

アプリケーションシステムの効率的設計技法 “HIPACE”

Software Engineering Methodology for Development of Application Systems “HIPACE”

近年、コンピュータを用いたアプリケーションシステムの開発に当たっては、対象業務の複雑化、大規模化による開発コストの増大、及び環境の変化に応じたシステムの変更コストの増加ということが問題となっている。日立製作所はこのような問題に対処するため、アプリケーションシステムの効率的設計技法“HIPACE”を開発した。

“HIPACE”は、アプリケーションシステム開発のニーズ分析から運用保守に至るまでの一貫した標準手順と、これに対応したワークシートを設定した。また手順に基づいて、システム開発を効率的に行なうための技法と支援ツールの開発により、システム開発をより効果的に行なうことができる。

宮副英彦* *Hidehiko Miyazoe*
津田道夫* *Michio Tsuda*
小林正和** *Masakazu Kobayashi*
前澤裕行** *Hiroyuki Maezawa*
堀内 一*** *Hajime Horiuchi*

1 緒 言

近年、コンピュータを用いた情報システムの開発、運用では、各ユーザーとも開発工数、保守工数の増大に悩まされている。このため、システム開発での生産性の向上が急務とされている。

生産性向上についてのこれまでの多くの施策は、主としてプログラミングやテストに関するソフトウェア生産技術に重点が置かれてきた。確かに、システムによってはプログラミングやテストの段階に70～80%の工数が占められるものもあり、こうしたソフトウェア生産技術を重点課題とすることは避けられない。しかし、ここ数年、システムエンジニアの慢性的な不足の中で、「エンドユーザーの真のニーズに合致した質の高いシステムのタイムリーな実現」がなかなか達成されないことが大きな問題となってきている。この問題を解決するためには、システム開発作業の中でも前工程に当たる分析や設計の作業を有効かつ迅速に進めるための方策が不可欠であり、その開発が強く望まれてきている。

“HIPACE”(Hitachi Phased Approach for High Productive Computer System's Engineering)は、こうした要望にこたえて開発され、体系化されたアプリケーションシステムの効率的開発技法の総称であり、特にシステム開発作業の前工程に重点を置いたものである。

本稿では、以下、この“HIPACE”の開発背景、ねらい及びその概要について紹介する。¹⁾

2 “HIPACE”のねらいと構成

コンピュータを用いた情報システムの開発では、

- (1) 経営層、利用部門からのニーズの多様化と速応性の要求
- (2) システム開発工数の増大と長期化
- (3) オンラインシステムの急増とコンピュータシステムへの依存性の増大に伴う高信頼性の要請
- (4) 既存システムの巨大化に伴う変更、保守コストの増大
- (5) コンピュータ利用コストに占める人件費比率の増大と有能なシステムエンジニアの慢性的不足

といった数々の問題が存在する。^{2)~3)} このため“HIPACE”では、次の目標を掲げ、そのための方法論を具体化し、標準手順、技法及び支援ツールの開発を行なってきた。

(1) エンドユーザー参加と前工程重視のシステム開発の実現

今後のシステム開発には、エンドユーザーの参加は必須と言える。それは集中開発体制の限界を打破するためにも、またエンドユーザーのノウハウを開発するシステムに注入するためにも望まれることである。“HIPACE”では、エンドユーザーをコンピュータの詳細を知る必要のない立場としてとらえ、(a) エンドユーザーとシステムエンジニアとの間のコミュニケーション手段の確保(SDF: Structured Data Flow Diagram)、(b) システム開発でのユーザー責任項目の明確化、(c) エンドユーザー参加局面(分析・計画段階)とコンピュータ化局面(設計・製造段階)の分離、を図った。また、これらの各工程に標準工程を定め、作業内容をワークシート化した(SPDS: Standard Procedure to Develop System)。これにより、システムエンジニアにより異なる作業の進め方、内容を均一化し、レビューや管理を容易にした。

(2) システム要求確定に関する技法の確立

システム保守コストの増大及びシステム開発混乱の多くは、不確定なシステム要求に基づいた設計、製造に起因する。それゆえ、システム仕様の十分な検討と確定を図る技法が要求されてきた。“HIPACE”はこれに対し、(a) ユーザーニーズを利害者関与関係でとらえながら分析する技法(PPDS: Planning Procedure to Develop System)、(b) システム計画段階をエンドユーザーと相互に進めるための構造分析技法、(c) 確定された要求仕様を記述し、仕様に内在する矛盾及び不整合を検出するための支援ツールなどを用意した。

(3) システムライフサイクルコスト削減のための技法の体系化

システムに関するコストをそのライフサイクル全体でとらえる必要性については、既に十分に認識されつつある。システム保守コストがライフサイクルコストの70%を超えるものとなることはよく知られている。“HIPACE”はこのような問題

* 日立製作所ソフトウェア工場

** 日立製作所システム開発研究所

*** 日立製作所コンピュータ事業本部

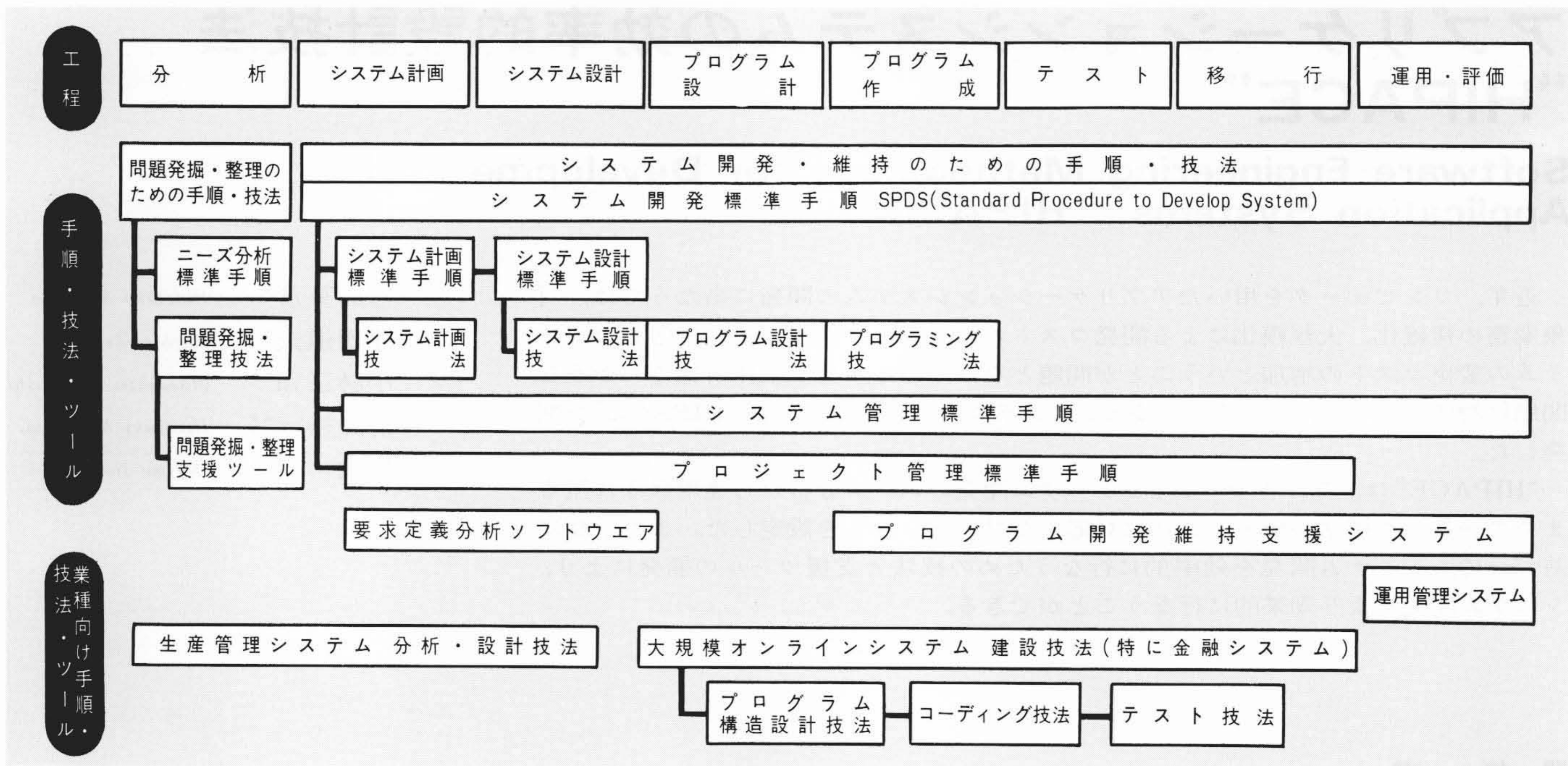


図1 “HIPACE”の構成要素の位置付け “HIPACE”(Hitachi Phased Approach for High Productive Computer System's Engineering)を構成する手順、技法、ツールの位置付けを示す。

点を重視し、生産性向上及び保守性向上のために、(a)構造分析技法(HIPACE-SA:“HIPACE-Structured Analysis”), (b)構造設計技法 (HIPACE-SD:“HIPACE-Structured Design”), (c)構造化プログラミング技法 (HIPACE-SP:“HIPACE-Structured Programming”)を開発した。

(4) システム開発、運用における管理統制の確立

情報システムでの問題点の多くは、開発及び運用での管理統制の不備にあり、その改善充実は急務と言える。そこで“HIPACE”は、(a)システム開発過程の管理を高めるため、開発過程を区分け(フェーズ分け)して、管理活動が着実に介入するフェーズドアプローチを標準手順に組み入れた。また、(b)プロジェクト相互及びユーザー相互の調整と管理統制、更に、開発されたシステムとシステム資源の管理を確立するシステム管理業務についての標準手順、及び技法も用意した。

図1に“HIPACE”を構成する諸要素を示す。

以下、その主なものについて紹介する。

3 システム開発標準手順

3.1 システム開発標準手順の考え方

システム開発標準手順(SPDS)は、業務システムの開発、運用及び保守を効率よく行なうために、システム開発作業の手順、管理の手順を標準化し、一貫した思想に基づいて体系化したものである。

SPDSのねらいは次にある。

- (1) システムライフサイクルコストの低減
- (2) エンドユーザー参画型システム開発の実現
- (3) プロジェクト管理の容易化
- (4) 共有資源管理の充実

図2はSPDSの適用概念を示したものである。SPDSは四つの標準手順から成り、作業及びその管理を標準的ワークシートに従って行なうことができる。これらは、

- (1) エンドユーザー自らがシステムの基本仕様を決め、システムの効果やコストの見積りなどシステム計画を確実に行な

える手順。

- (2) 業務システムの計画立案、設計工程での誤りを早期に発見するために、すべての作業に着実なレビューを行なうための手順。

を提供している。

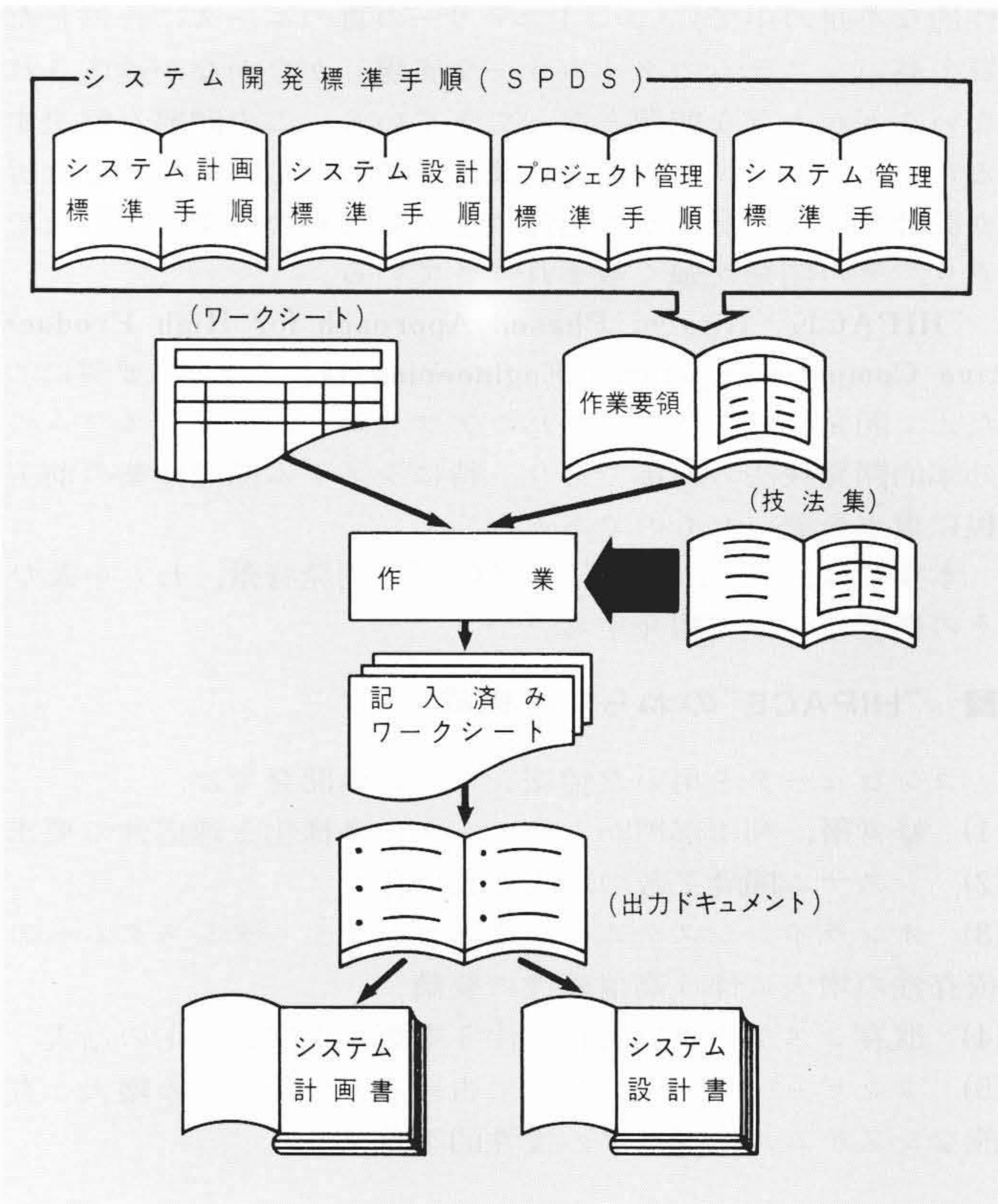


図2 システム開発標準手順の適用概念 システム開発標準手順は、ワークシートへの記入により作業を進めることを示す。

図3に、作業要領と具体的なワークシート記入例を示す。
SPDSを利用することにより、次に述べるような効果が得られる。

- (1) 作業手順の標準化
- (2) ドキュメントの標準化
- (3) システムの全体管理、プロジェクト推進管理の容易化
- (4) 信頼性、保守性の向上

3.2 システム計画標準手順

システム計画標準手順は、システム計画フェーズで使用する手順である。システム計画フェーズの目的は、分析フェーズで明確になったエンドユーザーのニーズと現状分析結果に基づいて、改善案の選定とコンピュータ処理への展開を行ない、実現すべきシステムのコストと効果を算定し、認可を受けることである。

図4は、システム計画標準手順の構成を示したものである。SPDSではシステム計画フェーズに重点を置き、できるだけ多くのシステムの要件を明確にすることを目標としている、このためエンドユーザが参加し、コンピュータ部門の管理者、システムエンジニアとともに作業を行なう。

3.3 システム設計標準手順

システム設計標準手順は、システム設計フェーズで使用する手順である。システム設計フェーズの目的は、システム計画フェーズで明確となったシステム要件を、与えられた環境の下で具体的な業務処理仕様、コンピュータ処理仕様に展開することである。

図5は、システム設計標準手順の構成を示したものである。システム設計フェーズは、システム管理(共有資源管理)と関係を取りながら作業を進めることが必要であり、エンドユーザー、コンピュータ部門の管理者及びシステムエンジニアが作業を行なう。

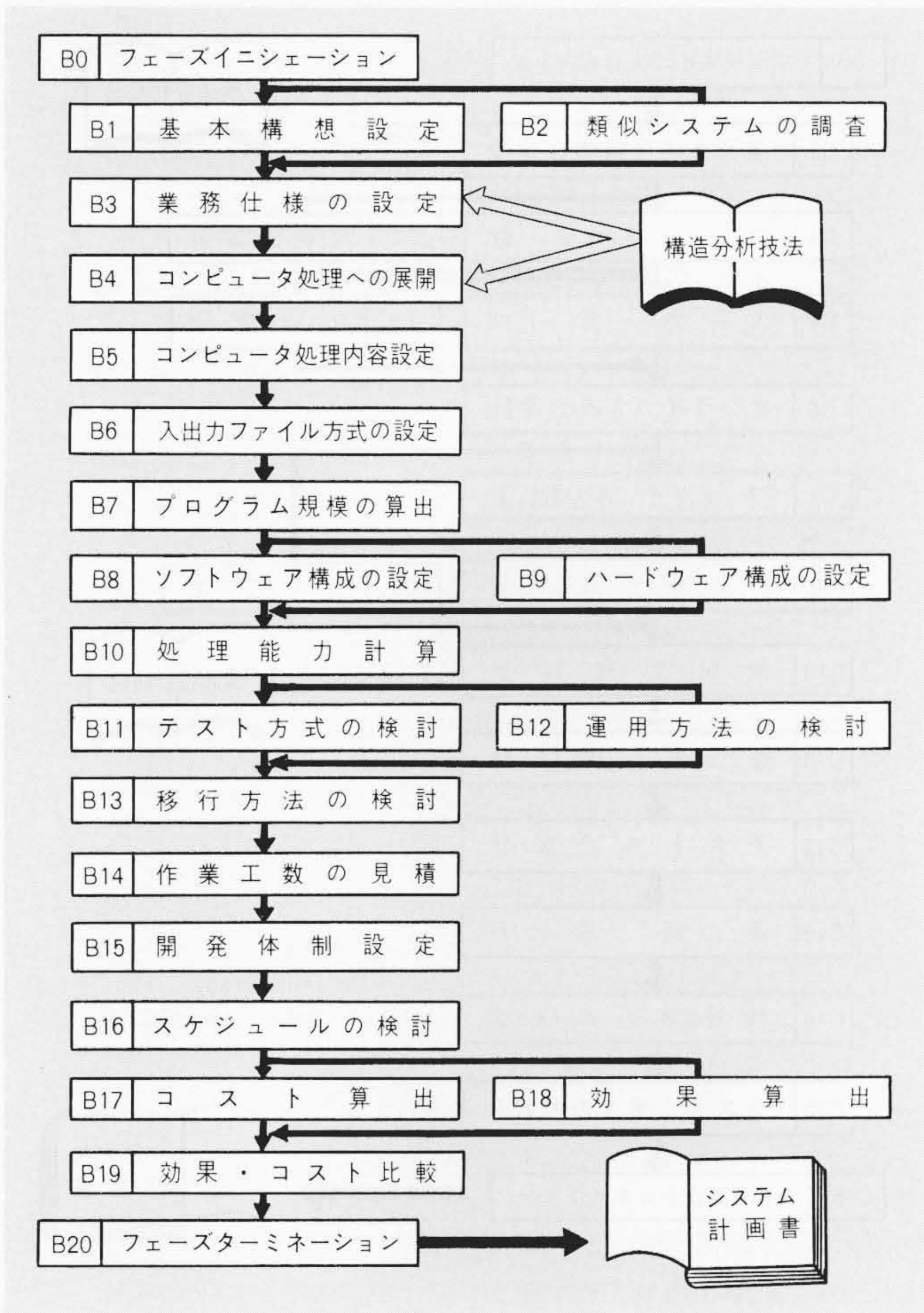


図4 システム計画標準手順の構成 システム計画標準手順は、21個のステップ(B0～B20)から構成されていることを示す。また、ワークシートは56種ある。

図3 システム開発標準手順の記入例 システム開発標準手順の作業要領、及びワークシート記入例を示す。

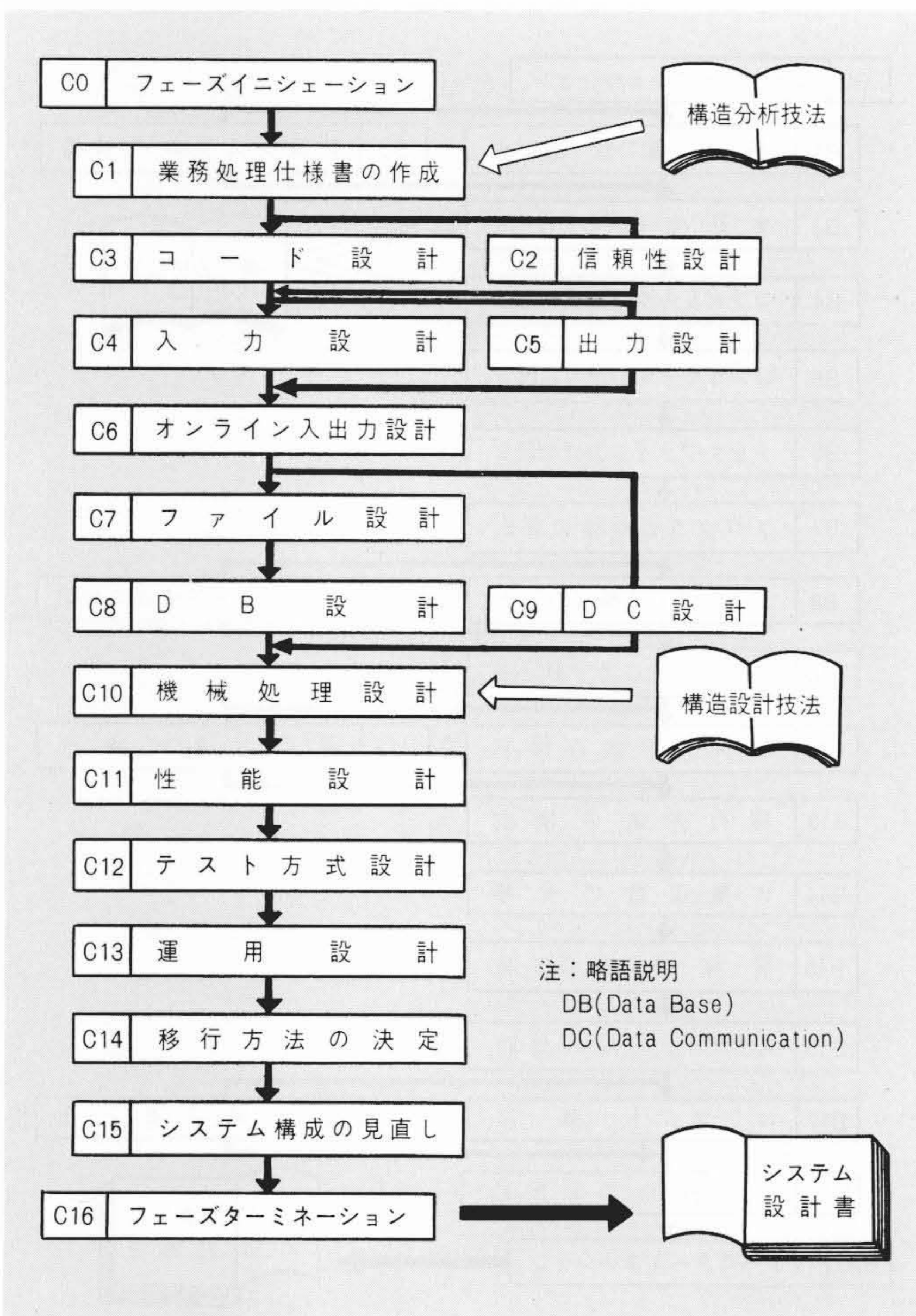


図5 システム設計標準手順の構成 システム設計標準手順は17個のステップ(C0～C16)から構成されていることを示す。また、ワークシートは64種ある。

3.4 プロジェクト管理標準手順

プロジェクト管理標準手順は、プロジェクトとして実施されるシステムの計画・設計・製造の作業を、計画・管理する手順である。

SPDSでは、プロジェクトの成果がユーザーによって機能的にも、性能的にも受け入れられ、しかも予定された納期と予算を守って達成されるように、次に述べるような方法を具体化している。

- (1) プロジェクト計画の充実のために、
 - (a) 計画の誤差を取り除くために、初期の計画を絶えず見直すためフェーズドプロジェクトプランニング手法の採用
 - (b) すべての計画の基礎となるプロジェクト作業項目の洗い出しと定義を行なうため、WBS(Work Breakdown Structure)の採用
 - (c) プロジェクトの発足に関する手順の定型化
 - (d) フェーズ別の組織体制の計画
- (2) プロジェクト管理の充実のために、
 - (a) フェーズドアプローチによる作業の進行と各フェーズでの開始、終了の管理者による着実なレビューの実施
 - (b) 管理のための基準としてマイルストーン計画とレビュー計画の事前立案
 - (c) プロジェクトファイルと管理レポートの標準化によるプロジェクト状況把握の容易化

プロジェクトの計画と管理は、主としてプロジェクト管理者により行なわれるが、その作業はプロジェクトメンバーによっても行なわれる。図6は、プロジェクト管理標準手順の記入例を示したものである。

3.5 システム管理標準手順

システム管理標準手順は、次に述べるような業務をシステム管理業務とし、システムの計画、設計、製造及び運用を通して実施する手順を示したものである。

(a) 作業要領

PJP1.3 プロジェクトWBS及び作業アクティビティネットワークの作成										ワークシート		
PJP1.3.1 WBS表										作成 田中	作成 80.5.6	
PJP1.3.1 WBS表										承認 山田	付	
PJP1.3.1 WBS表										バージョン		
PJP1.3.1 WBS表										フェーズWBS		
PJP1.3.1 WBS表										ステップWBS		
項目番号	最終成果物	大項目	中項目	小項目	作業名	作業番号	作業名	作業番号	作業名	作業番号	所要日数	備考
1	ニーズ分析書	1	ニーズの分析	1.1	ユーザーの明確化	1.1.1	ユーザーの定義	1.1.1.1	ユーザーの定義	1.1.1.1.1		
				1.1.2	ユーザーの洗い出し	1.1.2.1	ユーザーの洗い出し	1.1.2.1.1	ユーザーの洗い出し	1.1.2.1.1.1		
				1.2	ユーザーニーズの調査	1.2.1	ユーザーニーズの洗い出し	1.2.1.1	ユーザーニーズの洗い出し	1.2.1.1.1		
						1.2.2	ユーザーニーズの記述	1.2.2.1	ユーザーニーズの記述	1.2.2.1.1		

(b) ワークシート

注：略語説明 WBS(Work Breakdown Structure)

図6 プロジェクト管理標準手順の記入例 プロジェクト管理標準手順の作業要領及びワークシートの記入例を示す。

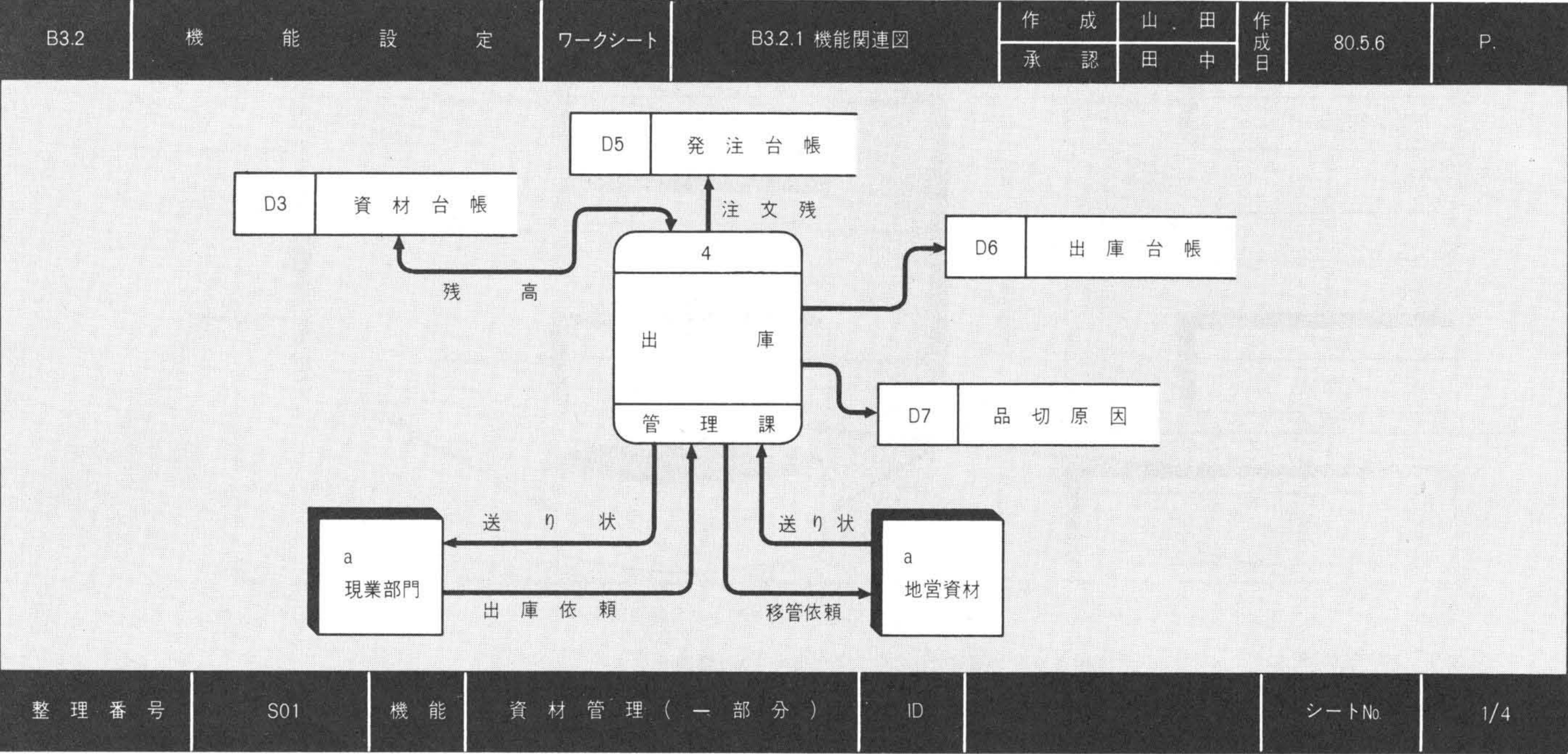


図9 SDFによる記述例 資材管理の一部をSDF (Structured Data Flow Diagram) で記述した例を示すもので、この図で記述イメージをつかむことができる。

ために用いる。図9は、資材管理業務をSDFで記述した例の一部を示したものである。構造化日本語は業務処理、コンピュータ処理の論理を正確に記述するための技法である。日本語のもっているあいまいな表現を避け、使用してよい文章、言語の制限及び段落付けによる文章の構造化、段落単位の番号付けなどにより、だれが読んでも正確に理解することができるように配慮している。

4.3 構造設計技法

構造設計技法は、主にプログラム設計フェーズを支援する技法である。本技法では「処理に基づく分割から機能に基づく分割」を原則とすることにより、個人的な技術に依存することなく作業を進めることができ、生産性、信頼性及び保守性の向上が図れる。

4.4 構造化プログラミング技法

構造化プログラミング技法は、プログラム作成段階での生産性、信頼性及び保守性の向上を達成するための技法である。本技法は、構造設計技法に基づいて設計されたプログラムのコーディング作業に適用され、見やすく分かりやすいプログラミング及びプログラムの部品化を実現する。

本技法では、COBOL及びPL/Iによるコーディング基準書も用意している。

5 支援ツール

支援ツールとして以下のものを用意している。

(1) 要求定義分析ソフトウェア⁶⁾

要求定義分析ソフトウェアは、システムについての要求、外部仕様、機能仕様などの決定事項を逐次データベースに蓄え、システムの利用者や設計者に必要なドキュメントを出力し、入力情報の抜けや矛盾をチェックするソフトウェアである。

(2) プログラム開発維持支援システム

CORAL(Customer Oriented Application Program Development System)は、コンピュータ部門でのプログラム開発と保守の作業を支援するソフトウェアである。仕様書言語と呼ば

れる日本語によるプログラミング言語と、プログラムを対話的に開発・保守するための機能を具備している。

(3) 運用管理システム

“HOPSS”(Hitachi Operation Support System)は、計算機の運用に関する作業の標準化を行ない、機械化することにより作業の効率向上と信頼性の向上とを図るものである。本システムは、ユーザー業務のスケジューリング、作業管理、データセットの管理、ボリューム管理、稼動実績管理までの一連の作業を管理する機能を具備している。

6 結 言

“HIPACE”は、アプリケーションシステム開発の標準化、及び体系化を行なったものであり、常に拡大の傾向にあるユーザーアプリケーションシステムを、長期にわたり建設、拡充していく際の基本となるものである。

今後、より多くのシステムへ“HIPACE”の普及を図るとともに、システム建設の効率向上のための技法及び支援ツールを更に開発する。また各種ソフトウェア製品がより効果的に使用される環境の整備を図る予定である。

参考文献

- 1) 日立製作所：アプリケーションマニュアル，HIPACEの紹介，G-7-348(昭55-2)
- 2) 日本情報処理開発協会：ソフトウェア開発・メンテナンスに関する調査研究報告書(1979-3)
- 3) 特集：効果的システム開発技法，ビジネスコミュニケーション(1978-3)
- 4) 春名：多関与者を有するシステムの計画技法—PPDS—，技術と経済，Vol. 8，No.10(1974-11)
- 5) K.Nakao et al.：A Structural Approach to System Requirements Analysis of Information Systems，COMPSAC '80，(1980-10)
- 6) 小林，外：要求定義言語及び解析システムの使用事例について，昭和55年度情報処理学会第21回全国大会前刷
- 7) 青山，外：アプリケーション・プログラム標準化法とソフトウェア工学，昭和54年度電気四学会連合大会前刷