解　説

2. ユーザインタフェース管理システムの適用事例

2.1 ユーザインタフェース管理システムのCASE 環境への適用†

秋　口　忠　三††

1. はじめに

ソフトウェアの開発は，(1)ユーザ要求の分析，(2)ソフトウェアとハードウェアの機能分担の決定，(3)ソフトウェアで実現すべき機能仕様の定義，(4)実現方式の決定，(5)概要から詳細に至るソフトウェア構造の設計，(6)プログラミング，(7)テスト，というプロセスを踏む場合が多い．いずれのフェーズをみても知識集約的である．CASE (Computer Aided Software Engineering) システム[1]は，このようなソフトウェア開発という知的な活動を，ソフトウェア工学の種々の成果を取り入れ，計算機の支援の下に行わせることを目的としている．

CASE システムが特に強く要求されるのは，大規模で複雑なソフトウェアを開発する場合であり，それぞれのフェーズで，ユーザおよび種々の役割をもったソフトウェア技術者が数多く参加する．一般に言われる，要求分析者，設計者，プログラマ，といった役割分担は，その作業結果である要求仕様書，設計書，プログラムといった成果物との入出力関係で捉えられる．この成果物を紙という静的なメディアではなく，計算機という動的なメディアを介して受け渡しすることによって，計算機による積極的なソフトウェア開発の支援を可能にしようというのが，CASE システムのねらいである．

ソフトウェアという論理的構造物を組み上げる作業を計算機で支援するためには，その論理構造物の表現の枠組みを決め，その枠組みで個々の表現のインスタンスを計算機内部に保持し，それをソフトウェア開発者に理解しやすい形で提示できなければならない．これがソフトウェアの視覚化の問題である[2]．成果物を視覚的に計算機のディスプレイ上に表示し，それの編集・加工を計算機の支援の下に対話的に進めることが，CASE システムが提供する主要な機能の一つである．ソフトウェア開発者の知的活動を妨げない快適な操作性を提供することは，CASE システムが，単なる紙にかかれたデータを計算機に入力し綺麗なレポートを出力するためのデータ管理システムなのか，真にソフトウェア開発を支援する知的なシステムなのか，の分かれ目になると思われる（仮にソフトウェア成果物の管理システムであるという位置付けにしても快適な操作性を提供することが重要であることは言うまでもないが…）．ここにCASE システムにおけるユーザインタフェース (User Interface, UI) の役割の重要性が強く感じられるのである．

ソフトウェア開発の成果物を表現する枠組みとしては，それが論理的構造物であるということから，(1)一定の構文をもったテキスト，(2)表，(3)階層構造，(4)グラフ構造，の4種類が重要であると考えられる．現場のソフトウェア開発者が作成している種々のドキュメントを観察してみると，人間と人間の間の非形式的で抽象的な概念の伝達を目的にしたもの以外は上記4種類の視覚メディアで表現される場合が多いからである．したがって，CASE システムで必要なユーザインタフェースとしては，これら4種類のメディアの編集を効率良く行えるものであって欲しい．さらにこれらのメディアを複合して扱えるものであって欲しい．

世に数多く存在する CASE システムでは，このようなメディアの編集機能は目的に特化して専用に作られているものと考えられる．しかしソフトウェアの論理構造の表記法は数多く存在し，ソ

† Application of UIMS to CASE Environment by Chuzo AKIGUCHI (Software Engineering Development Laboratories, NEC Corporation).

†† 日本電気(株)ソフトウェア生産技術開発研究所

フトウェア開発部門のそれぞれの文化やノウハウを反映して種々のバリエーションがある．そのためにカスタマイズが容易でさらに CASE システム全体の一構成要素として利用できる編集のための道具が必要とされる．

このような観点から，主に CASE システムを指向して開発した UIMS (User Interface Management System) として鼎 (かなえ)* がある．鼎は上記4種類のメディアに加えてベクトル図形とビットマップイメージの合わせて6種類のメディアの編集機能を部品として提供している．さらにメディアを入れ子にするネスト機能と編集単位のデータを相互に関連付けるハイパリンク機能をもっている[3]．

著者は鼎を利用したいくつかの CASE システムの開発に直接あるいは間接に係ってきた．そこで本解説では，鼎を利用して開発された CASE システムの事例を中心に UIMS の CASE への応用例を紹介する．

2. CASE システムで要求される UI

2.1 基本的な対話機能

CASE システムにおいても，GUI (Graphical User Interface) をもった一般のアプリケーションと同様に，各種のボタンやメニュ，スクロールバー，文字列入力のためのエントリボックスなどの，基本的な対話機能 (Interaction Techniques) が必要である．

これらは，コマンドの発行，CASE システムの各種内部パラメータの表示と設定，復旧不可能なコマンドを実行する際の確認，操作誤りに対する警告の発行，などを行うために多用される．

2.2 各種論理メディアの編集機能

ソフトウェア開発のおのおののフェーズで，ソフトウェアの論理構造を表現するために，多種多様な図式が利用される．一般に広く知られているものとして，データフロー図，モジュール構成図，状態遷移図，ER ダイアグラム，デシジョンテーブル，構造化プログラミング用の各種図式などがある[4]．またプロジェクト管理の面から考えると，プロジェクト体制図，PERT 図などの工程管理図，作業内容を詳細化した作業構造図 (work breakdown structure, WBS) などがある[5]．

これらのおのおのの図式は用いる記号やその付帯属性において，さまざまなバリエーションが存在する．しかしながら，これらの図式の構造的な側面をみると，その大半がすでに述べた4種類の視覚メディア—テキスト，表，グラフ構造，階層構造—を特化したものであることが分かる．以上の観点から，CASE システムを実現するための UIMS の機能としては，これらの視覚メディアの構造的な特性を考慮した編集機能が重要であるといえる．さらに，種々の目的に応じた特化を可能にする拡張性も必要である．

2.3 柔軟な情報の編成と検索

ソフトウェアの仕様は，その特性に応じて種々の視点で捉える必要があるが，データの視点，処理の視点，制御の視点の三つの視点が重要であるといわれている[6]．また開発プロセスの観点からは，要求分析，設計，プログラミングの各工程間で，異なった詳細レベルのソフトウェアの仕様を(行きつ戻りつがあるにしろ) 段階的に決めていくことになる．さらにこのプロセスが複数人のプロジェクトの協調作業として進行するわけである．

このようにソフトウェア開発全体の構造は，複数の視点から捉えたソフトウェア仕様を，複数の役割の人が，複数の工程をへて編み上げていく過程とその結果から生ずるものである．ここに複雑に関連しあう情報の編成と管理の問題が出てくる．この問題を UI の観点から捉えると，CASE システムの UI として，柔軟な情報の関連付けとその関連をたどる機能が要求される．また関連情報を一覧できることも必要である．

柔軟な情報の編成と検索，さらに一覧性の実現のために，前述した論理構造を表現するメディアの利用に加えて，ハイパメディアの機能，すなわち複合したメディアを相互に関連付ける機構が有効であると思われる．

2.4 UI 以外の CASE 機能からの独立性

アプリケーションの中で，UI 部分と，データベース管理，通信制御，計算処理などの UI 以外の部分(ここではアプリケーション固有部と呼ぶ)との独立性をできるだけ高めるために，UIMS では種々の工夫がされている．UI はユーザ層や各人の好みによって変更要求が頻繁に発生するからである．一方快適な対話性を追及しようとする

* 鼎は日本電気(株)の登録商標である．

と，アプリケーション固有部のもつ機能やセマンティクスをユーザに分かりやすく提示したいために，両者は強い依存関係をもたざるを得ない傾向がある．ここに UIMS を実現する際の難しさがある．

CASE システムを対象とした UIMS としては，ソフトウェア開発の成果物の格納庫であるレポジトリとの独立性，CASE ツール間の通信制御部分との独立性，成果物の整合性チェックや変換を行う計算処理部との独立性を，高める工夫が必要である．

3. CASE 用 UIMS

本章では，CASE 用の UIMS として鼎の概要を紹介する．鼎はXウィンドウシステム* で GUI を実現する枠組みであるXツールキット上にC言語のライブラリとしてインプリメントされている．以下では，CASE システム用の UIMS として重要と思われる機能を中心に鼎の紹介をする．

3.1 6種類の視覚メディアの編集機能

鼎では通常の GUI 構築用ライブラリが提供するボタンやメニュのような基本的な対話機能に加えて，以下の6種類のメディアのエディタ機能を部品として提供している．

テキスト　日本語を含む一般の文章．任意の部分文字列に対してフォントの種類・サイズなどを設定できる．

図形　文字列，イメージ，矩形，円，線，ポリゴン，自由曲線などの図形要素の組合せで構成される図形．

イメージ　イメージスキャナなどで読み込んだビットマップのカラーイメージデータ．

表　スプレッドシートのように二次元的に配列されたセルの集まり．複数の属性をもつ構造体データの集まりを一覧するときに用いられる．

グラフ構造　ノードとアークから構成される論理的な図式．ノード間にはアークによる関係と包含関係を設定できる．部分と全体の関係や要素間の相互の関係を全体的に捉えることを可能にする．

階層構造（トリー構造）　文章の章立てやソフトウェアの機能階層などを表現するメディア．全体の構造から細部の構造まで任意の詳細化のレベルで構造を捉えることができるので，段階的詳細化のようにアイデアを整理するときの強力なツールになる．

以上のメディアのうち，CASE システムでは，テキストに加え，表，グラフ構造，階層構造の3種類の視覚メディアが重要である．これら3種類のメディアはいわば論理的視覚メディアと呼べるもので，ソフトウェアの論理構造を視覚的に表現すると同時に，その論理構造をエディタ内部で保持しているので，計算機による解析，一貫性のチェックなどが可能になる．

3.2 エディタのアーキテクチャ

鼎では，これらのメディアをXウィンドウシステムのマルチウィンドウ環境でマウスとキーボードを用いて編集するために，**図-1** に示すように MVC (Model-View-Controller) モデルをエディタ

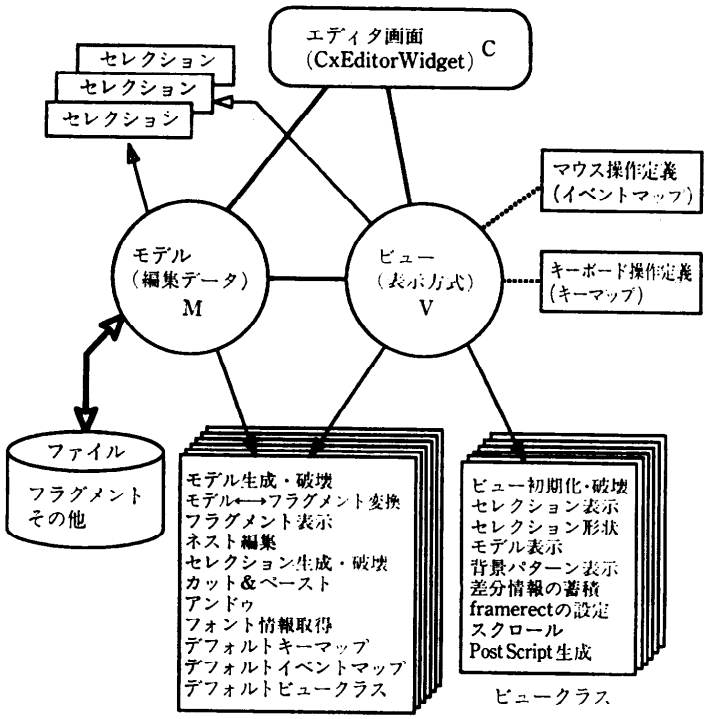

図-1　エディタのアーキテクチャ

* X-Window System は The Massachusetts Institute of Technology の商標である．

用に拡張した構成でエディタを実現している．ここでモデルは編集対象のデータの管理を，ビューは画面上へのデータの描画を，コントローラに当たる EditorWidget はマウスやキーボードのイベントの管理を，それぞれ司っている．一つのエディタは，EditorWidget，モデル，ビューの三つ組みによって実現できる．一つのモデルに対して複数の EditorWidget とビューの対を関連付けることによって，マルチビューのエディタを作ることができる．

鼎の中でメディアに依存しない部分をエディタ共通部と呼んでいる．エディタ共通部では，全てのメディアの編集に共通して必要とされる機能，すなわち，ウィンドウ上への描画・部分再表示，マルチビュー，拡大縮小，スクロール，マウスによる編集対象の操作，キーボードによるコマンド発行，ファイル入出力，PostScript☆ ファイルの生成，などを実現している．

各メディアに対応してメディアディスクリプタとビュークラスがあり，それぞれのメディアに依存する部分は全てここで実現されている．

3.3 カスタマイズ機能

エディタ機能を部品として提供し，目的に応じて特化して利用できるように，キーボードイベントとマウスイベントをカスタマイズする機能を提供している．キーボードイベントについては，キーマップを再定義することによって標準のキーバインディングを変更できる．マウスイベントについては，マウスボタンのプレス，リリース，ダブルクリック，マウスポインタの移動，ウィンドウへの出入りなど，全部で 10 種類のマウスイベントの解釈をイベントマップの再定義によって変更できる．

またエディタを構成するおのおのの要素の状態が変化するタイミングなどでアプリケーションプログラムに制御を渡せるように，約 40 種類の汎用コールバックと呼ぶフック関数を設定できるようにしている．たとえば，モデルの編集関数が実行される前と後，ビューとモデルが接続される直後と分離される直前，などである．

グラフ構造と階層構造はソフトウェアの構造を視覚化するために多用されるが，ノードとアークの形状は多種多様である．両メディアのエディタ部品を利用する場合，ノードとアークの形状の定義が大きな比重を占める．**鼎では，作図する図式**に対応してノードとアークの形状その他の属性を定義したものをスキーマと呼んでおり，このスキーマの定義を対話的に行うツールとして，スキーマ定義ツールを提供している．

3.4 ハイパメディアシステムの実現基盤

相互の関連を含めた情報を柔軟に表現し，情報間の論理関係を素早くアクセスしそれを即座に提示するために，従来のメディアを超えた情報の表現と活用を可能にした概念がハイパメディアである．定型化されていない情報の蓄積・管理・検索を行う際に有効であろう．CASE システムの種々の局面での応用が考えられる．

鼎でもハイパメディアシステムの実現の基盤として，メディアネスト機能とリンク機能を提供している．メディアネスト機能は，6 種類のメディアを相互にネストさせる機能であり，たとえば，表の一つのセルの中に，あるいは図形の一要素として，鼎の任意のメディアを張り込むことを可能にしている．

リンク機能は，モデルを構成する個々の編集対象（テキストの部分文字列やグラフ構造の個々のノード）に対して，任意の文字列をその属性種別を表すタグとともに設定し管理する機能である．リンクの表示，マウスによるピックアップなどが可能である．

4. UIMS を適用した CASE システム例

本章では，UIMS を CASE システムに適用した例として，鼎を利用して開発された 4 つの CASE システムを紹介する．次節でその適用効果と問題点を示す．

4.1 下流 CASE ツール：XSPD*C

XSPD*C は，整構造プログラムの表記法の一つである SPD[7] に基づき，プログラムの詳細設計を支援し，設計の結果から C 言語のソースプログラムと仕様書として SPD 図を生成する CASE ツールである（図-2 参照）．

エディタ機能としては，鼎の階層構造エディタを利用して段階的詳細化を支援しており，プログラムの全体の制御構造を見ながら，細部の設計を行えるようになっている．階層構造のノードには制御構造の種類を示す記号が付けられ，制御構造

☆ PostScript は Adobe Systems, Inc. の商標である．

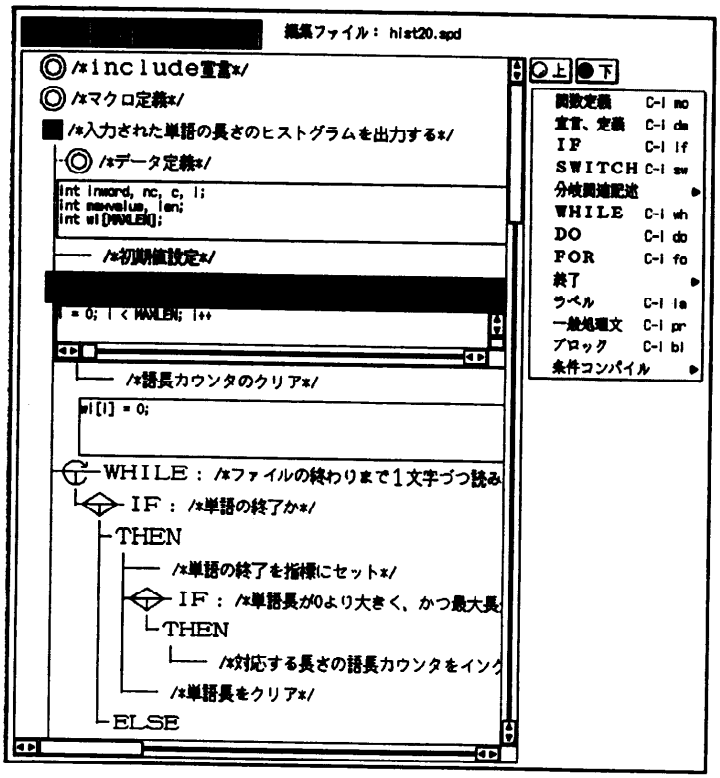

図-2 下流 CASE ツール：XSPD*C の実行時画面

を視覚的に捉えられるようになっている．各ノードにはその制御構造に応じて，条件文や制御変数などを設定できる．終端処理の説明は自然言語で行う．この部分にはテキストエディタ部品が用いられている．直接Cプログラムを書くこともできる．鼎では一つのノードに対して複数個のコンテンツ（内容物）をもてるので，文による説明とプログラムのソースコードを一つのノードに対して独立に管理できる．編集機能については鼎のエディタ部品をカスタマイズして利用することでほとんど無理なく実現できている．

設計情報は XSPD*C 独自のファイルフォーマットで保存される．印刷機能についても，画面上に表示される SPD 図を，独自のフォームシートに割り付けた設計書の作成が鼎の印刷機能を用いて実現されている．

4.2 上流 CASE システム SPECDESSIN/SA, FLOW

SPECDESSIN/SA と SPECDESSIN/FLOW はビジネスアプリケーションの開発の上流工程を支援する CASE システムである[8]．SA は構造化分析 (Structured Analysis) を，FLOW はジョブフローを中心とした概要設計を，それぞれ支援するもので，連携して利用される．どちらも，UI部は，テキスト，グラフ構造，階層構造，表の各エディタ部品を利用している（図-3 参照）．

SA は，機能階層図と，業務フロー図，機能情報関連図の各エディタ機能を中心に，詳細情報の設定を行うための数十のダイアログボックスが UI 部を形成している．FLOW は，ジョブフロー図，計算機処理フロー図，プログラム網図などの各フロー図の編集を中心にしたシステム概要設計の支援を行うが，ジョブフローのノードとしてプログラムパターンを用いた再利用ベースの設計を支援している点に特徴がある．再利用支援機能として，パターン仕様登録，パターン展開/組み込みなどの機能があり，グラフ構造と表を組み合わせたエディタを提供している．設計情報は階層的に管理されており，この部分の UI の実現には階層構造エディタ部品が利用されている．

ドキュメントとしては，各種一覧表，各種フロー図，機能情報関連図，プログラム仕様書などの生成機能が鼎の印刷機能を用いて実現されている．設計結果は SPECDESSIN 独自の形式でデータベース化されており，このデータベースから編集時に鼎のモデルを生成している．

グラフ構造上での各種の整合性チェックは，グラフ構造のモデルのデータ構造をアクセスする関数によって実現されている．

4.3 ソフトウェア工程管理システム：IX

IX は，ソフトウェア開発における計画策定や進捗管理を中心にシステム構成，作業項目，要員などの管理を総合的に支援するシステムである[9),10)]，各種管理情報を，グラフ構造，表形式，階層構造，線表など，その情報に適した表現形式で扱えるようになっている（図-4 参照）．

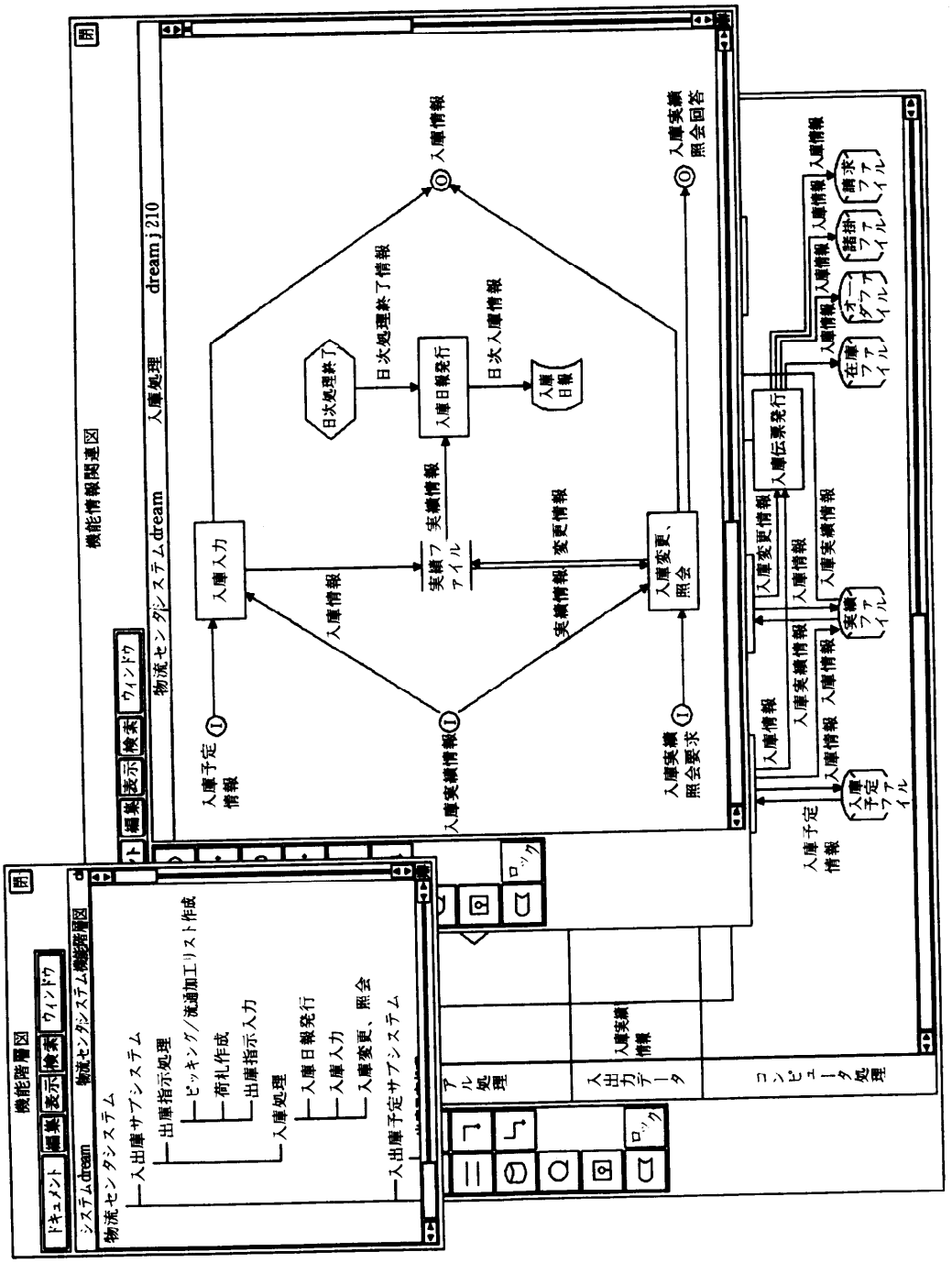

図-3 上流 CASE システム SPECDESSIN の実行時画面

主なツールとして，スケジュール管理，システム構成管理，作業項目管理，要員情報管理，進捗管理，カレンダ管理などを行うエディタが，鼎のテキスト，表，グラフ構造，階層構造の各エディタ部品を組み合わせて実現されている．スケジュール線表は表エディタとグラフ構造エディタを組み合わせ，両者を連動してスクロールできるようになっている．各ツールには詳細情報を設定するために，それぞれ十数個から数十個のダイアログボックスが用意されている．進捗状況を示す数値データを線グラフで表現した進捗グラフは図形エディタ部品を用いて実現されている．

管理情報は，SPECDESSIN の場合と同様，IX 独自の形式でデータベース化されており，この

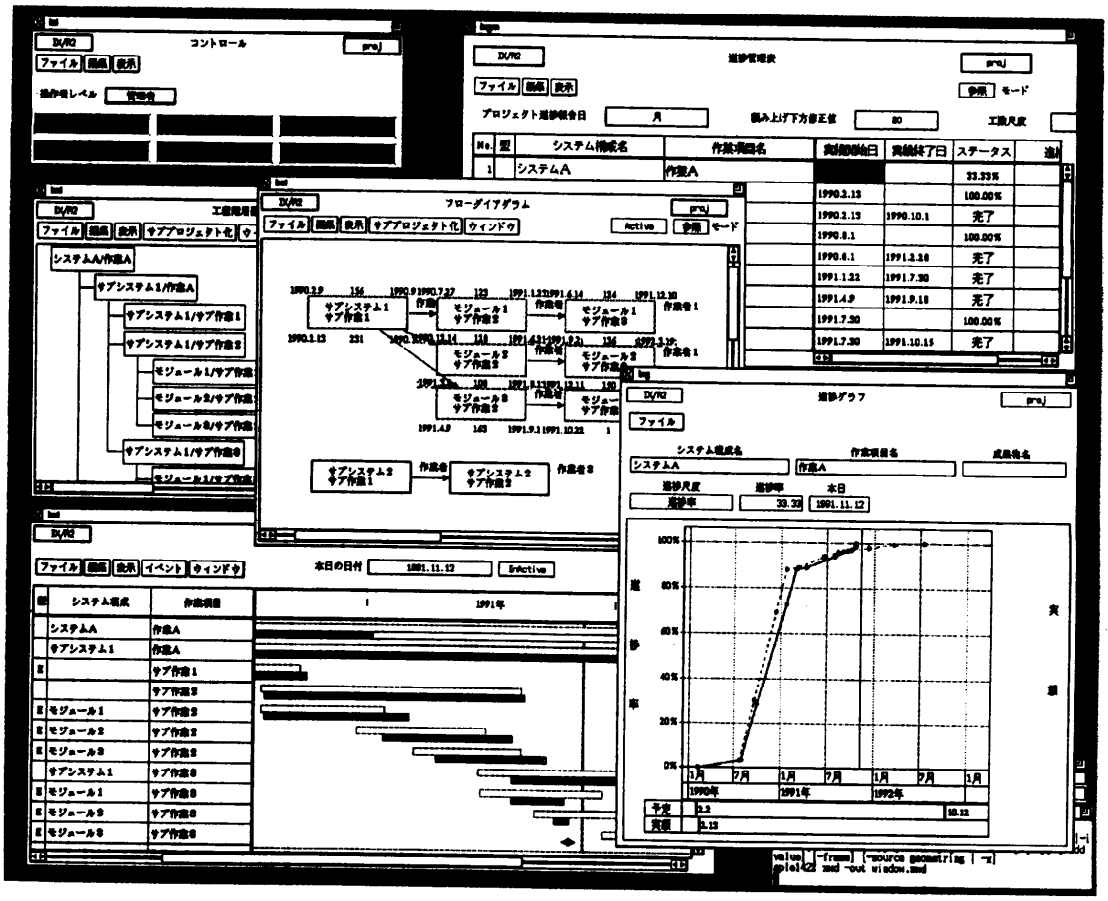

図-4 工程管理システムの実行時画面

データベースから編集時に鼎のモデルを生成している.

4.4 GUI 構築ツール：ゆず

「ゆず」は，鼎のアプリケーションプログラムの設計を対話的に行い，その設計結果からC言語プログラムを自動生成するツールである[11].「ゆず」自身 UIMS の一部と考えられるが，ここではUI 部を開発する CASE ツールという観点から，鼎の適用事例として「ゆず」を紹介する（**図-5** 参照）.

「ゆず」のUI部の中心はレイアウト機能である. GUI 画面を構成するボタンやメニュ，鼎のエディタ画面などを，マウスを用いて対話的に配置し，各種のダイアログボックスを用いて配色や対応させたい機能との関連付けを行う. レイアウトエディタは鼎のグラフ構造エディタ部品を用いて実現している. 画面に配置するボタンやメニュなどの部品の種類に対応してノードの型（ノードの形状と付帯属性）が定義されており，画面に配置される個々の部品のインスタンスの属性値はノードのインスタンスに保持される. 部品間の動的な関係，たとえば一つのボタンとそれを押したときにポップアップされるダイアログボックスとの関係など，の定義にはグラフ構造のアークが用いられている.

設計結果は，Lisp のS式の形式で保持される. なお「ゆず」の GUI 画面の大部分は「ゆず」自身を用いて開発された.

5. 適用効果と問題点

UIMS を用いて実際に開発・実用化された前述の CASE システムへの適用事例から，CASE 環境への UIMS の適用効果，現状の問題点および今後の改善課題を示す.

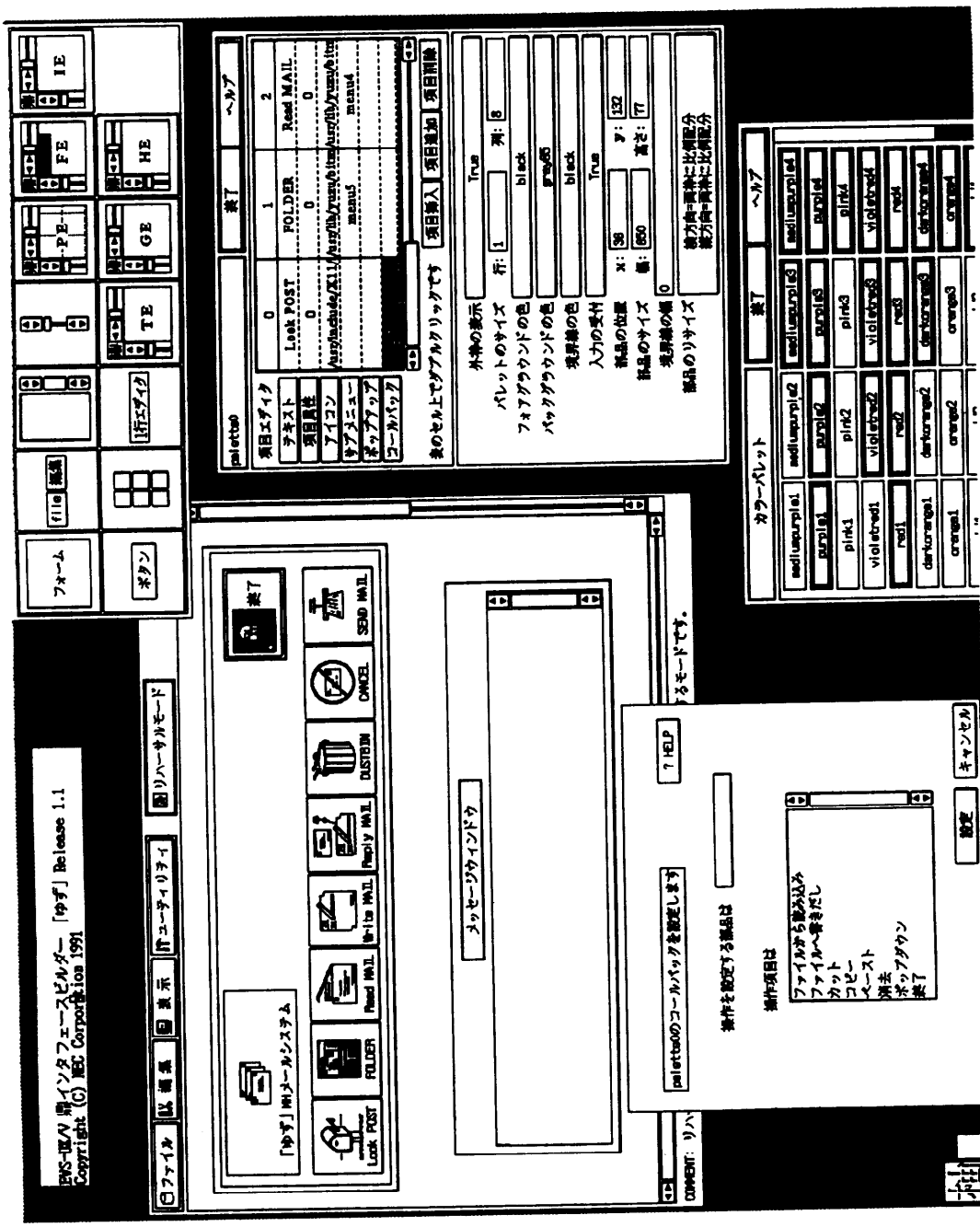

図-5 GUI 構築ツール「ゆず」の実行時画面

5.1 適用効果

鼎のエディタ機能の中でイベントの管理に関する部分はXツールキットの環境で動くウィジット(対話機能をもつ部品を意味するXツールキットの用語)として実現されている．一般にウィジットは GUI プログラミングの専門家が操作性と汎用性に十分な配慮をして開発するものである．GUI の対話処理の大部分はウィジットの中で実現されており，ウィジットを用いた GUI プログラムの開発は比較的容易である．複雑なデータ構造を扱うエディタの場合は，その対話処理全体の面倒をみるウィジットの効果はより顕著である．

ウィジットを用いた GUI プログラミングでは，対話部品やエディタ部品の画面上でのレイアウテ

ィングや配色などに，かなりの手間がかかっている．「ゆず」のような GUI 構築ツールを利用することによって，この部分の開発の手間は数分の1程度に削減できると思われる．

CASE 特有のエディタ機能の実現に関しては，(1)MVC の三つ組みを組み立てるだけで標準的なエディタを実現できること，(2)CASE システム独自の編集機能も CASE を指向して設計したエディタ部品（表，グラフ構造，階層構造の各メディアのモデルとビューを操作するプリミティブ関数）を利用して比較的簡単に実現できること，(3)これらとマウス操作やキーボード操作との結合もイベントマップとキーマップによって宣言的に行えること，などから，実現の手間を大きく軽減できたと思われる．

定量的な評価は難しいが，CASE システムの主要なエディタ機能が数千ライン程度で実現できており，さらに部品を組み合わせるような比較的単純なプログラム構造でできていることから，エディタ機能を独自に開発する場合と比較して，プログラムの開発規模にして 1/2〜1/3 程度の削減効果が，また開発するプログラムの複雑さにおいても同程度の低減効果があっただろうと思われる．プログラム規模の比較による定量的な評価については文献 3) を参考にされたい．

5.2 適用範囲

CASE システムで必要とされるエディタ機能は，テキスト，表，グラフ構造，階層構造の4種類のメディアでほぼ十分カバーできる．ただしIXでは統計的な数値データを視覚化するために図形エディタの描画機能も利用された．興味深い利用例として階層構造と表を横に並べ連動してスクロールするエディタも作られている．

各種ドキュメントの出力は CASE システムの主要な機能のひとつである．これらのドキュメントは，エディタで表示される内容をページ単位に分割し，日付，作成者，その他関連情報とともに，一定のフォーマットに割り付けて作成される場合が多い．このようなドキュメント作成において鼎エディタ部品の印刷機能である PostScript 生成機能が利用されている．フォーマットは個別に対応する図形データをプログラムで作成しているが，フォーマット生成を対話的に行えるツールが望まれるところである．

5.3 成果物の管理

鼎のエディタ部品の中には編集の結果をファイルに保存する機能があるが，事例で紹介した CASE システムではこの機能はほとんど利用されていない．独自のデータベースからエディタが扱える形式のデータを一時的に作成し，編集が終了したらそれをデータベースに書きもどしている．成果物をエディタの編集単位で管理するのではなく，相互に関係をもつ統合された情報のデータベースとして管理し，幾つかの視点から同じ情報を異なるメディアで提示したいからである．また編集対象の個々の要素に対して，各 CASE システム独自のデータを設定し保持する必要があったことも理由のひとつとしてあげられる．

CASE システムが扱う情報をデータベースとして管理する場合には，同時に複数のエディタで同じ情報が編集されることを防ぐための排他制御の問題や，あるエディタで編集している情報を別のエディタで参照するようなエディタ間の同期制御の問題の解決が必要になる．現状では，各 CASE システムにおいてそれぞれ独自の方式によって実現されている．CASE システムのように人間の編集作業が介在する長時間のトランザクションにおけるデータの共有は，データベース機能とプロセス間通信機能に主に依存するものであるが，UIMS 側からのアプローチも必要と思われる．

5.4 ハイパメディア機能の利用

現在までのところ鼎のハイパメディア機能を CASE システムへ適用した例はない．適用の時点でまだハイパメディア機能が利用できなかったこともあるが，それ以上にも，5.3 で述べたように，成果物の相互の関係も含めて独自のデータベースで管理する必要があったためと考えられる．

しかしソフトウェアの開発では，分析，設計，プログラミングの各フェーズで最終結果だけではなく，その結果に至る検討や議論の経緯を最終結果の補助的な資料として残しておくことは後日貴重な情報となることが多い．このような非形式的で非定型の情報の管理にはハイパメディア機能が有効に利用できると考えられる．

また，プログラムとドキュメントの一体化をねらった文芸的プログラミング (literate Programming) 環境を実現するためにも，ハイパメディア機能は有用であろうと思われる[12]．

6. おわりに

マウスやキーボードなどの入力装置からのイベントの制御とそれに対する実時間の視覚的なフィードバックを司る機構が UIMS の骨格である．この骨格の上に基本的な対話を実現する手段としてボタンやメニュのような対話用の部品が作られる．さらに部品を組み合わせて目的にかなうユーザインタフェースを実現するためにインタフェースビルダと呼ばれるツールが提供される．UIMS は，これら骨格，部品，組立ツールが一体となって発展するものであろう．

本稿で紹介した鼎は，ユーザインタフェースの実現において最も困難である編集機能をエディタ部品として提供している．当初より，CASE システムへの適用を第一の目的に開発したものである．その適用効果については，4 種類の CASE システムへの適用事例を紹介し，CASE システム特有の編集機能を自由に組み込めるエディタ部品が，高い有効性をもつことを示した．CASE システム以外にも，各種メディアの編集機能が必要とされる適用領域は数多く存在し，そこでのエディタ部品の有効性も確認されている．

複数のエディタ機能を利用してひとつの複合した成果物を複数の人が協調作業で作成していく CASE システムにおいては，エディタ機能，データベース機能，エディタ機能間の通信機能が密に関係をもってくる．それぞれに独立性をもたせ，なおかつ密な関係を実現できるようにするには，それぞれの機能間インタフェースをどのように設定すればよいか，非常に重要で困難な課題である．

参考文献

1) 原田 実監修：CASE のすべて，オーム社(1991).
2) Chang, Shi-Kuo : Visual Languages : A Tutorial and Survey, IEEE Software, Vol. 4, No. 1, pp. 29-39 (1987).
3) 暦本，垂水，菅井他：エディタを部品としたユーザインタフェース構築基盤：情報処理，Vol. 31, No. 5 (1990).
4) Martin, J. and Carma, M. : Diagraming Techniques for Analysts and Programmers, Prentice-Hall, Inc., Englewood Cliffs (1985). [ソフトウェア構造化技法―ダイアグラム法による，國友義久・渡辺純一訳，近代科学社 (1986).]
5) W.H. ローツハイム著，深沢士郎訳：システム・インテグレーションのための構造化プロジェクト管理，近代科学社 (1992).
6) Hatley, D. J. and Imtiaz, A. P. : Strategies for Real-Time System Specification, Dorset House Publishing Co., Inc. (1988). [リアルタイム・システムの構造化分析，立田種宏監訳，日経 BP 社 (1989).]
7) 遠藤裕香著，甲斐義久監修：構造化プログラム設計図法 SPD―わかりやすいプログラムへの招待状―，共立出版 (1992).
8) 納富，大島，柿沢，三輪，牛島：ソフトウェア開発支援，NEC 技法，Vol. 43, No. 7, pp. 40-48 (1990).
9) 原田，安部，寺門，岩崎，東出：ソフトウェア工程管理システムIX(1)，情報処理学会第43回全国大会講演論文集(5)，pp. 347-348 (1991).
10) 安部，原田，寺門，岩崎，東出：ソフトウェア工程管理システムIX(2)，情報処理学会第43回全国大会講演論文集(5)，pp. 349-350 (1991).
11) 杉山他：鼎(かなえ)インタフェースビルダ「ゆず」の構築，情報処理学会第45回全国大会講演論文集(5)，pp. 99-100 (1992).
12) Smetinger, J. and Gustav, P. : A Hypertext System for Literate C++ Programming, Journal of Object- Oriented Programming, Vol. 4, No. 8, pp. 24-29 (1992).

(平成4年7月16日受付)

秋口 忠三（正会員）

1954 年生．1977 年山梨大学工学部電子工学科卒業．1979 年同大学院修士課程修了．1982 年静岡大学電子科学研究科博士課程満期退学．静岡大学工学博士．同年，日本電気(株)入社．CASE システム，GUI 構築環境関連の研究開発に従事．1985 年より 1 年間カリフォルニア大学サンフランシスコ校客員研究員．現在，同社ソフトウェア生産技術開発研究所技術課長．電子情報通信学会，日本ソフトウェア科学会各会員．