

シンセティック・メディア・システム IntelligentPad における協調作業場の実現

長崎 祥[†] 田中 譲[†]

計算機上で協調作業を行うためには、作業で利用されるドキュメントやツールやアプリケーション・プログラムを複数の計算機間で共有する必要がある。この共有にあたって、ユーザが自分の必要とする協調作業環境を自分自身で容易に構築できるためには、作業環境とその共有環境のツールキット化が必要である。著者らが開発したシンセティック・メディア・システム IntelligentPad では、ドキュメントやツールやアプリケーション・プログラムを電子的な紙として扱う。この電子的な紙をパッドと呼ぶ。パッドに対してユーザが行う操作イベントを共有することで、作業環境の共有を実現している。操作イベントの共有に要する機能をパッドとして部品化するために、ユーザ操作イベントの検出、データ化機能、操作イベント・データの再生機能の3つの機能を提供するフィールドパッドを開発した。フィールドパッドは、協調作業への応用が考えられていないパッドを、複数ユーザで共有して利用できるようにする。複数サイト間におけるパッドに対するユーザ操作イベントの共有は、パッドをフィールドパッドの上に貼り、合成されたパッドの共有コピーを各サイトに配布することによって行われる。イベント共有によるイベント競合は、各コピーに優先度を与えてイベント再生の制御を行うことで解決した。フィールドパッドを階層的に貼り合わせることで、作業環境の部分共有も実現できる。

Realizing the Cooperative Work Field in IntelligentPad

AKIRA NAGASAKI[†] and YUZURU TANAKA[†]

This paper proposes a construction-set architecture for shared workbenches. To share the multimedia documents and tools among multiple users at different sites, our system does not share each of them but shares every user event applied to them. The IntelligentPad system developed in Hokkaido University considers multimedia documents and tools as sheets of paper with different functions. They are called pads. The pasting of various pads on another pad defines a new pad with a composite function. The event sharing in this system is realized by a special pad called a field pad that works as a workbench. A field pad with various pads pasted on itself defines a work environment. Shared copies of the same field pad with some pads on it allow to share every user event in this same work environment. These shared copies can be distributed among different sites that are connected through a network, which easily realizes a CSCW environment.

1. はじめに

計算機の高性能化と低価格化によって、1台の計算機をひとりで占有することが可能となった。計算機は単なる計算道具からユーザが行う様々な作業を支援する環境を提供するものへと変わってきている。我々が行う作業には多くの協調作業が含まれているため、計算機は個人作業を支援するとともに協調作業を支援することも必要である。協調作業を計算機によって支援しようとする試みが CSCW (Computer Supported Cooperative Work) である。協調作業を支援するシ

ステムとしては電子会議システムや共同執筆支援システムがあり、協調作業支援は NLS¹⁾ や Xanadu²⁾ の提案に見られるように、ハイパーテキスト・システムでは、歴史の当初より目標の1つとされてきた。

協調作業システムとしては Colab³⁾ や gIBIS⁴⁾ などがあり、国内においては MERMAID⁵⁾ などがある。これらのシステムは予め用意されたツールしか利用できない閉じたシステムである。閉じたシステムでは、システムに用意されていないツールを使う作業は行うことができない。様々な作業が行えるように、あらゆるツールを予めシステム側に用意しておくということもできない。新しく開発されるツールを随時追加できるシステムが望ましい。新しく開発されたツールを追加できるシステムには、Report^{6),7)}, MMConf⁸⁾,

[†] 北海道大学工学部電気工学科
Electrical Engineering Department, Faculty of
Engineering, Hokkaido University

Rendezvous⁹⁾, Dialogo^{10),11)}, ConferenceDesk¹²⁾, などが、国内においては ASSOCIA¹³⁾ などがある。これらのシステムは、作業を部分的に共有するサブグループを作ることができない。Rapport 以外のシステムは、個人作業用に開発されたツールをそのまま利用することはできない。

筆者らは、個人作業を支援するために開発された様々なツールが、協調作業においても利用できるアーキテクチャを提案する。本論文では、協調作業システムが持つべき機能について3章で述べる。3章であげた機能は、共有場という作業参加者全員が共有するひとつの仮想空間の中に実現される。協調作業に利用されるツールは、共有場に存在する。共有場の表現には、イベント共有、共有場へのツール追加、共有場への新規参加の3つの機構が必要である。各機構の実現については、4章、5章、6章でそれぞれ述べる。このアーキテクチャでは、共有場の部分的な共有もできる。この機構については、7章で実現機構を述べる。このアーキテクチャは、著者らが開発しているシンセティック・メディア・システム IntelligentPad¹⁴⁾⁻¹⁸⁾ 上に実現される。IntelligentPad については、2章で簡単にまとめる。

2. IntelligentPad の概要

共有場のアーキテクチャの実現を行ったシンセティック・メディア・システム IntelligentPad について、その概要を述べる。著者らは、IntelligentPad を SparcStation 2 上で開発している。開発言語としては UNIX 上で動作する Smalltalk-80¹⁹⁾ を用いている。図1にこのシステムのディスプレイ・ハードコピーの一例を示す。

計算機上には、音声や画像など多様な型の情報を処理するための媒体として、現実世界に存在するメディアのメタファが導入されている。計算機上に実現された電子的なメディアを以後、単にメディアと呼ぶ。シンセティック・メディアとは、互いに組み合わせることで、各々が持つ機能を合成した機能を持つ新しいメディアを容易に定義することができるメディアのことをいう。

IntelligentPad は、様々なメディアを紙のイメージを持つ「パッド」として統一的に扱う。図1中にある矩形のウィンドウがすべてパッドである。パッドは図1のように、ディ

スプレイ上で2次元の表示形態を持つ可視オブジェクトとして表現される。IntelligentPad は各種マルチメディアのほか、ボタンのような機能部品や、様々なアプリケーション・プログラム、さらにはデータベース・サービスやメール・サービスのような各種サービス・システム・プログラムにも紙のイメージを与え、これらすべてをパッドとして統一的に扱うことができる統合環境を提供する。ユーザは、パッドの移動やコピー等の操作をマウスを使って直接適用することができる。様々な情報やツールに対して、異なる機能ごとに異なる種類のパッドが用意され、ユーザはそれらをディスプレイ上で移動し、重ね合わせることによって互いの持つ機能を組み合わせ、新しい機能を持つパッドを定義することが可能である。パッドを重ね合わせる操作を、このシステムでは「パッドを貼り合わせる」と呼ぶ。

パッドの内部構造は、Smalltalk-80 のウィンドウのモデリングに利用されている MVC モデルを単純化したものである。このシステムではモデル (M)、ビュー (V)、コントローラ (C) の3つのオブジェクト間には図2中のパッド P_i に示されるように、M から V への更新伝搬、C から V へと V から M へのメッセージ送出力リンクが張られている。パッドは自分の状態をモデル部に持ち、その一部はスロット・リストという形で定義される。各スロットは他のパッドからアクセスすることができる。すべてのパッドはひとつ以上のスロットを持ち、その中に値だけではなく手続きを入

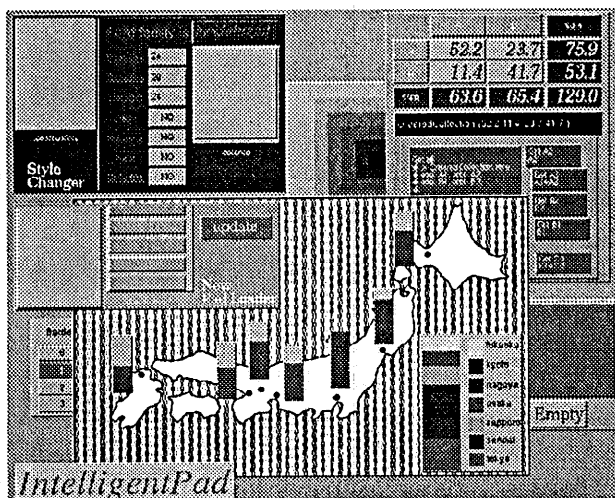


図1 システムの画面コピー

Fig. 1 A display hardcopy of the IntelligentPad system.

れることもできる。手続きは、スロットをアクセスすることで実行される。

パッドの機能合成は、貼り合わされたパッドの間でスロット値を授受することにより行われる。図2ではパッド P₂ がパッド P₁ に貼られている。P₂ から P₁ へのデータ通信は、set メッセージによって P₂ が P₁ のスロットへ値を送ることにより行われる。P₁ から P₂ へのデータ通信は、P₁ が update メッセージを P₂ に送り、このメッセージを受け取った P₂ が gimme メッセージによって P₁ のスロット値を読み出すことにより行われる。パッド間で送られるメッセージには、データ通信メッセージのほかにジオメトリカル・メッセージがある。ジオメトリカル・メッセージというのは、パッドの表示に関する操作を行うためのメッセージの総称で、メッセージ名は標準化されている。このように、パッド間で授受されるすべてのメッセージが標準化されているので、ユーザはパッドを自由に貼り合わせることができる。

3. 共有場のアーキテクチャ

パッドが協調作業においても利用できるようにするため、協調作業支援システムに必要とされる機能を考える。協調作業支援システムのアーキテクチャには、表1に示すような機能が必要になると考えられる。表1にある機能の中で、(2)は、作業に参加しているユーザのうち、特定の1人が持つツールやドキュメントを表示のみ他の参加ユーザ全員にも見せる機能である。この共有では、ツールやドキュメントに対して、それらのオーナー以外のユーザが操作を行うことはできない。(3)は、ある作業参加ユーザがツールなどに

対して行ったイベントと同一のイベントを、他の参加ユーザ全員の持つツールの同一コピーに対して適用する機能である。(10)は、共有場とその外部空間との間でツールを出し入れする機能である。(12)は、ひとつの協調作業において使用しているツールの中の一部だけが使用可能であるサブグループを可能とする機能である。(13)は、作業ツールの大きさや見る向きを参加ユーザごとに変えたり、作業ツールに対する使用制限をするための機能である。

著者らは、様々な協調作業ができるシステムのアーキテクチャの開発を目標としている。ユーザが様々な協調作業を行うためには、システムは新しく開発されるツールが随時追加できなければならない。システムが予め用意されたツールしか使えないならば、ユーザは用意されていないツールを用いる協調作業が行えない。新しく開発されるツールを随時追加できるシステムである Rapport, MMConf, Rendezvous, Dialogo, ConferenceDesk, ASSOCIA と著者らが開発している IntelligentPad について、協調作業支援システムのアーキテクチャに必要と考えられる機能の有無を示したのが表1である。既存のシステムは、作業を中座する機能や、作業を部分的に共有するサブグループを自在に形成することはできない。

IntelligentPad においても、(2)、(11)、(14)や(15)の機能は現在のところ実現されていないが、それ以外の機能は既にも実現できている。これは、イベント

表1 協調作業アーキテクチャに必要な機能

Table 1 Comparing the functions in the architecture of CSCW system.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Rapport	×	×	○	○	×	×	○	○	×	×	○	×	×	×	×
MMConf	×	×	○	×	×	×	○	○	×	×	×	×	×	×	×
Rendezvous	○	○	○	○	×	×	-	○	×	×	?	-	○	×	×
Dialogo	×	×	○	×	×	×	○	○	×	×	×	×	×	×	×
Conference Desk	×	○	○	?	×	○	○	○	×	×	○	-	×	○	×
ASSOCIA	×	×	○	○	×	×	-	○	×	×	×	×	×	×	×
IntelligentPad	○	未	○	○	○	○	○	○	○	○	○	未	○	○	未

○:有、×:無、?:不明、未:未実現、-:意味を持たない

1. 状態共有
2. ビュー共有
3. イベント共有
4. 協調作業への中途参加
5. 協調作業の中座
6. 操作履歴の記録
7. 協調作業場へのツールの追加
8. ツールの共同利用 (ツールの種類制限なし)
9. ツールの共同構築
10. 協調作業場へのツールの移動
11. テレポインタ
12. 部分共有操作場
13. 見え方、操作権の差異
14. 音声による対話
15. 他参加者との対面作業

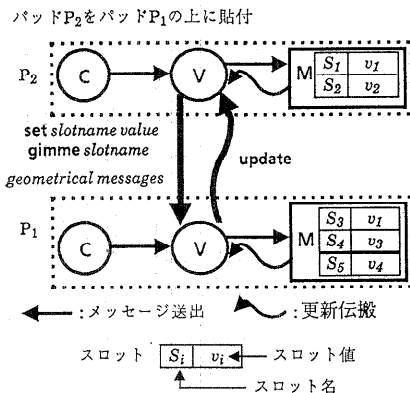


図2 パッド内部とパッド間のメッセージの流れ
Fig. 2 Messages between objects in a pad.

共有機構の部品化によって実現されている。計算機で協調作業を行うモデルとして、作業参加ユーザ全員が1つの空間を共有し、その空間内にあるツールを利用して作業を行うモデルを考える。共有される空間を共有場と呼ぶ。共有場の内部にあるツールは、参加ユーザ全員によって共有され、同時に参照や操作が行われる。共有場には、任意のパッドを持ち込むことができる。協調作業用に開発されたツールのみが利用できるわけではない。共有場は、操作場とそのコピーによって実現される。操作場は、イベント共有機能を持つ空間である。操作場は自身のコピーとの間で、作業参加ユーザが操作場内のツールに対して行ったイベントを共有することができる。このとき、操作場とそのコピーは作業参加ユーザによって共有されているという。イベントの共有によって、操作場内にあるツールへの入力は常に同一に保たれる。操作場を持つユーザからは、あたかも1つの空間を共有しているように見える。

著者らは、表1の機能を実現するため、(a)操作場を利用した作業参加ユーザ全員によるイベント共有、(b)操作場の転送、(c)操作場への新たなツールの中途追加の各機構を実現した。(a)、(b)は(3)の機能を実現するため、(c)は(7)と(10)の機能を実現するために必要となる。操作場は、その場にあるツールへのイベントを内部状態として持つフィールドパッド²⁰⁾という部品として実現される。イベントの共有は、フィールドパッドの状態共有コピーを作ることによって実現される。(4)、(5)、(6)、(8)、(12)、(13)はフィールドパッドを用いて以下のように実現できる。(4)は、フィールドパッドの共有コピーを作り、新規参加者へ転送することによって実現できる。新規参加者へのフィールドパッドの転送には、(b)の機能を用いる。(5)は、フィールドパッドとキューの機能を持つパッドとを組み合わせることによって実現できる。中座している間にフィールドパッドが検出したイベントは、キューパッドに蓄えておく。作業に復帰するとき、キューパッドからイベントを順次取り出して再生する。(6)は、検出されたイベントを蓄えることによって実現される。(8)は、個人作業用に開発されたパッドをフィールドパッドに貼るだけでよい。(12)については、フィールドパッドに別のフィールドパッドを貼り、部分共有場を作ることによって実現できる。(13)にある操作権の差異は、特定の参加ユーザが行ったイベントのみを通過させるパッドとフィールドパッドを

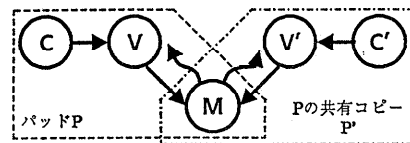
組み合わせることによって実現される。表示の差異は、イベント再生時の座標系を変換するパッドと、フィールドパッド内に貼られているパッドの表示イメージ変換機能を持つパッドによって実現できる。これらのパッドをフィールドパッドと組み合わせることによって、表示の差異が実現できる。

4. イベントの共有

本章では、共有場の実現に必要な基本機構である異なるサイト間でのイベント共有機構の実現について述べる。イベント共有には、(1)イベントの検出と再生を行う機能と(2)検出されたイベントを異なるサイトに分配する機能が必要となる。これらの機能は、個人による作業のために開発されたツールを作り直す必要がないように実現される。(1)の機能はフィールドパッドとして、(2)の機能はフィールドパッドの共有コピーを異なるサイトに分散させることによって実現される。イベント共有の基盤となる状態共有について述べたあと、(1)と(2)の機能について述べる。

4.1 共有コピーによる状態の共有

メディアのコピー操作には、メディアの状態を共有しない非共有コピー操作と、これを共有する共有コピー操作の2種類がある。静的なメディアでは、状態が変化することはないので、2種類のコピー操作の違いは生じない。これに対して、パッドのような動的なメディアは状態が変化し得るので、非共有コピー操作と共有コピー操作を区別する必要がある。IntelligentPadにおけるパッドの共有コピーを図3に示す。パッドの共有コピーは、コントローラ部とビュー部は独立なコピーを持ち、モデル部はオリジナルのパッドと同一モデル部を持つ。共有コピーのビュー部からモデル部の状態を変更する操作が行われると、モデル部は自分を保持しているすべてのビュー部に対して更新要求を出す。各ビュー部はこの要求により、表示の更新を行う。パッドを貼り合わせて作られた合成パッドの共有コピーは、図4に示すように定義される。合成



パッドPとその共有コピーP'は同一のモデルMを持つ

図3 パッドの共有コピー

Fig. 3 Shared copy of a primitive pad.

パッド P の共有コピー P' は、P の一番基底にあるパッド P₀ のモデル部だけを共有し、その上に貼られているパッド P₁ と合成パッド P₂ の状態は共有しない。

4.2 入力イベントの検出と再生機構の部品化

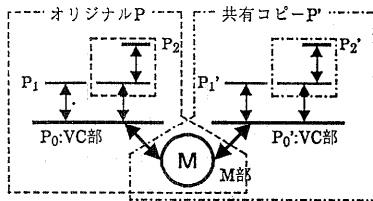
操作場は、イベントの検出と再生の機構を持つフィールドパッドによって定義される。フィールドパッドの上にはあらゆる種類のパッドを貼ることができる。図5に示すように、全体の共有コピーを作ることによって、フィールドパッドによる操作場の共有が行われ、共有場を実現することができる。

操作場の共有を実現するためには、各パッドがユーザから受けた操作イベントを検出する機構、検出したイベントを即座には実行しないでイベント情報として保持する機構、保持されているイベント情報を対象パッドに対して再生する機構の、3つの機構が必要となる。これらの機構は、検出がイベント検出パッドによ

り、保持と再生がイベント再生パッドによって実現される。フィールドパッドは、イベント検出パッドをイベント再生パッドに貼るることによって作られる合成パッドである。図6にフィールドパッドの共有コピーによるイベント共有の過程を示す。

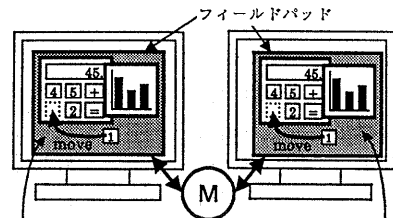
● イベントの検出

フィールドパッド上に貼られているパッドに対するユーザ・イベントの検出は、ユーザ・イベントを受け取るパッドがイベント検出パッドにイベント情報を通知することにより実現される。イベント情報は、イベントの種類、イベントの発生座標など、イベント再生時に必要なデータと、イベント発生サイト名からなる。ユーザ・イベントは、そのイベントの対象パッドによって受け取られるため、イベント検出パッドがこれ



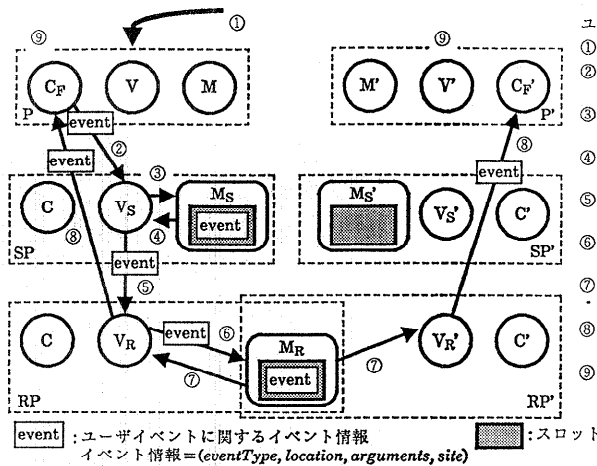
最も下層のパッドP₀とP₀'だけがモデルを共有する

図4 合成パッドの共有コピー
Fig. 4 Shared copy of a composite pad.



この2つの操作場で発生するユーザイベントは共有され、操作場内にある各パッドの状態や配置は常に同一に保たれる

図5 フィールドパッドの共有コピーによる操作イベントの共有
Fig. 5 Event sharing through shared copies of a field pad.



- ユーザイベントの検出・再生過程
- ① パッドPにユーザイベント発生
 - ② イベント検出パッドSPへイベント情報を通知
 - ③ イベント情報をSPが持つスロットへ格納
 - ④ SPのモデルの状態が変更されたことをビューへ通知
 - ⑤ イベント情報をイベント再生パッドRPへ送る
 - ⑥ イベント情報をRPが持つスロットへ格納
 - ⑦ RPのモデルの状態が変更されたことをRPとRP'のビューへ通知
 - ⑧ イベントを再生すべきパッドPとP'へイベント情報を送る
 - ⑨ イベント情報からイベントを再生

SP: イベント検出パッド RP: イベント再生パッド P: 任意のパッド
SP': SPの非共有コピー RP': RPの共有コピー P': Pの非共有コピー

図6 操作場上のパッドへのユーザ操作実行
Fig. 6 The process of user operation for the pad on field pad.

を横取りしてイベント検出を行うことはできない。そのため、ユーザ・イベントの検出は各パッドに行わせざるをえない。検出結果をイベント検出パッドに通知する機構が各パッドに必要となる。各パッドのコントローラ部は、そのパッドがイベント検出パッドの上方に置かれたときには、検出したユーザ・イベントからイベント情報を構成し、これをイベント検出パッドに通知する。このとき、マウスのドラッグ中の動きも逐一イベント検出パッドに通知すると、フィールドパッドの状態変更が頻繁に発生し、性能を著しく劣化させる。そこで、マウス移動は操作の発生地点から次の操作の発生地点までの直線移動とみなし、移動過程の情報に関しては検出しないことにした。

イベント検出の過程を図6中の①から⑥に示す。イベント検出パッド SP の上方にあるパッド P がユーザの操作を受ける(①)と、P は受けた操作イベントを即座に実行しないで、SP にイベント情報として送る(②)。これは、一度実行された操作を無効にするのが一般に困難なため、操作はイベントを共有しているフィールドパッド間でイベント競合の回避が行われた後に初めて実行される。②によって SP に送られてきたイベント情報は SP のスロットに格納される(③)。それにより、SP の状態が変化し(④)、パッド間のデータ通信機能によってイベント情報はイベント再生パッド RP へと送られて(⑤)、RP のスロットへと格納される(⑥)。

● イベントの再生

イベント情報を用いて操作を再生する過程は図6中の⑦から⑨である。⑥によってイベント情報が RP のスロットに格納されると、RP のモデル部の状態が変化するので RP と RP' のビュー部に更新要求が送られる(⑦)。RP のビュー部はイベント発生座標からイベントを再生すべきパッドを特定する。イベントを再生すべきパッドは、イベント情報に含まれているイベント発生座標の位置に貼られている最上位のパッドである。RP は特定されたパッド P のコントローラ部に対してイベント情報を送り(⑧)、そのコントローラ部はイベント情報を受け取ると、操作イベントの再生を行う(⑨)。RP' においても⑦から⑨が RP 上と同様に実行される。

4.3 フィールドパッドによるイベント共有例

フィールドパッドの上には任意の合成パッドが置けるので、複雑な対話処理も容易に複数利用者間で共同作業可能な形態に拡張することができる。本節では、

説明を簡便にするため、極めて単純な合成パッドを例として用い、これをフィールドパッドの上に配して CSCW 環境を定義した例を示す。例には、三目ならべを行う合成パッドを用いる。この例は単純ではあるが、パッドの移動やコピー操作を必要とし、パッドに対する操作イベントの共有過程を説明するのに格好の例である。図7がこのパッドをフィールドパッドの上に配し、共有コピーをとってイベント共有を実現した実例である。フィールドパッド上にはゲームに必要な○や×の絵が描いてある駒のパッドが貼ってある。この共有コピーの一方をゲームの対戦相手に送ることができる。ゲームの対戦相手へのパッドの転送は、6章で説明する転送パッドを利用して行われる。

プレイヤーの1人が自分の駒パッドをコピーしようとする、この操作は即座には実行されずにイベント情報に変換され、駒パッドを介してフィールドパッドに送られる。イベント情報はフィールドパッドを構成しているイベント再生パッドのモデル部に送られ、それを共有する自分の側と対戦相手の側との2枚のフィールドパッド上でコピー操作が同時に再生される。コピーされた駒パッドの移動操作も同様に両方のサイトで実行される。移動要求が駒パッドからフィールドパッドへと送られ、自分の側と対戦相手の側にある2枚のフィールドパッド上で各々独立に移動操作が再生される。駒パッドは、移動操作によってフィールドパッド内の同じ相対位置に移動させられる。

フィールドパッドを用いると、マウスイベントだけでなく、キーボードイベントも複数サイト間で共有することができるので、フィールドパッド上にテキストパッドも配しておくことにより、ゲーム中に相手にメッセージを送ることもできる。

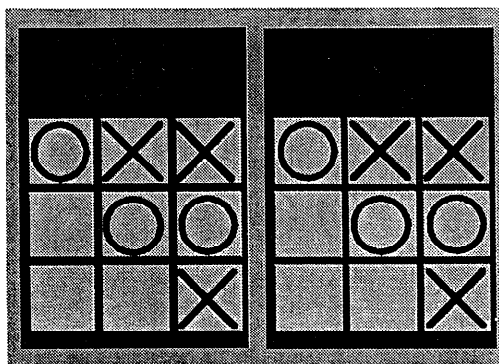


図7 操作場共有による三目ならべの画面コピー
Fig. 7 A display hardcopy of a tick-tac-toe game on shared copies of a field pad.

4.4 分散システムにおけるパッドの状態共有

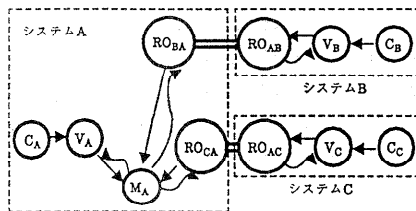
フィールドパッドによる操作場の共有には、異なるサイト上にあるフィールドパッド間において、検出されたイベントを共有する機構が必要である。本節では、パッドの分散状態共有の機構について述べる。

4.4.1 パッドの分散共有コピー

ネットワークで結合されている複数サイト上のシステムに分散した共有コピーを実現するには、サイト間でメッセージ交換を行う機構が必要である。この機構は、リモート・オブジェクト (RO) を用いて実現される。RO は2つで対をなし、それぞれが異なるシステム上におかれ、ネットワークを通じてメッセージの交換を行う。RO はオブジェクトを1つ持つ。RO は、自分と対になっているROが持つオブジェクトの代理オブジェクトとして機能する。RO はメッセージを受け取ると、対になっているROに対して受け取ったメッセージをデータとして送る。対になっているROは、このデータをメッセージとして自分が持つオブジェクトに送る。分散共有コピーでは、図8に示すようにパッドのモデル部とビュー部を切り離し、これら2つのオブジェクトを一对のROに1つずつ持たせる。図8では、システムA上にモデルMAがあり、このMAをシステムBとシステムC上のパッドがシステムA上のパッドと共有している。MAには、依存物としてビューVAのほかにもROBAとROCAが登録されている。ROBAがVBの、ROCAがVCの代理である。またVBとVCは、モデル部としてそれぞれROAB、ROACを持っている。これらのROはシステムAにあるROBA、ROCAと対をなすリモート・オブジェクトでMAの代理として働く。これらのROを通じて、3つのパッドのビュー部VA、VB、VCと、それらが共有するモデルMAとの間でメッセージ交換を行うことができ、分散共有コピーが実現される。

4.4.2 システム間通信の実現機構

ROによる異なるサイト間のデータ通信は、実際に



ROを介して異なるシステム上にあるモデル部とビュー部がメッセージを交換

図8 パッドのシステム間共有コピー
Fig. 8 Shared copies at different sites.

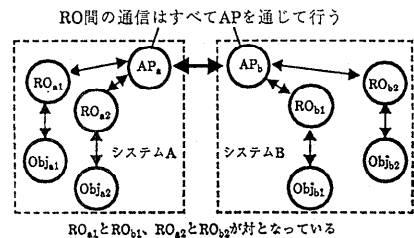
は図9のようにアクセスポイント AP という通信管理オブジェクトを介して行われる。AP は各サイトの IntelligentPad システムごとに1つずつ用意され、他サイトのシステムにあるROへのデータ転送機能と他のサイトの IntelligentPad システムへのパッド転送機能を持つ。ネットワークを介する他サイトの IntelligentPad システムとの通信は、すべて AP が管理する。AP 間の通信には、UNIX のプロセス間通信機構であるソケットが利用されている。

ROによる異なるサイト上の IntelligentPad システム間のデータ通信は、まず転送元サイトのROがデータと相手のアドレスから構成されるパケットをこのサイトのAPに送る。転送元サイトのAPはアドレスから転送先サイトのAPを探し、そのパケットを送る。転送先のAPはパケットを受け取ると、その中のアドレスからデータを受け取るべきROをこのサイトで探し、それに対してデータを送る。

4.5 共有コピーにおけるアクセス競合の解決

フィールドパッドの分散共有コピーでは、共有場に対して作業に参加しているユーザからのアクセスが非同期的に発生する。共有場の整合性を保つためにはアクセス競合を回避するための制御が必要である。フィールドパッドで実現された共有場へのアクセスは、共有モデル部に対する複数のビュー部からのスロット・アクセス要求を生じる。したがって、共有モデル部のスロットへのアクセス競合を回避する制御が必要となる。既存 CSCW システムにおいては、以下のような方法が競合回避のために提案されている。

- (1) 変更するデータに対してロックをかける。
- (2) 操作権を表すトークンが1つ用意され、操作はこのトークンを獲得したユーザだけに許される。
- (3) システムは操作競合が生じたことのみを検出し、ユーザが競合の解消を行う³⁾。



RO間の通信はすべてAPを通じて行う

RO_{a1}とRO_{b1}、RO_{a2}とRO_{b2}が対となっている

図9 パッド間通信の実現機構
Fig. 9 Communication mechanism between pads at different sites.

(4) 操作競合により、操作の実行順序がサイトごとに異なっても、同一結果が得られるように各サイトで操作を変形する²¹⁾。

しかしこれらの方法は、制御権の切り替え操作を必要としたり、使われるツールの機能に関する予備知識を必要とする。利用するツールによっては適用できない方法もあり、1つの制御方法であらゆるツールの操作競合を解消することはできない。IntelligentPadでは、同一のモデル部を持つパッドのビュー部に優先度を与え、その優先度によりモデル部へのアクセス競合を制御している。優先度によるアクセス制御を基盤として用いると、様々な制御方法がイベント情報の流れを制御するパッドとフィールドパッドを組み合わせることにより実現できる。例えば、(2)の制御を考えてみよう。この制御方式は優先度だけでは実現することができないので、特定のユーザによる操作だけが通過できるゲートの役割を果たすパッドを利用する。イベント検出パッドとイベント再生パッドの間にゲートパッドを貼っておくと、特定ユーザによる操作だけを再生することができる。ゲートパッドは、スロットにサイト名を1つ持ち、イベント情報に含まれるサイト名と自分が持つサイト名とが同一の場合だけイベント情報を通過させる機能を持つ。サイト名が特に指定されないときには、すべての参加ユーザの操作を通過させる。ゲートパッドが持つサイト名は、イベント検出パッド上に貼られているボタンパッドから変更される。パッドは、スロットアクセスが自分に対して行われたとき、アクセスされるべきスロットを自分が持っているなければ、アクセスを自分が貼られているパッドへ委譲する。この機能により、ボタンパッドがイベント検出パッドを通過して、ゲートパッドのスロット値を変更できる。ユーザは、自分のいるサイト名を持つボタンを押して操作権を獲得し、作業後にリリースボタンを押して操作権を放棄する。ゲートパッドを利用すれば、ユーザが操作権を獲得した後に操作競合は発生しない。しかし、操作権獲得時、ゲートパッド自身に対する操作が競合する。これを回避するため、優先度による制御を利用する。

4.5.1 優先度によるスロット・アクセスの制御

優先度の高いスロット・アクセスを優先的に処理するには、優先度の高いアクセスが到着したときに現在の処理を中止して新たに到着したアクセスの処理を開始する機構が必要である。パッドのモデル部でのスロット・アクセスの処理は、(1)スロット選択、(2)ス

ロット値変更、(3)ビュー部の更新要求の順に行われる。パッドのモデル部の状態が、ビュー部からスロットに入力される値だけで決まり、なおかつ先行アクセスの処理が(1)、(2)の段階であれば、このスロット・アクセスを中止することができる。パッドのモデル部の更新後の状態がビュー部からの入力値だけでは決まらないパッドの場合は、スロット値が一度変更されてしまうと、そのスロット・アクセスを無効にすることはできない。そのため、先行アクセスの処理が(1)の段階にある場合にしかアクセスを中止することはできない。フィールドパッドは、前者であるので(1)、(2)の段階までアクセスを無効にできる。

4.5.2 優先度の決定

ビュー部からのスロット・アクセスに対する優先度の決定は、モデル部が持つ優先度表を利用して行われる。ビュー部とコントローラ部のみを持っているユーザが勝手に優先度を変更するのを防ぐため、この表はモデル部で管理される。優先度表は、ビュー部に与えられたID番号と優先度との対からなる。パッドのビュー部は、スロット・アクセスを行う時に、自分の持っているID番号を同時に送る。モデル部は、ID番号と優先度表からスロット・アクセスの優先度を知る。

各ビュー部に対する優先度の初期値は、ビュー部とモデル部の間に依存関係ができた時、その時点で最も低い優先度に設定される。各ビュー部の優先度は、モデル部にあるpriorityスロットの値を読むことで知ることができる。操作場を共有するユーザが勝手に優先度を変更することを防ぐため、優先度の変更は変更が許されたビュー部を持つ特定ユーザだけに限定される。

優先度変更が許されていないビュー部を持つユーザが優先度を変更するためには、まずモデル部のpasswordスロットに正しいパスワードを入力して変更権を得なければならない。passwordスロットの値は、他のパッドから参照することはできない。パスワードには、passwordスロットに最初に書き込まれた値が利用される。パスワードの変更は、現在のパスワードと新しいパスワードとの組をpasswordスロットに書き込むことで行う。

5. 共有場へのパッドの持込み

前章では、協調作業に利用できるパッドは操作場の共有開始時点でフィールドパッド上に貼られているパ

ッドであった。共有開始後、作業に新たなパッドが必要となったときには、共有場内にパッドを持ち込むための機構が必要となる。著者らは、共有場内と共有場外とは異なる空間であると考え、それらを結ぶためにワームホールのメタファを利用することとした。ワームホールの入口としてブラックホールパッドを、出口としてホワイトホールパッドを定義し、これらを組で用いることとした。

組となるブラックホールパッドとホワイトホールパッドは、同一のモデル部を持つ。ブラックホールパッドのビュー部は、自分に貼られるパッドをモデル部のスロットに入れる機能を持ち、ホワイトホールパッドのビュー部は、モデル部のスロットに保持されているパッドをコピーして自分の上に貼る機能を持つ。ブラックホールパッドにパッドが貼られると、そのパッドはブラックホールパッドのモデル部のスロットに入れられる。ブラックホールパッドのモデル部は状態が変化するので、ブラックホールパッドとホワイトホールパッドのビュー部に更新要求がモデル部から送られる。このとき、ホワイトホールパッドのビュー部は、スロットにあるパッドをコピーして自分の上に貼りつけるが、ブラックホールパッドは何もしない。これによって、ブラックホールパッドに貼られたパッドが、ホワイトホールパッドの上に現れる。一方、ホワイトホールパッドにパッドが貼られた場合には何も起こらない。

6. 他のシステムへの操作場の転送

フィールドパッドによって協調作業を行うには、フィールドパッドの共有コピーを他サイトのユーザに転送する必要がある。フィールドパッドの共有コピーを他のシステムに転送するには、パッドを他のサイトの IntelligentPad システムに送る転送パッドを用いることができる。パッドを他のサイトの IntelligentPad システムに転送するとき、 ΔP はそのパッドをバイナリデータに変換して転送先の AP へと送る。共有コピーを持つパッドは、これらが図 8 に示された分散共有コピーとして機能するように、モデル部とビュー部の間に一対の RO が挿入される。転送パッドは、自分の上に貼られているパッドを AP に送る機能と、他の AP から送られてきたパッドを自分の上に貼りつける機能を持つ。この転送パッドを利用することで、ユーザはパッドの転送も貼り合わせ操作によって行うことができる。

7. 操作場の部分共有

協調作業において、作業の一部だけを共有するサブグループが必要とされる場合がある。IntelligentPad では、操作場の部分共有は、2枚以上のフィールドパッドの貼り合わせによって実現することができる。例えば、ユーザ A、B の間で操作場の共有を行うとする。共有操作場で行う作業の一部を他のユーザ C とも共有したい場合には、図 10 のように操作場のフィールドパッド FP_0 に別のフィールドパッド FP_1 を貼って、操作場の階層構造を作る。ユーザ A が FP_1 の共有コピー FP_1'' を作り、転送パッドによってユーザ C に送ることにより、ユーザ C はユーザ A、B と操作場を部分的に共有することができる。

フィールドパッドを階層的に重ねて貼った場合のイベント情報の流れを図 11 に示す。 SP_1 上に貼られているパッド P に対して操作が行われたとき、イベント情報は $P \rightarrow SP_1 \rightarrow SP_0 \rightarrow RP_0 \rightarrow M_0$ と検出される。イベントの再生では、イベント情報が $M_0 \rightarrow RP_0 \rightarrow RP_1 \rightarrow P$ と送られ、P によってイベントが再生される。また

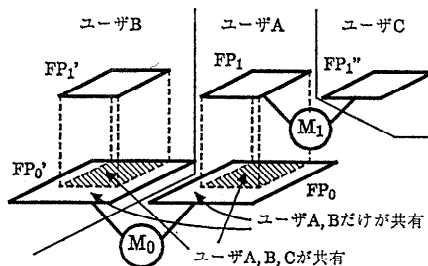


図 10 操作場の部分共有 Fig. 10 Nested shared fields.

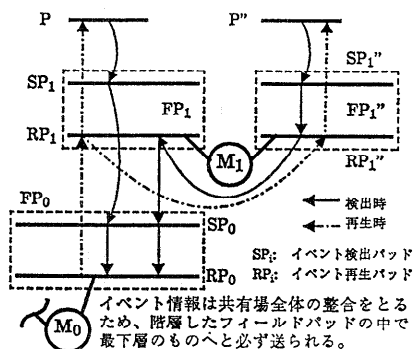


図 11 操作場階層化時のイベント情報の流れ Fig. 11 The event information flow among nested shared fields.

RP₁に送られてきたイベント情報はM₁を通じてRP₁'に送られ、さらにP''へと送られ、P''によってイベントが再生される。

イベント再生パッドは、イベント情報を保持するスロットの他に再生制御スロットを持つ。再生制御スロットの値が true の場合に限ってイベントの再生を行うことができる。このスロットの初期値は true であるが、イベント再生パッドが他のフィールドパッドに貼られると値が false になる。このスロット値は、RP₁に対する RP₀ のように、そのイベント再生パッドの直下のフィールドパッドからイベント情報が送られてきたときに true になる。

P'' に対して操作が行われたときには P''→SP₁'→RP₁'→M₁→RP₁→SP₀→RP₀→M₀ とイベント情報が送られる。再生はPに対して操作が行われたときと同様に行われる。RP₀の共有コピーからイベント情報が送られてきた場合も、Pに対して操作が行われたときと同様にイベントが再生される。

このように、どの階層のどのフィールドパッド上で操作が行われても、イベント情報は最下位にあるイベント再生パッドへと送られ、そこで操作イベントの競合が解消される。そして、最上位のイベント再生パッドが、イベント情報をもとにしてイベントを再生すべき対象パッドを特定する。

8. ま と め

計算機による実時間協調作業環境を容易に構築するためには、作業環境もツールキット化することが必要である。そのため、操作場というものを考え、その上に置かれたツールに対するユーザイベントを共有できるようにした。これによって、複数ユーザが同一の作業環境を共有することができ、作業環境の共有が可能となった。操作場はフィールドパッドによって実現される。フィールドパッド上に任意のパッドを配置し、その共有コピーを作業に参加する各メンバーのシステムに送っておけば、それらのパッドを作業対象として持つ同一の環境をすべてのメンバーが共有しながら操作を行うことが可能になる。

様々なツールは、複数ユーザがこれを共有して利用することを想定して作られていない場合でも、協調作業環境を操作場という形で部品化することにより、協調作業に利用することができる。各ユーザは、日常利用している作業ツールを用いて協調作業環境を構築することができる。この作業環境の構築も「フィールド

パッドに作業ツールのパッドを貼る」という操作のみによって行うことができる。

現在 IntelligentPad の開発は、イーサネットによって接続された SparcStation 2 上で行われている。IntelligentPad 自体は、Smalltalk-80 が動作する環境であれば利用することができるが、フィールドパッドによる協調作業を行うためには、サイト間通信を行う機構を持つ OS が必要となる。開発中のシステムでは、サイト間通信に UNIX のプロセス間通信機構であるソケットが利用されている。

参 考 文 献

- 1) Engelbart, D. C.: NLS Teleconferencing Features: the Journal, and Shared-Screen Telephoning, *Proc. Fall COMPCON 75 Digest of Papers*, pp. 173-177 (1975).
- 2) Nelson, T. H.: A File Structure for Complex, the Changing, and the Intermediate, *Proc. of the ACM National Conference*, pp. 84-100 (1965).
- 3) Stefik, M., Foster, G., Bobrow, D. G., Kahn, K., Lanning, S. and Suchman, L.: Beyond the Chalkboard: Computer Support for Collaboration and Problem Solving in Meeting, *Comm. ACM*, Vol. 30, No. 1, pp. 32-47 (1987).
- 4) Conklin, J. and Begeman, L.: gIBIS: A Hypertext Tool for Exploratory Policy Discussion, *ACM Trans. Office Inf. Syst.*, Vol. 6, No. 4, pp. 303-331 (1988).
- 5) 渡部和雄, 阪田史郎, 前野和俊, 福岡秀幸, 大森豊子: 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).
- 6) Ahuja, S. R., Ensor, J. R. and Horn, D. N.: The Rapport Multimedia Conference System, *Proc. Conference on Office Information Systems*, pp. 1-8 (1988).
- 7) Ahuja, S. R., Ensor, J. R. and Lucco S. E.: A Comparison of Application Sharing Mechanisms in Real-Time Desktop Conferencing Systems, *Proc. Conference on Office Information Systems*, pp. 238-248 (1990).
- 8) Crowley, T., Milazzo, P., Baker, E., Forsdick, H. and Tomlinson, R.: MMConf: An Infrastructure for Building Shared Multimedia Application, *CSCW '90 Proceedings*, pp. 329-342 (1990).
- 9) Patterson, J. F., Hill, R. D. and Rohall, S. L.: Rendezvous: An Architecture for Synchronous Multi-User Applications, *CSCW '90 Proceedings*, pp. 317-328 (1990).
- 10) Lauwers, J. C., Joseph, T. A., Lantz, K. A. and

- Romanow, A. L.: Replicated Architectures for Shared Window Systems: A Critique, *Proc. of Conference on Office Information Systems*, pp. 249-260 (1990).
- 11) Lauwers, J. C. and Lantz, K. A.: Collaboration Awareness in Support of Collaboration Transparency: Requirements for the Next Generation of Shared Window Systems, *CHI '90 Proceedings*, pp. 303-310 (1990).
- 12) Bonfiglio, A., Malatesta, G. and Tisato, F.: Conference Toolkit: A Framework for Real-Time Conferencing, *Studies in Computer Supported Cooperative Work*, Bowers, J. M. and Benford, S. D. (eds.), pp. 63-77, North-Holland (1991).
- 13) 中山良幸, 森賢二郎, 中村史朗, 山光 忠: 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).
- 14) Tanaka, Y. and Imataki, T.: IntelligentPad: A Hypermedia System Allowing Functional Composition of Active Media Objects through Direct Manipulations, *Proc. of the IFIP 11th World Computer Congress*, pp. 541-546, San Francisco (1989).
- 15) Tanaka, Y.: A Toolkit System for the Synthesis and Management of Active Media Objects, *Proc. of the 1st Int'l Conf. on Deductive and Object-Oriented Databases*, pp. 259-277, Kyoto (1989).
- 16) Tanaka, Y.: A Synthetic Dynamic-Media System, *Proc. of International Conference on Multimedia Information Systems*, pp. 299-310, Singapore (1991).
- 17) Tanaka, Y., Nagasaki, A., Akaishi, M. and Noguchi, T.: A Synthetic Media Architecture for an Object-Oriented Open Platform, *Proc. of the IFIP 12th World Computer Congress*, pp. 104-110, Madrid (1992).
- 18) 長崎 祥, 弓崎 潔, 田中 譲: シンセティックメディアシステム IntelligentPad, 1991 情報学シンポジウム講演論文集, pp. 147-156 (1991).
- 19) Goldberg, A. and Robson, D.: *Smalltalk-80: The Language and Its Implementation*, Addison-Wesley (1983).
- 20) 長崎 祥, 田中 譲: IntelligentPad における協調作業場の実現, 第 41 回情報処理学会全国大会論文集第 5 分冊, pp. 90-91 (1990).
- 21) Ellis, C. A. and Gibbs, S. J.: Concurrency Control in Groupware Systems, *Proc. of the ACM SIGMOD '89 Conference on the Management of Data*, pp. 399-407 (1989).

(平成 4 年 7 月 20 日受付)

(平成 5 年 1 月 18 日採録)



長崎 祥 (正会員)

1966 年生。1989 年北海道大学工学部電気工学科卒業。1991 年同大学院工学研究科修士課程修了。現在同大学院工学研究科博士後期課程在学中。ヒューマンインタフェース、共同作業の計算機支援 (CSCW) に関する研究に従事。ソフトウェア科学会, 人工知能学会各会員。



田中 譲 (正会員)

昭和 25 年生。昭和 47 年京都大学電気工学科卒業。昭和 49 年京都大学電子工学専攻修士課程修了。工学博士。現在、北海道大学電気工学科教授。データベースマシン, データベース理論, メディア・ベース, 論理型プログラミング等の研究に従事。主たる著書, 「コンピュータ・アーキテクチャ」(オーム社, 共著)。IEEE, ソフトウェア科学会, 人工知能学会各会員。