

## マルチメディア分散在席会議システム (MERMAID) を利用したグループアプリケーションの分散協調制御 方式の提案

阿部 豊子† 前野 和俊†  
阪田 史郎† 福岡 秀幸†

本論文では、マルチメディア分散在席会議システム (MERMAID) の提供機能を利用したアプリケーションプログラムの分散協調制御方式とその実現例について述べる。MERMAID は、地理的に離れた利用者が自席の WS から会議を開催するなど、広域多者間のグループ協同作業を支援する。参加者は、動画、音声の共有に加え、マルチメディア文書をリアルタイムで共有、編集できる。1989年3月に開発し、北米を含む遠隔10地点間で利用している。マルチメディア情報の共有はグループウェアにとって最も基本的な機能であるが、さらに多様な利用形態に対応するにはアプリケーション共有 (AP 共有) を実現する必要がある。本論文では、従来のイベント同報機能に加え、AP 情報同報によりシングルユーザ AP の共有における外部プロセスとのデータ交換を実現し、さらに MERMAID が提供する広域多者間の通信機能、データ同期機能、情報操作権の排他・移行制御機能などグループ管理機能をすべての AP に共通に提供する方式を提案する。提案方式を具体的な AP に適用することによりその有効性を示した。

### Distributed Cooperative Control for Sharing Applications Based on MERMAID (Multimedia and Multiparty Desktop Conferencing System)

TOYOKO ABE,† KAZUTOSHI MAENO,† SHIRO SAKATA† and HIDEYUKI FUKUOKA†

This paper proposes a control scheme for sharing applications, based on MERMAID, and presents prototype examples to which the proposed scheme is applied. A distributed multiparty desktop conferencing system, called MERMAID, was developed for supporting a wide range of group cooperative work in early 1989. It allows a group of users to conduct a meeting from their desktop workstations. This system has been used in our daily research work between ten remote locations. Sharing applications in a distributed environment is highly effective for giving participants a tool more suited to their work. Proposed scheme consists of the events distribution and the application specific-messages distribution, and maintains the data consistency among all sites. The application specific-messages distribution enables data access to external resources. Group management functions required in sharing applications are commonly supported to group applications, using functions provided MERMAID. The proposed scheme is applied to several applications, showing that it provides an effective platform for sharing applications.

#### 1. はじめに

企業や組織の分散化、国際化により人や情報が分散し、高度情報化社会に対応するために、業務の複雑化、多様化が進展している。それに伴い、グループで遂行する作業の重要性が増大してきている。コンピュータやネットワークを有機的に結合することにより、地理的に離れた複数の利用者間で情報を共有、交

換し互いに協力しながら各種の問題解決や業務遂行を図るグループウェア、CSCW (Computer-Supported Cooperative Work) の実現要求が高まっている<sup>1)~3)</sup>。

マルチメディア分散在席会議システム: MERMAID は通信技術と情報処理技術とを統合し、地理的に離れた複数の人達が自席の WS から会議に代表される多様なグループ協同作業を支援する<sup>4)~6)</sup>。1989年3月に開発し、約3年半の間、NEC 北米研究所 (米国、ニュージャージー州、プリンストン) を含む日米の10地点間で技術会議、分散ソフトウェア開発、分散シミュレーションなどに使用している。参加者は、音声、動

† 日本電気(株) C&C システム研究所ネットワーク研究部  
Network Systems Research Laboratory, C&C  
Systems Research Laboratories, NEC Corporation

画に加え、マルチメディア文書をリアルタイムで共有、交換できる。グループウェアでは、このようなマルチメディア情報の交換、共有に加え、より対象業務に適したグループ協同作業環境を提供するために、グループ（複数人）でアプリケーション（AP）を同時に利用する方式（アプリケーション共有方式）の確立が重要な課題となっている。

アプリケーション共有（以後、AP 共有）とは、グループ協同作業において、ある参加者が AP へ入力した時、その入力に対する実行結果をすべての参加者がリアルタイムに共有することである（例えば、文献7）～11）。AP の処理結果に関し、その結果を時々刻々同期を取る場合もあれば、あるまとまった処理後にその結果の同一性が保証されればよい場合もある。AP 共有により、参加者が同じスプレッドシートを見ながら計算を行ったり、エディタを共有して、同時に協同で文書を作成するなど、様々な機能を柔軟に参加者へ提供することが可能になる。

共有の対象となる AP には、単一ユーザ用に開発されたシングルユーザ AP と複数人でリアルタイムで共有されることを前提として開発されたマルチユーザ AP（以後、グループ AP と呼ぶ）がある。シングルユーザ AP の共有は、分散方式と集中方式に大別されるが、集中方式では応答性の悪さ、サイトごとに表示画面の自由度を持っていないこと（全員が全く同じ画面しか見ることができないこと）<sup>9)</sup>、また分散方式においては外部プロセスとのデータ交換の実現がそれぞれ問題となっていた<sup>7)</sup>。一方、グループ AP は複数の AP を同じグループで同時に共有する場合でも、グループメンバの招集、論理的通信路の設定などのグループ管理を個別に AP が行う必要があることが問題であった<sup>12),13)</sup>。本論文では応答性やサイトごとの表示情報の自由度の高さから分散方式を採用し、従来の研究において未解決であった

- シングルユーザ AP の共有における外部プロセスとのデータ交換の実現
- グループ AP に対するグループ管理機能（招集、論理的通信路設定、参加者管理など）の統一的な提供

を同時に解決する AP 共有制御方式について提案する。従来のイベント同報に加え、新たに AP 情報同報機能を提供することにより外部プロセスとのデータ交換を可能にし、サイト間の一貫性制御を行う。イベントはユーザ操作によって OS が発生する情報であ

り、AP に依存しない。AP 情報は AP の提供機能に依存した情報である。さらに、既に開発されている MERMAID の通信制御に関する基本機能を AP から利用可能にすることにより、グループ管理機能がすべての AP に共通に提供できる。

2章では従来の AP 共有方式について述べることによって、本論文で解決する問題を明らかにし、3章では本論文で提案する AP 共有方式を、4章ではその適用例について述べる。

## 2. アプリケーション共有方式

AP 共有方式は、既存のシングルユーザ AP を共有する、もしくはグループ AP の開発を支援するの2つに分類される。既存のシングルユーザ AP を共有するシステムは、AP が起動されている場所によって、集中方式と分散方式に分けられる。

集中方式では、AP を一箇所で起動し、出力をすべてのサイトに同報することにより参加者が同じ画面を共有する<sup>9)~11)</sup>。参加者間の表示画面の一貫性制御が容易であるが<sup>14)</sup>、入力情報に比べ表示情報は、特にマルチメディアを扱う AP の場合データ量が大きくなるため、ネットワークトラフィックが大きく、応答時間が長くなるという問題がある。

分散方式では、各サイトで同一 AP がローカルに起動され、ある参加者の入力情報をすべてのサイトに同報し、各サイトでローカルに処理を行うため、ネットワークトラフィックが小さく、応答時間も短い<sup>7),8),10)</sup>。各サイトでは AP 管理サーバが存在し、AP へのイベントを横取りし、同報する。例えば、MMConf<sup>8)</sup>では、Conference Manager が参加者ごとに1つ存在し、イベント同報制御を行っている。しかし、単なるイベント同報では制御できない処理として他 AP とのカットアンドペーストや、データベースへの多重アクセスなど外部プロセスとのデータ交換の制御が課題として残っている<sup>7)</sup>。

シングルユーザ AP 共有方式は、普段利用している AP の画面を参加者間で共有することが可能になるため、多種のグループ協同作業に対応することができる。グループ AP の開発の場合、最初から複数人で利用することが前提となっているため、個人ごとに異なる画面を見るような協同作業を支援する AP を作成することも可能である。グループ AP の開発に関しては、グループエディタを作成するためのツールキットとして DistEdit<sup>12)</sup>、AP 作成用の言語として、

RENDEZVOUS<sup>13)</sup>が提案されている。いずれもグループ生成機能、論理的通信路設定などグループ管理機能も AP ごとに実装する必要があるため、2つ以上の AP を同時に同じグループで利用する場合、参加者の招集などを AP ごとに行わなければならない。つまり、文書エディタと表計算ソフトを同時にグループで共有する場合、文書エディタで参加者を呼びだし、さらに表計算ソフトでも参加者を呼びださなければならぬという欠点がある。

柔軟にグループ協同作業に対応するためには、シングルユーザ AP のグループ AP 化と最初からグループ AP を構築することの両方を支援する必要がある。本論文では、シングルユーザ AP のグループ AP 化支援から AP 共有を検討するが、その際、シングルユーザ AP への修正は AP の提供機能に依存せず、簡単に行え、グループ AP へも適用することを考慮する。シングルユーザ AP 共有方式には、分散方式と集中方式があり、ネットワークトラフィックと一貫性制御のトレードオフのみで分散/集中が比較されてきた<sup>14)</sup>。さらに、サイトごとのビューの自由度(サイトごとに異なった表示を見る)について比較する。集中方式の場合、一箇所で共有する AP を起動し、同一の表示情報を同報するため、すべての参加者は同じ画面を見ることしかできない。一方、分散方式は、同一の入力情報を同報し、各サイトで処理を行うため、すべての参加者が同じ画面を共有するだけではなく、サイトごとに異なった処理を行わせ、違う画面を提供することも可能である。提案方式では、サイトごとの表示情報の自由度が高く、遠隔リアルタイム型の協同作業を支援する上で重要となる高い応答性を得るようにするために分散方式を採用する。提案方式の特徴として、

- イベント同報のみで制御を行う従来の分散方式の外部プロセスとのデータ交換の問題を、AP 情報同報機能新たに加えることによって実現する。
- MERMAID を利用したグループ AP 管理機能により AP 共有を実現し、すべての AP を一元的に管理する。

があげられる。以下では具体的な実現方法について述べる。

### 3. MERMAID の機能を利用した アプリケーション共有制御

分散方式では、すべてのサイトの AP の処理の一貫性を維持するために、次の3つの同期制御の問題を

解決しなければならない<sup>7),8)</sup>。

- AP の初期状態の統一
- ユーザ入力の同期
- 外部プロセスとのデータ交換制御

この章では、上記の問題を解決し、MERMAID の機能を利用した AP 共有制御方式を提案する。

#### 3.1 同報情報の比較

分散方式では、ユーザ入力を各サイトで起動されているグループ AP に送る必要がある。この時、どのような情報をどのようなタイミングで送るかが問題となる。この節ではシングルユーザ AP を構造化することにより、グループ AP の構造を明らかにし、同報する情報の内容とタイミングの問題について検討する。

既存のシングルユーザ AP は一時点においては一人で利用されることを前提として作成され、ユーザ入力は各 WS でローカルに処理される。シングルユーザ AP を論理的に、OS、イベントディスパッチャ、イベントプロセッサの3つのモジュールに分割する(図1)。OSはユーザ入力で発生したイベントを各アプリケーションキューに入れる。イベントディスパッチャはアプリケーションキューからイベントを取り出し、イベントプロセッサに送信する。イベントプロセッサは、受けとったイベントを実行する。同報は各部において行うことが可能であるが、表1にその比較を示す。

(a) OS での同報:

ユーザ操作によってイベント(キーボードイベント、マウスイベントなど)が発生した時に、すべてのサイトにそのイベントを同報する。

(b) イベントディスパッチャでの同報:

ユーザ操作によってイベントが発生した時、アプリケーションキューから取り出したイベントを同報する。

(c) イベントプロセッサでの同報:

イベントプロセッサによって実行される処理に

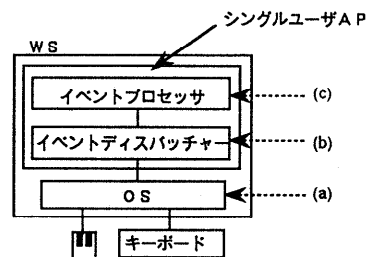


図1 シングルユーザAPの構造  
Fig. 1 Single user application structure.

表 1 同報通信機能の比較  
Table 1 A distribution comparison.

| ユーザ操作の同報通信機能    | AP への変更 | OS への変更 | 同報するメッセージ    | 特徴  |
|-----------------|---------|---------|--------------|---|
| (c) イベントプロセッサ   | 大       | 無       | AP 独自のメッセージ  | イベントを解析後、メッセージを同報するため、共有 AP 間の同期制御が容易である。         |
| (b) イベントディスパッチャ | 小       | 無       | OS で発生するイベント | イベントを解析せずにメッセージを同報するため、AP への変更点が様々な AP に対して共通である。 |
| (a) OS          | 無       | 有       | OS で発生するイベント | OS がメッセージを同報するため、シングルユーザ AP を変更せず共有できる。           |

必要な情報 (AP 情報) を作成し、すべてのサイトに同報する。AP 情報とは AP ごとに定義されたものである。

OS を変更する方法 (a) は、既存のシングルユーザ AP を修正なしにグループで共有できるが、OS に特殊なインタフェースを設けることになり、OS の汎用性を失わせ、実用的ではない。イベントディスパッチャで同報する方法 (b) は、AP の提供機能に依存しない修正によって共有が可能である<sup>15), 16)</sup>。イベントプロセッサで同報する方法 (c) は、AP の提供機能 (処理内容) に特化した情報を同報できるので、グループ AP 独自の動作を行うことが可能で、一貫性の維持が容易である。イベントディスパッチャにおいてイベントを同報することをイベント同報、イベントプロセッサで AP 固有の情報を同報することを AP 情報

同報と呼び、これらにより AP 共有を実現する。

3.2 AP 共有制御

図 2 に MERMAID の提供機能を利用したグループ AP 共有制御方式を示す。イベント同報部は同一の入力イベントが各サイトの AP で同一順で処理されることを保証する。AP 情報同報部は、AP の処理に依存した AP 情報 (処理指示、処理データなど) を作成し、各サイトの AP が同一の実行結果を得ることを保証する。MERMAID の AP マネージャは、MERMAID の基本機能を AP が利用するためのインタフェースを提供する。

3.2.1 MERMAID の提供機能

MERMAID は、参加者間で音声、動画の共有に加え、共有画面上でのマルチメディア文書をリアルタイムで交換・編集することによって遠隔地間のグループ

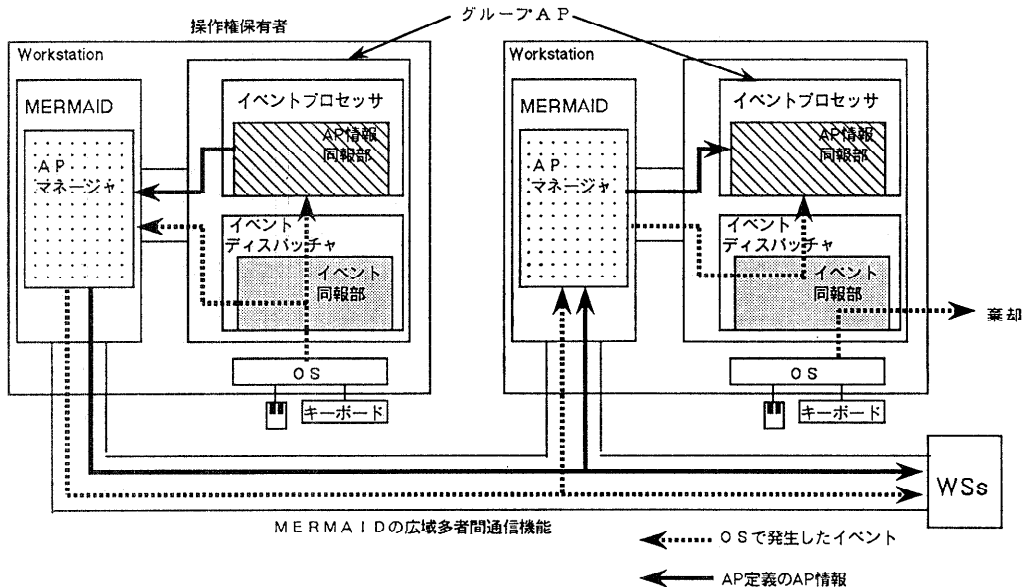


図 2 MERMAID における AP 共有制御方式  
Fig. 2 Sharing application control scheme based on MERMAID.

協同作業を支援している。その基本機能として、

- 広域多者間のリアルタイム同報通信機能
- 共有画面の一貫性制御機能（操作権移行制御など）
- グループ管理/運営機能（グループ生成、情報管理）
- ファイル管理/転送機能
- 映像・音声制御機能

を提供している<sup>9)~11)</sup>。これらはグループ AP に共通のグループ管理機能であり、AP 共有の基盤となる。グループ AP に対してこれらの機能を利用できるようにすることによりグループ AP の構築を容易にする。

### 3.2.2 AP マネージャ

AP マネージャの役割は、

- MERMAID の通信制御に関する基本機能（グループ管理、広域多者間通信機能、操作権移行制御）をグループ AP に提供する、
- イベント同報、AP 情報同報に関する AP 共有制御機能を提供する、

ことであり、以下の機能を有している。

- グループ AP の起動/管理：すべてのグループ AP は AP マネージャから起動され、グループ AP は AP マネージャとセッションを張る。AP マネージャは起動 AP と通信路の対応管理を行い、グループ AP はこのセッションを通じて MERMAID の提供機能を利用する。
- 広域多者間の情報通信：AP 共有に関して送受信されるメッセージには制御情報/イベント情報/AP 情報が存在する（図 3）。MERMAID の通信制御に関する基本機能（グループ管理・広域多者間通信機能、操作権制御）により、これらの情報のサイト間の送受信を行う。
- 制御情報処理：制御情報とは、AP マネージャ間でのみ送受信される情報であり、AP の起動/終了要求、操作権移行情報がある。操作権は AP ごと

に存在し、AP の操作権移行制御は MERMAID で一元的に管理されている。AP マネージャは操作要求や操作権情報の問い合わせを MERMAID に対して行う。

- イベント同報処理：受信イベントが送信元と対応している AP で処理されることを保証するために、AP ごとに受信イベントの管理を行う。
- AP 情報同報処理：受信した AP 情報に対応するグループ AP の処理を駆動する。また外部プロセスとのデータ交換の制御として、AP 情報の受信処理、イベントの再送信処理、データのグループ AP への送信などの処理を行う。詳しくは、3.2.5 項「AP 情報同報」で述べる。

MERMAID に対して新たに AP マネージャを設けることによって、グループ AP の一元的な管理が可能になる。また、利用者の管理や参加者の招集などは MERMAID が行うため、AP ごとにグループ管理機能を実装する必要がなくなる。

### 3.2.3 イベント同報

イベント同報部は、各サイトのグループ AP が同じ順番で、同一の入力イベントを受けとり、実行することを保証するために以下の機能を提供する。

- (1) 多地点間リアルタイムの入力イベント同報機能：MERMAID の広域多者間情報通信機能で実現する。

- (2) 入力イベントの分類：

分散方式による AP 共有システムでは、AP へのイベントをすべて同報しているが<sup>7), 8), 10)</sup>、サイト間のグループ AP の一貫性を維持するためには、すべての入力イベントを同報する必要はない。グループ AP の同期をとるために送受信する必要があるグローバルイベントと自 WS でのみ処理されるローカルイベントの 2 つにイベントを分類する。例えば、ある参加者がマウス操作によって

スクロールをした場合、マウスイベントはグローバルイベントであり、その結果生じる再描画イベントはローカルイベントである。

- (3) イベントの順序制御：すべてのサイトのグローバルイベントを同時に交換すると、各サイトで異なる順序で届く可能性がある。

通信メッセージ種別

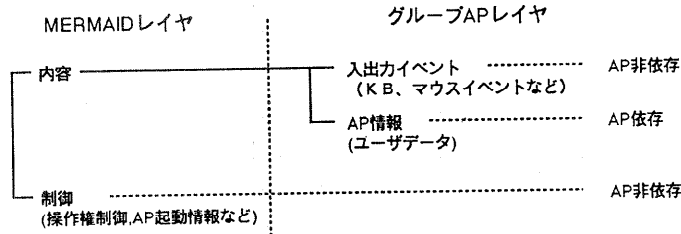


図 3 AP 共有制御におけるメッセージ体系

Fig. 3 Message hierarchy in sharing applications.

グローバルイベントの順序制御のために操作権を設ける。操作権保有者のグループ AP においてイベントが発生すると、イベント同報部はグローバルイベントをすべてのサイトへ同報する。操作権を保有していない場合、自 WS で発生したグローバルイベントを処理の対象から外す。操作権の移行制御は MERMAID の機能を利用する。

#### (4) ウィンドウ管理：

一般に OS は、複数存在するウィンドウをローカルなウィンドウ識別子 (WinID) を割り当てることによって管理し、自 WS で起動されている AP へのイベントを取り扱うのに利用している (例えば、UNIX における X-Window システム)。WinID は同じ AP が生成したウィンドウであってもサイト (WS) ごとに異なる。グローバルイベントがすべてのサイトで同等のウィンドウで処理されることを保証するために、サイト間で一意な (グローバルな) ウィンドウ名 (Win-Name) を設ける。イベント同報部はグループ AP の各ウィンドウにグローバル WinName を割り当て、ローカルな WinID との対応を管理する。

#### 3.2.4 イベント同報の限界

イベントは AP が処理を行うためのトリガであり、そのイベントによって行われる処理は AP の提供機能によって異なっている。イベントを同報しただけでは制御できない処理である外部プロセス (他 AP、データベース、周辺機器など) とのデータ交換の問題が依然として残っている<sup>7), 8)</sup>。例えば、マウス操作によってイメージを読み込む AP を参加者 A, B, C の 3 人で共有する場合、参加者 A がイメージ 1 を読み込むために、マウスでメニューを選択したとする。マウスイベントが同報されることにより B, C でもイメージの読み込み動作が行われるが、イメージ 1 は参加者 A しか持っていないため、B, C ではイメージ 1 を読み込むことができずサイト間の一貫性が崩れてしまう。また、グループ AP への要求である個人 AP とグループ AP 間のデータ交換<sup>9)</sup>においても同様に一貫性が維持できない。つまり、各サイトの AP が同じイベントを受けとり、同じ処理を行うだけでは十分ではなく、AP の処理内容に対応した制御が必要となる。次の項では、イベント同報に加え、AP の処理に対応した AP 情報を同報することにより、一貫性制御を行う方法について述べる。

#### 3.2.5 AP 情報同報

グループ AP での外部プロセスとのデータ交換について、シングルユーザ AP の拡張の点から述べる。個人的に起動している AP の画面 (例えば、Xterm) 上のデータをグループ AP へコピーする場合を例として説明する。図 4 に従来のイベント同報で個人 AP からのコピーを行った場合の処理シーケンスを示す。イベント同報の場合、

- (1) 個人 AP においてコピー処理が行われる。
- (2) グループ AP においてペースト操作が行われ、イベントが発生する。
- (3) イベント同報によって、ペースト処理のトリガイベントがすべてのサイトに同報される。
- (4) 各サイトで個人 AP へデータ要求が行われる。の処理シーケンスにおいて、イベント同報以外はシングルユーザ AP のオリジナルの処理である。AP 共有のためにイベント同報機能 (3) を追加しても、個人 AP において選択した領域のデータは他のサイトのメモリ上には存在しない。そのため、他のサイトではデータを取り出すことができず、一貫性が崩れる。以下に示す汎用的な関数を、シングルユーザ AP の外部プロセスへのデータ要求処理 (AP のオリジナル処理) に付加することにより制御を行う (図 5)。

##### ●操作権チェック関数

操作権は各 AP ごとに存在し、自サイトが AP に対して操作権を有しているかどうかをチェックする関数である。操作権の移行制御・管理は MERMAID のグループ管理機能を利用している。

##### ●AP 情報同報関数

操作権を保有している時に、AP 情報を同報するための関数である。この関数は AP マネージャに対して要求を行うもので、実際は AP マネージャが MERMAID の通信機能を利用して、他サイトに同報し、一貫性の制御を行っている。関数のパラメータとして、外部プロセスへのアクセスのトリガとなるイベント、送信データ (AP 情報) が含まれる。

##### ●AP 情報受信関数

AP マネージャに他サイトから同報されてきたデータ (AP 情報) を要求する関数である。AP マネージャにデータが存在している場合はデータが、存在しない場合は NULL が返される。

図 6 に AP 情報同報による外部プロセスとのデータ交換の制御シーケンスを示す。操作権保有サイトが、

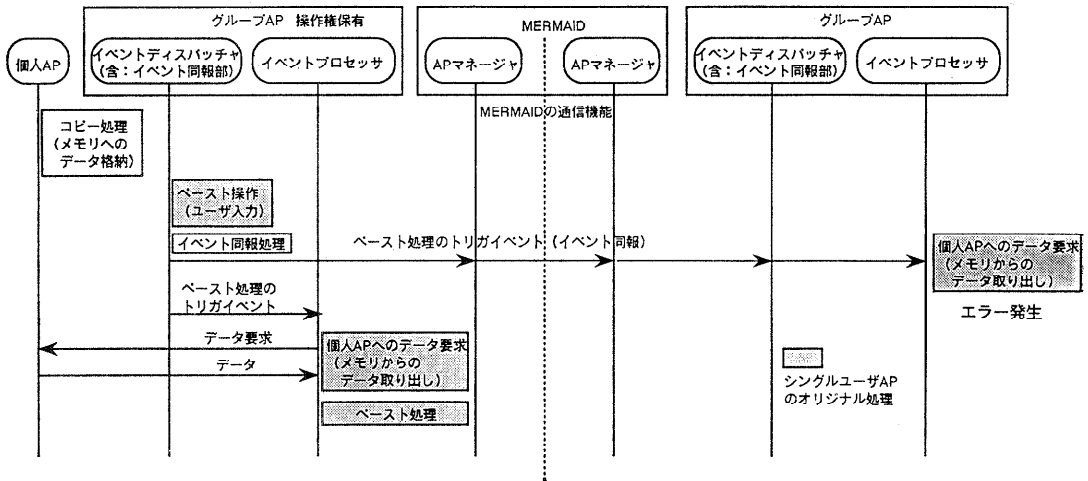


図 4 イベント同報による外部プロセスとのデータ交換エラーシーケンス

Fig. 4 Data inconsistency by interaction between a local application and a group application in the event distribution.

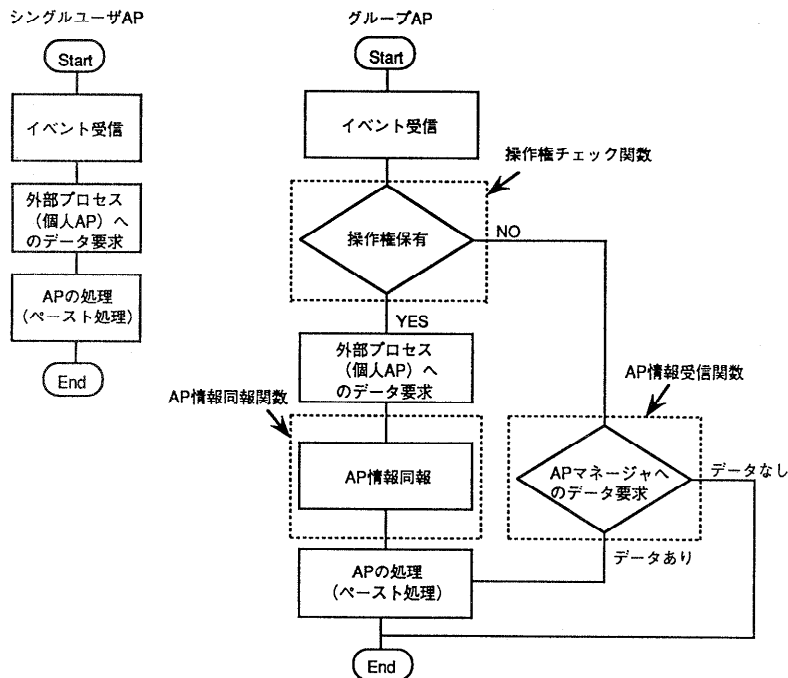


図 5 AP情報同報によるイベントプロセッサの処理

Fig. 5 Local processing at the event processor in the application-specific message distribution.

ベスト処理のトリガイベントをすべてのサイトで同報する処理までは、図4に示すイベント同報シーケンスと同じであるが、そのあとにデータの同報 (AP 情報同報) を追加する。受信サイトの AP マネージャは、受信した AP 情報をメモリに格納し、ベスト処

理のトリガイベントを再送する。受信サイトのグループ AP は、AP 情報受信関数を利用して、AP マネージャにデータを要求することによって個人 AP からのデータを受け取る。その後の処理はシングル AP のオリジナル処理 (ベスト処理) と同じである。

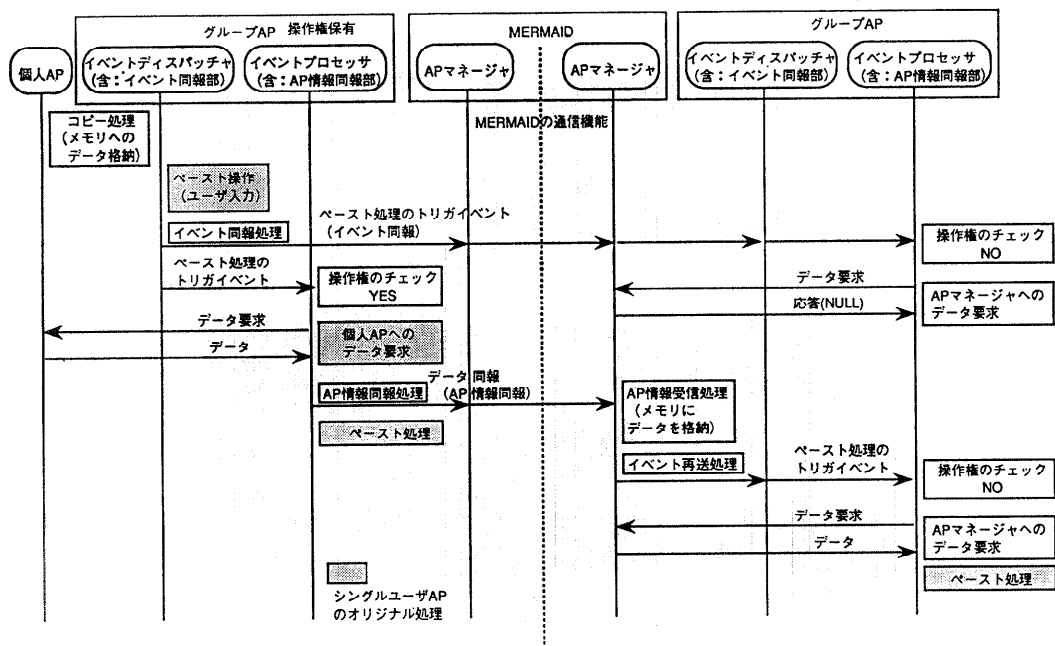


図 6 AP 情報同報による外部プロセスとのデータ交換処理シーケンス

Fig. 6 Data exchange between a local application and a group application in the application-specific message distribution.

この例では、外部プロセスとのデータ交換のイベント（ベスト処理のトリガイイベント）はグローバルイベントとして同報するが、イベントの時点で認識できる AP の場合、ローカルイベントとすることによって受信側の無駄な処理をなくすることができる。このように AP の処理に依存した AP 情報を同報し、AP マネージャがイベント同報、AP 情報同報を制御することにより、イベント同報だけでは行えなかった処理を含むシングルユーザ AP のグループ AP 化が可能になる。また、汎用的な関数を提供しているのでシングルユーザ AP のグループ AP 化における AP への変更点は小さく、画一的な変更となる。この方法は、最初からグループ AP を構築する場合の外部プロセスとデータ交換の制御にも適用可能である。

#### 4. グループアプリケーションへの適用例

グループ AP をイベント同報、AP 情報同報の組み合わせで同期制御することにより、イベント同報のみでは制御できなかった外部プロセスとのデータ交換が可能になる。組み合わせの割合は AP の提供機能によって異なるが、イベント同報方式はシングルユーザ AP を共有するのに有効である。AP 情報同報方式は

AP の処理に対応した細かい制御が行えるので、最初からグループ AP を構築するのに有効であるが、シングルユーザ AP の機能を生かした拡張にも対応可能である。

3.2 節で提案した方式に従って、一般に市販されているシングルユーザ AP であるハイパーメディア AP を修正し、グループ AP 化した（図 7）。ハイパーメディア AP は、マウス操作によってマルチメディアデータベースから情報検索を行うものである。この AP では、表示されるデータの一部は一元的にマルチメディア DB に保管されている。そのデータベースへのアクセス要求イベントは、操作権保有者の AP でのみ処理され、アクセス結果がすべてのサイトに AP 情報として同報される。その他のすべてのサイトが保有しているデータへのアクセスに関しては、マウスイベントの同報によって制御されている。すべてのデータが各サイトに存在すれば、イベント同報のみで制御が可能である。

最初からグループ AP として開発された AP として、MERMAID のグループ協同作業環境を提供している共有画面（黒板）がある。黒板は各サイトごとに起動され、参加者は黒板上でマルチメディア文書（テ





図 7 日本の遠隔4地点と米国とを結んで、ハイパーメディアAPを共有している様子  
 Fig. 7 Screen example of shared hypermedia application between Japan and U.S.

テキスト、イメージ、図形、手書き、ポインタ)をリアルタイムで編集・交換・共有を行うことができる。黒板では、文書の検索・表示を行う場合、最初の検索時に各サイトに文書を同報し、その後の編集作業はAP情報同報により各サイトが行う。マルチメディアデータの入力の場合、文字コード、座標情報、形、色などの情報が、拡大、縮小などの編集処理の場合、処理指示コマンドがAP情報として同報される。表2に黒板へのマルチメディアデータの入力処理におけるAP

情報の定義を示す。黒板における描画処理、編集処理は、各サイトが受信したAP情報に従って行っているため、転送時間は小さく(一秒未満)、遠隔地でも問題なく作業が行える。イメージの読み込みの場合、AP情報として実際読み込んだデータを各サイトに同報するため、データ量に比例して転送時間がかかる。しかし、実際に協同作業を行っている最中は、手書きによるコメントの付加が多く、イメージを読み込む頻度は比較的少ない。また、文書は作業に先だって

表 2 黒板におけるメディア入力に関するAP情報  
 Table 2 Application-specific messages in the blackboard.

| メディア           | AP 情報         |                           |
|----------------|---------------|---------------------------|
|                | 処理指示コマンド      | データ                       |
| テキスト           | DrawString    | 開始座標、描画文字列                |
| 図形(点)          | DrawPoint     | 座標、色                      |
| 図形(線)          | DrawLine      | 始点、終点、色、幅                 |
| 図形(長方形)        | DrawRectangle | 左上座標、サイズ、色、幅              |
| 図形(長方形(塗りつぶし)) | FillRectangle | 左上座標、サイズ、色                |
| 図形(弧)          | DrawArc       | 左上座標、サイズ、弧の開始角度、弧のサイズ、色、幅 |
| 図形(弧(塗りつぶし))   | FillArc       | 左上座標、サイズ、弧の開始角度、弧のサイズ、色   |
| イメージ           | DrawImage     | 左上座標、サイズ、イメージ形式、データ形式、データ |
| 手書き(色設定)       | SetColorHand  | 色                         |
| 手書き(太さ設定)      | SetWidthHand  | 幅                         |
| 手書き(開始)        | StartHand     | 座標                        |
| 手書き(描画)        | DrawHand      | 座標                        |
| ポインタ表示/非表示     | ShowPointer   | 座標、状態                     |

MERMAID のファイルサーバから各サイトに同報しておくことで、検索時間を小さくできる。

## 5. おわりに

本論文では、グループ AP の構造、要求条件、MERMAID の機能を利用した AP 共有分散協調制御方式とその実現例について述べた。提案する方式は、分散方式を基本とし、イベント同報と AP 情報同報によって分散している AP の同期を制御している。

この方式によって、従来のイベント同報のみでは不可能であった外部プロセスとのデータ交換が可能になった。また、AP マネージャによって、グループ AP を一元的に管理することにより、グループ管理などの基本的な機能を AP ごとに実装する必要がなくなった。現在、ソフトウェア開発 (CASE) ツール、仮想現実感 (VR) システム<sup>17)</sup>を AP として本方式の利用評価を行っている。本論文では、シングルユーザ AP をグループ AP とすることを中心に述べているが、提案する方式は最初からグループ AP を構築する場合にも適用可能である。

AP 共有は、グループごとに異なる業務形態に柔軟に対応することを可能にし、個人と組織、組織と組織を結ぶネットワーク環境において、協同作業を行うのに有効な手段となり得ると考えられる。今後の課題としては、本方式をさらに多くの AP に適用し、MERMAID のプラットフォームとしての汎用性を高める、高速ネットワークに対応するために、グループ AP を提供機能や利用形態から分類し、グループ AP に共通な機能、各 AP に対して最適な方式 (分散方式、集中方式) とそのトレードオフを明らかにし、最初からグループ AP を構築支援するための環境を整備する、ことが挙げられる。

## 参 考 文 献

- 1) Ellis, C. A., Gibbs, S. J. et al.: Groupware—Some Issues and Experiences—, *Comm. ACM*, Vol. 34, No. 1, pp. 38-58 (1991).
- 2) 阪田 : CSCW におけるマルチメディア技術、計測と制御, Vol. 30, No. 6, pp. 497-504 (1991).
- 3) 石井 : グループウェア技術と研究動向, 情報処理, Vol. 30, No. 12, pp. 1502-1508 (1989).
- 4) Watabe, K., Sakata, S. et al.: Distributed Multiparty Desktop Conferencing System: MERMAID, *Proc. Conf. on Computer Supported Cooperative Work*, pp. 329-342 (Oct. 1990).
- 5) Watabe, K., Sakata, S. et al.: Distributed Desktop Conferencing System with Multi-user

Multimedia Interface, *IEEE Journal on Selected Areas Communications*, Vol. 9, No. 4, pp. 531-539 (1991).

- 6) 渡部, 阪田ほか : マルチメディア分散在席会議システム MERMAID, 情報処理学会論文誌, Vol. 32, No. 9, pp. 1200-1209 (1991).
- 7) Lauwer, J. C., Joseph, T. A. et al.: Replicated Architecture for Sharing Window System: A Critique, *Proc. Conf. on Office Information Systems*, pp. 249-260 (Apr. 1990).
- 8) Crowley, T., Milazzo, P. et al.: MMconf: An Infrastructure for Building Shared Multimedia Applications, *Proc. Conf. on Computer Supported Cooperative Work*, pp. 329-342 (Oct. 1990).
- 9) Greenberg, S.: Sharing Views and Interactions with Single-user Applications, *Proc. Conf. on Office Information Systems*, pp. 227-237 (Apr. 1990).
- 10) 中山, 森ほか : 多者間電子対話システム ASSOCIA, 情報処理学会論文誌, Vol. 32, No. 9, pp. 1190-1199 (1991).
- 11) Abdel-Waahab, H. M. and Feit, M. A.: A Framework for Sharing X Window Client in Remote Synchronous Collaboration, *Proc. Tri-Comm '91*, pp. 159-168, IEEE (Apr. 1991).
- 12) Knister, M. J. and Prakash, A.: DistEdit: A Distributed Toolkit for Supporting Multiple Group Editors, *Proc. Conf. on Computer Supported Cooperative Work*, pp. 329-342 (Oct. 1990).
- 13) Patterson, J. F.: Rendezvous: An Architecture for Synchronous Multi-User Applications, *Proc. Conf. on Computer Supported Cooperative Work*, pp. 317-328 (Oct. 1990).
- 14) Ahuja, S. R., Ensor, J. R. et al.: A Comparison on Application Sharing Mechanisms in Real-time Desktop Conferencing Systems, *Proc. Conf. on Office Information Systems*, pp. 238-248 (Apr. 1990).
- 15) Ohmori, T., Maeno, K. et al.: Distributed Cooperative Control for Sharing Applications based on Multiparty and Multimedia Desktop Conferencing System: MERMAID, *Proc. of the 12th International Conference on Distributed Computing Systems*, pp. 538-546 (June 1992).
- 16) 大森豊子, 前野和俊ほか : マルチメディア分散在席会議システム (MERMAID) を利用したグループアプリケーションの分散協調制御方式とその実現例, 情報処理学会マルチメディア通信と分散処理研究会 53-8, pp. 53-60 (Jan. 1992).
- 17) 阿部豊子, 前野和俊ほか : マルチメディア分散在席会議システム MERMAID における分散協調機構と、そのネットワーク VR への応用, 情報処理学会グループウェア研究グループ 2-8, pp. 57-64 (Sep. 1992).

(平成 4 年 11 月 5 日受付)

(平成 5 年 4 月 8 日採録)



**阿部 豊子 (正会員)**

1965年生。1988年お茶の水女子大学理学部数学科卒業。同年、日本電気(株)入社。以来、C & Cシステム研究所にて、マルチメディア分散会議システムをはじめとするネットワークとコンピュータを利用した遠隔型のマルチユーザアプリケーションなどグループウェアに関する研究開発に従事。現在、C & Cシステム研究所ネットワーク研究部に勤務。



**前野 和俊 (正会員)**

1958年生。1980年東京理科大学工学部情報科学科卒業。1982年同大学院修士課程修了。同年、日本電気(株)入社。以来、C & Cシステム研究所にてコミュニケーションネットワークアーキテクチャ、高速マルチメディアLAN、マルチポイントテレコンファレンスシステム、マルチメディア分散会議システムをはじめとするグループウェアに関する研究開発に従事。現在、C & Cシステム研究所ネットワーク研究部主任。電子情報通信学会、日本オペレーションズ・リサーチ学会各会員。



**阪田 史郎 (正会員)**

1949年生。1972年早稲田大学理工学部電子通信学科卒業。1974年同大学院修士課程修了。同年、日本電気(株)入社。以来同社C & Cシステム研究所においてコンピュータ・ネットワーク・アーキテクチャ、分散処理、マルチメディア通信システム、グループウェア、ネットワーク管理システムなどコンピュータと通信との統合領域に関連する研究開発に従事。著書「マルチメディアとネットワークによるグループウェア実現技術」(共著、ソフト・リサーチ・センター)、「マルチメディア情報通信」、「図解グループウェア」(オーム社)ほか。現在、C & Cシステム研究所ネットワーク研究部部長。工学博士。IEEE、電子情報通信学会各会員。



**福岡 秀幸**

1959年生。1983年早稲田大学理工学部電子通信学科卒業。1985年同大学理工学研究科電気工学修士課程修了。同年、日本電気(株)入社。以来、ソフトウェア生産技術研究所、C & Cシステムインタフェース技術本部、C & Cシステム研究所、関西C & C研究所にてプロトコル処理、コンフォーマンステスト、グループウェア、ヒューマンコミュニケーション、マルチメディア通信の研究開発に従事。現在、C & Cシステム研究所ネットワーク研究部主任。電子情報通信学会会員。