

# 技 報

UNIVAC TECHNOLOGY REVIEW

1983年2月 第4号

---

## 論 説

- オペレーティング・システムにおけるセキュリティ  
——権利の役割分離……………E.J. Weyuker 1

## 論 文

- 任意節点配置差分法による連続体解析……………藤野 勉 69  
精密ロック……………J.R. Jordan, J. Banerjee, R.B. Batman 84

## 報 告

- UNIVAC 1100/90シリーズの新アーキテクチャ  
……………古沢純一, 城川孝二, 田中厚司 14  
MAPPERを利用した製造管理システム“TACTICS”……………L.F. Rogney 24  
ASHLARにおける自動レイアウト——VLSIの“ジェネラル・セル”  
レイアウトの諸問題へのアプローチ……………J.E. Hassett 38  
DSS概念の実現……………小坂 武 52

---

## TECHNOLOGY TREND

- ソフトウェアの品質管理と品質コスト……………西島政信 94  
データ通信回線の第2次開放……………守田洋一 98

BOOKS……………101

CALENDAR……………104

MEMORANDUM……………105

EDITORS' NOTE……………表2

---

今日、オペレーティング・システムのセキュリティと対象へのアクセスを制御するプロテクション機構に多くの関心が寄せられている。E. J. Weyuker のオペレーティング・システムにおけるセキュリティ——権利の役割分離は Harrison, Ruzzo, Ullman が導入したプロテクションの一般モデルと Jones, Lipton, Snyder が定義したプロテクション・システムを考察する。それぞれのモデルの非現実的な機能を指摘して、どのように修正すれば、モデルをより直観的に合理化できるかを論じている。また、これらの修正がプロテクション・システムについての重要な問題の決定可能性と、計算量に与える影響を検討している。

急激に増大する情報量、加速度的に拡大していく利用層、多様化し高度化する情報処理形態のもとでコンピュータに望まれることは、高い処理能力と同時に使いやすさである。古沢らの UNIVAC 1100/90 シリーズの新アーキテクチャは、こうした時代の要請に応えて開発された UNIVAC 1100/90 シリーズの新しいアーキテクチャのうち、その柱であるアドレス空間の拡張とリング/ドメイン概念によるメモリ保護を中心に解説している。同シリーズは第4世代機とも呼べるもので将来への方向づけを行ったもので、昨年全世界に発表された。

各段階における製造と検査の自動化を目的としたシステムは、今や製造部門での日々のオペレーションに不可欠なものとなっている。L. F. Rogney の MAPPER を利用した製造管理システム“TACTICS”は、MAPPER を使って構築されたデータ収集および製造工程制御用のリアルタイム・システム TACTICS について報告している。TACTICS は中央集中型の分散システムであり、その実現には MAPPER が必要不可欠であった。MAPPER はアプリケーション・ジェネレータであり、多種多様な報告書的设计・作成・修正をエンド・ユーザ自らの手で作成できるプログラムレス・ツールとして評判が高い。

J. E. Hassett は、ASHLAR における自動レイアウト——VLSI “ジェネラル・セル”レイアウトの諸問題へのアプローチで、任意寸法セルをもつ階層的 VLSI 設計のための会話型レイアウト・システムを解説している。とくに、配置、電源バス配線、多結晶シリコン内部結線などの諸問題への対応に注意

を向け、プロトタイプの実行結果も言及している。

ビジネス分野に対するコンピュータの適用として、OA とともに、近年 DSS が大きな関心を呼んでいる。小坂の DSS 概念の実現では、筆者らが IFIP の「Information & Management」誌に発表した論文と対をなすもので、DSS の基本形態を主に現実の場におけるマネジメントの問題解決の面から論じている。すなわち、汎用的な DSS の必要性を述べ、その必要を満たすには DSS パッケージが有効であることを示し、さらに具備すべき基本形態を考察している。また、具体的な DSS パッケージとして、日本ユニバック(株)の DSS 1100-J を取り上げ、その基本形態を実際に紹介し、いかに多様な DSS の構築に有効かを論じている。

複雑な形状をもつ連続体の解析には有限要素法が効果的に利用されているが、若干の欠点をもっている。藤野の任意節点配置差分法による連続体解析では、この点を改善するために任意節点配置による差分法を提案している。

差分法としては、Taylor 展開法、補間関数法、アイソパラメタ法の三つの方法に対し、主として2次元差分法について述べている。また、これらの方法の特徴を比較検討している。

J. R. Jordan らの精密ロックでは、データベースの並行処理を制御する方法を論じている。この方法、すなわち精密ロックは、データベースの整合性を保証するのに必要なレコードの集合のみを精密にロックするということから名づけられたもので、レコード・レベルのすべてのロックの中で並行処理が最大となる。本稿では、ロック・システムが犯す誤りを表示する枠組を導入し、既存のロック方法を論じたのち、精密ロックが良型かつ2段階の任意の述語を受け入れることや、物理的レコード・レベルのロック・システムとの間の単純な実行コストの比較などを論じている。

☆

## 論説

オペレーティング・システムにおけるセキュリティ  
——権利の役割分離

## Security in Operating System: Separating the Roles of Rights

E. J. Weyuker

**要約** オペレーティング・システムのプロテクション機構の二、三のモデルを論じる。Harrison-Ruzzo-Ullman のモデルと Jones-Lipton-Snyder モデルを手直して、原モデルがもつ非現実的な機能を修正する。

これらの修正が重要な問題の決定可能性、および計算量の問題に与える影響を検討する。それぞれの修正を加えた場合、決定可能性と計算量についての結果は原モデルの場合に比べて決して悪化しないことを証明する。

**Abstract** Several possible models of protection mechanisms for operating systems are discussed. These models represent modifications of models introduced by Harrison, Ruzzo, and Ullman, and Jones, Lipton and Snyder. The modifications represent attempts to rectify certain unrealistic features.

The effects of the modifications on the decidability of key questions is investigated, as well as complexity questions when appropriate. It is demonstrated that for each modification considered, decidability and complexity results are never worse than the less reasonable, original models.

## 1. はじめに

今日、オペレーティング・システムのセキュリティと対象へのアクセスを制御するプロテクション機構に多くの関心が寄せられている。そこで、Harrison, Ruzzo, Ullman<sup>[4]</sup> が導入したプロテクションの一般モデルと Jones, Lipton, Snyder<sup>[5], [6]</sup> が定義したプロテクション・システムを考察する。それぞれのモデルの非現実的な機能を指摘して、どのように修正すれば、モデルをより直観的に合理化できるかを論じる。また、これらの修正がプロテクション・システムについての重要な問題の決定可能性と、計算量に与える影響を検討する。

Minsky<sup>[7]</sup> の指摘によると、対象に対する操作を許可する権利は命令の適用を制御する権利でもある。これらの命令は順に(主体が所与の対象に対してもつ権利を記述している)アクセス(あるいはプロテクション)行列の修正を制御する。この機能は、Harrison, Ruzzo, Ullman の一般モデルにも、Jones, Lipton, Snyder の特定システムにも共通しているという事実を考えると、権利のこの二重の役割は妥当であるかという疑問が生じる。すなわち、同じ権利の集合が対象に許可されている操作を制御し、同時にアクセス行列の修正を制御できるかという疑問である。

これらのモデルには、いろいろなタイプの制限を設けることができる。ここでは上の疑問に答えて修正を加える。われわれが課する最も厳しい要請は、命令の実行の結果アクセス行列に記入、あるいは削除できるような権利の集合からコントロール権(命令を実行可能にする権利の集合)を分離するということである。この前者のタイプの権利を“影響を受ける(affected)権利”と呼ぶことにする。

さらに、分離の制約の緩和についても考えて、それが決定問題や計算量の問題や権利取得の難易度に与える影響を調べる。

最後に、権利が示す異なる役割を分離し、主体が新しい権利を取得しにくくさせるために、Jones-Lipton-Snyder システムのある修正法を検討する。

## 2. 形式的モデル

本章では、Harrison, Ruzzo, Ullman のモデルと Jones, Lipton, Snyder のシステムを概説する。詳細については論文<sup>[3], [4], [5], [6], [8], [9]</sup>を参照されたい。

### 2.1 Harrison-Ruzzo-Ullman モデル

プロテクション・システムは、

- 1) 権利 (right) の有限集合
- 2) 命令 (command) の有限集合

からなる。各命令は次の形をしている：

```

command  $\beta(X_1, \dots, X_k)$ 
    if  $\alpha_1$  in  $(X_{s_1}, X_{o_1})$  and
        ...
         $\alpha_n$  in  $(X_{s_n}, X_{o_n})$ 
    then  $op_1$ 
        ...
         $op_m$ 
end
  
```

ここで、 $\beta$  は名前、 $X_1, \dots, X_k$  はパラメタ、 $\alpha, \alpha_1, \dots, \alpha_n$  は権利、 $s_1, \dots, s_n, o_1, \dots, o_n$  は 1 から  $k$  までの範囲の整数である。また各  $op_i$  は以下にあげる基本操作の内の一つである。

基本命令 (primitive operation)

```

enter  $\alpha$  into  $(X_s, X_o)$ 
delete  $\alpha$  from  $(X_s, X_o)$ 
create subject  $X_s$ 
create object  $X_o$ 
destroy subject  $X_s$ 
destroy object  $X_o$ 
  
```

権利  $\alpha$  が命令の **if** 部に現われるとき、 $\alpha$  を“コントロール権 (controlling right)”という。権利が命令の **then** 部にある操作の中に現われるとき、その権利を“影響を受ける権利 (affected right)”という。**if** 部を命令の“条件 (condition)”，**then** 部を“本体 (body)”ということもある。

命令にコントロール権がない場合もある。この場合、命令は次の形である：

```

command  $\beta(X_1, \dots, X_k)$ 
     $op_1$ 
    ...
     $op_m$ 
end
  
```

プロテクション・システムの“構成 (configuration)”とは三つ組  $(S, O, P)$  である。ここで、 $S$  は“現主体 (current subject)”の集合、 $O$  は“現対象 (current object)”の集合で

あり、 $P$ は“アクセス行列 (access matrix)”である。

$P$ は $S$ 内の各主体 $s$ に対して1列を、 $O$ 内の各対象 $o$ に対して1行をもつ。 $S \subseteq O$ であるから、各主体には1列および1行が対応することに注意する。

$R$ は権利の集合であり、 $P[s, o]$ は主体 $s$ が対象 $o$ に対してもつ権利を表わす $R$ の部分集合である。

上記の六つの基本操作のそれぞれは、アクセス行列の修正方法を表わしている。形式的には、それらは次のように定義される。ここで、 $(S, O, P)$ と $(S', O', P')$ とをプロテクション・システムの構成とし、 $op$ を基本操作とすると、操作 $op$ を実行した結果、 $(S, O, P)$ が $(S', O', P')$ となることを、

$$(S, O, P) \xrightarrow{op} (S', O', P')$$

と書くとする。このとき、この関係は次のいずれかのときに成り立つ。

1)  $op = \text{enter } \alpha \text{ into } (S, O)$  かつ  $S = S', O = O', s \in S, o \in O$  であり、かつ、

$$(s_1, o_1) \neq (s, o) \text{ かつ } P'[s, o] = P[s, o] \cup \{\alpha\} \text{ ならば,}$$

$$P'[s_1, o_1] = P[s_1, o_1].$$

2)  $op = \text{delete } \alpha \text{ from } (s, o)$  かつ  $S = S', O = O', s \in S, o \in O$  であり、かつ、

$$(s_1, o_1) \neq (s, o) \text{ かつ } P'[s, o] = P[s, o] - \{\alpha\} \text{ ならば,}$$

$$P'[s_1, o_1] = P[s_1, o_1].$$

3)  $op = \text{create subject } s'$  で、 $s'$ は $O$ にない新しい記号であり、 $S' = S \cup \{s'\}$ 、 $O' = O \cup \{s'\}$ で、

$$\text{すべての } (s, o) \in S \times O \text{ に対して, } P'[s, o] = P[s, o],$$

$$\text{すべての } o \in O' \text{ に対して, } P'[s', o] = \phi \text{ かつ,}$$

$$\text{すべての } s \in S' \text{ に対して, } P'[s, s'] = \phi.$$

4)  $op = \text{create object } o'$  で、 $o'$ は $O$ にない新しい記号であり、 $S' = S$ 、 $O' = O \cup \{o'\}$ で、

$$\text{すべての } (s, o) \in S \times O \text{ に対して, } P'[s, o] = P[s, o] \text{ かつ,}$$

$$\text{すべての } s \in S \text{ に対して, } P'[s, o'] = \phi.$$

5)  $op = \text{destroy subject } s'$  で、 $s' \in S$ 、 $S' = S - \{s'\}$ 、 $O' = O - \{s'\}$ であり、

$$\text{すべての } (s, o) \in S' \times O' \text{ に対して, } P'[s, o] = P[s, o].$$

6)  $op = \text{destroy object } o'$  で、 $o' \in O - S$ 、 $S' = S$ 、 $O' = O - \{o'\}$ であり、

$$\text{すべての } (s, o) \in S' \times O' \text{ に対して, } P'[s, o] = P[s, o].$$

## 2.2 Jones-Lipton-Snyder TAKE-GRANT システム

Lipton らの論文<sup>[6]</sup>で定義されている TAKE-GRANT システムは次の3種類の命令をもっている。ここで  $x, y, z$  は主体を表わし、 $\alpha \in \{r, w\}$  である。(  $r$  と  $w$  は “read” と “write” と考えてよい。彼らの論文<sup>[6]</sup>では、権利  $c$  と CALL, REMOVE操作があるが、Jones らの論文<sup>[5]</sup>で述べているように、あまり使われない。)

1) **command** TAKE ( $x, y, z$ )

**if**  $r$  in ( $x, y$ ) and

$\alpha$  in ( $y, z$ )

**then** enter  $\alpha$  into ( $x, z$ )

**end**

2) **command** GRANT ( $x, y, z$ )

**if**  $w$  in ( $x, z$ ) and

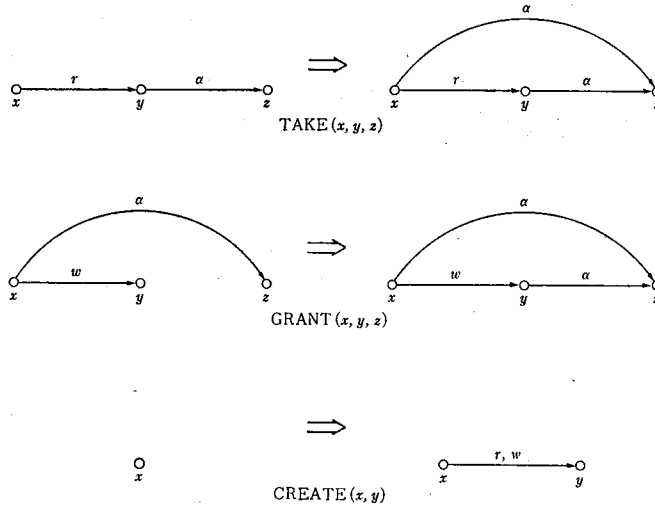


図 1 命令のグラフ表現

Fig. 1 Commands represented graphically

$\alpha$  in  $(x, z)$   
**then** enter  $\alpha$  into  $(y, z)$   
**end**  
 3) **command** CREATE  $(x, y)$   
     create subject  $y$   
     enter  $r$  into  $(x, y)$   
     enter  $w$  into  $(x, y)$   
**end**

これらの命令は図 1 に示すようにグラフで表現できる。

Jones, Lipton, Snyder に従って、ここでも命令のグラフ表現を使う。Jones の論文<sup>[5]</sup>では、主体と対象を区別できるようにシステムを修正した。このシステムでは上の各命令において  $x$  は主体でなければならず、 $y$  と  $z$  は対象でもよい。

$x\alpha y$  という記法で、「 $x$  は、 $y$  への権利  $\alpha$  をもつ」ことを表わす。これをグラフでは

$$x \overset{\alpha}{\longrightarrow} y$$

と表わす。さらに、 $x \text{ can } \alpha y$  で、「 $x$  は  $y$  への権利  $\alpha$  をもつか、あるいはある命令列を実行すれば  $x$  が  $y$  への権利  $\alpha$  を取得できる」ことを表わす。 $G$  をアクセス行列を表わすプロテクション・グラフとすると、命令の実行によってプロテクション・グラフの列

$$G = G_0, G_1, \dots, G_n$$

が得られよう。

### 3. 新しいモデルと結果

われわれの主要関心事の一つは、主体が対象に対する何らかの権利を取得できるか否かということである。

初期のアクセス行列が与えられたとき、ある命令列が存在して、命令列の実行前には ( $\alpha$  が) 存在しなかった行列要素に命令列の実行後に  $\alpha$  が入るならば、権利  $\alpha$  が“漏れた (leaked)”という。システムでどの権利も漏れないとき、システムは“安全 (safe)”で

ある。Harrison, Ruzzo, Ullman<sup>[4]</sup>はこの概念を長々と論じ、実践的な考察によっていろいろな代わりの定義を提案している。

### 3.1 Harrison-Ruzzo-Ullman モデルの修正

Harrison, Ruzzo, Ullman は、彼らのモデルについて、任意に与えられた権利に対して任意のプロテクション・システムが安全か否かを決定するアルゴリズムが存在しないことを証明した。われわれは、ここで Harrison-Ruzzo-Ullman モデルに対する四つの制限を提案し、この4種のプロテクション・システムに対する安全性の問題の(帰納的)可解性を調べる。制限  $k$  を満たすシステムを“タイプ  $k$  のプロテクション・システム”と呼ぶ。

命令  $C_i$  に対するコントロール権の集合を  $R_i$  とし、

$$R = \cup R_i$$

とする。命令  $C_i$  によって影響を受ける権利の集合を  $A_i$  とし、

$$A = \cup A_i$$

とする。

#### 制限 1

$$R \cap A = \phi.$$

#### 定理 1

任意のタイプ 1 のプロテクション・システムと、そのアクセス行列の任意の初期構成が与えられたとき、それが任意に与えられた権利  $\alpha$  に対して安全であるか否かを決定するアルゴリズムが存在する。

#### 〔証明〕

$R \cap A = \phi$  より、 $R$  の要素とアクセス行列に加えたり削除したりすることはできない。したがって、影響を受ける集合  $A$  のみを考えればよい。 $\alpha \in A$  が漏れるのは、任意の命令の実行前に対象のあるベクトルに対して“条件”が満たされるような命令  $C_i$  が存在し、 $C_i$  の一つの操作が enter  $\alpha$  into  $(X_s, X_o)$  で、かつ  $\alpha$  が  $(X_s, X_o)$  内にはないときであり、このときにかぎる。この状態は明らかに有限回の操作でテストできる。 ■

Harrison らの論文<sup>[4]</sup>に概説されているのと同様の  $k$ -クリーク問題からの直接の還元を使うと、制限したプロテクション・システムに対する決定問題は  $NP$ -完全であることが証明できる。詳細については Aho, Hopcroft, Ullman の論文<sup>[1]</sup>を参照されたい。したがって、次の定理を得る：

#### 定理 2

タイプ 1 のプロテクション・システムに対する安全性問題は  $NP$ -完全である。

タイプ 1 システムに対しては、安全性問題は決定可能であるが、 $NP$ -完全であることがわかったところで、そのようなシステムがどのように制限されているかを考えよう。この制限はコントロール権を追加も削除もできないことを述べている。とくに、この制限によれば新たに生成された主体も対象も、決してコントロール権を取得しえないことになる。このようなシステムは、きわめて静的である。そこで、安全性問題の決定可能性の限界を見出すために、また、とくに、この決定問題の決定可能性に対する十分条件としての安定性の役割を理解するために、この制限を緩和してみよう。(Jones-Lipton-Snyder システムに対するこの問題の決定可能性によって、あらかじめこれが必要条件ではないことがわかっている。)

#### 制限 2

$\alpha \in (R \cap A)$  ならば、 $\alpha$  に関する操作は delete 操作にかぎる。

この制限は、コントロール権の除去を許すが、挿入は許さない。この場合、システムはコントロール権の観点では完全に静的ではないが、縮んでいくシステムである。したがって、本質的にはタイプ1のプロテクション・システムに対する安全性の決定手続きと同じ手続きを、タイプ2システムの安全性の決定に使うことができる。

ゆえに、次の結果は定理1から、ただちに得られる。

### 系 1

任意にタイプ2のプロテクション・システムとそのアクセス行列の任意の初期構成とが与えられたとき、システムが任意に与えられた権利に対して安全か否かを決定するアルゴリズムが存在する。

制限1のその他の自明な緩和は、コントロール権の挿入は許すが、削除を許さないようにする修正である。

destroy 操作で主体や対象が消滅することによってコントロール権が削除されることのないよう、destroy 操作は禁止することにする。

この制限によって、コントロール権の追加のみが行われるという意味で、システムは増大する。Harrison<sup>[9]</sup> は delete, destroy object および destroy subject 命令を許さないシステムのことを“単調なシステム (monotone system)”と呼んでいる。そこで Graham, Denning の論文<sup>[2]</sup> の定理1をいいかえると、「単調なシステムに対して安全性を決定するアルゴリズムは存在しない。」ことになる。

単調性は、われわれの次の制限3より厳しいから、Harrison の定理1から、ただちに次の結果を得る。

### 制限 3

$\alpha \in (R \cap A)$  ならば、 $\alpha$  を含むすべての操作は enter 操作である。さらに、destroy 操作は禁止する。

### 定理 3

タイプ3システムに対する安全性を決定するアルゴリズムは存在しない。

したがって、安全性問題の決定可能性と決定不能性の境界はシステムに許されるダイナミズムの程度という観点からとらえることができよう。いままで、コントロール権を用いて、システムが静的ではあるが縮むのであれば、安全性は決定可能であることをみてきた。しかし、もしシステムが単調に増大するか、増大し同時に縮小することを許すならば、安全性は決定不能である。

特定のシステムについてのこの性質に及ぼす制限の影響を考える前に、まず次の（最後の）制限した Harrison-Ruzzo-Ullman モデルを導入する。

### 制限 4

$$(\forall i)[R_i \neq \phi \rightarrow (R_i \cap (R - \overset{\circ}{A})) \neq \phi].$$

ここで、 $\overset{\circ}{A} = \{\alpha | (\exists i)(\alpha \in A_i \wedge R_i \neq \phi)\}$  である。

直観的にいえば、この制限は「命令が条件をもつならば、その命令はコントロール権を含む任意の命令の中で影響を受ける権利とはならないような、ある種のコントロール権を含んでいる」ことを示している。 $\overset{\circ}{A}$  は、コントロール権の空でない集合をもつ任意の命令の中で影響を受ける権利として現われている権利の集合であることに注意する。

$A$  でなく  $\overset{\circ}{A}$  によるこの定義は、Jones-Lipton-Snyder システムの場合のような CREATE 命令をここで排除しないようにしたいということによる。このような命令は、新しい主体を生成するだけでなく、その主体を生成した主体にあるコントロール権を含む権利を付与



する。たとえば、Jones-Lipton-Snyder の TAKE-GRANT モデルを修正して、 $r$  や  $w$  以外の任意の権利を主体が取得したり、許可したりできるようにすれば、タイプ4のプロテクション・システムの例を得る。これについてはあとで研究しよう。

**定理 4**

任意のタイプ4のプロテクション・システムとそのアクセス行列の任意の初期構成を与えられたとき、システムが任意に与えられた権利  $\alpha$  に対して安全であるか否かを決定するアルゴリズムは存在しない。

**【証明】**

任意の Harrison-Ruzzo-Ullman のプロテクション・システム  $\mathcal{P}$  に対応して、 $\mathcal{P}$  が  $\alpha$  を漏洩するとき、そのときに限って  $\mathcal{P}'$  が  $\alpha$  を漏洩するようなタイプ4のプロテクション・システム  $\mathcal{P}'$  が存在することを示そう。さらに、 $\mathcal{P}$  は  $\mathcal{P}'$  にエフェクティブに翻訳できる。したがって、タイプ4システムの安全性を決定するアルゴリズムによって任意のプロテクション・システムの安全性を決定するアルゴリズムが得られるが、これは矛盾である。

$\mathcal{P}$  と任意の Harrison-Ruzzo-Ullman のプロテクション・システムとし、そのアクセス行列を  $M$ 、現主体の集合を  $S$ 、現対象の集合を  $O$ 、また  $T$  を権利の集合、 $C$  を命令の集合とする。

次のように定義するアクセス行列  $M'$ 、現主体の集合  $S'$ 、現対象の集合  $O'$ 、権利  $T'$ 、命令  $C'$  をもつシステム  $\mathcal{P}'$  を次のように考える。

$\delta$  を  $\delta \in T$  であるダミーの権利とし、

$$T' \triangleq T \cup \{\delta\}$$

とする。  $s_a$  を  $s_a \in O$  であるダミーの主体とし、

$$S' \triangleq S \cup \{s_a\}, O' \triangleq O \cup \{s_a\}$$

とする。

$M'$  を、行列  $M$  に  $s_a$  と対応する一行一列を付加した行列とし、 $M'(s_a, s_a) = \{\delta\}$  とする。すなわち、 $s_a$  は  $s_a$  に対する権利  $\delta$  をもつが、他の主体はすべての対象に対して権利  $\delta$  をもたないようにする。  $C$  の各命令  $C_i$  に対して、 $C_i$  と同じ条件の集合をもち、さらに条件：

$$\text{if } \delta \text{ in } (s_a, s_a)$$

を追加した命令  $C_i'$  を  $C'$  の命令とする。なお、 $C_i'$  の本体は  $C_i$  の本体と同じとする。

各命令  $C_i'$  に現われる追加条件は常に満たされることに注意しよう。さらに、 $C'$  のすべての命令は影響を受ける権利ではないコントロール権  $\delta$  を含んでいるから、 $\mathcal{P}'$  は制限4の条件を満たす。

最後に、 $C'$  の各命令に追加した条件は常に満たされるから、所望のごとく権利  $\alpha$  が  $\mathcal{P}'$  から漏れ出るとき、そのときに限って、 $\alpha$  は  $\mathcal{P}'$  から漏れ出る。 ■

したがって、制限4で表現される安定性の尺度は、安全性の決定可能性にとって不十分である。さらに、安全性が決定可能である場合でも、安全性の決定問題は NP-完全である。

**3.2 Jones-Lipton-Snyder の TAKE-GRANT モデルの修正**

Lipton らの論文<sup>[5], [6]</sup>に、Jones-Lipton-Snyder システムにおいて  $p$  can  $\alpha$   $q$  か否かを決定する線形時間量のアルゴリズムが示されている。このシステムを考察すると、主体が対象を取得できるか否かをこのように容易に決定できる理由の一つは、権利が非常に容易に取得できるということがわかる。

さらに、権利の二つの役割が、このように権利の取得を容易にしている。そこで、まずこれらの役割を区別して Jones-Lipton-Snyder システムを修正する。すなわち、図2のシステムを考える。ここで  $\alpha$  は  $r, w$  以外の任意の権利である。これは Snyder の論文<sup>[1]</sup>で導入された不活性権利の概念に似てくる。この修正システムは、われわれがタイプ4のプロテクション・システムと呼んだシステムの特別の場合である。このような TAKE-GRANT システムは制限1の要請を満たさないことに注意する。われわれは直観的に  $r$  と  $w$  をコントロール権と考え、( $r, w$  以外の権利)  $\alpha$  を影響を受ける権利と考えているが、形式的には、 $\alpha$  は命令の *if* 部と *then* 部双方に現われうるという意味でコントロール権であり、同時に影響を受ける権利である。この考察によって、われわれは制限4を導入したわけである。さらに、先に述べたように新たに生成された主体に対するあるコントロール権を、それを生成した主体に与える CREATE 命令の存在を許すために、集合  $\dot{A}$  を制限4の定義で導入した。この場合、 $\dot{A}$  は  $r, w$  以外の権利の集合である。

プロテクション・グラフの二つの節点間に方向を無視して道が存在するとき、二つの節点は“連結している”という。

#### 定理 5

$\alpha$  が  $r, w$  以外の権利であるようなタイプ4の TAKE-GRANT システムにおいて、 $p$  が次の形の道で  $q$  に連結しているとき、そのときに限って  $p \text{ can } \alpha q$  である。

- 1)  $((\vec{r}) * (\overleftarrow{w}) * \vec{r}) \alpha$ , または,
- 2)  $((\vec{r}) * (\overleftarrow{w}) * \overleftarrow{w}) \alpha$

ここで  $\vec{r}, \overleftarrow{w}, \alpha$  は自明な方法で定義する。また  $*$  は Kleene の  $*$  である。

#### [証明]

必要性は帰納法で容易に示されるので、ここでは十分性を示す。まず、すでに連結していない既存の節点を連結する規則はない。さらに、既存の節点  $x$  が他の既存の節点  $y$  への権利  $\alpha$  を取得できる唯一の場合は、 $\alpha$  を  $r, w$  以外の権利とすると、 $(xrx)$  または  $(zwx)$  であり、かつ  $(x\alpha y)$  となる節点  $z$  が存在するときである。また、既存の節点が他の節点への権利  $r$  または  $w$  を取得する方法は存在しないことに注意しよう。したがって、権利  $\alpha$  が伝わりうる場合は二つの節点間に1あるいは2の形の道が存在する場合に限られる。■

ここで、基本となる Jones-Lipton-Snyder システムと異なり、新しく生成された主体は既存の主体が新しい権利を取得する際に、どんな役割も果たさないことに注意する。

#### 系 2

タイプ4の TAKE-GRANT システムにおいて  $p \text{ can } \alpha q$  か否かを線形時間で決定するアルゴリズムが存在する。

#### [証明]

所要のタイプの道が存在するか否かを検証する問題は、線形時間で連結性問題に還元できるが、この問題には線形のアルゴリズムが存在する。■

したがって、制限4を満たすように Jones-Lipton-Snyder システムを修正しても、線形時間で  $p$  が対象  $q$  への権利  $\alpha$  を取得できるか否かを決定できるシステムを得る。

このシステムに制限を追加すると、 $p$  が権利  $\alpha$  を取得することがより困難になり、さらに、そのような権利の取得が実際の状態に一層合うようにできることがわかる。

いままでの考察によると、 $p \text{ can } \alpha q$  となる唯一の場合は、 $p$  が権利をわたす権利をもつ主体と権利とをわたす方向へ作用する場合であった。しかし、それでも主体は新しい権利を比較的容易に取得できる。つぎに少々上のシステムを修正して、権利の取得をさら

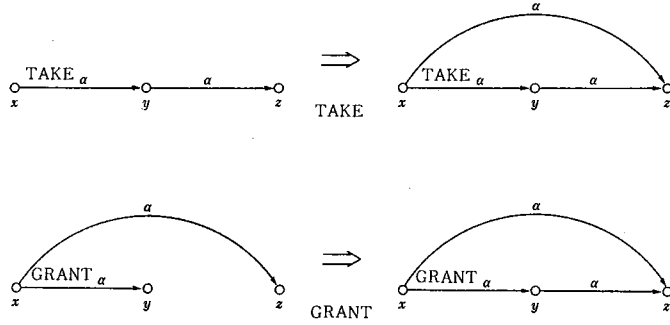


図 2 命令のグラフ表現  
Fig. 2 Commands represented graphically

に困難にする.  $r$  や  $w$  に他のすべての権利  $\alpha$  の取得をコントロールさせる代わりに, 各権利  $\alpha$  に対してコントロール権  $\text{TAKE}_\alpha$  と  $\text{GRANT}_\alpha$  の対を導入しよう. そこで, プロテクション行列に現われうる権利が  $n$  個存在すれば, われわれのシステムは図 2 に表現したような  $2n$  個の規則の集合をもつ.  $\text{CREATE}$  命令は生成主体に新しい主体に対する任意の, ないしはすべての権利を与えることができる. しかし, 新しく生成された主体は既存の主体が他の既存の主体に対する権利を取得する際に, 何の役割も果たさない. したがって, 使用者は自分がコントロールし, 順に欲する権利を取得し, それらをその生成主体にわたすような代理の主体を生成することはできない.

そこで定理 5 の証明と本質的に同じ証明によって, ただちにその定理を得る.

**定理 6**

図 2 の  $\text{TAKE-GRANT}$  システムにおいて, 主体  $p$  が  $q$  に対する権利  $\alpha$  を取得できる必要十分条件は,  $p$  が次の形の道によって  $q$  に連結していることである.

$$((\overline{\text{TAKE}_\alpha})^* (\overline{\text{GRANT}_\alpha})^*)^* \overline{\text{TAKE}_\alpha} \vec{\alpha}$$

または,

$$((\overline{\text{TAKE}_\alpha})^* (\overline{\text{GRANT}_\alpha})^*)^* \overline{\text{GRANT}_\alpha} \vec{\alpha}$$

ここで, われわれは任意に与えられた権利に対する安全性を線形時間で決定できる. さらに, ちょうど一つの他の権利の取得をコントロールする権利の概念を導入することによって, 権利取得のむずかしさを, やや増大させたことになる. 第 1 章で述べたように, このタイプの特権化は直観的かつ実際的に道理にかなう. このシステムにおいて, 主体が新しい権利を取得できる必要十分条件は, とくに  $\alpha$  をわたす権利をもった主体と作用できて取得することである.

さて, Jones-Lipton-Snyder システムの異なった修正版を考える. 同様に, この修正システムにおいても, もとのシステムより権利の取得はよりむずかしいが, 安全性は線形時間で決定可能である. 前述のシステムと異なり, このシステムで使用できる追加コントロールは権利の役割の厳密な分離から導かれるものではない. この修正システムには Graham と Denning が彼らの論文<sup>[2]</sup> で論じた“複製フラッグ (copy flag)” の概念を組み入れる. 複製フラッグの概念の背景となる直観的考察は, 対象に何らかの操作を行う主体の権利と, この権利を他の主体にわたす権利とを区別できるようにしたいということである. 明らかに, われわれはこの後者の一層強力な権利をもっと注意深く制限したい. この動機から, われわれは図 3 に示す  $\text{TAKE-GRANT}$  システムを導入する.

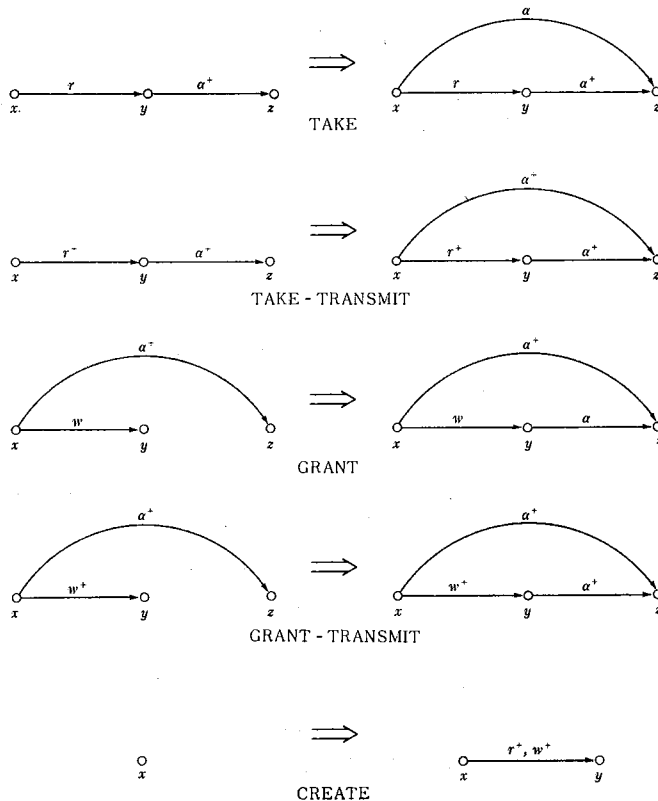


図 3 複製フラッグ付きの TAKE-GRANT システム  
 Fig. 3 A copy flag TAKE-GRANT system

もとの Jones-Lipton-Snyder システムと同様に、 $\alpha$  は  $r$  または  $w$  である。さらに、 $p$  が  $\alpha^+q$  に対する権利をもつならば、 $p$  は  $\alpha q$  への権利をもち、また上述の枠組の中でこの権利を他の主体にわたす権利をもつ。この  $\alpha^+$  を“フラッグ付きの権利 (flagged right)”と呼ぶ。もし  $p$  と  $q$  が、すべてのアークがフラッグ付きの権利でラベル付けられているような道で連結しているか、または  $p=q$  のとき、 $p$  と  $q$  は“フラッグ付きで連結している (flag connected)”という。

このような複製フラッグ付きの TAKE-GRANT システムにおいて、どのような環境のもとで、 $p$  は  $q$  への権利  $\alpha^+$  を取得できるか、またどのような環境のもとで、 $p$  は  $q$  への権利  $\alpha$  が取得できて  $\alpha^+$  が取得できないかを調べたい。

われわれの次の二つの結果は、この問題に関する結果である。このうち、前者の証明は Lipton らの論文<sup>[6]</sup>の主定理の証明と似ているから、ここでは重複して述べない。前に導入したシステムと対照的に、このシステムにおける権利取得の鍵は権利を取得して、それを自分を生成した主体へわたすような新しい主体の生成にある。

**定理 7**

複製フラッグ付き TAKE-GRANT システム (図 3) のプロテクション・グラフ  $G$  において、 $p, q$  を相異なる節点とする。  $\alpha$  を  $r$  または  $w$  とする。このとき、 $p$  can  $\alpha^+q$  の必要十分条件は次の 2 条件を満たすことである：

- 1)  $p$  と  $q$  は  $G$  でフラッグ付きで連結している。かつ、
- 2)  $G$  で  $s\alpha^+q$  となる  $G$  内の節点  $s$  が存在する。

**定理 8**

複製フラッグ付き TAKE-GRANT システムのプロテクション・グラフ  $G$  において、 $p, q$  を相異なる節点とする。また、 $\alpha$  は  $r$  または  $w$  とする。このとき、 $p \text{ can } \alpha q$  の必要十分条件は、

- 1)  $G$  で  $p \alpha q$  であるか、
- 2)  $G$  節点  $s, t$  が存在して、
  - (a)  $(prt)$  または  $(twp)$ 、かつ、
  - (b)  $t$  と  $q$  はフラッグ付きで連結している、かつ、
  - (c)  $s \alpha^+ q$

を満たす。

**[証明]**

(←) 条件 1) または 2) が成り立つとする。前者の場合、定理が成り立つのは自明である。そこで、条件 2) が成り立つと仮定する。定理 7 によって  $t \text{ can } \alpha^+ q$  である。

したがって、段階  $n$  が存在して、プロテクション・グラフ  $G_n$  において

- 1)  $(prt)$  かつ  $(t \alpha^+ q)$  であるか、
- 2)  $(twp)$  かつ  $(t \alpha^+ q)$  が成り立つ。

上の場合には、命令を 1 (TAKE) を作用させると、 $G_{n+1}$  において  $p \alpha q$  となり、したがって  $p \text{ can } \alpha q$  である。

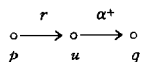
同様に、下の場合には命令 3 (GRANT) を 1 回作用させると、 $p \text{ can } \alpha q$  となる。

(→)  $G_i$  で、 $p \alpha q$  となることをグラフの段階  $i$  についての帰納法で証明する。

$p \text{ can } \alpha q$  と仮定する。このとき  $G_i$  で  $p \alpha q$  となるような  $i$  が存在する。もし  $i=0$  ならば、 $G$  で  $p \alpha q$  であるから、条件 1) が成り立つ。

$i>0$  ならば、ラベル  $\alpha$  をもつアークを追加できる唯一の方法は TAKE あるいは GRANT 操作、すなわち、命令 1 あるいは 3 による。

もし  $\alpha$  が命令の作用によって追加されるならば、段階  $i-1$  において



である。 $i-1=0$  ならば、 $pru$  で、 $u$  は  $q$  にフラッグ付きで連結されており、 $u \alpha^+ q$  である。したがって、条件 2) を満たす  $G$  内の節点  $s$  と  $t$  が存在する。

定理が  $0 < i-1 \leq n-1$  に対して成り立つと仮定しよう。 $i-1=n$  を考える。 $0 \leq j < n$  を  $u$  がシステムに追加された段階とする。 $(j>0$  ならば、 $u$  は CREATE 命令の作用によって追加されており、そうでなければ、 $u$  は  $G=G_0$  の節点である。) よって、 $G_j$  において、 $u \text{ can } \alpha^+ q$  である。したがって、定理 7 より、 $u$  と  $q$  は  $G$  でフラッグ付きの連結であり、 $v \alpha^+ q$  となる節点  $v$  が存在する。

さらに、 $G_j$  で  $p \text{ can } r u$  である。

帰納法の仮定から、 $prb$  または  $bwp$  となる (ピボット要素と呼ぶ)  $b$  が  $G_j$  内に存在し、 $b$  と  $v$  はフラッグ付きで連結している。よって、 $G_j$  において  $b$  と  $q$  は節点  $v$  を介してフラッグ付きで連結している。

もし  $j=0$  ならば、証明は終わっている。もし  $j>0$  ならば、 $b$  が追加された段階  $k$  を考え、同様の証明を繰り返せばよい。

段階  $m$  のピボット要素がシステムに追加された段階は、常に  $m$  以下であるから、帰納法の仮定と定理 7 をたかだか有限回適用すれば、 $prt$  または  $twp$  となる節点  $t$  が得ら

れ,  $t$  と  $q$  は  $G$  でフラッグ付きで連結している. さらに, 定理 7 によって,  $G$  で  $s\alpha^+q$  となる節点  $s$  が存在する.

$\alpha$  が段階  $i$  で命令 3 の作用で追加されたときも, 同様である. ■

#### 4. おわりに

われわれは Harrison-Ruzzo-Ullman モデルに対するいろいろな制限を検討した. これらの制限は, 規則の適用を許しプロテクション行列を修正するような権利のクラスと, 主体が対象に対して何らかの操作を行うことを, 許すような権利のクラスとの間の相互作用を制限しようという動機にもとづいていた. われわれは, 最も厳しい制限に対しても安全性問題は  $NP$ -完全であり, 制限をやや緩和すると, 問題が決定不能になることをみた.

ついで, Jones-Lipton-Snyder システムの修正版を調べた. これらの特定のシステムに対しては, 主体が対象への権利を取得できるか否かを問う決定問題は, 決定可能であるばかりでなく, この問題を決定する線形アルゴリズムが存在することを示した.

このシステムに制限を加えることによって, われわれは第 1 章で概説した意味でシステムをより魅力的にできたばかりでなく, 主体が対象への権利を取得することを一層むずかしくできたし, また権利取得の問題の線形時間量での決定可能性が保存できた.

われわれは他のセキュリティ・システムに対しても同様の問題を追及し, この論文で導入した権利のクラスの区別がもたらす結果を調べたいと考える.

もう一つの興味深い方向は, 対象を(主体と非主体の)二つに分ける代わりに, 規則の適用の能力にもとづいているいろいろなクラスに分割することである. したがって, 一つの規則のすべての条件を満たさなければならないというだけでなく, さらに使用者も規則を適用する権利をもたなければならないようになろう. このような区別は限定された意味で Harrison, Ruzzo, Ullman の論文<sup>[4]</sup>で行われている. この場合, 主体は権利を取得 (TAKE) あるいは許可 (GRANT) し, 対象を生成 (CREATE) できるが(すなわち, 規則を適用できるが), 対象はこの能力をもたない. あるいは, 各タイプが規則の適用能力で異なっているような一つのタイプの階層が, システム内に存在すると考えることもできよう. このようなシステムは, 各対象に付けられたある不変のコントロール権と各規則に必要なそうした権利を入れて模倣できるであろう.

最後に, この分野に対する私の関心を喚起してくれたことについて, N. Minsky に感謝し, 有益なコメントをくれた T. Ostrand と P. Abrahams に感謝する.

(テクニカル・パブリケーション室 山田 真市 訳)

- 参考文献 [1] A. V. Aho, J. E. Hopcroft, J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, Mass., 1974.
- [2] G. S. Graham, P. J. Denning, "Protection-principles and practice," *Proc. AFIPS 1972 SJCC*, Vol. 40, pp. 417-429.
- [3] M. A. Harrison, W. L. Ruzzo, "On Synchronization and Security," in *Foundations of Secure Computation*, Ed. R. A. DeMillo, et al., Academic Press, London, 1978, pp. 337-363.
- [4] M. A. Harrison, W. L. Ruzzo, J. D. Ullman, "Protection in operating systems", *Comm. ACM*, Vol. 19, No. 8, August 1976, pp. 461-471.
- [5] A. K. Jones, R. J. Lipton, L. Snyder, "A linear time algorithm for deciding subject-object security", *Proc. 17th Annual Foundations of Computer Science Conf.*, Houston, 1976, pp. 33-41.
- [6] R. J. Lipton, J. Snyder, "A linear time algorithm for deciding subject security," *J.*

ACM, Vol. 24, No. 3, July 1977, pp. 455-464.

- [7] N. Minsky, "The Principle of Attenuation of Privileges and its Ramifications", in *Foundations of Secure Computation*, Academic Press, 1978, pp. 255-276.
- [8] L. Snyder, "Formal Models of Capability-Based Protection Systems," Tech. Report 151, Yale University, 1979.
- [9] L. Snyder, M. Bishop, "The Transfer of Information and Authority in a Protection System," *Proc. 7th Symposium on Operating Systems Principles*, Pacific Grove, Ca. Dec. 1979, pp. 45-54.

執筆者紹介 Elaine J. Weyuker

New York 州立大学 Binghamton 校 (Harpur College) より数学で A. B. を, Pennsylvania 大学より計算機科学で M. S. E. を, Rutgers 大学より計算機科学で Ph. D. をそれぞれ取得。1968年から1969年まで IBM でシステム・エンジニアとして勤務。1969年から1975年まで New York 州立大学 Richmond 校で計算機科学の講師を務める。1977年より New York 大学の Courant Institute of Mathematical Sciences の助教授。Sperry Univac 社のソフトウェア工学研究グループに属す。研究分野はプログラム・テストと検証, プログラム図式, アルゴリズムの設計と解析など。



## 報告 UNIVAC 1100/90 シリーズの新アーキテクチャ

### New Architecture of UNIVAC 1100/90 Series

古沢 純一, 城川 孝二, 田中 厚司

**要約** 急激に増大する情報量, 加速度的に拡大していく利用層, 多様化し高度化する情報処理形態といった状況のもとで, コンピュータに望まれることは高い処理能力と同時に使いやすさである。UNIVAC 1100/90 シリーズでは, こうした時代の要請に応じて, 従来のアーキテクチャに加えて新しいアーキテクチャを開発したが, これは第4世代機とも呼べるもので, 将来への方向づけを行ったといえる。

本稿では, UNIVAC 1100/90 シリーズに採用された七つの新アーキテクチャのうち, その中心となるアドレス空間の拡張と, リング/ドメイン概念によるメモリ保護を中心に解説した。

**Abstract** Explosion of the amount of information, augmented class of end users and variety of sophisticated data processing are the fundamental characteristics of the environment in which today's computers are utilized. Among numerous expectations of computers in these days, high performance and ease of use are highly requested. UNIVAC 1100/90 series computers are fully employing new architectural technology.

This new architecture gives suggestions of the direction of computer technology in 1980's. This paper describes the major parts of seven new architectural items; the addressing architecture and the Ring/Domain protection architecture.

#### 1. はじめに

今日のコンピュータの利用傾向をいい表わすとすれば, 「個人使用の拡大」と「情報処理量の増大」である。その典型的な現われは, 分散処理の急激な発展であり, オフィス・オートメーション(OA)の普及とその導入となっている。1980年代後半にはこれらは定着し, 従来のEDP処理の形態と肩をならべる形になる。そのときに要求されることに, 不特定多数の人々に提供する徹底した使いやすさと安全な環境とがある。

Sperry Univac社では, この変化に対応して従来のアーキテクチャに加えて新しいアーキテクチャを開発し, 将来への方向づけを行った。この開発はまた, 高いコスト・パフォーマンスの実現と過去の資産に対する保護, および新しい機能の追加を目的としている。

本稿では, 1100/90 シリーズで実現した七つの新しいアーキテクチャ

- 1) 事実上限界を意識せずに使える巨大な仮想アドレス空間 (69 ギガ語相当)
- 2) 安心したパーソナル・ユースの世界を実現するリング/ドメイン概念の導入
- 3) ファームウェア化による OS パフォーマンス機能の向上
- 4) 専用化された各種プロセッサ間の接続を容易にする UPI (Universal Processor Interface) の開発
- 5) ホスト・コンピュータ能力を増大させる I/O プロセッサの処理機能強化
- 6) テスト環境を大幅に改善する仮想計算機機能の採用
- 7) 新しい世界を実現するエクステンデッド・モード, 従来との互換性を守るベーシック・モードおよびエクステンデッド・モードへの移行を容易にするミックスド・モードの採用



のうち、1) と 2) の最新アーキテクチャを中心に紹介する。

新アーキテクチャは、アドレス空間の拡張とリング/ドメイン概念による保護とが二つの柱をなしている。アドレス空間(仮想, 実空間とも)の拡張を実現したことによって、利用者はプログラミングにおけるユーザ領域の制限をまったく意識する必要がなくなり、開発の生産性向上が大幅にはかれるようになった。エンド・ユーザ層が急速に広がり、不特定多数の利用者が増大しても、リング/ドメイン概念によるメモリ保護の実現によって、それぞれの利用者が機密保護を遵守し、他から干渉されずに安心して利用できるようになる。

## 2. 新アドレッシング・アーキテクチャ

### 2.1 アドレッシング・アーキテクチャの変更点

1100/90 シリーズの多重仮想記憶方式は、1100/80 シリーズと同じ多重バンク方式を採用した。この 36 ビット仮想アドレス方式の採用に対応したアーキテクチャ上の変更点は次のとおりである。

- 1) プログラム・アドレス・レジスタ (PAR) の形式変更: 24 ビット実アドレスから 36 ビット仮想アドレスへの変更。
- 2) バンク記述子 (BD) の形式変更: バンクのサイズが 262 キロ語ごとに一つの BD を対応させ、BD を 4 語に拡張。
- 3) バンク記述子テーブル (BDT) の拡張: BD を最大 65,536 ( $=2^{16}$ ) 個収容する BDT を 1 システム当たり四つに拡大。
- 4) ベース・レジスタ (Bレジスタ) の新設: 従来の四つのバンク記述子レジスタ (BDR) を廃し、ユーザ用に 16 個、システム制御用に 16 個の B レジスタ (72 ビット) を準備 (これにより、268 メガ語ものアドレス空間が直接参照可能)。
- 5) 命令語形式の変更: 従来 16 ビットでオペランドを指示していた r フィールドが、B レジスタ指示のための b フィールド (4 ビット) とオペランド指示のための d フィールド (12 ビット) に分割。また、ユーザは 24 ビット (16 メガ語) の相対アドレッシングが可能。
- 6) 指示レジスタの変更: 実行中のアクティビティの稼動条件などを保持する指示レジスタは、システム制御部分とプログラム制御部分とに明確に分離。
- 7) インジケータ/ドメイン・レジスタ (I/D レジスタ) の新設: リング/ドメイン概念の導入に従って、I/D レジスタを設置。
- 8) 新命令語の追加: 多くの新命令語を追加し、同時にバンク操作の命令語群を再定義。
- 9) システム・テーブルのハードウェアによる定義: 従来、ソフトウェアで定義されていた一部のシステム・テーブルを、ハードウェア・アーキテクチャとして定義。

### 2.2 主な変更点の詳細

#### 2.2.1 B レジスタと PAR

36 ビット仮想アドレス方式を実現するために B レジスタを新しく導入し、PAR の形式を変更した。

B レジスタは、アドレスの形式とアドレスの妥当性確認、およびアクセス条件の検査のために使用される。その内容は BD から複写し、形式は図 1 の右上のような形をしている。ユーザ用に 16 個、システム制御用に 16 個、合計 32 個ある。

実メモリ上の次に実行すべき命令語の番地を指示する PAR は、従来の実アドレスを直接保持する 24 ビット形式から、仮想アドレス形式で保持する 36 ビット形式に変更され

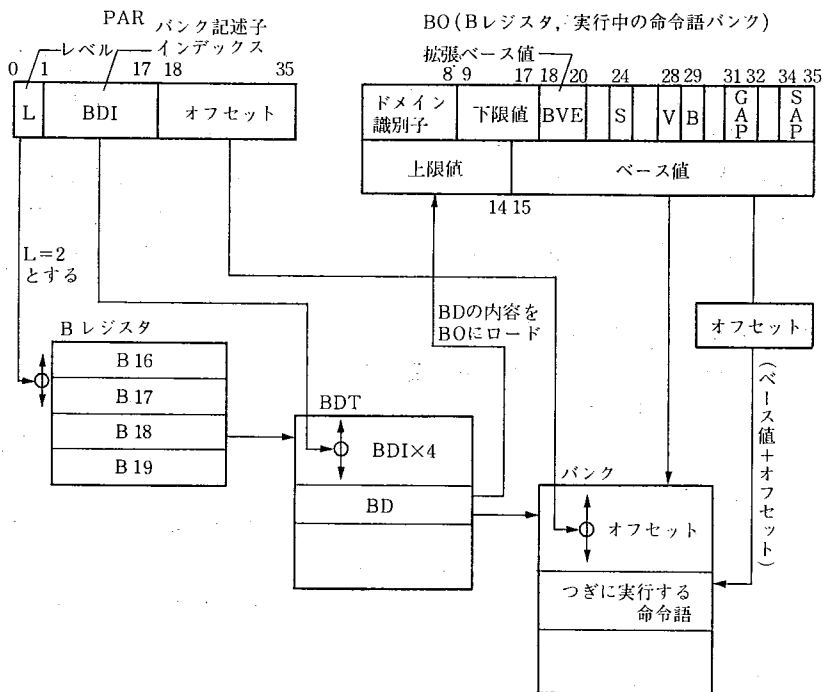


図 1 命令語の取り出し  
Fig. 1 Instruction fetch

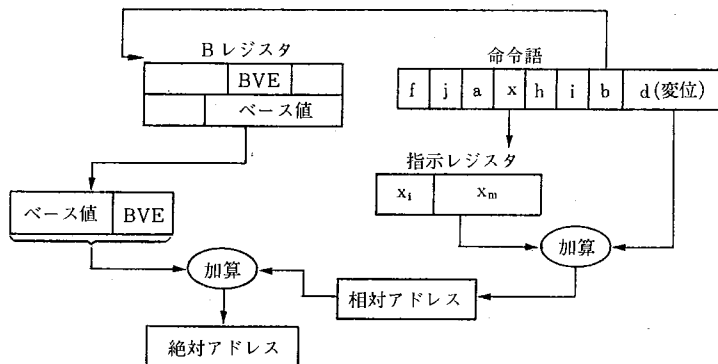


図 2 取り出した命令語のアドレス計算例  
Fig. 2 Example address calculation concerning fetched instruction

た。この仮想アドレスは左の 2 ビットで BDT の四つのレベルを表わし、次の 16 ビットでバンクの選択に使用するバンク記述子インディックス (BDI) を、右の 18 ビットでそのバンクのオフセットを表わす (図 1)。

これら B レジスタと PAR の関係を図 1 に、また取り出した命令語のオペランドのアドレス計算例を図 2 に示す。

### 2.2.2 バンク記述子 (BD) とバンク記述子テーブル (BDT)

BD と BDT は、アドレッシングの管理に用いる基本的ストラクチャである。BD は、B レジスタにロードするときに必要な主記憶領域の管理、および保護の情報をもっている。BD はバンクごとに存在するのが基本であるが、16メガ語までは表現できる。ラージ・バンクの場合 262 キロ語ごとに 1 個の BD をもつ。BD は 4 語長に拡張され、図 3 のような

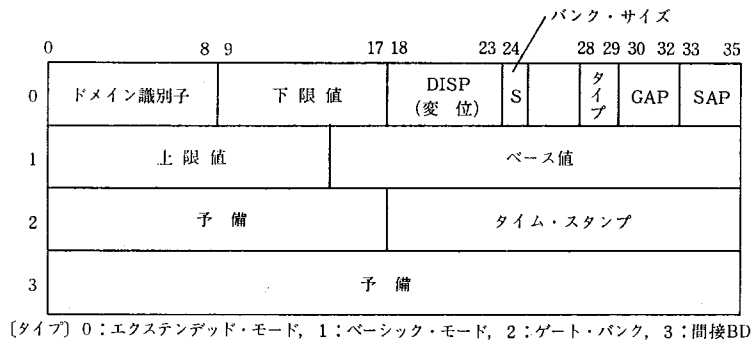


図 3 BD の形式  
Fig. 3 BD format

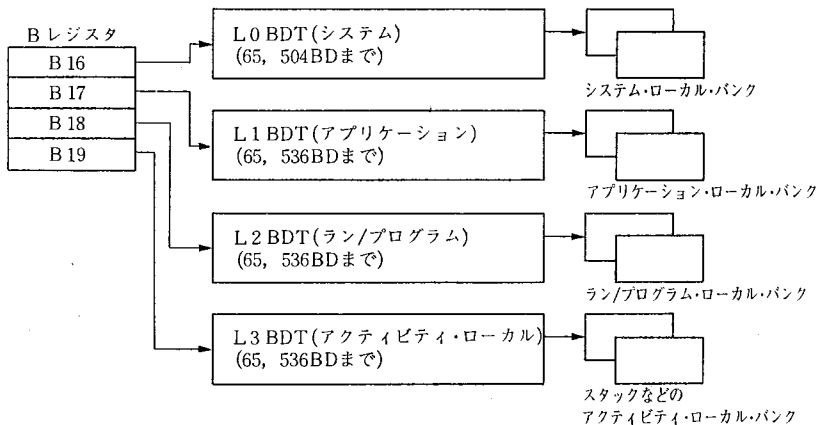


図 4 B レジスタと BDT の関係およびアドレス空間の構成図  
Fig. 4 Diagram of address space

形式をしている。

システム当たり四つの BDT で、アクティビティがアドレス可能な全仮想アドレス空間を表現する。各 BDT 内には最大 65,536 個までの BD が作れる。図 4 より、ユーザ・プログラムのアクティビティからみた全仮想アドレス空間の大きさは、 $(65504 + 65536 \times 3) \times 262K \approx 69$  ギガ語となり、非常に大きな仮想アドレス空間がもてるようになった。従来の機種全仮想アドレス空間は 1.1 ギガ語にすぎないので、今回の拡張により約 60 倍に拡大したことになる。

### 2.2.3 ラージ・バンクのサポート

16 メガ語までの実アドレス空間をサポートするため、 $2^{24}$  語までの相対アドレスが使用可能となった。

バンクのサイズには“ラージ”（最高  $2^{24}$  語）と“スモール”（最高  $2^{18}$  語）の二つがあり、その判別のために B レジスタと BD にサイズ (S) ・ビットを設けている（スモール・バンク：S=0，ラージ・バンク S=1）。実アドレスの形成方法は、バンクのサイズによって異なるということはないが、BD の扱いに違いがある。すなわち、一つのラージ・バンクを表現する各 BD には、何番目の BD であるかを示す DISP (Displacement) フィールドと呼ばれる欄が設けられている（図 3）。

ユーザはエクステンデッド・モード下では、命令語の i フィールド (図 2 参照) を指定して 24 ビットの相対アドレッシングができる。また、バンクの一部分だけを B レジスタにロードできるサブセッティングと呼ぶメカニズムがある。このメカニズムにより命令語の d フィールド (図 2) が 12 ビットであるというプログラミング上の制約が解消する。

#### 2.2.4 システム制御レジスタ

システム制御レジスタでは、指示レジスタの形式を変更するとともに I/D レジスタを追加した。指示レジスタは操作モードと CPU 稼働の条件を制御するレジスタで、システムとユーザを制御する部分に分かれる。主な変更点は、システム制御指示部分に対するプロセッサ特権 (PP) と CPU の動作モード (BM) フィールドとの追加である。各命令語には実行可能な PP の値が決められている。指示レジスタは図 5 のような形式である。I/D レジスタは命令の実行途中でインタラプトされたときの状態表示、インタラプトの表示およびドメインの保持に使用し、形式は図 6 のようである。

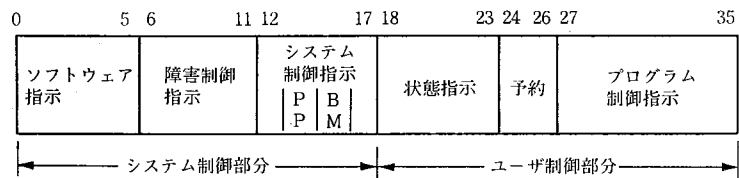


図 5 指示レジスタの形式

Fig. 5 Format of designator register

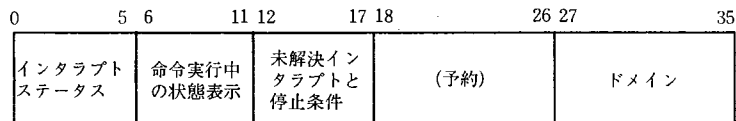


図 6 I/D レジスタの形式

Fig. 6 Format of I/D register

### 2.3 メモリ・アクセスの保護能力の向上

1100/90 シリーズでは、メモリ・アクセスの保護にキー (鍵) とロック (錠) のメカニズムを取り入れ、その能力を飛躍的に向上させた。

キー情報は I/D レジスタ中のドメイン・フィールドの内容に相当し、ロック情報は BD, または B レジスタ中のドメイン識別子に相当する (図 1, 図 3)。このドメインとドメイン識別子とのアクセス条件が一致しなければ、アクセスは許されない。

ドメインおよびドメイン識別子は、リング番号 (2 ビット) とドメイン番号 (7 ビット) に分けられる。リングとはアドレス空間を 4 階層に分けたもので、さらにその各々を 128 分割した一つずつがドメインとなる (図 7)。

すべてのプログラム・モジュールは、その機能や特性に応じて、どこかのリングおよびドメインに配置される。リングでは、リングの中心ほど共通度の高いモジュールを配置し、通常はリングの外側に置かれたモジュールから内側のモジュールにはアクセスできない。また、あるドメインに配置されたモジュールは、同一リング内でも、ある条件が満足されないと他のドメインに対してアクセスできない。

この仕組みを詳しくみてみよう。キーとロックの条件が一致したとしても、単に参照のための条件が整ったというだけで、ただちに参照が許可されるとは限らない。どのような

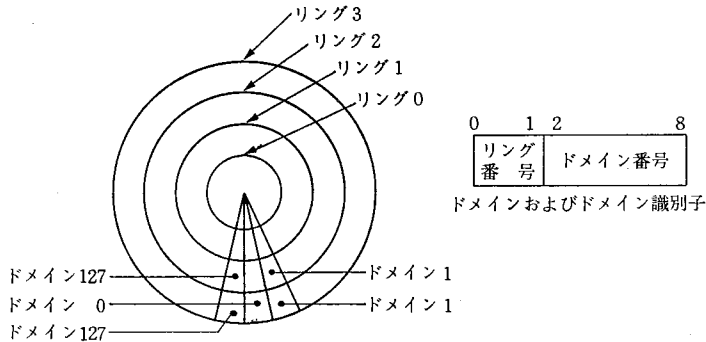


図7 リング番号とドメイン番号の対応

Fig. 7 Ring number vs domain number

参照形態が許されるかは、BD または B レジスタ中のアクセス制御フィールドに保持されている。アクセス制御フィールドには 2 種類あり、キーとロックが一致したときに検査される SAP (Special Access Permission) と、不一致とみなされたときに検査する GAP (General Access Permission) とがある (図1, 図3)。

SAP と GAP はともに 3 種類の参照形態 (読み出し, 書き込み, バンクへの進入の可否) を表現できる。

制御プログラム, あるいはデータベース・ルーチンと通常のユーザ・プログラムとは実行環境 (使用レジスタの種類, CPU 特権使用の許容度, ドメイン番号) が異なる。したがって、キーとロックの条件が一致し、要求する参照形態が許可されたとしても、目的のバンクで正しく稼働するためには、前もって要求者の実行環境を目的のバンク (ドメイン) での実行環境に合わせなければならない。すなわち、ゲートを介して要求者の実行環境を参照したい目的バンクの正しい実行環境に変更する。これをゲート処理という (3.4 節参照)。

正しく定義されたゲートを使用しなければ絶対にアクセスできないし、ゲート処理を正しく実行するにはゲート・バンクへの参照を示す BD 中のロック情報, およびゲート中のロック情報と要求者のキー情報の条件が一致しなければならない。このように、1100/90 シリーズではメモリ保護が十分にもなされている。

### 3. アクセス制御と保護

1100/90 シリーズのアドレッシング・アーキテクチャで規定されたハードウェアの動きのいくつかを 2 章においてみてきたが、本章では、OS 1100 がドメインとゲートの機構を用いたアクセス制御の詳細を述べる。

コンピュータ・システムの長時間連続運転を可能にし、各種の破壊からの情報保護、機密情報の漏えい防止などといったシステムの高信頼性安全性を実現するための基本原理は、ソフトウェアを機能別の論理的単位に区切り、それぞれが必要最小限の特権のみで実行するよう制御することである。メモリの保護についていえば、アクティビティ (CPU からみた仕事の最小単位) は、その時点で必要なバンクのみがアクセス可能のように制御することである。この制御された最小の命令バンクとデータ・バンクの集合をドメインと呼ぶ (図7)。

上記のドメインの慎重な運営を通じて、ハードウェアの誤動作やソフトウェアのバグなどによるエラーの局所封じ込めや早期回復が可能となり、システムの効率的な稼働が達成

されることとなる。さらに、ソフトウェア・メカニズムとして、ドメインの独立性によって独立に導入あるいは置換するのに有効な単位、またリンクの範囲を規定するリンケージ・グループとしての性格をも合わせもつような単位として定義できる。このようなソフトウェアの論理的単位をソフトウェア・パーティション、あるいは PIRL (Protection, Installation, Replacement, Linking) と呼ぶ。この考え方は、システム制御プログラム (EXEC) を含む OS 1100 に適用され、ユーザ・プログラムにとっても有効である。

### 3.1 アクセス制御

アクセス制御の基本メカニズムは、各ユーザの仮想アドレス空間の分離と同一仮想アドレス空間内の各バンクのアクセス範囲の限定である。

従来の UNIVAC 1100 シリーズでは、システムの仮想アドレス空間を EXEC BDT とユーザ BDT とで記述する 2 レベル BDT 構造であった。最初のレベルは全ユーザ共通で、コモン・バンクが対応する。2 番目のレベルはユーザ固有で、各ユーザ・プログラム・バンクが対応する。このような各ユーザ固有の仮想アドレス空間の分離により、他のプログラム・バンクへのアクセスができないような構造となっていた。

第 2 段階に相当するアクセス制御は、バンクごとにアクセス範囲の設定 (バンクの読み出し/書き込みの指示、バンクへのエントリ・ポイントの設定) を行うことにより実現する。しかし、これではある時点のアクティビティのアクセスをその処理に最小限必要なバンクとアクセスの範囲にしか限定できない。

このようなアクセス制御の下で EXEC をユーザ・プログラムから保護するために、EXEC とユーザの仮想アドレス空間を完全に分離し、ユーザ・プログラムが EXEC のサービスを受けるときには ER (Executive Return) 命令で内部割り込みを生じさせていた。

1100/90 シリーズでは、仮想アドレス空間の階層化を図 8 のように 4 レベルに強化した。レベル 0 の BDT はシステムにただ一つ存在し、システム・ローカル BDT と呼び、EXEC やコンパイラの実行時ライブラリなど、システム全体で共通に使用するルーチンやデータのバンク群を記述する。レベル 1 の BDT はアプリケーション・ローカル BDT と呼び、アプリケーションごとにグループ化されたバンク群を記述する。レベル 2 の BDT はラン/プログラム・ローカル BDT と呼び、ラン/プログラムのバンク群を記述する。レベル 3 の BDT はアクティビティ・ローカル BDT と呼び、一つの BDT で記述されるバ

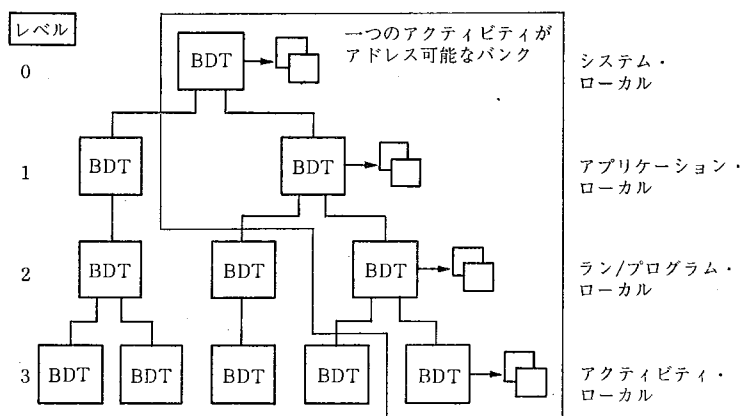


図 8 4 レベル・アドレス・ツリー

Fig. 8 4 levels address tree

ンク群は一つのアクティビティでのみアドレス可能である。このようなアドレス・ツリー構造によって、あるアクティビティがアドレス可能な仮想アドレス空間内のバンクは、アクティビティ・ローカルからシステム・ローカルに連なる四つの BDT で記述されるバンクである。バンクがアドレス・ツリー内のどの BDT に記述されるかは、そのバンクが属するパーティションをどのレベルに配置するかによって決まる。

この構造では、後に述べる保護メカニズムによって、従来機種と異なり EXEC でもユーザの仮想アドレス空間に入っている。これにより、ユーザは必要に応じて通常のサブルーチンと同様に EXEC を呼び出すことが可能となった。

### 3.2 保護メカニズム

アクセス制御の第 2 段階、つまりアクティビティのアドレス可能なバンクへのアクセス制御は、前述したように 1190/90 シリーズでは、キー/ロックのメカニズムを採用している。

各パーティションにはドメイン番号が割り当てられている。この番号は、そのパーティションを構成するバンクのロックのためのドメイン番号であり、その中で実行されるアクティビティのキーのドメイン番号になる。パーティション内での実行中のアクティビティが外部参照を行うと、ダイナミック・リンカを介して関連するバンクをメモリにロードするとともに、外部参照を外部定義にリンクする。ロードされたバンクのロックには、そのパーティションのドメイン番号を与える。同時に、バンクの初期状態として、バンクのリング番号に 0 を与える。一方、アクティビティのリング番号は通常 3 である。したがって、一つのパーティションに属するバンクの GAP フィールドをすべて 0 にしておくことで、そのパーティション内の各バンクをアクセスできるのは、そこで実行するアクティビティだけとなり、他のパーティションからの不正アクセスが防止できる。

未参照のパーティションの初期状態はインアクティブである。呼び出されるとアクティブになり、ドメインの割り当て（ドメイン番号 1~127）が行われる。ドメイン番号を割り当てる基本は、すでにアクティブなパーティションのそれとは異なるユニークなものを割り当てようになっている。

### 3.3 アクセス制御からみたバンクの分類

プログラムの一連の処理は一つのパーティション内で完了するわけではなく、処理の進行に従って、あるパーティションから他のパーティションへと制御が移る。したがって、パーティション間での適切に制御されたバンクの共通のメカニズムが必要となる。

まず、アクセス制御の観点からバンクをながめると、

- 1) パブリック・バンク……不特定多数のパーティション間で共有されるもの
- 2) セミパブリック・バンク……リング・メカニズムにより、特定のパーティション間で共有されるもの
- 3) セミプライベート・バンク……アクセス・ロック・リストにより、特定のパーティション間で共有されるもの
- 4) プライベート・バンク……属するパーティション内でのみアクセスが許されるものの四つに分類される。これらの分類はバンク・アクセス可能性カテゴリと呼び、リンキング・システムの一部であるスタティック・リンカあるいはパーティション宣言で指定する。パブリック・バンクは、その GAP フィールドに適切なアクセスの種類を設定することにより実現する。

セミパブリック・バンクはリング・メカニズム、すなわちキーのリング番号がロックの

それより小さい場合に両者のドメイン番号が異なってもキーとロックが一致し、SAP フィールドに指定が可能であるというメカニズムを利用して共有されたバンクである。

セミプライベート・バンクは、共通対象のバンクへのアクセスを許可するパーティションのドメイン番号と、そのときのアクセスの種類を示す SAP の内容とからなるアクセス・ロック・リストを用いて共有するバンクである。

### 3.4 ゲート

プログラム制御の遷移にともないキーのドメイン番号の切り替えや、セミパブリック・バンクの共有のためにキーのリング番号の変更を必要とすることがある。これらキーの変更を可能にするのが、ゲート・バンクに作られるゲートである。ゲート・バンクのロック情報はゲートの作成や管理を行うパーティション(ゲート・マネージャ)だけが読み出し/書き込みできるよう設定されている。

決められたエントリ・ポイントからの進入を保証するため、命令バンクのその場所にゲートを設け、ゲートの経由は、ゲートの SAP と GAP の両フィールドに E(バンクへの進入)が設定されているか否かで決められる。ゲートのロックを特定のキーに合うよう設定し、(ゲートのロックのドメイン番号には呼び出したパーティションのドメイン番号が、リング番号には 0 を設定)、GAP フィールドに E を設定しないようにすると (SAP には E を設定)、特定のパーティションだけがそのゲートを經由できる。このゲートをプライベート・ゲートと呼ぶ。ゲートの GAP フィールドに E を設定すると (ロックは 0, SAP は E を設定)、パブリック・ゲートとなり、不特定多数のパーティションが經由できるようになる。

他のパーティションからの進入を許すエントリ・ポイント(グローバル・エントリ・ポイント)に対しては、常にゲートを作成して不正な呼び出しからパーティションを保護している。

ゲートには、この他に内部ゲートがある。これは、パーティション内部からのみ参照可能なローカル・エントリ・ポイントとして指定されるポイントでのゲートで、ロックのドメイン番号にはそのパーティションのドメイン番号を、リング番号には 0 を設定し、SAP =E, GAP= 0 と設定して作られる。

### 3.5 導入と置換

以上、メモリ保護とリンケージ・グループとしての面からソフトウェア・パーティションについてみてきたが、ここでは導入と置換について簡単に述べる。

あるパーティション内のデータを外部から参照するには、用意されたインタフェースを呼び出さなければならない。これにより、レジスタ経由あるいはバンク共有のメカニズムのもとでデータ参照が可能となる。したがって、ソフトウェア・パーティションの考え方を適用するには、ソフトウェアを機能別にモジュール化することが必要となる。

ひとたび、このモジュール化が実現すれば、ソフトウェアの導入に際しては必要とする機能に合わせてパーティションを導入すればよく、未導入の機能が新たに必要となったときにも対応するパーティションを単に追加すればよい。さらには、すでに導入されているパーティションをシステム稼働中に、更新されたパーティションで置換することも削除することも可能である。

以上のことが容易にかつ安全に行えることから、メモリ保護に加えて、IARMIS (Installability, Availability, Reliability, Maintainability, Integrity, Security) が一段と向上することを期待できる。



#### 4. おわりに

以上、アドレス空間の拡張とリング/ドメイン概念によるメモリ保護との二つの新しいアーキテクチャを中心に述べてきた。1100/90 シリーズに採用された最新のアーキテクチャは、利用者のニーズを満足させる将来のコンピュータ・システムのあり方を方向づけるものとなっている。

1100/90 シリーズは、このほか

- 1) 新しいハードウェア技術の開発と、その適用による大幅な価格性能比の向上（従来システムの1/8の大きさと3.5～4倍の高速性能を実現し、価格性能比は3倍以上になる）
- 2) 今後のコンピュータ利用者の中心となるエンド・ユーザに「使いやすさ」を提供するソフトウェア群

の二つの大きな特徴をもつが、これらについては、別の機会で報告されるものと思う。

#### 執筆者紹介 古沢純一 (Junichi Furusawa)

昭和20年生、43年武蔵大学経済学部工業経営学科卒業、同年日本ユニパック(株)入社、現在に至る。プロダクト・サポート統括二部1100/90導入プロジェクトでベーシック・ソフトウェアなどのサポートに従事。



#### 城川孝二 (Kouji Shirokawa)

昭和45年防衛大学校航空工学科卒業、同年、日本ユニパック(株)入社。以来デマンド・シンビオント、TIP、HIPOSの開発および保守、ベーシック・ソフトウェアの保守などを行う。現在、プロダクト・サポート統括二部1100/90導入プロジェクトに所属し、UNIVAC 1100/90シリーズの導入準備を担当。



#### 田中厚司 (Atsushi Tanaka)

昭和46年防衛大学校電気工学科卒業。同年、日本ユニパック(株)入社。以来、ベーシック・ソフトウェアを担当し、現在プロダクト・サポート統括二部1100/90導入プロジェクトに所属。



著者注記 本稿は、「日経コンピュータ」1981年10月4日号に著者らが寄稿した「最新の7つのアーキテクチャを取り込んだ UNIVAC 1100/90 シリーズ」を抜粋し、再編集したものである。

## 報告 MAPPER を利用した製造管理システム “TACTICS”

### TACTICS—Automated Manufacturing with MAPPER

L. F. Rogney

**要 約** TACTICS は、データ収集、および製造工程制御用のリアルタイム・システムで、MAPPER を使って構築された。このシステムの目的は、製造現場における品質と生産性の向上にある。このシステムは Sperry Univac 社 Roseville 工場の開発部門で開発されたもので、コンピュータ自動設計システム UCADS に始まり、製造工程を経て、出荷後の特定の使用サイトの監視に及ぶ、各段階における製造と検査の自動化を狙いとしていた。しかし今や、このシステムは、製造部門での日々のオペレーションに不可欠なものとなっている。

TACTICS は中央集中型の分散システムであり、個々のユーザ部門を支援する中央データベース・システムを備えるとともに、システムを分散し TACTICS の主要部がダウンしても、個々の組織が自分自身の自立型のハードウェア資源とソフトウェア資源を使って、その機能を果たし続けられるように設計されている。なお、この中央データベース・システムを運用している UNIVAC 1100/80 システムは通常のデータ収集に加えて、プログラムのロードや更新などを行うためのリンク機能を備えている。

TACTICS の実現は、MAPPER なくしては不可能であった。MAPPER は、エンド・ユーザがアプリケーション・プログラマの手を借りずにレポート（情報）の設計・作成・更新を行うための豊富なレポート処理機能を用意している。

**Abstract** TACTICS is a MAPPER-based real-time data gathering and process control system interconnected and uniquely designed for the specialized tasks in manufacturing with the objective of improving quality and productivity. This system, developed by Sperry Univac, Major Systems Division in Roseville, was designed to automate manufacturing and test beginning with input from UCADS thru the manufacturing process and monitoring of selected customer sites. TACTICS is an essential part of the manufacturing operation.

TACTICS is a centralized distributed system, that is; it provides the Central Data Base (System) necessary to support each unique user organization and it is distributed, in that each using organization can continue to function with its own free standing hardware and software resources if any major segment of TACTICS should fail. The Central Data Base System, an 1100/82, provides link capability, program loading and update in addition to data collection functions.

MAPPER is the prime element, that makes TACTICS possible and successful. MAPPER provides the user with a wide variety of report processing functions to design, produce and modify reports without need of an application programmer.

#### 1. はじめに

Sperry Univac 社の Roseville 工場におけるコンピュータの利用は製造工程の制御管理および検査などの各々の局面で活発に行われてきたが、その後これらの個々のシステムを有機的に結合するとともに、各種の試験装置や製造装置ともインタフェースを取りながらデータ収集や資源管理が自動的にできるような総合的システムが必要となり、開発された

のが、MAPPER を母体とした TACTICS である。

TACTICS は、MAPPER を使って構築されたデータ収集、および製造工程制御用にシステム間の結合も含めて独自の形で設計されたリアルタイム・システムであり、その目的は製造現場における品質と生産性の向上にある。TACTICS という名前は、Total Assurance and Control by Testing and Interactive Collection System の頭文字を採ったものである。

このシステムは、Sperry Univac 社 Roseville 工場の開発部門で開発されたもので、コンピュータ自動設計システム UCADS (Unified Computer Automated Design System) に始まり、製造工程を経て、出荷後の特定使用サイトの稼動状況監視とフィード・バックも含めて各段階における製造と検査の自動化を狙いとしていた。そして、現在ではこのようなシステムは製造部門での日々のオペレーションに不可欠なものとなっている。

TACTICS は、中央集中型の分散システムであり、個々のユーザ部門を支援する中央データベース・システムを備えるとともに、システムを分散し万一 TACTICS の主要部分がダウンしても個々の組織が自分自身の自立型のハードウェア資源とソフトウェア資源を使って、その機能を果たし続けられるように設計されている。なお、この中央データベース・システムを運用している UNIVAC 1100/80 シリーズ・システムは、通常のデータ収集に加えてプログラムのロードや更新を行うためのリンク機能を備えている。そして、このシステムは、1台の 2×2 システム (2台のインストラクション・プロセッサと2台の入出力プロセッサで構成される) としても、あるいは、状況によっては2台の 1×1 システムとしても利用可能である。

MAPPER は、アプリケーション・ジェネレータであり、これなくしては TACTICS の実現は不可能であった。MAPPER は、MAintaining Preparing and Producing Executive Reports の略称で、多種多様な報告書の設計・作成・修正をエンド・ユーザがアプリケーション・プログラムの手を借りないでも行えるようにするものである。エンド・ユーザは、簡単なキーインで MAPPER のラン (MAPPER のコマンドを組み合わせて処理手順を登録しておくもの) を実行することができる。また、ある種の付加装置を付けることにより、コンピュータの製造用装置、各種試験装置、検査中のコンピュータまで含めてレポート処理を拡大できる。

## 2. TACTICS のネットワーク

TACTICS は、6種類のアプリケーションを運用するコンピュータ・ネットワークであり、10の主要機能を支援する全部で16の独立なシステムで構成される (図1)。

10の主要機能とは、MRCC (Midwest Regional Computer Center), UCADS, 工場品質管理, サポート・エンジニアリング (support engineering, 製造部門にあって、開発段階以後の製品の技術的問題を解決するスタッフ機能), 製品評価・試験, 環境監視・制御, 製造支援, 評価計画, システムと品質の支援, 傾向分析である。なお、Roseville 工場には、約4,300人の従業員がいるが、製造部門の各部署にいる3,500人の従業員が TACTICS のユーザとして登録されている。

TACTICS には、およそ1700台の UNISOPE 端末装置と1000台のその他入出力端末装置に加えて、20種140台に及ぶ試験装置、検査装置も接続されている。これらの16システムは TACTICS のネットワークとして互いに結合されている。

また、TACTICS の中核をなすデータ収集システムは MRCC である。(訳注: MRCC は現在 MRS (Midwest Region Service) の略称で呼ばれている。)

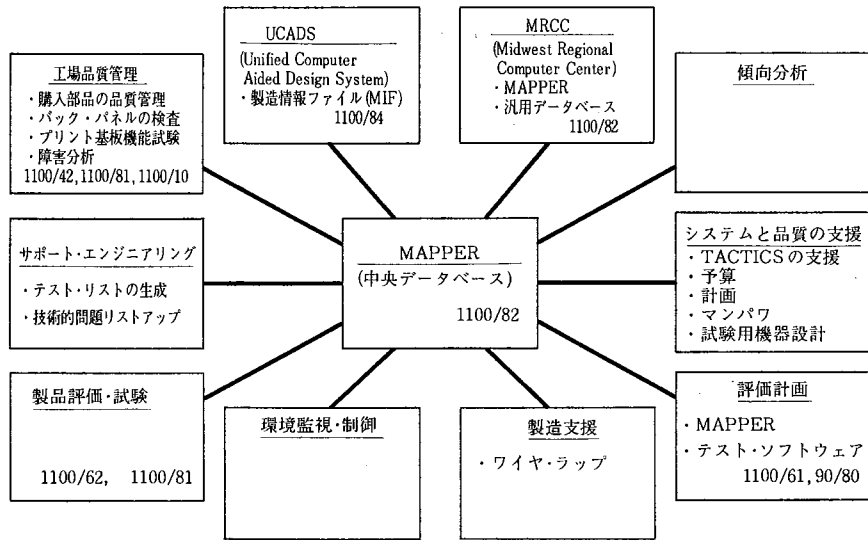


図 1 Roseville における TACTICS システム

Fig. 1 TACTICS at Roseville plant

MRCC は、ツウィン・シティズ地区 (Minneapolis および St. Paul を指す) をカバーするデータ処理システムで、UNIVAC 1100/82 を使用している。この MRCC システムをアクセスすることにより TACTICS のデータ収集および分析能力が拡張される。(訳注: MRCC のシステムを経由して UNIDATS と呼ばれる Sperry Univac 社全体のネットワークとも接続されている。)

なお、TACTICS は、ツウィン・シティズ地区の工場だけでなく、MRCC と専用回線で結ばれた四つの遠隔地の工場 (Canada Dorval の電源工場, Utah 州 Salt Lake City のプリント基板工場, Minnesota 州 Jackson プリント基板組み立て工場, Iowa 州 Clear Lake 工場) から専用回線で接続されており、このほか公衆電話回線を介して米国全土から、多数のユーザが TACTICS を利用している。

### 3. 自動設計

TACTICS における製造および試験用のデータベースは UCADS によって与えられる。UCADS そのものは、設計・開発・生産準備の支援システムであり、UCADS だけで1台の UNIVAC 1100/84 システムを専用している。UCADS は、TACTICS に対して製品の組み立て完了試験および品質管理に必要なデータベースを与える。UCADS の設計データの中で、TACTICS と関係をもつものに次のものがある。

- 1) 方程式(論理設計)ファイル……最も詳細なレベルの回路の記述であり、機能設計を構成する各回路の論理的相互接続について述べたもの。
- 2) ハードウェア記述ファイル……プロジェクトで使用される論理的ファミリの物理的・機能的・電子的特性 (たとえば、ワイヤリングの規則など) を記述したもの。
- 3) アルゴリズム・ファイル……レジスタ間のデータの移送や、主要な制御信号に関する高水準な記述を含むもので、論理設計の表現として用いられる。
- 4) 相互接続ファイル……多層基板、ワイヤ・ラップ、アートワーク、テスト・データ生成などの各モジュールで使用される主要な出力ファイル。

- 5) アートワーク記述ファイル……多層基板, あるいは多層バック・パネルの各層に関する一般的な特性を記述したもの.
- 6) タイトル・ブロック・ファイル……ロジック・ダイアグラム, 付記事項, 改訂の履歴などの各資料のタイトルをリスト・アップしたもの.

そして, これらのファイルは, MIF (Manufacturing Information File, 製造情報ファイル) としてまとめられ, 製造部門で使える形に変換され, ワイヤ・ラップ, 多層基板の検査, 再加工の指示や他の技術データとして利用される. そして, この MIF の情報の最新版がタイムリに得られることによって, 製造工程の作業が円滑に進められる.

このほか, TACTICS の機能の中で MIF を利用しているものにサポート・エンジニアリングがある. これは工場のテスト・リスト生成ソフトウェアの作成供給や, ときには再加工の指示を行う. なお, サポート・エンジニアリングは, MIF の系統履歴図を管理している.

#### 4. 会話型電子試験

TACTICS の主要機能の一つに部品とアセンブリの品質に責任をもつ工場品質管理があり, このために3台のコンピュータ, つまり UNIVAC 1100/42 システム (Roseville), UNIVAC 1100/81 システム (Clear Lake), UNIVAC 1100/10 システム (Jackson) が利用されている. そして, このシステムは, 図2の構成をもち, QUIET (Quality Unique Interactive Electronic Testing) と呼ばれ, 購入資材の品質管理, バック・パネルの検査, プリント基板の機能試験, 障害分析などを支援する.

QUIET システムの中心は UNIVAC 1100/42 システムであり, Roseville にあって電子部品の試験, サンプリングの自動計画, MAPPER を利用した種々のアプリケーション

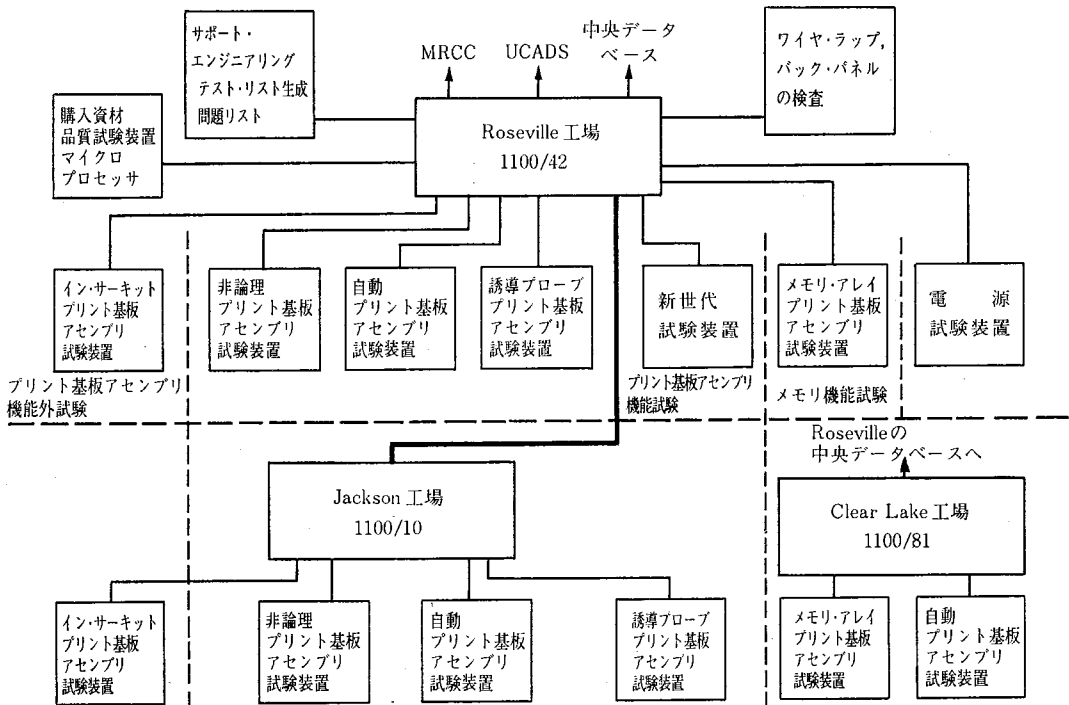


図2 QUIET システム

Fig. 2 Support engineering and factory quality

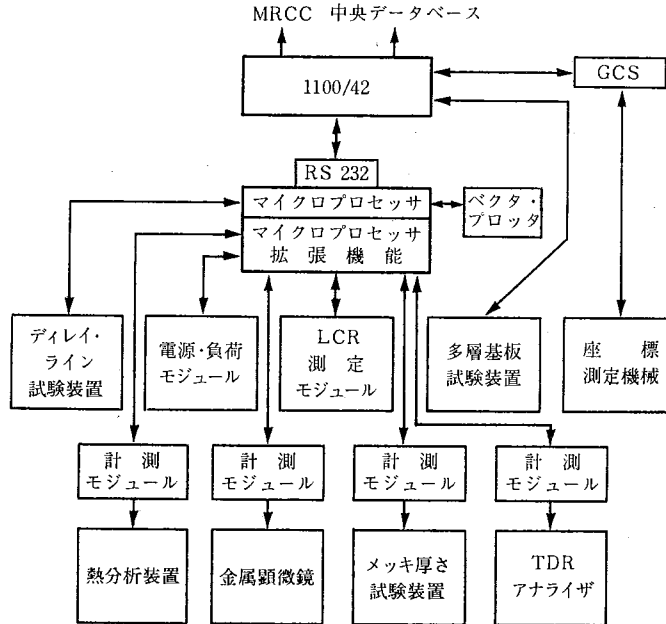


図 3 QUIET システム (購入部品の品質管理を中心として)

Fig. 3 Factory quality-PMQ

ンなどを通じて購入資材の品質管理の自動化を支援している (図 3)。

そして、専用の試験および品質検査システムがマイクロプロセッサを媒介にして QUIET システムに接続されている。なお、QUIET は、TACTICS の中央データベース、MRCC、UCADS システムとも接続している。

QUIET システムは、試験装置とのインターフェースを採るために開発された MAPPER のアプリケーション・プログラムを用いており、試験装置をコンピュータで制御することによって、より整合性のとれた正確な試験が可能である。また、測定とその結果の比較を自動化することによって人間の誤りが排除され、一方、テスト・ファイルを最新の版にすることによって正確な検査が保証される。そして、特定の部品あるいは特定の納入業者に偏って発生する問題などは自動的にリアルタイムで検出・分析されるので、製造工程の最も早い段階で解決できる。このほか、障害データが標準的なレポートあるいはグラフの形で表現できることも特長として挙げられる。

QUIET システムで使用されているマイクロプロセッサは、COLLATE (Control Log and Link Automatic Test Equipment) と呼ばれるもので、現在 65 キロ・バイトの記憶容量をもっているが、131 キロ・バイトまで拡張できる。COLLATE は、20 メガ・バイトのディスク記憶を有し、マルチプレクサを経由して、16 台の UNISCOPE が接続されている。そして、これらのディスプレイは、マイクロプロセッサの制御に使用したり、QUIET システムのディスプレイとしても利用できる。(訳注: COLLATE の制御をしているときは、MAPPER によく似たコマンドで受け入れ検査を行うが、このとき内部的には COLLATE から QUIET に対しテスト・リストの取り出しや試験成績の入力などが MAPPER コマンドをシミュレートする形で自動的に行われている。) なお、COLLATE システムは、IEEE データ・バスを介して 20 台までの試験装置が接続可能である。

また、QUIET システムによって、納入された各ロットに対するサンプリング計画が自

動的に生成され、コンピュータの画面から試験の段取りの指示やテスト・パラメタが検査係に与えられる。このほか、試験結果は表示後に自動的に TACTICS の中に記録され、前もって定められている試験基準値と比較される。そして、サンプルのすべての部品が検査されると、そのロット全体が受け入れられるかどうかが決められる。その後でワーク・ステーションのプリンタによって障害部品データのハード・コピーが採られ、そのロットの要約報告書が出力される。

なお、図4はロット番号 928945 におけるサンプリングによる試験結果をまとめた品質管理図である。この例では、130 個のサンプルのうちで1個だけが管理限界の外にあることを示している。

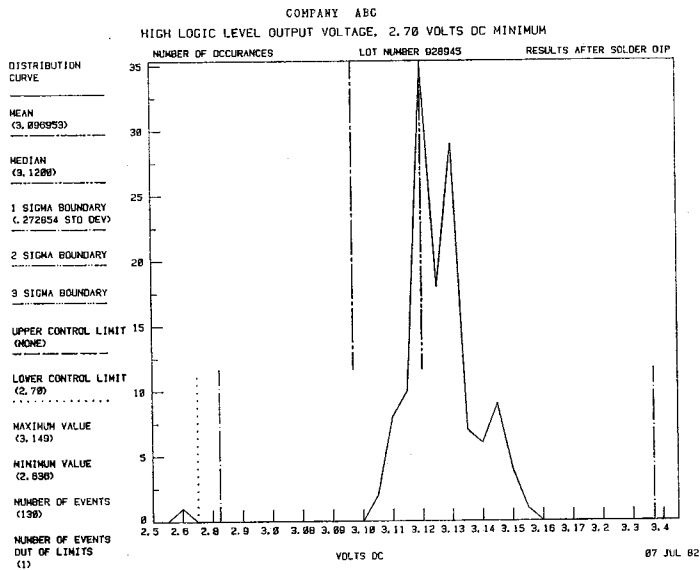


図4 購入部品の品質管理図 (PMQ)

Fig. 4 Purchased material quality graph of component

また、QUIET システムには、購入部品の試験を行う以下の装置がマイクロプロセッサを介して、あるいは直接接続されており、これらの装置はプログラムの格納、データの記録およびデータ操作の機能などに関してコンピュータの力を利用できる。

- 1) 座標測定機械……多層基板の寸法検査に用いられる。
- 2) 多層基板試験装置……多層基板の開放および短絡をチェックするのに用いられる。
- 3) デイレイ・ライン試験装置……部品レベルにおける能動および受動デイレイ・ライン、レジスタ・ネットワークをテストするのに用いられる。

このほか、現在開発中の試験装置として電源・負荷モジュール、LCR 測定モジュールがある。(訳注: 半導体部品については半導体事業部で担当しており、この中には触れていない。)

電源・負荷モジュールは、スイッチング・モジュールを介して接続されるソリッド・ステート負荷を使用している。この装置は部品の仕様で決められている負荷と電圧を自動的に加えるものである。

LCR 測定モジュールは、抵抗、静電容量、インダクタンス、誘電率を自動測定し、仕様と比較し、受け入れるかどうかを決定するのに用いられる。

このほか、次の機器が開発中であり、完成した場合には試験パラメタの決定、自動記録、データの傾向分析などに利用されることになっている。

- 1) 熱分析装置……主として多層板（ラミネート）のチェックに使用される。
- 2) 金属顕微鏡……メッキ厚さの測定や多層基板の内部の構成のチェックに用いられる。また、図3におけるメッキ厚さ試験装置は、主として金メッキされたコネクタやスイッチなどのメッキの厚さを測定するのに用いられる。
- 3) TDR アナライザ……TDR (Time Domain Reflectometer) アナライザは、多層プリント基板およびケーブルの回路インピーダンスを測定するのに用いられる。

購入部品の品質は、早期から完全な信頼性データを業者と与え、業者と協力し問題の解決を図ることによって改良される。また、テストの自動化によって生産性も向上し、受け入れから資材倉庫へ送られる時間が短縮するため在庫量が少なくなる。

TACTICS は、購入部品の品質の向上にとって不可欠なシステムといえよう。

## 5. プリント回路の試験

QUIET システムは、プリント基板の試験システムの中核としても働く。図2で示されるように、二つの主要なプリント基板試験設備が、Roseville と Jackson とにある。これらの設備のすべてのプリント基板試験装置は QUIET システムに接続され、通常の試験の運行監視や障害データの記録は QUIET システムによって行われる。また、最近の試験装置の中には、マイクロプロセッサと記憶装置を備えた独立型のももあり、その場合は QUIET システムとの通信は定期的に行う障害の記録とテスト・リスト・ファイルの更新に限られる。

さて、プリント基板は、まず、短絡・開放、レジスタ・パックの故障などの配線作業レベルでの誤りを抽出するため、イン・サーキット・テストでチェックされる。このテストは、マイクロプロセッサを内蔵しており、3600ポイントを測定できる装置である。たとえば、レジスタ・パックでは、このテストによる測定値に5パーセントの偏差がある場合は不合格となる。なお、基準プリント基板を用い自分自身のテスト・リストを生成できることも、このテストの特徴としてあげられる。

つぎに、機能レベルでの試験が行われる。ここで使われるテスト・リストは、UCADS からの情報をもとにサポート・エンジニアリングが QUIET システムによって生成したものである。このほか、テスト・リストは、試験装置自身が試験中の基板の応答と基準プリント基板の応答とを比較することによって生成されることもある。なお、機能試験では、電圧と温度のマージンもチェックされる。

また、メモリ・アレイは、0°C~75°C 間を周期的に変化するバーン・イン・チャンバー中でランダム・データ・パターンを用いて行われる。バック・パネルの検査装置は、マイクロプロセッサを介して QUIET システムに接続されている。なお、この検査装置は、試験対象のバック・パネルへのインタフェースの方法として、いわゆる“針の山 (bed of nails)” の方式を採用している。このほか、ワイヤ・ラップ機械もこの検査装置と同様な方法で QUIET システムに接続されている。これらの装置は、自立性の高いインテリジェント・システムであり、中央側でのデータ・ファイルの更新の場合以外は中央側コンピュータの支援が不要である。Roseville, Jackson, Clear Lake におけるこれらの試験機能は、データ収集、障害記録、傾向分析用のデータベースの提供を行う分散型システム TACTICS のすべてであるといってもよいくらいである。なお、試験装置の概念設計とその思想の開発は、新



製品の設計と並行して製品開発部門によって行われる。

### 6. 会話型診断システム

TACTICS の他の主要な機能に製品の評価・試験システムがある。このシステムは、5台の UNIVAC 1100/62 システムと3台の UNIVAC 1100/81 システムを用い、コンピュータおよびメモリの試験を支援するもので、MIDAS (MAPPER Interactive Diagnostics And Support) と呼ばれている (図5)。ここでも、MAPPER のアプリケーション・ソフトウェアが、特殊な試験や機能を提供するために用いられている。

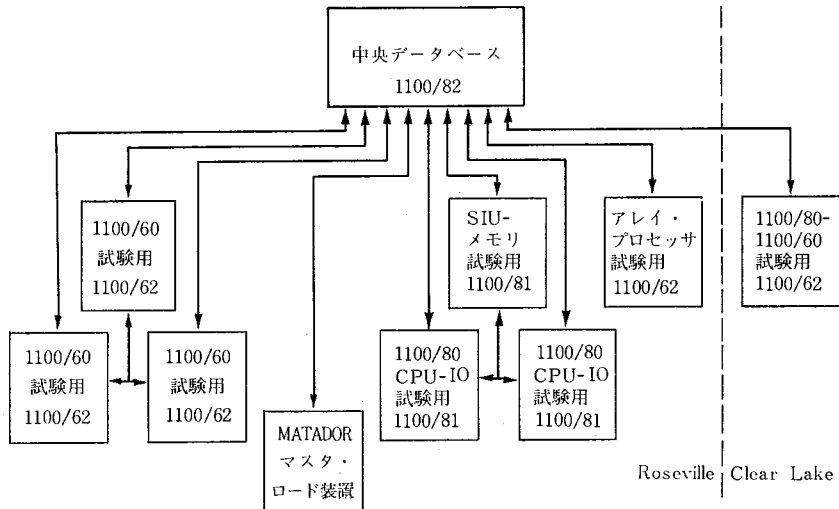


図 5 MIDAS システム

Fig. 5 Product evaluation and test

Roseville と Clear Lake における製品の評価および試験部門は、試験のほとんどすべての段階で MIDAS を利用している。なお、この試験は、電源投入前のチェックに始まり、プリント基板の挿入、診断および機能試験、ユニットの最終的受け入れ試験にも及んでいる。そして、UNIVAC 1100/60 システムと 1100/80 システムは、MIDAS システムを使って試験される。MIDAS の使用効果としては、従来の方法に比して試験時間を半減できること、経験に乏しい人でも試験できること、システム相互間でリアルタイムにデータを共有できること、保守プロセッサ・インタフェースを介して試験中の装置と接続できることなどがあげられる。

そして、試験はスキャン・セット、つまり基準ユニットで生成されたファイルの内容と試験中のユニットの応答とを比較することで行われる。また、ユニット間のオンライン試験は、マスタ・ロード・システム(ディスク装置)から入力された MATADOR テスト・ソフトウェア (MATE AnD wORKers の略) によって実施される。なお、レビジョン管理と必要なデータの分配は、TACTICS の中央側システムによって行われる。また、これらの試験におけるチェック・アウトの手順は MAPPER ランによって制御され、障害が発生するか人間の介入が必要になるまでは自動的に繰り返し行われる。これらの試験システムの運転操作は簡単で容易に修得できる。また、障害が発生すると、学習ファイルのデータにもとづき、障害の原因、信号線の名前、論理回路の種別、論理図のページ番号、チェック・アウト手順、発生した問題に対する過去の解法などが表示されるので、試験担当者は先に

進む前に問題を解決しなければならない。試験担当者は、問題を解決した後に、その手順におけるテスト・ログ部分にデータをキーインすることにより、チェック・アウトは続行される。さて、MIDAS システムは、チェック・アウト手順における重要なマイル・ストーンを達成した時点でその項目を自動的に記録するが、障害の記録の投入については試験担当者のイニシャルのキーインを必要とする。そして、この記録が状況報告の作成や傾向分析のデータとなる。MIDAS システムによって障害誤動作報告書 (FMR, Failure Malfunction Report, 図 6) および障害部品表示ステッカが自動的に作成される。つまり、障害の発生時に MIDAS の要請にもとづき試験担当者の投入したデータが、状況報告、障害誤動

```

.DATE 29 OCT 82 18:27:27 REPORT GENERATION ROOPLY

*****
FMR ASSEMBLY ANALYSIS
FOR P/N 4234567
FROM 820726 THRU 824024
*****

RSUL/CLR LK MANUFACTURING FACILITY

***** FOR PERIOD STARTING: 820726 AND ENDING: 824024 *****
*PART NUMR .TYPE/FEAT .TSTQTY.FMRQTY.% DEF.TLS.TLX. U/C.DESCRPTION
=====
4234567 F222200 454 143 74.8 400 429 PC ASSEMBLY--- UNIT TYPE
=====

*****
FMR ASSEMBLY SUMMARY
FOR WORKMANSHIP
DEFECTS FROM 820726 THRU 824024
*****

          QTY          % OF TOTAL      DESCRIPTION OF DEFECT
          =====
          24          41.6          WRONG COMPONENT
          40          4.9          DAMAGED CERS OR OVRVOLTAGE OR OVRCCURR
          -----
TOTAL =          34

*****
FMR ASSEMBLY SUMMARY
OF COMPONENTS BY P/N
FROM 820726 TO 824024
*****

          COMPONENT P/N          TOTAL
          =====
          3333333          146
          4444444          5
          2222222          2
          8888888          1
          7777777          1
          -----
TOTAL =          155

COMPONENT VNO VEND FMR DESCRIPTION
PART NO  CDE D/C REP  OF FAILURE
=====
4444444 MEP 8245 1 BAD/WRONG/INCORRECT LOGIC LEVEL
5555555 FAI 8144 1 BAD/WRONG/INCORRECT LOGIC LEVEL
6666666 FAI 8452 1 BAD/WRONG/INCORRECT LOGIC LEVEL
7777777 FAI 8243 1 BAD/WRONG/INCORRECT LOGIC LEVEL
8888888 UND 8439 1 VOLTAGE MARGINS
4444444 UND 8448 1 BAD/WRONG/INCORRECT LOGIC LEVEL
4444444 UND 8203 1 BAD/WRONG/INCORRECT LOGIC LEVEL
3333333 UND 8245 1 VOLTAGE MARGINS
5555555 MPT 8247 1 VOLTAGE MARGINS
6666666 MPT 8446 1 UNKNOWN- (ANALYSIS INCOMPLETE)
8888888 UND 8444 1 BAD/WRONG/INCORRECT LOGIC LEVEL
8888888 UND 8447 1 VOLTAGE MARGINS
8888888 UND 8448 2 VOLTAGE MARGINS
8888888 MPT 8454 1 VOLTAGE MARGINS
8888888 MPT 8242 20 VOLTAGE MARGINS
9999999 UND 8243 6 UNKNOWN- (ANALYSIS INCOMPLETE)
9999999 UND 8245 35 VOLTAGE MARGINS
5555555 MTK 8028 1 VOLTAGE MARGINS
6666666 MTK 8033 3 BAD/WRONG/INCORRECT LOGIC LEVEL
4444444 MTK 8034 1 VOLTAGE MARGINS
4444444 MTK 8204 1 VOLTAGE MARGINS
4444444 UND 8004 1 VOLTAGE MARGINS
4444444 UND 8052 1 BAD/WRONG/INCORRECT LOGIC LEVEL
4444444 UND 8204 3 VOLTAGE MARGINS (VOLTAGE COMPENSATION PROBLEM)
4444444 UND 8205 1 BAD/WRONG/INCORRECT LOGIC LEVEL
4444444 UND 8204 6 BAD/WRONG/INCORRECT LOGIC LEVEL
4444444 UND 7949 1 VOLTAGE MARGINS
-----
TOTAL =          155
..... END REPORT .....

```

図 6 アセンブリの障害誤動作報告書 (FMR) の例

Fig. 6 Example of failure malfunction report for an assembly

作報告, 学習ファイル・データベースなどへの入力として用いられる。

MIDAS は, 学習ファイル・データによる障害説明時間の節約および現在の状況報告のマネジメントへの提供を可能にするリアルタイム・システムである。製品の品質は MIDAS システムを用い, より完全かつ正確に試験することによって改良される。また, 製品の評価・試験における生産性は試験の順番を自動的に決定したり, 学習ファイルを利用することによって向上できる。MIDAS は, 単一入力方式による試験結果の記録, 状況データの投入での重複や障害を減らすことのできるペーパーレスなシステムである。

傾向分析は, TACTICS のもう一つの重要な機能である。TACTICS システムのデータ収集能力が向上するにつれて, データの加工と分析がシステムの有効利用とユーザの便益にとってより重要であるとの認識が生まれた。工場品質障害分析グループは, MAPPER ランを使ってデータベースをスキャンして, 部品・アセンブリ・ユニットの各レベルで通常の障害発生率よりも高い障害発生率を示す異常な傾向を発見する。また, これらの MAPPER ランは, すべての試験・品質管理従事者も利用できるようになっており, 障害部品の早期発見を支援している。また, 障害の傾向分析は, 工場品質障害分析グループが障害分析を行う個所を絞るのに有効である。なお, 同グループは高度な機械を利用しており, 先端的技术に潜む問題を把握するのに効果を挙げている。

さて, 前述のように, TACTICS システムは, MAPPER があって初めて実現したものである。当初, 評価計画部門によって開発された MAPPER システムは, TACTICS を経営情報システムとして利用しているすべての部門の主要なツールとなっている。MAPPER は, 急速に普及しつつある Sperry Univac 社の製品であり, 209 社の顧客および Sperry 社の 35 の部門で稼動中である。(訳注: 今日, それぞれ 500 社, 50 部門を越えている。)

MAPPER は, セキュリティ機能によって保護された各部門独自のレポート構造データベースを提供するだけでなく, MAPPER ランを高度に利用することによって, 業務を単純化し迅速化できる。さらに, MAPPER ランの設計でメニュー選択方式を採用すれば, MAPPER の使用法やドキュメントの構造についてほとんど知識をもっていないユーザでも抵抗なく作業が進められる。

## 7. 資源計画

TACTICS の主要機能に「システムと品質の支援」があり, 同名のグループがエンド・ユ

```

***** DEPRESS F1 OR ENTER 'RSM' WHEN REPORT IS UPDATED *****
.(PEN COLORS - BLACK, BLUE, RED, GREEN) SPERRY LOGO? Y (Y OR N)
.STATION 554 (556 OR 557) INCLUDE LEADING 'S'? N (Y OR N)
.HEADER 1( MATADOR PROGRAM DISK TRACK USAGE
.HEADER 2( PERCENTAGE BY PROCESSOR TYPE
)
)

```

NAME	VALUE	EXPLODE?	PEN COLOR	SHADE?
418III	19P		RED	Y
11P8	192		BLUE	Y
110P/4P	2P4		BLUE	
110P/6P	91		BLUE	Y
110P/8P	351	Y	RED	Y
494DP	117		BLACK	Y
9P/8P	2P2	Y	GREEN	
9P/7P	111		GREEN	Y
C/SP	6P	N	GREEN	
376P	8P		BLUE	Y
DCP	59		BLUE	

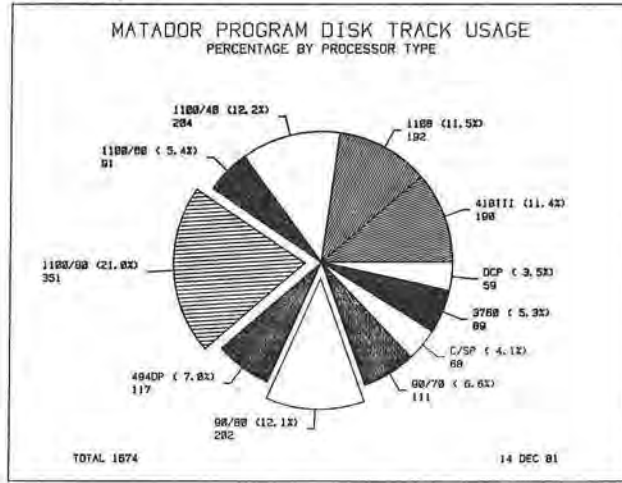
..... END REPORT .....

(a) 円グラフ用パラメタ

図 7 MAPPER 使用例 1

Fig. 7 Example of MAPPER run

SPERRY+UNIVAC



(b) 実行結果

図 7 MAPPER 使用例 1

Fig. 7 Example of MAPPER run

一ザの在庫管理・訓練・予算・資源計画・検査計画・状況報告などへの TACTICS の利用を支援している。このため、これらの業務の報告書にスケジュールの変更を反映させる場合、従来ならば人手によって数日間要したものが、即座に反映できるようになった。また、図形出力は、中央データベース・システムのプロッタを使って作成される。MAPPER ランを利用するとパラメタをセットするだけで、各種グラフやヒストグラムの作成がプロッタを使って容易にできる (図 7)。

そして、マイル・ストーン・チャートの作成・更新も容易であり、常に最新の状況をフ

```

***** DEPRESS F1 OR ENTER 'RSM' WHEN REPORT IS UPDATED *****
.STATION 556 (556 OR 557)          SPERRY LOGO? Y (Y OR N)
*
*
*TEXT                               START   END     INTERIM
                                DATE    DATE   DATE
-----
.TITLE UNIVAC SUPER PROCESSOR
.SUBTITLE PRODUCTION SCHEDULE
.TITLECOLOR GREEN
.SURCOLOR BLACK
.QUARTERS
.DATES
.CHARTDATE R11214
.LINECOLOR GREEN
.TEXTCOLOR BLUE
BUILD LEAD MACHINE                R10501  R20301
TEST LEAD MACHINE                 R10501  R10801
*PRE-POWER TESTS                 R10501  R10515
*ON-LINE TESTS                   R10515  R10615
*ACCEPTANCE TESTS                R10615  R10801
.LINECOLOR RED
.TEXTCOLOR RED
FIRST LPS                          R10915
.LINECOLOR GREEN
.TEXTCOLOR BLUE
UNIT TEST                          R10915  R11115
SYSTEM TEST                        R11115  R20301  R11215
PCD                                R20301
.INITIALS MJG
.PAGEID 67H

```

..... END REPORT .....

(a) マイル・ストーン・チャート用パラメタ

図 8 MAPPER 使用例 2

Fig. 8 Example of MAPPER run



フォローすることが可能である (図 8)。

さらに、TACTICS では、カラー・プロッタによるオーバーヘッド・プロジェクタ用のトランスペアレンシの作成なども行われている (図 9)。

このほか、クラス・ルーム方式の訓練や説明会などにおいてプレゼンテーション効果を高めるために、画像投影装置が TACTICS に接続されている。

### 8. 環境監視

TACTICS における他の重要な機能に、環境監視・制御がある。静電放電は、IC チップの障害・誤動作の主な原因とされており、Roseville の開発部門ではその影響を小さくするために、導電タイルの床、非帯電性上着、個人用のアース・ストラップの採用や、湿

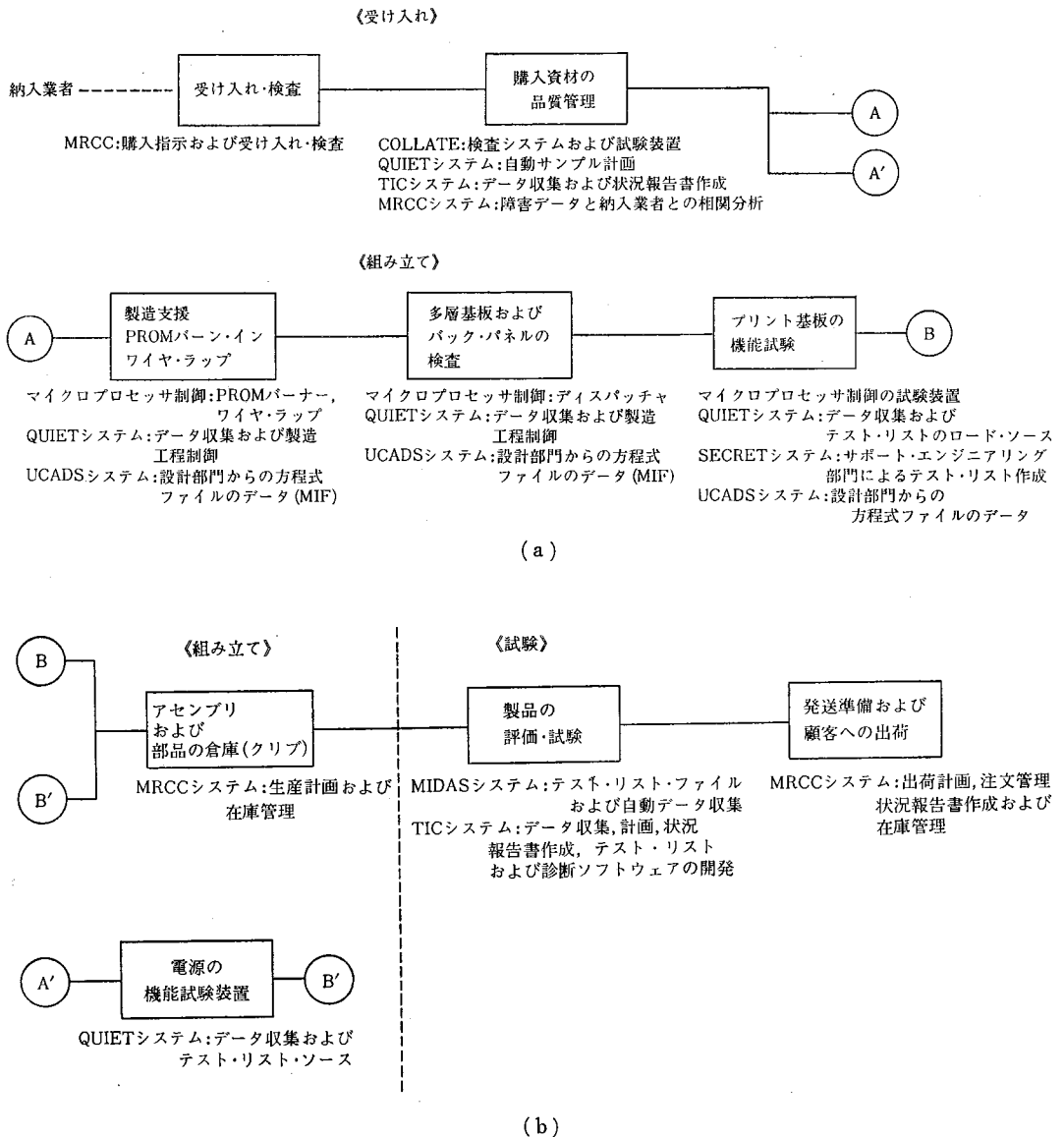


図 10 TACTICS プロセスの流れ  
 Fig. 10 TACTICS process flow

度制御の向上などによって防止に努めてきた。

これらの対策によって、従来、原因不明として処理されていた障害の数が大幅に減少した。このシステムは特別な制御領域を継続して監視し、ハードウェアの障害データと相関をもつ温度および湿度を記録するもので、もし、この制御領域における温度および湿度が設定値を越えると、空調関係の技術者に自動的に知らされる。

## 9. プロセスの流れ

以上で述べてきた TACTICS の主要機能を 購入部品の受け入れ・組み立て・試験・出荷の順に位置付け、そのプロセスをまとめると図 10 (a, b) のようになる。

## 10. 今後の開発計画

TACTICS に関し計画されている将来のプロジェクトとしては、前述の新規試験装置およびインタフェースのほかに次のものがある。

- 1) 光学式読取装置 (OCR)……OCR は、IC チップのサンプル・データを部品番号、外注業者名、外注業者コードによって決定したり、配線レベルでの検査を行うのに用いられる。
- 2) 音声認識の応用……現在開発中の音声認識装置は、まず購入部品の入荷処理・受け入れ検査時や、試験装置の操作時でのデータ投入を迅速化するのに用いられる。
- 3) 環境監視・制御……環境監視だけに止まることなく、自動環境制御へと移行する。なかでも、温度・湿度や各種の照明・電源回路の制御は重要であり、エネルギー効率の向上にもなる。

(訳注：本稿は 1981 年春のもので、現在ではさまざまな局面で新しい技術が応用されており TACTICS 自体も進化しつづけている。)

## 11. おわりに

TACTICS は、コンピュータ製造に関する業務全般を対象としたリアルタイムの総合的データ収集および製造工程制御システムであり、エンド・ユーザを指向しており、そのフレーム・ワークとして MAPPER を利用している。このため、TACTICS を用いると、一つのサブシステムから他のサブシステムのデータを必要に応じて利用できる。TACTICS は、品質と生産性の向上を実現する不可欠なシステムであるといえよう。

(プロダクト・サポート統括一部 企画推進室 横山 正敏 訳)

### 執筆者紹介 Larry F. Rogne

ハードウェア・プロダクト製造部門の品質管理およびシステム・サポート・グループに所属し、TACTICS の担当者である。Sperry Univac 社に入社以来 20 年近く 周辺機器および UNIVAC 1100 コンピュータ・システムの評価・試験関係に従事。



## 報告 ASHLAR における自動レイアウト

### —VLSI の“ジェネラル・セル”

#### レイアウトの諸問題へのアプローチ—

### Automated Layout in ASHLAR: An Approach to the Problems of “General Cell” Layout for VLSI

J. E. Hassett

**要約** 現在, Sperry Univac 社 Defense Systems Division で開発中の ASHLAR プログラムの自動レイアウト機能を解説する. ASHLAR は, 任意寸法のセルをもつ階層的 VLSI 設計のための会話型レイアウト・システムである. 配置, 電源バス配線, 多結晶シリコン内部結線など諸問題への対応に特別な注意を払っている. 最後にプロトタイプの実行結果も示す.

**Abstract** The automated layout facilities of the ASHLAR program, currently under development at the Defense Systems Division of Sperry Univac, are described. ASHLAR is an interactive layout system to be used in developing hierarchical VLSI designs with cells having arbitrary dimensions. Treatment of the problems of placement, power bus routing, and polysilicon interconnect usage are given particular attention. Finally, some results of a prototype implementation are presented.

#### 1. はじめに

ASHLAR (石積み) は, 石工が矩形の石を積み上げることに因んだ語である. 石積みには, 図1に示すように明瞭な二つのスタイルがある. coursed ashlar (列状石積み) は列あるいは層上に石が積まれ, 一つの層上では, ほぼ同じ高さである. random range ashlar (任意石積み) はでたらめの, あるいは規則性のない石積みとして知られ, 列上でなく近接適合配置で石が積まれる. 石工のこの二つの石積みスタイルは, 定義ずみの矩形セルを使う自動 IC レイアウトの二つのアプローチと強い類似がある.

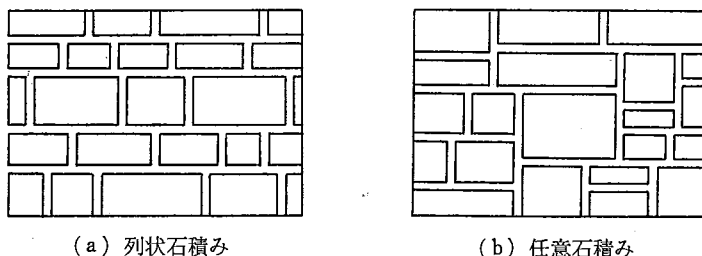


図1 石積み工の型

Fig. 1 Styles of ashlar stonemasonry

図2(a)に示される列上に置かれたセルのレイアウトは, しばしばポリセル・レイアウトとして知られるものである. 図2(b)にみられる任意寸法の矩形セルのレイアウトはビルディング・ブロックあるいは“ジェネラル・セル”レイアウトと呼ばれてきた. 本稿で

© 1982 IEEE. Reprinted, with permission, from ACM/IEEE 19th Design Automation Conference Proceedings (June 14-16), 1982, pp. 777-784.

© 1982 ACM. Reprinted, with permission, from ACM/IEEE 19th Design Automation Conference Proceedings (June 14-16), 1982, pp. 777-784.



はこれらレイアウトの形を区別するために、列状配置および任意配置という語を石工から借り受ける。なぜならば、両者の差異はセル自体の特性によるのではなく、セルの配置にもとづくものであるから。

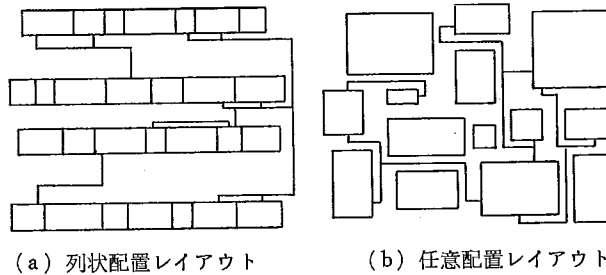
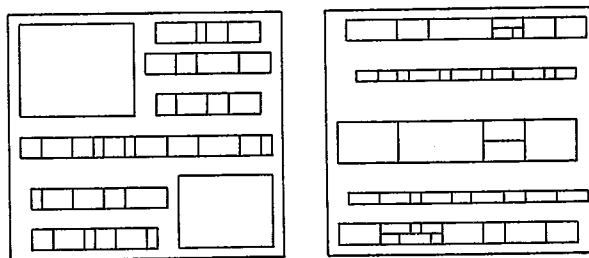


図 2 主配線を含む IC レイアウトの型

Fig. 2 Styles of cellular IC layout with representative routes

列状配置は、レイアウト MP 2 D<sup>[6]</sup>, LTX<sup>[14]</sup> や CALMOS<sup>[1]</sup> など数多くのレイアウト・システムにうまく使われてきた。これら多くのシステムではすべてのセルは同一の高さである。このことはセルの複雑さ (cell complexity) の幅に制限を加えることになる。なぜなら、簡単なセルに適した高さを選ぶと複合セルのレイアウトは幅広くなり、水平方向の内部結線の問題を生じる。このセルの複雑さの制限は、単一の配置設計で使えるセルの数に実質上の制限を加えるのと合わせて、チップ全体の複雑さにも制限を課することになる。

異なる列上には異なるセル・サイズを許容したり<sup>[17]</sup>、チップの端にいくつかの大きなセルを置く<sup>[6]</sup>などの基本列レイアウト法の拡張によって、問題の複雑さを幾分緩和するものの、容易ならぬ配置上の制約を生み出すことになる。



(a) 大型セル

(b) 異なる高さのセル

図 3 列状配置レイアウトの変種

Fig. 3 Coursed layout variations

任意配置レイアウトは任意寸法のセルを制約なく配置させるため、自動レイアウトのもう一つのアプローチとして、ここ数年の間議論されてきた<sup>[2], [9], [10], [12], [15], [16], [19], [21]</sup>。この方法はチップ・レイアウトのみならずセル・レイアウトにも適用でき、階層的なアプローチを可能にする (図 4)。

レイアウトごとに、きわめて限られた数のセルで、階層的レイアウトは相当な複雑さまで構成できるが、この方法を採用して作りあげられたチップはあるにしてもその数は少ない。成功例が少ないのは、この方法の採用に当たって生じたいくつかの問題に起因する。

一つの重要な問題は配置作業である。相互のセル寸法の異なるコンポーネントを互いによく組み合わせたいうで、短い配線距離を得るような妥当なセル配置を生む作業は複雑なものになる。より規則的な大きさのセルのレイアウトに適用される多くの配置アルゴリ

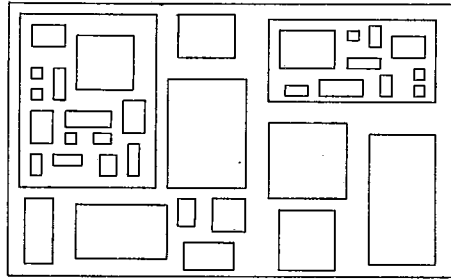


図 4 階層的レイアウト（内部レイアウト付きの二つのセルも示す）

Fig. 4 Hierarchical random range layout, showing the internal layout of two cells

ズムは、コンポーネントの交換によるものであったが、任意配置レイアウトではセルの寸法と形状の違いのために交換が不可能となる。

任意配置に関して、いくつかの配置アルゴリズムが発表されてきた。いくつかの論文<sup>[7], [9], [10], [16], [21]</sup>が、それらを比較したうえでの効果について一致した意見はない。いくつかの任意配置システムは、人手配置によるか、会話形配置機能を採用して配置問題の解決には人間の洞察力を利用するかによっている<sup>[2], [9]</sup>。

任意配置におけるもう一つの重要な問題は、電源線の分布である。列状配置では、セル同士が互いに接して配置されたとき、すべての列全体に電源が供給されるように、セル群は一定の位置に電源バス・セグメントを作り付けた形で設計される（図 5）。

任意配置では、そのような簡単なスキームは使えない。電源接続は、他のセグメントと同様な配線によってなされなければならないが、直流および過渡電圧降下やメタル・マイグレーションによる問題を避けるため、電源線の線幅を十分広くしなければならない。電源バス全体をとおしてその最適な配線幅は変化し、種々のコンポーネントへバス分岐が出るにつれて幅は狭くなる。適切なバス幅の決定と種々の幅をもつバスを配線することは、注意を要する重要な問題である。

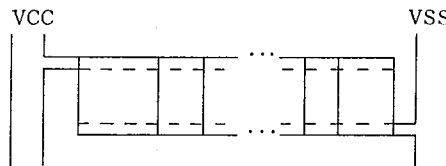


図 5 列状配置レイアウトの電源バス

Fig. 5 Coursed layout power bus

任意配置で出会う最も重要な問題は、おそらくメタルによる配線が単層でしか許されず、その結果、配線の大部分に高抵抗多結晶シリコンを使わざるをえない場合に生じる。列状配置レイアウトでは、主として配線チャンネルとの比較的短い配線に多結晶シリコンが使われる。しかし、任意配置レイアウトでは、垂直および水平両方向のほとんどすべての配線セグメントに使われる。

このことは、結果としてパフォーマンスを、とくに電源バスのパフォーマンスを低下させる。この問題の簡単な解決策は当然メタル 2 層配線を採用することであるが、2 層配線ができないとき、多結晶シリコンの使用を最小にするためのなんらかの方法を見つけなければならない。

特定の任意配線手法においては他にもさまざまな問題が生じるが、前記の三つ、すなわち、配置、電源線分布、そして多結晶シリコン使用の問題は、最も一般的かつ困難な問題である。

## 2. ASHLAR

ASHLAR (Automated System for Hierarchical Layout with Arbitrary Rectangles; 任意矩形セルを使った階層的自動レイアウト・システム) は Sperry Univac 社の Defense Systems Division で開発中の LSI の任意配置配線のシステムである。

このシステムは、ゲート・アレイおよび列状配置レイアウト (ポリセル・レイアウト) のために設計された他の総合回路レイアウト・プログラム (CGAL, SCROUL) を補足することを意図する。ASHLAR は、人間の洞察力とアルゴリズムの力を結合するカラー・グラフィック・ターミナルによる会話形のプログラム形式をとっているものの、基本的には完全自動レイアウトが可能である。

ASHLAR の自動レイアウト機能の主たる特徴を以下に述べる。ただし、ユーザ・インタフェースと会話形レイアウトに関する詳細な議論は本稿の範囲ではない。

## 3. 階層的レイアウト・セル・ライブラリ

階層的レイアウト・セル・ライブラリは、使用できるセル群に関する諸情報入力源であるばかりでなく、次のより高レベルのレイアウトで使われるセルの (内部) レイアウトを含むシステム出力先でもある。セル・ライブラリはセル・レイアウトのダイナミック・データベースである。

ASHLAR セル・ライブラリでは、セル定義の概念は単なるレイアウト寸法と端子の記述以上の内部を含んでいる。すなわち、レイアウトが決まる前のレイアウトすべき回路記述、すでに配置済みのさまざまな寸法のセルの集合、レイアウト規則、端子位置、配置・配線の詳細などを含む。そのような個々のレイアウトは、セル定義で記述された回路を正確に物理的に実現したものである。

回路記述は回路の外部端子の仕様、回路コンポーネントのリスト (各コンポーネントは他のセルの場合もある)、および内部結線の記述などを含んでいる。これは厳密に論理的記述であり、物理的特性を含まない。

もっとも、一般的なレイアウト記述はレイアウトが施される際のレイアウト規則、レイアウトの全体寸法、入出力端子の仕様 (セル端子に至るポート) および内部レイアウトの詳細な記述である。

実際の問題としては、セルの定義はその目的と開発の各段階に応じて逐次変更されていく。基本セル (primitive cell) は階層木の葉として現われるセルであり、他のセルから作られることはなく、その内部のレイアウトは他の手段によってあらかじめ用意されているものとする。基本セルの回路記述はセルの外部端子仕様だけである。複合セル (composite cell) は他のより簡単なセル群から作り上げられるセルで、完全な回路記述を含むけれども、前もってレイアウトは定義されていない。レイアウト情報は ASHLAR によって設計されて付加される。使用者は部分回路について順次レイアウトと配線を完成し、その結果を保存していくので、複合セル全体としてのレイアウトの内容は変化していく。そのような部分的に作られたレイアウトは、全体寸法とか端子仕様などの情報に欠けることになる。

他の種類の特別セル (special cell) も可能である。非機能セル (nonfunctional cell) は電

氣的機能をもたず、テキストやアラインメント用の目印などのマスクの特徴を表わすものである。ここでは回路記述はなく、レイアウト記述はレイアウト規則、全体寸法、内部アート・ワークなどが含まれる。他の特別セルとして stab セルがあり、会話型のチップ設計をする場合に使うことができる。ここでは、設計すべき機能、回路記述とダミー・レイアウトのみが与えられる。ダミー・レイアウトは仮想寸法と端子データのみからなり、内部レイアウトは与えられていない。完全なレイアウトは、より高レベルのレイアウトを設計する際にセル形状や端子の影響を見積もった後、ユーザによって完成される。

ASHLAR のセル記述に関するこの汎用性と柔軟性は、ASHLAR の多才な機能にとってきわめて重要なものである。セル・ライブラリはレイアウトの構成管理のうえで重要であるが、自動レイアウト技術によって数分のうちに新しいレイアウトが作り出されることが可能になるにつれて、階層的なシステムの上でますます重要な役割を果たすことになる。

#### 4. レイアウト・プロセス

ASHLAR のレイアウト・プロセスは、複合セルの回路記述を読み出すことと、あわせてコンポーネント・セルについて利用可能なレイアウトの記述から始まる。基本作業は、定義した回路を実現するためにコンポーネント・レイアウトを並べ内部結線することであるが、この仕事の完結までにはさまざまな制約が顕在化するに違いない。

レイアウト規則は、最小配線幅や各層での配線間隔などの物理的制約と遅延、電圧降下、電流制限値の計算に必要な配線層の電気的特性を定義している。

使用者が指定する制約には、クリティカル・ネットの指定コンポーネント選択の限定とその配置方向を限定すること、望ましい外形比（幅と高さの比）の指定（他のさまざまな制約を与える）などがある。当然、レイアウト・プロセスは、レイアウト面積、配線長およびスルホールの数を最小にすることを目指さなければならない。

Preas の論文<sup>[15]</sup>に従って、われわれはレイアウト・プロセスを三つの異なるフェーズに分解する。

- 1) 配置 (placement)……コンポーネントの概略配置をみつける。配線長を最小に、余空間を最小にすべくコンポーネントを互いにすり合わせる（ただし、各コンポーネント端に接続するネット数から、およそ概算できる配線空間を確保する）。そして、望みの外形比に近い配置を決定する。ここでの配置は近似的なものである。なぜなら、次の配線設計過程において、すべての配線を達成できるように必要十分なチャンネル幅（セル間の配線領域）が調節されるからである。
- 2) 大局的配線 (global routing)……各ネットごとに概略経路が使うべきチャンネル単位で見出される。基本目標は配線距離の最小化である。
- 3) トラック配線 (track routing)……各チャンネルに割り当てられたすべてのネットの詳細な配線が決められる。すべての経路を達成するようにチャンネル幅を調節しながら、トラック配線は各チャンネルごとに、そこに割り当てられたすべての配線について一度に実行される。必要なチャンネル幅にもとづいて、最終的コンポーネント位置がこのフェーズの終わりに決定される。

この方法、すなわち、配線は概略配置にもとづき、最終配置は配線結果によることは、完全な配線（100 パーセント配線）を保証するのに必要である。

概略配置はチャンネル位置グラフ<sup>[10], [16]</sup>を使って表現されるが、Szepieniec ら<sup>[21]</sup>と類似の木構造（コンポーネント間の接線関係を調べ構造を木で表現する）は配置とトラック配線

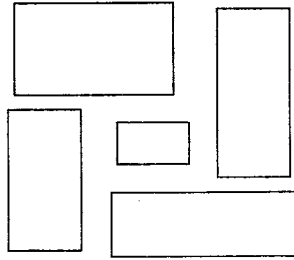
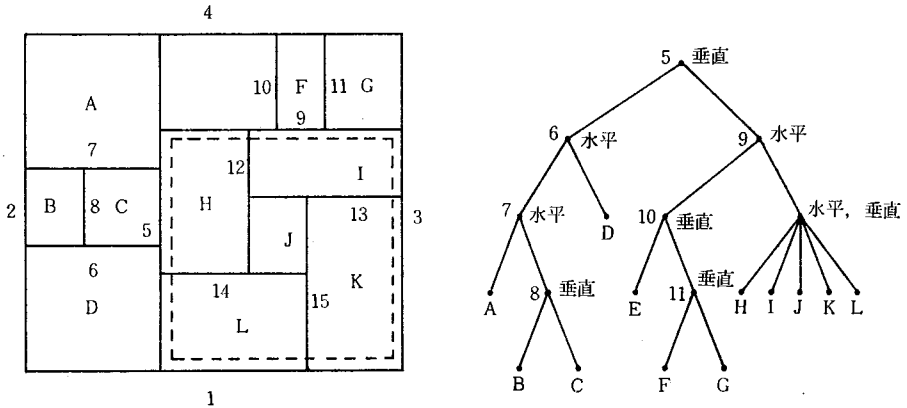


図 6 簡単なピンホイール配置

Fig. 6 A simple "pinwheel" arrangement of components

アルゴリズムに有効である。しかしながら, Szepieniec<sup>[21]</sup> と違い, ピンホイール (pin-wheel) までを含めたコンポーネントの任意の配置をわれわれは許している (図 6)。

ASHLAR では, より一般的配置は会話型配置によってのみ達成されるが, 簡単なピンホイールは自動配置アルゴリズムで作上げられる。そのような配置を木表現に含めるため, チャンネル位置グラフを木ノードと関連づける (図 7)。



(a) ピンホイールを含む分割された空間

(b) 相当する木

(c) ピンホイールの垂直チャンネル位置グラフ: (d) ピンホイールの水平チャンネル位置グラフ:  
 頂点は垂直チャンネルを, 辺は二つの垂直チャンネル間にあるコンポーネントを表わす。 頂点は水平チャンネルを表わす。

図 7 木とチャンネル位置グラフを使ったレイアウト空間の表現

Fig. 7 The representation of a partitioned layout space using trees and channel position graphs

#### 4.1 配 置

配置 (placement) の過程での主作業は, 設計者が指定した外形比に近づけながら, 配線距離と余白スペースを最小化する回路コンポーネント配置をみつけることである。この作業では, 次の 3 段階アルゴリズムが使われる。

- 1) クラスタリング (clustering)……回路の接続が解析され、互いに密な接続をもつコンポーネントは一つのクラスタに結合される。これは Murai ら<sup>[13]</sup>のものに似たクラスタ木を生じるが、極端に指定面積が不揃いの部分クラスタをもつようなクラスタが形成されるのを避けるために、特別な注意が向けられる。
- 2) 空間分割 (space partitioning)……この段階で取り扱われる最初の矩形空間は、要求外形比と最終レイアウト面積の見積りにより決まる寸法をもつ。この矩形空間は、クラスタ木に属する部分クラスタの見積り面積相当の空間に、繰り返し分割されていく。この分割方法は、Murai<sup>[13]</sup> や Lanther<sup>[10]</sup> らの方法と関係がある。数個の（7個以下）コンポーネントからなるクラスタまで分割が進むと、クラスタを形成する各コンポーネント相互の関係と全コンポーネントのレイアウトの両方からみて、うまく組み合わされているかどうか、別の方法で徹底的にチェックされる。そして、一つのコンポーネント・レイアウトは実現可能なレイアウト案の中から選択され、それぞれの初期配置方向もあわせて決定される。空間分割が完了した後、全体のレイアウト寸法とコンポーネント初期位置を評価するのに、クリティカル・パスにもとづく方法とられる（図8）。

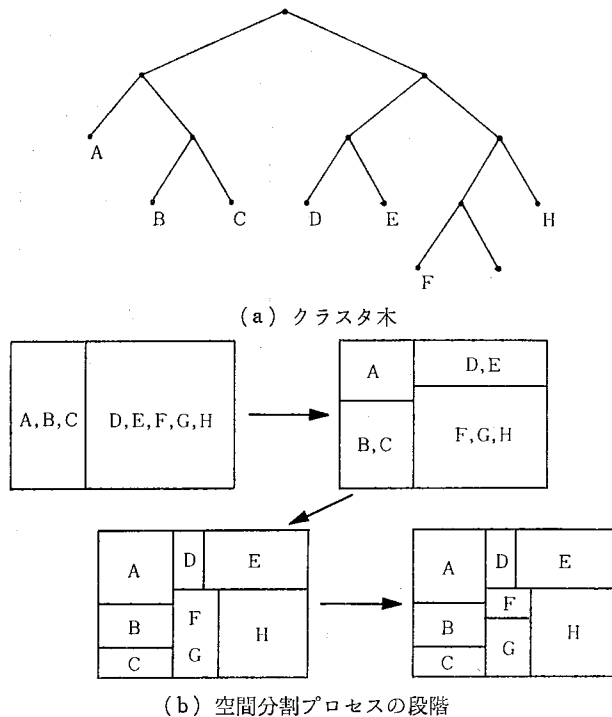


図8 空間分割プロセス(A~Hはクラスタを表わし、各クラスタは7個以下のコンポーネントからなるクラスタを分割し、コンポーネントが割り当てられた空間に合うように配置する。)

Fig. 8 The space partitioning process (A through H represent clusters of fewer than seven components. A separate process does the partitioning of these small clusters to arrange the components to fit into the spaces allocated here.)

- 3) 配置 (arrangement)……コンポーネントそれぞれは実質上その大きさと形状が一樣でないため、通常とられる交換法は任意配置レイアウトには適用できない。しかしながら、すべての分割が並列であれば、いずれの空間もその部分空間同士の交換は可能である（図9）。この交換により、全体の寸法を一定に保ったままで配線距離を短縮

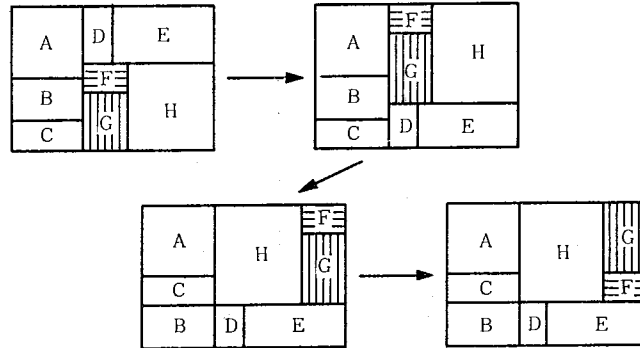


図 9 図 8 の例での配置プロセスの段階（配線長を短縮するためクラスタは相互に交換される。割り当てられた空間の大きさは変わらない。）

Fig. 9 Several stages in the arrangement process of the example shown in Fig. 8 (Whole clusters are interchanged to reduce routing distances. Note that the dimensions of the allocated spaces do not change.)

することができる。われわれのとった方法においては、そのようなすべての交換をクラスタ木を通じてトップ・ダウンにチェックする。個々のコンポーネントの位置が確立した時点で最後にコンポーネントの配置方向が選択される。

以上が自動配置の基本アルゴリズムの概要であるが、数多くの変形が可能であり、われわれも多くの改良を加えてきた。とくに、自動配置の前と後の両方に会話型配置機能を適用できるようにした。

#### 4.2 大局的配線

大局的配線 (global routing) あるいは大雑把な配線のプロセスでは、最終的に配線を割り付ける配線チャンネルを選択して、だいたいの経路を決定する。配置プロセスで定められた概略コンポーネント位置にもとづいて各ネットの初期配線が施される。経路はコンポーネントの位置にもとづいて推測された配線距離を最短にするように選ばれる。次の再配線のプロセスでクリティカル・チャンネルの密集部分にあるネットを見分け、それら密集経路を避けるように再配線をする。この再配線の目標は、全体 (チップ) のレイアウト寸法に影響を与えるチャンネル幅の縮小におかれる。

大局的配線では、チャンネル相互の交点と接続すべき点 (接続すべきネットのポート) を表わす“頂点” (vertex) とチャンネルを表わす“辺” (edge) で構成されるグラフを用いて、チャンネル配置を表現する。

実際には、各チャンネルは一つあるいは複数の辺で表現されることになる。なぜならば、一つのチャンネルは、その途中にある頂点によっては複数のサブ・チャンネルに分割されることがある。このグラフをサブ・チャンネル交差グラフと呼ぶ (図 10)。

大局的配線アルゴリズムは、グラフ内の最短経路を見つける Dijkstra のアルゴリズムにもとづく<sup>[4]</sup>。初期配置においてグラフの一つの辺に付加されることに生じるコストは、チャンネル長そのものである。その結果、2 個の頂点を結ぶ最短経路は、頂点間の Manhattan 距離 (格子長  $\Delta x + \Delta y$ ) より短いことはありえない。

このことを利用すると、目標点から遠すぎて最短経路に表われてこないような頂点を経路探索の対象から除くことができるので、アルゴリズム効率の改善に役立つ。

#### 4.3 電源バス配線

二つの重要な問題が電源バス配線で生じる。第 1 に、1 層のメタル配線だけが許される

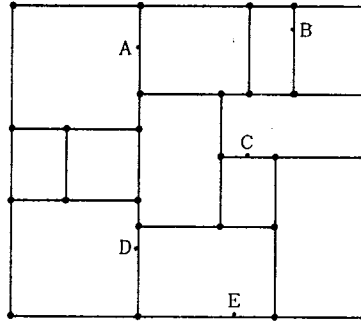


図 10 大局的配線のためのサブチャネル交差グラフ  
(A~E は配線すべきネットのポート位置を示す。他の頂点はチャネル交差を示す。)

Fig. 10 A subchannel intersection graph used for global routing  
(A through E represent the locations of ports of a single net to be routed. Other vertices represent channel intersections.)

とき、すべての電源バスをそのメタル層上に保持することが必要である。第2に電源バス配線は、メタル移動、直流過渡電圧降下による問題を避けるに十分広い線幅をもたなければならない。

ASHLAR では、2種類の大局的配線モードが電源バスの配線に使うことができる。一つのモードは通常の大局的配線法も採用し、電源バスを含むすべてのネットに2配線層を要求する。これは2メタル層が使えるとき、採用可能なモードである。第2のモードはメタル1層のみが使えるときに適し、Syed や Gamal<sup>[20]</sup> が記した配線技術を採用する。これは、1層上に2本の電源ネットを互いに交差することなく結線させる(図11)。他のネットは、通常両層を行う。

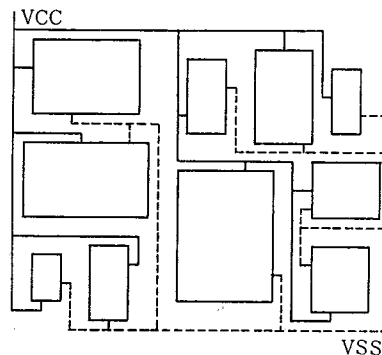


図 11 交差のない2本の電源バス配線の例  
(メタル1層が2本の電源バス配線に使える。)

Fig. 11 An example of two power busses routed without crossing each other, so that a single metal level can be used for both

いずれかの方法により、一度電源バスの大局的配線が完了すると、それぞれの電源バス・セグメントに必要な(線)幅が計算できる。この計算には、配線層の抵抗と最大電流密度、各電源端子から流れ出す直流およびピーク電流、各電源端子での最大許容直流過渡電圧降下値などのデータが必要である。われわれの配線方法は木構造バスを与えるため、計算は比較的容易である。

そのような木に属する各配線セグメントを流れる直流電流値は容易に計算でき、過度な



電流密度によって生じるメタル移動を避けるのに必要とされる最小線幅を、ただちに決定できる。一方、直流電圧降下と過渡電圧降下を考慮したセグメントの要求線幅は、相互に依存しているため決定するのが困難である。われわれは、すべての電源端子についての要求が満たされるまで、繰り返し手法を使って、種々の配線セグメント幅を増やしている。大局的配線で定まったすべての配線セグメントと、その算出された線幅はトラック配線 (track routing) プロセスへ引き渡される。

#### 4.4 トラック配線

矩形探索法<sup>[11]</sup>や迷路法<sup>[19]</sup>など、最終配線のさまざまなアプローチが論じられてきた。われわれはチャンネル幅が調節でき、すべての配線完了を保証するチャンネル配線法を選択した。そのアルゴリズムは Deutsch<sup>[3]</sup>のそれと似ており、制約グラフの中のループを崩すことができ、急に折れ曲る配線を取り入れている。

チャンネル内配線に当たっては、どのチャンネルから配線していくかがきわめて重要である。一般に、1番目のチャンネルが2番目のチャンネルに行き当たって終わるとき、2番目のチャンネルの配線以前に1番目の配線をすべきである。そうしなければ、2番目のチャンネル領域の境界に存在するポートの正確な位置がわからない。なぜならば、1番目のチャンネルの幅はまだ決まっていないからである。われわれの場合、コンポーネントの任意配置を許しているため、チャンネル順序の干渉を引き起こすチャンネル配置が生じる。たとえば図12に示すごとく、チャンネルAはチャンネルB以前に、BはC以前に、CはD以前に、DはA以前に配線しなければならないという矛盾が生じる。

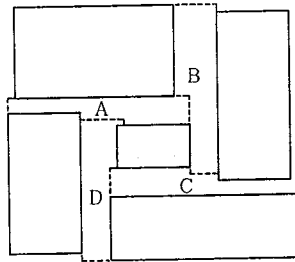


図 12 ピンホイールのチャンネルでトラック配線順序が干渉しあう。(AはBより、BはCより、CはDより、DはAより以前に配線すべきである。)

Fig. 12 Track routing order conflict for the channels of a pinwheel. (A should be routed before B, B before C, C before D, and D before A.)

この問題に対するさまざまな解決法が提案されてきた。その解決法とは干渉を排除するため、そのような配置を禁止または変更する方法<sup>[2], [21]</sup>や、チャンネルの1本を互いに直接接続のない隣接した2本の平行なチャンネルに分割する方法<sup>[9]</sup>などである。われわれは Preas の論文<sup>[16]</sup>に従うことを選んだ。1本のチャンネル幅を見積り、この見積りにもとづいて交差する他のチャンネルを配線し、最後に見積った(最初の)チャンネルを配線するものである。その見積りは、ほとんどのケースで配線が完了できる程度に十分寛大になされる。もし、失敗のケースが生じれば新たに見積りをやり直し、他のチャンネルも配線のやり直しとなる。

ASHLAR は、元来、メタル2層チップに適用しようとするものである。この場合、主として1層は垂直配線セグメントに、もう一方の層は主として水平配線セグメントに使われる。その結果、垂直チャンネル軸に平行な幹線あるいは配線セグメントはいずれか1層に、水平チャンネルに平行な他の配線はもう一方の層に置かれる。これによって、比較的簡単な

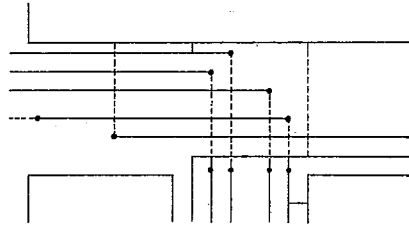


図 13 チャンネル交点でメタル 2 層を使用

Fig. 13 Use of two metal levels in channel intersections

チャンネル交差が実現できる (図 13)。

メタル 1 層と多結晶シリコン層に使うときは、長い多結晶シリコン・セグメントを避けるため、メタル層はすべてのチャンネルの幹線に使われる。したがって、この方法は、多結晶シリコン経路に接続するスルーホールをたくさん作り出すチャンネル交差を複雑なものにする (図 14)。すべての電源バス配線をメタル層に置くために、特別な電源バス配線スキームが使われる。

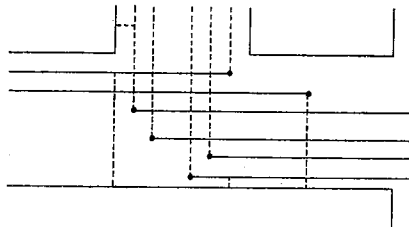


図 14 チャンネル交点でメタル 1 層 (実線) と多結晶シリコン層 (点線) を使用

Fig. 14 Use of one metal level (solid lines) and one polysilicon level (dashed lines) in channel intersections

ASHLAR では、すべての寸法は“レイアウト単位”で表現される。これは、レイアウト単位と使用者の与えるマイクロメータ ( $\mu\text{m}$ ) を対応させたものである。1 レイアウト単位は、代表的配線幅よりずっと短い距離を表現できる。この意味でチャンネル配線アルゴリズムは格子がないアルゴリズム<sup>[18]</sup>と同様に、いかなる幅の配線セグメントも、コンポーネント上のいかなる位置にあるポートでも、いかなる幅のチャンネルをも取り扱うことを可能にする。

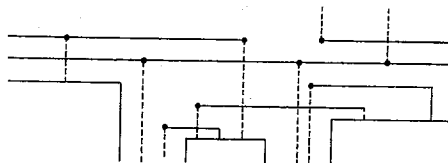


図 15 チャンネルに沿った湾の配線

Fig. 15 Routes in bays along a channel

トラック配線アルゴリズムのもう一つの特徴を図 15 に示す。コンポーネントの寸法が変わるため、その辺は配線チャンネルの側辺上に一線に並ばない。この結果、チャンネルに沿って湾 (bays) が現われる。ASHLAR トラック配線プロセスは、この空間を配線に利用しチャンネル幅を縮小する。

## 5. プロトタイプ実行

ASHLAR で採用される基本アルゴリズムを、実験用プロトタイプ・プログラムの上に実現した。このプロトタイプは ASHLAR に意図した多くの機能（たとえば、会話型機能、ピンホイールの処理、電源バスの特別処理、さまざまなレイアウト最適化機能など）を削っている。しかし、基本アルゴリズムの有効性は証明する。

図 16 は、このプロトタイプ・プログラムで生成されたレイアウトを示している。この図は、33 個のコンポーネントと 134 ネットからなる 8 ビット・マイクロプロセッサを表わす。このコンポーネント群は、簡単なゲートとインバータから RAM, ROM, ALU ブロックなどの高度な複合セルの範囲にまで至る。このレイアウトに要した CPU 時間は、UNIVAC 1100/80 シリーズ (2.1 MIPS) のうで 35 秒であった。その内訳は、配置 5 秒、大局的配線 6 秒、トラック配線 24 秒である。

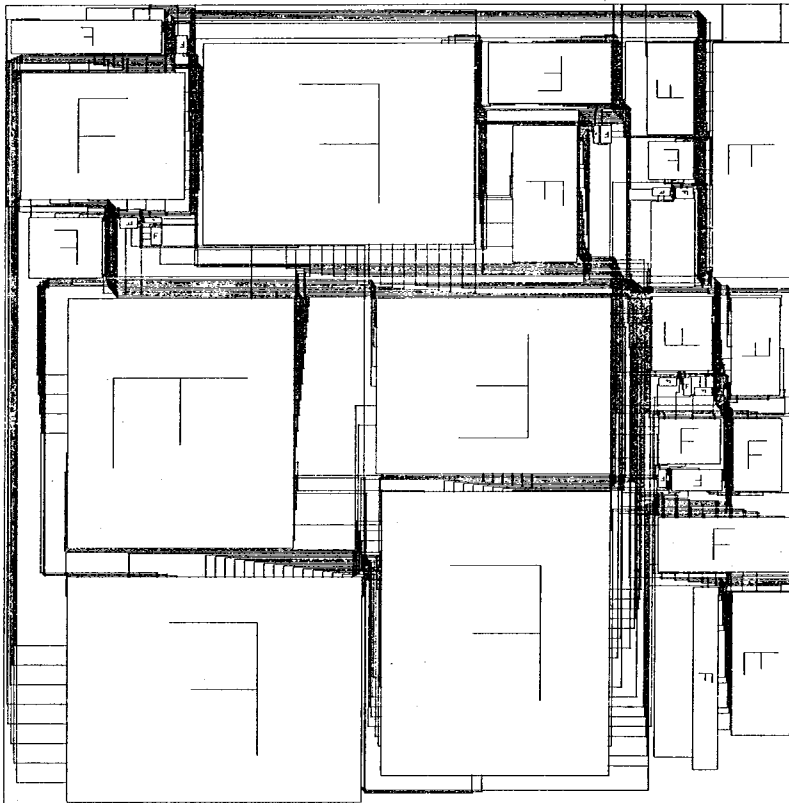


図 16 プロトタイプにより作られたレイアウト図

Fig. 16 A layout produced by the prototype program

## 6. おわりに

ASHLAR プログラムのために開発中の自動レイアウト機能の概要を示し、主要な特徴を考えることに注目した。ASHLAR セル定義は、可能なレイアウトと回路定義を取り込み、ASHLAR データベースを形成する柔軟、かつダイナミックなセル・ライブラリの基礎となる。たいていの自動配置アルゴリズムにとって適していると考えられる“3段階配置アルゴリズム”の概要を述べた。さらに、大局的配線とそれに続くトラック配線からな

る“配線プロセス”を，電氣的要求を満たすように電源バスに特別な配慮をしたことを強調した．また，多結晶シリコン配線セグメントの使用を最小に押さえる方法に言及し，メタル2層製造プロセスが可能であれば，われわれの見込んだほとんどの応用上で最小化の問題を取り除くことができることも述べた．

本稿でふれなかった ASHLAR のきわめて重要な側面は，ユーザ・インターフェースである．ASHLAR の会話型機能は，自動プロセスの制御，レイアウト結果と他の案との比較・評価・セル階層のレベルにまたがるレイアウトの注意点（外形比やポート位置など）の管理のために欠かせないものである．

われわれが ASHLAR の使用から得た階層的レイアウトの経験から，最後にはレイアウト・プロセスをより十分に自動化できることが期待される．しかし，ごく近い将来では，この分野の使用者の技量に頼らねばならない．

ASHLAR プロジェクトのメンバの T. Goessling, M. Newstrom, J. Glynn, J. Hildebrandt, J. Harra, T. Freuler, K. Kehlenbeck の協力を感謝する．

（応用ソフトウェア部 渡部 義維 訳）

- 参考文献
- [1] H. Bake, W. Sansen, “CALMOS: A portable software-system for the automatic and interactive layout of MOS/LSI,” *Proceedings of the 16th Design Automation Conference* San Diego, 1979, pp. 102-108.
  - [2] T. Chiba, N. Okuda, T. Kambe, I. Nishioka, T. Inufushi, S. Kimura, “SHARPS: A hierarchical layout system for VLSI,” *Proceedings of the 18th Design Automation Conference*, Nashville, 1981, pp. 820-827.
  - [3] D. Deutsch, “A ‘dogleg’ channel router,” *Proceedings of the 13th Design Automation Conference*, San Francisco, 1976, pp. 425-433.
  - [4] S. Even, *Graph Algorithms*, Computer Science Press, 1979.
  - [5] A. Feller, “Automatic layout of low-cost quick-turnaround random-logic custom LSI devices,” *Proceedings of the 13th Design Automation Conference*, San Francisco, 1976, pp. 79-85.
  - [6] A. Feller, R. Noto, “A speed-oriented, fully-automatic layout program for random logic VLSI devices,” *National Computer Conference Proceedings*, 1978, pp. 303-311.
  - [7] K. W. Fiduk, Jr., D. W. Hightower, F. A. Coady, “A non-modular placement algorithm with applications,” *Proceedings: IEEE international Conference on Circuits and Computers*, 1980, pp. 83-86.
  - [8] C. M. Harris, *Dictionary of Architecture and Construction*, McGraw-Hill, Inc., 1975.
  - [9] C. S. Horng, M. Lie, “An automatic/interactive layout planning system for arbitrarily-sized rectangular building blocks,” *Proceedings of the 18th Design Automation Conferences*. Nashville, 1981, pp. 293-300.
  - [10] U. Lauther, “A min-cut placement algorithm for general cell assemblies based on a graph representation,” *Proceedings of the 16th Design Automation Conference*, San Diego, 1979, pp. 1-10.
  - [11] U. Lauther, “The SIEMENS CALCOS system for computer aided design of cell based IC layout,” *Proceedings: IEEE International Conference on Circuits and Computers*, 1980, pp. 768-771.
  - [12] R. Malladi, G. Serrero, A. Verdillon, “Automatic placement of rectangular blocks with the interconnection channels”, *Proceedings of the 18th Design Automation Conference*, Nashville, 1981, pp. 419-425.
  - [13] S. Murai, H. Tsuji, M. Kakinuma, K. Sakaguchi, C. Tanaka, “A hierarchical placement procedure with a single blocking scheme,” *Proceedings of the 16th Design Automation Conference*. San Diego, 1979, pp. 18-23.
  - [14] G. Persky, D. N. Deutsch, D. G. Schwelkert, “LTX-A system for the directed automatic design of LSI circuits,” *Proceedings of the 13th Design Automation Conference*, San Francisco, 1976, pp. 399-407.
  - [15] B. T. Preas, C. W. Gwyn, “Methods for hierarchical automatic layout of custom LSI

- circuit masks," *Proceedings of the 15th Design Automation Conference*, Las Vegas, 1978, pp. 206-212.
- [16] B. T. Preas, *Placement and Routing Algorithms for Hierarchical Integrated Circuit Layout*. Ph. D. Dissertation, Dept. of Electrical Engineering, Stanford University, 1979.
- [17] K. Sato, T. Nagai, H. Shimoyama, T. Yahara, "MIRAGE-A simple-model routing program for the hierarchical layout design of IC masks," *Proceedings of the 16th Design Automation Conference*, San Diego, 1979, pp. 297-304.
- [18] K. Sato, H. Shimoyama, T. Nagai, M. Ozaki, T. Yahara, "A 'grid-free' channel router," *Proceedings of the 17th Design Automation Conference*, Minneapolis, 1980, pp. 22-31.
- [19] J. Soukup, J. Royle, "Cell map representation for hierarchical layout," *Proceedings of the 17th Design Automation Conference*, Minneapolis, 1980, pp. 591-594.
- [20] Z. A. Syed, A. El Gamal, "Single layer routing of power and ground networks in integrated circuits," *Journal of Digital Systems*, Vol. 6, No.1, Spring 1982, p. 53.
- [21] A. A. Szepieniec, R. H. J. M. Otten, "The geneological approach to the layout problem," *Proceedings of the 17th Design Automation Conference*, Minneapolis, 1980, pp. 535-542.

執筆者紹介 James E. Hassett

1971年に Minnesota 大学で数学の BA を、また 1973年に Wisconsin 大学でコンピュータ・サイエンスの MS を得ている。1973年に Sperry Univac 社に入社後、国防システム部門の設計自動化部において、主としてプリント基板の自動配線および集積回路の自動配置に従事している。現在はプリンシプル・サイエンティフィック・プログラマであり、同部門の自動配置設計グループのスーパーバイザを務めている。なお、ASHLAR プロジェクトに 1979 年より参加している。



## 報告 DSS 概念の実現

### Realization of DSS concept

小坂 武

**要約** ビジネス分野へのコンピュータの適用として、OA（オフィス・オートメーション）とともに、近年 DSS（意思決定支援システム）が大きな関心を呼んでいる。

DSS と呼ばれるシステムには多くの種類があり、1人のマネジメントがそれらの複数のシステムを使い分けなければならないのが現状である。本稿では、この使い分けのむずかしさから、汎用的な DSS の必要性を述べ、その必要を満たすには DSS パッケージが有効であることを示す。さらに、意思決定の場で使用するデータとマネジメントの問題解決の様式から、DSS パッケージが具備すべき基本形態を明らかにする。

具体的な DSS パッケージとして、日本ユニバック(株)が開発中の DSS 1100-J を取り上げ、その基本形態の実際を紹介し、DSS 1100-J が多様な DSS の構築に有効であることを論じる。

**Abstract** Decision support systems as well as office automation these days arouse increasing interest of people concerned with computer applications in business.

In DSS's there is a problem that managers are compelled to select and use multiple systems because facilities supporting their decisions are scattered in many systems. This paper shows the needs of a general-purpose DSS and proposes a DSS package which is effective for eliminating the problem. And then, it formulates an architecture necessary for the DSS package, based on the types of data used in decision making and the study of unstructured problem solving.

As a specific DSS package this paper presents DSS 1100-J under development at Nippon UNIVAC. With the discussion of its actual architecture it shows that DSS 1100-J is effective to build a variety of DSS's.

#### 1. はじめに

日本ユニバック(株)は、ここ数年来、意思決定支援システム (Decision Support System, 以降 DSS と略す。) を構築するためのソフトウェアの開発に力を注いできた。DSS 1100-J (Decision Support System for UNIVAC 1100-Japanese) はその成果であり、日本語が使える意思決定支援システムとして昨年登場し、さらに、各種のビジネス・グラフを低価格の端末機から利用するために継続して開発が行われている。

DSS 1100-J は、最近各社から発表されている DSS 用のソフトウェアと異なり、ホスト・コンピュータの特徴を生かして、マネジメント用の大規模なデータベースを取り扱うことができる。このデータベースには、時系列データを自然に表現するための工夫がなされている。また、データベースからデータを効率的に取り出し、プログラムを組むことなく、マネジメントの思考に沿ってデータを自在に加工することができる。

次章では、まず最初に DSS の概念を再確認する。つぎに、3章で現実的な要請からインプリメントされてきた多くの DSS を分類・概観し、その多様さを確認する。4章、5章では、マネジメントが望む本来的な DSS を明らかにするために、DSS が使用される環境と DSS のかわり、および人間の問題解決の様式と DSS のかわりを考察する。最後に6～11章において、DSS パッケージである DSS 1100-J が、DSS に対する基本的な要

件をどのように実現しようとしているかを紹介する。

本稿は、IFIP のジャーナル「Information & Management」(1982 年 Vol. 5, No. 1, North-Holland) に掲載された論文「An Effective Architecture for Decision Support Systems」(T. Kosaka and T. Hirouchi)<sup>[1]</sup> との関係のもとに執筆したものである。同論文は DSS の基本形態を主にコンピュータ・ソフトウェアの面から詳細に論じた。これに対し、本稿は DSS の基本形態を主に現実の場におけるマネジメントの問題解決の面から論じている。

## 2. 多様な DSS

言葉の出所や概念が明確でありながら、解釈が実に多様な用語の一つに意思決定支援システム (DSS) がある。DSS という概念は M. S. Morton により、1971 年に、MIS (Management Information Systems) に対比する概念として提唱され、研究者やマネジメントやコンピュータ産業界の間に大きな関心を引き起こした。その後、P. G. W. Keen らによって、その概念が洗練された。ここで、以降の議論を明確にするために、その概念を引用しよう。

DSS は、マネジメントの意思決定過程におけるコンピュータの役割りについて一つの見方を提示するものである。意思決定支援 (decision support) とはコンピュータを次の三つの観点で使用することを意味する<sup>[1]</sup>。

- 1) 半構造的 (semistructured) な仕事における意思決定の過程で、マネジメントを支援する。
- 2) マネジメントの判断に取って代わるのではなく、これを支援する。
- 3) 意思決定の効率 (efficiency) よりも、むしろその効果 (effectiveness) を改善する。

ここで、半構造的という用語には若干の注釈を要する。構造的という用語は H. A. Simon によって提唱された<sup>(注1)</sup>。問題解決における手順が事前に明確にされている意思決定を構造的な意思決定 (structured decision) と呼び、事前に明確になっていないものを非構造的な意思決定 (unstructured decision) と呼ぶ。

半構造的な意思決定とは、次の分野の仕事の意味する。これは、問題の大きさや計算の煩雑さ、あるいは要求される精度のために、人間の判断だけでは不十分な仕事である。また、問題の解決に当たって判断や主観的分析が必要とされるため、モデルやデータだけでも不十分な仕事がある。これらの仕事では、人間とコンピュータの両者からなるシステムが有効であり、そして、人間がコンピュータをコントロールできることが望まれる<sup>[2]</sup>。

一般的に、DSS に関する定義は上述の概念のように、人間とコンピュータのかかわりに関するものが主で、コンピュータ・ソフトウェアとしての定義はほとんどないといっよい。そのため、世の中における DSS の具体的な姿は非常に多様である。コンピュータによる情報システムとしての DSS があると思えば、一方には、パーソナルな分析・作図ツールまでもが DSS と呼ばれている。それは今日の OA の議論に似て、ビジネス分野の問題の宿命のように、DSS は百人百様の様相を呈している。

## 3. DSS の分類と DSS パッケージ

現実的な要請から構築されてきた数多くの DSS を分類するための有用な基準が、S. Alter によって提案されている。彼の分類基準は、人々が DSS について議論する場合のコミュニケーション上の枠組を提供し、多様な DSS の中でどのタイプの DSS につい

て議論しようとしているかを明確にして、不必要な混乱を回避するのに有用である。彼は、さらに DSS の成功・不成功要因について、そのキー・ポイントを DSS のタイプごとに提示している。DSS の成功は、システムがもつデータや機能の豊富さ以上に、導入後の運用様式にかかっている。それゆえ、彼が提示する DSS の運用上のキー・ポイントは各企業の DSS 担当スタッフに有用な指針となる。

ここでは、Alter の DSS の分類を DSS の多様さを明らかにするために使用し、その多様さの中に潜む問題点を洗い出す。

Alter は、システムの出力が最終的な意思決定にどの程度直接的なかかわりを有するかという点を分類基準として、調査した 56 個の DSS と呼ばれているシステムを、次の七つのタイプに分類している<sup>[3]</sup>。

- 1) ファイル検索システム (file drawer system)……オンラインでファイルのアクセスを行わせる。
- 2) データ分析システム (data analysis system)……ファイルにある時系列データ、あるいは非時系列データの分析を一般的に用意した、あるいは特別に用意したコマンドを使用して行わせる。
- 3) 分析的情報システム (analysis information system)<sup>(注2)</sup>……マネジメントのために用意された一連のデータベースと、小さなモデルへのアクセスにより経営情報を提供する。
- 4) 会計モデル (accounting model)<sup>(注3)</sup>……主に、会計上の定義にもとづいて、計画されたアクションの結果を計算する。
- 5) 説明モデル (representational model)……部分的に定義的でない式を含むモデルにもとづいて、アクションの結果を推定する。
- 6) 最適化モデル (optimization model)……アクションのための指針として、一連の制約条件を満足する最適解を提供する。
- 7) 提案モデル (suggestion model)……十分構造化された仕事に対して、具体的な意思決定そのものを提案するために機械的な作業をする。

これらの DSS は、図 1 のようにデータ指向とモデル指向の DSS として、さらに分類することができる。

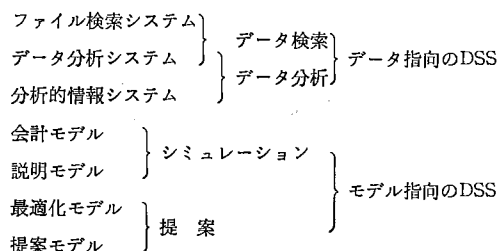


図 1 意思決定支援システムの分類

Fig. 1 A taxonomy of decision support systems

ここでは、これらの分類の詳細について説明する余裕はないが、非常に多様なシステムが DSS という言葉で呼ばれていることが理解されよう。

このように各種のシステムが DSS と呼ばれる背景の一つには、各システムにはそれぞれ中心的役割を果たす手法・技法があり、そのまわりに補助的機能が付加され、それぞれ独立なシステムとして作成されていることがあげられる。中心的役割を果たしている手法・



技法は、データ指向の DSS ではデータベース問い合わせ技術であり、モデル指向の DSS ではモデル言語や線形計画法である。これらの固有な手法・技法は、それぞれが比較的大きなソフトウェアであるため、個々の手法・技法だけで特別に作られた専用のシステムが流通しており、それらを使用して具体的な DSS が構築されてきているわけである。

現実の意思決定の場では、これらのシステムが単独で使用されることはまれで、一般的には複数のシステムが使用される。たとえば、債券ポートフォリオ・マネージャの場合、ポートフォリオの最適な組み合わせを決定するのに最適化モデルのためのシステムを使用するが、手持ちの債券を調べるためには別の分析的情報システムを使用したり、あるポートフォリオのもとにおける会社の財務状況を予測するためには別の会計モデルのシステムを使用したりする。

マネジメントの機能(仕事)は、状況把握(管理)と企画立案(計画)の二つからなる。おおまかにいえば、状況把握はデータ指向の仕事であり、企画立案はモデル指向の仕事である。このため1人のマネジメントがデータ指向の DSS とモデル指向の DSS を併用することは、彼の役割を全うするために、ごく自然な行動である。

しかし、現実には、個々のシステムはバラバラなため、マネジメントによる DSS の直接的な使用は阻害されている。システム間のギャップを埋めるために、システムの直接的な使用はスタッフにより行われ、マネジメントは間接的な使用者となっているのが現状である。

ここでは、特別に用意された専用のシステムの必要性を否定しようとしているのではない。スタッフが高度な分析を行うためには、専用のシステムが要求されることがある。しかし、マネジメントが最終的な使用者である DSS は、マネジメントが理解できるモデルでなければ何の意味もないことから、高度な専用システムである必要性は少ないといえよう。コンピュータの専門家でないマネジメントにとって、いくつものシステムを使い分けることは事実上不可能である。

DSS の概念の中に、DSS の直接的使用者はマネジメントであるという項目はない。しかし、マネジメントが直接の使用者であれば、モデルの中に彼の考えを反映しやすく、DSS をより有効に使える可能性がある。そのため、DSS の設計にはマネジメントの直接的な使用を考慮に入れることが望まれる。

複数のシステムの使い分けをなくし、さらにマネジメントがシステムを直接使用できるようにする有効な一つの手段として、本稿では DSS パッケージを取り上げる。DSS パッケージは、DSS システムとも呼ばれ、図2に示すように特定の分野のデータとモデルを

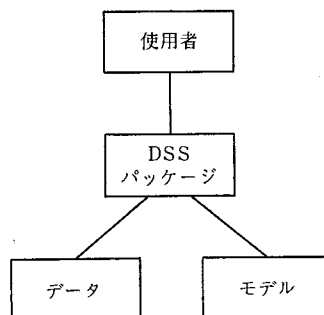


図2 DSS パッケージ

Fig. 2 A DSS package

与えれば、その分野の DSS を生成する、いわば半製品のソフトウェアである。

使用者は、DSS パッケージを通して、データとモデルに接近する。この DSS パッケージは、使用者がデータとモデルへ接近する場合に、一つの統一的な相互作用形式を提供するのに役立つ、柔軟性と使いやすさを実現するうえでの重要な手掛りを与える。DSS パッケージのもとでは、一つのシステムから複数のデータベースとモデルへ接近できるので、DSS パッケージはマネジメントが直接使用しうる DSS の構築手段となりえる。

DSS パッケージは、各種のモデルをシステム内に同化する必要があることから、個々の問題解決技法にとらわれることなく、その基本形態が意思決定支援の観点から、決定されるべきものである。DSS パッケージの設計では、マネジメントが使用するデータの型に着目し、かつ、それらをどのように処理するかという問題解決の過程に着目する必要がある。そこで、4章と5章では、マネジメントに使用されるデータの型と問題解決の様式を分析することにより、DSS パッケージの基本形態を明らかにする。

#### 4. データ

意思決定の場で使用されるデータは、業務で使用される特定時点での取引記録であるオペレーショナル・データの単なる集積ではない。J.F. Rockart と M.E. Treacy によれば、取引記録を次の三つの観点で分類・要約し、表現したデータが経営情報となりえる<sup>[4]</sup>。

- 1) ビジネス単位 (business unit)
- 2) ビジネス変数 (business variable)
- 3) 時間 (time)

ビジネス単位は、一般に人に関連した組織単位を意味し、ビジネス変数に対して責任を有する単位、あるいはそれを所有する単位をいう。具体的には販売店、部、地区などである。ビジネス変数は俗にいう、物や金であり、商品や財務科目などの種類を表わす。この三つの観点が軸を構成し、経営情報は立方体で表わされる。たとえば、売上情報は図3(a)のような立方体で表現できる。

現実の経営情報は、さらに複雑で上記の三つの軸の他に、

- 4) 計画・実績の区分
- 5) データの計量区分 (金額表示・物量表示、たとえば金額表示では平均残高や期末残高の区分)

を必要とする。そのため、売上情報といっても、この五つの指示をしないと一意に決定で

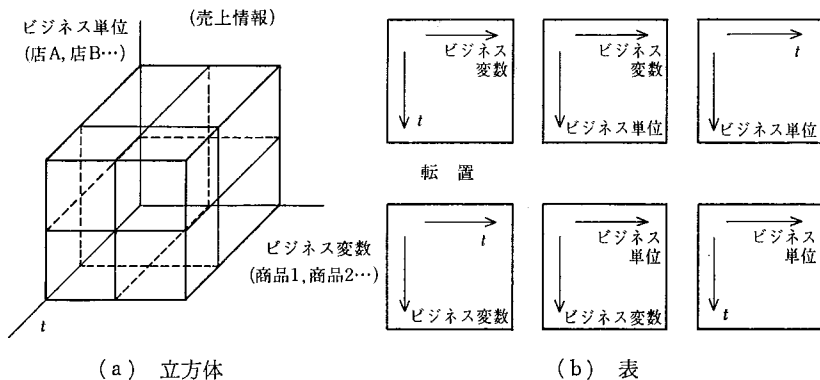


図3 経営情報

Fig. 3 Management information

きない。

経営情報を取り扱う場合、人間はこの立方体あるいは超立方体を自然に眺めることはできない。図3(b)に示すように立方体の任意な断面を切り出してはじめて、人間はそれを理解できる。

この断面の切り出し方は人それぞれ、あるいはその時々目的によって異なり、同一の断面でも、表を転置した形式で使う場合もあるため、一つの分野でも6個の種類の表が使用されることになる。

これら6種類の表は、それぞれ異なった目的のために切り出されたものであるから、一種の判断基準のもとに作成された情報としての意味あいを有する。一方、もとの立方体は、情報としての6種類の表を作り出す源泉となったもので、価値判断が入っていないということから、相対的なデータとして認識される。

オフィスにおけるマネジメントは、孤立した意思決定をすることはまれである。あるマネジメントによりなされた意思決定は、上司あるいは部下の関連した意思決定と相互作用し合う。あるいは、他部門のマネジメントの意思決定との間にも関係を有するであろう。この種の相互作用を有効にするには、その企業内にマネジメント用のデータベースが構築されるべきで、それは大規模な共用データベースとして位置付けられる(注4)。

マネジメントが使用するデータは企業内の内部情報ばかりではない。外部情報や彼自身が集めた情報も内部情報とともに利用される。意思決定の場で、当然個人的な情報も使用されるわけで、すべてが共用データベースでまかなえるものではない。また、たとえ共用データベースから取り出した情報でも、自分なりにそれを加工し、付加価値を高めることをする。このような情報の加工や、個人的な情報の保存のためにパーソナル・ファイルが共用データベースとともに存在する。

共用データベースは、データ間の不整合を回避し、かつデータの一元管理を図るために立方体の形式を自然に表現したものであることが望まれる。しかし、パーソナル・ファイルは見通しのきくファイルであることが操作性を高める重要な要因であることから、表の形式が自然といえよう。

共用データベースとパーソナル・ファイルの構造をこのように違えれば、立方体から断面を切り出して表を作成するという方式が必要となるが、これは決して欠点ではない。共用データベースは多様な情報要求に応えねばならず、これをデータの形式ではなく、情報の形式でもった場合、それらは互いに冗長性の高いものとなり、かえって不必要に情報量が多いために情報への接近を困難にし、混乱させかねない。一方、パーソナル・ファイルを立方体でもった場合は、その可視性が悪く操作性を低下させるものとなる。

DSS と呼ばれている多くのシステムは、この共用データベースとパーソナル・ファイルの両方を一つのシステム内で扱っておらず、どちらか一方というのが多い。そのため、パーソナル・ファイル形式のデータだけを扱っているものはモデル指向の DSS に多く、共用データの取り扱いに不便さがある。また、データ指向の DSS ではこの逆となっている。

この二つの形式のデータ形式を DSS がそなえることは、それが広い範囲の問題に対応するための条件といえよう。また、これら二つのデータ形式が別個に取り扱ってはじめて、大量データをもつ場合の効率への配慮や作業ファイルとしての使いやすさを十分に実現できることになる(表1)。

表 1 共用データベースとパーソナル・ファイル

Table 1 Databases and personal files

	共用データベース	パーソナル・ファイル
形 式	立 方 体	表
ファイル形態	データベース	個別ファイル
加 工 度	データ形式	情報形式
保 存 量	大 量	小 量
強 調 点	効 率	操 作 性
冗 長 性	低 い	高 い
作 成 者	データベース管理者	エンド・ユーザ

## 5. モ デ ル

DSS とは、マネジメントに取って代わる意思決定システムではなく、マネジメントの問題解決のための支援システムである。この基本的な考え方が DSS の概念を形成している。すなわち、DSS がシステムの技術的な都合でその形態が決定されるのではなく、マネジメントの問題解決の様式の都合によって、その形態が決定されるべきであることを意味する。

従来、DSS と呼ばれてきたシステムは、その多くがデータベース問い合わせ技術や分析手法など、いわば目玉となるソフトウェアの一つを中核として構築されている。マネジメントがこのような手法や技法を中心としたシステムを使用するには、マネジメント側が彼の問題解決の様式をシステムに合わせねばならない。また、これらのシステムでは特定の手法や技法を中心としたシステムであるため、問い合わせ用の DSS や分析用の DSS など複数個のシステムを使い分けることが強要されるのである。

支援を目的とする DSS は、本来、マネジメントの問題解決様式にシステム側が合わせるべきである。ここでは、H. A. Simon による人間の問題解決様式の研究成果をもとに、DSS に要求されるシステムの基本形態を考察する。

Simon は、人間の問題解決が次の三つの過程からなることを見出した<sup>[5]</sup>。

- 1) 発見過程 (intelligence phase)……問題を発見し定義するための情報活動の過程
- 2) 設計過程 (design phase)……問題を解決するためのいくつかの代替案を作成する過程
- 3) 選択過程 (choice phase)……代替案を比較検討し、一つの代替案を選択する過程

問題解決の過程は、最初から、このように三つの過程を明確に認識できるわけではない。初期の段階では、これらが相互にオーバーラップしている。問題が徐々に明確になるにつれて、これらの過程は鮮明になる。

半構造的なあるいは非構造的な問題を解決するとき、最初から一つの問題の全体に対処するのではなく、人間はより小さな副問題 (subproblem) に分解し、問題解決を図ろうとする。すなわち、問題を発見し定義する発見過程は、よりレベルの低い発見過程と設計過程と選択過程からなる。たとえば、問題を定義する発見過程では、問題を発見するための情報活動を行い、つぎにいくつかの候補となる問題を明確にし (代替問題)、その中から解決すべき一つの問題を選択するのである。

一つの問題をいくつかの副問題に分け、その副問題をさらに小さな副問題に分けるといふ分解行為は、副問題が構造的な副問題、すなわち直接解くことができるレベルの副問題、になるまで繰り返される。また、副問題の解決がなされたのであれば、もとの上位の副問

題の解決を図るといふ総合行為を行う。この分解と総合が繰り返行われて、最終的な問題解決がなされる。

この分解と総合は非常に人間的な行為であり、各個人によって大きな差異がある。この分解と総合には、人間がもつ洞察力・分析力・直観・経験が有効に作用し、その大小によって解決できる問題の範囲が決まる。

直接、解決できる構造的な副問題とは、人間がかつて学習したことのある問題解決様式や問題解決の手法が直接適用できるレベルの問題である。これも各個人によって、そのレベルは異なる。たとえば財務分析をする場合、多変量解析に精通した人であれば有用な総合的指標を作り出し分析できるが、そうでない人は若干の財務指標を四則演算で求め、分析する程度にとどまる。

コンピュータがマネジメントを支援できるのは、この構造的な副問題の問題解決における人間による機械的な情報処理の作業である<sup>(註5)</sup>。この情報処理には、ファイル検索・予測・分析・分類・データ加工・作表・作図などの情報処理の手法や技法が使用される。

問題解決は、1個以上の構造的な副問題の解決の集積からなることをすでに指摘した。問題が解決されたとき、そこには試行錯誤の軌跡として、副問題の解決からなる一つの連鎖とみることができる。この連鎖を、ここでは意思決定支援モデルと呼ぶことにする。この意思決定支援モデルは、副問題の解決に使用された情報処理手法・技法の組み合わせによって形成されている。これら情報処理手法・技法は意思決定支援モデルを形成する要素となっていることから、ここではそれらをモデル・ユニットと呼ぶ。

人間は問題解決をするとき、部分的にでも、ある処理手順を言葉を使用して頭に浮かべたから行う。たとえば、パレート図を描いて問題点を探るのであれば、次のように頭に浮かべるであろう。前月の売上実績を“抜き出して(ファイル検索)”，その実績の大きい順に販売店を“並べ(分類)”，実績の累計を“計算し(データ加工)”，それを折れ線“グラフ”にしてみる。この例でもわかるように、人間はキー・ワードを使って処理手順を頭に描く。なお、このキー・ワードの存在は Keen によっても指摘されている<sup>[6]</sup>。

モデル・ユニットの設定単位は、構造的な副問題に対応するレベルのものであるが、これはモデル・ユニット自身への接近の容易性を大きく左右する要因である。モデル・ユニットの名前は、人間とコンピュータとの間のコミュニケーションに使われる言葉である。人間がモデル・ユニットへ容易に接近できるようになるためには、モデル・ユニットの機能と名前が、意思決定支援モデルを作るうえでのキー・モード相当のものでなければならぬ。すなわち、モデル・ユニットの名前が、そのモデル・ユニットの内容を容易に連想させる一般的なものであることが要求される。そして、このことは、一つのモデル・ユニットが一つの言葉で表現できる唯一の情報処理手法・技法の機能をもつことが適切であることを意味する。

支援を目的とする DSS パッケージは、この条件を満たすモデル・ユニットを豊富に用意し、マネジメントに任意なモデル・ユニットへの接近を可能にして、彼独自の意思決定支援モデルの構築をさせるシステムになるものと考えられる。

## 6. DSS 1100-J における実際

具体的な DSS パッケージとして、日本ユニパック(株)の DSS 1100-J を取り上げ、以上論じてきた DSS の基本形態がどのように実現されているかを、具体的に記述する。

DSS 1100-J の構成は、図 4 に示すように、モデル・ユニット、モデル・ユニットを保

存するモデル・ベース，大量データを保存するデータベース，少量データを保存するパーソナル・ファイル，そして作業に使用するワークスペースから構成される。

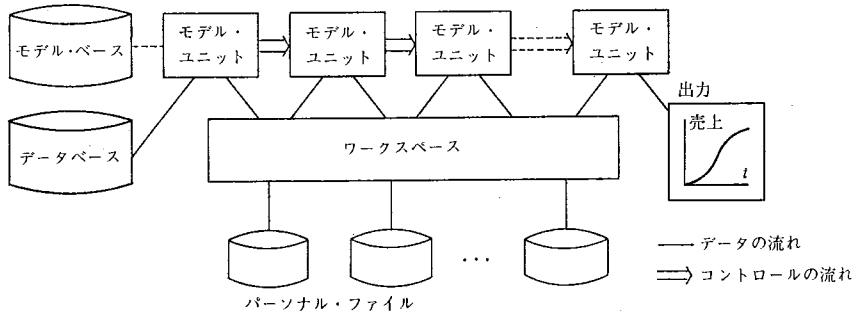


図 4 DSS 1100-J の構成

Fig. 4 Architecture of DSS 1100-J

DSS 1100-J は，この他，データベースのディクショナリとしてのメタ情報や，会話や出力を司る基本要素を有するが，紙幅の関係から，本稿ではデータ関係とモデル関係についてのみ説明する．なお，ここでは，DSS 1100-J の基本形態が，データ指向の DSS にもモデル指向の DSS にも適合することを強調する。

7. データベース

DSS 1100-J においては，次の二つの形式のデータを取り扱う。

- 1) CODASYL 型データベース……共用データベース
- 2) 表形式の独自のファイル……ワークスペース，パーソナル・ファイル

共用データベースでは大量データの管理と効率的な検索，さらにデータの一元管理と共同利用が要求されるため，DSS 1100-J では共用データベースを実現するために CODASYL 型データベースを採用した。

CODASYL 型データベースで 3 次元以上の経営情報を表現するために，次のような配慮を行っている．図 5 に示すように，ビジネス単位を一つのレコード・タイプ，ビジネス変数を下位の別なレコード・タイプとし，その間をセット関係で結んでいる．そして，時系列方向をビジネス変数のレコード・タイプの中に入れた。

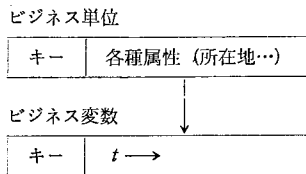


図 5 CODASYL 型データベースによる経営情報の表現

Fig. 5 Representation of management information under CODASYL database

キー	計画 (平残)	実績 (平残)	実績 (未残)
	t →	t →	t →

図 6 属性

Fig. 6 Attributes

また，経営情報が計画と実績に分かれている場合や，データの計測単位が金額表示や物

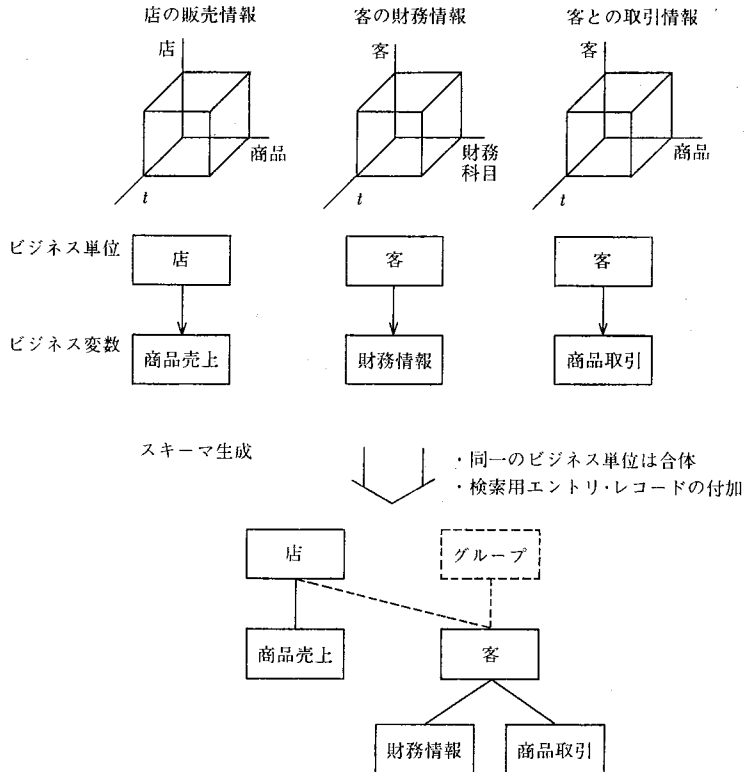


図 7 スキーマ生成

Fig. 7 Creation of a database schema.

量表示のものが混在している場合、図 6 のようにビジネス変数のレコード・タイプの中をいくつかの属性に分けてもつようになっている。

一つの企業を取り上げても、経営情報は各種の分野のものから構成される。CODASYL 型データベースでは、これらを一つのスキーマのもとで表現するわけであるが、そのスキーマの生成は具体的に次のように行う。

たとえば、銀行の営業店管理と主要顧客管理を取り上げた場合、経営情報は図 7 のように三つの立方体から構成される。それぞれの立方体は、二階層のレコードで表現される。スキーマを生成する上で、同一のビジネス単位をもつレコード・タイプが複数存在する場合、それらを一つのレコード・タイプに合体する。つぎに、検索上、異なったビジネス単位間に有用な関係があれば、それらのレコード・タイプ間にセット関係を設定する。また、ビジネス単位のレコード・タイプの上に任意なレコード・タイプを付加して分類などのために使用してもよい。このようにして、スキーマは作成され、経営情報の間に有機的な関係が自然に理解できる形で設定され、使用に供されるようにした。

## 8. ワークスペースとパーソナル・ファイル

データベース上の断面を抽出し、ワークスペース上で加工をし、必要に応じてパーソナル・ファイルにデータを保存するという利用形態が、DSS 1100-J の一つの基本的な利用形式である。これに対し、もう一つの基本的な利用形式は、エンド・ユーザ自身がワークスペースにデータをキー・ボードから入力し、それを必要に応じてパーソナル・ファイルに保存し、後に再使用するという形態である。なお、これら二つの利用形式の混合も可能

である。

エンド・ユーザ自身がデータを入力し、加工を行うためには、通常のオフィスにおける書類の取り扱いと同様な感覚で作業を行えることが必要で、そのためにはデータの見通しやすさと操作のよさが重要となる。これらの要請を満足するために、DSS 1100-J では表形式の特別なファイル構造をワークスペースとパーソナル・ファイルに採用した。

ワークスペースとパーソナル・ファイルは基本的に同一の形式のファイルで、いずれも以下に説明する複数の表を保持することができる。一つの表は複数の列からなるものをいう。一つの表は、図8に示すように、表側、表頭、データおよびデータ名から構成される。表には、時系列表と一般表がある。時系列表は、連続した時系列データを保持する表であり、一般表は時系列以外のデータを保持する表である。

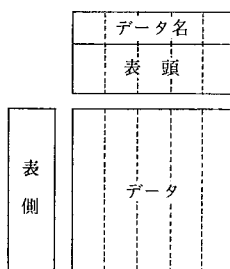


図 8 表の形式

Fig. 8 Structure of a table

表側と表頭には、次のいずれかの項目が入る。

- 1) ビジネス単位のキーあるいはそのデコード情報
- 2) ビジネス変数のキーあるいはそのデコード情報
- 3) 属性コードあるいはそのデコード情報、たとえば実績値（平均残高）、計画値（期末残高）
- 4) 時間

ただし、時系列表では、時間が表側となり、それ以外の項目が表頭となる。表頭には列を一意に決定する基本項目だけでなく、グループ項目を入れることが可能である。表の取り扱いを容易にし、表への列の追加と抽出を自由に行うことができるようにするために、表の各列にはデータ名が付けられる。四則演算などの加工はこのデータ名によって行う。行方向のデータを取り出すために、時系列表では時間を、一般表では、行番号が使用される。

DSS 1100-J のワークスペースは使用時に割り当てられる一時的ファイルである。一方、パーソナル・ファイルは個人ごとに設定できるだけでなく、部門別にいくつかの共同利用ファイルとしても設定できるカタログ・ファイルである。とくに、パーソナル・ファイルは十分確定していないデータや小量データを保存して、DSS を試行し始めるために使える有用なファイルである。

## 9. モデル・ユニット

DSS 1100-J におけるモデル・ユニットには、次の3種類のものがある。

- 1) システム・モデル・ユニット
- 2) レポート・ライター
- 3) ユーザ・モデル・ユニット



システム・モデル・ユニットは、DSS 1100-J にあらかじめ含まれている標準的なモデル・ユニットで、次のモデリング・プロトコルに従って、作成されている。

- 1) 一つのモデル・ユニットは、意思決定支援モデルを構築するうえでの、一つの機能から構成される。
- 2) モデル・ユニットは相互に独立である。
- 3) モデル・ユニットの入出力は基本的にワークスペースに対して行う。
- 4) モデル・ユニットの会話および出力は、DSS 1100-J の基本要素である会話モジュールおよび出力モジュールを通して行う。

レポート・ライタは、データベースにあるデータをもとに定形的な帳表を作成するプログラムで、データベース検索、加工、分類、編集などの複合機能を有する会話型 RPG に相当する。これは、構造化された仕事に対しデータベースを利用して、情報提供を行う MIS を DSS の一側面として実現するために提供されている。

ユーザ・モデル・ユニットは、システム・モデル・ユニットあるいはレポート・ライタで満足されないニーズに対処するために利用者によって作成される。ユーザ・モデル・ユニットは、ある特定の分野における固有なモデルとして認識されるが、これも DSS のモデリング・プロトコルのもとに作成されると、他のモデル・ユニットとの組み合わせにより、その有用性は増す。

システム・モデル・ユニットの一部を表 2 に示す。システム・モデル・ユニットは構造化された副問題を解決するための一つの手法・技法からなるもので、基本的に一つの機能だけをもつ。ただし、加工演算ユニットと表操作ユニットは、使用者が演算式を記述し新たな項目を作ったり各種の表操作をして、新たな表を作り出すためにその内部に手順記述機能をもつ。

表 2 システム・モデル・ユニットの例  
Table 2 An example of system model units

区 分	システム・モデル・ユニット
データベース検索	データベース断面検索 データベース統計検索
予 測	指数平滑法予測 センサス周法予測 伸び率予測
分 析	重回帰分析
加工演算	四則演算, 比較演算, 論理演算
表 操 作	結合, 選択, 転置, 分類
作 表	報告書作成
作 図	棒グラフ 折線グラフ レーダ・チャート
外部データ入力	マニュアル・データ入力 マニュアル・データ更新
パーソナル・ファイル処理	表 格 納 表 復 帰

使用者が DSS 1100-J を使い作業をするためには、まずデータベース断面抽出ユニット、あるいはマニュアル・データ入力ユニット、あるいはパーソナル・ファイルの表復帰ユニットを使用して、ワークスペースにデータを表形式でもち込む。なお、データベース断面抽出ユニットは、データベースについて知識のない使用者にも大規模データベースか

ら任意の断面が容易にかつ効率的に検索できるように工夫されたモデル・ユニットであり、DSS 1100-J における最も基本的で重要なユニットである。このモデル・ユニットは、ワークスペースへ表を出力するだけでなく、確認のために表を CRT 上にも出力することができるため、データを問い合わせるだけであれば、このモデル・ユニットを使用するだけでよい。

ワークスペース上に取り出された 1 個以上の表のデータをもとに、予測・分析・加工・表操作などのモデル・ユニットを用いて、データを有用な情報へ変換することができる。このような作業の途中あるいは最後に、必要に応じて、作表あるいは作図ユニットを使用して、一覧表やグラフを作成し、問題の分解で総合に役立てることができる。使用者は作業の中断や再開のために、パーソナル・ファイルへ中間的な表を格納したり、ワークスペースへそれを復帰するために表格納・表復帰ユニットを使用することができる。

問題解決は、何度も行われると学習効果が現われ、その解決様式がしだいに洗練され、ついには、その解決様式が構造化される。すなわち、モデル・ユニットの使い方とその組み合わせが決定されるという進化の側面がある。この構造化に対応するために、DSS 1100-J では、各モデル・ユニットの使い方とモデル・ユニットの組み合わせの手順を一つの名前で登録することが許される。構造化しつつある、あるいは構造化したものに対しては、一つの名前と若干の変動的なパラメタを与えるだけで同一の結果が得られる。

図 9 のような形式的に同一な複数の表を一度に処理する繰り返し処理が、基本的にすべてのシステム・モデル・ユニットに組み込まれている。たとえば、ビジネス単位だけが異なる複数の表を、データベース断面抽出ユニットで一度にワークスペースに移し、東京の表に対して行った処理を名古屋、大阪についても適用する。これは、使用者が最初の表に対して行った処理をシステムが 2 枚目以降の表に同一の処理を自動的に適用することである。この繰り返し処理機能により、使用者は大量のデータを効率的に処理することができる。

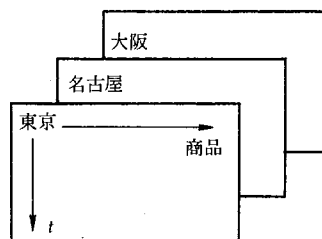


図 9 繰り返し処理

Fig. 9 Repeat processing

## 10. DSS 1100-J の適用範囲

DSS 1100-J は固有な問題から独立した基本形態を有する。この基本形態のもとでは、一つのシステムで Alter がいうところのデータ指向の DSS とモデル指向の DSS とが実現できる。すなわち、ファイル検索システムとして、データベース断面抽出ユニットとレポート・ライターがその機能を果たし、分析的情報システムとしてデータベースと各種システム・モデル・ユニットが有効である。また、会計モデルとしては加工演算ユニットの手順記述機能でモデルを記述しパラメタを変えて“what if”分析などを行うことができる。さらに、最適化モデルの DSS を実現するには線形計画手法をユーザ・モデル・ユニ

ットとして作成し、他のモデル・ユニットとともにそれを有効に作動させることができる。このように Alter が分類した各種の DSS を DSS 1100-J のもとで作動させることができるだけでなく、それらと使用者の間の相互作用は DSS 1100-J の会話機能を通して統一的行われる。このため、DSS 1100-J のもとでは複数のシステムを使い分ける必要がなく、マネジメント自らがシステムを操作し、多様な問題に対処することが可能である。

## 11. 使用例

ここでは具体的な意思決定問題を取り上げ、DSS 1100-J を使用してマネジメントが意思決定支援モデルを作り上げる様子を、マネジメントの思考に沿って、モデル・ユニットとデータとの間の有機的な動きで示す。

販売店の要員計画を担当するマネジメントを想定する。彼は、すべての販売店に適正な要員を配置する責任がある。年度末(1982年度)が近づいたため、来年度の要員計画を策定することとした。ただし、要員の増員にはきびしい制限がある。

そのため、彼は要員不足の大きい販売店に重点的に人材を配置したいと考えた。この問題を解決するために、次のような指針を採用することとした。

- 1) 必要人員数は、売上高に比例するものとする。来年度の売上高は、実績から予測する。
- 2) 1人当たり売上高は、次式で求める。ただし、来期は1人当たり売上高は10パーセント程度向上するものと仮定する。

$$1 \text{ 人当たり売上高} = \frac{\text{全販売店の売上合計}}{\text{全販売店の人数合計}}$$

- 3) 表示形式はパレート図を用いて、人材不足の大きい販売店の順に表示し、人材不足の状況をみる。
- 4) 売上高実績は販売管理データベースから入手する。各販売店の人数はパーソナル・ファイルに入っているものを利用するものとする。

彼が考えたモデル・ユニットの実行順序に従って、それぞれの作業内容を説明する(図10)。

- 1) 来年度の売上高予想
  - 1-1) 検索……販売管理データベースには、商品別の売上高実績と全商品の売上高合計実績が保存されている。ここでは、全商品の売上高合計実績を全販売店について時系列データで抽出し、作業領域に移す。
  - 1-2) 予測……指数平滑法を用いて、この先き1年間(12か月分)の予測を全販売店に対し行う。
  - 1-3) 転置……つぎに、四則演算を行うために表を転置する。四則演算は表の列単位に行う。転置して得る表には1982年3月の実績値と1983年3月の予測値を入れる。
- 2) 配属済みの人数を読み込む
  - 2-1) 表復帰……パーソナル・ファイルに入っている各販売店の人数を作業領域に移す。
- 3) 不足人数の算出
  - 3-1) 加工演算……不足人数を求めるために、1人当たり売上高を1982年3月の実績値を用いて算出する。1983年3月の予想売上高をいま求めた1人当たり売上高で

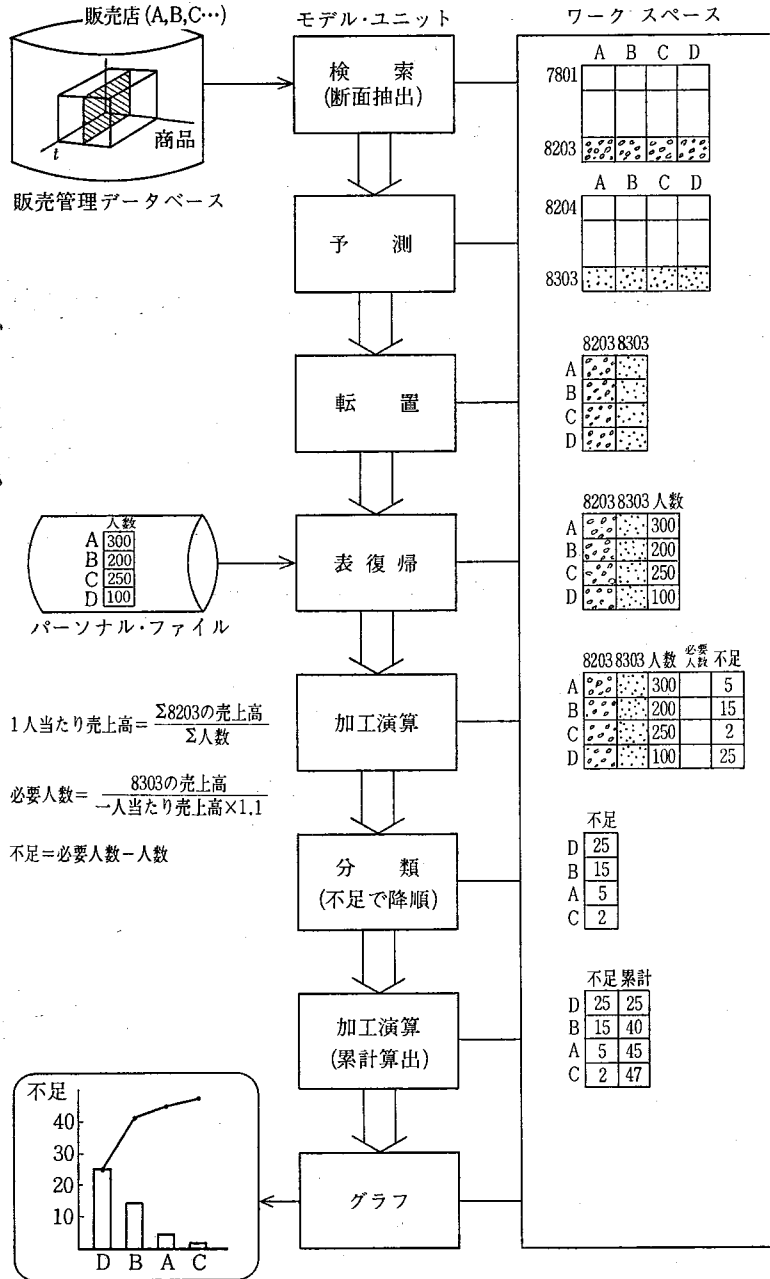


図 10 意思決定支援モデルの構築過程

Fig. 10 Building process of a decision support model

割り、必要人数を求める。ただし、この段階で生産性の向上を10パーセントと仮定している。最後に不足人数を配属済みの人数との差から求める。

4) パレートの作成

- 4-1) 分類……人員を重点的に増員するために、不足人数の大きいもの順に表の内容を分類（ソート）する。
- 4-2) 演算加工……パレートを作成するために、不足人数の累計を求める。
- 4-3) グラフ……不足人数を棒グラフ、累積不足人数を折れ線グラフとして作図する。

以上の結果、場合によっては不足人数の合計が来年度確保できる人数より多いことがあるであろう。このような場合、彼は 3-1) にもどって 1 人当たり売上高の生産性の向上分を大きくして再計算をしたりする。また、場合によっては各販売店から提出されている売上目標額を考慮に入れ、予測売上高との間で少ない方を計算の基礎にして、再評価することもできる。DSS 1100-J を使用することによって、このように、その時々状況を加味し短時間の間にいくつもの代替案を作り、一つの結論を得ることができる。

## 12. おわりに

DSS を成功させるには、システムに柔軟性と使いやすさを求めるだけでは不十分である。DSS は、コンピュータによるオペレーショナルな情報システムと異なり、システム開発の終了とかシステムの本番開始といった仕切りのつけられないシステムである。その構築と運用は、まったく従来のコンピュータによる情報システムとは異なる性格を有する。DSS の成功はマネジメントを継続的に支援し、システムを運営するスタッフの体制と資質にも大きく左右される。本稿は、このように等しく重要な適用技術 (implementation) など、DSS に関する諸々の側面について言及する余裕がなかった。これらについては、著者が広内哲夫氏と共著で、竹内書店新社からこの 4 月に出版予定の「意思決定支援システム：システムの構築と実際」(仮称) を参照していただきたい。

- 参考文献 [1] T. Kosaka, T. Hirouchi, "An Effective Architecture for Decision Support Systems," *Information & Management*, Vol. 5, No. 1, 1982.
- [2] P. G. W. Keen, M. S. Scott Morton, *Decision Support Systems: An Organizational Perspective*, Addison-Wesley, 1978.
- [3] S. L. Alter, *Decision Support Systems: Current Practice and Continuing Challenges*, Addison-Wesley, 1980.
- [4] J. F. Rockart, M. E. Treacy, "The CEO Goes On-line," *Harvard Business Review*, January-February 1982.
- [5] H. A. Simon, *The New Science of Management Decisions*, Revised edition, Prentice-hall, 1977.
- [6] P. G. W. Keen, "Decision Support Systems: Translating Analytic Techniques into Useful Tools," *Sloan Management Review*, Spring, 1980.
- [7] A. M. McCosh, M. S. Scott Morton, *Management Decision Support Systems*, Macmillan Press, 1977.
- [8] 小坂 武, 広内哲夫, "経営情報システムにおけるデータベース", *Computer Report*, Vol. 20, No. 3, 1980.
- [9] 小坂 武, "デシジョン・サポート・システムと OA", *事務と経営*, Vol. 33, No. 408, 1981.

- 執筆者注 (注 1) Simon は意思決定を programmed decision, nonprogrammed decision に分類した。G. A. Gorry と M. S. Scott Morton は、コンピュータとの関連を少なくし、問題の基本的な性格に注目するために、Simon の用語に代えて structured decision と unstructured decision という用語を使っている<sup>[7]</sup>。
- (注 2) 業務システムのデータベースを使うのではなく、マネジメントのために用意されたデータベースを使うことがこの分析的情報システムの特徴であり、コンピュータによる情報システムの側面が強調されている。
- (注 3) Accounting model は次の Representational model との関係から、会計モデルと訳したが、本来は説明モデルと訳するのが適切である。ここでの会計は財務会計ではなく、むしろ物量計算を含む管理会計での会計を意味する。
- (注 4) 共用データベースの必要性については小坂の論文<sup>[8]</sup>に詳しい記述がある。
- (注 5) 現在のコンピュータ・システムは構造的な副問題の解決における情報処理を支援するが、副問題の分解と総合という人間的な側面までも、コンピュータ・システムは知識工学の成果を利用して近い将来支援すると予想される。

**執筆者紹介** 小坂 武 (Takeshi Kosaka)

昭和 22 年生, 45 年上智大学工学部電気電子工学科卒業, 46 年日本ユニバック(株)入社, 現在に至る. 応用ソフトウェア部  
情報検索グループにおいて, 意思決定支援システムのサービス  
などに従事.



## 論文

## 任意節点配置差分法による連続体解析

## Analyses of Continuum by Means of The Finite Difference Methods with Arbitrarily Distributed Nodes

藤野 勉

**要 約** 複雑な形状をもつ連続体の解析には有限要素法が効果的に利用されているが、この方法は要素積分などを含み、計算手続きが多少複雑になるおそれがある。この点を改善するため、ここに任意節点配置による差分法を提案することとする。差分法の方法としては、① Taylor 展開法、② 補間関数法、③ アイソパラメータ法、などが考えられる。これらの方法は2次元、3次元解析ともに利用できるが、ここでは主として2次元差分法について述べることにする。Taylor 展開法についてはすでに発表している<sup>[1]</sup>ので、ここでは簡単に述べることにする。

Taylor 展開法は、任意節点配置差分法では最も正当な方法と思われるが、逆行列の計算を必要とする。しかし、計算法は簡明でプログラム化も最も容易と思われる。②の補間関数法は逆行列の計算を要しない代わりに計算式が多少複雑となり、プログラムの作成も①より容易さを失うものと思われる。③についても同様である。計算精度、時間については、今後の経験を待たなければならぬ。有限要素法との比較についても同様である。

**Abstract** The finite element method (FEM) is effectively used for analyses of continuum, but its calculation procedure and programming are somewhat not easy due to element integration and others. In this respect, the finite difference methods (FDM) are simple and straight-forward compared with the former. Hitherto, the finite difference method with orthogonal linear mesh were used, but it is not applicable for analyses of continuum with complicated boundary shape, while this method with curvilinear lattice is suitable for this means.

This paper is a study on the finite difference methods with arbitrarily distributed nodes which are classified to

- (1) Taylor expansion method
- (2) Interpolation function method
- (3) Isoparametric method.

Calculation process of these methods are shown in this paper. Further studies on calculation accuracy and execution time of these methods and finite element method are necessary.

## 1. Taylor 展開法

$x$ - $y$  座標平面で定義される変数  $u(x, y)$  は、原点  $x=0, y=0$  の近傍において次のように展開される。

$$u(x, y) = \frac{1}{m!n!} \left( \frac{\partial^{m+n} u}{\partial x^m \partial y^n} \right)_0 x^m y^n = a_n f_n(x, y) \quad (1-1)$$

$$a_1 = u, \quad a_2 = u_x, \quad a_3 = u_y, \quad a_4 = u_{xx}, \quad a_5 = u_{xy}, \quad a_6 = u_{yy}, \quad \dots \quad (1-2)$$

$$f_1 = 1, \quad f_2 = x, \quad f_3 = y, \quad f_4 = x^2/2, \quad f_5 = xy, \quad f_6 = y^2/2, \quad \dots \quad (1-3)$$

ここで微係数  $u_x, u_y, \dots$  を評価するため、原点近くの点の情報(変数値)を利用することとする。このように情報を提供する点を節点と呼び、これらは番号によって指定される。便宜上、微分値を評価する原点に番号1を与え、その他に2次精度ならば5個、3次精度ならば9個、4次精度ならば14個の節点を配置する。これらの節点の配置は任意でよいが、正則性を保つためには直交に近い曲線格子系の格子点に置くことが好都合である。

いま節点  $m$  の座標を  $x_m, y_m$  とし、

$$u_m = u(x_m, y_m), \quad f_{mn} = f_n(x_m, y_m) \tag{1-4}$$

とすれば、(1-1) により

$$f_{mn} a_n = u_m \tag{1-5}$$

が導かれ、(1-5) を  $a_n$  について解き、

$$a_m = g_{mn} u_n \tag{1-6}$$

を得る。ここに、 $g_{mn}$  は  $f_{mn}$  の逆行列で、その存在が保証されるためには  $f_{mn}$  が特異でないこと、すなわち  $\det f_{mn} \neq 0$  が満たされていることが必要である。そのためには、図 1 に示されるような正則な節点配置を行うことが要請される。

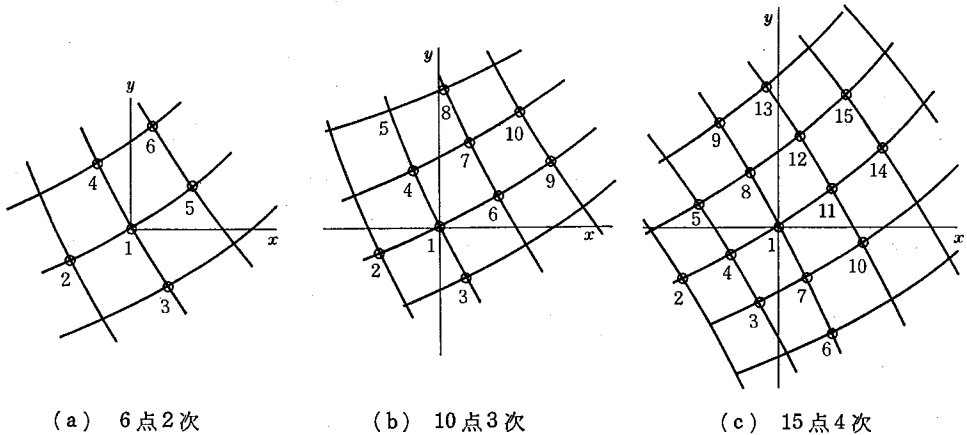


図 1 差分家族節点配置 (1)

Fig. 1 Node distribution of differential family (1)

1) 節点 1 における変数ならびにその微係数值

式(1-2) と (1-6) によって、 $u$  ならびにその微係数の原点における値は次のように与えられる。

$$\begin{cases} a_1 = u \\ a_2 = u_x \\ a_3 = u_y \\ a_4 = u_{xx} \\ a_5 = u_{xy} \\ a_6 = u_{yy} \\ a_7 = u_{xxx} \\ \vdots \end{cases} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ g_{21} & g_{22} & g_{23} & g_{24} & g_{25} & g_{26} & g_{27} & \dots \\ g_{31} & g_{32} & g_{33} & g_{34} & g_{35} & g_{36} & g_{37} & \dots \\ g_{41} & g_{42} & g_{43} & g_{44} & g_{45} & g_{46} & g_{47} & \dots \\ g_{51} & g_{52} & g_{53} & g_{54} & g_{55} & g_{56} & g_{57} & \dots \\ g_{61} & g_{62} & g_{63} & g_{64} & g_{65} & g_{66} & g_{67} & \dots \\ g_{71} & g_{72} & g_{73} & g_{74} & g_{75} & g_{76} & g_{77} & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{cases} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ \vdots \end{cases} \tag{1-7}$$

2) 節点 1 における微分方程式ならびに境界条件式の差分表示

与えられた領域内で満たされる微分方程式、境界条件式を

$$Au + B \frac{\partial u}{\partial x} + C \frac{\partial u}{\partial y} + D \frac{\partial^2 u}{\partial x^2} + E \frac{\partial^2 u}{\partial x \partial y} + F \frac{\partial^2 u}{\partial y^2} + G \frac{\partial^3 u}{\partial x^3} + \dots = f \tag{1-8}$$

とする。ただし、境界上では 1 次微係数まで含む線形条件式で

$$D=0, E=0, F=0, \dots \quad (s \text{ 上で}) \tag{1-9}$$

がある。これらの係数は一般に  $x, y$  の関数である。(1-8) は (1-7) により、次のように差分表示される。

$$k_{mu} u_m = f \tag{1-10}$$



ただし,

$$\begin{pmatrix} k_1 \\ k_2 \\ k_3 \\ k_4 \\ k_5 \\ k_6 \\ k_7 \\ \vdots \end{pmatrix} = \begin{pmatrix} 1 & g_{21} & g_{31} & g_{41} & g_{51} & g_{61} & g_{71} \cdots \\ 0 & g_{22} & g_{32} & g_{42} & g_{52} & g_{62} & g_{72} \cdots \\ 0 & g_{23} & g_{33} & g_{43} & g_{53} & g_{63} & g_{73} \cdots \\ 0 & g_{24} & g_{34} & g_{44} & g_{54} & g_{64} & g_{74} \cdots \\ 0 & g_{25} & g_{35} & g_{45} & g_{55} & g_{65} & g_{75} \cdots \\ 0 & g_{26} & g_{36} & g_{46} & g_{56} & g_{66} & g_{76} \cdots \\ 0 & g_{27} & g_{37} & g_{47} & g_{57} & g_{67} & g_{77} \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} A \\ B \\ C \\ D \\ E \\ F \\ G \\ \vdots \end{pmatrix} \quad (1-11)$$

である.

式(1-11)の係数行列は  $g_{mn}$  の転置行列である. このように微分方程式ならびに境界条件式は, ただちに差分化される.

### 1.1 2次展開による2階微分方程式の差分化

2階の微分方程式を解くためには, 少なくとも2次展開が必要である. いま領域内で満たされる線形微分方程式を

$$Au + B \frac{\partial u}{\partial x} + C \frac{\partial u}{\partial y} + D \frac{\partial^2 u}{\partial x^2} + E \frac{\partial^2 u}{\partial x \partial y} + F \frac{\partial^2 u}{\partial y^2} = f \quad (1-12)$$

とし, 境界上で満たされる線形条件式を一般に

$$Au + B \frac{\partial u}{\partial x} + C \frac{\partial u}{\partial y} = f \quad (1-13)$$

とする. ここで, 係数  $A, B, C, \dots, f$  は一般に  $x, y$  の関数である. 条件式(1-13)は一般に自由境界の条件式を与えるが, とくに  $A=1, B=C=0, f=\bar{u}$  とおけば, 固定境界条件式となる. 計算手順は次のとおりである.

- 1) 境界に適合する直交に近い曲線格子系をつくり, その格子点に節点を置き付番する.
- 2) 節点座標  $x_i, y_i$  を読み取る.
- 3) 差分家族を定める. (ここでは, 2次精度であるから, 微分評価節点  $i$  を中心として6個の節点群で  $i$  差分家族を構成する.  $i$  節点が領域内にあるときと境界上にあるときの節点配置は, 図2に示す.)

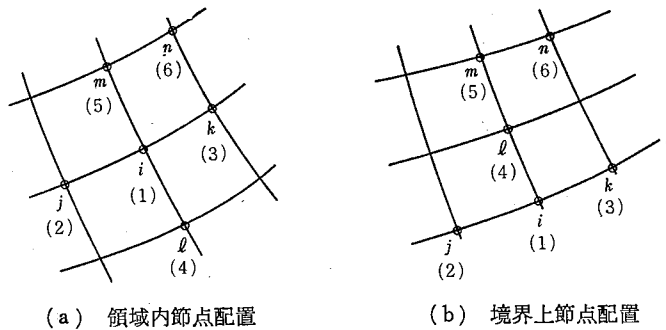


図2 差分家族節点配置 (2)

Fig. 2 Node distribution of differential family (2)

#### 4) 局所座標

$$\begin{aligned} x_1^i &= 0, x_2^i = x_j - x_i, x_3^i = x_k - x_i, x_4^i = x_l - x_i, x_5^i = x_m - x_i, x_6^i = x_n - x_i \\ y_1^i &= 0, y_2^i = y_j - y_i, y_3^i = y_k - y_i, y_4^i = y_l - y_i, y_5^i = y_m - y_i, y_6^i = y_n - y_i \end{aligned} \quad (1-14)$$

$$5) \quad f_{mn}^i = f_n(x_m^i, y_m^i) \quad (1-15)$$

$$6) \quad f_{mn}^i \text{ の逆行列: } g_{mn}^i \quad (1-16)$$

7) 係数行列

$$\begin{Bmatrix} k_{ii} \\ k_{ij} \\ k_{ik} \\ k_{il} \\ k_{im} \\ k_{in} \end{Bmatrix} = \begin{bmatrix} 1 & g_{21}^i & g_{31}^i & g_{41}^i & g_{51}^i & g_{61}^i \\ 0 & g_{22}^i & g_{32}^i & g_{42}^i & g_{52}^i & g_{62}^i \\ 0 & g_{23}^i & g_{33}^i & g_{43}^i & g_{53}^i & g_{63}^i \\ 0 & g_{24}^i & g_{34}^i & g_{44}^i & g_{54}^i & g_{64}^i \\ 0 & g_{25}^i & g_{35}^i & g_{45}^i & g_{55}^i & g_{65}^i \\ 0 & g_{26}^i & g_{36}^i & g_{46}^i & g_{56}^i & g_{66}^i \end{bmatrix} \begin{Bmatrix} A_i \\ B_i \\ C_i \\ D_i \\ E_i \\ F_i \end{Bmatrix} \quad (1-17)$$

ただし,  $A_i = A(x_i, y_i)$  (以下  $B_i, C_i, D_i, E_i, F_i$  同様)

$$8) \text{ 連立方程式} \quad k_{ij}u_j = f_i \quad (1-18)$$

## 1.2 2次展開による2変数, 2階微分方程式の差分化

二つの変数  $u_1 = u, u_2 = v$  に関する微分方程式ならびに境界条件式を一般に,

$$\begin{aligned} A^{11}u + B^{11}u_x + C^{11}u_y + D^{11}u_{xx} + E^{11}u_{xy} + F^{11}u_{yy} \\ + A^{12}v + B^{12}v_x + C^{12}v_y + D^{12}v_{xx} + E^{12}v_{xy} + F^{12}v_{yy} = f^1 \end{aligned} \quad (1-19)$$

$$\begin{aligned} A^{21}u + B^{21}u_x + C^{21}u_y + D^{21}u_{xx} + E^{21}u_{xy} + F^{21}u_{yy} \\ + A^{22}v + B^{22}v_x + C^{22}v_y + D^{22}v_{xx} + E^{22}v_{xy} + F^{22}v_{yy} = f^2 \end{aligned} \quad (1-20)$$

とする。ただし境界上では  $D^{pq} = E^{pq} = F^{pq} = 0$  とする。ここに  $p, q$  は1または2を表わす。これらの係数は一般に  $x, y$  の関数である。

上の微分方程式ならびに境界条件式(1-19), (1-20)は, 図2に示す差分家族について, 下記のように差分表示される。

$$\begin{aligned} k_{ii}^{11}u_i + k_{ij}^{11}u_j + k_{ik}^{11}u_k + k_{il}^{11}u_l + k_{im}^{11}u_m + k_{in}^{11}u_n \\ + k_{ii}^{12}v_i + k_{ij}^{12}v_j + k_{ik}^{12}v_k + k_{il}^{12}v_l + k_{im}^{12}v_m + k_{in}^{12}v_n = f_i^1 \end{aligned} \quad (1-21)$$

$$\begin{aligned} k_{ii}^{21}u_i + k_{ij}^{21}u_j + k_{ik}^{21}u_k + k_{il}^{21}u_l + k_{im}^{21}u_m + k_{in}^{21}u_n \\ + k_{ii}^{22}v_i + k_{ij}^{22}v_j + k_{ik}^{22}v_k + k_{il}^{22}v_l + k_{im}^{22}v_m + k_{in}^{22}v_n = f_i^2 \end{aligned} \quad (1-22)$$

上式は,  $u_i, v_i$  に関する連立方程式であり, 置き換え  $x_{2i-1} = u_i, x_{2i} = v_i, k_{2i-1, 2i-1} = k_{ii}^{11}, k_{2i-1, 2j-1} = k_{ij}^{11}, \dots$  を行うことにより, 次の連立方程式に変換される。

$$k_{ij}x_j = f_i \quad (1-23)$$

$$\begin{Bmatrix} k_{ii}^{pq} \\ k_{ij}^{pq} \\ k_{ik}^{pq} \\ k_{il}^{pq} \\ k_{im}^{pq} \\ k_{in}^{pq} \end{Bmatrix} = \begin{bmatrix} 1 & g_{21}^i & g_{31}^i & g_{41}^i & g_{51}^i & g_{61}^i \\ 0 & g_{22}^i & g_{32}^i & g_{42}^i & g_{52}^i & g_{62}^i \\ 0 & g_{23}^i & g_{33}^i & g_{43}^i & g_{53}^i & g_{63}^i \\ 0 & g_{24}^i & g_{34}^i & g_{44}^i & g_{54}^i & g_{64}^i \\ 0 & g_{25}^i & g_{35}^i & g_{45}^i & g_{55}^i & g_{65}^i \\ 0 & g_{26}^i & g_{36}^i & g_{46}^i & g_{56}^i & g_{66}^i \end{bmatrix} \begin{Bmatrix} A_i^{pq} \\ B_i^{pq} \\ C_i^{pq} \\ D_i^{pq} \\ E_i^{pq} \\ F_i^{pq} \end{Bmatrix} \quad (1-24)$$

ただし, 境界上では  $D_i^{pq} = 0, E_i^{pq} = 0, F_i^{pq} = 0$  とする。たとえば, 平板の面内変形問題では  $u, v$  を  $x, y$  方向変位成分として, 微分方程式ならびに係数行列はそれぞれ,

$$u_{1xx} + \alpha u_{1yy} + \beta v_{1xy} = f^1, \quad \beta u_{1xy} + \alpha v_{1xx} + v_{1yy} = f^2 \quad (1-25)$$

$$\begin{aligned} k_{ii}^{11} = g_{41}^i + \alpha g_{61}^i & \quad k_{ii}^{12} = k_{ii}^{21} = \beta g_{51}^i & \quad k_{ii}^{22} = \alpha g_{41}^i + g_{61}^i \\ k_{ij}^{11} = g_{42}^i + \alpha g_{62}^i & \quad k_{ij}^{12} = k_{ij}^{21} = \beta g_{52}^i & \quad k_{ij}^{22} = \alpha g_{42}^i + g_{62}^i \\ k_{ik}^{11} = g_{43}^i + \alpha g_{63}^i & \quad k_{ik}^{12} = k_{ik}^{21} = \beta g_{53}^i & \quad k_{ik}^{22} = \alpha g_{43}^i + g_{63}^i \\ k_{il}^{11} = g_{44}^i + \alpha g_{64}^i & \quad k_{il}^{12} = k_{il}^{21} = \beta g_{54}^i & \quad k_{il}^{22} = \alpha g_{44}^i + g_{64}^i \\ k_{im}^{11} = g_{45}^i + \alpha g_{65}^i & \quad k_{im}^{12} = k_{im}^{21} = \beta g_{55}^i & \quad k_{im}^{22} = \alpha g_{45}^i + g_{65}^i \\ k_{in}^{11} = g_{46}^i + \alpha g_{66}^i & \quad k_{in}^{12} = k_{in}^{21} = \beta g_{56}^i & \quad k_{in}^{22} = \alpha g_{46}^i + g_{66}^i \end{aligned} \quad (1-26)$$

と表わされる。ただし,  $\nu$  を Poisson 比として  $\alpha = (1-\nu)/2, \beta = (1+\nu)/2$  となる。

このとき、自由境界および固定境界の条件式は以下のとおりである。

固定境界条件式  $u = \bar{u} = f^1, v = \bar{v} = f^2$  (1-27)

自由境界条件式  $\begin{cases} \sigma_x = l_x(u_x + \nu v_y) + l_y \alpha(u_y + v_x) = \bar{\sigma}_x = f^1 \\ \sigma_y = l_x \alpha(u_y + v_x) + l_y(\nu u_x + v_y) = \bar{\sigma}_y = f^2 \end{cases}$  (1-28)

上式 (1-27), (1-28) を差分表示すると、

固定境界  $A_i^{11} = 1, A_i^{22} = 1$  (1-29)

自由境界  $\begin{cases} B_i^{11} = l_{xi}, C_i^{11} = \alpha l_{yi}, B_i^{12} = \alpha l_{yi}, C_i^{12} = \nu l_{xi} \\ B_i^{21} = \nu l_{yi}, C_i^{21} = \alpha l_{xi}, B_i^{22} = \alpha l_{xi}, C_i^{22} = l_{yi} \end{cases}$  (1-30)

となる。

1.3 1 変数, 非線形 2 階微分方程式の差分化

変数  $u(x, y)$  に関する 2 次の非線形微分方程式

$$Au + Bu_x + Cu_y + Du_{1xx} + Eu_{1xy} + Fu_{1yy} + Gu^2 + Huu_x + I + Gu^2 + Huu_x + Iuu_y + Ju_{1x}^2 + Ku_{1x}u_y + Lu_{1y}^2 = f$$
 (1-31)

とする。この式の差分表示は下記のように表わされる。

$$\begin{aligned} &k_{ii}u_i + k_{ij}u_j + k_{ik}u_k + k_{il}u_l + k_{im}u_m + k_{in}u_n \\ &+ G_i u_i^2 + H_i u_i u_{xi} + I_i u_i u_{yi} + J_i u_{xi}^2 + K_i u_{xi} u_{yi} + L_i u_{yi}^2 = f_i \end{aligned}$$
 (1-32)

ただし、 $u_{xi}, u_{yi}$  は下記のとおりである。

$$\begin{aligned} u_{xi} &= g_{21}^i u_i + g_{22}^i u_j + g_{23}^i u_k + g_{24}^i u_l + g_{25}^i u_m + g_{26}^i u_n \\ u_{yi} &= g_{31}^i u_i + g_{32}^i u_j + g_{33}^i u_k + g_{34}^i u_l + g_{35}^i u_m + g_{36}^i u_n \end{aligned}$$
 (1-33)

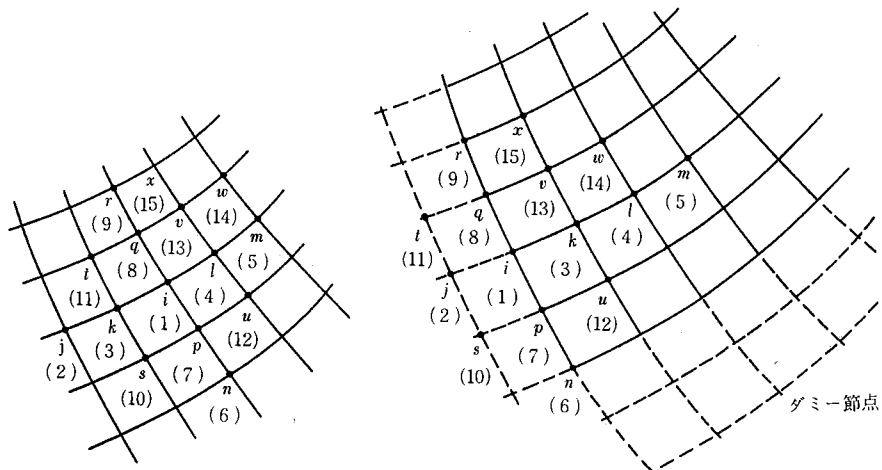
この非線形方程式は単純反復法、Newton-Raphson 法などにより求解することができる。

1.4 4 階微分方程式の差分化

4 階微分方程式ならびにその境界条件式を一般に、

$$Au + Bu_x + Cu_y + Du_{xx} + Eu_{xy} + Fu_{yy} + Gu_{xxx} + Hu_{xxy} + Iu_{xyy} + Ju_{yyy} + Ku_{xxx} + Lu_{xxy} + Mu_{xyy} + Nu_{yyy} + Pu_{yyy} = f$$
 (1-34)

とする。ただし、境界上では  $K=L=M=N=P=0$  とする。なお、境界では 2 または 3 個の条件式が満たされ、それに応じて 1 または 2 個のダミー節点を設ける必要がある。



(a) 領域内 (b) 固定, 支持境界上

図 3 15 節点 4 次差分家族構成

Fig. 3 Fifteen nodes, fourth order differential family

1.4.1 差分家族節点配置

上記の場合の差分家族節点配置例を図3に示す。

以下、計算手順を示す。

1)

行列:  $f_{mn}^i$

$$\begin{aligned} f_1=1, f_2=x, f_3=y, f_4=x^2/2, f_5=xy, f_6=y^2/2, \\ f_7=x^3/6, f_8=x^2y/2, f_9=xy^2/2, f_{10}=y^3/6, \\ f_{11}=x^4/24, f_{12}=x^3y/6, f_{13}=x^2y^2/4, f_{14}=xy^3/6, f_{15}=y^4/24 \end{aligned} \tag{1-35}$$

2)

$$f_{mn}^i = f_n(x_m^i, y_m^i) \tag{1-36}$$

3)

$$f_{mn}^i \text{ の逆行列: } g_{mn}^i \tag{1-37}$$

4) 節点  $i$  における微係数ならびに係数行列を求める。

$$\begin{pmatrix} u^i \\ u_x^i \\ u_y^i \\ u_{xx}^i \\ u_{xy}^i \\ u_{yy}^i \\ u_{xxx}^i \\ u_{xxy}^i \\ u_{xyy}^i \\ u_{yyy}^i \\ u_{xxxx}^i \\ u_{xxx}^i \\ u_{xxy}^i \\ u_{xyy}^i \\ u_{yyy}^i \end{pmatrix} = [g_{mn}^i] \begin{pmatrix} u_i \\ u_j \\ u_k \\ u_l \\ u_m \\ u_n \\ u_p \\ u_q \\ u_r \\ u_s \\ u_t \\ u_u \\ u_v \\ u_w \\ u_x \end{pmatrix} \tag{1-38}$$

$$\begin{pmatrix} k_{ij} \\ k_{ij} \\ k_{ik} \\ k_{il} \\ k_{im} \\ k_{in} \\ k_{ip} \\ k_{iq} \\ k_{ir} \\ k_{is} \\ k_{it} \\ k_{iu} \\ k_{iv} \\ k_{iw} \\ k_{ix} \end{pmatrix} = [g_{mn}^i]^T \begin{pmatrix} A_i \\ B_i \\ C_i \\ D_i \\ E_i \\ F_i \\ G_i \\ H_i \\ I_i \\ J_i \\ K_i \\ L_i \\ M_i \\ N_i \\ P_i \end{pmatrix} \tag{1-39}$$

5) 連立方程式

$$k_{ij}u_j = f_i \tag{1-40}$$

を計算する。たとえば、2重 Laplace の方程式では、 $K=1, M=2, P=1$  とおけばよい。

1.4.2 計算精度の検討

一様流れ中に置かれた円柱周りポテンシャル流れについて、差分法における節点配置ならびに有限要素法との精度比較を行った (図4, 5, 6, 7, 8, 9, 10)。

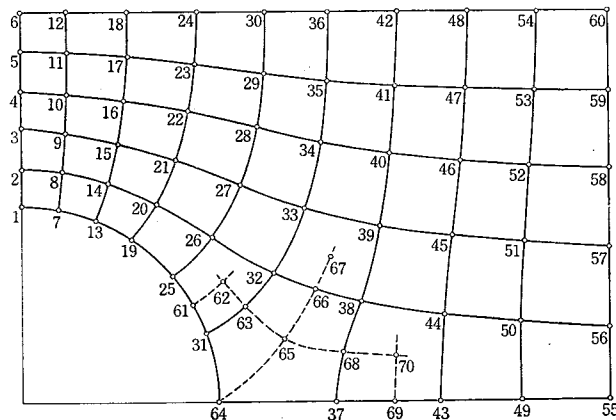


図4 節点配置 (理論値:  $\phi = y - y/(x^2 + y^2)$ )

Fig. 4 Node distribution

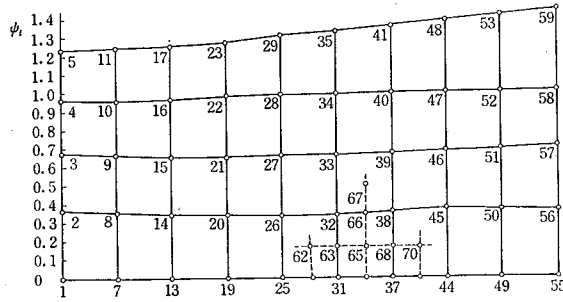


図 5 流れ関数  $\phi$  の節点値

Fig. 5 Node value of stream function  $\phi$

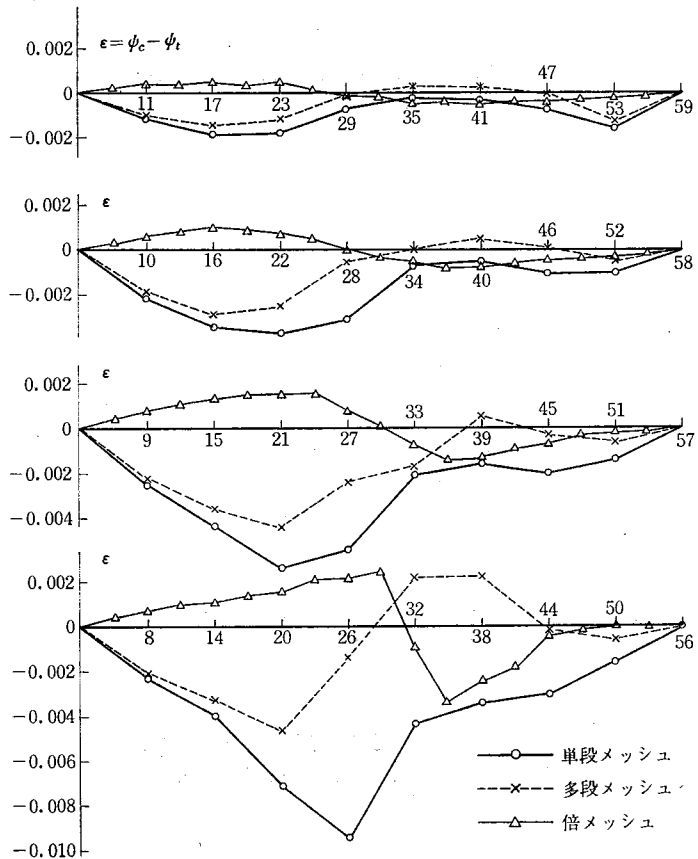


図 6 精度の比較 ( $\epsilon = \phi_c - \phi_t$ ,  $\phi_t = y - y/(x^2 + y^2)$  = 理論値,  $\phi_c$  = 計算値)

Fig. 6 Accuracy comparison

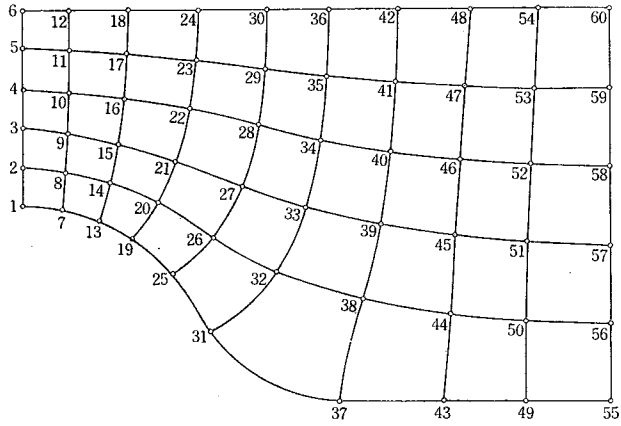


図 7 差分法の節点配置

Fig. 7 Node distribution of difference method

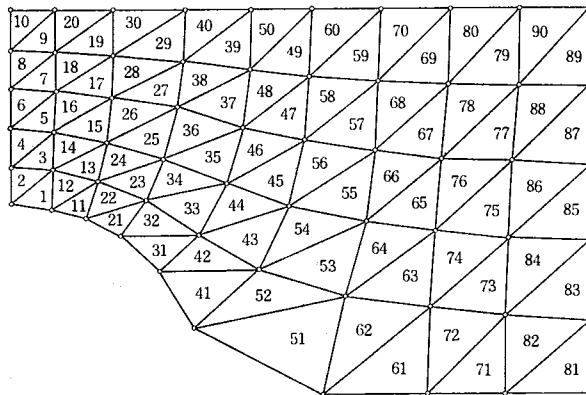


図 8 有限要素法の要素分割

Fig. 8 Element division of FEM

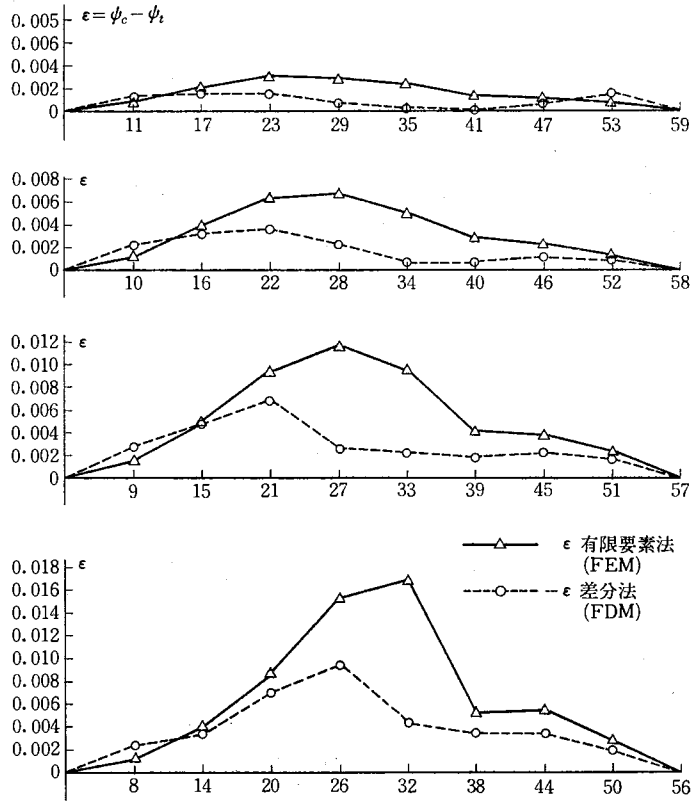


図 9 有限要素法と差分法の精度比較  
Fig. 9 Accuracy of FEM and FDM

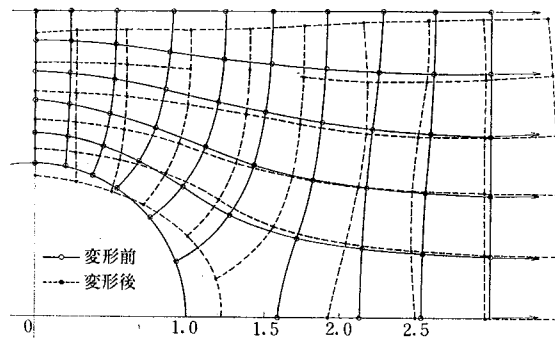


図 10 引張りによる有孔平板の変形 (節点 1 における形状係数は 3,768)  
Fig. 10 Deformation of plate with a circular hole by traction

2. 補間関数法

Taylor 展開法では逆行列を求めることが必要となり、多少計算量が多くなるおそれがあるので、ここにその必要がない計算法の提案を行う。

いま  $M$  個の節点により構成される差分家族について、節点  $m$  に関する補間関数  $p_m(x, y)$  が求められていれば、変数  $u(x, y)$  は差分家族の内部で

$$u(x, y) = p_m(x, y)u_m \tag{2-1}$$

によって近似展開される。補間関数は当然、

$$p_m(x_n, y_n) = \begin{cases} 1 & (n = m) \\ 0 & (n \neq m) \end{cases} \tag{2-2}$$

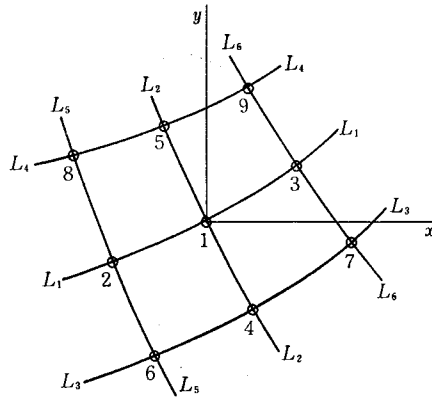


図 11 差分家族

Fig. 11 Difference family

の性質をもっている。したがって、微分方程式の差分化を行うとき補間関数を求めることが必要である。そのため、図 11 に示すように 3 点を通る曲線  $L_m$  をつくり、その節点  $n$  における値を

$$L_{mn} = L_m(x_n, y_n) \tag{2-3}$$

として、補間関数  $p_1(x, y) \sim p_9(x, y)$  を下記のように定義する。

$$\left. \begin{aligned} p_1 &= L_3 L_4 L_5 L_6 / L_{31} L_{41} L_{51} L_{61}, & p_2 &= L_2 L_3 L_4 L_6 / L_{22} L_{32} L_{42} L_{62} \\ p_3 &= L_2 L_3 L_4 L_5 / L_{23} L_{33} L_{43} L_{53}, & p_4 &= L_1 L_4 L_5 L_6 / L_{14} L_{44} L_{54} L_{64} \\ p_5 &= L_1 L_3 L_5 L_6 / L_{15} L_{35} L_{55} L_{65}, & p_6 &= L_1 L_2 L_4 L_6 / L_{16} L_{26} L_{46} L_{66} \\ p_7 &= L_1 L_2 L_4 L_5 / L_{17} L_{27} L_{47} L_{57}, & p_8 &= L_1 L_2 L_3 L_6 / L_{18} L_{28} L_{38} L_{68} \\ p_9 &= L_1 L_2 L_3 L_5 / L_{19} L_{29} L_{39} L_{59} \end{aligned} \right\} \tag{2-4}$$

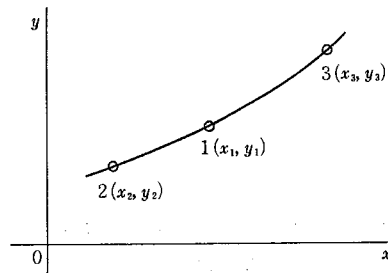


図 12 3 点を通る線分

Fig. 12 Line through three points



このように補間関数を定めるためには、任意の3点を通過する2次曲線を定めることが必要である。いま任意の3点1, 2, 3を通過する曲線は、次の2次式で表される(図12)。

$$L = a + bx + cy - \alpha(x^2 + y^2) = 0 \quad (2-5)$$

式(2-5)の諸係数は、次のように求める。

$$\left. \begin{aligned} \alpha_1 &= x_2y_3 - x_3y_2, \quad \beta_1 = y_2 - y_3, \quad \gamma_1 = x_3 - x_2, \quad f_1 = x_1^2 + y_1^2 \\ \alpha_2 &= x_3y_1 - x_1y_3, \quad \beta_2 = y_3 - y_1, \quad \gamma_2 = x_1 - x_3, \quad f_2 = x_2^2 + y_2^2 \\ \alpha_3 &= x_1y_2 - x_2y_1, \quad \beta_3 = y_1 - y_2, \quad \gamma_3 = x_2 - x_1, \quad f_3 = x_3^2 + y_3^2 \\ \alpha &= \alpha_1 + \alpha_2 + \alpha_3 \\ a &= \alpha_1 f_1 + \alpha_2 f_2 + \alpha_3 f_3, \quad b = \beta_1 f_1 + \beta_2 f_2 + \beta_3 f_3, \quad c = \gamma_1 f_1 + \gamma_2 f_2 + \gamma_3 f_3 \end{aligned} \right\} \quad (2-6)$$

このようにして、曲線  $L_m$  は、

$$L_m = a_m + b_m x + c_m y - \alpha_m(x^2 + y^2) \quad (2-7)$$

によって与えられ、補間関数  $p_1 \sim p_9$  も定められる。なお、3点を通る曲線でも表わすこともできるが、式の形は(2-5)と同様である。

$$\left. \begin{aligned} d_1 &= L_{31}L_{41}L_{51}L_{61}, \quad d_2 = L_{22}L_{32}L_{42}L_{62}, \quad d_3 = L_{23}L_{33}L_{43}L_{53} \\ d_4 &= L_{14}L_{44}L_{54}L_{64}, \quad d_5 = L_{15}L_{35}L_{55}L_{65}, \quad d_6 = L_{16}L_{26}L_{46}L_{66} \\ d_7 &= L_{17}L_{27}L_{47}L_{57}, \quad d_8 = L_{18}L_{28}L_{38}L_{68}, \quad d_9 = L_{19}L_{29}L_{39}L_{59} \end{aligned} \right\} \quad (2-8)$$

ただし、 $L_{mn} = L_m(x_n, y_n) = a_m + b_m x_n + c_m y_n - \alpha_m(x_n^2 + y_n^2)$  である。節点1における補間関数ならびに、その微係数値は以下のとおりである。

$$P_m = \begin{cases} 1 & (m=1) \\ 0 & (m=2 \sim 9) \end{cases} \quad (2-9)$$

$$\left. \begin{aligned} p_{1,x} &= (b_3 a_4 a_5 a_6 + a_3 b_4 a_5 a_6 + a_3 a_4 b_5 a_6 + a_3 a_4 a_5 b_6) / d_1 \\ p_{2,x} &= b_2 a_3 a_4 a_6 / d_2, \quad p_{3,x} = b_2 a_3 a_4 a_5 / d_3, \quad p_{4,x} = b_1 a_4 a_5 a_6 / d_4 \\ p_{5,x} &= b_1 a_3 a_5 a_6 / d_5, \quad p_{6,x} = p_{7,x} = p_{8,x} = p_{9,x} = 0 \end{aligned} \right\} \quad (2-10)$$

$$\left. \begin{aligned} p_{1,y} &= (c_3 a_4 a_5 a_6 + a_3 c_4 a_5 a_6 + a_3 a_4 c_5 a_6 + a_3 a_4 a_5 c_6) / d_1 \\ p_{2,y} &= c_2 a_3 a_4 a_6 / d_2, \quad p_{3,y} = c_2 a_3 a_4 a_5 / d_3, \quad p_{4,y} = c_1 a_4 a_5 a_6 / d_4 \\ p_{5,y} &= c_1 a_3 a_5 a_6 / d_5, \quad p_{6,y} = p_{7,y} = p_{8,y} = p_{9,y} = 0 \end{aligned} \right\} \quad (2-11)$$

$$\left. \begin{aligned} p_{1,xx} &= 2 \{ -(\alpha_3 a_4 a_5 a_6 + a_3 \alpha_4 a_5 a_6 + a_3 a_4 \alpha_5 a_6 + a_3 a_4 a_5 \alpha_6) \\ &\quad + b_3 b_4 a_5 a_6 + b_3 a_4 b_5 a_6 + b_3 a_4 a_5 b_6 + a_3 b_4 b_5 a_6 + a_3 b_4 a_5 b_6 \} / d_1 \\ p_{2,xx} &= 2 \{ -\alpha_2 a_3 a_4 a_6 + b_2 b_3 a_4 a_6 + b_2 a_3 b_4 a_6 + b_2 a_3 a_4 b_6 \} / d_2 \\ p_{3,xx} &= 2 \{ -\alpha_2 a_3 a_4 a_5 + b_2 b_3 a_4 a_5 + b_2 a_3 b_4 a_5 + b_2 a_3 a_4 b_5 \} / d_3 \\ p_{4,xx} &= 2 \{ -\alpha_1 a_4 a_5 a_6 + b_1 b_4 a_5 a_6 + b_1 a_4 b_5 a_6 + b_1 a_4 a_5 b_6 \} / d_4 \\ p_{5,xx} &= 2 \{ -\alpha_1 a_3 a_5 a_6 + b_1 b_3 a_5 a_6 + b_1 a_3 b_5 a_6 + b_1 a_3 a_5 b_6 \} / d_5 \\ p_{6,xx} &= 2 b_1 b_2 a_4 a_6 / d_6, \quad p_{7,xx} = 2 b_1 b_2 a_4 a_5 / d_7 \\ p_{8,xx} &= 2 b_1 b_2 a_3 a_6 / d_8, \quad p_{9,xx} = 2 b_1 b_2 a_3 a_5 / d_9 \end{aligned} \right\} \quad (2-12)$$

$$\left. \begin{aligned} p_{1,xy} &= \{ (b_3 c_4 + c_3 b_4) a_5 a_6 + (b_3 c_5 + c_3 b_5) a_4 a_6 + (b_3 c_6 + c_3 b_6) a_4 a_5 \\ &\quad + (b_4 c_5 + c_4 b_5) a_3 a_6 + (b_4 c_6 + c_4 b_6) a_3 a_5 + (b_5 c_6 + c_5 b_6) a_3 a_4 \} / d_1 \\ p_{2,xy} &= \{ (b_2 c_3 + c_2 b_3) a_4 a_6 + (b_2 c_4 + c_2 b_4) a_3 a_6 + (b_2 c_6 + c_2 b_6) a_3 a_4 \} / d_2 \\ p_{3,xy} &= \{ (b_2 c_3 + c_2 b_3) a_4 a_5 + (b_2 c_4 + c_2 b_4) a_3 a_5 + (b_2 c_5 + c_2 b_5) a_3 a_4 \} / d_3 \\ p_{4,xy} &= \{ (b_1 c_4 + c_1 b_4) a_5 a_6 + (b_1 c_5 + c_1 b_5) a_4 a_6 + (b_1 c_6 + c_1 b_6) a_4 a_5 \} / d_4 \\ p_{5,xy} &= \{ (b_1 c_3 + c_1 b_3) a_5 a_6 + (b_1 c_5 + c_1 b_5) a_3 a_6 + (b_1 c_6 + c_1 b_6) a_3 a_5 \} / d_5 \\ p_{6,xy} &= (b_1 c_2 + c_1 b_2) a_4 a_6 / d_6, \quad p_{7,xy} = (b_1 c_2 + c_1 b_2) a_4 a_5 / d_7 \\ p_{8,xy} &= (b_1 c_2 + c_1 b_2) a_3 a_6 / d_8, \quad p_{9,xy} = (b_1 c_2 + c_1 b_2) a_3 a_5 / d_9 \end{aligned} \right\} \quad (2-13)$$

$$\left. \begin{aligned}
 p_{1,yy} &= 2 \{ -(\alpha_3 a_4 a_5 a_6 + a_3 \alpha_4 a_5 a_6 + a_3 a_4 \alpha_5 a_6 + a_3 a_4 a_5 \alpha_6) \\
 &\quad + c_3 c_4 a_5 a_6 + c_3 a_4 c_5 a_6 + c_3 a_4 a_5 c_6 + a_3 c_4 c_5 a_6 + a_3 c_4 a_5 c_6 \} / d_1 \\
 p_{2,yy} &= 2(-\alpha_2 a_3 a_4 a_6 + c_2 c_3 a_4 a_6 + c_2 a_3 c_4 a_6 + c_2 a_3 a_4 c_6) / d_2 \\
 p_{3,yy} &= 2(-\alpha_2 a_3 a_4 a_5 + c_2 c_3 a_4 a_5 + c_2 a_3 c_4 a_5 + c_2 a_3 a_4 c_5) / d_3 \\
 p_{4,yy} &= 2(-\alpha_1 a_4 a_5 a_6 + c_1 c_4 a_5 a_6 + c_1 a_4 c_5 a_6 + c_1 a_4 a_5 c_6) / d_4 \\
 p_{5,yy} &= 2(-\alpha_1 a_3 a_4 a_5 + c_1 c_3 a_4 a_5 + c_1 a_3 c_5 a_6 + c_1 a_3 a_5 c_6) / d_5 \\
 p_{6,yy} &= 2c_1 c_2 a_4 a_6 / d_6, \quad p_{7,yy} = 2c_1 c_2 a_4 a_5 / d_7 \\
 p_{8,yy} &= 2c_1 c_2 a_3 a_6 / d_8, \quad p_{9,yy} = 2c_1 c_2 a_3 a_5 / d_9
 \end{aligned} \right\} \quad (2-14)$$

原点の近くで、変数  $u(x, y)$  ならびにその微係数は下記のように展開される。

$$\left. \begin{aligned}
 u &= p_m u_m = u_1, \quad u_x = p_m, x u_m, \quad u_y = p_m, y u_m \\
 u_{xx} &= p_m, xx u_m, \quad u_{xy} = p_m, xy u_m, \quad u_{yy} = p_m, yy u_m
 \end{aligned} \right\} \quad (2-15)$$

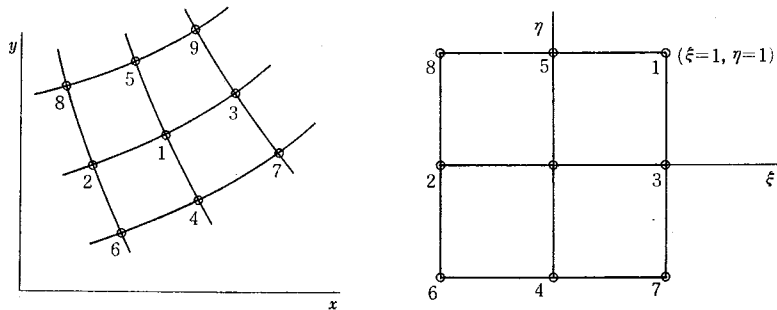
いま与えられた領域内で満たされる微分方程式ならびにその境界条件式を (1-8) と同様とし、これらの式は 1 節点 (原点) で (1-10) の形に差分表示されるものとする。ただし、係数行列は次のとおりである。

$$\left\{ \begin{matrix} k_1 \\ k_2 \\ k_3 \\ k_4 \\ k_5 \\ k_6 \\ k_7 \\ k_8 \\ k_9 \end{matrix} \right\} = \left[ \begin{matrix} 1 & p_{1,x} & p_{1,y} & p_{1,xx} & p_{1,xy} & p_{1,yy} \\ 0 & p_{2,x} & p_{2,y} & p_{2,xx} & p_{2,xy} & p_{2,yy} \\ 0 & p_{3,x} & p_{3,y} & p_{3,xx} & p_{3,xy} & p_{3,yy} \\ 0 & p_{4,x} & p_{4,y} & p_{4,xx} & p_{4,xy} & p_{4,yy} \\ 0 & p_{5,x} & p_{5,y} & p_{5,xx} & p_{5,xy} & p_{5,yy} \\ 0 & p_{6,x} & p_{6,y} & p_{6,xx} & p_{6,xy} & p_{6,yy} \\ 0 & p_{7,x} & p_{7,y} & p_{7,xx} & p_{7,xy} & p_{7,yy} \\ 0 & p_{8,x} & p_{8,y} & p_{8,xx} & p_{8,xy} & p_{8,yy} \\ 0 & p_{9,x} & p_{9,y} & p_{9,xx} & p_{9,xy} & p_{9,yy} \end{matrix} \right\} \left\{ \begin{matrix} A \\ B \\ C \\ D \\ E \\ F \end{matrix} \right\} \quad (2-16)$$

与えられた領域内で微分方程式を解く場合は、前節と同様に直交に近い曲線格子系をつくり、その格子点に節点をおき、差分化を行う節点  $i$  を中心とする局所座標、差分家族構成を定め、上の計算を行えばよい。

### 3. アイソパラメタ法

第 2 章において述べたように、曲線格子系の一つの差分家族について補間関数を求める計算手続きも、かなり複雑で大きな計算量を伴う。しかるに直交直線格子系では、この手続きは非常に簡単であるから、この系を媒介して補間関数を求めることが有効と考えられる。このために設けられた  $\xi$ - $\eta$  をパラメタ平面、 $x$ - $y$  を物理平面とよぶこととする。



(a) 物理平面 (b) パラメタ平面

図 13 物理平面とパラメタ平面

Fig. 13 Physical plane and parameter plane

いま図 13 に示される (a) 物理平面内の曲線格子は, (b) パラメタ平面内の定数 1 の直交直線格子に対応するものとして, その変換式を

$$x = p_m(\xi, \eta)x_m, \quad y = p_m(\xi, \eta)y_m \tag{3-1}$$

とする. したがって,  $p_m(\xi, \eta)$  は

$$p_m(\xi_n, \eta_n) = \begin{cases} 1 & (n=m) \\ 0 & (n \neq m) \end{cases} \tag{3-2}$$

を満たさねばならない. すなわち,  $p_m(\xi, \eta)$  は補間関数である.

### 3.1 補間関数ならびにその微係数

各節点に対し, (3-2) の補間関数とその微係数を求める.

$$\begin{array}{lll}
 p_1 = (\xi^2 - 1)(\eta^2 - 1) & p_{1,\xi} = 2\xi(\eta^2 - 1) & p_{1,\eta} = 2(\xi^2 - 1)\eta \\
 p_2 = (\xi - \xi^2)(\eta^2 - 1)/2 & p_{2,\xi} = (1 - 2\xi)(\eta^2 - 1)/2 & p_{2,\eta} = (\xi - \xi^2)\eta \\
 p_3 = (\xi + \xi^2)(1 - \eta^2)/2 & p_{3,\xi} = (1 + 2\xi)(1 - \eta^2)/2 & p_{3,\eta} = -(\xi + \xi^2)\eta \\
 p_4 = (\xi^2 - 1)(\eta - \eta^2)/2 & p_{4,\xi} = \xi(\eta - \eta^2) & p_{4,\eta} = (\xi^2 - 1)(1 - 2\eta)/2 \\
 p_5 = (1 - \xi^2)(\eta + \eta^2)/2 & p_{5,\xi} = -\xi(\eta + \eta^2) & p_{5,\eta} = (1 - \xi^2)(1 + 2\eta)/2 \\
 p_6 = (\xi - \xi^2)(\eta - \eta^2)/4 & p_{6,\xi} = (1 - 2\xi)(\eta - \eta^2)/4 & p_{6,\eta} = (\xi - \xi^2)(1 - 2\eta)/4 \\
 p_7 = (\xi + \xi^2)(\eta^2 - \eta)/4 & p_{7,\xi} = (1 + 2\xi)(\eta^2 - \eta)/4 & p_{7,\eta} = (\xi + \xi^2)(2\eta - 1)/4 \\
 p_8 = (\xi^2 - \xi)(\eta^2 + \eta)/4 & p_{8,\xi} = (2\xi - 1)(\eta^2 + \eta)/4 & p_{8,\eta} = (\xi^2 - \xi)(2\eta + 1)/4 \\
 p_9 = (\xi^2 + \xi)(\eta^2 + \eta)/4 & p_{9,\xi} = (2\xi + 1)(\eta^2 + \eta)/4 & p_{9,\eta} = (\xi^2 + \xi)(2\eta + 1)/4
 \end{array}$$

$$\begin{array}{lll}
 p_{1,\xi\xi} = 2(\eta^2 - 1) & p_{1,\xi\eta} = 4\xi\eta & p_{1,\eta\eta} = 2(\xi^2 - 1) \\
 p_{2,\xi\xi} = 1 - \eta^2 & p_{2,\xi\eta} = (1 - 2\xi)\eta & p_{2,\eta\eta} = \xi - \xi^2 \\
 p_{3,\xi\xi} = 1 - \eta^2 & p_{3,\xi\eta} = -(1 + 2\xi)\eta & p_{3,\eta\eta} = -(\xi + \xi^2) \\
 p_{4,\xi\xi} = \eta - \eta^2 & p_{4,\xi\eta} = \xi(1 - 2\eta) & p_{4,\eta\eta} = 1 - \xi^2 \\
 p_{5,\xi\xi} = -(\eta + \eta^2) & p_{5,\xi\eta} = -\xi(1 + 2\eta) & p_{5,\eta\eta} = 1 - \xi^2 \\
 p_{6,\xi\xi} = (\eta^2 - \eta)/2 & p_{6,\xi\eta} = (1 - 2\xi)(1 - 2\eta)/4 & p_{6,\eta\eta} = (\xi^2 - \xi)/2 \\
 p_{7,\xi\xi} = (\eta^2 - \eta)/2 & p_{7,\xi\eta} = (1 + 2\xi)(2\eta - 1)/4 & p_{7,\eta\eta} = (\xi^2 + \xi)/2 \\
 p_{8,\xi\xi} = (\eta^2 + \eta)/2 & p_{8,\xi\eta} = (2\xi - 1)(2\eta + 1)/4 & p_{8,\eta\eta} = (\xi^2 - \xi)/2 \\
 p_{9,\xi\xi} = (\eta^2 + \eta)/2 & p_{9,\xi\eta} = (2\xi + 1)(2\eta + 1)/4 & p_{9,\eta\eta} = (\xi^2 + \xi)/2
 \end{array}$$

節点 1 ( $\xi=0, \eta=0$ ) における補間関数およびその微係数値は表 1 のようになる.

表 1  $\xi=0, \eta=0$  での補間関数とその微係数値  
Table 1 Interpolation function and differential coefficient value on  $\xi=0, \eta=0$

$m$	$p_m$	$p_{m,\xi}$	$p_{m,\eta}$	$p_{m,\xi\xi}$	$p_{m,\xi\eta}$	$p_{m,\eta\eta}$
1	1			-2		-2
2		-1/2		1		
3		1/2		1		
4			-1/2			1
5			1/2			1
6					1/4	
7					-1/4	
8					-1/4	
9					1/4	

### 3.2 座標変換

物理  $x$ - $y$  平面とパラメタ  $\xi$ - $\eta$  平面間の変換式  $x=x(\xi, \eta)$ ,  $y=y(\xi, \eta)$  は陽に与えられているが, その反対に  $\xi=\xi(x, y)$ ,  $\eta=\eta(x, y)$  は陽に与えられていない。したがって,  $x, y$  に関する微分は下記の変換式による必要がある。

$$J = x_{\xi}y_{\eta} - x_{\eta}y_{\xi} \quad (3-9)$$

$$J_{\xi} = x_{\xi\xi}y_{\eta} + x_{\xi}y_{\xi\eta} - x_{\xi\eta}y_{\xi} - x_{\eta}y_{\xi\xi} \quad (3-10)$$

$$J_{\eta} = x_{\xi\eta}y_{\eta} + x_{\xi}y_{\eta\eta} - x_{\eta\eta}y_{\xi} - x_{\eta}y_{\xi\eta} \quad (3-11)$$

これら (3-9), (3-10), (3-11) から,

$$\xi_x = y_{\eta}/J \quad (3-12)$$

$$\xi_y = -x_{\eta}/J \quad (3-13)$$

$$\eta_x = -y_{\xi}/J \quad (3-14)$$

$$\eta_y = x_{\xi}/J \quad (3-15)$$

$$\xi_{xx} = (y_{\eta}y_{\xi\eta} - y_{\xi}y_{\eta\eta})/J^2 + y_{\eta}(y_{\xi}J_{\eta} - y_{\eta}J_{\xi})/J^3 \quad (3-16)$$

$$\xi_{xy} = (y_{\xi}x_{\eta\eta} - y_{\eta}x_{\xi\eta})/J^2 + x_{\eta}(y_{\eta}J_{\xi} - y_{\xi}J_{\eta})/J^3 \quad (3-17)$$

$$\xi_{yy} = (x_{\eta}x_{\xi\eta} - x_{\xi}x_{\eta\eta})/J^2 + x_{\eta}(x_{\xi}J_{\eta} - x_{\eta}J_{\xi})/J^3 \quad (3-18)$$

$$\eta_{xx} = (y_{\xi}y_{\xi\eta} - y_{\eta}y_{\xi\xi})/J^2 + y_{\xi}(y_{\eta}J_{\xi} - y_{\xi}J_{\eta})/J^3 \quad (3-19)$$

$$\eta_{xy} = (y_{\eta}x_{\xi\xi} - y_{\xi}x_{\xi\eta})/J^2 + x_{\xi}(y_{\xi}J_{\eta} - y_{\eta}J_{\xi})/J^3 \quad (3-20)$$

$$\eta_{yy} = (x_{\xi}x_{\xi\eta} - x_{\eta}x_{\xi\xi})/J^2 + x_{\xi}(x_{\eta}J_{\xi} - x_{\xi}J_{\eta})/J^3 \quad (3-21)$$

を求める。したがって, 変数  $u(x, y)$  ならびにその微係数は

$$u(x, y) = p_m(\xi, \eta)u_m \quad (3-22)$$

$$u_x = \xi_x u_{\xi} + \eta_x u_{\eta} \quad (3-23)$$

$$u_y = \xi_y u_{\xi} + \eta_y u_{\eta} \quad (3-24)$$

$$u_{xx} = (\xi_x)^2 u_{\xi\xi} + 2\xi_x \eta_x u_{\xi\eta} + (\eta_x)^2 u_{\eta\eta} + \xi_{xx} u_{\xi} + \eta_{xx} u_{\eta} \quad (3-25)$$

$$u_{xy} = \xi_x \xi_y u_{\xi\xi} + (\xi_x \eta_y + \xi_y \eta_x) u_{\xi\eta} + \eta_x \eta_y u_{\eta\eta} + \xi_{xy} u_{\xi} + \eta_{xy} u_{\eta} \quad (3-26)$$

$$u_{yy} = (\xi_y)^2 u_{\xi\xi} + 2\xi_y \eta_y u_{\xi\eta} + (\eta_y)^2 u_{\eta\eta} + \xi_{yy} u_{\xi} + \eta_{yy} u_{\eta} \quad (3-27)$$

となる。節点 1 ( $\xi=0, \eta=0$ ) における微係数値は次のとおりである。

$$\left. \begin{aligned} x_{\xi} &= (-x_2 + x_3)/2 \\ x_{\eta} &= (-x_4 + x_5)/2 \\ x_{\xi\xi} &= -2x_1 + x_2 + x_3 \\ x_{\xi\eta} &= (x_6 - x_7 - x_8 + x_9)/4 \\ x_{\eta\eta} &= -2x_1 + x_4 + x_5 \end{aligned} \right\} (3-28) \quad \left. \begin{aligned} y_{\xi} &= (-y_2 + y_3)/2 \\ y_{\eta} &= (-y_4 + y_5)/2 \\ y_{\xi\xi} &= -2y_1 + y_2 + y_3 \\ y_{\eta\xi} &= (y_6 - y_7 - y_8 + y_9)/4 \\ y_{\eta\eta} &= -2y_1 + y_4 + y_5 \end{aligned} \right\} (3-29)$$

$$\left. \begin{aligned} u_{\xi} &= (-u_2 + u_3)/2 \\ u_{\eta} &= (-u_4 + u_5)/2 \\ u_{\xi\xi} &= -2u_1 + u_2 + u_3 \\ u_{\xi\eta} &= (u_6 - u_7 - u_8 + u_9)/4 \\ u_{\eta\eta} &= -2u_1 + u_4 + u_5 \end{aligned} \right\} (3-30)$$

以上により  $\xi_x, \dots, \eta_{\eta\eta}$  の節点 1 における値が求められる。

つぎに, 変数ならびにその微係数の差分表示を求める。まず, (3-12)~(3-21) を (3-22) ~ (3-27) に代入して, 次式をつくる。

$$u = u_1 \quad (3-31)$$

$$u_x = (-\xi_x u_2 + \xi_x u_3 - \eta_x u_4 + \eta_x u_5)/2 \quad (3-32)$$

$$u_y = (-\xi_y u_2 + \xi_y u_3 - \eta_y u_4 + \eta_y u_5)/2 \quad (3-33)$$

$$u_{xx} = -2(\xi_x^2 + \eta_x^2)u_1 + (\xi_x^2 - 0.5\xi_{xx})u_2 + (\xi_x^2 + 0.5\xi_{xx})u_3 + (\eta_x^2 - 0.5\eta_{xx})u_4 + (\eta_x^2 + 0.5\eta_{xx})u_5 + 0.5\xi_x\eta_x(u_6 - u_7 - u_8 + u_9) \quad (3-34)$$

$$u_{xy} = -2(\xi_x\xi_y + \eta_x\eta_y)u_1 + (\xi_x\xi_y - 0.5\xi_{xy})u_2 + (\xi_x\xi_y + 0.5\xi_{xy})u_3 + (\eta_x\eta_y - 0.5\eta_{xy})u_4 + (\eta_x\eta_y + 0.5\eta_{xy})u_5 + 0.25(\xi_x\eta_y + \xi_y\eta_x)(u_6 - u_7 - u_8 - u_9) \quad (3-35)$$

$$u_{yy} = -2(\xi_y^2 + \eta_y^2)u_1 + (\xi_y^2 - 0.5\xi_{yy})u_2 + (\xi_y^2 + 0.5\xi_{yy})u_3 + (\eta_y^2 - 0.5\eta_{yy})u_4 + (\eta_y^2 + 0.5\eta_{yy})u_5 + 0.5\xi_y\eta_y(u_6 - u_7 - u_8 + u_9) \quad (3-36)$$

微分方程式 (1-11) は、

$$k_m u_m = f \quad (3-37)$$

の形に差分化される。ただし、係数は次のとおりである。

$$\left. \begin{aligned} k_1 &= A - 2(\xi_x^2 + \eta_x^2)D - 2(\xi_x\xi_y + \eta_x\eta_y)E - 2(\xi_y^2 + \eta_y^2)F \\ k_2 &= -0.5\xi_x B - 0.5\xi_y C + (\xi_x^2 - 0.5\xi_{xx})D + (\xi_x\xi_y - 0.5\xi_{xy})E + (\xi_y^2 - 0.5\xi_{yy})F \\ k_3 &= 0.5\xi_x B + 0.5\xi_y C + (\xi_x^2 + 0.5\xi_{xx})D + (\xi_x\xi_y + 0.5\xi_{xy})E + (\xi_y^2 + 0.5\xi_{yy})F \\ k_4 &= -0.5\eta_x B - 0.5\eta_y C + (\eta_x^2 - 0.5\eta_{xx})D + (\eta_x\eta_y - 0.5\eta_{xy})E + (\eta_y^2 - 0.5\eta_{yy})F \\ k_5 &= 0.5\eta_x B + 0.5\eta_y C + (\eta_x^2 + 0.5\eta_{xx})D + (\eta_x\eta_y + 0.5\eta_{xy})E + (\eta_y^2 + 0.5\eta_{yy})F \\ k_6 &= 0.5\xi_x\eta_x D + 0.25(\xi_x\eta_y + \xi_y\eta_x)E + 0.5\xi_y\eta_y F \\ k_7 &= -k_6, \quad k_8 = -k_6, \quad k_9 = k_6 \end{aligned} \right\} \quad (3-38)$$

#### 4. おわりに

以上、任意節点配置差分法について三つの方法論について述べた。Taylor 展開法は逆行列を求める計算が必要であるが、計算手順やプログラム化が容易で精度も良好と思れる。補間関数法は3点を通る補間曲線の積により補間関数を求めるもので、補間関数は不完全多項式表示となり、計算手順やプログラム化もかなり複雑となる。アイソパラメタ法はパラメタ平面の補間関数を物理平面に再変換するもので平面の歪を伴うおそれがある。パラメタ平面では、有限要素法におけるアイソ・パラメトリック要素と同様に、変数の変換は座標の変換と同様とする。この方法も計算手順やプログラム化は、Taylor 展開法に比べ複雑である。いずれの方法が計算精度、時間に関して最も優れているかは今後の研究に待たねばならない。さらに有限要素法とも同様の比較検討が必要である。

参考文献 [1] 藤野勉, 渡部義雄, “新しい差分法の提案と数値実験”, 技報, 日本ユニパック(株), No. 0, 1981.  
 [2] W. H. Frey, “Flexible finite-difference stencils from isoparametric finite elements,” International Journal for Numerical Methods in Engineering, Vol. 11, No. 11, 1977.

執筆者紹介 藤野 勉 (Tsutomu Fujino)

明治45年生, 昭和11年東京帝国大学理学部物理科卒業, 同年三菱重工(株)入社, 主に応力・振動・流体力学などの解析法(有限要素法を含む)の研究に従事, 32年, 工学博士号を取得, 47年, 同社技術本部顧問となる。また, 48年より東海大学工学部教授, 52年依嘱教授を歴任, 51年より日本ユニパック(株)の技術顧問となり, 現在に至る。「コンピュータによる構造工学講座Ⅱ-4-B-熱伝導と熱応力」(培風館, 1972)などの著書がある。



## 論文 精密ロック

## Precision Locks

J. R. Jordan, J. Banerjee, R. B. Batman

**要約** 本稿は、データベースの並行処理を制御する新しい方法、精密ロック (Precision Locks, PL) を記述したものである。精密ロックという方法名は、データベースの整合性を保証するのに必要なレコードの集合のみを精密にロックする、という事実由来している。このとき、ファントム (存在しないレコードのこと) をもロックする。この方法によると、レコード・レベルのすべてのロック方法 (述語ロックも含めて) の中で並行処理の度合いが最大となる。また、ロック・システム相互の比較をするために、ロック・システムが犯す誤りを表示する枠組を導入する。さらに既存のシステムはこれらの誤りを犯すが、PL では誤りが発生しないことを示す。つづいて、他の論理的なロック・システムでは許される述語の複雑さに制限があるが、PL では任意であることを述べる。また、PL の下のトランザクションは良型かつ2段階であり、そのことが整合性を保つのに十分であることを示す。PL の実行コストを伝統的なレコード・レベルのロック方法と比較し、PL の方が通常安上がりだという結論を得る。

**Abstract** This paper describes a new method for controlling data base concurrency, called Precision Locks (PL). The name is derived from the fact that they lock precisely the set of tuples required to guarantee data base consistency. phantoms included. This results in maximum concurrency among all tuple-level locking methods, including predicate locks. A framework for comparing locking systems which indicates the errors committed by them is introduced. It is shown that existing systems commit these errors while PL does not. While other logical locking systems are limited in the complexity of permissible predicates, PL allows them to be arbitrary. It is shown that transactions under PL are well formed and two phased which is sufficient to maintain consistency. The execution cost of PL is compared to a conventional tuple-level locking method. It is found that PL usually is cheaper to execute.

## 1. はじめに

本稿はデータベースの並行処理を制御する新しい方法、精密ロック (PL) を記述したものである。精密ロックという名前は、この方法がデータベースの整合性を保証するのに必要なレコードの集合のみを精密にロックするという事実由来している。このとき、ファントム (存在しないレコードのこと) をもロックする。

これによると、述語ロックも含めた、レコード・レベルのすべてのロック方法の中で、並行処理の度合いが最大となる。ロック・システム相互の比較をするために、ロック・システムが犯す誤りを表示する枠組を2章で導入する。既存の (物理的および論理的な) ロック方法の批判は3章で行う。精密ロックは4章で記述する。ここではPLが良型かつ2段階の任意の述語を受け入れることを述べる。さらに、物理的なレコード・レベルのロック・システムとPLとの間の単純な実行コストを比較する。5章では要約とわれわれの結論を述べる。

## 2. 比較のための枠組

二つの命題, すなわち「データベースの整合性を保つために, 対象のしかるべき集合をロックしなければならない」と「どのシステムも対象となる, ある集合をロックする」を考える. これら二つの命題が, 同時に真となるか, ならないかによって四つのケースが考えられる (図1).

		ロックしなければならない	
		真	偽
ロックされている	真	—	$\beta$
	偽	$\alpha$	—

図1 ロック・システムが犯す誤り

Fig. 1 Errors committed by locking systems

図1での横棒は正しい動きを表わしているが,  $\alpha$  と  $\beta$  は誤りを表わしている.

誤り  $\alpha$  は, その内容によって次の4種類に分けられる.

- 1) 失なわれた更新: あるトランザクションが更新したものの上に, 他のトランザクションが重ねて更新の結果を書いてしまう.
- 2) 不整合な結果: あるトランザクションは, 二つの関係あるデータ  $x$  と  $y$  を  $x'$  と  $y'$  に変えた. しかし, 他のトランザクションはこれを参照せず,  $x$  と  $y'$ , または  $x'$  と  $y$  というようにみてしまう.
- 3) 不安定なレコード: あるトランザクションは  $x$  を一旦  $x'$  に変えて, 最後に  $x''$  に変えた. しかし他のトランザクションは  $x'$  をみてしまう.
- 4) ファントム: あるトランザクションが存在していないレコードを要求する. その後, 他のトランザクションの動きの結果そのレコードが生成される.

誤り  $\alpha$  には三つの原因が考えられる. すなわち,

- 1) あるレコードをロックしなければならないことを知っているプログラマが, ロックに失敗した.
- 2) プログラム上のバグにより, 場合によってはあるレコードをロックしない.
- 3) 目的のレコードをわたすときに, プログラムに代わってシステムがレコードを触れるが, このときシステムの欠陥によって, そのすべてをプログラムにみせてしまう. この場合, プログラムはロックする機会をもっていない.

ユーザは 1, 2 番の誤りについては正せるが, 3 番目の誤りは正すことができない. しかし, システムは三つのすべてを正すことができる. すなわち, システムがユーザに代わって, 触れるすべてのレコードを単純にロックしてしまえばよい. こうすれば, ユーザにはロックするという負荷がまったくなくなる.

誤り  $\beta$  はそれによって不整合になることはないが, 並行処理を禁止してしまうので, システムのパフォーマンスを悪化させる. また, 不必要なロックもしてしまうので, オーバーヘッドを増やすことにもなる. この誤り  $\beta$  はロックの方法に依存する.

物理的ロック・システムでは, ユーザがデータに対して何をしたいのかを知る術がないので, 必要以上にロックしがちである. 論理的ロック・システムではこれまでのところ<sup>[1][3]</sup>, 起こるかもしれない衝突と実際に起こる衝突とを区別する効果的な術がないので, やはり, 過剰にロックしがちである.

第4章でみるように, PL ではこれらの誤り ( $\alpha$  と  $\beta$ ) は起こらない.

### 3. 既存の方法に対する批判

#### 3.1 物理的ロック・システム

われわれの知る限り、物理的ロック・システムに関してまとまった有用な文献はない。しかし、このシステムの本質は次のようにまとめることができる。すなわち、

ある物理的な集合体を  $A$  とする。

ただし、 $A \in \{\text{レコード, ページ, エリア, ファイル, など}\}$ 。

とすると、 $A$  のロックは直接  $A$  に印を付けるか、またはリスト上に  $A$  の名前を記録することによってなされる。もし、 $A$  に印を付けた後に主記憶装置が足りなくなったときは、 $A$  をデータベースに書き込んでその印がなくならないようにしなければならない。もし、その印の中に時刻が記入されているならば、トランザクションの終了時に印を消してロックを解くために  $A$  を読み込んでくる必要はない。一般的に言えば、印を付ける方法は余計な I/O が必要となるので好ましくない。

$A$  の大きさ  $|A|$  は粒 (granularity) と呼ばれ、上記のリスト中では左から右にいくにつれて大きくなる。Potier と Leblanc<sup>[7]</sup> は、システムのスループット  $T$  を  $|A|$  の関数として考察した。ロックのオーバーヘッドと並行処理の可能性は、 $|A|$  が大きくなるにつれて単調に減少する。 $|A|$  が大きくなるにつれて、衝突の発生頻度が高くなるので、並行処理されるトランザクションの数は減少する。しかし、 $|A|$  が大きくなるとロックを調べる回数が少なくなるので、オーバーヘッドは小さくなる。彼らは次のことを示した。すなわち、 $|A|$  が小さくなるにつれてスループット  $T$  は最少点に達し、その後着実に増加するという。そして結論は、細かい粒にすることがシステムのスループットを上げる最良の方針であるとされる。

Ries と Stonebraker は文献<sup>[5]</sup> の中で次のことを結論として述べている。すなわち、資源の要求がトランザクションの始まる時になされるならば、大きな粒にする方が好ましい。この方法ではデッドロックが起こらないし、オーバーヘッドが小さい。しかし、彼らは別の文献<sup>[6]</sup> の中で、もし資源が必要となる都度要求されるのであれば“…最も細かい粒にすることが必須である”と述べている。

#### 3.2 論理的ロック・システム

整合性と述語ロックの考えは、Eswaran, Gray 他による優れた文献<sup>[1]</sup> に述べられている。その中でフェントムの問題(註)は“述語ロック”によって解決することが示されている。彼らのやり方は各々のトランザクションと述語ロックを結びつけるものである。すなわち、述語  $L$  は  $L=(R, P, a)$  の形をしていて、 $R$  は関係名、 $P$  は求めるレコードを記述した述語、 $a$  はアクセス方法で読み出しまは書き込みである。もし二つのロックを同時に満たし、その中の一つが  $a$ =書き込み であるようなレコード  $t$  が存在しうらば、トランザクションは衝突を起こすことがあり、そのとき、トランザクションは一つ一つ順番にスケジュールされる、と論じている。

彼らはまた、二つの述語ロック  $P$  と  $P'$  が衝突する可能性を調べるために、両者を比較するという方法を示唆している。その手続きは、 $P \wedge P'$  を論理和標準形に直して、いずれの素命題も満たされるか、否かを調べるというものである。

この方法は大変興味深いものである。なぜなら、この方法は、ロックを調べる回数を減らすことができる。すなわち、トランザクションによってアクセスされる一対のレコードごとにテストを1回する代わりに、一対のトランザクションごとに1回のテストとすることができるからである。



一方、この方法は、衝突のテストにかなりの時間がかかるという欠点がある。彼らの中の何人かによるその後の文献<sup>[2]</sup>では、時間がかかりすぎるという理由でこの方法を捨てている。

Lee と Yeh は文献<sup>[3]</sup>の中で、述語ロックがあまりに大きな論理空間をロックし、不必要に並行処理を制限する、という理由でそれを非難している。彼らの主張は2次元（行と列）のロックの“構造的ロック”である。この方法によると、ある場合には二つのトランザクションが同じレコードを操作することができる。ただし、これの前提条件に、二つのトランザクションが衝突の可能性のあるやり方で同じ属性を使うことがないとしている。彼らの主張によると、この方法が並行処理の度合いを最大にするという。その理由は、単に起こるかもしれない衝突ではなく、実際のレコードについて考えているのだから、としている。しかし、並行処理の度合いが最大になるという彼らの主張は疑わしい。なぜならば、二つのトランザクションが互いに妨害し合わないのに、(PL は二つが並行して走ることを許しているが) 彼らの手続きでは二つが順番に一つ一つ走るようになってしまうような例を容易にみつけ出すことができるからである。問題は、選抜のアルゴリズムの中に“衝突の可能性”のテストが入っていることである。すなわち、彼らのアルゴリズムでは、あるトランザクションが選抜の目的で使う属性を他のトランザクションが変更しようとする、いつもそれを妨げるようになっている。

述語ロックに関する Lee と Yeh の批判には根拠がない。なぜなら、述語ロックは任意に細かい述語の仕様を認めているのであるから。すなわち、述語ロックの方法は書かれるレコードによって表現される  $n$ -空間 ( $n$ は属性の数) の中の点を正確に指示することを認めているのである。

述語ロックに関するわれわれの批判点は次の三つである。

- 1) 読み出しに関して述語を指示することはまったく自然（おそらく必要）だが、書き込みに関して述語を指示することは不自然である。すなわち、レコードを要求した人に、データベース・システムがその人の必要とするレコードの（任意の）部分集合を効率的にわたすことができるように、読み出しに関して述語を指示する。しかし、データベースの任意の部分集合を書き込むなどということほしない。その代わりに、ある決められたレコードの集合をデータベースへ書き込むようにする。この決められたレコードの集合を記述するために、述語を組み立てるというのは不自然である。
- 2) Eswaran らの文献<sup>[1]</sup>で触れられ、Kleene の文献<sup>[4]</sup>に記述されているように、演算子が制限されない限り、二つの述語が衝突するかどうかを決定する問題は計算不可能である。
- 3) Eswaran らが自ら結論づけているように、実行コストが高い。われわれがこれから述べるロックの方法には、これらの欠陥はない。

#### 4. 精密ロック

精密ロックは述語ロックの一つの実現方法であると考えられる。すなわち、実際に衝突が起こるときにのみ、それを防ぐようにして、ファントム問題を効果的に解決している。この方法の並行処理の度合いが、すべてのレコード・レベル・ロックの方法の中で最大であると考えている。この方法は述語の中の演算子のある部分集合の範囲に限定することほしない。

#### 4.1 方 法

データベースの整合性に関して、次の定義を仮定する。データベースは次の場合に限り整合性がある。すなわち、トランザクションの並行処理の度合いに関係なく、(すべてのトランザクションが終了したことによって) システムがある静止した状態に達したとする。このとき、同じデータベースの静止状態を生じるようなトランザクションを、一つずつ取り上げては処理する順序関係が存在しなければならない。

すなわち、トランザクションは一時に一つが走るように現われなければならない。このことを保証する必要十分条件は、次のようなものであることは明らかである。

あるトランザクション  $T_2$  が走っている間、別のトランザクション  $T_1$  が  $T_2$  のリード空間を変更してはならない。いいかえれば、 $T_1$  によるデータベースの変更は、 $T_2$  が始まる前か、終わった後になされたかのように見えなければならない。

次の議論の中で、われわれはデータベースは互いに素な関係に分割され、かつ二つのトランザクションの行為は同じ関係に向けられているか、または、二つはまったく何かかわりもないかのどちらかであることを仮定している。

レコードの変更は、古いのを削除して新しいものを挿入することによってなされると仮定する。さらに、レコードを削除するためにデータベース・システムがそのレコードの存在を検証する必要があること、それゆえ、データベース・システムは削除したことを知らせる前に、そのレコードを見つけるためにデータベースを読みに行くことと仮定する。このようにしてデータベース・システムは、そのレコードが価値のあるものであることを証明するのに、そのイメージを手に入れる。

各々のトランザクション  $T_i$  ごとに、二つの順序集合  $P_i = \{p_{ij}\}$  と  $U_i = \{u_{ik}\}$  が関連している。ここで、 $p_{ij}$  は  $T_i$  による  $j$  番目の“述語 (predicate)”， $u_{ik}$  は  $T_i$  による  $k$  番目の“更新 (update)”を表わす。“述語”とは、データベースから読み出してくるべきレコードに関する論理的記述をさす。すなわち、“職種名=プログラマ、かつ、所在地=ロースビル”というように。また“更新”によって、データベースへ挿入すべき、またはデータベースから削除すべき実在のレコードが表わされる。もし実在するレコードが変更される場合、更新の操作は削除とそれに続く挿入として取り扱われる。

$T_i$  が動いていくとき、述語リスト  $L_p$  の中へその(読み出し)述語を投げ入れ、また更新リスト  $L_u$  の中へその更新を投げ入れる。 $p_{ij}$  が  $L_p$  に投げ入れられると、更新リスト  $L_u$  の中に  $p_{ij}(u_{kl}) = \text{真} (k \neq i)$  となる  $u_{kl}$  があるかどうか調べられる。もし真となるものがあれば、 $T_k$  は  $T_i$  のリード空間を修正しているところであるから、 $T_i$  は  $T_k$  が終了するまで抑えられる。(実際には  $T_i$  が  $T_k$  より先行している可能性があるが、われわれはそうならないと仮定している。)  $T_k$  が終了すると  $u_{kl}$  は  $L_u$  から取り除かれ、 $T_i$  が動き出す。その結果、あたかも  $T_k$  (変更トランザクション) が  $T_i$  より先に走ったかのようになる。

$T_i$  が更新  $u_{ij}$  を  $L_u$  に投げ入れた場合は、 $p_{kl}(u_{ij}) = \text{真} (k \neq i)$  となる  $p_{kl}$  があるかどうか調べられる。もし真となるものがあれば、 $T_i$  は  $T_k$  のリード空間を修正しようとしているところで、 $T_i$  は  $T_k$  が終了するまで抑えられる。 $T_k$  が終了すると、 $p_{kl}$  は取り除かれ  $T_i$  は動き出す。その結果、 $T_k$  (検索トランザクション) が  $T_i$  の前に走ったかのようになる。

検索トランザクション相互、変更トランザクション相互の間では何も調べないことに注目すべきであるが、両方とも調べる必要はない。検索トランザクション相互が同じ対象に

ついて干渉し合うことがないことは明らかだが、変更トランザクション相互が干渉しないということは自明ではない。変更は実在のレコードの修正（削除を含む）か、または新しいレコードの挿入である。もし、修正ならばトランザクションは古いコピーを読み出しておかなければならない。そして、他のトランザクションによる更新を防ぐために（読み出し）述語を投げ入れておくことになる。もし、挿入で重複キーが許されているならば、二つのトランザクションが同じレコードを挿入しても一向にかまわない。その結果は、二つのトランザクションが一つ一つ順番に走ったのと同じになる。もし、挿入でかつ重複キーが許されていない場合は、そのレコードが、すでに存在していないかどうかを検証するために読み出しが行わなければならない。この場合には、 $L_p$  に投げ入れられたままになっている読み出し述語が、ふたたび他の変更トランザクションを妨げる。

PL は実際の読み出し述語だけを調べるので、レコードは絶対必要になるまでロックされない。物理的ロックと対称的に、パス集合（“要求集合”を引きわたす過程で読み出さなければならないもの）はロックする必要がない。

述語ロックとの対比では、

- 1) ファントム問題を解決するために範囲の広い述語を指示することによって、不必要なレコードをロックしてしまうことがない。
  - 2) 述語相互を時間をかけて調べる必要がない。
- などの特徴がある。

#### 4.2 任意の述語

許される述語に関して唯一の制約があることは明らかである。あるレコード  $t$  と述語  $P$  が与えられ、他には何の情報もないとき、 $P(t)$  が真か偽かを決定することが可能でなければならない。  $P$  がどんな演算子を含むかについては何の制限もない。たとえば  $P$  は次のようなものでもよい。

$$b^2 - 4ac \geq 0$$

すなわち、「 $a, b, c$  によって表現される 2 次方程式の根は実数か」でもよい。

#### 4.3 良型かつ 2 段階

良型かつ 2 段階の考え方は Eswaran らの文献<sup>[1]</sup>に定義されている。“良型 (well formed)”とは、対象が使われる前にロックされることを意味し、“2 段階 (two phased)”とは、一旦どれかのロックを解放したトランザクションが二度と新しいロックをかけてはならないことを意味している。すべてのトランザクションが上の二つの条件を満たせば、データベースの整合性は保たれることを、彼らは示した。

2 段階の考え方は実際には“ロックの方法”の現象ではなくて、ロックの方法いかにかわらず、システムで守られるべき規律である。システムは、コミット点に達するまでトランザクションから要求されたすべてのロックを単に保持する。

精密ロックの方法では、良型という性質を似たような状況にして実現する。すなわち、トランザクションは実際に何もロックしないという状況である。トランザクションは単に検索または変更の要求を出す。システムはそれを認める前に、その要求をまずロックに転換する。たとえば検索の場合、述語はその述語を満たしているレコードを他のトランザクションが変更することのないように用いられる。また変更の場合、更新は、他のトランザクションがその更新を満たす述語を出すのを妨ぐのに使われる。

変更レコードは書き込まれる瞬間までロックされないというのは、「ロックしてから実行する」という規則に違反しているので、奇妙に思えるかもしれない。しかし、この規則

は読み出されたデータに対してのみ意味をもつ。トランザクション  $T_i$  は、それを書き込む前に  $u_{ik}$  をロックし損ねても不整合の結果にはならない。なぜなら、他のトランザクションを傷つけないかどうかを  $u_{ik}$  を投げ入れることによって、システムが監視できるようにしている。そのため、更新は何か他のトランザクションの（述語）ロックを満たすと、止まってしまう。

もちろん、すべての資源を最初に要求することはないので、デッドロックになったらトランザクションをロールバックすることが必要になる。

#### 4.4 実行コスト

ロック・システムの実行コストを見積もることは、複雑な問題である。手続きそれ自体によって費やされる資源そのものは比較的単純である。より微妙な問題はこの方法によって許される並行処理の度合いである。もし、ボトル・ネックの避けられない個所が物理的システムのどこかにあるのであれば、並行処理の度合いを最大にする手続きをもつことには実際上何の価値もない。しかし、豊富な資金があるとすれば、このボトル・ネックは理論上除くことができる。したがって論理的にシステムの能力を決定するのは、論理的な並行処理の度合いのみである。

もしロックの手続きが長時間を費やすものであれば、ロックの手続き自体が CPU ボトル・ネックをひき起こす大きな原因になる。ここでは、PL が普通の物理的、レコード・レベルのロック TL に比べて通常速いことを示す。

本稿では、一つの環境を仮定する。この環境では、すべての資源をトランザクションの初めに宣言するようにと要求することはありそうもないものとする。われわれは Ries と Stonebraker の結論<sup>[5]</sup>に賛成である。したがって、以下、レコード・レベルのロックをするシステムと比較する。

$n$  = 並行処理されるトランザクション数

$n_i$  = パス集合の中のレコード数

$n_r$  = トランザクションによって要求されるレコード数

としたとき、

$$n_i \geq n_r$$

であるとする。また、

$n_u$  = 典型的なトランザクションの変更回数

$c_t$  = レコード（物理的）ロックを1回調べるのに必要な費用

$n_p$  = 1個のトランザクションによって生じる述語の数

$c_p$  = 述語と更新の突き合せを1回行うのにかかる費用

とする。

TL のロック・リストを初めから順番にみていくと仮定して、1個のトランザクションによる TL のオーバーヘッドは次のようにして求められる。  $n_u$  回の更新ごとに、他のトランザクションによってロックされている  $(n-1)n_i$  個のパス・レコードの各々と突き合せしなければならない。同時に  $n_i$  個のパス・レコードの各々は、他のトランザクションによってロックされている  $(n-1)n_i$  個の変更レコードの各々と突き合せしなければならない。したがって、オーバーヘッド  $O_i$  は、次のようになる。

$$O_i = [n_u \underset{\substack{\uparrow \\ \text{変更}}}{(n-1)n_i} + n_i \underset{\substack{\uparrow \\ \text{検索}}}{(n-1)n_u}] c_t = 2(n-1)n_i n_u c_t$$

同様に、1個のトランザクションによる PL の平均オーバーヘッドは、

$$O_p = 2(n-1)n_p n_u c_p$$

となる。このときの  $O_p$  と  $O_t$  の比率,

$$\frac{O_p}{O_t} = \frac{c_p \cdot n_p}{c_t \cdot n_t}$$

は、重要なパラメタを示している。すなわち、精密ロックは

$$\begin{aligned} c_p n_p &< c_t n_t \\ \therefore c_p / c_t &< n_t / n_p. \end{aligned}$$

の場合には安上がりである。

これは次のことも示している。すなわち、PL の述語と更新の突き合せを 1 回行う費用の増加が、突き合せ回数の減少に見合っている限り、PL は TL より安上がりである。確かに  $c_p \geq c_t$  であるが、しかし、それゆえ  $n_p \leq n_t$  である。

1 個の述語はたくさんのレコードを記述するだろうから、通常  $n_r \gg n_p$  である。システムの索引機能やアクセス方法では通常たくさんのレコードに触れることになるので、 $n_t \geq n_r$  である。これら二つの条件から、通常  $n_t \gg n_p$  となる。

たとえば、 $p = a \vee b \vee c$  としよう。また、 $c_t = 1$  と仮定する（すなわち、1 回のデータベース・キーの比較）。 $c_p$  の妥当な予測値は 5、すなわち各項と各演算子に 1 単位ずつであらう。

システムが 5 個以下のレコードを調べることで済ませられない限り ( $n_t > 6$ )、PL の方が安上がりである。もし、 $n_r = 1$  でかつシステムがそのレコードの場所をハッシュ法で直接みつけることができたとすると、TL の方が 5 倍速い。しかし、200 レコード入っているページをシステムが端から端まで捜すとすると、PL の方が 40 倍速い。また、 $p$  を満たすレコード数  $n_r$  が 5 より大きければ、 $n_t$  がいくつであっても、PL の方が速い。

TL はロック・リストを初めからみていくような方法は採らないだろう。動的線形ハッシュ法に関する最近の文献<sup>[6]</sup>の結果から、われわれは TL のオーバーヘッドの式を導いた。平均専有率 80 パーセントと仮定すると、およそ 8 回の読み出し/書き込みが、同じバケツの大きさをしたハッシュ表を捜すときの捜し損いの回数として必要である。すべての  $(n_u + n_t)$  回のトランザクションのロックをハッシュ表に配置する費用は、したがって次のようになる。ただし、 $c_h$  はハッシュの費用を表わす。

$$O_t' = (8c_t + c_h)(n_u + n_t) \tag{4-1}$$

われわれは、前に PL のオーバーヘッドを次のように導いた。

$$O_p = 2(n-1)n_p n_u c_p \tag{4-2}$$

$r = O_t' / O_p$  を求めると、次式のごとくなる。

$$r = [(8c_t + c_h) / c_p] [(n_u + n_t) / \{2(n-1)n_p n_u\}] \tag{4-3}$$

どのような妥当な述語に対しても、

$$(8c_t + c_h) / c_p > 1 \tag{4-4}$$

であり、通常、

$$n_t \gg n_u \tag{4-5}$$

である。

式 (4-4) と (4-5) を (4-3) に適用し、かつ  $n-1$  を  $n$  で置き換えると、

$$r > n_t / (2nn_p n_u) \tag{4-6}$$

となる。

PL は  $r > 1$ 、または

$$n < N = n_t / (2n_p n_u) \tag{4-7}$$

の場合に TL より速い。

述語のメカニズムはユーザのデータに対するすべての要求を指示できるので、 $n_p$  はしばしば 1 である。このことと (5-5) とをあわせると、 $N$  は大きな数になる。

興味ある二律背反がここで生じる。PL は並行処理の度合いが非常に大きくない限り、TL より速い。われわれは、PL が TL よりも並行処理の度合いを大きくすることを知っている。

$n_0$  = ロックの費用が等しくなるようなロックのレベル

$n_1$  = TL による最大の並行処理の度合い

とする。もし、 $n_1 < n_0$  ならば PL は TL より常に優れている。しかし、もし  $n_1 > n_0$  ならば TL の方が速くなるような場合がある。

$n_1$  は  $n_i$  の単調減少関数、 $n_0$  は  $n_i$  の単調増加関数なので、 $n_i$  が大きくなるにつれ  $n_1$  は  $n_0$  より小さくなる。再び索引メカニズムやアクセス方法が  $n_i$  を小さくできない限り、PL は TL より速い。上の費用分析はロック・リストの変更の費用を考慮していないので、まったく不完全である。しかし、この分析により TL と PL の費用の差が十分に示されている。

## 5. おわりに

ロック・システムが犯す 2 種類の誤りを述べた。どのシステムもその中の一つを簡単に取り除くことができる。しかし、筆者らが知っているどのシステムも 2 番目の種類の誤りを犯すことが明らかになった。精密ロックは、この誤りも取り除くことを述べた。すなわち、整合性を保つのに必要なレコードの精密な集合がロックされることを示した。

トランザクションを良型かつ 2 段階であるようにつくる負担は、ユーザが精密ロックを用いられなくなってしまうことを示した。また、同時に述語が任意でよいことも示した。

精密ロックの実行コストを物理的なレコード・レベルの方法と比較した。そこに表わされたパラメータと例題からは、精密ロックの方が通常速いという結論を容易に得ることができる。

(プロダクト・サポート統括 2 部 開発 2 部 本田都南夫 訳)

- 参考文献 [1] K. P. Eswaran, J. N. Gray, R. A. Lorie, I. L. Traiger, "The Notions of Consistency and Predicate Locks in a Data Base System," *CACM*, Vol. 19, No. 11, 1976.
- [2] M. M. Astrahan, (and 15 other authors), "A History and Evaluation of System R," *IBM Research Report RJ 2843(36129)*, IBM Research Laboratory, 1980.
- [3] S. Lee, R. T. Yeh, "Structural Locking for Concurrency Control in Data Base Systems," *COMPSAC Proceedings*, Nov. 1979, pp. 578-582.
- [4] S. C. Kleene, *Introduction to Metamathematics*, Van Nostrand, 1952, p. 204.
- [5] D. R. Ries, M. Stonebraker, "Effects of Locking Granularity in a Database Management System," *ACM TODS*, Vol. 2, No. 3, Sept. 1977, pp. 233-246.
- [6] D. R. Ries, M. Stonebraker, "Locking Granularity Revisited," *ACM TODS*, Vol. 4, No. 2, June. 1979, pp. 210-227.
- [7] D. Potier, Ph. Leblanc, "Analysis of Locking policies in data base management systems", *CACM*, Vol. 23, No. 10, Oct. 1980.
- [8] W. Litwin, "Linear hashing: a new tool for file and table addressing," *Proc. 6th International Conf. on Very Large Data Bases*, Oct. 1980, pp. 212-223.

訳者注 [注] あるトランザクションが動いている間、初めのうちは求めてもいなかったレコード、すなわちファントムがそのトランザクションが終了する前に、別のトランザクションによって追加される問題をさす。

執筆者紹介 John R. Jordan

1963年に数学で B. S., 1970年と1974年にコンピュータ・サイエンスでそれぞれ M. S. と Ph. D を Iowa 州立大学より取得。1963年から1965年までは Ames 研究所で技術計算プログラミングに従事。1965年と1966年は Boeing 社でプログラミングに従事。1966年から1974年には Iowa 州立大学で教鞭をとる。1974年に Sperry Univac 社に入社、現在 Consulting Computer Scientist である。



Jayanta Banerjee

1979年に Ohio 州立大学よりコンピュータ・サイエンスで Ph. D. を取得。同年 Sperry Univac 社に入社、現在 Principal Systems Programmer。関係データベース・システムの製作に従事。興味をもつ分野は、データベース・システム、データ・モデル、データベース・マシン、分散処理などである。



Ronald B. Batman

Temple 大学より数学で B. A. を取得。Case Western Reserve Medical School で医学を学ぶ。Sperry Univac 社に入社後、Ohio Bell, SAS, ALCOA などのプロジェクト・マネージャに従事し、1963年よりトランザクション処理システムとデータベース・システムのシステム設計と開発に従事。現在、データベース処理系などを含む周辺機器とデータ・ストレージ・システムの設計担当マネージャ。興味をもつ分野は、データ・モデル、分散処理システム、高性能データ管理システムの実装などである。

## ソフトウェアの品質管理と品質コスト

西島 政信

## 1. はじめに

総合的な見地から品質管理を進めていくうえで重要なことは、企業あるいは組織における品質コストの確認、解析および管理である<sup>[1]</sup>。

近年、ソフトウェア・ライフ・サイクルのトータルコストの低減が重要視されているが、これを限られたリソースの中で、全体的なバランスをみながら効果的に進めていくためには実績ベースのデータはもちろんのこと、長期的なレンジでソフトウェアの品質および生産性向上を検討していけるような戦略的なデータが必要になる。

製造工業で広く一般に使われている品質コストは、総合的な品質管理を進めていくうえでの重要な概念として、米国 G. E. 社の A. V. Feigenbaum 氏<sup>[1][3]</sup>によって示された。その主たる目的は、品質管理活動のどの部分にコストを投入するのが最も適切かを決定したり、計画したりするところにある。この品質コストからのアプローチは、従来の検査中心の品質管理活動から総合的かつ経済的な品質管理活動を展開していくうえで、一つの指針を与えてくれるものである。

本稿は、ソフトウェアの品質管理活動を分析評価し、製品品質の管理と改善を効果的に進めていった例を紹介する。すなわち、製造工業で一般に使われている品質コストとそのコスト概念を、ソフトウェアの品質管理にいかに応用していくべきかについて述べる。

## 2. 品質コストとコスト概念

J. M. Juran は、その著書<sup>[3]</sup>の中で、製品のライフ・サイクル・コストとして、次の9種類をあげている。

- ① 何が顧客の要求している品質であるのかを発見するための市場調査のコスト
- ② これらの品質要求に合致する製品の構想を生み出すための研究・開発のコスト
- ③ 製品の構想を実際に製造が行えるようにする、いわゆる設計のコスト
- ④ 製造計画のコスト
- ⑤ 基礎的な製造コスト
- ⑥ 試験あるいは検査のコスト

- ⑦ 不良防止のコスト
- ⑧ 品質失敗のコスト
- ⑨ 品質保証のコスト

J. M. Juran は、「種類④、⑤は製品を製造すれば必ず発生するコストである。種類⑥、⑦、⑧、⑨は品質試験、品質検査あるいは品質保証に要するコストで、程度の差はあるが避けることができるコスト（これがいわゆる品質コストと呼ばれるもの）である。残りの①、②、③は調査、研究、設計に要するコストで、このコストは品質コストと品質コスト以外のコストに分解するのは困難である」と述べている。

品質コストとは製品のライフ・サイクル・コストの中の⑥、⑦、⑧、⑨に関連するコストで、品質コストを分析する主たる目的は回避可能な⑥、⑦、⑧、⑨のコストをできる限り減少させるところにある。

A. V. Feigenbaum は、その著書<sup>[1]</sup>の中で、品質コストとコスト概念について詳細に述べているので、その概要を次に紹介する。

## 2.1 品質コストの体系

A. V. Feigenbaum は、生産活動に要する諸コストの中で、品質コストは製品品質の管理と改善に大きく関連するものとして、次の三つに大別している。

- ① 予防コスト……品質計画および不良を未然に防止することに関連したコストの総称である。これは、評価コストおよび失敗コストを低減させる投資の意味における予防コストであり、実質的には品質管理技術の研究、品質管理教育関連のコストがこの中に含まれる。
- ② 評価コスト……設定された品質水準を維持しているかどうか、製品の評価に関連したコストの総称である。品質試験、品質検査関連のコストがこの中に含まれる。
- ③ 失敗コスト……失敗コストは、社内失敗コストと社外失敗コストの二つに大別される。

●社内失敗コスト：社内における設計、製造仕様に合わない不良品によって生じるコストの総称である。手直しなどにもとづくコストがこの中に含まれる。

●社外失敗コスト：顧客に提供された製品の不良対策、あるいは不良処理に要するコストの総称である。顧客からのクレーム処理などに要するコストがこの中に含まれる。

## 2.2 コスト概念の概要

A. V. Feigenbaum は、予防コスト、評価コスト、失敗コストの関連を次のように述べている。



「一般に検査中心の品質管理下におけるコストの割合は、失敗コスト70パーセント、評価コスト25パーセント、予防コスト5パーセントである。歴史的にみて、検査中心の品質管理下における失敗コスト、評価コスト、予防コストの関係は、失敗コストが増加すれば、評価コストも増加する傾向をもっている。評価コストは一担増加すれば、それを引き下げるのは非常にむずかしい。なぜなら、不良品がたくさん作られたとすれば、当然の結果として失敗コストは高くなる。失敗コストが高くなったときに一般にとられる対応策は、より厳重な品質試験あるいは品質検査である。したがって、評価コストは失敗コストが高いままでいる限り少なくなならない。しかも、この失敗コスト、評価コストが高くなればなるほど、ますますその予防手段に手が回らなくなる。もし、このような状況にあるならば、総合的な見地から正しい予防法を確立し、このコストの悪循環を断ち切らなければならない。すなわち、品質コストのバランスをとりながら全体的なコストが下がるように予防コストを増加させ、失敗コスト、評価コストを減少させることが必要である。このコストの増加分は失敗コスト、評価コストの節約分でまかなえるものであり、会社全体の品質コストを実質的に長期にわたって増加させるものではない。なぜなら、予防コストの増加による不良品の減少は、厳重な品質試験あるいは品質検査の必要性を緩和し、評価コストをも減少させるからである。」

これが、Feigenbaum のコスト概念である。

Feigenbaum のコスト概念の要点は、「予防コストの増加は、失敗コストの減少を誘導し、相対的に評価コストも減少させる」というところにある。これは、「コストの悪循環を断ち切り、全体のコストを切り下げていくには、品質管理活動に費やされている諸々のコストを三つの品質コストに分解し、どの部分にコストを重点的に投入すれば、品質管理活動を費用対効果の高いものにしていけるのかが分析できなければならない」ということを意味している。

### 3. 品質コストのソフトウェア分野への適用

このコスト概念をソフトウェアの分野に適用するには、次のような項目について検討する必要がある。

#### 3.1 品質コストの適用可能性の検討

品質コストとコスト概念については、いくつかの論文の中で使用上の問題点<sup>[2]</sup>が指摘されている。その本質的な問題は、品質コストが原価低減のためのコストであって、収益拡大のためのコストではないということである。したがって、本稿の例でとり上げた部署（基本ソフトウェアの開発、改良、保守およびこれに付随する各種サービスを主業務として担当している）のように、収益拡大よりも原価低減を目標としなければならないような組織（コスト・センター）においては、この品質コストは適用可能であり、管理のための有用な手段となりうるものである。

#### 3.2 品質コストの分類と再編成

品質コストを使用するうえで重要なことは、業務を構成している諸々のコストがどのコストに属する

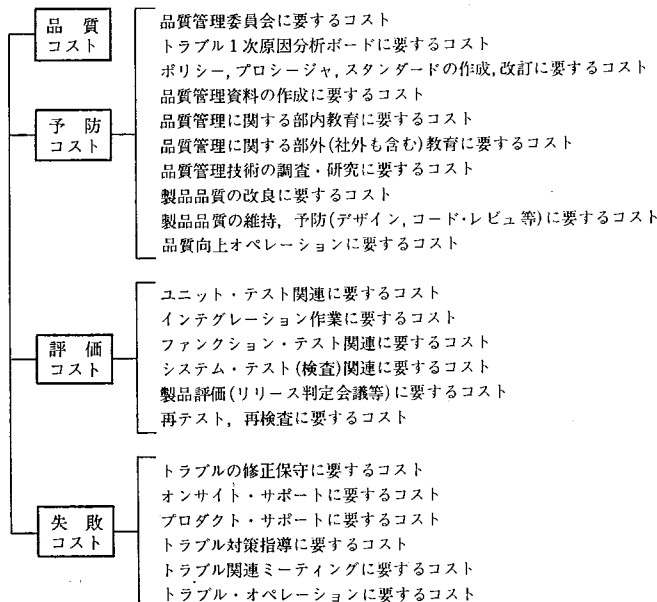


図1 品質コストの再編成

かを明確にし、組織の性格と業務に合った形で品質コストを分類し、再編成することである。図1は、とり上げた部署における業務および作業の主なものを用意、評価、失敗の各コストに再編成したものである。この品質コストの再編成が不適切であれば、品質コストを正しく把握できず品質管理活動を正しく評価することができなくなる。

### 3.3 品質コスト・データの収集と集計

品質コスト・データは、できる限り正確に把握できなければならない。しかし、そのためにデータの収集と集計に手間と時間がかかるようでは、かえってコスト高になってしまうおそれがある。したがって、データの収集、集計を機械化できるような仕組みが必要になる。現在、この業務別あるいはプロジェクト別のコスト実績の把握と生産性測定を目的とした全社的なレポーティング・システム (Project Reporting System: PRS) が運用されているが、このようなシステムを活用すれば、品質コスト・データの収集と集計を容易に機械化できる。

### 3.4 各種品質データとの関連

品質コストは品質問題の確認や潜在的な管理不良を発見するために役立つものであるが、具体的なアクションは他の品質関連データと有機的に結合させたうえで総合的に決定されるべきものである。したがって、品質コスト・データと他の品質データ（たとえば、ソフトウェアのユーザ使用報告の統計データ、トラブル1次要因分析データなど）との関連を明らかにし、各種品質情報の整備も合わせて検討しておく必要がある。

### 3.5 品質コストの分析方法の明確化

品質コストの使用目的を明確にし、その使用目的に合った分析法をあらかじめ検討しておく必要がある。品質コストによる問題確認の着眼点は、

- ① 期間の品質コストの総額は減少しているか。
- ② 予防コストが増加しても失敗コスト、評価コストは増加しているか。
- ③ 失敗コストより予防コストが多くかけられていないか。
- ④ 失敗コストの減少度合はどうか、またその内容の変化はどうか。それは社内失敗コストの減少か、社外失敗コストの減少か。
- ⑤ コスト項目の計画と実績はどうか。
- ⑥ 評価コストと社外失敗コストの関係はどうか。評価コストを減少させても社外失敗コストは増えていないか。

などである<sup>[4]</sup>。

## 4. 品質コストの応用

品質コスト・データは、総原価の最小点の発見や維

持あるいは低減ということに関して、ソフトウェアの分野においても以下のような応用が可能である。

- ① 品質管理活動の効果を測定する手段として……予防のために費やされるコストは、失敗コスト、評価コストの減少を導くものでなければならない。予防のために先行投資を行ったにもかかわらず、失敗および評価コストが下がらなかったとすれば、それは予防コストの使い方に誤りがあると考えられ、品質管理活動の再点検の指針とすることができる。
- ② 工程を解析する手段として……どの工程のどの業務、あるいは作業にどの程度の失敗コストを費やしているかを明らかにし、トラブルの1次原因の分析結果などをもとに、何がその原因になっているのかを解析するための手段とすることができる。
- ③ 品質管理活動の予算編成の手段として……どの工程のどの業務・作業に問題があったのか、またどの工程の業務・作業で事前に処置されるべきであったのか明らかになったならば、その問題の程度に応じた品質管理プログラムを作って段階的に改善を進めながら工程を遂行する。それには、全体的なバランスを考慮して品質コストを適切に配分していかなければならない。品質コストは、決定された品質管理プログラムを達成するために必要な経費を予算化するうえで、重要な役割を果たすことになる。

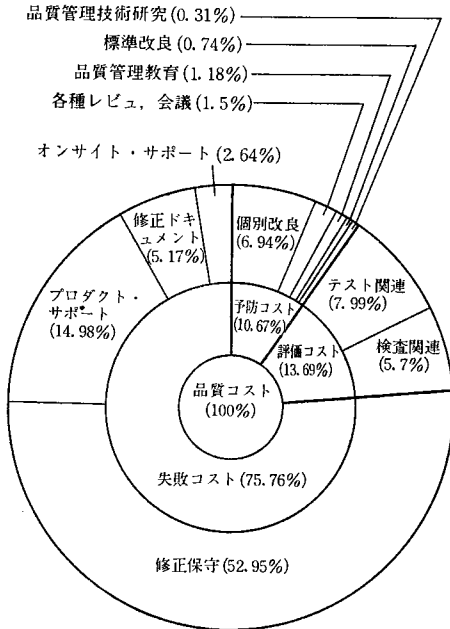
## 5. 品質コストの分析例

A. V. Feigenbaum は失敗コスト65パーセント、評価コスト25パーセント、予防コスト5パーセントが典型的なコストのバランス（検査中心の品質管理下における失敗コスト、評価コスト、予防コストの割合は、それぞれ70パーセント、25パーセント、5パーセント）であると述べているが、これは製造工業における品質コストのバランスである。これが、そのままソフトウェアの分野に適用できるかどうか、現在のところ明らかでない。したがって、半期あるいは四半期ごとの品質コストのデータから、現状の品質コストの配分がどうなっているのかを把握することが先決である。

図2はある期間のPRSのデータをもとにしたプロダクトAの品質コストの分析結果である。表1は同期間に処理されたクレームの統計結果であり、表2はA. V. Feigenbaumの品質コストのバランスからみたプロダクトAの品質コストの比較表である。

問題を簡単にするために、前提条件

- ① 既知のトラブルは、新規のトラブルに比べて



数字は金額ベースではなく、投入時間ベースのもの

図2 プロダクト A における品質コスト分析結果

表1 プロダクト A のトラブル統計結果

分類	割合 (%)	件数
新規トラブル	45.8	487
既知トラブル	54.2	576
計	100.0	1063

表2 品質コストの比較表

品質コスト	Feigenbaum の品質コスト	プロダクト A の品質コスト
	割合 (%)	割合 (%)
予防コスト	10	10.67
評価コスト	25	13.69
失敗コスト	65	75.76
計	100	100.00

件数が多いけれども処理済みであるため、そのコストは無視できる。

- ② 社内で発見されたトラブルに要するコストは、社外で発見されたトラブルに要するコストの 1/2 とする。
- ③ 当分は同じ水準で新規のトラブルが発生する。

のもとで考えると、種々のことがいえる。

- ① 新規トラブル 1 件に要している失敗コストはどのくらいなのか。
- ② 評価コストにどれだけのコストを先行投資すれば、テストあるいは検査で年間何件以上の新規のトラブルを発見する必要があるのか。
- ③ この目標が達成できれば、失敗コストをどれ

くらいに切り下げられるのか。

- ④ 削減できたコストをどの部分に、どのくらい予防コストとして投入できるのか。

などである。

予防あるいは評価のために、どの程度のコストを具体的に投入するかについては、さらに高度なマネジメント・レベル (社内および社外環境など) の判断が必要である。しかしながら、少なくとも品質コストの分析は、マネジメント・レベルにとって製品品質を管理し、改善していくうえでの貴重な情報の一つになりうるものである。

6. おわりに

ソフトウェアの品質および生産性向上のためになさなければならないことは非常に多い。品質管理活動もその一つである。品質コストは、製品品質の管理と改善を効果的に進めていくための一手段となりうるものであるが、これをソフトウェアの分野に適用し、本来の効果を発揮させていくには品質コスト・データを積み上げ、さらに研究していかなければならないだろう。

また、品質コスト以外のコスト、たとえば市場調査コスト、研究開発コスト、設計コストなどについても品質コストとの関連性を明らかにしていく必要があるだろう。

なお、第 2 回ソフトウェア品質管理シンポジウム (昭和 57 年 9 月、(財)日科技連主催) において、本稿と同主旨のものが発表されたことを付記する。

参考文献

- [1] A. V. Feigenbaum, *Total Quality Control*, (日立製作所訳, 「総合的品質管理」, 日科技連, 1967).
- [2] 千住鎮雄, 水野紀一, 「品質管理のための経済計算」, 日科技連, 1971.
- [3] J. M. Juran, *Quality Control Handbook*, (東洋レーヨン(株)訳, 「経営革新のための品質管理」, 日科技連, 1966).
- [4] 深井秀夫, 「動態コスト・マネジメント」, 中央経済社, 1971.
- [5] 市川竜三, 「総合的品質管理」, 日刊工業新聞社, 1964.
- [6] 永作忠一他, 「品質管理の計画と実施」, 日科技連, 1968.
- [7] P. B. Crosby, *Quality is Free*, (小林宏治監訳, 「クオリティ・マネジメント」, 日本能率協会, 1980).
- [8] 森口繁一, 「品質管理」, 岩波書店, 1979.

(プロダクト・サポート統括二部 開発 2 部)

**データ通信回線の第2次開放**

— 守 田 洋 —

**1. はじめに**

公衆電気通信法が改められ、昭和57年10月23日施行された。この改訂は昭和46年の、いわゆるデータ通信回線の第1次開放に続く第2次開放といわれ、データ通信回線の利用できる分野が法律上、大幅に広げられた。郵政省は、これに関連し二つの省令を同日付けで公布している。以下、その主な変更点を述べる。

**2. 共同使用**

共同使用とは、同一の特定通信回線を2人以上で共同して使用する形態を指す。回線の利用目的がデータ処理を行うことであれば、今次の開放により回線の自由な使用が認められるようになった。また、メッセージ交換と電信電話的利用についても無条件ではないが、認められることとなった。

**2.1 中途コンピュータでメッセージ交換をする共同使用**

図1に示した例では、A社の中途コンピュータでメッセージ交換ができる。交換されるメッセージは、A社またはB社のコンピュータで処理するためのものでなければならない。

**2.2 端末機間通信を行う場合の共同使用**

図2の例は、データ処理のための端末機間通信である。

データ処理、中途コンピュータなどの解釈については、これに対応した郵政省令や運用実績を注目する必要がある。

**2.3 電信電話的利用ができる共同使用**

共同使用する当事者間の関係が一定の条件を満たした場合は、特定通信回線を共同使用し、メッセージ交換はもちろん、データ通信のためだけでなく、電話回線としての使用、ファクシミリ回線としての使用など、電信電話的利用も可能となる。

求められる主な場合の条件は、業務上緊密な関係があり、その間に通信を必要とし、さらに次のいず

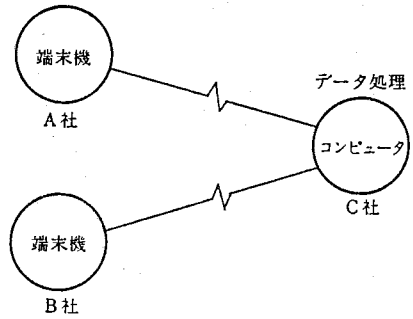


図2 データ処理のための端末機間通信を行う共同使用

れかに該当しなければならない。

- ① 発行済み株式総数、または出資総額の10パーセントを超える株式や資本の所有関係にあり、業務上継続的な取り引きなどの関係にあること。
- ② 少なくとも当事者の一方について、相手方との取り引き高が自己の総取り引き高の20パーセント以上であり、業務上継続的な取り引きなどの関係にあること。
- ③ 業務提携などの継続的な契約があり、密接な共同関係にあること。

**3. 他人使用**

特定通信回線の使用契約は、電電公社の直接の顧客（2者以上のこともある）と電電公社との間でなされる。

ところで計算センターなどの例では、計算センターが回線使用契約当事者となるが、計算センターの顧客はこの契約当事者でないという意味で、他人といわれる。他人使用とは、このような形態で他人が特定通信回線を利用することを指す。他人使用についても、今次開放によってユーザのコンピュータとの接続、中途コンピュータでのメッセージ交換、端末機間通信などが、制度上可能となってきた。

ただし、メッセージ交換サービス（付加価値通信

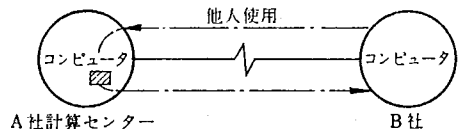


図3 コンピュータ間接続をする他人使用の形態

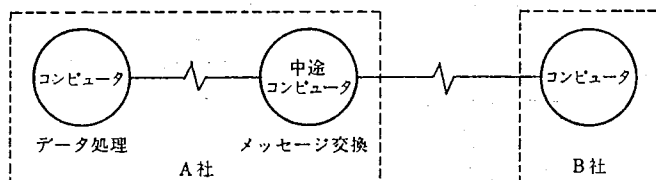


図1 中途コンピュータでメッセージ交換を行う場合の共同使用

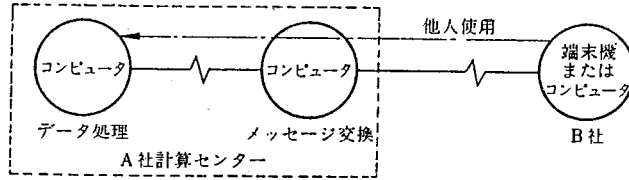


図4 データ処理のためのメッセージ交換をする他人使用の形態例

など)については、主として中小企業を対象を限定し、暫定的に認められるものであり、実施細則としての郵政省令では、おおむね、

- ① 対象となる取り引き関係を中小企業相互間に限定せず、中小企業と大企業との取り引き関係にも拡大する。
- ② 業務上、その間の通信を必要とする。
- ③ 中小企業の範囲は資本金1億円以下とする。

などとされている。

3.1 他人使用の形態 (1)

図3の例は、他人に相当する第三者が端末機かわりにコンピュータを接続してもよいことを示す。

3.2 他人使用の形態 (2)

図4の例では、中途コンピュータでメッセージ交換ができる。交換されるメッセージはデータ処理のためのものであることが必要である。

3.3 他人使用の形態 (3)

図5の例は、入力データを受け付けたコンピュータと出力コンピュータが異なってよいことを示す。ただし、いずれか一方のコンピュータでデータ処理が行われることが必要である。

3.4 他人使用の形態 (4)

図6の例は、データ処理のための端末機間通信である。

3.5 他人使用の形態 (5)

図7の例は、計算センターが付加価値通信網を用

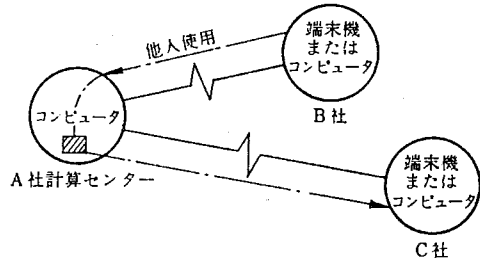


図6 データ処理のための端末機間通信を行う他人使用の形態例

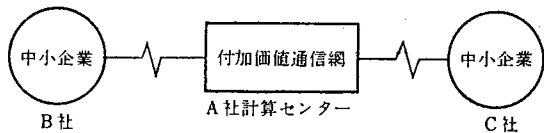


図7 中小企業間の通信を付加価値通信網により媒介する他人使用の形態例

いて、互いに業務上緊密な関係にある中小企業B、C間の通信を媒介することを示している(この取り扱いとは暫定的なものである)。

4. 相互接続

特定通信回線と公衆通信回線との接続、特定通信回線と私設通信回線(構内回線、自営回線)との接続などを相互接続という。ある地域の多数の端末機(装置)を集配信装置と公衆通信回線により結び、集配信装置と中央のコンピュータとを特定通信回線で結ぶと(図8)、多数の端末装置を直接中央のコンピュータと公衆通信回線で結んだ場合(図9)に比べ、データ通信回線の使用料金を少なくしうる。料金低減の程度は、集配信装置と中央コンピュータ間の距離、集配信装置に収容する公衆通信回線の数、特定通信回線の実使用時間などにより変わってくる。今後の開放により、

- ① いわゆる 公一特一公 の接続とならない
- ② データ処理のための接続である
- ③ 他人の通信の媒介を行わない

という三つの基準に適合すれば公衆通信回線と特定回線との接続が自由にできる。さらに上記の基準に適合しない場合でも、郵政大臣の個別認可を受ければ可能となっている。また、特定通信回線とDDXの回線交換サービス網・パケット交換サービス網との相互接続についても同様となった。このほか、特

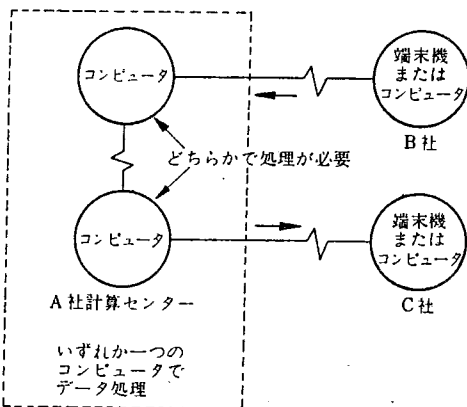


図5 入力データを受け付けるコンピュータと出力コンピュータが異なる他人使用の形態例

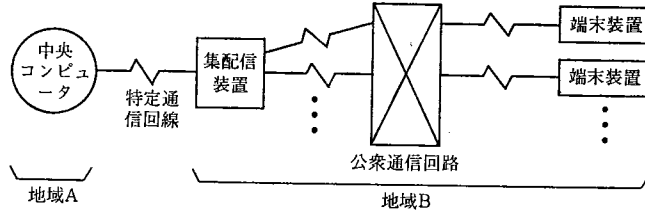


図 8 相互接続の例

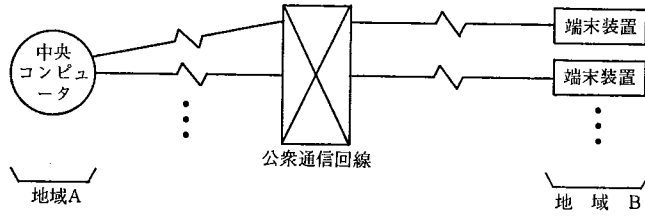


図 9 相互接続をしない例

定通信回線と私設通信回線との接続，異名儀の特定通信回線の接続，構内交換電話（PBX）とコンピュータの接続が条件付きであるが，可能となった。

参考文献

- [1] テクニカル・ダイジェスト・シリーズ1. データ通信入門＝制度編，日本ユニパック（株），1982.

（プロダクト・サポート統括二部 開発企画部）

—J. D. Foley, A. Van Dam 著—

### “Fundamentals of Interactive Computer Graphics”

Addison-Wesley, xx+664 pp., 1982.

グラフィック・ディスプレイを用いて対話形式で図形を操作することに代表されるコンピュータ・グラフィックスの世界は、グラフィック・ディスプレイが開発された当初から、その活用は大きな広がり可能性をもっていた。にもかかわらず、最近まで、そのハードウェアが高価であり、プログラム開発がむずかしいという理由から比較的限られた分野のごく一部の人々にしか利用されていなかった。ところが、ここ数年、エレクトロニクス技術の急速な発展と相まって、容量が大きく読み書きの速い記憶装置を必要とするラスタ・スキャン型グラフィック・ディスプレイが、コンピュータの入出力装置の一つとして急速に普及しつつあり、またグラフィック・ディスプレイ装置そのものも機能を拡大しつつある。これにより、これまでの文字によるコンピュータとの対話に代わって、図形や映像による対話の道が開けた。これらのことから、コンピュータ・グラフィックスは、複雑な現象のシミュレーションによる解析や設計、さらに教育や美術の分野での利用の拡大と発展が期待される。

本書は、その序文にも示してあるとおり、コンピュータ・グラフィックスに関心のある者に対しての教科書、あるいは参考書として活用できるものであり、プログラミングと構造をもつデータの操作に関する基本的な知識、およびコンピュータ・ハードウェアの構成に関する知識があれば、グラフィックスの経験がとくになくても理解できるように編集されている。

コンピュータ・グラフィックスを理解するうえで、グラフィックスのハードウェア、ソフトウェア、データ構造、座標変換など図形モデルの数学的操作、オペレータ・インタフェースの問題など、多岐にわたる分野の事柄を理解する必要がある。しかしながら、本書は基礎から複雑で特異な問題までを段階的に分類し順序立てて解説しているので、本書の章立てに従って読むことにより、容易に理解を深めていくことができる。全体は、17の章からなっているが、大きく四つのカテゴリに分かれる。

第1部は“コンピュータ・グラフィックスの基礎”である。コンピュータ科学の他の分野でもみられるように、グラフィックスの分野でも自ら試してみるものが理解を早める道である。本書においても、第1章でコンピュータ・グラフィックスの意味、応用分野、歴史など概要を述べたあと、第2章ですぐにグラフィックスを応用したプログラミングを始めている。そのため、節を進めるたびにグラフィックスの応用の中ではどのような問題があるのかが具体的に理解できる。コンピュータと対話するということが対話型グラフィックスの生命であることから、ここでは操作者がなじみやすい操作性を作り出すためのプログラム構造について、とくに詳しい解説がある。そして、徐々に展開されていくプログラム例の中でグラフィックスに描画するうえでの留意点、あるいは入力操作に対する考慮などが述べられている。プログラム例を記述するために用いている言語は、Pascal と米国 ACM-SIGGRAPH/GSPC によって提唱された標準グラフィック・サポート・システム CORE のサブセットである。

第3章と第4章では、ハードウェアの基礎と第2章で使用した CORE のサブセットである単純なグラフィックス・パッケージ SGP の具現化を想定したソフトウェアのコンセプトを述べている。第3章では、主に描画の手法の違いからくるハードウェアの種類、手法、構成などが記述されており、第4章では主にウィンドウ操作、クリッピング処理、表示データのセグメンテーションとその管理、入力操作のロジックについて詳述されている。つづく第5章では、グラフィック・ディスプレイと接続して使用できる種々の入力装置の機能や目的の紹介とともに特性を生かした入力技法が述べられ、さらに第6章では、操作者とコンピュータのインタフェースについての設計基準設定の考え方が述べられている。ともに、オペレータ・インタフェース設計のガイドランスとして有用である。

第2部は“図形操作に用いられる数字、データ構造、表示制御のハードウェア構成”の解説である。対話型での図形操作は平行移動、回転、拡大縮小などの基本操作の組み合わせとして行われるが、本書では第7章、第8章で、2次元、3次元の幾何的操作に関する数学的手法と投影に関する基礎について詳述している。そして、第9章では表示したい対象モデルについての図形の階層的構造の組み立て方について言及する。さらに第6章では、ディスプレ

イ・プロセッサの構成に関しての解説や、個々の機能をどう分化して配置すべきかについての解説があり、例として具体的な高級型のグラフィック・ディスプレイの紹介もある。

第3部は“ラスタ・スキャン型ディスプレイ”についてである。すなわち、第11章では、第2章で述べたラスタ・スキャン型の基本技術をさらに拡張し、適切に装置独立の論理を適用できるような手法を紹介し、ラスタ・スキャン型ディスプレイでベクタ・スキャン型ディスプレイと同様の描画をするラスタ・グラフィックス用ソフトウェアを紹介する。そして第12章では、現在のラスタ・スキャン型ディスプレイのハードウェア技術について言及している。

第4部は“グラフィック・ディスプレイで写真を作る”である。この部分はコンピュータ・グラフィックス分野で最も技術的に高度な分野であり、また同時に、技術的にも新しい分野で多くの人々の興味をそそる分野でもある。第13章では3次元の曲面を表示するための基本的な手法を紹介し、第14章では、3次元の物体をより現物に近い表現で画を作り出すための一般的な方法を紹介している。また、従来からあった隠面消去の問題の記述が第15章にある。第16章では陰影や表面反射のアルゴリズムや種々の明暗モデルについて詳述し、第17章には色調の定義と表示に関する解説がある。

以上、本書の構成についてごく簡単に紹介してきたが、本書はこのような章立てから、コンピュータ・グラフィックス分野の初心者からこれまで長くこの分野にかかわってきた者まで、幅広く参考書として活用できる内容である。本書の中で索引された文献が巻末にすべてまとめられているので、その内容の幅の広さと奥行の深さに圧倒されるばかりでなく、特定の応用分野でのグラフィックスの利用はコンピュータ・グラフィックスの世界全体からみると幅の狭い利用でしかなく、また実用面では今後の課題として残されている分野もまだ多くあると認識させられる。

山下 英男 監修  
日本ユニバック総合研究所編

共立 総合コンピュータ辞典 第2版

共立出版, A5判, xviii+1400 pp., 1982.

本辞典が、はじめて出版されたのは昭和45年3月のことであった。途中、51年に出版元が現在の共

立出版(株)に移ったため、今回の改訂は同社にとって「第2版」であるが、前出版元から数えると実質的には「第4版」となる。本辞典の最大の特色は、過去12年余にわたり、同じ編著者によって継続的にその内容が更新されてきたことにある。計算機科学の進歩はあまりにも急速であるため、その全体像を把握することは容易でなく、また各分野で使用される用語も統一した概念で使われているとはいえない。

このため、4～5年に1回の割合で全面改訂を施さないと、この種の「辞典」はたちどころに陳腐化するということを考えると、初版以来3回の改訂はむしろ当然であるともいえる。

今回の新版は、用語編の3700語に新しい用語を収録した努力の跡がみられ、解説編については、「プログラムと人工知能」、「文献検索」の2項目を削除し、新たに「マイクロコンピュータ」、「自然言語処理」、「オフィス・オートメーション」の3項目が追加されたほか、他の項目についてもそれぞれの状況に応じて大幅な加筆、および修正が加えられている。

今回の改訂の概要は次のとおりである。

#### (1) 用語編

- 前版の見出し語のうち、一般によく知られている用語や比較的重要度の低い語など、約400語を削除し、ACMやIEEEなどの最近の各種文献にみられる専門用語の中から新たに約1700語を選んで追加、合計で約3700語とし、全面的な書き直しが行われている。
- 計算機分野の発展に重要な貢献のあった科学者や技術者の人名が掲載され、その略歴や業績、エピソードなどが紹介されている。

#### (2) 解説編

解説編は、さらに「データ処理技術の背景」、「基本的なソフトウェアとプログラミング」、「計算機のハードウェア技術」、「応用分野」に分かれる。

- データ処理技術の背景：単に計算機ハードウェア、ソフトウェアにとどまることなく、その基礎となる計算の理論、計算機利用形態、関連分野への影響などについて歴史的な経緯を解説しているほか、プログラミングの技術や、そのあり方を考えるうえで必要となる基礎理論については、計算量理論やScott理論に関する解説が追加されている。
- 基本的なソフトウェアとプログラミング：「プログラム言語」の項では、Adaや抽象データ型言語に論及しているほか、言語の形式的定義の解説



が追加された。「データベース技術」の項では、関係型モデルに重点を置いた全面的な書き直しが行われたほか、従来とかく混乱しがちであったファイル編成法に関して、解説を試みている。「ソフトウェア工学」の項では、ソフトウェアのライフ・サイクルに沿って解説を展開していき、プログラマの職業的側面について言及した Ershov の講演でしめくくっている。

- 計算機のハードウェア技術：「論理回路とデバイス技術」、「記憶装置」は、半導体素子を中心に最新の記憶デバイスについて解説を試み、「計算機の制御」では、並列処理とデータ・フロー機械に言及している。「入出力装置」で日本語処理と音声認識を重点的にとりあげたことや、「マイクロコンピュータ」を追加したのも時代の要請を入れたものといえよう。
- 応用分野：「構造解析」、「数理計画法」、「CAD/CAM」、「CAI と CMI」の項は、最新の技術動向を反映して書かれており、「自然言語処理」では、構文に関する処理と意味に関する処理について

基礎的な事項が解説されている。また「オフィス・オートメーション」は、最近の議論や関心の高まりを考えると、時宜を得たもので、現状の問題点を把握するうえでの手助けとなろう。

### (3) 規格編

ISO, ANSI, ECMA などの情報処理分野の標準化活動に関する解説は、全面的に加筆、訂正されている。

JIS 規格として制定されているプログラム用言語については、前版と同じく ALGOL, COBOL, FORTRAN の最新言語仕様が収録されており（ただし、FORTRAN は上位水準だけ）、またマイクロコンピュータの普及を考慮して、最近制定された「基本 BASIC」も収録され、利用者の便宜がはかられている。

以上のように、用語編における見出し語数の増加、解説編における項目の補筆、追加によって総ページ数も旧版の 1132 ページに比べて、276 ページの増加になっている。

●SIGGRAPH '82 開催——Boston, 1982年7月  
26日～30日

本年は、チュートリアルとセミナーに分かれ、24のコースが開かれた。コースの主なテーマは、低価格グラフィック・システム、ラスターグラフィックス、ユーザ・コンピュータ・インタフェースのための心理学、装置独立なグラフィック・ソフトウェア、2次元および3次元アニメーション、画像合成、ビジネス・グラフィックス、グラフィックスとデータベース、オフィス・アニメーションと電子出版、地図作成、ソリッド・モデリング、自由曲面などである。このほか、パネル討論のテーマとして、グラフィックスの標準化の影響、ソリッド・モデリングへのアプローチ、音声認識、CAD/CAM教育、グラフィック入力技術、Videotex、コンピュータ・グラフィックスにおける健康と安全問題などである。このほかテクニカル・プログラムと称する一般講演が多数行われた。

●第6回ソフトウェア工学国際会議開催——東京、  
1982年9月13日～16日

今回は倉成正氏（情報産業振興議員連盟会長）を基調講演者に、G. M. Weinberg氏とF. L. Bauer氏、山田尚勇氏とを招待講演者として開かれ、26か国からの約1,300名（海外200名）の参加者があった。またパネル討論のテーマは、高品質ソフトウェアの保守、新しいプログラミング、形式化は有用か、ソフトウェア工学の教育と技術移転、知識ベース・システム、ソフトウェア工学に対する新技術の影響であった。このほか、9月13日には、同会議の内容の理解を助けるためのチュートリアル・セミナーが、ライフ・サイクル、工具・環境、言語、管理・品質保証・計量、理論、ヒューマン・ファクタに分かれて開かれた。主なテーマとして、ソフトウェア工学の工具と技法、要求仕様技術、ソフトウェア工学の理論的基礎、ソフトウェアのCADとCAM、ソフトウェア・プロジェクトの管理と人間工学、ソフトウェアの品質管理と保証、ソフトウェアの保守と評価、新しいコンピュータ・アーキテクチャに対応するソフトウェア工学、分散処理のソフトウェア、コンピュータ・ネットワークのソフトウェア、ソフトウェア生産の事例、などがとりあげられ、約240編の応募のうち、39編が選ばれた。

●COMPCON 82 Fall 開催——Washington D.  
C., 1982年9月20日～24日

本年はコンピュータ・ネットワークをテーマに開かれた。恒例のE. R. Piore賞がBell研究所のD. M. Ritchie氏とK. L. Thompson氏に与えられた。受賞理由は、UNIXやプログラミング言語Cの開発などの貢献によるものであった。基調講演のセッションでは、V. J. Goldberg氏、D. Hann氏らが「80年代の機会と課題」をテーマに講演を行った。また、今回は「日本の技術」と題して日本の大学・企業などにより光ファイバ技術の応用、大規模ネットワーク、パフォーマンス評価・セキュリティ・完全性管理の三つのセッションが開かれ注目された。

●情報処理学会第25回全国大会（昭和57年後期）  
開催——福岡、1982年10月19日～21日

今回の特別講演は、線形計画法の開発者として知られるG. B. Dantzig氏による「知と情一脳内情報処理機構と計算機」であった。一般講演のソフトウェア工学、人工知能・自然言語処理、アーキテクチャ、言語関係の主なテーマは次のとおり。ソフトウェア工学では、ソフトウェア品質の目標管理モジュラ・プログラミングを志向した適応プロダクション・システム、代数的仕様の検証支援システム、システム設計、疑似自然言語によるソフトウェアの形式的仕様記述、ソフトウェア生産性向上のための作業環境、プログラミング行動における個人差など。言語関係では、関数型言語による統合的ソフトウェア開発手法、Equationにもとづくプログラム生成、自然言語理解のためのPrologの拡張、設計システム記述言語ADL、漢字Prolog、述語論理言語NORM、Adaの言語システムの共通中間言語とその効率的実現法など。人工知能・自然言語処理では、フレームによる知識の形式的記述、知識ベース・システムのプログラム環境、知識ベース・システムのCADへの応用、Prologを使った日本語解析、Prologを用いた翻訳システムにおける意味処理、Montague文法による機械翻訳システムにおける訳文生成など。アーキテクチャでは、Pascalマシン、Prologマシン、抽象データ型支援アーキテクチャ、分散型データ・フロー計算機EDAC、高水準並列計算機などであった。

## 〈UNIVAC 関係の論文・講演等の要約〉

●インタラクティブ・コンピュータ・システムにおける人間工学およびユーザ支援技術——会話型システムの最大の関心事は使用の容易性である。この使用の容易性に関連し、システムの使用法の学習の容易性、使用能力の保持のしやすさ、誤りの防止、誤りからの復帰の容易性などの人間工学的要件が求められている。このため、研究室での実験、現場での調査、観察記録の分析などがシステムの設計や評価に採り入れられた。

まず、研究室における実験の最大の課題は、被験者の能力や実験内容に至るまでの各要因を制御しなければならないことである。

現場での調査は実験を制御しにくい欠点があるが、低コストで実施でき大量のデータが集められる利点がある。

また、観察記録（プロトコル）の分析は、研究室でも現場でも実施されているもので、初期システムの設計に有用である。

一方、ユーザを支援するオンライン・アシスタントとしては、オンラインでアクセス可能なマニュアル、パーザへの援助機能の組み込み、人工知能技術による高品位の援助など多岐にわたる技術が必要となる。

このほか、援助データベースへのユーザの要求処理を集中化した援助プロセッサを提案している。

また、人間と機械とのインタラクションを考える場合には、①コンピュータを道具とみなしてユーザ側に制御がある場合、②コンピュータを知的なアシスタントとみなし、コンピュータは主体的に行動してその結果と必要な情報を人間に知らせる場合、との二つの見解がある。そして、後者からコンピュータに援助の主体性をもたせるという考えが生まれており、R. R. Burton と J. S. Brown のコーチング・システムはその代表例である。(N. K. Sondheimer, "Human Factors and User Assistance in Interactive Computing Systems," *IEEE Transactions on Systems, Man and Cybernetics*, March/April 1982)

●コマンド言語の設計における人間工学的配慮の必要性——コマンド言語に対する人間工学的配慮は、設計の最初の段階から必要であり、人間とソフトウェアとのすべてのインタフェースを考慮しなければならない。

このため、インタフェースの設計で考慮すべき点

として、システムに対するユーザの見方、ユーザに対する知覚・記憶・思考に関する認知的側面、ユーザが使用できるツールなどがあげられる。また、ソフトウェア・システムに対する経験の蓄積とともに、これらが変化していくことも考えなくてはならない。

さて、以上の諸点を踏まえたうえで、コマンド言語の設計プロセスを考えると、そのプロセスは、①ユーザによるソフトウェア・システムの概念的モデルの決定、②技術レベルによるユーザのグループ分け、③コマンド言語の構文の決定、④コマンド言語のテストと評価、の各ステップに分けられる。

ここで、最初の二つのステップはシステムの概念的モデルとユーザ・モデルの定義で、これらはコマンド言語のインタフェースを決定する際のフレームワークを与える。つまり、概念的モデルはシステムの機能を、ユーザ・モデルはコマンドの構文の大綱を決める基礎となる。たとえば、ユーザ・モデルはコマンド言語の学習能力の発達を、①オウム（機械的にまねる人）、②初級（新米）、③中級（中堅）、④上級（エキスパート）、⑤名人（マスタ）、の五つに段階づけるものである。

そして、これらの設定が終わると、まず最初に主な（多数を占める）技術水準のユーザを対象としたインタフェースが構築され、つぎに他の水準のユーザへと対象を拡大していくことになる。

さて、会話型システムを対象とする人間工学の現状は、コマンド文の研究はすでに十分行われており、構文・コマンド名・省略型作成などに関し、ガイドラインが得られている。しかし、メニューや対話を対象とした研究は現在発展中であり、まだガイドラインが設定できるまでに至っていない。(M. L. Schneider, "Ergonomic Considerations in the Design of Command Languages," *International Ergonomics Association Congress*, Aug. 1982, pp. 23—27)

●ソフトウェアのエラー・データの分類法——ソフトウェアの開発法や検査法を評価する際の障害として、実用規模のプロジェクトにおけるエラー・データおよびその統計情報の不足があげられる。本研究は、高水準言語で1万行の会話型の特殊エディタを対象として、9か月に及ぶコーディング、ユニット・テスト、機能テスト、システム・テストの各段階におけるソフトウェア・エラーを分析し、開発プロセスとエラーとの関係を明らかにする新しいエラー分

類法を提唱するものである。

この分類法はエラーをその属性間の関係で記述するもので、関係データベースという属性との類似性から属性分類スキーム (attribute categorization scheme) と命名された。この分類法におけるエラー分類木の第1レベルは、大分類 (major category) と呼ばれ、どのタイプのコードが修正されたかを示すもので、①データ定義 (定数・記憶域・制御コードを定義するコード)、②データ処理 (変数の値を修正あるいは初期化するコード)、③テスト (条件を評価し、その結果によって分岐するコード)、④テスト・プラス・プロセス (条件を評価し、その結果によって特定の計算を実行するコード)、⑤ドキュメンテーション (プロダクトの記述)、⑥システム (オペレーティング・システム、コンパイラ、ハードウェアなどのプログラム環境におけるエラーの記述) などに大別される。

また、これらの大分類を補正するものとして、タイプが設定されている。タイプは、①アドレス (アレ・インデックス、リスト・ポインタ、記憶域へのオフセットなど)、②コントロール (ブランチ・テーブルへのエントリや出力フォーマットの指定などの制御の流れを決定する情報)、③データ (処理・入力・出力される主要情報)、④ループ (ループをチェックするテスト)、⑤分岐 (IF あるいは CASE 文をチェックするテスト) に分けられる。

そして、これらの分類体系によって、174個のエラーを分析し、エラーの原因、エラーが作られた時点および発見された時点、発見したときの状況、これらに要した時間、使用した技術、成功した技術などの事実を収集する。(T. J. Ostrand, "Collecting and Categorizing Software Error Data in An Industrial Environment.")

●人間工学と未来社会についての展望——電子技術の発展は、職場や家庭に対してすでに影響を及ぼし始めているが、この影響とデータ処理から通信への展開によって電子住宅 (electronic cottage) が実現した場合に引き起こされる変化と比較すると、ほんのわずかなものである。これらの技術が一般の人々の生活に与える影響が増大するにつれて、心理的要因および人間工学が最も重要な課題となった。そして、人間とコンピュータとの相互作用の増大によって、ハードウェア機器に対するよりよい人間工学的設計と認知の人間工学の必要性が求められている。

家庭での就労が実現するためのハードウェアは、高解像度ディスプレイ、キーボード、ハードコピー・プリンタなどで構成されるインテリジェント端末機

である。さらに、それらは今後短小軽薄化し、多重化された高品質なものとなろう。

また、ソフトウェアの面では、コンピュータの知識をもたないエンド・ユーザが機器を利用し、かつ端末機側にはエンド・ユーザを教える人もいないため、広汎な支援システム (help system) が不可欠となる。同システムからのメッセージは、語数の少なさよりも単純な文法で、かつ肯定形の表現にすべきである。そして、エラー・メッセージは、まず熟達者用の簡潔なものを出力し、初心者用には補助情報 (問題、原因、誤りの訂正あるいは防止法) が得られるようにすべきである。

個々のシステムの機能としては、通信と職業用ツールが求められる。そして、専用のデータベースや LISP のような記号処理言語も必要となろう。いずれにせよ、ワード処理、電卓機能、通信ネットワークのアクセスの機能は必須である。

将来のコンピュータは、人々のコミュニケーションの中核的存在として公式および非公式の情報交換を統合して行うものとなるだろう。そして、将来の電子住宅では電子秘書 (electronic secretary) が発展するであろう。また、コンピュータの媒介によるコミュニケーションの短所は、プライバシーの保護のむずかしさにあり、一方長所として、情報の共有によって新入社員の新しい職場への参加などは容易になる。

電子住宅は社会構造に大きな変革をもたらし、仕事の時間帯と個人生活は現在ほど切り離されたものではなくなる。また、住居の選択に通勤時間は問題とならない。このほか、通勤不能なために就労の機会がなかった身体不自由者や老人にも雇用の機会が与えられる

なお、電子住宅の実現は個人の自主性を高めるため、全体として生産は12~20パーセント向上するとみられている。また、電子住宅では、遠隔地にいる部下の働いている状況がマネージャからみえないため、現在と異なる新しい管理上の問題が発生しよう。(F. B. Arble, "Ergonomics and Future Technology;" NTC)

●アプリケーション・プログラムのコンバージョン技術——実際にいくつかのプログラム・コンバージョン・プロジェクトの作業実績を分析し、現実的な計画が立てられていないこと、コンバージョン・プロジェクトに明文化された規格や規程がないこと、コンバージョン・プロジェクトにおける役割分担が不明確なことなどの問題が発見された。

ユニバックでは、これらの問題を解決するため

に、コンバージョン・プロジェクトの各フェーズを構造化し、現実的で情報の取りやすい計画・管理システムを開発している。ユニバックでのアプローチは、コンバージョンの二つの重要な側面、計画とトランジションに主眼を置いている。

まず、計画は二つの主なアクティビティで構成され、第1のアクティビティは、コンバージョンのコンポーネント、つまりプログラム、サブプログラム、ライブラリ・プロシージャ、ジョブ、データ・ファイルを分析し、識別し、これらの倉庫（インベントリ）を作成し、相互参照リストなどを出力する。第2のアクティビティは、相互に関係をもつプログラムとジョブをとりまとめて、コンバージョンにおける基本単位である CIP (conversion input package) を作成する。第3のアクティビティは、コンバージョン・スケジュールの作成である。そして、コンピュータによって作られたコンバージョン計画を用いて、プロジェクト・マネージャはプロジェクトの運営を行う。

さて、プログラム、ジョブ、データの実際のコンバージョンは、三つの構造化されたフェーズに分けて実施される。第1のフェーズは、準備であり前述の CIP が生成される。第2は、パッケージ・コンバージョンで CIP 中のプログラム、データ・ファイル、ジョブが新しい環境の言語で書き換えられ、変換後のプログラムが旧環境での機能と同等に働くかどうかテストされる。そして、第3には、実施 (implementation) であり、実施フェーズでは COP (conversion output package) が受け入れられるものかどうかを確認したり、生データで実行結果をチェックしたり、変換後のシステムのカット・オーバーを行う。いずれにせよ、このアプローチのメリットは、コンバージョンをこのようにパッケージングすることによって、責任とその出力を明確にすることにある。

このほか、コンバージョンにおける構造的アプローチを支援するツールとして、いくつかの専用コンバージョン・ソフトウェアによって構成される IMPACT (Integrated Management, Planning, Analysis and Conversion Tool) と呼ばれるシステムが開発されている。これらの諸ツールの存在によってコンバージョンが管理可能となる。(I. M. Romanov, "Computer Assisted Conversion Technology—Issues and Innovations," Federal ADP Procurement Conference, Washington D. C.)

#### ▶テクニカル・コーディネータ

伊東 玄 (プロダクト・サポート統括一部・アドバンスド・ハードウェア・プロダクト室)、中村 脩 (事業企画部 商品企画1室長)、長島 毅 (応用ソフトウェア部 技術計算グループ)、野本 雄一 (画像処理システム部 グラフィック・システム企画室長)、原 潔 (プロダクト・サポート統括二部 1100/90 導入プロジェクト)、室 信治 (プロダクト・サポート統括二部 開発1部 EXEC 1 グループ・マネージャ)、森 俊治 (応用ソフトウェア部 経営科学/情報検索室情報検索グループ・マネージャ)、高山龍雄 (応用ソフトウェア部 CAD 開発部長)、渡部義維 (応用ソフトウェア部 技術計算グループ・マネージャ)

#### ▶エディトリアル・スタッフ

広野和夫 (広報部 テクニカル・パブリケーション室長)、山田真市 (主任研究員)、桑野龍夫、高橋 肇、青柳幸久、丹野敬子

#### ●Technical Coordinators

K. Hara, K. Itou, T. Mori, N. Muro,  
T. Nagashima, O. Nakamura, Y. Nomoto,  
T. Takayama, Y. Watanabe

#### ●Editorial Staff (Technical Publications)

K. Hirono, S. Yamada, T. Kuwano,  
H. Takahashi, Y. Aoyagi, K. Tanno

---

### 技 報

#### UNIVAC TECHNOLOGY REVIEW

No. 4

---

発行日	昭和 58 年 2 月 28 日
発行人兼編集人	富田 和夫
発行所	日本ユニバック株式会社 東京都港区赤坂 2-17-51 〒107 TEL (03) 585-4111 (大代表)
頒布価格	1,500 円
印刷所	三美印刷株式会社

禁無断複製転載

○通産省・電子協・メーカー6社の全面協力による新時代の必備書

# コンピュータの事典

編集委員長 元岡 達

編集委員

岡松 壯三郎 鳴村 和也  
川口 融 神宮 寺順  
若曾 根和之 二宮 昭一  
石井 治 藤木 勝美  
天羽 浩平 吉岡 忠  
石井 善昭

今後ますます変革が期待されているコンピュータ技術の全体を把握できるよう、通商産業省・日本電子工業振興協会・メーカー6社の全面協力により総纂されたもので、情報処理に従事する方々や、コンピュータ関連部門担当者にとっての必携書。

〔特徴〕○コンピュータに関する必要事項の基礎から応用までを体系的に説明。○歴史や諸外国の技術動向、今後のコンピュータ政策についても解説。○システム評価などは単なる理論ではなく、実際に役立つよう説明。○ハード面と同時に情報処理としてのソフト面も充実。○コンピュータに関する資料やメーカー説明書ではわからない点を解決できるよう配慮。○基礎的な記述も重視したので入門書としても役立つ。

A5判700頁 予価12000円  
【4月刊予定】

○基礎理論から設計上の問題点や応用技術を集大成

## ミニコン・マイコン応用ハンドブック

藤井克彦編集

ミニコン・マイコンの基礎をなすプロセッサ、メモリチップ、周辺機器等を説明し、あらゆる分野に広がる具体的な応用例(ロボット、家電製品、自動車、カメラ、複写機、テレビ、プラント、鉄道、医用、倉庫、視覚認識、音成合成、他)を解説。

A 5 判920頁 定価18000円

## マイクロコンピュータハンドブック

森 亮一編集

A 5 判 800頁 定価18000円

マイクロコンピュータ全般についてを集大成。応用技術や関連分野については実例をあげて具体的に記述。

## 集積回路応用ハンドブック

菅野卓雄編集

A 5 判 870頁 定価25000円

ICの作り方、基本構成素子、基本回路、応用機器、将来の展望等、使用方法を中心に基礎・応用を集大成。

## 光通信ハンドブック

柳井久義編集

A 5 判 700頁 定価16000円

急速に実用化されてきた光通信の基礎理論から応用技術までを網羅。光通信技術を初めて体系化し集大成。

## センサ工学

森村正直・山崎弘郎編

A 5 判 488頁 定価9000円

多種多様なセンサ技術を横断的視点で体系化し、応用の技術や将来の展望についても具体的に解説した。

## 現代人の統計 全7巻

林 知己夫 編集

1. 統計解析法の原理 鈴木義一郎著 定価2300円
2. 多変量解析法 柳井・高根著 定価2300円
3. 社会調査の基礎 杉山明子著 近刊
4. 意思決定の基礎 松原 望著 定価2300円
5. 実験計画法の基礎 早川 毅著 定価2300円
6. データ解析と管理技法 大隅 昇著 定価2500円
7. 多元的データ分析の基礎 駒沢 勉著 定価2300円

## 朝倉書店

〒162 東京都新宿区新小川町6番29号  
電話(03)260-0141(代)・振替口座 東京6-8673  
(本誌名ご記入の上お申込次第総合図書目録呈)