


共有透明ボードシステムの試作 ～ COGENT SHARE ～

堀川 桂太郎 桑名 栄二

NTT  ソフトウェア研究所


概要

従来の多くの共有ホワイトボードシステムでは、多様なメディアで表わされた議論の対象を共有する機能とユーザの振舞い(Awareness), 議論の焦点, 注釈メッセージ(Annotation)を共有する機能が別々に提供され, それらの間の情報交換やデータ変換に複数ステップの操作が必要であった. これに起因するコミュニケーションの即時性と連続性の損失に対して, 我々は単純なメカニズムで実現した透明なウィンドウを共有アプリケーションのウィンドウに重畳させ, AwarenessとAnnotationのグループインタフェースを整理することで, 議論対象・焦点・注釈メッセージを統合的に共有するシステム COGENT SHAREを設計・試作した.

本稿では, COGENT SHAREの機能設計と特徴, 透明ウィンドウの簡単なインプリメント法およびAnnotationツールとしての描画インタフェースなどについて述べる.

Development of the shared transparency board system : COGENT SHARE

Keitaro HORIKAWA, Eiji KUWANA

NTT  Software Laboratories

E-mail: {horikawa,kuwana}@slab.ntt.jp
3-9-11 Midori-cho Musashino-shi Tokyo 180 Japan

Abstract

It is important for shared whiteboard systems to support the functions to share the following factors: the objects of arguments, the behavior of users(awareness) the focuses of the arguments and the annotation messages(annotation). In traditional systems,however,these functions are usually supported by separately and a number of command steps are necessary for the information exchanging or data converting; therefore, instancies and continuities of the communication have been disturbed.

In order to eliminate interruptions in consideration originated by these loss of instancies and continuities and to support smooth and effective collaboration, COGENTSHARE has covered a shared application window(as a argument object represented by multimedia) with shared transparency windows. Consequently COGENTSHARE provides the functions to integrate and share the distributed argument object , focuses and annotation messages.

In this paper,we describe the COGENT SHARE design and its features ,a simple method of implementing transparent windows and the drawing interfaces as a annotation tool.

1. はじめに

我々の研究グループでは、協調的なグループワークにおいてより良い効果を引き出すグループウェアの実現をめざし、協調作業支援環境COGENTの設計・開発を進めている[13]。ここではCOGENT環境のひとつの構成要素として共有ホワイトボードを設計・試作したことを報告する。

今回試作したシステムCOGENT SHAREは、Maloneのコーディネーション理論[3]および文献[1][2]で提案された機能モデル(Groupware Functionality Model, GFM)を参考に、ユーザ間の効率的かつ直観的な意志の伝達をサポートするために、特にAwarenessとAnnotationなどのグループインタフェース機能を強化したシステムである。実時間の会議において、「どこの誰がどのようにして何を行っているか」を(暗に)表現するAwarenessと明示的な注釈メッセージであるAnnotationを共に提供することで参加者のコミュニケーションが効率化すること、そして(通常はテレポインティング技術などで提供される)AwarenessとAnnotationは、ベースとなるウィンドウシステムが透明なウィンドウをサポートしていれば容易に実現できる、ということは早くから指摘されていた[9]。しかしながら、既存のウィンドウシステムは透明ウィンドウをサポートできてなく、その実現は単純ではなかった[9][11]。

本稿で述べるCOGENT SHAREは、現在ワークステーションやパソコンUNIX上で一般に広く利用されているXウィンドウシステム(X11R4以降)をベースに、シェイプ拡張機能を適用することで非常に単純かつポータビリティのある手法で透明ウィンドウを実現し、それをアプリケーション共有ウィンドウと重ねることで統合的なグループインタフェースを提供するものである(図1参照)。

以下、2節にて共有ホワイトボードに対するグループインタフェースの必要性、3節にてCOGENT SHAREの機能設計と使用例、4節にて透明ウィンドウのインプリメント手法、5節にてAnnotationツールとしての描画インタフェースについて述べ、6節で考察および今後の課

題をまとめる。

2. 共有ホワイトボードに対する要件

Maloneのコーディネーション理論[3]および文献[1][2]の機能モデル(Groupware Functionality Model, GFM)の考え方を共有ホワイトボードシステムにあてはめると、以下のように「議論の対象, 焦点, 注釈メッセージ」の3つの統合的な共有とそのためグループインタフェース(Group Interface)を提供することが重要であると言える。

2.1 議論対象の共有

ミーティングなどに見られるグループの協調作業では、参加メンバーの会話(音声)だけによる説明や対話よりもレポートやメモなどの資料、議論対象となる実物や模型、あるいはそれらの図面・写真・映像を用いた方が明らかにコミュニケーションの効率がよい。これは[2]の共通オブジェクトの認知プロセスに相当し、通常はWYSIWISの原則として全ユーザに同じ画面情報を分配することで提供される。

2.2 議論の焦点の共有

会話と同時に実際に(どこの誰が)対象のどこに注目して説明・質問しているのかという焦点情報を他のメンバーに示しながら議論を進める方が相手との意志の疎通はるかに容易である。これは[2]のAwarenessに相当し、通常はマウスカーソルのテレポインティングなどで提供される。

2.3 注釈メッセージの共有

その場で議論対象の該当する部位に関連づけてコメント・意見・疑問点を書き記す事は、補足説明を与え議論を整理しメンバー間の共通の理解を促すという効果をもっている。これは[2]のコミュニケーションプロセスにおけるAnnotationに相当し、通常はホワイトボードへの描画機能によって実現される。

2.4 統合的な共有の必要性

共有ホワイトボードは現在までにいくつか実現されている[4][5][6][7][12]。例えば[6]などは上述の「議論の焦点の共有」に関し参加ユーザ1人につき1つのマウスカーソルを提供し指示棒の代わりにポインティング操作できるマルチカーソル機能を有し、「注釈メッセージの共有」に関しテキストに加えて矢印・スタンプ・マークなどを描画できる。

しかしながら「議論対象の共有」について、多様な情報メディアで表現された対象を共有することを考えると、例えば既存の個人用アプリケーション(AP)で作成されたフォーマットの異なる文書を引用して共有ホワイトボードに表示させるためには、そのシステムが要求する特定のファイルフォーマットへ変換して改めて読み込むか、あるいはAPの画面をスナップショットを用いて一度ビットマップイメージなどで保存した後に(必要ならフォーマット変換して)改めて読み込むというように複数ステップの操作が必要となる。

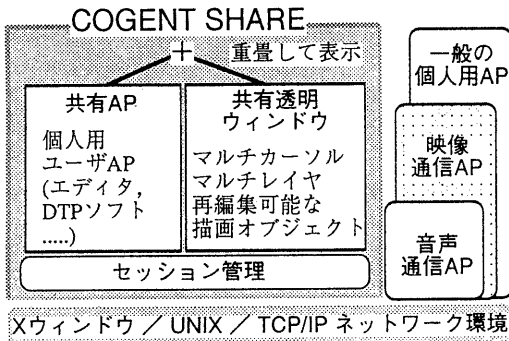


図1. COGENT SHAREの概念と位置付け

また、共有して説明したい対象が時間と共に動的に変化するような映像（連続）メディアで説明の焦点や注釈もそれらに連動させたい場合になると、スナップショットの連続保存は明らかに説明の連続性と即時性を損なうことになる。つまり、議論対象の分散共有の要件を満たすための機能と議論の焦点と注釈メッセージの分散共有のための機能が別々に動作し、それら間の情報交換やデータ変換に複数ステップの操作が必要になるような従来の環境では、著しくコミュニケーションの即時性と連続性を損なうことになる。

そこで、これらの即時性と連続性が損われることに起因する思考や作業の中断をなくし、協調作業を円滑かつ効率的に支援するために、上述の協調作業における議論対象・焦点・注釈メッセージを統合的に分散共有する機能を提供することが今回設計・試作したCOGENT SHAREに対する要求条件である。

3. COGENT SHAREの機能設計

図1にCOGENT SHAREの概念と位置付けを示したように、COGENT環境では音声通信と映像通信システムは既に別に用意されている。従ってCOGENT SHAREは共有ホワイトボードとして単体で使うのではなく、これらの音声と映像通信システムと共に利用することになる。前述の要求条件に向けて、COGENT SHAREでは以下の構成を導入している。

3.1 APの分散共有

多種多様な情報メディアで表現される議論の対象を分散共有するために、ワークステーションのウィンドウシステム上で稼働する任意のAPについて入力操作およびすべての出力表示を分散共有する機能を導入している。これは、機能モデルGFMの「共通オブジェクト認識プロセス」における対象の範囲を拡張する。つまり、議論の対象としてマルチメディア技術を駆使したAPを共有することで、より「百聞は一見にしかず」の効果が発揮され、コミュニケーションの効率向上が期待できる。

AP共有のもう一つの効果は、ユーザーが使い慣れた個人用のAPをグループワークの場でも同様に活用できるという意味の連続性である。COGENT SHAREの現在の設計では、AP共有の方式[8][10]として単純化の理由からxtv[5]やOpen GroupMedia [11]などと同様に集中型アーキテクチャに基づきXプロトコルを横取り・分配する疑似サーバを利用している。ただし、入力操作権の受渡し方式は、要求すれば直ちに権利が取得できる方式のみを実現している。

3.2 透明ウィンドウの重畳

図2にCOGENT SHAREにおける表示画面（ウィンドウ）のレイアウト構造を示した。共有APの表示領域に関連した注釈メッセージを共有するためにソフトウェアで実現した透明なウィンドウを直接かぶせている。これにより、APの表示内容をスナップショットすることなく、動いているAPの上に直接Annotationを与えることが

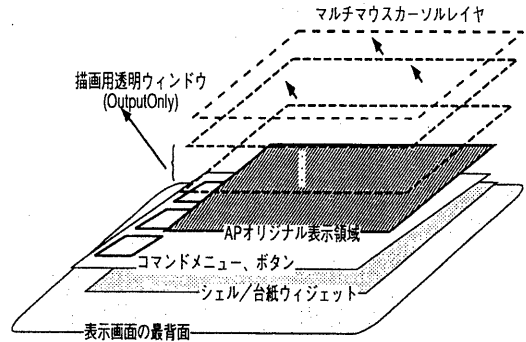


図2. 表示画面のレイアウト構造

可能となる。

透明ウィンドウ上に描画した図形や文字が、議論の対象であるAPを背景として描かれる注釈メッセージの役割を持つ。それらはすべて描画オブジェクトとして管理するので、一度書き終えた後に移動・複写・削除・編集取消など再編集が可能である。注釈メッセージが議論対象を補足したり強調するための単なる印ではなく、議論の過程でそれらの注釈やメッセージ自体が議論の対象になった場合にこの再編集機能が有効となる。また透明ウィンドウは（現在の設計では）基本的にメンバ1人につき1枚所有するという形でマルチレイヤ構造をもっているため、各メンバは他メンバの描画操作とは独立に任意の時点で注釈メッセージを書き込むことが可能である。

共有APの操作権を所有しているユーザーの入力データのみが共有APへ渡され、それ以外のユーザーの入力はすべて各自が所有する透明ウィンドウに対して送られる。

3.3 共有透明ウィンドウ上のマルチカーソル

共有透明ウィンドウは、他のメンバの動かすマウスポインタの位置を追跡し全メンバの画面上の透明ボードに疑似的なポインタとして随時表示する機能を有しているので、Awarenessおよびリアルタイムな議論の焦点の共有が可能となる。またマウスカーソルの形状と色をユーザーごとに一意に与え、ウィンドウの枠上にカーソルとユーザー名、ホスト名の対応を表示することで、「どこの誰が」という情報を示す。共有APウィンドウ上に移動したマウスカーソルの動きも追跡して表示する。

3.4 途中参加/退出可能なセッション管理

ユーザー情報と描画オブジェクトを管理することで、セッションの途中でユーザーが退出したり参加することをサポートする。またAPの表示画面と透明ウィンドウ上のAnnotation情報はビットマップ形式でマージしてセーブ/ロードできる。

使用例： 以上の機能を実現したCOGENT SHAREの利用画面の例を図3に示した。この例では、個人用のドローツールtgif+をAP共有し、その上のドキュメント（コンピュータネットワーク設計図）を議論の対象に、4人の参加者がそれぞれ透明ウィンドウを重ねてその上

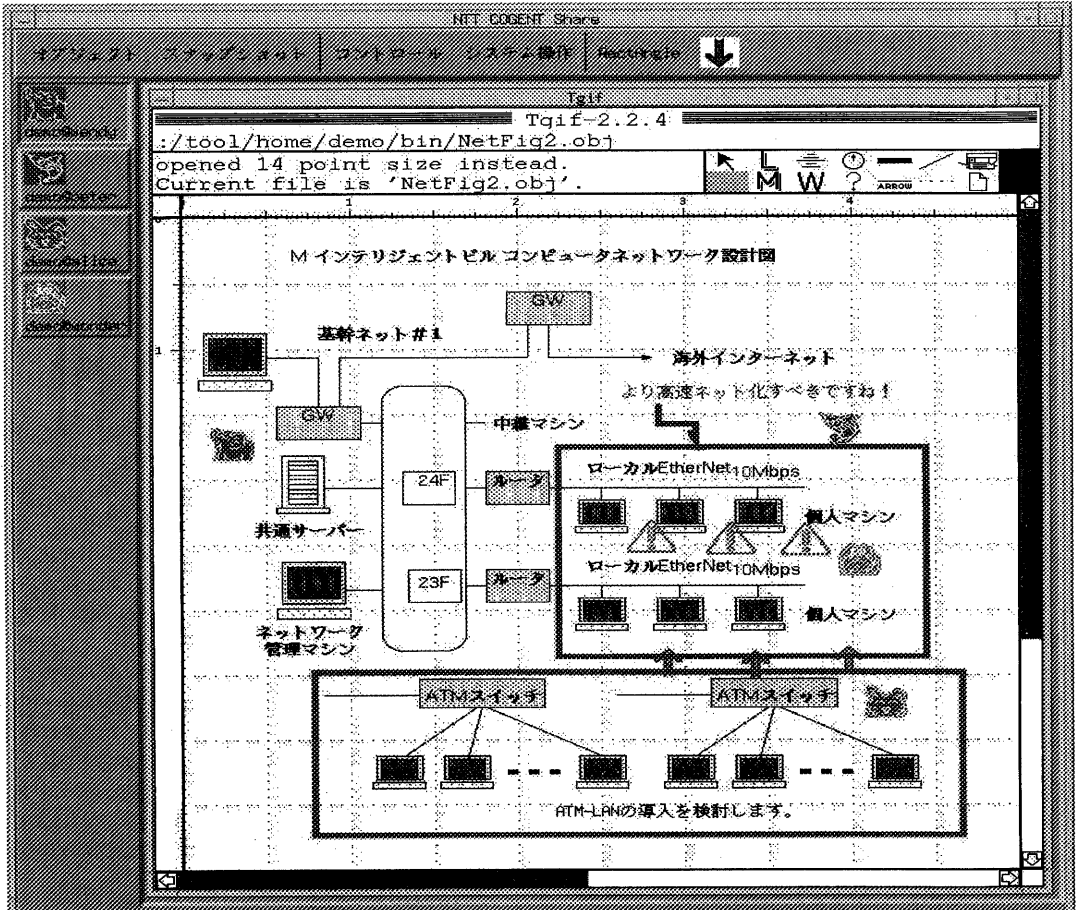


図 3. COGENT SHAREの利用画面 (tgif+を共有)

にAnnotation (矩形, 矢印, テキスト, スタンプ) を書き込んでいる様子を示している。外枠のウィンドウの左側に参加ユーザのマウスカーソルと対応してアイコンがユニークな形状 (動物の形) で表わされ, 各アイコンの下にユーザ名とホスト名が表記されている。

4. 透明ウィンドウのインプリメント

COGENT SHARE実現の中心的課題だったのは、「透明ウィンドウ」の簡潔なインプリメント手法である。

ベースとなるウィンドウシステムが透明なウィンドウ機能をサポートしているならば, テレポインティングや注釈メッセージの機能が容易に実現できる, ということは早くから指摘されていた[9]。

今回試作したCOGENT SHAREはUNIX/Xウィンドウ環境で動作する。Xウィンドウは基本的に透明なウィンドウはサポートしていないが, COGENT SHAREでは, X11R4以来MIT Xコンソーシアムの標準となっているシェイプ拡張機能[14]を利用することで非常に単純なメカ

ニズムで透明ウィンドウ機能を実現した。

4.1 シェイプ拡張機能の適用

シェイプ拡張機能を利用すると長方形でない任意の形状のウィンドウが定義できる。例えばXクライアントのoclockやxeyesなどは円形のウィンドウとして定義され, そのウィンドウの領域以外は透けて見えている。極端に言えば, 1本の線やビットマップ図形, テキストでウィンドウを定義することも可能である。COGENT SHAREでは, これを応用して描画オブジェクトを定義し, 透明ウィンドウ上の注釈メッセージやテレポインティング機能を実現している。

具体的な描画処理の手順としては, 図4に示すように (内部的な白紙の) Pixmapに現在の描画モードにしたがってオブジェクトを描き, それを既存のシェイプウィンドウと併合することを繰り返す。なお, 描画の際はラバーストリップを表示する。

4.2 共有AP上への透明ウィンドウの重畳手法

共有APウィンドウの親ウィンドウをルートウィンド

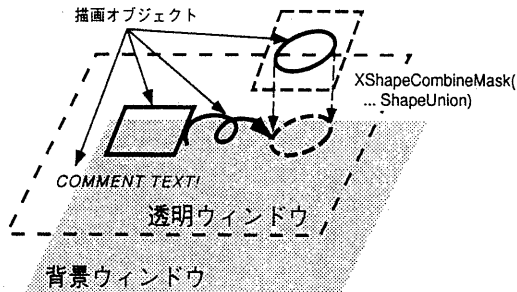


図4. オブジェクトの描画と透明ウィンドウ

ウからCOGENT SHAREウィンドウのシェルウィンドウ, 台紙ウィンドウに変更し (XReparentWindow()), 予め用意したシェイプウィンドウをraise (XRaiseWindow()) することで, 共有APのウィンドウ上に透明ウィンドウを重畳することができる。

4.3 性能上の問題

シェイプ拡張を使った透明ウィンドウは簡単に生成される反面, 実際の運用には以下の点に注意する必要がある:

- ・クライアント側では再描画の処理を考える必要がない反面, Xサーバの負担が大きい。
- ・曲線形状や斜め線形状の描画オブジェクトを多用すると再描画時のXサーバの負荷が急激に増大する。

したがってシェイプ拡張を多用することは性能に大きく影響する。この点について今回の試作では, 実現の容易性とポータビリティの観点を優先する意味で, シェイプ拡張方式を採択している。

5. Annotationツールとしての描画インターフェース

COGENT SHAREの共有透明ウィンドウ部では, 再編集可能なオブジェクトを基本とする描画機能を提供している。つまり一般のドローツールと同様に, 一度描いたオブジェクトを選択, 削除, 変更, 移動, 取消, 再実行, 複写, 貼付することができる。

ところが, 描画の大半の用途が注釈メッセージ(Annotation)を書き込むことである, という事実に着目すれば一般のドローツールと同様な自由度の高いインターフェースは, 複数ステップの操作を必要とする分, 迅速なコミュニケーションにとってマイナスであると考えられる。

そこで典型的な注釈の与え方を分析し, そのパターンを<region,link,message>の3つで一般化してまとめて扱うことで, より少ない操作ステップで注釈を書き込めるようにした。

図5にAnnotationの表記法, region,link,messageそれぞれの部品, そしてlinkの方向とデフォルトの接続形態の例を示す。予めregion, link, messageの部品 (と属性) を選択しておき, 注釈を与える時に注釈の対象領域(region)と注釈を書きたい領域(message)をマウス操作で

指定すれば, regionとmessageそれぞれ現在選択されている部品がマウスで指定した領域分の大きさで描画され, その間をlinkの選択部品が (デフォルトの接続形態で) 接続し, message領域内でテキスト入力待ち状態になる。このようにregion, link, messageの3つを一括してAnnotationとして扱う機能を提供することで, 繰り返される冗長な操作を省き迅速なAnnotation操作を可能とする。

6. 考察

プロトタイプ在完成とそれをデモンストレーションで実際に利用したユーザからの意見を参考に, 以下の考察を行う。

6.1 COGENT SHAREによる協調作業支援

上述の機能を実現したCOGENT SHAREを用いれば, 次のように協調作業を支援できる。

- (1) APの表示領域を背景に透明なウィンドウを重ねて注釈メッセージを随時書き込める。またAPの入力操作を共有しているのでその場で意見やコメントを反映し

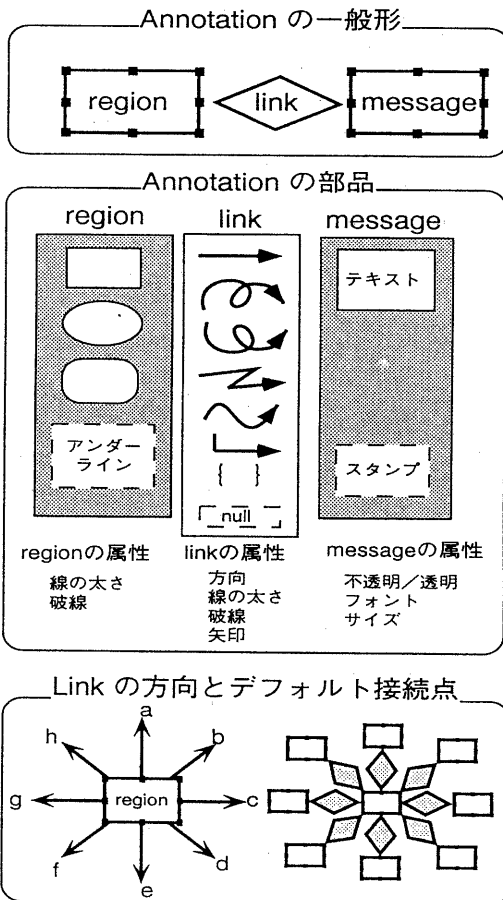


図5. Annotationの表記法

た修正や対処が可能となる。

(2) 動作しているAPの出力表示を議論の対象として直接共有できるので、紙ベースの資料や電子ファイルの配布が不要になる。またマルチメディア情報に対するWYSIWISが確立する。

(3) 各ユーザはマウスポインタを指示棒の代わりに使って発言できるので、「資料xページの図yの右からz番目の色が…」などという冗長な説明を「この色が…」で済ませることができる。

上記(2)(3)は従来のシステムでも個々の機能として提供されている。COGENT SHAREでは共有APと共有透明ウィンドウを統合した上でその機能を提供していることに意味があると考えられる。つまりCOGENT SHAREでは、多様なメディアで表現された議論対象/議論の焦点/注釈メッセージの共有を効率的に統合し、作業の連続性・即時性を提供することでコミュニケーションの円滑化と協調作業を支援する効果をもたらす。

また、COGENT SHAREにおけるAP共有メカニズムは集中型アーキテクチャかつXプロトコル横取り分配型であり、通常の個人用AP(Xクライアント)ならば、それ自体に何も手を加えることなくグループで共有できる。さらに透明ウィンドウ実現メカニズムはシェイプ拡張を利用した非常に単純なものであり、現在広く利用されている(X11R4以降の)Xサーバなら問題なく動作する。従ってより広範な環境で活用できると考えられる。

6.2 関連研究との比較

MERMAID[10]は、AP共有方式として分散型アーキテクチャかつAP変更型であり性能向上を重視し、Annotationについては予めグループAPとして用意した黒板を使うことになる。共有APウィンドウに透明ウィンドウを被せるという考え方ではない。Open GroupMedia[11]は、例えばOHPシートなどの透明ウィンドウメタファの必要性を唱えていて、その設計思想においてCOGENT SHAREと共通する部分が多い。COGENT SHAREは透明ウィンドウをシェイプ拡張を利用した非常に単純なメカニズムで実現している。ShowMe[6]は共有APと共有ホワイトボードがサブシステムとして別々に動作している。共有ホワイトボードからLook Throughモードで共有APを操作することができるが、それを全員で共有するためには、別途スナップショット化する必要がある。

6.3 今後の課題

COGENT SHAREの実現により、動いているAPの表示画面とそれに対応づけたAnnotationとの連携で、非常に興味深い新しい問題が得られている。つまり、現時点でAP表示画面と透明ウィンドウ上のAnnotationを結び付けているのは、単にオーバーレイしているという関係だけであり、その連携を保存するための方法としてはビットマップ形式ファイルしかサポートできていない、AP画面のスクロール(などの変化)とAnnotationの連携の保存するための有効な方法を現在検討している。例えば時

間軸方向の表示画面の記録・制御機構などを導入することが考えられる。

7. おわりに

Xウィンドウにおけるシェイプ拡張を利用した単純なメカニズムにより透明ウィンドウをインプリメントし、それを共有APに被せる構成で共有透明ボードシステムCOGENT SHAREを設計・試作した。これにより、きわめて簡潔な機構で協調的なグループワークにおける議論対象・焦点・注釈メッセージを統合的に分散共有することが可能となった。現在はプロトタイプをデモンストレーションや遠隔地との会議に利用し、さまざまな評価データを集めているところである。今後は性能面で更なる改良が必要である。

参考文献

- [1] Olson, G.M., McGuffin, L., Kuwana, E., Olson, J.S., Designing Software For A Group's Needs: A Functional Analysis of Synchronous Groupware, in User Interface Software (Bass & Dewan Ed.), John Wiley, 1993.
- [2] 桑名, 坂本, コーディネーションプロセスから見た協調作業支援機能モデル, 情報処理学会グループウェア研究会, Vol.93, No.95, pp.95-102, Oct. 1993.
- [3] Malone, T.W., Crowston, K., Toward an Interdisciplinary Theory of Coordination, Tech. paper CCS TR#120, MIT, 1991.
- [4] Jacobson, V., McCanne, S., Using the LBL Network 'Whiteboard', Lawrence Berkeley Laboratory, Feb. 1993.
- [5] Abdel-Wahab, H.M., Feit, M.A., XTV: A Framework for Sharing X Window Clients in Remote Synchronous Collaboration, IEEE TriComm'91, 1991.
- [6] Sun Solutions, Inc., The Complete Guide to ShowMe 2.0, Sun Technology Enterprises, Inc., Nov. 1993.
- [7] InSoft, Inc., Communique! Users Manual, InSoft, Inc., Jun. 1992.
- [8] Ahuja, S.R., Ensor, J.R., Lucco, S.E., A COMPARISONS OF APPLICATION SHARING MECHANISMS IN REAL-TIME DESKTOP CONFERRING SYSTEMS, in ACM conf. on Office Information Systems, pp.238-248., 1990.
- [9] Lauwers, J.C., Lantz, K.A., COLLABORATION AWARENESS IN SUPPORT OF COLLABORATION TRANSPARENCY: REQUIREMENTS FOR THE NEXT GENERATION OF SHARED WINDOW SYSTEMS, ACM Proceedings of CHI '90, pp.303-311, 1990.
- [10] 阿部, 前野, 阪田, 福岡, 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, 情報処理学会論文誌, Vol.34, No.6, pp.1406-1416, 1993.
- [11] 村永, 阿久津, 兎玉, 今井, 守安, Open GroupMedia - 個人用アプリケーションを共有するグループウェア・フレームワーク, 情報処理学会グループウェア研究会, Vol.93, No.95, pp.121-128, Oct. 1993.
- [12] Intel Corporation., Intel ProShare(TM) Personal Conferencing, www.intel.com, 1994.
- [13] 桑名他, 協調的な創造活動のためのコラボレーションルーム: COGENT, 情報処理学会グループウェア研究会, 63-2, 5-2, 1993.
- [14] Ferguson, P.M., 長方形でないウィンドウ形状への拡張, UNIX USER, Vol.2, No.1, JAN., 1993.