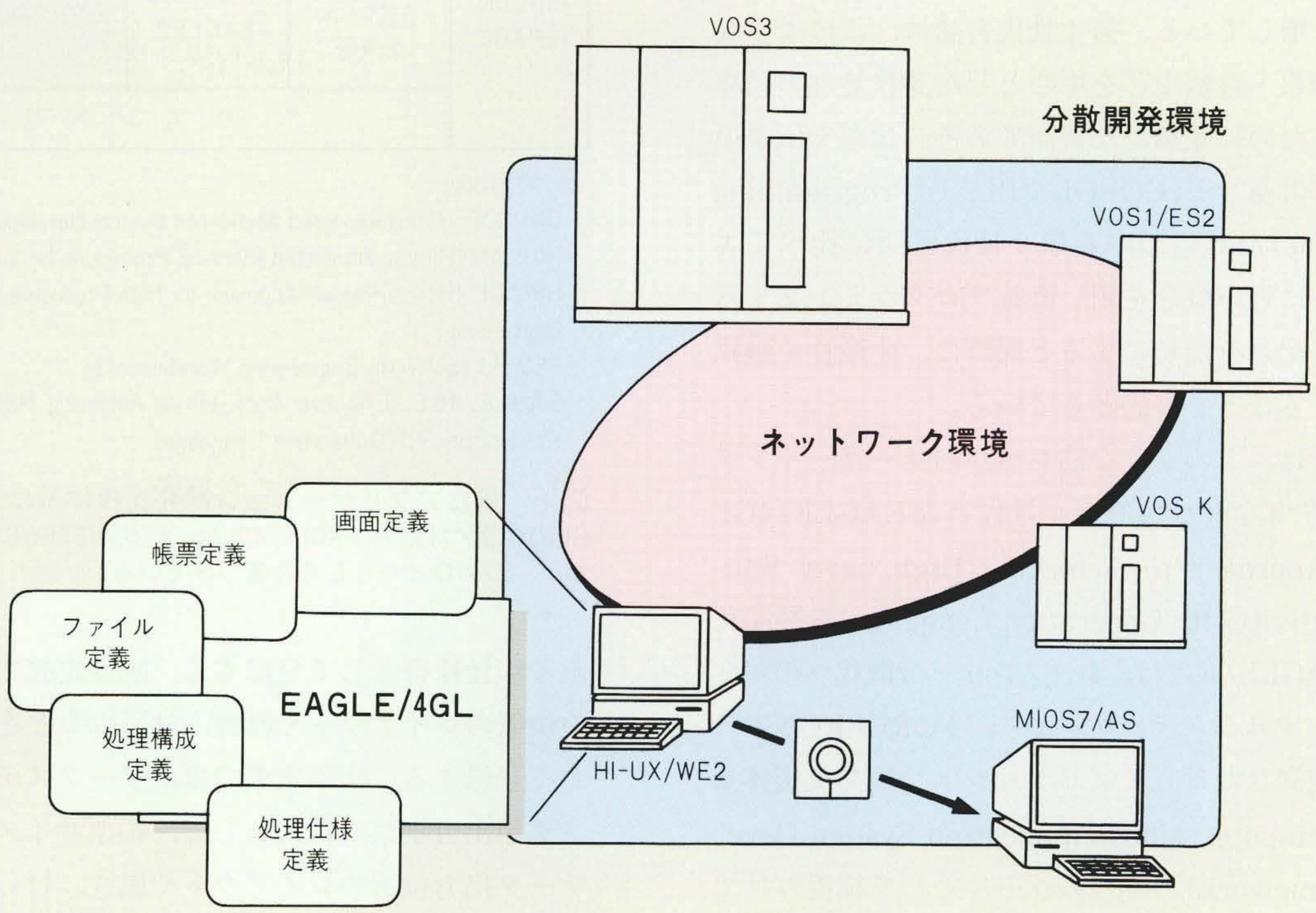
アプリケーションの分散開発を実現する第4世代言語

4th Generation Languages and Distributed Development Environment

西尾高典* Takanori Nishio
秋山美登* Yoshinori Akiyama
今城哲二* Tetsuji Imajyou



注：略語説明
VOS3 (Virtual-storage Operating System 3)，VOS1/ES2 (VOS1/Extended System 2)，VOS K (VOS Kindness)
HI-UX/WE2 (Hitachi UNIX*/Workstation Extended 2)，MIOS7/AS (Multiple Office Information Operating System 7/Advanced System)，EAGLE/4GL (Effective Approach to Achieving High Level Software Productivity/4th Generation Language)
＊ UNIXオペレーティングシステムは，UNIXシステムラボラトリーズ社が開発し，ライセンスしている。

EAGLE/4GL分散開発環境の概念　ワークステーション上のEAGLE/4GLで各種定義を行うことにより，さまざまなターゲットのCOBOLプログラムを開発することができる。

開発期間の短縮，開発・保守コストの低減など，ソフトウェア開発への厳しい要求に対する切り札が，日立製作所が開発した第4世代言語EAGLE/4GLである。EAGLE/4GLは，VOS3，VOS1/ES2，VOS K，MIOS7/ASおよびHI-UX/WE2上のアプリケーションをワークステーション上で開発するクライアントサーバ型の分散開発環境であり，以下の特長を持つ。

(1) 業界標準のX Window System[*1]およびOSF/Motif[*2]準拠の日立Motif上で，統一されたオブジェクト指向操作環境を実現した。(2) ボタンやポップアップメニューなど，GUI(Graphical User Interface)機能付き画面を持ったワークステーション向けアプリケーション開発が可能である。(3) 伝票入力，伝票訂正，マスタメンテナンス，検索という四つの業務に対して対話サブパターンを設け，多種多様な対話系業務プログラム開発の高生産性を実現した。(4) サーバステーション上のリポジトリ情報を画面・帳票定義へ展開する機能，および画面・帳票のテスト表示機能(プロトタイプ技法支援)を実現し，システム設計での生産性を飛躍的に向上させた。

* 日立製作所 ソフトウェア開発本部

1 はじめに

開発期間の短縮，高品質かつ低コストな開発依頼の増加，膨大な既存システムの維持コストの低減など，変化する企業環境に合わせてソフトウェア開発の要求はますます厳しさを増している。第4世代言語は，このような中で伝票・帳票・台帳などを中心とした業務処理の効率を向上させるために登場した言語である。伝票や伝票項目，帳票，業務計算など，COBOLやPL/I (Programming Language One)などいわゆる第3世代言語に比べてエンドユーザーに近い概念を用い直接プログラミングすることによって記述を簡易にすると同時に，生産性を飛躍的に向上させることを目的としている。

日立製作所は，このような高レベルの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)を製品化している。EAGLE/4GLは，オープン化・分散化への対応を柱とするトータルシステムコンセプト“FOREFRONT”[1]に基づいて確立した日立アプリケーション開発支援体系CAPSDF (Computer-aided Application System Development Framework)の一つのツールとして位置づけて(**図1**)，アプリケーションの分散開発環境を実現するものである。

ここでは，EAGLE/4GLの概要，分散開発環境として見たときのEAGLE/4GLの特長，および周辺の日立製品との関連について述べる。

2 EAGLE/4GLの概要

2.1 システム開発手順

EAGLE/4GLは，基幹業務開発用の高生産性言語として開発されたものである[2]。EAGLE/4GLを用いたアプリケーションシステムの開発手順を以下に述べる(**図2**参照)。

(1) システム設計

対象アプリケーションが使用するデータベースやファイル，画面，帳票，バッチ帳票などを定義する。

データベース，ファイル定義では，フィールド名，フィールドの属性，ファイル編成，排他制御区分などをファイル仕様書として登録する。画面定義では，データ出力位置のレイアウトや属性，けい線などを画面仕様書として登録する。帳票定義では，ワークステーションプリンタ用出力帳票の様式として，帳票サイズ，印刷形式，データ出力位置のレイアウトや属性，けい線などを帳票仕様書として登録する。バッチ帳票定義では，帳票サイズ，印刷形式，レイアウト，行種別(見出し行，明細行，合計行)などをバッチ帳票仕様書として登録する。

日立アプリケーション開発支援体系 CAPSDF				
計画技法・開発技法	支援ツール			
HIPLAN/HIPACE	ワークステーション型支援ツール SEWB3	ホスト型支援ツール EAGLE2	第4世代言語 EAGLE/4GL	リエンジニアリングツール
	リポジトリ			

注：略語説明
CAPSDF (Computer-aided Application System Development Framework)
HIPLAN (Hitachi Integrated Planning Procedure for Information Systems)
HIPACE (Hitachi Phased Approach for High Productive Computer Systems Engineering)
SEWB3 (Software Engineering Workbench 3)
EAGLE/4GL (Effective Approach to Achieving High Level Software Productivity/4th Generation Language)

図1 日立アプリケーション開発支援体系におけるEAGLE/4GLの位置づけ　EAGLE/4GLは，本体系でSEWB3などとともに支援ツールのひとつとして位置づけている。

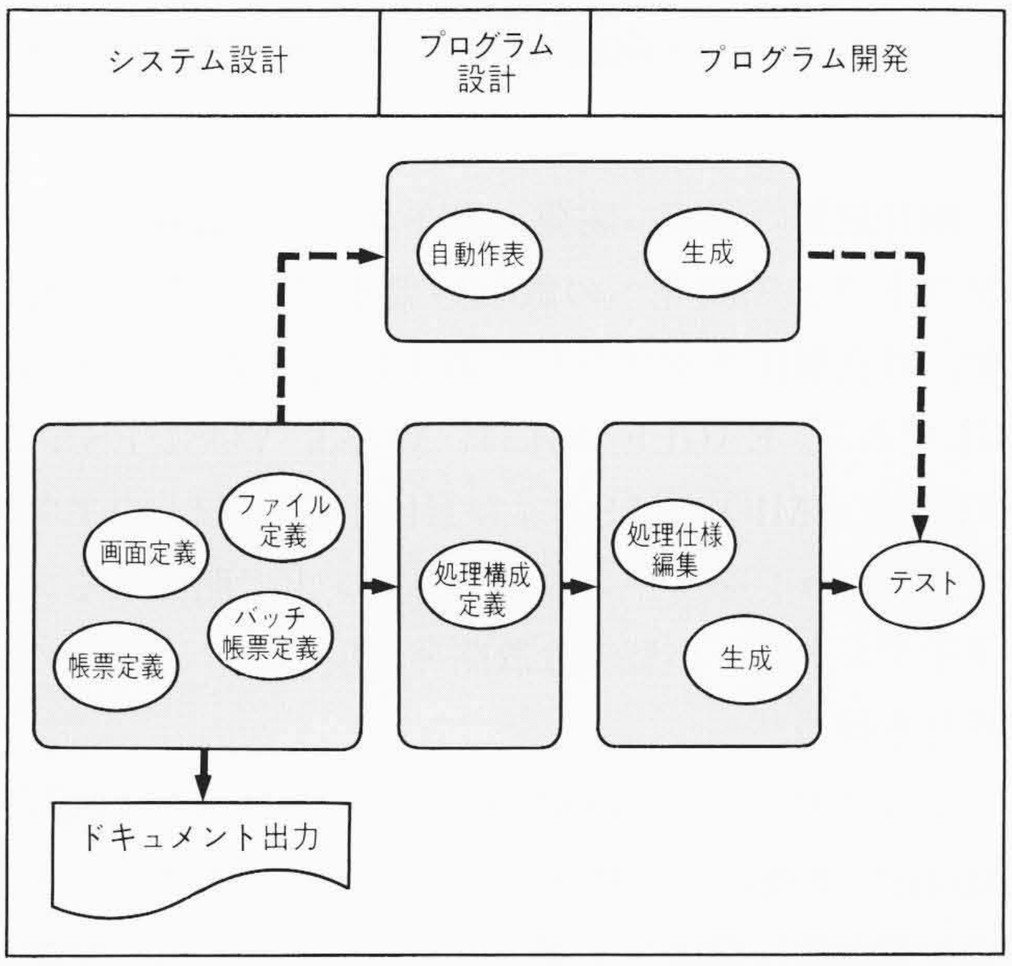


図2 EAGLE/4GLにおけるシステム開発手順　システム開発手順は，自動作表機能を用いた開発(図中破線のフロー)と通常の開発(実線のフロー)に分かれる。

※1) X Window Systemは，米国MIT(マサチューセッツ工科大学)のシステム名称である。
※2) OSF/Motifは，Open Software Foundation, Inc.の商標である。

(2)　プログラム設計

プログラム設計では，まず対象システムの業務目的に従ってEAGLE/4GLが用意した業務パターンの中から適合するパターンを選択する。さらに，パターンに応じた処理構成定義を行う。処理構成定義では，入出力仕様書名〔ファイル，画面，帳票など先の(1)で登録した仕様書〕などが登録される。

(3)　プログラム開発

EAGLE/4GLは，処理構成定義から半完成の処理仕様書(EAGLE/4GLソース)を自動生成する。この半完成処理仕様書に対して，業務固有の処理の追加・修正などの編集を施して処理仕様書を完成させる。EAGLE/4GLは，最終的に処理仕様書からCOBOLプログラムを自動翻訳することにより，対象アプリケーションを完成させる。

2.2　プロトタイプ技法

画面，帳票定義では，プロトタイプ技法を採用している。ワークステーション画面上で，ラフスケッチした画面と帳票のイメージをテスト表示することができる。従来は，プログラムの作成，テスト終了後に初めて画面や帳票をユーザーに確認してもらうという手順を踏んでいたため，手直しがあるとシステム設計の段階にまでさかのぼって画面を再設計する必要があった。このプロトタイプ技法ではシステム設計の段階で，文字の大きさ，レイアウト，カラーなどの詳細仕様に至るまで適宜ユーザーに確認してもらえるため，早期に画面や帳票イメージを確定することができ，生産性が向上する(**図3**)。

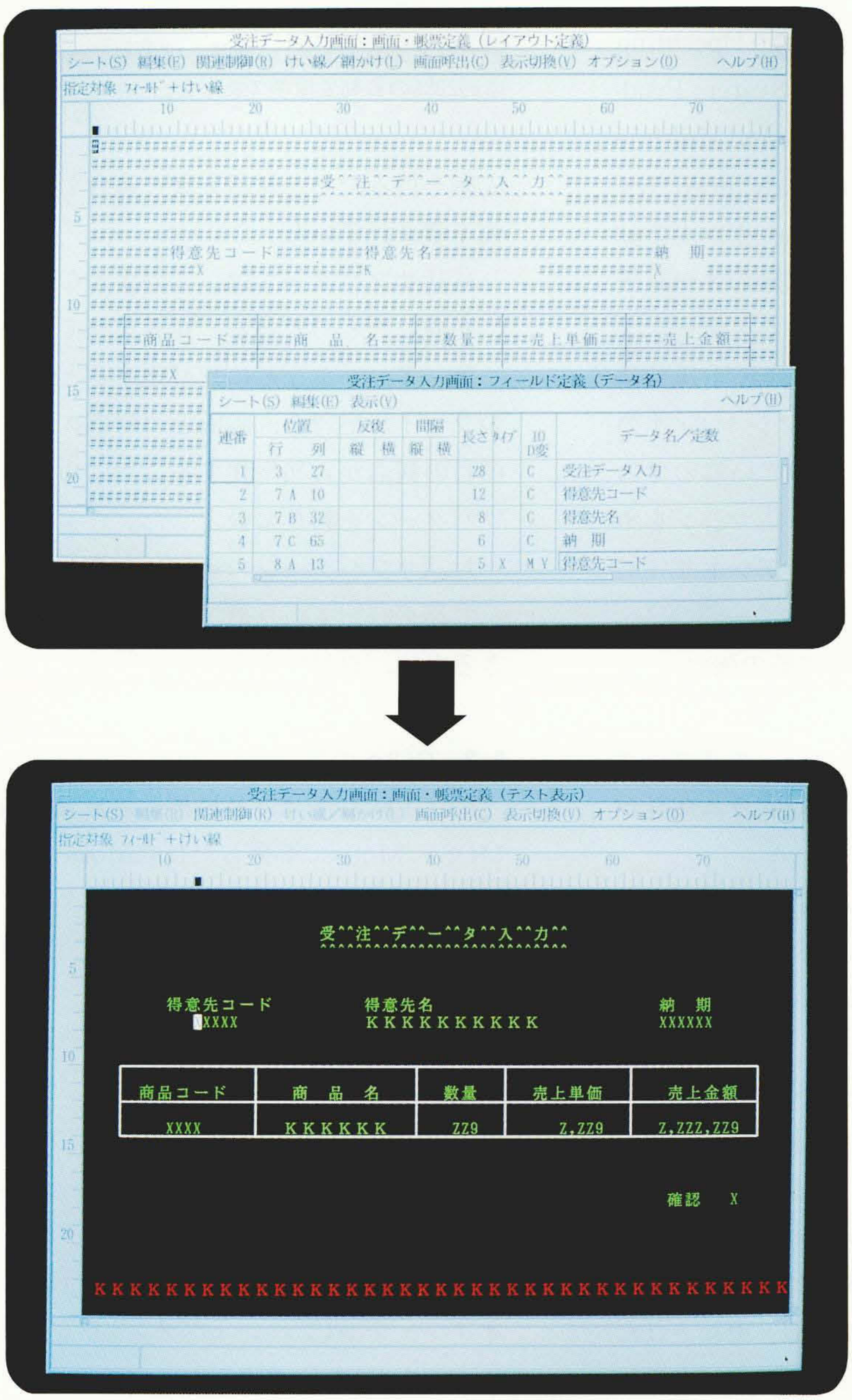


図3　テスト表示機能によるプロトタイプ技法支援　画面のレイアウト定義，各種詳細定義に基づいて自動的に画面イメージをテスト表示することにより，設計結果の早期確認が行える。

2.3 自動作表機能

データベースやファイルの内容を直接印字する一覧表や，小計や合計を印字する集計表などの帳票を出力する業務のために自動作表機能がある。この機能により，データベースやファイルの形式表示画面上で，帳票として出力したい項目を選択するだけでCOBOLプログラムを完全自動生成することができる。すなわち，処理構成定義や処理仕様書編集などの処理が不要となる。

3 EAGLE/4GLの特長

EAGLE/4GLの基本的な機能については，文献2)を参照願う。この章では，分散開発環境上のツールとしてEAGLE/4GLを強化した部分，およびその特長について述べる。

3.1 マルチターゲットの分散開発環境

EAGLE/4GLでは，VOS3(Virtual-storage Operating System 3)，VOS1/ES2(VOS1/Extended System Product 2)，VOS K(VOS Kindness)，MIOS7/AS(Multiple Office Information Operating System7/Advanced System)，HI-UX/WE2(Workstation Extended 2)上で動作するアプリケーションの開発を，それぞれのオペレーティングシステムに依存することなくワークステーション上で開発することができる。同一のEAGLE/4GL言語ソースから，それぞれのターゲットマシン向けのCOBOLプログラムを生成する機能を持っているのである。また，ワークステーションとターゲットマシンをネットワークで接続することにより，クライアントサーバ型の分散開発環境を実現することができる。

3.2 対話サブパターンによる対話業務定義の強化

従来のEAGLE/4GLによる生産性向上は，COBOLで開発を行った場合に比べて開発工数比でバッチ系プログラム開発が平均5～6倍程度，対話系プログラム開発が平均2～3倍程度と評価されている。

バッチ系プログラム開発では，自動作表機能などによって生産性向上が期待できる。一方，対話系プログラム開発では，従来からサポートしている対話パターンが基本的な画面遷移だけしか提供しないため，業務に合わせた処理を追加する必要があり，これが生産性向上のネックになっている。**図4**に示すとおり，処理仕様書の自動生成率がバッチ系では約80%なのに対し，対話系では20～30%前後であることからも明らかである。ここで自動生成率とは，処理構成定義から生成される(半完成)処理仕様書のステップ数と，完成までに追加・更新されたステップ数の比率のことである。対話系での高い自動生成率を得るため，対話業務に適合したパターンの提供が強く望まれていた。

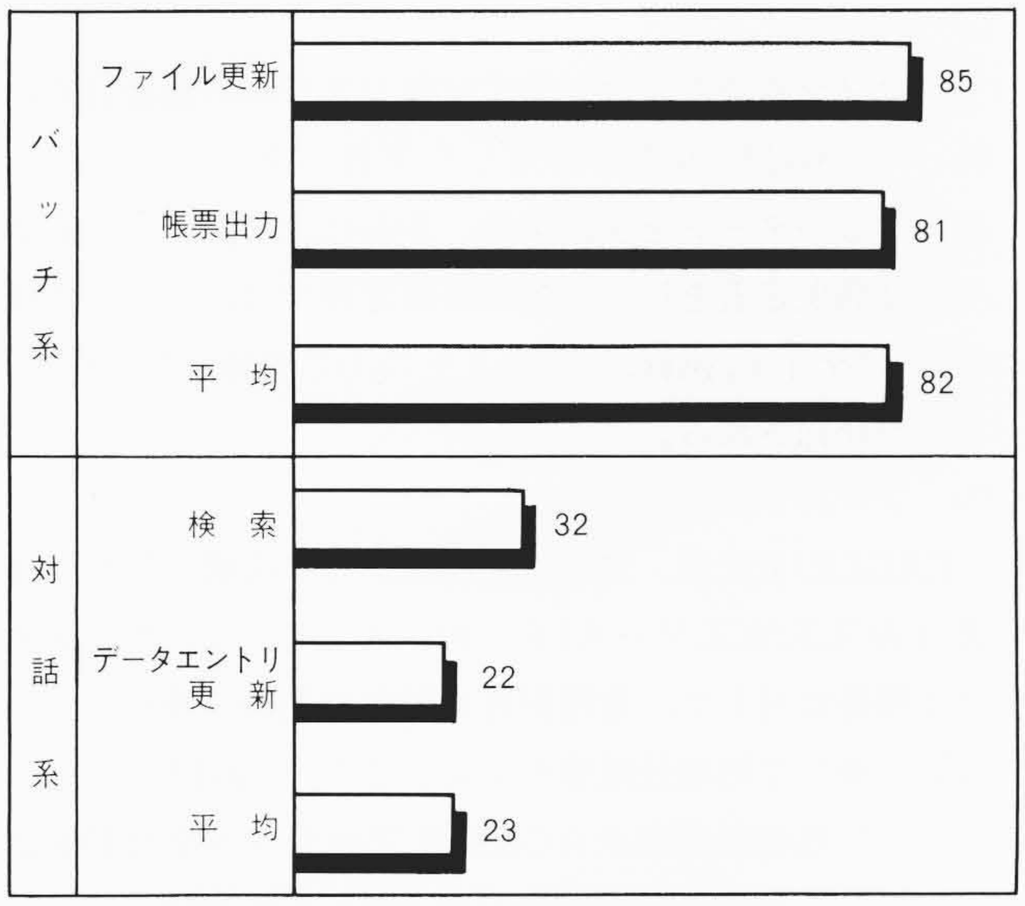


図4 処理仕様書の自動生成率データ ファイル更新や帳票出力などのバッチ系は約80%，検索やデータエントリなどの対話系は約20%の自動生成率であった。

対話サブパターンは，このような要求にこたえるために開発したものである。対話業務での画面処理を，一品一葉，多品一葉などに類型化すると同時に，対話業務でのファイル処理を，画面との対応，入出力処理などの4項目の観点から類型化することによって**図5**に示す4サブパターンを新たに設定した。

対話サブパターンの適用により，半完成の仕様書に対して従来行っていた追加・更新処理の約5割を自動化できるため，自動生成率を約60%にまで向上させることができると考えられる。

3.3 GUIへの対応

EAGLE/4GLは，業界標準のX Window SystemおよびOSF/Motifに準拠した日立Motif上で構築している。オブジェクト指向のユーザーインタフェースを実現する統一的な操作環境で，画面・帳票や処理構成などの定義ができる。

さらに，ワークステーション向けのアプリケーション開発に対応するため，ボタンやポップアップメニューなどを備えたウインドウ風の画面の定義を可能にした(**図6**参照)。同図中，GUI(Graphical User Interface)定義用ボードによって，すでにレイアウト定義された画面に付加すべきGUI機能の定義を行う。画面中に付加するオブジェクト(プッシュボタン，ラジオボタン，スクロールバーなど)を定義することができる。

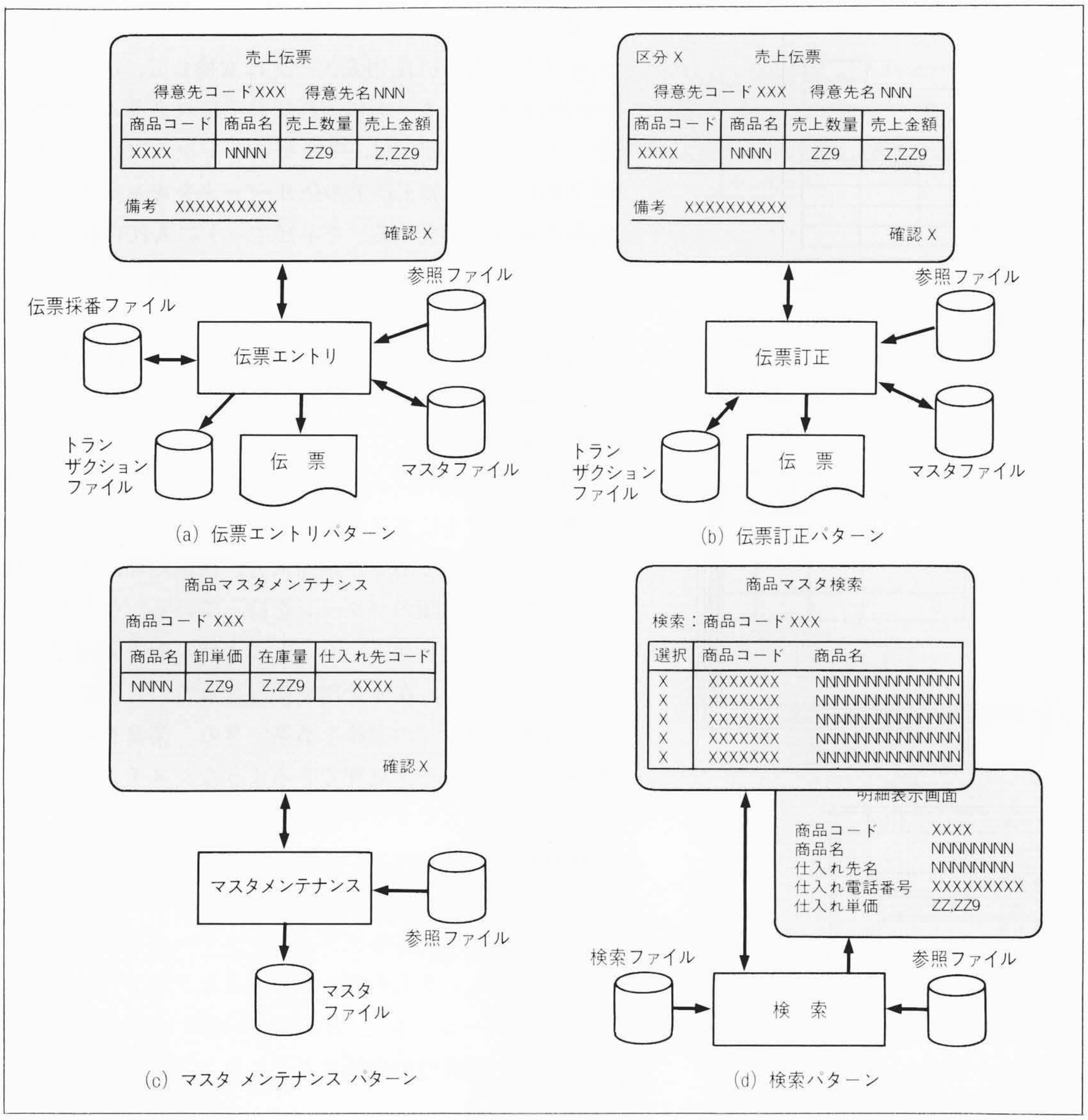


図5　対話サブパターンの概要　4種類の対話サブパターンを前提とするシステム構成を示している。

3.4　リポジトリ支援

サーバステーション上に構築されたリポジトリ内には，データ項目辞書として仕様情報が定義蓄積されている。クライアント側のEAGLE/4GLの画面・帳票定義で，これらのデータ項目情報を参照・利用することができる。

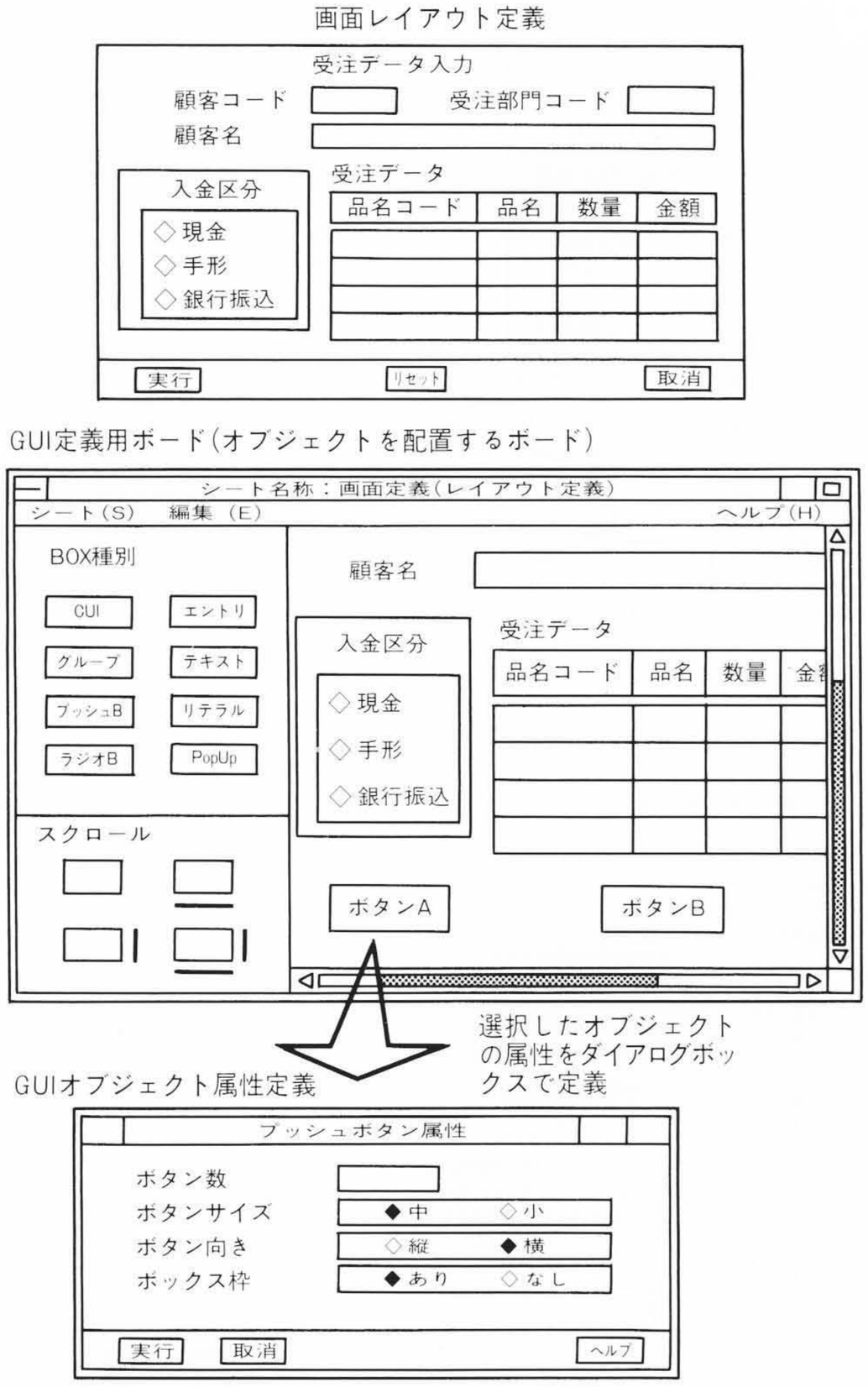
これにより，画面・帳票定義，ファイル定義などでの項目名称や属性などの再度入力を不要にすることができるので，項目定義を容易にすると同時に，データ項目の統一を図ることができる。また，各種定義，仕様書間関連の一覧をクロスリファレンス印刷でき，保守性の向上を図ることができる。

4　関連製品

第4世代言語には，非定型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)が基幹業務処理を開発するための高生産性言語と呼ばれるものまである。上述のEAGLE/4GLは，後者に位置づけられるものである。日立製作所にはこのほか，前者に位置づけられる製品として，OFIS/POL(Office Automation and Intelligence Support Software/Problem Oriented Language)，OFIS/FORM，ETOILE/OP(Effective Object Oriented Language for End Users/Office Processor)[3]などがある。これらの簡易言語的な第4世代言語でも，分散処理への対応が進んでいる。以下に，これらのいくつかについて述べる。

4.1　OFIS/POL

OFIS/POLは，パーソナルコンピュータ(以下，パソコンと略す。)からワークステーションまですべての機種上で稼動する，作表・集計機能を持った事務処理用ソフトウェアである。高精度の計算，表エリア不足の防止，べ



注：略語説明 GUI (Graphical User Interface)

図6 GUI機能付き画面の定義 画面レイアウト定義を行った画面について，GUI定義用ボードでボタンやポップアップメニューのレイアウトを定義し，GUIオブジェクト属性定義では，個々のオブジェクトの属性を定義する。

クトル概念を持つ計算式，ホストマシンとの連携がよい，などの特長を持つ。

OFIS/POLは単なる非定型業務支援だけでなく，分散基幹業務のインタフェース開発のためのプロトタイプづくりを支援することもできる。情報システム部門のメンバーとエンドユーザーとが協力して，OFIS/POLを用いて書式設計や計算式設計など業務の基礎づくりを行う。情報システム部門側では，ホストデータベースに蓄えたデータを検索しPOL用表データに変換して，ホストマシンの電子キャビネットに入れておくプログラムを作成する。エンドユーザーは，キャビネットからPOLデータを取り出したり，加工結果のPOLデータをキャビネットへ入れる部分を開発する。キャビネットに入れられたPOLデータはホスト側で一括処理される。情報システム部門側は，POL表の作成作業だけでよいし，エンドユーザー側はホストの出力リストをもとにしたPOL表の作成作業から解放される。

4.2 OFIS/FORM

OFIS/FORMは，小規模な定型業務をコンピュータで処理できるようにするエンドユーザーコンピューティングソフトウェアである。画面入力，検索処理，加工処理など全部で16種類のパターンを持っている。これらのパターンを組み合わせていくことにより，販売管理での売り上げ，仕入れ，在庫管理や給与計算での月例給与，賞与，年末調整などの業務を構築できる。業務を行う人が簡単にコンピュータ処理できるようなシステムを作ったり，必要に応じてすぐ手直しすることができる。

OFIS/FORMはスタンドアロン型で使用する以外に，クライアント側のOFIS/FORMからサーバ上のデータベースをアクセスするクライアントサーバ型の形態，ホストコンピュータ上のデータベースをダウンロードし，ワークステーション上のOFIS/FORMでデータ処理を行うホスト連携型の形態などをとることができる。

5 おわりに

日立製作所での第4世代言語の考え方と分散開発環境への対応について述べた。この論文で扱ったツールは，いずれもホスト中心の時代に生まれている。その基本概念は当初の理念を保ち続けているものの，オフィスコンピュータ，さらにはワークステーション，パソコンの時代へのめまぐるしい移行とともに，新しい機能の追加，分散化への対応と，急速に変容を遂げつつある。その状況を少しでも伝えることができたならば幸いである。

参考文献

1) 森，外：新コンセプト“FOREFRONT”，日立評論，**74**，5，376～380(平4-5)
2) 今城，外：VOS K第4世代言語“EAGLE/4GL”，日立評論，**71**，11，1119～1124(平1-11)
3) 土屋，外：オペレーティングシステム“MIOS7/AS”用第4世代言語“ETIOLE/OP”，日立評論，**71**，11，1145～1152(平1-11)