

リモート協調作業によるソフトウェア開発知的環境

小泉 寿男 1) 鈴木 昌則 1)
白鳥 則郎 2)

- 1) 三菱電機株式会社生産システム本部
Ⓔ100 東京都千代田区丸の内2-2-3
- 2) 東北大学電気通信研究所
Ⓔ980 仙台市青葉区片平2-1-1

あらまし

本稿では、協調作業に基づくソフトウェアの知的開発環境について論じる。始に、プロトタイプの開発、実用版開発、更にユーザ使用後の改良開発までの連続的なソフトウェアライフサイクルを論じる。次に、このライフサイクルを、CSCW (Computer Supported Co-operative Work) によって遠隔地間で協調作業を行なう開発環境、および協調作業に於ける知的インターフェースについて提案する。

最後に、製品在庫最適化システムの構築によって実証した評価について考察を述べる。

和文キーワード ソフトウェア開発、遠隔地協調作業、プロトタイプ、画面共有

A Software Intelligent Development Environment based on Co-operative Work Among Remotely Separated Sites

Hisao Koizumi 1) Masanori Suzuki 1)
Norio Siratori 2)

- 1) Headquarters-Engineering, Manufacturing & Information Systems
Mitsubishi Electric Corporation
2-2-3 Marunouchi, Chiyoda-ku, Tokyo, JAPAN
- 2) Research Institute of Electrical Communication Tohoku University
2-1-1 Katahira, Aoba-ku, Sendai-shi, JAPAN

Abstract

This paper describes a software intelligent development environment which is based on cooperative work among developers. First, a software life cycle which covers prototyping, productization, user evaluation, and the improvement is explained. Then, to support such a life cycle, a software development environment which realizes cooperative work among remotely separated sites by CSCW and an intelligent interface between cooperative workers are proposed. Finally, the evaluation of the proposed system which was applied to the development of an optimal inventory system is discussed.

英文 key words Software Development, CSCW, Remote Sites, Prototype, Screen Share

1.はじめに

ソフトウェアの開発作業は複数メンバーの協調作業が基本となる。要求仕様を出すユーザ側とソフトウェアを開発する側の協調、ソフトウェアを分担して開発する協調作業等多数な形態がある。コンピュータとネットワークを活用して各種の協調作業を支援するCSCW(Computer Supported Co-operative Work)をソフトウェア開発に適用する意義は大きい。

ソフトウェアのライフサイクルに於て、仕様決定、設計・製作、デリバリ、保守・改良の各段階ではそれぞれの専門技術と知識、役割をもった人間が参画し、その協調の仕方、コミュニケーションの効率がソフトウェアの出来具合を大きく左右する。しかも、異なった専門技術者同志は、常に一堂に会して仕事をするのではなく、空間的、時間的に異なった環境で仕事するのが通常であり、空間と時間のギャップを埋めるCSCWへの期待は大きい。

情報技術の急速な発達によって、ダウンサイジング、オープンシステム、ネットワーク化が浸透してきた。こうしたトレンドの結果、実用的なソフトパッケージが出現し、コンピュータ使用範囲がユーザ部門へも広がり、ユーザが簡易なものであれば自らソフトウェアをつくるエンドユーザコンピューティングも一部の部門では行なわれている。

ソフトウェアライフサイクルの各段階に於けるCSCWの適用もこうした動向を実質にとりこんで行く必要がある。

本稿では、ソフトウェア開発者側と要求仕様を渡すユーザ側との協調関係のもとに、プロトタイプによる基本仕様の確認の段階から実用版開発、ユーザ使用後の改良版作成までのライフサイクルを進化的にすすめる方式を論じる。特に、このライフサイクルと、遠隔地の人間同志がCSCWによって協同作業を行なうための知的インタフェース方式として画面共有方式を提案する。

本稿の構成は、第2章に、ソフトウェア開発に於ける協調作業の課題を述べ、第3章に提案方式の概要、第4章に画面共有による協調作業の交信方式を提案する。第5章に適用したシステムの検証と考察を述べる。

2.ソフトウェア開発に於ける協調作業の課題

2.1 ソフトウェアのライフサイクル

ソフトウェアのライフサイクルで一般的に使用されているウォーターフォール型は、開発対象の分析、設計、プログラミング、テスト、保守の段階をとるが、仕様決定に長い時間を要することと設計段階での変更が上流工程へフィードバックしにくいことに難点がある。

仕様決定の段階では、ユーザ側のニーズが多様で優先度が決めがたく、かつ盛り込むべき多種機能の相互関係によってそれぞれの仕様に変更の必要性が生じる。そのため仕様が仲々確定しにくく、ソフトウェアが出来上がって見ないと判らない部分が多く残るという特質をソフトウェアが内在している。設計の途中で仕様のもれや誤りが検知されても、上流への仕様変更、関連する他部分への影響度が特定しにくいためそのまま作業をすすめ、プログラミング後のテスト段階で上流仕様の見直しを行なうことになってしまう。こうした繰返しがソフトウェア開発の工数を増大させ、時として収束困難な開発プロジェクトが発生する要因となる。

ソフトウェアができ上がりユーザ部門へ引き渡されると作成側は一定期間の支援のあとは保守・改良のフェーズに入る。一方、ユーザ側には実使用してみて出力表示内容、操作性、不足機能の各事項についての仕様変更要求がおきる。こうした変更要請が出されても作成側は保守・改良フェーズに入っているため要求が実現するまでには期待に反し多くの時間を要し実情に合わないことになり、ソフトウェアライフサイクル上の問題となる。

こうした課題に対して各種の研究が報告されている。仕様確認の段階でプロトタイプをつくる方法、対象システム内のデータをモデル化してエンティティリレーション図(ER図)を作成し、ソースコードの自動生成までもっていくものもある。

いずれも仕様確認や自動生成の観点からは有意義な技法であるが、協調作業との面からは別のアプローチが必要であろう。

筆者らは、プロトタイプによる仕様、確認性に着目し、ソフトウェアパッケージを組み合せ

ることにより、非定型業務の処理を主体とする情報系システムのプロトタイプを作り上げる手法をソフトウェアライフサイクルのベースに置いている。そして、ソフトウェア開発者とユーザ側との協調作業により、プロトタイプから実用版ソフトウェアへ連続的に進化させていく方式を開発した。

2.2 ソフトウェア開発の協調作業

CSCWは、人間の協調作業を支援し時間と距離のギャップを埋める方法として各種の研究開発と試行が報告されている。ディスプレイ装置とLANを接続した電子会議システム(1)他(2)、離れた場所にいる人間同志が自席にいたままワークステーションを使ってリアルタイムに会議や情報交換を行うMERMAID(3)が主な例である。ソフトウェア開発への応用に関しても各種の例が報告されている(4)(5)。以上の内容は電子会議を主体としたCSCWであり、ソフトウェア開発に於ける情報の交換、仕様書、ソースコードの伝送確認が主体となっている。筆者らの研究は、ソフトウェアの開発プロセスそのものを、仕様にもとづく計算処理の結果による通信によって、異なった専門者同志の協調作業により行うものであり、ソフトウェアライフサイクルへの新しい提案をでもある。

3 本方式の概要

本章では、ソフトウェアパッケージを組み合わせ、ソフトウェア開発側とユーザ側が協調作業によりプロトタイプを作成し、これをベースに実用版に進化させていくソフトウェアライフサイクルについて述べる。特に、協調作業に於いて、遠隔地間の異った専門者同志が画面を共有し、エージェントの支援によってコミュニケーションを行なう方式を第4章で提案する。

3.1 ソフトウェアパッケージ活用によるプロトタイピング

3.1.1 ソフトウェアパッケージ

情報技術の急速な発達はいくつかのプロセッサとオープンシステムを現実なものとし、データ

ベース、スプレッドシート、グラフ処理等の汎用ソフトウェアパッケージが実用的なものとして普及するに至った。ソフトウェアパッケージは、その内部構造を知らなくとも必要な処理を動作させるコマンド、マクロ言語を記述することにより所要の機能を果たすことができる。

また、利用者が機能を付加しカスタマイズするためのマクロコマンド、他のソフトウェアパッケージの機能を組み合わせるインタフェースコマンドによりソフトウェアを構築できる。

ソフトウェアパッケージの出現は、ソフトウェア作成にかゝる人間の中を広げ、開発者とユーザのギャップを縮める。ソフトウェア開発に関する従来からのパラダイムを変える1つの革新と言っても過言ではない。

ソフトウェアパッケージを更に有力にするものとしてクライアント・サーバシステムがある。データの蓄積や共通の計算処理を行うサーバに、LAN経由した複数のクライアントを接続したシステムであり、クライアントからの処理要求によりサーバがデータを供給したりサーバでの計算処理結果を要求元のクライアント側へ提供するシステムである。

データベースソフトをサーバに、スプレッドシートソフトをクライアントに搭載してソフトウェアを構築する方式を図-1に示す。

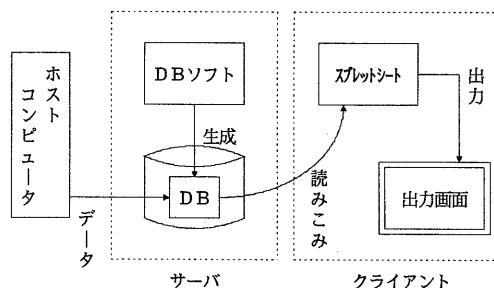


図-1 クライアント・サーバシステム上のソフトウェア

サーバには、データベースソフト(DBソフト)によって編成されたデータベース(DB)に、ホストコンピュータから必要なデータが転送され格納される。クライアント上のスプレッドシートソフトは与えられたコマンド処理に応じてDBから必要データを読み込み、同様にコマンド処理に則ってデータの加工を行った結果をグラフ、表等の出力形式に変換して画面に描

く。

3.1.2 プロトタイプからの進化

以上のようにソフトウェアパッケージの組み合わせによって構築する情報システムは一般にプロトタイプと呼ばれる。プロトタイプは、ソフトウェア開発者とユーザ間でソフトウェア仕様を確認する有力な手段であるが、仕様確認の目的が達せられると通常は捨て去られるのが従来の考えであった。

しかし、上述のようにソフトウェアパッケージの組み合わせによってプロトタイプが構築できれば状況は大きく変る。プロトタイプをベースとして、ソフトパッケージの各コマンドによる調整、チューニングを繰返しながら実用化システムに進化させることができプロトタイプからの連絡性が保てる。

この場合の開発のステップを図-2に示す。

筆者らはこうした進化の過程をソフト開発者側とユーザ側との遠隔地間協調作業によって行なう方式を提案する。

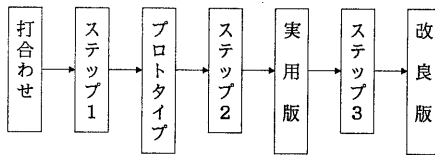


図-2 ソフトウェア開発のステップ

3.2 協調作業の方式

3.2.1 協調作業のモデル

ソフトウェア開発における協調作業は、異った専門技術者間の作業のケースと同一の専門者間の作業のケースがあり、次の様に分けられる。

<同一専門内メンバー>

専門者群 1	S 11, S 12, ... S 1n
// 2	S 21, S 22, ... S 2n
...	...
m	S m1, S m2, ... S mn

本研究では特に、異った専門者間の協調作業に焦点をあてる。この理由の1つは専門の違いをCSCWによってどれだけカバーし得るかを見極めること、理由の2つは異った専門者の方

が同一専門者群より離れた地域にいる可能性が高く、リモート協調作業によって距離的空間のギャップが埋まれば、CSCWによるソフトウェア開発効率化の意義が大きいからである。

異った専門者群は1~m群存在するが、ソフトウェア開発の協調作業という主点から本稿では次のように3群とする。

- ・ソフトウェア開発者：情報技術を基盤とし、ソフトウェアパッケージの知識と組み合わせ技術をもつ。……A群と略称する。
- ・ユーザ側の仕様のとりまとめ者：A側との接点となって協調作業を行なう。……B群と略称する。
- ・エンドユーザ：開発したソフトウェアを実際に使って仕事を行なう専門者。……C群と略称する。

3.2.2 協調作業と進化的ステップ

A, B, C群の各専門者の協調作業により、プロトタイプ→実用版→エンドユーザ使用後の改良版へと進化させていく関連を図-3に示す。

A, B, C群は通常場所的に離れて地点にいる。しばしば会合を持つことなく、始めの打合わせ後はCSCWの支援により、リモート間での協調作業を原則とする。

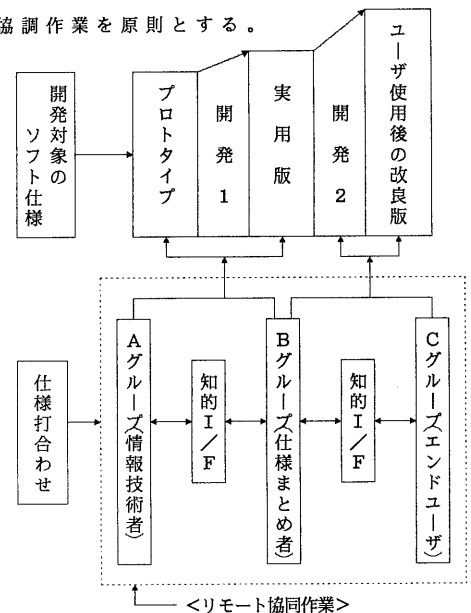


図-3 ソフトウェアライフサイクルとリモート協調作業関連図

3.2.3 協調作業間の交信方式

本方式の提案は、画面共有によるリアルタイム型を基本方式としている。仕様の確認を、画面に表示される計算処理結果をベースとした画面上の対話によって行なう方式が有効な協調関係をつくる。

協調作業を補足するため、電話と電子メールを活用する。したがって、協調作業のコミュニケーション方法は次の3種類の組み合わせで行なう方式をとる。

(1)画面共有（リアルタイム同期型）

同じ時間帯に協調者同志がワークステーションの画面をシェアしながら交信する。

(2)電話

①の補助として使う。

(3)電子メール（非同期型）／ファイル伝送 相手の時間帯に関係なく送受信する。

4.ソフトウェア協調開発の画面共有方式

4.1 画面共有の内容

ソフトウェア開発協調作業間交信のベースを共有画面におくに関して必要な要素は次項である。

- 1)開発すべきソフトウェアの仕様を仕様記述または仕様の計算処理結果として表、グラフにて表示するもの。
- 2)表示結果の内容を相手方が理解する助けとして、計算処理の方法・式、データ構成を示すもの。
- 3)相手先へのコメントをメッセージとしてやりとりするもの。
- 4)交信のシーケンスを指示し、ステータス表示を行なうもの。
- 5)交信の履歴をログしておき、交信の途中必要に応じて前に遡り、理解を支援するための編集をして表示するもの。

以上の要素は、協調作業の交信の観点から次の様に分類される。

- 仕様記述、仕様計算処理結果の出力表示
 - 表示内容の理解支援、交信のガイド支援
 - 交信シーケンス指示、ステータス表示
- これらを画面上の論理的なエリアに区分して、図-4に示すようにそれぞれ○出力エリア

○エージェントエリア ○コミュニケーションエリアとよぶ。

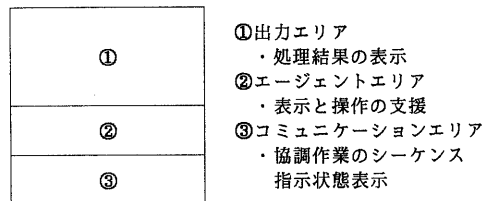


図-4 画面共有の各区分機能

①出力エリア：開発対象ソフトウェア仕様の記述、およびコンピュータによる処理結果をグラフ、表の形で表示する。①の表示内容は、処理結果が画面シェアソフトの機能によってそのまま協調作業の相手側に伝送される。

②エージェントエリア：①に表示される内容の理解を支援する役割を持つもので、エージェントによって表示される。計算処理に使った式の内容、データの内容を表示する。協調作業中の交信履歴ログを活用し、交信を支援する役目も持つ。メニューによってエージェントの機能を選択させることができる。

③コミュニケーションエリア：協調作業の交信シーケンスの指示、状態表示、コメントメッセージを伝達する（電子白板機能）。

4.2 エージェント機能

図-5にエージェントの位置づけと役割を示す。

スプレッドシートとDBソフトの組み合わせによって構築されたソフトウェアは、業務DBの内容を読み出し、仕様に基づいて計算処理、加工処理を行ない、結果を出力エリア①に表示する。エージェントは、各種情報と知識を活用して、出力エリア①に表示された内容を理解する支援をエージェントエリア②によって行なう。まず、出力エリア①の表示結果を説明するためのメニューをエージェントエリア②に示す。この中には、結果を出すに至った処理方式・計算式、データ構成が含まれ、メニューによって選択され、その内容が出力エリア①（物理的には別ウィンドウ）に表示される。

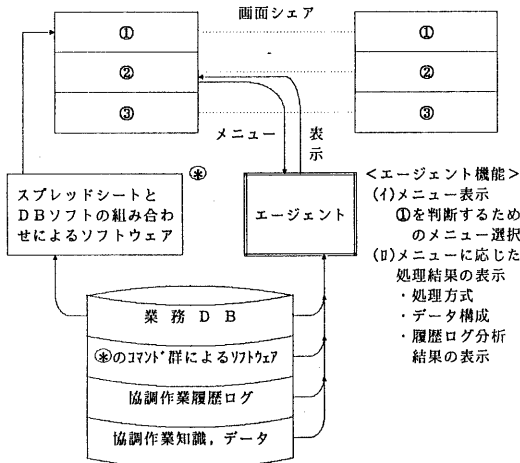


図-5 エージェントの位置づけと役割

更にエージェントは協調作業間のコミュニケーション履歴をログしておき、交信の過程で参照、編集し協調作業への便を図る役割をもつ。編集の際、協調作業A, B, Cの属性を知識をして記憶しておき、交信の組み合わせ(A ↔ B, B ↔ C, A ↔ C)によって編集内容を適応化させる機能を持たせる。

4.3 コミュニケーションの方式

4.3.1 コミュニケーションエリアの内容

コミュニケーションエリア③の内容と役割を表-1に示す。

仕様名:	エリア①, ②, ③により協調作業を行っている仕様名(ソフトウェア名)を示す。
仕様の確認:	相手方へ、仕様を処理結果に基づいて確認し妥当か否かをOK, NOにて知らせる。
コメント:	相手方へのメッセージを伝える。
ステータス:	相手方へのステータスを知らせる。
表示:	相手方への処理結果を表示し、判断を依頼する意味をもつ。
検討:	エリア①の表示内容を見、エリア②のメニューによって処理の内訳を確認し妥当性を検討する。確認の結果は、上記「仕様の確認」によりOK, NOを知らせる。理由は、コメント欄を利用する。
変更:	相手方からNOの回答を得て、変更処理を行っているステータスを示す。
次の仕様:	次の処理を行っているステータス
前の仕様:	すでに行った処理を再表示。
終了:	一連の協調作業の終了。

表-1 コミュニケーションエリアの内容説明

4.3.2 コミュニケーションの方式

画面共有によってソフトウェアの開発の協調作業を行なう方式を述べる。先ず図-6によって、協調者A, Bが協調者を行なう交信の状況を示す。

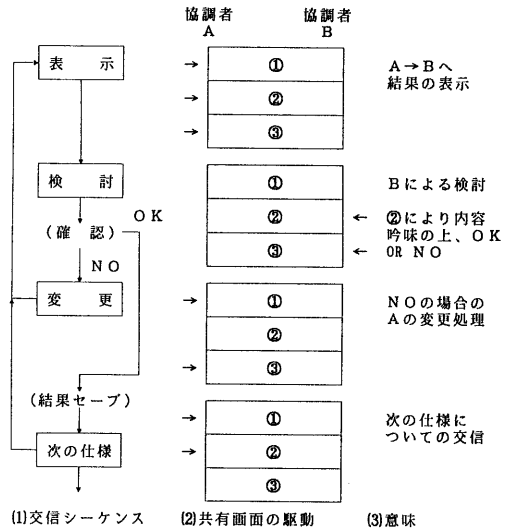


図-6 画面共有によるコミュニケーション図

協調者A(以下Aと略す)は処理結果を出力エリア①に表示し、理解の為のメニュー選択をエリア②に表示して表示をオンにする。次に協調者B(以下Bと略す)は検討をオンにし、結果の吟味・検討を始める。この作業において、エージェントの機能により①に出力された結果を吟味し確認OKであればOKを指定することにより、Aは次の仕様の作業に入る。

確認の結果が納得いかなければNOを指示し、コメントをコメント欄に表示する。

AはBがNOのレスポンスをしたことを知り、ステータスを変更しに切りかえて変更の作業に入り、望ましいと思われる結果が出たら再び表示を指示して相手方の検討を待つ。この一連の作業の繰り返しによって1つの仕様に対応したソフトウェア作成を行なう。これが終了すると次の仕様に関する交信シーケンスに移る。

協調者BとC, AとCについても交信の方式は同様である。

3.2.2の図-3に示したように、AとBはプロトタイプから実用版までの進化的開発を行なう。BとCはエンドユーザが使用后、実際の業務上からの改善的をおこむ為の協調作業を行なう。BがCの仕様要求に応じた対応が専門知識上困難の場合は、AとCとによって行なう。また、始めの仕様打合わせ後のプロトタイプ作成までのプロセスに於いては、B側の方が先ず要求仕様のイメージを簡単な作図、作表によってA側に伝え、それをもとに、A側がソフトウェアパッケージの組み合わせによりプロトタイプをつくり上げていく交信が行なわれる。

5. システムの検証と考察

5.1 システムの検証

本方式の検証を最適在庫配置モニタシステムのソフトウェア開発に適用して行なった。

開発すべきソフトウェアの仕様は、製品毎の在庫推移、品切れ・過剰検知、在庫シミュレーション、長期解析等の計算処理である。ホストコンピュータのデータベースに蓄積されているリアルタイム情報から業務に必要な情報を、クライアントサーバシステムのサーバに毎日オフロードし、部門単位の業務DBを生成する。実証システムは、この業務DBに対し、DBソフトウェア、スプレッドシートソフトウェアの組み合わせで作成したコマンド群が仕様に対応して処理を実行する方法とした。

検証システムのネットワーク網構成を図-7に示す。

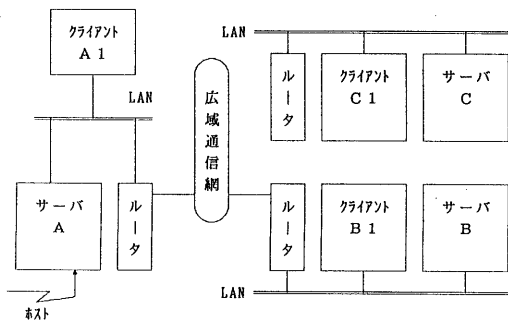


図-7 検証システムのネットワーク網構成

クライアントとサーバはLANによって接続され、それぞれのシステムはルーターと広域通信網経由にて接続される。

各クライアントは、他のサーバにアクセス可としている。

AとB、BとC、AとC間による協調作業の実施試行した。

リモート協調のパターンを図-8に示す。実体は画面共有を主体として、電話、電子メールの組み合わせとなった。

画面共有のシステム例（在庫管理における長期グラフ図）を図-9に示す。

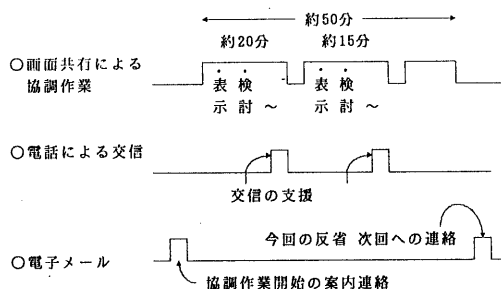


図-8 リモート協調作業の支援パターン

5.2 考察

ソフトウェアパッケージの組み合わせによって実証システムを構築してプロトタイプを作成し、これをベースに実用化レベルのソフトウェアに進化させていくプロセスを、異った専門者同志によるリモート協調作業で行なう方式の基本的な有効性は確認できた。具体的事項として次項があげられる。

- (1)非定型の情報システムをソフトウェアパッケージの組み合わせで作成できることはソフトウェア開発に於けるユーザの参画を促進させる。リモート間での協調作業によって更に加速される。
- (2)本稿で提案している画面共有方式は、各メンバーが事前に相手に確認すべき事項を整理、レビューしておくことによって協調作業が可能となる。ただし、計算結果の内訳を知るためのメニューの種類については今後の充実が必要である。コメントメッセージの伝達は相手先への理解を助ける。

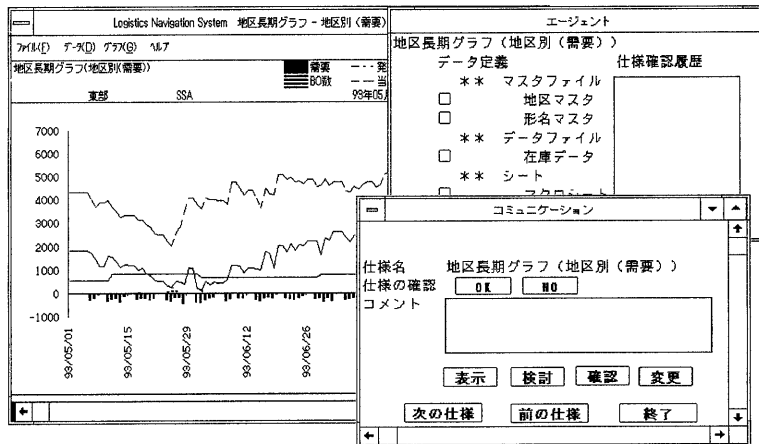


図-9 実験システムの画面例

- (3)表示→検討→確認→変更のシーケンスは、協調対話上妥当と考えられるが、相手側にこのステータスを常に認識しておく為の表示、レイアウトには今後工夫を要する。今後の課題としては次項のものが残されている。
- 1)画面共有における論理的3分割とウィンドウとの関連をわかり易くすること。
 - 2)専門者の属性をエージェントの中に効果的におりこむこと。
 - 3)LAN間接続において伝送されるパケット数を減少させるためルータに有効なフィルター機能を持たせること。
 - 4)今回の実験では対象外としたが、データのセキュリティに対応すること。

6.むすび

CSCWをソフトウェア開発に適用して生産性向上に役立てていくことは今後の大きな課題の1つである。リモート協調作業によって、ユーザの参画が加速され、ソフトウェア開発に新しいパラダイムの生じる可能性もある。

協調作業の方式確立の為には、方式の提案、実験による検証の繰り返しが必要であり、今回の提案がこうした研究の一助になることを望んでいる。

参考文献

- (1)Stefik, Mark et al : Beyond the Chalkboard: Computer Supported for Collaboration And Problem Solving In Meetings,Comm. of the ACM, Vol.30,No.1, pp.32-47,1987.
- (2)I.T.Hawrgozkiewicz : Supporting Coordination in CSCW Supems, Proceedings ISADS 93,pp.225-231
- (3)阪田史郎, 上田鉄雄 : 構内型マルチメディア在席会議システムの実現とその評価, 情報処理学会論文誌, Vol.31,No.2,pp.249-258(1990).
- (4)垂水浩幸 : グループウェアのソフトウェア開発への応用, 情報処理学会誌, Vol.33,No.1,pp.22-31(1992).
- (5)Prasum Dewan :Toward Computer-Supported Concurrent Software Engineering, COMPUTER,Jan.1993,pp.17-27.