

# “SEWB”の開発思想と機能

## Concept and Functions of “SEWB”

ソフトウェア開発の生産性向上、信頼性向上を目的として各種の図式設計技法が開発され定着化してきた。ソフトウェア開発の一連の工程を、これらの図式設計技法を用いたビジュアル化ソフトウェア開発支援ツール群で一貫支援することを目的に、SEWBを開発した。SEWBの開発では、対話管理、ライブラリ管理を統括した「SEWB基本部」を核とする「核(カーネル)化構造形統合方式」によって、ツール間のデータの授受、操作の統一を実現した。また、コミュニケーションモデルに基づき人とコンピュータとの対話方式を標準化し、利用者にとって快適なマンマシンインタフェースを確立した。SEWBはホストコンピュータとも連携し、ソフトウェアの開発に一貫して利用できる統合開発環境を構築している。

葉木洋一\* *Yōichi Hagi*  
 末宗英利\*\* *Hidetoshi Suemune*  
 前澤裕行\*\*\* *Hiroyuki Maezawa*  
 印東 功\*\*\*\* *Isao Indō*  
 関沢 智\*\* *Satoshi Sekizawa*

### 1 緒 言

企業の意思決定などの戦略的業務でのコンピュータの活用や、分散処理ネットワークの普及に見られるように、コンピュータシステムは高度化し、また地域的な広がりを持つようになった。ソフトウェアに対する重要性はますます高まり、より早く、より高品質なソフトウェアを開発するための種々の開発技法が開発されてきた。その中でも図や表を利用したビジュアル化表現が、ソフトウェアをより人間に理解しやすくするものとして定着化してきた。図や表での対話によるプログラムの開発ができる環境が整えば、大多数を占める初級、中級のプログラムの理解を助け、生産性の大幅な向上を図ることが期待できる。

一方、ハードウェアの発達とともに図形表現を容易にする高精細ビットマップディスプレイ、高性能、大容量メモリ、マルチウィンドウ機能などを持ったワークステーションが登場し、図式設計技法のツール化が可能となっている。

日立製作所では大形オペレーティングシステムVOS3 (Virtual-storage Operating System 3)の会話処理で稼動するEAGLE2 (Effective Approach to Achieving High Level Software Productivity 2)を既に開発し、ソフトウェア開発の効率を上げてきた。今回、EAGLE2を相補うものとして、高性能、多機能のワークステーション2050シリーズを利用した新たなソフトウェア分散開発環境SEWB (Software Engineering Workbench)を開発し実用化した。

SEWBは「効率のよいソフトウェア開発を実現するために、工程別(設計、製造、テスト、保守)の支援ツールをワークステーション上に有機的に組み合わせた統合システム」である。

主な特長は以下のとおりである。

#### (1) 図形によるソフトウェア開発の推進

構造化技法に基づくソフトウェア開発を支援するものとして、業務処理仕様図SDF (Structured Data Flow Diagram)、問題解析図PAD (Problem Analysis Diagram)などの図式がある。これらの図形情報をコンピュータで手軽に扱えるようにして、より高品質なソフトウェア開発のための図式設計技法を強力に支援する。

#### (2) 分散形ソフトウェア開発環境

大形コンピュータを共同利用したときの運用上の制約やホストコンピュータの負荷による応答性のばらつきを解消するため、また開発者の好みに応じたフレキシブルな対話インタフェースの実現を目的に、ワークステーションを活用した分散形ソフトウェア開発環境を構築する。

#### (3) 一貫した操作性を実現した開発環境

従来のソフトウェア開発の各工程ごとの開発支援ツールの品ぞろえだけでなく、それらを有機的に接続した統合化された環境を提供する。

本稿では、SEWBの開発思想を概説するとともに、その代表的な機能について紹介する。

## 2 開発思想

### 2.1 開発の方針

SEWBの開発上の基本方針は、コンピュータ処理に都合のよい従来のコンピュータ寄りのマンマシンインタフェースではなく、人間の思考を助けるための、利用する人の立場に立

\* 日立製作所ソフトウェア工場 \*\* 日立ソフトウェアエンジニアリング株式会社 \*\*\* 日立製作所システム開発研究所  
 \*\*\*\* 日立製作所大森ソフトウェア工場

ったマンマシンインタフェースを確立することである。

このために、以下に示す具体的な方針を設定し実現を図った。

(1) 「習→慣」ユーザーインタフェースの確立

個々のツールごとに異なる操作方法でなく、SEWB全体を通して均一な操作方法に統一する。これは、今後SEWBを構成するツールが増えても、同一操作方法で統一していける論理性を持ったものとする。すなわち、一つのツールの操作をマスターすれば、その結果が他のツールにもそのまま応用できる、操作の類推が可能なユーザーインタフェースを確立する。

(2) 支援ツール群の「有機的結合」

支援ツールを組み合わせて利用したとき、個々の効果は相乗的に増加する。このために、ソフトウェア情報の一元管理を行い、一元管理したソフトウェア情報を基盤として、支援ツール群を有機的に結合する。

(3) 「性能重視」の設計

利用者が快適に使用できる開発環境とするためには、優れたマンマシンインタフェースとともに、迅速な応答を満足するものでなければならない。一般に利用者は応答が3秒を超えると「遅い」と感じると言われている。SEWBでは使用頻度の多い操作に対しては、3秒以内の応答性能を確保した設計とする。

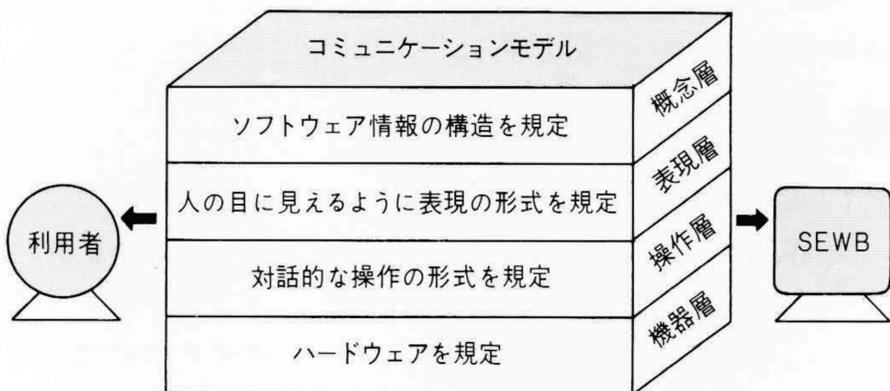
2.2 採用した方式

これらの方針を実現するため、各支援ツールに共通な機能を抽出し、各支援ツール共通の「核(カーネル)」となる「SEWB基本部」を開発した。また対話方式の設計では、人とコンピュータとのコミュニケーションモデル(図1)に基づき、それぞれでの規定を標準化することで使いやすい対話方式を実現した。

なお、「SEWB基本部」を核とする統合方式は、文献ではシームレスシステム(Seamless Systems)と呼ばれている<sup>2)</sup>。

(1) 核(カーネル)化構造形統合方式

従来のツールごとに異なる対話方式や関連ツール間だけでソフトウェア情報の管理を行う粗い結合方式では、ソフトウェア開発を一貫した操作で統一するのは困難である。このため、SEWB基本部を核とする核化構造形統合方式を採用し、



注：略語説明 SEWB(Software Engineering Workbench)

図1 コミュニケーションモデル 人とコンピュータとのコミュニケーションを四つの層に明確に分類し、それぞれの層での規定を標準化することで、利用者の立場に立った使いやすい対話方式を実現した。

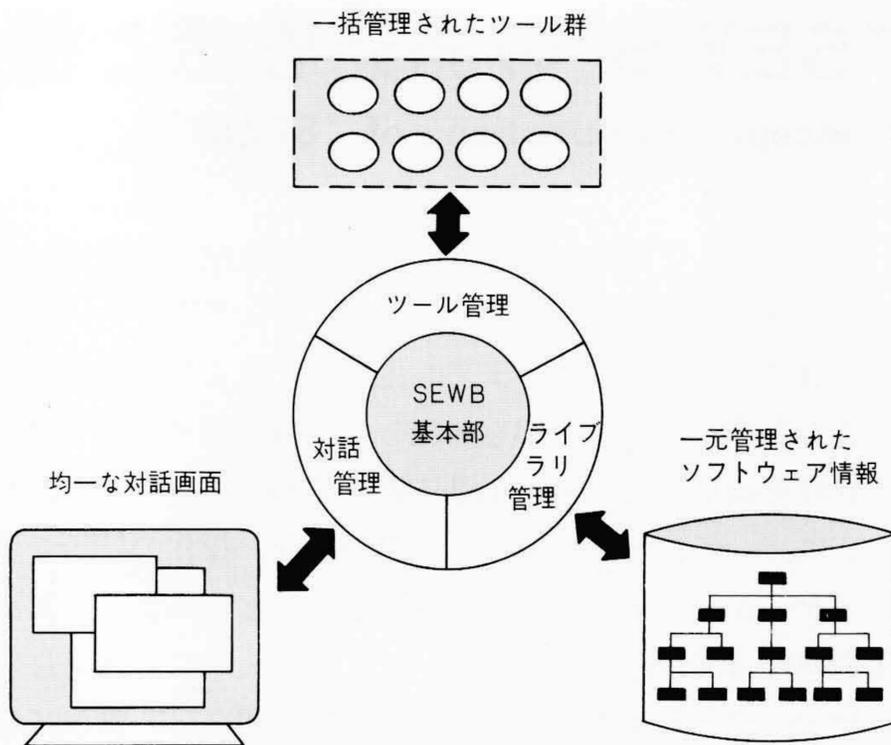


図2 核(カーネル)化構造形統合方式 ツール管理によって支援ツール群を一括管理するとともに、対話管理やライブラリ管理を支援ツールから分離して、SEWB全体の基盤となるSEWB基本部に統合した。

ツール管理により支援ツール群を一括管理するとともに、対話管理やライブラリ管理を支援ツールから分離してSEWB全体で統合する密結合方式とした(図2)。

なおSEWBでは、SEWB基本部を使用したユーザー独自ツールやソフトハウスの製品を容易に開発できる仕掛けも提供している。

(2) ソフトウェア情報の一元管理(概念層の標準化)

仕様情報やプログラムコードなどのソフトウェア情報を、最小の単位である「シート」として管理し、それらをまとめて管理できるように「グループ」と呼ぶ入れものを設け階層化した。これによって、ソフトウェア情報全体を木構造の階層で一元管理できる。シートやグループは画面上にアイコンとして表示し(図3)、各支援ツールの起動は、この一元管理されたソフトウェア情報の中で目的の「もの」を選び、コマンド(ポップアップコマンド)を選択することで自動的に行われる。

(3) SEWB画面の統一(表現層の標準化)

各支援ツールの画面をSEWB基本画面(図4)として統一し、以下に述べる項目の標準化を行った。

(a) 画面構成

ガイダンス領域、シンボル領域、編集領域などの配置。

(b) 共通的な画面操作方法

(i) マウスカーソル種別

「もの」をつかむときの道具(マウス)を示すもので、全体をつかむ「図+文」、一部の文字列だけをつかむ「文字」など4種類を用意した。

(ii) 拡大、縮小

表示内容の大きさを変えるレンズに該当し、最小、標準、最大の3段階を用意した。

(c) 画面の色調

日立製作所デザイン研究所の指導により、編集領域はグ

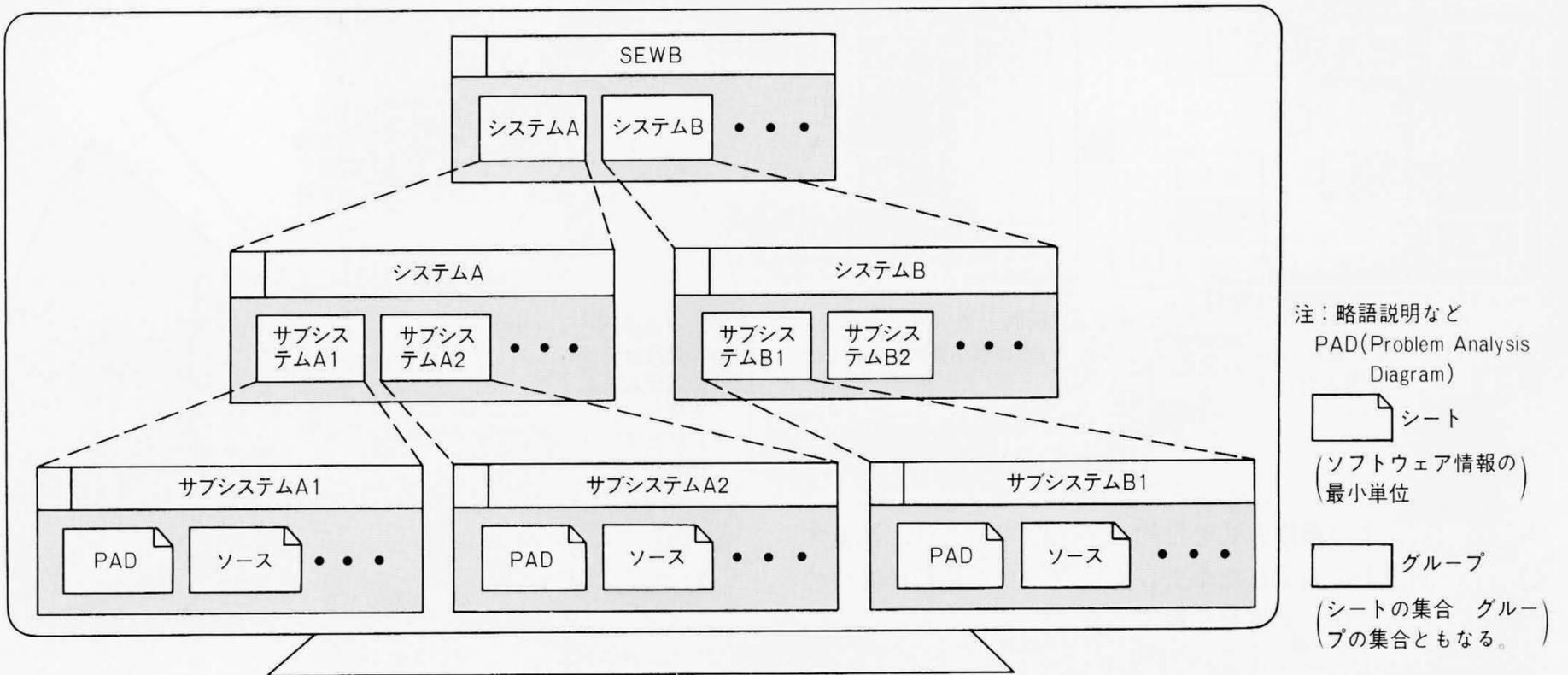


図3 ソフトウェア情報の一元管理 ソフトウェア情報を木構造の階層で一元管理する。“PAD”シートを選び、「開く」コマンドを選ぶと、当該PADを編集するPADエディタが起動され、編集可能なPAD画面が表示される。

レーを基調とし、画面上部のガイダンス領域やシンボル領域は明るいオレンジ系統にまとめ、人間工学に基づいた人が疲れにくく、認識しやすい色調とした。

(4) オブジェクト指向形直接操作方式(操作層の標準化)

(a) オブジェクト指向形

操作対象を「目的語(オブジェクト)」, 操作命令を「述語(コマンド)」に分解して、すべての操作命令を「オブジェクト+コマンド」の文法とし、「何をどうする」という人間の思考に合わせたものとした(図5)。

(b) 直接操作方式

オブジェクトやコマンドなど、すべての選択物を画面上に表示しておき、それらを直接マウスで選ぶだけで操作を完了できるようにした。キーボードの使用は文章の入力など最小限の場合に限定している。

(5) 高機能ワークステーションの活用(機器層の標準化)

ワークステーションで用意されているマルチウインドウ, マルチプロセスの機能を活用し、支援ツールはそれぞれ独立のプロセス(タスク)として動作させた。これは、自由度のあ

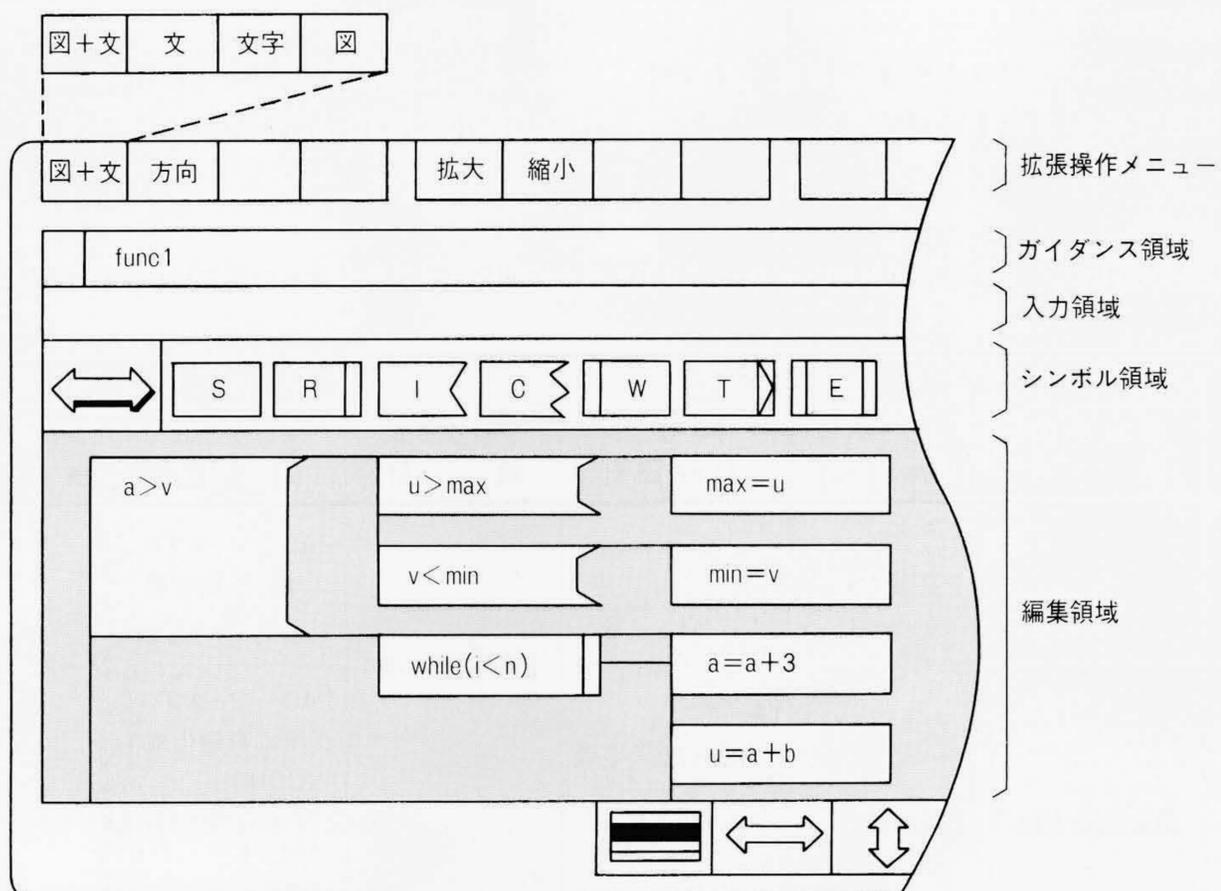


図4 SEWB基本画面形式 シンボル領域, 編集領域の内容だけがツールごとに異なるだけで、画面レイアウトをツール全体で統一している。

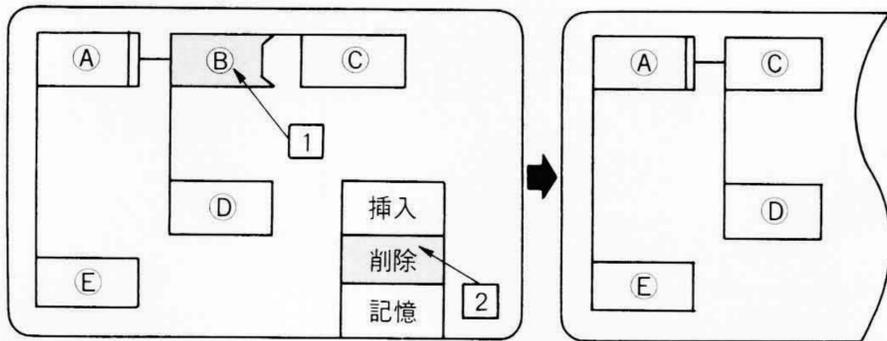


図5 オブジェクト指向形直接操作方式 画面上で「もの」(オブジェクト)を選ぶだけで簡単に操作できる。例えば、Bのボックスを削除したいときは1,2を選ぶだけでBが削除できる。

る使い勝手のよい操作方式を目的としたものであり、関連する情報を画面上に表示したまま各支援ツールの並行作業ができ、またソフトウェア情報を検索する場合にも、表示しているグループの任意のグループから自由に作業を開始できる。

(6) 差分表示方式による表示の高速化

性能に対しては、内部処理の高速化はもとより画面への表示をいかに速くするかがキーポイントとなる。SEWB基本部と各支援ツール間では論理的なインタフェースで表示データをやりとりする方式を採用しているが、全面的な書き直しだけでは性能を劣化させるだけでなく不要な「ちらつき」を発生させてしまう。このため、変化した部分だけのデータによる差分表示方式を開発し、書き直しの範囲を局所化するようにした。

3 機能概要

SEWBにはソフトウェア開発のライフサイクルと対応させ、図6に示す4種類のSEWBを検討しており、今回、このうち、システム設計者用とプログラマ用の二つのSEWBを製品化した。

3.1 ソフトウェア開発の流れ

SEWBを構成する各ツールが、ソフトウェア開発ライフサイクルのどの工程に位置するかを表1に示す。

表1 支援ツールと工程との対応 SEWBの各支援ツールは、ソフトウェア開発ライフサイクルの各工程に対応して図形による表現で一貫支援している。

対象ソフトウェア	システム設計			プログラム開発			
	業務処理設計	バッチ処理設計	オンライン処理設計	プログラム設計	プログラム自動生成	プログラム編集	テスト
ワークステーション、マイクロコンピュータ用ソフトウェア	該当せず			PADエディタ	PAD↔ソースプログラム自動生成(C)	ソースエディタ	PADテスト(C)
ホストコンピュータ用ソフトウェア	業務設計支援	システムフロー設計支援	画面・帳票定義		PAD↔ソースプログラム自動生成(COBOL, FORTRAN, PL/I)		COBOLソース実行テスト
			画面遷移シミュレータ	デシジョンテーブル→ソース自動生成(COBOL)			

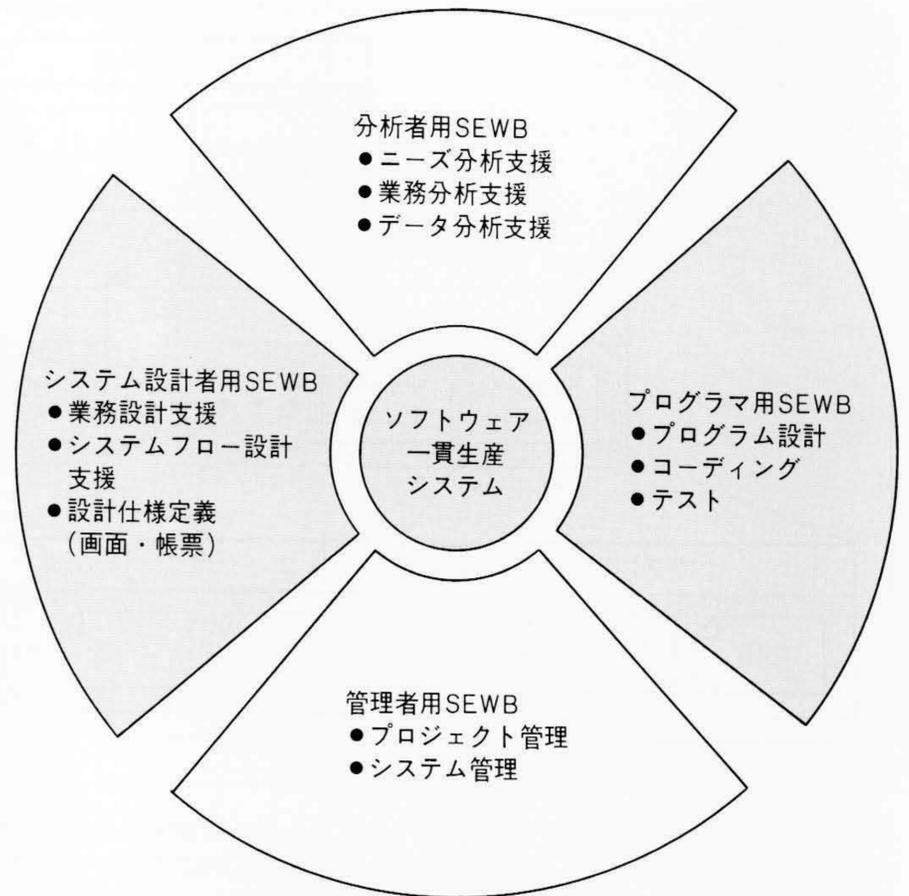


図6 SEWBのサポート範囲 4種類のSEWBのうち、システム設計者用SEWBとプログラマ用SEWBを製品化した。

SEWBはホストコンピュータ用ソフトウェアの開発支援以外に、ワークステーション、マイクロコンピュータ用ソフトウェアの開発支援も用意し、様々なターゲットマシンを対象としている。

各工程に位置するツールは、ソフトウェアを人間に理解しやすく、またビジュアルにする「図形」による表現で一貫支援している。

3.2 システム設計者用SEWB

(1) 業務設計支援

システム設計技法の一つにSDFと呼ぶ業務処理仕様図がある。SDFとはTom DeMarcoらが提案した構造化分析(SA:

Structured Analysis)<sup>3)</sup>に基づき、データの流りに着目して業務処理仕様を図形で記述する技法で、次のような特長がある。

- (a) 正確なモデルが記述できる。
- (b) トップダウンによる階層ごとの分析ができる。
- (c) 「もの」(データ、ファイル)と「できごと」(処理)に明確に区分したモデル表現となっている。

SEWBでは、このSDFの作成、編集を支援し、アイコン、マウスなどを利用して簡単に描画することができる。

SDFの編集例を図7に示す。

(2) システムフロー設計支援

バッチ処理設計を支援する図式に、コンピュータの処理フローを示すシステムフローがある。システムフローは、対象業務のプログラム間の関連、プログラムとファイルとの関連、及び処理手順を表したものである。SDFで記述される最下位レベルの1個の機械化処理対象ボックスがシステムフローに展開される。

システムフローの編集では、各種ボックスの追加や削除に対応してシステムフロー全体をバランスよく自動再配置する機能を持ち、変更の手間を省けるようにしている。また、ホスト側でSDFを元に自動生成したシステムフローをワークステーションへ転送し、それを編集することもできる。

(3) 画面・帳票定義

オンラインシステムの端末用画面、帳票のレイアウトやフィールドの属性などを、対話形式で定義する。定義された画面や帳票はテスト表示ができ、その場でエンドユーザーと仕様の確認が行える。

従来からEAGLE2でオンライン処理設計を支援していたが、今回、ホストコンピュータの負荷分散を目的にワークステー

ション側にも本機能を設けた。

編集方法は、まずラフスケッチを行い、その結果から各フィールドの細部の情報を入力する。最後に、定義結果をテスト表示する3段階で構成し(図8)、画面・帳票イメージから具体的なレイアウトの確認まで人間の思考に合わせた方式とするとともに、マウスなどワークステーションの機能を活用し、使いやすくしている。

(4) 画面遷移シミュレータ

画面・帳票定義で作成した画面・帳票をユーザーの設定した遷移図に従って遷移させ、オンラインシステムの実際の動きをワークステーション上でシミュレートすることができる。

すなわち、まず画面遷移図を作成し、続いて一つの画面・帳票から別の画面・帳票へ遷移する条件を、画面遷移図の作成過程で自動的に生成される遷移条件表の条件フィールドに記述する。条件は、=, !, =, <, >, &, |などの記号を使用し容易に定義できる(図9)。

オンラインシステムの動きをシミュレートするこのプロトタイプを使って、エンドユーザーの合意に基づいた高品質設計が実現できる。また、画面操作の事前学習といった効果をもたらす。

3.3 プログラム用SEWB

(1) PADエディタ

PADは構造化プログラム技法に最適な図式であり、2次元のプログラム構造を複雑な文章形式から単純化した図式として表現できる。SEWBでは接続、選択、反復の三つの基本ボックスを含め、図10に示す10種類のボックスをシンボルとして使用できる。

PADエディタは以下に述べる特長を持ち、PADの作成、変

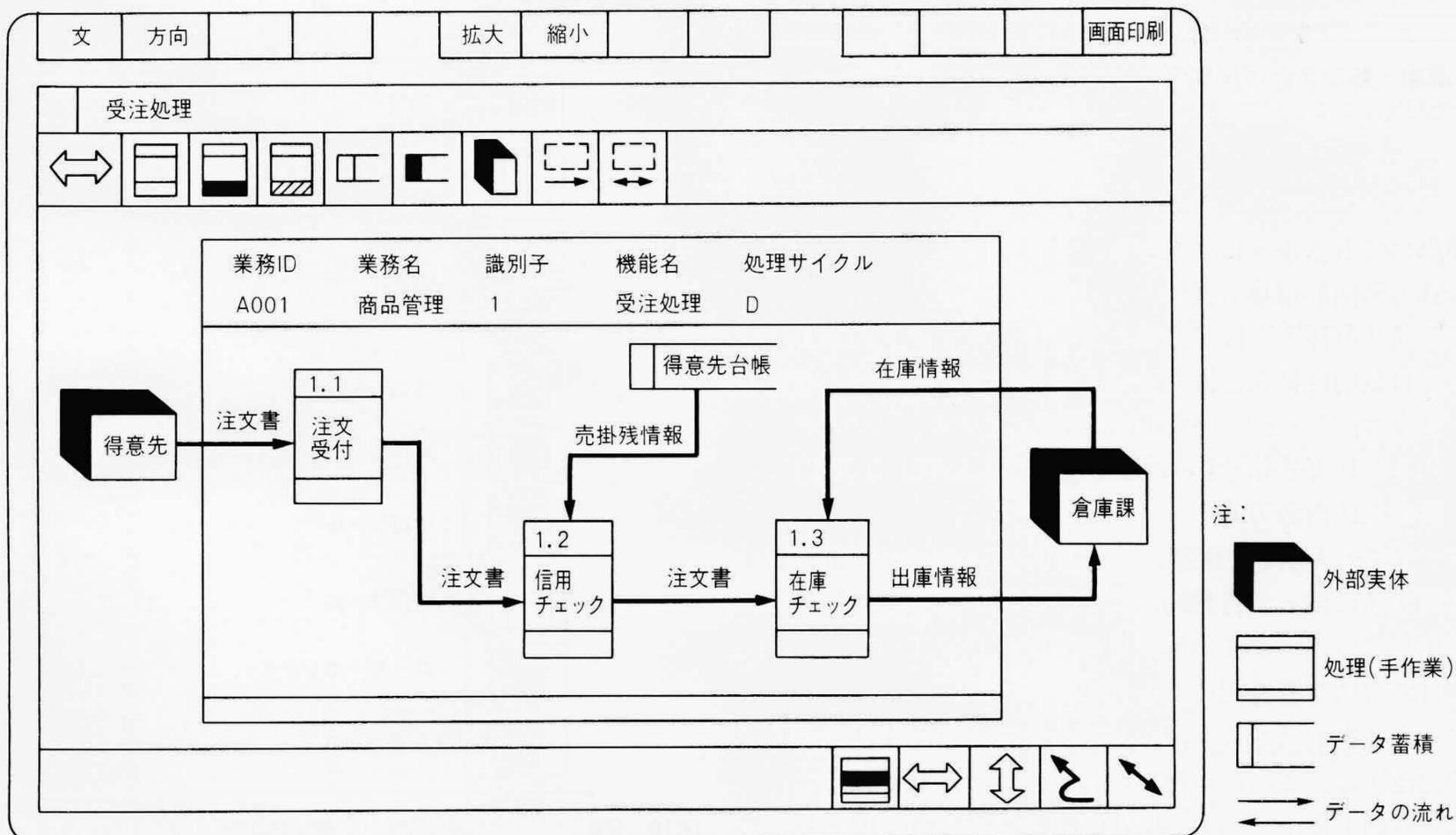
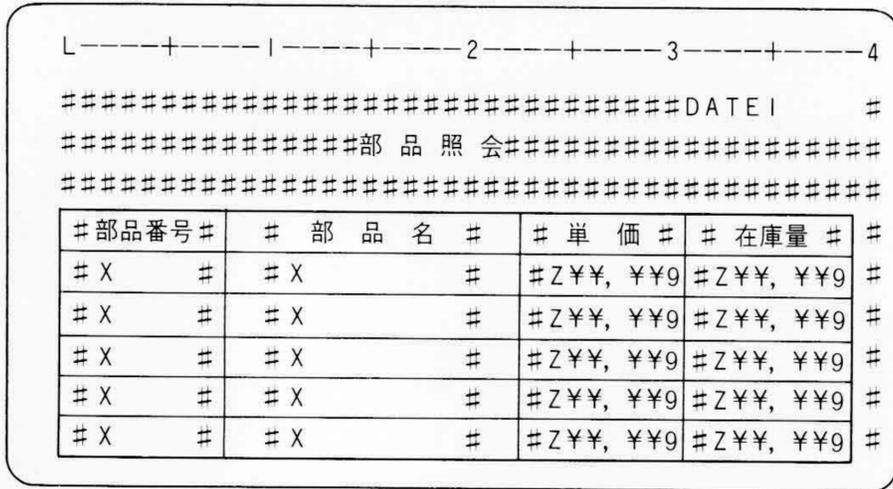


図7 SDFの編集例 図面上のシンボルを選び、ポップアップコマンド(挿入、記憶、置換など)を選ぶだけで簡単に編集できる。

レイアウト定義



フィールド定義

	位置		長	反復		間隔		IO・D変	レベル	タイプ	色	印字	転送
	行・ID	列		縦	横	縦	横						
001	1	32	8					R			G	Y	
002	2	16	8					C			R	Y	
003	4A	3	6					C			G	Y	
004	4B	12	10					C			G	Y	
005	4C	24	6					C			G	Y	
006	4D	32	6					C			G	Y	

テスト表示

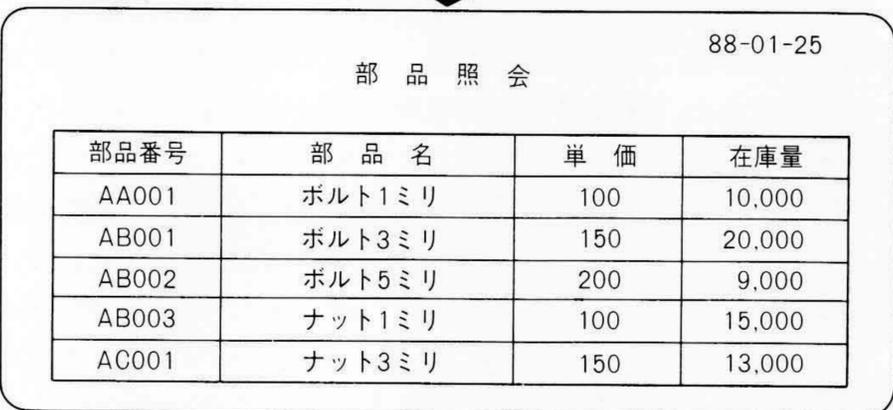


図8 画面・帳票定義の編集例 定義情報はホストに転送して、ホスト側で利用することができる。

更が簡単にできるようにしている。

(a) 誘導形図形編集方式<sup>4)</sup>

ボックスの挿入では、直前に挿入したボックスの形状に応じて自動的に挿入位置を決定し誘導していく。挿入位置は、誘導ボックスとして画面上に明示している。このため、ボックスの形状を指定するだけで連続挿入が行える(図11)。

(b) 図面文法内蔵方式<sup>5)</sup>

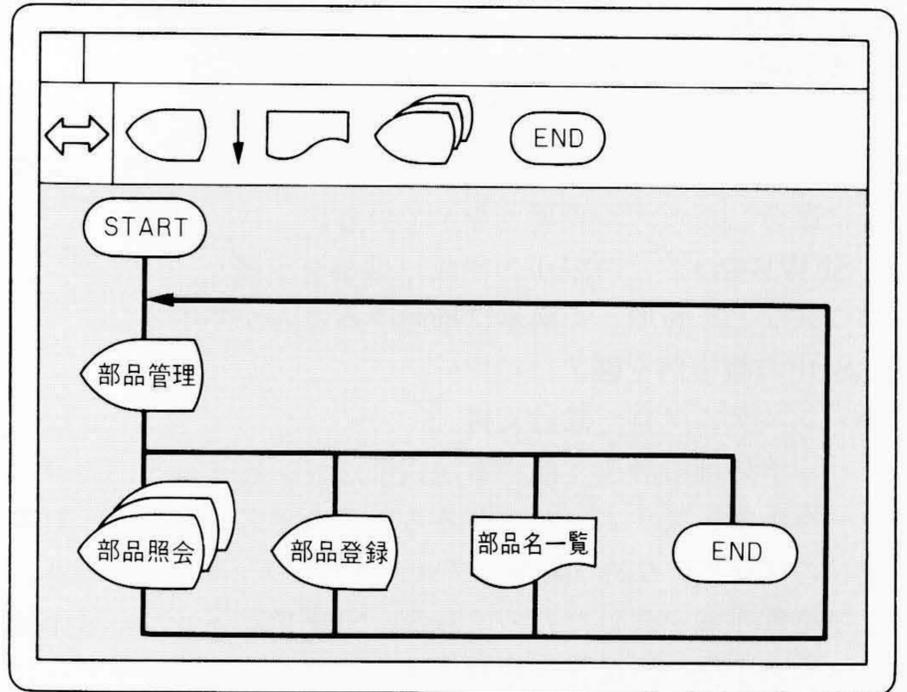
ボックスの大きさ、配置、結合をPADエディタ内部に持つ図面文法に従って自動的に行う。このため、編集操作では編集したいボックスの形状を考えるだけでよい。

(c) 目的言語を直接記述する方式

ボックス内のテキストは、目的の言語の言語文法そのもので記述する。このため新しい言語を覚える必要はなく、従来の言語を使用したままPADによるプログラム開発技法を適用できる。

画面遷移図の作成

画面・帳票の遷移を作成



遷移条件の定義

遷移条件を完成させる。

SEQ	FROM	TO	遷移条件	TIME
1	START	部品管理	PF1~PF24で起動します。	
2	部品管理	END	PF12	
3	部品管理	部品名一覧	CODE=3 & NO=SPACE	
4	部品管理	部品登録	CODE=2 & NO=SPACE	
5	部品管理	部品照会	CODE=1	
6	部品名一覧	部品管理	印刷	
7	部品登録	部品管理	PF1	
8	部品照会	部品照会	NEXT, BACK	2
9	部品照会	部品管理	PF1	

図9 画面遷移シミュレータ 遷移条件表のFROM, TOの内容は、画面遷移図から自動的に作成されるので、遷移条件だけを定義すればよい。

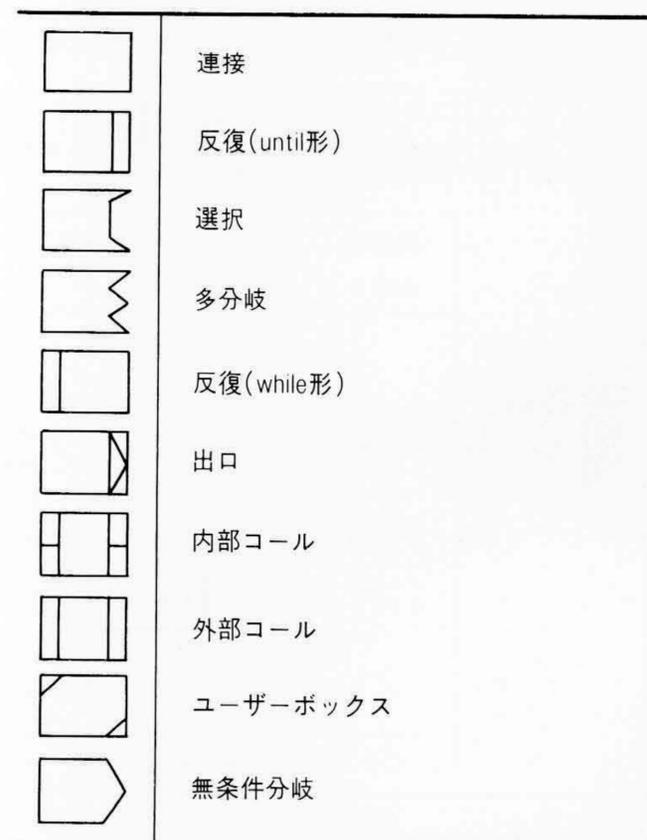
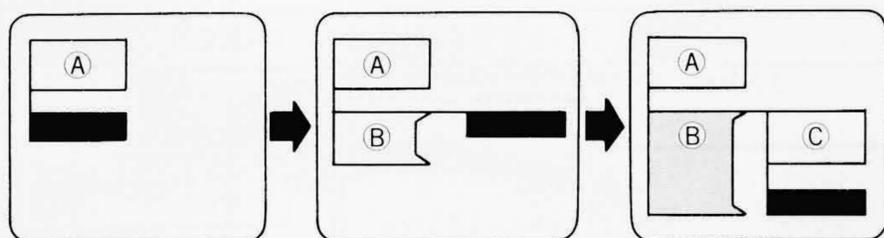


図10 PADのボックス形状 SEWBでは、PADを作成するのに10種類のボックスを使用できる。



接続ボックス(A)の下に誘動ボックスが明示されている。  
 選択ボックス(B)を作成すると、次に挿入すべき位置が誘動ボックスとして明示される。  
 更に接続ボックス(C)を作成すると、選択ボックス(B)が自動的に大きくなる。

図11 PADの編集例 直前に挿入したボックスの形状から、次のボックスの挿入位置を誘導し、また影響する他のボックスの大きさや配置を自動的に変更する。

なお、SEWBではユーザー出口付きPADエディタも用意しており、ボックスごとのシンタクスチェックや簡略語による入力などの機能を容易に実現できる。

(2) ソースエディタ

ホストコンピュータ上で動作するASPEN (Advanced Editor System for Programming Environment)と親和性を持たせたフルスクリーンエディタFSED/WS (Full Screen Editor/Work Station)を開発した。FSED/WSは、2050シリーズの特長であるアイコンやマルチウィンドウなどを利用して操作性を格段に向上させている。

(3) PAD↔ソース 自動生成

PADからソースを自動生成したり、また逆に、ソースからPADを自動生成する。サポートしている言語は、C、COBOL、FORTRAN、PL/Iの4種類である(なお、FORTRAN、PL/IのソースからPADへの自動生成は現在開発中である)。

(4) デシジョンテーブル→COBOL 自動生成

デシジョンテーブルを画面上で編集し、完成したデシジョンテーブルからCOBOLソースを自動生成する。

(5) C PADテスト

PADによるテストを支援するもので、実行したボックスの色を刻々と変えたり、変数の内容を実行に合わせて同時に表示する。すなわち、PAD上でアニメーションしてテスト状況を表示していく。テスト対象の関数の一覧表示、変数の一覧や内容の表示、実行速度の段階的調整、1ボックスごとに中断する逐次実行、ボックスごとの実行回数の表示、テストカバレッジ情報の表示など、豊富な機能を持ち、テストを分かりやすくビジュアル化させる効果がある。図12にC PADテストの実行例を示す。

(6) COBOL PADテスト

機能的にはC PADテストと同一である。C PADテストの場合、ワークステーション内で独立に動作するが、COBOL PADテストの場合、ホストコンピュータと連動して被テストプログラムはホスト側で動作する。すなわち、ホスト側でテスト実行させ、ワークステーション側ではその実行プロセス、及び結果をビジュアル表示する役割を担っている。

(7) COBOL ソース実行テスト

COBOLソースプログラムを解釈し、ワークステーション単独でテストできる。また、ソースプログラムをフルスクリーン

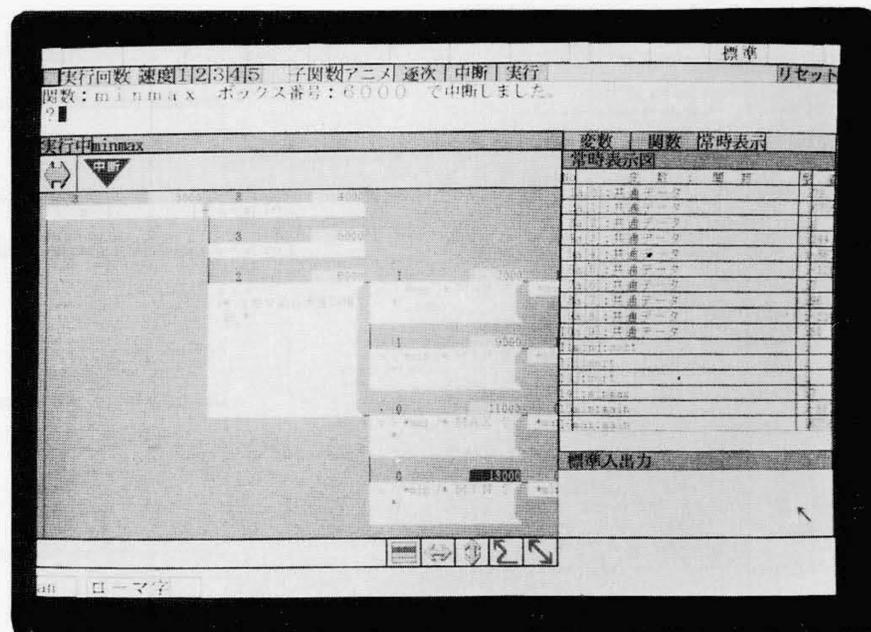


図12 C PADテストの実行例 C PADテストの画面は、モニターウィンドウ、PADウィンドウ、変数・関数ウィンドウ、標準入出力の四つのウィンドウで同時表示を行う。

で表示し、途中でエディタモードに切り替えることもできる。

4 AI応用によるソフトウェアの再利用

SEWBは部品の再利用を強力に支援するEAGLE2と連携し、部品化による生産性、信頼性の向上を図っている。図13は、ホスト事務処理システム(COBOL)をVOS3、EAGLE2と連携して開発した作業の流れを示したものである。

既存ソフトウェアを最大限に再利用するためには、より上流工程の仕様書のレベルでの再利用が必要である。日立製作所ではICAS-REUSE<sup>6)</sup> (Integrated Computer Aided Software Engineering System-Reusable Software Engineering)と呼ぶソフトウェア再利用方式の基盤技術を確立している。ICAS-REUSEは、要求文解読のための日本語処理機能、要求分の欠落情報を補足するための業務上の概念依存関係(知識)を用いた推論機能、類似仕様書の推論検索機能などを備えたものである。

5 SEWBによる効果

SEWBによるシステム開発、保守の効果を表2に示す。

次に、SEWBを適用した場合の定性的な効果を述べる。

(1) ソフトウェアのビジュアル化

SDF、システムフローによる設計仕様のビジュアル化、PAD、デシジョンテーブルによるプログラムロジックの図形表現を促進し、ソフトウェアを理解しやすくする。

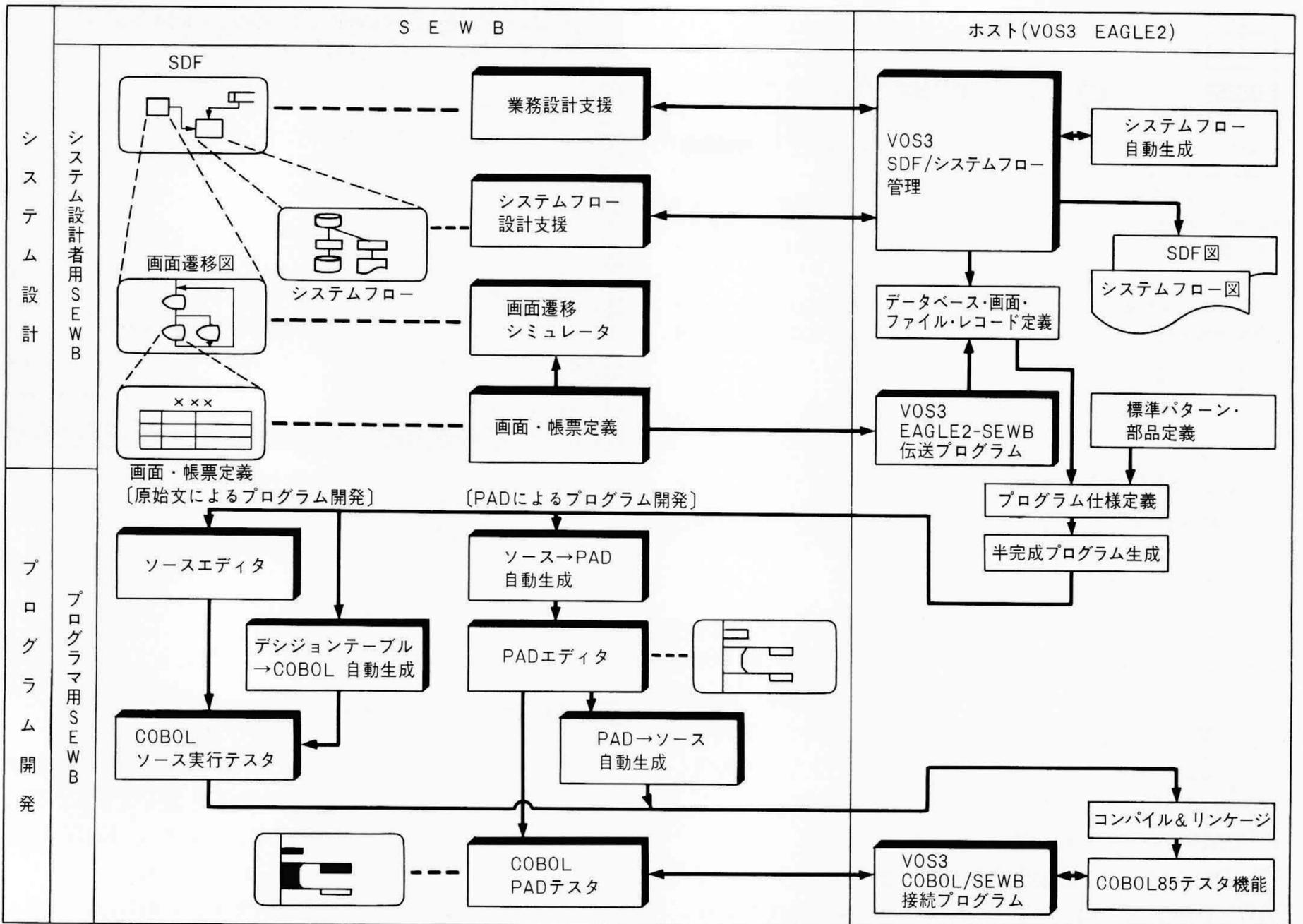
(2) プロトタイピングの実現

画面遷移シミュレータによって、オンラインシステムの稼働イメージを設計工程でエンドユーザーと画面を見ながら打ち合わせることができ、要求仕様の手戻りを大幅に減らすことができる。

(3) システム設計要員の強化

操作性、応答性の良い一貫化された開発支援ツール群により、経験の少ない人でもシステム設計が容易に行える。

(4) ホストコンピュータの負荷軽減



注：略語説明など VOS3(Virtual-storage Operating System 3), EAGLE2(Effective Approach to Achieving High Level Software Productivity)

SEWB関連ツール, VOS3 EAGLE2 既存支援ツール  
 → データの流れ, ---- 対応する画面例

図13 SEWBとEAGLE2を使ったシステム開発の流れ(COBOL) SEWBはホストDOS3 EAGLE2と連携して、システム開発を一貫支援している。

表2 SEWBによる効果 SEWBを適用することにより1.5倍から3倍の生産性向上が図れる。

分類	ユーザー	効果
生産性	EAGLEユーザー	ソフトウェア開発のライフサイクルを通してEAGLE適用時の1.5倍
	EAGLE未導入ユーザー (COBOL, PL/I)	ソフトウェア開発のライフサイクルを通して2~3倍(EAGLEとともに導入すると4~5倍)
	FORTTRANユーザー	プログラム設計からプログラム作成まで2~3倍
	C言語ユーザー	

システムの開発作業をワークステーション主体に行えるため、ホストコンピュータの負荷を軽減できる。

## 6 結 言

SEWBはソフトウェアを開発するための分散形一貫支援システムを目的とするもので、ワークステーションの機能を最大限に活用したマンマシンインタフェースを確立し、またホストコンピュータとの機能配分に留意して開発した。

今後、SEWBの適用結果のフィードバック、LAN(Local Area Network)を利用した水平分散形開発環境の実現、分析者や管理者用のSEWBの開発など、ソフトウェア開発に一貫して利用できる統合開発環境として、より使いやすく拡張していく考えである。

## 参考文献

- 1) 葉木, 外: システム開発支援ソフトウェア“EAGLE”-EAGLE拡張版“EAGLE2”-, 日立評論, 68, 5, 373~378(昭61-5)
- 2) Mark Dowson: Integrated Project Support with Istar, IEEE Software, 6~15(Nov.1987)
- 3) T.DeMarco(黒田, 高梨訳): 構造化分析とシステム仕様, 1986, 日経マグローヒル社(昭61-12)
- 4) 前澤, 外: 図面を用いたプログラム開発方式と支援ツール“SDL/PAD”, 日立評論, 68, 5, 357~360(昭61-5)
- 5) 前澤, 外: ソフトウェア開発指向のワークステーション, 日立評論, 67, 3, 235~238(昭60-3)
- 6) 千吉良, 外: システム仕様書の再利用によるソフトウェアの開発技法(ICAS-REUSE), 日立評論, 69, 3, 249~254(昭62-3)