

統合コミュニケーションシステムを用いた グループプログラミング支援の試み

凍田和美, 宇津宮孝一, 吉田和幸

大分大学

分散形計算環境の充実により、数人がデスクの近くにあるワークステーションを使って、協同して1つの仕事を行ういわゆるグループウェアが注目されるようになった。我々は、身近にある情報処理教育のグループプログラミングをグループウェアの題材に選び、開発中の統合コミュニケーションシステムの実時間会話機能を適用することで、分散環境下における協調作業の支援を試みている。

現在、協調作業の基本的な支援ツールとして、他のメンバのプログラムを共有する機能、グループ全員で1つの情報を作る機能、実行時の状況を共有して見る機能を考察している。また、設計中のツールの検討を目的として計算機を介したグループプログラミングの予備実験を実施している。

本稿では、グループプログラミングの概要、支援ツールの考察、統合コミュニケーションシステムの援用法およびグループプログラミング実験についての我々の経験を述べる。

A Group Programming Environment using an Integrated Communication System

Kazuyoshi KORIDA, Kouichi UTSUMIYA and Kazuyuki YOSHIDA

Faculty of Engineering, Oita University

Groupware, software designed for supporting groups of cooperating users, raises a number of new issues along with the wide spread of distributed computing environments. Taking group programming in computer software education as a cooperative work in order to attack these issues, we are trying to support it using our Integrated Communication System under development.

This paper describes the group programming environment, this experiment and our experiences.

1. はじめに

分散形計算環境の普及に伴い、数人がデスクの近くにあるワークステーションを使って、協同して1つの仕事を行ういわゆるグループウェア^[1]が注目されるようになった。近い将来、広域分散環境においてもこうした協同作業が盛んになることが予想される。大分大学組織工学科においても、約60台のエンジニアリングワークステーションをイーサネットで相互に接続した分散形計算環境下で情報処理教育を行っている。分散形計算環境の充実は、学生のプログラミング能力を当初の予想以上に向上させた。その反面、各個人の能力の格差を生じやすい。したがって、プログラミングの演習においては、従来の等難易度の小さな問題を個々の学生に与える方法ではなく、比較的大きな問題を個人の能力に応じて分担させ、各人の能力をさらに高めながら、個人間の協同作業を通じてプログラミングのノウハウを伝え合うことにより、全体のレベルを向上させる必要がある。

我々は、1つの事例として身近にある情報処理教育のグループプログラミングをグループウェアの題材に選び、開発中の統合コミュニケーションシステム^[2]の実時間会話機能をグループプログラミングに適用することで、分散環境下における協同作業の支援を試みている。

本稿では、グループプログラミングの概要、支援ツール

の考察、統合コミュニケーションシステムの採用法およびグループプログラミング実験についての我々の経験を述べる。

2. グループプログラミングの概要

一般にプログラミングには、①期限内の生産性を高めるために、職業的プログラマのチームにより巨大プログラムを作成させようとするものと、②個々のプログラマが各人の必要に応じてプログラム作るものがある。従来、ソフトウェア工学は、①に対するアプローチとして発展した。しかしながら、専門情報処理教育の現場で行われるプログラミングは②の形態を採るものが多く、特別な能力をもつ個人が、高機能・高品質のプログラムを作り出すことも少なくなかった。しかし、優れた個人プログラマは自分が作成したプログラムやその手法の説明は不得手なことが多く、せっかくのノウハウが他へ伝搬しにくい。我々は、こうした優れたプログラマが、容易にその手法を他のプログラマに伝えられる環境を構築し、教育の立場から他のプログラマのレベルを向上したいと考えた。

我々が試作するグループプログラミングの環境を図1に示す。10人程度の学生が、共有画面を通して、協同して1つのプログラムを開発する。学生は、問題のモジュール分割、分担、そのプログラミング、グローバルな変数の定義、部分的なテストと全体での実行などの各段階で実時間

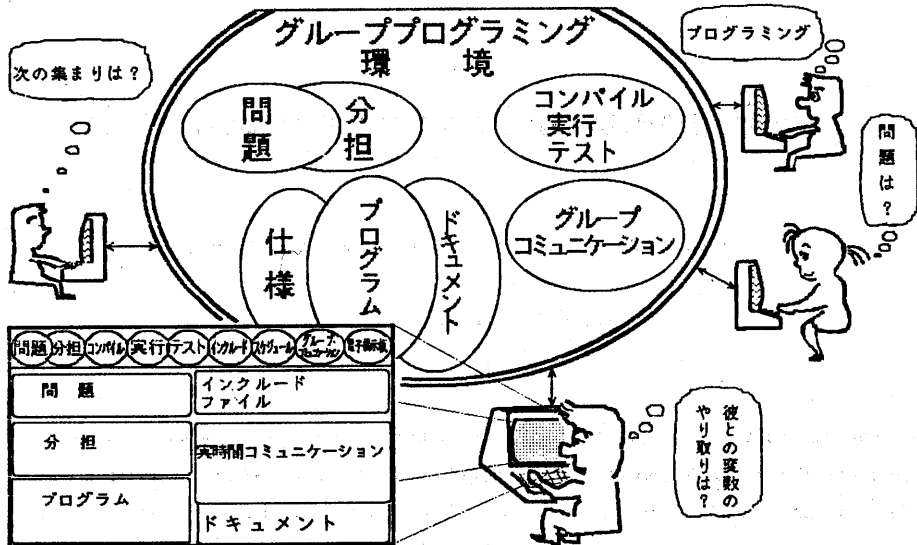


図1 グループプログラミング環境

会話機能を用い、お互いに情報をやり取りしながら、分担したプログラムモジュールを作成していく。

このような協同作業を支援するためには、①グループプログラミング支援ツールの開発および②グループ作業を支援するコミュニケーションツールの開発が必要となる。分散環境下で、グループプログラミングを実際に体験することを通じて、①に対しては、協調作業の基本となるツールの考察と開発を、②については、統合コミュニケーションシステムの実時間会話機能の充実を図る。

3. 支援ツールの考察

支援ツールを主に以下の3つの立場から考察する。

- (1) 問題やプログラムおよびそのドキュメントなどメンバー個人の資源をグループ内で共有する機能。これらの情報は互いにリンクされた図2のようなハイパertextの概念^[3]を導入した構成になると考えている。利用者は、グループ共有の資源をあたかも各自のカレントディレクトリにあるかのように取り扱うために、リンクを辿って必要なファイルを読み込む、また新しいプログラムを作成したときはそのファイルを適当なノードに保存することができる。
- (2) 共有変数や同一ファイル外から参照する副プログラム名など、グループ全員が利用する情報を取り扱うインクルードファイルに関する支援ツール。このファイルには、同時に書き込みができないように、ファイルの施錠、開錠の手続きを必要とする。
- (3) コンパイル、テストおよび実行時におけるその状況や出力情報をグループで共有するための支援ツール。

図3にグループプログラミング支援環境を示す。グループプログラミング支援システムは、(1)、(2)の支援を行うサブシステムと(3)の支援やグループ内のコミュニケーションを支援するサブシステムからなる。また、付加的な支援を行うサブシステムとして、グループメンバーのスケジュール管理とグループ行動を広報する電子掲示板が必要だと思われる。電子掲示板は、玄関先にあるディスプレイにより計算機を使用していないときにも情報を広報できるものである。

4. 統合コミュニケーションシステム

4.1 非同期会話

我々は、計算機を介したコミュニケーションを円滑に行うため、さまざまなコミュニケーションの機能を1つの統一した概念に基づく機能として扱う統合コミュニケーションシステムを構築している。このシステムは、従来のコミュニケーションの概念に対応して分類されたボックスをもつ。これらのボックスはデータと機能が一体化されたもので、階層構造をもち、下位のボックスは親の機能を継承する。利用者は利用を統合したユーザインタフェースを使い、所定のボックスに対し必要な情報を送る。ボックスの属性 (mail, readnewsコマンド) を定義することで、従来の電子メール、電子ニュースの設備を簡単に用意できる。ボックスに新たな機能を付加することにより、電子レポート等のボックスを系統的に追加できる。利用者は、「所定のボックスに必要な情報を送る」だけで、さまざまなコミュニケーションを同一の考えで実行できる。

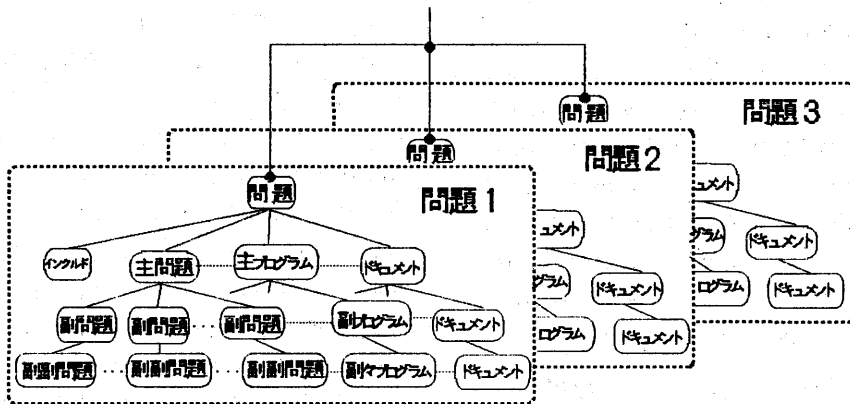


図2 文書ファイルの構造

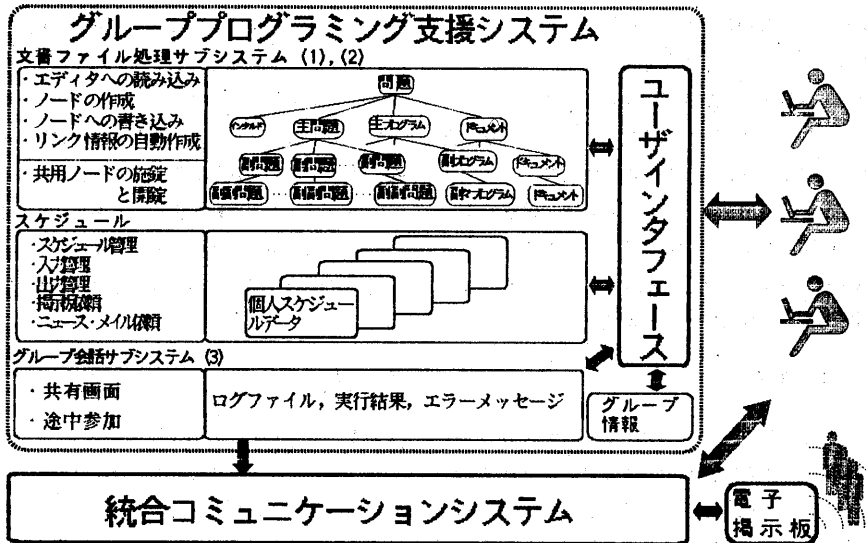


図3 グループプログラミング支援環境

4.2 実時間会話機能

近年、身近の常時使える状態の計算機がイーサネットなどの高速通信回線で結ばれるようになり、実時間（同期）会話が可能な計算環境となってきた。実時間会話を非同期方式と連続した機能として提供することにより、計算機を使用中のあるいは使える状況の人とは実時間で迅速な情報伝達とその応答が、他の人とは時間や場所にとらわれない情報伝達が可能になる。実時間会話機能はX-windowのXlibを用いて実現した。実時間会話の利用画面を図4に示す。利用手順は、次のとおりである。(1)利用者は、図式化した実時間会話アイコンをマウスで指定する。(2)ウィンドウ①が現れ、現在のWS利用者が表示される。利用者名の中から会話の相手を選ぶ(図4ウィンドウ②参照)。(3)ウィンドウ③、④が現れ、実時間会話が開始される。共有ウィンドウ③には参加者の会話が表示される。各自の発言はウィンドウ④で作成し、改行により文章が転送される。さらに、会話に途中から参加する機能(CALL)、既存のファイルを共有画面で見る機能(FILE)などを準備し、グループの協調作業を支援する。(4)ウィンドウ⑥は、プログラムソースファイルの共有画面である。マウスを使って手書きで印を付けることができる。

5. グループプログラミングのための予備実験

設計中の支援ツールの検討、問題点の洗い出しを目的に

計算機を介したグループによるプログラミング実験を実施している。まだプログラミングに慣れていない学部1年次生を対象にすることで、グループプログラミングの潜在的な問題点をも把握しようとした。

5.1 実験方法

学部1年次開講の「プログラミング演習」受講生を対象に8名のメンバからなる7つのグループを作り、1グループに1名の出題者(ある程度の情報教育を受けた者)を当てた。問題は、出題者の身近にあり、テキストファイルを扱うものとし、受講生が一人で担当しても作成可能な程度の問題とした。各出題者が作成した問題を、難易度が片寄らないように調整したのち、電子ニュースに掲載した。使用言語はPascal言語とし、課題の目標を「分かりやすいプログラム(モジュール化)」とした。期間は約1か月とし、できるだけ計算機を使った話し合いを行いながら問題を解くことにした。計算環境は、Unixワークステーション分散環境下で、マルチウィンドウ使用可能状況とした。グループプログラミングのための準備としては、各グループごとにグループのディレクトリを与えた。前述の実時間会話のログファイルは、このディレクトリ下に保存することにした。実時間会話、電子ニュース、電子メールの使用を奨励し、やり取りした情報をできるだけ保存するようにした。他に週一度の割合で対面の会議を行った。

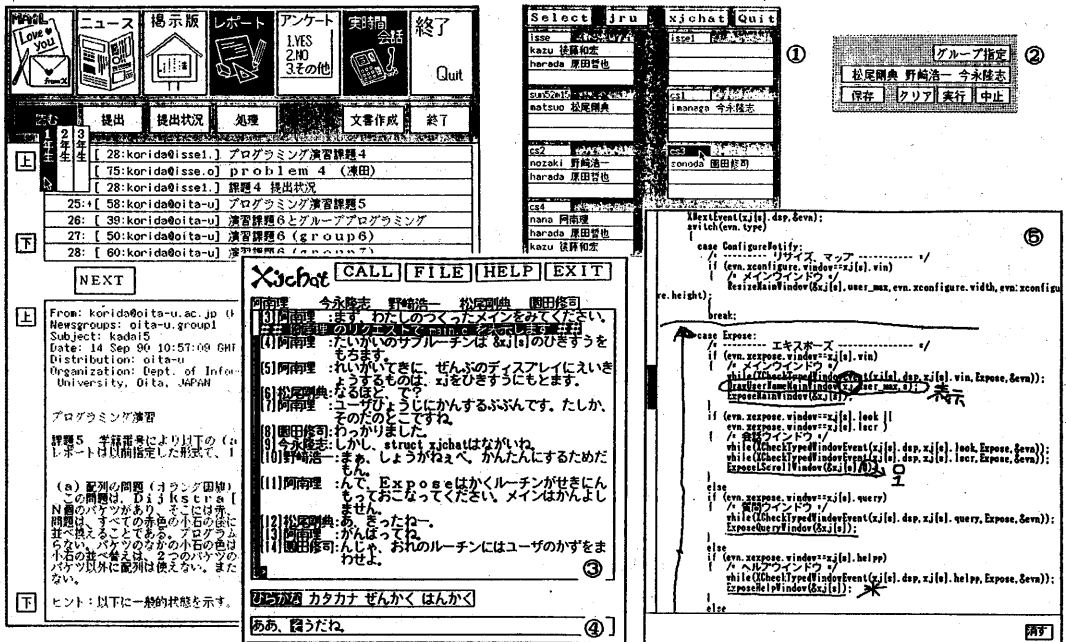


図4 実時間会話利用画面

5.2 実験の経験と検討

プログラミング準備段階に予想以上の時間を要した。本稿をまとめている時点で6週目であるが、ほとんどのグループがまだ最終段階の全体テストを行っている状況である。以下、途中経過であるが、計算機を介してやり取りした情報の解析結果を示す。

(1) 実時間会話

実験は各人の空き時間を使って行ったが、ワークステーション (WS) 利用中のグループメンバとはできるだけ実時間会話で相談させた。実時間会話利用回数は19回で、1回の平均参加者は約6名、平均継続時間は約2時間であった。全実時間会話で取り扱われた45項目のうち主な項目を以下に示す。

- ・グローバル変数について ----- 11
- ・日程について ----- 11
- ・分割と分担について ----- 9
- ・問題について ----- 5
- ・進行状況について ----- 2

(2) 電子ニュースによる伝達

全体やグループ内の情報は電子ニュースで通知した。総記事数は151であり、投稿者は28名であった。

●全員への通知 (全25記事)

- ・実時間会話について ----- 7
- ・問題について ----- 3
- ・分割コンパイル ----- 3
- ・標準入力でのファイル名指定 ----- 2
- ・呼び出し ----- 2

●グループ内通知 (全126記事)

- ・日程について ----- 61
- ・分割と分担 ----- 31
- ・問題について ----- 24
- ・グローバル変数 ----- 14

(3) 中間レポート

3週間後と5週間後にレポートを提出させた。それぞれ、40、47名が提出した。感想等に書かれていた問題事項で多かった項目を以下に挙げる。

●3週間後 (全85項目)

- ・デバッグに関すること ----- 19
- ・日程について ----- 12
- ・グローバル変数に関すること ----- 6
- ・実時間会話について ----- 5
- ・グループプログラミング ----- 4

●5週間後 (全108項目)

- ・テストや実行 ----- 24
- ・問題の難しさ ----- 20
- ・グループプログラミング ----- 15
- ・日程について ----- 12
- ・グローバル変数 ----- 4

(4) アンケート

5週目に、グループプログラミングに関するアンケートを行った。56人中27人が回収された。

●時間がかった順

・設計, 作成, デバッグ	-----	7
・作成, 設計, デバッグ	-----	3

●難しい順

・作成, 設計, デバッグ	-----	6
・作成, デバッグ, 設計	-----	4

●グローバル変数の連絡

・口頭	-----	10
・実時間会話	-----	9

●ミーティングの日程を知った手段

・ニュース	-----	10
・口頭	-----	10
・メール	-----	4

テスト、実行などのデバッグに関する検討課題が出ている。個人モジュールテストのための何らかのデバッグが必要になると考えられる。

メンバの空き時間などの情報伝達、グループ活動の即座な情報伝達が日程に関する検討課題として出てきた。これらは、メンバのスケジュール管理機能や計算機を使わなくても、緊急の伝達事項が伝えられる電子掲示板の実現により解決できると考えられる。

グループプログラミングに関して、以下の意見があった。

- ・みんなでやると心強い
- ・とても詳しい人がいて驚いた
- ・リードしてくれる人がいて助かった
- ・他人に頼ることが出来ずに自力でやった
- ・オモシロイ、良い勉強になった
- ・みんなに迷惑がかかるので心配になる
- ・自分のペースでやっていけない
- ・強力なリーグがないと進まない
- ・まとめ役の苦労が大変大きい
- ・一人で作る方が楽

アンケート結果には、実際の情報伝達に口頭によるものが当初の予想よりも多く現れた。このことは、我々の現在の分散計算環境は、学生自身にとっては、実際に計算機のある場所に行かなければ利用できない環境であり、その場は顔が合わせられる状況であったなど実験環境が将来予想される真の分散環境とはかけ離れていたことによると思われる。今後は、実験環境の見直し、問題の分割過程、グループのリーダー的存在の役割および完成したプログラムの評

価などのさらに詳細な分析が必要である。

6. 検討

支援ツールについては、基本的な手続きを洗い出し、いかに協調作業に取り込むかを模索している。現時点での検討課題として以下の事項が挙げられる。

- (1) 実験結果に基づく支援ツールの検討
- (2) プリミティブな支援手続きの実現
- (3) 個人モジュールのデバッグ機能の実現
- (4) 実時間会話機能への漢字入力機能の実現
- (5) 実行時共有ウィンドウの実現
- (6) 図などのイメージ情報を送る機能の実現

実際の演習科目にグループプログラミングを取り入れる場合には、上記の支援ツールの早期実現のほかに、チュータの制度化やその指導体制の問題が残っている。また、グループプログラミングを優れたプログラマにとっても有効になるものにするための検討も必要となろう。これらの問題は、一般の協同作業の場合にも言えることである。

7. おわりに

グループプログラミングを1つの事例として情報処理教育に適用するために、予備実験を行い必要な支援ツールの考察を行った。これらの経験に基づいて、基本的なツールを早急に実現し、実際のグループプログラミングに適用したい。その後、より一般的な観点から、人が協調してグループプログラミングを行うメカニズムの解明^[4]と広域分散環境における協同作業のための仕組みの確立^[5]を図る必要がある。

参考文献

- [1] 石井: グループウェア技術の研究動向, 情報処理学会誌, 30, 12, pp. 1502-1507 (1989).
- [2] K. Korida, K. Utsumiya, K. Yoshida: An Integrated Communication System for Distributed Educational Computing Environments, Proc. of 5th IJWCC, pp. 173-180 (1990).
- [3] Jeff Conklin: Hypertext: An Introduction and Survey, IEEE COMPUTER, pp. 17-41 (1987).
- [4] 松浦, 市村, 平岩, 岡田, 松下: 協同作業のためのチーム指向環境, 情報処理学会研究会報告, 90-DPS-45, 45-20 (1990).
- [5] 松本, 楠本, 菊野, 鳥居: プログラム開発におけるチーム性能のモデルに基づく実験的評価, 情報処理学会論文誌, 31, 12, pp. 1812-1821 (1990).