

単アプリケーション方式によるグループライティング

小郷 直言・米川 覚

(平成3年10月31日受理)

要 旨

われわれはかねてより、「もんじゅ」と呼ぶ協働支援システムを構築し、相互学習法による学習・授業の支援に利用してきた。しかし、コンピュータシステムとしてTSSをベースとしていた関係上、リアルタイム性には限界があった。本稿では、もんじゅのLAN上への移植と、リアルタイム性を十分に発揮したグループライティングについて述べる。リアルタイム・グループライティングでは複数の利用者が共有ウィンドウにいつでも、自由に書き込め、しかも、十分な応答性能とWYSIWIS環境を提供することができた。

キーワード

グループウェア, グループエディタ, CSCW, 相互学習法, LAN

1 はじめに

オフィスと同じように、大学においても学習者一人一人に一台のワークステーションが割り当てられ、そのワークステーション同士が高速なネットワークにより相互に接続されるという環境が次第に普及して¹⁾きた。オフィスにおけるコンピュータ利用の発展経緯をみると、コンピュータとコミュニケーション・メディアの普及により、個人作業の支援から、最近では協調作業の支援を目指したグループウェアへの関心が高まっ²⁾ている。ただし、グループウェアにはその目標として、グループワークの効率的な支援、ひいてはオフィスワーク全体の生産性を向上させることを目指しているところがある。そのため、グループウェアを単純に教育環境に当てはめるとい³⁾うわけにはいかない。グループウェアの教育環境に

おける応用を目指すわれわれにとっては、教授形態の中の相互学習法にグループウェア的なシステムを活用し、その有効性や実現可能性を調べることが第1の目標である。

相互学習法というのは教師と学習者間の、もしくは学習者相互間の交渉が基礎になっ³⁾て行われる学習の様式である。代表的なものとして問答、討議、対話、劇化などの共同活動がこれに含まれる。相互学習法では学習者が主役であり、学習者が主体的に討議することによって、相手から学ぶところに意義がある。ただし、相互学習法の難点は、講義や自習法という他の教授法にくらべて、教師側にかなりの負担がかかり、学習者側にも相当な緊張を強いることである。このことが、相互学習法を積極的に取りにくいものになっている原因の一つである。このような困難を乗り越えて、講師・学習者双方ともが相互学習法を

積極的に活用できるようにするために、われわれはコンピュータシステムを導入し、相互学習法を支援するべきであると考えた。一般に相互学習法では、目に見える距離での対面的 (face-to-face) なコミュニケーションを想定する。しかし、情報技術の発展によって、地理的・距離的な制約を少なくして、人々はたとえ分散し離れていても十分なコミュニケーションが取れるようになってきた。オフィスで最近よく利用される電子会議、テレビ会議などにその徴候が見える⁴⁾。

2 教育の時空

グループウェアという言葉の意味内容は、今のところ人によって様々に解釈がなされているが、一般的には、人間の協調活動をコンピュータにより支援することと解釈される⁵⁾。グループウェアを大きく分類すると、リアルタイム (同期型) のものと、蓄積分配 (非同期型) のものとに分けられる。さらに、これらは空間的な特性により対面型 (face-to-face)

と分散型 (遠隔地との双方向性) とに細分される⁶⁾。いまこれらを2つの座標軸として、教育場面における教授方法の実際的な応用に当てはめてみよう。図1はそれを表している。

図では、長方形の外枠に近づくほど(⇐)コンピュータや新しい通信技術を利用したり、コンピュータで支援するという考え方は無縁な教授の方法が集められている。一方内部(中心)に向かうほど(→)今度は反対に、コンピュータを積極的に利用したり、コンピュータを教育の支援道具として用いようとする姿勢が見られる教授の方法が集められている。現在、教育におけるコンピュータの利用が進んできているとはいえ、この両者はまだまだ相入れない考え方と受け取られている。コンピュータと通信技術がいま以上に融合し、人間にとって本当に使いやすいメディアになったとき (いまかりにそれが長方形の中心に位置していると考えことにしよう)、そのとき初めて今までの教授の方法に大きな影響を与えるものになると想像される(図2参照)。

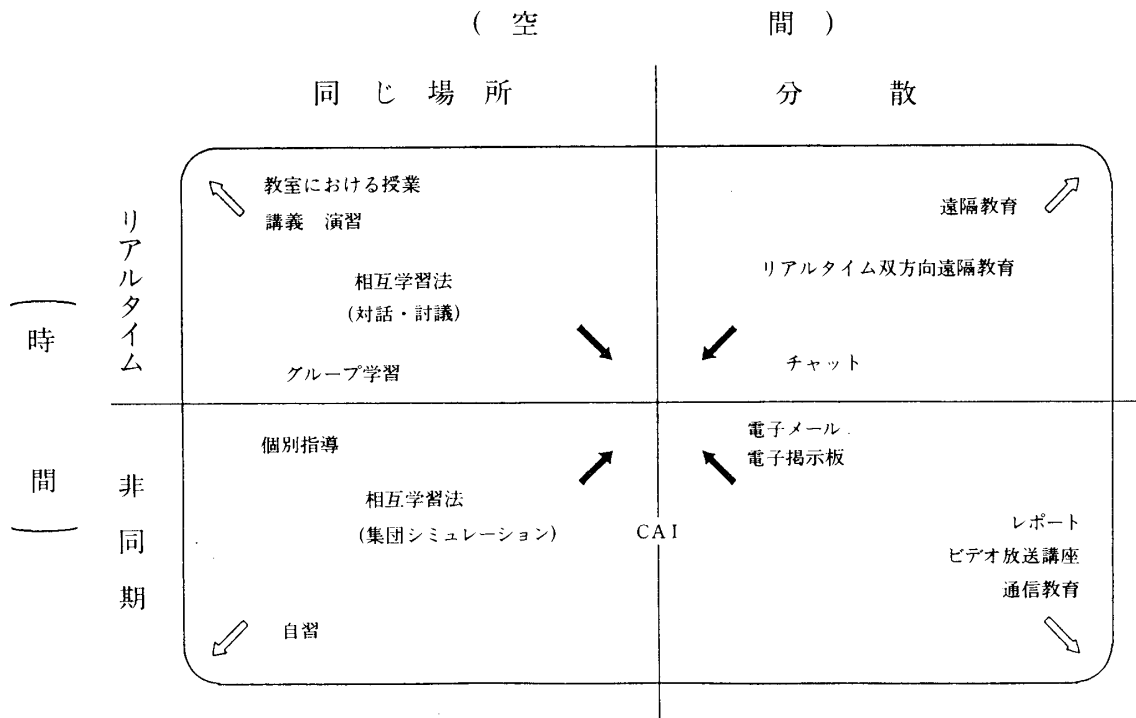


図1 時間・空間による教授方法の分類

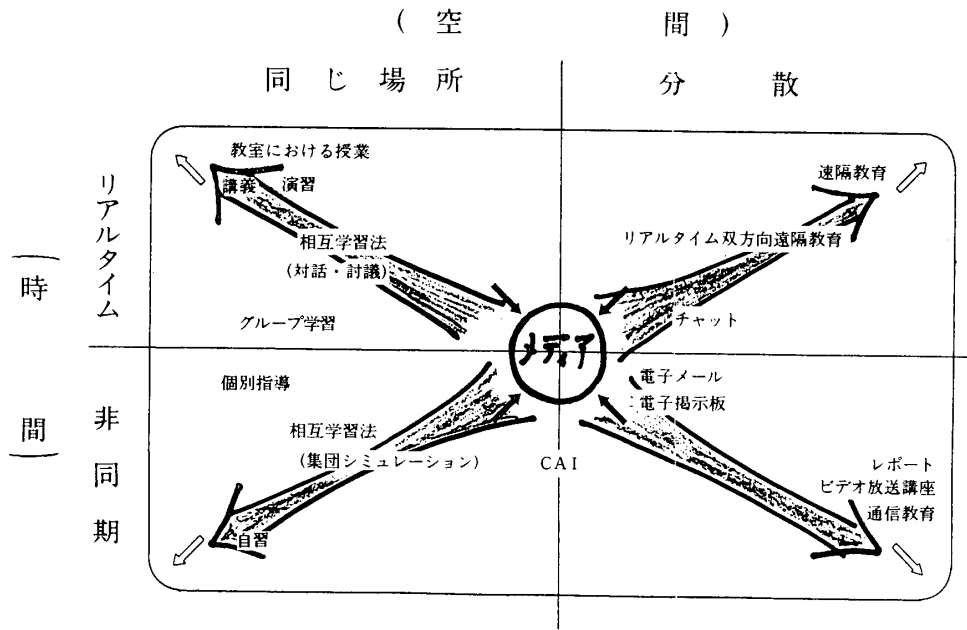


図2 メディアへの進化とその影響

2.1 もんじゅの時空

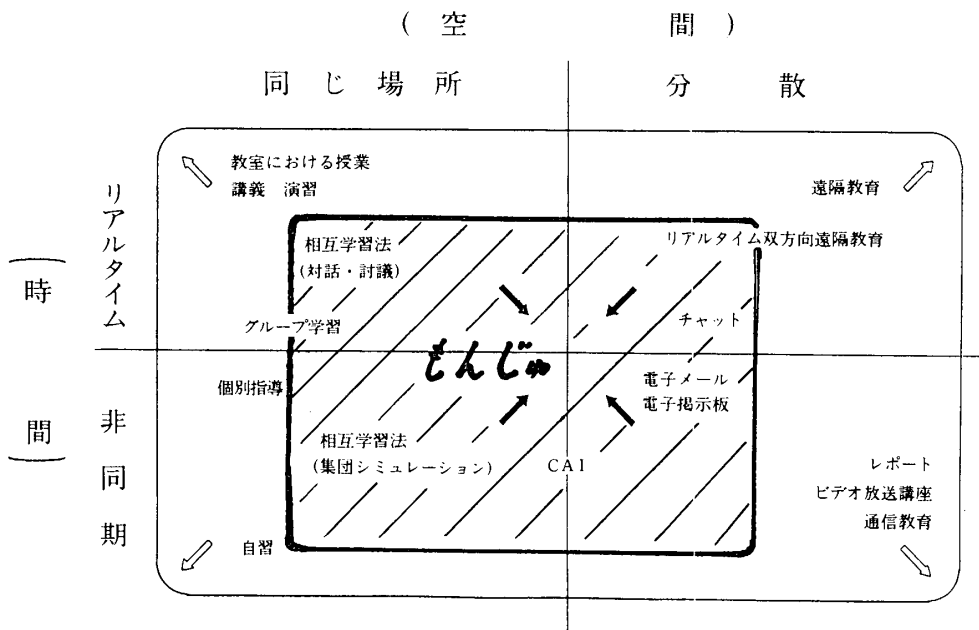


図3 もんじゅの時空

さてここで、グループウェアの教育的場面での応用を目指してきたもんじゅは、先の図1の中でどのような位置付けとなるかを示そう。図3は、もんじゅが主として機能的にカバーしている領域を示している（斜線部分）。もんじゅが本来的に機能する領域（ホームグ

ラウンド）は空間的には同じ場所で、時間的には非同期という左下にある。ただし、この領域はグループウェアの同じ分類では通常空白部分になっている場所である。もんじゅがここに本拠を置いたことには、いくつかの理由がある。

もんじゅが支援したいと考えた教育実践上の対象の多くは、実は「同じ場所でリアルタイム」(図で示すと左上の領域)な教育環境であった。しかし、実際に利用実験できるようなコンピュータシステムとしてはTSS (Time Sharing System)を前提としたため、まずこのTSSをあたかもクライアント/サーバモデルのごとく模擬して使用できるようにした⁷⁾。TSSの擬似的な並行動作では、リアルタイムなグループウェアが提供する、高度でかつ高速な並行性をサポートすることはできない。これを達成するためには、複数のワークステーションが高速なネットワークで結ばれることが、どうしても必要であった。そこで、「同じ場所でリアルタイムに」という強い指向性をもちながらも、「同じ場所で非同期」な環境から、「同じ場所でリアルタイム」性にできる限り近づけるように、様々な努力を行ってきた。電子黒板、メッセージボード、メディアの共有などがそれである。

もんじゅにおける非同期的な特性は、それぞれが思い思いに、自分が必要と感じた時点で、もんじゅのキーボードをたたけばよい。しかし、他の人と同時になにかの作業を行いたい、一つのことに注目を集めたいというときには、厳密な意味では「同時」とはいかないが、人間の時間感覚からすれば相当に速い応答性で、対話や伝達が行える環境が、それなりに用意されていた。そのために、もんじゅでは同時進行や共通な関心を持続させる技術的な工夫が要求された。プロダクション・システムによるイベント駆動方式、マルチメディアによるプレゼンテーション、合成音声の出力などがそのために利用された。

さて、図1では時間と空間の各次元に広がる平面をペダゴジー (pedagogy) として考えてきたといえる。しかし、われわれの意図している平面は実のところペダゴジーとは大きく異なる地平 (平面) で捉えた方が、より適切であることが次第にわかってきた。

2.2 新しい地平⁸⁾

ペダゴジーでは、子どもの教育については教師の指導性がほとんど全面を支配し、子供は教育の「単なる対象」、受動的な存在として扱われている。これに対して相互学習法では、教師 (指導者) と学習者、そして学習者同士の「相互協力」が、その大きな特徴になっている。この「相互的」とは、自立した学習者の意志を出発点に、その相談に応ずるという形で学習者・教師が協力する関係である。学習場面の構成のすべての面で、学習者が主体的に行動することが前提とされる。

もんじゅは、相互学習法による学習・授業を支援することを目指してきた。そして、われわれが主張してきた相互学習法は、対等な者同士の相互の学び合ひであり、学ぶことを通してお互いの成長を支援し合う関係を育てることが意図されている。そこでは教師は知識を伝達するよりも、場のコーディネーターとしての役割がより重要になるのである。もっとも重要なのは学び手である人間の主体性であり、学習の局面での学び手としてのイニシアチブである。このように考えると、教育システムは学び手の自己形成を支援するシステムでなければならないということになる。自己形成を支援するためには、学び手のイニシアチブが発揮される条件を整えることが重要である。

3 もんじゅシステムのLANへの移植

3.1 グループウェアともんじゅの比較

もんじゅは、これまでの説明からもわかるように、グループウェアあるいはCSCWと呼ばれるソフトウェアの考え方に非常に近い。しかし、以下の点でもんじゅは、一般的にイメージされているグループウェアと若干異なっている。

(a) ハードウェアの形態

グループウェアのハードウェアとしては、

パソコンネットワーク、LAN (Local Area Networks), 大規模なコンピュータネットワークなど、その用途や目的によって様々である。中でも、複数のワークステーションをLANで接続したコンピュータシステムの形態が、もっとも典型的なタイプと考えるべきだろう。一方、われわれのもんじゅシステムは、大学の授業などで多く利用されている汎用機のTSSを用いて開発した。この理由は、授業を協働の一つのプロトタイプとして取り上げようとした、われわれの考え方による。

(b) グループの大きさと形態

グループウェアの利用形態は、会議室のようなお互いの顔を見ながらできるものから、各自のオフィス、部屋同士を結んだもの、地球的規模にまで広げたものまで様々な形態が考えられる。もんじゅでは、人々が基本的には一緒のところにいることを前提に設計された(TSS端末がおかれているプログラミング演習室を利用したことによる制約)が、各自が離れたところにおいても、システムとしては何ら変わるところはない。ただし、そのような環境では全員で共有するメディアによって提供される機能は、多くの場合提供できなくなる。ハイパーメディア・システムの開発は、この欠点をカバーしたいという気持ちから始まった⁹⁾。

(c) 提供される機能

もんじゅでの電子メール、電子掲示板、電子会議などは、パソコン通信などで提供される一般的な機能に比べてかなり手が加えられており、この点では、グループウェアと変わりが無い。しかし、現在のもんじゅでは端末側のハードウェア性能から、マルチタスク機能は無く、ワークステーションのビットマップディスプレイのようなウィンドウ管理機能を必要とするサービスは提供できない(カレンダーを用いたリアルタイムな打ち合わせ、共有ウィンドウや画面上で共有される指示棒を用いての会議など)。

(d) 関係の強さ

会議を支援するシステムでは、その性格上、参加者の関係が密に発展する場合が多い。一方、もんじゅでは会議よりも多人数な授業形態を想定しているため、疎な結合が主となっている。この違いが及ぼすシステム構成上の差については、まだ検討がなされていない。¹⁰⁾ Colabが共同を意識し、少数の専門家を対象にワークステーションをベースにしたシステムを指向しているのに比べ、われわれのもんじゅは、ルーズな結び付き(Loose Coupling)を意識し、多人数(数名から50人くらいまで)を対象にTSSをベースにしたシステムを指向している。またもんじゅでは、会議(ミーティング)のような場を主に考えるのではなく、「授業」というような環境を第一義的に取り上げることから出発した。

3.2 LAN上でのもんじゅの基本構成

(1) なぜLANか

もんじゅシステムには、図3のパーспекティブからも明らかなように、もともとリアルタイム性への強い願望があった。しかし、コンピュータシステムとしてTSSをベースとしていた関係上、いま以上のリアルタイム性には限界があった。LANへの展開がその壁を打ち破る取っ掛けとなることは、他の多くのグループウェアがLAN上に構築されていることから理解されよう。LANにより全てが解決できるということでは決していないが、LANとワークステーションという組合せが、新しい可能性を秘めていることも、また確かである。

(2) ハードウェアとソフトウェアの基本構成

ハードウェアの基本構成は、図4に示すように、サーバマシン一台とユーザー用のワークステーション(クライアント側)とがイーサネットワークで接続される。ソフトウェアの基本構成はOSがUNIX¹¹⁾、ウィンドウ管

理はXウィンドウ⁽¹²⁾を用いる。汎用性を高め、
 ためにもこのような構成をとることにした。

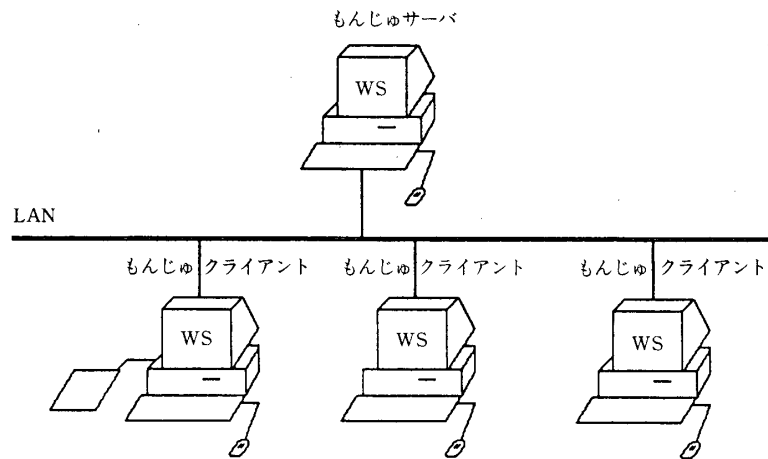


図4 LANの基本構成

(3) サーバ側

図5は、サーバとクライアントの通信路を構成する各種部品を略図化している。サーバ側では、ユーザーのワークステーション数の通信口（ソケット）が用意され、並列的にクライアントからの要求を受け入れる。マスター（プログラム）では、待ち行列からの処理要求を逐次実行していく。結果は出力側専用の通信路を使って、クライアント側に送り返される。マスターはTSSのマスターと同様に、

オブジェクト指向プログラミングが基本である。ただし、言語はC++を用いている。TSS環境でのマスターでは、Lispの特性を活かした動的環境とプロダクションシステムが提供されたが、LAN上でのこの機能はマスターに登録され独立に動作する一つの下位プロセスとして動作させ、その特性を引き継がせることを考えている。4.以降で述べるグループライティングのアプリケーションも、同じくマスターの下位プロセスの一つである。

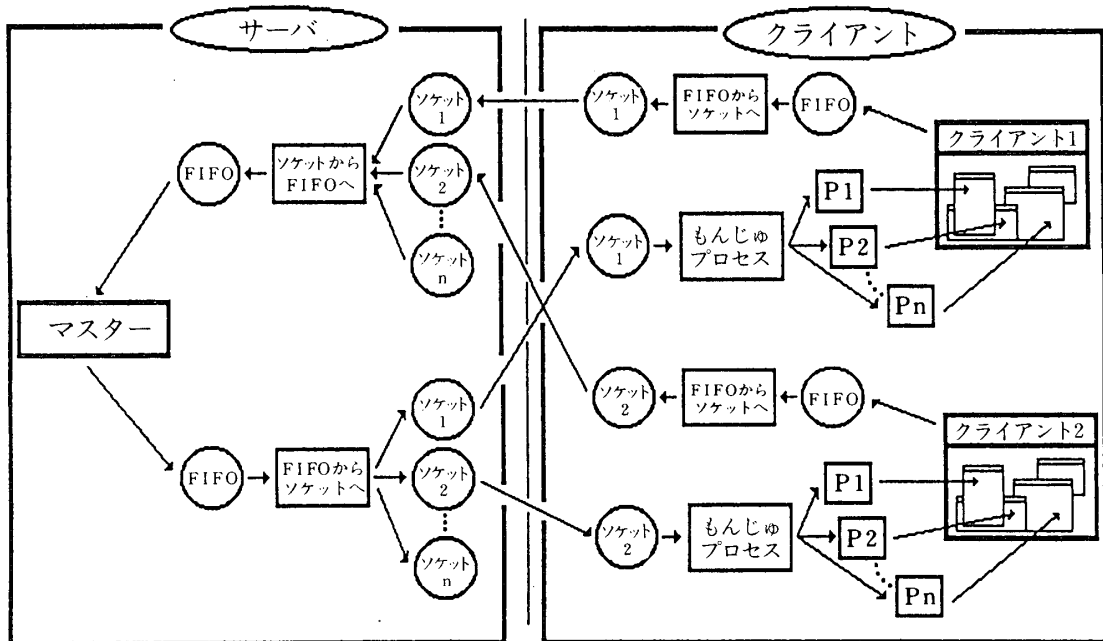


図5 クライアントとサーバ

4) クライアント側

一方クライアント側では、もんじゅプロセスと呼ぶ一つのアプリケーションが、ユーザーの処理要求に対する窓口となる。クライアント側で用意される各種機能は、もんじゅプロセスが生成する。生成された各プロセスは、統一性のあるユーザーインターフェース機能を利用者に提供する。利用者はそれをウィンドウを通して享受できる。ほとんどの機能プロセスは、個別のウィンドウに対応する形でプログラムされる。図6はその様子を示している。各プロセスからマスターへの要求は、マニュアルあるいは自動に関わりなく、統一したパケットの形で送られる。各機能プロセスの間に、もしもローカルな関係をつけたい時には、もんじゅプロセスがその中継を行う。

ドウを通して享受できる。ほとんどの機能プロセスは、個別のウィンドウに対応する形でプログラムされる。図6はその様子を示している。各プロセスからマスターへの要求は、マニュアルあるいは自動に関わりなく、統一したパケットの形で送られる。各機能プロセスの間に、もしもローカルな関係をつけたい時には、もんじゅプロセスがその中継を行う。

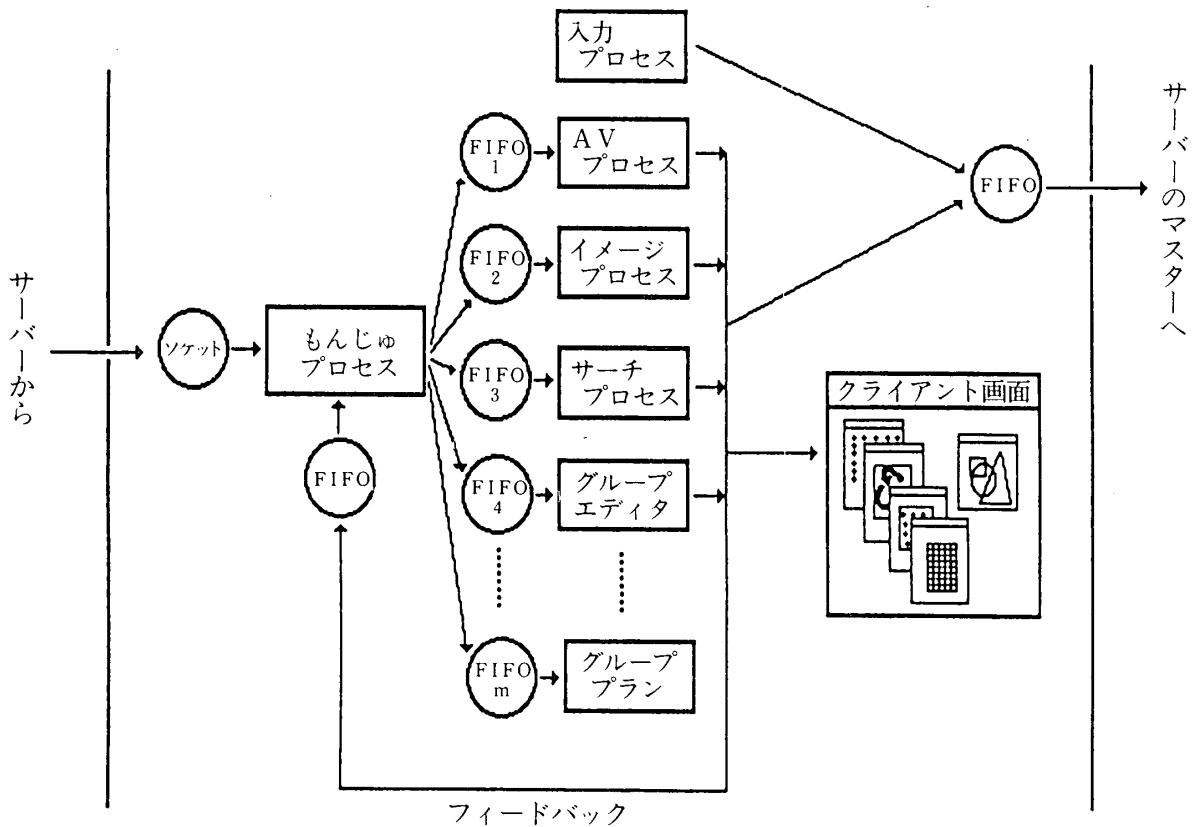


図6 クライアントの各モジュール

4 リアルタイム・グループライティング

もんじゅシステムをLAN上に移植するにあたり、まず最初にいままでのもんじゅには無かった、リアルタイム性の機能を追加することにした。その一つがここで取り上げる、リアルタイム・グループライティングと、われわれが名づけた共有ウィンドウ上での、共同執筆用のアプリケーションである。グループエディタ¹³⁾を指向してはいるが、図形も同時

に書き込みができるようにして、まず、(分散した)利用者がブレンスティングできるようなものを考えた。図7はその概略を示している。

4.1 リアルタイム・グループライティング設計上の指針

グループライティングを作成するに当たり、採用した基本的な考え方は以下の通りである。

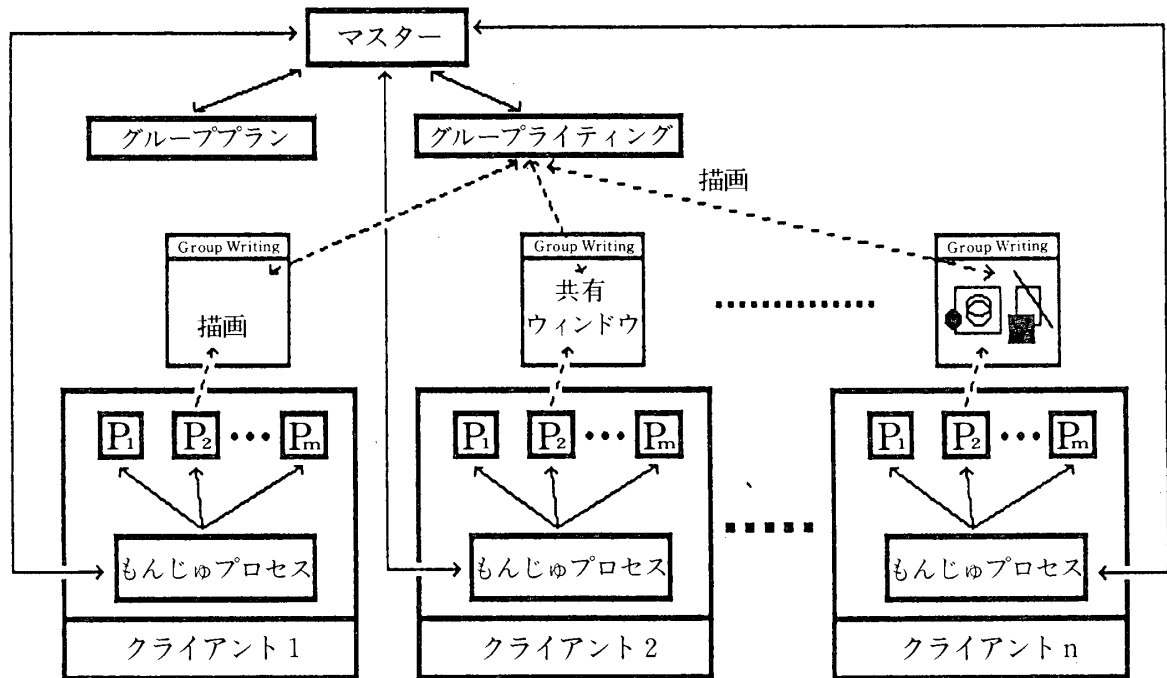


図7 グループライティング

- 利用者が書き込みたいと考えた時にはいつでも書き込めるようにする。
- 応答性能（レスポンス）を最重要視する。
- WYSWIS（What You See is What I See）環境の提供

この三つの方針は、リアルタイムなグループライティングを最も特徴づけるものといえる。ユーザーが利用したいとき、利用権や発言権のような、ある時点では一人のユーザーだけにその権利を与える方法もあるが、ここでは、ユーザー全員がなんの制約もなく、いつでも自由に思いつくままに、自分のやりたいことを行う。利用権を次々に渡していくような機能は、グループライティングの一つの下位機能として提供されるかもしれないが、基本的には全員がいつでも自由に書き込みができるようにする。

レスポンスの速さはハードウェアとコミュ

ニケーション・ネットワークのスピードにも大きく依存するが、ユーザーインターフェースを重視するという立場からも、どうしても満たされなければならない達成目標である。書く、描くという思考を伴う作業に百パーセント熱中できるようにしなければならない。レスポンスの悪さが思考のスムーズさや中断を招かないようにすることが大切である。

WYSWISの考え方は¹⁴⁾、グループライティングに参加している全員が見る共有ウィンドウの内容が常に同じであることをシステムに課している。このことで、利用者間で意図や状況の把握に統一性がとられ、しかも矛盾も少なく、利用者同士が同じように認識できる条件が整う。このことが、人間同士の協働の支援にグループライティングが役立つかどうかという問題に直接関係する。

以上の他に、ユーザー間でマルチメディア情報が利用できるようにする。グループライ

ティングでは、利用者が共有ウィンドウに表現できるメディアとして、テキストと図形（簡単な図形、矢印、手書きを含む）が用意される。しかも共有ウィンドウ上に両メディアの完全な混在を可能にしている。イメージ像をも共有ウィンドウ上に表示させることは可能であるが、もんじゅではイメージの表示機能（イメージのデータベース検索とイメージウィンドウ、それにイメージ像の同報機能など）は別に用意される。

その他に、動画と音声については、LANと並行して敷設されているCATV網を利用して、動画はディスプレイ上の1つのウィンドウとして出力される（音声はスピーカーより）。ただし、現在のところ利用者全員に開放することはできず、設備の制約上、3つのチャンネルのみを利用して動画像と音声を送受信することができる。このようなことから、多人数でのシステム利用に際して、共通の動画情報や動画の一瞬を静止画にしたイメージ情報を利用者全員で共有して、同時に見るというような使い方が有効であると考えられる。これはTSS上のもんじゅが採った方針と同じである。

4.2 グループライティング用アプリケーションの作り方

ここで、グループライティングを可能にするアプリケーション作成上の問題点、すなわち、単一のアプリケーションで行うか、複数の分散したアプリケーションで行うかについて一般的な評価を試みてみよう（表1参照）。

表1 単一あるいは分散アプリケーションの評価

アプリケーション 目 標	単一	分散
利用上の制約なし (いつでも利用でき、かつWYSIWIS)	容 易	難 しい
レスポンスの確保	難 しい	容 易

アプリケーションが単一のとき、整合性の¹⁵⁾問題は存在しないか、あったとしても、アプ

リケーションが分散¹⁶⁾して、それらを調整しなければならぬときと比べれば、明らかに解決しやすいといえる。ただし、その代わり十分に速いレスポンスを保とうとしたときには、ワークステーション数の増加に大きく左右される。単一アプリケーションの場合、レスポンスの確保は利用者の数により、一般的に次のように言える（表2参照）。

表2 利用者数とレスポンス

		レスポンス
利用 者 数	少	○
	中	△
	多	×

すなわち、利用者数が増加するにしたがってレスポンスが悪くなっていく傾向がある。ただし、利用者が少ないとはいえ、利用者が広く分散して、LAN以外の方法で広域をカバーしなければならないとすると、レスポンスを確保することは難しい。そこで上の表のように言えるのは、あくまでもLANのような高速のネットワークが利用できる時に限られる。

4.3 コンカレンシー・コントロール

まず、アプリケーション（リアルタイム・グループライティング）へのアクセスの方法から説明しよう。アクセスは、もんじゅシステムのマスターに依頼することから始まる。また、グループライティングへの途中からの参加、退席を自由に認める。

利用者の数により、次のようなアプリケーションの使い分けが考えられる。

ケース1) 3～4人まで

：1アプリケーション+Xサーバ

ケース2) ～8人ぐらいまで

：1アプリケーション+Xサーバ

ケース3) 10人以上～

: 1アプリケーション+Xサーバ
+もんじゅサーバ+もんじゅプロセス
[多人数・広域分散
: アプリケーションの分散化が必要か?]

ここでアプリケーションとは、任意のワークステーション上にあるリアルタイム・グループライティング用のプロセスのことである。

グループライティングで考慮すべき点を述べると、まず、ロング・インタラクティブなグループライティング環境を提供すること。そして、共有ウィンドウへの書込みは常時全員に許可し、アクセス権（議長権、発言権）は特に設けない。これらは結局のところ、共

有ウィンドウ制御と整合性（consistency）の確保をどのように実現するかということである。共有ウィンドウ制御は、基本的には1つのアプリケーション・プログラムとXウィンドウを用いて行った。このシステムのもとで、早い応答性を満たしながら整合性の確保をどのように達成するかが大きな目標である。

(1) ケース1 (図8参照)

これは、一つのグループライティング用アプリケーションだけで、全てのワークステーションからの描画要求、ウィンドウ表示を行ってしまう中央制御方式であるといえる。アプリケーションはワークステーションから要求のあった描画依頼を、当のワークステーションをも含め、他の全てのワークステーションに同報する（実際には逐次であるが）。描画要求が一個所に集まるグループライティングのアプリケーションでは、要求は到着順に処理されるので、整合性の問題は発生しない。同報される描画命令は、ワークステーション側のXサーバに渡されて共有ウィンドウ内に表示される。/

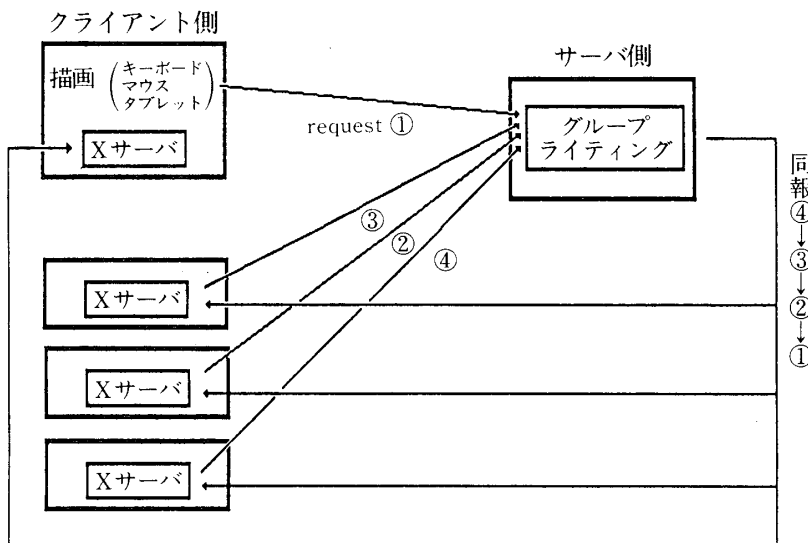


図8 ケース1

この方式の最大の欠点は、参加者が利用するワークステーションの数(正確には、共有ウィンドウの数)が増えるにしたがって、応答性が2乗のオーダーで悪くなっていくことである。これはリクエスト受付と同報しなければならない総数が、ワークステーション数×ワークステーション数で増加していくためである。

(2) ケース2 (図9参照)

このケースは、ケース1の欠点であったユーザー数増加による応答性の悪化を、見かけ上回避しようとする方式である。ケース1では、同報が無条件にすべてのワークステーションに対してなされる。しかし、応答性が問題となるのは、なによりもまず描画要求を出した

本人に対してである。これを最優先することによって、思考過程のスムーズさを妨げないようにできる。同報すべき他のワークステーションへの描画命令は、この時点では出さず、描画内容そのものを各ワークステーションごとの（スタック形式の）記憶場所に一旦しまっておく。そして、別のワークステーションからの描画要求の処理にとりかかる。

この方式は、手を動かさない者は後回しにして、描画要求者を最優先するという考え方である。ただし、全てのワークステーションからの要求がとぎれて、グループライティングのプロセスがアイドル状態になった時には、記憶テーブルを見てスタックの一番深いものから順に、たまっている描画命令を並び換え、該当するワークステーションのXサーバに渡していく。

なお、描画の整合性をとるために、あるワークステーションからの描画要求があった時には、まず、そのワークステーションの記憶場所に残存している描画要求がないかどうか調べる。もし残っていれば、まず、その描画命令を順序通りに実行した後、いま出された描画要求を処理するという手順がとられる。

この方式の欠点は、ある時点をとれば厳密な意味でWYSIWISが守られていない状況を生み出すことである。もう一つ、応答性は確かによくなるが、思いもかけない描画が唐突に表示される時には戸惑いが感じられるかもしれない。さらに、この方法では早い応答性を保つための根本的な解決にはなっていないため、ワークステーションの数が増えてくれば必然的に、応答性がケース1と同様に悪

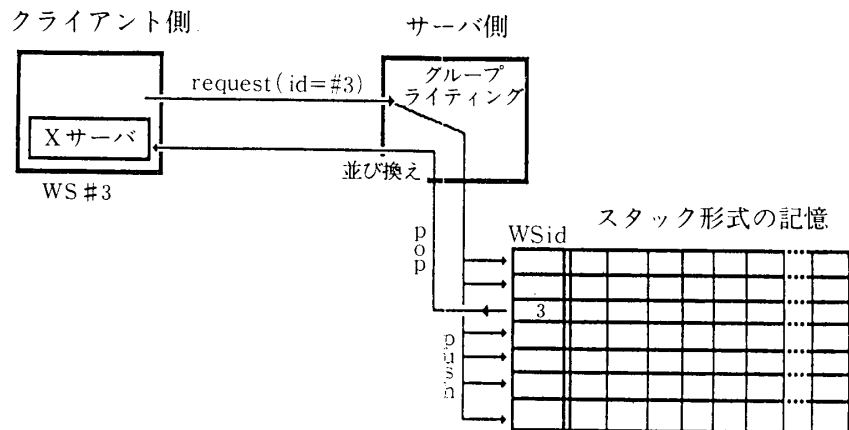


図9 ケース2

くなっていく。それとともに、WYSIWISの時間的遅延も大きくなっていくという欠点を持っている。

(3) ケース3 (図10参照)

ケースの1, 2では、グループライティングのアプリケーションそれ自身を独立して利用するシステムであった。しかし、われわれがグループライティングを導入した趣旨は、もんじゅシステムの機能充実というところにあった。グループライティングが独立して使用できるということに、別段支障があるわけではない。しかし、できればもんじゅシステム（ここでは3.2で示したLAN上のもんじゅを念頭においている）とうまく統合されているようなグループライティング・アプリケーションが望ましい。ケース3はこの目的と、ケース2が持つ欠点をなくするという考えから設計された。

ケース3の大きな特徴は、これまでグループライティング・アプリケーションがもっていた他のワークステーションへの同報機能を、全てもんじゅサーバが受け持つようにしたことである。もっとも、同報機能はもんじゅサーバが本来的に持っている機能であり、このためにわざわざ特別なプログラミングが必要に

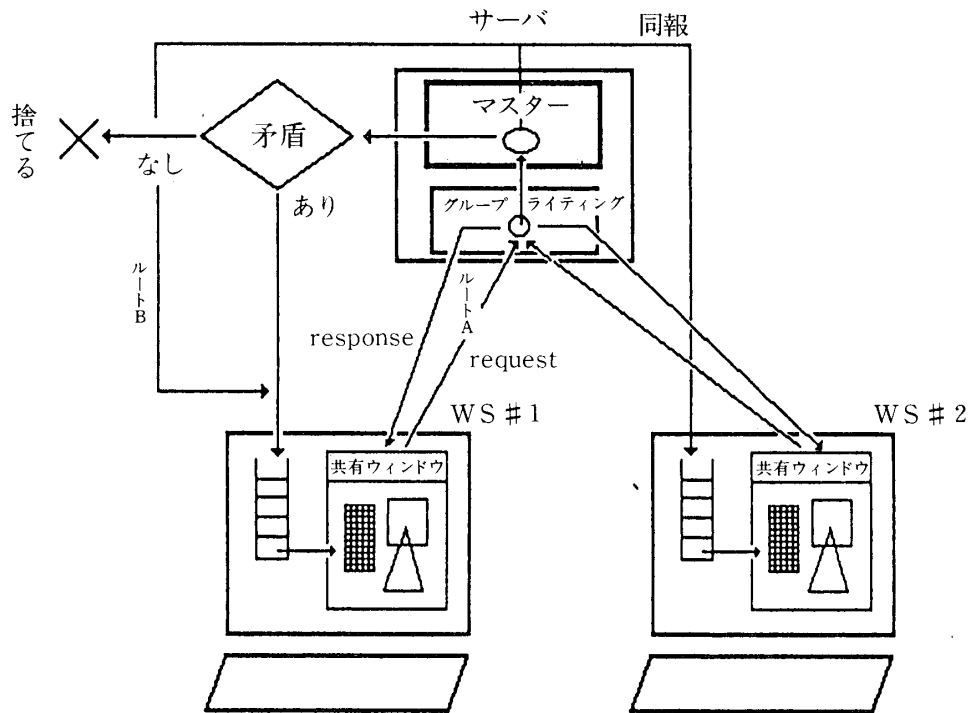


図10 ケース 3

なったわけではない。グループライティング・アプリケーションは同報機能が無くなった分、今まで以上に各ワークステーションへの応答性向上に貢献できるようになった。この場合、ワークステーション数の増加による応答速度の遅れは、ケース1とは異なり、ワークステーション数に単比例するにすぎない。

しかし、ケース1、2で起こりえなかった、整合性の確保を意識的に行わなければならないという厄介な問題が起る。一つのアプリケーションで中央制御するのではなく、始めから複数のアプリケーションでグループライティングを実現しなければならないときには、当然のこととして起る問題がここにきて少し違った形ではあるが現われる。

このような整合性の確保に対処する方法が、リアルタイムなグループエディタ実現のために、いろいろと試みられている。これは、テキスト処理の場合相当に厄介な問題となる。例えば矛盾が起こった時に、以前に実行して

しまった一連の処理を「Undo」して、もう一度正確な順序で処理を進めるという方法がとられる。また、矛盾の発生をすぐに知らせ、矛盾したままでは先に進めないようにする方式も考えられる。

応答時間を少し犠牲にするとか、自由にいつでも書き込み可能という条件を変更、あるいは緩めるかして、問題を解決するシステムも多い。MERMAID (日本電気)⁽⁷⁾では遠隔地とのリアルタイム通信ということもあり、書き込みは一時に一人だけという方式を採っている。

われわれのグループライティングでは、共有の文書領域で文字を書き込みするときに発生する矛盾の発見と、その解消に関わる論理的で挑戦的な問題を今しばらく先送りして、実用上、大きな障害にはならない程度のテキスト記入の機能を提供することで、今回はこの問題を避けることにした。

図形描画の場合には、書かれた(あるいは

消された) 順序が端末ごとに違った順番になっても、自らの矛盾を起こした描画要求を、それが実際には起こるべきタイミングで再実行されることで矛盾は解消される場合がほとんどである。

図形の場合の矛盾の発見とその解消は、以下の手順で行われる。

①グループプライティング・アプリケーションで順序付けられた描画要求は、要求してきたワークステーションに対して、すぐに実効命令が出され、ワークステーション上のXサーバによって描画される(これで応答性能が満たされる)。一方、同じ描画内容がもんじゅサーバにも送られる。

②もんじゅサーバでは、まず矛盾の検出が行われる。図形の場合の矛盾とは、他の誰かが書いている(あるいは消している)領域と同じ領域で何か(図形を描く、消すなど)しようとした時に起きたと見なす。円の内部に重ならない範囲でより小さい四角を描いても、実際には決して衝突にはならないが、ただし、

この場合にも機械的に矛盾していると解釈されることになる。

③矛盾検出は、各ワークステーションが最近時に描いた図形の二次元的な平面(矩形)領域(テーブル化されている)と比較して、領域侵害がないかどうかによって決められる。

④もしも矛盾がない時は、自分以外の他のワークステーションに同報する。

⑤もしも矛盾があった時には、自分を含めて全てのワークステーションに同報する。これはあらためて順序どおりに、自分のワークステーション上でも再描画するためである。

⑥テーブルの自分の記憶領域を、最新の使用中領域のものに書き換える。同時に時刻をも書き入れる。時刻は②, ③の他のワークステーション利用者との描画領域との衝突を検出する前に、まず、現在の時刻とテーブルに書き込まれている時刻とを比較するために使われる。少し前の描画は、すでに全てのワークステーション上に描かれているであろうから、矛盾は発生しそうにないと判断される。

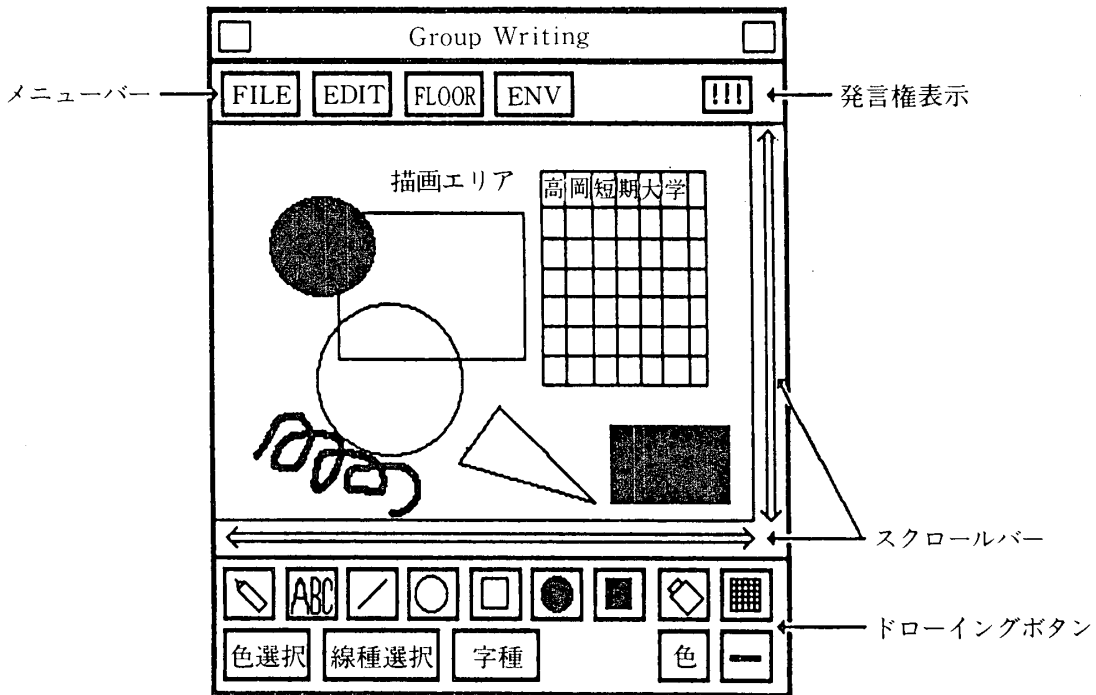


図11 共有ウィンドウ

テーブルにかかれた時刻が現在の時刻よりかなり前なら、そのワークステーションの領域内容は消し、検査は行わずにテーブルの次を調べる（これを繰り返す）。

⑦もんじゅサーバより送られてきた描画命令は、順次ワークステーション側のもんじゅプロセスが受け取り、順番にXサーバに渡されていく。

文字の場合の矛盾回避の方法については4.4(3)の「テキスト入力」で説明する。

なお、この方式では、WYSIWISの要請は若干の時間的遅れを伴うが、十分に満たされている。しかし、どれぐらいのワークステーション数まで実用に耐えられるかは、今のところまだ把握していない。

4.4 グループライティングの各種機能

ここではケース3を前提に、グループライティングの各種機能について述べる。

図11は、グループライティング用の共有ウィンドウである。このウィンドウを用いて、文字、図形、手書き、[イメージ、指示棒、音声]の利用者間での交換と共有が行われる。

(1) 図形の描画

描きたい図形を決め、イメージ化されたアイコンから一つを選び、MacPaint¹⁸⁾風に図形を描く。色も同様に選ぶ。図形には、点、直線、四角、円などが用意されている。(図12参照)

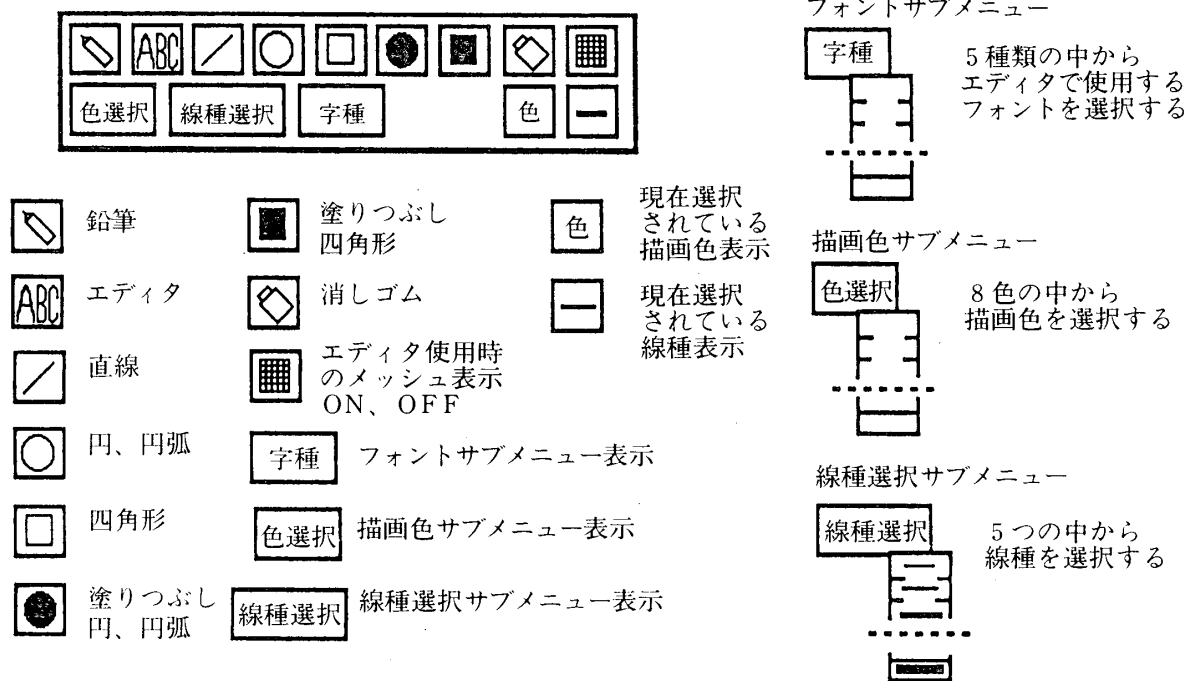


図12 描画機能のいろいろ

(2) 手書き

手書きは、点の連続として実現されているが、相談しているときや話の内容と連動して

注目をひきたい時、ちょっとしたひらめきをメモしたいとき、ある言葉とある言葉を関連づけられることを示すためなどに、とくに重

要な手段である。そこで、手書き用に特にタブレットを用意して、ペンと同じように書ける工夫を行っている。

(3) テキストの入力

まず、新規にテキストを入力する場合の手順を以下に示す。

①テキスト入力のために、共有ウィンドウの空いている（他のユーザーが現在使用中でない）領域を選ぶ。

②領域の指定は矩形（長方形）の左上と右下をマウスでクリックする。その前に、活字フォントの形とサイズを選ぶ。これにより、矩形の内部領域が、ちょうど原稿用紙のように一文字単位のメッシュに自動的に分割される。

③この時点で、グループライティングのアプリケーションは、上の②の領域要求が他のユーザーの現在のテキスト入力活動と衝突していないかどうかを検査する。

④もしも、先にその領域の一部あるいは全部が、他のユーザーにより現在使用中である領域とかち合っ、領域確保の要求が許可されない時には、その旨が知らされ、なんの変化も起きない。このときには、もう一度①に戻って作業をやり直す。

⑤領域確保が許可された時には、原稿用紙のメッシュ状の領域が黄色の枠で表示される。そして、当然のこととして、グループライティングを利用している他のすべてのユーザーにも、その矩形領域が黄色ではなく赤色の枠で、それぞれの共有ウィンドウ上にほぼ同時に現われる。（このように文字領域の確保を要求する時のみ、例外的処理となる。）

⑥カーソルをメッシュの任意の位置に置き、キーボード入力を開始する。Insert, Delete, BS など最小限のエディタ機能が使用できる。

⑦テキスト入力を終了する時は、黄色の枠の上にマウスを持っていきボタンを押す。そうすると、自分をも含めた他の全ての共有ウィ

ンドウ上の当該のテキスト領域が赤色から、水色に変化する。これにより、誰でもがその領域の取得要求を出すことが可能となる。

次にテキストの更新について示すと、

①マウスを共有ウィンドウ内の水色で囲まれた矩形の中にもっていき、ボタンを押す。（赤色で囲まれた領域は現在他の人が使用中であり、テキストの更新は行えない。）

②その領域の使用が許可されると、枠が黄色に変えられる。それと同時に他の共有ウィンドウ上の同じ領域は水色から赤色に変化し、現在使用できないことを知らせる。

③あとは、最小限の許されたエディタ機能を使って文字の追加、修正、削除を行う。

書き加えたい、修正したいテキストは、グループライティングの目的からして、自分のものと他人のものという区別はない。

(4) イメージ、動画像、指示棒、音声

イメージと動画像の制御は、クライアント側のもんじゅプロセスから起動されるイメージウィンドウ、動画ウィンドウを操作する各プロセスが受け持つ。音声は対話を可能にするためにどうしても必要であるが、現在ワークステーションが遠隔地に散らばっているわけではないので、声を出して喋ればよく、特別な装置は用意していない。指示棒は注目個所を誘導したり、焦点をはっきりさせるのに使用する。

(5) その他の機能

- ・ビデオより取り込んだ（動画の一瞬を静止画に変換した）資料をその場で、全員で共有できるようなサービス。

- ・構内CATVによる画像・音声の双方向通信が可能である。テレビ会議とグループライティングを併用した効果的な共同作業支援が提供できる。

- ・必要を感じた時には、各自の判断でいつで

も画面の情報を保存（SAVE）し、個人用のファイルとすることができる。（図13参照）。

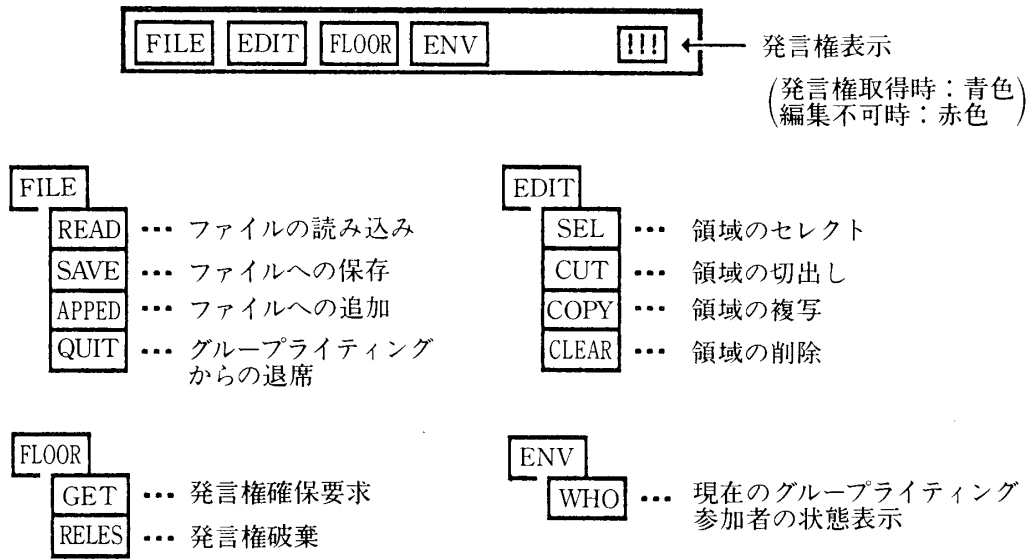


図13 その他の機能

4.5 いろいろな利用法

(1) 共同デザイン

グループライティングの作図機能を使って、複数の人の間で共同して図形、絵、表に対するデータの書き込みなどを行える（図14参照）。複数で協力して図形を描くというようなことに、これまでわれわれ自身経験がないので、何をすればよいのか、どのように協力すればよいのかわからず、しばらく暗中模索の実験的利用の範囲に留まるであろう。ただし、このような新しい試みは、デザインの過程、方法について改めて考え直す機会を与えることになるかもしれない。

(2) ブレーンストーミング

これまで説明してきたグループライティン

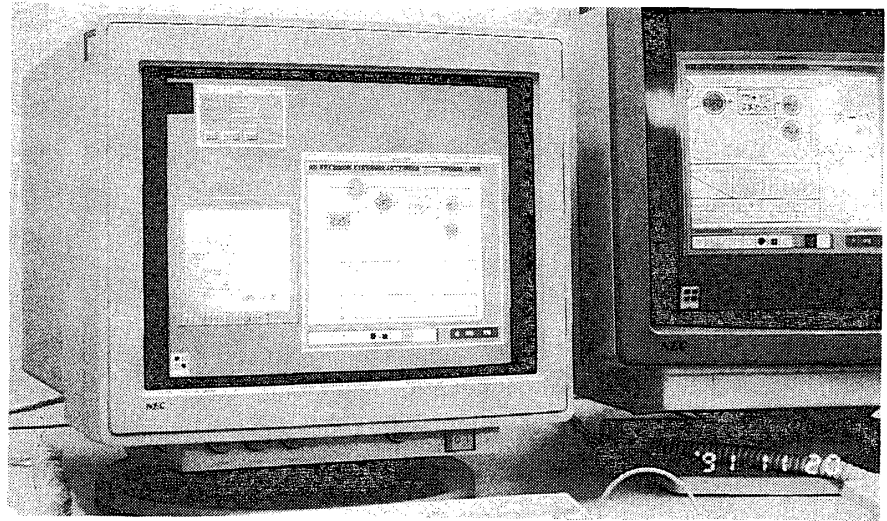


図14 描画の様子

のごく一般的な使い方としては、ブレーンストーミングがある。各自が思い思いに共有ウィンドウ（電子黒板）に、概念を表わす単語、簡単な図形、概念間の関連を示す矢印を書き込む。（図15参照）

ここで大切なことは、ブレーンストーミングが進行していく様子をよく観察して、その過程を各フェーズとして抽象化し、その各フェー

ズに最もふさわしい支援機能がなんであるかを見極めることである(4.6の2を参照)。

(3) 文章のリアルタイム添削

このグループライティングは文章と図形を混在させながら、複数の人とリアルタイムに協同執筆ができる環境になっている。ある人の文章について、意味がわからない箇所や修正してほしい箇所に手書きの線を入れ注意を喚起するなど、様々な方法で共同で文章を添削し合いながら、一つの文章に完成させていく使い方ができる。

(図16参照)

4.6 グループライティングの課題

(1) 利用実験を繰り返す

現在のところ、このグループライティングをすぐに実用化すること考えているわけではない。まずは地道な利用実験を続けていくことが必要である。また、応答性能の向上、具体的な場面にそった機能の提供、操作性などについて検討していかなければならない。

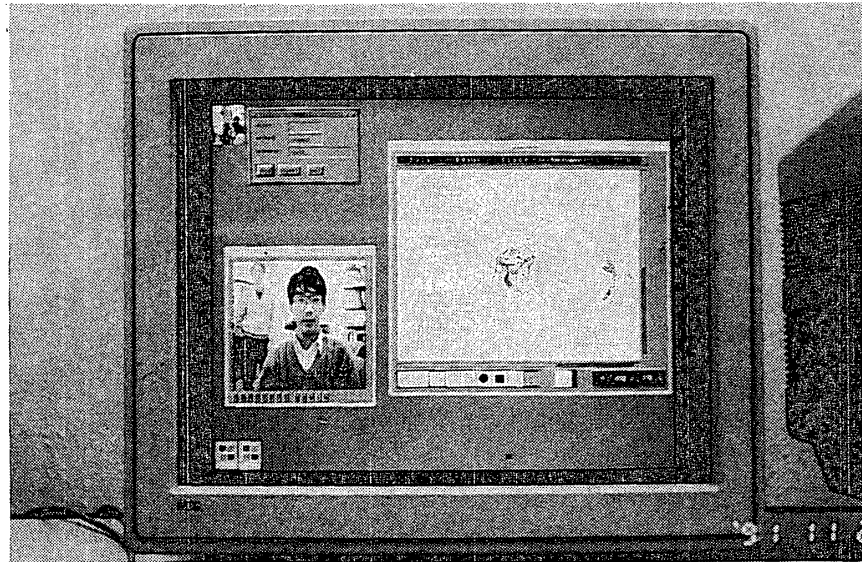


図15 ブレーンストーミング

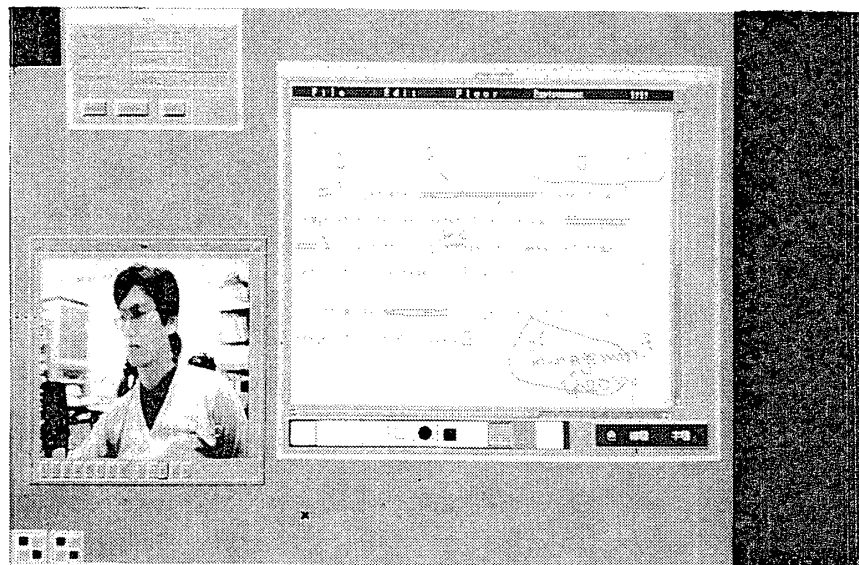


図16 文章の添削

共有ウィンドウ機能は、リアルタイムの共同作業環境を実現する手段としてきわめて有効である。しかし、複数の利用者が同時にライティング、ドローイングを行うために、今までに経験したことのない状況に遭遇して、しばしば戸惑う様子がうかがえる。これを制御するいろいろな方法も工夫できないわけではないが、今しばらくは利用実験を積み重ねることが先決であるように思われる。その意味からも、マルチユーザインターフェースと呼ばれる研究が今後注目される分野である。

(2) フェーズごとの共同支援機能

利用実験がある程度進んだところで、現在でも強く感じていることではあるが、協働支援をグループライティング内部に取り入れたシステムにしていかなければならない。現在考えているその手順を示すと、

- ①セッションの通知
- ②ブレーストローミング
- ③意見のまとめ（矢印、囲みなど）
- ④短い文章化（書込み場所不定）
- ⑤整形
- ⑥本格的な文章化
- ⑦図形の挿入

の各作業フェーズごとに支援機能を用意する。

(3) テキストの自由書込み

テキストにも、図形と同じように、「いつでも、どこにでも」自由に書き込みたいという要望が、もしもあったとしたらどのような対処すればよいのだろうか。先のケース1、2では整合性が取れているため、実際の利用上の不便さを考慮に入れないならば、そのまま利用することもできる。しかし、最も実用性の高いケース3では、整合性の問題がより深刻になり、なんらかの妥協が図られなければならないだろう。応答性を確保しながら、文字単位の低レベルまで整合性を考えて処理するとすると相当に難しいといわなければならない。

らない。

4) アプリケーションの分散化の必要性

もしも今のようなLANよりも、もっと広域で、ワークステーションが分散した環境で、同じようなグループライティングの機能を提供しようとするならば、単一のグループライティング・アプリケーションによる中央制御では当初のシステム目標は多分満たせないことになる。そこで、どうしてもアプリケーションの分散化が必要になる。

もんじゅシステムのもとで、アプリケーションの分散化を試みようとするならば、整合性を達成するためにある程度の制約はやむを得なくなるが、いまのグループライティング・アプリケーションを各ワークステーションのもんじゅプロセス配下に持ってくるのが考えられる。少し加工すれば、グループライティングをグループエディタとして模様替えすることも考えられる。

5. おわりに

グループウェアが注目された背景には、オフィスワークにおけるグループワークに対する関心の高まりがある。コンピュータ利用は帳票や文書の作成、保存、検索、配布など個人作業に対する支援を中心に発達してきた。ところが、オフィスワーカーが仕事で費やす時間を調べてみると、個人作業であるデスクワークの他に、会議、交渉、調整、面談などグループでの活動と電話などによるコミュニケーション活動にかなりの時間がさかれていることがわかる。この点に注目したグループウェアは、その目標として、グループワークの効果的かつ効率的な支援、ひいてはオフィスワーク全体の生産性を向上させることを目指していると言えよう。

しかし、われわれがグループウェアを教育環境に応用しようとする意図は、グループワークの効率的支援に対する関心よりも、いまま

での教育や学習のあり方では、実施するのが困難であると考えられていた相互学習法に、コンピュータによる支援を適用してみたいという点にあった。

その試みの一つとして、本稿では、LAN上に移植された「もんじゅ」に、グループライティングと呼ぶアプリケーションを構築した。グループライティングはリアルタイムなコミュニケーション環境で、ユーザーが同じ場所、あるいは離れているにかかわらず、どちらでも利用できる。ソフトウェア的には共有ウィンドウの制御と、整合性の確保に新しい方式を導入した。

開発されたリアルタイム・グループライティングでは、ユーザーはいつでもテキスト、図

形を書き込め、速い応答性能とWYSIWIS環境が提供され、良好なユーザーインターフェースを享受できる。ただし、今のところ、もう少し利用実験を重ねることと、具体的な場面にそった機能充実に務める必要がある。

謝辞

産業情報学科の平田道憲助教授には貴重なご助言と示唆をいただきました。厚く御礼を申し上げます。

付記

この研究は平成3年度文部省科学研究費(一般研究C)および本学特定研究経費の補助によって行ったものである。

引用文献・脚注

- 1) ただし、いまのところ、学習者はプログラミング演習などの特定の科目の授業が行われる時にだけこのようなコンピュータ環境が提供されるにすぎない。
- 2) Engelbart, D. and H. Lehtman: "Working Together", *BYTE*, 245-252(1988).
Winograd, T.: "Where the Action Is", *BYTE*, 256-258(1988).
Greif, I. ed.: *Computer-Supported Cooperative Work: A Book of Readings*, Morgan Kaufmann Pub., 1988.
- 3) 細谷俊夫著: 教育方法 第三版, 岩波書店, 1980.
- 4) Hiltz, S.R., and M. Turoff: *The Network Nation-Human Communication via Computer*, Addison-Wesley, 1978.
Vallee, J.: *Computer Message Systems*, McGraw-Hill, 1984.
Rapaport, M.: *Computer Mediated Communications*, John Wiley & Sons, 1991.
- 5) Winograd, T.: "Groupware: The Next Wave or Just Another Slogan?", *Proceedings of IEEE COMPCON*, Spring, 1989, pp.198-200.
- 6) 石井裕: "グループウェア技術の研究動向", 情報処理, **30**-12, 1502-1508(1988).
- 7) 小郷直言, 米川覚: "コンピュータによる協働支援システムの基本的枠組(1)——メッセージ通信とサーバの設計——", 高岡短期大学紀要, **2**, 19-33(1991).
- 8) 倉内 史郎: 社会教育の理論, 第一法規出版, 1983.
新井 郁男: 学習社会論, 第一法規出版, 1982.
池田秀男他: 成人教育の理解, 実務教育出版, 1987.
社会教育基礎理論研究会編著: 学習・教育の認識論, 雄松堂出版, 1991.
- 9) 小郷直言, 米川覚: "ハイパーメディアシステムの構築とその教育における利用", 高岡短期大学紀要, **1**, 47-66(1990).
- 10) Stefik, M., Foster, G., Bobrow, D., Kahn, K., Lanning, S., and Suchman, L., "Beyond

the Chalkboard:Computer Support for Collaboration and Problem Solving in Meetings”,
Communications of the ACM,30-1,32-47(1987).

11) UNIXはAT&Tベル研究所が開発したオペレーティングシステムである。

12) X-window はMITが開発した。

13) 例えば, Colab (Xerox Corporation), GROUVE (MCC, Microelectronics and Computer Technology Corporation)

14) Stefik,M., Bobrow,D., Foster,G., Lanning,S., and Tatar,D.: “WYSIWIS Revised:Early Experiences with Multiuser Interfaces”, *ACM Transaction on Office Information System*, 5-2,147-167(1987).

15) Greif, I. and S.Sarin: “Data Sharing in Group Work,” *ACM Transactions on Office Information*, 5-2,187-211 (1987).

Sarin, S. and Greif, I. : “Computer-based Real-time Conferencing Systems”, *IEEE Computer*, 18-10, 33-45(1985).

16) Ellis,C.A., S.J.Gibbs, and G.L.Rein : “Groupware : Some Issues and Experiences”,
Communications of the ACM, 34-1, 39-52(1991).

17) Watabe,K., et.al.:“Distributed Multiparty Desktop Conferencing System:MERMAID,”
CSCW'90, Los Angeles, October, 1990.

18) MacPaint は Apple Computer 社のソフトウェアである。

Group-Writing with Single Application in Local Area Networks

Naokoto KOGOU and Satoru YONEKAWA

(Received October 31, 1991)

ABSTRACT

We have developed a real-time group-writing system in LAN and workstations. This system is designed so users can write and draw on the common window at anytime, and users can receive a rapid response and WISWIS environment.

KEY WORDS

Groupware, Groupeditor, CSCW, LAN