

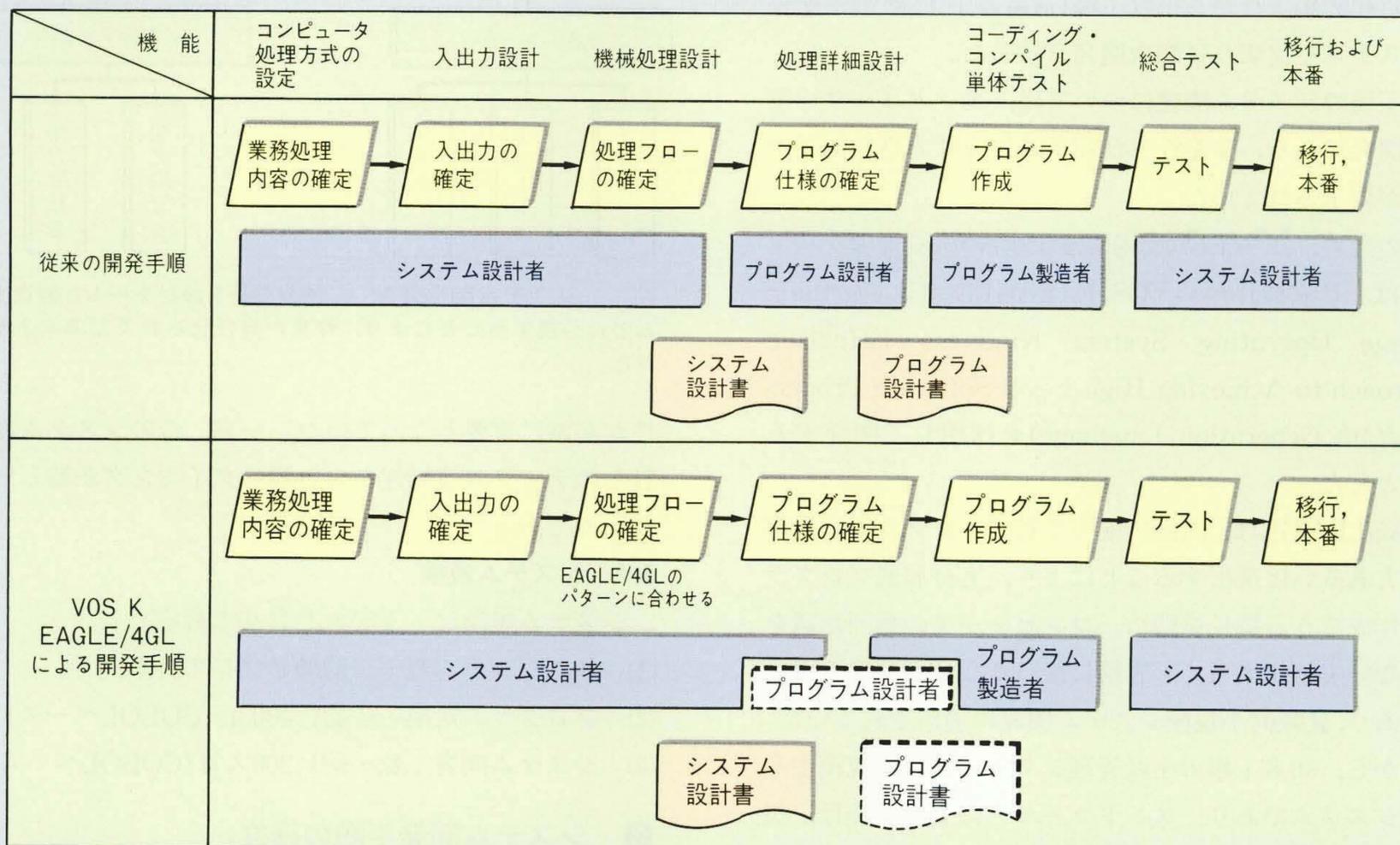
第4世代言語を適用したシステム開発手順の確立

—日本ガイシ株式会社—

Production Control Systems Development Using 4th Generation Language

林 英之* *Hideyuki Hayashi*

廣田忠夫** *Tadao Hirota*



注：略語説明 VOS K (Virtual-storage Operating System Kindness), EAGLE/4GL (Effective Approach to Achieving High Level Software Productivity / 4th Generation Language)

第4世代言語を適用したシステム開発工程 EAGLE/4GLのメリットを生かした大規模システム向け開発手順を確立したことにより、従来の開発方法と比較して効率のよい開発が可能となった。

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心要であったり、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、信頼性や生産性が今一步伸びていなかった。

日立製作所はこの問題を解決するために、システ

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)を提供している。

これまで、大規模システム開発にEAGLE/4GLを適用した日本ガイシ株式会社(以下、日本ガイシと言う。)をはじめとして、他システムの開発をとおして大規模開発を考慮したシステム開発手順を確立した。この論文では、日本ガイシの事例を紹介する。

* 日本ガイシ株式会社 ** 日立製作所 情報システム事業部

1 はじめに

日本ガイシ株式会社(以下、日本ガイシと言う。)では、全社情報システム構築の展開方針「経理・財務システムの全面再構築および事業部(製品)特性にあった分散処理システムの順次構築」に基づき、分散処理システムとして、昭和60年1月から小牧工場(送電碍子生産)の生産管理システムを皮切りに順次開発を始めた。

各工場のシステム開発については、エンドユーザー主体の開発であり、生産性・信頼性・保守性の良い開発ツールが要求されていた。

そのため、知多工場(変電碍子生産)の生産管理システムでは、日立製作所のVOS K-EAGLE/4GL(Virtual-storage Operating System Kindness - Effective Approach to Achieving High Level Software Productivity/4th Generation Language)を採用して開発することとした。

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により、実行形式プログラムが作成できる特長を持つ。コンピュータの専門知識を持たない実務担当者でも容易に使いこなせるソフトウェアであり、比較的小規模システム開発に適用されていた。

しかし、知多工場の生産管理システムは、大規模生産管理システムであり、エンドユーザーによる開発は、規模・開発期間の面で不可能と判断した。

そのため、開発には日立製作所も参画することとなったが、開発要員はピーク時約70人となり、**図1**に示すような設計と製造を分離した開発体制をとった。また、通常EAGLE/4GLでは、ラフスケッチと設計メモをもとに試作して修正を加えながら業務仕様を決定していくプロトタイプングアプローチをとるが、今回は大規模開発を行う必要があったため、従来の開発手順にEAGLE/4GLのメリットを生かしたシステム開発手順を確立し、開発を行うこととした。

ここでは、EAGLE/4GLによる大規模システム開発でのシステム設計手順の確立を中心に述べる。

2 システム概要

2.1 システム概要

このシステムは、受注～生産計画～作業指示～各種実績収集～在庫管理～品質管理に至る総合生産管理システムであり、大規模でかつ複雑なシステムであることから各サブシステム間の整合性とデータ処理ロジックの統一

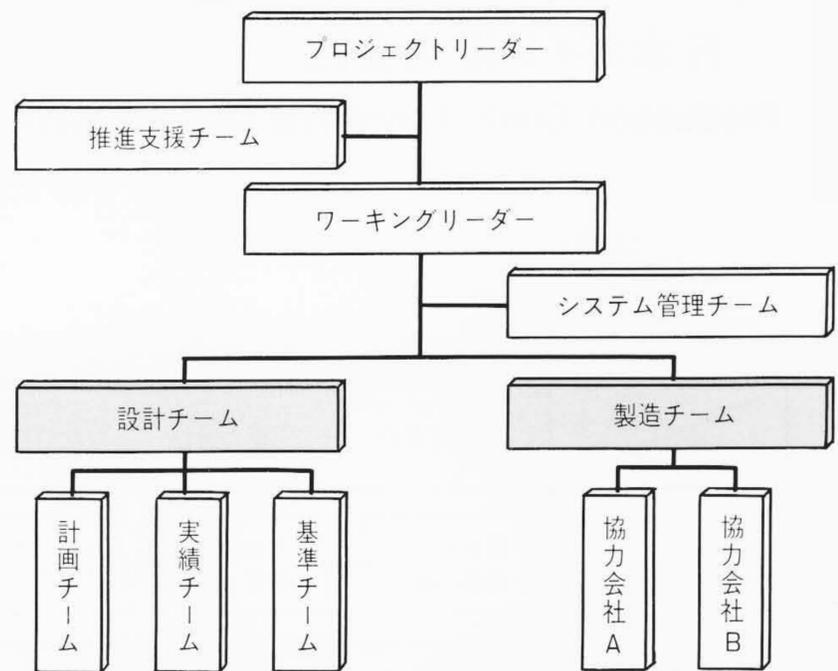


図1 システム開発体制 開発要員を設計チームと製造チームとに分離することにより、作業が専任化されて効率向上が図れた。

性が非常に重要となっていた。一方、このシステムは本社とのデータの連動性から処理のタイミングが難しいシステムであった。

2.2 システム規模

システム規模は、下記のとおりである。

- (1) サブシステム数……約30サブシステム
- (2) プログラム規模……約1,400 ks(COBOLベース)
- (3) システム開発工数…約1,200人月(COBOLベース)

3 システム開発手順の確立

EAGLE/4GLによるシステム開発では、高生産性、高品質を得るためにシステム設計時からEAGLE/4GLの処理パターンとその機能を十分理解して設計を行う必要がある。そのため、システム設計時に、EAGLE/4GLの処理パターンにあったプログラム分割までを行えば、プログラム1本ごとが単一機能になり、プログラムごとのユーザー固有ロジックを同時に検討し、システム設計者が引き続きプログラムを製造することが、効率のよい開発となる。

しかし、大規模システムでは、段階的稼動を行うケースが多く、設計と製造を同一人が行うことは難しい。

以上のことから、従来のシステム開発と異なり、プログラム設計の一部をシステム設計に統合し、プログラム設計者によるプログラム設計フェーズを除外したシステム開発手順を確立し、それに伴うシステム設計ワークシートを作成した。システム開発手順を**図2**に示す。

4 システム設計手順の確立

システム設計手順と設計ワークシートの体系は次に述

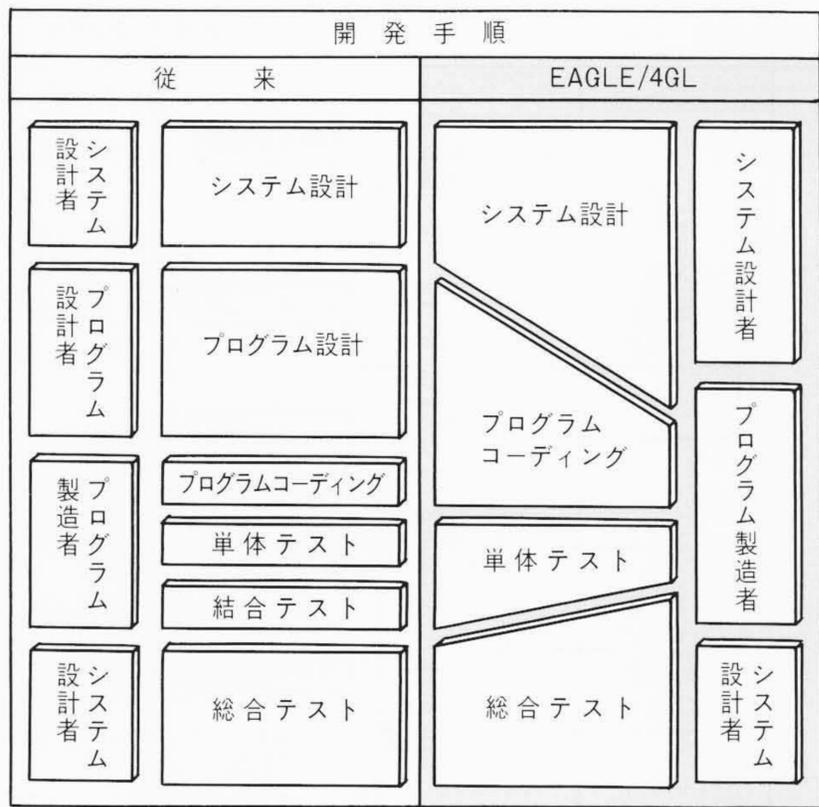


図2 システム開発手順 システム設計とプログラム設計を一体化することにより、設計工程およびプログラム製造工程での生産性が大幅に向上した。

べるとおりである。

システム設計手順は図3に示すようにEAGLE/4GLの対話処理とバッチ処理および共通処理とに分けた。

ここでは、EAGLE/4GLのパターンを十分理解したうえで処理設計を行う。EAGLE/4GLは、固定パターン方

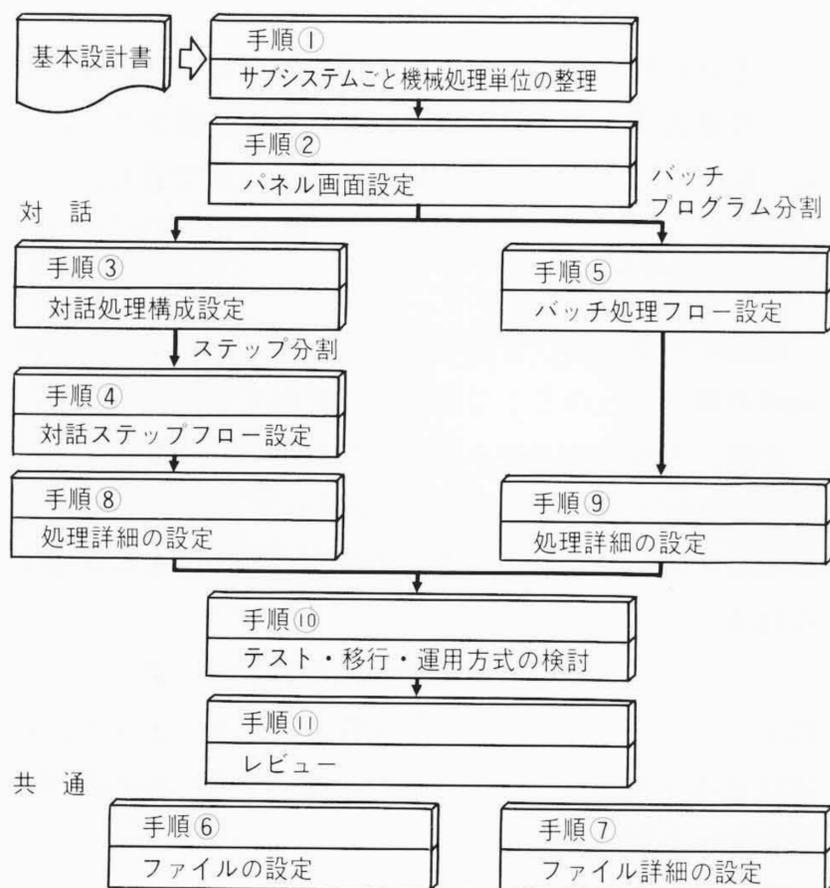


図3 システム設計手順 システム設計手順は、対話処理部分とバッチ処理部分および共通部分とに分け、ドキュメントも同様に体系化した。

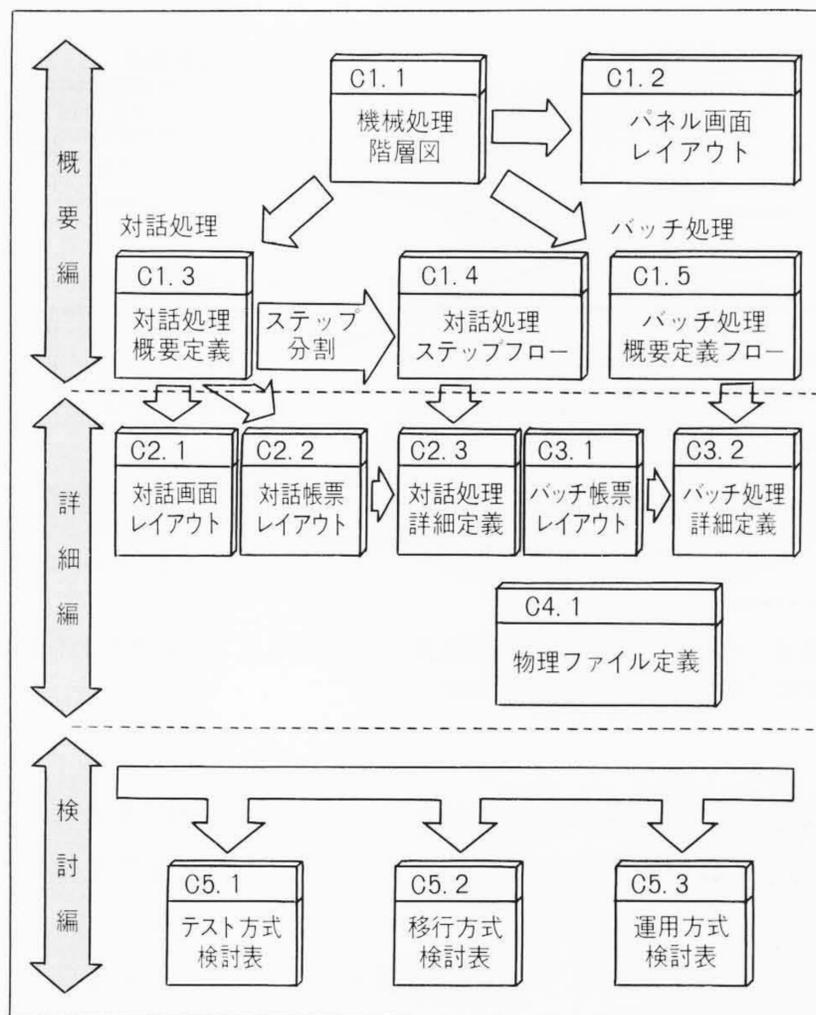


図4 システム設計ワークシート体系 処理機能を中心とした概要編、プログラムの機能を中心とした詳細編、テスト・移行・運用方式を中心とした検討編から成る。

式(バッチ2パターン、対話1パターン)のため、あいまいなまま処理設計を行うと、プログラム製造工程からの戻り作業が発生し、かえって工数増大の原因となる。したがって、バッチと対話処理の手順を明確に分けることが必要となった。

今回のシステム設計のドキュメント体系は、すでに日本ガイシプロジェクトで業務の基本設計は完了していたことにより、システム設計の機能処理設計からプログラム設計、およびテスト・運用・移行設計を対象とした。

設計手順に対応したシステム設計ワークシートは機械処理単位の整理からファイル設定までを行う概要編、ファイル詳細設計から画面・帳票レイアウトの設計、および処理詳細設計を行う詳細編、総合テスト方式、移行方式、運用方式の検討およびレビュー報告を行う検討編から構成されている。システム設計ワークシートの体系を図4に示す。その他、品質向上、生産性向上を目的に、図5に示すように各種のシステム設計標準書を作成した。

5 システム設計手順適用上の考慮点

対話処理に対する考慮点は次に述べるとおりである。EAGLE/4GLで提供されている対話処理パターンは、

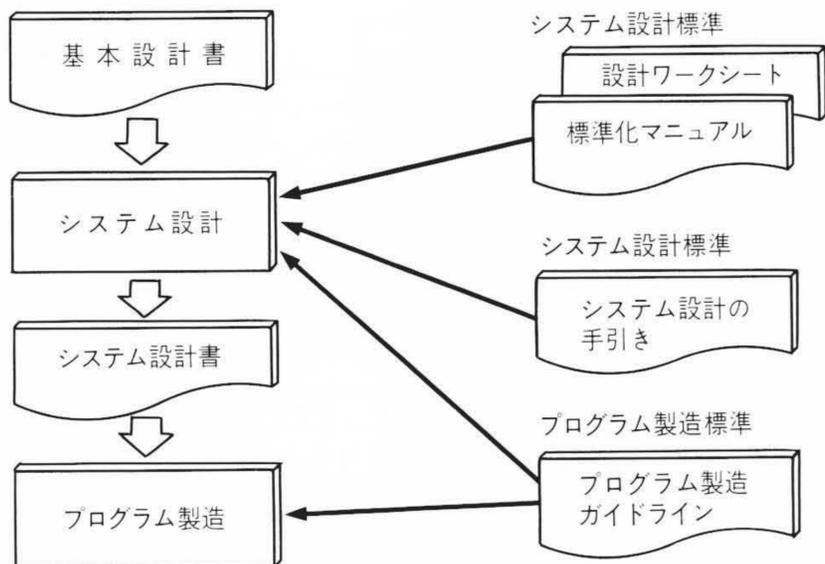


図5 システム設計標準の体系 品質向上、生産性向上を目的に、各種の設計標準書を作成した。

処理内容が設計者の自由に任されている部分が多いので設計品質が均一にならない。したがって、生産性や信頼性が低下する可能性があるため、設計ワークシート上に図6に示す例のような対話処理ステップフローを事前に印刷することとした。

以下にその対話標準パターンを示す。

- (1) 削除処理パターン..... 1種類
- (2) 登録処理パターン(繰り返し含む)..... 2種類
- (3) 変更処理パターン(繰り返し含む)..... 2種類
- (4) 検索処理パターン..... 1種類

対話処理のパターン化により、設計者は対話処理フローを作る必要がなくなり、パターンを選択しプログラムに固有な情報を記述するだけとなった。

これにより、設計品質の向上を図ることができた。

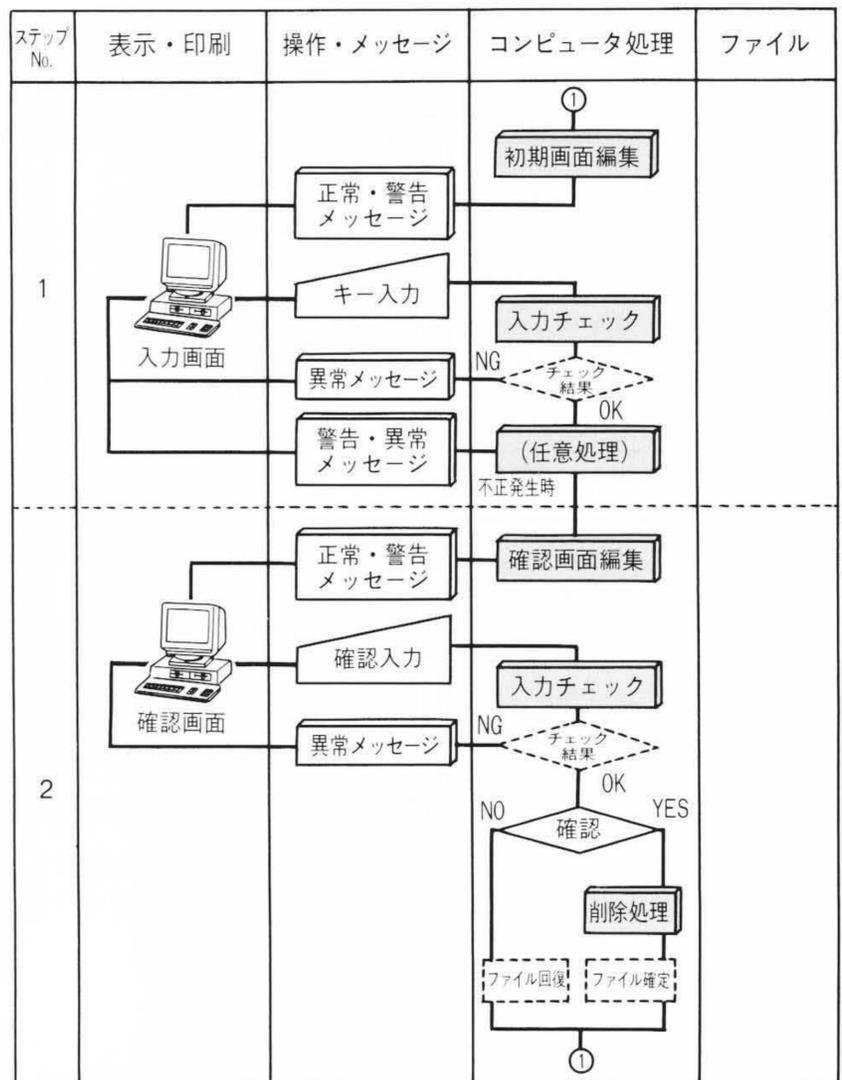
6 成果

今回、EAGLE/4GLを適用して、大規模システム開発のシステム設計手順の確立を図ることができた。

これまで、EAGLE/4GLによるシステム開発では、EAGLE/4GLの特長の一つであるプロトタイプング手法により、システム設計工程およびプログラム製造工程での生産性が飛躍的に向上すると言われている。

しかし、ある程度以上のシステム開発規模の場合は、システム開発参加者が多く、プロトタイプング手法の場合、対話形式で作業が行われるので、端末台数がネックとなるなどの問題がある。

EAGLE/4GLでの大規模システムの開発は、当初、難



注: [] は自动生成後のユーザーロジック追加記述部分を示す。

図6 EAGLE/4GL対話処理ステップフローの例 対話処理ステップフローを事前に印刷することにより、設計者は対話処理フローを作る必要がなくなって、パターンを選択してプログラムに固有な情報を記述するだけとなり、設計品質の向上を図ることができた。

しいと言われていたが、この手順の確立により、日本ガイシ生産管理システムの場合のような大規模システムでも、EAGLE/4GLの適用で開発することができた。

7 おわりに

EAGLE/4GLは、生産性が高く、よりエンドユーザー向けの開発ツールとして位置づけされている。

今回、大規模システム開発の設計手順の確立により、今後のEAGLE/4GLによる大規模システムの開発手順を明らかにしたことは、きわめて有意義であったと考えられる。

今後は、日本ガイシ用に作成した設計ワークシート、標準パターンなどを品質、生産性の観点から、より良い設計技法への改良・改善を進め、他ユーザーへの適用に努めていく考えである。