U.D.C. 681.32.06.002:[658.5.012:681.322]

総合ソフトウェア生産管理システム“CAPS”

Computer Aided Production Control System for Software

芝田寛二* *Kanji Shibata*
横山陽一* *Yôichi Yokoyama*

近年，ますます大規模かつ高度化するソフトウェア製品生産に対応する生産性及び信頼性向上施策として，各種の生産技術開発の重要性とともに，生産管理強化の必要性もいよいよ高まっている。永年，日立製作所でも種々の方式を展開してきたが，このたび，全生産工程にわたる総合ソフトウェア生産管理システム“CAPS”を開発した。

CAPSは，製品のプログラム階層構造に即した部品対応の工程管理，品質管理情報を総合データベース化し，このデータをもとに，不可視的なソフトウェア生産工程の可視化と膨大化する生産管理業務の省力化を図るとともに，管理精度の飛躍的向上を実現しようとするものである。

1 緒　言

大規模なソフトウェアを，所望のコスト，納期及び品質を満たすように生産することは非常に大きな努力を要する。

本論文では，日立製作所で大規模ソフトウェア製品の生産管理に適用してきた種々の管理システムやツールについて，総合ソフトウェア生産管理システム“CAPS”（Computer Aided Production Control System for Software）を中心に紹介する。CAPSは，製品の急激な大規模，複雑化に対応してますます複雑化かつ膨大化する管理データを，構造設計技法[1)]によるソフトウェアの各階層構造に対応づけて，設計工程から検査工程まで一貫して把握するための，総合生産管理データベースであり，オンライン検索機能，選択予約出力機能，ノンインパクトプリンタによる図形出力機能などにより，タイムリーに見やすいレポートを提供することによって，管理者，設計者双方の管理作業の大幅な省力化と管理の質の向上を実現した。

2 生産管理上の諸問題と対策

ソフトウェア生産管理上の諸問題への対策は，そのまま生産管理システムやツールに必要な条件であり（**図1** 参照），特にソフトウェア生産過程の可視化と管理工数の低減は，最重要目標である。また，管理対象があくまで設計部門であることから，最終的には設計者の自主管理を助けるシステムを指向して，設計者の利用しやすさを最重点とした設計現場密着形のシステムとする必要がある。

3 生産計画の策定と実績の評価

CAPSと諸ツールの機能について**図2** に示す。

ソフトウェアの生産管理を行なうためには，まず原価，納期及び品質の管理基準が必要である。

3.1 資源の基準値と実績フォロー

ソフトウェアの製造原価の90%以上は，加工費（人件費）と計算機費で占められている。このため，まず工数，計算機時

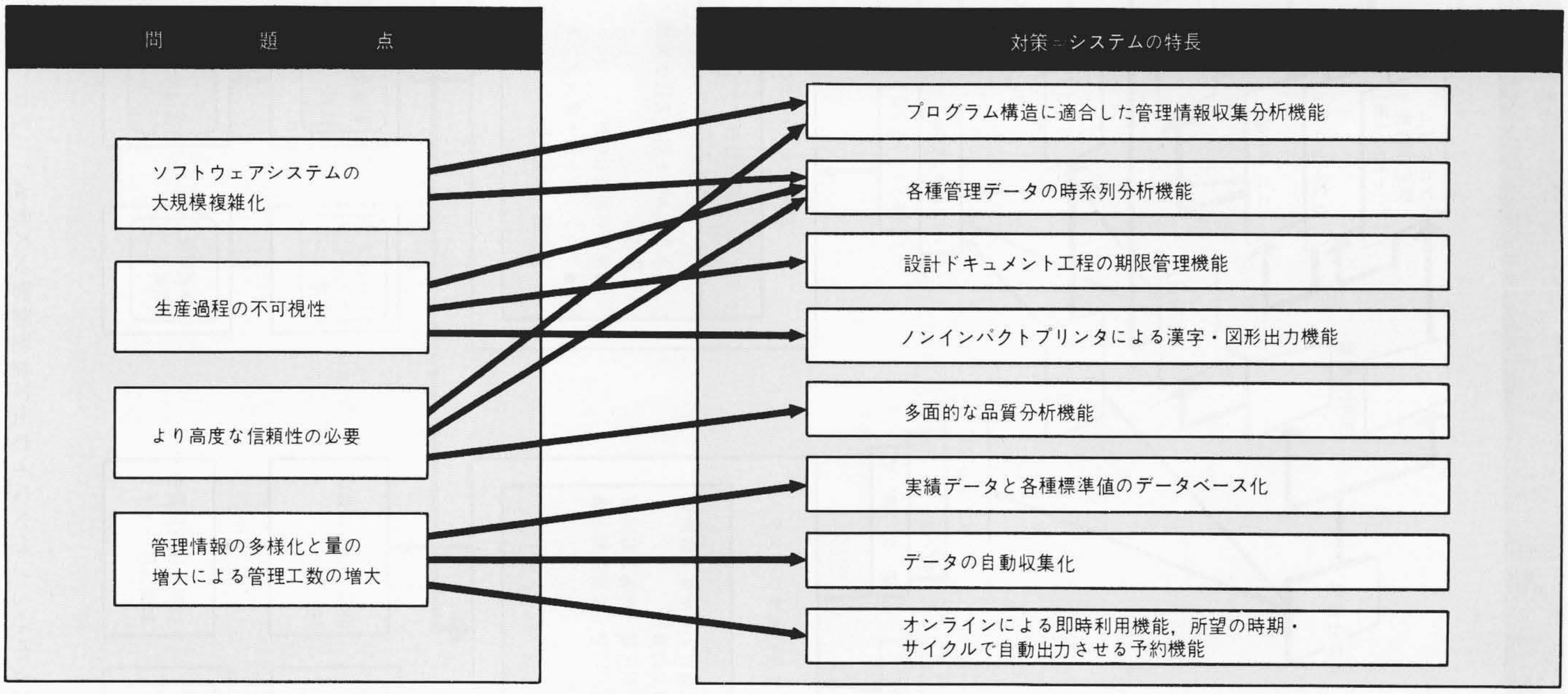

図1　ソフトウェア生産管理上の問題点と対策　問題点の解決を目標に，システムの仕様を設定した。

* 日立製作所ソフトウェア工場

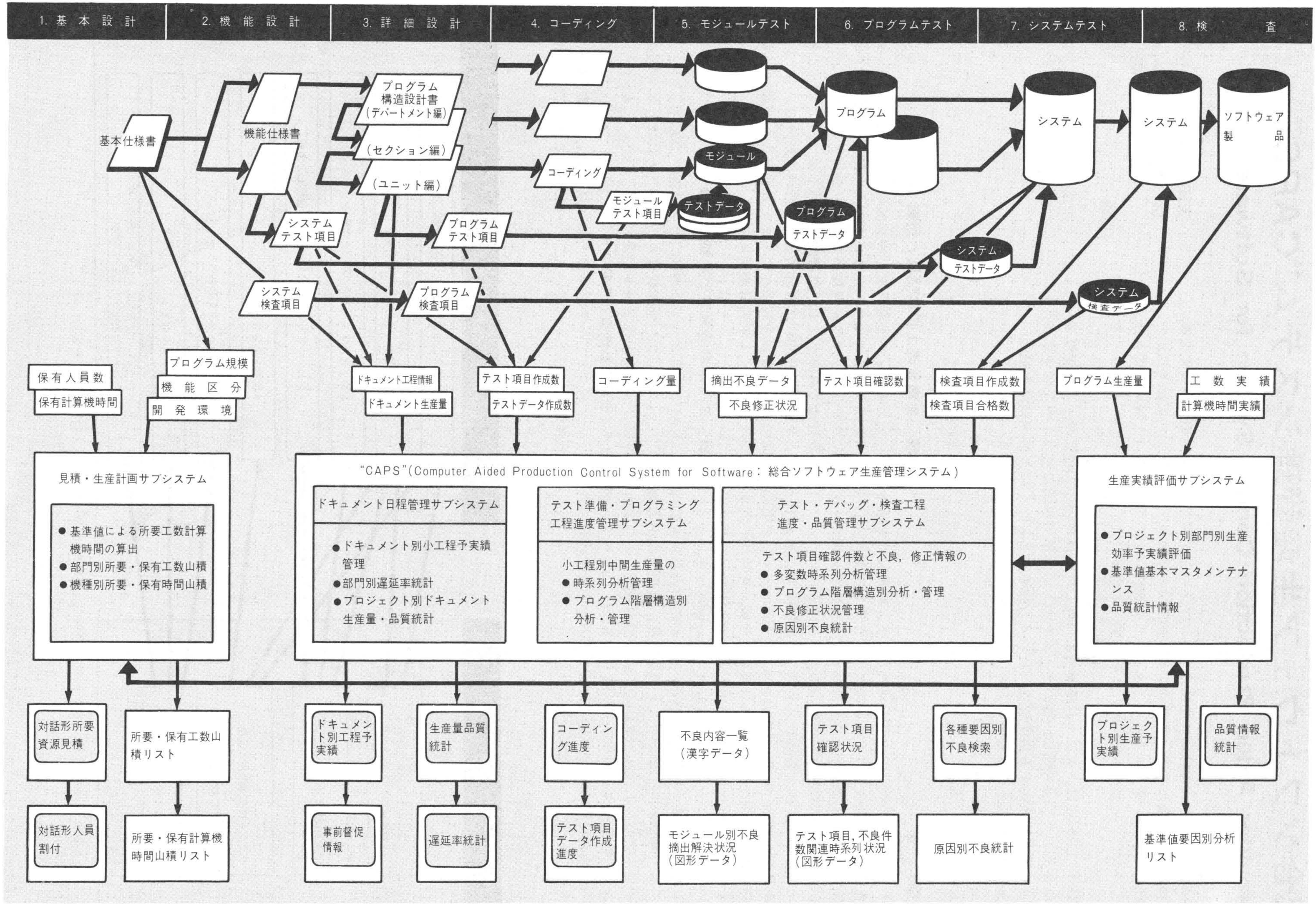

図2　CAPSを中心としたソフトウェア生産管理システム　　プログラム階層構造別，時系列別に各工程の管理情報をデータベース化した。

間についての見積基準値を設定し、実績をフォローするためのシステム作りを行なった。

また基準値の設定には事前に作業標準を定め，プログラマを階層付けして能力係数を設定した。

基準値は，使いやすく改訂が容易なように，できる限り単純な構成にした(**図3** 参照)。また，基準値の適用範囲は多方面であり，見積りは緊急を要し，多くの条件に対してシミュレーション的に適用する場合が多いため，オンラインによる算定プログラムを用意し活用している。

基準値は，工場の長期計画の策定から期予算，個別のプロジェクトの所要量計画，見積りに至るまで，工場共通の見積尺度として広く利用されている。

また，基準値運用の定着化のためには，実績のフィードバックと生産条件の変化に即応した定期改訂が必須である。そのためには，プロジェクト別の生産予実績がデータベース化され，オンライン端末から随時検索できるとともに，一定期間ごとに部門別，要因別に集計されて，能率評価や基準値の改訂に活用できるシステムとなっている。

3.2 標準日程

資源管理と日程管理は車の両輪の関係にある。ソフトウェアプロジェクトの混乱は大幅な納期遅延や原価超過を来すが，実際にはこれらが一つの現象となって起こる場合が多い。これらの失敗の主な原因に，当初の無理な日程計画がある。ソフトウェア開発といえども標準日程を設定し，事前チェックを強化することで，これらの失敗のかなりの部分を未然に防止することができる。

標準日程は，所要工数と投入人員からプログラムで自動算定する。この算定アルゴリズムには，下記の2点を考慮してある。

(1) 投入人員の実働時間率

(2) ソフトウェアの各規模及び基準値要因に応じた工程ごとの最小必要時間

3.3 品質尺度

近年，ソフトウェアの高信頼化への要求は高くなる一方である。品質向上の最上策は，前工程から不良を作り込まないことである。そのためには，種々の生産技術を活用することはもちろんのこと，各工程ごとに正しく品質の評価を行なえることが大切である。このための品質尺度としては，各工程ごとにステップ数，テスト項目数，ドキュメント枚数などによって，それぞれノーマライズされた不良件数である。これらのプログラム種別ごとに設定された品質評価尺度が，次項で述べるCAPSによって把握される工程ごとの品質実績と対比されて，早期工程での品質評価に結びつけられ，事前対策へとフィードバックされる。

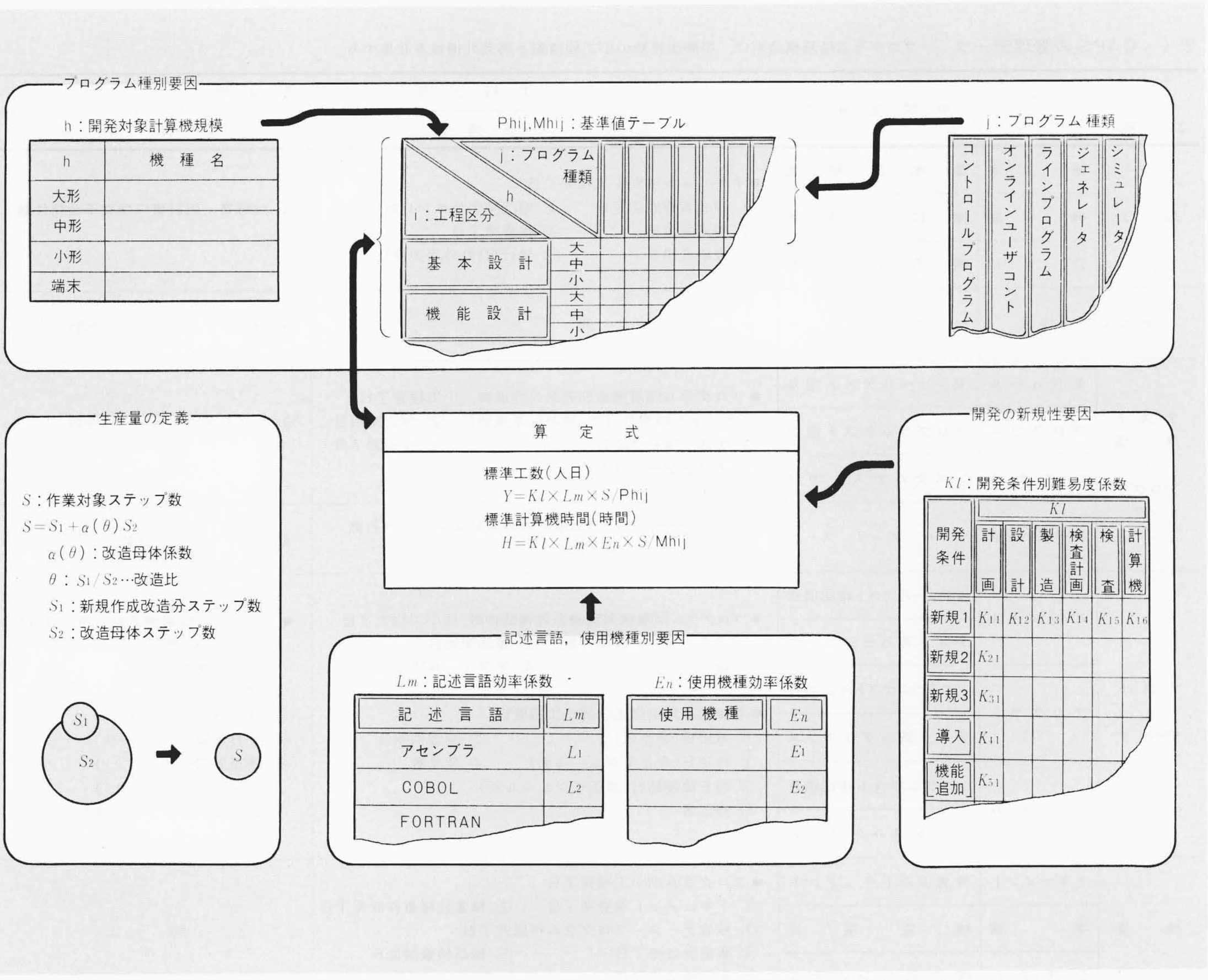

図3 基準値の構成と要因　ソフトウェア生産性を左右する多数の要因から，特に影響の大きいものを構造に取り入れた。

4 CAPS

ソフトウェアの生産管理にとって，進度，品質の相互関連は非常に強いものがあり，両情報を関連づけて一体で可視管理することが，どうしても必要である。これらの情報を，データベース化した総合生産管理システムとしてのCAPSについて次に述べる。

4.1 CAPSによる管理の特長

CAPSの重点課題は，不可視的なソフトウェア生産工程をいかに可視化するかであり，そのためには複雑な論理構造をもつソフトウェアが，「現在どの箇所が，どのような状態にあり，今後どのようになるのか。」を随時客観的に把握できるシステム作りが必要である。

日立製作所では，ソフトウェア生産の工業化を図るため，職人技巧的な設計を排除し，ソフトウェアを整然としたモジュール(部品)階層構造に設計する構造設計法[1]を全面的に適用し，ソフトウェア構造そのものに可視性を与えている(**図2**参照)。

CAPSは，階層構造向きのデータベース管理システムADM(Adaptable Data Manager)を活用し，構造設計によるモジュール階層に適合した管理データベースで構成されているので，実際のソフトウェア構造に即した可視管理が可能である。データベースに格納される管理データは，進度率や予実算比などの抽象的な百分率指標を一切排除し，各作業の中間生産物の生産量を時系列に収集したものである。そのため構造面の可視管理機能とともに，過去，現在の状態から将来の予測をも行なうことができる時系列面の可視管理機能をもっている。

これらの基本データとともに，担当者や各種要因コードなど管理に必要な，種々の付帯データを一緒に格納し，多面的な分析が行なえる(**表1**参照)。

また，出力機能そのものについても，生産量の時系列予実績推移などのグラフや，不良内容などの漢字データをそのまま出力するノンインパクトプリンタを活用し，**図4**中の出力例に見られるように，出力リストそのものの可視化も図っている。

また出力方式としても，必要な部分の情報を，所望のサイクルで，必要な量だけノンインパクトプリンタに自動出力するよう事前に予約できる選択予約機能や，ADMのデータコミュニケーション機能を活用したオンライン即時検索機能などの，オフィスオートメーション化を目指した諸機能をもっている。そのため，貴重な第一線管理者や上級設計者の工数を大幅に低減できる。

更に検索機能は，モジュール階層構造別の検索集計に加えて，各種の付帯データのすべてを検索項目として指定できるようにしてあるため，手作業とは比較できないほど高密度の可視管理を行なうことができる。

表1 CAPSの管理データ　プログラム階層構造別に，中間生産物の小工程情報と時系列情報を収集する。

工程		中間生産物	工程別進度・品質情報	
			進度・期限情報	品質情報
設計	基本	基本仕様書	●ドキュメント別小工程完了日 ① 担当者作成完了日　② 部門間デザインレビュー日 ③ 検査受付日　④ 検査終了日 ⑤ 最終合格日　⑥ 発行日(社外用のみ)	① 仕様書，設計書の検査不合格件数
	機能	機能仕様書		
	詳細	構造設計書		
マニュアル作成		マニュアル	同上	② マニュアル検査不合格件数
テスト準備・プログラミング	モジュール	モジュールテスト項目	●プログラム階層構造別時系列作成数，小工程完了日 ①テスト仕様書作成数(時系列累積値)　② 作成開始日 ③ テスト項目作成数　④ 作成完了日 ⑤ テストデータ作成数	③ テスト項目検査不合格件数
	プログラム	プログラムテスト項目		
	システム	システムテスト項目		
	コーディング	ソースリスト	① モジュール，プログラム単位の時系列コーディング行数 ② 開始日　③ 終了日	④ コーディング不良数
テスト・デバッグ	モジュール	モジュールテスト確認項目	●プログラム階層構造別時系列確認件数，同小工程完了日 ① テスト項目の確認件数　② 確認開始日 ③ 確認終了日　④ モジュール組立完了日 ●各摘出不良別修正状況小工程管理 ① 摘出日(発生日)　② 原因判明日 ③ 修正日(修正モジュール別)　④ 発見者 ⑤ 修正後確認日(修正モジュール別) ⑥ 修正者	●プログラム階層構造別，不良原因別，工程別，時系列の分析 ① 摘出不良数問題点 ② 修正中不良数 ③ 修正確認済み不良数 ④ 不良現象(漢字データ) ⑤ 不良原因(同　上) ⑥ 不良対策(同　上) ⑦ 修正ソースリスト行番号
		テスト済みモジュール		
	プログラム	プログラムテスト確認項目		
		テスト済みプログラム		
	システム	システムテスト確認項目		
		テスト済みシステム		
検査	ドキュメント	検査済みドキュメント	●プログラム別小工程完了日 ① ドキュメント検査完了日　② 検査仕様書作成完了日 ③ 検査データ・プログラム作成完了日 ④ 事前評価完了日　⑤ 製品検査開始日 ⑥ 製品検査終了日	同上
	準備	検査項目		
	システム	検査済みシステム		

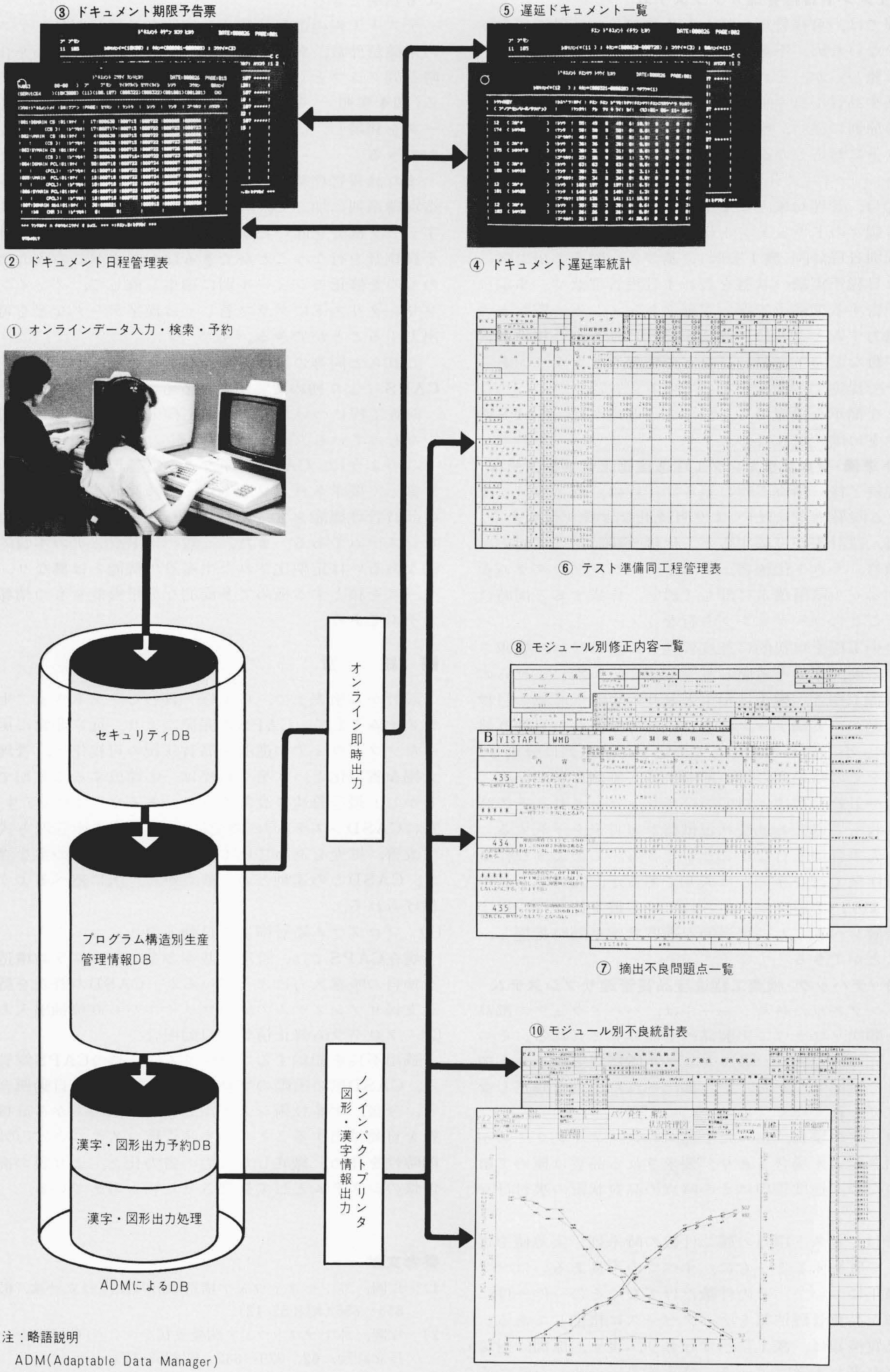

図4　CAPSの出力例　　オンライン端末画面出力とノンインパクトプリンタ図形・漢字出力から成る。

4.2 ドキュメント日程管理サブシステム

設計工程では，可視管理の核となるモジュール階層構造がまだ明確でないため，不可視性は特に強く，十分に可視管理することは難しいが，それを理由に管理を怠ると，後で大幅な工程遅延や品質不良を引き起こす場合も多い。設計工程も可視管理の原則に従い，唯一の中間生産物である個々の設計ドキュメントに対応した進度・品質管理を行なうだけでも効果は大きい。

CAPSでは，管理対象となるシステムの各プログラム単位に作成する個々のドキュメントに対応して，登録された作成検査小工程別日程計画(**表1**参照)に基づき，**図4**の出力例にあるような日程予実績の状態を表わす日程管理票や，事前に完了日を予告する期限予告票，遅延ドキュメント一覧などの督促票を出力するとともに，遅延率統計，検査不合格件数統計，ページ数などの生産量統計などの情報をオンラインあるいはバッチで提供する機能をもっている。

このような簡単な機能でも，設計完了遅延率は，CAPS適用以前の約$\frac{1}{4}$に削減された。

4.3 テスト準備・プログラミング工程進度管理サブシステム

設計工程終了後，検査工程に至る諸工程は，構造設計によるプログラム階層構造に従い，より可視的な管理が行なえる。

各仕様書，設計書が完成すると，仕様や設計論理に対応したテスト項目，テスト仕様書，テストデータ，プログラムなどを，モジュール階層構造に即して設定，作成すると同時にモジュールごとにコーディングを行なう。

これらの小工程を可視的に進度管理するためには，要求される管理の密度，水準を考慮して，モジュール別にこれらの小工程の目標工期と，**表1**に示すとおりの中間生産物の目標生産量を，過去の実績データに基づいた標準値から見積り設定する。更に，小工程の工期を必要に応じて時系列に分割し，目標生産量をその時系列に従い展開する。実績は各時系列ごとの目標値や最終目標値と対応づけてフォローされ，その時点での達成状況と過去の進度状況推移から将来を予測することにより，先手管理を行なうことができる。このような作業をすべて手作業で行なうことは大変であるが，CAPSを利用することにより，データが入力されている最下位レベル以上の任意の階層レベルごとに可視的な進度状況を随時把握し，管理することができる。

4.4 テスト・デバッグ・検査工程進度品質管理サブシステム

コーディングされた各モジュールは，ハードウェアの部品と同様に，順次ソフトウェア製品へと組み立てられる。その際同時に，先に設定された各モジュール階層対応のテスト項目一件一件を確認しながら，かつ摘出された不良を修正しながら組み立てられてゆく。

ソフトウェアの場合，たった1箇所のコーディングミスが大障害を引き起こす場合もあり，要求される品質は極めて厳しい。そのため，進度管理はその時点の品質状況の裏付けが必要である。

CAPSでは，テスト項目の確認件数の時系列予実算値を進度情報として収集するとともに，すべての不良あるいはソフトウェア修正について，その件数だけでなくその一件一件について詳細な品質管理情報をデータベースに格納してある。

個々の品質情報は，**表1**に示すとおり，現象，原因，対策などの内容を表わす漢字データ，修正モジュール，システム名称，担当者，各種不良原因・要因コード，更には修正作業の作業別実績完了日などであり，これらのすべてのデータを管理情報として出力できるほか，分析のための検索項目としても利用できる。

テスト工程の進度状況や全体的な品質状況は，テスト項目の未確認件数，発生不良件数，同解決済み件数とを合わせて時系列グラフとして出力することにより，可視的に把握できる(**図4**参照)。更に，修正一件一件の作業別実績完了日のデータを利用し，個々の修正作業の詳細進度管理を行なうことができる。

また品質管理関係の機能は，基本となるモジュール階層構造別時系列に加えて，入力された詳細品質データのうち，漢字データ部分を除いたすべてのデータ項目を検索条件として，不良統計を行なうことができるほか，すべての入力データそのものを修正モジュール別に編集し直して，ノンインパクトプリンタリストにグラフ若しくは漢字データなどを可視的に出力することができる。

これらと同等の管理を手作業で行なうことは不可能であり，CAPSにより初めて実現されるものである。

検査工程についてもテスト工程と同様の進度・品質管理機能をもっている。

このように，CAPSは単なる進度管理機能以外に，ますます厳しく要求されるソフトウェアの高品質化を達成するための品質管理機能をもっており，むしろその機能に重点をおいたシステムである。また，一般ハードウェアの本質的に可視的なあるいは定型化された生産管理機能とは異なり，データベースを核とする極めて多面的な分析機能をもつ情報検索システムである。

5 結　言

設計から検査まで，全工程一貫したソフトウェア生産管理システムとしてのCAPSの開発により，従来非常に困難であったソフトウェアの進度・品質状況の可視化と，管理業務の大幅な省力化という第一目標は一応達成することができた。しかし，最近急速に改善されつつあるソフトウェア生産技術，特にCASDシステム[2]の開発に対応して，生産管理方式も絶えず改善，拡充を進めていく必要がある。当面の主な課題として，CASDとの連動という観点から，次に述べるような点が挙げられる。

(1) プログラム階層構造の自動入力

現在CAPSでは，管理の基本となるプログラム構造の登録を独自の帳票入力によっているが，CASDの普及を待って設計支援サブシステムの出力ファイルから直接抽出入力する。

(2) プログラム修正情報の自動照合

摘出不良を記録するプログラム修正票のCAPS側管理情報と，CASD側の実際のプログラム修正状況を自動照合するなど，今後，生産技術システムとしてのCASDから直接各種情報を自動抽出することで，生産管理システムとしての即物性，即時性を高め，徹底した入力の省力化と，より質の高い管理情報のシステムとして発展させたいと考えている。

参考文献

1) 片岡，外：ソフトウェア構造設計技法，日立評論，**62**，853～856（昭和55-12）
2) 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**，879～882（昭和55-12）
3) 岩元，外：プロジェクト管理のツール，情報処理，Vol. 20，No. 8，671（1979）
4) 芝田，外：ソフトウェア製品の生産計画と工程管理，情報処理，Vol. 21，No. 10，1035（1980）