構造化マトリックスによる給与計算システム

舟木康雄．奥津繁治．阿比留幸仁．増沢貴一
（富士写真フイルム㈱ システム管理部）

1.開発の背景

EDPシステムが次々に開発されるに従い、システムのレベルUPや制度変更に伴うプログラムのメンテナンス工数の増大は、EDP部門の運営上、大きな負担となっている。さらにメンテナンスに対する生産性の低下は、システムの信頼性をも低下させる結果となり、その影響が単にEDP部門のみでなく、利用部門も含めた全社的広がりをもつ問題となってきている。

当社の給与計算システムは、開発以来10年を経、度重なる制度変更への対応で複雑化、巨大化し、しかも給与制度を扱う労政部門からは、システムの複雑さが全くわからない状況になっていた。その結果、EDP部門では、常時3名の部員を給与計算システム維持に当てていたにもかかわらず、ちょっとしたメンテナンスでも致命的なトラブルが発生し、苦情があいつぐ一方で、給与制度そのものが益々、細分化、複雑化する状況に追い込まれていた。

そこで、システムのメンテナビリティーの向上と信頼性の確保に向けて、「誰でもが（給与計算を扱う労政部門担当者が）、漏れなく、正確にメンテナンスできる」給与計算システムの開発に取り組み、構造化マトリックスの概念を活用して、この解決を試みた。

このシステムは、昭和57年2月に完成し、所期の目的を達成して、本稼働に入っている。

2.給与システム運用上の問題

システム運用を担当する現場では、次の様な問題を抱えていた。

(1)メンテナンス工数の増大
労政部門から発行される変更依頼をプログラムメンテナンスに結びつける際、大きな時間を要していた。

(2)プログラムの複雑化
メンテナンスに際して、期間との関係でその都度、最も安全な方法を選ぶことにより、プログラムの構造が複雑化し、結果的に担当者の習熟に時間がかかっていた。

(3)給与制度面の複雑さの潜在化
給与制度をあずかる労政部門からは、EDPシステムの状況がわからず、データ処理を含めた給与制度全体の複雑さが、潜在化していた。

(4)信頼性の低下
プログラムが複雑化したため、すべてのPASSを短期間に把握しきれず、ダブルチェックに頼らざるを得なかった。結果として特殊ケース等のバグが潜在化し、後日トラブルとなるケースが発生していた。

そして、これら日常かかえる問題は、給与計算システムの変更が、プログラムの変更という形で行なわれるという問題に根ざしていた。

3.新システム開発のねらいと課題

新システム開発にあたり、目標としたのはシステムメンテナビリティーの向上である。そして

(1)業務のエキスパートであるユーザー部門の担当者が、
(2)習得時間も短かく、特別なスキルも必要とせずに、
(3)変更したい部分のみを確実に、メンテナンスできる。

という仕組みを、本システムの要件とした。この要件を満たすためには、プログラムの知識を持たない人でも、システムメンテナンスを行なえる環境を整備する必要があり、

(1)システム変更要因を集約し
(2)プログラムから分離する

ことがシステム設計上の課題であった。

4.開発への取り組み

4－1. プログラムメンテナンスの限界

システム変更をプログラムメンテナンスで対応している現状を考察して、プログラムメンテナンスの限界を整理すると、以下の項目が挙げられる。

(1)コボル等によるプログラミングは、言語自体の教育のみならず、構造化設計手法の教育を要するが、使用機会の少ないユーザー部門ではその教育の効率は低い。
(2)プログラムが構造化されていても、給与制度の変更の様にその構造的変更が加わるものに対し、常に合理的な論理構造を維持して行くことは、担当者の資質の差などを考慮すると、非常にむずかしい。
(3)ブラン千条件が増えれば、全てのＰＡＳＳを検証する為のテストデータは、幾何級数的に、増加する。この為メンテナンス時に全てのケースをテストするには、工数がかかりすぎる。

従って、いかに構造化されたプログラムであろうと、設定したシステム要件を満たせず、一切の外部仕様をプログラムに持ちこまない非プログラム化こそ、この要件を満たす方法であるとの結論に至った。

4－2. 非プログラム化の展開

4-2-1. プログラムの機能

コボルのプログラミングを例にとってみると

IF～ COMPUTE ～ GO TO ～
⋮

この様に、IF～に表わされる様な判断部分、COMPUTE～に表わされる計算部分、そしてこれらの文がPREFOMやGO TOにより幾層にも構造化され、順序づけられている。

ところで、データの性格についてその分類をすると、次ページのようになる。

ⓐ数値データ ―― 金額等の数量データ　(Quantitative Data.)
ⓑ非数値データ ― コード等の指示データ　(Indicative Data.)
　　└ 略称等の表示データ　(Descriptive Data.)

図1　データの性格による分類

そこで判断部分について見ると、コード等の非数値データを判断条件として、倍数等の数値データを設定したり、コードマッチングにより複数の数値データを結びつけ、その後の計算に依する働きと考えることができる。つまり、IF～等の判断部分は、数値演算のために非数値データを数値データに変換する機能として、とらえられる。

以上のことから、プログラムの機能を以下のように分離する事で、判断、計算とその順序づけの複合によるプログラムの複雑さを除去し得ると考えた。

給与支給額 ― 早残手当の計算 ⟷ 早残手当支給基準
　　家族手当の計算 ⟷ 家族手当支給基準
　　住宅手当の計算 ⟷ 住宅手当支給基準

↓ 演算および順序づけの機能
COMPUTE ～ GO TO ～
(Quantitative の世界)

↓ 条件による判断、数値化の機能
IF ～　変換
(Indicative → Quantitative)

図2. プログラム機能の分割

本システムでは、この演算および順序づけの機能に対し、構造化マトリックスの手法を適用し、それに対応した判断、数値化の機能に条件テーブルを活用して、システムメンテナンスの非プログラム化を、図った。

4-2-2. 構造化マトリックスとその特徴的な適用分野

構造化マトリックスの特徴的な適用分野について、参考文献(1)で述べられているもののうち、本システムへの適用の動機となった特徴は、次のものである。

(1)定型データを取扱う分野
(2)単純なロジックの複雑な積み上げで構成される分野
(3)全体を鳥瞰し、部分に入っていくことを要求される分野
　　(コンピュータで何を行っているか知らせる必要がある。)
(4)エンド・ユーザにシステムメンテナンスをさせたい分野

4－3. 構造化マトリックスの適用

4-3-1. マトリックス演算と条件テーブル

マトリックス演算を給与計算に適用すると、次の手順で計算が行なわれていく。まず左辺と上辺とに同一配列で位置する計算項目がある。基本給、勤務給(手当)などである。上辺は計算への入力に使われ、結果の出力は左辺へ入る。上辺と左辺とから作られるマトリックスには係数が持ち込まれ、上辺の各項目の値とそれに該当する係数の掛け算と、その結果の横軸での合計が左辺に出力される。(縦積横和計算)このマトリックスに持ち込まれる係数は、計算対象となる人の属性に従って条件テーブルより持ち込まれる。

計算結果は左辺より上辺に移され、順にすべての計算項目が処理されていく。

(例)

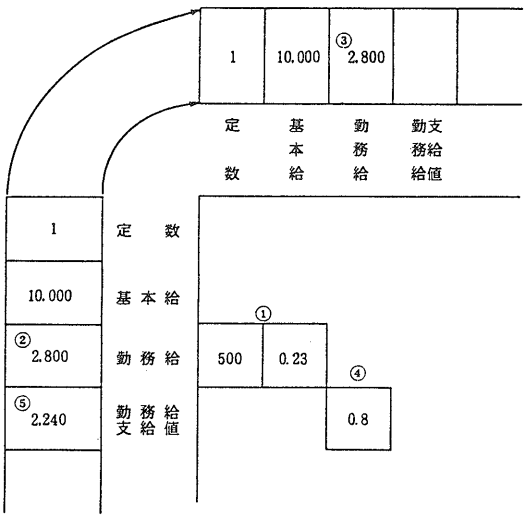


図3. マトリックス演算の例

計算式：勤務給＝(基本給×系数＋定額)×支給率

A さんの系数値　　0.23　　500　　0.80

基本給1万円

勤務給の計算例として処理の流れを説明する。

(1)勤務給行の系数を条件テーブルより持ち込む。

(2)勤務給のマトリックス演算

$1 \times 500 + 10,000 \times 0.23 = 2,800$

(3)上辺への移送

(4)支給勤務給行の系数の持ち込み

(5)支給勤務給のマトリックス演算

$2,800 \times 0.80 = 2,240$

となる。ある条件の人に支給しない場合、条件テーブルの値は0となる。

4-3-2. マトリックスの構造化

係数を持ち込むマトリックス部分、即ち係数マトリックスは、N個の計算項目に対してN×Nの係数マトリックスを準備すれば良い。しかし、係数マトリックスを部分マトリックスにより分割、構造化することで、次のメリットを出す工夫を行った。

(1)係数がゼロの要素が激減し、また同一の部分マトリックスを兼用して処理の効率化を図った。

(2)もともと、各計算項目は、何段階もの計算の論理構造をもっている。そこでこの構造を反映した、部分マトリックスを作ることにより、係数マトリックスをわかり易くした。

(3)メンテナンスで関連する項目の範囲が限定され、変更確認を容易とした。

図4に構造化の一例を示す。本システムでは640000(800×800)のマトリックスを約130の部分マトリックスに集約し、構造化した。

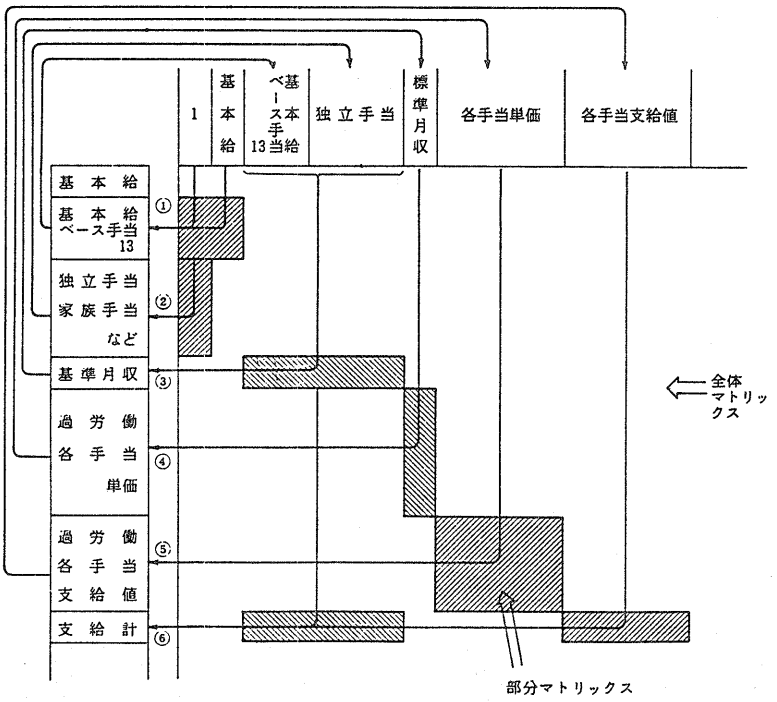


図4. 構造化マトリックス

4-4. EDPシステムへの展開

4-4-1. 処理エリアの設定

本システムでは、処理エリアの機能とその相互関連を図5に示すように設定した。

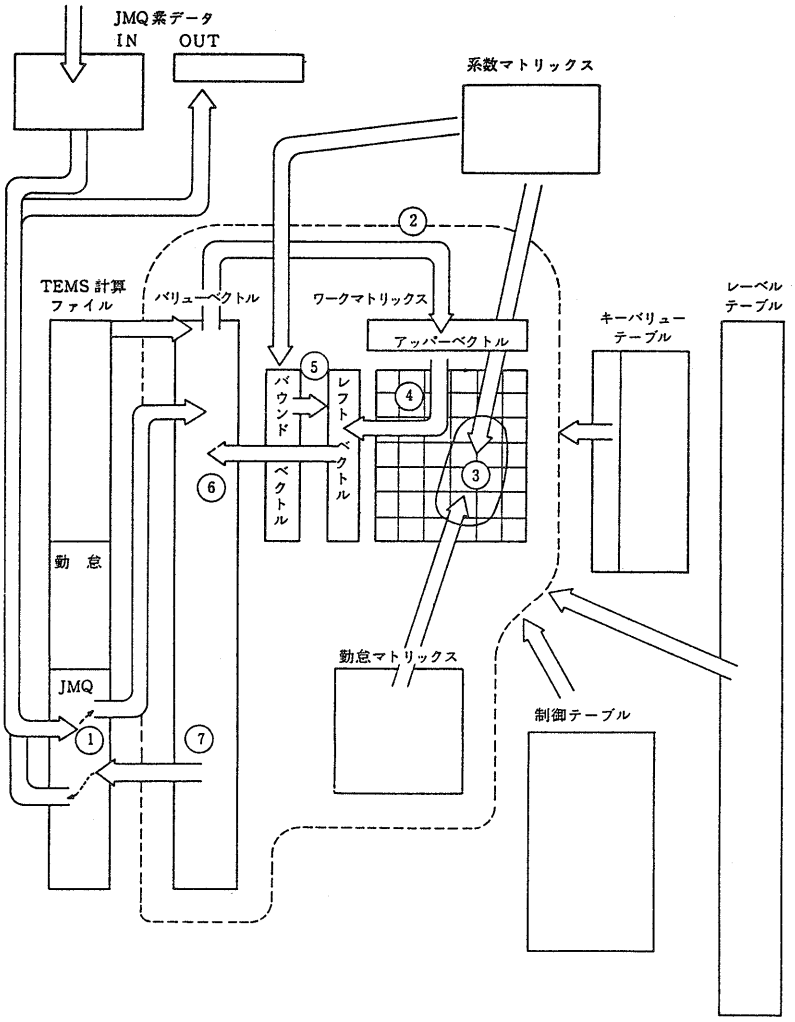


図5. 処理エリアと相互関連

表1. 処理エリアの内容説明

処理エリアの名称	内　　容
JMQ素データ	賃金計算の各項目別に入力した支給あるいは控除する金額データ
TEMS計算ファイル	TEMS マスター（人事マスター）から作られたファイルで メイン処理を行う前に、各人毎に基本属性、勤怠データ等 計算に必要な項目を設定したファイルである。
系数マトリックス	マトリックス演算を行う為の系数マトリックスファイルであり、給与計算の各項目の計算式を表わしたものである。このマトリックスは キーのタイプごと値ごとに複数 用意されている。
バリューベクトル	給与マトリックス 演算を行う為の各項目の値をストアしておくワークエリアである。各項目の数は、レーベルテーブルと対応している。

キーバリューテーブル	TEMS計算ファイルを読む度に、性別などのキーのタイプとその値を格納する。
ワークマトリックス	マトリックス演算を行う為のワークマトリックスエリアである。係数マトリックス(20×20)に上辺、左辺のベクトルおよびバウンドテーブルを追加したもの。ここで演算が行われる。
バウンドベクトル	マトリックス演算に使用する上限および下限の項目別の値を保有する。
レーベルテーブル	バリューベクトルの名称と位置。バリューベクトルにデータを格納する時の条件およびバリューベクトルとJMAエリアとの対応づけを行う。
勤怠マトリックス	TEMS計算ファイルから勤怠項目をセットしたマトリックスで、過勤働、カット額などを計算する際、使用する。
制御テーブル	部分マトリックスの大きさ、係数ファイルの使用条件、サブルーチンの利用の制御、および部分マトリックスの実行順序を制御する。

4-4-2. 条件テーブルから係数マトリックスへの変換

メンテナンス対象となる条件テーブルを、EDP処理に使用する係数マトリックスへ変換するプロセスを以下に示す。

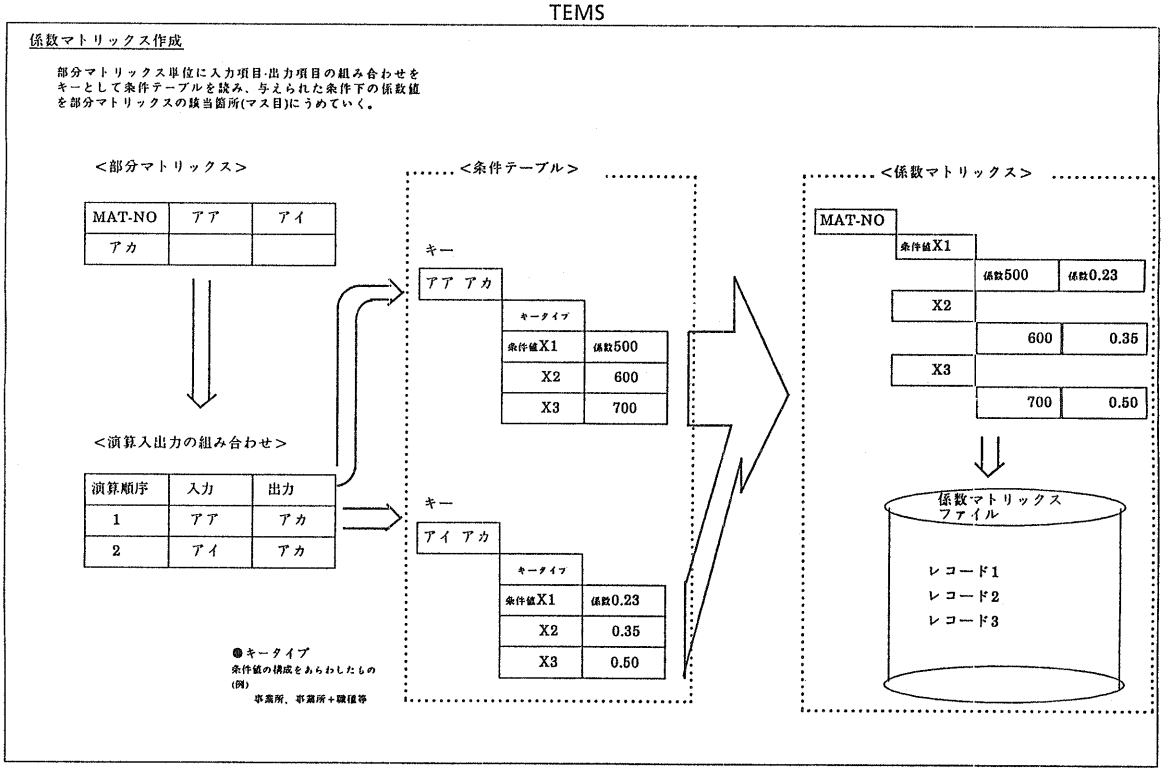


図6. 係数マトリックスの作成

4-4-3. 各テーブルの構成

本システムで使用する、制御テーブル、レーベルテーブル、条件テーブルについて、その構成を以下に示す。

(1) 制御テーブルの構成

ARG

コード MATRIX No. (8)

RES

POSI		SIZE		SAME MAT No	KEY TYPE	MAT TYPE	事セットタイプ	う回マーク		V. V LABEL
R (3)	C (3)	R (3)	C (3)	(8)	(2)	(2)	(1)	(1)	(2)	(4)

- POSI R (3): バリューベクトルのアウトプット開始位置
- POSI C (3): バリューベクトルのインプット開始位置
- SIZE R (3): アウトプット項目数
- SIZE C (3): インプット項目数
- SAME MAT No (8): 他のMAT No.又はサブルーチン名を指定して実行できる
- KEY TYPE (2): 条件値のタイプ
- MAT TYPE (2): 部分マトリクスの処理タイプを決定する
- 事セットタイプ (1): 条件値の事業場タイプを決定する
- う回マーク (1): このマトリクスを実行するか否かをコントロール

図7. 制御テーブルの構成とサンプル

```
==================================================
ｺｰﾄﾞ          ﾅｲﾖｳ
==================================================

L1402140  586475  1 11L140203008F  8
L1402150  587486  1 11L140210008F  8
L1402160  588497  1 11          08F A8
L1402170  589508  1 11          08F A8
L1402180  590519  1 11          08F A8
L1402190  591530  1 11          08F A8
L1402200  592541  1 11          08F A8
L1403000  575552  6 18          08F A1
L1403010  586570  1  2          08F A1
L1404000  593288  1  1          06F A1  ﾎｲﾀ2
L1404010  594000001000JXB12T5508UB 1
L1405000  594  0  7  0            MQ 1  ﾎJ
```

(2) レーベルテーブルの構成

ARG

コード (2)

RES ---→ バリューベクトルのレーベルを表わす

名称	JMQ レーベル		MOVE	LIM	BOUND		V. V SEQ	JMQ SEQ	
(15)	(2)	(4)	(1)	(1)	(1)	(1)	(3)	(3)	(4)

- 名称 (15): バリューベクトルの略称を指定する
- JMQ レーベル (2): TEMSファイルのJMQレーベル名
- MOVE (1): バリューベクトルからTEMSファイルへの戻し方 (MOVE / ADD)
- LIM (1): 限度チェックの有無
- BOUND (1): 結果をバリューベクトルに戻す際のラウンドの仕方
- V. V SEQ (3): コードがバリューベクトル上の何番目にあるかを指定
- JMQ SEQ (3): コードがTEMS JMQエリアの何番目に対応するかを指定

図8. レーベルテーブルの構成とサンプル

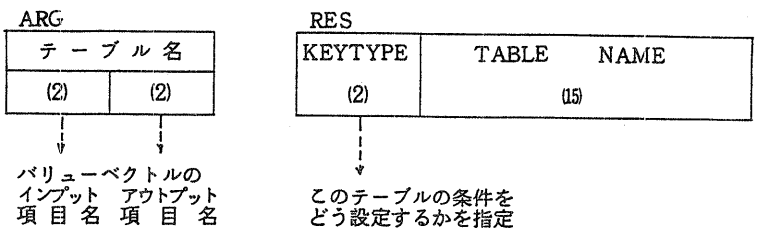
```
==================================================
ｺｰﾄﾞ          ﾅｲﾖｳ
==================================================

ｱｱ          ﾃｲｶﾞｸ ｺﾝｽﾀﾝﾄ                    1
ｱｲ          ｷﾎﾝｷｭｳ ﾘﾛﾝﾁ        MU      A   2 21
ｱｶ          ｷﾝﾑｷｭｳ ﾘﾛﾝﾁ        MX      A   3 24
ｱｷ          ｶｷｭｰ ﾘﾛﾝﾁ          MX    A A   4 24
ｱｹ          3ｰ3ｺｳﾀｲ ﾘﾛﾝﾁ       NG      A   5 33
ｱｺ          4ｰ3ｺｳﾀｲ ﾘﾛﾝﾁ       NF      A   6 32
ｱｻ          2ｺｳﾀｲA ﾘﾛﾝﾁ        NI      A   7 35
```

(3) 条件テーブルの構成

条件テーブルは、マトリックスに応じた系数値の検索条件（キータイプ）と検索条件の内容に対応した係数値を持つ。

A）テーブル名テーブル



B）条件テーブル

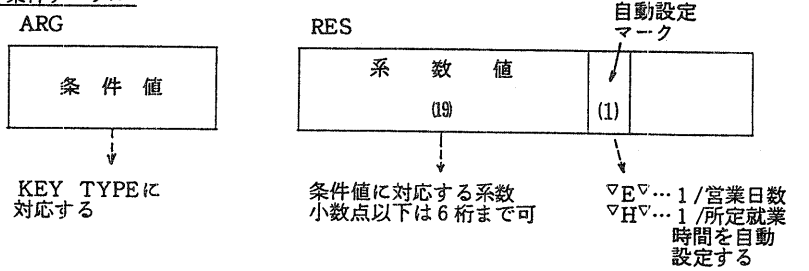


図9. 条件テーブルの構成とサンプル.

ARG	RES
86HD	.068323
86HE	.068323
86H3	.071429
86H4	.071429

5. 効果

本システムは、昭和57年2月より本稼働に入っている。構造化マトリックスを活用することにより、メンテナンスの対象が、条件テーブル、制御テーブル、キーバリューテーブル、レーベルテーブル、ファイルレイアウトに限定され、プログラムメンテナンスから解放された。

これにより以下の効果が出ている

(1) 条件テーブルのメンテナンスが労政部門に移管でき、EDP部門は、給与システムのメンテナンスから解放された。
(2) EDP部門での給与関連業務の維持管理工数が1/3に減少した。
（担当は3名より1名に減）
(3) クレーム件数は皆無になり、EDP部門への問い合わせが1/5に減少した。
（50件/月 から 10件/月に減）
(4) EDP部門の担当者の引き継ぎに要する期間が短縮した。
（従来1年かけていたものが、3ヵ月で可能となった。）

(5)労政部門では、EDP部門との調整なしで、タイムリーな制度変更を行う事が可能になった。
(6)制御テーブルの変更だけで、賞与計算、差額計算への適用が容易に行えた。

6.今後の課題

6－1. 汎用化への取り組み

今回、構造化マトリックスを活用した給与計算システムが実現できたが、この計算部分のインターフェースを明確にして、計算用部品システムとして汎用化を試みたい。

6－2. 構造化プロセスの明確化

部分マトリックスの構成にあたっては、EDP部門の担当者の知見による試行錯誤で行なわれ、その視点としては
(1)条件テーブルの検索キーが同一であるもの
(2)インプット項目が支給値、単価、手当などグルーピングとしての意味をもつもの
(3)全体として、部分マトリックスの意味づけが可能なもの

として行なった。
プロセスを明らかにして、ある程度機械的に行なえるようにする必要がある。

参考文献 (1)日本IBM資料 AM0H-010
「構造化マトリックスによる経営管理システムへのアプローチ」
日本IBM
(2)阿比留、粟津、増沢
「構造化マトリックスによる新給与計算システム」
第2回IBMユーザシンポジウム論文集
昭和59年