

オブジェクト指向データベース ワークシートプログラムの構造

渥 美 亮

三井造船(株) 技術総括本部

手によるオフィスワークの分析を進め、これにもとづいて、先に概念を示したツールの構造を提案する。先に、文書・作業／サービスを含めた部品展開、部品表内の住所を定める形の分類、文書の帳表・台帳・管理台帳への還元、データと手順をもつワークシートによる、手によるオフィスワークを現実の世界のモデルとする、手作業をそのまま画面上の作業にするツールの概念を提案した。本稿では、手によるオフィスワークの文書体系では管理台帳により詳細に管理が行われていること、管理の階層を浅くするための努力がなされていること、時々刻々の変化に対応するための努力がなされていることを示し、これにもとづいて上記ツールの構造を提案する。

THE STRUCTURE OF AN OBJECT ORIENTED DATA BASE  
WORKSHEET PROGRAM

Ryo Atsumi

Corporate Technical Research & Development Headquarters,  
Mitsui Engineering & Shipbuilding Co., Ltd.  
6-4, Tsukiji 5-Chome, Chuo-ku, Tokyo, Japan

The structure of tool, the real-world of which is office work by handwriting, is proposed. The concept proposed involves explosion including document and work/service; classification defining address in the bill of materials; document reduction into three categories: complete table, list and control register; and a worksheet containing both data and procedures. In this report, it is shown that the control register controls documents in detail, and efforts are made to keep shallow layers and to cope with continuous occurrence of changes. Based on the analysis, I propose the structure of tool which transplants handwriting work as is to work on the display.

## 1 はじめに

これだけシステム化・OA化を進めているのに、現実のオフィスワークでは机の上を書類の山にしているのが実情である。このため、実際のデスクワークから出発して、手なれたそして洗練された今の手順をそのまま画面上の作業にする、新しいシステム像の登場が期待される。

筆者は、手によるオフィスワークが

- (1) 単位組織毎の分散処理
- (2) ホロニックアーキテクチャ
- (3) 高度な情報処理
- (4) シースルーオープン
- (5) 暫定処理

という特徴をもち、さらに手によるオフィスワークにおいては

- (1) 情報（文書）、作業／サービスを含んだ形で部品展開出来ること
- (2) 予め部品表内の住所を定める形（部品表アドレス）で分類出来ること
- (3) 帳表・台帳・管理台帳の3種類で情報（文書）が包含出来ること
- (4) データと手順を一体化したワークシート（オブジェクト）が有効であること

を示し、これにもとづいて、現実世界のモデルとして手によるオフィスワークを撰定した、手作業

をそのまま画面上の作業にするツール、ワークシートプログラムなる概念を提案した [1]。

本稿では、ワークシートプログラムを実現するため、さらに手によるオフィスワークの分析を進め、ワークシートプログラムの構造を得たので、ここに報告する。

## 2 現実の文書体系

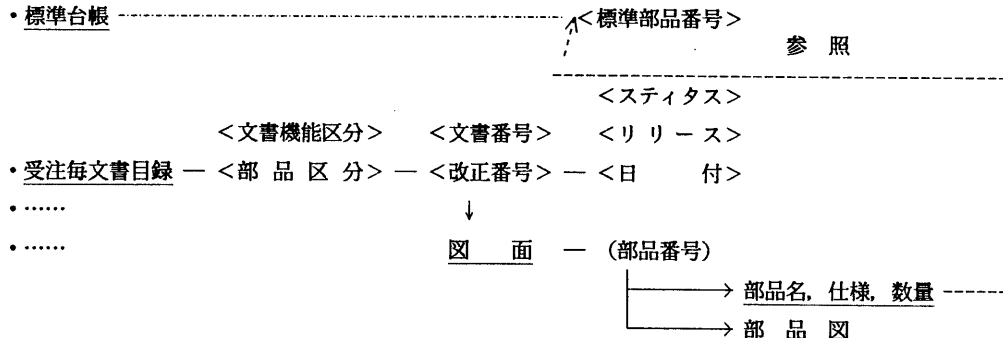
通常、手によるオフィスワークでは、第1図に示すように、単位組織毎に

1. 標準台帳、受注台帳、受注毎文書目録などの管理台帳を用意する。
2. 必要に応じて、文書機能区分、部品区分などの分類を管理台帳に設ける。
3. 文書番号、改正番号などにより識別して文書を管理台帳に登録する。
4. ステータス、リリース、日付などの管理のための属性を記録する。
5. 図面の場合、部品表を用いて、部品番号をインデックスとして、部品名、仕様、数量、部品図を特定する。このとき、標準品などのように、具体的内容は他の文書に記載し、これを参照するだけの形式が存在する。

この文書体系のなかで、次のような工夫がされている。

### 単位組織

- 受注台帳
- 標準台帳



第1図 現実の文書体系

### (1) 浅 層 化

管理の階層を浅くするためおよび一覧性向上のため、分類コードに工夫をして、一管理台帳あるいは一文書内の要素を多くする努力が払われている。

例えば、管理台帳レベルでは日常使用する名称を分類コードとする、区分レベルでは作業に従事していれば覚えてしまう形にする、文書レベルでは部品表ないしは相当するものを設けて情報を集中する、同一管理台帳の他の図面を参照するというものがあげられる。

### (2) 時 系 列 化

時々刻々の変化に対応するため、発行した文書のオリジナルおよび管理台帳の該当レコードを保持することが行なわれている。ここで改正番号が異なる同一文書番号の複数の文書が、次のような場合には、共存すること、および過去の姿を再現するためには最新の文書

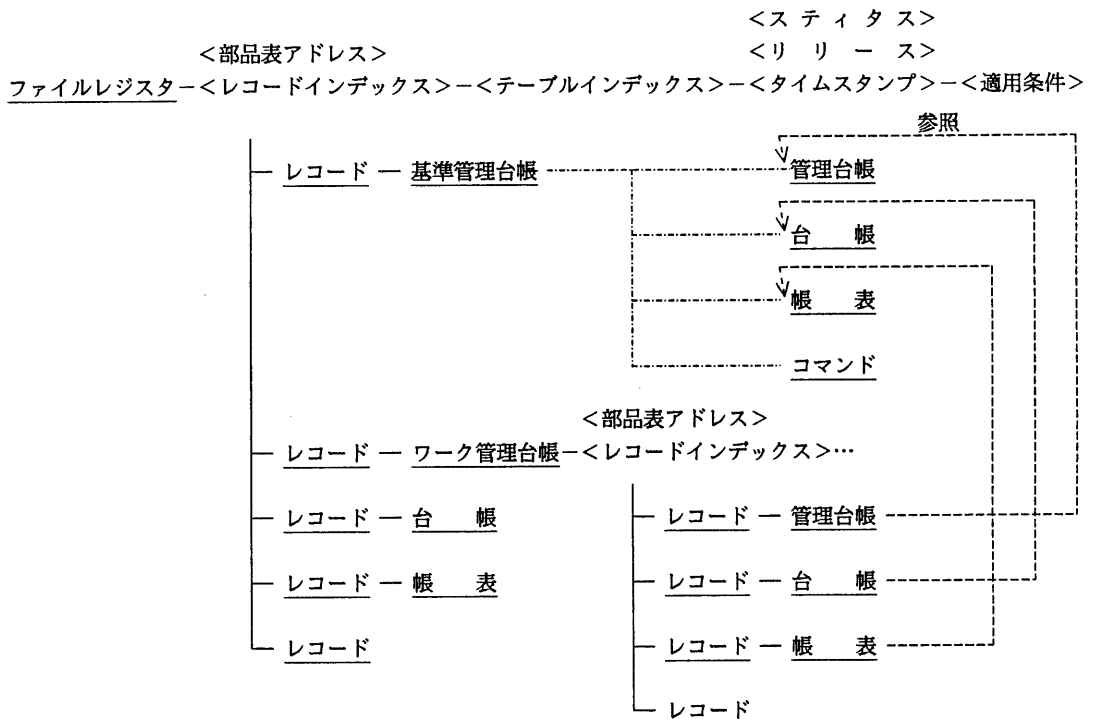
のみでは不十分であることに注意されたい。

- 発効日指定    ～情報伝播の遅れを吸収する、あるいは既にスタートしている活動を効力から除外する手段
- 範囲指定      ～特定地域・顧客・注文を特別扱いする手段
- 号機管理      ～同一図面で複数製造するとき、例えば、1号機完成後、2号機以後を変更する手段

### 3 データの構造

現実の文書体系を実現するため、ワークシートプログラムではデータの構造を第2図に示すものとする。

1. 帳表・台帳・管理台帳の3種類のテーブル／表をもつ。



第2図 データの構造

2. 頂点の管理台帳としてファイルレジスタを設ける。
3. ファイルレジスタを含むすべての管理台帳は、そのレコードの中に帳表・台帳・管理台帳のインデックスを持ち、これらの管理を行なうことが出来る。
4. ファイルレジスタを除くすべての管理台帳は、そのレコードの中で、自己あるいは他の管理台帳の支配下にある帳表・台帳・管理台帳を参照出来る。
5. 管理台帳のレコードは、取消・復活操作を除いて、一度登録された後は変更出来ない。
6. 管理台帳のレコードは、時系列管理のために、タイムスタンプをもつ。
7. 管理台帳のレコードは、レコードインデックスの他に、分類コードである部品表アドレスをもつことが出来る。
8. 管理台帳のレコードは、管理パラメータとして、リリース、ステータスさらには適用条件をもつことが出来る。
9. コマンドは、帳表の一種として、管理台帳の支配下におく。

#### 4 テーブル／表の構造

前記のデータ構造の単位である帳表・台帳・管理台帳そしてコマンドは第3図に示す構造とする。

##### (1) 単 構 造 化

帳票・台帳・管理台帳そしてコマンドはすべて同一構造とし、これを単位テーブルと呼ぶ。単位テーブルは単数ないしは複数種類の単数ないしは複数のファイルと1対1に対応するテーブルから構成する。後者を要素テーブルと呼ぶ。

1. 要素テーブルはワークシートプログラム固有のテーブル（WSPテーブル）あるいはワークシートプログラムの配下でない外部システムのテーブル（NonWSPテーブル）のいずれであってもよい。

2. WSPテーブルは次に述べる複数種のテーブルのすべてあるいは幾つかで構成するが、これらに限定するものでなく、必要に応じて追加出来る。
3. テーブルを固定部、可変部に分けて、それぞれを独立させて、フォームテーブル、データテーブルとすることが出来る。固定部の重複格納・内部領域割当を回避すること、1回の固定部の変更をすべての単位テーブルに及ぼすことが可能となる。
4. WSPテーブルの自由度を増すため、WSPテーブルをフォーマットによる構成とし、これをフォーマットテーブルに納める。フォーマットテーブル中のフォーマットはファイル領域・内部領域・画面さらには外部入出力毎に作成格納することが出来、さらに、ファイル領域対応のものを除いて、追加格納することも可能である。
5. ユーザの記述した手順は、コンパイルして、プロセス実行モジュールとして格納する。このとき、同時にプロセス内部変数のためのプロセス内部テーブルを設け、プロセスをリエントラント可能なものとする。
6. フォームテーブル、データテーブルの任意のデータ項目を指定し、これによるインデックステーブルを複数種類もつことが出来る。とくに、台帳・管理台帳についてはレコードインデックスをキーとするインデックステーブルが必須である。
7. 単位テーブル上でワークするために必要な参照テーブル呼出しのための検索条件は、レファランステーブルを設け、その中に格納することが出来る。
8. 単位テーブル上でワークに使用した参照テーブルを明示するため、引用テーブルリストを設け、記録することが可能である。
9. 大容量のファイルの処理のため、中間ファイルを設ける。ただし、保管することは出来ない。
10. WSPテーブルの要素テーブルの一つとし

てNonWSPテーブルを含めることが可能である。

## (2) 仮想化

単位テーブル（帳表・台帳・管理台帳・コマンド）はテーブルインデックスでくくられた複数種類かつ時系列上の複数の要素テーブルから構成される仮想的存在であり、実体をもたない。

さらに、台帳・管理台帳では、レコードもレコードインデックスでくくられた時系列上の複数のレコードから構成される。したがって、管理台帳レコードで支配される単位テーブルはレコードインデックスおよびテーブルインデックスで二重にくくられた要素テーブル群の複合体となる。

## (3) 実体化

単位テーブル（帳表・台帳・管理台帳・コマ

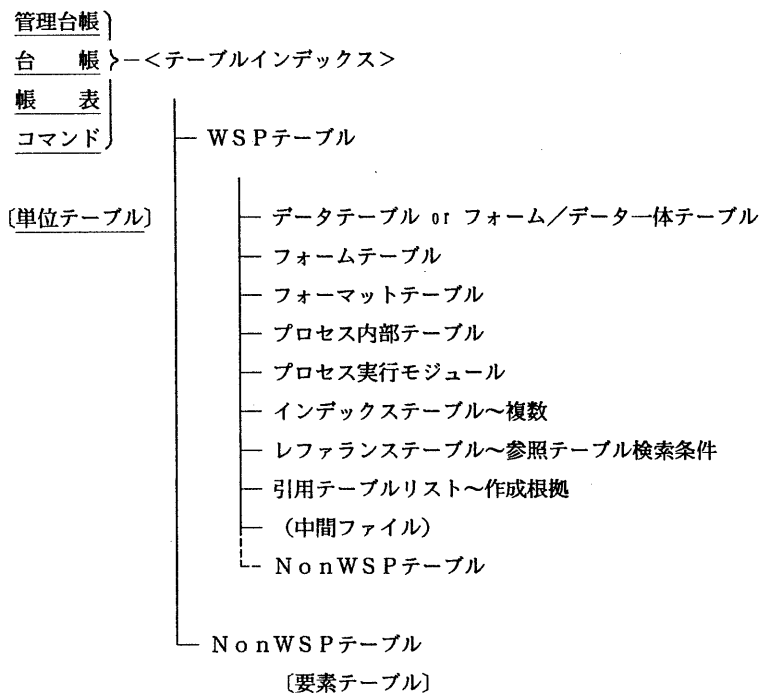
ンド）は、レコード側のレコードおよびテーブルインデックス、タイムスタンプ、リリース、ステータス、適用条件に対してユーザ検索条件の時点指定、変更指定、リリース指定、ステータス指定およびその他の指定にもとづいて検索を行うことで、はじめて、内部領域上の、リンクで関連づけられた複合実体として実現される。

台帳・管理台帳のレコードも単位テーブルと同様な検索を行なうことで、はじめて、内部領域上の実体となることが出来る。

## 5 オブジェクト指向データベースとしてのワークシートプログラム

ここで、ワークシートプログラムを、いわゆるオブジェクト指向の視点から、見直してみる。

### (1) 主オブジェクトと上下方向および論理リンク



第3図 テーブル/表の構造

ワークシートプログラムは、第4図に示すように、管理台帳を基本ノードとして相互関係を展開している。このことから、管理台帳を主オブジェクトと呼ぶことにする。

ここで、図の破線で示す、自己の管理下に実体を持たない参照の形が不可欠であることに注意されたい。

ファイルレジスタから要素テーブルに至る下方向リンクは順次ファイルを辿ることで実現される。

要素テーブルからファイルレジスタに至る上方向リンクは、内部領域に展開する時点で、上位管理台帳へのリンクを行う形で実現される。

さらに、主オブジェクトである管理台帳の中の各レコードは部品表アドレスにより、主オ

ブジェクト内の階層上の位置を示す形で、論理的にリンクされている。

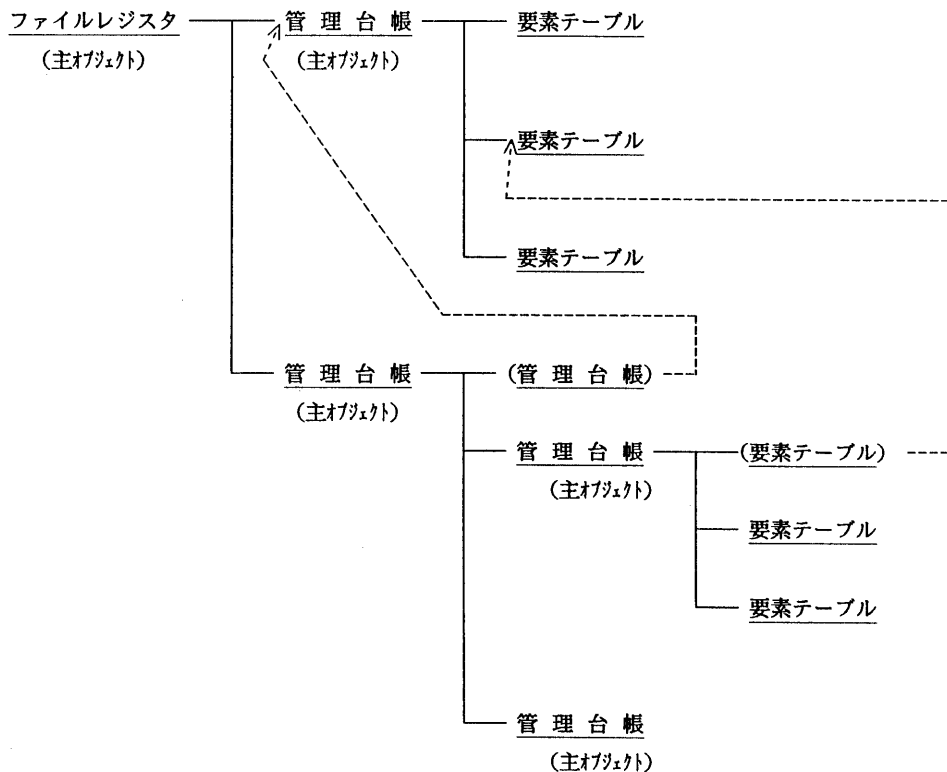
(2) クラスとインスタンス

第5図に示すように、フォームテーブル系の要素テーブルからなる単位テーブルとデータテーブル系の要素テーブルからなる単位テーブルとの関係はクラスとインスタンスの関係を構成する。ここで、単位テーブルを単位オブジェクトと呼ぶことにする。

なお、クラス相当の単位テーブルは、インスタンス相当の単位テーブルと同一の管理台帳内にあってもよい。

(3) 物理ブラウザと論理ブラウザ

第6図に示すように、管理台帳はそれ自体が



第4図 主オブジェクトと上下方向リンク

ブラウザである。管理台帳をそのままブラウザとしたものを物理ブラウザと呼ぶことにする。

一方、同一管理台帳内で、部品表アドレスおよびレコードインデックスで検索したレコード群で構成される画面の系列でブラウザを作ることが出来る。これを論理ブラウザと呼ぶこととする。

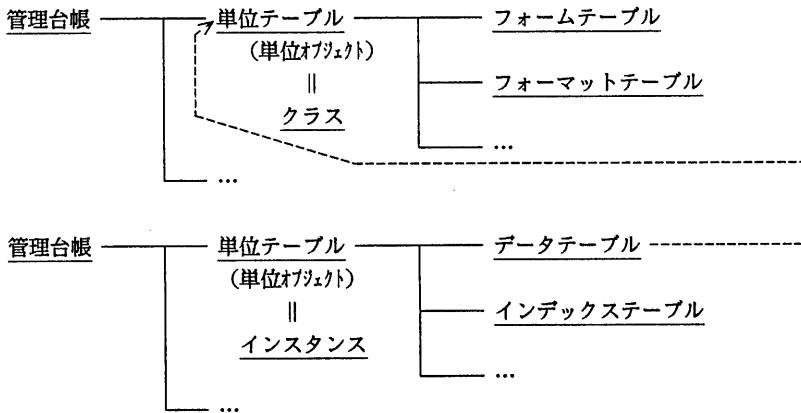
## 6 データの経路

3項および4項で述べた複雑な構造をもつデータをファイルから実体化するための手順として、画面表示の系列とコマンド実行の系列との2系統のデータの経路を用意する。

### (1) 画面表示

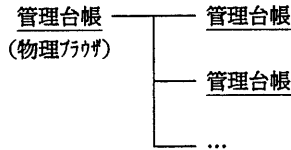
画面上に表示される個々の画面（以後、単画面と呼ぶ）は、第7図に示すように、画面番号レジスタの支配下におく。

1. 単画面はそれぞれ画面レジスタ、表示テーブルほかの固有のテーブルをもつ。

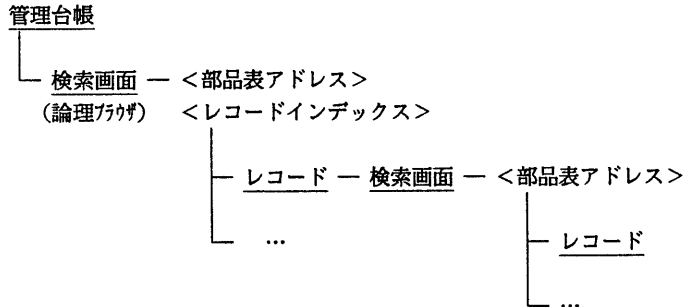


第5図 クラスとインスタンス

### 物理ブラウザ



### 論理ブラウザ



第6図 物理ブラウザと論理ブラウザ

2. 画面レジスタには表示されるテーブル（以後、主テーブルと呼ぶ）のレコードをもつ。
3. 主テーブルのレコードから自己の属する管理台帳（以後、親管理台帳と呼ぶ）のレコードおよび表示あるいは処理のため必要となる要素テーブル（以後、サブテーブルと呼ぶ）のレコードへのリンクを張る。さらに、あるテーブルに属するものとして指定された場合は、自己の属する（以後、親テーブルと呼ぶ）のレコードへのリンクを張る。  
ここで、各要素テーブルは独立の存在であり、リンクを張ることによって、はじめて、その役割が定まることに注意して欲しい。
4. 画面レジスタの各レコードは実体を持たず、内部テーブルレジスタの対応レコードにリンクを張ることによって、これに代えている。
5. 内部テーブルレジスタの各レコード間の関係は画面レジスタのそれらと同一である。ただし、内部テーブルレジスタの各レコードは、内部領域上に要素テーブルの実体をもつことが可能である点で、画面レジスタ

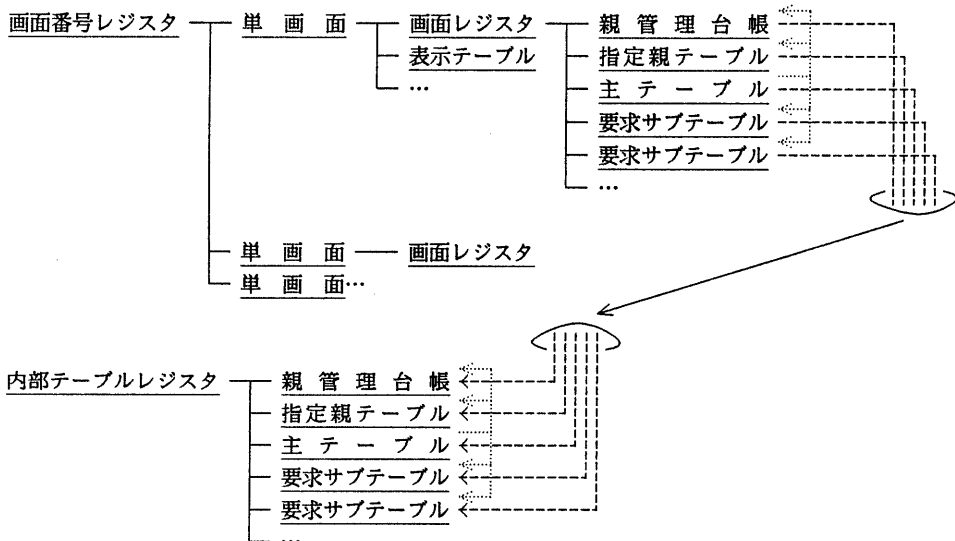
と異なる。

以上の形をとることで、複数の異なる様式の画面間で同一の内部領域上の要素テーブルを共有することが可能となる。

(2) コマンド実行

コマンドおよび独立プロセスは、第8図に示すように、それぞれメッセージレジスタ、プロセスレジスタの支配下にある。

1. コマンド実行要求は、メッセージの形（以後、一般メッセージと呼ぶ）でメッセージレジスタに登録し、画面レジスタと同様に、主テーブル、親管理台帳、要求サブテーブル、指定親テーブルのレコードをもつ形にする。  
画面表示の内部テーブルレジスタに対応するものとして、コマンド内部テーブルレジスタをもつ。これと一般メッセージとの関係は画面表示の内部テーブルレジスタと画面レジスタとのものとまったく同一である。
2. 独立プロセス実行要求は、メッセージの形（以後、ダイレクトメッセージと呼ぶ）でプロセスレジスタへプロセス毎に登録する



第7図 画面レジスタと内部テーブルレジスタ



が、画面レジスタ、一般メッセージと異なり、複数のレコードを持つ形はとらない。画面表示の内部テーブルレジスタに相当するものとして、プロセス内部テーブルレジスタがあるが、これとダイレクトメッセージとの関係は、画面レジスタ、一般メッセージとは異なり、ダイレクトメッセージから、直接、プロセス内部テーブルレジスタへリンクを張る形をとる。

## 7 プロセス実行の流れ

ワークシートプログラムのプロセス実行は、第9図に示すように、メッセージレジスタおよびプロセスレジスタで制御する形とする。

### (1) メッセージレジスタ

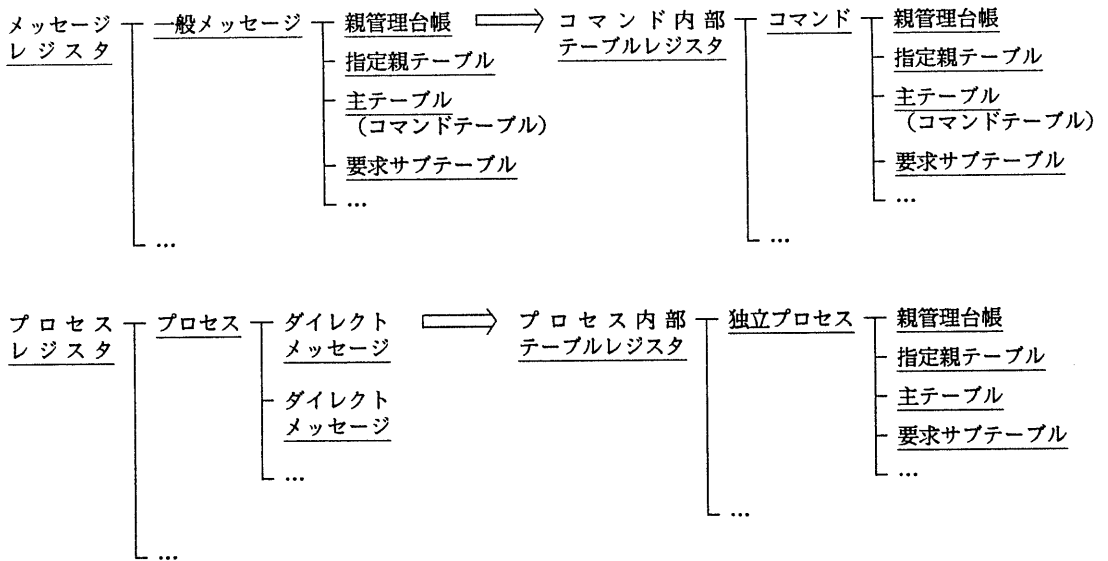
メッセージレジスタはコマンドおよびワークシート化された帳票・台帳・管理台帳の実行指示をプロセスレジスタで実行可能なものに編集する。

そして、実行モジュール名を付けて、ダイレクトメッセージをプロセスレジスタに送附する。

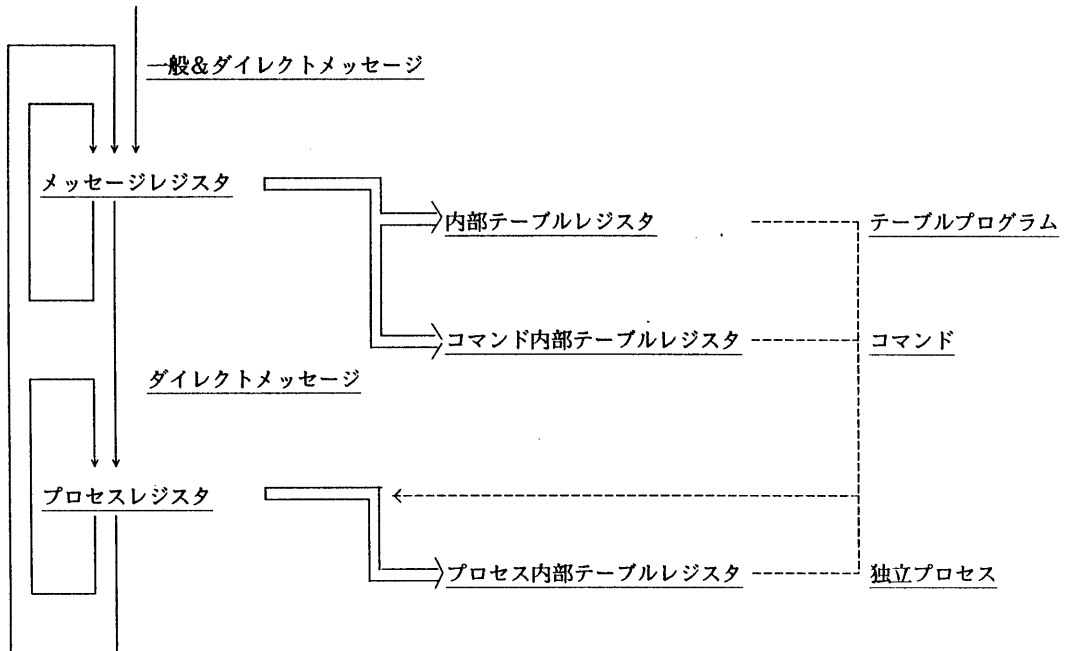
### (2) プロセスレジスタ

プロセスレジスタは受取ったダイレクトメッセージをプロセス毎に受け、優先順位にしたがって、プロセスの実行を指示する。

何らかの結果を得たとき、実行要求元へダイレクトメッセージを返すことが出来る。



第8図 メッセージレジスタとプロセスレジスタ



第9図 プロセス実行の流れ

## 8 おわりに

手作業をそのまま画面上の作業にするツール、ワークシートプログラムの実現を目指して、現実世界のモデルとしての、手によるオフィスワークの分析をさらに進めた。

この結果にもとづいて、ワークシートプログラムのデータの構造およびそのためのシステムの構造を提案した。この後、ワークシートプログラムの構成さらには論理を明かにして、コンピュータを知らない人でもシステム化出来る、あるがままをそのままシステム化出来る、喜んで使ってくれるシステム化出来る、使えるものは何でも使う、必要なところからシステム化出来る、そして気が

ついたら統合システムになっているという夢のツールの実現を目指したい。

本研究を行なうにあたって、多大な援助を下された日本自動車工学研究所長 故石原智男先生ならびに富士通SE課長 故佐田弓海彦氏に深く感謝を捧げる。

## 参考文献

- 〔1〕渥美: オブジェクト指向データベース  
ワークシートプログラムの概念, 情処  
データベース・システム研究会研報,  
84-2, 1991. 7. 17