

## 協調作業と個人作業を融合する オブジェクト指向ネットワーク作業環境

越 智 剛<sup>†,☆</sup> 松 田 昇<sup>†</sup> 岡 本 敏 雄<sup>†</sup>

本稿では、グループウェア構築、実行の汎用のプラットフォームとなる分散システムの機構について述べる。本システムは、ネットワーク計算機環境における協調作業と個人作業の基盤となる統合作業環境の実現を目的として構築した。本来の人間の作業は、協調作業や個人作業といったある作業形態だけが独立して存在するのではなく、形態の違う複数の作業を同時に行ったり、作業形態を変えたりしながら進められる。個人の計算機作業環境から利用者が違和感なく協調作業を行えるように、個人作業と協調作業を融合するメカニズムが必要となる。本研究で構築したシステムでは、協調作業空間となるすべてのオブジェクトが同じ共有のメカニズムを持ち、システム上に構築されたツールはすべて動的に作業形態を変更できる。また、本システムで実現したユーザのオブジェクト化によって協調作業空間の形成を容易にするとともに、作業支援機構を協調作業として後から組み込むことが可能である。このシステムの有効性の検証のために、システム上で作業ツールと支援システムを構築した。

### An Object Oriented Distributed Working Environment to Integrate Cooperative Work and Personal Work

TAKESHI OCHI,<sup>†,☆</sup> NOBORU MATSUDA<sup>†</sup> and TOSHIO OKAMOTO<sup>†</sup>

In this paper, we describe the mechanism of a distributed computing system which provides us with a common base of groupware. The system has been built for the purpose of offering a working environment which integrates cooperative and personal works. Naturally, human work is made of various basis cooperative and personal works, and goes on while working parallel or changing form. An integration mechanism is necessary so that user could do cooperative work without any inconvenience on a personal computing environment. Our distributed computing system has an object sharing mechanism which is designed for groupware, and all tools which are constructed on the system can change their cooperation form dynamically. An encapsulation of user makes it easier to build a shared workspace, and makes it possible to build in work support system into the work environment. We built work tools and a work support system in order to verify effectiveness of the system.

#### 1. はじめに

近年、ネットワーク上での複数の人間のインタラクションの機会が飛躍的に増大し、協調作業を支援する分散型のグループウェアシステムの必要性が認められてきた。現在では、様々なグループウェアシステムが研究開発され、実用化されている<sup>1)</sup>。またその技術の適用範囲も、オフィスシステムから協調学習<sup>9)</sup>、ゲーム等大きく広がってきている。グループウェアのユーザインタフェースは協調作業向けに構築されており、

その利用によって効率良く協調作業を行えると考えられる。しかしながら本来の人間の作業は、ある協調作業だけが独立して存在するのではなく形態の違う複数の作業を同時に行ったり、作業形態を変えたりしながら進められる。そのような作業の中では、特定の協調作業だけでなく作業全体を統合した作業支援環境が必要となる。計算機上における作業全体の支援には、作業ツールすべての土台として機能するグループウェアの基盤システムが有効と考えられる。

個人作業と共有作業の融合の方法として、個人作業ツールと共有作業ツールの統合化が考えられる。その方法として、従来の個人作業に使われてきたシングルユーザアプリケーションの共有があげられた。リアルタイムグループウェアについての論文<sup>4)</sup>では、2つの共有方法を集中型アーキテクチャ、分散複製型アーキ

<sup>†</sup> 電気通信大学大学院情報システム学研究科  
Graduate School of Information Systems, The University of Electro-Communications

<sup>☆</sup> 現在、株式会社エフエム  
Presently with FM Inc.

テクチャとして定義している。しかしながら、いずれの方法にしても実装、あるいは実行時において問題点が存在し、根本的対処法とはいえない。

それ以外の個人作業ツールと共有作業ツールの統合化の方法として、共有作業、個人作業のどちらでも使用できるアプリケーションの開発が考えられる。リアルタイム分散会議システムである MERMAID<sup>6)</sup>では、共有作業用ドローツールを起動時の指定によってシングルユーザのツールとして利用できる。また、分散オブジェクトを使用した作業融合支援の研究では、分散オブジェクト指向の作業支援プラットフォーム、Fusion Worksを開発している<sup>5)</sup>。ここでは、プラットフォーム上で動作するツールを共有作業に使用するか個人作業で使用するかによってツールのイベントメッセージを制御し、ツールをどちらの方法でも利用可能にしている。このように、単一のツールを個人、共有どちらにおいても利用可能にする方法は、すでに有効性が確認されている。

一方、CSCL (Computer Supported Collaborative Learning) の分野では、協調作業や協調学習の支援の目的で様々な研究が行われている。たとえば、稲葉ら<sup>10)</sup>によって分散協調学習環境における知的議論支援の研究が行われている。この研究では、ネットワーク上に分散した学習者たちの幾何学の協調学習を支援する知的グループ学習支援システムを開発し、利用を行ってきた。この研究の主体である議論支援機構は、領域知識を用いないドメイン非依存の議論支援を実現していた。しかしながら実際にはグループ学習システムの一部として動作せざるをえないために、利用される状況が限定されているといった問題点が指摘されていた。協調作業支援の枠組みの研究は、グループウェア構築基盤として有用かつ普及しているシステムが存在しないために、独自のアーキテクチャで構築された閉鎖的なシステムとなりがちである。そのため協調作業支援の既存のグループウェアへの適用や、他の協調作業支援システムとの組合せといった要求を実現するのは困難である。これら協調作業支援システムを同一の汎用プラットフォーム上で動作させれば、これらの要求を解決できる。

これらを背景として本研究では、分散型の協調および個人作業の統合環境アーキテクチャの実現を目的とし、グループウェアを構築/実行するための汎用的な基盤となるオブジェクト指向分散システムの構築を行う。この基盤システム上で動作するすべてのツールが動的に作業形態を変更できる機能と、共有作業空間を構築するフレームワークを提供する。また、協調作業

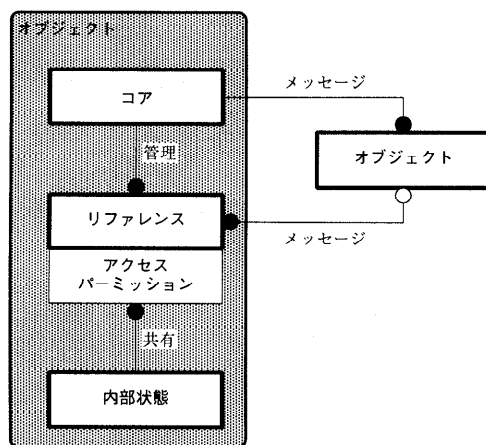


図1 オブジェクト共有のクラス図  
Fig.1 Class model of object sharing.

支援を目的とするソフトウェアの開発および利用をより容易にするプラットフォームとしての機構を持つ。これらによって、作業者の要求に合わせた作業形態をより柔軟に実現するとともに協調作業支援システムによる作業支援環境への拡張性を提供する。

## 2. グループウェア基盤機構

本研究では、上述したオブジェクト指向分散システム（以下、基盤システムと呼ぶ）を実現するために、オブジェクト共有とユーザ隠蔽の機構を提案する。本章では、それらの概念を説明する。

### 2.1 オブジェクト共有

図1は、オブジェクト共有機構のクラス図を示す。クラス図はオブジェクト図の一種で、オブジェクトおよびクラスとその間の関係をモデル化するための形式的図式記法である<sup>13)</sup>。システムを構成するオブジェクトクラスおよびクラス間の関連を、四角形と線分で図示する。四角形はクラスであり、四角形の中にはクラス名と、必要に応じて属性、メソッドを記述する。線分はクラス間の関係で、端点には多重度、すなわちそのクラスのいくつのインスタンスが関連するクラスの単一のインスタンスと関連するかを示す記号を記述する。○は多重度が0から1、●は多重度が0からn、記号のない端点は多重度1を表す。

我々の開発した基盤システムでは、オブジェクトは1つのコアおよび0個以上のリファレンスからなる。コアは非排他（複数のオブジェクトからの通信を受ける）、リファレンスは排他（1つのオブジェクトからの通信のみ受ける）の性質を持つ。コアは、リファレンスを管理しており、リファレンスを生成、破棄できる。またそのためのメソッドを他のオブジェクトに提供す

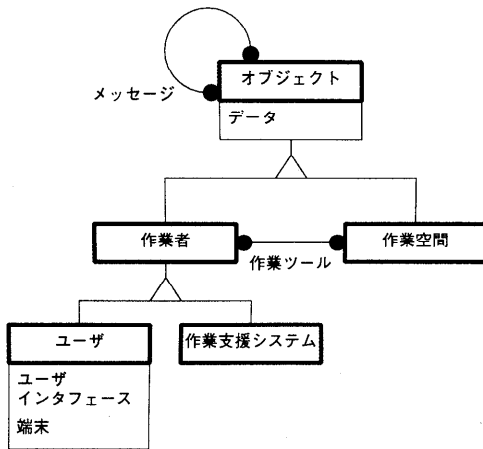


図2 本研究のグループウェアシステムのクラス図  
Fig.2 Class model of our groupware system.

る。オブジェクトは、オブジェクトのリファレンスを獲得するとそれに対して排他的通信が可能となる。この状態をリファレンスを所有していると呼ぶ。リファレンスは、共有データであるオブジェクトの内部状態にアクセスできる。またリファレンスは、共有データへのアクセス制限としてアクセスパーミッションと呼ぶ内部状態を持つ。

本基盤システムにおけるオブジェクトの共有は、この共有モデルで次のように表現される。オブジェクトのリファレンスが1つ存在する場合、そのオブジェクトは1つのオブジェクトによって作業されている個人作業状態である。リファレンスが複数存在する場合、そのオブジェクトは複数のオブジェクトに共有されている共有作業状態にある。リファレンスの生成や破壊によって、オブジェクトの共有状態は動的に変化していく。

## 2.2 ユーザ隠蔽

本基盤システムでは、作業者そのものをオブジェクトとしてカプセル化し、アクセス透過性を実現する。これをユーザ隠蔽と呼ぶ。これによって、協調作業空間の生成はオブジェクトをノード、通信をアークとするネットワークの構成問題となり、容易に共有作業環境を構築できる。ユーザ隠蔽を実現する機構を持つオブジェクトをユーザオブジェクトと呼ぶ。このオブジェクトは作業者1人に1つ割り当てられ、作業者の作業時以外でもつねにシステム上で動作する。

本研究におけるグループウェアシステムのクラス図を図2に示す。クラス図では、クラスの派生関係は、線分中の△で表す。作業者および作業空間は、分散オブジェクトから構築される。作業者にはユーザおよびユーザの協調作業を支援する支援システムがあり、こ

れらが作業空間を使用して協調作業を行う。オブジェクト共有により、個人作業空間と共有作業空間のクラスとしての違いがなくなり、作業空間として統一できる。これにより、個人作業と協調作業の間のデータ交換が不要となる。それとともに、従来システムに組み込まれていた協調作業支援が1つのオブジェクトとして独立し、再構成や再利用が容易に可能となる。

## 3. システムの構成

本基盤システムは、実行環境を実現するモジュールと、オブジェクトやユーザインタフェースを構築するための開発環境から構成される。本章では、本システムの各部の構成を説明する。

### 3.1 システムの概要

図3に、本基盤システムの構成を示す。本システムは、TCP/IP ネットワークに接続されたUNIX ワークステーション上で動作する。ネットワークモジュールは、他のワークステーションからきたメッセージを配送する役割を持つ。オブジェクトマネージャは、オブジェクトの起動とネームディレクトリ・サーバの役割を持つ。アプリケーションプログラマが構築したオブジェクトは、基盤システムの機能によりノード間の通信やメッセージ処理、オブジェクト共有等の基本的機能を持つ。ユーザオブジェクトは、作業者1人に1つ存在し作業者にユーザインタフェースを提供するとともに、他のオブジェクトに利用者の指示を伝える。利用者は、ユーザオブジェクトのユーザインタフェースを用いて作業空間を構築し、作業を行う。

### 3.2 オブジェクトの実装

本基盤システムのオブジェクトは、ノードと呼ぶ通信単位プロセスの集合である。すべてのノードは、ユニークなノードアドレスを持ちノード間の通信を行う。

図4に、オブジェクトの内部構造を示す。ノードは、本基盤システムのライブラリによるシステムプログラムとアプリケーションプログラマの記述したユーザプログラムからなる実行モジュールである。ユーザプログラムは、システムプログラムによるメソッドの呼び出しにより実行され、基盤システムが提供するC++関数の呼び出しによってメッセージの送信等のサービスを利用できる。ノードは、オブジェクト共有を実現するため特殊な役割を持つものがある。以下にノードの分類を示す。

コアノード オブジェクト共有のコアの役割を持つノードで、オブジェクト内部に1つ存在する。リファレンスノードを管理し、リファレンスノードの起動、終了によってオブジェクトのリファレンス

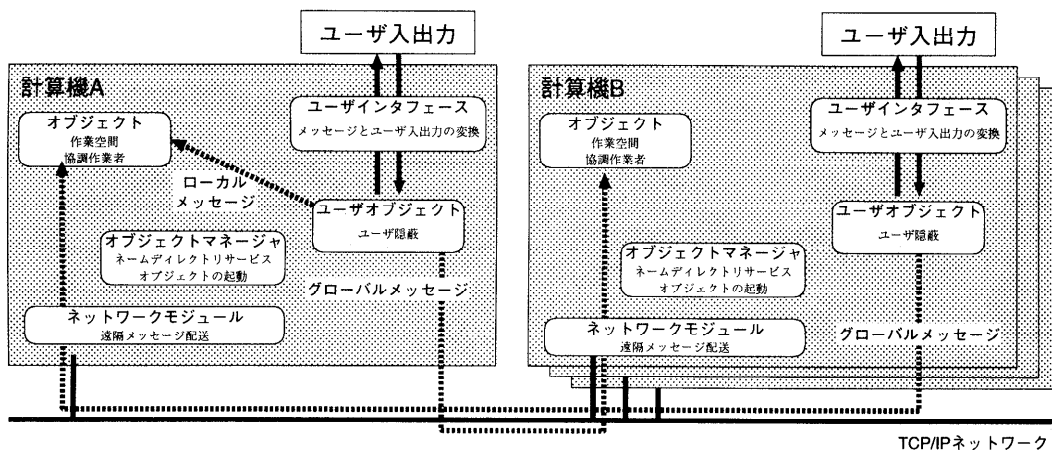


図3 本システムの構成  
Fig.3 Structure of the system.

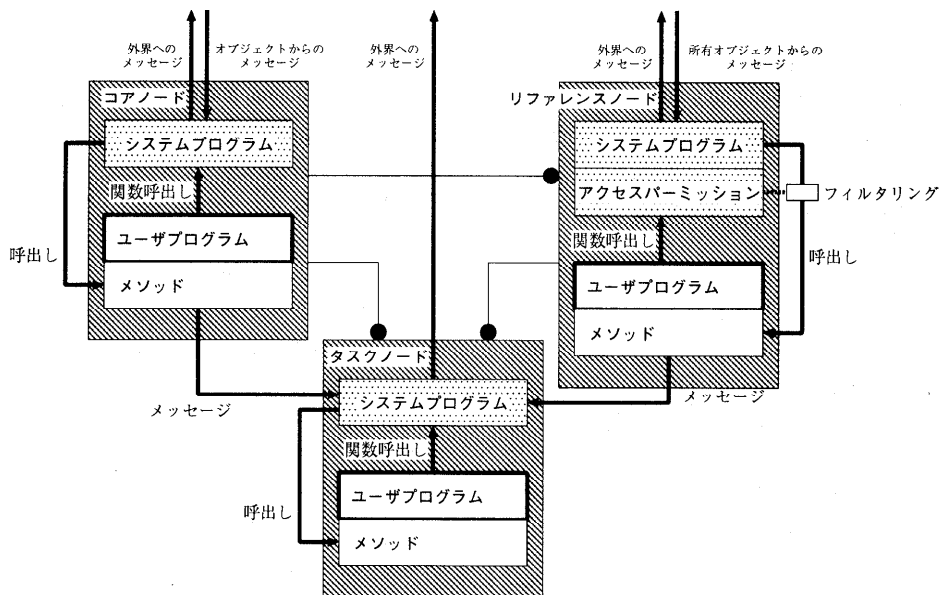


図4 オブジェクトの内部構造  
Fig.4 Inner structure of object.

を制御する。リファレンスノードで共有するデータもここに置かれる。

**リファレンスノード** コアノードによって管理されるノードで、0個から複数個存在する。オブジェクト共有におけるリファレンスの役割を持つ。共有データと作業オブジェクトとの中継を行うプログラムが記述される。また競合解消の方法の1つとしてアクセスパーミッションと呼ぶ内部値が存在し、共有データへアクセスする前に参照される。この内部値はコアノードによって設定される。

**タスクノード** コアノード、リファレンスノードの子として起動するコアノード、リファレンスノード

以外のノード。各コアノード、リファレンスノードにつき0個から複数個存在する。オブジェクト共有における役割はなく、アプリケーションプログラムが必要に応じて使用できる。

### 3.3 ユーザオブジェクト

本基盤システムでは、システムで作業を行うユーザ1人1人にユーザオブジェクトが割り当てられる。このオブジェクトは、ユーザが作業中であるか否かにかかわらずつねにシステム上で動作し、他のオブジェクトからのメッセージを処理している。ユーザオブジェクトは、ユーザに対して所有オブジェクトのユーザインタフェースを提供しなければならない。そのため

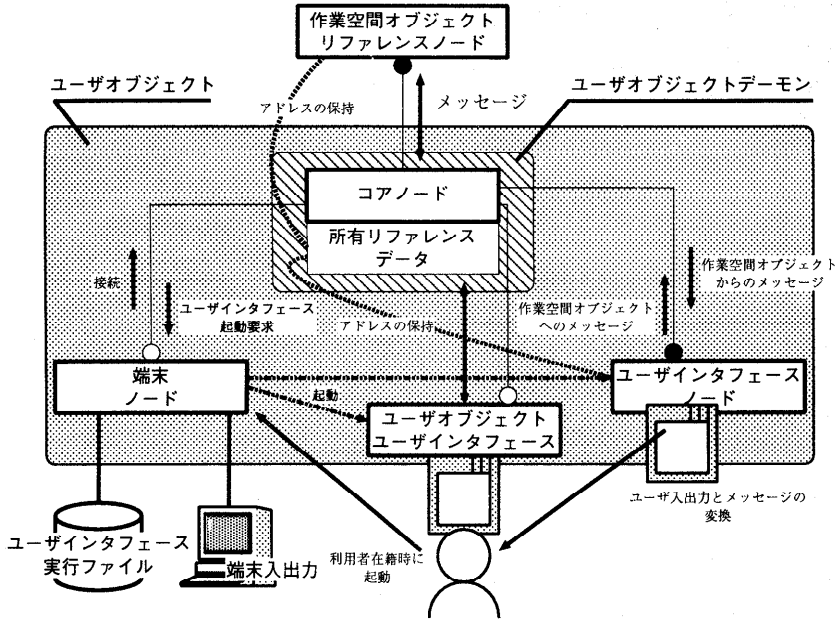


図5 ユーザオブジェクトの機構  
Fig. 5 A mechanism of user object.

に、ユーザオブジェクトは所有オブジェクトに対応するユーザインタフェースノードを実行する機能を持つ。ユーザインタフェースノードは、アプリケーションオブジェクトとともにアプリケーションプログラマによって構築されるタスクノードである。

図5に、ユーザオブジェクトの構成を示す。

ユーザオブジェクトのコアノードは、所有している作業空間のリストを持っており、それに対応するユーザインタフェースノードの情報等とともに管理している。端末ノードはタスクノードの一種で、ユーザがシステムにログインしたときに起動し、コアノードに接続する。コアノードは、端末ノードが接続するとユーザがログインしたと判断し、ユーザオブジェクトに命令を与えるためのユーザインタフェースを提供する。また、作業空間情報を参照して必要なユーザインタフェースノードを起動する。コアノードから起動されたユーザインタフェースノードは、対応する作業空間オブジェクトからデータを得て、ユーザに提示する。またユーザの入力をメッセージに変換し、作業空間オブジェクトに転送する。

図6は、ユーザオブジェクトのユーザインタフェースを示す。このユーザインタフェースは、ユーザオブジェクトに送信されたメッセージを見るためのウィンドウ、およびメッセージを編集し送信できるウィンドウからなる。ユーザはここでオブジェクトの生成や所有といった、作業空間構築に関するオペレーションを

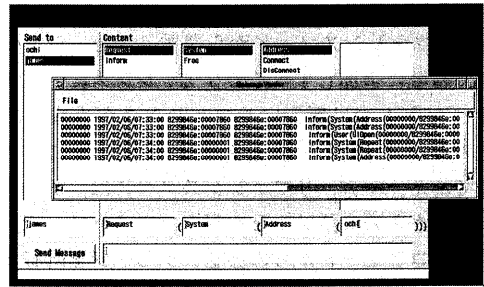


図6 ユーザオブジェクトのユーザインタフェース  
Fig. 6 User interface of user object.

実行する。

#### 4. システムの利用と評価

本基盤システムの有効性を確認するために、実際に基盤システムを用いて協調作業を行うためのアプリケーションを構築した。本章ではこれらについて説明する。

##### 4.1 協調作業環境の構築

実験用のアプリケーションとして、オセロおよび会話の2つの作業空間を実現するアプリケーション、および議論支援を行う作業支援アプリケーションを構築した。

作業者は、オセロと会話アプリケーションを共有し作業を行う。その際、さらに共有対象として議論支援システムを選択し、利用することができる。

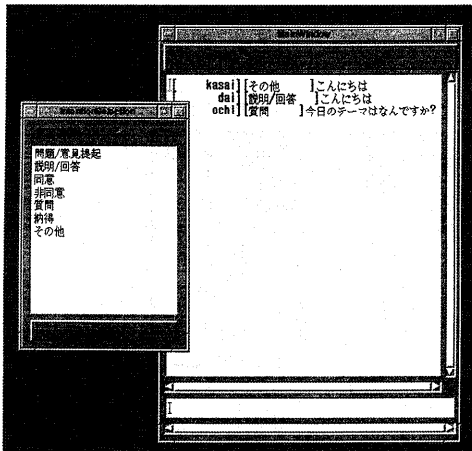


図7 会話オブジェクトのユーザインタフェース  
Fig. 7 User interface of talk object.

**会話アプリケーション** 協調作業参加者が会話をを行う作業空間を実現する作業空間オブジェクトとユーザインタフェースノードを構築した。このオブジェクトのコアノードは、共有データとして発言そのものとその意図を表す文字列からなる発言メッセージの履歴を保存している。また2つのメソッドが実装されている。1つはリファレンスノードから発言メッセージを受け取ると履歴に追加しすべてのリファレンスに発信する機能、もう1つは新しいリファレンスが生成されると他の参加者へ新しい参加者があったことを知らせる機能である。一方、リファレンスノードは、所有オブジェクトからの会話メッセージをコアノードへ転送する機能、コアノードからの会話メッセージを所有オブジェクトへ転送する機能が実装されている。図7は、会話アプリケーションのユーザインタフェースを示す。ユーザインタフェースノードは、リファレンスノードからの発言メッセージを画面に表示し、ユーザの入力（テキスト入力と発言意図の選択）を発言メッセージに変換してリファレンスノードに発信する機能を持つ。

**議論支援アプリケーション** 議論支援アプリケーションは、会話アプリケーションを利用者と共有して議論支援を行う作業支援オブジェクトである。これは、幾何学の協調学習を支援する協調学習支援システム(iDCLE)<sup>10)</sup>で実装された機能である。領域知識を使わず発言の意図を基に議論状態を把握し、議論への介入発言によって円滑な議論の進行を促す。iDCLEにおいては、この機能はサーバの一部であったが、今回の構築では利用範囲をより広げるために、作業支援オブジェクトとして独立して構築した。本アプリケーションは、会話アプリケーションのオブジェクトを他の作業支援オブジェクトと同じように共有し、ユーザの発言

メッセージを受信する。受信した発言メッセージの意図文字列を取り出し、議論を分析するとともに、問題が発生していると判断すると介入発言を共有オブジェクトに送信して議論支援を行う。

ユーザ隠蔽は、協調作業を支援するモジュールを、ユーザオブジェクトと同じように作業となるオブジェクトとして構築できる利点を持つ。すなわち、オブジェクトから見てどのオブジェクトが実際に人間であるかは関係がなく、ユーザとのインタラクションを持たないオブジェクトでも容易に共有作業に参加できる。これによって、協調作業支援モジュールの動作形態がシステムに束縛されず、また他のオブジェクトの変更なしに支援モジュールを構築できるようになった。

**オセロアプリケーション** 作業空間オブジェクトとしてオセロゲームを行うためのオブジェクトとユーザインタフェースを構築した。練習や戦略を練るときの個人作業にも利用できるよう、アンドゥ機能が設けられている。このアプリケーションオブジェクトのコアノードは、オセロの盤面情報を共有データとして持ち、下に示すメソッドによってオセロオブジェクトを実現した。

- 盤面情報を送信する機能
- 配置する駒の位置（またはパス）情報を受け取ると盤面情報を更新し、結果を全リファレンスノードに送信する機能。
- 盤面初期化命令を受け取ると盤面情報を初期化し、結果を全リファレンスノードに送信する機能。
- アンドゥ命令を受け取ると最後の盤面更新を取りやめ、結果を全リファレンスノードに送信する機能。
- リファレンスノードが生成されるとそれに対しプレイヤー権限を割り当て、全リファレンスノードに伝える機能。生成される順に白、黒、観戦者（参照のみの権限）のプレイヤー権限を割り当てる。
- リファレンスが破棄されるとそのリファレンスの権限をチェックし必要に応じてプレイヤー権限を他に割り当て、全リファレンスノードに伝える機能。

またリファレンスノードは、基本的に会話アプリケーションのものと同様の機能を持つが、自身のプレイヤー権限に応じたフィルタリングを行う。たとえば、白プレイヤーのリファレンスノードは、所有オブジェクトから駒配置のメッセージが送信されても白プレイヤーの手順でなければコアノードにメッセージを転送しない。図8は、オセロアプリケーションのユーザインタフェースを示す。盤面上をマウスでクリックすると、駒の配置情報がリファレンスノードに送信される。ま

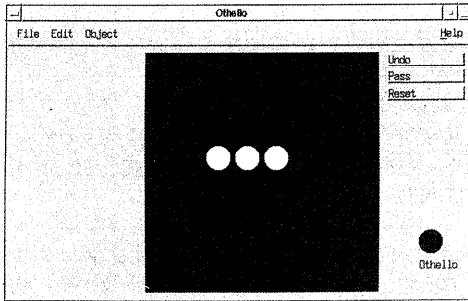


図8 オセロオブジェクトのユーザインタフェース  
Fig.8 User interface of othello object.

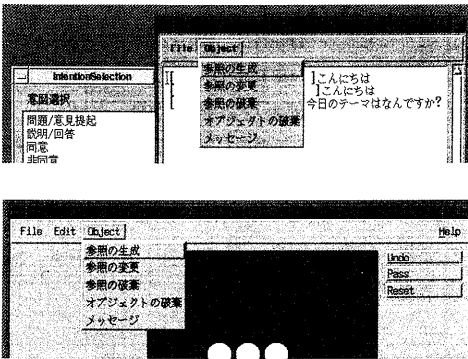


図9 ライブラリによるメニューの実例  
Fig.9 Menu which is included by system library.

たアンドゥ、パス、盤面の初期化等のボタンを押すと、それぞれに応じたメッセージがリファレンスノードに送信される。

**共有ユーザインタフェース** 共有状態と固有状態が動的に変化する統合グループウェアにおいては、作業者がいつでも作業空間の共有状態を変更、確認できる機能が必要である。本基盤システムでは、対象オブジェクトに対してリファレンスを制御するための決まったメッセージを送信すれば任意のアプリケーションでいつでも共有状態を変更できる。今回のアプリケーション構築の実験の中で、この機能を実行するユーザインタフェースを追加する共通の関数が構築された。

図9は、これを用いたオセロ、および会話のメニューを示す。作業空間の共有に関するコマンドはすべてメニュー“Object”に集められており、ツールにかかわらず、つねに同じ方法で共有状態を変更できる。これによってアプリケーション共有にかかわるプログラミングに必要なコードは少量ですみ、プログラマはアプリケーション固有の問題に集中できた。

4.2 考 察

本基盤システム上でのノードプログラミングは、C++上での一般的イベント駆動プログラミングであ

る。このためC++について学んでいるプログラマは抵抗なくプログラミングが可能である。一方、コアノードとリファレンスノード、ユーザインタフェースノードの機能の設計に関しては、本基盤システムの基本概念の理解が必要であり、各ノード間のインタフェースを設計するためにある程度のオブジェクト指向プログラミングの能力が必要とされる。構築技術を持つプログラマにとっては、起動時の他のノードとの接続やメッセージ受信、共有にともなうノード構成の変更等、共有作業空間として必要な基本的処理はすべて基盤システムのプログラムが行うので、固有のメソッドの記述のみでノードを実装可能である。この点でグループウェアアプリケーションの作成成率は向上したといえる。

また、アプリケーションプログラムとそのユーザインタフェースとの分離により、ユーザと協同作業を行うアプリケーションの構築が可能となった。今回の実験では議論支援アプリケーションが構築された。このアプリケーションが持つ議論支援機能は、会話内容のドメインに依存せず、会話参加者の変動にも対応している。したがって、作業者は作業の内容、形態にかかわらず支援システムを利用することができる。その中で、発言の意図入力が必要とする現在のユーザインタフェースの使いにくさが再確認される等、支援システムへのフィードバック効果も現れている。一方、現在のオセロオブジェクトを使用してオセロプレイヤーオブジェクトの作成が考えられている。これによってユーザのプログラムとの対戦や、プログラムどうしの対戦の観戦等が可能となる。また、そのアルゴリズムに不満を持ったプログラマが新しいオセロプレイヤーオブジェクトを作成してプログラムどうしの対戦を行うといった利用が考えられる。

構築されたアプリケーションの利用実験では、作業環境の利用時にオセロのプレイヤーが個人作業で使用していたオセロオブジェクトを観戦者と共有し、戦法について話し合うといった作業が見られた。これは本基盤システムの共有空間において、共有-個人作業間を動的に形態変化させることが可能であり、その方法が簡単であることを示していると考えられる。

ユーザオブジェクトは、永続的に存在する作業代理オブジェクトである。したがって、作業者不在時に他者から作業空間が提供された場合、ユーザオブジェクトはその作業空間を作業者がログインするまで保持する。これによって本システムは、非同期型のグループウェアも兼ねられるため、個人作業、同期型共有作業、非同期型共有作業を同時に扱える。よってこのシ

ステムは、広い範囲の作業形態をカバーできるといえる。

## 5. おわりに

分散型の協調および個人作業を支援する統合環境アーキテクチャの実現を目的とし、コンピュータネットワークにおけるグループウェアの開発運用基盤となる分散システムのアーキテクチャを設計、構築した。より具体的には、オブジェクト共有によって共有作業空間と個人作業空間を融合し、ユーザ隠蔽によって作業者のネットワーク資源化と、協調作業支援システム構築運用を行いやすい環境の構築を行った。

また、これまで幾何学の協調作業支援の中だけで動作していた議論支援モジュールの独立した構築によって、議論支援の利用範囲を広げられた。今後は、他の協調作業支援モジュールの導入により、複数の作業支援モジュールによる支援環境が期待される。

## 参考文献

- 1) 石井 裕：CSCW とグループウェア，オーム社 (1994).
- 2) 坂下善彦，井手口哲夫，滝沢 誠，水野忠則，分散システム入門，近代科学社 (1993).
- 3) Crowe, M.K.: *Cooperative Work with Multimedia*, Springer-Verlag (1994).
- 4) 石井 裕：リアルタイムグループウェアのデザイン，情報処理，Vol.34, No.8, pp.1017-1027 (1993).
- 5) 木本陽介，服部進実：分散オブジェクト指向による作業融合支援プラットフォーム Fusion Works の開発，情報処理，Vol.37, No.12, pp.2352-2361 (1996).
- 6) 渡辺ほか：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) 阿部豊子，前野和俊，阪田史郎，福岡秀幸：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.1406-1415 (1993).
- 8) 村永哲郎：グループワークのための情報共有技術，情報処理，Vol.34, No.8, pp.1006-1016 (1993).
- 9) O'Malley, C.: *Computer Supported Collaborative Learning*, NATO ASI Series, Series F: *Computer and Systems Sciences*, Vol.128 (1995).
- 10) 稲葉晶子，柳場泰孝，岡本敏雄：分散協調型作業/学習環境における知的議論支援，電子情報通信学会論文誌，Vol.J79-A, No.2 (1996).
- 11) 小島圭一，岡本敏雄：CSCW の対話における発言意図の推定に関する研究 (3)，信学技報，Vol.AI95-53, pp.87-94 (1996).
- 12) 越智 剛，松田 昇，岡本敏雄：マルチユーザ指向マルチエージェント環境における知的グループウェアの構築 (2)，信学技報，AI95-62, pp.33-38 (1996).
- 13) ランボー，J.，ブラハ，M.，プレメラニ，W.，エディ，F.，ローレンセン，W.: オブジェクト指向方法論 OMT モデル化と設計，トッパン (1992).

(平成9年4月7日受付)

(平成9年11月5日採録)



越智 剛

1971年生。1995年，日本大学文理学部応用数学科卒業。1997年，電気通信大学大学院情報システム学研究科修了 (情報システム設計学専攻)。修士 (工学)。グループウェアシステムの研究・開発に従事。現在 (株) エフエム，Social 開発事業部勤務。



松田 昇 (正会員)

1962年生。1988年，東京学芸大学大学院修了 (数学教育学専攻)。教育学修士。金沢工業大学工学部助手を経て，1993年，電気通信大学大学院情報システム学研究科助手，現在に至る。学習者モデルの構成手法を中心に，知的教育システムの研究に従事。人工知能学会，電子情報通信学会，日本教育工学会，教育システム情報学会各会員。



岡本 敏雄 (正会員)

1975年，東京学芸大学大学院修了。工学博士 (東工大)。金沢工業大学工学部，東京学芸大学講師，助教授，教授を経て，1992年より，電気通信大学大学院情報システム学研究科知識処理システム学講座教授。知的 CAI システム，協調分散グループ作業・学習支援システム，情報教育カリキュラム開発などの研究に従事。(訳書)「人工知能と知的 CAI システム」，「知的 CAI システム」，(著書)「教育における情報科学」など。電子情報通信学会 (人工知能と知識処理専門委員会副委員長)，人工知能学会，日本教育工学会 (理事)，教育システム情報学会 (副会長) 各会員。