

オペレーティングシステム “MIOS7/AS” の基本技術

Basic Technology for Operating System “MIOS7/AS”

OPS (Office Processor) の利用については、単にオフィス業務の効率化を図るだけでなく、ネットワークやデータベースによって情報をより戦略的に活用するといったニーズの高度化、多様化の要求が高まっている。HITAC L-700シリーズでは、このようなニーズに対応し、OS (Operating System) であるMIOS7/AS (Multiple office Information Operating System 7/Advanced System) を開発した。

MIOS7/ASでは、L-700シリーズのハードウェアアーキテクチャを生かし、使いやすく高性能なシステムが実現できるように新機能を支援した。特に使い勝手については、イメージ処理技術を適用してプログラムの自動生成ができる機能や、自律分散技術を応用し高信頼化および負荷分散化が図れるエバーラン機能を開発した。これらの使い勝手の向上により、多様化するユーザーニーズに対応でき、より高度なシステムの構築が可能となった。

野口耕平* *Kôhei Noguchi*
竹内増幸* *Masuyuki Takeuchi*
土屋雅幸** *Masayuki Tsuchiya*
織茂昌之*** *Masayuki Orimo*
武田晴夫*** *Haruo Takeda*

1 緒 言

OPS (Office Processor) の利用については、オフィス業務の効率化を図るだけでなく、情報を多角的に活用した戦略的な情報処理へと拡大してきている。また、処理形態も対話によるトランザクション処理に加え大形ホストコンピュータとの連携による垂直分散、業務の分散化に対応した水平分散処理など、集中と分散を融合したネットワーク処理へと進んでいる。これらに対し、HITAC L-700 (以下、L-700と略す。) シリーズのOS (Operating System) としてMIOS7/AS (Multiple office Information Operating System 7/Advanced System) を開発した。

MIOS7/ASの開発に当たっては、システムコンセプト¹⁾に基づき、L-700シリーズのハードウェアアーキテクチャを生かし、使いやすく高性能なシステムを実現できるように、OSの基本部(制御プログラム)についても各種の新しい機能を支援した。

特に使い勝手については、帳票や伝票フォーマットをコンピュータで処理するときの画面の定義や処理の手続きなど、非常に煩雑なプログラミング作業が、帳票認識技術によって自動的にプログラミングできる知的帳票認識プログラムを開発した。また、OPSの利用層が拡大した一方で高度利用も進んできた現在、ひとたびコンピュータの故障が発生した場合には、その影響も大きいものがある。これに対応するため、自律分散技術²⁾を適用し、システムの無停止化を確保できるエバーラン機能の技術を確立した。

本稿では、これらMIOS7/ASでの制御プログラムの基本機能、プログラムの自動生成機能およびエバーラン機能の基本技術を中心に述べる。

2 制御プログラムの基本機能

L-700シリーズのハードウェアアーキテクチャを生かし、使いやすく高性能なシステムを実現させるために、MIOS7/ASの制御プログラムについても各種の新しい機能を支援した。その主なものについて次に述べる。

2.1 ダイナミックリロケーション

プログラムを動的に再配置制御し、不連続メモリ領域をまとめて一つの論理空間として利用できるダイナミックリロケーション機能を支援した。その制御方式を図1に示す。

物理メモリを4kバイト単位のページに分割し、それらを論理空間にマッピングしている。これにより、メモリのフラグメンテーションによる使用不可能な物理メモリの最大をページサイズ(4kバイト)に抑えることができ、メモリの有効利用が可能となった。

また、論理空間は全体で4Gバイトあり、それをOSの空間(2Gバイト)とユーザー空間(2Gバイト)に分けている。ユーザー空間は図2に示すように、ジョブごとに独立した空間を設け、信頼性の向上を図っている。

ただし、ジョブ空間での論理アドレスは、互いに重複しな

* 日立製作所 ソフトウェア工場 ** 日立ソフトウェアエンジニアリング株式会社 *** 日立製作所 システム開発研究所

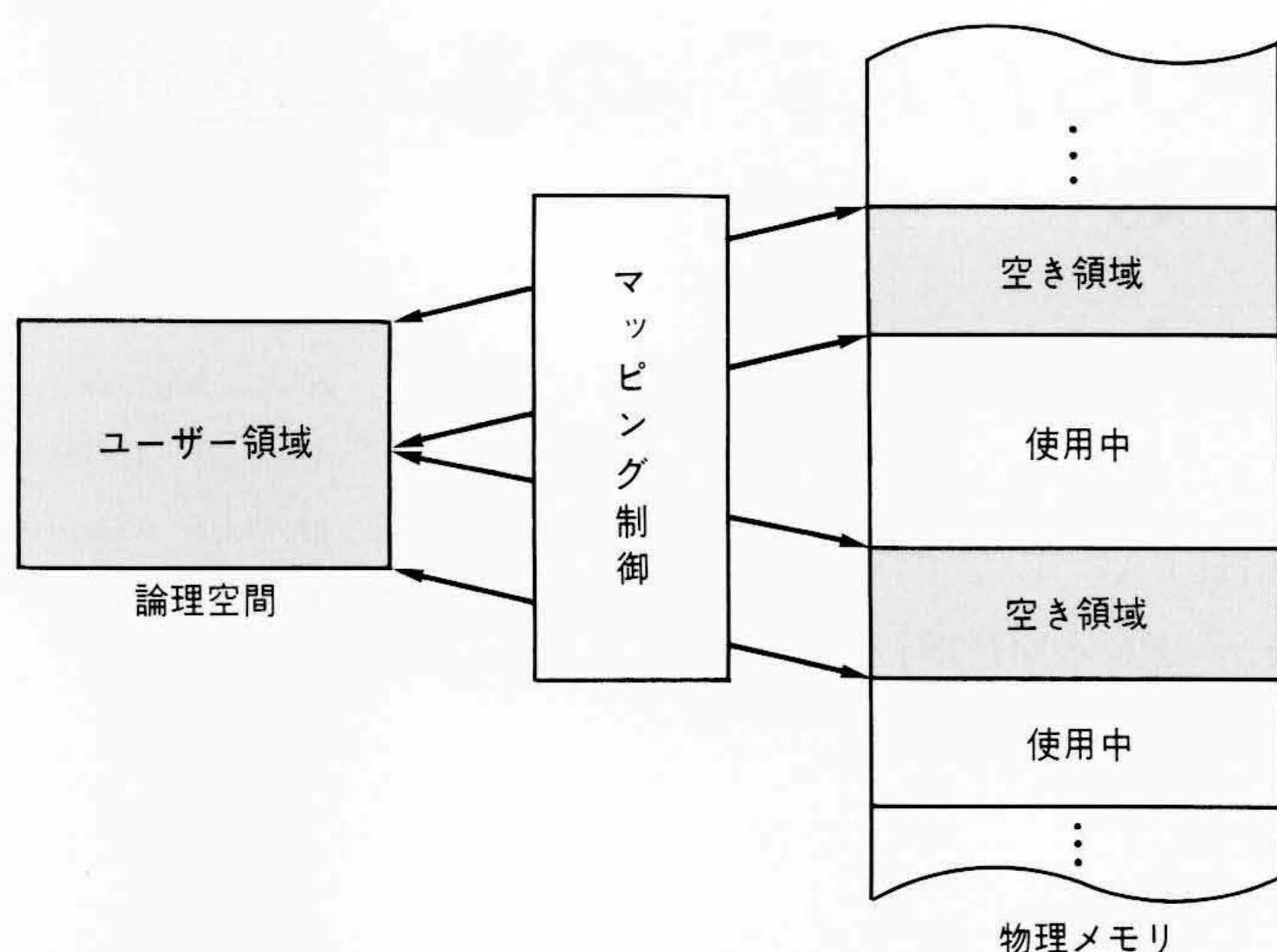


図1 ダイナミックリロケーション機能 物理メモリ上不連続であっても、マッピング制御により論理空間上に連続してマッピングしている。

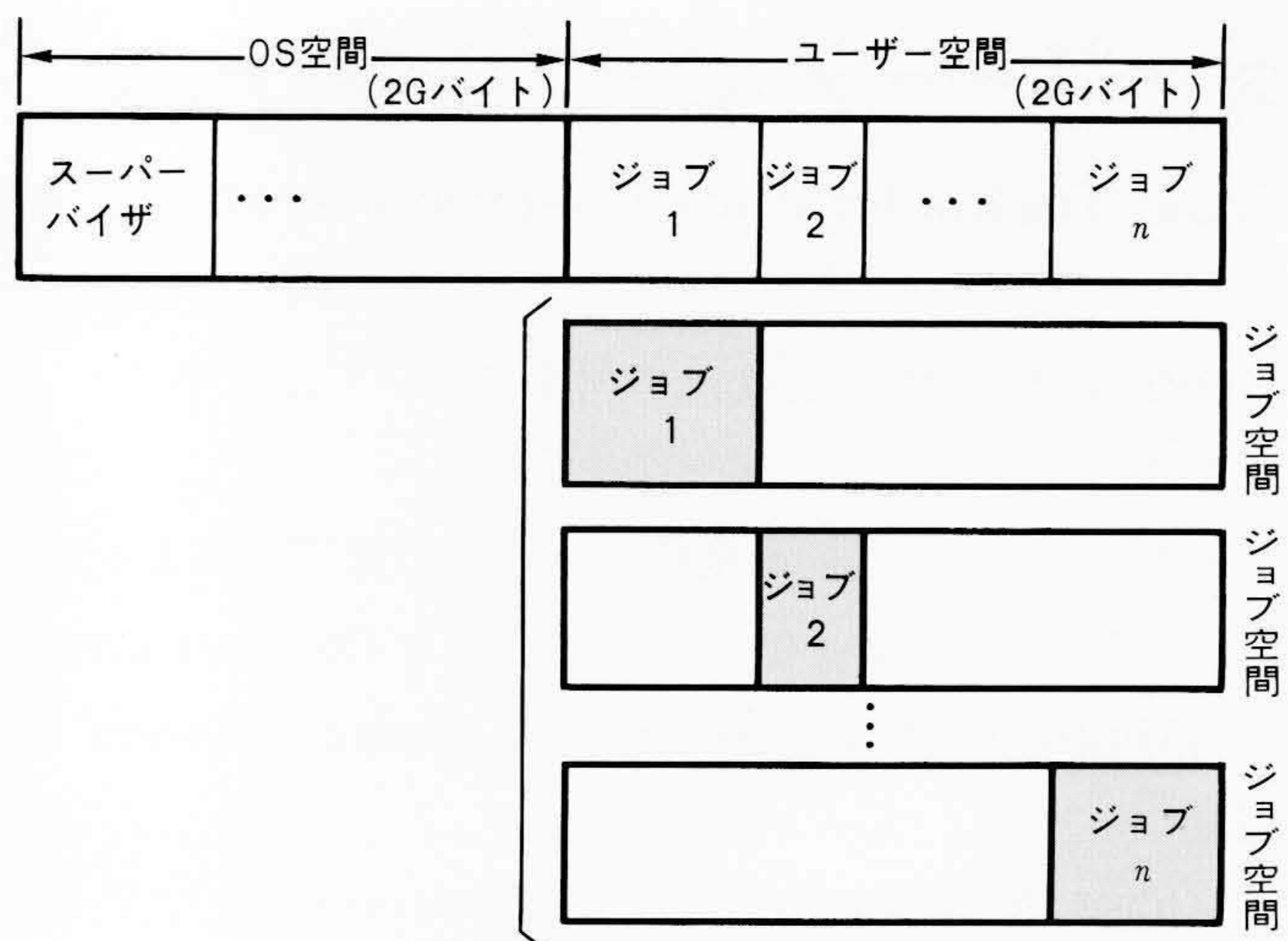
のように割り付けている。これによって論理アドレスで実行するジョブが一意に決まり、通常のマルチバーチャルで必要な空間切替制御オーバーヘッドを不要にすることができ、高性能化を図っている。本機能は、OSの基本機能として独立した機能であり、ユーザープログラムにはまったく影響を与えずに実現している。

2.2 データアクセスの高速化

L-700シリーズのハードウェアの高性能化と合わせて、データアクセス機能についても以下の高性能化を図った。

- (1) システムバッファリング
- (2) 仮想ファイル空間制御

特に仮想ファイル空間制御について次に述べる。



注：略語説明 OS (Operating System)

図2 論理空間の構成 OS空間およびユーザー空間の対応を示している。

仮想ファイル空間は、通常の仮想メモリアクセス制御をファイルアクセスに適用したものである。

ファイル利用時に、ファイル全体を仮想ファイル空間上にマッピングする。ユーザープログラムがファイルアクセス時(Read/Write)、ファイルのアドレスを仮想ファイル空間上のアドレスに変換する。この制御によって、通常の仮想メモリアクセス制御が適用できる。すなわち、仮想ファイル空間に対応したデータが物理メモリ上にあれば、そのデータをユーザープログラムに渡す。対応したデータが物理メモリ上になければ、ページングアルゴリズムによってディスク装置上にある要求されたデータを、物理メモリに読み込む。この方法により、アクセス頻度の高いデータは、自動的に物理メモリ

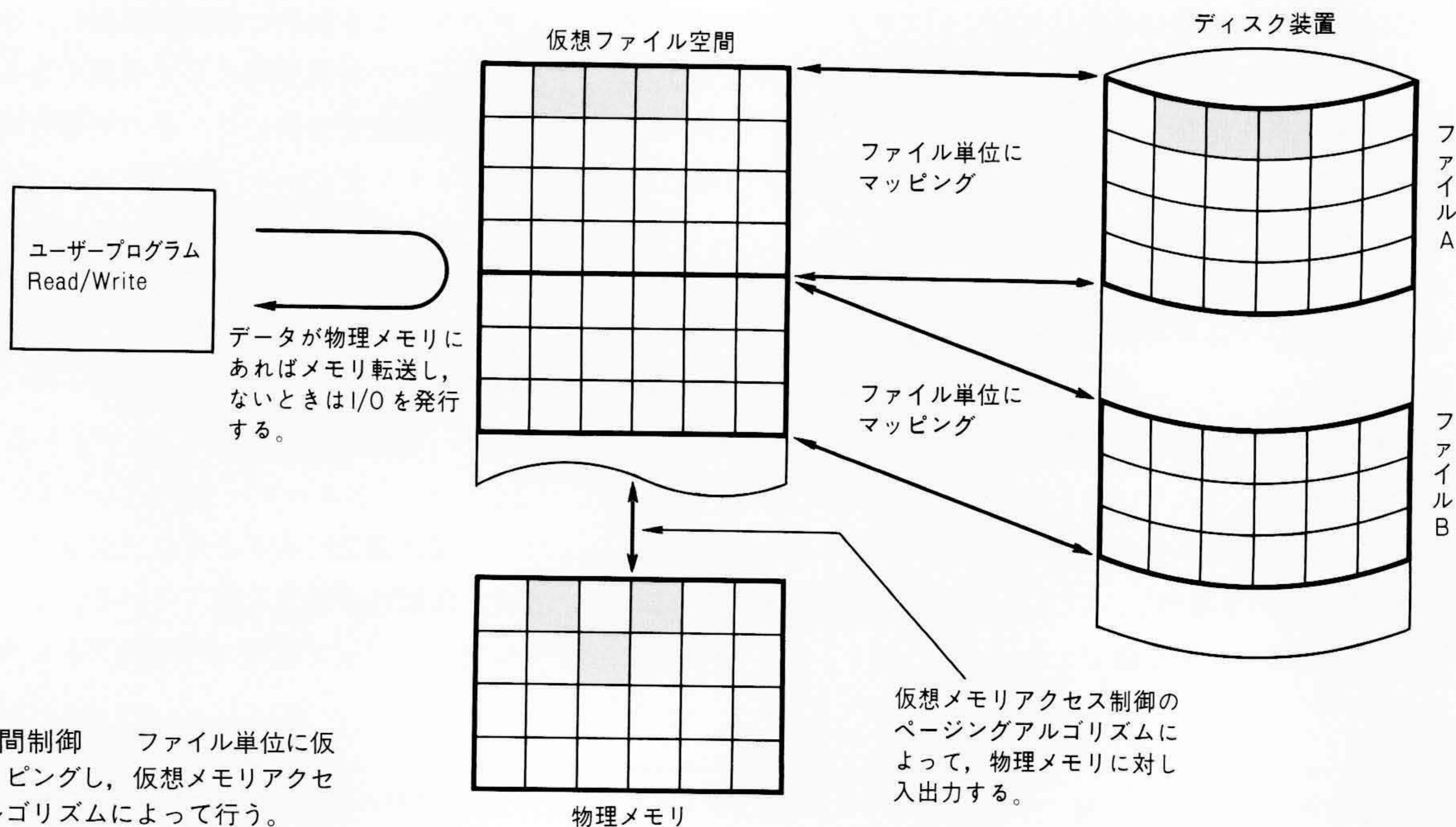


図3 仮想ファイル空間制御 ファイル単位に仮想ファイル空間上にマッピングし、仮想メモリアクセス制御のページングアルゴリズムによって行う。

上に確保され、ファイルアクセス時間を短縮することができる(図3)。

以上、制御プログラムでの主な基本機能について、使い勝手および高速化の向上という観点から新機能を述べてきたが、もちろんユーザー財産の移行性および互換性については十分考慮し、ソースプログラム、オブジェクトモジュール、ロードモジュールなどについてアップワードの互換性を持っている。

3 基本技術

2章で述べた制御プログラムの基本機能をベースとして、イメージ処理技術および自律分散技術といった高度技術の適用により、使い勝手のいっそうの向上を図っている。ここでは、第4世代言語ETOILE/OP(Effective Object Oriented Language for End Users on Office Processor)での知的帳票認識プログラム(BELIEVE)およびエバーラン機能での基本技術を中心に述べる。

3.1 知的帳票認識プログラムBELIEVE

3.1.1 概要

知的帳票認識プログラムBELIEVEは、帳票や伝票のフォーマットをイメージリーダーから入力するだけで、コンピュータ処理で使用する画面や帳票のスケッチ画面と処理の手続きを自動生成でき、プロトタイピング形の開発がより効率的に行えるプログラムである。このBELIEVEは、ETOILE/OPから起動される方式と単独で起動される方式がある。ETOILE/OPから起動された場合は、ETOILE/OP用の画面情報と処理手続きを生成し、単独で起動された場合は、COBOL用の情報を出力することができる。

3.1.2 目的

通常ユーザーの業務の大部分は、入力として画面、出力として帳票から成り立っている。現状では、この画面や帳票の形式の作成は、紙上に描かれている形式をディスプレイ画面上のカラム単位に変換しディスプレイ画面上に1カラムずつ描いていき、その後、画面上の形式に必要な入出力フィールドの位置や大きさ、入出力属性などを一つ一つ手入力で指定をしていく方式をとっている。しかし、これらの方式であると位置合わせの計算が難しく、試行錯誤しながらの作成となるため、作成工数が多くかかっているのが現状である。また、フィールドの位置や属性の指定も画面を見ながら一つ一つ指定をしていくため、作成作業に要する時間は多大なものがある。さらに、行ピッチや文字ピッチが一定でない指定伝票(図4)を使つての業務作成は、事前に印刷するフィールドの位置を指定し、ピッチを意識しながらデータストリームを作成していかなければならず、プログラミングが非常に難しいものとなっている。これらのことから、画面や帳票の作成方法やプログラミング言語を十分習得していない初心者は、手戻りが多く習熟度が低いため、即戦力とはならず特定の担当者に作業が集中してしまうことが多い。

BELIEVEは、
(1) すべて新規に作成していくよりも、ある程度出来上がったもの(プロトタイプ)を修正、変更していくほうが簡単である。
(2) 帳票、伝票の形式や文字列から、どこに何をするかがかなり理解できる。
ことに注目し、帳票や伝票の形式をイメージリーダーから入力するだけでフィールドの生成、属性の設定、各フィールド対応の処理手続きの生成、さらに認識した情報をもとにした実

図4 行ピッチの異なる伝票例
品名と品名の次のフィールドでは印字位置が異なっている。ユーザーの伝票では一般的な形であるが、ピッチの異なるストリームを生成するのはかなり難しい。

〒244
横浜市戸塚区品濃町
5 4 9 - 6

日立 太郎 殿

仕 入 伝 票
株式会社 日立製作所
〒140
東京都品川区南大井6-23-15
日立大森別館
TEL 03-765-3111

取引区分名	倉 庫 名	計 上	伝票日付	伝票番号
仕 入	日立		891020	123456

商品コード	商 品 名	課	単位/数量	単 価	金 額	消費税額	税込金額
1 A 0 0 1	事務用ハサミ		10	450	4500	135	4635
2 A 0 0 2	油性サインペン		70	200	14000	420	14420
3 A 0 0 3	ボールペン		80	250	20000	600	20600
4 A 0 0 4	自動式鉛筆削り		20	2000	40000	1200	41200
5 A 0 0 5	電動式鉛筆削り		10	5300	53000	1590	54590
6 A 0 0 6	消しゴム		100	150	15000	450	15450
合 計					146500	4395	150895

発注日 891020 発注番号 1 部 課 担当者 日立 次郎

検 1 2 3
印

行結果の印刷位置合わせを自動化し、作成コストの大幅低減、作成スピードの向上、初心者への即戦力化を図るものである。

BELIEVEの特長を以下に述べる。

- (1) 既存の指定伝票をイメージリーダから読み込むだけで画面レイアウトができ、一段とスピーディに業務開発ができる。
- (2) 煩わしい定義なしで指定伝票印刷(細かな行，文字ピッチなど)ができる。
- (3) コーディングレスでのプロトタイピング開発ができる。即実行によって動作イメージの早期把握ができる。
- (4) 自動生成したプログラムは，ETOILE/OPまたはCOBOLで作成したフォーマットと同じであり，自由に修正ができる。
- (5) 自動生成中，途中の結果を表示し，途中での修正が可能である。より正確なプロトタイプの生成ができる。
- (6) ワードプロセッサで帳票や伝票の形式を作成し，これを使うことによってプロトタイプの作成ができるため，開発のための開発マシン占有が少なくて済む。

3.1.3 帳票認識の方式

帳票や伝票の形式は、人が見ればどこがフィールドとなりうるかがすぐに理解できる。例えば、通常伝票の形式を見れば、「住所」の右横が住所フィールドであり、「年」の左横、「月」の左横、「日」の左横、「氏名」の右横がそれぞれフィールドとなることがわかる。フィールドの認識は、以下のことから認識が可能となる。

- (1) 枠内は、入出力フィールドとなりうる。
- (2) 枠内に文字がある場合は、文字列の内容で枠内にフィールドが必要か否かが判別できる。
- (3) 枠外でも文字列の内容で、フィールドが必要か否かが判別できる。
- (4) フィールドがどこにできるかは、文字列の内容によって判別できる。例えば、**図 5**の①は、「品名」を処理するフィールドであり、②は、文字列から「年」の左と「月」の左と「日」の左に年月日処理するフィールドが必要であることがわかる。

フィールドの位置もまたこの文字列によってある程度推測できる。例えば、図6の「年月日」は右側であり、「年」、「月」、「日」と分かれている場合は左側である。「××銀行××支店」とある場合は、フィールドは不要であるが、④のように「銀行」、「支店」の文字が離れている場合は、その間にフィールドが必要となることがわかる。

これらを正しく認識し、作成した入出力フィールドに対応して処理の手続きを組み立てるためには、その文字列が何を
 するのかという意味、内容の理解が重要である。

一般に帳票や伝票の用語では、その用語の語尾の単語ほどその用語の内容、意味を表している。例えば、「前期売上金額」は、語尾の「金額」が性質を表している。図7でみると、「前期支払金」は語尾の「支払金」から「支払」を、「その他払出

納 品 書

住 所
氏 名

②

平成 年 月 日

品 名	単 価	数 量	金 額
①			
計			

図5 伝票の形式 一般的な伝票の形式である。フィールドの処理する項目は、接している。枠内の文字列の内容によってわかる。

① 年 月 日

② 年 月 日

③ 日立銀行 横浜支店

④ 銀行 支店

図6 フィールドの位置 用語によって一般的なフィールドの位置が決まる。これは、漢字特有な性質を利用することによって理解できる。

①	②	③	④
前期支払金	前期収入金	その他払出金	臨時入金

⑤ 支払合計

語尾の単語の性質で判断

① 前期支払金 → 支払

② 前期収入金 → 収入

③ その他払出金 → 支払

④ 臨時入金 → 収入

性質

↓

⑤ 支払合計 = 前期支払金 + その他払出金

図 7 処理手続きの生成例 用語そのものが知識ベースにあればよいが、ない場合は語尾の性質で判断する。マスタファイル内のフィールドの関連づけも同じである。

金」も語尾の「払出金」から「支払」を意味している。この性質を利用して処理を行う。同図の⑤の支払合計のフィールドを考えると、このフィールドは「支払」の「合計」を求めることであるから、各項目の中から「支払」の性質を持つものを選べばよい。各項目の語尾の単語から「前期支払金」と「その他払出金」が支払の性質を持つものであると理解し、この二つの合計を支払合計に表示する処理手続きが生成される。BELIEVEは、これらの規則を知識辞書としてまとめ、これを利用して処理を組み立てていくことを行っている。すなわち、BELIEVEの認識方式は、以下の2点に要約される。

- (1) 帳票や伝票の規則や業務の一般的な規則、日本語の性格などを知識辞書としてまとめ、これを利用して帳票や伝票の様式や意味を理解する。
- (2) 文字列の切り出しや性格を理解する手段として、業務の一般用語の中から意味、内容を持っている単語を知識辞書としてまとめ、文字列の語尾単語ほどその内容を表すことを利用し、フォーマット上の文字列と知識辞書の語尾マッチング

により意味の理解を行う。

3.1.4 BELIEVEの機能

BELIEVEの流れを図8に示す。まず、イメージリーダーから用紙を読み込みます(同図の①)と読み込んだ形式を認識し、画面に表示する(同図の②)。ユーザーは表示内容を確認し、修正があればその場でけい線、文字などを修正できる。このとき、形式情報が認識情報として格納される。認識が終了すれば「続行」キーを入力することによってフィールドの自動生成に入る。認識が終了し、フィールドが自動生成されると画面にフィールドを伴って形式が表示される。フィールドは、リバーズ表示される(同図の③)。ここで出来上がっているフィールドの位置、属性、フィールド名称の確認を行う。属性や名称は、該当のフィールドにカーソルを位置づけることによって内容がポップアップ表示され、その場で内容を確認、修正できる。また、フィールドは結合、分割、移動、削除が行える。フィールドの確認が終了したら「続行」キーを入力し、行・けたモード画面の生成に入る。このとき認識フィー

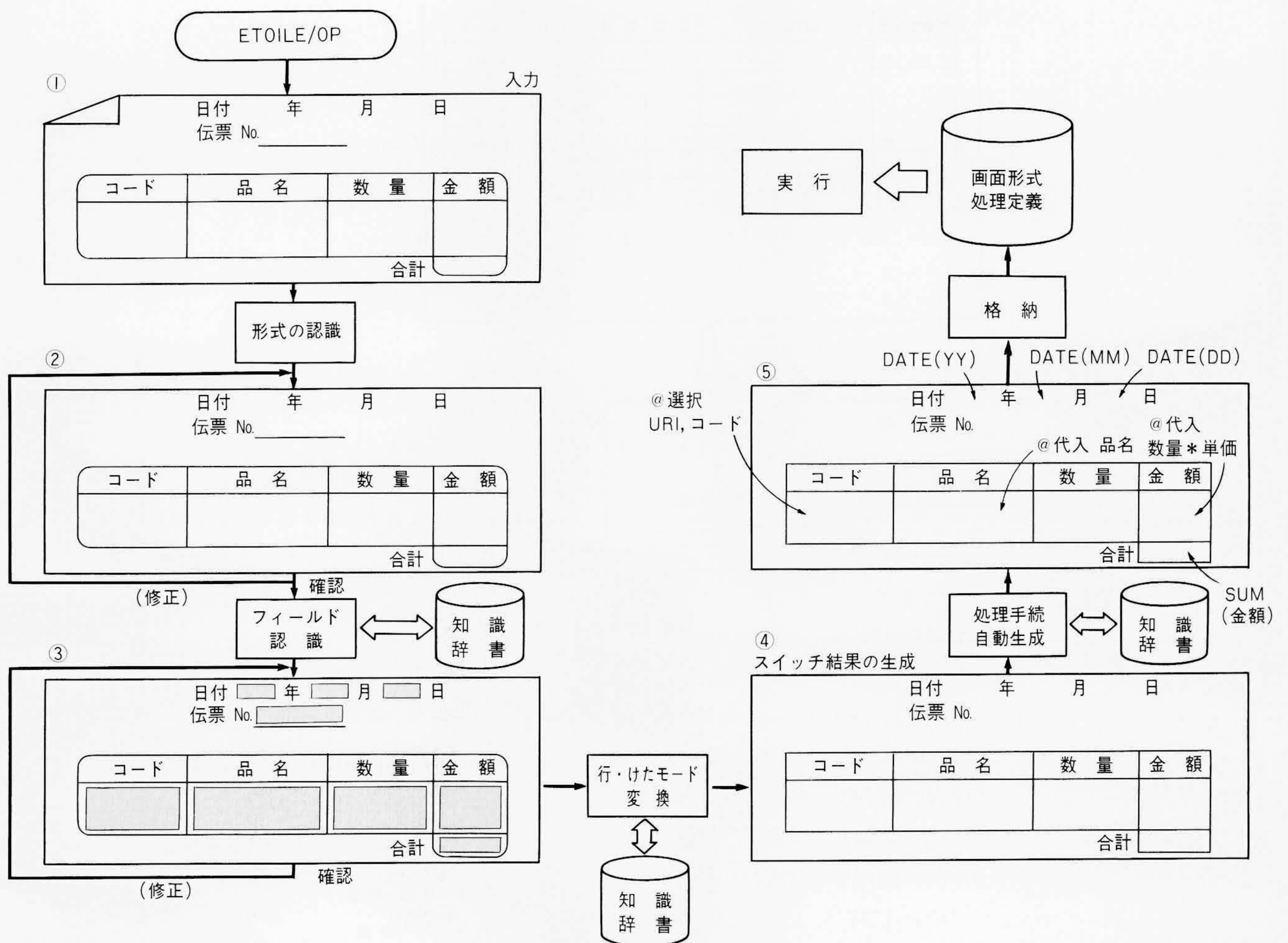


図8 帳票認識による自動生成処理の流れ 伝票や帳票のフォーマット用紙を入力してから処理手続き生成までの流れである。生成処理中の途中表示をしないダイレクトモードもある。

ルド情報が格納される。行・けたモードの画面生成処理では、ミリメートル単位の画面表示から従来の行・けたの画面を自動生成する。このとき、画面内のセンタリングやなるべく画面内に入るように調整を行うことができる(同図の④)。その後、行・けたモードの画面生成処理では、ETOILE/OPの画面、帳票情報またはCOBOLの画面、帳票ソース情報を出力する。

ETOILE/OPからBELIEVEが呼ばれた場合は、さらにプログラムの自動生成に移る。プログラム自動生成処理では、認識したフィールド情報と知識辞書をもとに処理を組み立てていく。作成した処理は、ETOILE/OPの処理定義として出力され、修正、追加はそのままETOILE/OPの変更機能で行うことができる。作成する処理の定義例を図9に示す。

作成した業務を実行する場合、実行開始時に認識情報ファ

イル名称を指定すると、データを印刷するときに行列のピッチを何も考えずに読み込んだ形式の所定の位置に印刷することができる。

その他の機能としては、途中の認識結果画面を出さずに一気に処理の手続きまで生成するモードや帳票の引き伸ばし(2倍まで)を行うことができる機能がある。A3サイズの帳票は読み込めないためA4に縮小し、これを引き伸ばし率の指定で大きく作成することができ、便利な機能である。

3.1.5 知識辞書

BELIEVEで認識やプログラムを生成するとき使用する一般規則や演算のルールは、知識辞書として格納されている。この知識辞書は標準提供されるが、辞書の内容の追加および変更に対しても対話形式で簡単に行うことができるようになっている。したがって、BELIEVE実行前にこの知識辞書の内

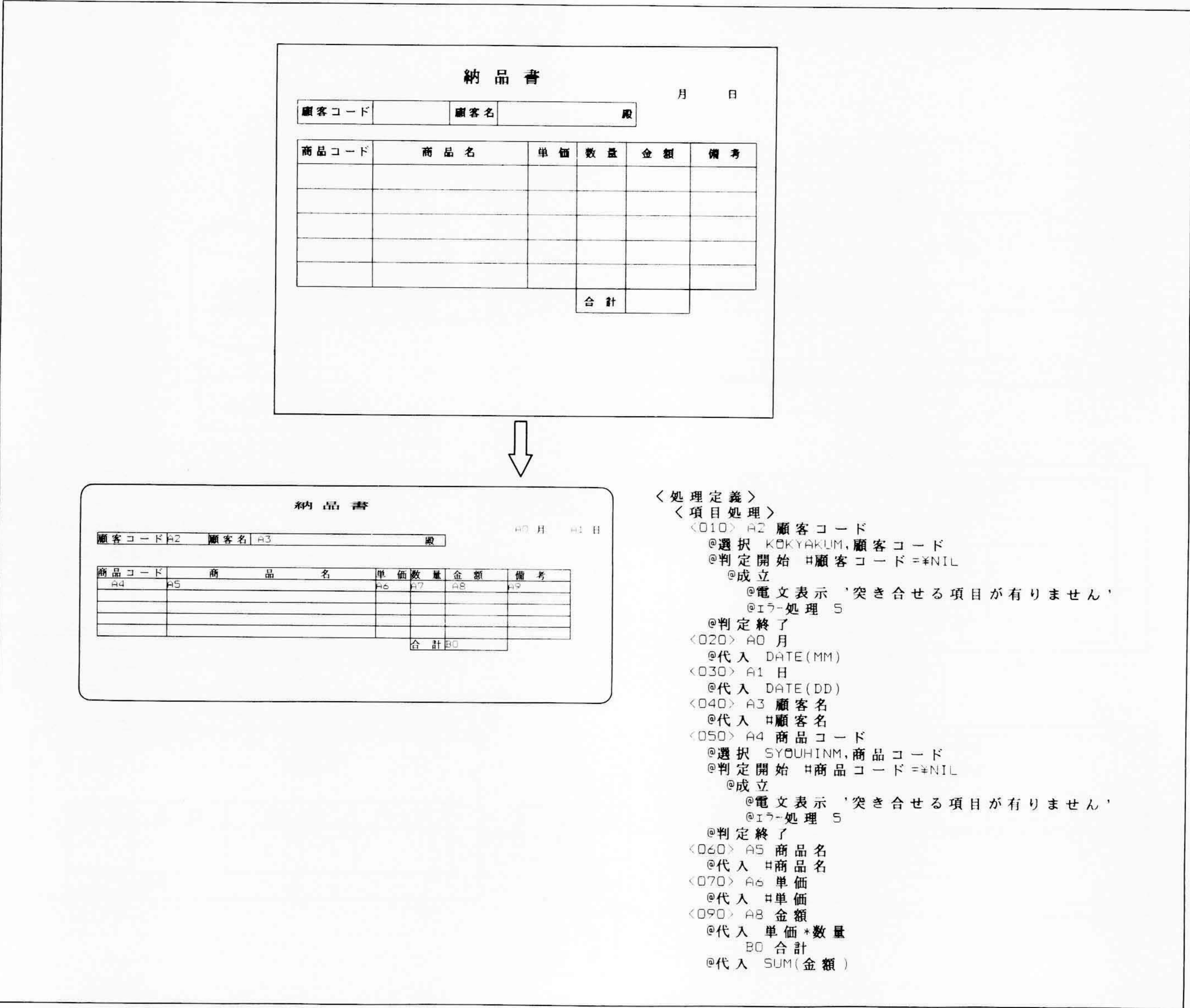


図9 生成手続き例 知識ベースの中の各項目に対して、処理ルールが何もないときの作成手続きである。処理のルールがあれば、上記の作成手続きにさらに各項目の手続きが生成される。

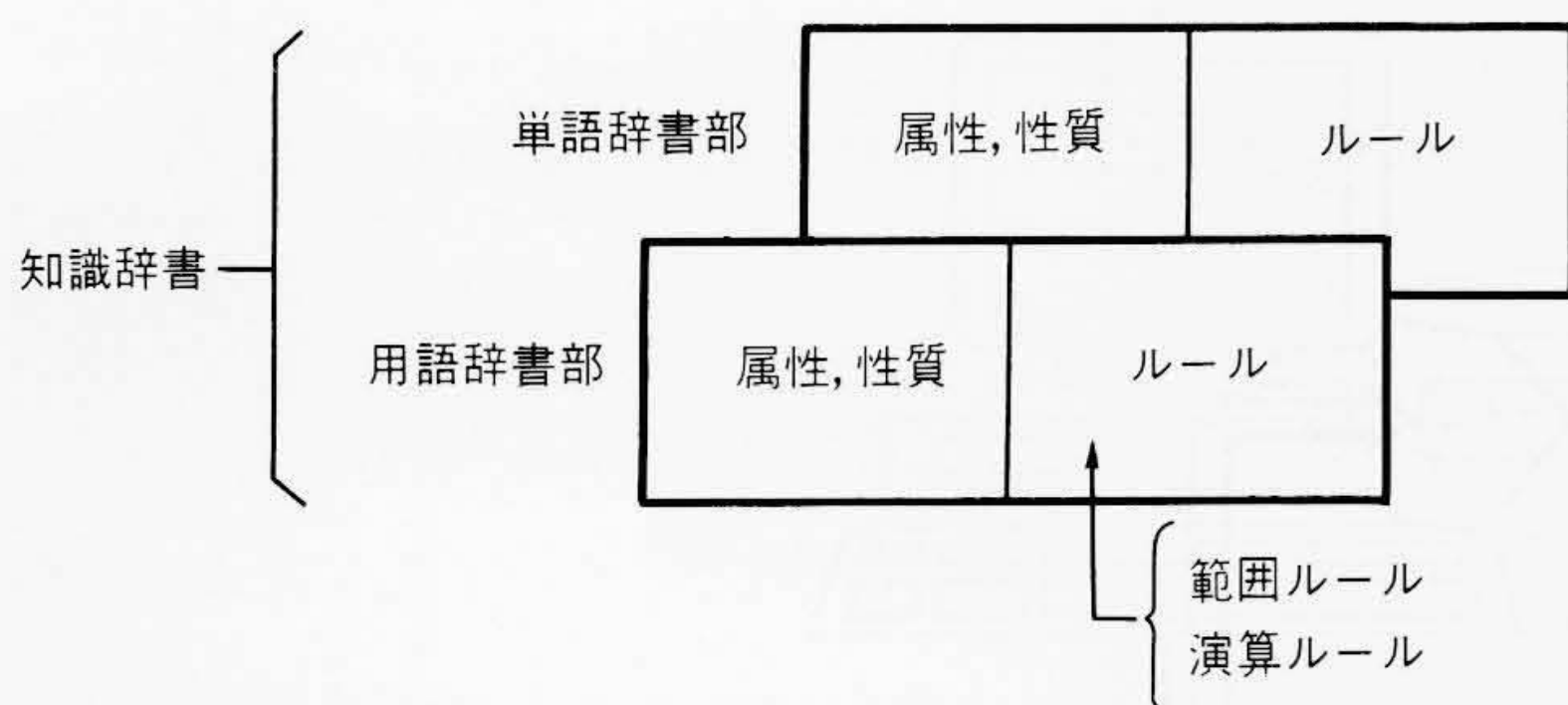


図10 知識辞書の内容 用語辞書部と単語辞書部は同じ形式である。ユーザーは、この知識辞書をあらかじめ業種対応にメンテナンスしておくことによって、認識率を向上させることができる。

容を利用する業種、業務用に利用者が追加、変更すれば、それだけ認識率を向上させることができ、より正確なものができる。この知識辞書は用語辞書部と単語辞書部とで成り立っており、各辞書部はさらに属性、性質部とルール部に分かれている。ルール部は範囲ルールと演算ルールから成り立っており、用語、単語対応に、ユーザーがチェック範囲と演算処理をルールとして設定しておくことができる。知識辞書の形式を図10に示す。

3.2 エバーラン

3.2.1 ねらい

通信回線やLANで接続されたOPS間でのリソースの共用を実現するための機能として、水平分散機能が提供されている。水平分散機能は、スタンドアロンシステムの延長でファイルアクセス(リモートファイルアクセス)やプログラムの実行(仮想ワークステーション)を可能にするものである。

MIOS7/ASでは、これら機能に加えて複数のOPSがLANで接続された構成を対象とし、ユーザーがシステムの物理構成を意識することなくシステム内リソースを利用できるようにするエバーラン機能を提供する。エバーラン機能は、システム内の障害発生や拡張、保守などによるシステム構成変化時にも、ユーザーが実行中の業務を止めることなく、そのまま業務を続けられるようにすることを目的とするもので、分散環境下でのOPSの使い勝手向上をねらうものである。

エバーラン機能は、日立製作所独自の技術である自律分散システム技術に基づくものである。自律分散システムとは、システムを構成する各サブシステムが、他のサブシステムに依存することなく、おのおの独自の判断で自処理の実行、また他サブシステムとの連携を行うことを可能とするシステムである。このようなサブシステムの自律化を達成するため、自律分散システムでは、データフィールド(データの流れる場)という概念を導入している。各サブシステムはデータフィールドに接続され、データフィールド上を流れるデータの中からみずからに必要なデータを判別し、自内に取り込み処理を行う。また、処理結果のデータは、そのデータを受け取る相手(あて先)ではなくデータの内容を示すコードを付け、デー

タフィールドに送出する。このようなアーキテクチャをとることにより、各サブシステムは他サブシステムの障害や拡張、保守などの影響を受けることなくみずからの処理を行うことが可能となる。すなわち、自律分散システムは、いつ、どこに、いくつでもサブシステムを追加・拡張でき、また障害があったとしても他サブシステムに影響を与えることのないアーキテクチャを持つシステムであり、従来の高信頼化をねらった多重化システムとは異なるものである。以上示したデータフィールドは、LANに、またサブシステムはLANに接続するプロセッサに対応する。

今回、MIOS7/ASで提供するエバーラン機能は、この自律分散アーキテクチャに基づき、特に耐故障性を実現するものである。なお、エバーラン機能は現行の水平分散機能と共存して使用することができる。以下に、今回提供するエバーラン機能について説明する。

3.2.2 エバーラン機能

(1) エバーランファイル機能

エバーランファイルは、上記で示したデータフィールドに接続するファイルであり、基本的に任意の多重度に多重化することが可能なファイルである。ただし、OPS分野では二重化で十分であり、今回は特に三重化以上の多重化については支援しない。また、従来の多重化方式では、ディスクボリューム全体を多重化する方式がとられているのに対し、本機能では個々のファイル単位での多重化を可能とする。これにより、例えばマスタファイルなどの重要なファイルだけを二重に持ち、他のファイルは一重に持つといった構成が可能となり、必要最小限のディスク容量で、システムに要求される信頼性を満足させることができる(図11)。

(2) エバーラン対話機能

データエントリなどの業務の実行途中に障害となった場合に、その業務を最初からやり直すことは、使い勝手の面から好ましくない。業務実行途中に障害が発生した場合でも、他のOPSで実行途中時点から業務を引き継げるようにしたのがエバーラン対話機能である(図12)。例えば、対話の業務実行中に使用しているOPSの画面がアクセス不能となった場合、別のOPSに移り、そのまま業務を続けることができる。このとき、それまで使っていた画面を引き継いでそのまま業務を続行できる。

さらに、画面の障害にかかわらず業務実行途中に別のOPSに移動した場合でも、移動先のOPSを用いて実行中業務をそのまま引き継ぐことができる。

本機能により、単にオペレータ負荷を軽減するだけでなく、空いているOPSの有効利用が可能となる。

4 結 言

システムコンセプトに基づき開発したOSであるMIOS7/AS

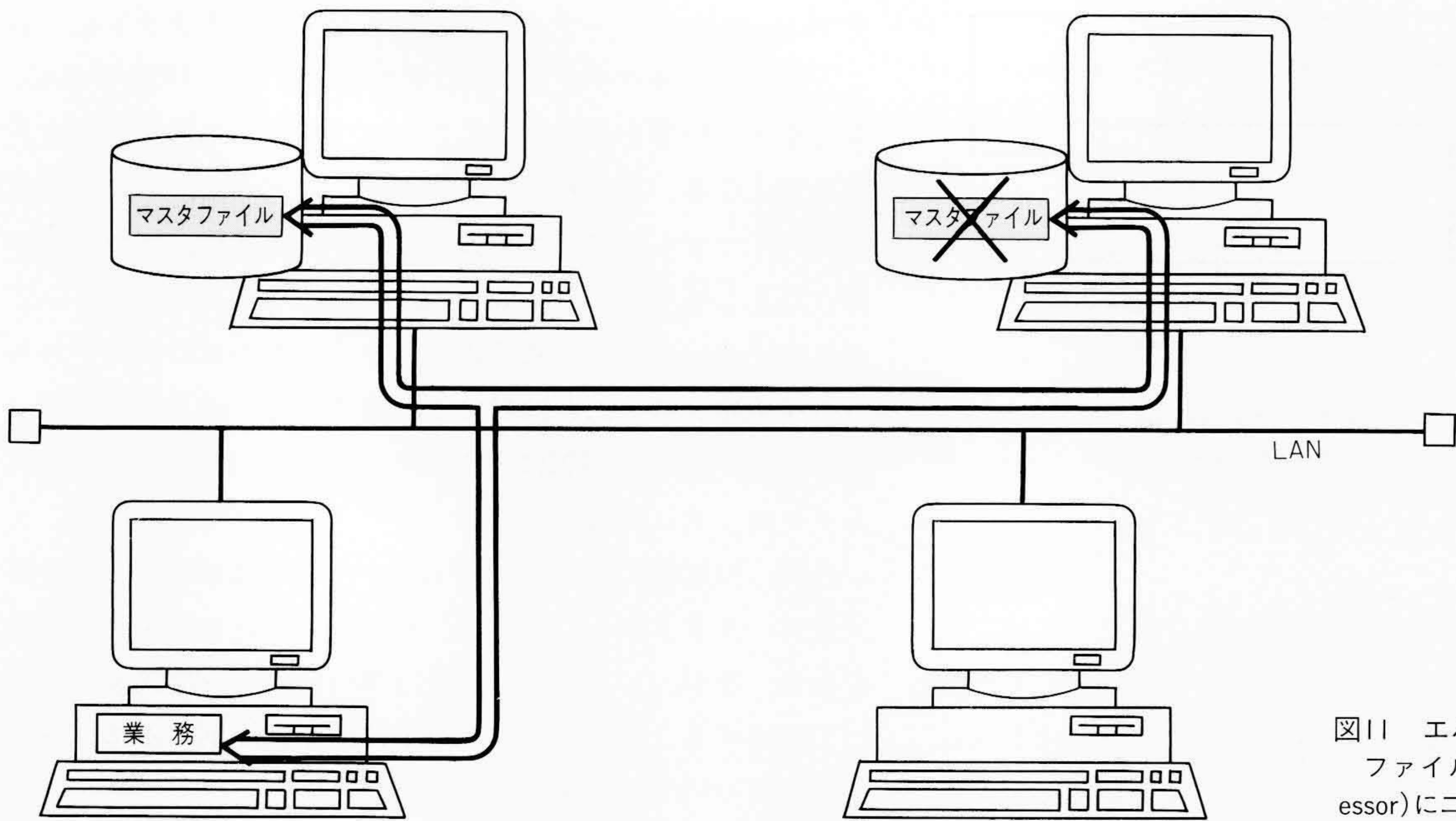


図11 エバーランファイル機能
ファイルを別々のOPS (Office Processor) に二重化して管理できる。

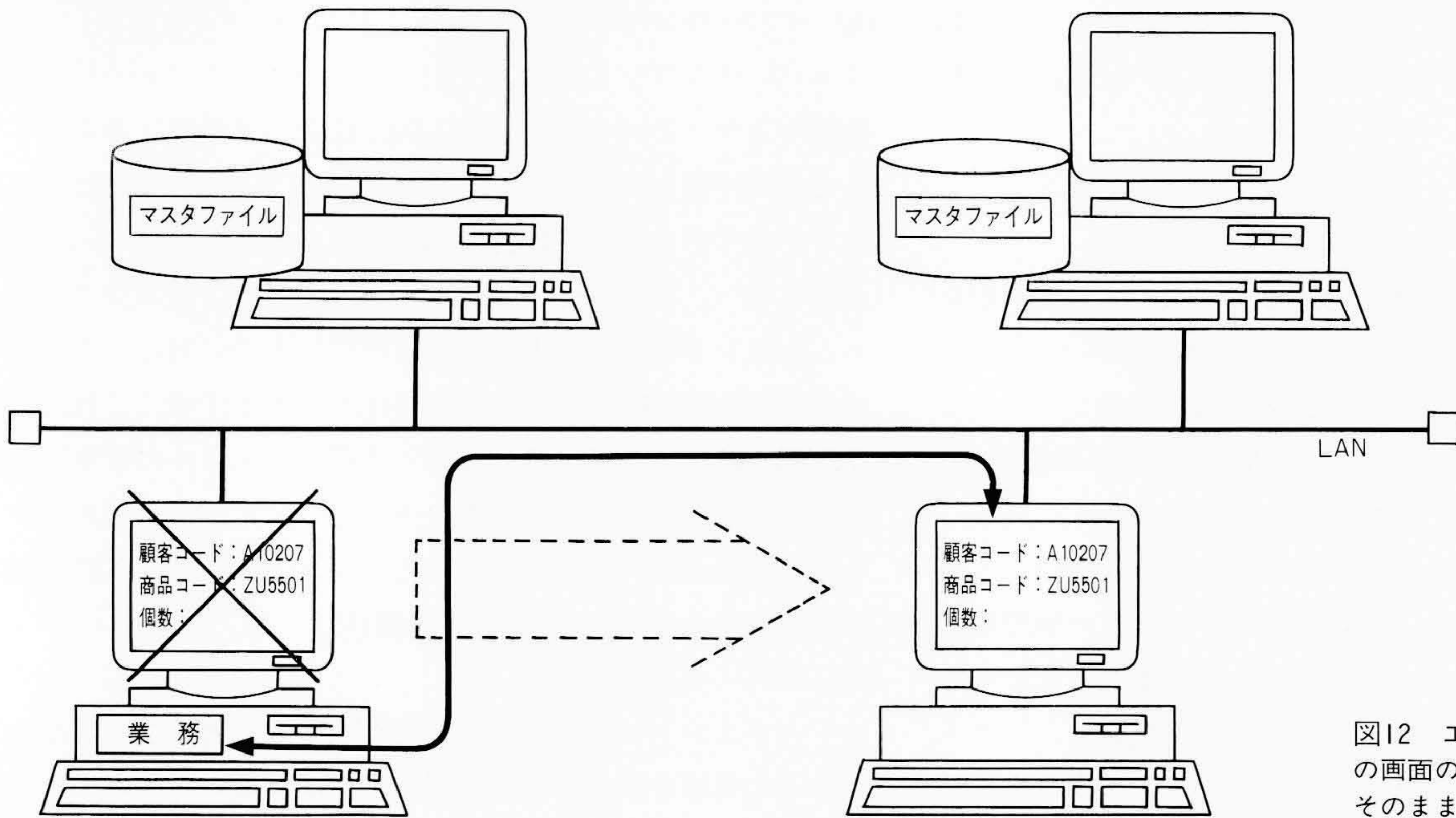


図12 エバーラン対話機能 OPS
の画面のアクセス不能時、別のOPSに
そのまま画面を引き継げる。

の主な新機能について、その基本技術という観点から述べた。

L-700シリーズのハードウェアアーキテクチャを生かし、使いやすく高性能なOPSとするためにMIOS7/ASの制御プログラムについては、ダイナミックアロケーション機能およびデータアクセスの高速化機能を実現した。

特に、使い勝手の向上ということで、帳票認識技術を用いた第4世代言語ETOILE/OP向けの帳票自動生成プログラムにより、今後の業務プログラムの開発効率の大幅向上が期待できる。また、自律分散技術を適用したエバーラン機能により、システムの無停止化と高度な負荷分散が可能となった。

OPSの分野では、使い勝手が重要なファクターであり、こ

れらの新技術の確立により、使い勝手のいっそうの向上を図ることができたと確信している。

MIOS7/ASでは、今後とも使い勝手の向上のために新技術を積極的に取り込み、より戦略的な情報システムの構築ができるよう充実させていく考えである。

参考文献

- 1) 鈴木, 外: HITAC L-700シリーズオフィスプロセッサの開発のねらいと特長, 日立評論, 71, 11, 1125~1130(平1-11)
- 2) 森, 外: 自律分散概念の提案, 電気学会論文誌C, 104巻, 12号, 303~310(昭59-12)