

生産管理用コマンド形言語

Production Management Command System

生産管理ソフトウェアの生産性向上と段階的拡張を可能とするため、生産管理用コマンド形言語を中心とする生産管理用統合ソフトウェアの体系化を進めている。生産管理用コマンド形言語は、計算機やアルゴリズムの専門家でなく、生産管理の実務担当者にとって理解しやすいように、生産管理でのデータ処理を入力表から出力表への変換過程としてとらえて、各々の命令(コマンド)を生産管理の基本情報処理単位に対応させている。生産管理ロジックを記述する基本的なコマンドと生産管理に標準的に用いられる機能を部品化したコマンドとがある。これらコマンドを組み合わせるにより、容易に生産管理ソフトウェアを構築でき、従来言語と比べてステップ数は約 $\frac{1}{14}$ となる。

大成 尚* Hisashi Ônari
土屋 昇* Noboru Tsuchiya
杉野和宏* Kazuhiro Sugino
土谷健次** Kenji Tsuchitani

1 緒 言

計算機の性能向上と価格低下に伴い、CIM (Computer Integrated Manufacturing)やFA (Factory Automation)と称する計算機を積極的に利用した生産システムの総合自動化が盛んである。

この総合自動生産システムでは、単に自動機を導入した作業の自動化だけでなく、材料と設備や人的資源を効率良く運用することで製造システム全体の効率化を図る必要がある。

したがって、製造活動の計画と統制をサポートする生産管理システムに対するニーズは高く、その重要性も増加している。しかし、生産管理システムの実現に必要なソフトウェアの作成工数と期間が多岐であることと、その拡張、変更柔軟な対応がとれないという問題がある。

また、個々の製造現場に置かれた専用計算機を用いて、担当者が現場で発生する諸問題をその場で即時解決する分散形生産管理システムのニーズが高い。分散形生産管理システムでは、基本的生産管理機能は同じであるが、個々の現場の要求仕様に合わせて多数の生産管理ソフトウェアを作成しなければならず、更に多大の工数と期間を要する。

上記の問題を解決するために、生産管理ワークステーションにより作業計画、進捗管理、現場事務などの対話形処理をつかさどり、更にリアルタイム制御系へのSPC (Supervisory Production Control)を行なう生産管理統合ソフトウェアを開発した(図1)。

この生産管理用統合ソフトウェアは、リレーショナルDB (Data Base)を核としており、次の特徴をもっている。

- (1) 生産管理用コマンド形言語により、生産管理ロジックを容易に構築できる。
- (2) 作画、作票などの各種支援ツールにより、エンドユーザプログラマブルな使いやすさをもっている。
- (3) リアルタイム制御系など他システムとの通信が容易である。
- (4) UNIX^{*)}ベースの流通ソフトウェアが活用でき、アプリケーションソフトウェア作成量が低減される。

本論文では、統合ソフトウェアの1番目の特徴である生産

※) UNIXは、米国ATTベル研究所が開発したオペレーティングシステムの名称である。

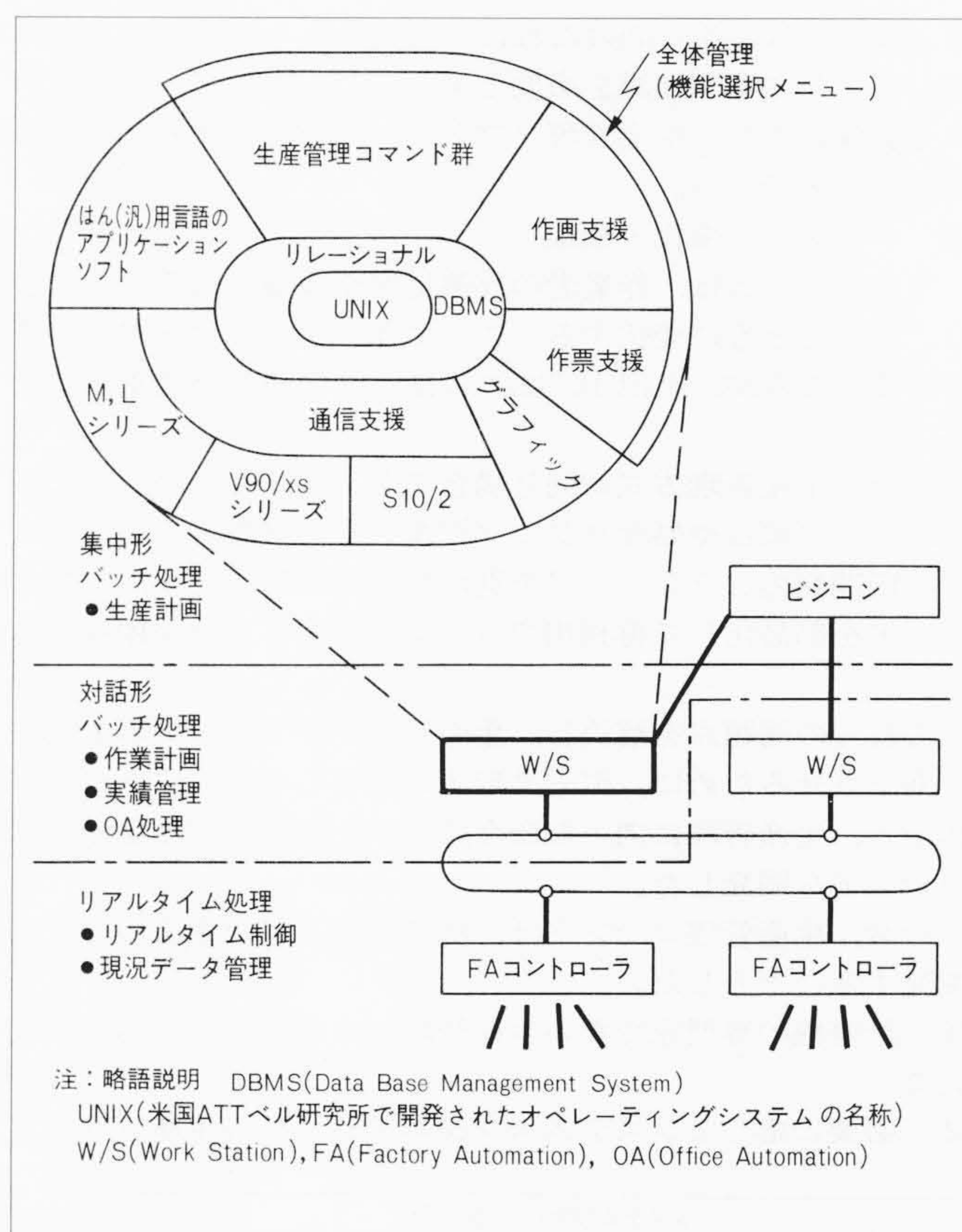


図1 生産管理用統合ソフトウェアのシステム構成 リレーショナルデータベースを中核とした生産管理統合ソフトウェアにより、ソフトウェアの大幅な生産性向上が実現される。

管理用コマンド形言語(以下、生産管理コマンドと略す。)^{1),2)}について述べる。

生産管理コマンドは、計算機やアルゴリズムの専門家でなく、生産管理の実務担当者に理解しやすいように、各々の命令(コマンド)を生産管理の基本情報処理単位に対応したものである。したがって、ユーザーがコマンドを入力することにより、容易に生産管理の各種機能を実現できる。

2 開発の目的

ソフトウェア生産性向上を妨げている従来の生産管理ソフトウェアの特徴を以下に記す。

(1) ファイル中心のソフトウェア

生産管理ソフトウェアは、ディスクファイル中にあるデータを取り出して加工し、結果をファイルに戻すという処理が中心の情報処理系である。したがって、前もってシステム全体の仕様を設計し、データファイル構造を確定してからでなければソフトウェア設計を行なえない。また、ファイルの変更はシステム全体に大きな影響を及ぼすことになる。したがって、部分的に小さなシステムを開発し、段階的に大きなシステムに発展させるという方法を採用することは困難である。

(2) はん(汎)用言語での開発

プログラミング言語としてCOBOL、FORTRANなどのはん用言語を用いており、生産管理システムに必要なプログラムの規模が100kステップを超える場合も多く、多大な開発工数だけでなく、プログラムの明示性に欠けるという問題がある。また、はん用言語はだれにでも使えるわけではなく、計算機システムの専門知識を必要とする。したがって、よりマクロな表現により、生産管理ソフトウェアを実現する問題向き言語が必要である。

(3) システムの変化・相違

生産システムは、作業者の改善提案や設備の導入などにより、時間とともに変化する。これに伴いソフトウェアの変更が必要となるが、上記(1)、(2)の特徴から簡単には変更できない。

また、生産管理方式が同じ場合でも、製品や工場が異なると、データ項目や処理ロジックが少しずつ異なる。したがって、既開発のソフトウェアを別の職場で利用したり、ソフトウェアを部品化して再利用できるようにすることが困難である。

これらの問題点を解決し、生産管理ソフトウェアの生産性を向上させるために、データ形式に依存しないソフトウェア体系と、生産管理に向けた命令語体系を特徴とする生産管理コマンドを開発した。

なお、生産管理コマンドは、次の要求を満たすものとして開発することとした。

- (1) 計算機の専門家でない生産管理の実務担当者が容易に使える。
- (2) 必要に応じてシステムエンジニアリングの生産管理ソフト

- トウェア開発ツールとなり得る記述能力をもつ。
- (3) 生産管理ソフトウェア部品化の手段となる。
- (4) ソフトウェアの独立性、拡張性を高くする。

3 表変換によるソフトウェア体系²⁾

生産管理ソフトウェアは、帳票のように表形式でファイル中にあるデータを取り出し、必要な計算を行ない、その結果を表形式としてファイルに戻すといったファイル中心のデータ処理システムである。

生産管理コマンドでは、表形式データの入出力を基本として、生産管理ロジックを記述できるように言語体系を定めた。ここで、表形式データを入出力とするデータ処理過程を表変換と呼ぶ。

つまり、生産管理に必要なデータ処理を整理し、共通的な表変換として部品化したコマンドを準備しておき、この組合せで生産管理ロジックを記述する。

従来の記述法では、数値や文字列の単位で処理を考えていたのに比べ、生産管理コマンドではデータの集まりである表を単位として考えるため記述効率が良い。また、入力表から出力表に至る処理の一つ一つがそれぞれコマンドで表現されるため、プログラムの明示性が良く、変更が容易である。

図2に表変換と従来のはん用言語による部品所要量展開の記述例を示す。これは、生産計画を実行するのに必要となる部品要求表、すなわち各部品の所要員数と期日を求める処理であり、二つの表変換で表わされている。第1の表変換UPDATEでは生産計画表の着手日計算を行ない、第2の表変換INSERTでは着手日に間に合わせた部品要求表を作成する計算を行なう。

従来の方法では、サブルーチンの中まで見ないと処理内容が不明確であるが、表変換では、コマンドに続いて生産計画、部品構成などの入力表を書き、その次にSETに続いて計算式を書いており、処理内容が明確になっている。また、システム設計での人間の思考は各種の表を中心に展開しており、その思考過程に合った表変換コマンドを用いることで明示性の高いプログラムとなる。

4 基本コマンドと管理コマンド

生産管理コマンドには、はん用言語での数十ステップに対応する処理を1命令で行なう基本コマンドと、はん用言語での数百ステップに対応する処理を1命令で行なう管理コマンドがある。また、デバッグツールとして対話形修正コマンド



図2 生産管理機能と表変換コマンド 生産管理ソフトウェアは、入力表から出力表への変換過程であり、本システムは表の変換過程を記述するための命令(コマンド)を用意している。

も準備されている。

4.1 基本コマンド

生産管理のロジックを記述する基本的なコマンドでINSERTなど30種類ある(図3)。

基本コマンドのいくつかは、RDB(Relational Data Base)言語と類似の計算機能をもっている。ただし、RDB言語で扱う関係演算の基本は、物理的に複数存在する表を論理的に一つの統合した表として扱い、特定の条件を満足するレコードだけを取捨選択するにすぎず、生産管理に要求される計算能力をもつとは言えない。例えば、図4に示すロット設定機能では、所要量表中の第2、第3及び第4レコードの一部をまとめて、手配表の第1レコード、3/19納期で手配数50を作り、第4レコードの残りから手配表の第2、第3レコードを作るといった計算を行なっている。生産管理ではこの例のように複数のレコードから一つのレコードを作り、一つのレコードから複数のレコードを作るといった計算能力が必要となる。このために、基本コマンドはRDB言語に内部変数(同図の:不足)と繰り返し出力(同図のWHILE節)などの機能追加を行なっている。

基本コマンドの言語形式は、ISO(International Organization for Standardization)の定めるリレーショナルDB言語SQL(Structured Query Language)に準拠した仕様としている。

4.2 管理コマンド

生産管理の標準的機能(ロジック)を部品化したコマンドであり、現在、34種類用意している(図3)。

同様の考え方に基づくコマンド形式の生産管理用ソフトウェアとして、米国GE社(General Electric Company)のMIMS³⁾とノルウェー工業大学のBASEOPERATIONS⁴⁾がある。

管理コマンドとして実現した生産管理の標準機能は、米国市販生産管理ソフトウェアの調査結果⁵⁾と日立製作所での生産管理システム開発経験を分析整理して選定しており、そのはん用性は高い。

また、管理コマンドは、基本コマンドを組み合わせで作成しており、その機能変更や拡張が必要な場合は、基本コマンドによる記述を変えることによって容易に実現できる点で柔軟性が高く変更が容易である。

基本コマンド		管理コマンド		
機能分類	コマンド名	機能名	コマンド名	タイプ
表変換	INSERT	部品展開	EXPLODE	2
	UPDATE	ロット設定	LOTSIZE	4
表形式入出力	INPUT	工程展開	ROUTE	1
	DATALOAD	在庫引当て	LOCATE	4
	DISPLAY			
グラフ出力	CHART			
単純計算	SET	ライン順列	SEQLINE	1
テーブル作成・削除	CREATE	着手指示	DISPATCH	5
	DROP			
・	・			
・	・	進行監視	MONITOR	2
・	・	実績集計	MEASURE	4
合 計	30	合 計		34

図3 コマンドの例 生産管理のロジックを記述する基本コマンドと、基本コマンドの組合せで記述された管理コマンドがある。生産管理システムは、これらコマンドの組合せで容易に構築できる。

(ロット設定)

手配基準：固定数量(50)

所要量表		手配表	
日付	必要数	納期	手配数
3/18	0	3/19	50
3/19	20	3/22	50
3/20	20	3/22	50
3/22	90	・	・
・	・	・	・
・	・	・	・

INSERT INTO 手配表(納期, 手配数)
SELECT 日付, :固定数
FROM 所要量表 KEY=(日付)
CREATE VAR :不足=0, :固定数=50
SET :不足=:不足+:必要数
WHILE :不足>0
SET :不足=:不足-:固定数
ENDWHILE ;

図4 ロット設定の記述例 生産管理には、データベース検索言語では根本的に実現不可能な計算能力が必要である。これを実現したものが本システムである。

5 データインタフェース

生産管理コマンドの処理系のソフトウェアは図5に示すように、下記の3階層とデータベースシステムから構成されている。

- (1) コマンドプロセッサ
- (2) データマニピュレータ
- (3) データインタフェース

投入されたコマンドは、コマンドプロセッサで解釈され、実行しやすい形に変形されてから、データマニピュレータで実行される。表変換コマンドでは、入力表中にある全レコードに対して1レコードずつ取り出しながら同様の計算を繰り返す必要がある。このために、コマンドを解釈しながら即実行を行なうのではなく、繰返し処理に不要な処理を前もってコマンドプロセッサで済ませておき、最小限必要な処理だけをデータマニピュレータで実行することにより、効率の良い処理を実現している。

生産管理コマンドでは、データ形式に依存しないソフトウェア体系とするため、RDBを採用し、データ形式の相違や変更に対するソフトウェアの独立性を高めている。

データインタフェースは、DBに対する本システムの移植性を高めるため、DBからのデータ取り出しや、書込み処理をRDBとデータ操作部から独立させたものである。

現在、データインタフェースとして、オペレーティングシステムUNIX(V90/X5用)とVOS(Virtual-storage Operating System)(Mシリーズ用)の各々に生産管理コマンド専用RDBとはん用(市販)RDBに対するインタフェースを準備している。

表1にUNIX用のRDBの比較を示しているが、専用RDBは、

表1 UNIX上でのRDB比較 専用DBは動的な表の作成機能などが優れている反面、セキュリティなどの機能が弱い。はん用のデータベースシステムと用途により使い分けることによって、両方の長所を有効に活用できる。

	応 答 性	セキュリティ	動的な表作成	
専 用RDB	中	×	○	⇒ 開発向き
はん用RDB	高	○	×	⇒ 定形業務向き

注：略語説明 RDB(Relational Database)

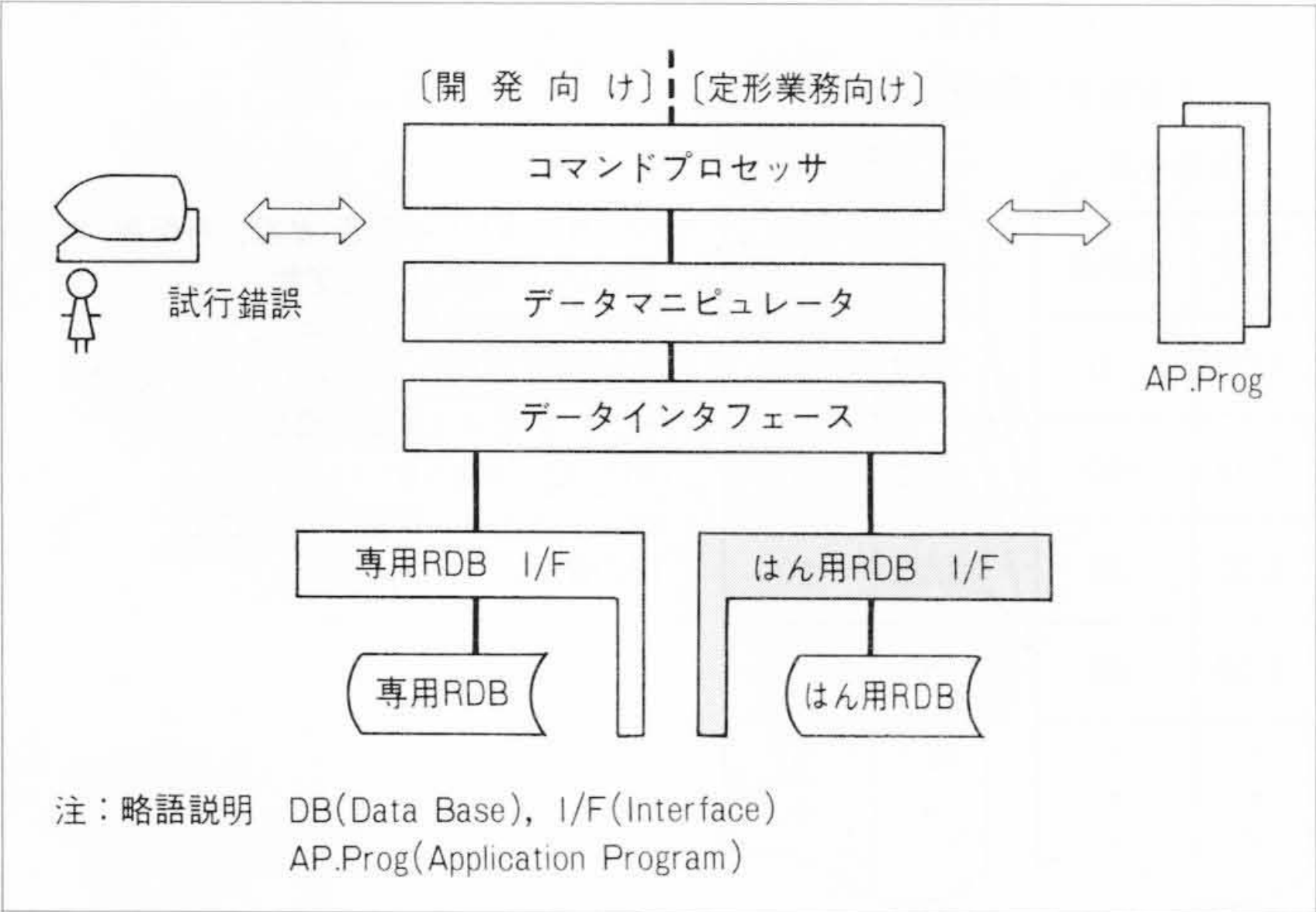


図5 ソフトウェア構成 オペレーティングシステムやデータベースシステムからの独立性を高めるために、データインタフェース部を設けた。この結果、本システムは大形コンピュータからマイクロコンピュータまで、どの計算機でも実行可能である。また、異なる2種類のデータベースシステムを同時に使用することも可能である。

生産管理システムの記述に便利な機能が多いが、処理が遅く、はん用RDBは処理は速いが、動的表作成などの柔軟性に欠ける。生産管理コマンドでは、図5に示すように、これら二つのDBを同時に使用可能であり、専用RDBによる開発時の使いやすさとはん用RDBによる定形業務時の応答性の両方を満足させる構成となっている。

6 適用事例

生産管理コマンドとその周辺ソフトウェアを含む生産管理用統合ソフトウェアの適用性の検討を行なった。対象とする生産管理システムとして、ごく最近開発したもののの中から中小規模の典型例を選んだ。図6にその検討結果を示すが、生産管理用統合ソフトウェアで実現できない(又は、性能的にすべきでない)生産管理機能のデータ処理部分は、プログラム容量比で13~40%と小さく、生産管理用統合ソフトウェアの適用範囲が広いことが分かる。また、生産管理コマンドの特徴である表変換部分だけでも、最大50%であり、生産管理ソフトウェア開発ツールとして効果がある。

生産管理用統合ソフトウェアは、計画や実績評価機能を中心とする生産管理ソフトウェアの開発に適しており、生産統制機能を中心とするリアルタイム性の強いソフトウェア開発には適さない。

なお、適用範囲外のデータ処理部分については、従来のはん用言語を用いて作成した専用プログラムとして別途開発し、生産管理用統合ソフトウェアと連結して使用する。

図7に生産管理コマンドの効果を検討した結果を示す。これは、典型的生産管理機能である(1)日程計算、(2)負荷山積と(3)作業指示を例として、基本コマンドとC言語の記述性を比較したものである。本例では、生産管理コマンドは、C言語に比べてステップ数で約1/14となり、データ処理単位の日安となるモジュール数とコマンド数の比較では約1/2となり記述性が大幅に向上していることが分かる。

生産管理コマンドは、(1)表形式のデータ入出力、(2)機能対応コマンドなどにより、プログラム内容が分かりやすく、テーブル設計などを含めた開発期間全体は、C言語を用いた場合に比べて約1/5となった。

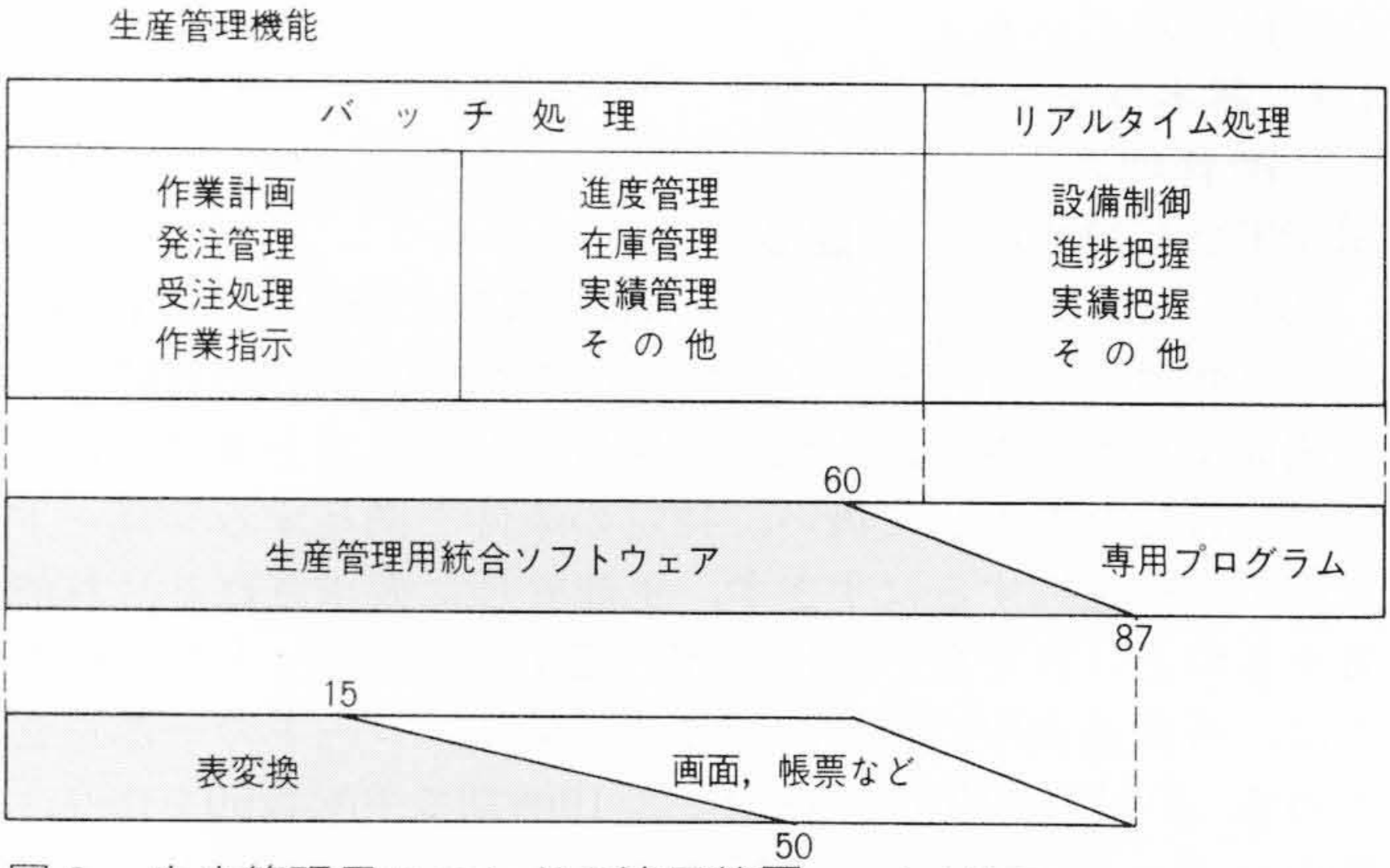


図6 生産管理用コマンドの適用範囲 生産管理ソフトウェア全体のうち、60~87%を占める部分に適用可能である。他の部分については、個別に開発したプログラムを本システムからダイナミックに起動することにより、全体システムを実現する。

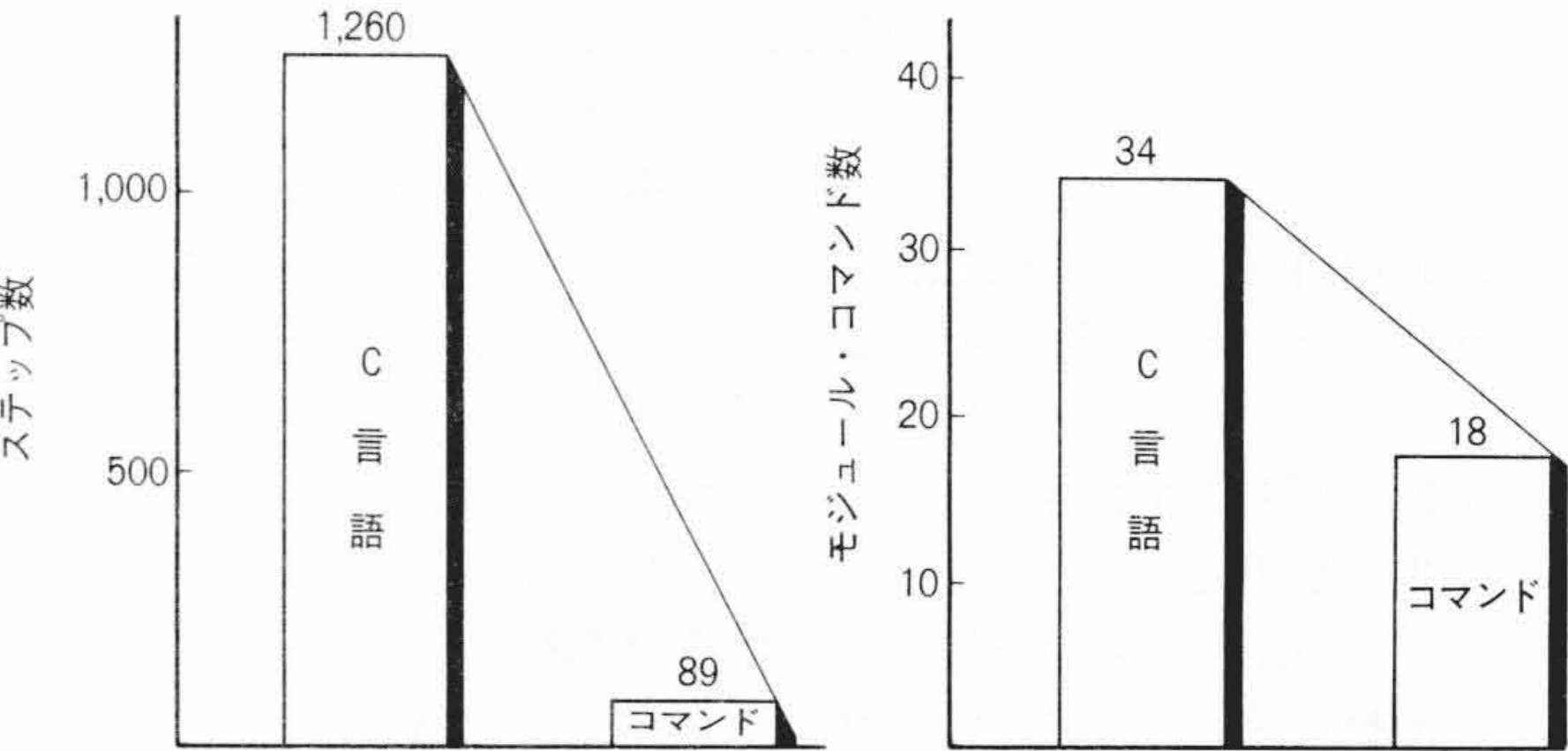


図7 生産管理用コマンドの効果例 従来のはん用言語と比較した場合、ステップ数で1/14となる効果を確認した。

7 結 言

FAシステムの高効率稼動に不可欠な生産管理ソフトウェアの生産性向上と、その段階的拡張を可能とするために、ユーザーがコマンドを入力することにより容易に生産管理の各種機能を実現できる生産管理用コマンド形言語を開発した。

本言語は、生産管理でのデータ処理を入力表から出力表への変換過程としてとらえ、計算機の専門家でない生産管理実務担当者にとって理解しやすい以下のコマンドから成る。

- (1) 基本コマンド：生産管理ロジックを記述する基本的データ処理命令
 - (2) 管理コマンド：生産管理の標準的機能を部品化した命令
- これらを組み合わせることにより、従来言語に比較して見やすく、ステップ数も1/14となり、大幅な記述性の向上が期待できる。

参考文献

- 1) 杉野, 外: 生産管理コマンドシステム: 情報処理学会第30回大会予稿集, 2113~2114(昭60-3)
- 2) 土屋, 外: 生産管理コマンド型言語: 情報処理学会第32回大会予稿集, 1895~1896(昭61-3)
- 3) MIMS入門: 電通国際情報サービス
- 4) O. Otherhals: Baseoperations in Factory Management: Dr. Thesis, Univ. of Trondheim & Norwegian Inst. of Tech. (1980-2)
- 5) H. Onari: Survey of Commercially Available Production Management System: CAM-I (1981-12)