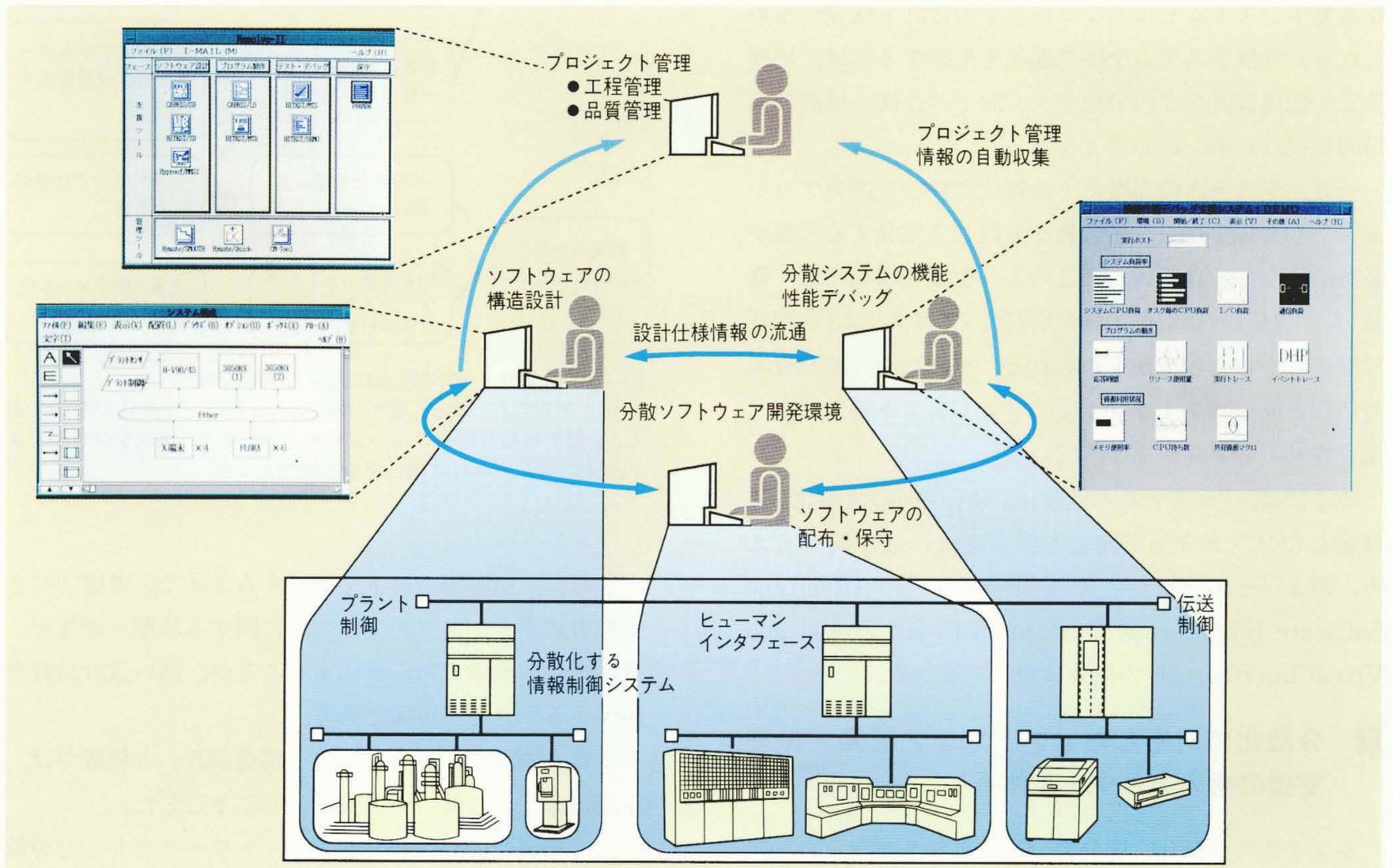


# 分散化が進む情報制御システムの ソフトウェア開発・保守支援システム“Resolve-Ⅱ”

Computer Aided Software Engineering System for Distributed Realtime Systems

和田 裕\* Yutaka Wada 高橋勇喜\*\* Yūki Takahashi  
岡部幸弘\*\* Yukihiro Okabe 大脇隆志\*\* Takashi Ōwaki



## 分散システムを多人数で開発する環境を支えるソフトウェア開発支援システム

一元管理された設計仕様情報・管理情報を、多人数にわたって整合をとるスムーズに流通させることにより、設計効率・設計品質の向上に寄与する。

コンピュータのネットワーク化に伴い、情報制御システムでも分散化が急速に進展している。また、超高速ネットワークの普及に伴い、開発環境の分散化に拍車がかかっている。

これらの分散化に起因して、分散システムの設計・テストの難しさや、設計仕様・プロジェクト動態管理の複雑化など、ソフトウェア開発・保守上に新たな課題が生じている。

これらの課題に対応するため日立製作所は、従来

のソフトCAD・CAM・CAT(Computer Aided Testing)一貫支援システムを強化・拡張し、「分散システムに対応した開発・保守支援」、「分散開発環境における多人数協調開発の支援」をねらいとした「情報制御システム向けソフトウェア開発支援システム」を開発した。

これにより、分散システムの開発・保守がいつでも効率よく行うことができ、高品質な情報制御ソフトウェアの開発が実現できる。

\* 日立製作所 大みか工場 工学博士 \*\* 日立製作所 大みか工場

## 1 はじめに

近年の情報制御システムでは、高い信頼性、速い応答性、柔軟な拡張性への要求から、複数の処理装置がネットワーク上に接続された分散アーキテクチャが多く採用されている。分散システムでは、従来の単一マシンからなる集中システムとは異なる視点での設計・保守が求められる。分散システム全体の機能を把握するには、処理装置・伝送装置などの分散マシン間での連携・協調の大局的な視点が新たに必要となる。

一方、システムの高度化・複雑化に伴って開発プロジェクトも大規模化し、多人数が分散して開発する形態が必然的になってきている。このような開発環境では、資源(人、設備)と作業成果物(仕様書、プログラム)が相互に関連しながら作業が進められる。そのため、開発資源の情報と成果物をより効率よくプロジェクト内で共有、伝達させる必要性が高まっている。

ここでは、「システムの分散化」と「開発の分散化」に対応したソフトウェア開発・保守支援の基本的な考え方、およびその支援システム“Resolve-II”(Realtime Software Engineering Platform with Integrated and Visual Interface II)の概要について述べる。

## 2 分散化に対応したソフトウェア開発・保守支援の考え方とツール体系

システムの分散化、開発の分散化に伴うソフトウェア開発上の課題とその解決策を図1に示す。

### 2.1 システムの分散化に伴う課題とその解決策

分散システムの利点、例えば「機能を分割しマシン間で負荷配分できる」、「分散マシン間で相互に処理を監視しあい、異常時には処理分担を変更できる」などを享受するには、集中システムに比べて「開発対象が複雑化する」問題を、システム開発の各段階でどう克服するかが鍵(かぎ)となる。

#### (1) システムの設計、製作上の課題

分散システムで自動化・支援される業務は、分散マシン上の機能を連携させて実現される。顧客側の業務要求を、開発側で確実にシステム機能に反映させるには、双方が共通の視点で機能をとらえる必要がある。

分散システムの機能を把握するには、(a)分散マシン間の機能の関係、(b)機能の細部の両面から、ソフトウェアモジュールへの振り分けと分散マシンでの処理分担を明確に定義することが不可欠である。

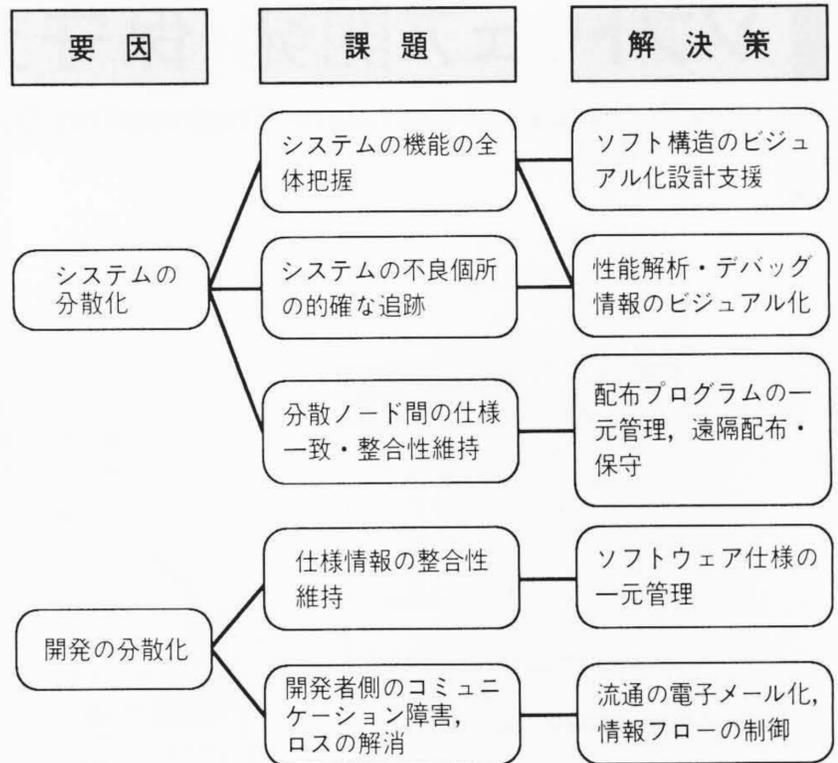


図1 「分散化」の問題点に対する解決方針

分散システムの構造、動作の把握にはビジュアル化が有効である。分散する設計情報には一元化が有効であり、情報をプロジェクト全体にスムーズに流通させる必要がある。

さらに、顧客側の要求を反映するうえで、考慮すべき処理方式やインタフェース仕様に関する基準・条件を、設計の早い段階で明らかにすることが、高い設計品質を確保するうえで不可欠である。

これらの設計作業を「ソフト構造設計」と位置づけ、下記の考え方に基づいて支援することとした。

- (a) 機能間のつながりを、ネットワークを介した分散マシンの接続図の上でビジュアルに定義できること。
- (b) 機能に含まれるソフトウェアモジュールの処理・データフローや機能分割範囲を、だれが見ても同じ解釈ができる一定の記法で定義できること。

この支援により、顧客側にとっては、システムに対する要求が的確に表現でき、また開発側にとっては、要求に対応させて設計作業を展開させる出発点を得ることができる。

#### (2) システムのテスト、デバッグの課題

従来、システムのテストには、制御対象プラントの動的ふるまいを模擬するシミュレータが利用され効果を上げてきた。また、デバッグ段階では、プログラムの処理履歴を記録し、プログラムのどの部分に問題があるかを見つけ出すことが有効である。

分散システムでは、ネットワークを介したプログラム起動や、分散したデータベースを介したデータ授受が多く用いられる。ここでは、個々のマシンごとに記録した

処理履歴を突き合わせながら、分散マシン間でのプログラム連携状態を把握する必要があった。

解決策として、同一時刻での複数マシンでのデータ設定・処理状況を網羅して記録し、これらを時系列順に一覧表示して、「分散システムの機能・性能デバッグ」を支援することとした。

この支援は、顧客側でのシステムの動作状況のモニタにも活用でき、ネットワーク負荷の異常、多量データ処理による性能悪化などの把握が可能となる。

(3) システム構築・保守上の課題

分散システムでは、プログラムを初期インストールするシステム構築時や、保守のための変更・修正時には、下記の点を配慮することが重要である。

- (a) 複数マシンに一括配布する共通プログラムのバージョン一致化
- (b) 共通仕様を持つ複数のプログラムを部分的に入れ替える際の整合性の維持

この際、漏れや誤りなくプログラムを配布・入れ替えて、整合性の有無を判断し、プログラム配布状況を確認するには、多大の労力を要する。

解決策として、すべてのプログラムを1台のマシンで一元管理し、一括してプログラムの配布・入れ替えを行う「遠隔保守・配布管理」を支援することとした。

この支援は、一覧表示画面上のマウス操作だけでプログラムの配布・入れ替えを可能にするもので、顧客側でもシステムの構築・保守が容易に行えるものとなる。

2.2 開発の分散化に伴う課題とその解決アプローチ

(1) システムの設計上の課題

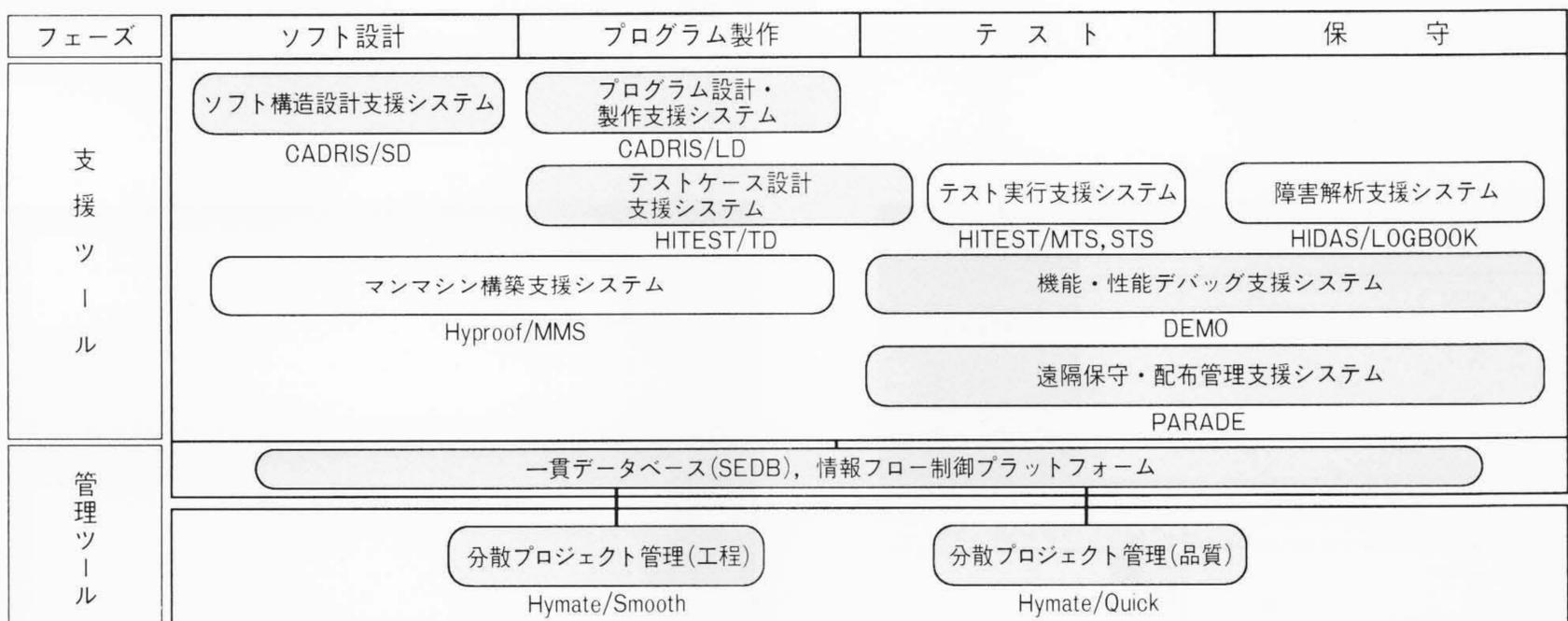
複数の機能を分散して開発する際には、最新の設計結果を、多数の開発者間で共有することが不可欠である。しかし、分散した地点間で、個々に設計情報間の整合性を常に維持し続けることは困難である。

解決策として、最新の設計結果を1台のマシンに集約し、仕様情報に自動反映させる「ソフトウェア仕様の一元管理」を支援することとした。

(2) プログラム製作、テスト上の課題

プログラム製作、テストでは、作業に必要な情報の依頼や作業結果の通知に、ツールの使用環境を離れた別の手段(電話、口頭、メモなど)を用いていた。人手を介したテスト作業「依頼」、デバッグ結果「通知」などのコミュニケーションのやり方では、漏れや誤りの防止に限界がある。

解決策として、テスト状況を一元管理し、テストケース・テスト結果が各分散地点の担当者間で共有できる「テストケース設計」を支援することとした。



注：略語説明 CADRIS (Computer Aided Design and Reuse Environment with Intelligent Support), CADRIS/SD (CADRIS/Structured Design) CADRIS/LD (CADRIS/Logic Design), HIDAS (HIDIC Diagnosis Assistant System), HITEST (Hitachi Integrated Test System) HITEST/TD (HITEST/Test Case Design), HITEST/MTS (HITEST/Module Test System), HITEST/STS (HITEST/System Test System) DEMO (Determinate Evaluation, Monitoring and Output System), Hyproof/MMS (Hyper Programming System by Object-Oriented Formula/ Man-Machine System Support), SEDB (Software Engineering Database), Hymate (Hyper Project Management System under Distributed Environment), Hymate/Smooth (Hymate/Software Development Process Monitor and Control Support System Using Hierarchical Network) Hymate/Quick (Hymate/Quality Improvement Check and Support System), PARADE (Program Remote Maintenance Aide System)

図2 「分散化」対応情報制御システム向け“CASE”のツール体系  
 ○の部分分散化に対応して、強化・拡張した。

(3) プロジェクト管理上の課題

プロジェクト管理者は、各工程での作業と品質の目標をすべてのプロジェクトメンバーに伝え、各作業の進捗(ちよく)状況、生産状況をすべての工程にわたって横断的にとらえる必要がある。しかし、人手を介した「指示」、「報告」などのやり方だけでは、現状把握と的確な管理をプロジェクトの隅々にまで行き渡らせることが困難となる。設計ツールの使用環境内で発生したプロジェクト管理にかかわる情報は、管理者や関連作業員などの受け取るべき相手に送り出されること、さらに、相手先の管理・設計ツールに誤りなく反映されることが必要である。

解決策として、人と人との間での情報交換として従来位置づけられてきた電子メールの利用形態を拡張し、支援ツールへのメール機能の付加を容易とする「情報フロー制御プラットフォーム」を導入することとした。

2.3 ツール体系

「分散化」に伴う上記の問題に対し、Resolve-IIが提供するツール体系を図2に示す。これは、従来のソフトCAD・CAM・CAT一貫支援システム“Resolve-I”<sup>3),4)</sup>に対し、「分散化」への対応のための機能を新たに導入、強化したものである。

Resolve-Iから継承したマンマシン構築支援、テスト支援、障害解析支援の各システムでは、集中システムに対応する支援の考え方が直接分散マシンに適用できる。

3 システムの分散化に対応した支援システム

分散システムの実体をとらえるには、その構成要素を直接目に見える形で表現するツールが必要である。また、分散マシンでの静的なプログラム配布管理やその動的な処理状況の監視には、表形式で情報を一覧し、一括指定できるツールが必要である。

3.1 ソフトウェア構造設計支援システム“CADRIS/SD”

CADRIS/SDは、ブロック図形式による「GUI(Graphical User Interface)ベース支援」、設計の要所で考慮すべき項目を提示する「設計ガイド支援」を実現している。

この支援システムの基本部は、製品分野に応じて確立しつつある設計のやり方を尊重し、多様な設計技法に合わせた記法・記号の再定義が容易な図形編集機能である。

CADRIS/SDを用いると、図3に示すように機能構造を階層化し、分散した資源が担う詳細機能を図形的に定義できる。

これにより、システムに対する要求仕様を顧客側で的確に表現でき、開発者側に伝えることが容易となる。

3.2 分散システムの機能・性能デバッグ支援システム“DEMO”

複数のマシンから構成されるシステムの解析情報の収集を効果的に支援するツールを開発した。その特徴は、複数マシンの連携処理の流れをミリ秒単位の分解能で追跡できることである。

全マシンの処理・負荷状況を1画面上に集約して表示

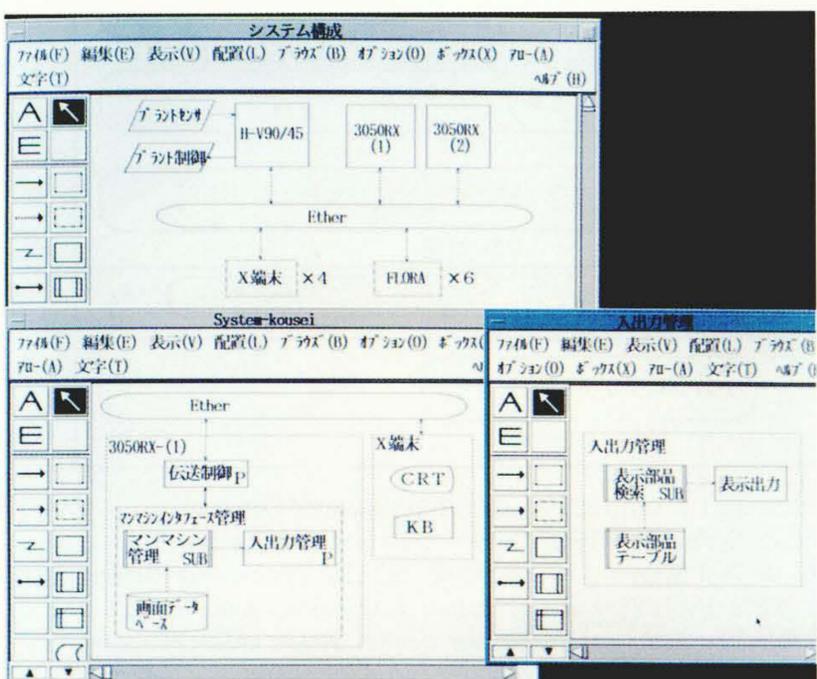


図3 ソフトウェア構造設計支援システム画面  
システムの機能を実現するソフトウェア構成を、一貫した記法を用い、目に見える形で組み立てていく。

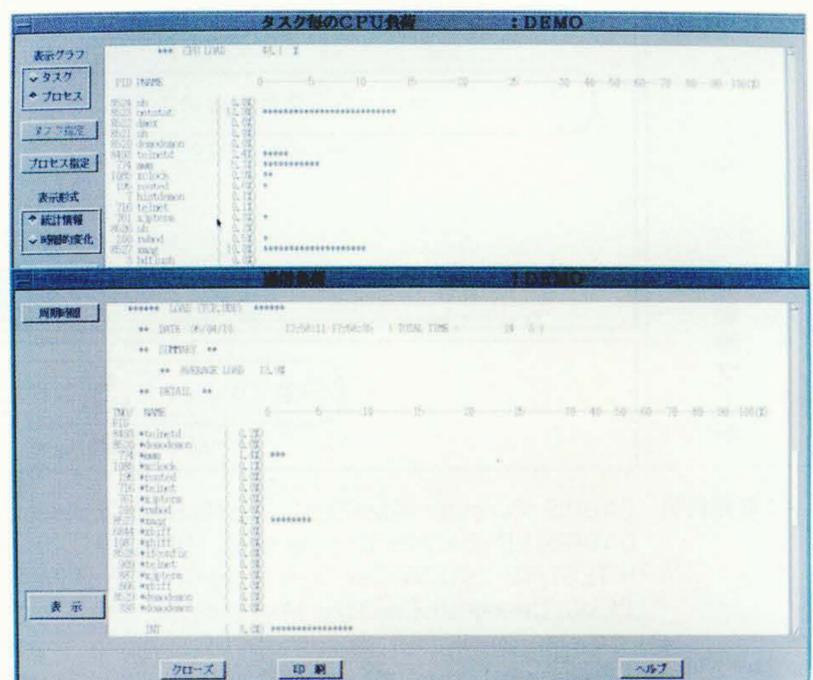


図4 分散システムの機能・性能追跡結果  
時間経過とともに変化する各マシン内の処理内容・負荷状況がミリ秒単位で追跡できる。

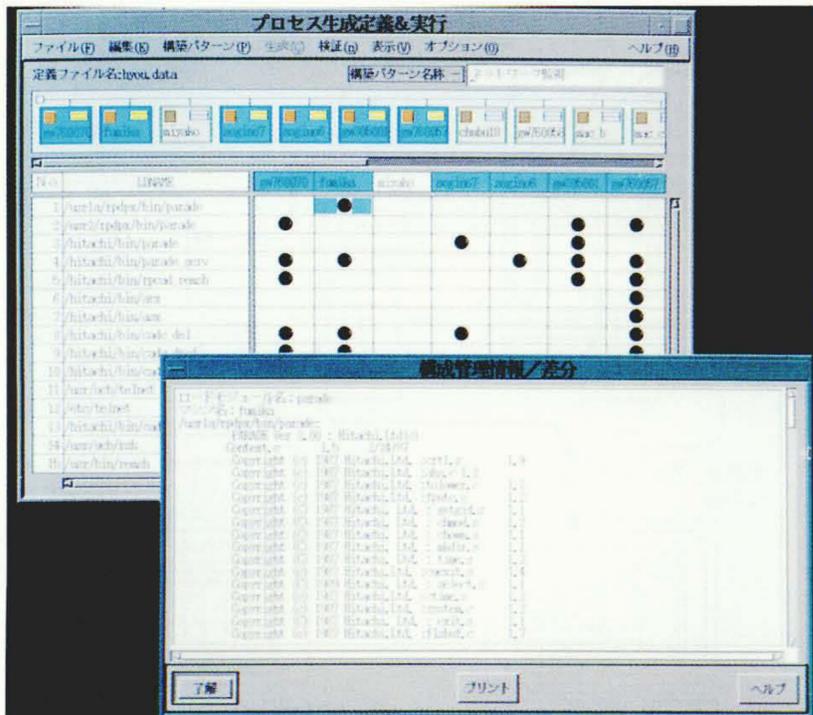


図5 遠隔保守・配布管理支援システム “PARADE”

プログラムの登録状況と版の一致状況が一覧で把握でき、不整合部分はワンタッチで解消できる。

することによって、例えば、

(1) 待ち状態となる処理が性能を悪化させていないか、(2) 複数の処理からの単一メモリへのアクセス競合が発生していないかなどが把握できる。これにより、顧客側でも、性能悪化などの監視が容易となる(図4参照)。

### 3.3 遠隔保守・配布管理支援システム “PARADE”

PARADEは、プログラムごとに登録した構成情報を収集することにより、マシン間のプログラムの整合状況を管理する。遠隔地点の保守・管理マシン上で、(1) 分散システム内の計算機構成・プログラム配布・登録状況がビジュアルに概観でき、(2) ワンタッチで、指定したプログラムの配布・登録ができる特徴を持つ。

プログラム管理者は、配布状況に不整合が生じると画面上に警告表示されるソフトウェアに対し、ワンタッチで再配布・再登録を指示するだけで済むので、容易にマシン間のプログラムの整合性が維持できる(図5参照)。

## 4 開発の分散化に対応した支援システム

ソフトウェア仕様の一元管理には、分散開発のために分割して開発した個々の仕様・プログラムを一つのシステムとしてまとめ上げるときに、ソフトウェアの機能構造や階層関係の一貫性が保てる仕掛けが必要である。

一方、開発者間のコミュニケーション障害の解消には、情報の発生した地点で即時これをとらえ、開発環境内に流通できる仕掛けが必要である。

### 4.1 ソフトウェア仕様の一元管理支援

#### (1) プログラム設計・製作支援システム “CADRIS/LD”

CADRIS/LDは、定型化された図表形式のモジュール仕様やPAD(Problem Analysis Diagram)を用いて、トップダウンかつ高品質なプログラム設計・製作を支援する。

CADRIS/LDは、複数のソフトウェアモジュールをまとめて管理する単位として、「パッケージ」の概念を採用する。パッケージは、システムの機能の単位と作業分担の単位の両側面から定義できる(図6参照)。パッケージ単位の版管理機能を用いることにより、多人数で分担開発した複数のパッケージを連結した際には、「どのプログラムをだれが開発し、版は何か」を管理することができる。これにより、分散開発されたソフトウェア全体の構成管理が容易になる。また版管理情報は、プログラム製作以後のトレース情報として利用され、変更作業の制御・管理が容易となる。

#### (2) テストケース設計支援システム “HITEST/TD”

テストケース設計とテスト実行を分担して行った結果(テスト状況・件数など)を集約し、一元管理する。機能間の組み合わせテストに先立って、各機能に対して分散地点で実行したテスト状況が把握でき、単体機能での漏れを防止し、テスト品質の向上を図る。

各テストケースについては、各機能への入力情報とその組み合わせ条件をマトリックス上に集約することにより、テストケース設計が可能となる(図7参照)。入出力情報の名称とテスト条件はデータベース上に一貫保持

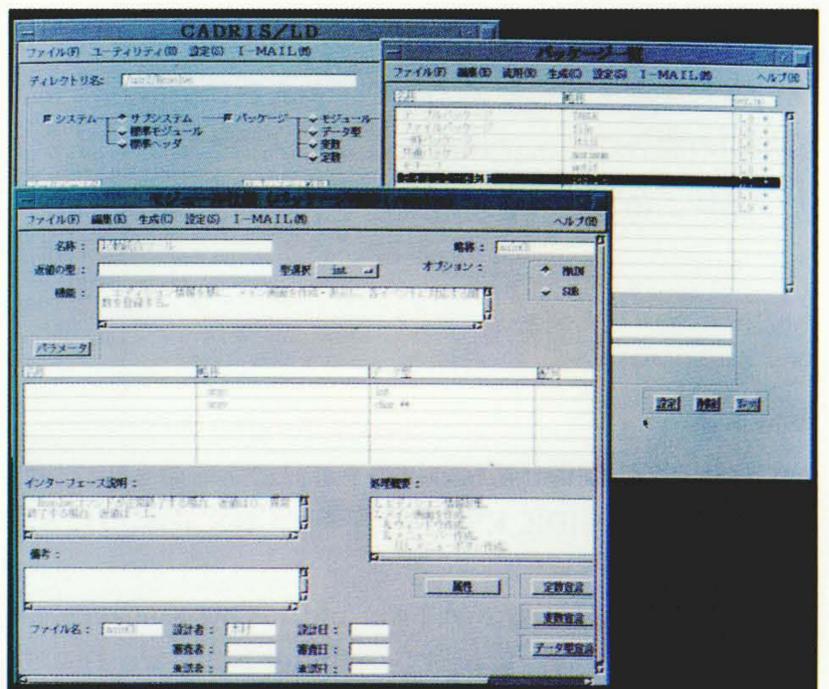


図6 プログラム設計製作支援システム “CADRIS/LD”

システムの機能の単位に「パッケージ」の概念を採用した。パッケージ単位の版管理が行え、変更作業の制御・管理が可能である。

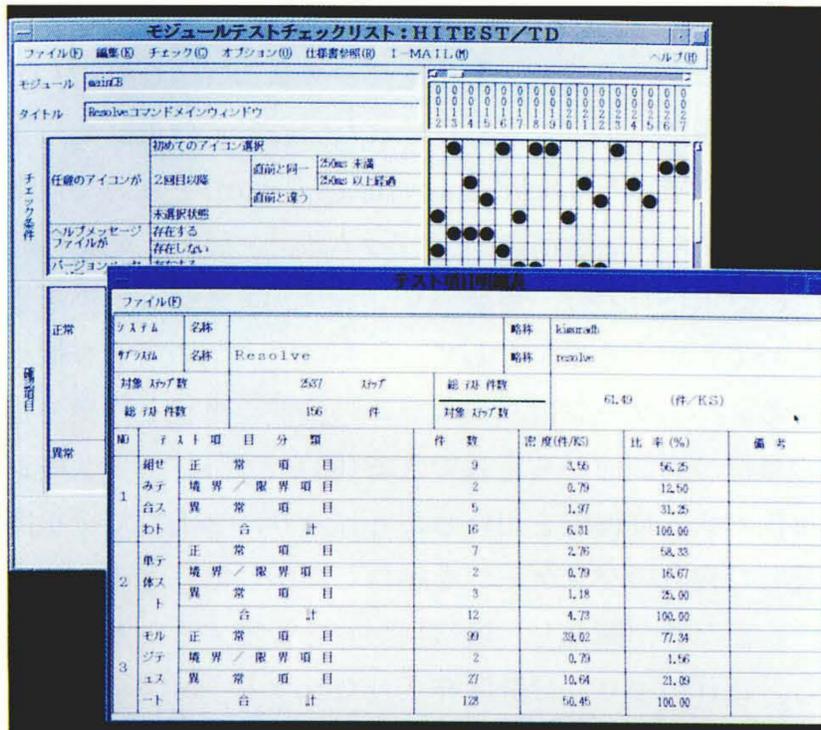


図7 テストケース設計支援システム“HITEST/TD”  
機能・項目ごとに分担設計したテストケースとテスト結果をマトリックス上に集約することにより、テスト状況が一目で把握できる。

し、CADRIS/LDで作成したソフトウェア構造に対応させて管理する。これにより、パッケージに対するテストケースを参照し、パッケージ内部のモジュールに対するテストデータへの継承が容易となる。

#### 4.2 情報フロー制御プラットフォーム

各ツールとの情報交換を仲介するためのコミュニケーションプラットフォームを設けることによって、人とツールの間、ツールとツールの間の情報交換を実現している。また情報の発信元は、送るべき情報が発生した時点で、受信側の支援システムの動作状態に依存せずに、即座に送信できる(図8参照)。

このプラットフォームを適用し、各支援ツール間の連携動作を実現することによって、多人数に分散した設計情報間の整合性を常に保持することができ、プロジェクトの進捗・品質状況の把握をタイムリーかつ的確に行う

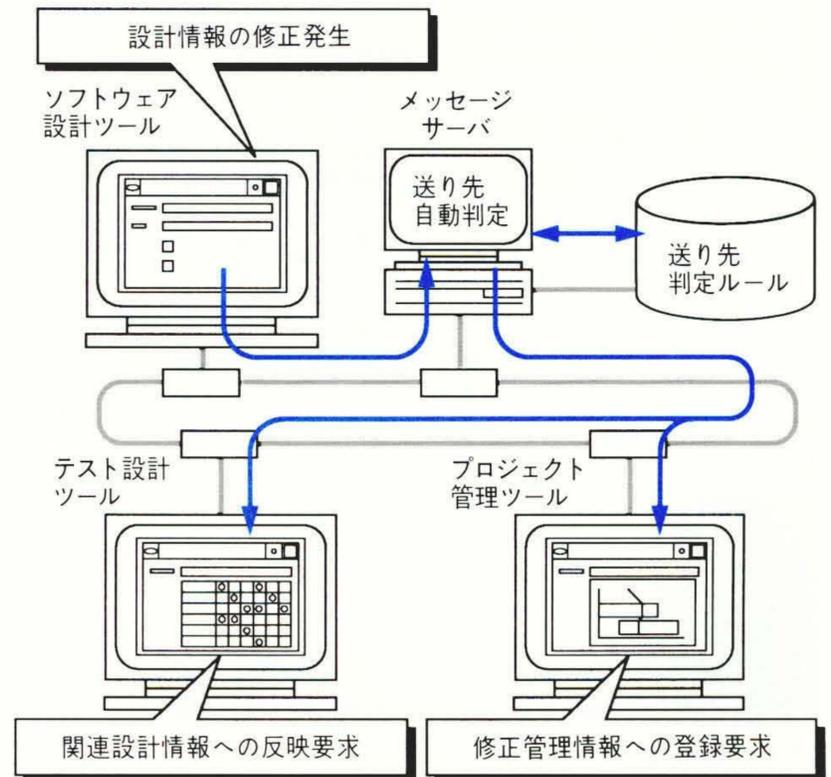


図8 コミュニケーションプラットフォームの働き  
ツール使用中に発生した修正情報は、ユーザーが指定しなくても必要とするツールに通知される。

ことができる。したがって、開発工程の遅れや品質上の問題発生といったトラブルを未然に防止することが可能となる。

#### 5 おわりに

ここでは、システムの分散化、開発の分散化に伴うソフトウェア開発・保守上の問題点に対し、その解決に有効な支援システム“Resolve-II”の概要について述べた。

ここで述べた支援システムは情報制御のさまざまな分野でのシステム開発に適用され、開発効率の向上と品質の確保に寄与してきた。

今後も、情報制御システムソフトウェアのいっそうの高品質化・生産性向上を目指し、ソフトウェア生産技術の開発を進めていく考えである。

#### 参考文献

- 1) 佐藤, 外: 制御と情報の統合を容易にする産業用情報制御計算機—HIDIC RS90シリーズ—, 日立評論, 77, 7, 465~468(平7-7)
- 2) T. Hayashi, et al.: A Lean and Visible Management Method and Its Support System for Software Testing Progress and Quality Control, First World Congress for Software Quality (1995-6)
- 3) 林, 外: 計算機制御システムのための高品質ソフトウェア開発方法, 電気学会論文誌C, 114巻, 6号, 645~653, (平6-6)
- 4) 高橋, 外: 情報制御システム向けソフトウェア開発・保守支援システム“Resolve-I”, 日立評論, 73, 8, 795~800(平3-8)