編集など長時間にわたるオブジェクトの更新に対応するために、MuHyme ではチェックイン・チェックアウト操作に基づくノードの更新と、そのバージョン管理を統合したデータ更新モデルを採用している。

3.2.1 チェックイン・チェックアウト操作による更新

図 3 にチェックイン・チェックアウト操作に基づくノード更新の概念図を示す。

ハイパーメディア・オブジェクトの記憶空間を、グループで共用する空間と個人作業用の空間に分割する。オブジェクトを編集する際には、グループ空間から必要なオブジェクトを適宜コピーして、個人作業空

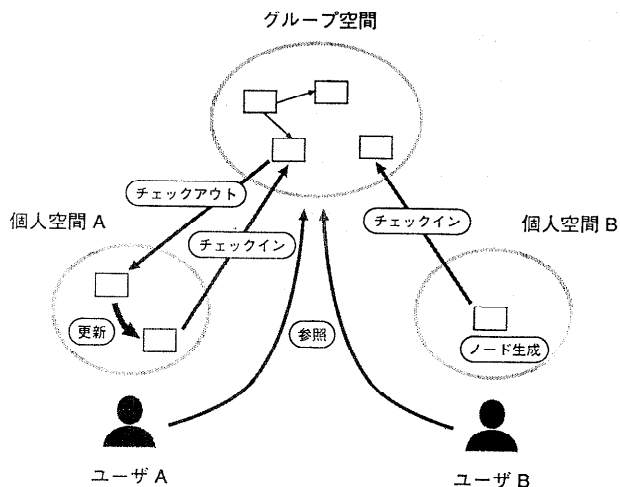


図 3 チェックイン・チェックアウト操作による更新
Fig. 3 Node updating with check-in/check-out operations.

間にチェックアウトしてからオブジェクトの変更を行う。そして個人作業空間で編集作業が終了した時点でそのオブジェクトをグループ空間に再びチェックインする。

共同文書作成作業においては、グループ空間には、いったん書き終えた下書き、つまり他のメンバにレビューしてもらった文書などを格納する。一方各ユーザの個人作業空間に格納されたデータは、そのユーザにのみアクセス権があり、現在作成中の下書き、文書になる前の断片的なメモ情報など、各ユーザが個人的に管理する情報が格納される。

チェックアウトされたグループ空間内のオブジェクトに対して、デフォルトでは更新ロックをかけない。したがってその間他のユーザはグループ空間上のこのオブジェクトを参照したり更新したりできる。これにより、長時間特定のオブジェクトが一人のユーザに占有されて、他のユーザがそれを参照できず作業が止まってしまう事態を避けることができる。

同じオブジェクトを二人以上のユーザが同時期に更新しようとした場合、第 3.3 節で述べる通知機能を用いて、あとからアクセスしたユーザにはそのオブジェクトが既にチェックアウトされて更新中であることが通知される。その通知に対して、そのオブジェクトを

あえて更新するか、あるいは参照するだけにとどめておくかは、ユーザの判断に任されている。

またたとえ二人以上が更新したとしても、第 3.2.2 項で述べるバージョン管理機能により、並列する分岐バージョンが作られ、グループ空間上にオリジナルのバージョンが残される。したがって元の状態に戻ることが可能である。さらに、同時に並列分岐するバージョンができないように、二人以上によるノードの同時更新を許さないような設定を、ユーザが行うことも可能である。

一方、リンクづけなどの短時間で終了する更新は、通常の一時的な更新ロックによって、グループ空間上のオブジェクトに対して直接行う。

3.2.2 バージョン管理

バージョン管理により、複数ユーザによる並行的な文書編集において元の文書を保存できる。さらにコメントのやりとりによる議論の過程を、文書のバージョンと関連づけて管理することができる。

図 4 に示すように、チェックイン・チェックアウト操作と統合する形でバージョン管理を導入する。たと

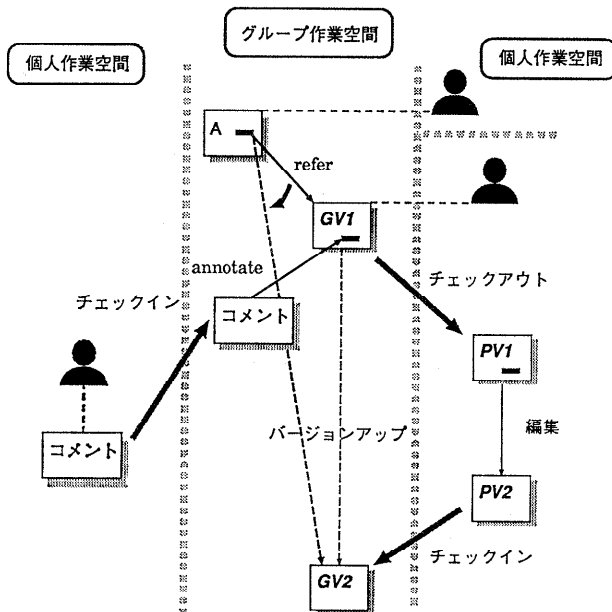


図 4 バージョン管理
Fig. 4 Version management.

例えばグループ空間のノード GV1 を個人作業空間にチェックアウトすると、PV1 という新たなバージョンが作られる。その後これを個人作業空間内で更新して PV2 としてから、グループ空間にチェックインすると、バージョン木をルートに向かってたどり、PV1 のオリジナルは GV1 であることを見つけ、GV1 から導出されたバージョン GV2 として、PV2 はチェックインされる。

またノードのバージョンが進んだ際に、リンク先の変更・維持も行われる。図 4 に示すように、refer リンクにより参照されたノードが更新された場合、refer リンクは新バージョンを指すように変更される。一方 annotate リンクによりコメントづけされたノードが更新された時は、リンク先の変更は起こらず、コメントノードからは旧バージョンを指したままとなる。

3.3 状態変化の通知

MuHyme では、協調作業を促進するためにユーザの行動を相互に通知するメカニズムを提供する。われわれはユーザの行動の通知を「共有されている状態変化の通知」という視点でとらえる。また通知先となるユーザをノードに対する「関わり (commitment)」に基づき決定する。

共有されているノードの編集、コメント付加など、グループ空間である共有データベースの状態が変化した際に、その由に関わりのあるユーザに通知する。共有データへのアクセスによる状態変化、ひいてはユーザの行動を別のユーザに知らせることにより、競合する操作を避けたり、競合が生じた場合の調整を促したりすることができる。

何を、どのタイミングで、誰に、どういふインタフェースで通知するかについて、MuHyme の通知モデルを以下に示す。

通知する情報は、他ユーザの行動の中で共有状態の変化を引き起こす操作であり、ハイパーメディアのノードの更新 (編集) とリンクづけである。MuHyme では以下の情報を通知メッセージとして、他のユーザに送信する。

- 操作 (ノードの更新、またはリンクづけ)
- 操作対象のノード
- 操作者
- 操作時刻

通知のタイミングは、操作の開始および終了時である。MuHyme では、チェックイン・チェック

アウト操作に基づいてノードの更新を行うため、チェックアウトとチェックインのそれぞれのタイミングで通知を行う。前者は競合アクセスの警告、後者は「更新した」という報告に相当する。

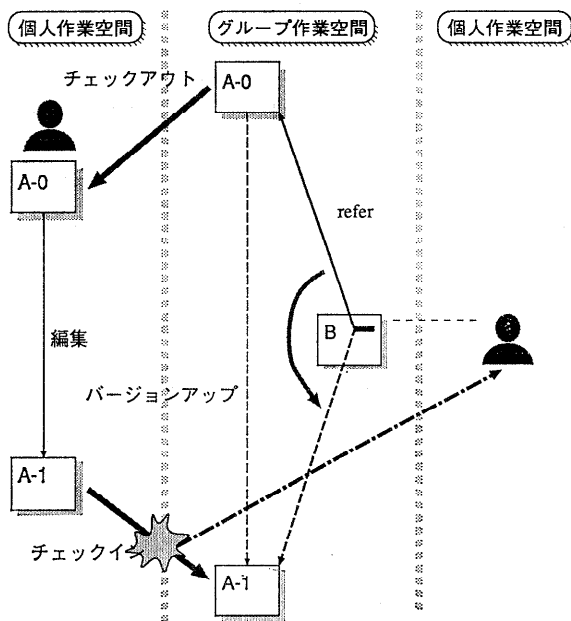


図 5 ノードの更新の通知
Fig. 5 Notification of node updating.

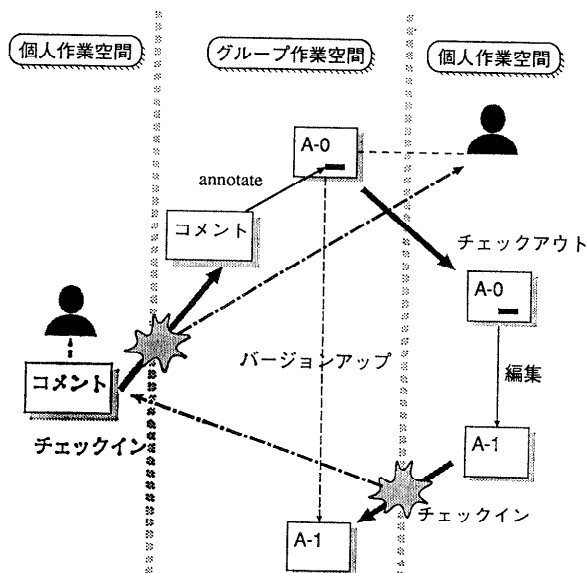


図 6 コメントづけの通知
Fig. 6 Notification of annotation.

通知の相手先は、操作対象であるノードに関わりのある人である。MuHyme では

- ノードの著者
 - ノードを参照しているノードの著者
 - コメントづけをしたユーザ
 - ノードを現在表示して読んでいるユーザ
- に関わりがあるとして、通知を行う。

図5に、ノードの更新による通知の様子を示す。ノードAを参照しているノードBの著者は、ノードAに関わりがあるため、ノードAが更新されるとその由が通知される。

また図6に、コメントづけに関わる通知の様子を示す。まずノードAへコメントづけが行われると、ノードAの著者に通知が行く。その際ノードAの内容にコメントした人は、ノードAに関わりが生じる。したがって後ほどノードAが更新されるとその由が通知される。ここではノードAの著者と、それにコメントした人に関わりが生じ、互いの操作が相互に通知されるため、結果的に議論が促進される。

4. MuHyme の実現

4.1 システムの基本構成

MuHyme の基本的なソフトウェア・アーキテクチャを、図7に示す。HAM (Hypertext Abstract Machine)¹⁵⁾と同様の記憶管理、データモデル操作、プレゼンテーション処理の3層からなる実現モデルを採用している。

記憶管理層ではハイパーメディアのノード、リンクを C++¹⁶⁾のオブジェクトとして扱い、それをオブジェクト指向データベース¹⁷⁾上に永続的に記憶する。またユーザインタフェースとなるプレゼンテーション処理層には、OpenWindows*を用いている。ハイパーメディアの各ノードに一つのウィンドウを対応させ、マルチウィンドウ上に表示する。MuHyme ではウィンドウをビューという C++ オブジェクトとして扱えるようにカプセル化を行っている。

データモデル操作層は、プレゼンテーション処理層と記憶管理層とを仲介するハイパーメディア操作のインタプリタに相当する。つまりプレゼンテーション処理層からの入力 (例えばマウスクリック) に基づき、

ハイパーメディア操作を実行して、記憶管理層中のノードの検索・更新を行う。

MuHyme のプロセス構成は、最下位の記憶管理層がハイパーメディアのデータ・サーバの役目を果たす

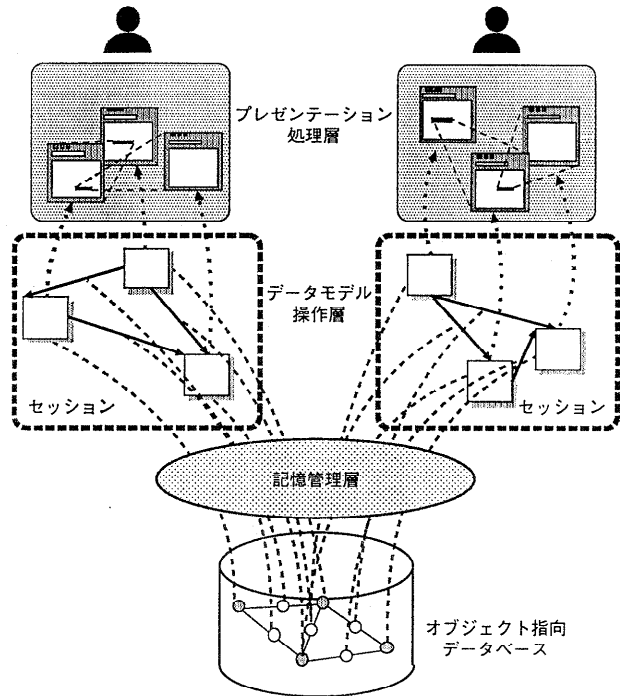


図7 MuHyme のアーキテクチャ
Fig. 7 Software architecture of MuHyme.

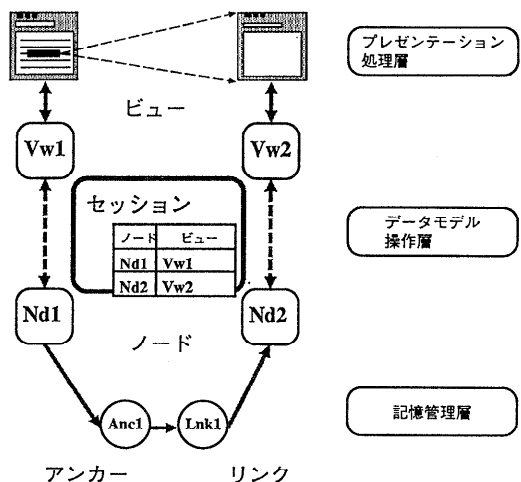


図8 セッションによるノードとビューの管理
Fig. 8 Nodes and views in a session.

* Open Windows は Sun Microsystems Inc. の登録商標である。

プロセスとなり、それにユーザごとのクライアント・プロセス（上位2層，データモデル操作層とプレゼンテーション処理層）がアクセスするクライアント・サーバ型の形態である。クライアントは，ネットワークを通してデータ・サーバにアクセスし，データを検索・更新する。

4.2 セッション

MuHyme ではデータモデル操作層に，セッションという概念を導入している。これは一人のユーザがある時点でアクセスしているハイパーメディアのノードとビューのオブジェクト集合を管理するものである。

セッションは，MuHyme のオブジェクト間のメッセージのやりとりを仲介する役目を果たす。セッションには現在アクセスしているノードとビューの対応関係を管理する表があり，実行に応じて動的に変化する。

ノードをはじめとするハイパーメディア・オブジェクトと，ビューオブジェクト，およびセッションの関係を，図8に示す。

ノードの航行を例にとりて，セッションの実行の様子を以下に示す。あるノードを表示しているビューからマウスによりアンカーを選択して，そのリンク先にあるノードをデータベースから検索し，別のビューに表示する場合，

1. ビューからアンカーを選択，対応するノードとア

ンカーを同定する。

2. 選択されたアンカーに followLink メッセージを送り，その先にあるリンクを得る。
3. 得られたリンクにさらに followLink メッセージを送り，リンク先のノードを得る。
4. 新しくビューを生成する。
5. 生成したビューに，リンク先のノードをマップして，ノードとビューの対応関係を表に登録する。
6. ビューを表示する。

という処理がセッションで実行される。

4.3 グループウェア機能の実現

MuHyme のデータ更新モデルと通知メカニズムは二つが統合する形で実現されている。オブジェクト指向データベースの基本バージョン管理機能上に，MuHyme のバージョン管理を導入している。そしてチェックイン・チェックアウト操作時に，リンク先の変更・維持と，関わりのあるユーザへの通知が実行される。通知先を探索するメカニズムを以下に示す。

第3.3節で示したように，関わりのある通知先は，ノードの著者，参照者のようにノード間のリンク構造によって静的に定まる場合と，現在アクセスしているユーザのように実行時に動的に決まる場合とがある。前者はリンクをたどることにより通知先を探す。後者については，セッションを利用して実行時に探索するメカニズムを実現した。

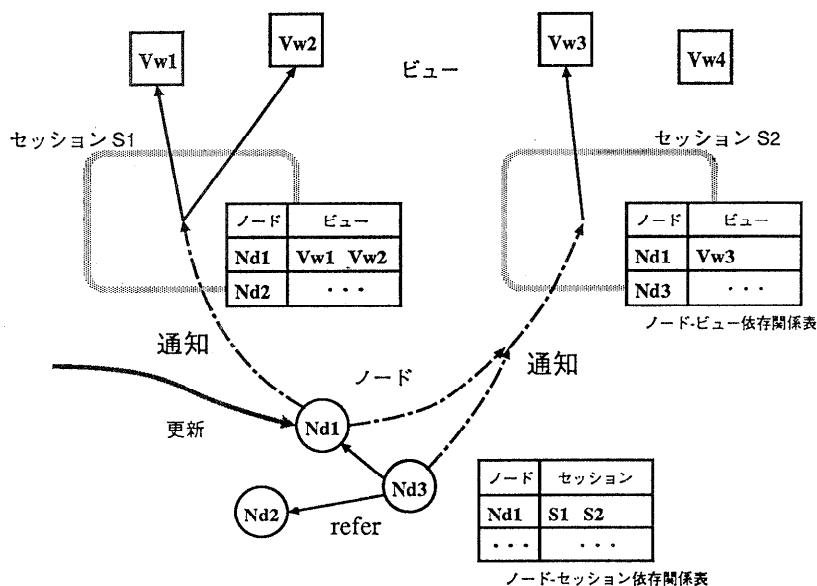


図9 状態変化の通知メカニズム
Fig. 9 The mechanism of notification.

通知における状態変化の伝播を実行時に管理するために、図9に示すような二つの表を用いる。一つは、セッション内のローカルな更新の伝播を管理する表で、セッションの中に存在する。この表には、あるノードの更新によって表示が影響を受けるビューが示されている。つまりそのセッションがローカルに管理するノードとビューの間の依存関係を示す表である(ノード-ビュー依存関係表)。これは先に述べたノードとビューの対応関係表を拡張したものである。もう一つの表は、ローカルセッションとリモートセッションの間の依存関係を表す。あるノードの更新によって影響を受けるセッションを、大局的に管理する(ノード-セッション依存関係表)。

あるセッションにてノードの更新が起こった際、それによって影響を受けるリモートのセッションがあるかを、ノード-セッション依存関係表により調べる。ない場合は、そのセッション内でローカルに影響を受けるビューを、ノード-ビュー依存関係表から調べて、ビューの表示の更新が行われる。影響を受けるリモートのセッションがある場合は、そのリモートセッションへのノードで更新があったかを通知する。その後リモートセッション内のノード-ビュー依存関係表を用いて、リモートセッションでのビューの更新が行われる。

送信した通知メッセージは、受信側のキューに保存される。現在受信側のユーザが MuHyme を使っているならば、直接画面を更新して通知し、そうでないときはシステム起動時にキューを見て通知が来ていればその由を伝える。

通知の基本メカニズムは、Unix の socket によるプロセス間通信により実現されている。セッションが通知メッセージを非同期に受けとる機構は、ウィンドウシステムがサポートするプロセス間通信イベントの監視機能を利用している。

MuHyme の別の実現方法として、ノードを一つのファイルとして扱い、Unix 上の SCCS などのバージョン管理システムと、メール機能とを組み合わせる手法も考えられる。しかしながら、

- ノード(ファイル)間のリンク管理、メールとの連動など、実装が複雑になる
- 同時にチェックアウトできないなどの制限があり、ユーザが使える機能が限定される
- ノード(ファイル)が増えると効率的なアクセスが困難になる

といったことを考慮して、オブジェクト指向データベース上にシステムを実現した。ノードをオブジェクトとして扱うことにより、ノードを単位とする操作(リンク管理、チェックアウト/チェックイン、バージョン管理)が容易に実現できた。またクラス階層に基づく継承関係を利用して、異種メディア間で共通の操作を上位クラスに定義することにより、効率的な実装を行うことができた。

5. 利用評価

MuHyme は実験システムの開発が終了し、現在社内で利用評価を行っている。その結果以下のような見が得られた。

- 通知は協同作業の効率化を図る上で有効である。通知しない場合は各ユーザは自主的にグループ空間を参照する必要があり、それを全員に徹底することは困難であった。
- 議論(コメント)の履歴を、文書のバージョンと共に保管することは有用である。これにより議論の繰り返しを防ぐと同時に、意見に対する責任が明確化された。
- 簡易なリンクづけ操作、音声によるコメントづけにより、議論が活発化した。単純なマウス操作によりコメントづけする部分を指定でき、かつ音声でコメントを入力できることは、気軽に意見を出すことを可能にした。これは特に管理者層、キーボード操作に慣れていないユーザに好評であった。

一方、以下が新たな課題として挙げられた。

- リアルタイム・コミュニケーション支援：共同文書作成においては個人作業が中心であるが、コメントと通知による非同期コミュニケーションだけでは不十分であった。たとえば、文書の構成・分担の変更、異なるバージョンのマージなどの作業では、同時に打合せをする機能が望まれた。
- 既存アプリケーションとの統合：MuHyme は一つの閉じたアプリケーションとして実装されており、内部に独自のエディタ機能、および既存 DTP (机上出版)とのインタフェースを持つ。しかし共同文書作成のためにわざわざ別のエディタを使いたくないとの要望が強かった。また DTP 処理後の文書のレイアウト表現についてもコメントをつけたいという要求が出た。

これらの要求の解決に向けて、共有ウィンドウによる同時コミュニケーションの導入、汎用性のあるア

アプリケーション・インタフェースへの拡張を検討している。

今回評価対象とした文書は、個人の分担が比較的明確な企画書が中心だったため、意見の相違から複数のバージョンが同時に生じるということは、あまり頻繁には起こらなかった。一般の文書作成においては、異なるバージョンが同時に生じた場合、CASE で行われているように内容に立ち入って、その対処をシステムが支援することは難しい。われわれは異なるバージョンのマージのために、同時コミュニケーションの導入をまず検討しているが、バージョンの相違をユーザにわかりやすく示す特殊な diff¹⁸⁾のようなツールも必要になると考えられる。またより分担が明確でない文書についての MuHyme の利用評価は、将来の課題としたい。

6. おわりに

本論文では、ハイパーメディア技術を基盤として、共同で文書作成するユーザ間の非同期型コミュニケーションをサポートするグループウェア MuHyme について、その特徴的なグループウェア機能の設計と実現について述べた。

MuHyme を利用することにより、共著者間でマルチメディア情報を長期間共有管理し、コメントづけを通して非同期的に議論することが可能となった。そしてチェックイン・チェックアウト操作とバージョン管理を導入して、長時間にわたる編集作業により他のユーザの作業の進行が妨げられないようにすると同時に、議論の繰り返しを防いでグループ作業の効率化を行った。また他のユーザの操作を通知し合うメカニズムを提供することにより、情報への協調的なアクセスを可能とした。

今後、MuHyme の実際の使用経験を重ね、そこで得られた知見をもとにさらなる改良を進めていく予定である。

参 考 文 献

- 1) Ellis, C. A., Gibbs, S. J. and Rein, G. L.: Groupware Some Issues and Experiences, *Communications of the ACM*, Vol. 14, No. 1, pp. 38-58 (1991).
- 2) 石井 裕: コンピュータを用いたグループワーク支援の研究動向, *コンピュータソフトウェア*, Vol. 8, No. 2, pp. 14-26 (1991).
- 3) 守安 隆: マルチユーザ・ハイパーメディア・システム, *ヒューマン・インタフェース講習会資料*, pp. 77-82, 計測自動制御学会, ヒューマン・インタフェース部会 (1991).
- 4) 村永哲郎, 守安 隆, 友田一郎: ハイパーメディアを用いたグループウェア MuHyme の設計と実現, *情報処理学会グループウェア研究グループ研究会*, Vol. GW-1, pp. 59-66 (June 1992).
- 5) Trigg, R. H., Suchman, L. A. and Halasz, F. G.: Supporting collaboration in NoteCards, *Proceedings of the 1986 Conference on Computer Supported Cooperative Work (CSCW '86)*, pp. 153-162 (1986).
- 6) Yoder, E., Akscyn, R. and McCracken, D.: Collaboration in KMS, a Shared Hypermedia System, *Proceedings of the Conference on Human Factors in Computing Systems (CHI '89)*, pp. 37-42 (1989).
- 7) Berlin, L. M. and O'Day, V. L.: Platform and Application Issues in Multi-user Hypertext. Gibbs, S. and Verrijin-Stuart, A. A. ed., *Proceedings of the IFIP WG 8.4 Conference on Multi-User Interfaces and Applications*, pp. 293-309 (1990).
- 8) 西尾章治郎: オブジェクト指向データベースにおける並行処理制御, *情報処理*, Vol. 32, No. 5, pp. 540-549 (1991).
- 9) Meyrowitz, N.: Intermedia: The Architecture and Construction of an Object-oriented Hypermedia System and Applications Framework, *OOPSLA '86*, pp. 186-201 (1986).
- 10) Catlin, T., Bush, P. and Yankelovich, N.: InterNote: Extending a Hypermedia Framework to Support Annotative Collaboration. *Hypertext '89 Proceedings*, pp. 365-378 (1989).
- 11) Conklin, J. and Begeman, M. L.: gIBIS: A Hypertext Tool for Exploratory Policy Discussion, *ACM Transaction on Office Information Systems*, Vol. 6, No. 4, pp. 303-331 (1988).
- 12) 市村 哲, 松浦宣彦, 岡田謙一, 松下 温: レイヤ構造と PilotCard 機構に基づく協同作業支援データベース, *情報処理学会論文誌*, Vol. 33, No. 9, pp. 1152-1160 (1992).
- 13) Fish, R. S., Kraut, R. E. and Leland, M. D. P.: Quilt: A Collaborative Tool for Cooperative Writing, *Proceedings of the Conference on Office Information Systems (COIS '88)*, pp. 30-37 (1988).
- 14) Leland, M. D. P., Fish, R. S. and Kraut, R. E.: Collaborative Document Production Using Quilt, *Proceedings of the 1988 Conference on Computer Supported Cooperative Work (CSCW '88)*, pp. 206-215 (1988).
- 15) Campbell, B. and Goodman, J. M.: HAM: A General Purpose Hypertext Abstract Machine, *Communications of the ACM*, Vol. 31, No. 7, pp. 856-861 (1988).

- 16) Ellis, M. A. and Stroustrup, B.: *The Annotated C++ Reference Manual*, Addison-Wesley (1990).
- 17) Cattell, R. G. G.: *Object Data Management: Object-Oriented and Extended Relational Database Systems*, Addison-Wesley (1991).
- 18) Neuwirth, C. M., Chandhok, R., Kaufner, D. S., Erion, P., James, M. and Miller, D.: *Flexible Diff-ing in a Collaborative Writing System, Proceedings of the Conference on Computer-Supported Cooperative Work (CSCW '92)*, pp. 147-154 (1992).

(平成4年10月28日受付)

(平成5年4月8日採録)



村永 哲郎 (正会員)

1961年生。1984年東京大学工学部計数工学科卒業。同年(株)東芝入社。以来、研究開発センター情報・通信システム研究所において、知識処理システム、グループウェアの研究開発に従事。共訳書「メンタルモデル」(産業図書)。ACM, 人工知能学会各会員。



守安 隆 (正会員)

1954年生。1977年京都大学工学部精密工学科卒業。1982年同博士課程修了。1983年東京芝浦電気(株)(現、東芝)入社。以来、研究開発センター情報・通信システム研究所において、AI, グループウェアの研究開発に従事。1987-89年 Edinburgh 大学 AIAI に駐在研究員。IEEE, 人工知能学会, 計測自動制御学会各会員。



友田 一郎

1964年生。1987年慶応義塾大学理工学部電気工学科卒業。1989年同修士課程修了。同年(株)東芝入社。以来、研究開発センター情報・通信システム研究所において、オブジェクト指向言語, グループウェアの研究開発に従事。日本ソフトウェア科学会会員。



水谷 博之 (正会員)

1947年生。1971年名古屋大学理学部数学科卒業。1973年同修士課程修了。同年東京芝浦電気(株)(現、東芝)入社。以来、システム工学, 知識処理, グループウェアの研究開発に従事。現在、研究開発センター情報・通信システム研究主任研究員。人工知能学会, AAAI 各会員。