

# VOS K第4世代言語 “EAGLE/4GL”

## VOS K 4th Generation Language “EAGLE/4GL”

今までのシステム開発は、COBOL、PL/Iなどの第3世代言語や、EAGLE (Effective Approach to Achieving High Level Software Productivity)などのシステム開発支援ソフトウェアによって行ってきた。しかし、これらの多くは言語知識が必要であったり、システムや他ソフトウェアとのインタフェースを意識する必要があったため、信頼性や生産性がいま一步伸びていなかった。この問題を解決するため、プログラム設計から運用、保守までを一貫して行える第4世代言語EAGLE/4GL (EAGLE/4th Generation Language)を開発した。

EAGLE/4GLは、画面に従って目的プログラムの情報を入力あるいは選択することで、実行形式プログラムが作成できる特長を持ち、コンピュータの専門知識を持たない実務担当部門でも容易に使いこなせるソフトウェアである。

今城哲二\* *Tetsuji Imajō*  
 秋山美登\* *Yoshinori Akiyama*  
 北尾修治\*\* *Shūji Kitao*  
 里本 健\*\* *Takeshi Satomoto*  
 脇坂隆則\*\* *Takanori Wakizaka*  
 大西俊治\*\*\* *Toshiharu Ōnishi*

### 1 緒 言

急激な情報化社会への進展に伴い、コンピュータシステムが企業活動のかなめを占めるに至り、各企業ではこれらのコンピュータシステムを、いかに効率よく開発・保守していくかが重要な課題となっている。しかも、エンドユーザーからのコンピュータシステムに対するニーズが多様化し、またシステム自身も大規模化・複雑化してきている。

これらの問題に対処するために、最近、システム開発の生産性や信頼性向上を目的とする総合的なシステム開発支援ソフトウェアが各種提供されてきている。しかし、それらの多くはCOBOLやPL/I言語の専門知識が必要不可欠であったり、保守効率が期待ほど上がっていないなどの問題点があり、生産性がいま一步伸びていないのが現状である。

日立製作所では、これらの問題を解決するために、言語を既存のCOBOL (PL/I)レベルにとどめずに、より高レベルなユーザーインタフェースを持つ言語として、第4世代言語EAGLE/4GL (Effective Approach to Achieving High Level Software Productivity/4th Generation Language)を開発した<sup>2)</sup>。

EAGLE/4GLは、HITAC M-620/M-630/M-640システムで稼動するオペレーティングシステムVOS K (Virtual-storage

Operating System Kindness)のもとで利用できる。VOS Kは、コンピュータの専門知識を持たなくても容易に使いこなせることを目指したシステムであり、基本機能としてRDB (リレーショナル形データベース)を具備し、そのRDB機能を前提として開発されたEAGLE/4GLを提供している。

一般的に、第4世代言語は次のように定義されている。

- (1) ユーザーフレンドリーであること。
- (2) 専門家でないプログラマにも結果が得られること。
- (3) COBOL (第3世代言語)の $\frac{1}{10}$ の命令数で結果が得られること (手続き形)。
- (4) データベースシステムを直接利用できること。
- (5) プロトタイプを手早く作成し、変更することができること。
- (6) 一般的ユーザーが2日間の学習で一応使えるようになること。
- (7) COBOLの $\frac{1}{10}$ 以下の期間で結果が得られること、など。

EAGLE/4GLは、これらの要件を満たす情報処理部門向け開発支援システムの第4世代言語である。

本論文では、EAGLE/4GLの開発方針、特長および代表的な機能について説明する。

\* 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 情報システム工場

## 2 EAGLE/4GLの概要

### 2.1 EAGLE/4GLの開発方針

第4世代言語は、非定形的な業務を利用部門で直接作成するための簡易言語と呼ばれるものから、コンピュータの専門家(情報処理部門)が基幹業務を開発するための高生産性言語と呼ばれるものまでであるが、EAGLE/4GLは後者の基幹業務開発用の高生産性言語として開発された。

### 2.2 EAGLE/4GLの特長

EAGLE/4GLには大きく次の4点の特長がある。

#### (1) 対話処理によるシステム開発

EAGLE/4GLは、VOS Kの対話処理環境下でシステム設計から運用・保守までの各作業を一貫して支援する。業務画面は、**図1**に示すように日本語メニューであり、ガイダンスやEAGLE/4GLからのメッセージも日本語で表示されるので、初心者でも容易に使用できる。

#### (2) プロトタイピングによる画面・帳票定義

**図2**に示すように、画面・帳票イメージを画面上で、ラフスケッチ定義することにより、直ちに確認画面を表示するプロトタイピング技法を採用している。これにより、システム設計の段階で最終的な画面・帳票様式、大きさ、色などの詳細な仕様確認が可能であるため、早期に出力フォーマットを確定することができ、システム設計工程での生産性が大幅に向上する。

#### (3) 処理のパターン化・日本語プログラミング

EAGLE/4GLでは、**図3**に示すようにコンピュータ業務処

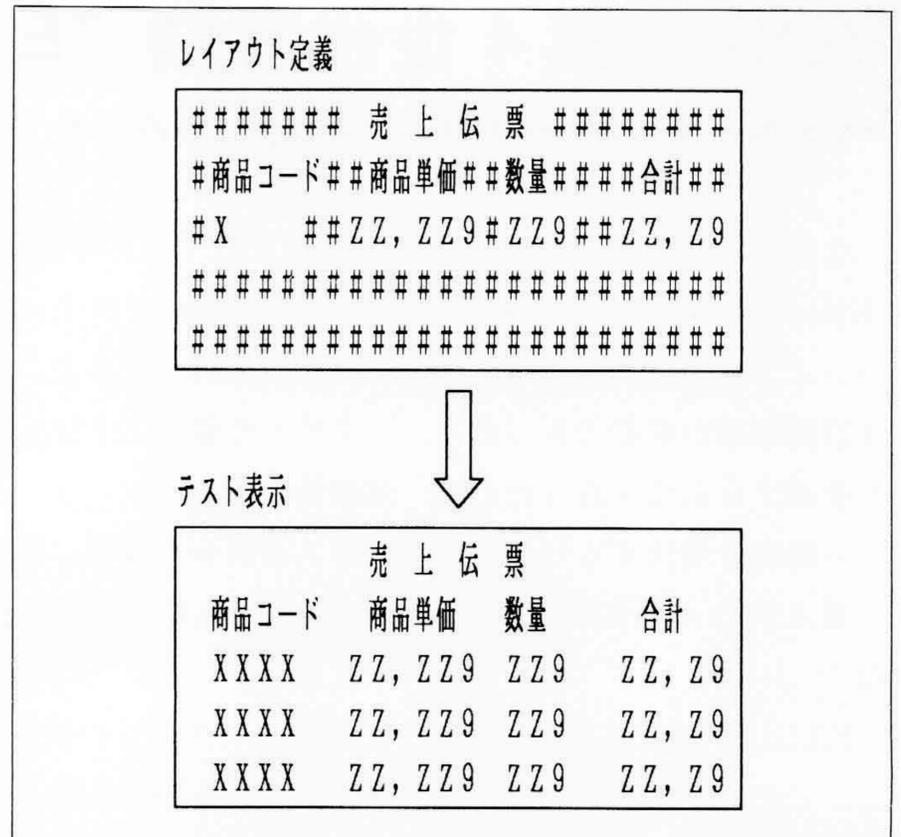


図2 プロトタイピングによる画面・帳票定義 画面、帳票の入出力項目を選択するだけでプログラムが自動生成され、イメージをその場で確認できる。

理のパターンを、

- (a) 対話パターン
- (b) ファイル作成・更新パターン(バッチ)
- (c) マッチングパターン(バッチ)

の3種類に集約した。このパターンと一致する業務処理については、基本アルゴリズムを考える必要がなく、対話形式で空きを埋めていくだけでプログラミングができるため、プログラム作成工程での生産性が飛躍的に向上する。

また、**図4**に示すように、プログラムの入出力仕様書名(ファイル、画面、帳票)などを指定するだけで、EAGLE/4GLが処理仕様書(EAGLE/4GLのソースプログラム)を自動生成する。ユーザーはこの自動生成(初期生成)された処理仕様書を編集して処理仕様書を完成させていけばよい。

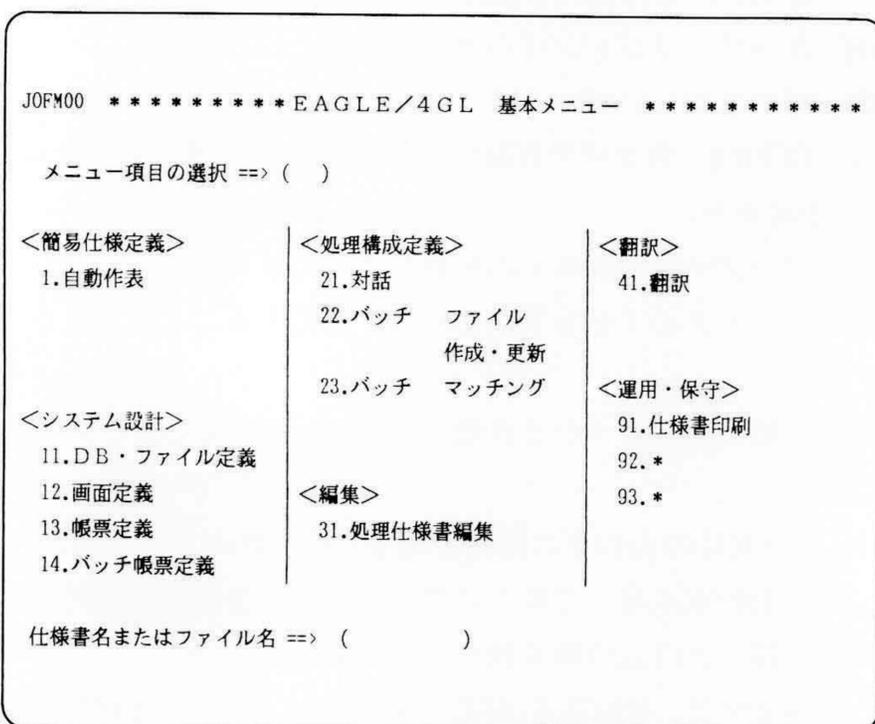
さらに、自動生成される基本ロジックにきめ細かな処理記述の追加が必要な場合のために、各種の処理記述文や関数をサポートしている。これらの処理記述文や関数、およびデータ項目名称には日本語が使用できる。

#### (4) 簡易仕様定義(自動作表)機能による業務仕様自動生成

帳票を出力するプログラムの開発に有効な機能であり、入出力構成および帳票の形、項目情報を選択するだけで、帳票出力のプログラムを自動生成できる。

## 3 EAGLE/4GLの機能

EAGLE/4GLでの開発方法には、通常の手順での方法と、簡易仕様定義(自動作表)機能を使った方法(簡単な帳票を出力するプログラムを作成する場合)の二とおりがある。本章では、



注：略語説明 EAGLE/4GL(Effective Approach to Achieving High Level Software Productivity/4th Generation Language) DB(Data Base)

図1 EAGLE/4GLの業務選択メニュー 目的業務の番号を選択することで業務を開始できる。

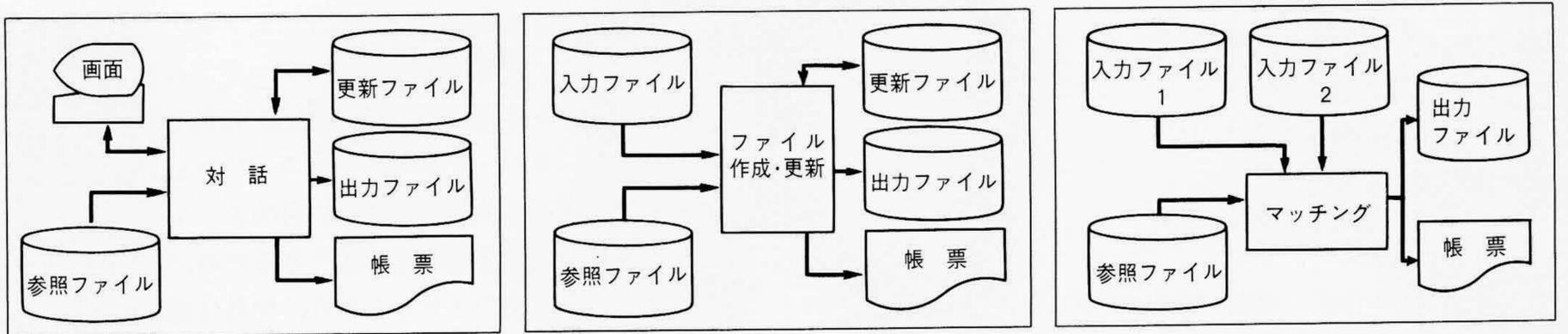


図3 EAGLE/4GLの処理パターン 対話1パターン、バッチ2パターンでプログラムが組める。

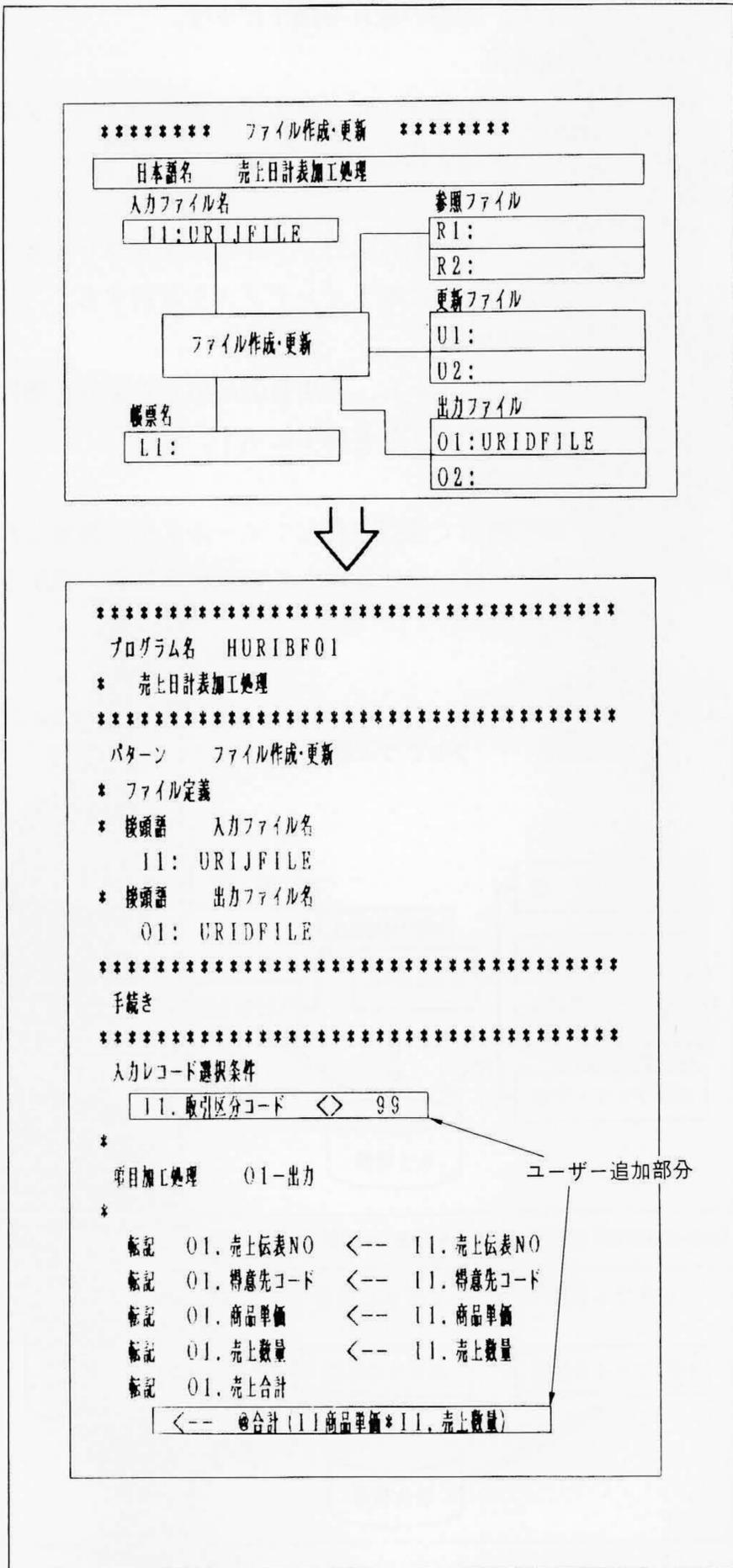


図4 処理仕様書 プログラム入出力情報を指定し、処理仕様書を自動生成する。データ項目の名称や処理の指示(文)に日本語が使える。

この流れの順にEAGLE/4GLの機能について説明する。

### 3.1 通常の開発

次の機能を利用して、実行形式プログラムを作成する。

#### (1) データベース、ファイル定義

データベース、ファイルおよびレコードを定義する。定義する内容には、フィールド名、フィールドの属性、インデクス情報、ファイル編成、排他制御区分などがある。定義した情報はファイル仕様書として登録される。

#### (2) 画面定義

作成するプログラムで使用する画面の様式を定義する。文字やデータ出力位置のレイアウトや属性およびけい線などが画面上でラフスケッチしながら簡単に定義できる。プロトタイプング技法の採用により、定義した内容をテスト表示して確認できる。また、定義画面とテスト表示画面を分割表示できるので、出力画面を確認しながら定義することもできる。定義した情報は画面仕様書として登録される。

#### (3) 帳票定義

対話プログラムで使用する出力帳票(ワークステーションプリンタ用)の様式を定義する。帳票のサイズ、印刷形式、文字やデータ出力位置のレイアウトや属性、けい線などが簡単に定義できる。プロトタイプング技法による定義方法は画面と同様である。定義した情報は帳票仕様書として登録される。

#### (4) バッチ帳票定義

バッチプログラムで使用する出力帳票(センタプリンタ用)の様式を定義する。定義する内容には、帳票のサイズ、印刷形式、文字やデータ出力位置、行種別(見出し行、明細行、合計行)などがある。定義した情報はバッチ帳票仕様書として登録される。

#### (5) 処理構成定義

EAGLE/4GLの処理パターンには、対話処理用で一つ、バッチ処理用で二つの計三つのパターンがある。処理構成定義では作成する処理に合うパターンを選択し、必要な入出力仕様書名(ファイル、画面、帳票)などを指定して処理の構成を定義する。定義した情報は処理仕様書として登録される。

各パターンで作成できる処理の種類を以下に示す。

- (a) 対話
  - (i) 業務振り分け
  - (ii) 問い合わせ
  - (iii) ファイル更新
  - (iv) ファイル更新と帳票出力
  - (v) データエントリ
- (b) ファイル作成・更新パターン(バッチ)
  - (i) ファイルの分配と抽出
  - (ii) ファイルの集計と照合
  - (iii) ファイルの更新とレコード追加
  - (iv) 帳票出力
  - (v) 「ファイル更新しながら帳票出力する」などの複合形
- (c) マッチングパターン(バッチ)
  - (i) 入力ファイル2本の照合(1:N)
  - (ii) ファイル照合時のエラーリストの出力
- (6) 処理仕様書編集

処理構成定義によって自動生成された処理仕様書の疑問符(?)部分を追記したり、修正したりしながら処理仕様書を完成させる。既存の処理仕様書を更新する場合も、処理仕様書編集機能で行う。

(7) 翻訳

編集した処理仕様書を翻訳することで、各定義情報(処理、ファイル、画面、帳票、バッチ帳票)から、実行形式プログラムを生成する。

3.2 簡易仕様定義(自動作表)による開発

本機能は、EAGLE/4GLの通常のプログラム開発をより簡単にするものである。開発対象は、ファイル内容を直接印字するような一覧表や、小計や合計を印字するような集計表を出力するプログラムであり、図5に示す入出力構成であることが条件である。

通常の開発と簡易仕様定義による開発の比較を図6に示す。

簡易仕様定義は、入出力構成およびバッチ帳票のフィールド情報を与えることで、簡易仕様書、処理仕様書、バッチ帳票仕様書を生成する。生成した処理仕様書は完成したもので

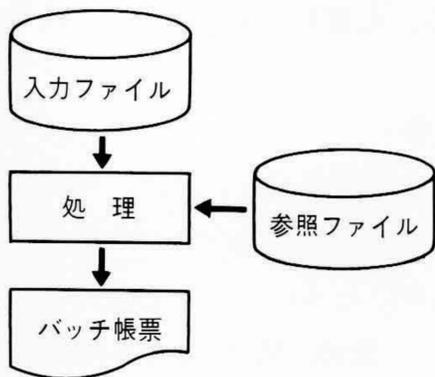


図5 簡易仕様定義(自動作表)の入出力構成 この入出力構成であれば、簡易仕様定義(自動作表)による開発が行える。

あるため、そのまま翻訳すれば実行形式プログラムを作成することができる。通常の手順では、プログラム変更時、処理仕様書およびバッチ帳票仕様書の変更が必要であり、翻訳失敗によるフィードバックが考えられる。しかし、本機能を使用すると簡易仕様書の変更だけで、目的のバッチ帳票仕様書および完成した処理仕様書を生成できるため、翻訳に失敗することはない。以上のメリットから、簡易仕様定義は通常のEAGLE/4GLによる開発作業をさらに削減し、信頼性の高い開発作業を実現できる。

簡易仕様定義(自動作表)機能による開発の一例を挙げて各画面の説明をする。画面の流れを図7に示す。

(1) 入出力構成画面

入力ファイル、参照ファイル、出力する帳票の1ページの行数および1行の字数を指定する。

(2) インデクス選択画面

表示されたインデクスから入力ファイル、参照ファイルからデータを読み込むときに使うインデクスを選択する。

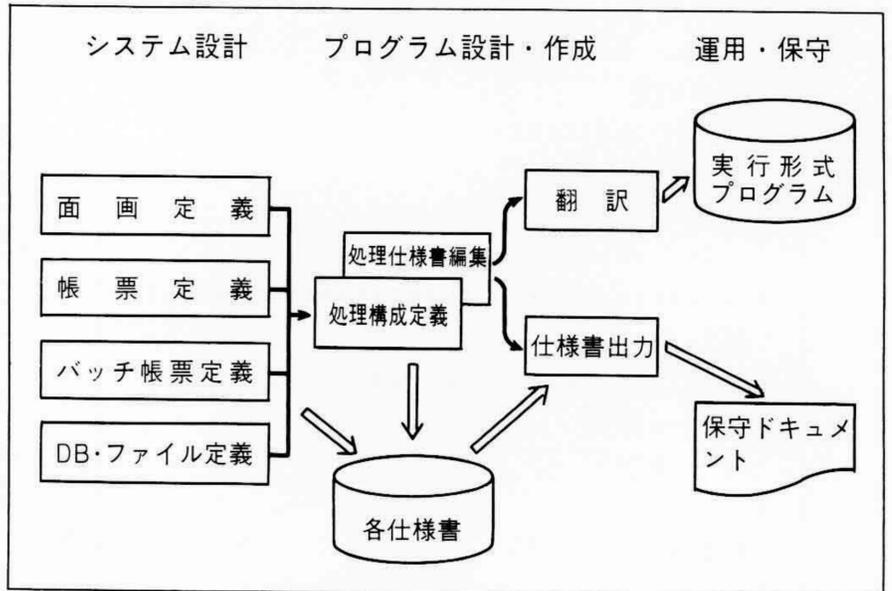
(3) フィールド選択画面

表示されたフィールドから、使用目的に応じた記号を順序欄に指定する。指定する記号を表1に示す。

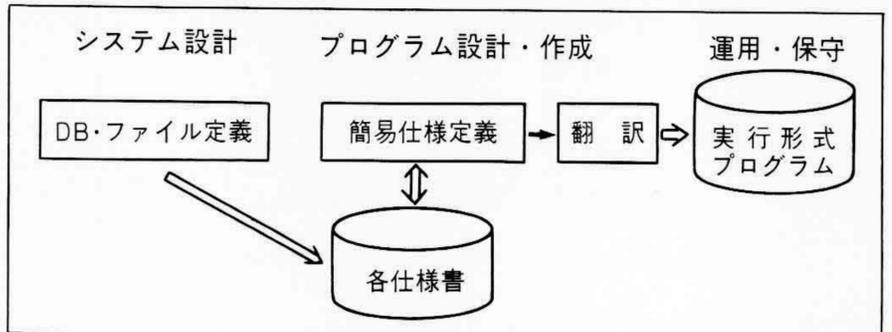
(4) フィールド定義画面

フィールド選択画面で選択されたフィールドが、指定した情報(順序、行数)に従い並び替えられて表示される。表示さ

通常の開発



簡易仕様定義(自動作表)による開発(帳票設計と処理仕様書編集が不要)



注: → (作業の流れ), ⇔ (データの流れ)

図6 EAGLE/4GLの開発手順とデータの流れ EAGLE/4GLによる開発方法は二とおりある。どちらかの方法でプログラムを開発する。

IO \*\*\*\*\*入出力構成(自動作表)\*\*\*\*\* URIAGE

日本語名(表題)	得意先売上リスト
コメント	

入力ファイル名  
I1:UTRN

参照ファイル名  
R1:SMASTER R2:TMASTER  
R3: R4:  
R5: R6:  
R7: R8:

自動作表

バッチ帳票

1ページの行数 ==>( 66 ) (3~99)  
1行の字数 ==>( 132 ) (21~225)

コマンド ( )

1	次画面		PA	行画面	順		割込
5			9			終了	

入出力構成画面

ID \*\*\*\*\*インデクス選択(自動作表)\*\*\*\*\* URIAGE

選択	接頭語	インデクス名	フィールド名
S	R1	SCODEI	商品コード
	R1	SNAMEI	商品名
S	R2	TCODEI	地域コード
	R2	TCHARGE	担当者名

コマンド ( )

1	次画面	前画面	PA	行画面	順		割込
5			9	↑	↓	終了	

インデクス選択画面

FD \*\*\*\*\*フィールド定義(自動作表)\*\*\*\*\* URIAGE

順序	行数	例示	接頭語	帳票のヘッダ(フィールド名)	式/ブレイクキー/合計/平均/キー値フィールド/編集形式
H1		@A	I1	得意先コード	B3
H2		@K	R2	番号	
H1	2	@L	R2	得意先名	
1		@D	I1	商品コード	
2		@F	R1	商品名	
3		@E	I1	売上数量	E(A:3)
S		@B	I1	地域コード	
S		@C	I1	通番	
S		@G	R1	商品コード	K(@D)
S		@H	R1	単価	
S		@I	R2	地域コード	K(@B)
S		@J	R2	通番	
4				金額	@H*@E,S3,SR,E(A:6)
S				実利益	@H*@E*0.8 - 5000,SR,E(A:6)

改ページするブレイクレベル ==> ( 3 ) (1~3)  
入力レコード選択条件 ==> ( @K > 100 )  
ブレイクキー(B1,B2,B3) 合計(S1,S2,S3,ST,SR) 平均(A1,A2,A3,AT,AR)

コマンド ( )

1		前画面	PA	行画面	順		割込
5			9	↑	↓	終了	

フィールド定義画面

FS \*\*\*\*\*フィールド選択(自動作表)\*\*\*\*\* URIAGE

順序	行数	例示	接頭語	レベル	入力ファイルと参照ファイルのヘッダ(フィールド名)	属性	長さ	繰返
H1		@A	I1	01	売上レコード			
S		@B	I1	02	得意先コード	文字	2	
S		@C	I1	03	地域コード	文字	3	
1		@D	I1	02	通番	文字	3	
3		@E	I1	02	商品コード	パック	3	
			R1	01	売上数量			
			R1	02	商品マスタレコード			
2		@F	R1	02	商品名	漢字	5	
S		@G	R1	02	商品コード	文字	3	
S		@H	R1	02	単価	パック	5	
			R2	01	得意先マスタレコード			
			R2	02	得意先コード			
S		@I	R2	03	地域コード	文字	2	
S		@J	R2	03	通番	文字	3	
H2		@K	R2	02	番号	ゾーン	4	
H1	2	@L	R2	02	得意先名	漢字	4	

コマンド ( )

1	次画面	前画面	PA	行画面	順		割込
5			9	↑	↓	終了	

フィールド選択画面

図7 簡易仕様定義(自動作表)の画面遷移 目的プログラムの情報を入力あるいは選択し、生成・翻訳することで実行形式プログラムを作成できる。

表1 フィールド選択 使用目的に応じた記号を順序欄に指定する。

順序欄に指定する記号	フィールドの使用目的	備考
H1~H9	● 頭書きとして帳票に出力する。	記号中の数字は、そのフィールドを帳票に出力する場合、左から何番目に配置するかを表す。
1~99	● 明細、または明細と脚書きとして帳票に出力する。	
S	● 脚書きとして帳票に出力する。 ● 帳票に出力しないで、式の要素、ブレイクキー、キー値フィールドとして使用する。	脚書きに出力したい場合、フィールド定義画面で合計(S3, S2, S1, ST, SR)および平均(A3, A2, A1, AT, AR)を指定する。

れたフィールドに対して、順序・行数・帳票のヘッダを更新し、式・ブレイクキー・合計・平均・キー値フィールド・編集形式を定義する。また、新しくフィールドを追加することもできる。

以上の情報を入力し、生成・翻訳することで実行形式プログラムを作成できる。実行出力結果を図8に示す。

#### 4 EAGLE/4GLの生産性

EAGLE/4GLの生産性を評価するため、受注・販売管理システムをEAGLE/4GLとCOBOLの両方で記述実験した。

プログラムの内容は、受注データの入力や発注商品一覧表、売上日報印刷処理など全9本である。ただし、EAGLE/4GLとCOBOLの定義部分は同等であるため、手続き部(Procedure

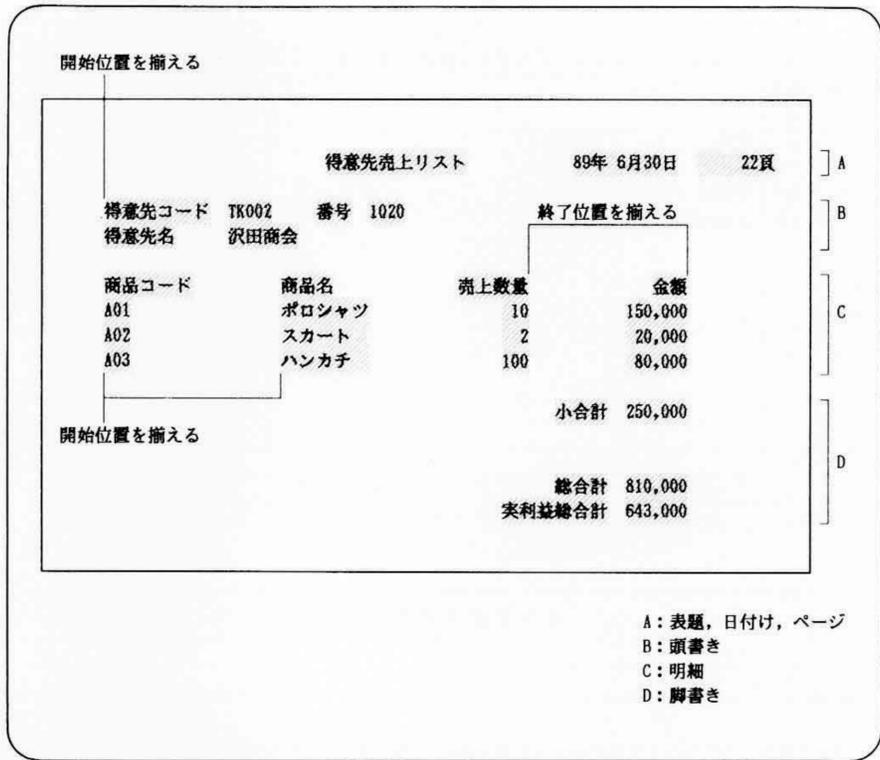


図8 バッチ帳票出力例 図7で作成した目的プログラムの出力結果である。

表2 EAGLE/4GLの生産性 受注・販売管理システムをEAGLE/4GLとCOBOLの両方で記述した結果である。

項番	パターン	手続部記述量		比率
		COBOL	4GL	
1	バッチ	96	11	1/9
2	バッチ	11	0	—
3	バッチ	59	6	1/10
4	バッチ	113	10	1/11
5	バッチ	21	1	1/21
6	バッチ	48	9	1/5
7	対話	232	72	1/3
8	対話	197	72	1/3
9	対話	116	43	1/3

注：開発ステップ数  
 バッチパターン：1/10に減少  
 対話パターン：1/3に減少

Division)だけのステップを対象として評価した。

評価結果を表2に、発注商品一覧表印刷処理のソースリストを図9に示す。

この記述実験で、EAGLE/4GLの記述量はCOBOLに比べ、バッチ処理で1/10、対話処理で1/3となった。

## 5 結 言

EAGLE/4GLでは、プログラム形態に応じたパターンを用いて実行形式プログラムを自動生成する。この結果、今までの第3世代言語でのプログラムの作成やデバッグに要していた工数が削減され、併せて信頼性の高いプログラムを作成できる。

以上のことから、EAGLE/4GLを業務に適用することにより、業務プログラムの生産性は飛躍的に向上する。

```

009300 01 TITLE-SPACE
009400 PROCEDURE DIVISION.
009500 OPEN INPUT HSORT-FD
009600 OUTPUT HATCHUU-FD
009700 I-O LHATCHUU-FD
009800 ACCEPT 日付 FROM DATE
009900 READ HSORT-FD
010000 AT END MOVE '1' TO FLAG
010100 END-READ
010200 PERFORM 主処理 THRU 主処理-終り
010300 UNTIL(FLAG = '1')
010400 CLOSE HSORT-FD
010500 HATCHUU-FD
010600 LHATCHUU-FD
010700 STOP RUN.
010800***** 主処理 *****
010900 主処理.
011000 MOVE 商品コード OF HSORT-REC
011100 TO 商品コード OF LHATCHUU-REC
011200 READ LHATCHUU-FD
011300 COMPUTE 仕入数量 = 最高在庫数量 OF HSORT-REC
011400 - 現在在庫数量 OF HSORT-REC
011500 MOVE 仕入数量 TO 予約在庫数量 OF LHATCHUU-REC
011600 REWRITE LHATCHUU-REC
011700 MOVE 仕入先名 OF LHATCHUU-REC
011800 TO 仕入先名 OF MEISAI-REC
011900 MOVE 仕入先電話番号 TO 電話番号
012000 MOVE 商品名 OF LHATCHUU-REC TO 商品名 OF MEISAI-REC
012100 MOVE 商品コード OF LHATCHUU-REC
012200 TO 商品コード OF MEISAI-REC
012300 MOVE 色 OF HSORT-REC TO 色 OF MEISAI-REC
012400 MOVE 現在在庫数量 OF LHATCHUU-REC TO 在庫数量
012500 MOVE 安全在庫数量 OF LHATCHUU-REC
012600 TO 安全在庫数量 OF MEISAI-REC
012700 IF LINE-COUNT > 17
012800 THEN
012900 COMPUTE PAGE-CNT = PAGE-CNT + 1
013000 MOVE PAGE-CNT TO ページ
013100 WRITE HATCHUU-REC FROM TITLE-SPACE AFTER PAGE
013200 WRITE HATCHUU-REC FROM TITLE-REC1 AFTER 2
013300 WRITE HATCHUU-REC FROM TITLE-REC2 AFTER 4
013400 MOVE 0 TO LINE-COUNT
013500 END-IF
013600 WRITE HATCHUU-REC FROM MEISAI-REC
013700 AFTER 2
013800 COMPUTE LINE-COUNT = LINE-COUNT + 1
013900 READ HSORT-FD
014000 AT END MOVE '1' TO FLAG
014100 END-READ.
014200 主処理-終り.
014300 EXIT.
    
```

COBOLの記述

```

003400 02 商品名
003500 色 形式
003600*****
003700 手続き
003800*****
003900*****
004000 項目加工処理 U1-更新加工
004100*****
004200 計算 U1.予約在庫数量 <-- U1.最高在庫数量
004300 - U1.現在在庫数量
004400*****
004500 項目加工処理 H-キー一致加工
004600*****
004700 転記 L1.日付 <-- U1.日付
004800 転記 L1.PAGE <-- U1.ページ編集
004900*****
005000 項目加工処理 B-キー一致加工
005100*****
005200 転記 L1.仕入先名 <-- U1.仕入先名
005300 転記 L1.電話番号 <-- U1.仕入先電話番号
005400 転記 商品名 <-- U1.商品名
005500 転記 色 <-- U1.色
005600 転記 L1.商品名 <-- 商品名色
005700 転記 L1.商品コード <-- U1.商品コード
005800 転記 L1.仕入数量 <-- U1.予約在庫数量
005900 転記 L1.在庫数量 <-- U1.現在在庫数量
006000 転記 L1.安全在庫数量 <-- U1.安全在庫数量
    
```

□：記述部分

EAGLE/4GLの記述

図9 発注商品一覧表印刷処理ソースリスト 手続き部は、COBOLで48ステップ、EAGLE/4GLで9ステップ(疑問符(?)が出力されたステップ)のコーディングが必要である。

今後、EAGLE/4GLの適用結果をフィードバックし、生産性向上への要件を積極的に反映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) 脇坂, 外: 中小形OS VOS Kの開発(4)第4世代言語EAGLE/4GL, 情報処理学会第38回全国大会講演論文集, 1~2(平1-3)