U.D.C. 681.32.06:〔681.32.004.14:658.512.2.011.56〕

# システム開発支援ソフトウェア“EAGLE”

# Integrated Software Development and Maintenance System “EAGLE”

葉木洋一* *Yôichi Hagi*
今城哲二* *Tetsuji Imajô*
津田道夫* *Michio Tsuda*
仁平博三* *Hirozô Nihira*

業務システムの大規模化・複雑化とエンドユーザー部門からのシステム化要求の急増は，EDP部門の開発能力を上回り，バックログの発生と累積をもたらしている。

このバックログ解消の かぎ を握るのが，EDP部門の開発能力の向上である。日立製作所はこのような問題に対処するため，HIPACEの標準手順・技法と一体となるシステム開発支援ソフトウェアEAGLEを開発・製品化した。

EAGLEは，対話処理によるシステム開発，データ辞書によるソフトウェア資源の一元管理，パターン・部品によるプログラムの生成，保守ドキュメントの自動作成の四つの特長をもち，バッチ処理からオンライン データベース処理までの業務システムの開発を支援する。

## 1 緒　　言

コンピュータシステムが企業活動の中枢となった現在，いかに効率的に業務システムを開発し維持するかが重要な課題となっている。しかし，業務システムの大規模化・複雑化とエンドユーザー部門からのシステム化要求の急増(開発需要は年20～30%の増)は，**図1**に示すようにEDP(Electronic Data Processing)部門の開発能力(要員は年ゼロ～数パーセントの増)を上回り，バックログ(開発待ち業務)の発生と累積をもたらしている。

このバックログ解消の方策として，(1)エンドユーザー自身によるシステム開発，(2)アプリケーションパッケージの購入及び(3)EDP部門の開発力向上の三つが考えられる。(1)は最も望ましい姿であるが，現状はエンドユーザー自身がシステム開発できる環境が整備されているとは言い難い。(2)は，国内のアプリケーションパッケージマーケットがようやく成長し始めた段階であり，まだ現実的ではない。すなわち，(1)と(2)とだけに頼るわけにはいかないので，1980年代のバックログ解消のためには，(3)のEDP部門の開発力をいかに向上させるかが重要である。

日立製作所は，EDP部門のシステム開発の効率向上を目的として，要求分析からプログラムの開発・保守までの作業手順を標準化し，技法とともにHIPACE(Hitachi Phased Approach for High Productive Computer System's Engineering)として体系化し，提供してきた[1)]。

従来からHIPACEでは，個々の支援ソフトウェアを提供してきたが，今回，よりいっそう体系化し，システム開発を支援するソフトウェアとしてEAGLE(Effective Approach to Achieving High Level Software Productivity)を製品化した。すなわちEAGLEは，**図2**に示すようにHIPACEの3本柱(標準手順，技法，支援ソフトウェア)の一翼を担うものである。

本報では，EAGLEの開発背景，ねらい及び特長について概説するとともに，その代表的な機能を紹介する。

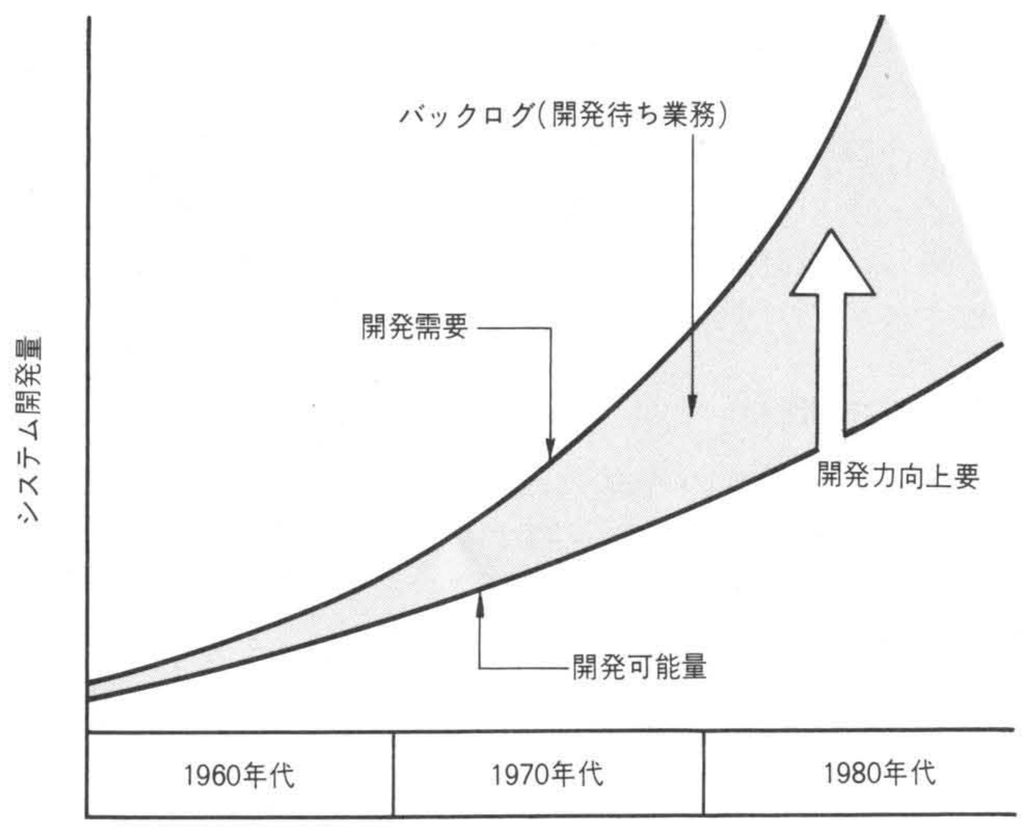
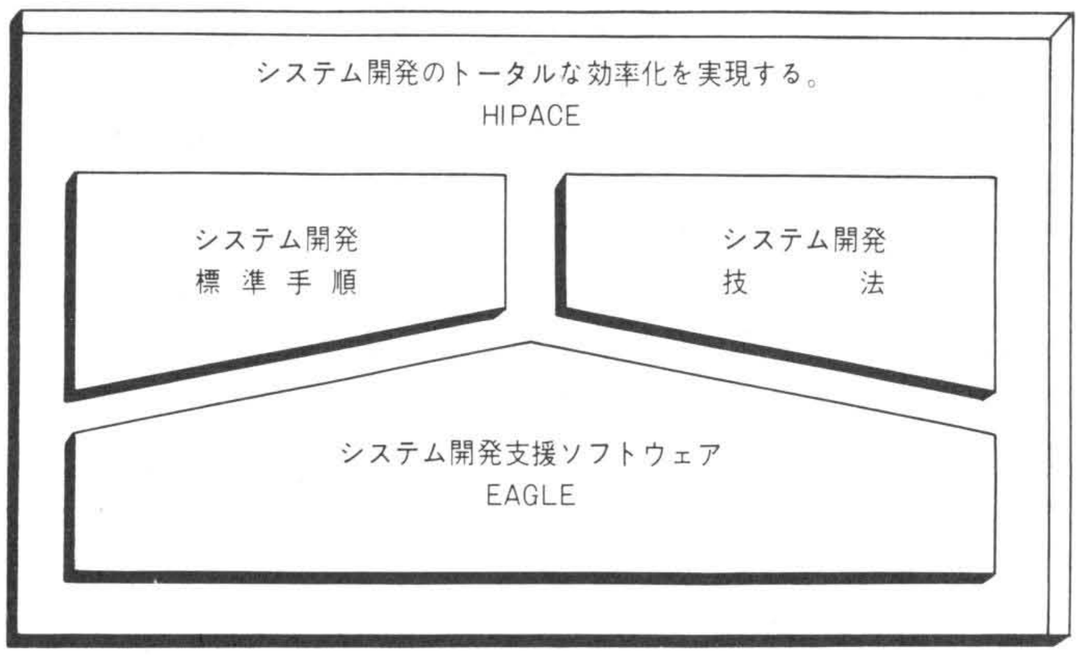



図1　システム開発量の推移とバックログ　　システム開発の需要と供給の差は広がる一方である。EDP部門の開発力の抜本的な向上が急務である。




注：略語説明
HIPACE(Hitachi Phased Approach for High Productive Computer System's Engineering)
EAGLE(Effective Approach to Achieving High Level Software Productivity)

図2　HIPACEの中のEAGLEの位置づけ　　EAGLEは，HIPACEの標準手順と技法と一体となったシステム開発支援ソフトウェアである。

* 日立製作所ソフトウェア工場

## 2 EAGLEの概要

### 2.1 既存の開発支援システムのノウハウを活用

EAGLEの開発に当たり，既存の総合的な開発支援システムの長所を積極的に取り入れることに努めた。特に参考にしたのは，利用実績が多く客観的に評価可能な次の三つであり，漢字CORAL（Kanji Customer Oriented Application Program Development System)からは対話処理と資源の一元管理を，CANDO（Computer Aided New Facilities for system Development and fixation to the Organization)からは標準パターンによるプログラム生成機能を，CASD(Computer Aided Software Development System)からはテスト支援機能を取り入れた。

(1) プログラム開発維持支援システム：漢字CORAL[2)]

オンラインデータベースシステムの業務プログラムの生産性向上を目的とする。日本語の仕様書言語，対話処理によるシステム開発，データベースを用いたソフトウェア資源の一元管理に特長がある。

(2) 構造化ソフト開発支援システム：CANDO[3)]

ファコム・ハイタック株式会社ハイタック本部の社内ツールである。HIPACEの考え方に基づき，システム設計，プログラム作成，テスト，ドキュメント出力の各支援機能をもつ。標準パターンによるプログラム生成及び単体テスト用パラメータの自動生成に特長がある。

(3) ソフトウェア開発支援システム：CASD[4)]

日立製作所の社内ツールである。オペレーティングシステムやコンパイラなどの基本ソフトウェアの開発用であり，設計，プログラミング及びテストの支援サブシステムで構成される。ソフトウェア製品の信頼性を高めることに重点が置かれている。

### 2.2 EAGLEの開発方針

EAGLEの最終目標は，EDP部門のシステム開発・保守の生産性と信頼性の向上である。この目標達成のため，次に述べるような開発方針を設けた。

(1) ソフトウェアの開発は必ず何らかの作業標準を前提にしている。EAGLEは，HIPACEのシステム開発標準手順SPDS(Standard Procedure to Develop System)に従って作業することを前提とする。すなわち，SPDSのマニュアル[5)]で定められたワークシートに必要事項を記入することにより作業を進め，このワークシートの記入結果をEAGLEの支援機能を用いてコンピュータに蓄積し，多方面に有効利用を図る。

(2) システム開発の全工程を対象とする総合的な支援ソフトウェアとする。また，この工程の中での一貫性を保証する考え方として，データ資源管理の概念を採用する。データ資源管理とは，データを，人，物，金などと同様に企業の目的達成のための貴重な資源とみなして管理することである[6),7)]。この管理の中核としてデータ辞書(データディクショナリ)を位置づける。

(3) しかし，システム開発の一部の工程に対して標準手順をもち，支援ソフトウェアを運用しているユーザーも少なくない。これらの多くは前記(1)，(2)の標準手順や概念と類似しているが，具体的なワークシートや手法となると必ずしも一致していない。今まで慣れ親しんだ方法を捨てて，(1)及び(2)に全面的に変えることが困難なユーザーのため，EAGLEの中の個々の支援機能を自由に取り出して部分適用できるように柔軟な構造とし，容易にEAGLEが導入できるようにした。

### 2.3 EAGLEの特長

**図3**に示すように，EAGLEには四つの特長があり，以下に説明する。

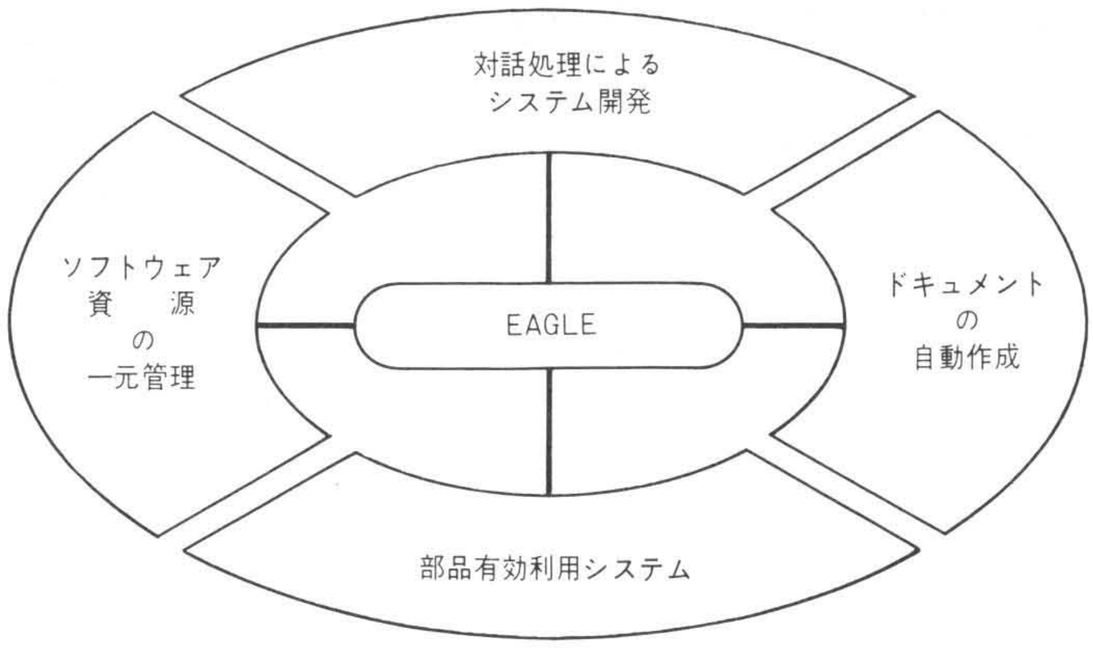



**図3 EAGLEの四つの特長** EAGLEには，(1)対話処理によるシステム開発，(2)データ辞書を用いたソフトウェア資源の一元管理，(3)標準パターン・部品の有効利用によるプログラム生成，(4)保守ドキュメントの自動作成の四つの特長がある。

(1) 対話処理によるシステム開発

システム設計からテストまでを対話処理の作業対象とする。**図4**に示すように日本語のメニュー選択だけで，だれでも容易に作業できる。また，作業ガイダンスやEAGLEからのメッセージも日本語で表示する。

(2) データ辞書を用いたソフトウェア資源の一元管理

膨大なソフトウェア資源(設計仕様，プログラムなど)は，データ辞書を用いて一元管理するので，システム開発や保守での資源間の不一致が防げる。また，データ辞書には資源間の関連情報が格納されており，必要な相互参照情報(クロスリファレンス)が検索できる。

(3) 標準パターンと部品の有効利用によるプログラム生成

プログラムの形態別に用意されている標準パターンと部品(サブルーチン)を使って，ソースプログラム(COBOL)を自動的に組み立てる。その結果，プログラム作成工数が削減でき，また，信頼性の高い標準的なプログラムとなるため保守性が良い。標準提供される22個の標準パターンは，CANDOでの利用実績を踏まえており，バッチCOBOL業務プログラムの大部分をカバーしている。

(4) 保守ドキュメントの自動作成

日本語の保守ドキュメント(設計ドキュメント，プログラムドキュメント，クロスリファレンスリスト)を自動作成するため，ドキュメントの清書や仕様変更に伴う書き直し工数が削減できる。更に，設計仕様やプログラム仕様とドキュメントが常に一致するので，効率の良い保守作業が行なえる。

## 3 EAGLEの機能

**図5**にEAGLE利用時の作業の流れを示す。本章では，この流れの順にEAGLEの機能を説明する。

### 3.1 データ項目の登録と更新

システム開発の対象となる業務システムのデータ分析を行ない，データリレーションを抽出し正規化する[7)]。この正規化されたデータリレーションを構成するデータ項目をデータ辞書に登録し，以後の各工程で利用する。

データ項目ごとに，名前(プログラム中で用いるデータ名，略称，日本語名とその振り仮名など)，属性(データタイプと長さ)，説明及びユーザーの独自項目(値の範囲，テスト時の標準値，含まれるデータリレーションの名前，データの管理元や発生源など)を指定できる。

＊＊＊ EAGLE システム開発メニュー ＊＊＊

次に行なう業務番号＝＝＞〔 〕プロジェクト名＝＝＞〔 〕

1. システム設計……画面・帳票仕様，データベース仕様，ファイルレコード仕様の定義
2. プログラム作成…プログラム仕様の定義，プログラム生成，エディト，コンパイルの実行，対話形テストの実行
3. 管 理……データ辞書作成，パターン・部品管理，情報検索，ドキュメント作成など

＊＊＊ EAGLE システム設計メニュー ＊＊＊

次に行なう業務番号＝＝＞〔 〕 仕様名＝＝＞〔 〕

| 〈データベース仕様〉 | 〈画面・帳票仕様〉 | 〈レコード仕様〉 |
|---|---|---|
| 11 作成・更新 | 31 レイアウト作成・更新 | 41 作成・更新 |
| 12 表示 | 32 項目定義 | 42 表示 |
| 13 定義パラメータ生成 | 33 書式オーバレイ定義 | 43 略称変換（データ辞書使用） |
| 〈ファイル仕様〉 | 34 テスト表示 | 44 原始文生成 |
| 21 作成・更新 | 35 定義パラメータ生成 | |
| 22 内容表示 | | |

＊＊＊ EAGLE プログラム作成メニュー ＊＊＊

次に行なう業務番号＝＝＞〔 〕プログラム名＝＝＞〔 〕

| 〈プログラム仕様〉 | 〈編集・コンパイル〉 | 〈テスト〉 |
|---|---|---|
| 11 作成・更新 | 31 プログラム編集（DESP） | 41 テストコマンド生成 |
| 12 表示 | 32 コンパイル | 42 テストコマンド修正（DESP） |
| 13 プログラム生成 | | 43 COBDBG実行 |
| 〈パターン・部品検索〉 | | |
| 21 一覧表表示 | | |
| 22 内容表示 | | |

＊＊＊ EAGLE 管理メニュー ＊＊＊

次に行なう業務番号＝＝＞〔 〕

| 〈データ辞書作成〉 | 〈情報検索〉 | 〈ドキュメント作成〉 |
|---|---|---|
| 11 データ項目定義 | 31 一覧表表示 | 41 データ辞書（データ項目） |
| 12 データ項目表示 | 32 標準パターン・部品 | 42 設計ドキュメント |
| 13 関連情報自動作成 | 33 設計仕様 | 43 プログラム・ドキュメントト |
| 〈パターン・部品〉 | 34 クロスリファレンス | 44 クロスリファレンス |
| 21 パターン作成・更新 | | |
| 22 部品作成・更新 | | |

図4 EAGLEの業務選択メニュー システムエンジニア，プログラマ及び管理者の三つの職能別に，システム設計，プログラム作成及び管理の三つのメニューを用意している。

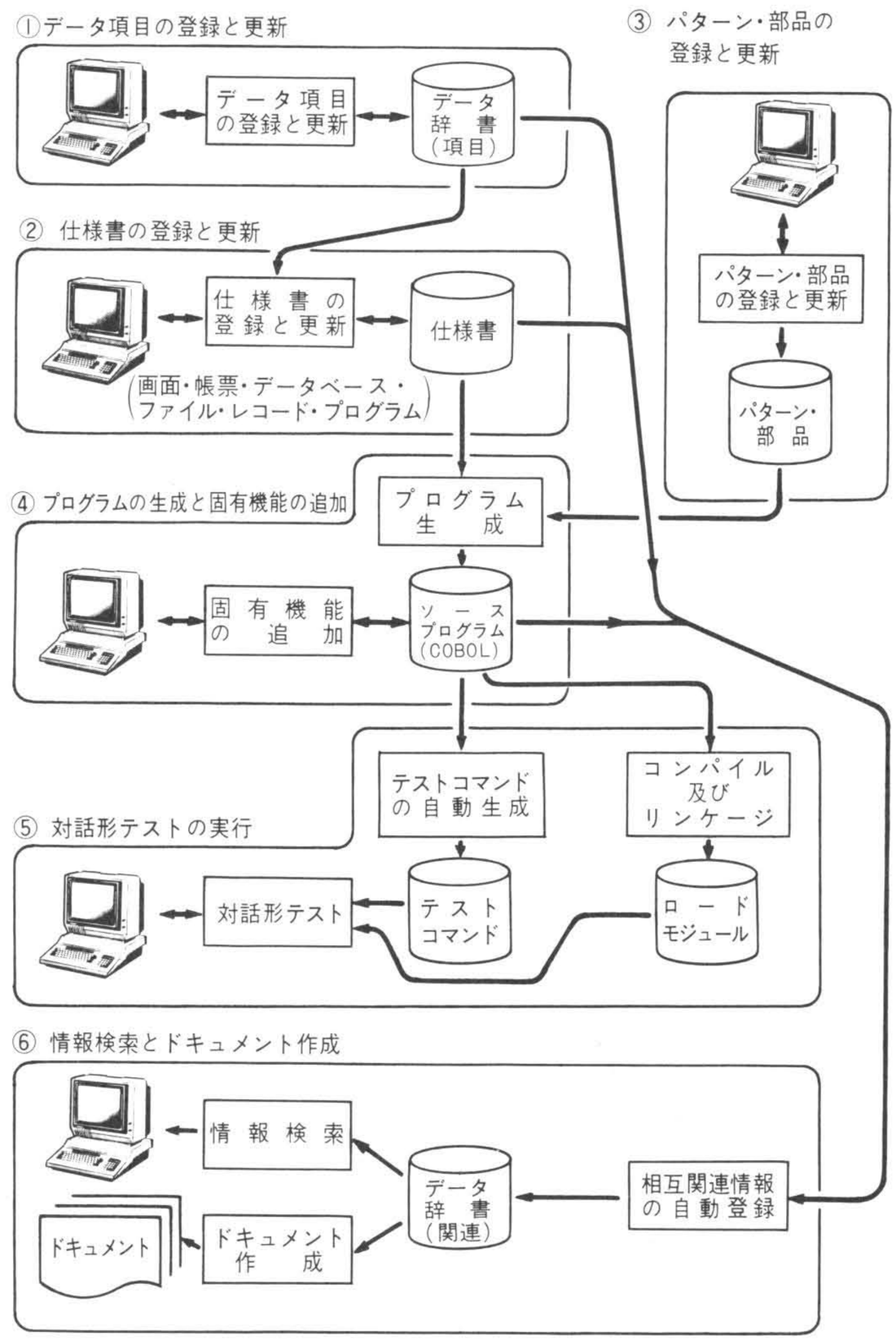



図5 EAGLEを用いたシステム開発手順 ①～⑤の順番でシステムを開発する。⑥はいつでも実行できる。

### 3.2 仕様書の登録と更新

システム設計工程でワークシートに記述した設計仕様を，仕様書ライブラリに登録する。登録する仕様書には，次のものがある。

(1) 画面・帳票仕様〔端末ディスプレイ，端末プリンタ，センタ(漢字)プリンタ，書式オーバレイなど〕

画面・帳票イメージをそのまま対話形で定義後，直ちに確認画面を表示する。それにより，エンドユーザーが本番時のイメージを確認できるので，最も仕様変更が多く本番直前まで仕様が安定しなかった画面・帳票仕様が早期に安定する。

(2) データベース仕様(**図6**に例を示す。)

(3) ファイル仕様

作成/更新 ＊＊＊ データベース仕様（ORDRDB） ＊＊＊

| | |
|---|---|
| プロジェクト名 | PROJECT1 |
| 概 要 | ORDER DATA BASE |

| 区分 | データセット名 | 編成法 | BF数 | スペースレンジ | ロードリミット | 回 復 |
|---|---|---|---|---|---|---|
| M | GOOD | KSDS | 50 | | | Y |
| M | CUST | ESDS | 60 | | | Y |
| V | ORDR | ESDS | 40 | 1,000 | 70 | Y |
| | | | | | | |
| | | | | | | |

作成/更新 ＊＊＊DBアクセス仕様（AORDR1）＊＊＊

| | | | |
|---|---|---|---|
| プロジェクト名 | PROJECT1 | | |
| 概 要 | ORDR DB UPDATE | | |
| プログラム区分 | MPP | 言語区分 | COBOL |

| | | | | | |
|---|---|---|---|---|---|
| DBM名 | ORDRDB | 使用DB | STD | | |
| プログラム名 | PORDR1 | | | | |

| データセット名 | I/O区分 | BASELK | データセット名 | I/O区分 | BASELK |
|---|---|---|---|---|---|
| GOOD | U | | CUST | U | |
| ORDR | U | GOODLK01 | | | |

図6 データベース仕様書 PDMIIデータベースの定義例である。データベース仕様では，データベースの構成を指定する。DBアクセス仕様では，各プログラムのデータベースに対するアクセス方法(検索や更新)を指定する。

(4) レコード仕様
(5) プログラム仕様

仕様書は，各種の定義パラメタやソースプログラムの生成など多目的に使用される。

### 3.3 パターン・部品の登録と更新

システム設計と並行して，プログラムの標準パターンと部品をパターン・部品ライブラリに登録する。汎用的なCOBOL標準パターン(**図7**)と部品は，製品として提供される。また，ユーザーが独自にパターンと部品を開発し，使用することもできる。なお，一般に処理アルゴリズムに関する部品(処理部品)は，なかなか汎用化しにくく，むしろデータに着目したデータ項目対応の部品(データ部品)のほうが汎用化しやすい。データ部品の標準化を推進すれば，部品組立てによるプログラム開発が容易となる。

### 3.4 プログラムの生成と固有機能の追加

システム設計工程で作成したプログラム仕様書には，プログラムの構成要素〔使用するパターン・部品，入出力(画面・帳票・データベース・ファイル)など〕が指定済みである。この指定に従い，ソースプログラムを生成する。この生成方法を**図8**に示す。

生成後のソースプログラムに対して，各プログラムの固有機能をエディタを用いて追加し，プログラムを完成する。

パターンや部品でカバーできない固有機能は，従来と同じ方法でプログラム設計とコーディングを行なえばよい。コーディング工程では構造化プログラミング技法が有効であることが1970年代の初めから実証されてきた。しかし，以前のCOBOLは，PASCALやPL/Iと比べ構造化プログラミング機能が弱体であったが，日立製作所の大形オペレーティングシステム用のCOBOLでは**表1**に示すような構造化プログラミング機能を支援しているので，質の良いプログラムの作成が可能である。なお，本機能は，アメリカのCOBOL改訂規格案[8](1984年中に制定の見込み，1987年ごろJIS化の見込み)に準拠している。

### 3.5 対話形テストの実行

完成したソースプログラムを解析してテストコマンドを自動生成し，対話形テストを実行する。対話形テストには，COBOLデバッグ支援を使用する。中間点でのデータ表示，副プログラム単独実行用のシミュレーション(**図9**)，ファイ

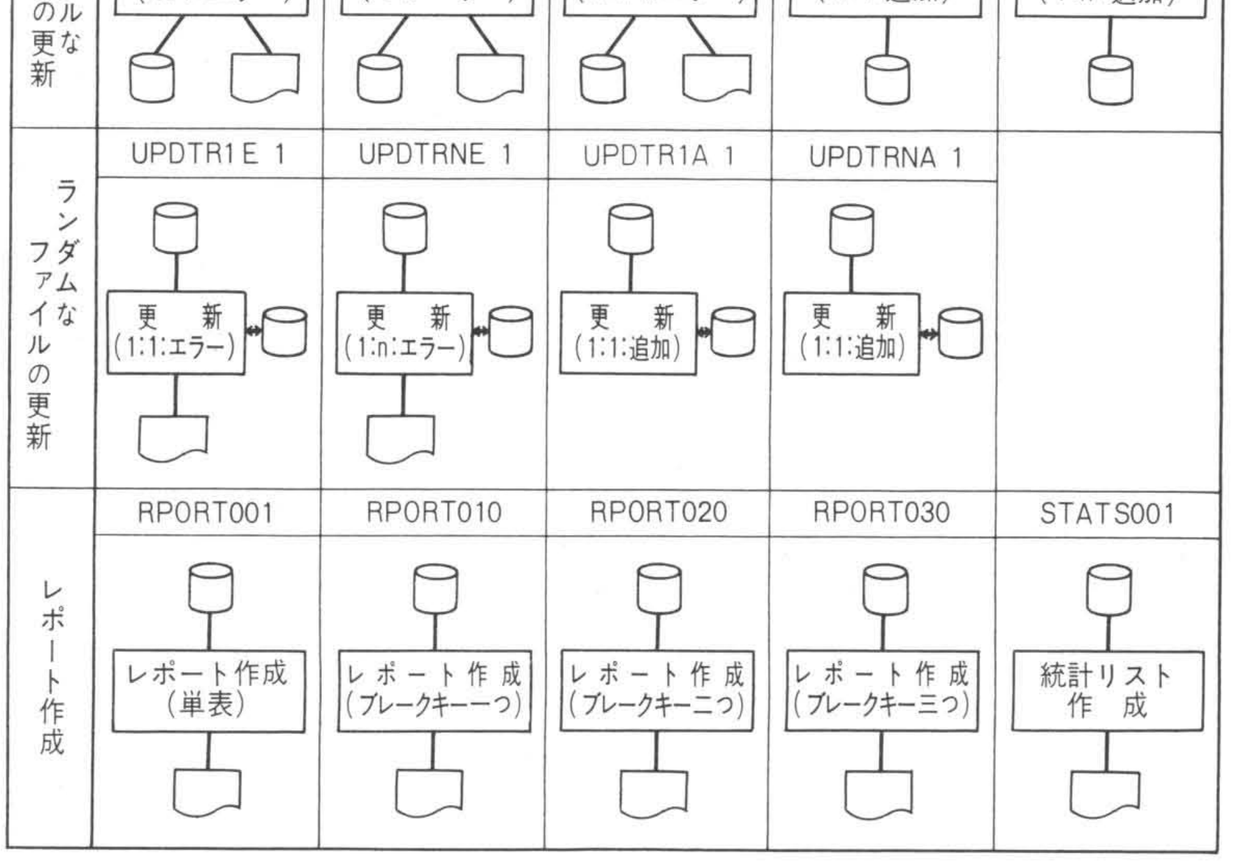



図7 22個の標準パターン　22個の標準パターンで，大部分のバッチCOBOL業務プログラムの生成ができる。

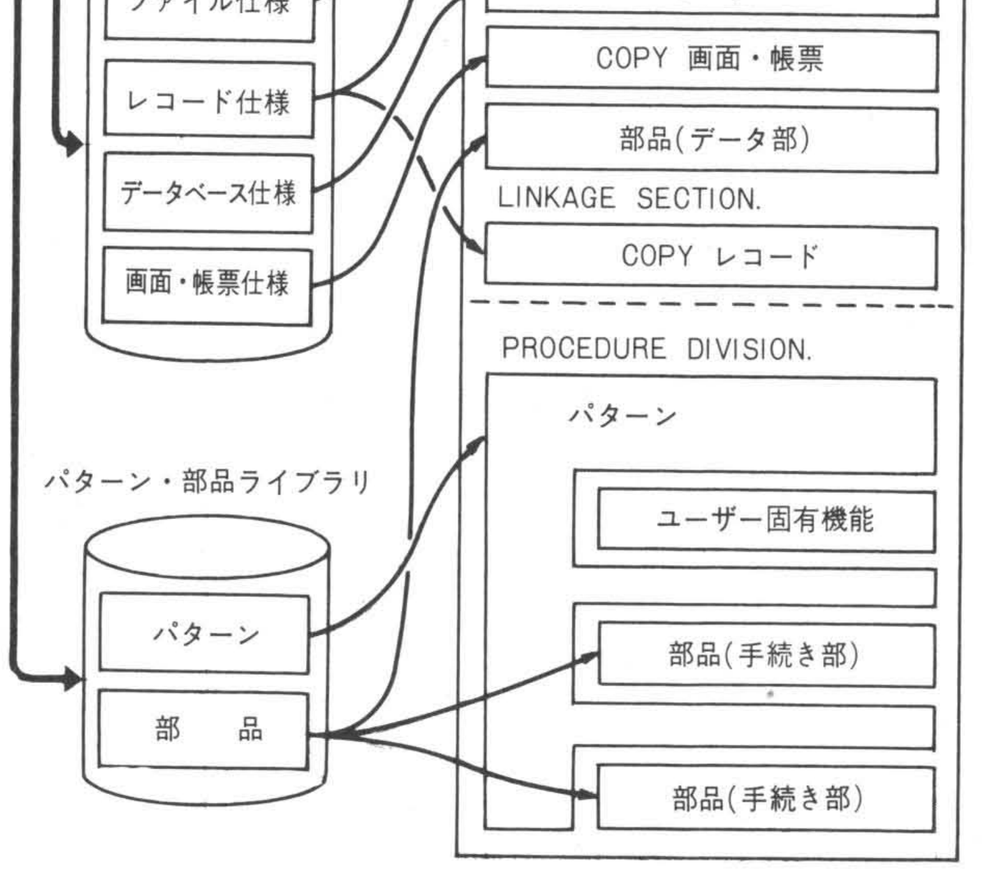
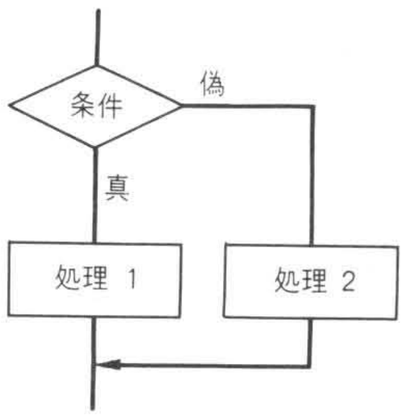
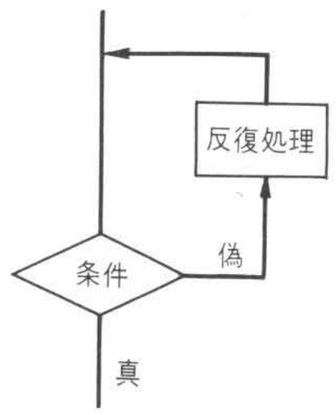
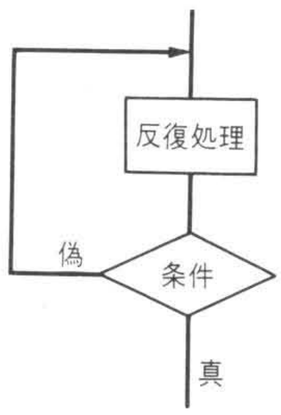
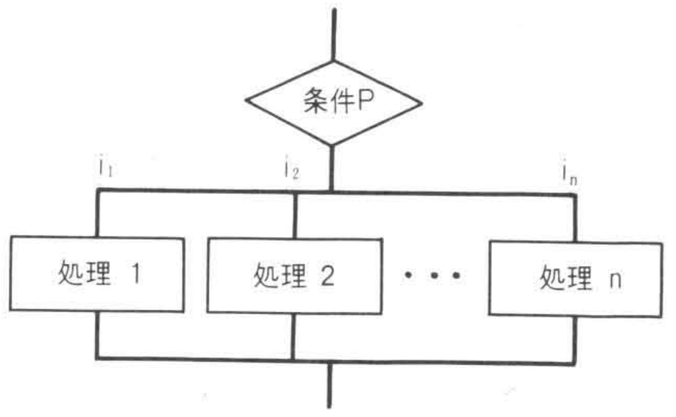
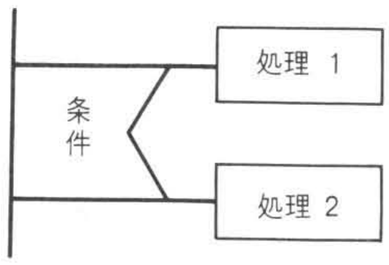
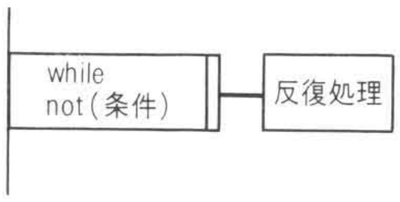
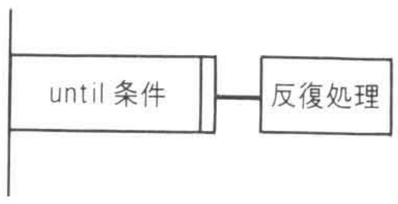
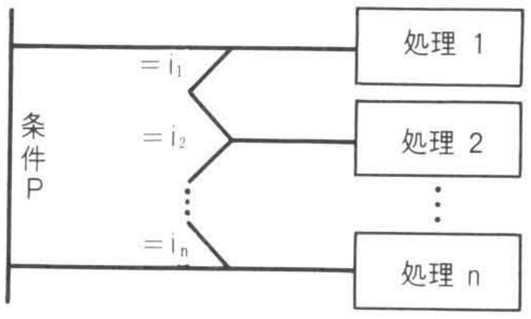



図8 EAGLEのプログラム生成方式　プログラム仕様書の指定に従い，データ仕様パターン及び部品を用いてソースプログラムを組み立てる。

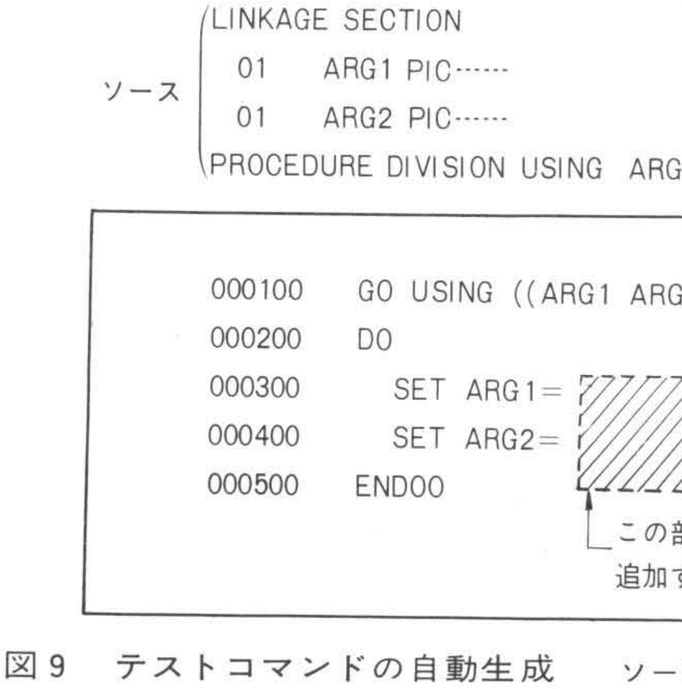
表1　COBOL構造化プログラミング用の命令　VOS3 COBOLの構造化プログラミング用の命令は，アメリカ規格COBOL改訂案(1984年中に制定の見込み)に準拠している。

| 種類 | if-then-else形 | do-while形 | repeat-until形 | case形 |
|---|---|---|---|---|
| フローチャート |  |  |  |  |
| COBOL | IF 条件 THEN<br>処理1の命令群<br>ELSE<br>処理2の命令群<br>END-IF | PERFORM<br>[WITH TEST BEFORE]<br>UNTIL 条件<br>反復処理の命令群<br>END-PERFORM | PERFORM<br>WITH TEST AFTER<br>UNTIL 条件<br>反復処理の命令群<br>END-PERFORM | EVALUATE 条件P<br>WHEN $i_1$処理1の命令群<br>⋮<br>WHEN $i_n$処理nの命令群<br>END-EVALUATE |
| PAD図式 |  |  |  |  |

ソース
```
LINKAGE SECTION
  01    ARG1 PIC……
  01    ARG2 PIC……
PROCEDURE DIVISION USING  ARG1 ARG2
```

```
000100    GO USING ((ARG1 ARG2))
000200    DO
000300       SET ARG1=
000400       SET ARG2=
000500    ENDOO
```
この部分をエディタで追加する。

図9　テストコマンドの自動生成　ソースプログラムを解析して，テストコマンドを自動生成する。この例は，副プログラム単独実行用のシミュレートコマンドの生成例である。




| | 走行済み命令 | テストカバレージ |
|---|---|---|
| 1回目 | 12 | 32% |
| 2回目 | 27 | 73% |

図10　テストカバレージ機能　COBOLデバッグ支援には，テストの進捗状況を数値化し，表示するテストカバレージ機能がある。走行済み命令数とその割合($C_0$カバレージ)及び走行済み分岐(パス)数と，その割合($C_1$カバレージ)を表示する。

ル シミュレーションなどのほか，テストカバレージ機能(**図10**)によるテスト十分性のチェックを行なう。

### 3.6　情報検索とドキュメント作成

データ辞書(項目)，仕様書及びソースプログラムを解析し，相互関連情報をデータ辞書(関連)に自動登録する。このデータ辞書を検索することにより，あるデータベースを参照しているプログラム一覧などを調べることができる。

また，データ辞書(項目及び関連)と各種ライブラリの情報を解析し，**表2**で示す種類の保守用ドキュメントを出力する。**図11**にADCAS(Auto Documentation Aid System)を用いて出力したPAD(Problem Analysis Diagram)図を示す。

## 4　EAGLEによる効果

### 4.1　稼動環境と適用分野

EAGLEは，HITAC Mシリーズの大形オペレーティングシステムで稼動する。日本語を全面的に採用しているため，T-560/20漢字ステーションと漢字プリンタを必要とする。開発対象業務はバッチ及びオンラインデータベースシステムであり，支援するプログラム言語はCOBOLである。

### 4.2　適用効果

EAGLEを用いることにより，以下に述べる効果があり，従来のシステム開発と比べて2～3倍の生産性及び信頼性が期待できる。

(1)　HIPACEの作業手順と技法の適用による作業の標準化
(2)　対話形の画面・帳票設計により，設計仕様の早期安定
(3)　設計仕様とプログラム標準パターン・部品の再利用
(4)　テストコマンド自動生成とテスト十分性チェック機能(テストカバレージ機能)によるテストの効率化と標準化
(5)　データ辞書を用いたソフトウェア資源の一元管理
(6)　日本語保守用ドキュメントの自動作成
(7)　日本語メニュー ガイダンスによる容易な操作

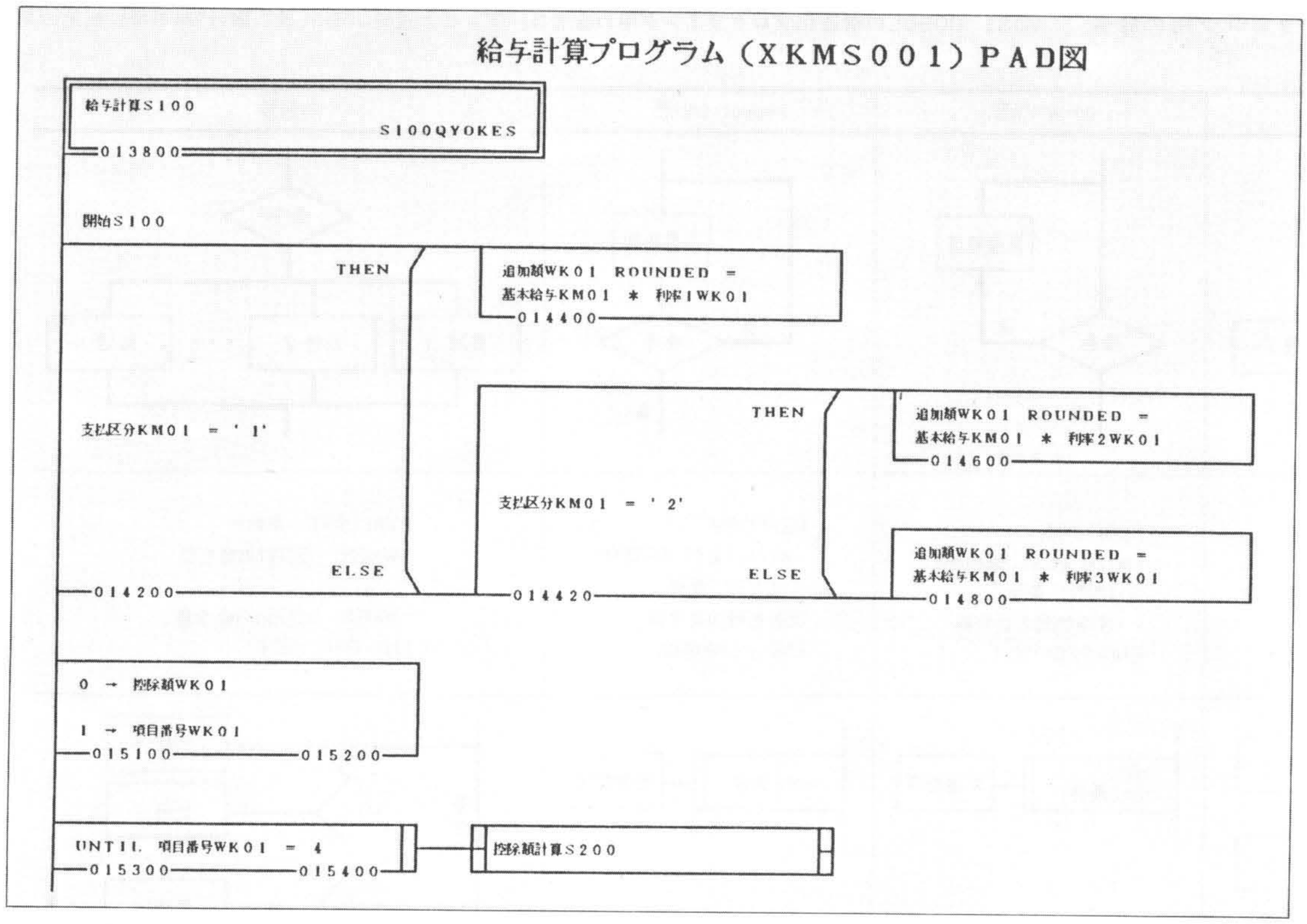



図11 保守用ドキュメント出力例(PAD図) ソースプログラムを解析して，PAD(Problem Analysis Diagram)図を出力する。データ名やコメントは，データ辞書(項目)や用語辞書を用いて日本語に変換している。

表2 EAGLEの保守用ドキュメント一覧 項番1～8は設計ドキュメント，項番9～16はプログラム ドキュメントである。プログラム ドキュメントは，EAGLEを構成するプログラムADCASで出力する。

| No. | ドキュメント名称 | 概要 |
|---|---|---|
| 1 | 一覧表 | 各定義データの一覧表 |
| 2 | データ項目仕様表 | データ項目辞書の内容を出力する。 |
| 3 | データベース仕様表 | データベース仕様と構造図を出力する。 |
| 4 | ファイル仕様表 | VSAM，SAMファイル仕様を出力する。 |
| 5 | レコード仕様表 | レコード仕様を出力する。 |
| 6 | 画面仕様表 | 画面のレイアウト，項目のデータを出力する。 |
| 7 | 漢字帳票仕様表 | 漢字帳票のレイアウト，項目のデータを出力する。 |
| 8 | プログラム処理概要図 | プログラム仕様の内容を出力する。 |
| 9 | プログラム関連図 | プログラムの呼ぶ・呼ばれる関係を階層構造図の形式で出力する。 |
| 10 | プログラム機能説明図 | プログラムの機能概要をデータフロー図(HIPO図)の形式で出力する。 |
| 11 | セクション関連図 | プログラムを構成するセクションの関係を階層構造図の形式で出力する。 |
| 12 | モジュール仕様書 | プログラムを構成するセクション単位に処理概要をデータフロー図(HIPO図)の形式で出力する。 |
| 13 | PAD図 | プログラムを構成するセクション単位に処理構造，手順をPAD図の形式で出力する。 |
| 14 | SPチャート | プログラムを構成するセクション単位に処理構造，手順をSPチャートの形式で出力する。 |
| 15 | 日本語ソースリスト | 原始プログラムの行単位に，入力原始文とこれに対応した日本語原始文を出力する。 |
| 16 | データ定義書 | 原始プログラム，登録集原文(COBOL)中のデータセグメント単位に，個々のデータの名称，属性，意味，用途などを一覧表の形式で出力する。 |

注：略語説明 VSAM(Virtual Storage Access Method)
SAM(Sequential Access Method)
HIPO(Hierarchy plus Input-Process-Output)
PAD(Problem Analysis Diagram)

## 5 結言

EAGLEは，HIPACEの標準手順・技法と一体となって，効率的なシステム開発を支援するソフトウェアであり，全工程を対象としていることによる総合性と，データ辞書を用いたデータ資源管理の考え方に基づく一貫性に特長がある。

今後，いっそうの効率向上を実現するために，EAGLEの機能拡張及び適用結果のフィードバックの推進に努力する考えである。

最後に，EAGLEの開発に当たり，御協力をいただいた関係各位に対し，感謝の意を表わす次第である。

## 参考文献

1) 宮副，外：アプリケーションシステムの効率的設計技法“HIPACE”，日立評論，**62**，12，861～866(昭55-12)
2) 原田，外：オンラインデータベースシステム向き業務プログラム開発維持支援システム「漢字CORAL」，日立評論，**64**，5，351～354(昭57-5)
3) 小野，外：構造化ソフト開発支援システム“CANDO”，日立評論，**66**，3，195～198(昭59-3)
4) 片岡，外：ソフトウェア開発支援システム(CASDシステム)，日立評論，**62**，12，879～882(昭55-12)
5) 日立製作所：共通技術マニュアル，システム開発標準手順SPDS概説編，SE-160-10(1982-10)
このほかに，SPDSには工程別に8冊のマニュアルがあり，作業標準手順が詳細に説明されている。
6) 堀内，外：データ資源管理の概念と動向，日立評論，**64**，5，317～322(昭57-5)
7) 味村，外：データベースシステムの設計と開発，1983年，オーム社
8) 今城：COBOLの標準化の動向，情報処理，**24**，9，1062～1069(1983-9)