

TECHNOLOGY

REVIEW

UNISYS

技 報

通巻

48

1996年2月発刊

Vol. 15 No. 4

論文

実行可能な形式仕様言語 CafeOBJ	谷津弘一	1
通信プログラム仕様からプロダクション 規則仕様への逆変換手法	深山 尚, 田倉 昭, 太田 理	17
ソフトウェアレビューの質を参加者への テストにより評価する方法	的場政明, 高田義広, 鳥居宏次	29
Ferguson 曲線の接線長決定の一手法	清水保弘, 宮地恵美	40
“統合化”と企業組織 ——異質性のダイナミズム	ト部正夫	73
OSI-TP ソフトウェアの3機種 (2200, U 6000, A シリーズ) への実装	山野茂美, 末永敏昭 河野典久, 石崎達也	89
複数拠点下の XTPA システム	加藤武彦	105
信金バックアップセンタの紹介	関 俊夫	117
食品物流センタにおける IND・IE の適用	平野静夫	133
コンピュータの電源高調波問題と対策	蛭本邦夫	154
8003 の高速・高品位レンダリング機能	渡辺 啓	172

新製品紹介	194
掲載論文梗概	表 2, 3

CafeOBJ は、代表的な代数仕様記述である OBJ に、José Meseguer が Maude 言語で提案した書き換え規則を導入し、拡張した言語である。CafeOBJ では、この書き換え規則を用いて、オブジェクト指向パラダイムの意味でのオブジェクトやその動的な振る舞いを陽に記述することができる。谷津弘一の**実行可能な形式仕様言語 CafeOBJ** は、筆者等が情報処理振興事業協会 (IPA) にて現在開発中の、**実行可能な形式仕様言語 CafeOBJ** の概要を紹介している。

サービスを分かりやすく記述でき、非専門家にも理解できる**通信プロダクション規則仕様**から、**通信プログラム仕様**へ変換することができる。しかし、**通信プログラム仕様**はネットワーク内の詳細な知識を必要とし、既存の**プログラム仕様**から**プロダクション規則仕様**へ逆変換できれば既存の**プログラム仕様**の再利用性が高まる。深山尚・田倉昭・太田理は、**通信プログラム仕様からプロダクション規則仕様への逆変換手法**の中で、**通信プログラム仕様から宣言型のプロダクション規則仕様の抽出法**について提案している。

開発中のソフトウェアに含まれるバグは、発見が遅れるほど、その修正に要するコストが高くなる傾向にある。そこで、一般のソフトウェア開発の各工程においては、その工程の成果物に含まれるバグを早期に発見するためにレビューが実施される。的場政明・高田義広・鳥居宏次は、**ソフトウェアレビューの質を参加者へのテストにより評価する方法**の中で、コード作成工程において実施される**コードレビュー**について、**コードレビューの直後、途中でレビュー者に簡単な筆記試験をばたすことにより、コードレビューの質を定量化する方法**を提案し、提案する方法を適用した実験について報告している。

CAD の幾何モデラの自由曲線を生成するには、いくつかの方法があるが、清水保弘・宮地恵美の**Ferguson 曲線の接線長決定の一手法**では、通過点列と各通過点での曲線の接線方向を指定して、区分的 3 次多項式曲線を生成する方法を取り上げ

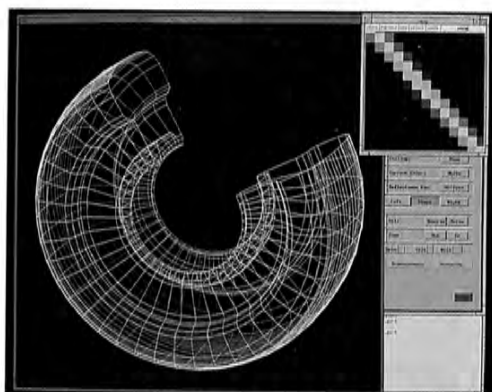
ている。隣接通過点間を補間する 3 次多項式曲線は Ferguson 曲線と呼ばれる。同曲線の接線長として、従来から経験的に用いられてきた方法 (接線長をセグメント始終点の弦長としたり、接円弧長とする方法) は理論的根拠が明らかでなかった。本稿では、線分・円弧を基本形状とし、これらのアフィン変換や 1 次ブレンドとして接線長を決定していくアプローチを提示し、とくに円弧形状セグメントの場合には等面積法と呼ぶ新手法による接線長決定法を提案している。

ト部正夫の**“統合化”と企業組織——異質性のダイナミズム**は、統合化と企業組織の成長について論じている。統合化の定義としては、「異質の要素の組み合わせによって、新奇性を織り込んだ、より高次の統一体ないし全体を構築する行動過程」を援用し、企業組織の成長過程を 6 段階の進化ステージとして設定した上で、各ステージの特徴およびステージ間の移動について論じている。

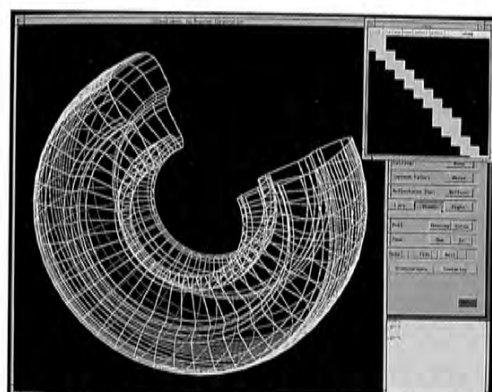
OSI-TP は、X/Open の DTP モデルに準拠した分散トランザクション処理システムであり、分散トランザクション処理アプリケーションが OSI あるいは TCP/IP ネットワーク内の他のアプリケーションと通信することを可能にしている。山野茂美・末永敏昭・河野典久・石崎達也は、**OSI-TP ソフトウェアの 3 機種 (2200・U6000・A シリーズ) への実装**の中で、2200 に実装したソフトウェアを U 6000 および A シリーズへ実装した時の開発経緯・実装方法・効果を概説している。

XTPA 技術を用いたノーダウンシステムは、従来の事例では単独の拠点運用形態であった。複数拠点運用下の XTPA システムでは単独拠点運用の XTPA 技術では対応できない問題が発生する。加藤武彦の**複数拠点下の XTPA システム**は、オンライン分野の XTPA 適用技術、とくに分散オンラインにおける**複数拠点コンピュータ運用**が必要となる機能について述べている。

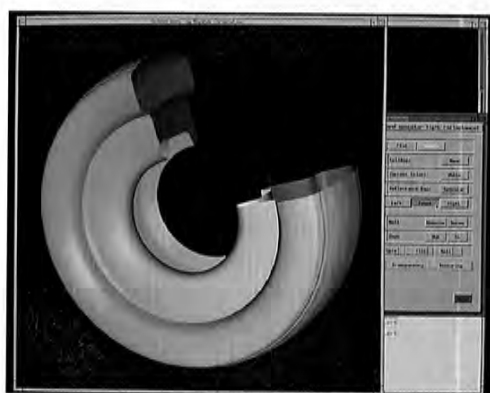
金融機関のコンピュータの利用範囲は対外接続の広がりと共に拡大し、社会的責任の増大に伴う



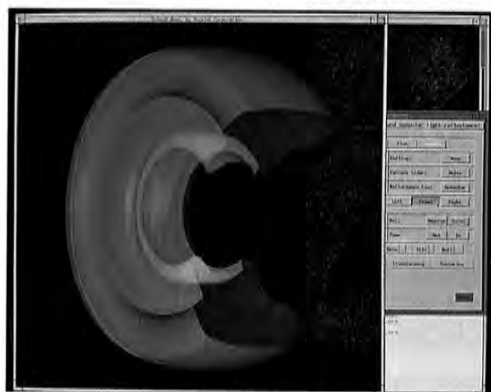
□絵1 タイヤのワイヤ・フレーム
(アンチ・エアシング ON)



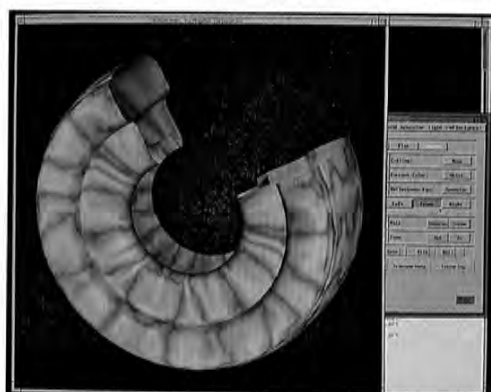
□絵2 タイヤのワイヤ・フレーム
(アンチ・エアシング OFF)



□絵3 タイヤ (グーロ・シェーディング)



□絵4 タイヤ (トランスペアレンシ)

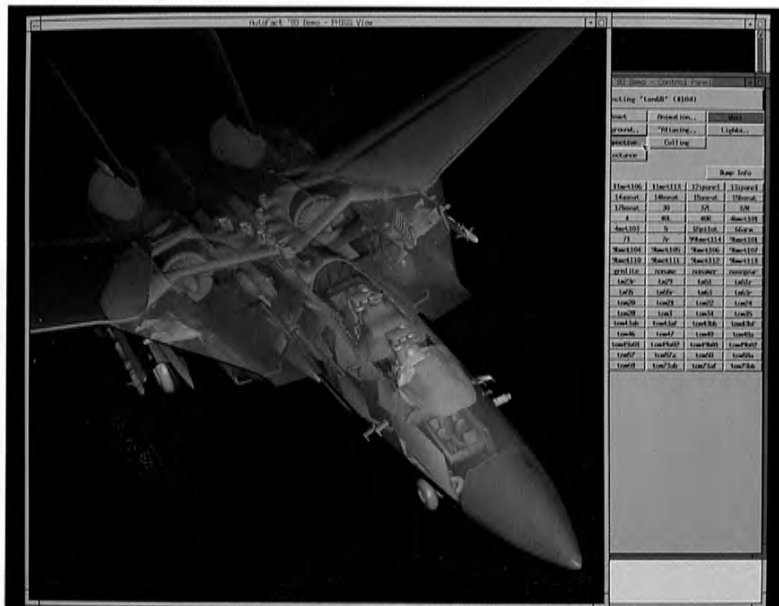


□絵5 タイヤ (テクスチャマップ)

詳細は、本文「8003の高速・高品位レンダリング機能」に掲載。



口絵6 車 (トランスペアレンシ)



口絵7 飛行機 (トランスペアレンシ)

詳細は、本文「8003の高速・高品位レンダリング機能」に掲載。

実行可能な形式仕様言語 CafeOBJ

An Executable Specification Language CafeOBJ

谷津 弘 一

要約 本稿では、筆者らが情報処理振興事業協会 (IPA) にて現在開発中の、実行可能な形式仕様言語 CafeOBJ の概要を紹介する。CafeOBJ は、代表的な代数仕様記述である OBJ に、José Meseguer が Maude 言語で提案した書き換え規則を導入し、拡張した言語である。CafeOBJ では、この書き換え規則を用いて、オブジェクト指向パラダイムの意味でのオブジェクトやその動的な振る舞いを陽に記述することができる。また、CafeOBJ は高階のパラメータ化機構を有する。これは、パラメータの制約となるモジュールのパラメータ化を許す機構であり、これにより、Standard ML で実現されている sharing constraint と同等な記述が可能になる。

Abstract In this paper, a primary view of an executable specification language CafeOBJ, whose development is in progress at Information-Technology Promotion Agency, Japan (IPA), is described. CafeOBJ is an extension of OBJ, one of the most marked algebraic specification languages, with rewriting rules introduced in Maude language of José Meseguer. In CafeOBJ, objects and their dynamic behavior in the sense of object-oriented paradigm can be represented explicitly with rewriting rules. Besides, CafeOBJ provides higher-order parameterization mechanism, which allows parameterization of restrictions of parameters and makes it possible to realize sharing constraint supported by Standard ML.

1. はじめに

Edsger Dijkstra の go-to-less プログラミング¹⁾から始まる構造化プログラミングの概念は、データの集合 (この集合を台集合と呼ぶ) とその上の演算を一体化したデータ型を経て、抽象データ型へと昇華した。抽象データ型は、データ型からデータの表現形式や演算の実現手段を捨象したもので、演算の振る舞いは演算間の関係によって定められる。オブジェクト指向パラダイムにおけるオブジェクトの概念も、抽象データ型の延長線上にあるものと言える。

代数仕様記述法は、1970年代の半ばに、抽象データ型を厳密に表現する方法として提案された。代数仕様記述法では、抽象データ型は、台集合、台集合上の演算、演算から生成される項の間の等式の三つで表現される。代数仕様記述法に基づき記述された仕様を代数仕様と呼び、代数仕様記述法に基づく言語を代数仕様言語と呼ぶ。

ソフトウェアシステムはよく状態機械で表現されるが、状態機械は抽象データ型で自然に捉えられる。抽象データ型はソフトウェアシステムの基本的な構造のモデルと考えられており、抽象データ型を厳密に表現する代数仕様は、ソフトウェアシステムを的確に表現しうる仕様の一つとして広く認められている。

代数仕様には、等式のみを述語に持つ等式論理を採用していることに起因する、大きな利点がある。まず、一階述語論理に比べて検証が容易であること、そし

て、等式の左辺を右辺に書き換える項書換え系という実行系を用いて、項の値の計算ができる（このことを仕様が実行可能であると呼ぶ）ことである。この実行可能性を生かして、代数仕様をラビッド・プロトタイピングに使うこともできる。

本稿では、筆者らが情報処理振興事業協会（IPA）にて開発中の実行可能な形式仕様言語 CafeOBJ^[10]の概要を紹介する。CafeOBJは、代表的な代数仕様言語である OBJ^[2,4]に一方方向の書き換えを表す書換え規則を導入し、拡張した言語である。この書換え規則を用いて、CafeOBJではオブジェクト指向パラダイムにおけるオブジェクトとその動的な振る舞いを陽に記述することができる。CafeOBJは、代数仕様言語の長所を損なうことなく、オブジェクト指向を無理なく取り入れた言語であると言える。

CafeOBJの大きな特徴の一つに、高階のパラメータ化機構^[12]がある。ここでいう高階とは、パラメータ付きモジュールにおいてパラメータの制約となるモジュールのパラメータ化を許すことを意味する。この機構は、多重継承を持つソフトウェアシステムの的確なパラメータ化を可能にするものであり、Standard ML^[9]における sharing constraintと本質的に同等のものである。

本稿の構成は以下の通りである。まず2章で代数仕様について簡単に説明した後、3章と4章でCafeOBJの基本的な構文と構造化機構を紹介する。そして、5章でCafeOBJによるオブジェクト指向モデリングについて述べ、6章でCafeOBJの高階パラメータ化機構について説明する。

2. 代数仕様

CafeOBJは厳密には代数仕様言語の範疇には入らないが、代数仕様言語 OBJ をサブセットとして含むので、代数仕様言語として扱われることが多い。CafeOBJについて述べる前に、基礎的な知識として、代数仕様と代数仕様を表す「もの」について説明しよう。

データ型は、データの集合（台集合）とその上の演算を一体化したものであり、数学的には代数として捉えられる。データ型を代数として表現しよう、というのが代数仕様の基本的な考えである。

代数仕様は、次の三つにより構成される。

- ・いくつかの台集合 (support, carrier)
- ・いくつかの台集合上の演算
- ・演算から生成される項の間に成り立つ、いくつかの等式

例として、スタックの代数仕様をお目につけよう。

仕様 STACK is

```

台集合  Elt, Stack
演算    empty : → Stack
         none  : → Elt
         push : Elt × Stack → Stack
         pop  : Stack → Stack
         top  : Stack → Elt

```

等式 $\forall s : \text{Stack}, e : \text{Elt}$

$$\begin{aligned} \text{pop}(\text{push}(e,s)) &= s \wedge \\ \text{top}(\text{push}(e,s)) &= e \wedge \\ \text{pop}(\text{empty}) &= \text{empty} \wedge \\ \text{top}(\text{empty}) &= \text{none} \end{aligned}$$

仕様 *STACK* は二つの台集合 *Elt* と *Stack* を持つ。 *Elt* はスタックの要素の台集合であり、 *Stack* はスタックの台集合である。 さらに、 *STACK* は演算 *empty*, *none*, *push*, *pop*, そして *top* を持つ。 定義域 (domain) が無い演算は定数を表す。 上の例では、 *empty* と *none* が定数である。 *push* は、 *Elt* と *Stack* の要素をもらって *Stack* の要素を返す演算であり、 *pop* と *top* は、 *Stack* の要素をもらって、 *Stack* と *Elt* の要素をそれぞれ返す演算である。 そして、 任意の *Elt* の要素 *e* と *Stack* の要素 *s* に対して、 上に示した四つの等式が成り立つ。

台集合と演算を併せて、 指標 (signature) と呼ぶ。 代数仕様は、 仕様に記述された等式を満たす指標を含む*データ型の集まりを表す。

また、 代数仕様は、 次の二つの条件

- 1) 台集合の要素 (項) は、 宣言された演算だけから生成される。
- 2) 二つの項の値はその間の等式が定義された等式群から導出されるときに限り等しく、 かつその場合には必ず二つの項は同じ要素になる。

を満たすデータ型を、 名前替えによる指標の違いを除いて一意に定めるという性質を持つ²⁾。 このようなデータ型を抽象データ型と呼ぶ。

以上からわかるように、 代数仕様が表す「もの」には、

- 1) 仕様を満たすデータ型の集まり
- 2) 仕様から一意に定まる抽象データ型

の二つがある**。 多くの代数仕様言語ではこの二つを区別して記述する。 例えば、 代表的な代数仕様言語である OBJ には、 データ型の集まりを表すセオリ (theory) と抽象データ型を表すオブジェクト (object) という、 2種類の記述単位がある。

仕様に記述された項の間の等式を、 左辺から右辺への項の簡約規則と見做すことにより、 次のように項の値の計算ができる。

$$\begin{aligned} &\text{top}(\text{pop}(\text{push}(e2,\text{push}(e1,\text{empty})))) \\ \rightarrow &\text{top}(\text{push}(e1,\text{empty})) \\ \rightarrow &e1 \end{aligned}$$

どの簡約規則を用いてもそれ以上簡約できない項を正規形 (normal form) と呼ぶ。 代数仕様における項の値とは、 項の正規形のことであり、 代数仕様の実行とは仕様に記述された等式を簡約規則と見做して項の正規形を求めることである。 記述された等式に基づき仕様の実行を行う系を、 項書換え系 (rewriting system) と呼ぶ。

代数仕様に基づく論理は、 等式論理 (equational logic) という、 等式のみを述語に持つ非常によい性質を持つ論理である。 先に述べた仕様の実行は、 等式論理に基づく推論と見做すこともできる。 よい性質を持つ論理に基づくことと項書換え系という推

* 指標の名前を付け替えて含んでも構わない。

** 圏論的に言えば、 データ型の集まりは一つの圏 (category) を成し、 抽象データ型はその圏の始対象 (initial object) である。

論実行系を持つことという二重の理由から、一階述語論理に基づく仕様に比べて代数仕様は仕様が満たすべき性質の検証が容易である、と言える。

3. CafeOBJの基本構文

前章で、OBJにはデータ型の集まりを表す記述単位（セオリ）と抽象データ型を表す記述単位（オブジェクト）がある、と述べたが、CafeOBJには、モジュール（module）と呼ばれる、その二つを併せた記述単位がある。本章では、CafeOBJのモジュールを記述するのに必要な構文要素について説明する。

3.1 モジュール

本節では、下のリストを表すモジュールの説明を兼ね、モジュールの記述に必要な、書換え規則以外のCafeOBJの構文要素を紹介する。

```
module LIST is
  sorts List Elt .
  subsorts Elt < List .
  op nil : -> List .
  op _,_ : List List -> List .
  vars L L' L" : List .
  eq (L,L'),L" = L, (L',L") .
  eq L,nil = L .
  eq nil,L = L .
endm
```

3.1.1 モジュールの宣言

モジュールの記述は、予約語moduleで始まり、予約語endmで終わる。

```
module LIST is
  ...
endm
```

moduleとisの間にモジュール名を書き、isとendmの間に、台集合、演算、等式、そして書換え規則を宣言する。

3.1.2 ソートの宣言

代数仕様では、台集合のことをソート（sort）と呼ぶ。予約語sortから始まる文で、仕様で用いるソートを宣言する。

```
sorts List Elt .
```

（上の例ではsortsと複数形になっているが、これは一つの文で複数のソートを宣言するための構文糖衣である。）CafeOBJでは、subsortから始まる文でソート間の包含関係を宣言することができる。

```
subsorts Elt < List .
```

この文は、ソートEltの要素がソートListの要素に含まれることを示している。この文のおかげで、List上で定義された演算をEltの要素に対して適用することができる。

3.1.3 演算の宣言

予約語 `op` に続くのは、演算の宣言である。

```
op nil : -> List .
op _,_ : List List -> List .
```

下線部 () は引数の位置を表す。コロン (:) の右に演算の「型」を記述する。-> の左には引数のソートを、右には値のソートを記述する。引数のソートの指定のないものは、定数を表す。nil はソート List の定数であり、_,_ は二つの List の要素を引数とし List の要素を値とする演算である。

3.1.4 等式の宣言

予約語 `eq` から始まる文で、項の間の等式を宣言する。

```
eq (L,L'),L" = L,(L',L") .
eq L,nil = L .
eq nil,L = L .
```

上の例の三つの等式は、演算 `_,_` が結合則を満たし、かつ単位元が nil であることを示している。等式に現われる変数は、あらかじめ予約語 `var` で始まる文で宣言されていないなければならない。

```
vars L L' L" : List .
```

モジュール LIST の等式群は次の文で置き換えられる。

```
attr _,_ : [ assoc id: nil ] .
```

`attr` から始まる文で、二項演算子の、結合的 (associative)、可換 (commutative)、冪等 (idempotent) といった性質*を持つことや、単位元 (identity) の指定ができる。上の文は、演算 `_,_` が結合的で、nil を単位元を持つことを宣言している。

CafeOBJ では、条件付きの等式を、予約語 `cq` で始まる文で記述することができる。

```
cq L = R if P .
```

上の文は、条件 P が真である時項 L と項 R は等しいことを示す。

3.2 書換え規則

CafeOBJ では、等式は正規形を求めるための簡約規則であるのに対し、書換え規則は状態の変化を表現するためのものであり、正規形を求めるための簡約規則ではない。書換え規則は、予約語 `r1` から始まる文で宣言される。

```
var E : Elt .
r1 loss : E,L => L .
```

コロン (:) の左にあるのは、この書換え規則に付けられたラベルである (ラベルは付けなくてもよい)。この loss とラベル付けされた規則は、空でないリスト E,L が L に変化するさまを表している。

等式と同様に、条件付きの書換え規則を書くことができる。

```
cr1 Lbl : L => R if P .
```

上の予約語 `cr1` から始まる文は、ラベル Lbl を持つ、条件 P が真であるときに左辺 L を右辺 R に書き換える規則を表している。

* あるソート s 上の二項演算子を `_∇_` : s s -> s とすると、`_∇_` が結合的、可換、冪等とは、任意の s の要素 x, y, z に対して、 $(x \heartsuit y) \heartsuit z = x \heartsuit (y \heartsuit z)$, $x \heartsuit y = y \heartsuit x$, $x \heartsuit x = x$ がそれぞれ成り立つことである。

代数仕様が仕様を満たすデータ型の集まりを表すのに対し、書換え規則のある CafeOBJ モジュールは「データ型もどき」の集まりを表す。ここでいうデータ型もどきとは、書換え規則を除いた部分を満たすデータ型に、書換え規則が表す値の動的な変化を加えたものである*。

しかし、書換え規則を加えても、データ型もどきの中から一意に抽象データ型もどきが定まることが知られている^[7]。本稿では、必要のない限りデータ型とデータ型もどきを区別しないことにする。

4. CafeOBJ の構造化機構

仕様を構造化して記述できる機構は、作ろうとするソフトウェアシステムの本質を理解する上でも、記述した仕様の再利用を図る意味においても、重要な仕組みである。とくに、大規模あるいは複雑なソフトウェアシステムの仕様の記述には必要不可欠なものであると言えよう。

CafeOBJ は、大規模あるいは複雑なソフトウェアシステムの仕様記述に必要な構造化のための機構を備えている。本章では、CafeOBJ の構造化のための機構である、モジュール輸入とパラメータ化機構について説明する。

4.1 モジュールの輸入

モジュールの輸入は、既存のモジュールを再利用するための最も簡単な方法である。前章で紹介したリストを参考にして、自然数のリストを表すモジュールをモジュールの輸入を用いて記述してみよう。

自然数を表すモジュールとして、次のような簡単なものを考える。

```
module NAT is
  sort Nat .
  op 0 : -> Nat .
  op s_ : Nat -> Nat .
endm
```

モジュール NAT では、定数 0 と後者関数 (successor function) のみから自然数が表される。例えば、0, 1, 2, ... は、それぞれ 0, s 0, s s 0, ... で表される。このモジュール NAT を輸入して、自然数のリストを表すモジュールを記述する。

```
module NATLIST is
  using NAT .
  sort List .
  subsort Nat < List .
  op nil : -> List .
  op _,_ : List List -> List .
  attr _,_ : [ assoc id: nil ] .
endm
```

上のモジュール NATLIST において、予約語 using から始まる文で、モジュール NAT

* 圏論的に言えば、ソートは 0-cell、演算は 1-cell、書換え規則は 2-cell を表す。ここでいうデータ型もどきは、書換え規則を除いた CafeOBJ モジュールが表す集まりの要素であるデータ型を、書換え規則が表す 2-cell により 2-圏の拡張 (2-categorical enrichment) したものである。(データ型はそれ自身圏と見做される。)

が輸入されることが宣言される。NATLISTにはNATで宣言されたソート、演算、等式、書換え規則がすべて含まれる。

CafeOBJには、usingの他に、includingというモジュールの輸入を宣言するための予約語がある*。includingの使い方は、usingのそれと同じである。この二つの予約語の意味的な違いは、usingが抽象データ型としてモジュールの輸入を宣言するのに対し、includingはデータ型の集まりとしてモジュールの輸入を宣言することである。

4.2 パラメータ化機構

仕様の抽象化を支援するパラメータ化機構は、ソフトウェアシステムの仕様記述にとって極めて重要な機構である。ソフトウェアシステムの再利用は、パラメータ化機構を抜きにしては語れない。モジュールの輸入も、パラメータ化機構を用いて説明される。

パラメータ化機構は、パラメータ付きモジュール、ビュー、例化 (instantiation) の三つからなる。この三つを順を追って説明しよう。

4.2.1 パラメータ付きモジュール

パラメータ付きモジュールは、モジュールから非本質的な部分を捨象することによって得られる、汎用的なモジュールである。例えば、前章で自然数のリストを紹介したが、リストの要素にはいろいろなものが考えられる。それらを逐次記述するのは非常に煩わしく、かつリストの本質を見失わせる。最初に示したリストのモジュールから、要素を表すソートを次のように捨象して、リストをパラメータ付きモジュールとして再定義しよう。

```
module TRIV is sort Elt . endm

module LIST[X :: TRIV] is
  sort List .
  subsort Elt < List .
  op nil : -> List .
  op _,_ : List List -> List .
  attr _,_ : [ assoc id: nil ] .
endm
```

ここで、TRIVはソートEltをただ一つ持つだけのモジュールである。TRIVの下にあるのが、パラメータ付きモジュールとして再定義したLISTである。モジュール名の横にある、括弧 ([]) に囲まれたX::TRIVは、LISTのパラメータとその制約**を表す (Xがパラメータであり、TRIVがその制約である)。このLISTに実モジュールを与えることにより、様々な要素のリストを表すモジュールを作ることができる。

4.2.2 ビュー

CafeOBJでは、ビューと呼ばれるものを介して、実モジュールがパラメータ付きモ

* 厳密に言うくと、usingはprotectionとincludingというモジュール輸入を宣言する予約語の総称であるが、この二つの予約語には意味的な差異はないので、本稿ではusingで統一する。

** プログラミング言語における型のようなものと考えてよい。

ジュールに与えられる。パラメータ付きモジュールに与えられるのは、モジュールではなくビューである。ビューから与えられた情報を下に、パラメータが実モジュールに置き換えられる。

ビューは、パラメータの制約と実モジュールとの間の、ソートと演算の対応を与えるものである。

```
view TRIV2NAT from TRIV to NAT is
  sort Elt to Nat .
endv
```

TRIV2NATは、LISTのパラメータの制約TRIVからNATへのビューである。このビューにより、TRIVのソートEltとNATのソートNatが対応付けられている。

4.2.3 例 化

パラメータ付きモジュールを例化して実モジュールを作るには、次のようにビューを代入すればよい。

```
make NATLIST is LIST[TRIV2NAT] endm
```

このようにすることで、4.1節のNATLISTと同じモジュールが作られる。

CafeOBJには、暗黙のビューと呼ばれるものがある。パラメータの制約がTRIVのようにソート一つしか持たない場合には、実モジュールをそのまま代入することができる。

```
make NATLIST is LIST[NAT] endm
```

TRIVのソートEltからNATの最初に宣言されたソート（すなわち、Nat）への対応を表すビューが自動的に作られる。

パラメータ付きモジュールに与える実モジュールがパラメータの制約で定義されているソートと演算を含む場合にも、同じように実モジュールをそのまま代入することができる。この場合には、同じ名前のソートと演算を対応付けるビューが自動的に作られる。

次のように、ビューの中身を直接代入することもできる。

```
make NATLIST is
  LIST[view to NAT is sort Elt to Nat . endv]
endm
```

この場合、ビューの名前と制約を記述する必要はない。

5. オブジェクト指向モデリング

ソフトウェアシステムのモデル化におけるオブジェクト指向パラダイムの重要性については、もはや言を待たない。ソフトウェアシステムの仕様で、オブジェクト指向パラダイムの意味でのオブジェクトやその動的な振る舞いを記述できるメリットは大きい。

代数仕様が基づく等式論理では、更新される状態を持つオブジェクトの振る舞いを表現することはできなかった。José Meseguerは、項の変化を表す一方向の書き換えを述語に持つ、書換え論理¹⁷⁾ (rewriting logic) を提唱している。書換え論理では、等式は双方向の書き換えとして表現される。書換え論理は、等式論理を含む論理であ

り、検証のしやすさや項書換え系という実行系を持つ等式論理の長所をそのまま継承している。また、書換え論理では、等式論理ではできない、メッセージを受けて状態を動的に変化させるオブジェクトの振る舞いを的確に表現できる。CafeOBJの書換え規則は、José Meseguerが自身の書換え論理に基づき作成した言語Maudeを参考にして作られたものである。本章で紹介されるCafeOBJのオブジェクトおよびオブジェクト指向システムの取り扱いも、Maudeのそれを参考にしている。

5.1 オブジェクトとコンフィギュレーション

CafeOBJでは、状態が変化するオブジェクトは次のような項で表現される。

$$\langle \text{Oid} : \text{Class} \mid a_1 = v_1, \dots, a_n = v_n \rangle$$

Oidはオブジェクトの識別子であり、Classはクラスの名前である。各 a_i はオブジェクトの属性の名前であり、 v_i はその属性の値を表す。オブジェクトの識別子を属性の値にしても構わない。

CafeOBJには、モジュールの形で、オブジェクトやオブジェクト指向システムを構成する基礎となるキットが用意されている*。上で述べたように、オブジェクトは識別子、クラス名、そして属性（属性名と値の組）の並びからなる。オブジェクトの属性の並びは、同じ要素の重複を認める多重集合（multi-set）として表現される。

```
module MULTI-SET[X :: TRIV] is
  sort Set .
  subsort Elt < Set .
  op empty : -> Set .
  op _,_ : Set Set -> Set .
  attr _,_ : [ assoc comm id: empty ] .
endm

module ATTRIBUTE is
  using (MULTI-SET * (sort Set to Attributes,
                    op empty to emptyattr))
  [TRIV * (sort Elt to Attribute)] .
  sorts AttributeId Value .
  op (_,_) : AttributeId Value -> Attribute .
endm
```

ここで、MULTI-SET * (sort Set to Attributes, op empty to emptyattr) は、モジュールMULTI-SETのソートや演算の名前をSetからAttributesに、emptyからemptyattrにそれぞれ付け替えたモジュールを表す。オブジェクトを構成する基礎となるモジュールは、上で定義された属性の並びを表すモジュールATTRIBUTEを用いて次のように記述される。

```
module OBJECT is
  using ATTRIBUTE .
```

* ここで紹介するのは、本稿から得られる知識だけで理解できるよう、実際にCafeOBJに組み込まれているモジュールを制限/モディファイしたものである。

```

    sorts ObjectId ClassName Object .
    subsort ObjectId < Value .
    op <_:_|_> : ObjectId ClassName Attributes -> Object .
endm

```

CafeOBJでは、オブジェクト指向システムはコンフィギュレーション (configuration) と呼ばれる、オブジェクトとメッセージが混在する項で表される。

```

module CONFIGURATION is
    using (MULTI-SET * (sort Set to Configuration,
                        op empty to emptyconfig,
                        op _/_ to _/_))
        [TRIV * (sort Elt to ConfigUnit)] .
    using OBJECT .
    sort Message .
    subsort Object < ConfigUnit .
    subsort Message < ConfigUnit .
endm

```

オブジェクトがメッセージを受けてその状態を変化させる様は、コンフィギュレーションの変化で表される。

$$M_1 \cdots M_n O_1 \cdots O_m \rightarrow O'_1 \cdots O'_m M'_1 \cdots M'_l$$

この変化は、書換え規則によりの確に表現される。

5.2 オブジェクト指向モデリングの例

本節では、図書館の蔵書をモデル化した例を紹介する。属性に、タイトル、著者名、貸出中か否かを示す状況、貸出先を持つ蔵書のオブジェクトは次のように書ける。

```

< B-851-028467 : Copy | title = Isabelle-A Generic Theorem Prover,
                        authors = Lawrence C.Paulson,
                        status = checkedout,
                        borrower = 谷津 弘一 >

```

Copyはオブジェクトのクラス名であり、title, authors, status, borrowerは、それぞれタイトル、著者、状況、貸出先を表す属性名である。このオブジェクトは、B-851-028467というIDを持つ、現在筆者が借りている、Lawrence Paulsonが書いたIsabelle-A Generic Theorem Proverという蔵書を表している。

蔵書オブジェクトのクラスは、前に定義したモジュールOBJECTを輸入することを前提として、次のように記述できる。

```

    sorts Copy CopyId .
    sorts Title Authors Status Borrower .
    subsort Copy < Object .
    subsort CopyId < ObjectId .
    subsorts title authors status borrower < Value .
    op Copy : -> ClassName .
    ops Title Authors Status Borrower : -> AttributeId .

```

```
op <_: Copy | title =_, authors =_, status =_,borrower =_> :
  CopyId Title Authors Status Borrower -> Copy .
```

CafeOBJは、構文糖衣としてクラス構文*を持つ。

```
class クラス名 { 属性名とソートの組の並び }
```

図書館の蔵書のクラスは、クラス構文を用いて次のようにして記述することができる。

```
class Copy { title : Title,
              authors : Authors,
              status : Status,
              borrower : Borrower } .
```

蔵書に渡されるメッセージとして、貸出と返却を考える。

```
op _borrows_ : Borrower CopyId -> Message .
op _returns_ : Borrower CopyId -> Message .
```

図書館の蔵書は次のようなモジュールで表される。二つの書換え規則により、メッセージを受けて状態を変化させるオブジェクトの振る舞いが的確に表現されている。

```
module COPIES is
  using CONFIGURATION .
  class Copy { title : Title,
                authors : Authors,
                status : Status,
                borrower : Borrower } .
  ops available checkedout : -> Status .
  op none : -> Borrower .
  op _borrows_ : Borrower CopyId -> Message .
  op _returns_ : Borrower CopyId -> Message .
  var A : Attributes .
  var B : Borrower .
  var C : CopyId .
  rl (B borrows C) < C : Copy | A, status = available, borrower
    = none > => < C : Copy | As, status = checkedout, borrower
    = B > .
  rl (B returns C) < C : Copy | A, status = checkedout, borrower
    = B > => < C : Copy | As, status = available, borrower
    = none > .
endm
```

* 本稿では紹介しないが、継承 (inheritance) を考慮したクラス構文もある。

6. 高階パラメータ化機構

CafeOBJの特筆すべき特徴の一つに、高階のパラメータ化機構がある。CafeOBJでは、パラメータの制約もまたパラメータ化することができる。本章では、高階のパラメータ化機構とその重要性を説明する。

```

module FUN[X Y :: TRIV] is
  op f_ : Elt.X -> Elt.Y .
endm

module MAP[X Y :: TRIV,
           L1  :: LIST[X],
           L2  :: LIST[Y],
           F   :: FUN[X,Y]] is
  op map_ : List.L1 -> List.L2 .
  var E : Elt.X .
  var L : List.L1 .
  eq map nil = nil .
  eq map (E,L) = (f E), (map L) .
endm

```

MAPは、リストの各要素に与えられた関数を作用させる、リスト上のマップ関数を表す高階のパラメータ付きモジュールである。MAPのパラメータL1, L2, およびFを制約するモジュールはそれぞれパラメータ化されている。さらに、L1とFの制約の間でパラメータXが共有され、L2とFの制約の間でパラメータYが共有される。

異なるパラメータが同じ名前のソートや演算を持つモジュールで制約されている場合、そのソートや演算はパラメータ名の修飾により区別される。上の例で言えば、FUNに現われるソートElt.XやElt.Yのようにソート名の後ろにパラメータ名を修飾させて、Xを通して輸入されるソートEltとYを通して輸入されるソートEltが区別される。

MAPを用いて、リストの各要素を2倍する、自然数のリスト上のマップ関数を表すモジュールを作ってみよう。自然数を2倍する関数を表すモジュールは、NATを使って次のように書ける。

```

module DOUBLE is
  using NAT .
  op double_ : Nat -> Nat .
  var N : Nat .
  eq double 0 = 0 .
  eq double (s N) = s s (double N) .
endm

```

MAPの例化は次のようにしてなされる。

```

make MAP-DOUBLE is
  MAP[NAT,NAT,LIST[NAT],LIST[NAT],

```



```

view to DOUBLE is op f_ to double_ . endv]
endm

```

ここで注意して欲しいのは、最初の二つのビューにより、パラメータ x およびパラメータ y を制約する二つの TRIV と NAT の間で、ソート Elt とソート Nat が対応付けられていることである。パラメータ x は $LIST[x]$ と $FUN[x,y]$ の間で、パラメータ y は $LIST[y]$ と $FUN[x,y]$ の間でそれぞれ共有されているので、残りの三つのビューで Elt と Nat を対応付ける必要はない。このため、 $LIST[NAT]$ へのビューでは暗黙のビューが使え、最後の $DOUBLE$ へのビューでは演算 f と $double$ を対応付けるだけで十分になっている。 $MAP-DOUBLE$ は、意味的には次のモジュールと等しい。

```

module MAP-DOUBLE is
  using NAT .
  using LIST[NAT] .
  using DOUBLE .
  op map_ : List -> List .
  var N : Nat .
  var L : List .
  eq map nil = nil .
  eq map (N,L) = (double N), (map L) .
endm

```

ここで、直接輸入されている、あるいは $LIST[NAT]$ や $DOUBLE$ を通じて輸入されている NAT は同一視される。

このような高階パラメータ化機構は仕様記述言語にとって必須の性質である。もし、この高階性がなければ、 MAP は次のような2組のモジュールで表すのを余儀なくされる。

```

module UNARYFUN is
  sorts Elt1 Elt2 .
  op f_ : Elt1 -> Elt2 .
endm

module UNARYMAP[F :: UNARYFUN] is
  using (LIST * (sort List to List1,
                 sort NeList to NeList1,
                 op nil to nil1,
                 op _,_ to _,1_))
  [view to F is sort Elt to Elt1 . endv] .
  using (LIST * (sort List to List2,
                 sort NeList to NeList2,
                 op nil to nil2,
                 op _,_ to _,2_))
  [view to F is sort Elt to Elt2 . endv] .

```

```

op map_ : List1 -> List2 .
var E1 : Elt1 .
var L1 : List1 .
eq map nil1 = nil2 .
eq map (E1 ,1 L1) = (f E1),2(map L1) .
endm

module ENDOFUN is
  sort Elt .
  op f_ : Elt -> Elt .
endm

module ENDOMAP[F :: ENDOFUN] is
  using LIST[F] .
  op map_ : Elt -> Elt .
  var E : Elt .
  var L : List .
  eq map nil = nil .
  eq map (E,L) = (f E), (map L) .
endm

```

最初の二つのモジュールは定義域 (domain) と値域 (codomain) が異なるマップ関数を表すためのものであり、あとの二つのモジュールは定義域と値域が同じであるマップ関数を表すためのものである。高階のパラメータ化機構を用いて記述したMAPに比べて、読みにくく、記述量が増えるのに加え、汎用性も著しく劣る。

7. おわりに

現在筆者らが研究開発中の実行可能な形式仕様言語 CafeOBJ の概要について説明した。

オブジェクト指向における多重継承で表されるような、複雑な輸出入関係を持つモジュールで構成されるソフトウェアシステムは現実にはたくさんある。本稿で説明した高階のパラメータ化機構はこのようなソフトウェアシステムの仕様を的確に記述する上で必要不可欠な仕組みである。

CafeOBJ は、仕様の検証のしやすさと実行可能性に加え、書換え規則の導入によりオブジェクト指向モデリングの支援ができることと、高階パラメータ化機構を持つことから、実用的な形式仕様言語であると言える。

高階パラメータ化機構はモジュールの継承であり、ソート間の包含関係は、ソーートの継承であると言える。本稿では述べなかったが、CafeOBJ ではクラスの継承はソーートの継承で説明される。この2種類の継承の自然な切り分け方を探ることは、ソフトウェアシステムの仕様記述の方法論の探求にも繋がる。これは、今後の大きな課題である。

CafeOBJの高階パラメータ化機構を含めた構造化機構は、ある仮定の下、Martin-Löfの型理論の意味論的枠組みである局所カルテジアン閉圏 (locally cartesian closed category) の中で極めて自然に解釈される^[17]。仮定の内容は数学的に過ぎるのでここでは紹介しないが、その真偽の決定は、代数仕様の分野において本質的な問題であり、筆者の今後の課題の一つである。

CafeOBJ言語処理系は、平成8年度よりIPA技術センタを通じて一般に公開される予定である。

謝辞

CafeOBJの研究開発は、電子技術総合研究所（電総研）言語システム研究室における研究成果に基づき、電総研や北陸先端科学技術大学院大学（北陸先端大）との緊密な協力関係の下に、IPAにおける実行可能な形式仕様言語の研究開発プロジェクトで進められています。

共に研究を進めるプロジェクトメンバーの中川中、本間毅寛の両氏、および、プロジェクトを御指導、御支援くださる、北陸先端大二木厚吉教授を始めとするコンサルティング委員ならびにワーキング委員の皆様感謝いたします。そして、筆者の研究に対して的確な批評と有益な助言をくださった、Oxford大学Joseph Goguen教授とSRIインターナショナルJosé Meseguer博士に感謝いたします。

最後に、日本ユニシス（株）の伊豆エグゼクティブセンター運営に携わる全ての方々に深い感謝の意を捧げます。この方々のおかげで、筆者らは伊豆エグゼクティブセンターにて、OBJとCafeOBJについての極めて有意義なワークショップを開くことができました。

-
- 参考文献 [1] E.Dijkstra, "Go to statement considered harmful", Communications of ACM, 1968.
 [2] 二木厚吉, "代数モデルの基礎", コンピュータソフトウェア, Vol.14, No.1, 1996.
 [3] K.Futatsugi, J.A.Goguen, J.-P.Jouannaud and J.Meseguer, Principles of OBJ2, Proc. of the 12th Symposium on Principles of Programming Languages, ACM, 1985.
 [4] K.Futatsugi, J.A.Goguen, J.Meseguer and K.Okada, Parameterized Programming in OBJ2, Proc. of the 9th ICSE, IEEE, 1987.
 [5] K.Futatsugi and T.Sawada, Design Consideration for Cafe Specification Environment, preprint, 1995.
 [6] J.A.Goguen, T.Winkler, J.Meseguer, K.Futatsugi and J.-P.Jouannaud, Introducing OBJ, Technical Report SRI-CSL-92-03, Computer Science Laboratory, SRI International, 1992.
 [7] J.Meseguer, "Conditional Rewriting Logic as a Unified Model of Concurrency", Theoretical Computer Science, Vol.96, 1992.
 [8] P.Martin-Löf, *Intuitionistic Type Theory*, Bibliopolis, 1984.
 [9] L.C.Paulson, *ML for the Working Programmer*, Cambridge University Press, 1991.
 [10] 谷津弘一, 本間毅寛, 中川中, 二木厚吉, 実行可能な形式仕様言語 CafeOBJ の概要, 第12回IPA技術発表会論文集, 1993.
 [11] 谷津弘一, 本間毅寛, 中川中, CafeOBJによるZ仕様の検証支援, 第13回IPA技術発表会論文集, 1994.
 [12] 谷津弘一, 本間毅寛, 中川中, 澤田寿実, CafeOBJのパラメータ化機構, 第14回IPA技術発表会論文集, 1995.
 [13] 谷津弘一, 二木厚吉, CafeOBJによるZ仕様の検証支援, 第1回ソフトウェア工学の基礎ワークショップ (FOSE'94) 論文集, 1994.also Inソフトウェア工学の基礎 I (日本ソフトウェア科学会 FOSE'94), 近代科学社, 1995 (short version).
 [14] 谷津弘一, 二木厚吉: 代数仕様言語によるZ仕様の検証支援, 投稿中.
 [15] 谷津弘一, 中川中, 本間毅寛, 澤田寿実, 二木厚吉, , 引数間の依存関係を許す代数

仕様の parameterization について, 日本ソフトウェア科学会第12回大会論文集, 1995.

- [16] 谷津弘一, 中川中, 本間毅寛, 澤田寿実, 二木厚吉, 代数仕様言語のモジュール同一視機構について, ソフトウェア工学の基礎 II (日本ソフトウェア科学会 FOSE'95), 近代科学社, 1995.
- [17] H.Yatsu and K.Futatsugi, Categorical Interpretation of Modular Specification in CafeOBJ, preprint, 1995.

執筆者紹介 谷津弘一 (Hirokazu Yatsu)

1985年東京大学理学部数学科卒業. 同年日本ユニシス(株)入社. 現在, 情報処理振興事業協会(IPA)技術センターに出向中, 実行可能な形式仕様言語の研究開発に従事. 情報処理学会, 日本ソフトウェア科学会各会員.



通信プログラム仕様からプロダクション規則仕様への逆変換手法

A Method of Reversely Converting Communications Program Specifications into Production Rule Specifications

深山 尚, 田倉 昭, 太田 理

要約 通信プログラム仕様から宣言型のプロダクション規則仕様の抽出法について提案する。サービスを分かりやすく記述できる通信プロダクション規則仕様から、通信プログラム仕様へ変換することができる。プロダクション規則仕様では、ネットワークをブラックボックスとする規則で記述することができ、非専門家にも理解できるものである。しかし、通信プログラム仕様はネットワーク内の詳細な知識を必要とする。既存のプログラム仕様からプロダクション規則仕様へ逆変換できれば既存のプログラム仕様の再利用性が高まる。そのため事例ベース手法とプロセス間通信の解析を組み合わせて逆変換を行う。これにより、既存通信サービスの再利用性が高まる。

Abstract This paper proposes a method of extracting declarative production rule specifications out of communications program specifications. It is possible to convert the latter which allows the plain description of services into the former. Production rule specifications consist of the set of rules which enables the regarding of any communications network as a black box; thereby making them accessible to even non-experts in communications networks. In the meantime, communications program specifications call for a good knowledge of networking. The availability of reverse conversion from existing program specifications into service specifications helps boost the re-usability of pre-converted ones. This reverse conversion entails a combination of a case study technique and a process-to-process communications analysis technique; thus contributing to the increased re-use of existing communications services.

1. はじめに

ますます高度化、多様化する傾向にある通信ソフトウェアの世界ではユーザプログラマビリティの必要性が叫ばれている。その実現のためには、ユーザにとって理解性、記述性ともに高い記述方式の提案を行うことである。

理解性の高いプロダクション規則言語としてSTR (State Transition Rule) が提案されている。この設計法では通信システムの外部から認識可能な端末の動作の変化を表現する規則の集合で記述することができる^[1]。

当研究所では、要求仕様であるSTR仕様から、プログラム仕様であるSDL^[2]への自動生成を行っている。SDL仕様は有限状態機械をモデルとしていて記述能力に優れ、グラフィカル表現も有し、最も普及している。しかし、SDL仕様よりソフトウェアの再利用性が高く、理解性に富んだ要求仕様にて再利用を行ったほうが保守や開発効率の面からも好ましい。

プログラム仕様から要求仕様への研究は数多くなされており、SDL仕様からメッセージシーケンスチャート (MSC) への逆変換がある^[3]。しかし、MSCにおいては、

遷移の過程を時系列に意識しなければならず、要求仕様での再利用性において問題が有ると考える。

また、リエンジニアリングにおける言語間の変換においても、既存のインプリメントから要求仕様を経由し、新しいインプリメントへの変換を行っている^{[4]~[6]}。しかし、既存のインプリメントから抽出された要求仕様は既存のインプリメントの影響を色濃く残している。

本稿では、状態遷移機械を用いて記述されたプログラム仕様（SDL仕様）から要求仕様であるSTR仕様を抽出するというリバースエンジニアリングを検討する。しかし、SDL仕様からSTR仕様へ逆変換には二つの問題がある。

SDL仕様では有限状態遷移機械なので状態を全て管理しなければならず、さらにローカル記述のため通信システム内部の動きを意識する必要がある。STR仕様では、部分適用化により扱う状態の量を減らすことができ、グローバル記述化により通信システム内部のプロトコルを意識する必要性が無い。また、状態の時系列を把握する必要性も無い。これら違った特徴をもつ記述方法で、SDL仕様からSTR仕様への変換には、完全適用から部分適用への問題とローカル記述からグローバル記述化への問題が存在する。

完全適用から部分適用への問題はユーザとの対話を用いた方法を使用し、ローカル記述からグローバル記述への問題には内部信号を動的に解析することにより解決を図った。

これにより、既存の多くの通信ソフトウェアをSTR仕様を用いた仕様環境下で、利用率を高めることができる。

本稿の構成は以下の通りである。2章では通信ソフトウェアの設計法とSTR仕様、SDL仕様の関係を述べる。3章ではどのようなSDL仕様ならばSTR仕様へ変換できるかを述べる。4章では逆変換の具体的方法を、5章でまとめとなる。

2. 通信ソフトウェア仕様設計法

STR-SDL自動変換システムでは、プロダクション規則STR仕様からプログラム動作仕様への変換が行われている。図1で示すように、非専門家対象のSTR仕様を使用し、グラフィカルなツールにて対話しながら通信サービスの内容を具体化していく。書かれたSTR仕様に矛盾や抜け落ちがないように各種検証ツール、補完ツールもサポートしている。このように要求された通信サービスを満足するSTR仕様から、システムを構築する個々のエンティティの振る舞いを表現するプログラム仕様への自動生成がなされる。生成されたプログラム仕様は文書としての価値をもち、さら

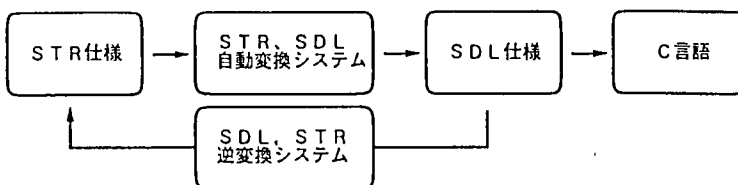


図1 通信ソフトウェア開発環境と逆変換概要

に実機を稼働するためにC言語であるプログラムへと変換される。

プログラム動作仕様は従来から形式化、標準化が進んでおり、特に有限状態遷移機械 (FSM) を基本にしたSDL仕様が広く使用され、多くのプロトコルがSDL仕様にて記述され資産として残されている。

これら、SDL仕様で書かれた膨大なプログラム動作仕様をSTR言語に変換できれば、STR言語は非専門家対象のため、保守、機能追加の容易性に大変魅力的である。今回、リバースエンジニアリングとして対象とするのはSDL仕様からSTR仕様への変換とする。

しかし、如何なる条件のSDL言語を全てSTR言語に逆変換することは不可能である。そのため研究のステップ1としてSTRからSDLへの変換するプロトコルを使用したSDLのみを逆変換の対象とした。

2.1 プロダクション規則仕様言語STR

STR仕様は「現状態」、「イベント」、「次状態」の三つの部分から構成される。状態は、端末の状態と、適当な2端末間の関係で表現される。状態は、状態要素から構成される。プリミティブは、属性の違いによる固有の名称と、端末対応のための変数を持つ。プリミティブが持つ端末対応のための変数には、1変数のものと、2変数からなるものがある。どちらのものも、第一変数には動作対象の端末を記述する。第二変数は、第一変数の端末に束縛されているという関係を示す。

例えば、図2のように記述される。

```
ringback(B, A), ringing(A, B) offhook(A) :
path(A, B), path(B, A)
```

図2 通信サービス記述例

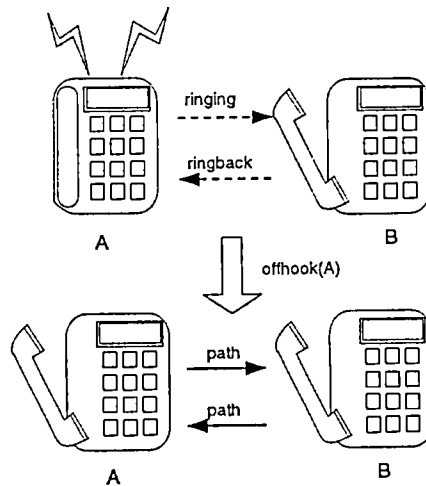


図3 STR記述と状態遷移イメージ

図2の規則は「端末Aが端末Bから呼び出しを受けている時、端末Aがoffhookすると、端末A、B間で通話状態に遷移する」ことを規定している(図3)。

STR仕様の現状態において、イベント生起点からSTR仕様内に記述されている端末間にパスが存在する。このように端末間の連結性を議論するには、論理式よりもグラフ表現の方が適している。

STR仕様中の現状態とイベントを合わせてラベルつき有向グラフとして表現でき、次状態もラベルつき有向グラフとして表すことができる。

図2のSTR仕様に対するグラフ表現を図4に示す。

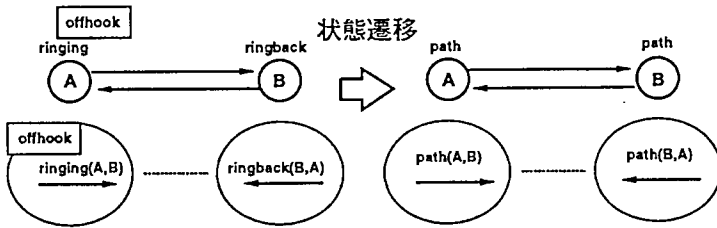


図4 STR記述に対するグラフ表現

2.2 STR仕様, SDL仕様の関係

STR仕様とSDL仕様の関係および特徴, SDL仕様からSTR仕様への変換過程を明記する。

部分適用はプロダクションシステムでは一般的なルール適用方法であり, STR仕様も同様な特徴を持っている。適用条件部を満足している場合, すなわち, ネットワーク内の状態(実状態と呼ぶ)がSTR規則の適用条件部を包含している時にSTR規則が適用され状態遷移がおこる。これに対し, SDL仕様では実状態とSDL仕様で記述されている状態が完全一致している時のみ状態遷移が行われる。このような規則適用方法を完全適用と呼ぶ。完全適用では, 遷移する状態を全て把握しなければならないのに対し, 部分一致適用では必要な条件のみを把握すればよいというメリットがある。

また, ネットワーク全体の状態遷移を記述する方法をグローバル記述, ネットワーク内の部分的な状態遷移を記述する方法をローカル記述と呼ぶ。グローバル記述で通信仕様を記述する場合, 通信プロトコルに関する知識が不要である。それに対し, ローカル記述の場合は, 通信プロトコルに関する知識, 具体的にはネットワーク全体の状態を知るために, 各々の状態を一つにまとめるために信号のやり取りをする。この信号を内部信号と呼ぶ。

したがって, ネットワークの非専門家にとって, ネットワーク全体を記述するにはグローバル記述で部分適用の言語仕様がふさわしいと考える。

仕様記述法を適用法の観点から部分一致適用と完全一致適用に, 記述範囲の観点からグローバル記述とローカル記述に分類した。それぞれの特徴を組み合わせることで4種類の記述法が考えられる。図5にて説明する。

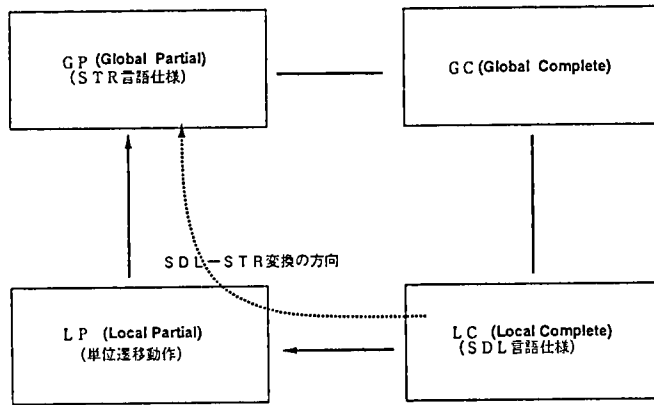


図5 STR, SDL変換概要

- ・ GP (Global Partial)

STR仕様のことを意味する。グローバル記述で部分一致適用を可能とする言語仕様となっている。
- ・ LC (Local Complete)

SDL仕様のことを意味する。ローカルな記述でかつ完全適用を意味している。
- ・ LP (Local Partial)

ローカル記述で部分適用の物をいう。これを単位遷移動作と呼ぶ。
- ・ GC (Global Complete)

グローバル記述でかつ完全適用の場合をいう。今回の変換過程では存在しない。

3. STR仕様へ変換できるSDL仕様のモデル

既存のSDL仕様を全てSTR言語仕様へ変換することはできないので、対象とするSDL仕様を制限する。以下の条件を満たしていればSTR言語仕様へ変換できる。

- SDL仕様上のプロセスの動きと端末の動きが1対1である

SDL仕様上のプロセスの動きと端末の動きが1対1であり、それぞれのプロセスは同型であること、プロセスはidle状態から始まりidle状態へ戻る必要がある。図6に示してあるように、STR言語の記述は外部認識可能な端末の状態変化を引き起こす入力に対してイベントを割り付け、端末の状態変化として仕様を記述している。STRからSDLへ変換するプロトコルにはSTR仕様の端末IDの動きをもとにプロセスを生成してある。そのため、SDL仕様上のプロセスの動きはSTR仕様の端末IDの動きと同じでなければならない。
- SDL仕様中の状態が明記されている

SDL仕様は有限状態遷移機械であるので、それぞれの遷移過程において状態が存在する。この状態がそのプロセスを中心に連結な有向グラフで表現で

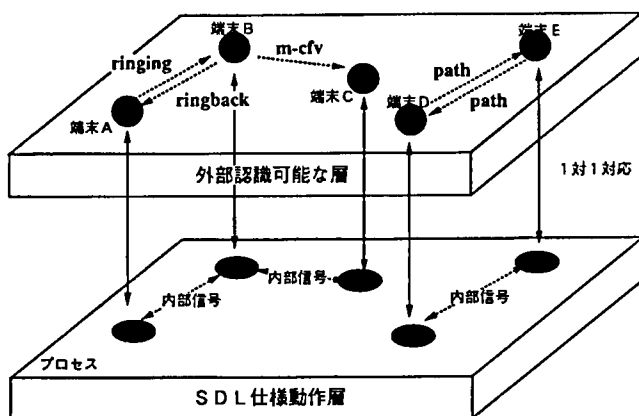


図6 SDL上のプロセスと端末の動き

きるものでなければならない。例えば、自プロセスから path がはられていたら path (Q0, Q1) となる。Q0 は自プロセス ID で Q1 は自分以外のプロセス ID のことである。これら、プロセス間の関係を示すものをプリミティブと呼ぶ。

- 画一的な内部信号の動き

SDL仕様はローカル記述であるので、内部信号にてネットワーク内を探索し、STR仕様に当てはまる状態遷移を行う。

変換する対象のSDL仕様がSTR-SDL自動変換時の内部信号の形態と同じプロトコルにて書かれていることが必要である^[7]。このプロトコルとは、イベント生起点のプロセスにてローカルな状態を把握した後、関係のあるプロセスへ探索信号を発信する。この時、SDL仕様上の状態中のプリミティブ先が存在するプロセスのみ内部信号を送信できる。内部信号を受信したプロセスは、このローカルな状態を把握したのち、他に探索をするプロセスがあれば内部信号を送信する。探索最終点のプロセスにてどの様に遷移するかを決定し、このプロセスから関係のあるプロセスへ遷移命令である内部信号を返信する。また、ローカルな状態にSTR規則との不一致が生じた場合は拒否信号を関係のあるプロセスに返信する。

4. SDL仕様からSTR仕様への変換方法

以下の経路にて変換を行う。

LC (SDL仕様) $\rightarrow LP \rightarrow GP$ (STR仕様)

GC経由では扱う記述量が多いので、LP経由での変換を手段として選ぶ。GCは完全適用記述であるので、実状態の全ての状態を処理しなければならない、LCと同数の状態を処理しなければならない。

LP経由では、部分適用化をGCからLPへの初期の段階で行ってしまうので、管理する状態数が少なくすむ。したがって、LP経由の方法を採用する。

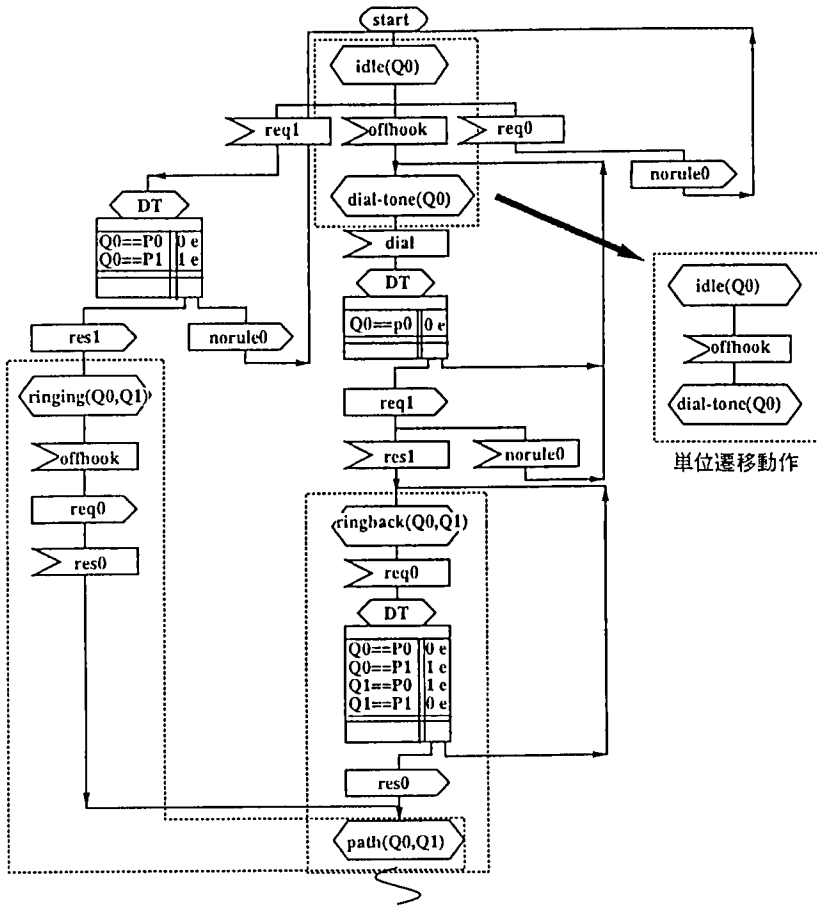


図7 LCからLPへの変換

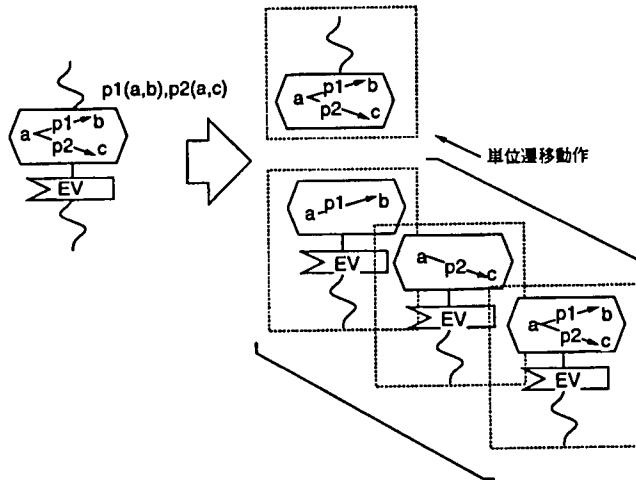


図8 完全適用から部分適用への変換

4.1 LCからLPへの変換

LP (単位遷移動作) への変換を行う。単位遷移動作とは現状態、イベント、次状態の三つの構成要素を持ち、現状態、次状態ともローカルな状態を記述してあるものである。

図7にあるように、プロセスをイベントごとに分割し単位遷移動作にする。

このとき、現状態での完全適用の状態から部分適用の状態へ各々の状態を分割しなければならない。図8に示すように、SDL仕様上の状態とはある単位遷移動作の次状態と別の単位遷移動作の現状態が重なったものである。また、部分適用とは、次状態に含まれるプリミティブで次への遷移が行われるものである。すなわち、部分適用部を見つけることは、これらプリミティブから必要なプリミティブを選び出すことである。

単位遷移動作化するために、AND手法、事例ベース手法を用いた手法を述べる。

図9に示すように、AND手法は二つ以上の同一イベントの単位プロセスで、現状態のプリミティブの内容のANDを取り部分一致適用部を見つけるものである。欠点としては、二つ以上同一イベントの単位遷移動作が存在しない場合と、プリミティブのANDにて設計者の意図した部分一致適用のプリミティブを抽出できないことの2点があげられる。

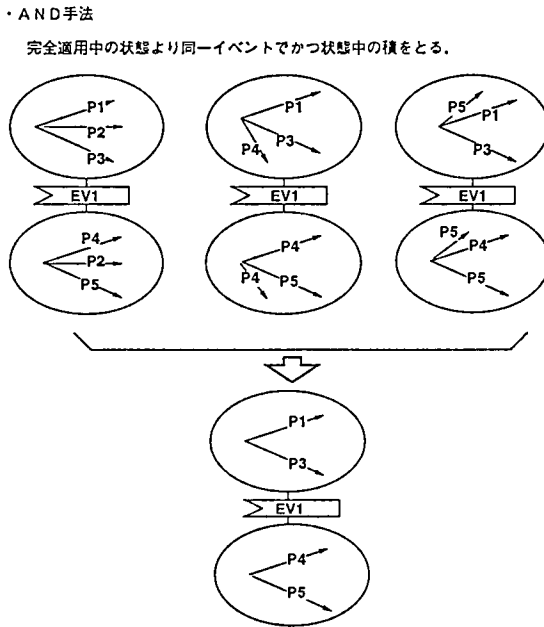


図9 部分適用解決方法その1

図10に示すように、事例ベース手法は過去の部分適用の事例から部分一致適用のプリミティブを見つけ出す方法である。

- 1) 過去の部分適用の例をイベントごとに分類してあるものから、対象とする

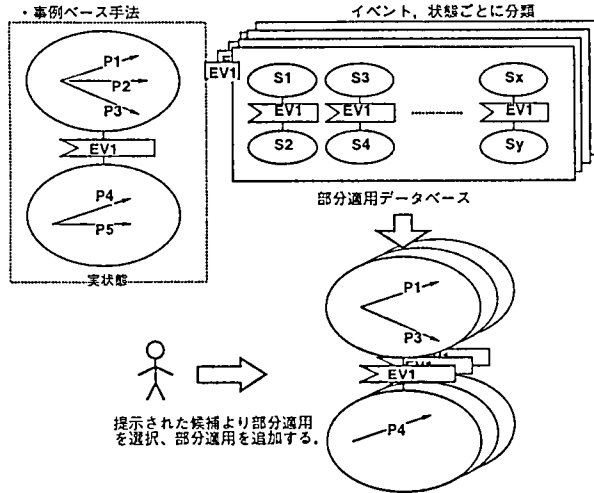


図10 部分適用解決方法その2

SDL仕様のイベントと同一のものを候補として出す。

- 2) 対象としているSDL仕様の現状態が事例ベース上の現状態を含んでいるものを候補として出す。
- 3) 対象としているSDL仕様の次状態が事例ベース上の次状態を含んでいるものを候補として出す。
- 4) 候補として残ったものを候補として提示しユーザに選択してもらう。

SDL仕様の現状態が候補として提示された状態と異なった部分適用を選択した場合は、この部分適用の状態および、イベントを部分適用データベースへ登録する。

事例ベース手法では、過去の事例よりSDL仕様上の実状態中に目的の部分一致適用プリミティブを見つけ出し、ユーザに候補を提示する。そのため最終的に判断するのはユーザとなる。ここでユーザとはSDL仕様からSTR仕様へ変換する人をいう。ユーザはSDL仕様を設計した人物の部分適用部を把握しなければならない。

さらに、事例ベース手法は過去の事例が存在するか否かによって変換の効率が変わってくる。

以上、AND手法、事例ベース手法について述べたが、実際は、最初にAND手法によってSDL仕様上の部分適用部分を絞る。さらに事例ベース手法によって最終的に部分適用部分を決定する。この一連の流れを、SDL仕様実状態上でプリミティブが二つ以上存在する状態全てに行う。

4.2 GPへの変換

LPでは単位遷移動作が存在する。これは、一つのプロセス上でローカル記述である現状態、次状態が存在し、かつ内部信号が存在する。これをグローバル記述である現状態、次状態に変換する。

グローバル記述化のために、単位遷移動作の振る舞い上の内部信号をつなげ合わ

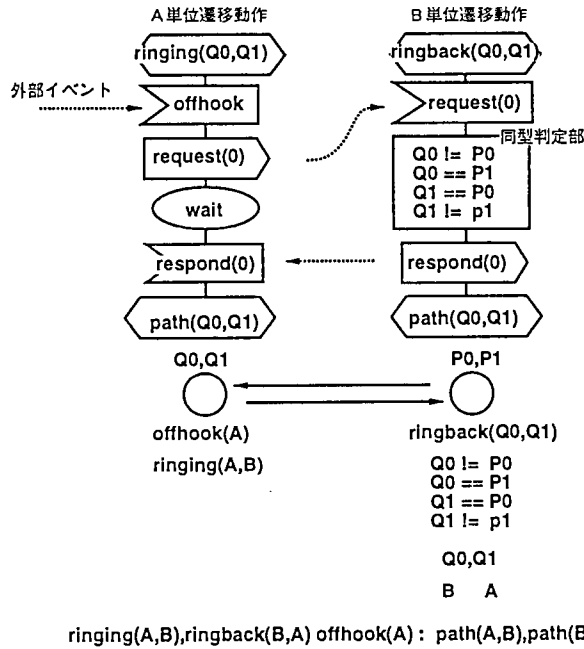


図 11 LP から GP への変換

せて消滅させ、グローバル記述化した端末IDの決定を行う。

グローバル記述化へのアルゴリズムを以下に示す。

- 1) 外部イベント生起点が存在する単位遷移動作を見つける。
- 2) 1)で抽出した単位遷移動作の内部信号IDを同じ受信信号をもつ単位遷移動作を見つける。
- 3) その単位遷移動作内にて、端末IDの決定を行う。
- 4) 内部信号の送信を持たない単位遷移動作に到着するまで2), 3)を繰り返す。

図 11 に本アルゴリズムによるグローバル記述化の例を示す。ringing(Q0, Q1) を現状態を持つ単位遷移動作は offhook の外部イベントを持っていて、request(0) の内部信号を送信している。一方 ringback(Q0, Q1) の始状態を持つ単位遷移動作は request(0) を受信している。

各々単位遷移動作には Q0, Q1 の ID を持っているが、Q0 が自分の単位遷移動作で、それ以外が相手の単位遷移動作である。Q0, Q1 の関係を示すものが同型判定部である。例では P0 が A 単位遷移動作の Q0 に相当し P1 が Q1 に相当する。Q0, Q1, P0, P1 の関係式よりイベント生起点の単位遷移動作を A とすれば、Q0 は B に、Q1 は A に相当することが分かる。

それらを STR 仕様の現状態、イベント、次状態に端末 ID として記入すれば図 11 の下の規則式になる。

5. 結 論

今回、プログラム仕様である SDL 仕様から、グローバル記述、部分適用型である

STR仕様への逆変換を行った。まず、リバースエンジニアリングの通信ソフトウェア仕様設計における位置付けを明確化しSTR仕様に変換できるSDL仕様の範囲を明らかにした。そしてその条件に合うSTRの抽出方法を提案した。以上により、より効率的で信頼の高い通信ソフトウェアの開発を期待することができる。

しかしながら、逆変換の問題点を述べると以下の通りとなる。

- 部分適用抽出部における抽出効率の問題

部分適用では、AND手法にて候補を絞った後、事例ベース手法にてプリミティブを候補として表示しユーザに選択抽出してもらうものである。事例ベースでの欠点として、過去の事例がどれだけ存在するかがSDL-STR変換効率の重要なポイントとなる。最悪の場合、プリミティブが二つ以上存在する状態全てにユーザが選択しなければならなくなる。

- 内部信号の振る舞いの条件

提示した内部信号の振る舞いは、イベント生起のプロセスから順次探索を行うものである。しかし、その内部信号の送信は、一方向でかつSDL上の状態のプリミティブに沿って内部信号を送信しなければならない二つの条件がある。グローバルの視点にて説明するならば、イベント生起点のプロセスから一方向に、同じプロセスを二度と経由しないで探索を行うことである。そして、探索の最後のプロセスにてどの様に遷移するかを決定し、関係するプロセス全てに遷移命令信号を送信する。

これら二つの条件のうち、プリミティブに沿って内部信号を送らなくてはならない条件を解除するには、具体的には、内部信号の動きにはプリミティブの方向とは違った内部信号の送信先が存在することを許すことである。

解決法として、プリミティブとは違った所への内部信号を送信したプロセス単独でプロセスIDを決定する解析はできないので、イベント生起点プロセスから探索したプロセス個々の全ての状態、内部信号受信IDの情報を全てスタックに積み上げておく必要がある。

内部信号の動きで、一方向の動き以外の振舞いとしては、あるプロセスから複数の他プロセスに向かい、同時に複数内部信号を送信する方法が考えられる。

この解決法としては、LCからGPへの変換過程において、複数の応答信号から同時にプロセスIDを決定できるように解析を行う必要がある。

今後、以上の問題点解決法についてのアルゴリズムの提示を行うつもりである。

本研究を進めるにあたり、御指導と励ましをいただきましたATR通信システム研究所の葉原耕平会長と寺島信義社長、ならびに有益な議論をいただいた研究室の皆様感謝の意を表す。

-
- 参考文献 [1] Y. Hirakawa, T. Takenaka, "Telecommunication service description using state transition rules". Proc. Sixth Int. Work. Software Specification and Design, pp. 140~147, Oct., 1991.
- [2] ITU-T: Recommendation Z.100, "Functional Specification and Description Language (SDL)", 1992.

- [3] 大友健治, 荒川則泰, 山中顕次郎「通信ソフトウェアにおけるリバースエンジニアリング」電子情報通信学会 交換システム研究会 SSE94-20, IN94-20, 1994年4月.
- [4] H. M. Sneed, G. Jandrasics, "Software Recycling", Proc. Conf. on Software Maintenance, 1987, pp.82~90.
- [5] E. J. Byrne, "Software Reverse Engineering: A Case study", Software Practice and Experience, Dec. 1991, pp.1349~1364.
- [6] R. C. Waters, "Program Translation via Abstraction and Reimplementation", IEEE Trans. on Software Engineering, Aug. 1988, pp.1207~1228.
- [7] 河田慶三, 田倉昭, 太田理「端末の動作記述からプロセス仕様を自動生成する手法について」電子情報通信学会 交換システム研究会 SS92-21, 1993年1月.

執筆者紹介 深山 尚 (Hisashi Miyama)

1986年茨城大学工学部卒業。1988年同大学院修士課程修了。同年日本ユニシス(株)入社。通信ソフトウェアの開発に従事。1993年4月ATR通信システム研究所に出向。通信ソフトウェアの自動生成に関する研究に従事。1995年4月, ATR知能映像通信研究所に出向。画像空間内, 動きのハンドリングに関する研究に従事。情報処理学会会員。



田倉 昭 (Akira Takura)

1979年東京工業大学理学部 情報科学科卒業。1981年同大学院修士課程修了。同年日本電信電話公社(現日本電信電話株式会社)入社。プロトコル設計支援および, その適用技術に関する研究に従事。1992年2月ATR通信システム研究所に出向。情報処理学会会員。IEEE会員。



太田 理 (Tadashi Ohta)

ATR通信システム研究所通信ソフトウェア研究室長。1968年九州大学工学部 電子工学科卒業。1970年同大学院修士課程修了。同年日本電信電話公社(現日本電信電話株式会社)入社。電子交換機のソフトウェア研究開発に従事。1987年からNTTにおける情報・通信機器向けの共通OSの開発に従事。この間トロン協会CTRON専門委員会の幹事を歴任。1992年にATR通信システム研究所に出向。現在に至る。工学博士。電子情報通信学会, 情報処理学会, IEEE会員。



ソフトウェアレビューの質を参加者へのテストにより評価する方法

An Evaluation Method for Software Review Quality by Imposing a Test
on the Reviewers

的 場 政 明, 高 田 義 広, 鳥 居 宏 次

要 約 開発中のソフトウェアに含まれるバグは、発見が遅れるほど、その修正に要するコストが高くなる傾向がある。そこで、一般のソフトウェア開発の各工程においては、その工程の成果物に含まれるバグを早期に発見するためにレビューが実施される。

ただし、レビューにおいて全てのバグが発見されるとは限らない。しかも、バグを発見する率は、レビュー者の能力や経験、レビューに費やした労力などに依存して、大きくばらつくので、バグの発見率が極端に低くなる場合もある。したがって、開発管理者は、そのようなレビューの質の低下を察知して対策をとる必要がある。ところが、バグを見逃したことは、後の工程で発見されるまで不明である。そのために、従来、そのような察知を行うことが困難であった。

そこで、本研究では、コード作成工程において実施されるコードレビューについて、実施の直後、あるいは、途中でその質を定量化する方法について検討した。本稿では、コードレビューの直後、あるいは、途中でレビュー者に簡単な筆記試験を課すことにより、コードレビューの質を定量化する方法を提案する。そして、提案する方法を適用した実験について報告する。

Abstract The later bugs are detected in the software being developed, the more it tends to cost to fix those bugs. That is why each process of general software development requires a review for an early detection of bugs in the resultant product of each process. The review, however, does not always help detect all bugs. Besides, the bug-finding rate largely fluctuates depending on reviewers' skill and experience as well as the amount of their labor required for reviewing, and so there are some cases where the rate marks an extremely low level. As a result, there arises a need for the development supervisor to take some precautions by perceiving the possibility of such a fall in the quality of reviewing. Once bugs are missed, they keep existing till they are detected in a later process. That has so far caused the difficulty of such perception.

For this reason, the authors joined forces to study an approach by which to grasp, in terms of quantity, the quality of code reviewing which takes place in the processes of coding, right after or during a review. This paper proposes a method of quantifying the quality of a code review by imposing an easy-to-answer test on reviewers immediately after or during their reviewing, and additionally reports on an experiment for which the proposed approach was adopted.

1. はじめに

高品質のソフトウェアを効率良く開発するためには、テスト工程でバグを検出・修

正するだけでなく、それ以前の工程であるレビューによってバグを検出・修正することが有効であることが知られている^[1]。したがって、ソフトウェア開発を管理する上では、テストだけでなく、レビューが十分に行われたかどうか(レビュープロセスの質と呼ぶことにする)を確認することが重要である。

ところが、レビューの方法については多くの提案があるが、レビュープロセスの質を確認する方法についてはほとんど提案がない^{[2][3]}。また、テストについては、その質を評価するための尺度(例えば、テスト網羅率や信頼度成長モデルに基づく残存バグ数の推定値)がいくつも提案され実用されているが、レビューについてはほとんどない。レビューに費やした時間や調べた項目の数を尺度とすることも考えられるが、それらの有効性については疑問が残る。レビュープロセスの質を評価するための実用的な尺度、および測定方法が必要である。

そこで、我々は、レビュー直後(または、途中)のレビュー参加者に対して簡単な筆記試験を実施することにより、レビュープロセスの質を間接的に定量化することが可能であるかどうかについて検討している。本発表では、レビュー参加者への試験の方法を提案し、試験の成績とコードレビュープロセスの質との関係についての実験結果を報告する。実験では、同一のプログラムに対して8名の被験者に独立にコードレビューを行ってもらった。試験問題の作成方法についても検討の余地が残されているので、3種類の問題を作成し、各被験者に全ての問題に回答してもらった。そして、各被験者の試験の成績とバグを検出した率とを比較した。また、各種類の試験問題がどの種類のバグの検出と関連しているかについて考察した。

一般に、コードレビューの目的としては、バグの検出・修正だけでなく、コードの可読性やプログラムの性能の分析・改善などが含まれることがある。ただし、今回、提案する方法では、バグの検出率だけに限定してレビュープロセスの質を評価することを目的としている。ここで言うバグとは、プログラムが仕様には合わない場合に原因となっているソースコードの部分である。構文エラーなどは含まない。

なお、我々は、文献^[4]においても、レビュー参加者に対する試験を試みた結果をすでに報告している。今回、発表する試験方法は、文献^[4]で示したアイデアを発展させた方法であり、より正確な評価値を得るための細部の手順が追加・改善されている。5章で述べるように、今回、報告する実験の結果と文献^[4]の実験の結果との間に一貫性があった。この分析から、提案する方法がプログラムの種類に依らずに有効であることを示唆するデータが得られた。

以降、2章では、提案する試験方法を説明する。3章では、実験で使用した試験問題の作成方法を説明する。4章では、実験の結果を報告する。5章では、実験結果に考察を加える。6章では、まとめと今後の課題とを示す。

2. 試験方法

提案する方法では、レビュー参加者以外に、試験官が必要である。試験官は、レビューに直接に参加せず、試験問題の作成と試験の実施だけを担当する。実際のソフトウェア開発では、プロジェクトリーダーや、プロジェクトと独立に存在する品質管理部などが、試験官を兼ねればよい。

試験官の作業は以下の通りである。

- 1) レビュー参加者に、ソースコード、プログラム仕様書を参照しながらコードレビューを行ってもらおう。コードレビューの方法については特に制約を設けない。
- 2) ソースコードが作成された後からコードレビューが終るまでの間に、レビュー参加者と独立に試験問題を作成する。問題は、ソースコードと仕様書とから作成する(3章参照)。
- 3) コードレビュー直後(または、途中)のレビュー参加者に試験問題を与える。そして、試験問題以外の資料を一切見せず、記憶だけを頼りにレビュー参加者に回答してもらおう。試験時間は特に限定しない。
- 4) 回答を採点し、コードレビューの質を間接的に評価する。

3. 試験問題

コードレビューとは、バグを検出するために、仕様とソースコードとを比較する工程であるとみなせる。そこで、我々は、仕様とプログラムとの両方について、レビュー参加者が細部まで読んで理解した程、バグを検出する率が高く、コードレビュープロセスの質が高いと考えた。そこで、試験問題として、仕様とコードとの理解の程度を試す3種類の問題、問1~3を作成した。

問1. コード把握度問題

ソースコードに対する理解の度合を試す問題であり、yes/noで答えられる2種類の多数の小問題からなる。各種類の問題は、ソースコードを基にそれぞれ次のように作られる。

- ソースコード中の分岐文、ループ文、関数呼び出し文、代入文の中から、無作為にいくつかの文を選び出す。そして、各文について、その文の動作を説明する記述を作る。説明を作る時、各記述には、5割の率で意図的に1個の誤りを混ぜる。この説明が実際のコードと一致するかどうかが問われる。
- いくつかの関数を無作為に選び出す。そして、その関数の中の関数呼び出しの回数や順番、または、条件分岐の回数や順番を説明する記述を作る。各記述には、やはり5割の率で誤りを混ぜる。この説明が実際のコードと一致するかどうか問われる。

小問題の作成の例を示そう。ここでは、C言語で書かれたソースコードの中に次のようなif文が入っていた場合を想定する。

```
if((fp=fopen(fname, "+r") == NULL)
    mainprog(fp, i);
```

このプログラムは、テキストファイルを読み出して何らかの処理を行うプログラムである。上のif文は、ファイルを読み出すためにファイルを開く関数 `fopen` を呼び出した後、エラーが発生しなければ関数 `mainprog` を呼び出すことを表している。このif文に対して、例えば、次のような説明を作成する。

関数mainの中にはファイルを開くための関数fopenを呼び出している文がある。この時、関数fopenには、ファイルの名前とリードオンリーを指定する引数 (“r”)とを渡している。

下線は、誤りの含まれている恐れのある部分をレビュー参加者に示している。上のif文を見るとわかるように、実際にはリードライトを指定する引数 (“+r”)を渡している。従って、この小問題に対する正解は“no”である。コードレビューの際にこの引数を確認したレビュー参加者にとっては、容易に答えられる問題である。

問題の採点について述べる。各小問題は2者択一の形式であるので、コードを全く理解していない者が1個の小問題にも正答しない訳ではない。無作為にyes/noを選択すれば、平均的に5割の小問題に正答する。そこで、問1の成績は次式のように計算する。

$$\text{問1の成績} = 2 * \left(\frac{\text{正答した小問題の数}}{\text{小問題の総数}} - 0.5 \right)$$

コードを完全に理解している者の成績は1、全く理解していない者の成績は0となることが期待される。

問2. 仕様把握度問題

プログラム仕様に対する理解の度合を試す問題であり、yes/noで答えられる多数の小問題からなる。各小問題は、仕様書を基に次のように作られる。まず、仕様書の中から、無作為に1個の文、または、図表中の1個の項目を選び出す。仕様書の中のそのような文や項目は、プログラムの入出力に何らかの関係があるはずである。そこで、その文や項目に関係する入力とその入力に対応する出力との一例を挙げ、それらの入出力を説明する記述を作る。説明を作る時、問1と同様に、各記述には、5割の率で意図的に1個の誤りを混ぜる。この説明が実際の仕様と一致するかどうか問われる。

小問題の作成の例を示そう。ここでは、ある言語の構文解析を行うプログラムの仕様書の中に、次のような構文定義が書かれていた場合を想定する。

変数宣言 = int変数名 [“,” 変数名] “ ; ”

変数名 = 英字 {英字 | 数字}

この部分は、正しく解析できる変数宣言の構文を表している。正しく解析できなかった場合は、適当なエラーメッセージを出力して、停止することになっていたとする。この構文定義に対して、例えば、次のような説明を作成する。

入力に “ int a, b, c; ” と言う変数宣言が含まれる場合、エラーメッセージを出力する。

上の構文定義を見るとわかるように、3個の変数を1度に宣言する構文は許されていない。したがって、この小問題に対する正解は“yes”である。コードレビューの際にこの定義を確認したレビュー参加者にとっては、容易に答えられる問題である。

問題の採点は、問1と同様に次式によって計算する。

$$\text{問2の成績} = 2 * \left(\frac{\text{正答した小問題の数}}{\text{小問題の総数}} - 0.5 \right)$$

仕様を完全に理解している者の成績は1, 全く理解していない者の成績は0となることが期待される。

問3. コード変更部発見問題

この問題は、問1と同様にソースコードに対する理解の度合を試す問題であるが、問1とは形式が異なる。問題は、ソースコード中のいくつかの箇所を試験官が変更することによって作られる。そして、この変更されたソースコードと変更箇所の数だけがレビュー参加者に提示され、どこが変更されたかが問われる。

問題の採点について述べる。コードを完全に理解している者は、全ての変更箇所を正しく言い当てるはずである。一方、コードを全く理解していない者は、全く答えられない、あるいは、変更されていない箇所を誤って指摘すると思われる。全く答えられなかった者と、変更されていない箇所を指摘した者との間では、コードの理解に対する程度に大きな差があるとは考えられない。そこで、問3の成績は次式のように計算する。

$$\text{問3の成績} = \left(\frac{\text{正しく指摘した変更箇所}}{\text{変更箇所の総数}} \right)$$

この成績は、問1の成績と同様に0～1の値をとる。

4. 実 験

4.1 レビュー対象のプログラム

実験に使用したプログラムは、酒屋倉庫管理問題のプログラムである。大阪大学情報工学科の演習で学生によって作成された。C言語で記述されており、608行の大きさであった。コンパイラが検出できる構文上の誤りは、あらかじめ除去されていた。

事前のテストにより、17個のバグが潜んでいることがわかっていた。バグの個数

表1 実験プログラム中のバグ

バグの原因	個数
変数の更新の洩れ	2
関数の戻り値の判定洩れ	2
NULLポインタの参照	2
ループの制御変数の誤り	2
式の誤り	2
定数の誤り	1
型の誤り	1
引数の誤り	1
その他の仕様との相違	4

は、もちろん、被験者に知らされなかった。潜んでいたバグの原因を表1にまとめる。

4.2 手 順

前述のプログラムに対して、8名の被験者に独立にコードレビューを行ってもらった。被験者は、奈良先端大の7名の大学院生と筆者らの1人が勤めた。一般にコードレビューは複数の参加者により共同で行われることが多いが、今回の実験では1名だけで行ってもらった。できるだけ多くの回数の実験を行うことを優先したためである。

各被験者には、ソースコードのハードコピーと、仕様書と、筆者らが作成したプログラムの概要を説明する文書とが渡された。このプログラムの説明書には、プログラム中に定義されている各関数の概要が書かれていた。そして、時間の制限を設けずにコードレビューを行ってもらった。レビューの方法は各被験者の自由に任された。ウォークスルーや文献¹⁵⁾で提案されている方式のように、コードの作成者(または、その役割を演じる者)がコードの説明を行い、被験者がバグを探す方式も考えられた。しかし、そのような方式では、説明の仕方が実験ごとに変化する恐れがあったので、各被験者には、文書だけを見ながら単独でレビューを行ってもらった。各コードレビューに要した時間は2~3時間であった。

作成した試験問題の規模は以下の通りである。まず、問1として40個の小問題、次に、問2として19個の小問題を作成した。最後に、ソースコード中の4関数より7箇所を削除して問3を作成した。

4.3 結 果

各被験者についての試験の成績とバグの検出率を表2に示す。ここで言うバグ検出率とは、潜んでいたバグの総数に対して、被験者が検出したバグの割合を意味する。表からわかるように、本実験のように、同じ条件の下でのコードレビューでも、被験者によってバグ検出率が大きく異なっている(18~65%)。

図1に問1~3の成績の平均とバグ検出率との相関を示す。図からわかるように、試験の成績と検出したバグの数との間に強い関連があった。相関係数は0.746であっ

表2 実験結果

被験者	バグ 検出率	成 績			
		問1	問2	問3	平均
A	.65	.56	.56	1	.77
B	.35	.46	.78	.86	.70
C	.35	.36	.44	1	.60
D	.35	.26	.34	.29	.30
E	.29	.30	.48	.71	.50
F	.24	.36	.56	.43	.45
G	.18	.46	.34	.43	.41
H	.18	.08	.34	.29	.24

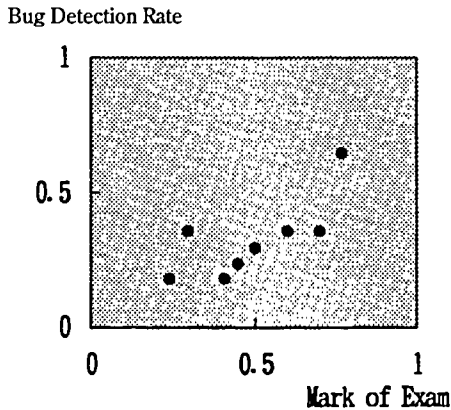


図1 試験成績とバグ検出率との相関

た。また、問1～3の各問題の成績と検出したバグ検出率との相関係数は、それぞれ0.609, 0.394, 0.692であった。各問題の成績よりも、3問の成績の平均の方がバグ検出率との相関が高かった点が興味深い。レビューの質を評価するためには、1種類の問題だけでは不十分であり、複数の種類の問題を使用する必要があることが伺える。

5. 考察

5.1 試験の成績とバグの種類

バグの種類と試験問題の種類との関係について考察する。4.3節でも述べたように、問1～3の成績の平均とバグ検出率とは相関が高かったが、各試験問題の成績とバグ検出率とはやや相関が低かった。この結果から、各試験問題の成績は、特定の種類のバグの検出率にのみ強く関連していることが推察される。そこで、我々は、コードレビュープロセスの単純なモデルを考え、そのモデルに基づいて、バグの種類と試験問題の種類との関係を説明することを試みた。

考案したコードレビュープロセスのモデルを図2に示す。このモデルでは、レビュー参加者の作業を次の3種類に分類している。まず、レビュー参加者は、仕様書を読んで理解する。つまり、仕様書に書かれている情報を基に、プログラムの抽象的な動作に関する概念を頭の中に形成する。また、レビュー参加者は、ソースコードを読んで理解する。つまり、ソースコードに書かれている情報を基に、プログラムの具体的な動作に関する概念を頭の中に形成する。そして、両種類の概念を比較することにより、バグを検出する。つまり、両種類の概念を基に推論を行うことにより矛盾を導く。

ただし、頭の中に形成される概念と仕様書に書かれている情報とは、一対一に対応する訳ではない。頭の中に取り込まれた情報は、仕様書のように、文字や図表で表されているとは限らないし、章や節に分かれているとも限らない。また、仕様書を読み漏らしていたり深く読んでいなかったために、仕様書に書かれていても頭の中には形

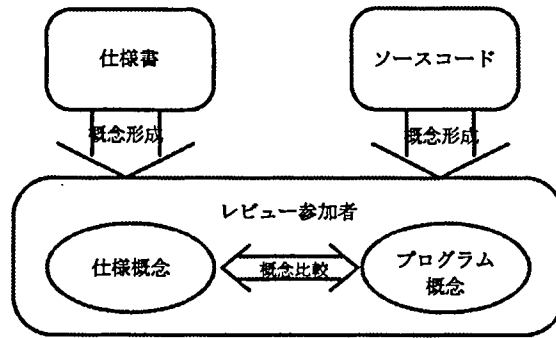


図2 コードレビュープロセスのモデル

成されていない概念があるかもしれない。思い違いをしたために、仕様書には書かれていないが頭の中に形成されている概念があるかもしれない。ソースコードについても、仕様書と同様に、頭の中に形成される概念と書かれている情報とが一對一に対応する訳ではない。

このモデルにおいては、バグが次の2種類の作業において発見される。一つは、仕様書とソースコードとを理解した結果を比較する作業である。もう一つは、ソースコードを理解する作業である。前者の作業では、仕様書とソースコードとから形成した概念の間に矛盾が導かれると、バグが検出される。例えば、ある入力を与えた時の出力が、仕様書とソースコードとの間で一致していない場合である。後者の作業では、ソースコードからプログラムの動作に関する概念を形成する際に、ソースコードの情報の中で矛盾が導かれると、バグが検出される。例えば、変数を初期化したり値を書き込む前に参照する恐れがある場合や、ある配列の添字が配列の大きさを越える恐れがある場合である。なお、このモデルでは、仕様書の中に矛盾が含まれている場合を考えていないので、仕様書からプログラムの動作に関する概念を形成する際には、バグが検出されない。

このモデルに基づけば、コードレビューにおいて発見される可能性のあるバグは、次の2種類に分類できる。

種類1 仕様書を見ていなくてもソースコードを理解する時に検出されるバグ

種類2 仕様とソースコードを比較する時に検出されるバグ

4章で述べた実験で検出されたバグの中で、種類1, 2のバグの個数を表3に示す。

なお、このモデルに基づけば、バグを見逃す場合には、次の3種類の原因が考えられる。

原因1 仕様書を正しく理解していなかった。つまり、仕様書に書かれていた情報と頭の中に形成された概念とが一致していなかった。

原因2 ソースコードを正しく理解していなかった。つまり、ソースコードに書かれていた情報と頭の中に形成された概念とが一致していなかった。

原因3 仕様書とソースコードとを比較していなかった。つまり、頭の中での推論の不足により矛盾を見逃した。

表3：実験で検出されたバグの分類

被験者	種類1	種類2	合計
A	8	3	11
B	2	4	6
C	6	0	6
D	2	4	6
E	4	1	5
F	3	1	4
G	2	1	3
H	0	3	3

Bug Detection Rate

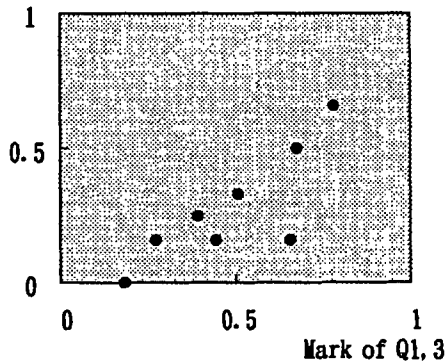


図3 問1, 3の成績と種類1のバグの検出率

これらの中で、種類1のバグを見逃す原因は、原因2だけである。種類2のバグは、原因1～3のどの原因によっても見逃す恐れがある。

以上の議論より、バグの種類と試験問題の種類との関係は、以下のように導かれる。まず、種類1のバグは原因2によってだけ見逃す恐れがあるので、種類1のバグの検出率は、ソースコードの理解の程度を試す問1, 3の成績に強い関連があると思われる。逆に、仕様の理解の程度を試す問2との成績には、全く関連がないと思われる。そして、種類2のバグは原因1～3のどの原因によっても見逃す恐れがあるので、問1～3のどの成績にも関連があると思われる。4章で述べた実験について、問1, 3の成績の平均と種類1のバグの検出率との相関を図3に示す。相関係数は、0.818と高い値であった。問2の成績と種類1のバグの検出率との相関係数は、0.246であった。これらの数値は、我々の推察と一致した傾向を示している。

5.2 試験成績のプログラムへの非依存性

1章でも述べたように、我々は4.1節で述べたプログラムと別のプログラムに対して同様の実験を行っている¹⁴⁾。そこで、試験の成績とバグ検出率とについて、今回

Bug Detection Rate

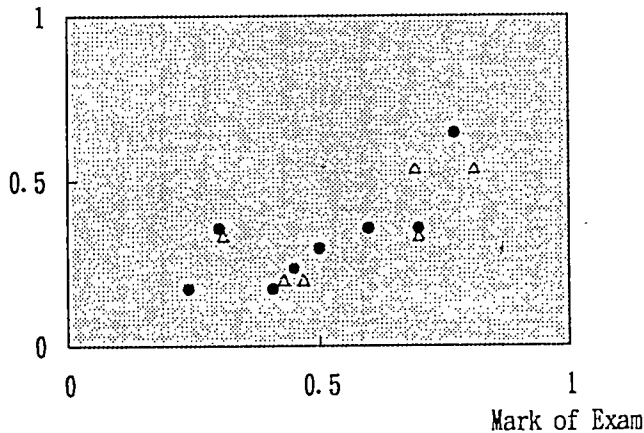


図4 二つの実験の試験成績とバグ検出率

の実験での相関と前回の実験での相関とを比較する。

前回と今回の実験では、プログラムの仕様もソースコードも異なるが、プログラムの規模はほぼ同じであった。また、問1～3の3種類の問題を使用したことも同じであった。主な相違は次の通りである。第1に、レビューの方式が異なっていた。今回の実験では、各被験者が単独でコードレビューを行った。前回の実験では、コード作成者の役割を演じる者が、被験者にコードの説明を口頭で行った。第2に、試験の実施方法が異なる。今回の実験では、試験の際に問題以外の資料を被験者が参照できなかった。前回の実験では、ソースコードや仕様書を読みながら回答することが許されていた。前回の試験方法では、仮に被験者がコードレビューの際にコードや仕様の細部を理解していなくても、試験の際に時間をかければ高い成績が得られる恐れがあった。今回の試験方法では、その点が改善されている。

前回と今回の両方の実験について、問1～3の成績の平均とバグ検出率との相関を図4に同時に示す。前回の実験の結果は“△”で、今回の実験の結果は“●”で表されている。今回と比較すれば前回の実験での相関係数はわずかに低いが、両方の実験で同様の傾向が現れていることがわかる。両実験を合わせたのべ14名の被験者についての相関係数は0.738であった。この結果は、対象とするプログラムに依らず、試験の成績がバグ検出率をよく表すことを示唆している。

6. おわりに

実験の結果より、提案した試験の成績とバグ検出率との間に相関があることがわかった。対象とするプログラムが異なる実験においても、同様の相関関係が見られた。また、コードの理解の程度を試す問題の成績は、特定の種類のバグの検出率に特に関連していることがわかった。

今後の課題としては、以下の点が挙げられる。

- これまでの実験では、レビューの方法を被験者の自由に任せていたため、場当たりの方法でレビューが行われていた。より系統的な方法を採用した場合にも提案した試験方法が有効であるかを、実験により調べる必要がある。
- これまでの実験では、各被験者が単独でレビューを行っていたが、一般には、複数のレビュー参加者が共同でレビューを行う。共同でレビューが行われる場合の試験方法について検討が必要である。

- 参考文献 [1] B.W. Boehm, "Software engineering", IEEE Trans. Computers, pp. 1226 ~ 1241, 1976.
- [2] M.E. Fagan, "Design and code inspection to reduce errors in program development", IBM Systems J., 15, 3, pp.182 ~ 211, 1976.
- [3] C.P. Hollocker, "Software Reviews and Audits Handbook", John Wiley & Sons, 1990.
- [4] 的場政明, 高田義広, 鳥居宏次, "ソフトウェアレビューの質を参加者へのテストにより評価する試み", 情処全大, pp.313 ~ 314, 1994 秋.
- [5] B.B David and R.Lyle James, "A two-person inspection method to improve programming productivity", IEEE Trans. Soft. Eng., 15, 10, pp. 1294 ~ 1304, 1989.

執筆者紹介 的場政明 (Masaaki Matoba)

昭和62年日本大学文理学部応用数学科卒。同年日本ユニシス(株)入社。証券会社のオンラインシステム開発に従事。平成7年奈良先端科学技術大学院大学情報科学研究科修士課程了。現在、証券システム部所属。情報処理学会会員。



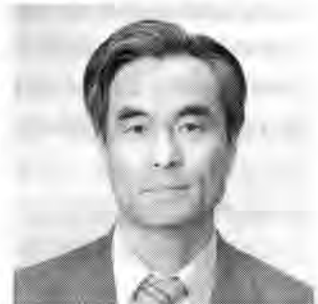
高田義広 (Yoshihiro Takada)

平成元年大阪大学基礎工学部情報工学科卒。平成6年同大学院博士課程了。同年奈良先端科学技術大学院大学・助手。工学博士。ソフトウェアの信頼性予測、プログラマの能力評価の研究に従事。電子情報通信学会、情報処理学会、日本音響学会各会員。



鳥居宏次 (Koji Torii)

昭和37年大阪大学工学部通信工学科卒。42年同大学院博士課程了。同年電気試験所(現電子技術総合研究所)入所。50年ソフトウェア部言語処理研究室室長。59年大阪大学基礎工学部情報工学科教授。平成4年奈良先端科学技術大学院大学・教授。工学博士。ソフトウェア工学の研究に従事。電子情報通信学会、情報処理学会、日本ソフトウェア学会、人工知能学会、ACM、IEEE各会員。



Ferguson 曲線の接線長決定の一手法

A Method for Determining the Tangent Lengths of Ferguson Curves

清水保弘, 宮地恵美

要約 CADの幾何モデラにおいて自由曲線を生成するには、いくつかの方法があるが、本稿では、通過点列と各通過点での曲線の接線方向を指定して、区分的3次多項式曲線を生成する方法を取り上げる。隣接通過点間を補間する3次多項式曲線（セグメント）は、Ferguson曲線とよばれる。一つのFerguson曲線は、セグメント始終点とそこでの接線ベクトルを指定すると決まるが、今の場合、始終点での接線ベクトルの長さ（接線長）の自由度を決定する必要がある。

Ferguson曲線の接線長として、従来から経験的に用いられてきた方法は、接線長をセグメント始終点の弦長としたり、接円弧長とする方法であるが、これらの理論的根拠は明かではなかった。実際、従来法では形状的に異様な盛り上がりを持つセグメントを生成する場合があった。

そこで、本稿では、線分・円弧を基本形状とし、これらのアフィン変換や1次ブレンドとして接線長を決定していくアプローチを提示し、与えられた条件を満たす高品質形状が得られることを示す。とくに、円弧形状セグメントの場合には、円弧と弦が囲む面積を最良近似する「等面積法」とよぶ新手法による接線長決定法を提案する。

Abstract Known are several methods of creating free-form curves for the geometric modeler of a CAD system. This paper addresses a method of creating a piecewise cubic polynomial curve by specifying both the string of points it passes through and the tangent directions of a curve at an individual point. Each cubic polynomial curve interpolating adjacent points is called a segment, which is represented by a Ferguson curve. Each Ferguson curve can be determined through the definition of coordinates and tangent vectors at the end points, and this instance requires the lengths of those tangent vectors (tangent lengths) to be determined in terms of their degree of freedom.

The traditional, experience-led method handled the tangent length of a Ferguson curve as a chord length or a circular arc length, but its theoretical background has not been made clear. In fact, the conventional method could generate a segment with an abnormal swell in its shape.

That is the reason this paper is intended to present an approach, with line segments and circular arcs being basic shapes, by which to determine the tangent lengths of Ferguson curves as their affine transformations and linear blendings; thus making available high-quality shapes that satisfy the given conditions. For circular arc segments especially, this paper proposes a new approach called the "equi-area method" which optimally approximates the areas enclosed with arcs and chords.

1. はじめに

広範な対象を扱うCADシステムの幾何モデラにとって、自由曲線・自由曲面形状を取り扱うことは必須の要件である。ここで、自由曲線・自由曲面とは、直線・円・楕円・平面・円柱面・円錐面・球面などの解析的な形状では表現できない曲線・曲面

形状の総称である。

多くの幾何モデラでは、曲線・曲面を表すのにパラメタによる表現（パラメトリック表現）が用いられている。自由曲線・自由曲面をパラメトリック表現で扱う伝統的なアプローチは、区分的3次多項式曲線・区分的双3次多項式曲面として表現するものである。

現在では、NURBS (Non-Uniform Rational B-Spline) 曲線・曲面を用いる新しい表現法が世界的に主流になりつつあるが、NURBSについては図形処理上の未解決問題も多く、3次多項式表現という簡便かつ直観的な表現法に基づいた伝統的アプローチの価値は依然としてあせていない。

自由曲線を創成するには、拘束する条件により、いくつかの方法が提案されている。本稿では、通過点列と各通過点での曲線の接線方向が与えられたときに、通過点列間を補間する区分的3次多項式曲線を求める方法を論ずる。各々の隣接通過点間は一つの3次多項式曲線に対応し、セグメントとよばれる。セグメント端点と端点接線方向ベクトルを補間する3次多項式曲線はFerguson曲線として知られているが、この場合、セグメントの形状は、端点での接線ベクトルの長さ（接線長）の決定法に依存している。従来から経験的に採用されてきた方法は、接線長をセグメント両端点の弦長（直線距離）としたり（弦長法）、両端点を通過し接線に内接する円弧の弧長とするもの（円弧長法）であるが、これらの方法の理論的裏付けは必ずしも明らかにされていなかった。実際に、両端点と接線方向の関係によっては不自然な盛り上がりのある形状ができる場合がある。

そこで本稿では、Ferguson曲線の接線長決定問題について理論的に考察することで、高品質なセグメント形状という観点から一つの指導理念を与えるとともに、従来法の位置づけを明確にし、従来法を越えた新しい方法を提案する。

本稿の構成は次の通りである。

続く2章では、Ferguson曲線をはじめとする、本稿で必要な基礎概念についての準備を行う。

3章では、Ferguson曲線の接線長決定問題について従来法を紹介し、どこに問題があるかを検討する。また、本稿全体で前提とする条件（基本仮定）を明確にし、接線長決定問題へのアプローチを提示する。

4章では、線分・円弧形状を基準とするケース（基本ケースとよぶ）を扱う。この場合の方針は、これらの形状を最も良く近似するFerguson曲線の接線長を決定することである。線分ケースは従来からよく知られた弦長法が最も良い接線長を与える。円弧ケースでは、円弧への近似度を測る「面積指標」とよぶ指標を新たに導入することで、「等面積法」とよぶ新手法による接線長の決め方が最適であることを主張する。

5章では、より一般のケースを扱う。ここでの方針は、楕円弧、平面S字曲線、空間曲線のケースをアフィン変換と1次ブレンド法により基本ケースに帰着させることである。本章で述べられる接線長決定法が、従来法に比べて、より適切な接線長を与えていることを実務データに適用して検証する。とくに、楕円弧のケースは、楕円弧を基準とする近似曲線の生成法を与える。

6章では、「基本仮定を満たさない場合」への展望、「複数セグメントで発生する拘

束条件がある場合」の問題を今後の課題として述べる。

7章では、本稿で得られた結果のまとめを行う。

2. 準備

2.1 Ferguson 曲線

t を 0 から 1 までの変域を持つパラメタとする。 t に関する次の四つの 3 次多項式を 3 次 Hermite 補間関数とよぶ (参考文献^[1]p. 82)。

$$\left. \begin{aligned} H_{0,0}(t) &= (t-1)^2(2t+1), & H_{0,1}(t) &= t^2(3-2t), \\ H_{1,0}(t) &= (t-1)^2t, & H_{1,1}(t) &= (t-1)t^2 \end{aligned} \right\} \quad (2-1)$$

$t=0$ で始点 Q_0 , $t=1$ で終点 Q_1 を通り、始点での (長さ込みの) 接線ベクトル T_0 , 終点での (長さ込みの) 接線ベクトル T_1 を持つ 3 次多項式曲線 $C(t)$ は 3 次 Hermite 補間関数を用いて次のように表現される。

$$C(t) = H_{0,0}(t)Q_0 + H_{0,1}(t)Q_1 + H_{1,0}(t)T_0 + H_{1,1}(t)T_1 \quad (2-2)$$

このように表現された 1 セグメント 3 次多項式曲線を Ferguson 曲線 とよぶ (図 1)。

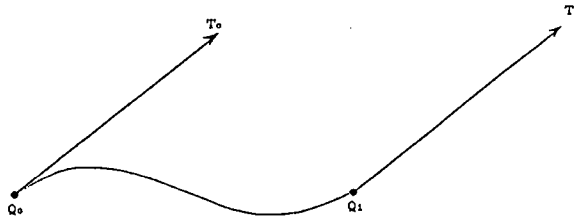


図 1 Ferguson 曲線

2.2 Bezier 曲線

上と同様に t を 0 から 1 までの変域を持つパラメタとする。

n を多項式の次数, i を $0 \leq i \leq n$ の範囲の整数とすると, t に関する次の $(n+1)$ 個の n 次多項式を n 次 Bernstein 基底関数 とよぶ (参考文献^[2]p. 15)。

$$B_{i,n}(t) = {}_n C_i (1-t)^{n-i} t^i \quad (2-3)$$

ここで ${}_n C_i$ は, $!$ を階乗の記号とすると

$${}_n C_i = n! / (i!(n-i)!) \quad (2-4)$$

であり, 二項係数とよばれる。

P_0, P_1, \dots, P_n を $(n+1)$ 個の点列とすると, Bernstein 基底関数を係数とする 1 次結合で表現される 1 セグメント n 次多項式曲線 $C(t)$ を n 次 Bezier 曲線 とよぶ。

$$C(t) = B_{0,n}(t)P_0 + B_{1,n}(t)P_1 + \dots + B_{n,n}(t)P_n \quad (2-5)$$

$(n+1)$ 個の点 P_0, P_1, \dots, P_n を Bezier 曲線の 制御点 (control points) とよぶ。制御点のうち, P_0, P_n は各々 $t=0, t=1$ での曲線の通過点であり, セグメントの始点・終点に対応する。その他の制御点は一般に曲線上にあるとは限らないが, 曲線を 1 次近似した概形を表しており, これらの点を動かして曲線形状を制御できる。制御点列を $P_0 \rightarrow P_1 \rightarrow \dots \rightarrow P_n \rightarrow P_0$ の順に結んで得られる $(n+1)$ 角形を 制御多角形

(control polygon) とよぶ。

1次Bezier曲線

$$C(t) = (1-t)P_0 + tP_1 \tag{2-6}$$

は一般に線分を表現している。これを3次Bezier曲線に次数上げて表現すると

$$C(t) = (1-t)^3P_0 + 3(1-t)^2t((2/3)P_0 + (1/3)P_1) + 3(1-t)t^2((1/3)P_0 + (2/3)P_1) + t^3P_1 \tag{2-7}$$

となる。3次Bezier曲線(2-7)の中間制御点は、線分の3等分点になる。

2次Bezier曲線

$$C(t) = (1-t)^2P_0 + 2(1-t)tP_1 + t^2P_2 \tag{2-8}$$

は一般に放物線弧を表現している。これを3次Bezier曲線に次数上げて表現すると

$$C(t) = (1-t)^3P_0 + 3(1-t)^2t((1/3)P_0 + (2/3)P_1) + 3(1-t)t^2((2/3)P_1 + (1/3)P_2) + t^3P_2 \tag{2-9}$$

となる。3次Bezier曲線(2-9)の中間制御点は、2次Bezier曲線の制御三角形 $P_0 P_1 P_2$ の2辺 $P_0 P_1$, $P_2 P_1$ を各々 2 : 1 の比に内分する点になる (図2)。

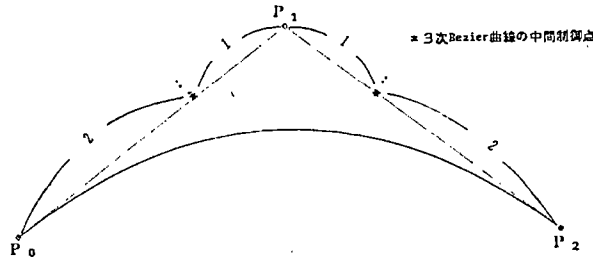


図2 2次Bezier曲線

2.3 Ferguson 曲線と3次Bezier曲線の関係

3次Bezier曲線

$$C(t) = (1-t)^3P_0 + 3(1-t)^2tP_1 + 3(1-t)t^2P_2 + t^3P_3 \tag{2-10}$$

と Ferguson 曲線(2-2)はともに3次なので相互に次の関係がある (図3)。

$$\left. \begin{aligned} Q_0 &= P_0, & Q_1 &= P_3, \\ T_0 &= 3(P_1 - P_0), & T_1 &= 3(P_3 - P_2), \\ P_1 &= Q_0 + (1/3)T_0, & P_2 &= Q_1 - (1/3)T_1 \end{aligned} \right\} \tag{2-11}$$

2.4 Bezier曲線のアフィン不変性

線形変換 M とベクトル D による平行移動の組み合わせで得られる変換 A

$$A(V) = M(V) + D \tag{2-12}$$

をアフィン変換とよぶ (ここで、 V は任意の位置ベクトル)。

アフィン変換は、直線を直線に写すこと、および直線上の点列の比を保つという性質がある。

A をアフィン変換、 $C(t)$ をBezier曲線とすると、曲線をアフィン変換すると、各制御点をアフィン変換した曲線に一致するという性質

$$A(C(t)) = B_{0,n}(t)A(P_0) + B_{1,n}(t)A(P_1) + \dots + B_{n,n}(t)A(P_n) \tag{2-13}$$

がある。これをBezier曲線のアフィン不変性という。

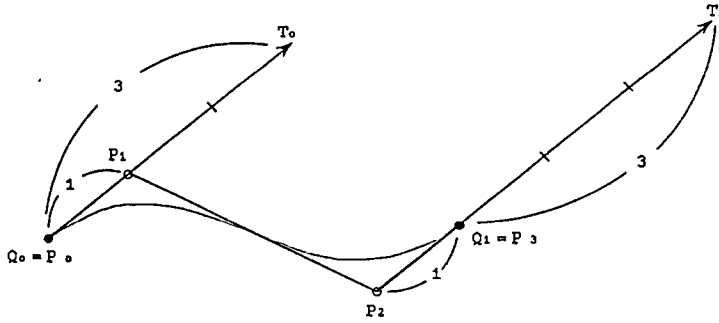


図3 Ferguson 曲線と 3 次 Bezier 曲線

この性質は Ferguson 曲線表現に対しては成り立たないことに注意する。

2.5 2 曲線の 1 次ブレンド曲線

$C_0(t)$, $C_1(t)$ を同じパラメタ t (変域は $0 \leq t \leq 1$) で表現された 2 曲線とする。このとき、 t の 1 次式 $(1-t)$, t の重みでブレンド (混合) した曲線

$$C(t) = (1-t)C_0(t) + tC_1(t) \tag{2-14}$$

を $(C_0(t)$ と $C_1(t)$ の) 1 次ブレンド曲線とよぶ。

1 次ブレンド曲線は次の性質を持っている。

$$C(0) = C_0(0), \quad C(1) = C_1(1) \tag{2-15}$$

さらに、 $C_0(0) = C_1(0)$ かつ $C_0(1) = C_1(1)$ (すなわち 2 曲線が端点を共有する) なら

$$(dC/dt)(0) = (dC_0/dt)(0), \quad (dC/dt)(1) = (dC_1/dt)(1) \tag{2-16}$$

ここで、 $(dC/dt)(t)$ は曲線 $C(t)$ のパラメタ t 位置での 1 階微分値 (接線ベクトル) である。

性質 (2-15), (2-16) により、端点を共有する 2 曲線 $C_0(t)$, $C_1(t)$ の 1 次ブレンド曲線は、始点付近で曲線 $C_0(t)$ の傾向を持ち、終点付近で曲線 $C_1(t)$ の傾向を持つ最も簡単な曲線であるといえる。

$C_0(t)$, $C_1(t)$ が 3 次多項式曲線なら、それらの 1 次ブレンド曲線は一般に 4 次 (以下の) 多項式曲線になる。ところが、2 曲線がある条件を満たすと 1 次ブレンド曲線が 3 次多項式曲線になる場合がある。次の補題が成り立つ。

補題 A

$$C_0(t) = H_{0,0}(t)Q_0 + H_{0,1}(t)Q_1 + H_{1,0}(t)T_0 + H_{1,1}(t)U_1$$

$$C_1(t) = H_{0,0}(t)Q_0 + H_{0,1}(t)Q_1 + H_{1,0}(t)U_0 + H_{1,1}(t)T_1$$

$$C_2(t) = H_{0,0}(t)Q_0 + H_{0,1}(t)Q_1 + H_{1,0}(t)T_0 + H_{1,1}(t)T_1$$

を端点を共有する三つの Ferguson 曲線とする (接線ベクトルの選び方に注意)。このとき、 $C_0(t)$, $C_1(t)$ の 1 次ブレンド曲線 $C(t)$ と曲線 $C_2(t)$ との誤差は

$$C(t) - C_2(t) = (1-t)^2 t^2 ((U_0 - T_0) - (U_1 - T_1))$$

で与えられる。したがって、 $(U_0 - T_0) - (U_1 - T_1) = 0$ なら、1 次ブレンド曲線は Ferguson 曲線 $C_2(t)$ になる

補題 A の証明は、付録 1 で与える。

3. Ferguson 曲線の接線長決定問題

3.1 従来法によるアプローチとその問題点

自由曲線を扱う幾何モデラが、その内部で解かなければならない典型的問題の一つは、与えられた通過点列を通り、各点で指定された接線方向を持つ自由曲線の決定である(図4)。これは、隣接する通過点間をセグメントとすると、各セグメントの形状決定に帰着する。3次多項式表現を用いる場合には、Ferguson 曲線セグメント(2-2)の接線長 $|T_0|$ 、 $|T_1|$ を如何に決定するかが問題となる。

従来は経験的に次の方法が採られていた。

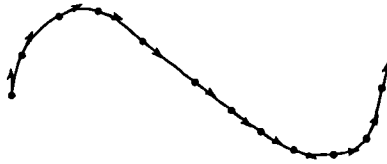


図4 通過点列と接点方向を与えた自由曲線決定

1) 弦長法(図5)

始終点 Q_0 、 Q_1 を結ぶ線分(これを弦とよぶ)の長さを l とするとき

$$|T_0| = |T_1| = l$$

とする方法。

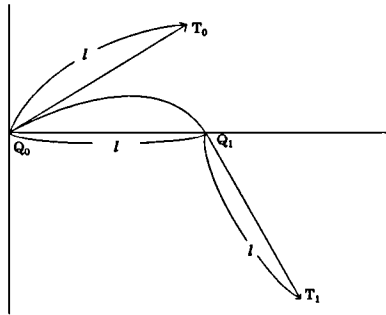


図5 弦長法による接線決定

2) 円弧長法(図6)

始終点 Q_0 、 Q_1 を通り、始点接ベクトル T_0 に接する円弧の弧長を l_0 、終点接ベクトル T_1 に接する円弧の弧長を l_1 とするとき

$$|T_0| = l_0, \quad |T_1| = l_1$$

とする方法。

3) 平均円弧長法(図7)

2)の平均円弧長を $(l_0 + l_1)/2$ とするとき

$$|T_0| = |T_1| = (l_0 + l_1)/2$$

とする方法。

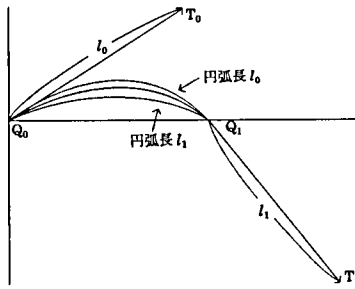


図6 円弧長法による接線長決定

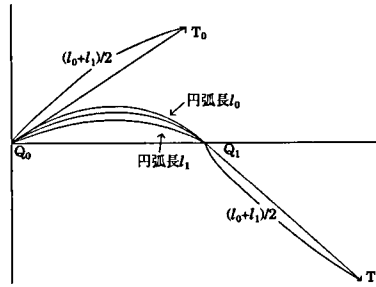


図7 平均円弧長法による接線長決定

これらの方法は理論的裏付けなしに用いられてきた。実際、これらの方法で作られた Ferguson 曲線セグメントの形状には「良い形状」とは言い難い場合が存在する。例えば、図8は、Ferguson 曲線の接線長を弦長法により決定して得られた1セグメント形状である。曲線の中央部に“不自然な”盛り上がりがある。曲線形状の評価によく用いられる曲率ベクトル分布を見ると、乱れていることがわかる。この原因は接線長が通過点と接線方向の位置関係にそぐわないためである。

1), 2), 3) の他に、教科書^[3]では、“経験的に” 接線長を弦長 l の 1.2 倍にとる方法なども提案されているが、理論的背景を欠いていることには変わりない。

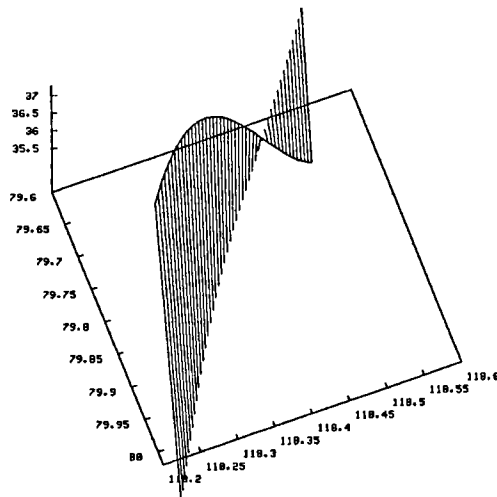


図8 従来法による曲率ベクトル分布

3.2 本稿でのアプローチの提示

本稿では、Ferguson 曲線の接線長決定問題を「基本ケース」と「一般ケース」に分けて理論的に考察する。基本ケースでは、「高品質」と判断できる基準形状を提示し、これを最も良く近似するように接線長を決定した Ferguson 曲線が「良い形状」であると考えます。また、一般ケースはアフィン変換と1次ブレンドにより基本ケースからの変形として理解できることを示す。

これらのケースでの接線長決定法は続く4章と5章で詳しく論ずるが、本節では、理論を複雑にしないために、本稿全体を通じて仮定する条件を述べる。

本稿で対象とする Ferguson 曲線は、次の仮定を満たすものとする。

仮定1： 弦長 $|Q_0Q_1|$ は0でない。

仮定2： 接線ベクトルの長さ $|T_0|$ 、 $|T_1|$ はいずれも0でない。

仮定3： 接線ベクトル T_0 、 T_1 と弦 Q_0Q_1 とのなす角度（ラジアン）をそれぞれ θ_0 、 θ_1 とするとき、

$$0 \leq \theta_0 < \pi/2, \quad 0 \leq \theta_1 < \pi/2$$

である。

これらの三つの仮定を本稿では基本仮定として引用する。仮定の根拠は次の通りである。

仮定1は、1セグメントの閉曲線（始終点一致する曲線）を扱わない表明である。

仮定2は、始終点が特異点（接線ベクトルが退化する点）にならないための要請である。

仮定3は、曲線が始点から終点に向かう弦の方向に沿っているとする要請である（図9）。n個の通過点列の補間という Ferguson 曲線セグメントの本来の使用場面を

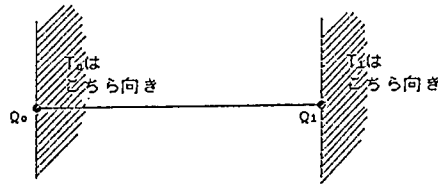


図9 仮定3の意味

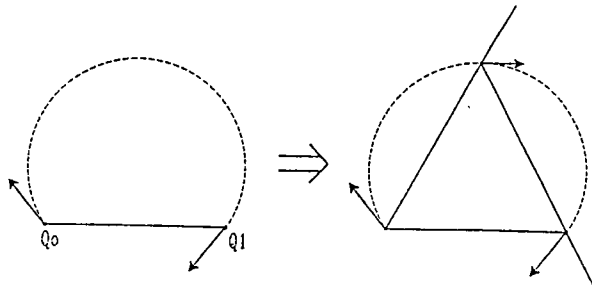


図10 仮定3を満たさない場合

想定したとき、仮定3を満たさない場合は、通過点列の与え方が粗すぎるはずである(図10)。したがって、必要なら通過点と接線方向を補って各セグメントで仮定3を満たすように要請することは強い制限を与えないと考える。

4. 基本ケースの接線長決定法

4.1 基本ケースと基準形状の定義

「良い形状」と判断できる曲線形状(これを基準形状とよぶ)を比較的容易に設定できるケースがある。それは次の2ケースである。

1) 線分ケース(図11)

$\theta_0 = \theta_1 = 0$ の場合である。この場合の基準形状は、弦 Q_0Q_1 と一致する線分である。

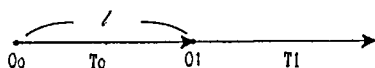


図11 線分ケース

2) 円弧ケース(図12)

次の3条件を満たす場合である。

- ① 弦 Q_0Q_1 と接線ベクトル T_0, T_1 が同一平面上にある。
- ② 接線ベクトル T_0, T_1 が弦 Q_0Q_1 に関して逆側を向く。
- ③ $\theta_0 = \theta_1 = \theta$ ($0 < \theta < \pi/2$)。

この場合は、 Q_0 を始点、 Q_1 を終点、 T_0, T_1 に接する円弧が一意に存在する。この円弧を基準形状とする。

以上の2ケースを総称して、基本ケースとよぶ。基本ケースでの方針は、各々の基準形状を最も良く近似する接線長を持つFerguson曲線を決定することである。ここで得られるFerguson曲線は、いずれも平面曲線であり、弦 Q_0Q_1 の垂直2等分線に関して左右対称であることを注意しておく。

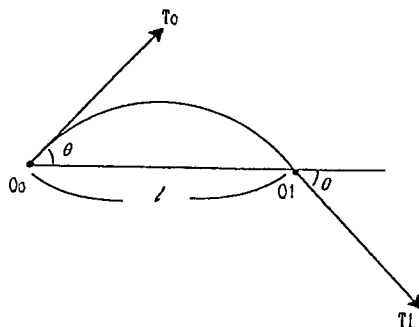


図12 円弧ケース

4.2 線分ケースの解法

線分は Ferguson 曲線により近似なしに表現できる。線分の標準的定義式（1次 Bezier 曲線表現）は、

$$C(t) = (1-t)Q_0 + tQ_1 \tag{4-1}$$

であるが、式(4-1)を Ferguson 曲線の定義式(2-2)にあてはめると、

$$T_0 = T_1 = Q_1 - Q_0$$

の場合に相当している。この結果、線分ケースの接線長の決め方は、弦長 $|Q_0Q_1|$ を l とするとき、

$$|T_0| = |T_1| = l$$

として接線長を決める弦長法になる（これが線分ケース以外にも弦長法を用いる従来の根拠になっている）。また、(4-1)から到る所で接ベクトル長が一定値になること（等速性）もわかる。線分ケースでは、原則として、この決め方を採用する。

ただし、形状が線分になることだけを重視し、等速性を断念するなら、 w を $0 < w < 1$ の範囲の任意定数とするとき、

$$|T_0| = |T_1| = wl$$

として接線長を決めても構わない。この不定性（自由度）は、5.3.2節で平面 S 字曲線ケースでの接線長決定に利用する。

4.3 円弧ケースの解法

4.3.1 等比変形

弦長 $|Q_0Q_1|$ を l とする。 Q_0 から始点接線ベクトル T_0 方向に延びた半直線と、 Q_1 から終点接線ベクトル T_1 の逆方向（ $-T_1$ 方向）に延びる半直線とは交点を持つので、交点を Q_m とする。三角形 $Q_0Q_mQ_1$ は底角 θ の二等辺三角形になる。等辺の長さは

$$|Q_0Q_m| = |Q_1Q_m| = l / (2\cos\theta) \tag{4-2}$$

である（図13）。

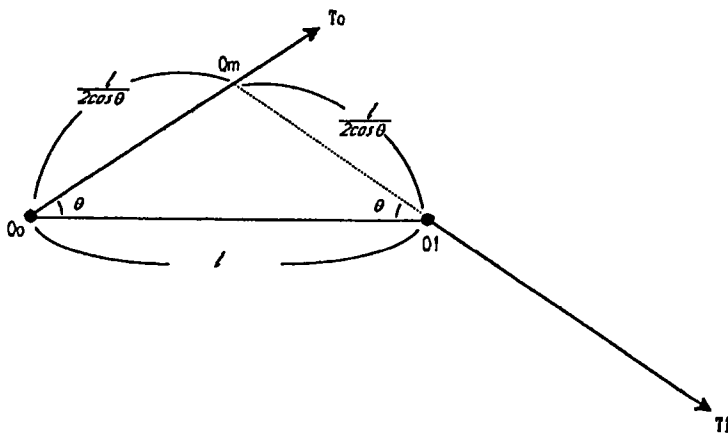


図13 円弧ケースでの二等辺三角形

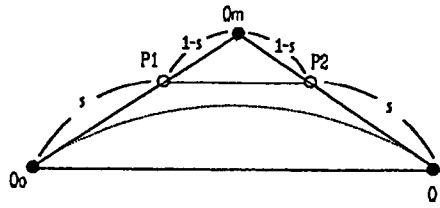


図 14 等比変形

ここで、三角形 $Q_0Q_mQ_1$ を二次Bezier曲線の制御三角形と考えてみよう。(2-8)から2次Bezier曲線 $C(t)$ は

$$C(t) = (1-t)^2Q_0 + 2(1-t)tQ_m + t^2Q_1 \tag{4-3}$$

となる。これは、制御三角形に内接する放物線弧を与えている。(2-9)より3次Bezier曲線に次数上げたときの中間制御点 P_1, P_2 は、辺 Q_0Q_m, Q_1Q_m を各々2:1の比に内分する点である。この内分比を Q_0, Q_1 に近づくように変化させて中間制御点を動かすと、3次Bezier曲線の形状は放物線弧から次第に円弧に近づいていく。すなわち、比 s を $0 < s \leq 2/3$ の範囲で動かして、3次Bezier曲線の変形族

$$\left. \begin{aligned} C(t) &= (1-t)^3Q_0 + 3(1-t)^2tP_1 + 3(1-t)t^2P_2 + t^3Q_1 \\ P_1 &= (1-s)Q_0 + sQ_m \\ P_2 &= (1-s)Q_1 + sQ_m \end{aligned} \right\} \tag{4-4}$$

を考える(図14)。中間制御点 P_1, P_2 は辺 Q_0Q_m, Q_1Q_m 上を動かしているもので、変形の過程で始終点での接線ベクトルの方向は不変であることに注意する。また、4.1節で注意した対称性から、比 s は辺 Q_0Q_m, Q_1Q_m で等しくとる。

この意味で比 s を等比、(4-4)による3次Bezier曲線の変形を等比変形とよぶ。

一般に、等比 s により等比変形した3次Bezier曲線をFerguson曲線表現すると、接線長は(2-11)と(4-2)より

$$|T_0| = |T_1| = 3sl / (2\cos \theta) \tag{4-5}$$

である。とくに放物線弧の場合は、 $s = 2/3$ より

$$|T_0| = |T_1| = l / (\cos \theta) \tag{4-6}$$

となる。求める曲線形状を放物線弧と見なして(4-6)により接線長を決める方法在本稿では「放物線法」とよび、従来法の一つと見なす。

4.3.2 円弧中点通過法

Ferguson曲線 $C(t)$ による円弧の近似に関して教科書(例えば文献^[11])に書かれている方法は、パラメタ $t = 1/2$ の点 $C(1/2)$ が円弧の中点を通過するように接線長を決める方法である。これを本稿では「円弧中点通過法」とよぶ(図15)。この方法で定まる接線長は

$$|T_0| = |T_1| = 2l / (1 + \cos \theta) \tag{4-7}$$

となる。

これを等比変形の観点から考えると、(4-5)と(4-7)から等比 s は

$$s = (4\cos \theta) / (3(1 + \cos \theta)) \tag{4-8}$$

となる。

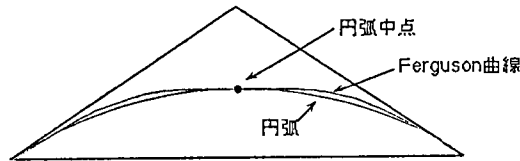


図15 円弧中点通過法

4.3.3 面積指標

従来の円弧近似の尺度は近似曲線と円弧との最大離れ距離に注目していたが、曲線全体を考慮する大域的な観点からは、曲線と弦で囲まれる領域の面積を円弧と弦で囲まれる領域の面積に最も近くするという尺度も検討するに値すると考える (図16)。

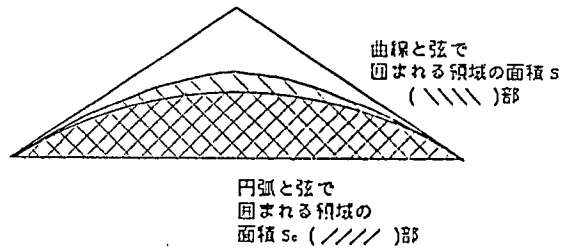


図16 面積指標 I=S/Sc

そこで、両面積の比をとって、次の指標Iを定義する。

$$I = (\text{曲線と弦で囲まれる領域の面積}) / (\text{円弧と弦で囲まれる領域の面積})$$

本稿では、これを面積指標とよぶ。Iが1に近いほど円弧を良く近似していると考ええる。

3次Bezier曲線の等比変形(4-4)について各々の面積指標Iを計算し、等比sと角度θ (ラジアン) の関数として表した公式が次である (証明は付録2で与える)。

定理B (円弧ケースでの面積指標公式)
 3次Bezier曲線の等比変形(4-4)の面積指標Iは

$$I = (3/10)s(4-s)(\tan \theta \sin^2 \theta) / (\theta - \sin \theta \cos \theta)$$
 で与えられる

円弧ケースについて、従来法によりFerguson曲線の接線長を決めた場合、対応する3次Bezier曲線を等比変形と考え、その等比と面積指標を求めると次のようになる。

1) 放物線法

等比: $s = 2/3$

面積指標: $I = (2/3)(\tan \theta \sin^2 \theta) / (\theta - \sin \theta \cos \theta)$

2) 弦長法

等比: $s = (2/3)\cos \theta$

面積指標: $I = (2/15)(6 - \cos \theta)(\sin^3 \theta) / (\theta - \sin \theta \cos \theta)$

3) 円弧長法

$$\text{等比: } s = (2/3)(\theta \cos \theta) / (\sin \theta)$$

$$\text{面積指標: } I = (2/15)(\theta \sin \theta)(6\sin \theta - \theta \cos \theta) / (\theta - \sin \theta \cos \theta)$$

4) 円弧中点通過法

$$\text{等比: } s = (4\cos \theta) / (3(1 + \cos \theta))$$

$$\text{面積指標: } I = (8/15)(\sin^3 \theta)(3 + 2\cos \theta) / ((1 + \cos \theta)^2(\theta - \sin \theta \cos \theta))$$

このように等比 s と面積指標 I は各手法毎に角度 θ の関数になる。

$0 < \theta < \pi/2$ の範囲での各関数のグラフとサンプリング値の表を図17にまとめる。これを見ると、円弧中点通過法が、どの角度 θ に対しても安定して $I \doteq 1$ (微量だけ1より大きい) を与えており、従来法の中では円弧への近似度が最良であることがわかる。

4.3.4 等面積法

ところが面積指標という新しい観点で考えると、厳密に $I=1$ を与える等比が存在する。

定理C (面積指標を1にする等比の存在性)

3次Bezier曲線の等比変形(4-4)の中に面積指標が1になる等比 s が存在する。これは次で与えられる。

$$s = 2(1 - (1 - f(\theta))^{1/2})$$

ここに、 $f(\theta)$ は θ の関数であり

$$f(\theta) = (5/6)(\theta - \sin \theta \cos \theta) / (\tan \theta \sin^2 \theta)$$

である。また、 $()^{1/2}$ は正の平方根を表す

定理Cの証明は付録3で与える。等比 s を θ の関数としてグラフに表したのが図18である。円弧中点通過法の等比関数のグラフ(図17)に極めて似ていることがわかる。

ここで示された等比から(4-5)で接線長を決める手法を「等面積法」とよぶことにする。等面積法は、面積指標の観点からは、円弧中点通過法を越える最良の円弧近似手法である。

4.3.5 円弧ケースでの各手法の評価

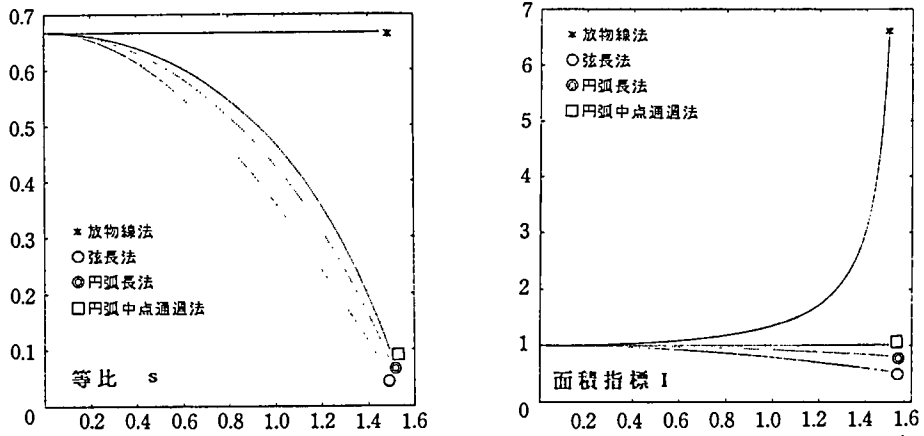
円弧ケースの始終点と接線方向ベクトルを与えて、各手法で得られるFerguson曲線の形状を比較したのが図19である。

円弧中点通過法と等面積法が適切な接線長を与えていることがわかる。この2手法は θ が大きいつきに形状の差が発生し、等面積法の方が円弧形状に近い。ただし、現実によく現れる θ の値(大きくても $\pi/3$ ラジアン=60度ほど)では円弧中点通過法でも充分である。

5. 一般ケースの接線長決定法

5.1 一般ケースの定義

基本仮定を満たし、基本ケースに合致しないケースを「一般ケース」とよぶ。一般



θ		放物線法	弦長法	円弧長法	円弧中点通過法
度	ラジアン				
1	0.017453	1.000061	0.999939	0.999980	1.000000
15	0.261799	1.014131	0.986251	0.995404	1.000009
30	0.523599	1.062250	0.944585	0.981292	1.000142
45	0.785398	1.167959	0.874250	0.956684	1.000771
60	1.047198	1.410040	0.775522	0.919927	1.002695
75	1.308997	2.192043	0.651443	0.868683	1.007534
89	1.553343	24.859588	0.519117	0.805187	1.017551

図17 従来法の等比sと面積指標I

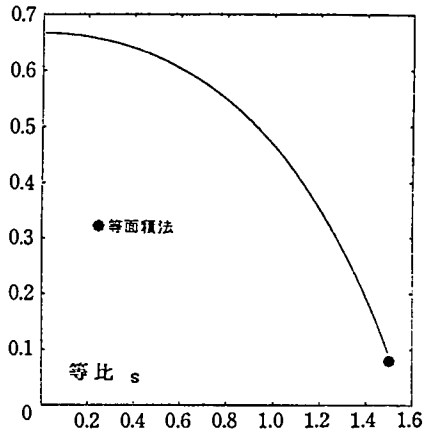


図18 等面積法の等比

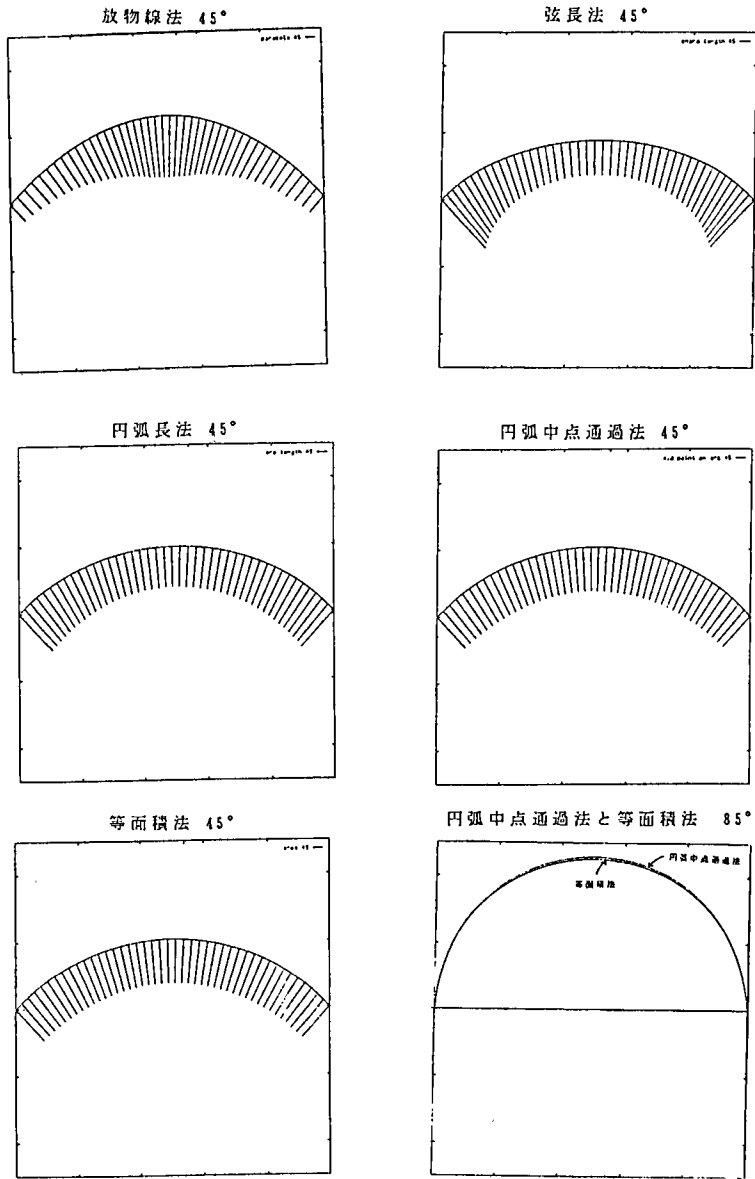


図19 圆弧ケースでの手法の比較

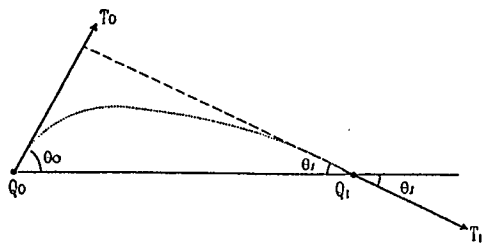


図20 精圆弧ケース

ケースは次の3ケースに分類される。

1) 楕円弧ケース (図20)

次の3条件を満たす場合である。

- ① 弦 Q_0Q_1 と接線ベクトル T_0, T_1 が同一平面上にある。
- ② 接線ベクトル T_0, T_1 が弦 Q_0Q_1 に関して逆側を向く。
- ③ $0 < \theta_0 \neq \theta_1 < \pi/2$ 。

条件①より, Ferguson 曲線は平面曲線である。また③より, 曲線は弦 Q_0Q_1 の垂直二等分線に関して左右非対称である。このケースを「楕円弧ケース」とよぶ理由は 5.2.2 項の議論で明らかになる。

2) 平面S字曲線ケース (図21)

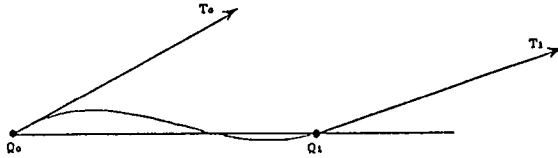


図21 平面S字曲線ケース

次の2条件を満たす場合である。

- ① 弦 Q_0Q_1 と接線ベクトル T_0, T_1 が同一平面上にある。
- ② 接線ベクトル T_0, T_1 が弦 Q_0Q_1 に関して同一側を向く、あるいは、一方の接線ベクトルが弦と平行である。

条件①より, Ferguson 曲線は平面曲線である。また②より, 曲線はS字を引き延ばした形状になる。そこで, このケースを「平面S字曲線ケース」とよぶ。

3) 空間曲線ケース (図22)

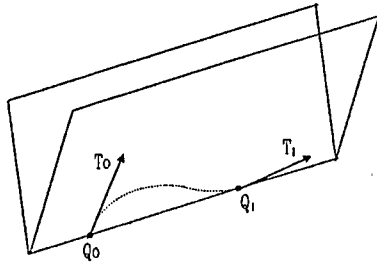


図22 空間曲線ケース

弦 Q_0Q_1 と接線ベクトル T_0, T_1 が同一平面上にないケースである。この場合, 得られる Ferguson 曲線はすべて空間曲線になる。そこで, このケースを「空間曲線ケース」とよぶ。

5.2 楕円弧ケースの解法

5.2.1 円弧ケースへの帰着

楕円弧ケースでは, 条件①②③から, 始点 Q_0 から始点接線ベクトル T_0 方向に伸びる半直線と, 終点 Q_1 から終点接線ベクトル T_1 の逆方向 ($-T_1$ 方向) に伸びる半直線

とが交点を持つのが円弧ケースと共通の特徴である。この交点を Q_m とする。

こうして三角形 $Q_0Q_mQ_1$ が得られるが、条件③のため円弧ケースと異なり二等辺三角形にはならない。ところが、頂点 Q_m を底辺 Q_0Q_1 に平行に移動して Q_m' とし、三角形 $Q_0Q_m'Q_1$ が二等辺三角形（二辺 Q_0Q_m' 、 Q_1Q_m' が等辺）になるようにできる（図23）。

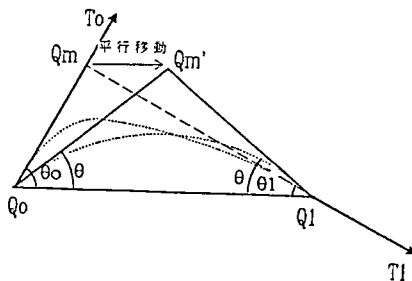


図23 楕円弧ケースの考え方

この変形は平面上のアフィン変換 A により実現される。このアフィン変換 A は

$$A(Q_0) = Q_0, \quad A(Q_1) = Q_1, \quad A(Q_m) = Q_m' \quad (5-1)$$

により特徴付けられる。

二等辺三角形 $Q_0Q_m'Q_1$ の等しい底角の角度を θ とすると、

$$\cot \theta = (\cot \theta_0 + \cot \theta_1) / 2 \quad (5-2)$$

の関係がある。これら二等辺三角形への変形の証明は付録4で与える。

こうして、角度 θ の円弧ケースに帰着されたので、円弧ケースの解法を適用する。等面積法（あるいは円弧中点通過法）により等比 s を決め、円弧を近似する3次Bezier曲線

$$\begin{aligned} C'(t) &= (1-t)^3 Q_0 + 3(1-t)^2 t P_1' + 3(1-t)t^2 P_2' + t^3 Q_1 \\ P_1' &= (1-s)Q_0 + sQ_m' \\ P_2' &= (1-s)Q_1 + sQ_m' \end{aligned} \quad (5-3)$$

を得る。アフィン変換 A の逆変換 A^{-1} （これもアフィン変換）を曲線(5-3)に施して、曲線 $C(t) = A^{-1}(C'(t))$ に写すと、アフィン変換が比を保つこと、およびBezier曲線のアフィン不変性(2-13)により、

$$\begin{aligned} C(t) &= (1-t)^3 Q_0 + 3(1-t)^2 t P_1 + 3(1-t)t^2 P_2 + t^3 Q_1 \\ P_1 &= (1-s)Q_0 + sQ_m \\ P_2 &= (1-s)Q_1 + sQ_m \end{aligned} \quad (5-4)$$

を得る。ただし、 $P_1 = A^{-1}(P_1')$ 、 $P_2 = A^{-1}(P_2')$ とする。

等比 s が保存されることに注意する。

3次Bezier曲線(5-4)に対応するFerguson曲線が求めるものである。

5.2.2 手法の幾何的意味

上述した手法は、アフィン変換で円弧ケースに帰着させ、円弧近似曲線を逆アフィン変換するものである。ここで、基準形状である円弧を A^{-1} で逆アフィン変換すると非対称な三角形 $Q_0Q_mQ_1$ に内接する楕円弧が得られる。したがって、曲線(5-4)は楕円弧の近似曲線を与えていることになる。本ケースを「楕円弧ケース」とよぶのは、

このためである。

5.2.3 従来法との比較

$\theta_0 = \pi/3$ (60度), $\theta_1 = \pi/6$ (30度) の場合に, 等面積法を用いて等比を決めて得られる形状と, 従来法 (放物線法, 弦長法, 円弧長法, 平均円弧長法) との形状比較を図24に掲げる。等面積法が従来法に比べ楕円弧形状をよく近似しているのがわかる。

なお, 等面積法による形状は始点付近に若干の曲率のピークを持っている。原因は, この場合の基準形状である楕円弧の長軸端点 (曲率極大点) が始点付近にあるためである。

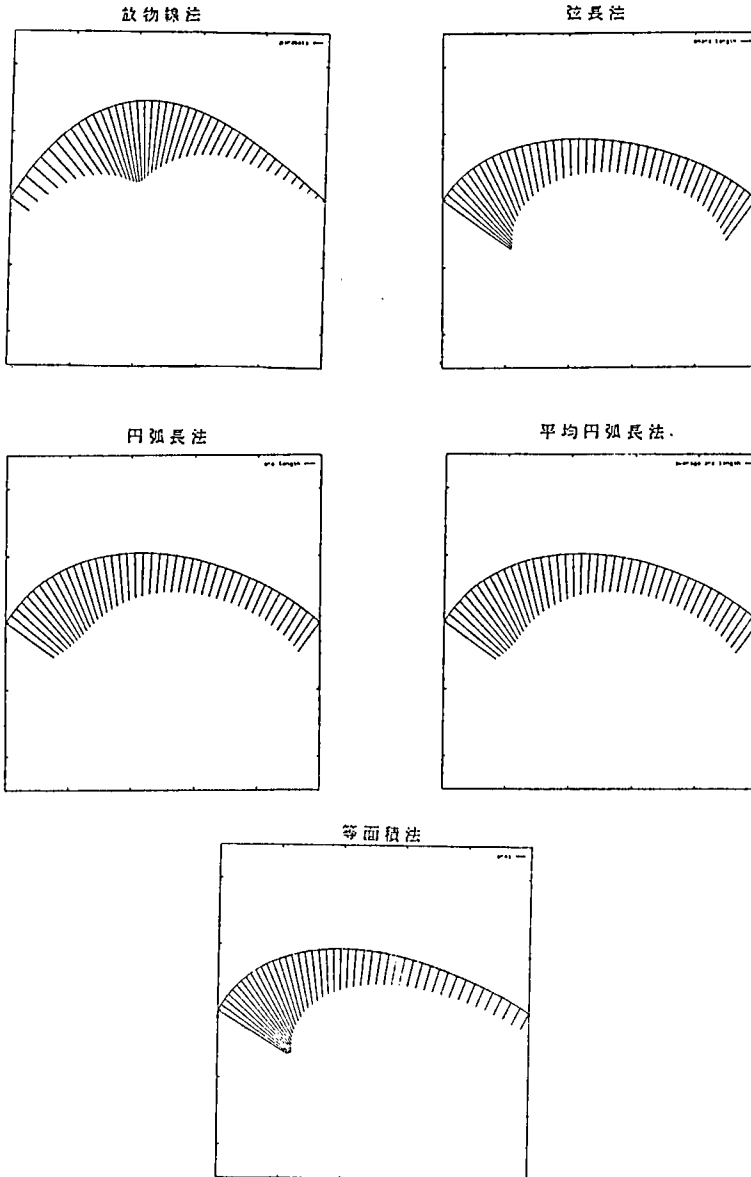


図24 楕円弧ケースでの手法比較

5.3 平面S字曲線ケースの解法

5.3.1 円弧ケースあるいは楕円弧ケースへの帰着

まず、どちらの接線ベクトルも弦 Q_0Q_1 と平行でない場合を扱う。

接線ベクトル T_0 を弦 Q_0Q_1 に関して折り返したベクトルを U_0 、同様に T_1 を弦に関して折り返したベクトルを U_1 とする。ここで、2本の Ferguson 曲線 $C_0(t)$ 、 $C_1(t)$ を次のように定義する (図25)。

$$\left. \begin{aligned} C_0(t) &= H_{0,0}(t)Q_0 + H_{0,1}(t)Q_1 + H_{1,0}(t)T_0 + H_{1,1}(t)U_1 \\ C_1(t) &= H_{0,0}(t)Q_0 + H_{0,1}(t)Q_1 + H_{1,0}(t)U_0 + H_{1,1}(t)T_1 \end{aligned} \right\} \quad (5-5)$$

2曲線は互いに合同であり、円弧ケースか楕円弧ケースになる。そこで、等面積法 (あるいは円弧中点通過法) による最適な等比 s を定めることができる (円弧あるいは楕円弧の近似曲線)。これにより接線長 $|T_0| = |U_0|$ 、 $|U_1| = |T_1|$ が決定される。

また、2曲線の2次Bezier制御三角形は合同であり (弦に関して折り返しの関係)、共通の等比で3次Bezier曲線の制御点が決まるから、

$$(U_0 - T_0) - (U_1 - T_1) = 0 \quad (5-6)$$

が成立する。証明の詳細は付録5で与える。

従って補題Aから、2曲線 $C_0(t)$ 、 $C_1(t)$ の1次ブレンド曲線 $C(t)$ は接線ベクトル T_0 、 T_1 を持つ Ferguson 曲線

$$C(t) = H_{0,0}(t)Q_0 + H_{0,1}(t)Q_1 + H_{1,0}(t)T_0 + H_{1,1}(t)T_1 \quad (5-7)$$

と一致する。こうして、どちらの接ベクトルも弦と平行でない平面S字曲線ケースの最適な接線長が決定された。

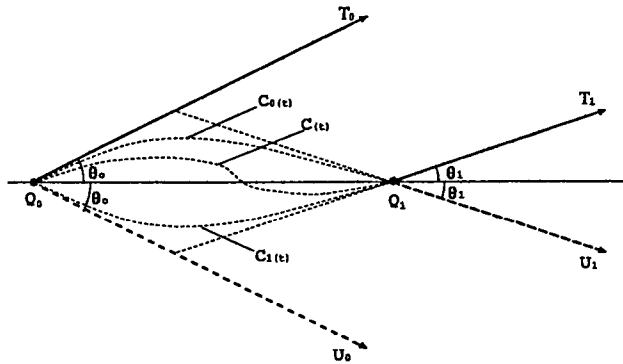


図25 平面S字曲線ケースの考え方

5.3.2 円弧ケースと線分ケースへの帰着

次に一方の接線ベクトルが弦 Q_0Q_1 と平行の場合を扱う。ここでは、 T_1 と弦とのなす角度 θ_1 が0の場合を考える。 T_0 と弦とのなす角度 θ_0 が0の場合も同様である。

始点側で角度0の接線ベクトルを考え U_0 とする。終点側で角度 θ_0 の接線ベクトルを弦に関して T_0 と逆側にとり U_1 とする。2本の Ferguson 曲線 $C_0(t)$ 、 $C_1(t)$ を次のように定義する (図26)。

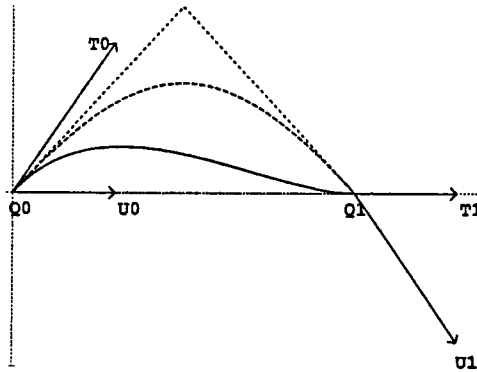


図 26 平面 S 字曲線：一方の接ベクトルが弦と平行

$$\left. \begin{aligned} C_0(t) &= H_{0,0}(t)Q_0 + H_{0,1}(t)Q_1 + H_{1,0}(t)T_0 + H_{1,1}(t)U_1 \\ C_1(t) &= H_{0,0}(t)Q_0 + H_{0,1}(t)Q_1 + H_{1,0}(t)U_0 + H_{1,1}(t)T_1 \end{aligned} \right\} \quad (5-8)$$

曲線 $C_0(t)$ は円弧ケースの条件を満たすので、等面積法（あるいは円弧中点通過法）による最適な等比 s を定めて、接線長 $|T_0| = |U_1|$ を決定する。

曲線 $C_1(t)$ は線分ケースであるが、曲線 $C_0(t)$ の等比 s は $0 < s < 2/3$ の範囲にあり、 $0 < (3s/2) < 1$ だから、4.2 節で注意した線分ケースでの接ベクトル長の不定性を利用し、敢えて弦長 l に補正係数 $(3s/2)$ を掛けて、 $|U_0| = |T_1| = (3s/2)l$ により接線長を決定する。

このように 2 曲線 $C_0(t)$ 、 $C_1(t)$ の形状を決めると、

$$(U_0 - T_0) - (U_1 - T_1) = 0 \quad (5-9)$$

が成立する。証明の詳細は付録 5 で与える。

したがって補題 A から、2 曲線 $C_0(t)$ 、 $C_1(t)$ の 1 次ブレンド曲線 $C(t)$ は接線ベクトル T_0 、 T_1 を持つ Ferguson 曲線

$$C(t) = H_{0,0}(t)Q_0 + H_{0,1}(t)Q_1 + H_{1,0}(t)T_0 + H_{1,1}(t)T_1 \quad (5-10)$$

と一致する。こうして、一方の接ベクトルが弦と平行の場合の最適な接線長が決定された。

5.3.3 従来法との比較

平面 S 字曲線ケースについて、上記方法で接線長を決めた場合と、従来法（弦長法、円弧長法、平均円弧長法）との形状比較を図 27 に掲げる。

5.4 空間曲線ケースの解法

5.4.1 円弧ケース・楕円弧ケースへの帰着

次のように記号を定める。

- PL_0 : 弦 Q_0Q_1 と接線ベクトル T_0 を含む平面、
- PL_0^+ : PL_0 のうち弦 Q_0Q_1 を境界とし、接線ベクトル T_0 側に延びた半平面、
- PL_0^- : PL_0 のうち PL_0^+ と反対側の半平面、
- PL_1 : 弦 Q_0Q_1 と接線ベクトル T_1 を含む平面、
- PL_1^+ : PL_1 のうち弦 Q_0Q_1 を境界とし、接線ベクトル T_1 側に延びた半平面、
- PL_1^- : PL_1 のうち PL_1^+ と反対側の半平面。

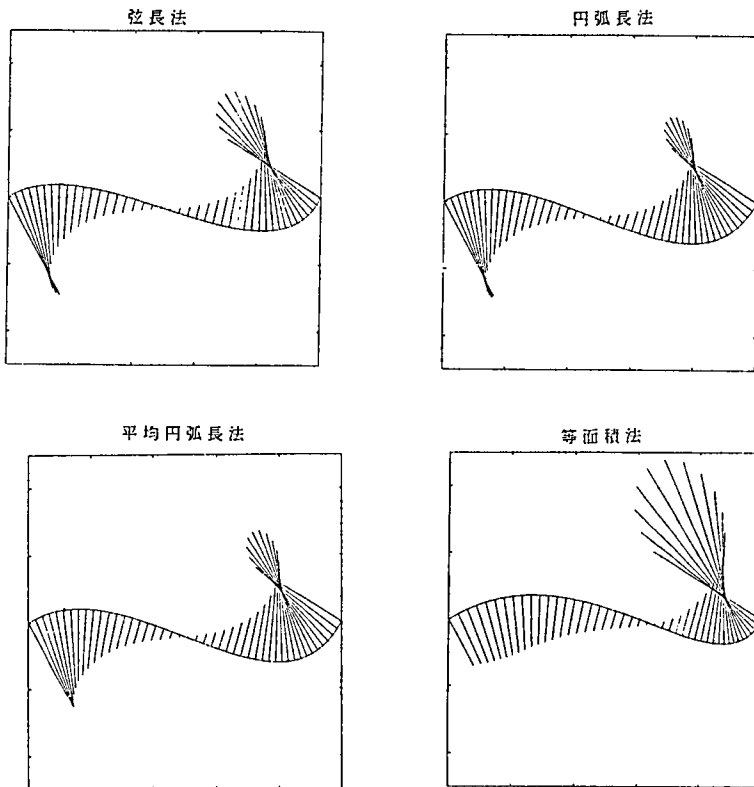


図27 平面S字曲線ケースでの手法比較

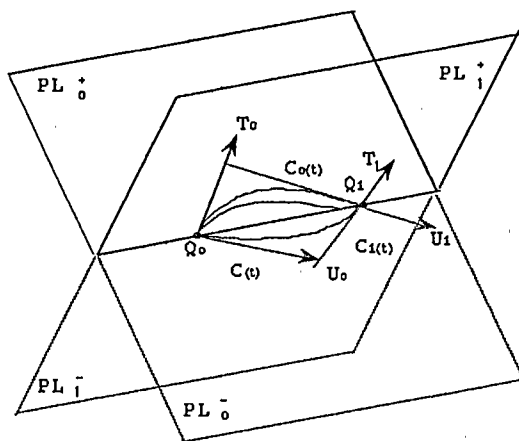


図28 空間曲線ケースの考え方

空間曲線ケースの定義により，2平面 PL_0 ， PL_1 は同一平面でない。

さて，弦 Q_0Q_1 を回転軸とし，半平面 PL_0^+ を半平面 PL_1^- に写す回転で，接線ベクトル T_0 を回転したものをベクトル U_0 とする．同様に，弦 Q_0Q_1 を回転軸とし，半平面 PL_1^+ を半平面 PL_0^- に写す回転で，接線ベクトル T_1 を回転したものをベクトル U_1

とする。これらを用いて、2本の Ferguson 曲線 $C_0(t)$ 、 $C_1(t)$ を次のように定義する (図28)。

$$\begin{aligned} C_0(t) &= H_{0,0}(t)Q_0 + H_{0,1}(t)Q_1 + H_{1,0}(t)T_0 + H_{1,1}(t)U_1 \\ C_1(t) &= H_{0,0}(t)Q_0 + H_{0,1}(t)Q_1 + H_{1,0}(t)U_0 + H_{1,1}(t)T_1 \end{aligned} \quad (5-11)$$

2曲線は互いに合同であり、円弧ケースか、楕円弧ケースになる。そこで、等面積法 (あるいは円弧中点通過法) による最適な等比 s を定めることができる (円弧あるいは楕円弧の近似曲線)。これにより接線長 $|T_0| = |U_0|$ 、 $|U_1| = |T_1|$ が決定される。

また、2曲線の2次Bezier制御三角形は合同であり (弦を中心にした回転の関係)、共通の等比で3次Bezier曲線の制御点が決まるから、

$$(U_0 - T_0) - (U_1 - T_1) = 0 \quad (5-12)$$

が成立する。証明の詳細は付録5で与える。

したがって補題Aから、2曲線 $C_0(t)$ 、 $C_1(t)$ の1次ブレンド曲線 $C(t)$ は接線ベクトル T_0 、 T_1 を持つ Ferguson 曲線

$$C(t) = H_{0,0}(t)Q_0 + H_{0,1}(t)Q_1 + H_{1,0}(t)T_0 + H_{1,1}(t)T_1 \quad (5-13)$$

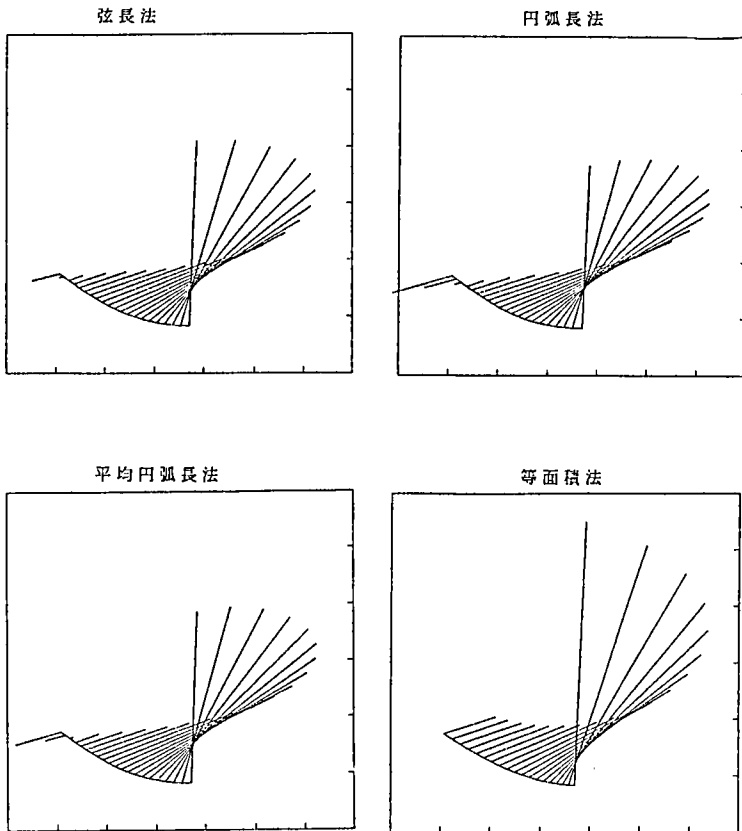


図29 空間曲線ケースでの手法比較

と一致する。こうして、空間曲線ケースの最適な接線長が決定された。

5.4.2 従来法との比較

実務でよく現れる空間曲線ケースについて、上記方法で接線長を決めた場合と、従来法（弦長法，円弧長法，平均円弧長法）との形状比較を図29に掲げる。本稿の方法で作られたセグメント形状の曲率分布が従来法に比べ穏やかに変化していることがわかる。

5.5 一般ケースのまとめ

以上の一般ケースの解法をまとめると、始点接線ベクトル \mathbf{T}_0 と弦とのなす角度 θ_0 ，終点接線ベクトル \mathbf{T}_1 と弦とのなす角度 θ_1 ，弦長 l が与えられれば，接線長を公式として書き下せることがわかる。

θ_0, θ_1 ともに0でない場合は，（ $\theta_0 = \theta_1$ の場合も含めて）楕円弧ケースの解法を用いて，(5-2)で角度 θ を決め（ $\theta_0 = \theta_1$ の場合は $\theta = \theta_0 = \theta_1$ ），

$$\left. \begin{aligned} |\mathbf{T}_0| &= 3sl(\tan \theta)/(2\sin \theta_0) \\ |\mathbf{T}_1| &= 3sl(\tan \theta)/(2\sin \theta_1) \end{aligned} \right\} \quad (5-14)$$

により接線長を決めればよい（ $0 < \theta < \pi/2$ に注意）。ここで， s は角度 θ の円弧ケースでの等面積法による等比である（定理C）。

$\theta_0 = 0$ の場合は，始点側に線分ケース，終点側に角度 θ_1 の円弧ケースを適用して

$$\left. \begin{aligned} |\mathbf{T}_0| &= (3s/2)l \\ |\mathbf{T}_1| &= (3s/2)l/(\cos \theta_1) \end{aligned} \right\} \quad (5-15)$$

となる。ここで， s は角度 θ_1 での円弧ケースの等面積法による等比である（定理C）。

$\theta_1 = 0$ の場合は，始点側に角度 θ_0 の円弧ケース，終点側に線分ケースを適用して

$$\left. \begin{aligned} |\mathbf{T}_0| &= (3s/2)l/(\cos \theta_0) \\ |\mathbf{T}_1| &= (3s/2)l \end{aligned} \right\} \quad (5-16)$$

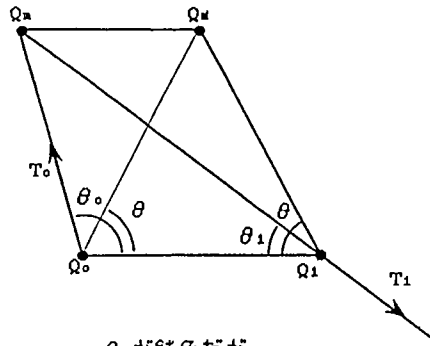
となる。ここで， s は角度 θ_0 での円弧ケースの等面積法による等比である（定理C）。

6. 今後の課題

基本仮定を満たさないケースは，本稿では対象としなかったが，処理プログラムの頑健性という観点からは，扱える範囲が可能な限り広い方が望ましい。実際，本稿での等比変形の考え方をそのまま適用できるケースも存在する（図30）。また，等比変形を適用できないが，半円を基準形状と見なして近似曲線を求められるケースもある（図31，付録6）。しかし，弦と接線ベクトルのなす角度が始終点とも鈍角になる場合（図32）には，新しい観点からの考察を要すると思われる。これらの点を改良し，理論の適用範囲を広げるのが，今後に残された第1の課題である。

また，本稿では，セグメントごとに独立にその接線長決定問題を考察したが，複数セグメントの場合は，隣接セグメントとの関係で他の拘束条件が課せられることがある。主なものを掲げると，

- 1) 曲率連続な接続問題……隣接するセグメントの共有点で曲率が連続になるようにセグメント形状を決定する場合は，3次Bezier曲線表現の中間制御点が拘束される^[1]，
- 2) スケール合わせ問題……二つ以上の複数セグメント自由曲線を拘束線として自



θ_0 が鈍角だが
アフィン変換で
円弧ケースに帰着

図30 基本仮定を満たさないが扱えるケース

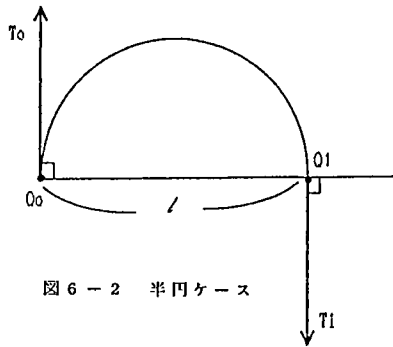


図 6 - 2 半円ケース

図31 半円ケース

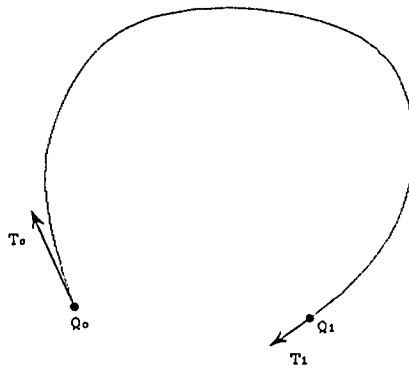


図32 始・終点とも鈍角のケース

由曲面を創成する場面では、隣接するセグメントの共有点での前後セグメントの接線長の比（スケールとよぶ）を各線の対応位置で共通にする必要がある、などがある。隣接セグメントからの拘束条件が課せられると、注目セグメントの接線長を最適に決めることができなくなる（等比変形の枠からはみ出てしまう）。複数セグメントで発生する拘束条件下での接線長決定理論の確立が残された第2の課題である。

このように本稿で提示したアプローチは、自由曲線の形状決定問題（さらには自由曲面の形状決定問題）への端緒に過ぎない。今後とも考察を継続・発展させていく予定である。

7. おわりに

本稿で得られた新しい結果は次の通りである。

- ① Ferguson 曲線による円弧の新しい近似手法（等面積法）を示したこと（定理 C），
- ② そのために等比変形，面積指標という概念を提示したこと（定理 B），
- ③ 基本仮定のもとで，一般の Ferguson 曲線の接線長決定問題が，アフィン変換と1次ブレンド法により基本ケースに帰着できることを示したこと，
- ④ 楕円弧の近似手法を示したこと，
- ⑤ 2本の Ferguson 曲線による1次ブレンド曲線が，再び Ferguson 曲線になるための条件を提示したこと（補題 A），
- ⑥ 新しい手法により作られる曲線形状を従来法での形状と比較し，その優位性を明らかにしたこと。

なお，本稿で参照した図版は PDS（Public Domain Software）の ghostscript 2.41J, gnuplot3.2, および KwikDraw1.21 を用いて作成した。

-
- 参考文献 [1] 山口富士夫，形状処理工学 [I]，日刊工業新聞社，1982。
 [2] 山口富士夫，形状処理工学 [II]，日刊工業新聞社，1982。
 [3] G.Farin 著，山口 泰監訳，CAGD のための曲線・曲面理論，共立出版，1991。

【付録 1】 補題 A の証明

1次ブレンド曲線の定義から，

$$\begin{aligned} C(t) &= (1-t) \cdot C_0(t) + t \cdot C_1(t) \\ &= H_{0,0}(t) \cdot Q_0 + H_{0,1}(t) \cdot Q_1 + H_{1,0}(t) \cdot ((1-t) \cdot T_0 + t \cdot U_0) \\ &\quad + H_{1,1}(t) \cdot ((1-t) \cdot U_1 + t \cdot T_1) \end{aligned}$$

一方， $C_2(t)$ の定義から，

$$C_2(t) = H_{0,0}(t) \cdot Q_0 + H_{0,1}(t) \cdot Q_1 + H_{1,0}(t) \cdot T_0 + H_{1,1}(t) \cdot T_1$$

両辺を引き算して，3次 Hermite 補間関数の定義式(2-1)を用いると

$$\begin{aligned} C(t) - C_2(t) &= H_{1,0}(t) \cdot t \cdot (U_0 - T_0) + H_{1,1}(t) \cdot (1-t) \cdot (U_1 - T_1) \\ &= (t-1)^2 \cdot t^2 \cdot (U_0 - T_0) - (1-t)^2 \cdot t^2 \cdot (U_1 - T_1) \end{aligned}$$

以上により

$$\therefore C(t) - C_2(t) = (1-t)^2 \cdot t^2 \cdot ((U_0 - T_0) - (U_1 - T_1))$$

証明終わり

【付録2】 定理Bの証明

弦 Q_0Q_1 の中点を原点 O ，弦の方向に x 軸，原点 O から円弧の中点に向かう方向に y 軸をとる (付図1)．この座標系で2次Bezier曲線の制御点は

$$\begin{aligned} Q_0 &(-l/2, 0) \\ Q_1 &(l/2, 0) \\ Q_m &(0, (l \tan \theta)/2) \end{aligned}$$

と成分表示される．等比 s での等比変形(4-4)に対応する Ferguson 曲線を

$$C(t) = (x(t), y(t))$$

と成分表示すると，3次Bezier曲線と Ferguson 曲線の関係(2-11)から

$$C(t) = H_{0,0}(t)Q_0 + H_{0,1}(t)Q_1 + H_{1,0}(t) \cdot 3s(Q_m - Q_0) + H_{1,1}(t) \cdot 3s(Q_1 - Q_m)$$

なので，成分関数の表式

$$\begin{aligned} x(t) &= (l/2)(-4t^3 + 6t^2 - 1 + 3s(2t^3 - 3t^2 + t)) \\ y(t) &= (l/2)(3s \cdot \tan \theta)t(1-t) \end{aligned}$$

が得られる．

曲線と弦で囲まれる領域の面積 S は成分関数の定積分により

$$S = 2 \cdot \int_{t=0}^{t=1/2} y(t) (dx/dt) dt$$

で表される．積分計算を実行すると

$$S = (3/40)s(4-s)(\tan \theta)l^2$$

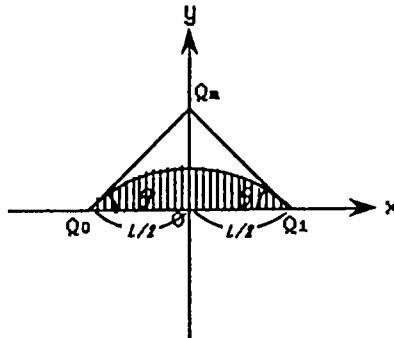
となる．

また，円弧と弦で囲まれる領域の面積 S_c は

$$S_c = (\theta - \sin \theta \cos \theta)l^2 / (4\sin^2 \theta)$$

であるから，面積指標 I は

$$\begin{aligned} I &= S/S_c \\ &= (3/10)s(4-s)(\tan \theta \sin^2 \theta) / (\theta - \sin \theta \cos \theta) \end{aligned}$$



付図1 定理Bの証明

となる。

証明終わり

【付録3】 定理Cの証明

円弧ケースで面積指標Iを1にする等比の存在証明を行う。

付録2で証明したように

$$I = (3/10)s(4-s)(\tan \theta \sin^2 \theta) / (\theta - \sin \theta \cos \theta)$$

であるから、 $I=1$ とおくと、角度 θ を $0 < \theta < \pi/2$ の範囲で任意に固定したときsについての2次方程式が得られる。具体的に係数を書き下すと

$$s^2 - 4s + (10/3)(\theta - \sin \theta \cos \theta) / (\tan \theta \sin^2 \theta) = 0$$

である。考えている θ の範囲では定数項の分母 $(\tan \theta \sin^2 \theta)$ は正の値であることに注意する。

2次方程式の判別式をDとおくと、

$$\begin{aligned} D/4 &= (-2)^2 - (10/3)(\theta - \sin \theta \cos \theta) / (\tan \theta \sin^2 \theta) \\ &= 4(1 - (5/6)(\theta - \sin \theta \cos \theta) / (\tan \theta \sin^2 \theta)) \end{aligned}$$

と計算される。ここで、

$$f(\theta) = (5/6)(\theta - \sin \theta \cos \theta) / (\tan \theta \sin^2 \theta)$$

とおくと、

$$D/4 = 4(1 - f(\theta))$$

である。

後で証明するように $0 < f(\theta) < 5/9$ なので、 $D/4 > 0$ となり2次方程式は実根を持つ。解の公式で2実根を求めると

$$\begin{aligned} s &= 2 \pm 2(1 - f(\theta))^{1/2} \\ &= 2(1 \pm (1 - f(\theta))^{1/2}) \end{aligned}$$

であるが、等比変形の等比の範囲 $0 < s \leq 2/3$ に適合するのは

$$s = 2(1 - (1 - f(\theta))^{1/2})$$

の方である。

残された $f(\theta)$ の範囲に関する証明を行う。

まず、 θ を0に近づけたときの $f(\theta)$ の極限值を求める。これは $0/0$ の不定形であるが、ロピタルの定理により分母と分子を独立に θ で微分して極限值 $\lim f(\theta)$ を求めると、

$$\begin{aligned} \lim f(\theta) &= (5/6) \cdot \lim (2\sin^2 \theta) / (\tan^2 \theta + 2\sin^2 \theta) \\ &= (5/6) \cdot \lim (2 / (\sec^2 \theta + 2)) \\ &= (5/6) \cdot (2/3) = 5/9 \end{aligned}$$

が得られる。すなわち、 $f(0) = 5/9$ と考えてよい。

次に、 θ を $\pi/2$ に近づけたときの $f(\theta)$ の極限值を求める。これは(有限)/(無限)になるので0である。すなわち、 $f(\pi/2) = 0$ と考えてよい。

最後に $0 < \theta < \pi/2$ の範囲で $f(\theta)$ が単調減少であることを示すために、 $f(\theta)$ を θ で微分して導関数 $df/d\theta$ を求める。

$$df/d\theta = (5/6)(\sin^2 \theta)(g(\theta)) / (\tan^2 \theta \sin^4 \theta)$$

ここで、

$$g(\theta) = 2 \tan \theta \sin^2 \theta - (\theta - \sin \theta \cos \theta)(\sec^2 \theta + 2)$$

とおいた。 $0 < \theta < \pi/2$ の範囲で $g(\theta)$ が単調減少であることを示すために導関数 $dg/d\theta$ を求める。

$$dg/d\theta = -2(\sin \theta \cos \theta)(\theta - \sin \theta \cos \theta)/(\cos^4 \theta)$$

であり、 $(\theta - \sin \theta \cos \theta) > 0$ だから(注)、 $dg/d\theta < 0$ 。 すなわち、 $g(\theta)$ は単調減少。
 $g(0) = 0$ に注意すると、

$$g(\theta) < 0$$

を得る。従って、

$$df/d\theta < 0$$

であり、 $f(\theta)$ は単調減少。こうして、 $0 < \theta < \pi/2$ の範囲で

$$0 < f(\theta) < 5/9$$

を得る。これより、 $(1 - f(\theta))^{1/2}$ の範囲は

$$2/3 < (1 - f(\theta))^{1/2} < 1$$

であり、等比 s の範囲は

$$0 < s < 2/3$$

となる。

証明終わり

(注) $0 < \theta < \pi/2$ の範囲で $(\theta - \sin \theta \cos \theta) > 0$ の証明は容易である。

$h(\theta) = \theta - \sin \theta \cos \theta$ とおく。導関数 $dh/d\theta$ を求めると

$$dh/d\theta = 2 \sin^2 \theta > 0$$

より、 $h(\theta)$ は単調増加。 $h(0) = 0$ だから、 $0 < \theta < \pi/2$ の範囲で

$$h(\theta) > 0$$

である。

[付録4] 楕円弧ケースでの二等辺三角形への変形手続きの証明

弦 Q_0Q_1 の中点を原点 O 、弦の方向に x 軸、弦の垂直二等分線で原点 O から Q_m のある側に y 軸をとり、局所座標系で考える。この座標系での Q_m の x 座標を q 、 y 座標 (= 三角形 $Q_0Q_mQ_1$ の高さ) を h とする。各点を成分表示すると

$$\begin{aligned} Q_0 &(-1/2, 0) \\ Q_m &(q, h) \\ Q_1 &(1/2, 0) \end{aligned} \tag{A4-1}$$

となる (付図2)。

三角形の二辺 Q_0Q_m 、 Q_1Q_m を底辺 Q_0Q_1 に正射影して

$$h(\cot \theta_0) + h(\cot \theta_1) = l \tag{A4-2}$$

に注意する。

Q_m を x 軸に平行に y 軸上まで移動し Q_m' とする。 Q_m' の成分表示は

$$Q_m'(0, h) \tag{A4-3}$$

である。

y 軸の定義から三角形 $Q_0Q_m'Q_1$ は二等辺三角形となる (底角の角度を θ とする)。二等

辺三角形の二辺を底辺に正射影すると、

$$2h(\cot \theta) = l \quad (\text{A4-4})$$

を得る。(A4-2), (A4-4)から

$$\cot \theta = (\cot \theta_0 + \cot \theta_1)/2 \quad (\text{A4-5})$$

が示された。

最後に、 Q_m を Q_m' に移動する変換がアフィン変換で実現できることを示す。

次の行列で定義される線形変換Mを考える。

$$M = \begin{bmatrix} 1 & -q/h \\ 0 & 1 \end{bmatrix} \quad (\text{A4-6})$$

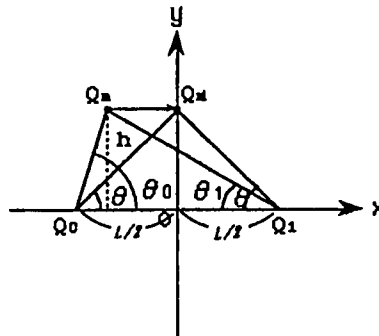
Q_0, Q_1, Q_m の成分表示を列ベクトルと考えたものに行列Mを左から乗ずることで

$$M(Q_0) = Q_0, \quad M(Q_1) = Q_1, \quad M(Q_m) = Q_m' \quad (\text{A4-7})$$

が示される。

以上は局所座標系での表示であるが、あらかじめ定められた全体座標系から局所座標系への変換は座標軸の回転と原点の平行移動で表されるから、線形変換Mは全体座標系ではアフィン変換で表される。

証明終わり



付図2 楕円弧ケースでの2等辺三角形への変形手続きの証明

【付録5】 平面S字曲線・空間曲線ケースでの補題Aの条件成立の証明

1) 証明の準備

まず、簡単であるが有用な補題を提示する。

補題D

2次Bezier曲線の制御三角形 $Q_0Q_m, Q_0Q_1, Q_0Q_m, Q_1Q_m$ から共通の等比 s ($0 < s \leq 2/3$) で等比変形した二つの3次Bezier曲線

$C_0(t), C_1(t)$ を次式で定義する。

$$C_0(t) = (1-t)^3 Q_0 + 3(1-t)^2 t P_{1,0} + 3(1-t)t^2 P_{2,0} + t^3 Q_1$$

$$P_{1,0} = (1-s)Q_0 + sQ_{m,0}$$

$$P_{2,0} = (1-s)Q_1 + sQ_{m,0}$$

(A5-1)

$$\begin{aligned}
 C_1(t) &= (1-t)^3 Q_0 + 3(1-t)^2 t P_{1,1} + 3(1-t)t^2 P_{2,1} + t^3 Q_1 \\
 P_{1,1} &= (1-s)Q_0 + sQ_{m,1} \\
 P_{2,1} &= (1-s)Q_1 + sQ_{m,1}
 \end{aligned}
 \tag{A5-2}$$

また、これらに対応する Ferguson 曲線の接線ベクトルを次とする。

$$\begin{aligned}
 T_0 &= 3(P_{1,0} - Q_0) = 3s(Q_{m,0} - Q_0) \\
 U_1 &= 3(Q_1 - P_{2,0}) = 3s(Q_1 - Q_{m,0}) \\
 U_0 &= 3(P_{1,1} - Q_0) = 3s(Q_{m,1} - Q_0) \\
 T_1 &= 3(Q_1 - P_{2,1}) = 3s(Q_1 - Q_{m,1})
 \end{aligned}
 \tag{A5-3}$$

このとき、補題Aの条件式

$$(U_0 - T_0) - (U_1 - T_1) = 0$$

が成立する

補題Dの証明

共通の等比sを用いているので、これらの接線ベクトル間には、関係式(A5-3)から直接計算して、

$$\begin{aligned}
 (U_0 - T_0) - (U_1 - T_1) &= 3s(Q_{m,1} - Q_{m,0}) - 3s(Q_{m,1} - Q_{m,0}) \\
 &= 0
 \end{aligned}$$

の関係が成立している。

補題D証明終わり

(注) 補題Dで本質的な条件は、等比sが2曲線で共通になることである。

2) 平面S字曲線ケース：どちらの接線ベクトルも弦と平行でない場合

本文での式(5-5)の2曲線に等比sによる等比変形(5-3)を施した3次Bezier曲線表現(A5-1)、(A5-2)を考えると、それぞれに円弧ケースか楕円弧ケースの解法を適用して、2次Bezier曲線の間接制御点 $Q_{m,0}$ 、 $Q_{m,1}$ が決められる。これらの制御三角形は合同であるから、等面積法（あるいは円弧中点通過法）による等比sは共通に決定される。したがって、補題Dにより補題Aの条件が成立する。

3) 平面S字曲線ケース：一方の接線ベクトルが弦と平行である場合

接線ベクトル T_1 が弦と平行である場合を考える。

本文での式(5-8)の2曲線のうち、曲線 $C_0(t)$ には円弧ケースの解法が適用されて、等面積法（あるいは円弧中点通過法）による等比sが決定される。補題Dの式(A5-1)は、円弧ケースの解法により2次Bezier曲線の間接制御点 $Q_{m,0}$ が決められる。

一方、曲線 $C_1(t)$ は線分ケースであるが、線分の中点を $Q_{m,1}$ として、つぶれた制御三角形 $Q_0 Q_{m,1} Q_1$ から2次Bezier曲線を考える。曲線 $C_0(t)$ の等比sをこれに適用して等比変形 $C_1(t)$ を考えると、補題Dの式(A5-2)になる。これは、線分ケースの接線長を

$$\begin{aligned}
 |U_0| &= |T_1| = |3s(Q_{m,1} - Q_0)| \\
 &= 3(s/2)l \\
 &= (3s/2)l
 \end{aligned}$$

で決定していることに他ならない。

等比 s を共通にとっているので、補題Dから補題Aの条件が成立する。

接線ベクトル T_0 が弦と平行である場合の証明も同様である。

(注) この証明から、線分ケースの接線長をなぜ弦長にしないかが、明らかになる。線分側の接線長を弦長 l にすると、 $C_1(t)$ の等比が $2/3$ になり、円弧ケースの $C_0(t)$ 側の等比 s と一致しない。これらの1次ブレンド曲線に対しては、補題Aの条件が成立せず、1次ブレンド曲線は4次多項式曲線になってしまう。

4) 空間曲線ケース

2)で扱った「平面S字曲線ケース：どちらの接線ベクトルも弦と平行でない場合」の証明と同様。

以上で証明終わり

【付録6】 半円ケースの解法

次の3条件を満たす場合を半円ケースとよぶ。

- ① 弦 Q_0Q_1 と接線ベクトル T_0, T_1 が同一平面上にある。
- ② 接線ベクトル T_0, T_1 が弦 Q_0Q_1 に関して逆側を向く。
- ③ $\theta_0 = \theta_1 = \pi/2$ 。

この場合は、弦 Q_0Q_1 を直径とする半円が基準形状である。

このとき、接線が平行で交点 Q_m を得られないので等比変形手法は使えないが、接線長を直接用いた変形を考えることで、半円への近似を円弧ケースと同じように扱うことができる。

弦長 $|Q_0Q_1|$ を l とする(半円の半径は $l/2$)。 $|T_0| = |T_1| = a$ とし、各接線ベクトルの単位化ベクトルを N_0, N_1 とする。Ferguson曲線表現は次のようになる。

$$C(t) = H_{0,0}(t)Q_0 + H_{0,1}(t)Q_1 + H_{1,0}(t)aN_0 + H_{1,1}(t)aN_1 \quad (A6-1)$$

面積指標 I を円弧ケースと同様に定義すると、(A6-1)を用いて I は次のように計算される(証明は定理Bと同様にできるので省略)。

$$I = (8a)/(5\pi l) \quad (A6-2)$$

従来法を面積指標で評価すると次のようになる(放物線法は適用できない)。

1) 弦長法

$$\text{接線長: } a = l$$

$$\text{面積指標: } I = 8/(5\pi) \doteq 0.509296$$

2) 円弧長法

$$\text{接線長: } a = (1/2)\pi l$$

$$\text{面積指標: } I = 4/5 = 0.8$$

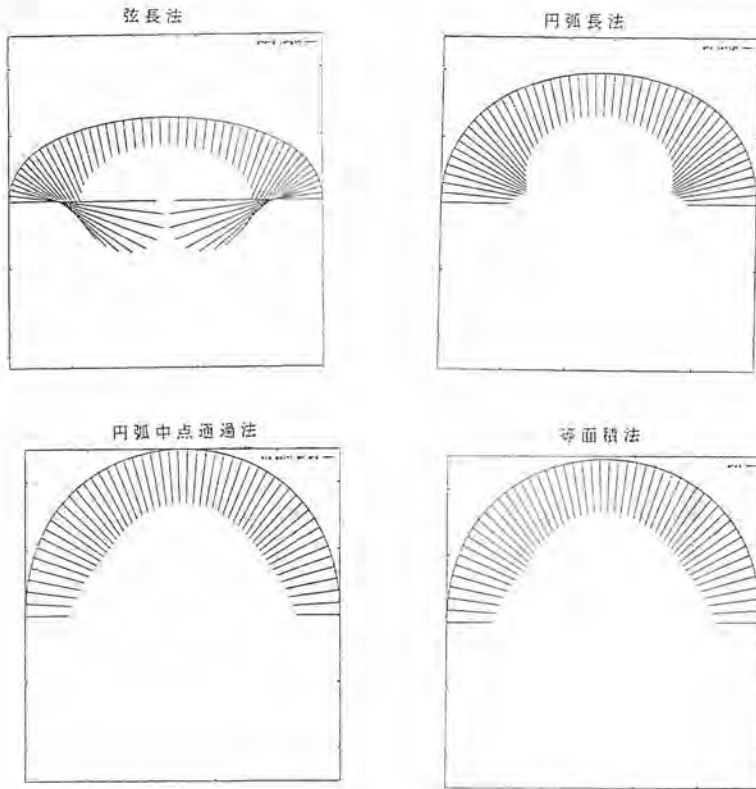
3) 円弧中点通過法

$$\text{接線長: } a = 2l$$

$$\text{面積指標: } I = 16/(5\pi) \doteq 1.018592$$

半円ケースでの等面積法は(A6-2)で $I=1$ と置くことにより、接線長 $a = (5/8)\pi l$ とすれ

ばよいことは容易にわかる。



付図3 半円ケースでの手法の比較

各手法で得られる Ferguson 曲線の形状を比較したのが付図3である。円弧中点通過法は、半円の中点を通過しているが、等面積法は中点の少し下を通過する。半円ケースでは、通過点重視か大域的な近似重視かで、どちらの手法を優位に置くかの評価が分かれる。

執筆者紹介 清水保弘 (Yasuhiro Shimizu)

昭和29年生。53年東京大学理学部数学科卒業。55年東京都立大学大学院理学研究科修士課程修了。59年同博士課程満期退学。60年日本ユニシス(株)入社。CAD/CAMシステムの開発に従事。現在、製造工業システム第一本部CAD/CAMシステム部に所属。日本数学会会員。情報処理学会会員。



宮 地 恵 美 (Emi Miyachi)

昭和32年生。57年慶応義塾大学大学院工学系研究科修士課程修了。同年日本ユニシス(株)入社。CAD/CAMシステムの開発に従事。現在、製造工業システム第二本部システム三部に所属。情報処理学会会員。



“統合化”と企業組織——異質性のダイナミズム

“Integrating” and the Corporate Organization ——Dynamism of Heterogeneousness

ト 部 正 夫

要 約 本稿は、“統合化”と企業組織の成長について論じたものである。“統合化”の定義として、「異質な要素の組み合わせによって、新奇性を織り込んだ、より高次の統一体ないし全体を構築する行動過程」を採用し、企業組織の成長過程を6段階の進化ステージとして設定した上で、各ステージの特徴およびステージ間の移動について論じている。ステージを分ける基本的な軸として統合範囲の拡大と対境積極性の増大を上げ、それぞれを部門内、部門間、組織間および環境維持、環境適応、環境創造に3分割した。

六つのステージのうち特に重要なステージはステージ1（部門内—環境維持）、3（部門間—環境適応）、6（組織間—環境創造）であり、その他のステージはこれらのステージを移動していく過渡的なステージと位置づけできる。ステージ1（部門内—環境維持）は未統合の段階であり、そこでは中央集権的な管理と統制によるマネジメントが展開される。ステージ3（部門間—環境適応）は一つの組織がその全体を統合した段階であり、各部門、各個人が組織全体の観点から、組織を取りまく環境の変化に適応していく姿が描かれている。ステージ6（組織間—環境創造）は統合化の最終段階であり、複数の異質な組織が組織間ネットワークを構成する。そこでは、異質な組織が相互を認知し、その知識の連鎖の中から共通のビジョンを創造し、ビジョン実現に向けて協働する。統合化の対象要素として、ビジネスの使命・戦略、経営資源、ビジネスプロセス、組織、経営管理を上げ、各々の側面から各ステージの特性を説明している。

企業組織の統合化のプロセスは、過去の分業による効率化のパラダイムから異質の取り込みによる新しい環境の創造へのパラダイム転換の過程であり、組織の態様として階層型組織からネットワーク型組織への転換がその実現の鍵を握っている。ステージ6（組織間—環境創造）は組織の形態としては統合化の最終の姿を示しているが、実際の企業組織の統合化のプロセスは終わることはない。環境創造とは環境と組織とのたゆまぬ交互作用の過程であり、組織の中に常に異質を取り込み、環境との対話の中から自らのビジョンを変革していく絶えざる進化の道程である。

Abstract This paper is intended to discuss relationships between “integrating” and the growth of the business organization. By citing a definition of “integrating” as “a process of action for creating a higher-level united body or whole with novelty woven in by combining heterogeneous elements,” this paper divides the growth processes of the corporate organization into six different development stages, and addresses the noticeable points of each stage and transitions from stage to stage. Assuming that the basic dimensions that separate those stages are an expanded scope of integration and an increased positiveness toward an external environment, the author separates the stages into three levels that are each called ‘intra-department,’ ‘inter-department,’ and ‘inter-organization’ or ‘adherence to the past

environment,' 'adaptation to the current environment' and 'creation of a future environment.'

Out of the six stages, the ones that are particularly important are stage 1 (intra-department — adherence to the past), stage 3 (inter-department — adaptation to the present) and stage 6 (inter-organization — creation of the future). The other stages can be positioned as transient stages the organization works its way through for growth. Stage 1 represents a not-yet integrated level featured by management based on bureaucratic control and regulations. Stage 3 stands for the phase in which one organization has been wholly integrated, depicting how each department and individuals are reacting to changes in an environment around them from an aspect of the organization as a whole. Stage 6 is synonymous with the final phase of integration that sees a composed inter-organizational network linking heterogeneous organizations. At this final stage, those heterogeneous organizations can now recognize each other, so that they can form common visions out of their chain of knowledge and join hands to get to work on the visions. As the elements involved in such integration, the author focuses on the mission/strategy of business, business resources, business processes, organization and business administration so the characteristics of each stage can be described from each of these aspects.

The integration process for the corporate organization is the process of shifting paradigms from the past work efficiency supported by the division of labor to the creation of a new environment with heterogeneousness brought in. The key factor in implementing this process is the transformation of an organizational style from a hierarchical structure to a networked organization. Stage 6 shows the ultimate form of integrated organization, but the actual process of corporate organization integration never sees the curtain falling. As a process of perpetual interaction between environment and organization, the creation of a new environment is a constant incorporation of heterogeneousness into the organization — a continued evolutionary process of changing an organization's own visions through the exchange of dialogs with environments.

1. はじめに

“統合”と言う言葉は、いろいろな分野でよく使用されている。例えば情報処理分野では、統合ソフトウェア、メインフレームと分散機の統合、経営と情報技術の統合等がある。ビジネスの分野では、製販の統合、事業部間にまたがる統合組織、企業間統合等の使われ方がみられる。しかしながら、一般的に統合という言葉が使われるとき、そこには多様な意味合いがあり、結合、融合、整合、連鎖等の言葉との相違は判然としない。

上記のような統合の使い方で想起される英語は“Integration”であるが、その動詞である“integrate”は、<部分・要素を>全体にまとめると言う意味である。海老澤は“統合化”を【異質な要素の組み合わせによって、新奇性を織り込んだ、より高次の統一体ないし全体を構築する行動過程】と定義し、統合化の理念を【組織体における要素間の関係性、連続性の範囲を広め、かつ内容を質的に変え、進化させること】としている¹⁾。

メインフレームと分散機の統合は、メインフレームはメインフレームの固有の良さを、分散機は分散機の良さを互いに生かしながら、それらを統合することによって単独では得られなかった新たな価値を導き出すことである。製販の統合は、製造は製造の論理、販売は販売の論理を振りかざすのではなく、相互の組織単位が協働することによって、新しい価値を創造したり、より質の高い企業活動を実現することである。

このように、統合化とは1+1を3にも4にも高めていこうとする創造的行為である。統合化される各要素はあくまでその固有の価値を失わずに、なおかつ協働することによって、各要素にはない新しい世界を切り開いていく極めて進化的な行為であると言える。しかしながら、あくまでこのような創造的進化は、異質な要素の組み合わせによってしか実現されない。同質的なものが集まるだけでは、量的なパワーを生かすことは可能であっても、その中から新奇なものは生まれてこない。

日本の多くの企業は、長い間右肩上がりの成長を謳歌した後、構造的とも言われる不況に直面し、縮み指向が蔓延しているように見える。量的拡大はもはや望めない。いかにして組織としての質的転換を図ることができるかが、企業組織の抱える大きな課題である。米国という先例が既になくなった日本の企業にとって、異質性のダイナミズムが生み出す統合化の過程を辿ることにより、新たな創造的進化を実現することが、今日の閉息状態を切り開く唯一の道と思われる。

2. 統合化による組織成長モデル

異質性の取り込みによる組織進化の概念を企業の戦略的なビジネス行動におきなおしたとき、次の二つの進化の軸を考えることができる。一つが統合範囲の拡大であり、他の一つが対境¹⁾積極性の増大である(図1)。統合範囲の拡大の軸は、文字どおり統合化された空間の広がりを示している。対境積極性とは組織が環境に対して能動的に働きかける姿勢の強さの程度を意味しており、その程度の低い組織の典型が城砦型組織であり、高くなればなるほど都市国家型の組織¹⁾に近づく。

城砦型の組織とは、自組織と環境の間に強固な垣根を築き、自組織の中への異質分子の進入を防御する。城砦型の組織にいる人々は、常に同質的な、自分達が築き上げてきた過去の成功体験に裏打ちされたパラダイムで環境を読み、環境に対応する。一方、都市国家型の組織においては組織の垣根は低く、組織員は外部の多様な人々と常に交わり、異質とのぶつかりあいの中から自分達のパラダイムを変革していく姿勢を持つ。都市国家型の組織においては、多様な外部組織との知識の連鎖を築きながら、

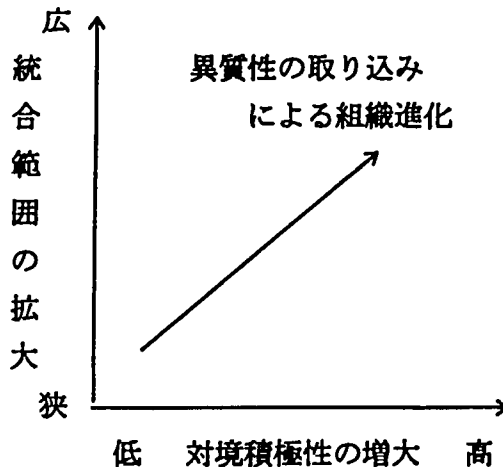


図1 統合化の方向

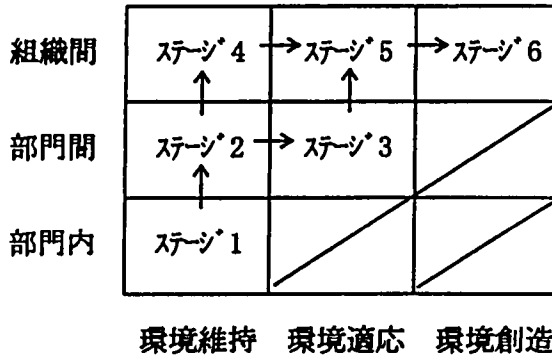


図2 統合化の進化ステージ

環境に単に適応して行くだけでなく、さらに積極的に環境を創造していく風土を組織内に育む。対境積極性の増大の軸は、組織の開放度の増大と基を一にしている。

ここでは、企業組織の統合化の発展段階を図2に示す六つのステージに分けて説明する。図2では、図1の縦軸すなわち統合範囲の拡大を部門内、部門間、組織間の三つの発展段階に分け、横軸である対境積極性の増大を環境維持、環境適応、環境創造のやはり三つに区分した。

縦軸の部門内の統合化とは、組織の各部門レベルでは統合化が図られ、業務の連結はスムーズに流れ、かつ部門の成員は互いの知恵を出し合いながら部門のパフォーマンスを上げて行くが、各部門間の連携は公式的なコミュニケーションルートによる情報の伝達だけであり、異質な部門同士が多彩な情報交換をしながら、組織全体のパフォーマンスを上げていく段階には達していない。部門間の統合化とは、このような統合化が組織全体にわたった段階であり、さらに組織間になると外部組織とネットワークを組み、その組織間ネットワークの中での統合化が図られるようになる。このような統合範囲の拡大は、組織の成長・拡大を求め、組織の最大限のパフォーマンスを達成しようと努力するさかりにおいて、半ば必然的に起こる現象と言える。

一方、横軸の変化すなわち組織の対境積極性を増大させることは簡単ではない。環境維持の段階とは、組織が過去の成功体験に固執し、その時点の環境が変わらないとの幻想のもとに、組織のパワーを「内なる効率化」¹⁴⁾に向けて集中している状況を指す。しかしながら、実際には環境はとどまることなく変化して行くために次第に組織は環境不適應を起こしているのであるが、組織成員は環境を見つめるという習慣が育っていないためにそのことに気づかないことが多い。環境不適應がだんだん大きくなり、その影響が組織の明示的なパフォーマンス指標（売上、利益、顧客数等）の大きな低下となって現れてはじめて組織はそれを感知し¹⁵⁾、次の段階への移行の必要性に気づくことが多いのが一般的である。

図2の横軸に対して時間の概念を使用すれば、次のような説明が可能である。環境維持の段階では、組織員の環境パラダイムは過去の一時点で止まった状態と言える。環境適応の段階になると、組織は変わりゆく環境に対してさまざまな適応行動をとる。しかしながらこの段階での環境適応は、組織の環境適応能力が十分に開発されているとは言い難く、多分に短期的視野にもとづく環境適応行動がとられやすい。環境創造

の段階に入ると、組織は環境への適応行動の積み重ねの中から環境の長期的トレンドをイメージすることが可能になり、より遠い将来の時点における自組織の理想像を描くようになる。そうすると、組織はその理想状態に早く近づくために環境に対して能動的に働きかける行動をとるようになり、対境積極性の非常に高い状態に至る。このような組織では、長期の目標状態を頭に描きながら短期的なさまざまな行動を起こし、かつその反応を常に見定めながら長期の目標状態を小刻みに変更していくことになる。換言すれば、環境創造段階に至った組織は、常に環境との対話をしながら創造すべき環境を模索している組織と言えよう。

統合度の最も低いステージ1から最も統合度の高いステージ6に至る過程としては、さまざまなルートが考えられる。しかしながら最も一般的なルートは、ステージ1→ステージ2→ステージ3→ステージ5→ステージ6の階段上に図2を右上がりにとどるルートであろう。それは、縦に一段統合範囲を拡大することにより視野が広がり、それが組織のパラダイム転換を促し横の一つ移動していくことになるためである。しかしながら、ステージ2からパラダイムの転換が起きずステージ4へ行く系列化の動きもあり得るし、ステージ1からステージ3ないしステージ6に途中の段階を経ずして組織が成長することもある。系列化は強圧的なパワーを持って弱者を従え同質的ネットワークを築くことであり、異質性を掲げる統合化の概念とは相入れないものである。系列化に対する概念としては連邦化ということになる。

3. 環境維持を目指す部門内統合化

3.1 部門内統合化の目的と特性

部門内の統合化にとどまっている組織の目的は効率化にある。企業はその成長過程において、市場とのインタラクションから効果的な企業行動の方向性を見定め、仕組みを築いていく。しかしながら、いったんその仕組みができあがると多くの企業はその精緻化を図り、その仕組みに則って内部の効率化に邁進するようになる。現業およびマネジメントの職務が細分化され、企業内の人々は、その仕組み全体と環境との不一致に気づかない。統合化のレベルが低い組織における最も大きな負の特性は、組織員の環境不感知の態度にある。

企業活動においては、常に効率 (Efficiency) の前に効果 (Effectiveness) を考えなければならない⁶⁾。ここで言う効率とは、投入量に対する産出量のように成果の量的側面を指しており、効果とは成果の有効性すなわち成果の質的側面を指している。さまざまな企業の行動は、市場との関連においてそれが効果的な行動であるかどうか、またさまざまな行動が作り出す全体の仕組みが市場と乖離していないかが常に検証されていなければならない。企業行動において、効果と効率は自転車に例えれば前輪と後輪に当たる。ハンドルに直結した前輪の指し示す方向に後輪を効率よく回すことによって、企業は自ら到達したい地点に向かう。途中の地形、障害物等の環境に合わせて企業はハンドルを小刻みに調整することが必要である。過去の環境に合わせて前輪の方向を一定に保ち、どんなに効率よく後輪を回しても、組織のパフォーマンスを上げることはできない。そのまま進めばそのうちに暗礁に乗り上げてしまう。

部門内統合化にとどまる企業の戦略展開は、トップダウンアプローチをとることが

多い。企業の行動は事前の計画により細かくスケジュール化され、統制が行われる。組織内には官僚性がはびこり、ラインおよびミドルマネジメントは上からの指示により行動する。ラインに市場を見つめる目がないために、市場の生き生きした情報が企業に取り込まれない。たとえそのような情報が下から上げられたとしても、多くの組織階層を通るうちに情報は解釈に解釈を重ねられ、異なった形でトップに伝えられる。

企業において本来、さまざまな情報が現場で感知され、伝達され、集約され、マイクロレベルの情報の解釈とマクロレベルの情報の解釈が多くの人々によって検証され、マクロレベルの重要な情報が再度組織のさまざまな部門に伝達され、部門ごとの行動に活かされていかなければならない。このような情報のマイクロマクロループ¹⁷⁾が企業内に存在してはじめて、企業は市場の動きと同期をとりながら効果のある企業の行動計画を立てることが可能になる。

統合化のレベルが低い組織においては、この情報のマイクロマクロループが存在しない、あるいはループが切れているために、トップダウンで立てられる事前の計画に市場の動きを組み込むことができず、いかにうまく統制され計画どおりの行動が遂行できたとしても、事業の成功はおぼつかない。

このように、市場を見ないあるいは現場の有効な情報が得られない一因として、ムダの徹底的排除という考え方がある。このような企業では、合理化とコストダウンが至上命題として組織員に要請され、既定の行動からの逸脱が許されない雰囲気ができあがる。部門内の仕事の仕組みから徹底的にムダが排除され、人々は市場に対して新しい行動をとっていくことをしない。市場の文脈はただ単に観察によって得られるものではなく、組織員がさまざまな行動を起こし、その結果の意味を解釈しながら次第に明らかになっていくものである。またこのような組織では、部門内の仕事の仕組みは整合性が図られムダが排除されても、部門間の協調行動は生まれず、組織と組織の間に不整合およびさまざまな重複が存在する。木を見て森を見ないのが、部門内の統

表1 部門内統合化のステージの特性

	ステージ1
ビジネスの使命, 戦略	<ul style="list-style-type: none"> ・ トップダウンによる事前設定 ・ 長期間一定 ・ 成功体験の固定化による既存戦略の強化行動
経営資源	<ul style="list-style-type: none"> ・ ハードな資源（物・金）の重視 ・ 過去の連続線上の実績による資源配分 ・ 資源の部門間重複
ビジネスプロセス	<ul style="list-style-type: none"> ・ 効率重視 ・ 経験による部門内ビジネスプロセスの定式化と精緻化 ・ 部門間のつながりの悪さ
組織	<ul style="list-style-type: none"> ・ 深い縦型構造 ・ 部門間の壁 ・ 公式的コミュニケーション・チャネルの重視
経営管理	<ul style="list-style-type: none"> ・ 公式的な計画策定と統制プロセス ・ QC活動

合化にとどまる企業の大きな弱点である。

表1には、部門内統合化のステージの特性をまとめている。

3.2 部門内統合化の限界

ステージ1のような企業は、過去の環境変化が少ない時代においては十分にその存在理由があった。組織をとりまく環境があまり変わらず、組織の進むべき方向が長期にわたって明白である状況においては、ハンドルの方向をあまり気にすることなく効率に気を配ればよかった。効率を最大に持つていくために、組織タスクの細分化が行われ、専門化が指向されたのである。このような状況の中で、そのタスクを管理するためのマネジメントの階層化が行われ、人々は部門の中に閉じ込めってしまった。しかしながら、現在においてはこのような部門内統合化にとどまることができる条件は完全に崩壊している。ビジネスをとりまく変化のサイクルは非常に速くなっているのである。

もう一つの重要な環境変化は、多様性の増大である。経済が高度化し、人々の生活が豊かになると、人々の価値観は多様化する。選択の幅の広さに価値が見いだされる時代になっている。環境が多様化することは、企業が取り得る戦略の方向性も非常に多様化してくることを意味している。多様化した環境に、自らが適応していくためには、自らの多様性を高めていく以外にない。すなわち、組織は必要多様性の法則⁹⁾に従わざるを得ないのである。

環境変化の速さと多様性の増大は、企業にビジネスのサイクルを速くすることを求める。多様な製品・サービスを速く開発し、市場を開拓し、製造・販売のサイクルを速めることによって、企業は強まる企業間競争の中で優位に立つことができる。さまざまな業種において、企業のビジネススピードが企業業績に大きな影響を与えることが指摘されている⁹⁾。ビジネススピードを上げ、環境変化に適応していくためには、分業化における逐次的なビジネスの処理形態を変えていかざるを得ない。各部門が共通の目的を認識し、仕事を同時併行的にオーバーラップさせながら進行させることによってはじめて、ビジネススピードを上げることができる。そのためには部門間の壁を取り払い、各部門が情報を随時交換しながら、協働する部門の状況を認識して動いていく必要がある。

環境に即応していくために現在の階層型の組織が大きな足枷になっていることは、すでに広く認識されている。多くの企業において階層の数を削減し、よりフラットな組織をデザインする試みが行われている。現在のツールな組織形態では現場のいきいきした情報は組織の上までは届かず、公式的なコミュニケーション・ルートによる情報伝達形態だけでは多様な情報を必要な部門間で共有することはできない。また、組織内での異質性の確保についてもその必要性の認識はかなり広まってきている。金太郎飴の人材から異能集団へとよく言われるが、実際にはなかなか思う方向へ進展しないのが実状である。組織は、継続的に策を施さない限り同質化する。部門内統合化の限界を打ち破るためには、組織の構造的な改革が必要である。

部門内統合化にとどまる企業は、結局のところ部分最適の穴に陥っている。企業は一つの有機体として外界に対して反応する必要があるが、顔は前を向き、手は横を向き、さらに足が後ろを向いていては、人はそこにじっと立ちすくむしかない。各部門

は壁の中で独自の視野で目標を立て、その実現のために自部門でできる範囲と方法で対応する。たとえ、ある部門が環境に適応する方向に対処しようとしても、有機体の一部だけが動いても他の部分動かない限りは前には進めない。一部の動きとその意図が他に伝わり、相互の議論の中で共感が生まれたとき、他も同じ方向に動きを始める。この相互理解が多く部門間で生まれてはじめて、全体が動く状況が生まれるのである。

実際に企業がステージ1から別のステージに移る場合には、統合範囲を部門間に拡大してステージ2に移る場合が多い。企業は常に競争圧力にさらされている。依然意識は効率化にあっても、競争に勝ちあるいは遅れないためにより大きな効率化を実現するには、部門を越えて仕事の仕組みを変えざるを得なくなる。このように効率をキーにして統合範囲を拡大してもそこで部門間の交流が生まれることによって、次の段階である環境適応を目指す部門間統合化に向かうことになる。

4. 環境適応を目指す部門間統合化

4.1 部門間統合化の目的と特性

部門間の統合化が目的とするところは、環境に対する組織有効性の向上にある。ステージ1からステージ2に上がった企業は、当初強烈な部門間コンフリクトに直面する。コンフリクトは統合のためになくしてはならないものであるが、ステージ1にとどまっていた時点では各部門が自分の殻に閉じ込もっていたために、深刻な部門間コンフリクトに直面することはなかった。部門間コンフリクトに直面したときに単に妥協によってその解消を図るならば、企業は統合化の道を歩むことはできない。部門間のコンフリクトを建設的に解消するためには、部門間が共通の目的を探索し、その目的を相互に認識することが重要である。部門内の統合化にとどまっていたときのように、互いが別々の視野を持ち異なる目的遂行を考えていたのでは、妥協による解決を図るしか道はない。

部門間が共通の目的を認識することは、各々の部門が部門の利益を考えていたのでは達成されず、組織として対環境にどのように対応すべきかを真剣に部門間で議論することによって始めて可能になる。視野が異なる部門は、同じ環境を見てもその見え方は異なる。相互の環境の認識を互いに交換することにより、旧来の視野とは異なるところから環境を見る態度が生まれてくる。部門としての視野でなく、組織としての視野で環境を見る訓練ができてくると、今まで見過ごしていた環境におけるさまざまな小さな変化の種が見えてくるようになる。いろいろな変化の兆候をつなぎ合わせてはじめて、市場の文脈が認識される。「戦略は市場に聞け」というマーケティングの根本原理が自覚されるようになる。対環境に対して企業として何をすべきかが部門間で共通に認識されれば、その目的を達成するための手段を共に考え、より自部門の能力に適した役割を各部門が能動的に引き受け、相互に必要な資源を配分し、情報を交換する流れができあがる。今まで高かった部門間の壁が溶解しはじめるのである。

部門間の統合化に至った企業は、戦略、計画の策定、変更において常にトップダウンだけでなくボトムアップのアプローチがとられる。トップのビジョンと現場から集約された環境の文脈情報が常におつけられ、必要に応じてビジョンそのものも修正さ

れる。一度策定された戦略、あるいは計画も不断に見直され、随時変更されていく。

しかしながら、このようなことができるためには、部門内の統合化の段階に比べてはるかに組織の階層がフラットになっていなければならない。各部門で感知された環境情報が、その本来の意味を失うことなく、上に、横に伝達される必要がある。コミュニケーションルートはフォーマルなルートと共にインフォーマルルートを通じても盛んにやりとりがなされる。部門はもはや分業化されたある部分の作業を担当する歯車ではなく、自ら全体を見通した上で、自分の持つ専門的能力に基づいて自らの役割を果たす能力センターのような機能を担うようになる¹⁰⁾。環境の変化に素早く対応するために、さまざまな部門横断型のプロジェクトチームあるいはタスクフォースチームが編成され、各部門の人間が動員される。組織の態様は、組織間の統合化の段階でのネットワーク型組織にいたる過渡期の様相をとる。部門間の統合化の段階を一言で表現すれば、速いサイクルで変化する環境に対して、組織の各部門が有機的な結合をとり、あるいは組み替えをすばやく行って適応していく姿であり、その目標は環境に対する組織の全体満足の実現にある。

表2には、部門間統合化の各ステージの特性をまとめている。

表2 部門間統合化の各ステージの特性

	ステージ2	ステージ3
ビジネスの使命 戦略	<ul style="list-style-type: none"> ・トップ・ダウン・ホリゾンタルのアプローチ ・公式的戦略策定プロセス 	<ul style="list-style-type: none"> ・環境シナリオ分析、コンテナ・エンベロープ ・環境変化に合わせた事後的な戦略の変更 ・新事業の探索
経営資源	<ul style="list-style-type: none"> ・人の重視(計画能力) ・公式戦略に基づく資源配分 	<ul style="list-style-type: none"> ・人の重視(環境探索・把握能力) ・新たな資源の取得(技術・情報) ・戦略変更に従った資源の再配分
ビジネスプロセス	<ul style="list-style-type: none"> ・ビジネスプロセスの再構成 ・部門間の接続プロセスの定式化 ・逐次処理 ・無駄の排除 	<ul style="list-style-type: none"> ・環境探索プロセスの制度化 ・情報の共有化と情報伝達のスピード化 ・同時併行処理 ・戦略変更をビジネスプロセスに反映
組織	<ul style="list-style-type: none"> ・ビジネスプロセスに合わせた組織構造の変更 ・階層の削減 	<ul style="list-style-type: none"> ・統合組織の出現 ・プロジェクトチーム、タスクフォース等の活用 ・人材センターとしての部門 ・インフォーマルコミュニケーションの重視
経営管理	<ul style="list-style-type: none"> ・MBOの積極的展開 ・TQC活動 	<ul style="list-style-type: none"> ・やわらかい統制 ・対市場有効性の評価

4.2 部門間統合化の限界

部門間の統合化であるステージ3に至った企業は、環境を見る目を持ち、また環境変化に対応する柔軟性を内部メカニズムとして備えている。しかしながら、変化した環境に適応するために自らが保持していない資源が必要になる場合で、かつその資源がオープンマーケットで調達できないものである場合には、企業がいかに環境適応の必要性を認識し、かつその方法を理解していても対応するすべがない。このような資

源とは人に付随する知識、技術であり、内部で獲得あるいは蓄積するにしても非常に時間を要するものである。企業が求められる製品・サービスが技術的に高度化し、多機能化し、複合化し、常により新しいものが求められる現在の環境において、企業が異質な資源の不足のために環境適応に失敗する可能性はきわめて高くなっている。このような事態を回避する有効な方策が、異質な知識・能力をもつ他組織との協働である。

ステージ3までは組織内の統合化は図られているが、より組織が進化して行くためにはもっと組織外に向かって自組織の開放度を高めていく必要がある。他の組織との協働が必要になった段階で協働する相手を探索するのでは、戦略の機を逃してしまうかもしれない。常日頃から、将来協働する必要性がでてくるとされる組織と、人の交流を通じてチャンネルを作っておくことが必要である。

組織の開放度を上げて異質な組織と交流することは、環境を読むことにおいてきわめて重要な意味を持つ。組織内の統合化が図られ、さまざまな部門が環境を読みその情報を組織内で交換したとしても、所詮同じ組織の中であり、かなり視点が同質化していることは避け難い。組織内のさまざまな人々が他の組織の人々と自由に交流することによって、すなわち真に異質な人々の環境の見方、考え方に触れることによって自分達の視野が大きく広がってくる。その結果、今まで環境の文脈を十分に読みとることができず、都度、環境に対応するしかなかった状態から、将来の環境変化のシナリオを描けるようになり、将来に対する自社のビジョンを描けるようになってくる。環境適応から環境創造へ向けて再度、組織はパラダイムを転換することになる。

企業が協働する相手は、異業種あるいは大学等の公的機関だけではない。今後は、同業の競争相手とも必要に応じて提携関係を持つことが多くなるであろう。特にわが国においては、有力な各企業が製品、サービス、チャンネル等で全く同質的な競争を繰り返してきた。競争が新製品の出現を促し、総需要を喚起してきた時代はそれでも良かったが、需要が飽和状態になってくると、同質的な競争は各企業のパフォーマンスを低下させるようになる。無益な競争対抗が企業の首を締めかねない時代になって、企業は同業者との間で競争と協調のバランスをとっていくことを迫られている。

企業が統合の範囲を部門間から組織間へと広げていくルートは、一般的に次の二つが考えられる。一つは依然環境維持の段階にあるステージ2の企業が、取引関係あるいは資本関係をとおして、統合範囲を企業間に拡大する場合である。もう一つのルートは、ステージ3に至った企業がより高い異質性の取り込みを目指して企業間統合化に入ってくる場合である。

5. 環境創造を目指す組織間統合化

5.1 組織間統合化の目的と特性

組織間統合化の究極的な目的は、環境創造にある。統合範囲を組織間に拡大したステージは、ステージ4、5、6の3段階があるが、ステージ4は未だ環境維持の段階にとどまっており、ステージ5は既に環境適応の段階まできている。ステージ6が環境創造の段階であり、企業組織の統合化が目指す最終的なゴールである。組織間統合の目的は、組織が複数の他の組織とネットワークを組み、自分だけでは不可能なことを

可能にしようということにある。

環境創造に至るメカニズムは次のようなものであろう。ステージ2から、より高い効率をもとめて統合範囲を組織間に広げてきたステージ4の企業は、系列化等に見られる階層的な支配と服従のネットワークを形成することが多い。このような企業間ネットワークは、そのうちに力のあるネットワークメンバーが徐々にネットワークから離脱する。残ったメンバー間ではいかにビジネスプロセスを効率化しようとも、過去の環境にフィットした戦略を維持するパラダイムのもとに作り出される製品・サービスは、現在の環境との間での不適応がだんだん大きくなり、ついには明らかなビジネスパフォーマンスの低下となって現れる。この時点ではじめて、ステージ4の企業は環境維持から環境適応へとパラダイムを転換せざるを得なくなる。

急速に変化していく環境に適時適切に対応して行くためには、ネットワークのメンバーは互いに異なる技術・能力を持つ組織であり、互いにその能力を補完しあうことが必要である。異なる能力、知識を持つメンバーがそれぞれ異なる視野から環境を見、そこで得た知見を互いに交換することによって、単に現在の環境を正確に把握するだけでなく、将来の環境をも予測できるようになる。将来の環境変化のシナリオをネットワークのメンバーが共有し、その将来の環境に対応するための方策を議論しあうことによって、企業間ネットワークの将来のあるべき姿にたいするビジョンが生まれてくる。将来ビジョンの生成は、その実現に向けてのネットワークメンバーの主体的行動を喚起する。ネットワークメンバーは互いに協働しながら、ビジョンの早期実現に向けて環境に働きかける。環境の変化を把握して、それに適応するだけの受動的行動から、ビジョンの実現に向けて環境に積極的に働きかけ、環境を変えていこうとする能動的な環境創造のパラダイムへの転換が起こるのである。

優れたビジョンを持つことは、組織間ネットワークを成長させる鍵となる。統制と服従のネットワークは、力のあるメンバーを遠ざける。しかしながら、優れたビジョンを持ち、ビジョンの共有とその実現への協力を惜しまないネットワークは、力のある新しいメンバーを引きつけ、ネットワークは拡張していく。新しいメンバーの参加は、ネットワークの異質性を保持するためにぜひとも必要である。新しい異質なメンバーを加えて、ビジョンは絶えず再検討が繰り返されていく。

ビジョンの形成には、直観の役割が非常に大きい¹⁴⁾。従来の経営戦略策定に見られるような分析的アプローチでは、人の共感を呼ぶビジョンの形成は難しい。環境適応の過程の中で、市場のさまざまな動きが実感として捉えられているならば、直観力に優れた少数のメンバーの中からビジョンの種が芽生える。ビジョンの種は多くのメンバーの議論の中から成長し、メンバーの実践的試みの結果からその正当性が補強される。ビジョンを実現するための戦略も、多くのメンバーの試行錯誤の中から有効なもの残り、組み合わせられ、事後的に戦略として形成される。ビジョンおよびその実現のための戦略は、固定化されてはならない。戦略もビジョンもその形成は試行錯誤的であり、その実行も常に検証の過程にほかならない。組織間ネットワークの環境創造力が増してくると、環境そのものが変わってくる。変わった環境は、また新たな課題と対応をネットワークに迫ってくる。環境適応の段階のように、環境は観察あるいは行為対象としての客体であり、組織は適応行為の主体であるという主・客の分離は環

境創造の段階には適用できない。環境と組織がともに影響を与えながら、ともに進化していくのが、組織間統合化の真の姿である。

このような生き生きとしたネットワークは、異質で開放度の高いメンバーと共生の理念によって支えられる。ネットワークのメンバーは、そのメンバー組織の代表者だけが他のメンバーと交流するのではなく、メンバー組織内のさまざまな部門のさまざまな階層の人達が、他のメンバー組織の多様な人々と交流する。市場の新しい方向性の発見は、このような現場レベルの担当者の交流の中から生まれてくることが多い。また、このような交流を可能にする多様な情報ネットワークの基盤を確立することも重要である。この情報ネットワークの上を流れる情報は、フォーマルな情報以外に、多様で大量なインフォーマル情報が行き来する。コミュニケーション・ネットワークは、自由で、遊びの要素も許すような気軽さが必要である。このように、ネットワークのメンバーが互いに知恵を出し合い、異質な資源を相互に供給し合うためには、組

表3 組織間統合化の各ステージの特性

	ステージ4	ステージ5	ステージ6
ビジョンの 使命 戦略	<ul style="list-style-type: none"> ・ネットワーク・リーダーの意思 ・ネットワーク・メンバーへの戦略の徹底 	<ul style="list-style-type: none"> ・ネットワーク・メンバーによる市場探査 ・市場知識の共有 ・ネットワーク・メンバーの合意による戦略形成 	<ul style="list-style-type: none"> ・異質なメンバーの知恵の交換によるビジョンの創造 ・共有するビジョンに向かう個々のメンバーの活動からの事後的戦略形成
経営資源	<ul style="list-style-type: none"> ・物/金を中心にしたネットワーク ・物/金に付随した情報 	<ul style="list-style-type: none"> ・経営資源の相互融通 ・情報/知識の共有 	<ul style="list-style-type: none"> ・異質なメンバーを積極的に取り込むことによる異質な資源のネットワーク内への取り込み ・新しい情報、知識、技術の生成
ビジョン プロセス	<ul style="list-style-type: none"> ・効率化指向 ・物/金/情報がスムーズに流れるプロセスの事前設計 	<ul style="list-style-type: none"> ・市場環境に合わせた絶えざる変更・調整 ・情報の積極的な交換と蓄積プロセスの設計 	<ul style="list-style-type: none"> ・ビジョンを共有するメンバーの自発的な相互接触からの事後的形成 ・非定型のプロセスの自由な組み込み
組織	<ul style="list-style-type: none"> ・スター型ネットワーク構造 ・取引関係にそった垂直構造(系列化) 	<ul style="list-style-type: none"> ・リンク型ネットワーク ・同業種の他メンバーの取り込み ・相互調整による役割分担 ・スラックの許容 	<ul style="list-style-type: none"> ・メッシュ型ネットワーク ・ネットワーク自体のオープン化 ・個のメンバーの独自能力によるパワー発揮と相互認知およびその利用
経営管理	<ul style="list-style-type: none"> ・始と鞭によるネットワーク管理 ・管理者派遣による支配 	<ul style="list-style-type: none"> ・キープ・アクト・テイクによるネットワークの維持 ・ネットワーク・リーダーによる管理・調整から主要メンバーによる合議への転換 	<ul style="list-style-type: none"> ・自由で多彩な情報交換の場とメカニズムの維持 ・意図的な異質メンバーのネットワークへの勧誘

組織間ネットワークの中に共生の理念がしっかりと植え付けられていなければならない。ネットワーク資源の3大要素は、開放度の高いメンバーとコミュニケーション基盤、および共生の理念にある。

部門間の統合化は、部分最適から全体満足への過程であった。最適化はその解を求める場を極めて限定することによって、始めて可能である¹²⁾。統合の範囲が部門内から部門間に広がることによって、不確実性は非常に高まる。部門間の統合化の指向するところが変転する環境への適応にあることを考えれば、最適化を求めることは不可能であり、せいぜい組織が求め得るのは満足解に過ぎない。組織間の統合化によって、解を求めようとする場はさらに広がる。一つの組織内の統合化である部門間統合化では、一時点の環境に適応することを共通の目的として各部門あるいは個人の行動を都度、組織的に調整することはある程度可能である。しかしながら、完全に独立した個々のネットワークメンバーの自律的な協働は、最適化を求めることによって引き出すことはできない。ネットワークメンバーの協働によって、全員が満足できる新しいものを生み出すときに、はじめて共生の理念がネットワークの中に醸成されるのである。

表3には、組織間統合化の各ステージの特性をまとめている。

5.2 組織間統合化の組織におよぼす影響

一つの組織が組織間の統合化に踏み込んだときには、組織自身さまざまな影響を受ける。組織のメンバーが組織間ネットワークの他のメンバー組織の個人と交流をする。トップマネジメントは、他組織のトップマネジメントとの交流の中で、世の中の流れについて異なった見方に触れる。組織構造、組織運営のあり方、組織風土の違いに対して大きな啓蒙を受ける。技術者は、他組織の技術者との共同開発の現場で新しい技術に触れるだけでなく、技術開発のやり方、技術者の市場を見る目の違い、マーケティング部門あるいは生産部門など技術部門以外の部門との協力の方法など、自分達の仕事のやり方との相違を身を持って体験する。組織のトップ、ミドルマネジメント、担当者レベルのさまざまな人々が、他の組織の人々との交流を通じて、現在までの自社の経営理念、目標、戦略、ビジネスプロセス、組織および経営管理のあり方についてさまざまな疑問を持つようになる。従来、組織内で保たれていたある種の秩序が崩れはじめる。人々は、現状の変革のためのさまざまな試みをする。まさに、異質とのふれあいから生じたカオスの出現である。

カオスの中の対話から、組織は新しい秩序を求める。新しい秩序作りは、組織の新しいアイデンティティを求めるところからはじまる。組織の多くの人々が、他組織との交流の中から自組織のあるべき姿を考える。ネットワークの中では、他の組織の良いところを模倣するだけでは、他のメンバーから尊敬を集めることはできない。ネットワークのメンバーが各組織に求めるものは、他の組織には求めることができない何かである。自組織の本当の強みは何か、それをどのように伸ばし活かして行けば、ネットワークのビジョン実現に向けて自組織が現在よりももっと貢献することができるようになるのか。組織の多くのメンバーが、この解を求めて対話を繰り返す。組織のアイデンティティについての合意が生まれると、組織の新しい秩序はそのアイデンティティ確立の過程で自ずと生まれてくる。アイデンティティを確立した組織は、他の

ネットワークメンバーから認知され、ネットワーク内でパワーを発揮できるようになる。このような組織の成員は、他組織との交流において指導的な役割を果たすことになり、組織およびその成員は深い満足を味わうことができる。人および組織の歓喜は、競争優位の中で他の組織に勝ち、より多くの経済的利得を得るところから得られるのではない。自分の能力を最大限に活かし、他のメンバーと協力し、共通の目的を達成し、その過程での個人あるいは組織の貢献が仲間から賞賛されるときにはじめて、大きな喜びを得ることができるのである。組織のアイデンティティを生み出すのは対話のプロセスであり、ネットワークのビジョンを生み出すのも対話のプロセスである。対話こそ、企業組織の統合化を実現するための鍵であるといっても過言ではない。

6. 絶えざる進化を求めて

企業の寿命は30年。言い古された言葉である。しかしながら、確かに組織が進化を忘れ、環境維持のパラダイムの中で30年の歳月を経過すれば、確実に死期を迎えざるを得ないであろう。昨今の環境変化の流れの速さの中では、その寿命はもっと短いかもしれない。組織が生き残り、しかも常に若さを保持して行くためには、絶えざる進化を求めて行かなければならない。進化とは、単なる変化とは異なる。どちらも自己が今までとは異なる存在になるということでは同じであるが、進化には上昇の意味を含む。現在の自己を基盤にして、かつ現在の自己のある部分あるいは全体を否定して、より高みに自己が生まれ変わる。単なる量的あるいは質的成長とも異なる。より高い次元への新しい自己の創造こそ、組織の進化¹³⁾である。

組織が創造的進化を遂げるためには、創造へのメカニズムを内包していなければならない。組織の自己創造を可能にするものが、自己組織性¹⁴⁾である。自己組織性とは、現在の自己を否定することによって現在の安定状態を自ら崩し、混沌の中から新しい秩序を生み出す能力を言う。有機的組織は、環境との対応の中から自己変革の必要性を関知し、自ら組織内にゆらぎを創出する。組織が自己変革の必要性を関知し、自ら組織内にゆらぎを起すためには、組織の中に異質性・多様性が存在することが必要になる。多様で異質な個が組織内に散在することによって、組織は自己革新の機会を捉えることができる。さらに、異質のぶつかり合いによって、組織内に混沌を生じさせる。組織内の異質性・多様性を高めることは、組織の自己組織性を高めることに他ならない。自己組織性によって、組織は安定と混沌を繰り返しながら、絶えざる進化の道を進んで行くのである。

組織が部門内から部門間の統合へと進んでいく。そのときに起こってくる部門間の強烈なコンフリクトは、組織を一時期混沌におとし入れるゆらぎである。組織はこの混沌の中から、環境維持から環境適応へと自己のパラダイムを変える。これは組織にとっての大きな進化である。パラダイムの転換をした組織は、戦略の形成方式を変え、ビジネスプロセスを変え、組織を変え、経営管理の方法を変える。環境適応のステージに移った組織は、環境維持のステージの組織とは異なる次元に生まれ変わっている。このような組織はまたいつの日か、自組織を組織外に開いて、組織間統合へ進む。最初は単に、環境適応のために他組織の知識・能力を利用するという功利的発想から組織間ネットワークを築いてきた組織も、他組織との交流の中で自分達とは異なる考え

方、価値観に触れることによって自組織のものの考え方、価値観がゆらいでくる。互いに従来の自身の価値観がゆらいだ組織どうしが、対話の中から新しいビジョンを創造し、環境創造という新しい次元へと生まれ変わる。この進化の流れは、組織間ネットワークにさらに異質なメンバーを次々と取り込み、その都度ビジョンを新たにしながら進んでいく。

組織は部門内、部門間、組織間へと統合範囲を拡大する。これは、組織が対境積極性の増大という自己創造を果たして行くために、異質性・多様性を高めていく過程である。しかしながら、組織は異質性を高める意図的な努力を怠ると必ず同質化してくる。この同質化を回避するためには、常に組織の開放度を上げておくことが重要である。城砦型の組織から都市国家型の組織への転換を、トップマネジメントは意図的に行って行かなければならない。また、組織に自由な個を認める風土がなければ、せっかく取り込んだ異質に対してそれを同質化しようとする強い力が働く。企業組織の統合化の実現にとって、自由な個を認める精神こそ欠くべからざる要諦である。

7. おわりに

統合化による組織進化は、組織のパラダイム転換を必要とするが故に、長く険しい道のりである。本論文で言う組織間の統合化の段階に踏み込んだところは、比較的小さな組織において散見し得るにすぎず、大きな組織では見あたらない。部門間の統合化に至った企業も極めて少ないのが実情であろう。ただ、この方向に向かって多くの企業でさまざまな試みが行われている。我国の企業は、過去、先進欧米諸国の経営システムを導入し、それを日本化することによって成功の階段を駆け上がってきた。既に我々の前に手本とする先例がないと言われる今日において、さらに今までの延長線上には明るい未来を想起することが困難な状況において、我々は敢然として統合化の道を模索するしかない。ネットワーク型組織構造にしても、慣れ親しんだ階層型の組織構造とは大きく異なるために、実際にそのような組織の運営が可能であろうかと言う不安が生まれるのは至極当然である。統合化された組織において個人に要求される能力水準は比較にならないほど高い。しかしながら乗り越えるべき障壁が高くとも、1歩1歩前に進むことが重要である。何故ならば統合化による組織進化は、真に人間が働く喜びを味わえる世界を築く道だからである。

筆者は現在、経営情報学会に所属している。本稿は、その学会で昨年度まで行われた統合化情報システム研究部会における活動での個人的成果をまとめたものである。この研究部会には、学者、システムベンダー、ユーザ企業に所属する多様な人々が参加していた。組織論に関してほとんど無知であった筆者が本論文をまとめるに至ったのも、部会の異質な人々との密度の濃い交流があったことに大きく依存している。まさに、異質との遭遇による個人の進化を経験したと言っても過言ではない。研究部会のメンバーの方々、特に親身にご指導頂いた神奈川大学海老澤栄一先生、さらに筆者が学会活動に時間を割くことを日頃応援して下さいる会社の上司、同僚の方々に厚く御礼申し上げる次第である。

- 参考文献 [1] 海老澤栄一, 組織体における統合化情報システムの概念的フレームワーク (1), 神奈川大学国際経営論集, No.4, P.23, 1993.
- [2] 「対境」という言葉は, 山城の言う「対境関係」の中で使用されている. 山城の対境関係とは, 企業に対する利害者集団との関係を意味しており, 本論文で使用した意味よりも広い概念である. 本論文で言う対境とは, 企業を取りまく市場, 顧客, 経済・社会環境と企業の相互関係を意味している.
山城章, 新講経営学, PP.93~99, 中央経済社, 1968.
山城章, 経営原論, PP.201~207, 丸善, 1970.
- [3] 城砦型組織と都市国家型組織は, バダラッコの城砦パラダイムと都市国家パラダイムに依拠している.
バダラッコ J.L. Jr. (中村元一・黒田哲彦訳), 知識の連鎖, PP.4~7, 20~24, ダイヤモンド社, 1991.
- [4] 嶋口充輝, 戦略的マーケティングの論理, P.4, 誠文堂新光社, 1983.
- [5] ジョンソン他は, インクリメンタルな戦略的意思決定の方法として組織が問題を発見するプロセスを詳しく説明している.
Johnson G. = Scholes K. = Sexty R.W., Exploring Strategic Management, PP.38~44, Prentice-Hall Canada Inc., 1989.
- [6] 嶋口充輝, 統合マーケティング, PP.23~29, 日本経済新聞社, 1986.
- [7] 今井賢一=金子郁容, ネットワーク組織論, PP.80~86, 岩波書店, 1988.
- [8] アシュビー W.R. (篠崎武訳), サイバネティクス入門, PP.126~127, 255~256, 259, 261, 宇野書店, 1967.
- [9] 松島克守, CIM 製造業の情報戦略, PP.134~138, 工業調査会, 1987.
- [10] サバージ C.M. (島戸一臣, 梅村守, 奥田省三訳), 第5世代マネジメント, PP.251~254, 294~295, 朝日新聞社, 1990.
- [11] 従来, 意思決定, 戦略立案等のマネジメント理論において, 「直観」は合理的でないという理由からその重要性を低く見られてきた嫌いがある. 近年, マネジメントにおける直観の役割を見直す論者が多くなっている.
ミンツバーグ H. (北野利信訳), 人間感覚のマネジメント, PP.87~121, ダイヤモンド社, 1991.
- [12] サイモン H.A. (稲葉元吉, 吉原英樹訳), 新版システムの科学, PP.329~334, パーソナルメディア社, 1987.
- [13] 海老澤栄一, 組織進化論, PP.243~264, 白桃書房, 1992.
- [14] ヤンツ E. (芹沢高志, 内田美恵訳), 自己組織化する宇宙, 工作舎, 1986.

執筆者紹介 ト部正夫 (Masao Urabe)

1947年生, 1970年九州大学法学部卒業, 1971年日本ユニシス(株)入社, 以降, システムエンジニアを経て現在, 客先の経営およびシステム・コンサルティングに従事, 1985年慶應義塾大学大学院経営管理研究科修士課程修了, 経営学修士, 中小企業診断士, 経営情報学会会員, 1993年より神奈川大学経営学部非常勤講師として, 経営情報論, 経営情報システム論を担当, 著書:『経営を見る目』(日経BP社, 共著)『統合化情報システム』(日科技連, 共著)他, 訳書:『プロセス・イノベーション』(日経BP出版センター, 共訳)『情報システム組織論』(オーム社, 共訳), 現在, I & C マーケティング部ソリューション2室に所属.



OSI-TP ソフトウェアの3機種 (2200, U6000, Aシリーズ) への実装

The Implementation of OSI-TP Software for Three Computer Models (the 2200, U6000 and A Series)

山 野 茂 美, 末 永 敏 昭
河 野 典 久, 石 崎 達 也

要 約 OSI-TPはX/OpenのDTPモデルに準拠した分散トランザクション処理システムである。分散トランザクション処理アプリケーションがOSIあるいはTCP/IPネットワーク内の他のアプリケーションと通信することを可能にしている。本稿では2200に実装したソフトウェアをU6000およびAシリーズへ実装した時の開発経緯、実装方法、効果を概説する。

Abstract OSI-TP is a distributed transaction processing system which conforms to the distributed transaction processing reference (DTP) model designed by X/Open. It allows distributed transaction processing applications to communicate with others that are processed by other systems via the OSI or TCP/IP network.

This paper gives a rundown on the milestones of development and the way of implementation involved in the accomplished effort to newly port OSI-TP software already embedded in the 2200 Series onto the U6000 and A Series products in addition to the resultant effects.

1. はじめに

オープン環境において、異機種システムの相互接続性の確保は、重要な課題の一つである。システム間相互接続の標準プロトコルとして、ISO (International Organization for Standardization)およびCCITT (International Telegraph and Telephone Consultative Committee)の協調のもと、標準化されてきたOSI (Open Systems Interconnection)は良く知られている。OSIの基本標準¹⁾は、国際規格または国際勧告の形態ではほぼ完成され、ファイル転送のFTAM (File Transfer and Access and Management)や電子メールのMHS (Message Handling System)においては、実用化の時期を迎えて数年が経過している。日本国内における情報処理業務の多くは、トランザクション処理の形態を呈しており、トランザクション処理用のOSI標準規格DTP (Distributed Transaction Processing: 分散トランザクション処理)への期待は大きい。

オープン環境のもう一つの課題は、サービス・インタフェースの標準化である。サービス・インタフェースの標準化は、ソフトウェアの移植性を高め、システム開発の生産性および信頼性を向上させる。UNIX*を標準化するためのPOSIX (Portable Operating System Environment)、共通のアプリケーション環境を規定するX/Open**

* UNIXはX/Openカンパニーリミテッドがライセンスしている米国ならびに他の国における登録商標である。

** X/Open: ソフトウェアベンダーとユーザから構成される国際組織。オープンシステム環境の標準を提案。

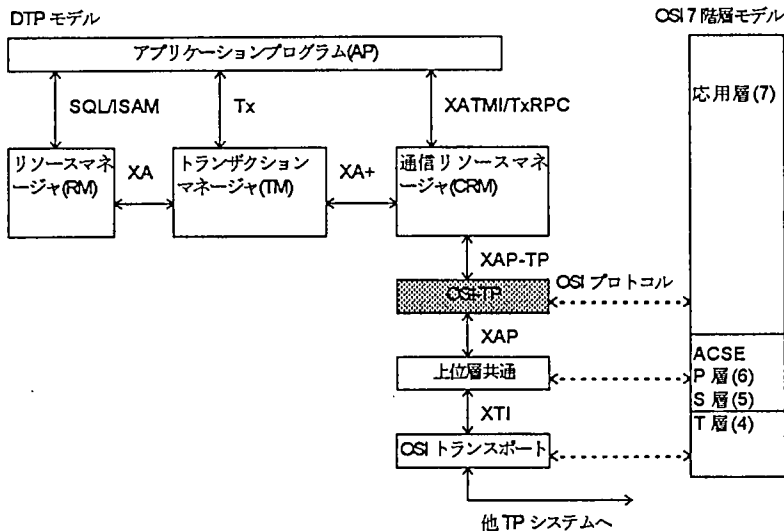
などが、この課題に取り組んでいる。サービス・インタフェースの標準化は、ユーザが開発するプログラムにとどまらず、計算機メーカーが新たに製品を開発する上でも有益である。

ISOは分散トランザクション処理プロトコルを相互接続性の観点より規定^[2]している。これに対して、X/Openはアプリケーション環境の観点より、分散トランザクション処理に関するDTPモデルを定義し、データベースなどトランザクション処理に係わる機能要素とのインタフェースも含めて明確に定義^[8]している。

通信ソフトウェアの製品開発において、実装するプロトコルそのものも重要であるが、X/Openなどにより規定された事実上の標準を、ユーザ・インタフェースとして提供することは、オープン環境において重要なことである。

2. X/Open DTP モデル

X/Openで定義されている分散トランザクション処理に関するDTPモデルとインタフェース、およびOSIで規格化されている通信プロトコルとの関係を図1に示す。今回の開発の対象となった部分は図1の中のOSI-TP部分である。



SQL/ISAM : Structured Query Language /Indexed Sequential Access Method
 XATMI/TxRPC : X/Open Application-Transaction Manager Interface/TxRPC
 ACSE : アプリケーション制御サービス要素
 XAP-TP : ACSE/Presentation : Transaction Processing API
 XAP : ACSE/Presentation Service API,
 XTI : X/Open Transport Interface
 P層 : プレゼンテーション層, S層 : セッション層, T層 : トランスポート層

図1 X/Open DTPモデルとインタフェース

X/Open DTPモデルは、分散トランザクション処理に関してアプリケーション・プログラム(AP)、トランザクション・マネージャ(TM: Transaction Manager)、リソースマネージャ(RM: Resource Manager)、通信リソースマネージャ(CRM: Communication Resource Manager)、の四つの機能要素を定義し、それぞれの要素間のインタフェー

スをC言語関数呼び出しの集合により、定義している。

- 1) APはTMが提供するTxインタフェース*を使用して、トランザクションの開始、終了およびトランザクションをコミットあるいはロールバック**するかの決定を通知^{[10][12]}する。

APはトランザクション内で、データベースなどのRMが提供するインタフェースを使用して、そのリソースにアクセスする。APはCRMが提供するインタフェースを使用して、他TPシステム上のAPとの会話を行う。

- 2) TMはトランザクションの開始、終了およびコミット、ロールバックを制御し、APに対してTxインタフェースを提供する。

TMはRMに対してXAインタフェース***^[11]、CRMに対してXA+インタフェース****^[13]を使用して、トランザクションのコミット、ロールバックを調整する。分散されたりリソースの同時更新を矛盾なく行うための手順として知られる2相コミットメントが、TMとRM、CRMの間でそれぞれ行われる。

- 3) RMはデータベースあるいはファイル・システムなどリソースを制御する機能要素で、APに対してそれぞれのサービス・インタフェースを提供する。SQL、ISAMなどのインタフェースがX/Openにより定義されている。RMはXAインタフェースによるTMからの情報により、リソースをコミットあるいはロールバックする。

- 4) CRMはAPが他のTPシステム上のAPと会話するためのサービスを提供する。X/Openは以下の通信機能要素とそのインタフェースを定義している。

XATMI	-クライアント・サーバ型会話機能 ^[9]
TxRPC	-遠隔プロシージャ呼び出し ^[10]

CRMはXA+インタフェースによるTMからの情報により、他TPシステムへコミットあるいはロールバックを通知する。X/Openは、OSIで規格化されているTPに対して、XAP-TP^[6]インタフェース*****を定義しており、CRMはこのインタフェースを使用して、他TPシステムへの通知を行う。

以上のインタフェースがX/Open DTPモデルの枠組みの中で定義されている。これに加えて、X/OpenはOSIで規格化されている他の通信エンティティに対しても、それぞれインタフェースを定義している。

- 5) XAP^[7]はOSIの上位層共通（セッション層、プレゼンテーション層、ACSE）に対するインタフェース
- 6) XTIはOSIの4層に位置するトランスポート層に対するインタフェース

* Txインタフェース：アプリケーション・プログラムの広域トランザクション開始および終了を可能とする。
 ** コミット：トランザクションを更新、ロールバック；トランザクションを更新前の状態に戻す。
 *** XAインタフェース：広域トランザクションのデータベース同期処理用に二相コミットを提供している。
 **** XA+インタフェース：TPシステム上のコンピュータ間のトランザクションの同期を可能にする。
 ***** XAP-TPインタフェース：CRMがTPシステム上のローカルとリモート・アプリケーション・プログラム間の通信を開始およびサポートするため使用される。

3. ユニシスにおける OSI-TP の開発

OSI-TPが制御を提供するトランザクションには応用支援トランザクション^{*[3]}と提供者支援トランザクション^{**[4][5]}がある。

FTAMをはじめとする他のOSIプロトコルが、1対1のプロトコル動作を取り扱うのに対して、OSI-TPの提供者支援トランザクションにおいては、1対Nのプロトコル動作を取り扱う。この処理の複雑さからDTP規格の国際標準の制定までに、長時間を要した。

弊社はこの間2200シリーズ上でTPの開発を行い、OSI-TPとしてリリースした。最終的に、そのインタフェースの重要性から、米国ユニシスはUA(Unisys Architecture)の中のオンライン・トランザクション・システム(Open/OLTP)として、X/Open DTPモデルとインタフェースを全面的に採用した。

その結果、DTPモデルのTMとCRM(XATMI)を実装するソフトウェアOpen/OLTPとOSI-TPの開発が2200シリーズだけではなく、A、U6000シリーズの3機種において実装することになった。米国ユニシスより、2200、A、U6000シリーズのOSI-TP同時リリース計画が発表され、米国ユニシスからの協力要請を受け開発に参加した。OSI-TPのすべての機能を実装し、ユーザ・インタフェースに、X/Open XAP-TPインタフェースを採用した。このときISO規格は、国際標準に制定された。

4. 2200シリーズへの実装

弊社は、トランザクション・システムの開発にあたりX/Open DTPモデルを採用した。上位アプリケーション・インタフェースとしてはX/Openで規定されているXAP-TPを採用した。標準化されている規格を採用することによりインタフェースの設計に要するマンパワーを削減できる。また、トランザクション処理をサポートするソフトウェア全体にも同様に標準化を取り入れた。OSI-DTP^{***}の上位ソフトウェアとしてOLTP-TM2200^{****}および関連ソフトウェア(データベースおよびOS)が同時に開発された。OLTP-TM2200およびデータベース・ソフトウェアを含めて標準インタフェース(X/Openで定義したもの)を採用した。標準化を取り入れた結果、OLTP-TM2200および関連ソフトウェアの開発も同時に開始された。

本章では主に2200シリーズにおけるOSI-DTPの開発処理について記述する。

4.1 モジュール構造

OSI-TPの開発は3機種共通で行うため、なるべく各システムに依存しない部分を多く作り出すように機能のモジュール化を行った。このようにすることにより、その部分の開発に要するマンパワーを削減できる。

OSI-TPプログラムは、大きく次に示す四つのモジュールから構成されている(図2)。

■共通モジュール

■システム・インタフェース・モジュール

* 応用支援トランザクション：OSI-TPサービス利用者が、コミットおよびロールバックの制御に責任を持つ。

** 提供者支援トランザクション：OSI-TPサービス提供者が、コミットおよびロールバックの制御に責任を持つ。

*** 2200では日本ユニシス製OSI-TPをリリースしており、区別するために米国ユニシス製をOSI-DTPと呼ぶ。ユーザアプリケーションのAPIに互換性はない。

**** 2200のOpen/OLTPプロダクトで、OSI-DTPと協調してトランザクション処理を行う。

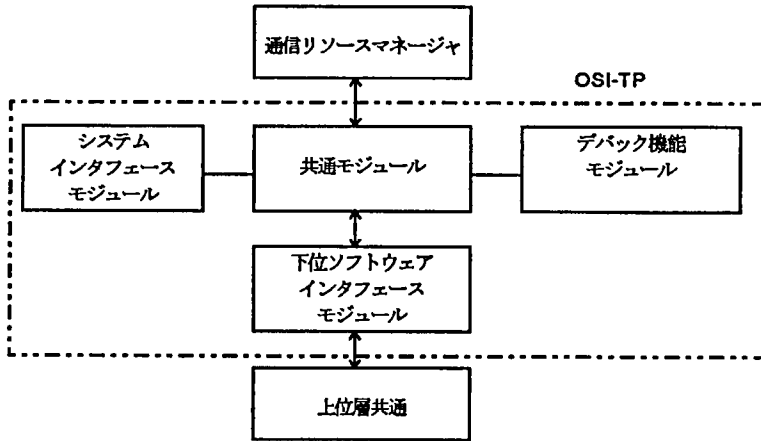


図2 OSI-TPモジュール構造

■下位ソフトウェア・インタフェース・モジュール

■デバック機能モジュール

これら四つのモジュールについて、簡単に説明する。

4.1.1 共通モジュール

共通モジュールは、上位アプリケーション・インタフェースおよびOSI分散トランザクション処理の規格で定義されているプロトコル・マシンの部分である。

OSIのプロトコル・マシン部分は、約100ページに及ぶ状態遷移表など量としては膨大で複雑かつ難解である。したがって、この部分を共通モジュールとして作成することは、開発チームにとって重要であった。また、テーブル・レイアウトも当然ながら共通に設計した。

4.1.2 システム・インタフェース・モジュール

システム・インタフェース・モジュールは、特にOSなどそのシステム固有の処理を必要とする部分である。

システム・インタフェース・モジュールには主に次のようなものがある。

- ロック機構
- アクティビティ管理
- メモリ管理
- オペレータ操作インタフェース
- コンティンジェンシ処理など

システム・インタフェース・モジュールは、各システムで最も異なる部分である。この部分は、外部インタフェースつまりシステム・インタフェース・モジュールを呼ぶ部分を共通化し、実際にシステム関数を呼んでいる部分のみ別モジュールにしている。このようにすることにより別のシステム関数の開発が発生した場合、そのモジュールを見れば何が必要かがすぐわかる。2200シリーズでは、システム関数を呼ぶ方法としてCの標準ライブラリを使用する。標準ライブラリにない部分が若干存在する。

この部分はMASM*で作成した。

4.1.3 下位ソフトウェア・インタフェース・モジュール

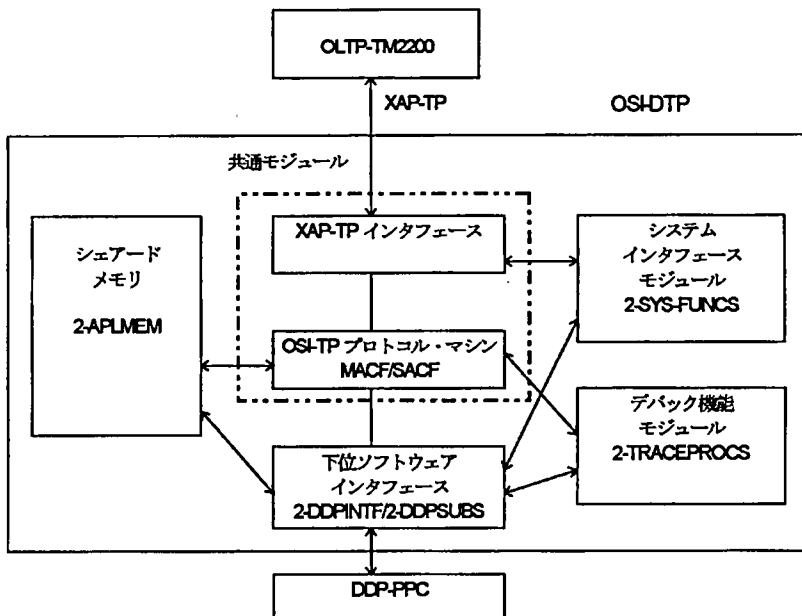
この部分は各機種により異なっている。つまり、すでにこの部分をサポートするソフトウェアは各機種とも他OSIプロダクトのためにリリース済みであるため、そのソフトウェアが提供しているインタフェースに合うように作成する必要がある。

2200シリーズでは、下位ソフトウェアとしてDDP-PPC (Distributed Data Processing Program to Program Communications)が存在する。DDP-PPCはOSI7層の内、5層(セッション)、6層(プレゼンテーション)および7層の共通部分(ACSE)を実装しているソフトウェアである。先に記述した通りこのソフトウェアにはOSI-DTPに必要な新しい機能を追加するが、インタフェースは従来の形を使用しなければならない。標準インタフェースを使用するという考えからすると本来は、X/Openで定義されているXAPインタフェースを使用するべきであるがこのような理由によりできなかった。

2200シリーズでは、OSI-DTPの構成情報をDDP-PPCが提供するコンフィグレーション・サービスを使用して作成/使用する。構成情報を定義する文の構文は、3機種共通にした。

4.1.4 デバッグ機能モジュール

デバッグ機能は、各モジュールの軌跡をトレースしたり、記憶領域の内容または障



MACF/SACF : Multiple Association Control Function/Single Association Control Function
 2-APLMEM, 2-SYS-FUNCS, 2-TRACEPROCS, 2-DDPINTF/2-DDPSUBS は 2200 OSI-DTP ソフトウェア・モジュール名。

図3 2200シリーズのOSI-DTP構造

* MASM ;メタ・アセンブラ言語

害情報を保存したりする機能である。4.1.2項のシステム・インタフェース・モジュールと同様に、個々のシステムで異なっている。しかしデバッグ機能呼び出すインタフェースを統一したため、呼び出す側ではシステムの差異を考えなくてもよい。

2200シリーズでは、これらの情報をDDP-PPCが提供するログ・トレース機能を使用して保存している。DDP-PPCのログ・トレース機能は、使用しているシステムに影響を与えず採取/編集ができるため有効である。また、DDP-PPCが提供している各層の情報も同時に採取できるため、広範囲の情報を基にデバッグができる。

以上を考慮した2200シリーズOSI-DTPの構造を図3に示す。

4.2 使用言語

2200, U6000, Aシリーズの3機種同時開発ということで、コスト/期間等の観点から各システムで使用する言語も統一した。各システム共通で使用する言語としてC言語を採用した。しかし、C言語も各機種で多少仕様が異なっているため、コーディング上、条件付きコンパイル指令*を使用して機能を区別している。

5. U6000シリーズへの実装

U6000シリーズへのOSI-TP実装の方針は、

- 1) OSI-TPのU6000シリーズ上の実装はユーザ空間においてなされる。
- 2) 2200シリーズの共通モジュールを完全に移植する。
- 3) U6000シリーズに特有なシステム・インタフェース・モジュールを新規作成する。
- 4) 下位ソフトウェア・インタフェースとしてX/Openが定義したXAPインタフェース¹⁾を新規作成する。
- 5) デバッグ・モジュールを新規作成する。

の上記5点に集約できる。

本章では各項目にしたがってU6000シリーズへの実装について述べる。

5.1 OSI-TPのユーザ空間での実装

トランザクション・システムにおいて効率性は重要である。効率面からみればユーザ空間での実装よりは、カーネル空間での実装がよい。

開発当初においては図4で示すようにOSI-TPをカーネル空間で実装することを検討したが、以下に述べる理由で、ユーザ空間での実装を選択することになった。

- 1) カーネル空間での実装はOSI-TPのストリーム・モジュール化を意味しており、一般のユーザ空間での実装に比べてプログラミングやデバッグが難しくなる。
- 2) ストリームはシステムV独特のものであり、BSD系のUNIX**ではサポートされない。すなわち他社UNIX機への移植性を大きく損なう。
- 3) OSI-STK***との関係上、ACSEを含めてプレゼンテーション層自体を独自に作成しなければならない。このことは大幅な工数の増加を意味する。
- 4) さらにOSI-STKがサポートするXAPインタフェースを直接使用することがで

* 条件付きコンパイル指令: #if/#else/#endifなど

** BSD系のUNIX: UnixのOSは大別するとUCLAパークレイ版(BSD)とATT版(システムV)に分けられる。

*** OSI-STK: OSI-Stackの略。OSIの上位層および下位層の機能を提供する。

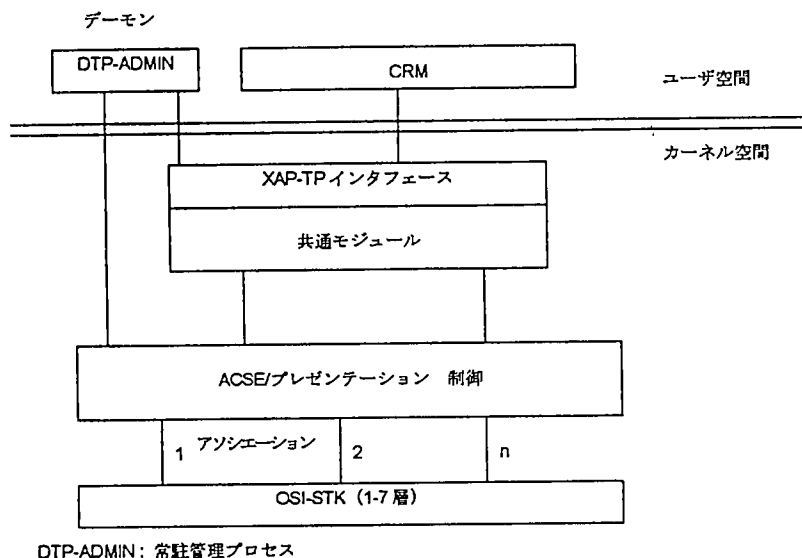


図4 カーネル空間での実装

きなくなる。カーネル空間での実装による効率面での利点は確かに捨て難いものではあるが、開発コスト、工数、移植性を考慮した結果、OSI-TPはユーザ空間での実装となった。

その後、米国ユニシスは将来のWindows NT*への移植も考慮し、OSI-STKとの機能分担を行い、ユーザ空間においてULS (TransIT Connect Upper Layer Service)モジュールを実装した。OSI-STKは1層から4層までのOSIトランスポート以下の部分、ULSは5層から7層までの上位層共通部分の機能分担となり、OSI-TPはULSに下位インタフェースを取るようになった。最終的にOSI-TPの実装は図5で示されるような位置付けとなった。

5.2 共通モジュールの移植

U6000シリーズでのOSI-TP実装段階において2200シリーズで開発した共通モジュールをそのまま使用した。この採用は、開発工数の大幅な削減に寄与した。

5.3 システム・インタフェース・モジュールの作成

5.3.1 シェアード・メモリ

シェアード・メモリに関する機能は2200シリーズでは、`salloc`, `csalloc`, `sfree`といった関数より実装されている。UNIXでのメモリー・シェアリングはIPC (Inter Process Communication)シェアード・メモリ・セグメントを使用することによってのみ可能である。2200シリーズと異なり、シェアード・メモリの多重プールのためのアドレス要求は存在しない。その代わりに一つのシェアード・メモリのブロックを使用するので、2200シリーズの関数は直接使用できない。そこでU6000シリーズでは、関数名

* WindowsNTは米国Microsoft社の商標である。

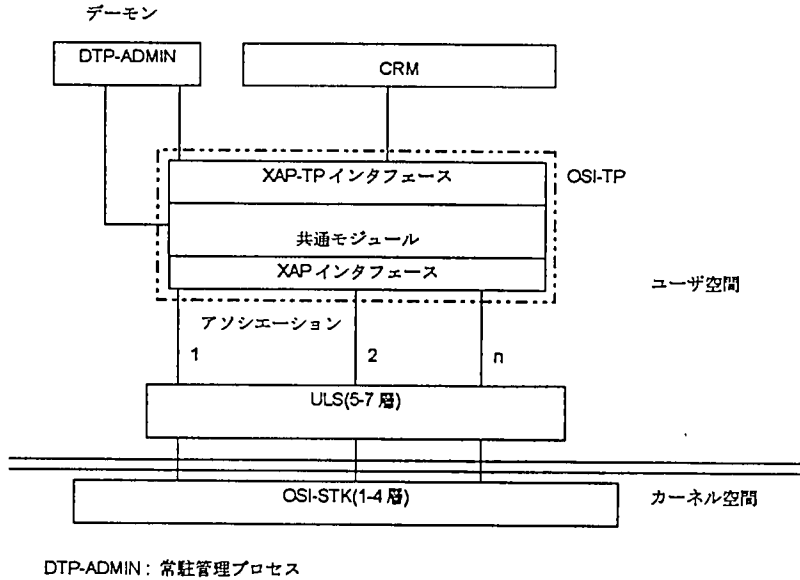


図5 ユーザ空間での実装

および引き数の変更は行わず内容をUNIXシステム・コール*に置換した。

5.3.2 ロック機構

ロック機構に関する機能は2200シリーズでは、ts_lock, ts_unlock, ts_unlock_waitそしてts_unlock_actといった関数より実装されている。UNIXでのロック機構はすべてIPCセマフォを使用することで行うことが望ましい。シェアード・メモリと同様に2200シリーズの関数を直接使用することができないため、U6000シリーズでは関数名および引き数の変更は行わず内容をIPCセマフォを用いてのロック機構へ書換えを行った。

5.3.3 下位ソフトウェア・インタフェース (XAPインタフェース) の作成

U6000シリーズ上のOSI-TPからみて下位ソフトウェアはULSである。ULSはX/Openが定義したACSE/プレゼンテーション層のユーザ・インタフェースとしてXAPを提供している。したがって2200シリーズのDDP-PPCインタフェースをU6000シリーズのXAPインタフェースに書き換えた。

XAPインタフェース作成にあたっては、DDP-PPCインタフェースを構成するモジュール (DDPINTF.C DDPSUBS.C) の関数名および引き数には一切の変更を加えず、DDP-PPCシステム・コール部分をXAPシステム・コールへ変更することで実現している。

5.3.4 デバッグ・モジュールの作成

2200シリーズのログ・トレース機能はDDP-PPCが提供しているが、U6000シリーズでは代替える機能がないため、すべて新規に作成する必要があった。実装したログ・トレース機能としては次の二つがある。

* UNIXシステム・コール: shmget()など

1) OSI-TPトレース

共通モジュールが出力するOSI-TP内部の情報と同様。

2) XAPプリミティブ・トレース

XAPシステム・コールごとの情報

いずれのトレース・ファイルもUNIXバイナリ・ファイルを複数個サイクル使用することで、トラブル時およびデバック作業時の解析ができる。

6. Aシリーズへの実装

AシリーズにもC言語が用意されており、2200 OSI-DTPのコードが使用できる。しかし、Aシリーズのシステム主言語はALGOLであり、システム関数など細かいインタフェースはC言語には用意されていない。また、マシン・コンセプトの違いがあり、すべてのコードをそのまま使うことはできない。問題となった主な項目は次である。

1) システム・インタフェース・モジュール関連

- ・シェアード・メモリが使えない
- ・C言語からのプロセス起動ができない
- ・ロック機構が2200シリーズと異なる
- ・CとALGOLのデータ構造が違う

2) 下位ソフトウェア・インタフェース・モジュール関連

- ・下位のインタフェースはポートファイル・インタフェースである

3) デバッグモジュール関連

- ・ログトレース機能の新規作成

これらの解決方法を検討した結果、次に示すような方法が見つかり、AシリーズのOSI-TPはC言語とALGOL言語を組み合わせれば実現可能となることが判明した。

本章では、各項目について説明する。

6.1 システム・インタフェース・モジュール

6.1.1 シェアード・メモリ

この機能はOSI-TPのモジュール間でのデータ授受に非常に重要である。Aシリーズにはこの機能がないが、この構造を変更することは作り直しと同じになる。

解決方法はシェアード・メモリでデータ授受を行っているモジュールを1本のプログラムとして組み上げることである。しかし、最初から1本のプログラムとして書くのであればあまり意味が無く、2200シリーズのコードを流用することが開発方法の目的に照らして重要である。

BINDER*というユーティリティを用いて複数のプログラムを1本のプログラムにできることが判明した。従来からALGOLとCOBOL用には提供されていたが、新たにC言語も加わり使用可能となった。

6.1.2 Cコードからのプロセス起動

OSI-TPは複数のプロセスが起動して協調動作するが、現在のAシリーズのCコー

* BINDER: 小さなモジュールで書いたプログラムを、1本の実行モジュールに組み替えるユーティリティ。

ドにはプロセスを起動する機能がない。

解決方法はALGOLコードをCコードから呼び、そのALGOLコードからプロセスを起動してCコードに戻る。また、起動したプロセスがある処理を終了するまで呼び出したCコードの処理を進めてはいけないうきなどは、呼び出されたALGOLコードでプロセスを起動しイベント待ちにする。起動されたプロセスは処理後イベント待ちになっているALGOLコードにイベントを送りALGOLコードを実行する。結果としてALGOLコードが終了しCコードへ戻る。

6.1.3 ロック機構

2200シリーズ独特のロック値を使用しているが、OSI-TPの共通モジュール部分に影響するのでAシリーズもその値を使用し、ロック機構をつくった。しかし、ALGOL部分で共通モジュール部分に依存しない部分では効率を考えALGOL専用のPROCURE, LIBERATE構文を使用した。

6.1.4 CとALGOL間のデータ移送

Cのデータ構造はデータ、ポインタ（アドレス）などが構造形式で格納されているが、ALGOLのアレイ構造と異なり、CのデータをそのままALGOLコードに渡すことができない。特にポインタで示された先のデータ構造が渡せない。これはX/OpenでのXAP-TPインタフェース部分で問題になる。X/OpenのインタフェースはすべてCデータ構造で定義されており、OSI-TP共通モジュールのXAP-TPインタフェース部分はCデータ構造になっている。これを使用することが大前提にある。

OSI-TPの上位にはOpen/OLTPがある。Aシリーズの場合、このOpen/OLTPはシステムに依存しているので、新規に作られるのではなく既存ソフトウェアにX/Open DTPモデルの機能を追加する方法になる。ここで、既存プロダクトのデータ・フォーマットをXAP-TP構造にデータを変換しなければならないという問題がある。このOpen/OLTPとOSI-TPとのデータ授受のためにOAM (OSI Access Module) というモジュールを用意しXAP-TPのデータ構造へ変換する。

OAMモジュールは2200シリーズではOpen/OLTPのCRMに含まれているが、AシリーズではOSI-TPに含めている。その理由は、AシリーズのCRMはALGOLコードであり、XAP-TPのデータ構造をとるのが困難であり、シェアード・メモリ機能がないのでデータ構造をそのまま渡すことができない。そこでALGOLデータ・フォーマットをOSI-TPに渡して、ここでXAP-TPのデータ構造に変換する方法を採った。したがって、AシリーズのOAMはOSI-TP内に存在し、ALGOLデータ構造とXAP-TPのCデータ構造の変換を行う。これでCRMとのデータ授受はALGOLフォーマットで可能となった。

6.2 下位ソフトウェア・インタフェース・モジュール

下位の実現は他のプロダクト(FTAM, MHS)でも共有されるものでシステムに依存する度合いが大きい。Aシリーズでは下位部分(ACSE以下)はMCP (Master Control Program)*およびBNAOSISUPPORT**ライブラリで実現されており、インタ

* MCP: Aシリーズのオペレーティング・システム。

** BNAOSISUPPORT: BNAやOSIネットワークをサポートするシステム・ソフトウェア。

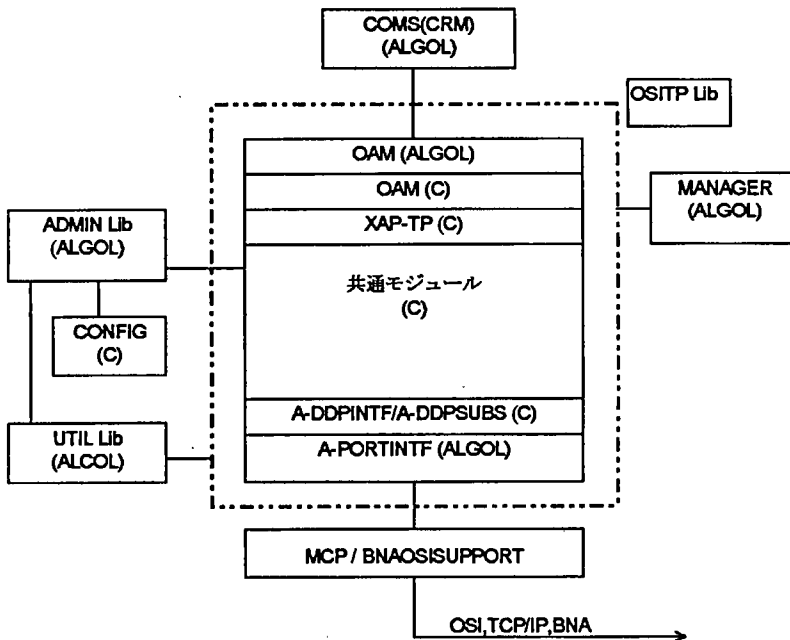
フェースはポートファイル*を使用する。このインタフェースはX/OpenのXAPインタフェースの部分であり、C構造で定義されている。このポートファイル・インタフェースはALGOL用のみ提供されているので、XAPをAシリーズにそのままの形で実現することはできない。同一機能を提供できるように独自にALGOLコードで作成した。ポートファイル・インタフェース・モジュールは2200シリーズのDDP-PPCの一部を実現していることになる。

6.3 デバッグ・モジュール

2200シリーズのログトレース機能はDDP-PPCが実行しているが、AシリーズではOSI-TP内に組み入れなければならない。MCP, BNAOSISUPPORTライブラリのロギングとの同期をとることからOSI-TPのログ・トレースをシステム・ログファイルに格納することにした。これにより下位層との一体のログが採取できデバッグも容易になった。ログファイルはシステム管理下のファイルなのでこのモジュールはALGOLコードで作成した。

6.4 結 果

これらの問題を解決し、AシリーズOSI-TPはC言語とALGOL言語の混在で作成した。OSI-TPの中核となるプロトコル・マシン部分は2200, U6000, Aシリーズとも共通であるので、この部分の開発期間、コストとも節約されている。この中核部分の回りにシステムに親和性のあるALGOLコードでインタフェースを作成することによ



(ALGOL), (C) は記述言語を表す。
 その他は、ソフトウェア名あるいはモジュール名

図6 AシリーズOSI-TP構造

* ポートファイル：BNA,OSI,TCP/IPなどで使用されるネットワークとの入出力をアプリケーションがファイルのread,writeで扱えるようにしたもの。

り、短期間に品質の良いプロダクトが作成できた。図6にAシリーズOSI-TP構造を示す。共通モジュールをALGOLコードが囲み、システムとのインタフェースを実行する構図になっている。

7. ソースコード制御

3機種(2200, U6000, Aシリーズ)の共通モジュールのほかに、3機種それぞれの専用モジュール、2200とAの共通モジュール、AとU6000の共通モジュールがある。各機種でそれぞれのOSI-TPコードを作るのであるが、別々にソースコードを保持していたのではモジュールに修正が発生したとき、修正漏れなどが発生して、3機種で品質にばらつきが起こり、品質低下につながる。そこで、1か所でソースコードを制御しなければならない。各機種での制御方法と現在環境を検討した結果、U6000シリーズでソース・コード制御をすることになった。

専用コードはそれぞれのマシンで管理することにし、他の共通コードのみU6000シリーズで管理することにした。管理する対象は3機種共通コードだけではなく、2機種共通コードも対象とする。どの機種用のコードかを簡単に判別できるように、ファイル名に名前付けの規則を設定した。それぞれのファイル名称とそのファイル数、ソースコード行合計数は表1のようになった。

2200シリーズ用コードは表1の(3)+(4)+(7)で、U6000シリーズ用のコードは(2)+(6)+(7)で、Aシリーズ用のコードは(1)+(4)+(6)+(7)でそれぞれ作る。

表1 ファイル情報

	ファイル名称	役 割	ファイル数	行数
(1)	A-xxxxxxxx	A専用コード	75	57339
(2)	u-xxxxxxxx	U6000専用コード	153	39573
(3)	2-xxxxxxxx	2200専用コード	57	21971
(4)	2A-xxxxxxxx	Aと2200共通コード	10	1924
(5)	2u-xxxxxxxx	U6000と2200共通コード	0	0
(6)	ua-xxxxxxxx	AとU6000共通コード	36	16341
(7)	xxxxxxxxxxx	3機種共通コード	91	97528

各機種のファイル数と総行数、共通コードおよび専用コードの占める割合は表2のとおりである。

表2 ソースコード行数

機 種	ファイル数	総行数	共通コードの占める割合
2200シリーズ	158	121423	81.9%
U6000シリーズ	280	153442	74.2%
Aシリーズ	212	173132	66.9%

8. テスト環境

マシン環境としては、2200, U6000シリーズはローカル折り返し機能があるので、一台で機能テストを行うことができた。外部接続ではLANを使用した。

テストプログラム環境は、C言語で作成し、XAP-TP指令を一つ一つ実行できるシステムを作った。テストプログラムは、2200シリーズで開発し、テスト・スクリプト・ファイルから指令を読みながら実行する仕様である。このプログラムとスクリプト・ファイルは、3機種で共用できるので、開発コストの減少、開発期間の短縮、均一プロダクト品質になる。

つまり共通モジュールのテストは一つのシステムで行うことによりある程度完了する。若干システムに固有の処理に関する部分が含まれているため、その部分は個別にテストした。

9. 考察

今回の開発の結果、各機種の総行数の3機種共通コード行数が占める割合を移植率とすると、U6000シリーズの移植率は63.6%である。これは2200シリーズと比較すると構成ファイル関連、トレース関連部分が増えているためである。2200シリーズではこの機能はDDP-PPCに存在する。またファイル数が280と多く、小さいファイルが多い。Aの移植率は56.3%である。これはOAM関連、構成ファイル関連、トレース関連部分が増えているためである。Aシリーズ専用コード内のALGOLは26668行(46.5%)を占めた。

今回の開発を通じて効率の良かった点は、膨大なOSI-TPの規格、X/Open DTPの規格を共通モジュールを作るメンバーのみが理解作成し、このモジュールをブラックボックスとして扱うことができたこと。また、C言語を採用したことにより、移植性が良かったことである。2200シリーズで開発した共通モジュールには150人月かかっており、A, U6000シリーズでそのまま使用することで個別に開発するより開発費用は節約できた。評価テスト実施も共通モジュールの部分のテストの重複が避けられている。保守作業では、共通コードについて各機種共通に保守可能であり、保守作業量の削減ができる。

開発効率が悪かった点は、機種それぞれにシステム・インタフェースが異なる点である。また、下位ソフトウェア・インタフェースは、U6000シリーズを除いて、既存プロダクトの拡張であるのでXAPインタフェースを持っていない。したがって、機能面からXAPと同じ機能を実現するように作成した。

10. おわりに

本技術報告では、分散トランザクション・システムのプロトコルを受け持っているOSI-TPの実装について述べた。実装したコードでは共通コードの採用により、品質を均一に保ち、かつ保守が容易となった。また、それぞれの機種についての実装方法について説明した。個別に開発するよりも、今回とった方法が開発期間の短縮になっている。移植効率も高く、今後同様のシステム開発の参考になれば幸いである。

- 参考文献
- [1] Information Processing Systems - Open Systems Interconnection - Basic Reference Model (ISO/IEC 7498) (1984).
 - [2] Information Processing Systems - Open Systems Interconnection - Distributed Transaction Processing (ISO/IEC 10026, Parts 1 - 3) (1992)
 - [3] Information Technology - Open Systems Interconnection - International Standardized Profile (ISO/IEC 10026: OSI-TP) (1994)
 - [4] Information Technology - Open Systems Interconnection - Service Definition for the Commitment, Concurrency and Recovery Service Element (ISO/IEC 9804) (1990)
 - [5] Information Technology - Open Systems Interconnection - Protocol Specification for the Commitment, Concurrency and Recovery Service Element (ISO/IEC 9805) (1992)
 - [6] ACSE/Presentation: Transaction Processing API (XAP/TP), Document Number P216 ISBN 1-872603-85-5. X/Open CAE Specification (March 1995).
 - [7] ACSE/Presentation Services API (XAP), X/Open CAE Specification (September 1993).
 - [8] Distributed Transaction Processing : Reference Model, Version 2. X/Open Guide (November 1993).
 - [9] Distributed Transaction Processing : The XATMI Specification. X/Open Preliminary Specification (July 1993).
 - [10] Distributed Transaction Processing : The TxRPC Specification. X/Open Preliminary Specification (July 1993).
 - [11] Distributed Transaction Processing : The XA Specification. X/Open CAE Specification (October 1991).
 - [12] Distributed Transaction Processing : The Tx(Transaction Demarcation) Specification. X/Open Preliminary Specification (October 1992).
 - [13] Distributed Transaction Processing : The XA+ Specification. X/Open Snapshot (March 1993).

執筆者紹介 山野 茂美 (Shigemi Yamano)

昭和24年生, 47年電気通信大学電波通信学科卒業。
同年日本ユニシス(株)入社, 以来, Aシリーズ通信関連プロダクト, NDL,BNA,CP2000の開発・保守に従事。現在, ネットワークソフトウェア部に所属。



末永 敏昭 (Toshiaki Suenaga)

昭和32年生, 51年下関工業高校電子学科卒業。同年日本ユニシス(株)入社, 以来, 2200通信関連プロダクトの開発・保守に従事。現在, ネットワークソフトウェア部に所属。



河野典久 (Norihisa Kono)

昭和32年生，56年甲南大学理学部経営理学科卒業。58年日本ユニシス(株)入社，以来，UNIX通信関連プロダクトの開発・保守に従事。現在，ネットワークソフトウェア一部に所属。



石崎達也 (Tatsuya Ishizaki)

昭和35年生，58年埼玉大学工学部電子工学科卒業。同年日本ユニシス(株)入社，以来，2200,UNIX通信関連プロダクトの開発・保守に従事。現在，ネットワークソフトウェア一部に所属。



複数拠点下のXTPAシステム

An XTPA System for Multi-user Sites

加藤 武彦

要約 XTPA技術を用いたノーダウンスystemは、従来の事例では単独の拠点運用形態であった。複数拠点運用下のXTPAシステムでは単独拠点運用のXTPA技術では対応できない問題が発生する。

ホスト障害を例にすると、これまでは障害リカバリの範囲は、XTPAを構成する未障害状態のホスト側に限定することができた。複数拠点間で分散オンラインを行う場合、障害となったホストが設定した情報が他拠点システム上に残っているため、それらの無効化処理を行う必要がある。言い換えれば、XTPAのメディアムリカバリを相互接続する拠点間で行う形態となる。

本稿は、オンライン分野のXTPA適用技術、特に分散オンラインにおける複数拠点コンピュータ運用で必要となる機能について述べる。

Abstract "No-down" systems adopting XTPA (eXtended Transaction Processing Architecture) have so far been operated at single, independent computer user sites. In the case of an XTPA system linking multi-user sites, there occur problems that cannot be handled with conventional XTPA technology for single-site use. To take host computer faults as an example, it has been possible to limit the range of fault recovery efforts only to the running XTPA host computer. With a distributed, on-line XTPA system for multi-site use, data from the host, when it breaks down, keep staying on systems installed at other sites; thus bringing about the need to invalidate those data. In other words, that does require XTPA medium recovery between all linked independent system user sites.

This paper discusses on-line XTPA application techniques, especially focusing on the functions needed for multi-site computer operation in the distributed, on-line system environment.

1. はじめに

XTPA (eXtended-Transaction-Processing-Architecture)^{[6]~[8]}を用いたシステムは、エアライン、TRITON[®]等の先行事例があるが、複数の拠点を連動する形態でのシステム事例はなかった。

複数拠点連動とは、コンピュータ運用が独立した複数の拠点間で、オンライン処理を拠点連動して行う形態、すなわち分散オンラインデータベース処理である。このためにXTPA単独拠点運用のみを想定した機能では、システム立上げ・終了・障害回復ができない。たとえば、XTPAにおけるホスト障害回復の考え方は、「健全ホスト側にて障害ホストの回復を行う」という定義であり、障害ホストから他拠点の分散ホストに設定した各種更新情報のロールバック処理を、加入する全てのホストで実行する

本稿に記載の会社名、商品名は、一般に各社の商標または登録商標である。

• TRITON：金融機関向けのポスト3次オンラインシステムインフラおよび業務アプリケーションパッケージ（技報40号参照）。

必要がある。言い替えれば、XTPAにおけるメディアムリカバリ*を分散する全拠点を連動させて行う機能が必要になる。

本稿では、複数の拠点に跨る分散オンライン機能をXTPA環境でどのように構築したかを紹介する。

2. XTPAシステム構築の目的

1) 複数拠点コンピュータ運用

拠点内の一部ホストコンピュータが障害などにより停止しても、複数拠点を連携する業務処理に影響を与えない。

2) 営業所業務の無停止

障害が発生した場合でも、営業所端末の全体停止がないこと。障害が発生していないホストにおける業務は正常に実行できること。

障害内容に係わらず業務端末における帳票出力は全て可能であること。

3) ホスト障害発生時、障害ホスト接続端末が再使用可能になるまでの時間短縮

従来システムでは30分要したが、新システムでは5分以内とする。

4) 処理能力を増強する場合の容易性確保

以前は、数年先まで見越した設備を初期投資する方式であったが、投資対効果に基づいて業務の伸びに合わせて能力増強が簡単にできる仕組みであること。

3. コンピュータ拠点・ネットワーク構成

コンピュータの運用拠点は図1の通り、I・M拠点およびS・N支店の計4拠点である。I・M拠点の本番機はRLP**を使用したXTPA構成であり、S・N支店は単独コンピュータ構成である。拠点内部はEthernet-LANで結ばれており、拠点間通信に使用する。また、全体のコンピュータ運用管理・各拠点の稼働監視・ネットワーク機器監視には専用機を使用する。端末は旧端末（UTSエミュレータ他社端末）と新端末（PW²500EでOS/2使用）を接続する。新端末はそのサーバにUNIX（U6000：SVR4）機が採用されており、各営業所でLANが敷設されホストとの基幹業務を行う以外にEUCを行える設定である。旧端末はパケット網を通じホスト直結であり、通信装置にはDCP（Data Communication Processor）をRC（リモートコンセントレータ）として使用する形態である。

4. 分散オンライン処理

4.1 分散処理方式

業務量の増大および安全性の確保のため業務ホストの分散化が要請された。具体的には、各業務を地区毎に分割しその拠点間を横断したシステムである。

分散システムでは、特にデータの分散とアクセス方式がそのシステムの拡張性や効率などの重要な目標に対して影響を与える。いかに、他との関連性が少なく独立した

* メディアムリカバリ：XTPA運用下で、構成上の任意ホスト障害時に健全なホストにて障害ホストのデータ更新ログを使用し、データベースや管理テーブルを回復する。

** RLP：レコードロックプロセッサの略。疎結合ホスト環境において、ホスト共用のロック管理を行う。

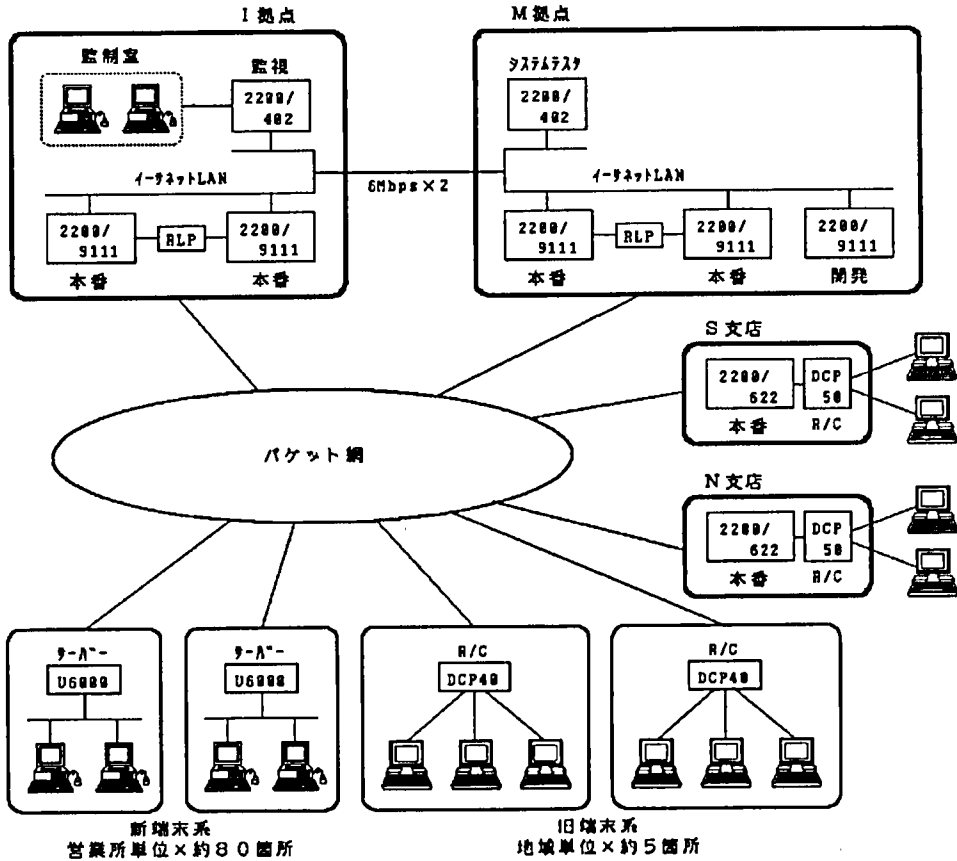


図1 コンピュータ構成図

サブセットにデータを分割できるかが鍵となる。本システムでは分散オンライン処理形態として、データサーバ・トランザクション転送・トランザクション会話の方式を使用する。各方式のポイントは下記の通りである。

1) データサーバ方式 (RDA：リモートデータアクセス)

- ① 業務アプリケーションに分散外部インタフェースを意識させない。
- ② 複数拠点間データベース更新のデータ保全性保証 (2フェーズコミットメント)。
- ③ ネットワークワイドなデータデッドロック検出。
- ④ 業務から独立したデータベースサーバの提供。
- ⑤ 拠点間ネットワーク制御 (サーバとクライアント間)。

処理上の特徴は、通信量が多く機構も複雑となるが、分散不可視性があり、柔軟性に富む方式である。適用業務形態は少量データ更新・複数拠点間同時更新型の処理である。

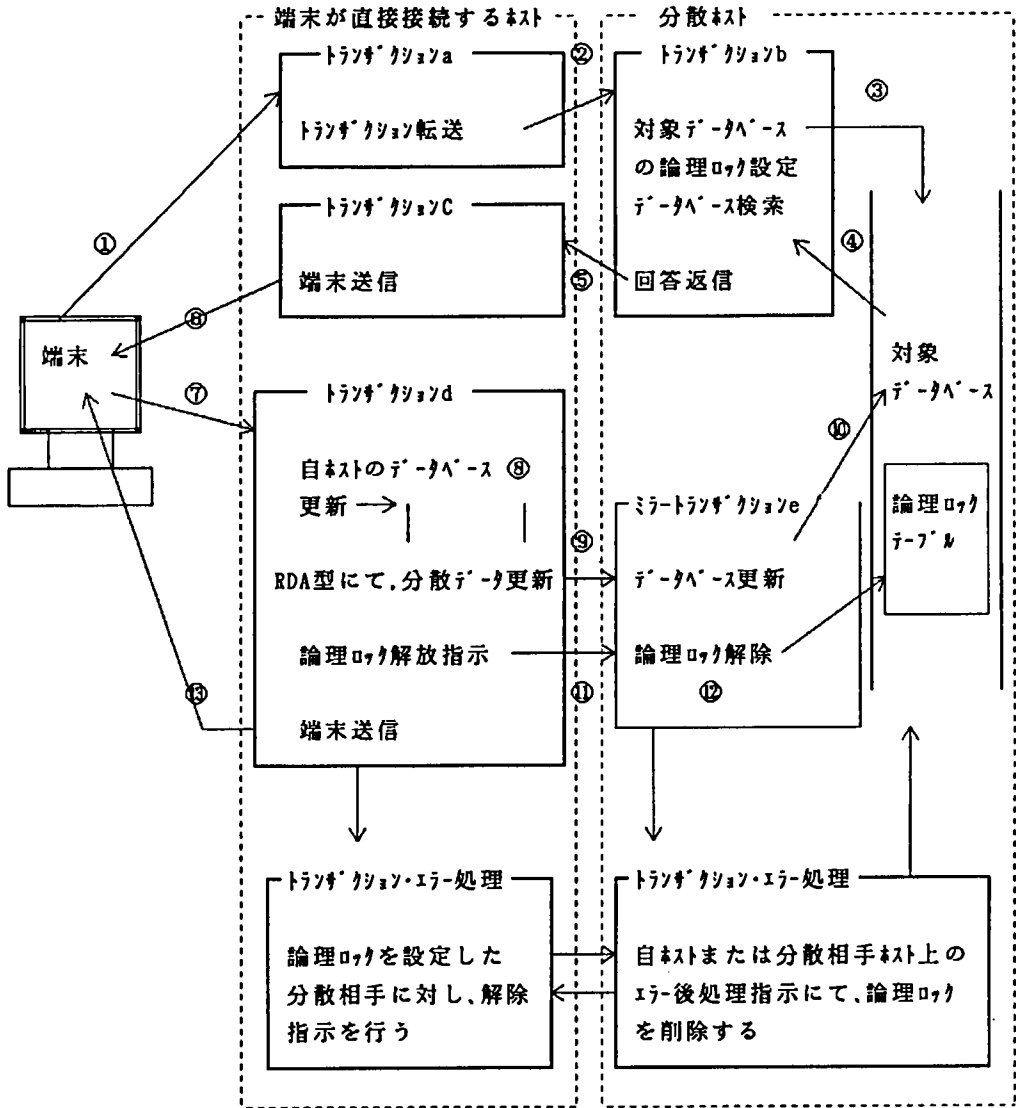
2) トランザクション転送方式

- ① 分散についての高度な知識なしにシステム構築ができる。
- ② トランザクション転送先でも通常のプログラミングができる。

- ③ エラー処理などの異常対応をアプリケーションから独立させ、開発者が業務処理に専念できること。
- ④ ネットワークの接続・切断の自動化。

処理上の特徴は、通信量が少ない、分散データ配置を意識した業務プログラムの考慮が必要となる。適用業務形態は、大量データ検索型の処理である。

3) トランザクション会話方式 (PPC)



ミラートランザクション：クライアントになる別コンピュータからのデータアクセス要求に対しデータベースアクセス（検索更新）を行い、レコードの供給を行うトランザクション。サーバ機能プログラムでスキーマの単位設定する。分散論理ロックなどの機能を持ち、2フェーズコミットメントも行う。

図2 分散処理

- ① 分散についての高度な知識なしにシステム構築ができる。
- ② 分散依頼元プログラムが主導でデータ制御でき開発するプログラム量、考慮すべき項目がトランザクション転送方式に比べ簡単になる。

処理上の特徴は、データサーバ方式とトランザクション転送方式の中間であり、通信量が多く、業務プログラムは分散データ配置を意識する必要がある。プログラム作成は容易であり、少量のデータ検索に向く。

4.2 トランザクション実行の流れ

端末入力から応答が返るまでの間における分散処理の流れを図2に示す。

4.3 分散ホスト接続

複数拠点ホスト間の接続形態を図3に示す。各ホストシステム内の印は、ホスト間接続パスでトランザクション転送時の仕向・被仕向経路・リモートデータアクセスやPPC時の会話パスになる。XTPAを構成するホストの二重障害（たとえばH1aとH2bの同時障害）でも分散処理が可能ないように全てのホストから網接続する。

オンラインセットアップ時に、各ホストは分散相手ホストに接続要求を発行し稼働中ホストとのインタフェースが確立される。各ホストは、確立が完了したパスのみを管理し使用不能パスを除去する。セットアップが順次完了すると、全てのホストパスが使用可能となる。実行時のパス選択は相手ホストへの全接続パスを循環使用する。

I拠点aホストの接続端末からM拠点に分散接続する例では、①を第1回目（M拠

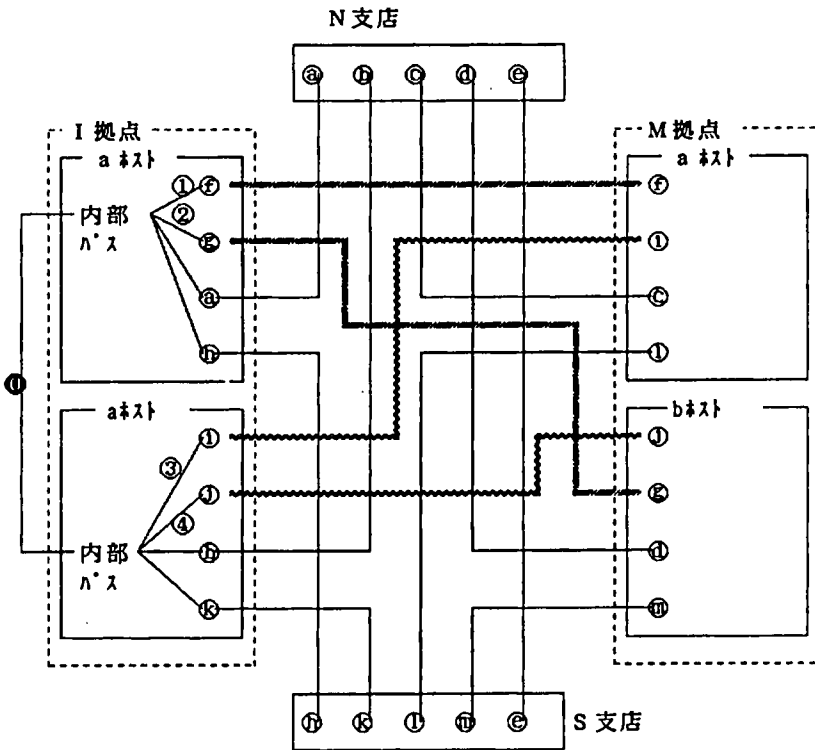


図3 分散ホスト間パス設定

点aホスト宛), ②を2回目 (M拠点bホスト宛), ③XTPA内部ホスト間パスを使用して③を3回目 (M拠点aホスト宛), ④を4回目として以降この繰り返しを行う。不通なパスを発見した時, 該当パスを閉塞し以後使用しないようにする。

5. XTPA実行時機能

5.1 障害回復

XTPA運用時のホストコンピュータ障害回復機能はOS2200・UDS*およびXISにより提供される。回復対象は各々OS2200にてFCSS**回復, UDSにてデータベース回復, XISにてメモリテーブル回復である。本システムでは分散処理にて論理ロック(プログラムが存在しない状態でも有効となるロック機構: 問い合わせから更新の過程で他の使用者にデータ変更されないようにする目的)を使用するため, 障害回復対応が独自に必要となる。当ロック情報は, XTPA以前の運用ではメモリ上に配置しており障害回復をホットスタンバイにて対応する場合は無効処理不要であった。XTPA運用では, ホスト共用ロックとする必要があり, ディスク上のロック配置に変更した。独自回復として, XTPAホスト障害回復(ミディアムリカバリ)にこれらの情報を無効化する処理を追加する。さらに本ロックは分散接続する別拠点にも設定されるため, 加入する全てのホストに通知し一斉に無効化処理を行う。図4にXTPA運用を行っているI拠点aホストが障害時の同報一斉ミディアム回復処理を示す。

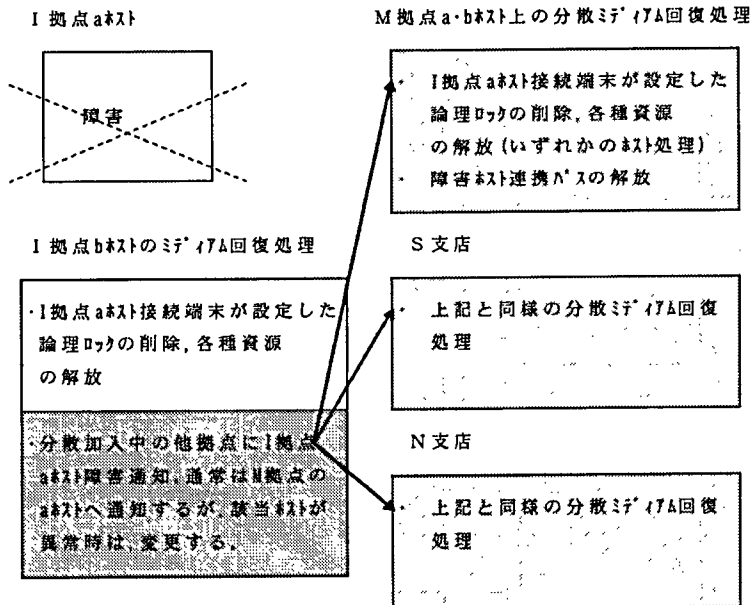


図4 拠点間XTPA回復機能

* UDS: Universal Data System. ネットワーク型・リレーショナル型データベース実行・運用管理システムであり, XTPA環境においてデータベース管理・回復・維持機能を提供する。
 ** FCSS: File Control Super Structure. TIP (Transaction Interface Package) が制御するリアル専用ファイル。

5.2 ホスト間パス管理

通信パスはホスト単位の管理であり、XTPAを構成するホスト間パスと拠点間(XTPA構成のシステム間)パスが存在し、各システムの立上げ・終了および障害回復時に設定・変更する。管理情報は、自ホストから見た分散ホストアクセス状態とする。

5.2.1 立上げ処理

ホストの立ち上げ時、XTPAを構成する相手ホストに対し接続要求を行い、正常な場合、内部ルート(XTPAホスト間内部転送パス)を開通し、通信可とする。一方、分散ホストに対しては、構成ホスト単位(例：I拠点aホストからはM拠点a・bホストおよびS・N支店)に同様の処理を行う。接続要求に対し回答がない場合は相手ホストは立ち上げ遅れと判定する。

5.2.2 加入処理

XTPA構成下の一方のホスト立上げが遅れた場合、XTPAホスト間の内部ルートは不通状態となっている。また、該当ホストと他拠点間も同様の状態で分散処理には加入していない状態である。このようなホストが立ち上がりXTPA環境に組み込まれると、立上げ処理と同様にXTPA相手ホストおよび他拠点ホストへ加入要求を行う。各拠点ホストの運用は独立しているため、立ち上げが同期しない場合でも、当機能により分散接続が可能となる。

5.2.3 終了処理(離脱)

加入処理と同様にシステムの終了は、同期して行うことができない。非同期に順次終了を可能とするため、ホストシステム終了時に離脱要求を発行し、XTPA相手ホストおよび他拠点の分散ホストから当ホストを使用できないよう設定を行う。

5.2.4 障害・再立上げ

ホスト障害時に、XTPA構成の健全ホスト側にてメディアム回復を行い、分散ホストに対して障害ホストへのパス切断要求を行う。障害ホストが復旧し再度使用可能になると再立上げ処理にて加入処理と同様の要求を行い、分散環境に復帰する。

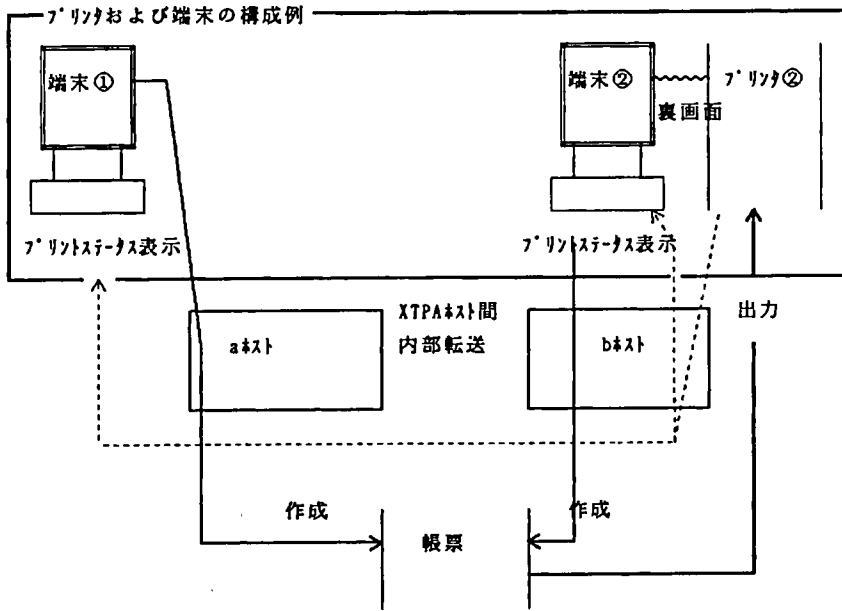
5.3 ホスト障害時のプリンタ接続変更

端末・プリンタのセッション(通信路)管理は、ホスト側にて行う。通常運用中はこのセッションを切り替えし、接続ホストを変更することはない。ホストの管理情報は、該当プリンタを制御する端末情報、任意の端末処理で作成された帳票の宛先プリンタ情報などを持つ(後述5.4節参考)。ホスト障害発生時、健全ホスト側へ端末を切り替える。会話処理を行う端末は電文入力トリガーとしてセッション変更を検知しホストの管理情報に反映可能であるが、プリンタは入力がないので、元のネットワーク状態より吸収した端末の管理情報を更新する。

5.4 プリント処理のホスト間制御

プリンタと端末は1対nにて設置され、端末からのプリント出力は場合により別のホストに接続されたプリンタとなる。このような構成を可能とするため、次の機能を追加する(図5に制御関係を示す)。

- トランザクションにて作成された帳票を、該当端末と対応するプリンタに、ホストを横断して出力する機能(端末接続ホストとプリンタ接続ホストは異なる)



端末①②は共用のプリンタを持ち作成した帳票はプリンタ②に出力される。プリンタ②の稼働状況は、①および②の端末に表示される。

図5 XTPAホスト環境下のオンラインプリント制御

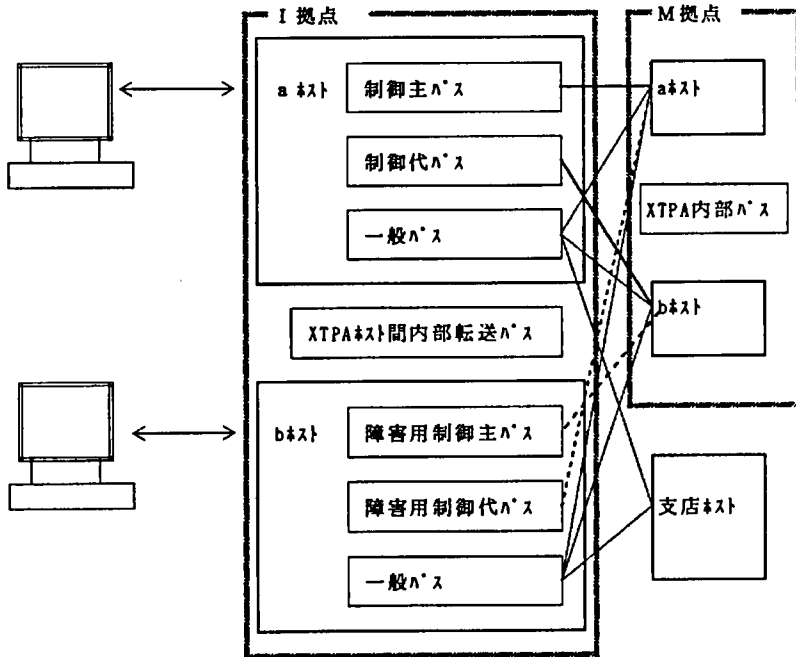


図6 分散ホスト間トランザクション転送パス設定

る)

- プリンタの稼働状況を、対応するn個の端末に表示する機能
- プリンタの切替えによるホスト変更、セッション管理、出力制御切替え

5.5 トランザクション転送ホスト間接続管理

トランザクション転送はXIS*の機能を使用する。分散ホスト間接続制御パスはXTPA構成ホスト中唯一の設定しかできない制約があるため、制御パスを保持するホストが障害の場合、分散トランザクション転送が全停止となる。当問題は複数拠点の二重障害（複数のXTPA構成コンピュータ群の障害が重なった場合）でも発生するものであり全停止防止のため網接続にて対応する。図6の設定を元に以下に説明する。

- ① I拠点-M拠点間を4本のパス接続する。
- ② 制御パスは、aホスト間パスを主パスとし、a-bホスト間を代行パスに設定する。bホストからの他拠点パスは障害対應用パスとして使用する。
- ③ 一般トランザクション転送パスは、全経路使用する。

6. 運用機能

6.1 XTPA オンライン環境の端末振り分け定義

端末をXTPA構成の複数ホストに割り付けするが、CMS**/TELCON***, XIS, 固有管理情報など関連するものが多く、且つその整合性を保つ必要がある。サーバ・端末構成など複雑であり、個々の情報定義の矛盾排除のため一元化したネットワーク辞書にて維持管理する。端末を複数のホストへ振り分けするが単純に半々の台数とした。この振り分け定義と6.2節の端末側セッション設定は矛盾しないよう設定する必要がある。

6.2 ホスト障害時の端末間引き運用について

ホスト障害時、該当ホスト分の端末を吸収する運用となる。このため障害環境下で、端末を無制約に使用するとホスト過大負荷によるパニック状態が発生する。このような状態では端末の応答遅延となるため、操作者はさらに各種操作を行い、より高負荷となる。対応として、障害時の端末間引き運用を行う。障害ホストに接続した端末に対してホスト側の回復を行い、且つ入力閉塞を設定し端末からの入力に対し閉塞の旨応答を行う。その後運転の負荷状況により順次閉塞を解除する。

プリンタ運用がこの際の問題となる。プリンタは端末の裏画面経由出力であり端末を閉塞するとプリンタも停止してしまう。プリンタはオンライン業務が可能であれば必ず稼働可能でなければならない。対応として画面入力は閉塞するがプリント制御入力は有効とするよう機能追加した。図7に縮退運用時の端末間引き状態を示す。

6.3 XTPA環境における端末回復

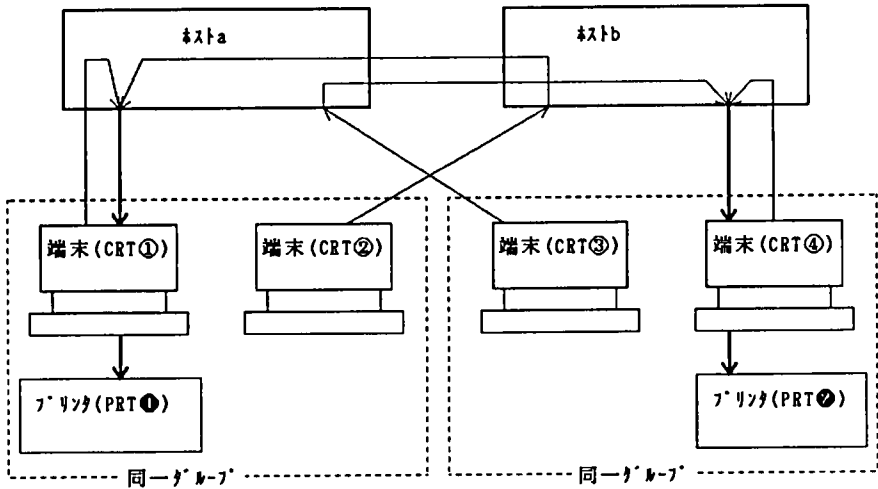
XTPA環境では、複数ホストのうち、1台のホストに障害が発生してもシステムは継続処理が可能である。しかし、障害ホストと接続中の端末は通信セッションが切断

* XIS：eXtended Information Systemの略で、拡張モード環境、XTPA環境を支援するプロダクト。

** CMS：2200上の通信制御ソフトウェアであり、アプリケーションインタフェースやDCPインタフェース機能を持つ。

*** TELCON：DCA (Distributed Communication Architecture) の概念に基づく通信制御ソフトウェアで、DCP (Distributed Communication Processor) 上で稼働する。

□ 端末とホスト接続初期構成



□ ホストa障害時の端末(CRT)操作変更

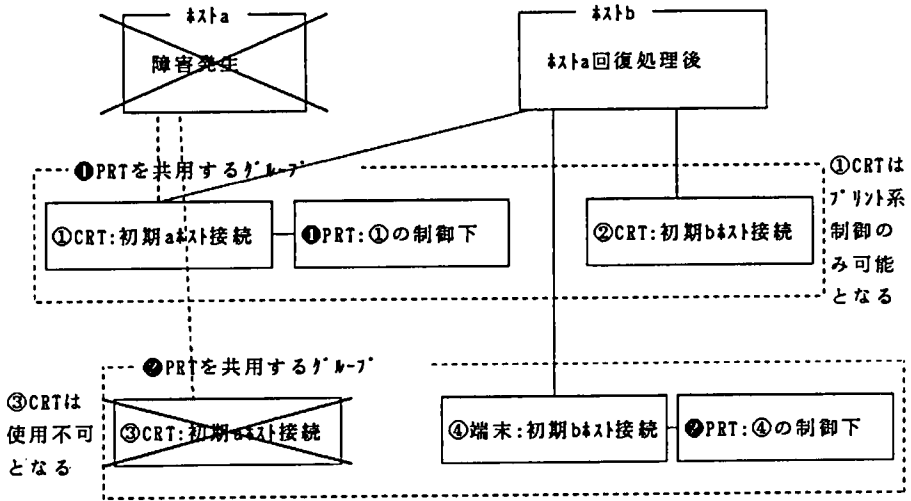


図7 ホスト障害時の端末間引きとプリンタ制御

されるため停止状態となる。この対策として予めバックアップセッションを設定しておき主セッションの通信停止時に切替えを行う「デュアルセッション」機能がある(図8)。

本システムでは接続する端末数が2000台となり、セッション回復時間が重要な課題となる。ノーダウン・停止時間の最小化のためデュアルセッションを検討したが、端末がOS/2・TELCON上DCAパススルー形態であり、採用することができなかった(デュアルセッション機能は、DP-M1型端末に特化したものである)。シングルセッションでホスト障害時の端末回復時間を測定すると目標値を大幅に越える値であり、別な対応策が必要となった。図9にネットワーク環境を示す。本システムの特徴

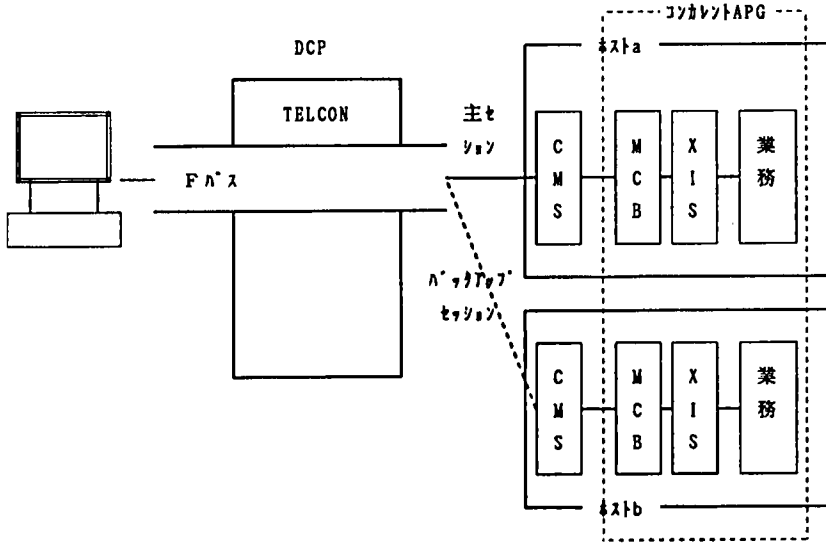


図8 デュアルセッション機能

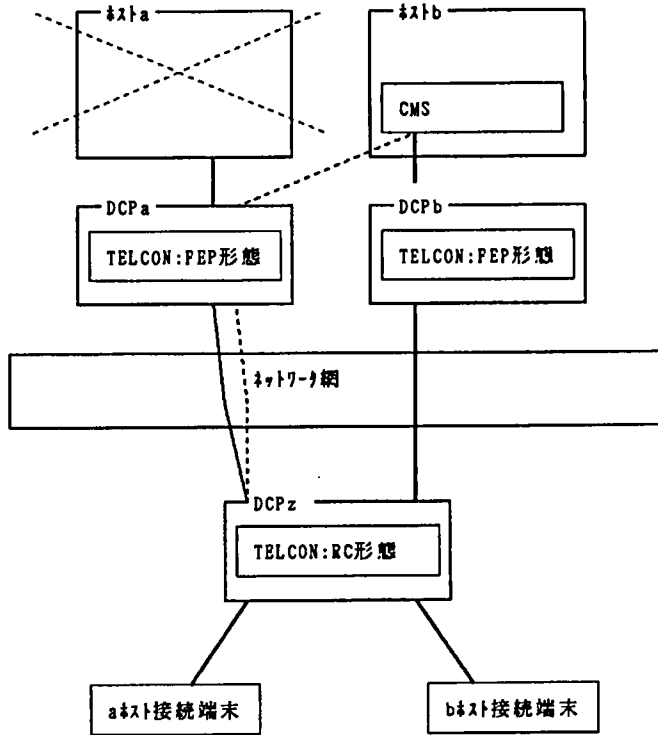


図9 単一セッション下の端末回復

は、ネットワーク要素にリモートコンセントレータ (RC) が存在していることであり、RC-FEP-CMSの3コンポーネント間の回復手続き通信回数が多く、長時間になる原因となっていた。対策はこの3者間の通信回数を少なくすることであり、複数手続きをまとめて行うようにした。

7. おわりに

複数拠点システムが、XTPA環境で且つ分散オンライン運用を行う事例があまりなく、運用定義が難しい。分散拠点運用とXTPA運用は本質的に関係ないが、「他システムまたは他ホストの障害回復を、別システム・ホストが行う」考え方・「独立運用の要件」と「協調業務の要件」の整理を行い検討すべきものである。今回の事例は2200ホスト間の分散・XTPAシステム構築であった。今後はホストがサーバとなり、サーバ間およびPCを含む業務連携運用が課題となると予想される。本稿が何らかの参考になれば幸いである。

本システムの課題はホスト負荷の平準化が動的に行うなどデータ通信を含めた形で能力強化である。個々のコンピュータ資源や負荷をバランスさせ全体スループット向上を行うことがテーマである。

最後に、今回のシステム構築・運用設計に、多大な協力を頂いた2200シリーズソフトウェア部とシステム企画開発部の方々に感謝する意を表す。

-
- 参考文献 [1] Mercury/XTPA適用ガイド。
 [2] シリーズ2200拡張トランザクション機能XTC概説書。
 [3] シリーズ2200拡張トランザクション機能XTC移行解説書。
 [4] 河田雄三, 奥田孝, 久野一輝, “ユニシス計算機の拠点拡充作業について”, ユニシス研究会入選論文集1994。
 [5] 中根義明, 山田健史, “クライアント/サーバ・システムの効率的運用を目指した基盤構築”, ユニシス研究会入選論文集1995。
 [6] 沢田啓, “XTPAに基づくノードダウン・システム”, 技報通巻40, 日本ユニシス(株), 1994. 2。
 [7] 石川清人, “XTPA環境におけるXISのデータ通信制御”, 技報通巻42, 日本ユニシス(株), 1994. 8。
 [8] 竹内久, “XISのXTPAシステム運用”, 技報通巻42, 日本ユニシス(株), 1994. 8。
 [9] 栗原幸代内久, “XISの分散トランザクション処理”, 技報通巻42, 日本ユニシス(株), 1994. 8。

執筆者紹介 加藤 武彦 (Takehiko Kato)

49年木更津工業高等専門学校機械科卒業。同年日本ユニシス(株)入社。オペレーティングシステムの開発・保守を始め、オンラインデータベース関連の開発・適用に従事。現在中部支社システム技術部に所属。



信金バックアップセンタの紹介

A Profile of the Nihon Unisys Back-up Center for *Shinkin* (Credit Association) Banking Systems

関 俊 夫

要 約 金融の自由化・国際化が進み金融業界とりわけ信用金庫を取り巻く経営環境は益々厳しくなっており、機械化に対する負荷も益々増大している。また金融機関におけるコンピュータの利用範囲は対外接続の広がりとともに益々拡大し社会的責任の増大に伴う安全性確保が重要な課題となっている。

日本ユニシス信金バックアップセンタは信金業界のシステムの安全性を低コストで保証することを目的として設立された。

本稿は信金のお客様がバックアップセンタをより理解し有効に利用して頂けるよう、設立の背景・目的、業務内容、今後の展望について紹介するものである。

Abstract Recent moves in the financial market toward deregulation and globalization have triggered an increasing difficulty of the business environment in the financial industry — specifically in the *shinkin* (credit association) banking sector and also boosted loads in computerization efforts. As computer uses at financial institutions are expanding to meet needs for linking with other systems outside, it has become more and more important to ensure the safety of those systems with the growth of their social responsibility.

The Nihon Unisys Shinkin Back-up Center was set up in order to guarantee the security of computer systems for *shinkin* banking institutions at a low cost.

With the aim of helping *shinkin* users get a good picture of and make more effective use of the Center, this paper is intended to present the background of its new creation, its objectives and the details of its operations as well as its future perspectives.

1. はじめに

今日、コンピュータにおける情報化は広く社会全般に浸透し、コンピュータシステムは、我が国の経済活動はもとより、国民生活の向上に欠かす事のできない社会的基盤を形成している。

とくに金融機関におけるコンピュータシステムは、技術の進歩に従って、業務のあらゆる面に組み込まれ、年々大規模化、複雑化してきている。

このため、災害、事故等予測しない緊急事態でその機能が停止すれば、社会的影響は甚大であり、これらに備えた安全対策、とくにバックアップ対策を講じる事は、金融機関にとっては、社会的責任、公共的使命といえる。

一方、バブル経済の崩壊、自由化に供う商品の複雑化、即応性への対応等機械化投資の増大等、金融機関をとりまく環境は非常に厳しいものとなっている。

弊社では信用金庫業界におけるトップメーカ（自営オン86金庫中43金庫、シェア

50%)の責任として、上記不測の事態に備えるため、信金バックアップセンタを設立し、コンピュータ障害に対する保証を低コストで提供している。

2. 信金バックアップセンタ設立の背景

我が国では地質的、気象的に様々な自然災害が発生する条件下にあり、また比較的安定しているとはいえ、人的災害も増加の傾向にある。

昭和59年11月に東京世田谷で起きた「地下ケーブル火災」による停電で、金融機関のオンラインが麻痺した経緯があり、「釧路沖地震」「北海道南西沖地震」と大規模災害が多く、発生している。

こういった背景の中で信用金庫業界のコンピュータおよびネットワークも年々大規模化し、経営にとっても不可欠のシステムであると同時に、我が国の経済活動および市民生活にも欠く事のできない社会的基盤の一部を形成している。

したがって、システムが長期停止に至った場合の社会的影響は大きく、何らかの対策を講じる必要が生じてきた。

2.1 バックアップセンタの検討

このような背景により、コンピュータ災害にそなえる必要性から、バックアップセンタ設立の機運が生じてきた。自営で設立した場合、共同で設立した場合、それぞれのメリット/デメリットを検討した結果

“自営でバックアップセンタを設立するのは不可”
との結論に達した。

その理由としては、すべてコストにはねかえってくるのだが、信用金庫という狭い地域での地域金融機関という属性に負う所が大きい。

つまり、営業区域内にバックアップセンタを設立したとしても、大規模災害の場合には同時に被災してしまう可能性が大きい。また、区域外に設立した場合、運用要員等を確保しておく事が非常に困難である。共同で設立するにしても、同様の問題、とくに平常時のコンピュータ有効利用および運用要員の確保の難しさが残された。

そこで、第三者委託の形でバックアップセンタ設立の要望が弊社に寄せられた。

2.2 信金バックアップセンタの設立

前記要望を受け、弊社本社内に信金バックアップセンタを設立することになったが、万が一の保険的システムであるとの考えから、コスト（開発費用、保守費用、運用費用）を極力押えた。

- 1) 共同利用型のセンタであり、運用は日本ユニシスが全面的にサポート。
- 2) 既存機器の流用を図る。
 - ホストマシン：日本ユニシス信金システム開発センタの開発マシンを利用。
 - 端 末：営業店既設置端末をそのまま利用。
 - 回 線：営業店回線バックアップ用に敷設されているINS回線を利用。
- 3) メンテナンス負荷軽減のため、統一アプリケーションとする。
- 4) 必要最小限の機能をもった特定業務バックアップ方式とする。

3. 信金バックアップセンタの概要

3.1 対象範囲

対象ユーザは全国の自営オンライン信用金庫（信用組合）で、勘定系を弊社のホストコンピュータで構築しているユーザである。

ここで弊社ユーザに限定しているのは

- 1) データベースのバックアップは、時間短縮の意味から物理的に吸い上げ保存する。そのデータベースの移行には、吸い上げたマシンと同等のものを用意する必要がある。
- 2) 端末接続プロトコルは、接続ホストマシンに依存する。

ことの2点の理由からである。

3.2 特徴

万が一の場合の保険的システムの位置づけからコストを最小限に押えるため、信金バックアップセンタは以下の特徴をもつ。

- 1) 特定取引に限定したシステムである。

被災時に必要な最小限の業務、言い換えればホストコンピュータがなければ、対顧客、対外部に対して大きな信用失墜を来たす業務に絞って構築した。

絞り込みは以下に示す手順で行った。

- ・コンピュータに依存している業務を全て洗い出す。（営業店、本部、センタ）

↓

被災時、顧客へのサービス低下と信用リスクを来たさないためには

↓

- ・被災時も行う業務の選定（被災対策対象業務）

↓

手作業では処理不可能なものは、

↓

- ・バックアップセンタシステムの対象業務の選定

この手順の詳細は図1に示すが、バックアップセンタシステムの対象業務は次のものを選定した。

- ・流動性預金（普通預金、当座預金）
 - ・窓口端末および自動機による無通帳入出金
 - ・対外接続による他行被仕向（普通預金）
 - ・事故届
 - ・外部自振
 - ・締め上げ（精査）
- ・為替
 - ・他行被仕向
- 2) 業務処理プログラムの統一、標準化

各ユーザで行っている勘定系処理は、当然ユーザごとの独自性をもたせた作り

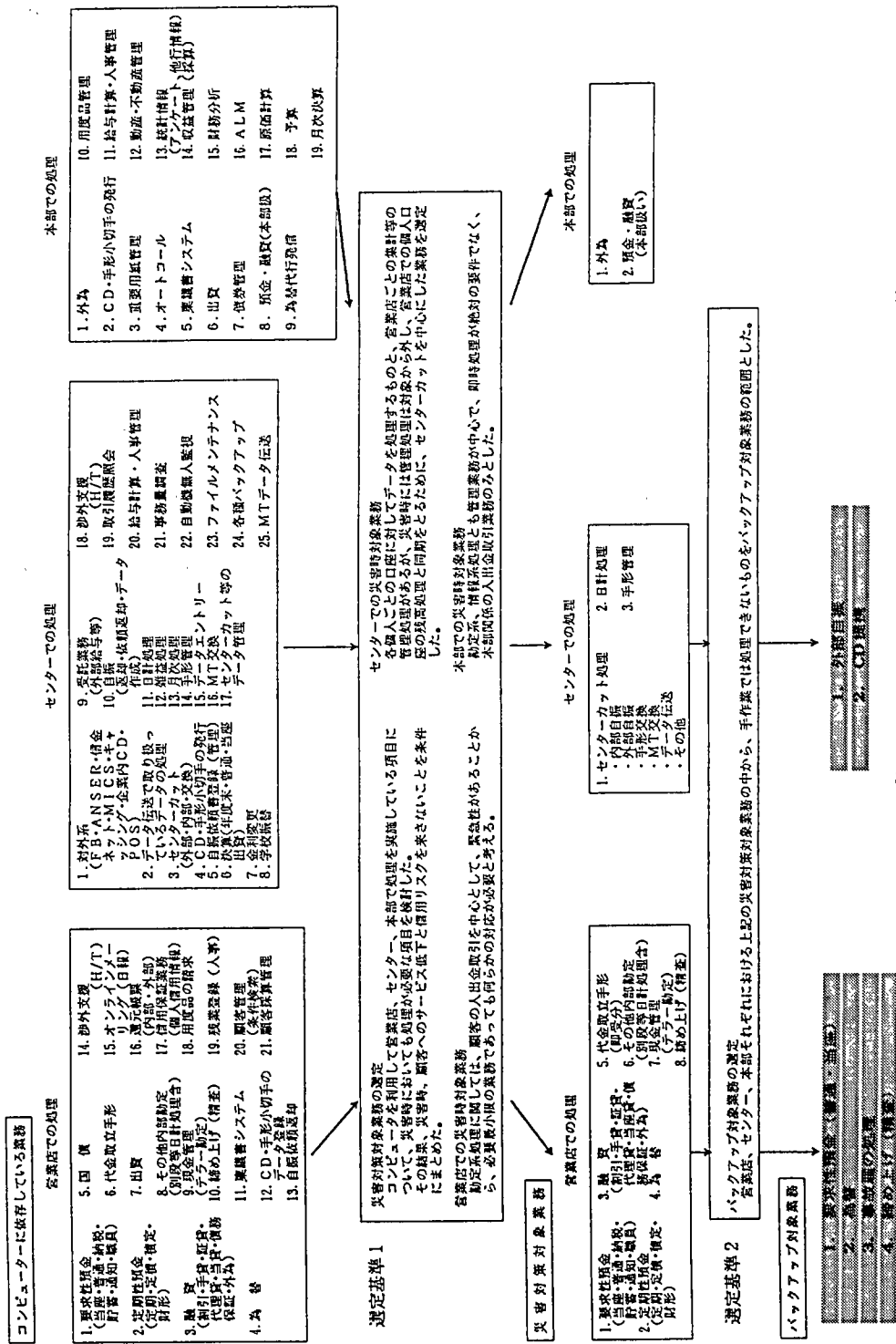


図1 バックアップセンター対象業務の選定手順
出典 「信用金庫業界におけるバックアップシステム構築の必要性について」 Ⅱ、平成6年7月 (社) 全国信用金庫協会

となっており、そのため、データベース構造、端末入力画面、業務処理プログラム等ユーザごとに異なったものとなっている。

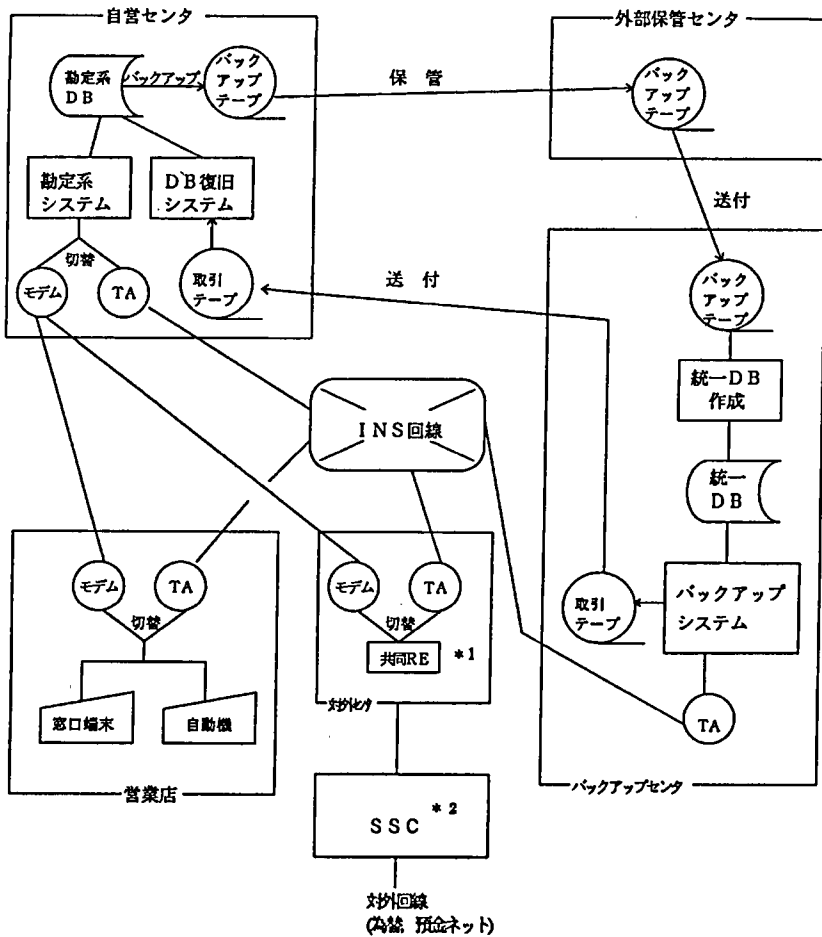
しかし、被災時に行う特定業務処理は、口座の管理が可能で、その口座に対する入出金、および付帯業務（設定、照会、精査等）が可能であればよいと考えれば、各ユーザごとに可変である必要はない。

また、バックアップセンタシステムの構築負荷軽減、構築後のメンテナンス負荷軽減を考え、業務処理プログラムはどのユーザで使用する場合でも同一である。

そのため

- ・統一したデータベース構造
- ・統一した端末入力画面、出力媒体
- ・各ユーザ毎可変部分のテーブルウェア化、OWNコード化

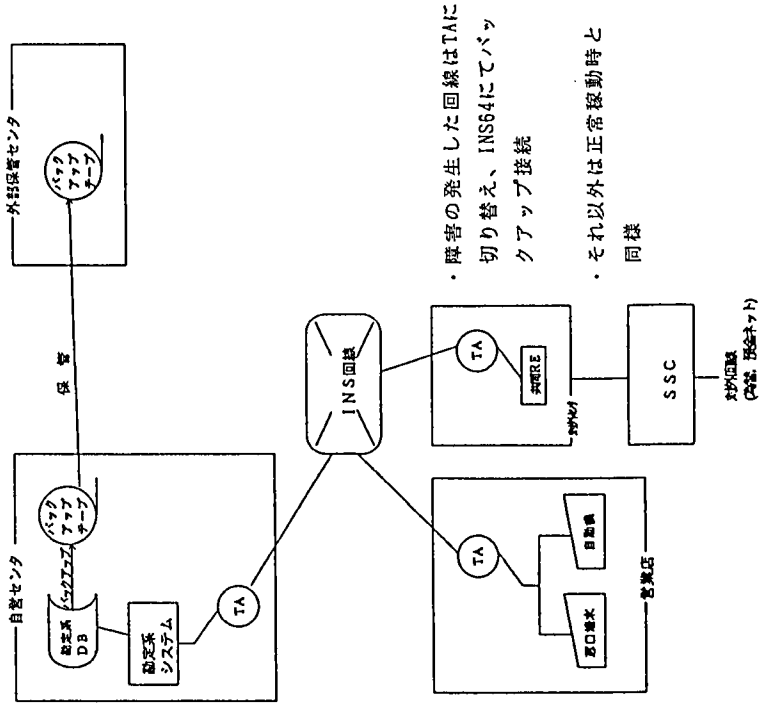
が図られている。



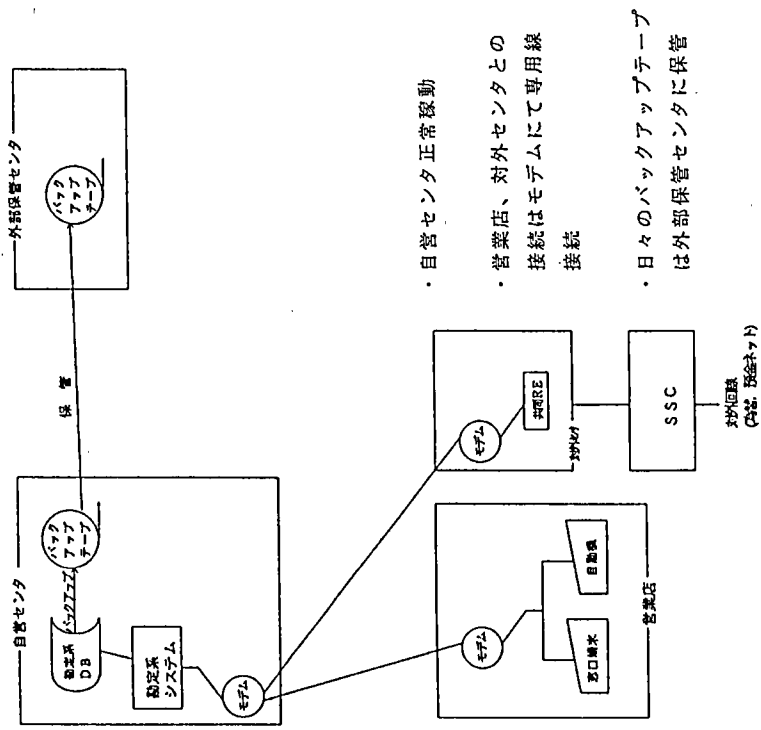
*1 共同RE：自営信用金庫の対外取引（預金ネット、為替等）の中継を行うコンピュータであり、全国七つの共同地区センタ内に設置されている。

*2 SSC：信金情報システムセンタの略であり、信用金庫業態内、業態外の取引の中継を行う。

図2 バックアップセンタ全体構成



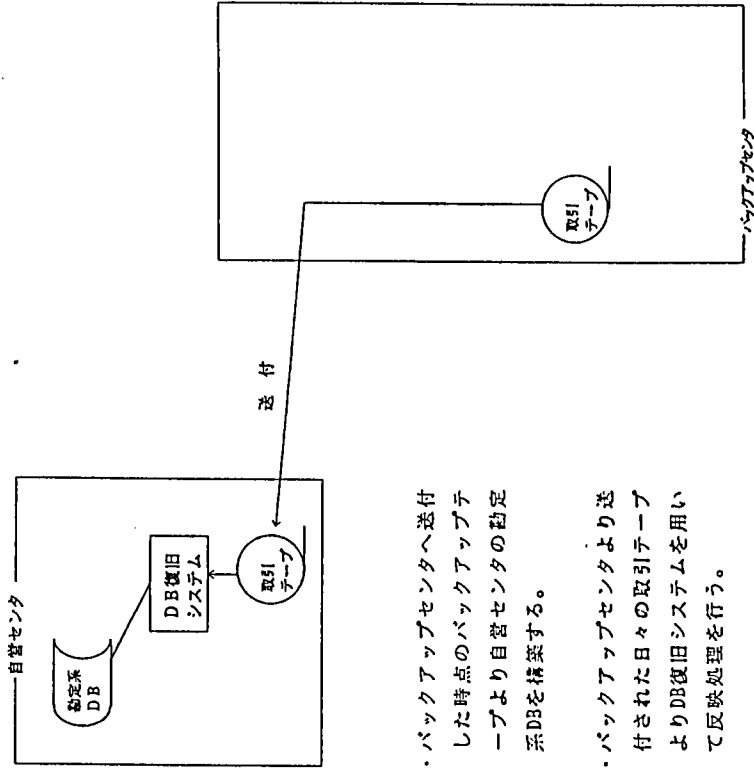
- ・ 自営センター正常稼働
- ・ 営業店、対外センターとの接続はモテムにて専用線接続
- ・ 日々のバックアップテープは外部保管センターに保管



- ・ 障害の発生した回線はTAに切り替え、INS64にてバックアップ接続
- ・ それ以外は正常稼働時と同様

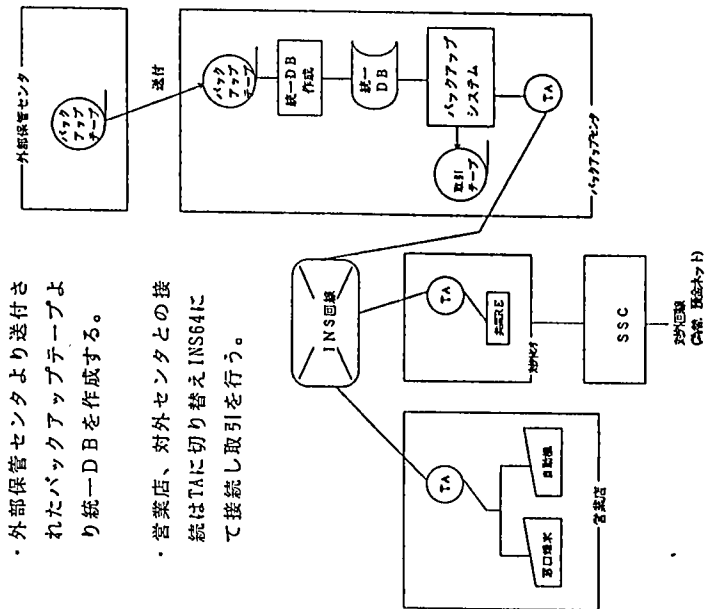
図3 正常稼働時の作動範囲

図4 回線障害時の作動範囲



- ・バックアップセンターへ送付した時点のバックアップテープより自管センターの勘定系DBを構築する。
- ・バックアップセンターより送付された日々の取引テープよりDB復旧システムを用いて反映処理を行う。

図6 反映処理時（自管センター復旧時）の作動範囲



- ・外部保管センターより送付されたバックアップテープより統一DBを作成する。
- ・営業店、対外センターとの接続はTAに切り替えINS64にて接続し取引を行う。

図5 バックアップセンター稼働時（自管センター被災時）の作動範囲

3) 同時稼働ユーザ数の制限

災害の規模、場所、およびバックアップセンタ加盟ユーザ数等により、同時被災ユーザの数を事前に特定する事はできない。

加盟ユーザ全てが同時稼働できる環境を構築する事が最良であるが、主にハードウェアの諸要素（ディスク容量、回線容量等）から最大3ユーザ同時稼働を前提としてシステム構築している。

4) 既存端末、既存回線の使用

バックアップセンタ業務処理の統一とは相反するが、トータルコスト（端末改造、新設費と業務処理プログラムの改造費）を考えると、端末は各営業店に設置されている既存のものを使用することとしている。

端末および対外センタとの接続は、自営センタとの間に設けられた専用線障害時のバックアップ回線（INS64）を流用する事で、バックアップセンタ側のTA、INS回線以外の費用負担を防いでいる。

3.3 バックアップセンタ全体構成

バックアップセンタの位置づけ、接続形態を説明するために、自営センタ、営業店、対外センタ、バックアップセンタを含めた全体構成（図2）と、正常稼働時（図3）、回線障害時（図4）、自営センタ被災時（図5）、および復旧処理時（図6）の各々の場合の作動環境を示す。

4. 信金バックアップセンタシステムの仕組み

4.1 データベースの移行

信金バックアップセンタシステムの前提としているデータベース形式は統一された形式となっているため、個々のユーザのデータベースからの移行作業が発生する。

各ユーザのバックアップテープは、通常業後処理の時間短縮のため高速（物理的な形式）で吸い上げられている。また、業中（データベース未確定状態）に吸い上げ、確定版（その日の最終の状態）は、オーディットより更新情報をはりつけることで、作成するユーザも増えてきている。

これらの物理的な状態のファイルより論理的な形式のファイルにするためには（確定版の作成も含めて）、ユーザで吸い上げたものと同一の機種上で処理を行う必要がある。このため、信金バックアップセンタには、UNISYS2200シリーズ、Aシリーズのマシンを用意して対応を図っている。

論理的に読める形となったユーザデータベースファイル（個々のユーザの形式）より移行プログラム（各ユーザ作成で持ち込まれたもの）を使用し、信金バックアップセンタ標準の統一データベース形式のファイルを作成する。

図7に移行作業の一連の流れを示す。

4.2 営業店端末、対外センタ接続

4.2.1 営業店端末の接続形態

営業店端末の接続形態は大きく分けて

- ・ スタンドアロン型
- ・ TC型（ターミナルコントローラがホストと端末の間に介在し、端末を制御す

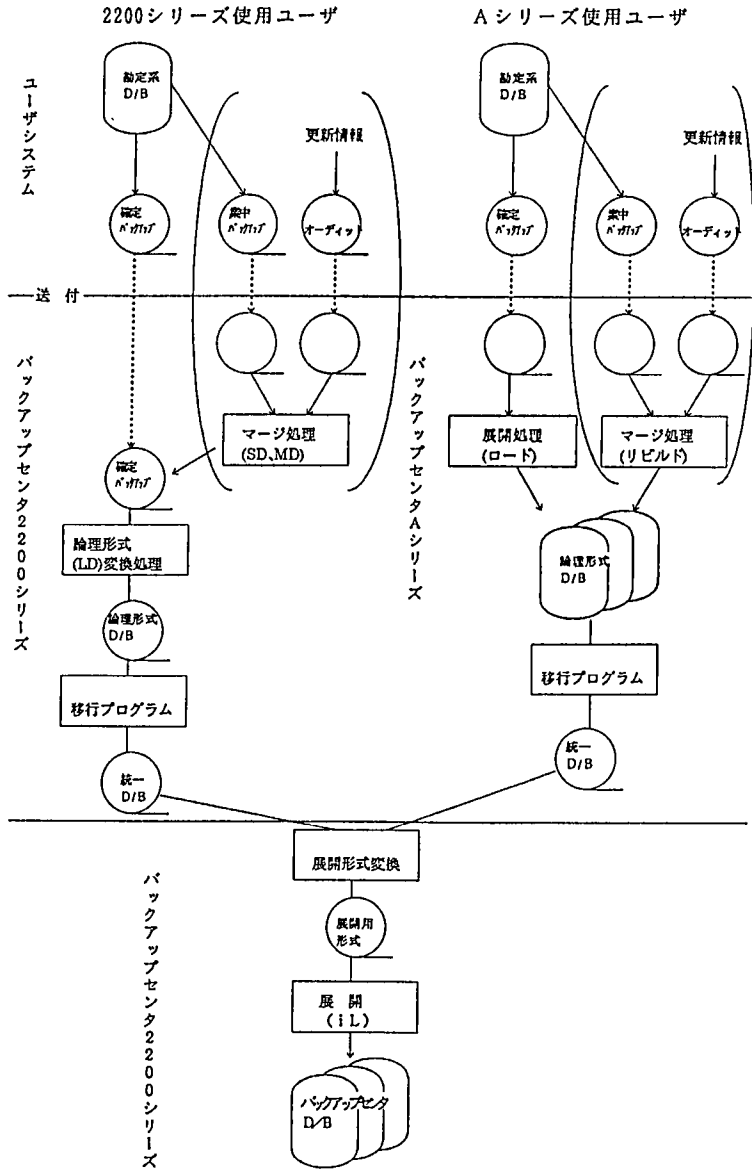


図7 データベース移行の流れ

る)
に分類される。

TC型の場合はさらにTCの設置場所によって

- ・センタ設置型
- ・母店設置型
- ・営業店設置型

に分類される。

スタンドアロン型は、端末単独でホストと接続できるため、バックアップセンタ側

にはINS回線およびTAを用意しておけば、それ以外に特別な機器は必要なく接続可能である。

しかし、TC型の場合、設置場所によってはTC自体も被災している事が考えられ、端末単独ではホストと接続する事ができない。

ユニシス自営信金43金庫中26金庫がこのTC型の端末を使用しているが、そのうち25金庫で使用している沖電気製のTCをバックアップセンタ側では用意しており、TC被災時のバックアップに備えている。

残り1金庫は、営業店設置型のため、バックアップセンタとの接続に問題はない。

4.2.2 営業店端末のバックアップセンタ接続

バックアップセンタと各営業店間の接続は、営業店側に用意されている回線障害時のバックアップのためのINS64をバックアップセンタに振り向ける事で行っている。

各接続形態ごとのバックアップセンタ接続を図8に示す。

4.2.3 対外センタのバックアップセンタ接続

特定業務バックアップで選定された対外センタととの間の業務には他金融機関との間の預金ネット取引および為替取引がある。

信金業態では、上記取引は共同RE（地区センタ内に設置されているコンピュータ）

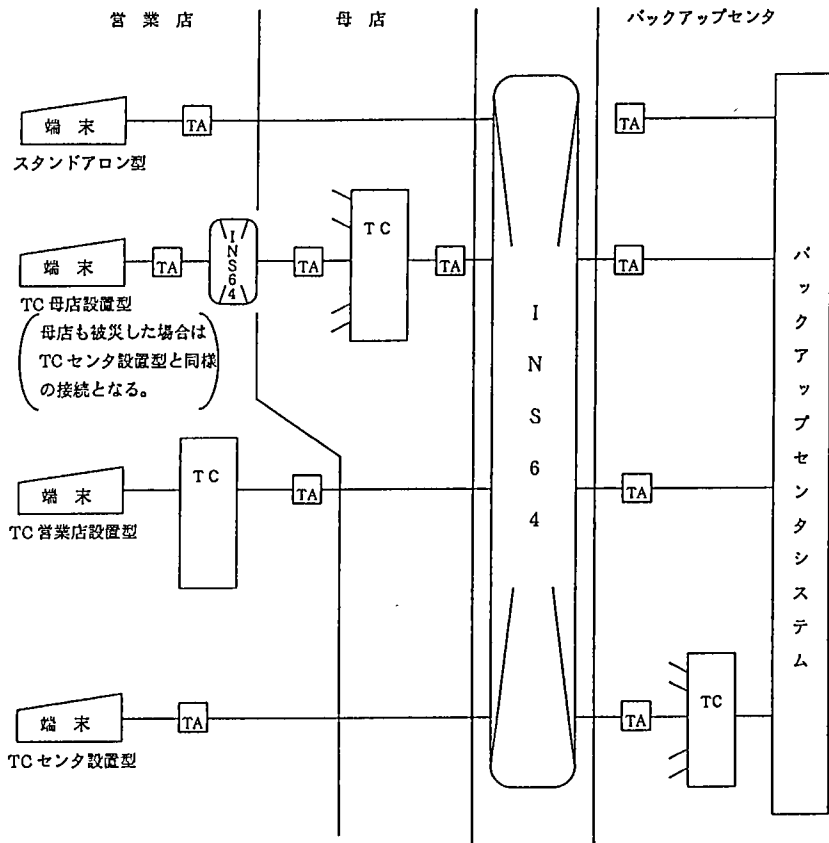


図8 営業店とバックアップセンタとの接続

を經由して行われる。

該共同REと自営センタとの間は、専用線で結ばれており、回線障害に備えてINS64による回線バックアップの考慮が施されている。

自営センタ被災時は、このINS回線を信金バックアップセンタに用意された共同RE専用のTAを通して接続することにより当該信金の対外センタ接続を可能としている。

接続図を図9に示す。

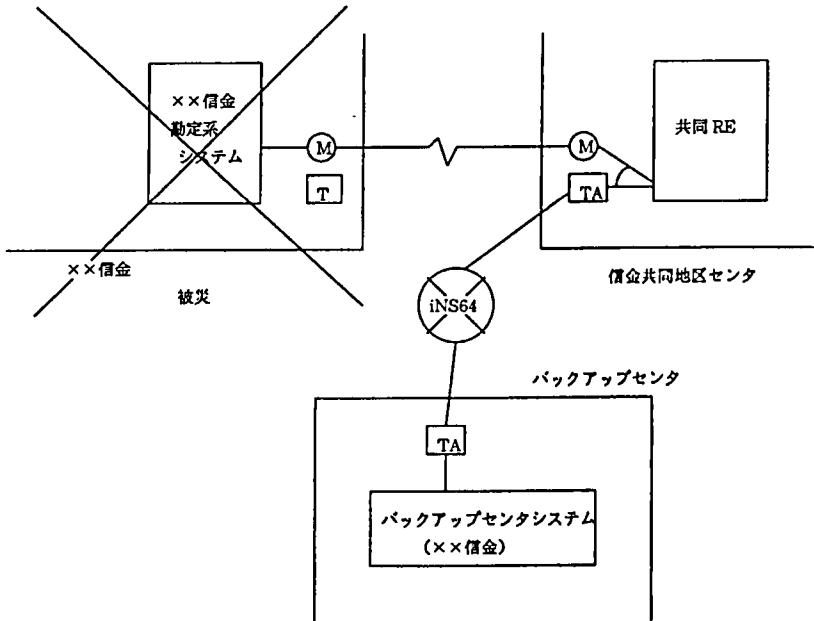


図9 対外センタとバックアップセンタとの接続

4.2.4 伝送制御手順と電文仕様

ホスト・端末間のインタフェースは、データ伝送を中心とした“伝送制御手順”と各々のアプリケーションに依存した“電文仕様”とに分けることができる。

伝送制御手順は、表1に示す6種類を用意することで勘定系システムをユニシス・マシンで構築している自営信金43金庫の営業店端末接続を可能としている。

また、共同REとの間の伝送制御手順は各ホストマシンの機種毎に統一されており、2200シリーズ用に‘SCCP-I’、Aシリーズ用に‘B-POOL’が用意されており自営43金庫全てで接続可能としている。

電文仕様は端末メーカ、端末機種によって異なっており、同一端末機種においても各客先ごとに相異している。

バックアップセンタシステムは統一アプリケーションであり、当然入力電文、出力電文に関しても電文仕様は、統一の形式を前提としている。そのため、各客先ごとの電文仕様とバックアップセンタシステムでの統一電文仕様との変換を行うプログラム

表1 バックアップセンタ保有伝送制御手順

伝送制御手順名 (2200シリーズでの名称)	対象端末機種	客先数	接続形態
DPM1-ABM	沖電気 2300	10	TC型
	ユニシス FSA(2200接続)	1	スタンドアロン型
DPM1-NRM	オリベッティ M380	1	スタンドアロン型
	ユニシス FSA(2200接続)	1	スタンドアロン型
DPM1-ARM	日立 FBT	1	TC型
1200 II	沖電気 1200 II	15	TC型
B-POLL	ユニシス FSA(Aシリーズ接続)	13	スタンドアロン型
U-BSC	オリベッティ M54	1	スタンドアロン型

を各客先ごとに作成し対応している。

4.3 バックアップセンタ業務処理プログラムの仕組み

4.3.1 異機種端末接続

伝送制御手順は各端末の仕様に則ったものをバックアップセンタシステムの中に用意しておくが、電文仕様に関しては、各端末の形式とバックアップセンタシステムの規定する形式（標準形式）との変換を行うプログラムを各客先ごとに用意し業務処理プログラムの統一化を計っている。

電文の流れと変換の流れを図10に示す。

なお、各客先個別の入力変換プログラム、および出力変換プログラムは4.3.2項で述べる客先個別の系列に、各客先のもが同一の名前で格納されており、どの客先の業務処理を行う場合でもプログラム構造は同一となっている。

4.3.2 複数ユーザサポート

通常システムを構築する場合、自分の所の環境に合わせて構築をしていくわけであるが、バックアップセンタでは、特定多数のユーザが存在し、なおかつ3ユーザ同時稼働を保證する必要がある。

どのユーザがどの組み合わせで稼働するかは、事前に一意に決定する事はできないため、バックアップセンタでは以下の仕組みを作る事で対応している。

- 稼働環境を3種類持つ必要があるため、各々必要なファイルを修飾子*を分ける事で3系列持つ。
- バックアップセンタシステムで共通に使用される要素を保持する系列をやはり修飾子を分ける事で1系列持つ。
- 各ユーザ固有の情報をもつため、各ユーザ別のファイルを修飾子を同一として1系列持つ。

稼働環境構築時にそのユーザがどの環境で稼働するかによって上記5系列より一つの稼働環境を組み立てる。

* 修飾子：同一ファイル名をもつ複数のファイルを区別するためのもの

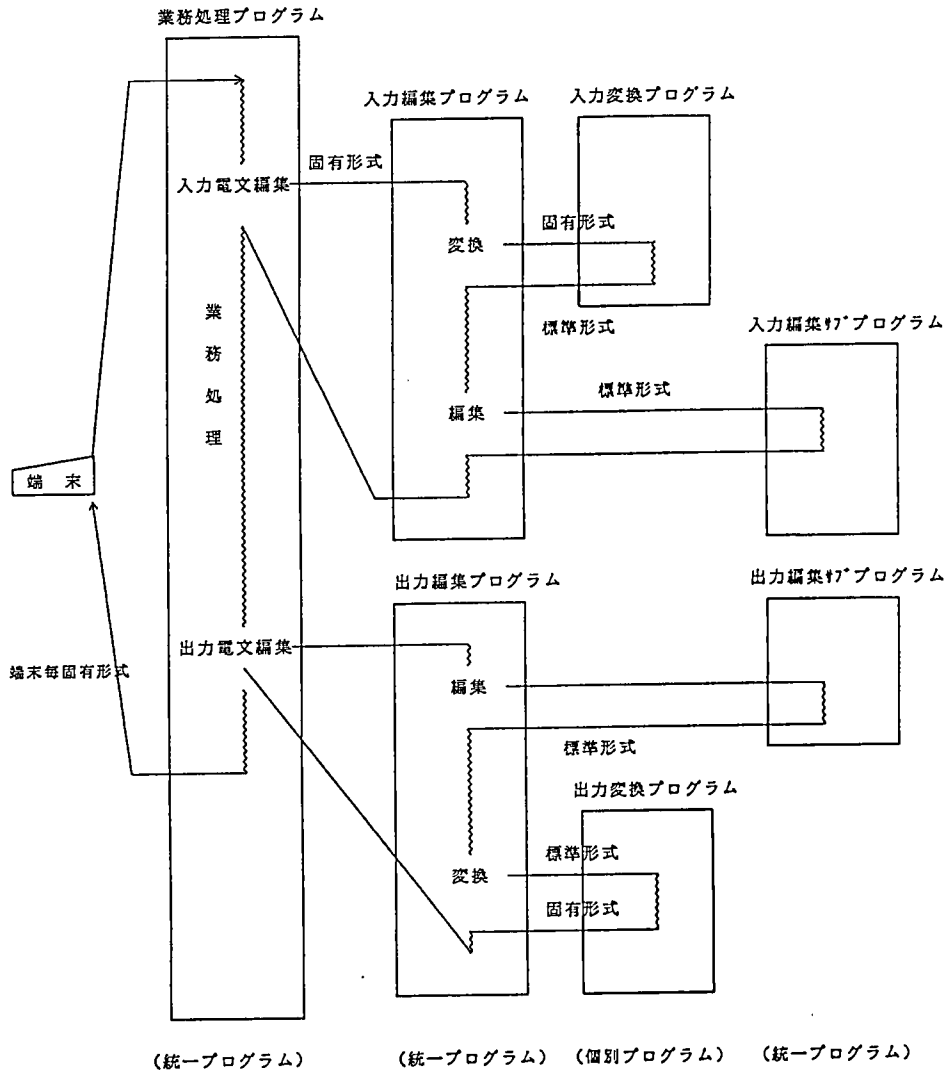


図10 電文変換の流れ

これによって不特定同時3ユーザ稼働を実現している。

各系列の具体的な内容は次のとおりである。

1) システム共通系列

バックアップシステムとして、共通に使用できるエレメントの集合体であり、
 ‘BUC0*’ という修飾子で修飾されるファイルの中に格納されている。

- 内容は ・ほとんどのアプリケーションプログラム
- ・システム共通パラメータ 等である。

2) 個別稼働系列

稼働環境に左右されるエレメントの集合体であり、‘BUC1*’, ‘BUC2*’,
 ‘BUC3*’ という修飾子で修飾される三つのファイル系列の中に格納されてい
 る。

- 内容は ・ BOSS*, DMS**等稼働環境に依存するもの
 ・ TIPプログラム番号***, TIPファイル番号****等システム一意に
 設定しなければならないもの 等である。

3) ユーザ個別情報系列

端末仕様, 通信プロトコルに影響をうける部分, および客先環境に依存するエ
 レメントが格納されている。

- 内容は ・ 伝送制御プログラム
 ・ 端末運用プログラム
 ・ 電文変換プログラム
 ・ ネットワーク構成パラメタ
 ・ D/Bサイズパラメタ
 ・ 金融機関店舗パラメタ 等である。

これらの系列の構成を図11に示す。

稼働環境構築は, 次に示す手順に従って行う。ここでは1番目の稼働環境でUSER-
 ABCなるユーザのバックアップシステムを構築する例で説明する。

- ① BUC1 * PF××\$ UPに, BUC0 * PF××\$ UPの内容をすべてCOPYする。
- ② BUSER * USER-ABC内にある個々のエレメントを所定のBUC1 * PF××

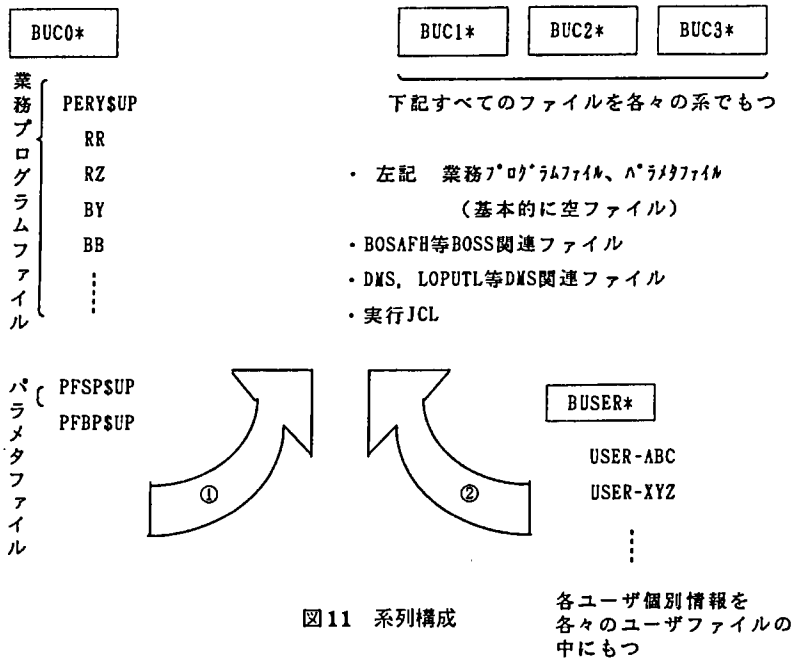


図11 系列構成

* BOSS:リアルタイム・バンキング・サポート・システム
 ** DMS:データベース・マネージメント・システム
 *** TIPプログラム番号:トランザクション処理を行う使用者プログラムにつけられた番号
 **** TIPファイル番号:TIPファイルコントロールを通してアクセスするファイルにつけられた番号

\$ UP内にCOPYする。

③ BUC1 *の系列の中で

- ・業務処理プログラムの生成 (コンパイル, マップ)
- ・客先固有のテーブルウェアの生成
- ・客先毎異なるサイズのファイルの確保 (D/Bファイル)

等を行い, 稼働環境を構築する。

2番目, 3番目の稼働環境で構築する場合は, 上記手順のBUC1 *がBUC2 *,

BUC3 *にかわるだけである。

4.4 復旧処理

自営センタが復旧した場合(ホストコンピュータが使用可能となった場合)は, 被災した時点からの取引を反映して復旧当日のデータベース確定作業を行う必要がある。

ここでいう取引とは

- ① バックアップセンタで処理した取引
- ② 手作業で処理した取引 (定期預金の払出し等)
- ③ 何も処理しなかった取引 (未処理分センタカット等)

の3種類が存在する。

これらの取引の反映方法は

- ① バックアップセンタへ送付したバックアップテープより自営センタ勘定系D/Bを構築。これで被災当時のD/Bとなる。
- ② バックアップセンタより返却された日々の取引テープより日付をおっかけながら反映処理を行う。
- ③ 手作業分, 未処理分を起算日付で処理。

の順で行う事で復旧当日のデータベースを確定する。

5. 今後の課題

5.1 通信網の整備

バックアップセンタと営業店および対外センタとの間の接続は, 自営センタとの専用線の予備として敷設してあるiNS64を経由して行っている。

今回の阪神・淡路大震災の例を見ると, 公衆回線ほどではないにしても, 輻輳状態が発生しており, 地上回線が寸断された場合等も考えて, 例えば通信衛星を利用したパケット通信回線等も考慮する必要がある。

ただし, バックアップセンタ単独で考えても意味のないことであり, 営業店/自営センタ/対外センタをあわせた形で考えていく必要がある。

5.2 端末の標準化

営業店の端末の種類は多種多様であり, その制御 (伝送制御手順, 電文規約) も端末の種類あるいはそれ以上存在することは前に述べたとおりである。

バックアップセンタシステムの中でいちばん複雑なことは, これらの多種多様な制御をいかに行うかであり, 一度構築したシステムも, 営業店端末の更改等で端末の種類が変更となった場合, 再度バックアップシステムを構築しなおす事も考えられる。

規格を統一した (例えばWINDOWS配下で動くPC) 端末をバックアップセンタ用

に営業店に配備し、通常はPCとして利用し、緊急の場合は当該端末をバックアップセンタ用端末として使用する等（自動機の接続に問題は残るが）も今後考えていく必要があると思われる。

また、将来のWOSA/XFS*等による営業店端末の規格統一も考慮すべきと思われる。

6. おわりに

先の阪神・淡路大震災において幸い長期的停止には至らなかったものの、多数の金融機関が被災し、一時システム停止となった地銀、信金が存在した。

災害は起こるかもしれないではなく、必ず起こるとの考えで発生したらこうすればよいとの備えを十分にしておく必要がある

本稿では、その一つの方法である第三者委託による特定業務のバックアップ方式について紹介してきた。

今後バックアップを考えると、データの保存、バックアップセンタの業務内容範囲、端末接続は切り離して考えることも可能である。

データの保存に着目すれば、最新の技術動向により

- ・リモートディスクミラーリング
- ・データベース更新情報の遠隔地への転送

等の機能により、被災直近のデータの保障も可能になってきている。

各々の特徴、メリット（コスト）、デメリット（リスク）を再整理し、それらを有機的に結びつけて、さらに検討していきたい。

-
- 参考文献 [1] 「信用金庫業界におけるバックアップシステム構築の必要性について」、平成6年7月、社団法人 全国信用金庫協会。
 [2] 「業界バックアップシステムの現状と課題」、社団法人 全国信用金庫協会。
 [3] 「金融機関等におけるコンテンジェンジープラン（災害時等の緊急時対応計画）策定のための手引書」、財団法人 金融情報システムセンタ。

執筆者紹介 関 俊 夫 (Toshio Seki)

昭和50年武蔵工業大学工学部経営工学科卒業。同年、日本ユニシス（株）入社。金融アプリケーションの開発、サービスに従事。現在、金融システム第二本部信金システム開発センタに所属。



* WOSA/XFS：Windows Open Services Architecture / Extensions Financial for Services (Windows上の金融システム向けアプリケーションインターフェース)
 Windowsは、米国Microsoft社の商標である。

食品物流センタにおける IND・IE の適用

An Application of the IND-IE System at a Food Distribution Center

平野 静夫

要約 昨今の消費動向の変化から「価格破壊」が流通業界のキーワードとなっており、低価格な商品を消費者に提供することが、小売業にとって生き残りの条件となっている。物流の効率化が、経営上の重要課題として取り上げられて久しいが、消費者の低価格指向に応えるため、企業単独の物流改善のみならず、川上から川下までの連携した物流改善が模索されている。

本稿では、特定のスーパーマーケットチェーンとの連携による日本型 ECR⁽¹⁾ (Efficient Consumer Response) の実現を目指した食品物流センタのシステム化事例を紹介する。

Abstract Changes in the recent consumer movement have given birth to the buzzword "price destruction" in the wholesale and retail business communities. It has become requisite for the survival of retailers to offer low-priced products and goods to consumers. It has been long since the higher efficiency of commodities distribution was taken up as a new challenge for individual wholesale/retail businesses, and in order to react to consumer's preference for lower prices, efforts are under way not only for a distribution reform by each company alone but for an integrated distribution innovation that covers all the aspects ranging from uppermost stream to lowest.

This paper presents an instance where a food distribution center has been computerized in cooperation with a certain supermarket chain group with the aim of letting the system provide Japanese-style efficient consumer response (ECR).

1. はじめに

「ECR」が近年における流通業界のキーワードとなっている。ECR (Efficient Consumer Response) は、「効率的な消費者への対応」と直訳され、米国での模範事例となったウォルマート社と P&G 社の記事は我が国でも注目され、色々な角度から取り上げられた。

また、同時期に日本では、花王とジャスコの EDI 取引が始まり、味の素とダイエーの商品開発共同化の発表やセブンイレブンのチームマーチャンダイジングの動きが紹介されるに至ると、流通業界はこれこそが低成長下における生き残り戦略であるとして、様々な書物に製販同盟の行方が論じられた。バブル崩壊の後消費者の価格志向が強まったこともあり、この結果として、またしても問屋無用論が浮上してきたのもこの時期である。しかし、ウォルマート社と P&G 社の組み合わせのような製販戦略同盟に代表される ECR は、あくまで米国経済の仕組みに適用される理論であり、市場条件や産業構造の異なる我が国においてそのまま適用できるものとはいえない。ECR の厳密な定義は別として、ECR が唱える最重要テーマである「共同で効率的な商品供給システム」を作り上げるという点から考えたとき、我が国では必ずしも製販

同盟が解答とはならない。

「食品業界のECRを推進するのは卸売業である」として大手卸売業が手を挙げた。自社物流を目指した大手チェーンストアなど小売業も意思に反して増え続ける物流コストに気づき、卸売業とのパートナーリングによる最適機能分担をもう一度考え始めている。

本稿ではスーパーチェーンとボランタリーチェーンストアを対象に、新しい形で物流サービスを展開し、ECRの実現を目指している事例とそれを支援する情報システムを紹介する。

2. システム化の背景

株式会社 小網は酒類と加工食品全般を取り扱う卸売業であり、その一部門であるRS部は小売業とのパートナーリングを推進する組織として機能している。該社の得意先であるスーパーマーケットチェーン（以下SCと記述）は、ローコストオペレーションの手段として、一括物流による店舗納品の効率化を目指し、高度な物流機能を持つパートナーを求めている。

今回該社RS部とSCが先に述べたECRの考え方にに基づき提携関係をより強め、共同で効率的な物流システムの構築を目指すことになった。該社はSC専用の物流センタを建設し、従来該社の関連会社が供給していた日配品に加え、菓子と加工食品も一括納入できる体制を整えることになった。SCと従来取引のあった卸売業との帳合関係から直接取扱不可能な商品群についても、後に述べる納品代行取引によりSCは仕入先の違いを全く意識せず一括発注でき、EDIをベースとした毎日発注/毎日納品と365日24時間体制でSCの期待する物流機能を実現できることとなった。オープン価格制への移行が進展する中で、SCは品揃え強化と物流効率化によるローコスト運営を求め、卸売業である該社は新しい物流サービスを提供することで小売業の要求に応え、双方がメリットを享受しながら、最終的に消費者への効率的対応を目指す戦略的同盟がスタートした。

3. システム化の要件と方針

該社の物流センタ（以下当センタと記述）開業に伴い、情報システム構築の必要性が生じた。システム化における要件は以下の通りである。

- 1) センタ開業日は、該社とSC両者の要望で平成6年10月と決まった。システムの要求定義が開始されたのは同年3月、開発期間は全工程を通し6か月であり、これは最も優先されるべき制約条件となった。
- 2) 得意先および仕入先との取引ルール（受注から出荷のタイムチャート、発注方法、決済方法など）を規定し、それに合わせたシステム作りを行う。当センタは複数の組織が共同で効率化を目指すため、ビジネスプロトコルの全てをゼロから作り上げる必要がある。
- 3) 得意先および仕入先とは、いわゆるEDIによってデータ交換を行う。

以上から、システム化の方針を次のように定めた。

- ①) 短期開発を最優先とする。そのため、当社の物流情報ソリューションパッケ

ージ「IND・IE」*の適用を基本とした開発を行う。要求定義は、IND・IEの個々の機能を評価しながら要/不要の決定とカスタマイズ仕様を固めていく形ですすめる。

- 2) 稼働後の変更にも耐える柔軟性を持ったシステム構造とする。新しいセンターのため、運用形態の変更があり得ると考えられるためである。もともとIND・IEの開発ツールであるLINCの柔軟性に加え、カスタマイズ（追加）する機能を外付けにすることで、この要件に対応する。

4. 物流ソリューションパッケージIND・IE概要

当システムは、IND・IE/LOG、IND・IE/SALをカスタマイズしたものである。当システムの内容の理解を助けるためにIND・IEの特徴を以下に説明する。また、IND・IEが提供する業務サブシステム（受注・発注・在庫・売上・仕入）の機能と各サブシステムの相互関連を図1に示す。本図では、卸売業としての基本システムの機能と情報の流れを説明している。

次にIND・IE開発時のねらいを若干説明する。IND・IEは、量販店を主要得意先としている年商100億円以上の卸売業をモデルとして作成されている。以下、モデルとした卸売業の要求とIND・IEでの対応について主な特徴を述べる。

モデルの主たる要求	IND・IEの対応
<ul style="list-style-type: none"> ・EOSによる取引が多く、多様な受注データとターンアラウンド伝票に対応できること。 	<p>EOS受信ファイルのフォーマットをデータベースに登録し、登録内容に従い受信先ごとにフォーマットの異なる受信ファイルをIND・IEの受注共通ファイルに変換する。 納品伝票のフォーマットデータベースに登録し、登録内容に従い得意先指定伝票を出力する。</p>
<ul style="list-style-type: none"> ・得意先への納品リードタイムが短く、多頻度少量の納品ができること。 ・取り扱い商品アイテム数が多く、在庫管理と発注の仕組みにより、適正な在庫を維持できること。 	<p>受注からピッキング開始までの処理を短期間で行えるよう設計されている。 補充受注のための発注点・最大保管量を過去の受注実績を用いて自動洗い替えし、適性な発注勧告を行う。</p>
<ul style="list-style-type: none"> ・取引データが多量であることから、請求・支払の自動照合ができること。 	<p>得意先、仕入先の仕入・売上データとIND・IEの売掛・買掛データをプログラムで照合しその結果をアンマッチリストに出力する。</p>

5. システム概要

5.1 物流センターの機能/特性

当センターは、特定スーパーマーケットチェーンに対する一括配送を目的として設立された。

- 1) 取扱い商品
 - ①加工食品（以下、加食と記述）
 - ②菓子
 - ③日配品
- 2) 取引形態

*IND・IE：INtegrated Distribution・Information Enviromentで、物流センター向けのソフトウェアパッケージである。

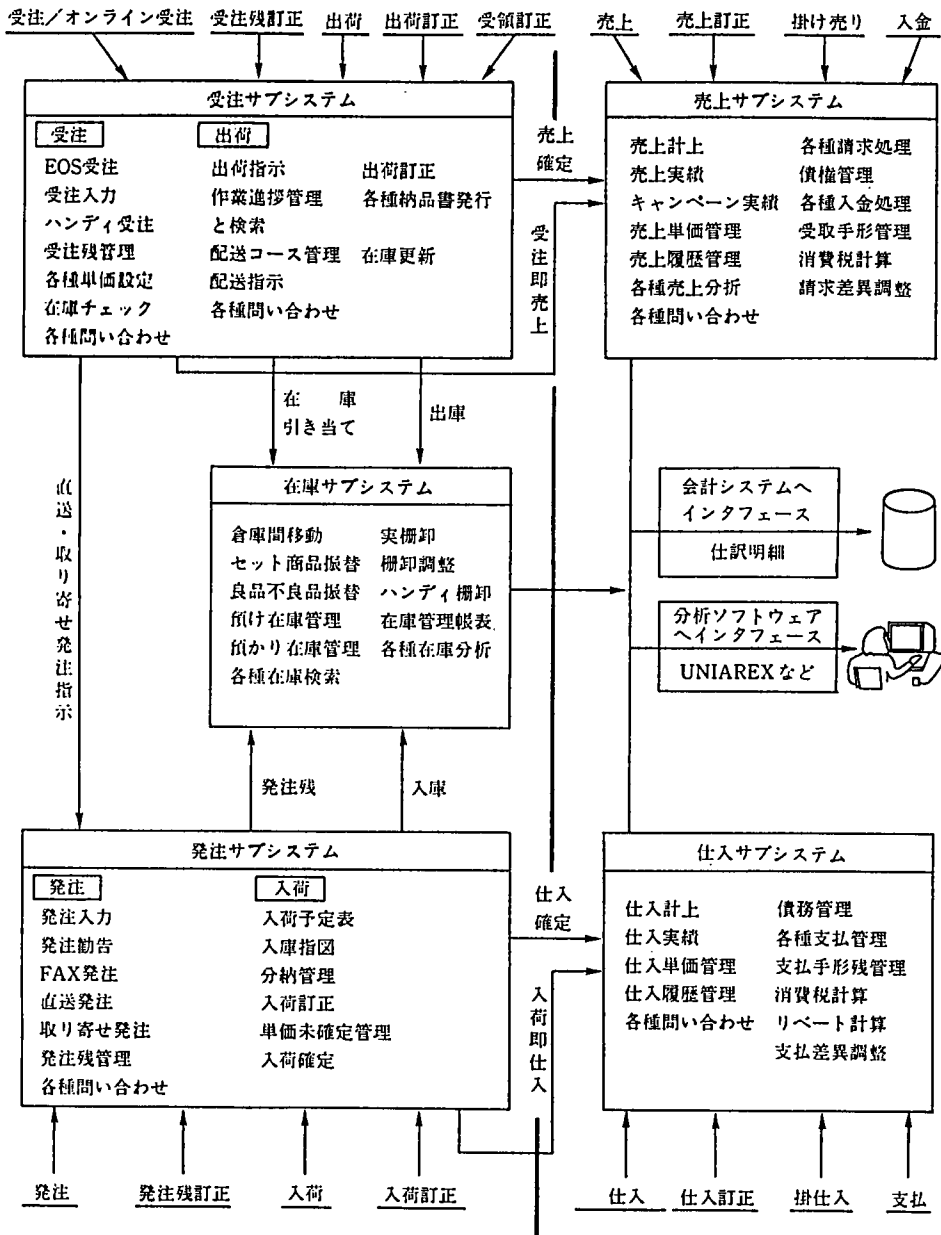


図1 サブシステム関連図²⁾

当センタの取引形態として、納品代行取引（以下代行取引と記述）と帳合取引がある。図2に取引のイメージを示す。

代行取引と帳合取引のそれぞれに対応する当センタの機能は表1の通りであり、物流面での大きな違いは見当たらない。

3) 物流形態

当センタの物流形態は、在庫型物流（以下DCと記述）、通過型物流（以下TCと記述）の2種類に分かれている。図3にDC、TCの概念図を示す。

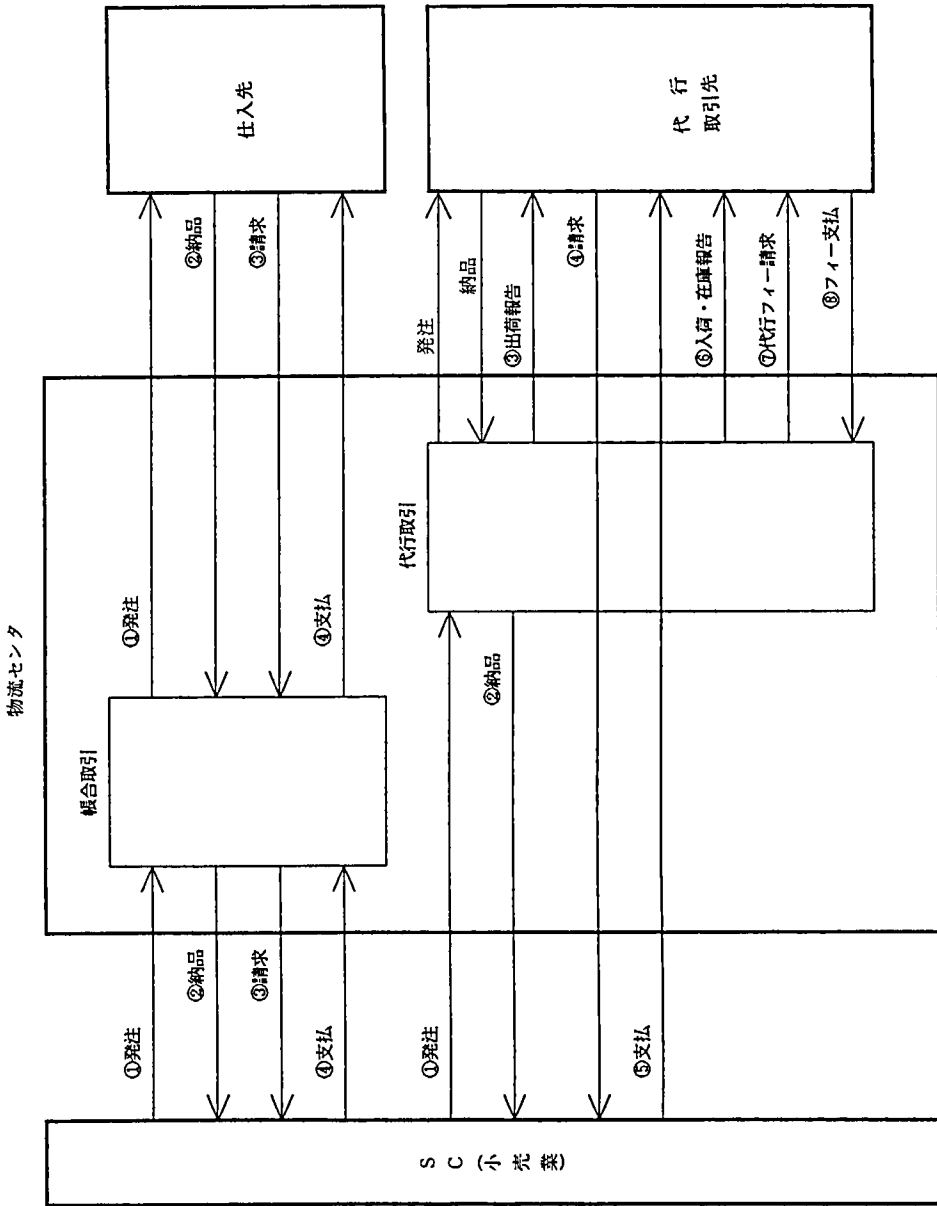


図2 帳合・代行取引概念図

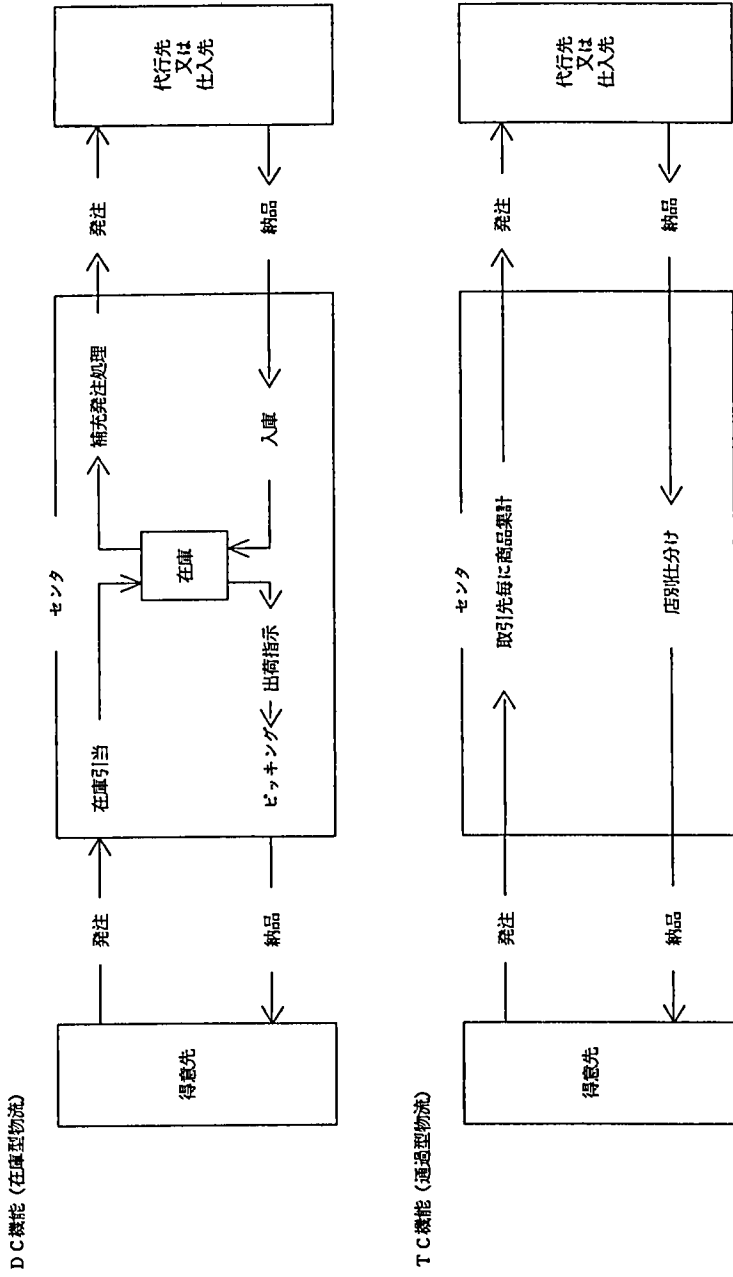


図 3 DC, TC機能概念図

表1 物流センターの機能と取引関係

機 能	代行取引	帳合取引
受注 (EOS 95%, FAX 5%)	○	○
ピッキング	○	○
物流加工	○	○
店別仕分け	○	○
納品伝票発行	○	○
配送	○	○
在庫管理	○	○
発注	○	○
入荷	○	○
売掛請求/買掛支払		○
納品代行フィー請求	○	○
代行先への取引報告 (売上, 入荷, 在庫等)	○	

① DC機能

加食・菓子の定番取引に適用される。センター内に在庫を保有し定番受注に対し商品をピッキング、店別に仕分け後、店舗に納品する。また、在庫商品の補充発注や棚卸など在庫管理機能を持つ。

② TC機能

加食・菓子の特売と日配品の取引に適用される。センター内に在庫は保有せず、受注データを商品ごとに集計し発注を行う。発注した商品は入荷次第、店別に仕分けされ、店舗に納品される。

4) 代行取引における補充発注

代行取引対象の在庫品は代行取引先からの預かり品である。当センターでは、代行取引においてもセンターが責任をもって在庫管理を行う結果、

- ① 過剰在庫
- ② 鮮度切れ
- ③ 得意先への欠品

に対する解消義務があり、一般の共同配送センターと比べリスクが高い一方、一括物流機能を有する当センターで一元的に実施される業務品質の向上が得意先へのサービス品質の向上に直接つながることが利点となっている。

5) 業務運用時間

当センターは元旦を除き、毎日納品体制をとっている。配送便は図4に示す通り、商品カテゴリ別に1日3便制をとっており、それに対応するため業務は24時間体制で運用されている。センターの主要業務であるピッキング作業が夜間に行われる事がセンターの作業上の特徴である。

得意先60店舗はA、B二つのグループに分かれており、受注データもそのグループ単位でEOSにより受信される。

6) ピッキング処理の流れ

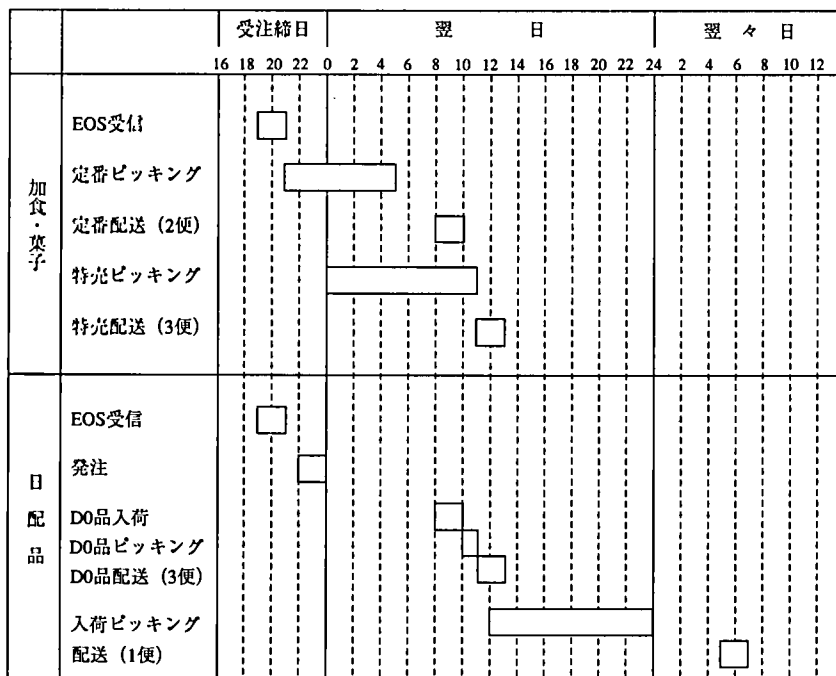


図4 物流作業タイムチャート

当センタの庫内レイアウトは図5に示す通りである。
各エリアとピッキング形態の関連は以下の通りである。

加食Aランク	バラ品保管エリア	グループ別トータルピッキング
菓子Aランク	バラ品保管エリア	〃
加食B, Cランク	ケース品保管エリア	店別摘み取りピッキング
菓子B, Cランク	ケース品保管エリア	〃
加食B, Cランク	バラ品保管エリア	〃
菓子B, Cランク	バラ品保管エリア	〃
輸入品保管エリア		〃
加食Aランク	ケース品保管エリア	〃
菓子Aランク	ケース品保管エリア	〃
日配品		種蒔きピッキング (さみだれ式)

ピッキング効率を考慮し、出荷頻度の高いAランクバラ品をトータルピッキング、その他を摘み取りピッキングの対象にしている。

エリアごとのピッキング作業手順は図6に示す通りである。

5.2 システム機器構成と主要ソフトウェア

図7にシステム機器構成を示し、構成上留意した点について次に述べる。

- 1) 障害防止対策として、データベースが割り当てられているディスクをミラー化している。
- 2) EOS受信時間を短縮するために、JCAの回線スピードをINS NET64を利用し

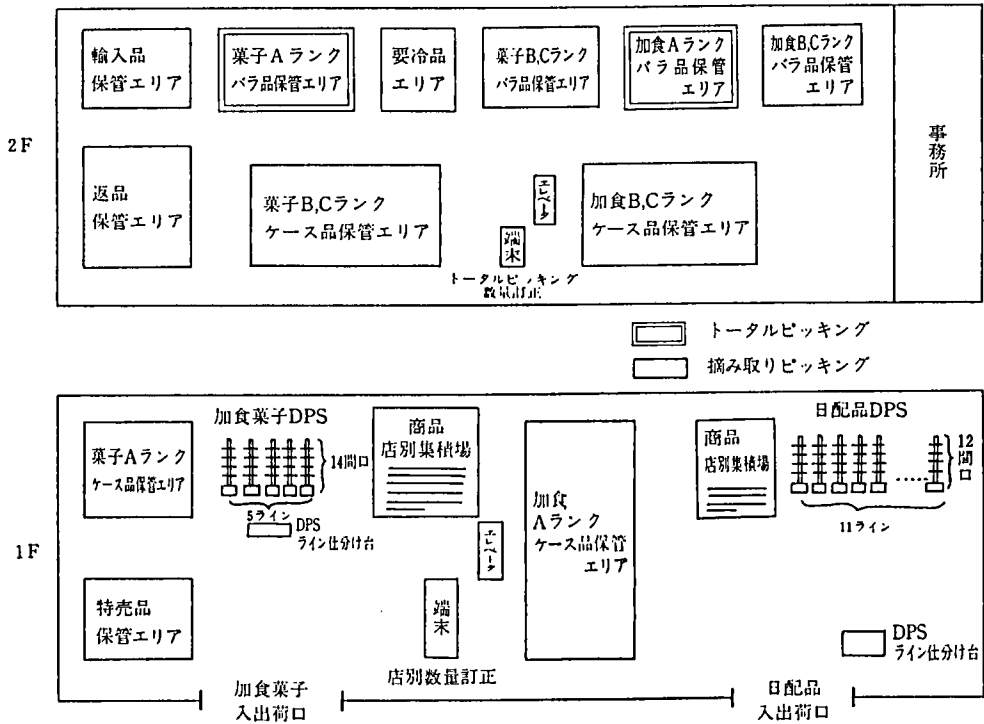


図5 庫内レイアウト

9600bpsにした。

- 3) 障害発生時の早期対応のため、当社とリモートサービス回線を用意している。
- 4) 発注先（即ち仕入先または代行取引先）への発注データ送信のためオンライン回線（JCAプロトコル）の他にファックス送信装置を備えている。
- 5) 機器名称概略と主要ソフトウェア等

A11：当センタ専用メインフレームで、カートリッジテープ装置および磁気ディスク9.22GBを内蔵

OS：MCP

言語：LINC, COBOL

データベース：DMS II

CP2000：通信制御装置

プリンタ（AP4）も同装置経由で接続。リモートプリンタ化も可能である。

X310-91：磁気テープ

PW24253：DOS/V PC

エミュレータによりT27-KNの役割も兼ねている

T27-KN：端末

H/T：フルノ社製ハンディターミナル

DPS：イシダ社製デジタルピッキングシステム

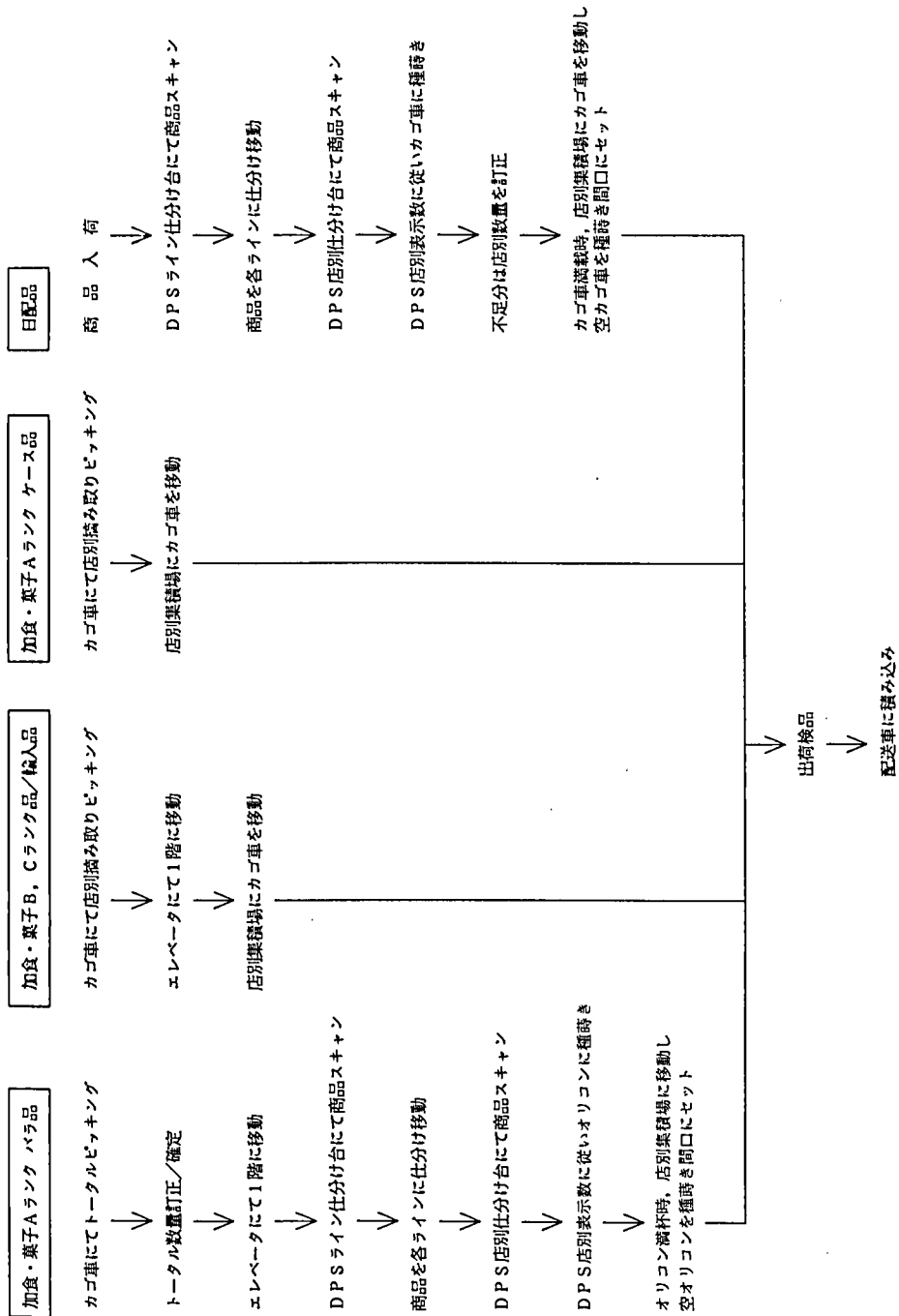


図6 ピッキング作業手順

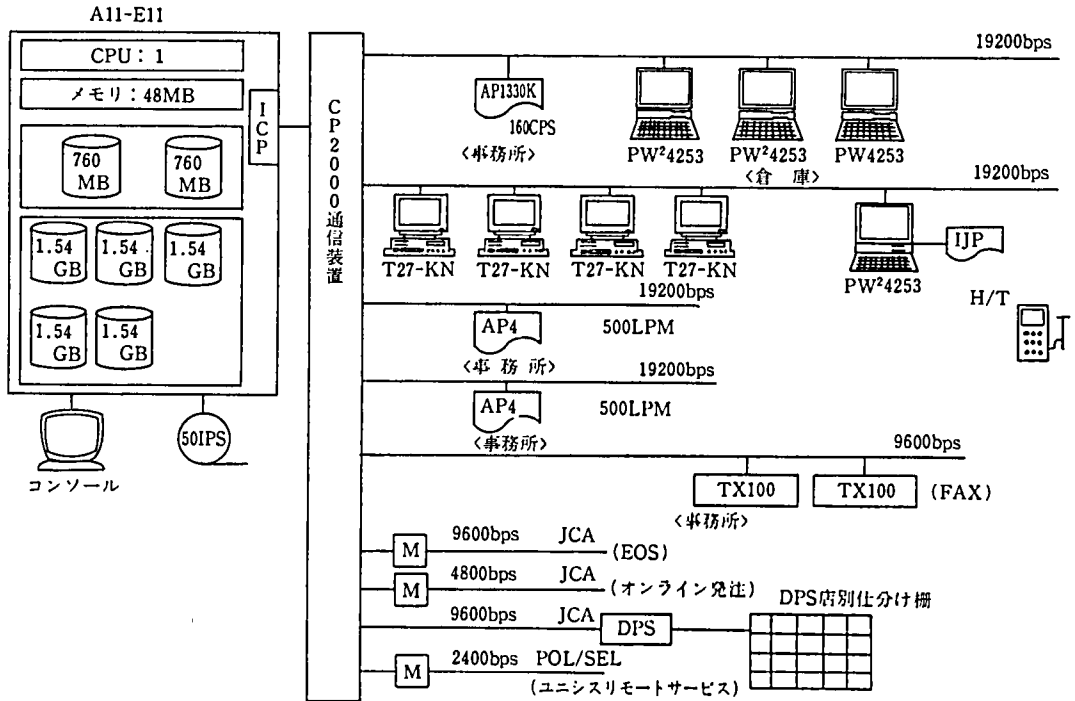


図7 システム機器構成図

TX100：ファクシミリ送信装置

5.3 システム機能概要

当システムは、大別して業務処理システムと共通基盤とで構成されている（図8参照）。各々の内容は以下のとおりである。

1) 業務処理システム

- ① LOG：物流情報システムを表す
- ② SAL：販売管理システムを表す

2) 共通基盤

① ソフトメニュー

メニュー画面およびメニュー構造をファイルに登録し、登録内容に従い画面を移動、プログラムを実行する機能を持つ。

② EOS受信

得意先ごとに異なるフォーマットの受信ファイルをIND・IEで処理する受注フォーマットに変換する汎用的な仕組み。受信ファイルのレイアウトは、受信ファイル書式定義ファイルに登録する。

③ 伝票編集/出力

得意先ごとに異なるフォーマットの納品伝票を編集/出力する汎用的な仕組み。納品伝票のレイアウトは、伝票書式定義ファイルに登録する。

④ FAX出力編集

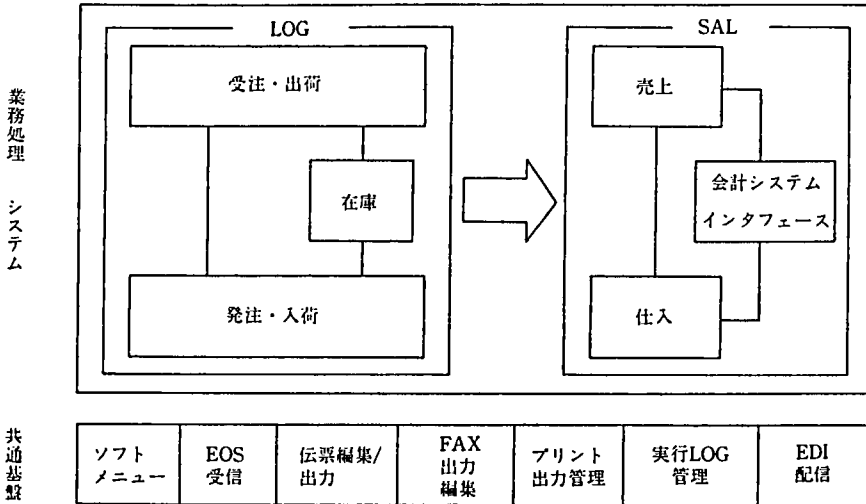


図 8 システム機能関連図

プリントイメージファイルをTX100（ファクシミリ送信装置）を通じて送信する。

⑤ プリント出力管理

帳表プログラムによって作成されたプリントバックアップファイルを管理し、画面からの指示に従い、プリンタに出力する機能を持つ。

⑥ 実行LOG管理

各プログラムの開始/終了およびプログラムの出力するメッセージを格納/照会する機能を持つ。

⑦ EDI配信

代行先取引内容をオンラインで報告するための配信ファイルを作成し、配信完了までを管理する機能。

以下、業務処理システムの特徴的な機能を説明する。

5.3.1 受注・出荷

1) EOS受注処理

EOS受注処理の主要機能は以下の通りである。

① EOSデータを受信し、当システムの受注共通フォーマットに編集する。

② 受注共通フォーマットの得意先商品コードを自社商品コードに変換する。変換時にエラーが発生した場合、出荷数を0とし、ピッキング対象から除外する。変換エラーは以下の2項目である。

- ・自社商品コードに変換できない。
- ・納品日が商品の取扱い開始日、停止日の範囲内でない。

③ 得意先との取引条件に基づき、受注データのチェックを行う。

(データチェック例)

④ 加食・菓子の特売品は、在庫品からの出荷ではなく、受注データを商

品別に集計・発注して取り寄せによる品揃えを行ってから出荷する形態をとっている。加食・菓子の特売受注で、納品日一受注日 $<N$ のときは出荷数を0として、発注・ピッキング対象から除外している。発注から入荷までのリードタイムを考えると、上記条件では商品手配ができない場合があるため、ビジネスプロトコル上、不正データとみなすためである。

このことは、業務上においてもシステム上においても、当センターの性格を良く表現している。これは、(様々な批判は別として)スーパーマーケットチェーンの主たる販売促進策である特売を、欠品なしで確実に実施するために用意された機能である。

- ⑥ SCからのEOS発注データ不備により、誤まった仕入先の商品を出荷することを防ぐためセンターの商品マスタと内容照合を行う。これは預り品の管理を正確に行うためである。

2) 在庫引き当て

図9に示すように、在庫引き当て処理はピッキング開始指示後に行われる。ピッキングリスト上には、受注数と出荷指示数(コンピュータで引き当てされた数)が表示される。ピッカーは出荷指示数によってピッキングするが、その時、実在庫があるにもかかわらず、出荷指示数が受注数を下まわる(すなわちコンピュータ在庫と実在庫に狂いが生じている)ことが発生すると、それはすぐに差異調査にまわされる。当センターは、自社商品だけでなく、代行先の商品も管理しているため、在庫の正確性には、とくに注意をはらっている。

実在庫が出荷指示数を下まわった時は、ピッキングリスト回収時にピッキング訂正入力が必要となる。

3) 欠品配分

トータルピッキングリスト*の出荷指示数にも引き当て数が出力される。引き当て数に基づくピッキングのため、トータルピッキングでの数量訂正の頻度は少ない。在庫管理の不備とロケーション管理ミス等により引き当て数とピッキング数とが異なる場合に数量訂正処理が行われる。

トータルピッキングの場合、訂正の結果、確定した欠品数を店別に自動配分し、出荷数を決定する。

4) DPS(デジタルピッキングシステム)による種蒔き指示

DPSは、加食・菓子のトータルピッキングの2次仕分け(店別仕分け)用と日配品の店別仕分け用の計2システムが使用されている。

DPSに送信している主な項目は、JANコード、商品名、種蒔きロケーションコード(店舗間口を示すコード)、数量である。

DPSに関して発生する主なエラーは以下の通りである。

- ① DPSへの送信データの種蒔きロケーションに不正(送信データのロケーションとDPSが管理している実ロケーションが異なる)がある場合、DPSは受

*トータルピッキングリスト：店別に商品をまとめてピッキングするのではなく、同一配送便に対し倉庫から同一商品をまとめて出庫するときのピッキングリストである。なお、まとめて出庫された商品は、何らかの方法で店別に小分けすることになる。当事例ではDPSで小分けしている。

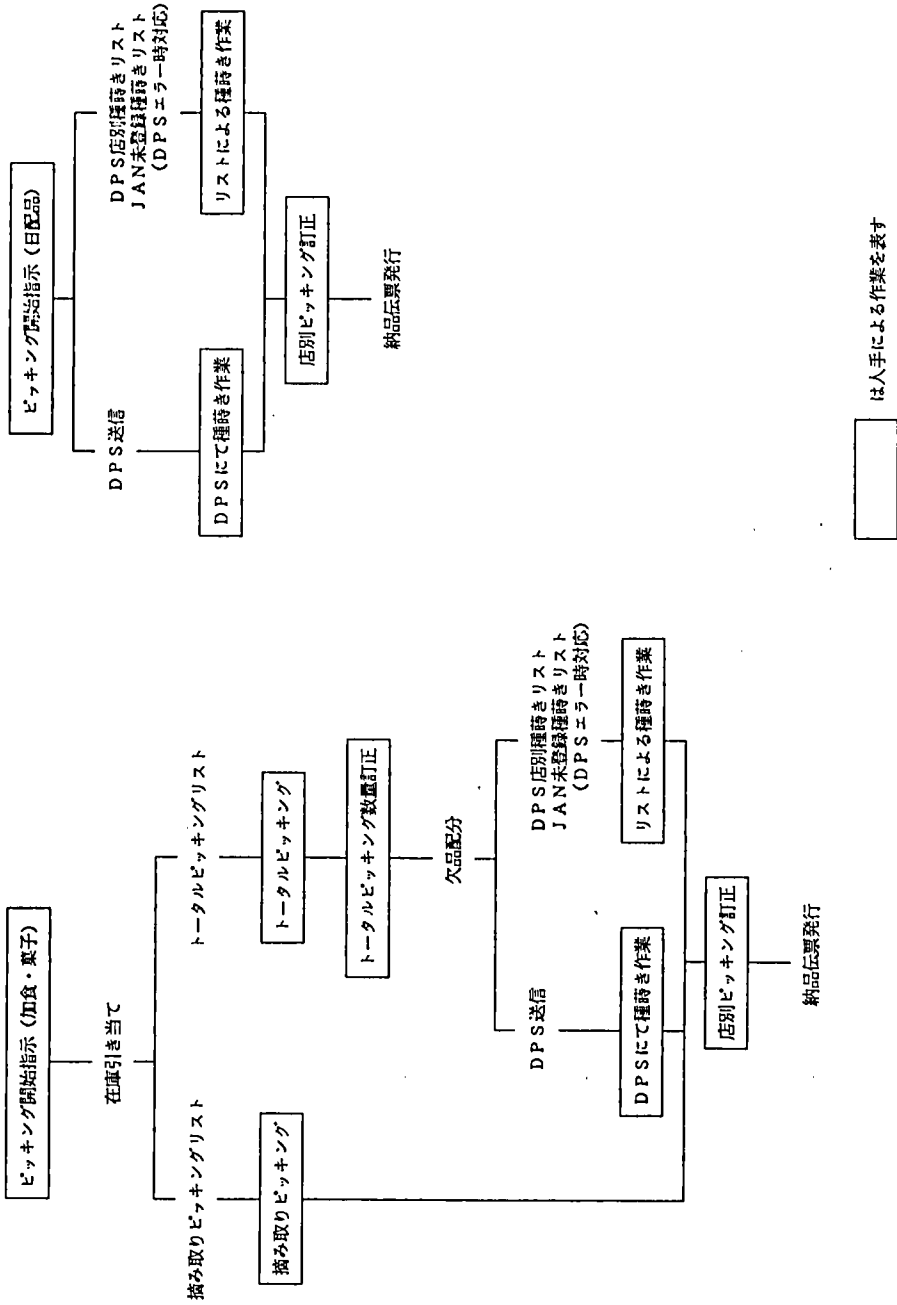


図9 ピッキング開始指示

信エラーとし、正しいデータを受信するまで作動停止となる。エラー時の対応とDPSそのものの異常に備えてDPS店別種蒔きリスト（商品店別に出荷指示数を出力している）を用意している。

- ② DPSへの送信データのJANコードに不正（スペースまたは商品に印字されているJANコードと異なる）がある場合、商品のJANコードをスキャンするときにエラーになる。JANコードエラーの対応として、JANコードスペースエラーにはJAN未登録種蒔きリストを、JANコード不一致エラーにはDPS店別種蒔きリストを用意している。

5) 納品伝票発行

一般の卸売業における納品伝票の取引先名には自社名を出力する。納品代行用伝票の取引先名には、小売店から見た取引先、すなわち、当センタから見た代行先名を出力している。

6) 積載量計算

近年トラックの積載量オーバーが社会問題化し、規制が厳しくなる傾向にある。当システムでは、商品ごとに重量を登録し、配送コース別に積載重量の算出が可能である。

7) 欠品率改善

欠品率改善のために2種類の情報を提供している。

- ① 下記カテゴリ別に欠品率を把握する

(例) EOS変換エラー、新商品、季節商品、補充発注対象品等

- ② 欠品アイテム別の原因を分析する

当システムでは、①のために

欠品数量構成表（図10に帳表イメージを示す）

仕入先別欠品数量構成表

②のために

欠品商品リスト（自社理由、取引先理由別）

欠品商品分析リスト（受注、欠品、出荷、発注、入荷の2週間日別実績）

を用意した。

当センタでは欠品率改善の責任者を定め、これらの資料を日々出力し欠品原因の分析を行うことにより、欠品率の改善に努めている。各資料の具体的な利用方法は以下の通りである。

・欠品数量構成表

カテゴリごとの欠品率を日別推移グラフに転記し、欠品発生の特性（週明けの欠品が多い等）を分析し、対策を検討する。

・仕入先別欠品数量構成表

全体の欠品率に大きな影響を与えている仕入先を把握し、該当仕入先の発注から入荷が取引ルール通りに行われているかを確認する。また、該当仕入先から仕入れる商品に対する発注点等補充発注のための在庫管理係数を見直す。

・欠品商品リスト/欠品商品分析リスト

<< 欠品数量構成表 >>			作成日 07/03/10 08:10
納品日 7年 3月 10日			ページ 1
	受注数	欠品数	欠品率 (%) *
A E O S変換エラー	300	300	0.31 (300÷96300×100)
マスタ未登録	230	230	0.24
取扱停止	70	70	0.07
B 在庫引当不可	1,400	1,400	1.45
短納期特売	0	0	0.00
ロケーション未登録	300	300	0.31
補充発注対象	900	900	0.93
補充発注対象外	200	200	0.21
季節商品	60	60	0.06
新商品	100	100	0.10
その他	40	40	0.04
C 出荷訂正	1,200	160	0.17
短納期特売	0	0	0.00
ロケーション未登録	0	0	0.00
補充発注対象	1,100	100	0.10
補充発注対象外	100	60	0.06
季節商品	0	0	0.00
新商品	100	60	0.06
その他	0	0	0.00
合計 (A+B+C)	2,900	1,860	1.93
D 納品数	93,400		
合計 (A+B+C+D)	96,300		

*欠品率 = $\frac{\text{欠品数}}{\text{合計(A+B+C+D)}} \times 100$
上記欠品率は架空のもので、実際の値とは異なる

図 10 欠品数量分析表

欠品商品リストでは、自社理由による欠品と取引先（得意先、仕入先）理由の欠品に分類されている。

自社理由による欠品の場合は、欠品商品分析リストを用い発注タイミングが正しいか、入荷予定日に入荷しているかの確認を行う。また、取引先理由による欠品の場合は、取引先に対し改善すべく勧告する。

8) 伝票レス取引

センタ稼働1年後の平成7年10月よりSC内での最大得意先であるA社とは納品伝票を廃止し、いわゆる「伝票レス取引」を開始した。これはセンタの安定稼働に伴ない納品精度が向上したことを受け、該社とA社両者の合意で、加食・菓子取引について試行したものである。

A社では、店頭でのスピード検品に加え、店舗からの伝票回収、仕分け、コンピュータ入力削減、該社では伝票購入・保管コスト削減や出荷時伝票仕分け、受領書回収・整理・保管など、いずれも取引合理化とコスト削減に大きな効果をもたらしている。

開始後、混乱や問い合わせもほとんどなく極めて順調に推移している。

5.3.2 発注・入荷

1) 発注形態

当センタの発注形態には、補充発注、取り寄せ発注があり、各々

- ① 補充発注 …加食・菓子の定番
- ② 取り寄せ発注…加食・菓子の特売, 日配品
を対象にしている。

2) 補充発注における発注数の決定

システムによる補充発注対象商品は単品ごとの在庫管理係数として、発注点・最大保管量を登録している。有効在庫数が発注点を割り込むと最大保管量まで補充するよう発注予定数が設定される。発注予定数は担当者が確認しながら、必要があれば修正することも可能である。

3) 発注点・最大保管量の洗い替え

発注点、最大保管量の洗い替えは人手による方法と自動で行う方法がある。

① 人手による洗い替え

コンピュータで算出した発注点、最大保管量および算出の要素となる項目を洗い替え勧告リストに出力し、発注担当者は、各担当仕入先の商品について洗い替えリストの内容を確認する。次に、発注担当者の判断により変更が必要な商品について、現在の発注点、または最大保管量の値を画面から保守する。

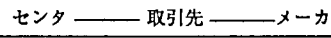
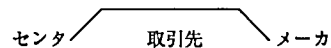
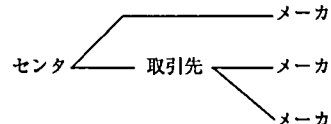
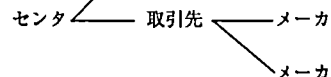
② 自動洗い替え

洗い替え勧告リストを出力し、発注担当者がリストの内容を確認するところまでは、人手による洗い替えと同じである。自動洗い替えとして登録された商品については、コンピュータで算出した新たな発注点と最大保管量でデータベースを自動的に更新する。

商品管理担当者が係数の修正が必要と判断した場合には、画面から保守することになる。

発注点・最大保管量を算出するには、単品ごとの過去の受注実績と、それ

表2 発注先と発注手段

発注先	発注手段	
	オンライン	FAX
取引先に発注		
	○	○
メーカーに直接発注		
	×	○
取引先、メーカー混在発注		
	×	○
	○	○

取引先：代行先または仕入先を表す。
 メーカー：センタにとって直接の取引先ではない、あくまでも取引先とメーカーが取引関係にあり、本来は取引先へ発注すべきだが入荷リードタイムの短縮を目的として直接メーカーに発注している。

をもとにした発注点係数，最大保管量係数が用いられるため，あらかじめ商品マスタに登録されている必要がある。

4) 発注先と発注手段の組み合わせは表2の通りである。

発注の仕組みとしては，発注先に関わらずオンライン，FAXの二つを用意しているが，加工食品メーカは独自のオンライン仕様が要求されるため，FAX送信が中心となっている。

5) 入荷欠品への対応

メーカから事前に入荷情報が入手できる場合，情報に合わせて発注内容を変更し，入荷予定表/入荷ラベルを出力している。

メーカから事前に欠品の連絡がない場合の対応として，1日の入荷が終わる頃未入荷リスト（一部入荷も未入荷リストの対象としている）を出力し，入荷のない発注先に欠品理由を問い合わせしている。また，欠品状態で入荷された商品についても欠品理由を問い合わせしている。

5.3.3 在庫（棚卸）

ハンディ端末に在庫数量の入力を行い，パソコンを通じてホストに取り込んでいる。

棚卸は，毎月最終日曜日の日中に行っている。理由として，

① 日曜日は商品の入荷がない。

② 元旦を除いて毎日納品があるが，在庫品の出荷作業は21時以降に行われる。があげられる。棚卸差異の発生を防ぐため日々のピッキング時に在庫数確認を行うようにしている。

当システムでは，ロケーション別の在庫管理は行っていないが，センタの商品保管ルールが1商品1ロケーションであるため，エリア別に循環棚卸しを行うことが可能である。

5.3.4 売上（売掛管理）

当システムでは一得意先に対し複数の請求先を設定することができる。請求先は，仕入先の支払い締め日パターンごとに設定される。

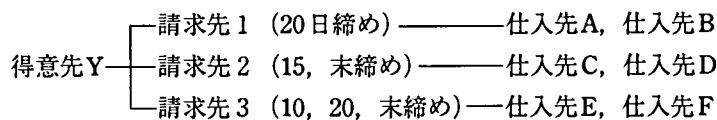
一得意先に複数請求先が発生するのは以下のためである。

① 日配品は商慣習として支払いサイトが短い。

② 仕入先に旬ごとに支払い，得意先からは月1度回収とすると，センタの運転資金が必要になる。

③ 従って，得意先の締め日を商品分類ごとに設定し，回収都度仕入先への支払いを行うこととした。

例)



5.3.5 代行取引

①) 取引先EDI

代行先からの預かり商品であっても、センタが得意先から直接受注を受け、代行取引先が介在せずに預かり在庫から出荷しているため、代行取引先では売上等取引の内容を把握することができない。このため、センタから代行取引先に対し毎日の取引状況を報告する必要がある。表3に代行取引先に対する報告データを示す。

表3 代行先への報告データ

報告内容	サイクル	報告媒体	
		オンライン	FAX
売上	日次	○	
売上返品	日次	○	
入荷	日次	○	
商品返品	日次	○	
在庫振替	日次	○	
棚卸し調整	棚卸し調整ごと	○	
在庫数量	月末又は、要求があるごと	○	○

2) 代行請求

代行取引先に対し、月末に代行フィーを請求している。

- 請求額＝代行フィー－センタ補償額

① 代行フィーは次の2項目の合計である。

- 売上高 × 率
- 得意先理由（店舗棚替え等）による売上返品高 × 率

② センタ補償額は次の3項目の合計である。

- センタ理由（輸送中破損等）による売上返品高
- 在庫振替（鮮度期限オーバー等）による在庫減耗数 × 卸単価
- 棚卸し調整による在庫減耗数 × 卸単価

6. IND・IE適用作業に対する評価

当システムは、食品卸売業をモデルとしたソリューションパッケージであるIND・IEをカスタマイズすることで完成された。今回カスタマイズした主な機能は

- ① 在庫管理、発注を支援する代行取引機能
- ② 日配品、加食・菓子特売品を対象としたTC機能

である。

要求定義終了後、センタ開設まで4か月という厳しい条件でのシステム構築であったが、初期トラブルによる混乱は発生したものの、予定通り稼働を開始することができた。短期開発が実現できた主な要因は以下の通りである。

- ① 開発期間を意識した機能の絞り込みを行い業務稼働に最低限必要な範囲とした。

開発から除外した機能の主な例として、ハンディ端末による入荷検品機能、実績/統計資料作成が挙げられる。

② IND・IEが持つソフトウェア構造の柔軟性

- ・各種機能がパラメタに従って実行されるようになってきているため、プログラム開発が不要となる（商品保管エリアごとにピッキング方法を設定できること等）。
- ・納品単価算出方法や商品保管ロケーション決定方法のようにカスタマイズが想定される機能は、個々のプログラムでの記述ではなく共通ロジックとして一か所に記述してある。
- ・統合CASEツールであるLINCをベースに作られているため、項目の属性を変更する際でもシステム内の整合性を保つのが容易である。また、開発ドキュメントがLINCから出力されるので常に最新の情報に基づき開発を進めることができる。

③ IND・IEの各プログラムが徹底した標準化のもとに作成されていたため、既存プログラムを変更する際、既存ロジックを解析する時間が少ない。

④ IND・IEを熟知した要員がカスタマイズの各工程を担当した。

7. 今後の課題

センタの今後の改善テーマは以下のとおりである。

1) 物流作業精度の向上

入出荷を中心とした「作業ミスゼロ」実現のため、検品システムの導入を行う。

2) 物流作業効率化

現在、庫内作業は物流会社に委託しているが、その作業効率（生産性）を数値化してとらえ、適正值への改善を行う。

3) 自動発注（補充）システム構築

納品リードタイムゼロのセンタ特性を生かし、単品売上情報をベースにしたノンオーダーシステムの実現を目指す。

8. おわりに

センタが、日本型ECRの考えに基づいて設立され、稼働後1年が経過した。すでに述べたように、改善すべき課題も多々あるが、実効あるECRを目指すため、また、今後の物流環境の変化に対応していくために、業務およびシステムの改善を継続的に行っていく予定である。

最後に本稿の作成にあたり、本システムの開発に対しご協力いただいた株式会社小網および関連会社の方々に、感謝の意を表する。

- 参考文献 [1] “TDB REPORT戦略流通”, VOL.11, 帝国データバンク.
[2] “IND・IE/LOGシステム解説書”, 日本ユニシス.

執筆者紹介 平野 静夫 (Shizuo Hirano)

昭和54年横浜国立大学工学部金属工学科卒業。同年日本ユニシス(株)に入社。流通関連のソフトウェアパッケージ開発、適用に従事。現在、I&Cシステム本部システム四部に所属。



コンピュータの電源高調波問題と対策

Harmonics Problems with Computer Power Sources and Actions Taken

蛭本邦夫

要約 コンピュータなど、直流を電源として動作する半導体素子を組み込んだ電子機器が世の中に普及してきた。交流100V/200Vの商用電源を、半導体素子が必要とする低電圧の直流に変換する電源部が、高調波を発生する。「電源高調波」である。高調波は電子機器から配電線を逆に伝播して問題を起こす。すでに弊社でもトラブルを経験した。新聞の1面に載る事故もあった。事故を防止するために機器の発生する高調波の大きさを制限する規制も始まった。コンピュータを販売、サービスする我々にとっても避けて通れない問題になってきた。筆者等がサポートした事例をもとに、高調波の発生する仕組み、経験した問題、測定方法、対策、規制などについて具体例を交えながら述べる。

Abstract Increasingly prevailing is electronic equipment with semiconductor devices loaded, like personal computers, which operates on a low-voltage DC power source. Harmonics; that is, "power line harmonics" are induced by power supplies that convert commercial 100/200V AC into low-voltage DC required for semiconductor devices. Harmonics cause problems by flowing reversely from electronic equipment through its power lines. The author and others at Nihon Unisys have already experienced some harmonics-caused machine trouble. There was an accident that went into headlines. To prevent such hazards, the guidelines have also been drawn up to restrict the magnitude of equipment-inducing harmonics. All these, combined, have developed into a problem that cannot be bypassed by us at Nihon Unisys, which specializes in marketing computers, and offering related services and maintenance. Based on several instances where support was rendered by Nihon Unisys Technical Support Operations and with some specific cases cited, this paper describes how harmonics are generated, what problems really happened, how to measure harmonics, and how to react as well as the related regulations.

1. はじめに

電源高調波問題が話題に上ようになってきた。電源の異常はコンピュータに不可解な誤動作を引き起こす可能性があるだけに厄介である。実はコンピュータの電源は高調波の発生源であり、保守担当者としては高調波と認識はしなくても長年かかわってきた。パーソナルコンピュータ、インバータ利用機器など高調波を発生する機器が一般家庭に広く普及してきた。高調波問題をコンピュータルーム内に限定された問題からどこでも起こり得る身近な問題に変身させることになった。送電系統のリアクトルやコンデンサに被害が発生する。コンピュータを保守する立場としては高調波に対する正しい理解が必要である。電源高調波に対する問い合わせやサポート依頼を経験したので、この機会にサポートした内容を中心にまとめた。

2. 高調波発生メカニズム

2.1 高調波とは

一般に交流、特に商用電源の50/60Hzについては、正弦波であるものと認識している。波形が正弦波からずれてくる、すなわちひずむ。これを周波数の観点からみると元の波形（基本波）にその整数倍の周波数の波形が重なったものとして捉えることができる。この整数倍の周波数の波形を総称して高調波と言う。また、電力の利用の面からは後述するように力率が悪化する。波形がひずむ、高調波が発生（増える）する、力率が悪くなる、同じ現象でもとらえ方により表現は異なる。電力会社の配電系統に使われているコンデンサやリアクトルに高調波電流が流れ、発熱・焼損などの事故を起こすことから、最近電源高調波問題としてクローズアップされてきた。

2.2 コンピュータが高調波を発生する仕組み

コンピュータが高調波を発生する原因の大部分は、電源装置が配電系統から見て非線形の負荷であることに起因する。コンピュータでは、半導体素子の電源電圧である直流低電圧（12V、5V、2Vなど）を作るために、スイッチング電源を使用している。スイッチング電源では図1のように入力を一旦整流平滑して直流にし、それを高周波スイッチングして所定の直流低電圧を得ている。ここに使われている整流器と平滑コンデンサの組み合わせである整流平滑回路（コンデンサ入力型整流平滑回路）が、非線形負荷であり高調波が発生する要因になる。コンデンサCには整流ダイオードDを通して電荷が蓄えられている。したがって、入力電圧がCの端子電圧を越えないと電流は流れ込まず、電圧のピークでのみ流れ込むひずんだ電流波形となる。コンピュータに限らず直流を使う機器では、交流の商用電源を整流して用いるので整流平滑回路で高調波を発生する。

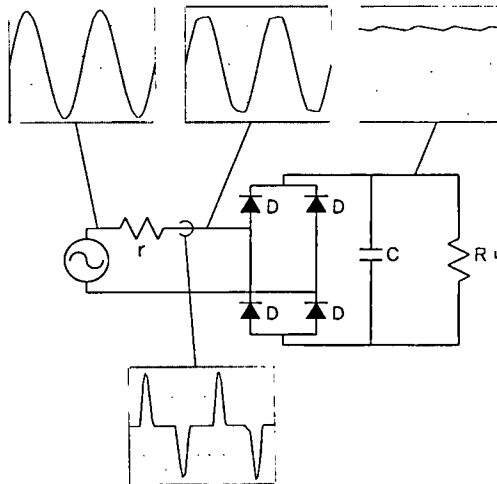


図1 コンデンサ入力型整流平滑回路と電圧（上三つ）、電流波形（下）

2.3 力率の悪化

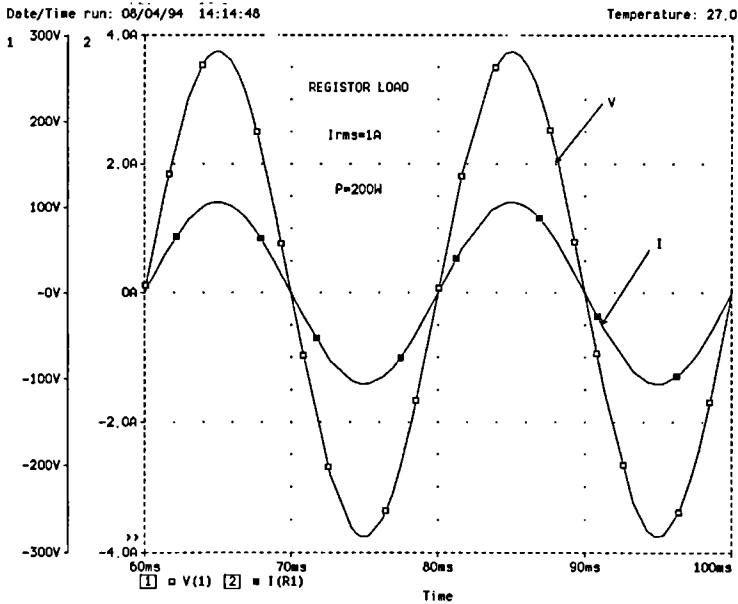
波形ひずみは力率にも関係する。

力率=消費電力 (W) /皮相電力 (VA)
 =消費電力/(電圧の実効値×電流の実効値)
 である。

従来力率は電圧と電流の位相差すなわち $\cos \phi$ としてとらえられてきた。波形がひずむと実効値が大きくなるために位相がずれていなくても力率が悪くなる。2者を区別するために弊社では力率として

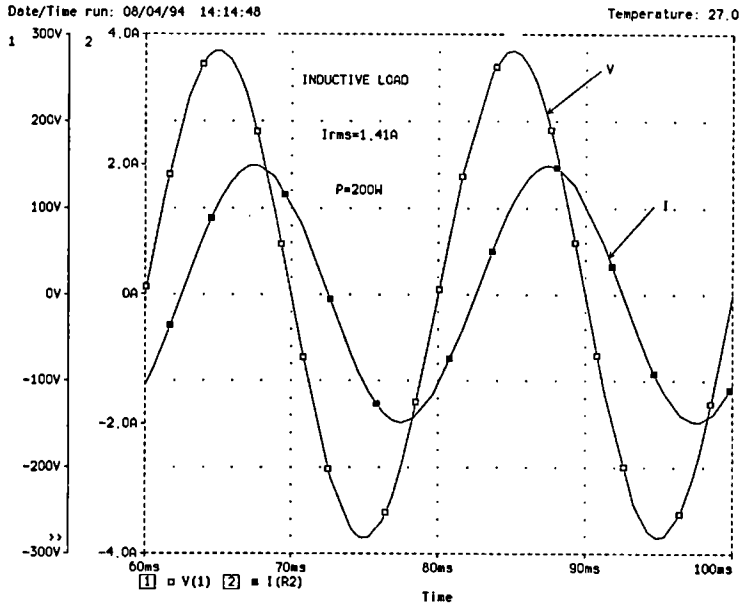
$$\text{力率} = \text{ひずみ係数(Distortion Factor)} \times \text{位相係数(Displacement Factor)}$$

と捉えることを提案している¹¹⁾。実際にはひずみ係数と位相係数を区別して測定することは困難だが、力率の悪化を二つの要因に分けて考えることは現象を把握する上で役に立つ。位相がずれることによる力率の悪化に対してはその対策上位相の進みか遅れか、言い換えれば符号がマイナスかプラスかが重要な意味を持つ。対策として進みなら遅れ、遅れなら進みの要素を回路に入れなければならないからである。一方ひずみが原因である場合には、測定器の性格上、符号はプラスになったりマイナスになったりすることがある。符号の正負は意味を持たない。歪みを減らさなければ力率は改善されないからである。力率を言う場合、相手によっては力率=位相のずれと取ってしまうこともあるので、ひずみによる力率であることを付け加えておくと良い。以下具体例を図2, 3, 4に示す。比較のため消費電力 (200W), 電圧 (200Vrms) は一定



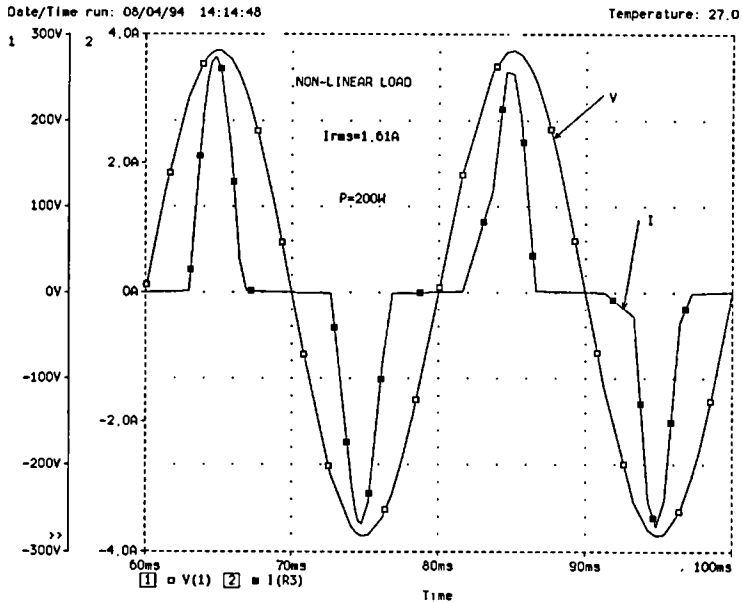
電圧と電流の位相が一致している。位相差は0, 力率は1である。

図2 抵抗負荷の電圧電流波形



電流の位相が45度遅れている例である。力率は+0.71である。

図3 誘導負荷の電圧電流波形



コンデンサ入力型整流平滑回路の例。
位相はずれていないが電流波形がひずんでいる。力率は0.62である。

図4 非線形負荷の電圧電流波形

にしてある。

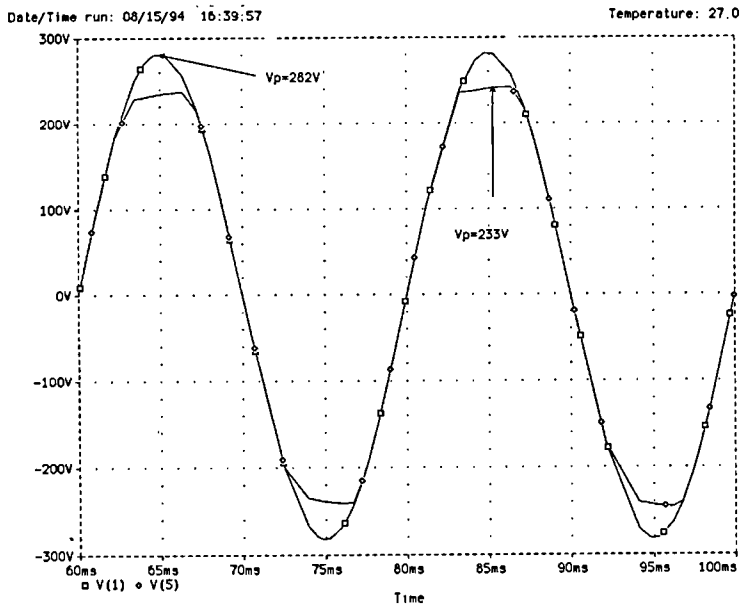
3. フィールドサービスにおいて経験した問題

すでに述べたように、コンピュータは高調波の発生源である。高調波発生に対する規制も始まった。被害では高調波による配電系統の事故が大きな問題である。高調波の発生源であるコンピュータが高調波の被害を受けることはほとんどない。最近フロア一杯にパーソナルコンピュータ、ワークステーションなどが設置されることも珍しくなくなった。しかも電源が一般の商用電源いわゆる壁コンセントの場合が多い。このような環境で高調波問題が発生しやすい。

3.1 ピーク電圧の低下—入力電圧低下を検出して機器が動作しなくなる

新たに機器を増設したところ、既設の機器の中に入力電圧低下を検出し動作しなくなるものが出た。テスター、DMM（デジタルマルチメータ）などで電圧を測ってみてもスペック内に収まっている。

コンデンサ入力型整流平滑回路では、電圧のピークのみ利用する。この客先では多数の機器の設置により、図5の例のように電圧のピークが削り取られた波形になっていた。233Vのピークを持つ正弦波は165Vと計算される。コンデンサ入力型整流平滑回路にとっては82%に低下したことになる。整流平滑後の電圧が十分上昇せず入力電圧低下が検出されたものである。しかしテスターあるいはDMMでは平均値、実効値などが用いられておりピーク値の低下を的確に検出できない。そこで電圧のピーク値を測定するアダプタを製作した。これを用いて電圧を測定することにより、電圧低



正常な波形と共に表示してある。

図5 ひずんだ電圧波形

下に敏感な機器を電圧の高い電源系統に移したり、負荷を振分けて極端に電圧の低い系統ができないように対策をした。

3.2 高調波電流の流入一力率改善用コンデンサに異常電流が流れた

電源用のモータオルタネータが使われている機器がある。力率改善のために進相用コンデンサが使われている。このコンデンサに、他の機器で発生した高調波が異常電流として流れ込み、ブレーカがトリップした。

コンデンサに流れる電流を測定したところ正常値の倍近くあった。電流波形を観測したところ図6のように高周波が重畳して、50Hzの波形がほとんど観測できないほどである。

高周波の原因をつきとめるために電流波形をFFTアナライザ*で分析した結果、図7のように50Hzの整数倍の高調波電流が流れており、これが電流値を増加させていることが分かった。同じ電源に接続されているスタティックインバータの整流回路が発生する高調波が、進相コンデンサ回路に回り込むことが原因であることが分かった。11, 13, 23, 25, 35, 37次の高調波が目立っている。スタティックインバータのメーカーの説明では、入力3相交流を12相整流しているので $12n \pm 1$ の高調波が大きく出る。そこでとりあえず進相コンデンサ回路を切り放すことで対処した。

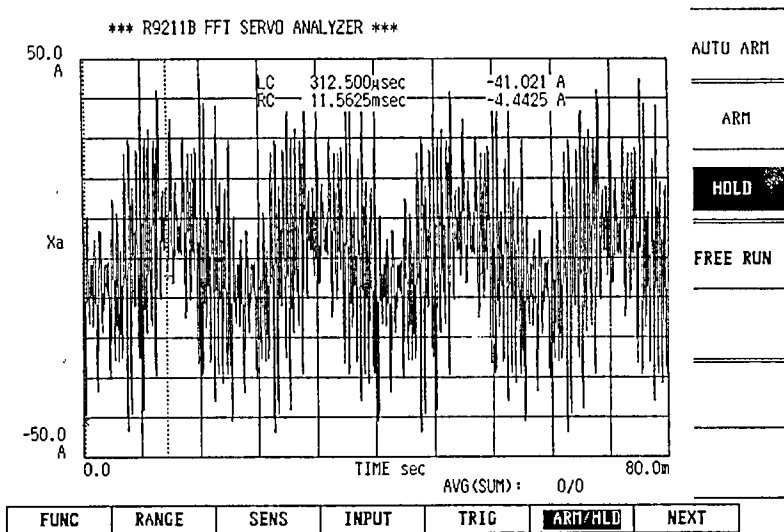
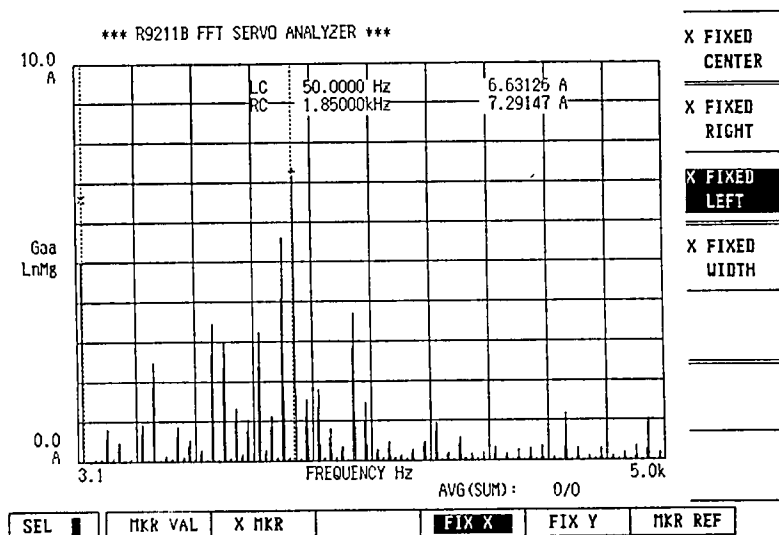


図6 進相コンデンサに流れる電流の波形

3.3 皮相電力の増大一力率の悪化

消費電力 (W) に比べて皮相電力 (V · A) が大きくなり電源設備に余分な投資が必要となる。場合によってはピーク電流も考慮しなければならない。ここに100Vで消費電力200Wの機器がある。実効電流は3.8A、ピーク電流は13Aである。力率は0.53である。UPS (Uninterruptible Power Supply, 無停電電器装置)を設置する場合を

* FFT : Fast Fourier Transform (高速フーリエ変換) の略, 周波数解析はFFTで行われるのでFFTアナライザが一般名となっている。



基本波6.6Aに対して37次高調波が7.3Aある。

図7 進相コンデンサに流れる電流の周波数分析

考えよう。力率が1なら200VAの容量があれば済むのに、力率が悪いために530VAの容量が必要である。しかし13Aのピーク電流に対応するためには $13 \div \sqrt{2} = 9.2$ (Arms) すなわち920VAの容量が必要となる(スイッチング電源用として実効値の3倍程度までピーク電流が流せるUPSもある)。流れる実効電流に合わせて電源の配線も太くしなければならない。200Wに対して920VAを用意しなければならないとは困ったものである。

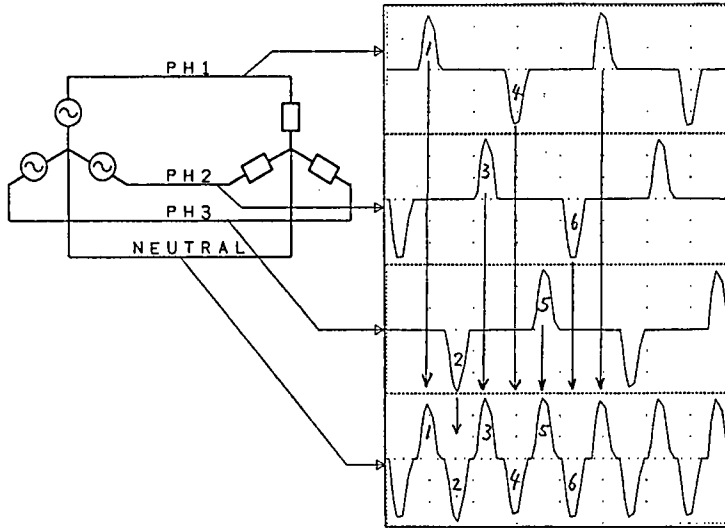
3.4 3相4線のニュートラル線の過大電流

常識的にニュートラル線は負荷のアンバランス分の電流が流れるだけなので、相線に比べて流れる電流は非常に少ないと思われる。しかし、3相4線式の電源線のニュートラルには、相線の $\sqrt{3}$ 倍の電流が流れる(図8)。ニュートラル線の異常発熱やニュートラル電位の変動をもたらす¹¹⁾。スイッチング電源を搭載した初期の機器でニュートラル線を変更するFCN*が出されたこともあった。

3.5 配電システムの事故に対する弊社の対応等の客先からの質問

1994年3月24日、名古屋においてビルの受電設備のリアクトルの爆発があった。それと同時に近隣の約10か所で、リアクトルの焼損や異常震動など高調波障害特有の事故が発生した。この件に関しユーザから、弊社のコンピュータの高調波問題に対する取組について質問を受けたものである。この客先については大型コンピュータのユーザであり、CVCFを使っているので、コンピュータの発生する高調波はCVCFで止まる。配電システムへはCVCFの入力側の高調波が問題となる。弊社のコンピュータにはすでに対策済みの製品がある。弊社のスペックでは電圧のひずみ率は5%以下と規

* FCN: Field Change Notice, ハードウェアの改善・改良のための変更指示。



ニュートラル線の電流は相殺されて0になるべきところが図のように重なり合う部分がないので実効値として $\sqrt{3}$ 倍の電流が流れる。

図8 3相4線における非線形負荷

定されている。配電系統に直接影響があるのはCVCFを介さないで商用電源を直接使っているパソコンを始めとする電気製品であることなどを説明した。

4. 測定方法

目的によって測定する（注目する）ものが異なる。たとえば、平均値、実効値、ピーク値などである。正弦波と違って互いに換算によって得ることができないので、目的に応じて測定することになる。ここでは高調波発生の原因としてコンデンサ入力型整流平滑回路を仮定している。最近では高調波に関する全ての測定が1台でできる便利な測定器（たとえばフルーク40高調波メータ）も市販され始めている。筆者等が使った測定器についても参考のため載せておく。高調波測定用に揃えたものではないので必ずしも最適とは言えない。

4.1 プリスケーラ、プローブ

安全のため絶縁することと、測定器に入力できる電圧・電流に低減するために使用する。電圧にはポテンシャルトランスフォーマ、電流にはクランププローブが便利である。

使用したものは安価な50/60Hzの定格の電源測定用だが、10KHz程度まで帯域が伸びている。高調波が原因であることの見極め、高調波発生元の特定などトラブルシューティングの用途には十分使えることが分かった。ポテンシャルトランスフォーマ：日置9303、クランププローブ：日置9008、フルークY8101、テクトロA6302などがある。図9に周波数特性の例を示す。

4.2 実効値

テスター、DMM（デジタルマルチメータ）などで測定する。プレーカの電流値、

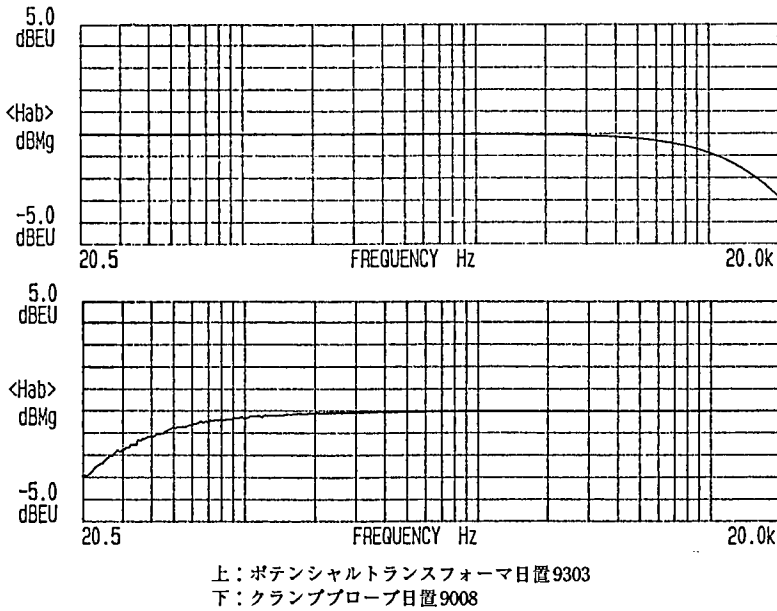


図9 プローブの周波数特性

電線の許容電流など実効値で作用するので実効値の測定が必要になる。真の実効値 (True rms) を測定できるものが必要である。高調波の周波数範囲まで周波数帯域が伸びているものでなければならない。フルーク 8060A, フルーク 87 などがある。

4.3 ピーク値

非線形負荷 (コンデンサ入力型整流平滑回路負荷) の特徴として電圧波形のピークが削られて低くなる (図5参照)。これは非線形負荷が交流のピークのみを利用していることによるが、裏を返せばピークが低くなることは、非線形負荷にとって入力電圧が低下したことになる。場合によっては電圧が低くなりすぎて動作しなくなることがある。通常の電圧計で測ってもピークの低下は分からない。ピーク電圧の測定ができる電圧計に良いものが見つからないのでRPS1085*で「電源ピーク電圧測定アダプタ」を作った。

4.4 波形

オシロスコープ、もう少し低速の波形記録装置などを使う。波形を観測することは原因究明のための大変有効な手段である。原因が分からない時はとりあえず波形を観測してみることを勧める。DMMなどのように何処にでも持ち運んで使うわけにはいかない。機動性の面で劣る。FFT機能が付いた機種もあるが後述するように分解能が十分ではない。HP54502A, 日置8830などがある。

4.5 電力、力率

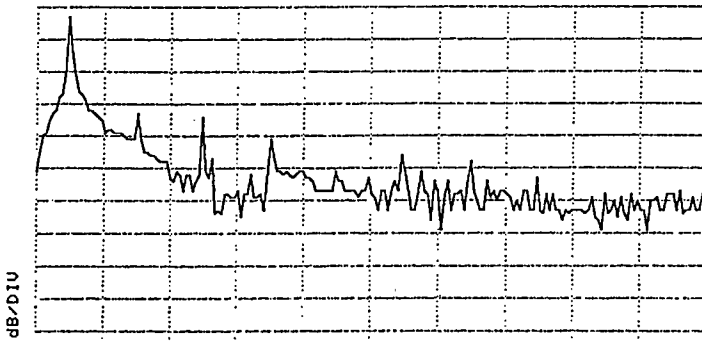
電力計で測定する。注意しなければならないのは力率の符号である。数値は1以下 (0.5~0.9) で力率が悪いことを示しているが符号は+になったり-になったりする。

* RPS: Request for Product Support, プロジェクト単位に筆者等の所属する部署で付けられる番号。仕事の区切りの単位。

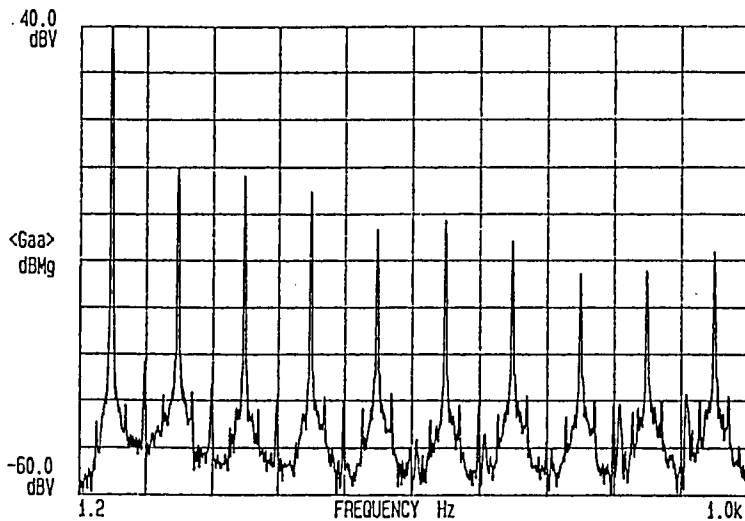
符号に意味があるのは位相のずれで力率が悪化する場合だけである。波形のひずみによる力率の悪化に対しては符号は意味を持たない。ただ測定器の性格上、符号判定器が符号を付けるので上述の結果になる。

力率 = ひずみ係数(Distortion Factor) × 位相係数(Displacement Factor)
 を思い出して頂きたい。位相のずれによる力率の悪化の場合の対策は、電流の遅れ(+)には進相コンデンサ、電流の進み(-)にはリアクトルを入れて位相を補正すれば良いが、波形のひずみによる力率の悪化に対してはまったく無意味である。

力率を言う場合、ひずみによるものか位相によるものかをきちんと告げないと、相

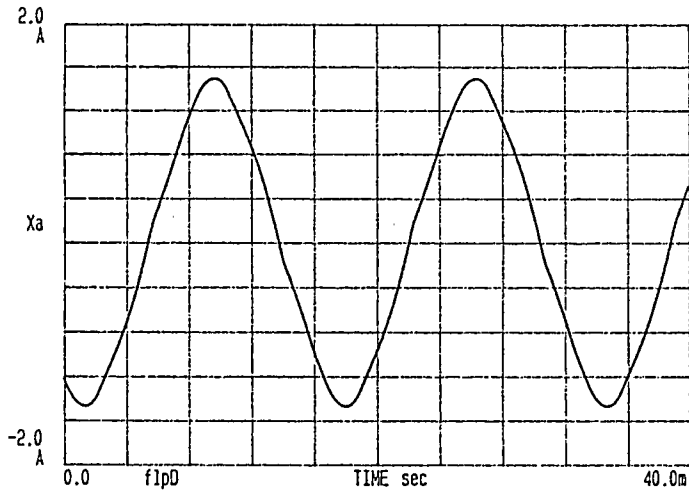


(a) 8ビット分解能でのFFT結果



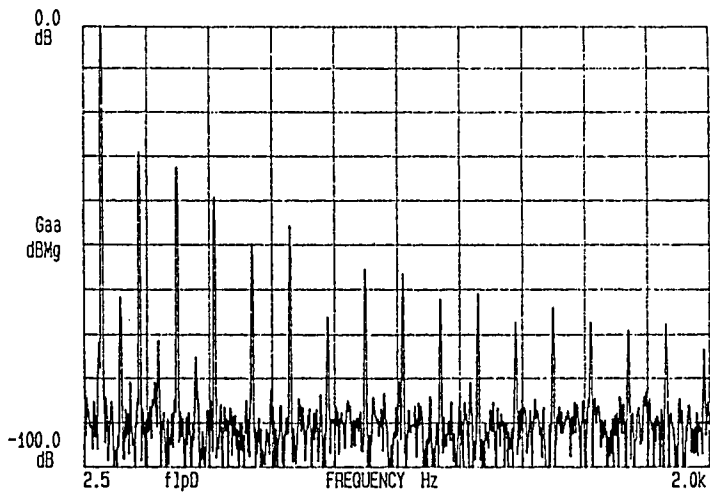
(b) 16ビット分解能でのFFT結果

図10 ビット数の違いによる分解能の差 (縦軸は一目盛り10デシベル)。



見逃してしまいそうなひずみ

(a) 交流波形



全高調波ひずみ率は4.4%。縦軸は一目盛り10デシベルである。

(b) aの交流波形をFFTアナライザで分析したもの

波形を見ただけでは見逃してしまいそうなひずみでも、FFTアナライザで分析すれば高調波があることがはっきりわかる。

図11 ひずみ率の例

手は誤解を受けることになる。筆者等のサービスするコンピュータの場合、力率の悪化はひずみによるものと考えて差し支えない。また測定器の特性として、ひずみ波形にどれだけ対応できるか注意しなければならない。日置3191などがある。

4.6 高調波解析

FFTアナライザで測定する。各次の高調波の大きさ、周波数、ひずみ率などを知る

ことができる。デジタイジングオシロスコープや波形記録装置などにも、FFT機能を持ったものがあるが、分解能が低い場合が多い。たとえば8ビットでデジタイズすると1ビットの分解能は $1/256=0.4\%$ となる。ただし波形は余裕を見てフルスケールの半分くらい振らせるのがせいぜいのため、分解能は1%（-40デシベル）止まりとなる。図10の例のように、高調波の存在が誤差の中に隠されてしまって使い物にならない。アドバンテストR9211Bなどがある。

4.7 全高調波ひずみ率

全高調波エネルギーの基本波のエネルギーに対する割合として定義される。波形のひずみ具合を定量的に表す数値である。ひずみ率測定専用のひずみ率計もあるが、FFTアナライザにもひずみ率測定機能があるのでそれを使えばよい。ひずみ率が数パーセント以下になると、図11のようにオシロスコープで波形を見ても、ひずんでいようには見えない。FFTアナライザで高調波分析をしたり、ひずみ率を測定すれば定量的に把握することができる。

5. 対策

すでに出来上がった製品に対して対策を施すことは非常に困難である。コンデンサ入力型整流回路に対しては、アクティブ平滑フィルタが有力候補である。しかし整流電圧を高くせざるを得ないので、既存のスイッチング電源に後から付加できるものではない。UPS、AVR(Automatic Voltage Regulator, 自動電圧調整器)などが使われている場合には、高調波対策済みのものを使うことによって高調波が伝播していくことを防止することができる。新たに設置するときには対策済みのものを選定するのが良いであろう。高調波に対する規制が始まる以前に発売された機器については、高調波抑制の対策は取られていない。高調波を抑制する安価で簡単な付加装置はない。対策は被害を食い止める消極的な対策が主流となる。大型機のようにCVCFなどコンピュータ用の専用電源設備を設置しているところでは、コンピュータの発生する高調波はCVCFで止められるので配電系統への影響は考えなくてよい。

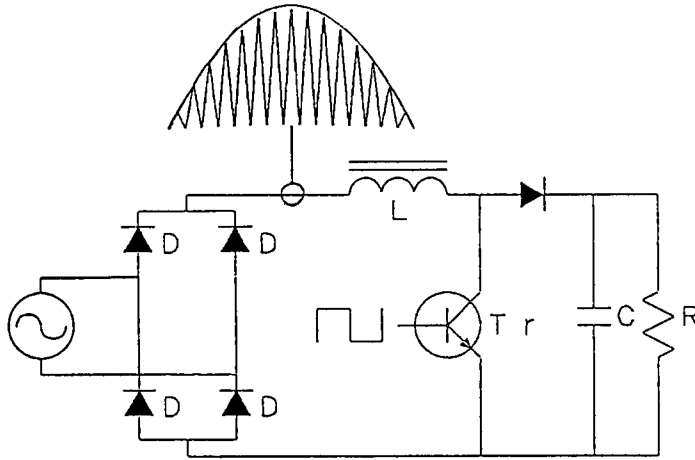
5.1 発生を押える対策

1) アクティブ平滑フィルタ

図12のように、整流回路と平滑回路の間に、LとTrによるステップアップコンバータを設け、スイッチング電流の平均が正弦波に近くなるように制御する。力率をほぼ1にすることができる。弊社においては、高調波発生の抑制の目的よりは、力率を改善して電源の負荷を軽減する（同じ電源容量ならより多くの負荷をつなぐことができる）目的で、SYSTEM11、2200/100-500などですでに採用されている。バルク電源の出力電圧が360～380Vと高いのが特徴である。

2) チョーク入力型平滑回路

整流回路と平滑回路の間に低周波チョークを挿入することにより、高調波の発生はコンデンサ入力型に比較して少なくなるが、形状も大きく重量も重くなるのであまり用いられない。そもそもスイッチング電源が出てきたこと自体、大きくて重く高価な鉄と銅の塊であるトランスを排除するのが目的であった。



高速スイッチングすることにより電流の平均値を正弦波に近付ける。

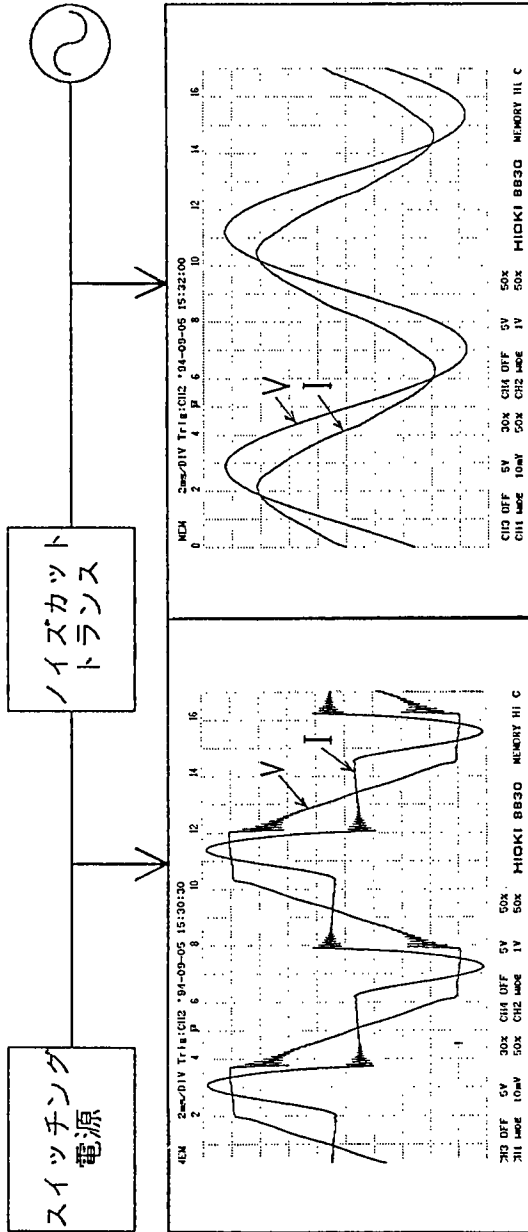
図12 アクティブ平滑フィルタ

5.2 伝播を防ぐ対策

常時インバータ方式のUPSでは内部で交流→直流→交流の変換を行っているので、負荷側の高調波が上流（1次側）に伝わることはない。UPS自身の1次側の整流平滑回路に高調波対策が取られていれば、UPSより上流では高調波の発生はない。伝播を食い止めたことになる。ノイズカット・トランス（1次2次間の絶縁だけでなく高周波ノイズも阻止する）の中には高調波を阻止する能力を持つ製品がある。高調波のフィルタとして使える。図13はノイズカット・トランス入出力の電圧電流波形の例である。2次側（負荷側）の電流はひずんでいるが、1次側は正弦波に近いものに整形されている。UPS、AVR、ノイズカット・トランスなどの装置を新たに設置するときには、高調波対策品を選定するのがよい。高調波防止だけの目的のために増設するには、コストや設置スペースの点で難点がある。

5.3 配電系統を工夫して被害をおさえる

高調波として捉えるより波形ひずみと考えた方がわかりやすい。電圧波形のピークが低くなると、ピークを利用するコンデンサ入力型平滑回路にとっては、入力電圧が低下したことになり動作しなくなる。配電系統が複数あれば負荷をうまく分配することによって被害を避けることができる。簡単でコストのかからない方法として使われている。また、増設に先だって測定しておくことで安心できる。例えば日本国内での機器の動作範囲は100%±10%である。ピーク値が127V以下であれば障害が起こってもおかしくない。なぜならピーク値127Vは正弦波に換算すれば90V、すなわちコンデンサ入力型平滑回路にとっては、100Vの-10%低下と同等だからである。コンピュータの電源はさらに低電圧で動作する実力があるが、電源電圧低下検出回路を持っている機器が動作を停止したりする。ピーク電圧が測定できる装置があれば作業がやりやすい。オシロスコープなどで波形を観測してもよいが多数の場所で測定するには機



電圧：120Vrms ひずみ率6.3%
 電流：2.16Arms ひずみ率88%
 消費電力：200W
 力率：0.75

(a) 2次側電圧電流波形

電圧：120Vrms ひずみ率0.04%
 電流：2.10Arms ひずみ率4.4%
 消費電力：240W
 力率：-0.94

(b) 1次側電圧電流波形

周波数は60Hz。比較のためスケールは同じにしてある。2次側の高調波が1次側では激減している。高調波を抑制している様子が分かる。電流のピークは2次側で5Aあったものが1次側では3Aに減少している。ノイズカット・トランスの損失が40Wある。

図13 ノイズカット・トランスの1次側・2次側、電圧・電流波形

動性がない。

5.4 ラインフィルタの効果

電源ラインを伝わってくるノイズを防止するために、ラインフィルタが使われている。このラインフィルタに期待できないのか。答えは否である。ラインフィルタの目的は伝導ノイズの規制に対処するためのものであり日本ではVCCI（情報処理装置等電波障害自主規制協議会）で、150KHz以上の周波数について規制されている。ラインフィルタがきき始めるのは低域の伸びているもので、10KHzあたりからである。50/60Hzの高調波は10KHz以下のものが大部分であり、ラインフィルタは全く役に立たない。

6. 規 制

平成8年1月1日から事務機器・電子計算機など一部の機器で規制が開始される。これから発売される新製品に対して規制がかけられる。電源高調波が次第に減少していくことが期待される。高調波が原因の事故やトラブルも無くなっていくであろう。ちょうど執筆時期に一部の機器で高調波規制の適用が始まった。ガイドラインなどが次々と発行されたり、見直されたりしており、規制に関する記述には最新の情報を反映できていない部分もある。

6.1 ユニシスの基準

(4000-0010, 4000-0028)¹²⁾に電圧波形についての記述がある。機器が発生する（機器が電圧波形をひずませる）全高調波ひずみ率、機器を接続する電源側の全高調波ひずみ率がいずれも5%以下となっている。電圧波形のひずみは内部インピーダンスが0でない電源に対してひずんだ電流が流れるために生じる。電源の条件が測定結果に影響する。測定に際しては電源の条件も規定されている。現実のコンピュータの設置環境においては特に対策をしなくても電圧波形のひずみ率はほとんど5%以下である。高調波メータなどで測定できる。

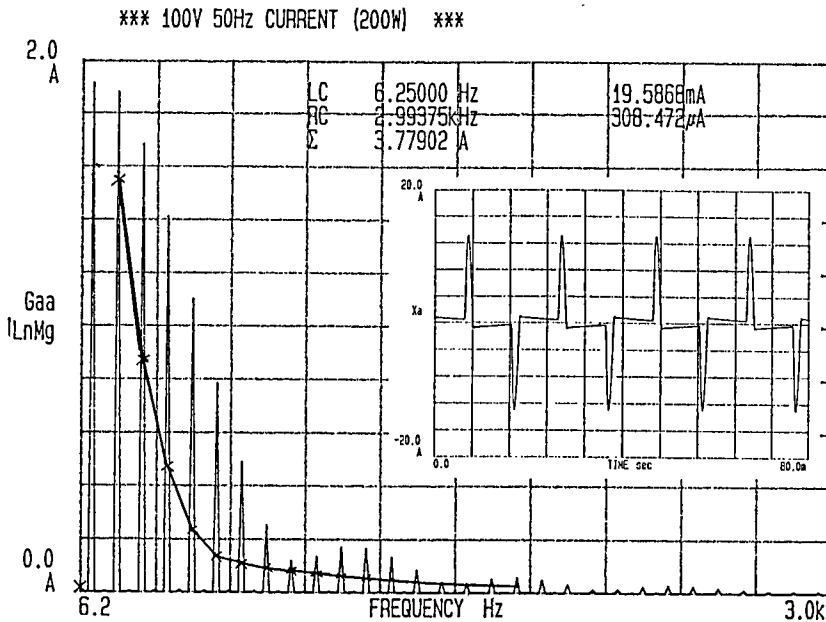
6.2 高調波問題専門委員会

電力会社では高調波による電力系統の障害に対し「高調波問題専門委員会」を設けて検討を重ねている。家庭用の電子機器は半導体化が進み、スイッチング電源が使われることにより高調波の発生が多くなった。加えてパソコンなど新たな電気製品の普及、エアコン、ヒーター、照明器具などのインバータ化により、高調波は増加の一途を辿っている。高調波が原因と見られる現象（異音の発生、コンデンサ/リアクトルの発熱・焼損、ヒューズの溶断など）が増加している。同委員会の報告書¹³⁾によれば、高調波による障害を防止するために配電系における総合電圧ひずみ率は5%以下、特高系における総合電圧ひずみ率は3%以下にすることを目標レベルとしている。高調波抑制対策ガイドライン制定の根拠として引用されている。94年3月24日、名古屋で起こったビルの受電設備のリアクトル焼損事故は高調波が原因の事故だとされ、新聞でも報道された。

6.3 IEC-1000

IEC-1000-3-2¹⁴⁾として国際規格がまとまった。新しく製造する機器は高調波の発生を厳しく制限することによって、高調波ひずみ率を現状維持しようとするものであ

る。欧州では1996年1月1日から規制が始まる。日本でも1996年または1997年以降に出荷する新製品から適用される予定になっている¹⁵⁾。負荷の種類によってクラスAからクラスDまでに分類される。電流の高調波を次数ごとに制限する。まさに高調波問題に対する規制である。スイッチング電源は何らかの対策をしないと規制をクリアできない。アクティブ平滑フィルタを内蔵した力率改善型スイッチング電源も製品が出回り始めている。これは力率が0.99と高調波の発生が極めて少ない。規制の例として消費電力200Wで動作しているスイッチング電源の高調波電流をIEC規格の制限値と共に図14に示す。



小さい枠内は電流波形、折れ線はIECの制限値。

図14 消費電力200Wのスイッチング電源の高調波電流

6.4 ガイドライン

日本においても高調波問題専門委員会の報告を受けてガイドラインが発行された。コンピュータの発生する高調波の規制に関連するものとして以下のものがある。海外ですでに同等の規制について認定/認証を取得している場合には、日本で再測定を省略できるよう要望が出されている。日本でも一部の機器については適用が始まった。弊社開発機器も製品開発時に配慮が必要である。

「家電・汎用品高調波抑制対策ガイドライン」¹⁶⁾の内容はIEC-1000に準拠している。ガイドライン適合品、ガイドライン準用品にはそれぞれその旨を製品の取扱説明書等に表示することになっている。

文献「高圧または特別高圧で受電する需要家の高調波抑制対策ガイドラインについて」¹⁶⁾は、高圧または特別高圧から受電する需要家から、商用電力系統に流出する高調波電流についてのガイドラインである。需要家が高調波発生機器を新設・増設・更

新する場合に適用される。負荷になっている個々の機器の高調波電流を集計して、高調波流出量を算出する。コンピュータを新設・増設する場合に、電源設備担当者から高調波流出量について質問を受ける場合がある。

「家電・汎用品ガイドラインの実施計画書」⁷⁾では、上述の「家電・汎用品高調波抑制対策ガイドライン」対象機器のうち電子計算機および関連機器についての適用時期が決定された。定格入力電力値1 KW以下の機器については平成8年1月1日以降出荷される新製品から、1 KWを超える機器については平成9年1月1日以降出荷される新製品から適用される。

7. おわりに

筆者等がサポートした事例の中から電源高調波に関係のあるものをピックアップする形でまとめた。挿入してある波形は実測のほか、測定環境の実現が困難なものはPSPICE*によるシミュレーションで作成した。

製造販売後の製品に対する対策は不可能に近い。弊社の製品には力率の改善の面からすでに対策済みの製品もある。高調波は身近に存在しながら、FFTアナライザ、ひずみ率計など馴染のない測定機が必要なので接する機会が少ない。これからは規制の対象になることも手伝って、高調波メータなど手軽で便利な測定器が登場する。フィールドサービスにおいては測定器の設置、高調波問題に対する理解などが必要となってくる。本稿が電源高調波問題理解の一助になれば幸いである。

最後に貴重な資料の提供をいただいた佐藤担当部長にこの場を借りて謝意を表する。

-
- 参考文献 [1] William Richardson, Power & Environment Engineering Computer Power Distribution, UNISYS, July 1989.
 [2] UNISYS CORPORATE STANDARD - TEST SPECIFICATION
 4000-0010 POWER ENVIRONMENT
 4000-0028 POWER ENVIRONMENT TEST
 [3] 高調波問題検討報告書 昭和62年5月7日 高調波問題専門委員会
 [4] Draft IEC 1000: Electromagnetic Compatibility (EMC) 1993-12-24
 Part3: Limits
 Section 2: Limits for harmonic currents emissions (equipment input current $\leq 16A$ per phase)
 [5] 「家電・汎用品高調波抑制対策ガイドライン」平成6年9月資源エネルギー庁発行。
 [6] 「高圧又は特別高圧で受電す需要家の高調波抑制対策ガイドラインについて」平成6年10月資源エネルギー庁発行。
 [7] 「家電・汎用品ガイドラインの実施計画書」平成7年6月日本電子工業振興協会発行。

* PSPICE：回路シミュレータソフトウェアSPICEのパソコン版

執筆者紹介 蛭本 邦夫 (Kunio Hiramoto)

1946年生。1969年防衛大学校電気工学科卒業。同年日本ユニシス(株)入社。カスタマーサービス、フィールドサポート、サポートエンジニアリングに従事。現在CS推進本部ロジスティクス企画室に所属。



8003の高速・高品位レンダリング機能

Instantaneous Visual Acuity on the 8003 Graphics Subsystem

渡 辺 啓

要 約 8003グラフィック・サブシステムの技術的課題は、ポリゴンアンチ・エリアシングと透過描画をシングル・パスで実現し、さらにシェーディング、テクスチャ・マッピングと統合し高速なハードウェア機構とすることである。これは世界で最初の試みである。

課題を実現するためには、描画の重なり関係管理を含むアルゴリズムが必要である。基本アルゴリズムとしてAバッファ法（アンチ・エリアシングは面積法）が採用され、グロー補間とミップマップ法によるテクスチャ・マッピングが組み合わされた。速度要求を達成するために、アルゴリズムは4種のASIC（Application Specific Integrated Circuit）として実現された。そのためにVHDLの導入を中心とする最新の大規模ASIC作成技術が試みられた。

現在本製品はポリゴンアンチ・エリアシングとz方向描画順の制限を持たない透過描画をシングルパスで行える業界唯一の製品である。また、速度性能に関しても100～200万ベクトル/秒級機ではGPCPLBベンチマークで世界最高速値を示している。

アルゴリズムのASICによる実現は**InstaCUIITY**テクノロジーと名付けられ、日米英で特許出願が済まされた。

Abstract The technical objective of the 8003 Graphic Subsystem was to implement in high-speed hardware a single-pass rendering of polygon anti-aliasing, random Z alpha blending, shading and texture mapping. To achieve this objective, the A buffer method was used to maintain the depth of each pixel fragment for proper blending of anti-aliasing and alpha. Combined with it were Gouraud interpolation and Mipmap-based high-speed texture mapping. In order to achieve the performance required, this algorithm was implemented into four new ASICs, using the latest giant-scale ASICs and VHDL technology.

The 8003 Graphic Subsystem is the first and only product in the world market which is equipped with this hardware accelerator to render both polygon anti-aliasing and random Z alpha blending in a single pass. It is also the fastest, in items of GPC PLB benchmarks, of 1 to 2 million vectors/sec graphics systems.

The ASIC implementation of this algorithm has been named **InstaCUIITY** (Instantaneous Visual Acuity) and is patented in the USA, Britain and Japan.

1. はじめに

日本ユニシスは、AGS3000グラフィック・ターミナルの後継商品として、8003グラフィック・サブシステム（開発コード名：Camellia）を開発し提供を開始した。本製品は、UNIX*ワークステーションのグラフィックス・アクセラレータとして近年のマルチ・ウィンドウ環境、フルカラーサーフェス・レンダリングの要求に応えるも

* UNIXは、X/Openカンパニーリミテッドがライセンスしている米国ならびに他の国における登録商標である。

のであるが、あわせてAGS5000*で実現した水準の高速かつ高品位なレンダリング能力が求められた。

本製品の開発作業は1992年より3年にわたりMegatek社(米国San Diego)で行われた。製品化において、ポリゴンアンチ・エリアシングと透過描画をシングル・パスで実現し、さらにシェーディング、テクスチャ・マッピングも含めて4者を統合し高速なハードウェア機構とすることが技術的課題とされた。とくに前2者については世界で最初の試みである。採用されたアルゴリズムのASICによる実現は、**InstaCUIITY**(Instantaneous visual acuity)テクノロジーと名づけられ、日米英で特許出願が済まされた。

本稿では採用されたアルゴリズムとASICによる実装のあらましを紹介し、解決されたレンダリング上の問題点を明らかにする。

2. レンダリングの諸方式と問題点

まず最初に、主なサーフェース・レンダリング機能について、すでに提案されている諸方式とその問題点を述べる。

コンピュータ・グラフィックスの分野では、写実的で高品位な描画結果を得るための手法として、レイ・トレーシング、ラジオシティなどが知られている。しかし、これらの手法は演算量が膨大で、長時間かけて静止画を一枚一枚作り上げるものである。一方、CAD/CAMなどの会話処理におけるヒューマン・インタフェースとして位置づけられる本製品においては、描画のリアルタイム性が必須の条件となる。ローカル・ビューイングに代表されるリアルタイム・アニメーション処理において秒10コマ程度以上の描画速度を持たさなければならず、いかに高品位な結果が得られようと複雑で多量の実数演算が伴うような方式を導入することはできない。以下の議論はLSI化が可能な程度の簡便法に限定する。

2.1 アンチ・エリアシング

ラスター・スキャン型グラフィックスの大きな欠点であるジャギー(jaggie:階段状ゆがみ)は、代表的なエリアシング(aliasing)問題である。このようなエリアシングを取り除く処理をアンチ・エリアシング(口絵1, 2参照)と呼ぶが、今日では目の錯覚を利用し仮想的に表示分解能を上げる方法が主流である。基本原理を図1に示す。二つのピクセル(画素)が隣合うとき、観察者には単一ピクセルに見える。その中心は両ピクセルの明るい方にずれて見え、その明るさは両ピクセル輝度の合計の印

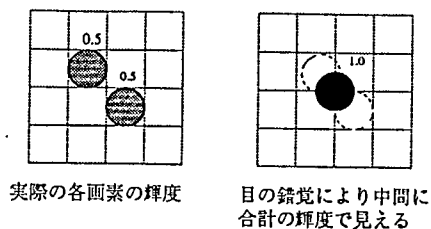


図1 視錯による仮想的表示分解能の向上

* 日本ユニシスは1989年、おもに国内自動車メーカーの意匠設計CADのニーズに応えるため、当時世界最高水準の線画描画品質と表示速度を持つAGS5000グラフィック・ディスプレイ・システム¹¹⁾を発表した。

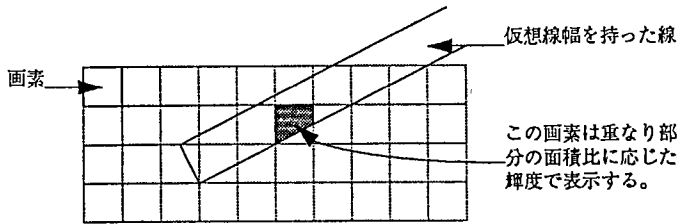


図2 面積法によるアンチ・エアシングの考え方

象を得る。

J.Truebloodらは、この原理に基づき、面積法(図2)の近似テーブル方式^[2]を提案した。

当方式はスクリーン・ピクセルに対するオフセットおよび方向を指標とした輝度テーブルを前もって用意しておき、高速に点・線のアンチ・エアシングを行うものである。ハードウェア実装に適し、最近の多くの点・線アンチ・エアシングハードウェア実装例が当方式を基本にしている。AGS5000も、その原理を当方式に求めた(図3)^[3]。しかし、当方式はポリゴンに適しない。輝度テーブルは前もって定められた線幅種(細線・中線・太線など)に関し複数個用意せねばならないが、任意の線幅(およびポリゴン)のためのテーブルをすべて用意することはできない。

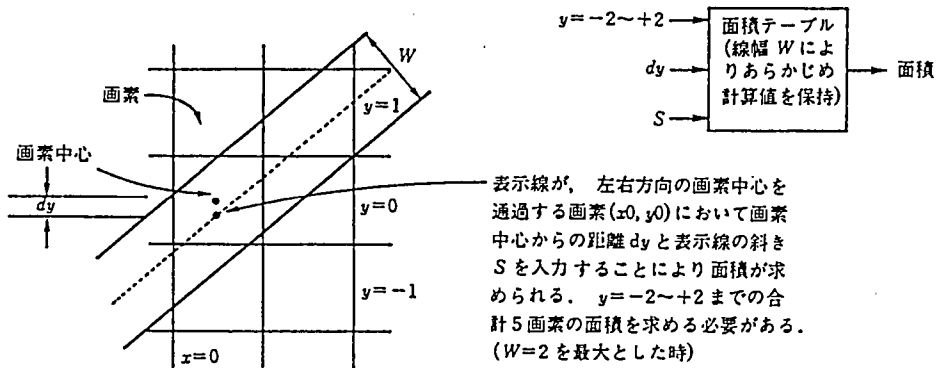


図3 AGS5000のアンチ・エアシング

ポリゴンに適した方式として、P.Haeberlyらは蓄積バッファ (Accumulation Buffer) 技術を提案した^{[4][5]}。現在、広く用いられている方式である。この方式ではスクリーン画像は何度もピクセル・メモリ内に描き直される (multi-pass rendering)。描画回数を n としたとき、各描画は $1/n$ の輝度 (色) で、ジッタ処理 (jittering)^[5] に基づくわずかに異なる位置決めにより n 回繰り返される。蓄積バッファ内で加算されることにより平均化され、エアラスが除去された最終画像となる (図4)。この技術は、点、線、ポリゴンのアンチ・エアシングに適している。さらに単純な透過を許し、モーション・ブラー (motion blur: アニメーション時間軸でのアンチ・エアシング)、

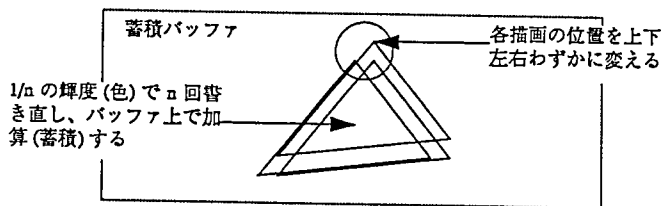


図4 蓄積バッファによるシーンアンチ・エリアシングの考え方

被写界深度など、他の特定の目的にも使用できるが、複数の描画が必要であるため非効率である（満足できるアンチ・エリアシング効果を得るためには、通常8回程度の書き直しを行う）。

一回の描画（single-pass rendering）で高精度にポリゴンアンチ・エリアシングを行う方式として、L.Carpenterはエリアス除去領域平均化蓄積バッファ技術（A Buffer）を提案した^[6]。当方式は、同一のスクリーン・ピクセルにおいていくつかの面境界が交差する場合に、ピクセルをカバーする領域の面積を持ったフラグメント（fragment）・リストとして蓄積し、最後にそれらをピクセルごとにブレンド（荷重平均）するものである（図5）。通常の面積法においては単純に決められた背景色（たとえば黒）あるいはすでにピクセル・メモリに書かれた色とブレンドするだけであるが、当リストは視点に近い順にz値で順序付けられており、複雑な点、線、ポリゴンの重なりを正確にブレンドすることができ、さらにランダムな透過ポリゴン同士の重なりにも対応できる。

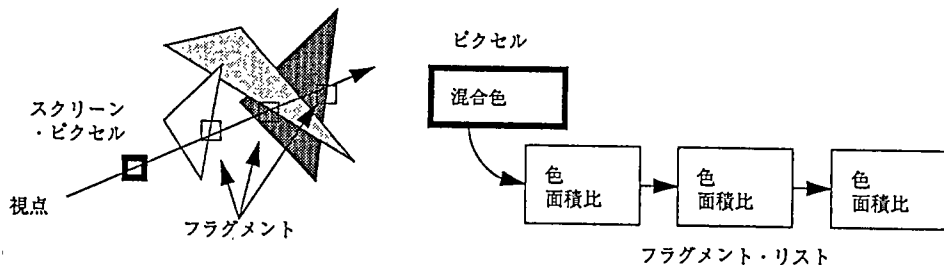


図5 フラグメント・リストによるアンチ・エリアシングの考え方

2.2 シェーディング

多角形（通常は三角形）によって近似された三次元オブジェクトを陰影づけ描画するとき、簡便的には各三角形の面を一様な光の強さ（色）で表示することがある（constant shading / flat shading）。しかし、本来曲面の場合にはその表面の光の強さは連続的に変化するのが自然であるから、滑らかな陰影づけ（smooth shading）をほどこすことがしばしば求められる。一般に、次のような2種のスムージングが行われる。

Gouraudは近似三角形の各頂点の光の強さを（頂点の法線ベクトルに基づいて）まず求め（lighting）、この光の強さをを用いて三角形内の各点の光の強さを線形補間によ

って求める方法を提案した (Gouraud shading ; 口絵3参照)¹⁷⁾。当方式は三角形のスクリーン変換に増分法を適用することにより比較的簡単に実現することができる。しかし、三角形が大きい場合、線形の色補間だけでは十分な滑らかさは得られない。

Phongは、光の強さのかわりに各頂点での法線ベクトルを用いて、Gouraudと同じ補間法により三角形内の各点の法線ベクトルを求め、その後各点の光の強さを求める方法を提案した (Phong shading)¹⁸⁾。この方法はより滑らかな効果が得られ、鏡面反射のハイライトを忠実に表現できる。しかし、ライティング計算が各点で求められ処理時間の増大を招く。また、累乗・累乗根などの実数演算が求められるライティング計算のLSI化はより困難である。現在まで、フォン・シェーディングのハードウェア機構を作成した例はない。

2.3 透明感表示

透過、すなわちオブジェクトを適当な割合で (半) 透明に描画する機能は、ガラス、プラスチックなどの材質感を表現するためのみならず、複雑な機械設計などで、内部構造を確認する、または周辺の部材を半透明にすることにより設計対象部材をフォーカスするなど、サーフェス・レンダリンググラフィックスにとっては重要な機能である (口絵4, 6, 7参照)。

これらの用途に正しく答えるためには、次のようなことがらがあわせて求められる。

- アンチ・エイリアシング効果との統合
- ライティング/シェーディング/鏡面反射効果との統合
- テクスチャ・マッピング効果との統合
- 描画順と一致しない、z方向ランダムな透明オブジェクト同士の重なりを許す。

すでに提案されている諸方式に、以上の要件を満たしリアルタイム・レンダリングとして実現できるものはない。

透過に関し良く用いられる簡便法に、スクリーン・ドア (網戸) 技術がある。ポリゴンは適当なマスクを用いて背景ポリゴンのピクセルと交互に描画される (図6)。この技術は単純であるが、解像度を犠牲にする混ぜ合わせであり、結果は粗雑である。近傍のピクセル色との組合せによる視覚効果に限定されるため、他の効果と正確に統合できず、複数の透明オブジェクト同士の重なりを許さない。

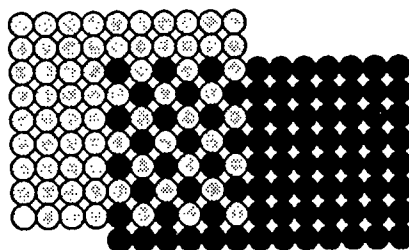


図6 スクリーン・ドア透過の考え方

より本格的な透過効果は、各ピクセルにNewellの色混合⁹⁾を施すことにより得ることができる。この方法は、背景のオブジェクトの色と（半）透明な手前のオブジェクトの色を透過率の割合で混ぜ合わせるにより透明感を表現するものである。

$$C = tC_2 + (1 - t)C_1$$

ただし、

C_1 : 手前の面 (透明な面) の光の強さ (色)

C_2 : 背景の面の光の強さ (色)

C : 全体の光の強さ (色)

t : 手前の面の透過率 ($0 \leq t \leq 1$)

$t=1$ 完全な透明、 $t=0$ 完全な不透明

この方法は屈折などを無視した簡便法である。さらにzバッファ法と単純に組み合わせられない問題がある。しかし一番の問題は、この混合を複数回繰り返す時（透過オブジェクトが複数個重なっている時）に、混合演算の交換が許されない点にある。正しい透過効果を得るためには、奥行き (z) 方向の重なり順番と混合演算の順番を一致させなければならない。

これらの問題の解決法として現在広く用いられているものに、ソートzリスト手法がある¹⁰⁾。まず、プリミティブは不透明と透明の二群に分離され、描画にはマルチパス・レンダリング手法が用いられる。最初に不透明のプリミティブのみが通常のzバッファリング処理でレンダリングされる。次に、透明プリミティブのリストはz方向でソートされ、重なり順と書き順を一致させながら、最初のパス（不透明レンダリング）の結果残されたzバッファの内容を用いて参照のみのzバッファリングを行いながら、Newellの色混合レンダリングがなされる。この手法により、上質の透過画像が得られるが、プリミティブの分離およびソートには大きな時間がかかり、大画量のリアルタイム・レンダリングに対応するのは困難である。

2.4 テクスチャ・マッピング

テクスチャ・マッピングとは、写真などから得られるデジタル画像を三次元オブジェクトの表面に張り付ける機能である（口絵5参照）。いろいろな素材の画像を用いることにより簡単に表示オブジェクトに素材感を与えることができる。また、建物の表面に窓・ドアの画像を張り付ける、あるいは飛行機の計器板の写真を張り付けることによりコックピットを作るなど、一般的に、本来必要な膨大で複雑な幾何データの定義を行わずに簡便に表示オブジェクトに細密感、リアリティを与える手法として有効である。

テクスチャ・マッピングでは、与えられたデジタル画像配列（その要素をテクセル: texelと呼び、スクリーン・ピクセルと区別する。）の対応座標値 (u, v) をポリゴンの頂点データに持たせ写像を行うことになるが、ポリゴンのスクリーン上のピクセルに対応するテクセル値が単純に一つ存在するとは限らない（図7）。

この時に、ピクセルの中心に一番近い位置にある一つのテクセル値を採用するなどの単純な手法では、次のような不都合が生じる。

■テクスチャの細部の欠落

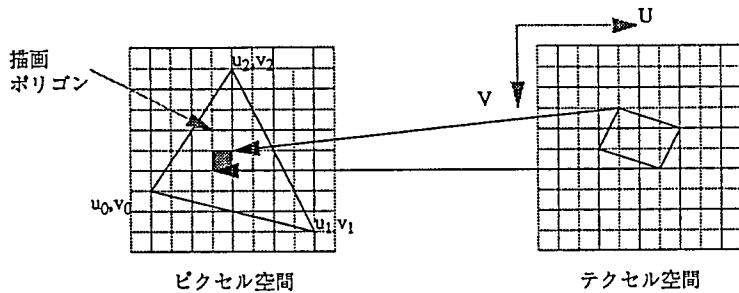


図7 ピクセルとテクセルの写像関係

■繰り返しパターンのマッピング結果のモアレ現象

■アニメーションにおけるマッピング結果の揺らぎ (swim) 現象

これらは、前に述べたジャギー現象と並ぶ、もう一つの大きなエイリアシング問題である。

これらの問題を解決するため、スクリーン上のピクセルに対応する位置の近傍の複数のテクセル値にフィルタ関数を乗じて荷重平均値を求める方法^[10]が考案されているが、大きな計算時間を必要とし、そのままではリアルタイム・レンダリングに対応できない。

これに対し Williams によって提案されたミップマップ法 (mip map)^[11]は、前もってフィルタリング処理によりすべての解像度のテクスチャ画像を作成しておき、レンダリング時の計算時間を大きく短縮するものである。

3. 8003の課題と解決方法

次に、これまでに述べた各種レンダリング方式の問題点をふまえ、本製品に与えられた課題を整理し、採用された解決方法を述べる。

3.1 課題

本製品の開発にあたり、そのレンダリング機能に与えられた課題は次のようなものである。アンチ・エイリアシング品質と速度性能は、ほぼAGS5000に準じたものである^[9]。AGS5000がモノクロ、ワイヤーフレーム描画機であるのに対し、本製品はフルカラー、サーフェス・レンダリング機であり、その差異よりいくつかの新たな課題(下線部)が追加された。

■フルカラー環境での高品位なアンチ・エイリアシング

- ・点, 線, ポリゴン (境界) すべてがアンチ・エイリアシングされる。
- ・点, 線分端点, ポリゴン頂点は水平・垂直方向, スクリーン・ピクセルの1/8以上の精度で正確に位置付けされる (10240×8192サブ・ピクセルの仮想分解能)。
- ・水平/垂直線と斜線で輝度差がない。
- ・線分端形状は, 線分方向直角に切りとる矩形とする (図8)。
- ・背景色を黒に限定しない。

■高品位なサーフェース・レンダリング機能

- ・ フラット、グーロ、フォン各シェーディング
- ・ 動的な透明感表示 (透過オブジェクトのサポート)
- ・ テクスチャ・マッピング

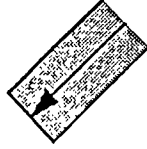


図8 接続部で輝度むらが起きない線分端形状
(AGS5000の経験より)

■世界最高水準の線・ポリゴン描画速度

- ・ 125万ベクトル/秒 (アンチ・エリアシング付き)
- ・ 60万ポリゴン/秒 (アンチ・エリアシング、グーロ・シェーディング付き)

AGS5000の経験より、点・線アンチ・エリアシング関連の諸課題は面積法で実現できることが解っている。新たな技術的課題を整理すると、次のようなものである。

- 背景 (たとえば背景ポリゴン) 色を限定しないポリゴンアンチ・エリアシング
- 任意の透過率で描画順と一致しない重なりを許す透過表示
- アンチ・エリアシング、透過、シェーディング、テクスチャ・マッピング各効果の正しい色混合 (統合)
- 以上の、シングル・パスによる高速リアルタイム・レンダリングとしての実現
- フォン品質のシェーディング

速度要求にてらすと、マルチパス・レンダリングを想定してさらに8~10倍の速さを追求することは得策ではない。シングル・パスでの実現は必須である。よってポリゴンアンチ・エリアシングおよび透過の課題を実現するためには、描画の重なり関係を管理するための何らかのメカニズムが必要となる。この問題を解決した製品は、まだ他社においても発表されておらず、本製品が始めての試みとなり技術的挑戦とされた。

3.2 解決方法

3.2.1 アルゴリズム

描画の重なり管理をとまなうシングル・パス手順として、Aバッファ法が適している。アンチ・エリアシング手法は面積法である。本手法にミップマップ法によるテクスチャ・マッピングが組み合わされた。処理手順の骨子は次頁のとおりである。

以上の手順により、前に述べた技術的課題 (フォン・シェーディングを除く) が、シングルパスで実現できる。もはや、マルチパス・レンダリング、ソート処理は必要ない。

3.2.2 ASIC化

前に述べた速度要求を達成するためには、30nsec/pixel程度のレンダリング速度を用意しなければならない。このアルゴリズムは500~800ステップのCプログラムと

- (1) 点・線・ポリゴンを、すべてある面積を持つポリゴンとしてあつかう。
- (2) ポリゴンをスクリーン・ピクセル空間に投影したときのピクセルカバレッジ領域小片、すなわちフラグメントに分解する。
- (3) フラグメントの面積 (0.0~1.0) 計算をおこなう (求めた面積は面積法アンチ・エリアシングに用いる)。
- (4) 次のポリゴン頂点情報から線形補間によってフラグメント情報を求める。
z 値、色、不透過率 (1-透過率)、テキスト座標
(色の線形補間はグーロ補間そのものである。)
- (5) フラグメントのテキスト座標よりミップマップ検索をおこないテクセル色を決定する。
- (6) 面積と不透過率の積を求め該フラグメントのアルファ値とする。
- (7) 色とテクセル色の積を求め該フラグメントの色とする。
- (8) 同一スクリーン・ピクセルに関しフラグメントが衝突したときは、z 値で順序づけられたフラグメント・リストを成長させる。
- (9) すべての描画終了後、フラグメントが衝突したピクセルに関し、フラグメント・リストを奥から順にたどりながらアルファ値 (α) とフラグメントの色 (Fc) による重なり順の Newell 色混合をおこない、ピクセル色を決定する。

```

C = 背景色 ;
for ( i = bottom → top ) {
    C =  $\alpha_i \cdot Fc_i + (1 - \alpha_i) \cdot C$  ;
}
ピクセル色 = C ;

```

して表現できるが、ソフトウェア (ファームウェア) として用意すると仮定すると、数1000 ミップスあるいは数ギガフロップスのプロセッサ能力が求められ、必要な価格性能比を得ることができない。ASICによるハードウェア・ロジック化が必須であり、さらに適当なパラレル・パイプライン構成とする必要がある。

本来Aバッファ法はソフトウェア的なアルゴリズムであるが、以下の事情によりASIC化は可能と判断された。

- 専用回路とすることにより、フレーム・バッファの多重インタリーブとバス幅の大幅拡張が可能となる。
- 最新のLSIテクノロジーの採用により20万~30万ゲート規模のASIC作成が可能になっており、ロジックの高多重化による高速化が可能である。
- VHDL (VLSI Hardware Description Language) の導入により、C言語で記述されたアルゴリズムのワイヤード・ロジックへの橋渡しが可能になっている。
- シミュレーション・エンジンの導入と表示ツールとの組合せにより、アプリケーションのデータでシミュレートするなど、膨大なデータでの機能・性能検証 (デバッグ) が可能である。

3.3 問題点

3.3.1 フォン・シェーディング

フォン・シェーディングに関しては、ゲート容量の制限よりライティング計算のASIC化を断念し、疑似フォンをソフトウェア (ファームウェア) 処理として用意することとする。用意される疑似フォンは、指定された、画面上一定のトレランスまで、

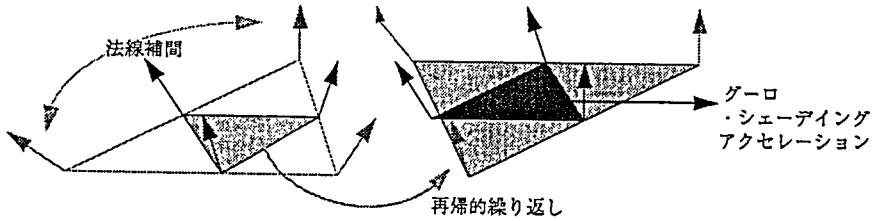


図9 再帰的三角形分割 (疑似フォン)

頂点法線ベクトル（および、他の頂点データ）の線形補間を伴う三角形の分割を再帰的に繰り返し、前に述べたゲロー・レンダリング機構にわたすものである（図9）。アンチ・エリアシング、透過表示、テクスチャ・マッピングなどのASICに実装される高品位レンダリング機能と共存しながら、円滑に描画速度・滑らかさのトレードオフを使用者に選ばせることができる。

3.3.2 フラグメント・リスト量の見積り

本方式による描画速度は成長したフラグメント・リストの数と長さ直接的に依存する。すなわち、描画オブジェクトの重なり具合により性能が“ゆるる”可能性がある。速度性能とフラグメント・リスト用メモリ総量の初期見積りは、ASICアルゴリズムのシミュレータ評価時にアプリケーション開発側から入手したいいくつかの典型的と思われる描画データを実測し、平均フラグメント・リスト長を1.4とした。この見積りの問題点およびとられた対応策を5章で述べる。

4. 実 装

以上のレンダリング方式は、8003においてハードウェア機構として実装された。次にそれらの実装をアルゴリズムの詳細とともに述べる。

4.1 8003の構成

8003の全体構成を図10に示す。8003本体はタワータイプの外付筐体であり、Sun SPARC StationのS-Bus 1スロットを占有するWorkstation Interface Board (WIB) 経由でホスト接続される。筐体内には次の3枚のU9ボードが格納されており、IGBよりビデオ・ケーブル経由で74Hzの高品位モニタに表示される。

■ Graphics Interface Board (GIB)

S-Busの8003側機構。WIBと対になり、ホストが変わった場合は置き換えられる。

■ PEX Geometry Board (PGB)

CSS（ディスプレイ・リスト）メモリを内蔵し、トラバース処理、さらにライティング計算、座標変換、クリップなどのいわゆるジオメトリ処理を行う。

■ Imaging and Graphics Board (IGB)

スキャンライン変換、zバッファリング、ピクセル混合、ビデオ信号発生などのいわゆるレンダリング処理を行う。

PGB、IGBともにパラレル・パイプライン構成であるが、前者が9個のi860マイクロ・プロセッサによるソフトウェア（ファームウェア）処理であるのに対し、後者は

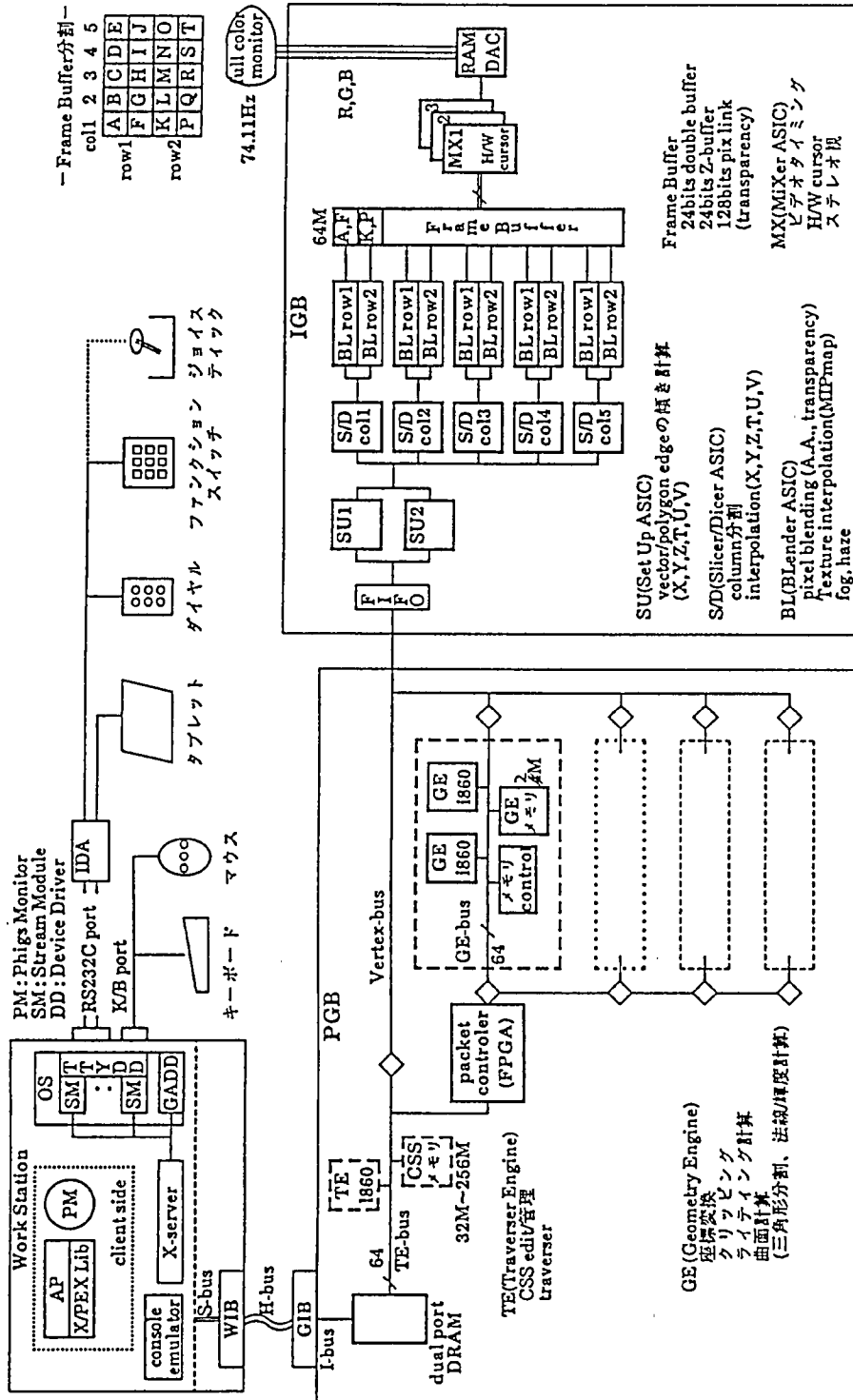


図10 8003のハードウェア全体構成

全てASICを中心とするハードウェア・ロジックである。本稿で説明するレンダリング機能は、全てIGBにおいて実現されており、以後当ボードについて述べる。

4.2 IGBの構成

8003のレンダリング機構を実現するために、次のような、それぞれ数十万ゲート規模のカスタムLSI (ASIC) が4種新規に開発された。計20個実装され、レンダリングパラレル・パイプラインを構成している。

■ Setup ASIC

点、線、三角形をプリミティブとして受けとり、ポリゴンに変換し、頂点データの増分 (辺の傾き、色・透過率・テクスチャ座標の変化率) 計算を行う。テクスチャ・マッピングのためのピクセル空間/テクセル空間拡大率 (ミップマップ・レベル) の計算を合わせて行う。

■ Slicer/Dicer ASIC

ポリゴンをフラグメントに分割する。フラグメントのスクリーン・ピクセルカバレッジ領域の面積計算を行う。さらに、ゲージ補間を含め、各頂点データの線形補間をおこない、フラグメント・データを求める。

■ Blender ASIC

スクリーン・ピクセルに対応するフラグメント・リストを生成する。zバッファリングに相当する処理、さらにアンチ・エイリアシング、透過、テクスチャ・マッピングのための色混合 (ブレンド) 処理を行いピクセルの色値を決定する。

■ Mixer ASIC

ピクセルの色値をビデオ信号に変換する。カーソル信号の混ぜ合わせを行い、モニタに表示する。さらにハードウェア・プリンクをサポートする。

前三者について、以下の項でさらに詳細に述べる。Mixer ASICについては割愛する。

4.2.1 Setup ASIC

レンダリング・パイプラインの最初のASICである。2個で並行処理がなされる。

1) 入力レジスタ

PGBより、点、線、三角形の3種のプリミティブを受けとる (図11)。それぞれ1, 2, 3個の頂点データ (P_n) を持つ。頂点データの内容は次のものである。すべて固定小数点表現である。整数部・小数部の桁数を [] 内に示す。

P_n : x_n, y_n, z_n …頂点のスクリーン座標 [x, yは11.8, zは0.24]

r_n, g_n, b_n ……頂点色 [8.4]

t_n ……頂点の不透過率 (1-透過率) [1.8]

u_n, v_n ……頂点のテクスチャ座標 (0.0~1.0) [9.15]

頂点のスクリーン座標は、モデリング・ビューイング変換、クリップ処理を終えたものである。頂点色は頂点法線ベクトルよりライティング計算を施して求められたものである。また、テクスチャ座標はテクスチャ投影計算を施して求められたものである。いずれもPGBで処理されている。さらに点にはサイズ、線には線幅情報が添えられている。

2) ポリゴンへの変換および頂点のソート

点、線プリミティブは、まず図11の通り、それぞれサイズ、線幅情報を用いて正方形、矩形ポリゴンへ変換される。以後8003のレンダリング・パイプラインでは、すべてある面積を持つポリゴンとして扱われる。次にポリゴンの各頂点は最小x値を持つものを P_0 とし、左回り (CCW) に並ぶように入れ換えられる。

3) 頂点データの増分計算

次に、全ての頂点データについて、各辺での傾き (増分) が求められる。これらは以降、Slicer/Dicer ASICにおけるフラグメント変換時に、フラグメント・データを求めるための線形補間における平面方程式の係数として用いられる。

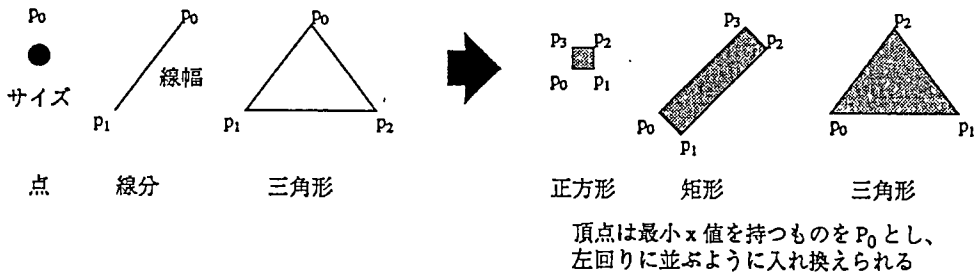


図11 IGBの基本プリミティブとポリゴンへの変換

4.2.2 Slicer/Dicer ASIC

Setup ASICからポリゴンの頂点データと増分データを受けとり、フラグメント変換を行い、フラグメント・データを求めるものである。5個で並行に処理される。

1) スライシング/ダイシング

Slicer/Dicer ASICにはSlicerモジュールとDicerモジュールがあり、それぞれ垂直方向でのスライシングと水平方向でのダイシング (図12) を行い、フラグメントのピクセル・カバレッジ領域を求め、面積 (a) が計算される (図13)。

2) フラグメント・データの算出

z, r, g, b, t, u, v の7種のデータに関し、頂点データより線形補間でフラグメント・データが求められる。補間には図14の平面方程式が用いられる。式の係数はSetup ASICにより前もって求められている。なお、ここにおいてなされる r_n, g_n, b_n の線形補間は、ゲーロ補間そのものである。

前項で求めたフラグメント面積 (a) は、そのまま面積法によるアンチ・エリアシングに使用できるが、透過が指定されていたときは、不透過率 t との積が求められる。

$$\alpha = a \cdot t$$

以降、アンチ・エリアシングと透過は統合されてブレンド処理される。

4.2.3 Blender ASIC

Blender ASICはSlicer/Dicerごとに2個ずつ用意されており、計10個の並列処理となる。一つのBlenderで垂直方向二つのピクセルを同時に出力するため、Slicer/Dicer・Blender群の一回の動作で水平・垂直 5×4 計20ピクセル分のスクリー

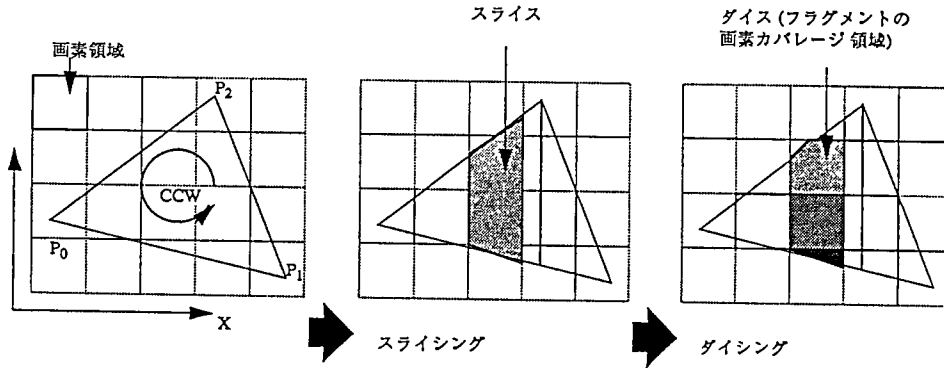


図12 ポリゴンのスライシング/ダイシング

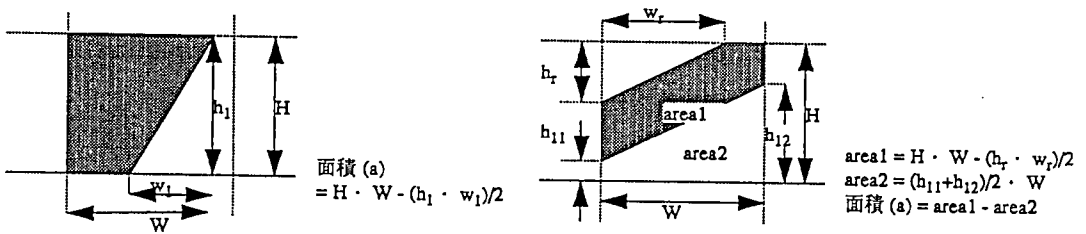
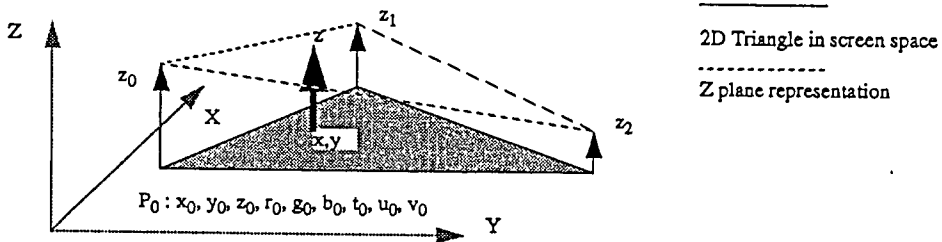


図13 フラグメント面積の計算法



General form: $Ax + By + Cz + D = 0$

Solve for Z: $z = A'x + B'y + D'$

Substitute constants: $z = dz/dx(x-x_0) + dz/dy(y-y_0) + z_0$

Solve for R: $r = dr/dx(x-x_0) + dr/dy(y-y_0) + r_0$

Solve for G: $g = dg/dx(x-x_0) + dg/dy(y-y_0) + g_0$

Solve for B: $b = db/dx(x-x_0) + db/dy(y-y_0) + b_0$

Solve for T: $t = dt/dx(x-x_0) + dt/dy(y-y_0) + t_0$

Solve for U: $u = du/dx(x-x_0) + du/dy(y-y_0) + u_0$

Solve for V: $v = dv/dx(x-x_0) + dv/dy(y-y_0) + v_0$

図14 平面方程式によるフラグメント・データの線形補間

ン領域を描画することになる。フレーム・バッファメモリはこの単位でインタリーブされている。20ピクセルに関し同時並行書き込みが可能であり、見かけのメモリ書き込み速度を上げている。

1) 入力レジスタ

BlenderはSlicer/Dicerより次のフラグメント・データ (F) を受けとる。

F:

- x, y, z… フラグメントのスクリーン座標
- r, g, b… フラグメントの色 (ゲージ補間済み)
- α ……フラグメント面積 (透過反映済み)
- u, v ……フラグメントのテクスチャ座標

2) フラグメント・リストの生成

フラグメント・リストはフレーム・バッファにルートを持つフリー・リストであり、同一スクリーン・ピクセルに二つ以上のフラグメント・データが発生した時に成長する (図15)。実際にはリストの各要素には二つまでフラグメント・データを格納することができる。通常の描画状態ではフラグメントの衝突は平均1.4であるため、たかだか一つ程度の要素数のリストである。

Blenderはフラグメント・データを受けとると、z値をチェックし、昇順になるように適当な要素位置に挿入する。結果として視線方向重なるの順にソートさ

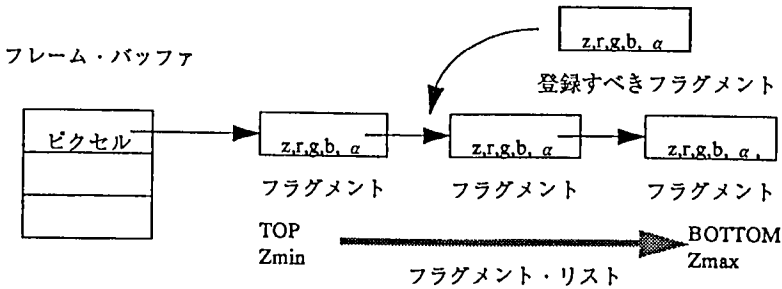
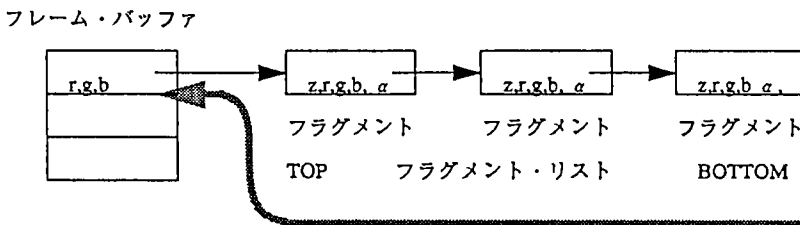


図15 フラグメント・リストの成長



```

C = 背景色;
for (l = BOTTOM → TOP) {
    C =  $\alpha_l \cdot [r, g, b]_l + (1 - \alpha_l) \cdot C$ ;
}
[r, g, b](フレーム・バッファ) = C;
    
```

図16 色の混合

れたリストとなる。

3) 色の混合

色の混合は、その画像に関する全てのフラグメント・データ登録終了後、一斉に行われる。リストのトップの要素の α の値をチェックし、すでに1.0（完全にスクリーン・ピクセルをカバー）ならば、その r , g , b 値をフレーム・バッファに書き込む。これは通常のZバッファリング処理と等価である。 α の値が1.0に満たないときは、1.0に達するまでリストをたぐり、 α 値の割合で r , g , b 値の加算処理を行う。これが色の混合（ブレンド）である（図16）。結果として面積法によりアンチ・エイリアシングと任意の透過率で書き順とは無関係に重なりあった透過物体の描画を得る。

4.2.4 テクスチャ・マッピング

テクスチャ・マッピング処理も、今までに述べた3種のASICの処理の中で同時に実行されるが、混乱をさけるため別項を設けて説明する。

1) テクスチャ画像の格納

テクスチャ画像は前もってミップマップ展開され、IGBのメモリに格納されている。ミップマップ展開のアルゴリズムについては、後述するが、ベース画像の最大解像度を 512×512 テクセルとし、下位画像の近傍テクセル群からガウシアン・フィルタ関数による荷重平均として上位の一つのテクセル値を求める処理を行い、解像度 $1/2 \times 1/2$ の画像を作成することを繰り返す、図17のようなイメージ・ピラミッドとしたものである。結果としてピラミッドの各レベルは $width = height = 2^{9-level}$ の解像度を持つ。

ベース画像そのものもフィルタリング処理で求められるため、使用者が用意する原画像は正確に 512×512 画素である必要は無い。

2) Setup ASICにおけるミップマップ・レベルの算出

まずSetup ASICにおいて、各ポリゴンのスクリーン・ピクセル座標系での面積 (A_{xy}) とテクセル座標系での面積 (A_{uv}) が求められ、その面積比よりミップマップ・レベル (MLF) が算出される（図18）。MLFも整数部・小数部を持つ固定小数点表現である。

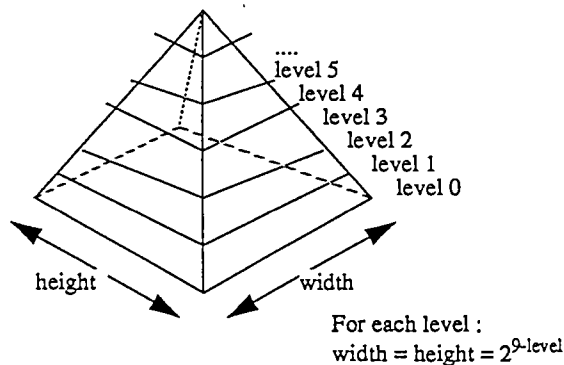


図17 ミップマップイメージ・ピラミッド

3) Blender ASICにおけるテクセル色の混合

Slicer/Dicer ASICにおいて、頂点データより線形補間により、スクリーン上の各ピクセルに対応するフラグメント・データ (x, y, z, r, g, b, α, u, v) が求められることは前に述べたが、ここで、そのフラグメントに対するテクスチャ・マッピング効果を反映した色決めの手順を述べる。

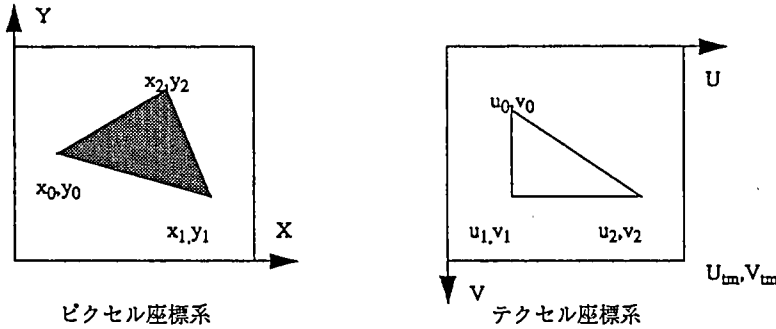
- ① Setup ASICで求めたミップマップ・レベル (MLF, 固定小数点) より近傍上下のミップマップ・レベル (MLH, MLL) を計算する。また、MLFの小数部 (MLFRAC) を求める。

$$MLL \text{ (下側, 高い解像度)} = \text{tranc} (MLF)$$

$$MLH \text{ (上側, 低い解像度)} = MLL + 1$$

$$MLFRAC = MLF - MLL$$

すなわち、MLF = 3.64 のとき、それぞれ、MLL = 3, MLH = 4, MLFRAC = 0.64 となる。



u_n, v_n は 0.0 ~ 1.0 の正規座標である。実際のテクセル座標 (u_m, v_m) は、1.0, 1.0 での最大テクセル数 U_m, V_m より、

$$u_m = u_n \cdot U_m$$

$$v_m = v_n \cdot V_m$$

- 面積 (スクリーン・ピクセル系) の 2 倍: $2A_{xy} = \text{abs}(x_0(y_1 - y_2) + x_1(y_2 - y_0) + x_2(y_0 - y_1))$
- 面積 (正規化テクスチャ系) の 2 倍: $2A_{uv} = \text{abs}(u_0(v_1 - v_2) + u_1(v_2 - v_0) + u_2(v_0 - v_1))$
- テクスチャ全体テクセル面積: $A_t = U_m \cdot V_m$
- ミップマップ・レベル: $MLF = ((\log_2(A_{uv}) + \log_2(A_t)) - \log_2(A_{xy})) / 2$

図 18 ミップマップ・レベルの算出

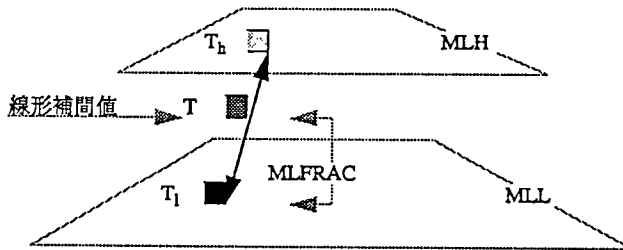


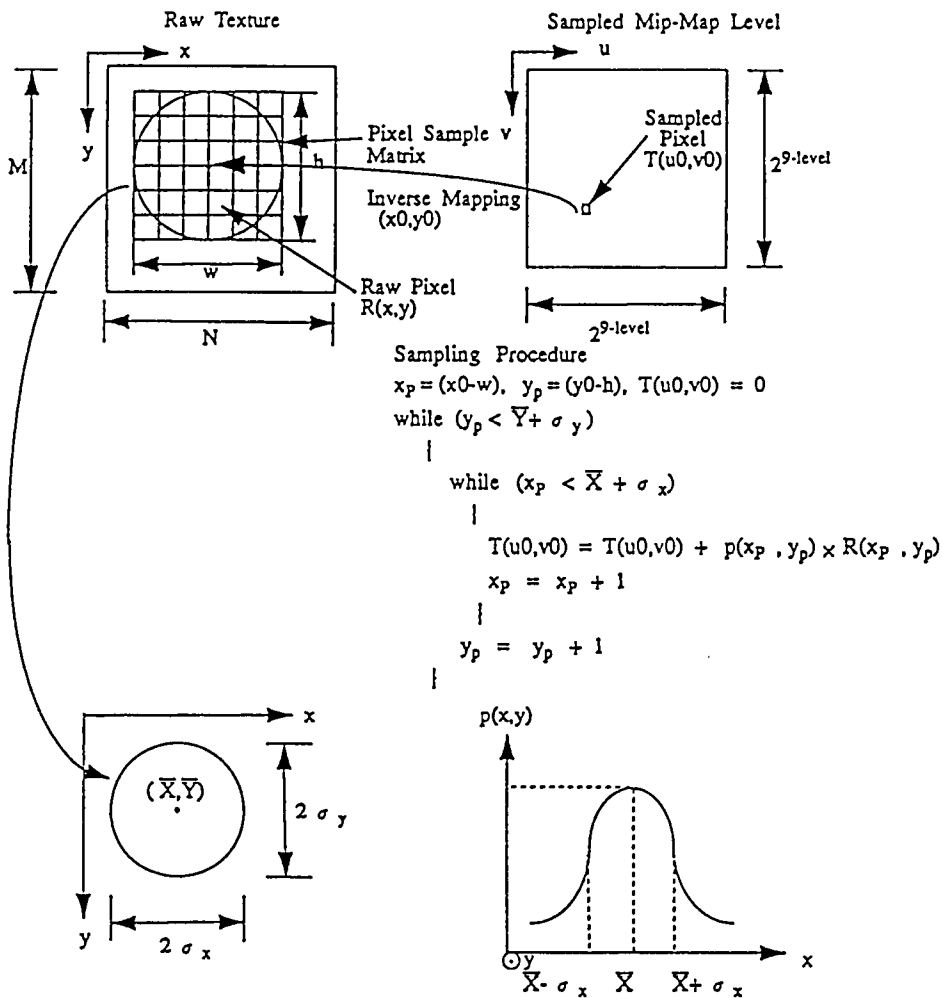
図 19 ミップマップ・レベル間の線形補間

- ② フラグメント・データ u, v より, MLH, MLLでのテクセル座標値 ($u_{th}, v_{th}, u_{tl}, v_{tl}$) を求め, それぞれの対応するテクセル値 $T_h(u_{th}, v_{th}), T_l(u_{tl}, v_{tl})$ を決定する.

$$u_{th} = u \cdot U_{im} \text{ (MLH)}, \quad v_{th} = v \cdot V_{im} \text{ (MLH)}$$

$$u_{tl} = u \cdot U_{im} \text{ (MLL)}, \quad v_{tl} = v \cdot V_m \text{ (MLL)}$$

- ③ 二つのテクセル値 (T_h, T_l) の間を MLFRAC により線形補間してマッピングすべきテクセル値 (T) を求める (図19).



Cross Sections of Bivariate Gaussian Density Function

$$p(x, y) = \frac{1}{2\pi\sigma_x \cdot \sigma_y \sqrt{1-\rho^2}} \exp \left\{ -\frac{1}{2(1-\rho^2)} \left[\left(\frac{x-\bar{X}}{\sigma_x} \right)^2 + \left(\frac{y-\bar{Y}}{\sigma_y} \right)^2 - \frac{2\rho}{\sigma_x \cdot \sigma_y} (x-\bar{X})(y-\bar{Y}) \right] \right\}$$

where

σ_x = standard deviation of $x > 0$

$-1 < \rho < 1$

σ_y = standard deviation of $y > 0$

図20 フィルタリングによるミップマップ・ピラミッドの作成

$$T = (1 - \text{MLFRAC}) \cdot T_1 + (\text{MLFRAC}) \cdot T_n$$

(T_n はr, g, b, α 各成分を持つ.)

- ④ フラグメント・データのr, g, b, α に求めたテクセル値 (T) をかけ合わせる.

$$F_{\text{textured}} = F \cdot T$$

(F, Tはr, g, b, α 各成分を持つ.)

4) Mip mapの展開

説明が前後するが、最後にミップマップ・ピラミッドの作成法を述べる。ミップマップ・ピラミッドとは、前に述べたとおり、ベースになるテクスチャ画像より、フィルタ関数により解像度1/2, 1/4, 1/8…の画像を前もって作っておくものであり、実際のテクスチャ・マッピング時に時間のかかるフィルタリング処理を行わなくて済み、高速なテクスチャ・マッピングを実現することができる。

このフィルタリングはガウスの分布関数を基にしており、荷重平均を取りながら低解像度のテクスチャ画像を作り出すものである。図20に処理の詳細を示す。

サンプリング・マトリクスのは大きさは、与えられたテクスチャ画像の大きさとミップマップ・レベルの大きさの比により決定される。また、ガウスの分布関数の標準偏差 (σ) を変更することにより各ミップマップ・レベル間の色変化のスムーズさが調整される。

5. 評価

5.1 機能

機能に関しては、ほぼ所定の課題を実現した。現在本製品はポリゴンアンチ・エイリアシングとz方向描画順の制限を持たない透明感表示をシングルパスで行える業界唯一の製品である。参考のため他社製品の実装状況を紹介すると表1のとおりである。

シングルパスで処理しきれなかった問題として、テクスチャ・マッピングと鏡面反射効果の組合せの問題をあげなければならない。この二つの効果を正しく組み合わせるには、テクスチャ・マッピング結果 (色) に鏡面反射効果成分 (色) を加算しなけ

表1 各社のレンダリング機能実装状況

ベンダー	点・線アンチ ・エイリアシング	面アンチ ・エイリアシング	透明感表示	シェーディング	テクスチャ ・マッピング
A社	○	△ (注1)	△ (注2)	△ (注4)	○
B社	○	×	△ (注3)	△ (注4)	○
C社	○	×	△ (注3)	△ (注4)	○
D社	○	△ (注1)	△ (注2)	△ (注4)	○
E社	○	×	△ (注3)	△ (注4)	×
日本ユニシス	○	○	○	△ (注4)	○

注1：蓄積バッファ方式を推奨 (処理はアプリケーションで用意する)

注2：ソートZリスト方式を推奨 (処理はアプリケーションで用意する)

注3：書き順による混合不正を保證せず

注4：フォン・シェーディングのハードウェア実装なし

ればならないが、シングルパスの8003では乗算で組み合わせざるを得ないため、鏡面反射効果が弱くなってしまう。

$$\text{正しい混合: } (Kd \cdot Lc \cdot \cos \theta \cdot Oc) \cdot Tc + (Ks \cdot Lc \cdot (\cos \phi)^{Oe} \cdot Os)$$

$$\text{8003の混合: } \frac{((Kd \cdot Lc \cdot \cos \theta \cdot Oc) + (Ks \cdot Lc \cdot (\cos \phi)^{Oe} \cdot Os)) \cdot Tc}{\text{ライティング計算 (PGB)、グーロ補間の結果}}$$

(平行光照射の例)

ただし、

Oc オブジェクトの色	Kd 拡散反射係数	θ 入射光の角度
Lc 光の色	Ks 鏡面反射係数	ϕ 正反射方向と視点方向の角度
Os 鏡面反射の色		Oe 鏡面反射の広がりの指数
Tc テクスチャの色		

この種のブレンド問題を汎用的に解決するには、Blender ASICにマルチパスレンダリング対応の機能を追加するのがより得策である。現在は、当面の代替策としてアプリケーションでの対応手法を提示するにとどまっている。

5.2 性能

速度性能に関しても、ほぼ目標を達成した。現在本製品は100～200万ベクトル/秒級機ではGPC PLBベンチマークで世界最高速値を示している。InstaCUIITYはフラグメントをリストでもつことによりはじめて可能になったが、フラグメント・リスト方式は性能がリスト長に依存する。8003は3.3.2項で述べたように平均リスト長を1.4として設計された。ポリゴンは通常zバッファonで描画されるため、プリミティブが重なっているピクセルは少なくリンク長は伸びにくい。1.4を越すことはほとんどない。しかしベクトルの場合は、データと表示状態によっては平均リンク長が2.0に近づき、性能がでないケースもある。こうした点を考慮し、8003ではフラグメントをリストにせず、常に直接スクリーン・メモリにブレンドしていくモードも用意してある。このようなブレンドでもベクトルの場合は描画品質はほとんど落ちない。しかしそれでもブレンド処理のコストはかかるため、ベクトルで画面全体を何度も塗りつぶすような過度の重なりがある場合—もはやワイヤーフレーム表示としての意味はないが—は8003のフル性能はでない。なお、他社製品のアンチ・エリアシングに見られるようなスクリーン・メモリとブレンドしない(背景の色を固定とする)単純置き換えの機構は、描画品質も落ちるうえに、上記のような実使用上ありえないケースでしか意味がないため、用意していない。

こうした直接ブレンドモードを用意せざるをえなかったという点で、シングルパス統合レンダリング機構としてのInstaCUIITYは完全には成功していない。しかし、アプリケーションデータで繰り返しシミュレーション、性能評価を行ったこともあり、実使用に強く、アプリケーションでピーク値に近い性能が出せるグラフィックスに仕上がっている。最近のグラフィックス製品にはカタログ性能は高いが実際に使用すると非常に低い性能にとどまるものが多く見受けられることを考えると、本製品のアドバンテージの一つである。

6. おわりに

本製品は1993年6月より客先設置を開始したが、その描画品質に関して高い評価を得た。現在市場に供給されているグラフィックス製品で、これだけの画質を持ち、この速さが出ているものは他にないと認識している。

本製品で実現できなかった課題としてはフォン・シェーディング(ハードウェア版)とテクスチャ・マッピング結果への鏡面反射の付加が上げられる。

残されたレンダリング上の重要な課題としては、影 (shadow casting) と写り込みの問題がある。いずれも本格的なハードウェア・アクセレーションは次世代を待たねばならない。

また、セールス側より、描画品質を落しても廉価な製品をとの声も大きい。本製品で試みたASICによる実装手法では、(パラレル数を落すことにより)速度を落して廉価版を用意することはできるが、描画品質を落して、高速かつ廉価な製品を用意することはできない。もう一つの今後の課題である。

最後に、本テクノロジーの実現に尽くしたMegatek社Advanced Rendering Groupの五人のエンジニア、Roman Kuchkuda, John Rigg, Manny Enriguez, Jim Henson, Curt Stehleyに敬意を表し、さらに、本製品の開発・提供に御努力いただいた、Megatek/弊社双方の多くの方々に厚くお礼を申しあげて、本稿を閉じることとする。

-
- 参考文献 [1] 森分芳男, AGS5000 グラフィック・ディスプレイ・システムの開発, ユニシス技報 24号1990, pp.3~17.
- [2] J.Trueblood/M.Mizuno, An Efficient Anti-Aliasing Software Algorithm, SID 86 DIGEST, pp.348~351.
- [3] 嶋田博明, ラスタ・スキャン型ディスプレイ用の高品位アンチ・エリアシング方式について, ユニシス技報24号1990, pp.18~30.
- [4] P.Haeberly, The Accumulation Buffer: Hardware Support for A High-Quality Rendering, ACM Computer Graphics, Vol.24 No.4 1990, pp.309~318.
- [5] J.Neider/T.Davis/M.Woo, OpenGL Programming Guide, Addison-Wesley, 1993.
- [6] L.Carpenter, The A Buffer and Anti-Aliased Hidden Surface Method, ACM Computer Graphics, Vol.18 No.3 1984, pp.103~108.
- [7] H.Gouraud, Continuous Shading of Curved Surface, IEEE Transactions on Computer, C-20(6) 1971, pp.623~628.
- [8] Bui-Tuong, Phong, Illumination for Computer-Generated Pictures, Comm.ACM, Vol.18 No.6 1975, pp.311~317.
- [9] M.Newell/R.Newell/T.Sancha, A New Approach to the Shaded Picture Problem, Proc. ACM, Nat. Conf. 1972, pp.443.
- [10] E.Feibush/M.Levoy/R.Cook, Synthetic Texturing Using Digital Filters, Proc. of SIG-GRAPH' 80, pp.294~301.
- [11] L.Williams, Pyramidal Parametrics, Computer Graphics, Vol.17 No.3 1983.
- [12] Imaging and Graphics Board (IGB) Design Specification Rev1.0, Megatek Corporation
- [13] 鷲島/西澤/浅原, 並列図形処理, コロナ社.
- [14] 中前/西田, 3次元コンピュータグラフィックス, 昭晃堂.

執筆者紹介 渡辺 啓 (Satoru Watanabe)

昭和47年福島大学教育学部理科卒業，48年同大同学部
研究生課程（情報処理教育）修了．同年日本ユニシス
（株）入社．エアライン予約システムの開発，グラフィック
関連基本ソフトウェアの開発に従事．62年より
AGS5000の開発に従事．平成4年より8003の開発に従事．
現在，オープンプロダクト本部ハードウェア開発部グラ
フィックシステム開発室に所属．



オープン環境との最適な共存を実現する
オープン・エンタプライズ・サーバ
-ITASCA 3800-
UNISYS 2200/3800 シリーズ

平成8年1月より販売を開始した新世代汎用機「オープン・エンタプライズ・サーバUNISYS 2200/3800シリーズ」の概要について紹介する（写真1）。

UNISYS 2200/3800シリーズはコードネーム「ITASCA 3800」（イタスカ）として開発が進められてきた製品であり、超大型機UNISYS 2200/900シリーズの後継機である。オープン環境と汎用機の上で蓄積されたユーザ資産の最適な共存を実現する新世代汎用機である。最先端の高性能なCMOS技術を採用し、UNISYS 2200/900の8倍の能力を一つの筐体で実現しつつ総コスト（TCO: Total Cost of Ownership）を飛躍的に低減した。また、業界第一の実績のある並列処理技術を一層充実させた。

1. はじめに

高性能なPCやUNIXに代表される情報技術いわゆるオープンプラダクト環境が急速に普及しつつある。ユーザの最大の関心事はこれまでに汎用



写真1 オープン・エンタプライズ・サーバ
ITASCA 3800 シリーズ

本稿に記載の会社名、商品名は、一般に各社の商標または登録商標である。

機の上で蓄積された膨大な情報資産を最大限に活用しつつこうした新しいオープン情報技術を安全に導入するにはどうすればよいかということである。情報投資の抑制とダウンサイジングの潮流は、分散処理の普及を急速に加速した。

一方、CMOSを採用した汎用機の価格性能比が急速に改善される中で、汎用機の重要性が再確認され、一方的なダウンサイジングではなく、システムをトータルなライフサイクルでみた投資効率、すなわちTCO (Total Cost of Ownership)を考慮したバランスのとれたシステムの在り方を示すことが求められている。規制緩和に伴う市場競争の激化(メガコンペティション)の中で企業の競争優位を獲得するためには積極的な情報技術の活用がますます必要とされる。CMOSと並列処理技術の成長により、一層低価格となる汎用機のリソースや性能をふんだんに活用し、新しいビジネスを展開することが可能となりつつある。こうした中で「企業情報モデル」と「情報基盤の整備」が急務となっている。

ユニシスは企業情報システムを四つのモデルで定義した。企業の基幹業務を支える大規模トランザクション処理やバッチ処理を行う「業務系モデル」、従来の情報系あるいはレポート処理を行う「情報系モデル」、情報を共有しつつ仕事のプロセスを情報化し業務の生産性を向上させる「オフィス支援系モデル」、そしてこれらを結合するネットワーク処理を行う「連携基盤モデル」である。とくに、前者の三つのモデルに対して、C/Sシステムの適用による分散処理を導入する。このときの汎用機の役割をオープン・エンタプライズ・サーバと位置付けた。

一方、ユニシスの並列処理技術の特徴は、単にプロセッサ処理分野だけではなく、入出力処理や通信処理などすべての分野に高性能のCMOSと並列処理技術を適用していることである。並列処理実現の中核となる製品が拡張データ処理装置「XPC」である。情報系モデルの分野では、データウェアハウスに代表される大規模DBの「高速検索」要求に応える「OPUS 2000」と、基幹業務用DBとリアルタイムに連動した「高速な更新と検索」の両方の要求に応える「DataCentral 4000」を提供する。

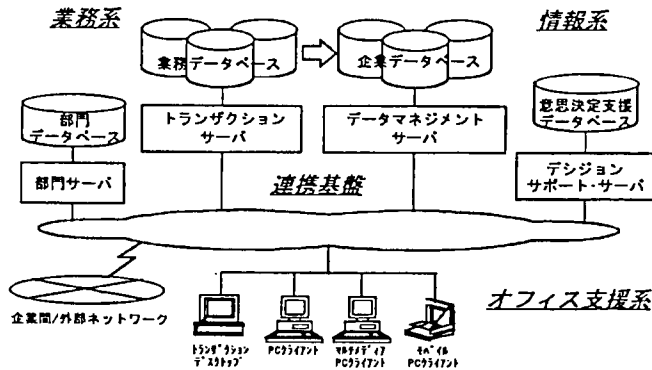


図1 企業情報システムモデル

2. 企業情報システムモデル

ユニシスは、汎用機の上で蓄積された情報資産を生かしながら、オープン環境で新たに構築されている情報資産を実績ある安定した汎用機の上でも活用できる分散協調型のオープン・エンタプライズ・サーバの実現を目指している。

企業情報システムは、従来の中央の汎用機と部門ごとに構築されたC/Sの併存型時代から、汎用機の上に蓄積された情報資源と連動した「分散協調処理システム」である第2段階のC/Sシステムへの変遷期にある。汎用機をエンタプライズ・サーバと位置付け、大規模トランザクション処理やDB処理などの中核とし、各部門のC/Sとの協調を図った「エンタプライズ・クライアント・サーバ・システム」である。

企業情報システムの対象を三つのAP分野と基盤となるネットワーク分野に分類した「企業情報モデル」を提案する(図1)。この企業情報モデルに最適な処理形態と製品(プラットフォーム)を配置する。

1) 業務系モデル

企業活動の基幹となる業務でそのシステムが停止すると、企業活動そのものが停止するAP(ミッションクリティカル)を対象とする。そのためには、エンタプライズ・サーバの持つ安定性、高信頼性、大規模な拡張性が不可欠である。トランザクション処理形態やバッチ処理形態がこのモデルの中核となる。ITASCA 3800シリーズが最適とされる分野である。適用業務には、勘定系システム、予約システム、受発注システムなどがある。

2) 情報系モデル

業務系APで蓄積されたデータを検索・分析し、企業戦略や戦術の立案を支援するAPが対象である。これらを実行するのは、エンタプライズ・デシジョン・サポート・サーバ(DSSサーバ)「OPUS 2000」とエンタプライズ・データ・マネジメント・サーバ「Data Central 4000」である。「OPUS 2000」は大量データの高速検索機能を重点において開発された製品であり、「DataCentral 4000」は同一のDB上でデータのリアルタイムの更新と検索の両方の機能を提供する。意思決定を支援するDSSサーバには、超並列型のプラットフォームが向いている。APの特性に応じていずれかの最適なプラットフォームを選択する。レポート処理やDSS処理形態がこのモデルの中心となる。顧客サービス、経営支援、市場分析・開拓などが適用業務である。

3) オフィス支援系モデル

企業レベルでのグループ協調作業の支援をする。ワークフロー管理、文字/画像/音声メール、分散ストレージ管理、分散プリント処理などのグループウェアAPが対象である。

4) 連携基盤モデル

企業の内・外の、データ・情報・知識のグローバルな共有と交換を可能とするネットワーク分野が対象である。CALSを実現するEDI(電子データ交換)やEC(電子商取引)など、企業外との広範なネットワーキングに対応する。

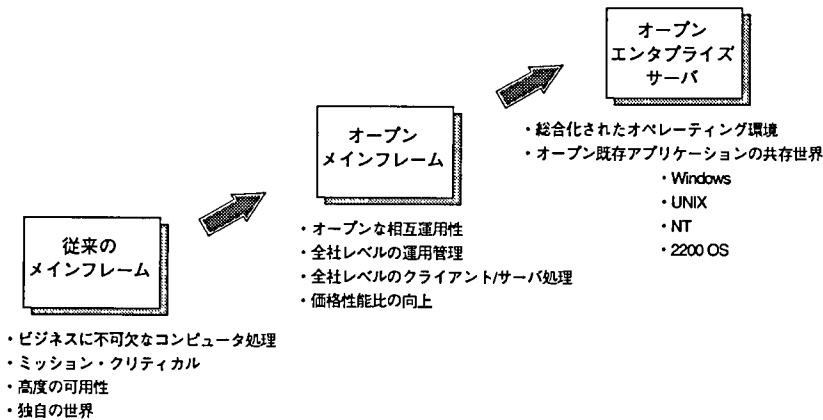


図2 オープンとの共存・共栄

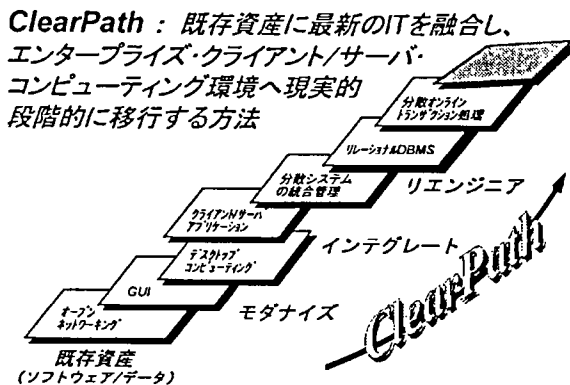


図3 クリアパス実現のステップ

3. オープン環境との共存・協調を実現

エンタプライズ・サーバとしてのITASCAのオープン化対応には二つの分野がある。

第一は、エンタプライズ・サーバとPC、UNIX間での分散処理や連携処理を可能とするインターオペラビリティの提供である。例えば、エンタプライズ・サーバ上でODBC機能を提供することによりODBC対応のPCからエンタプライズ・サーバ上の各種DBをアクセスしPC上の各種ツールにより中央のデータを活用することが可能となる。また、複数のプラットフォームの上で分散トランザクション処理を実現するミドルウェアの「TransIT Open/OLTP」を提供している。さらに、UNIX系の業界標準であるONC RPC(リモート・プロシージャ・コール)、マイクロソフト社のRPCを提供し、さらに将来的には、汎用機上のAPをオブジェクト化して活用することを狙いと

し、分散OLEの搭載も計画している。

第二には、エンタプライズ・サーバ上でのSVR4、SPEC 1170に対応するUNIX環境を実現し、APのポータビリティを提供することである(図2)。

● ユーザ資産の継承と付加価値の実現

現行のシステムで蓄積した情報資産を継承しながら、中央集中処理型のコンピューティング環境から、オープン技術を導入しつつ統合されたエンタプライズC/Sコンピューティング環境へ順次移行することが投資リスクが少なくかつ市場の変化に柔軟に対応できる。ユニシスでは、これを実現するアプローチを「ClearPath」(クリアパス)と名付けてサービスと共に提供する。これを支援するソフトウェア群の整備を図る。「ClearPath」実現のステップには3段階があり、モダナイズ、インテグレート、リエンジニアリングと呼ぶ(図3)。

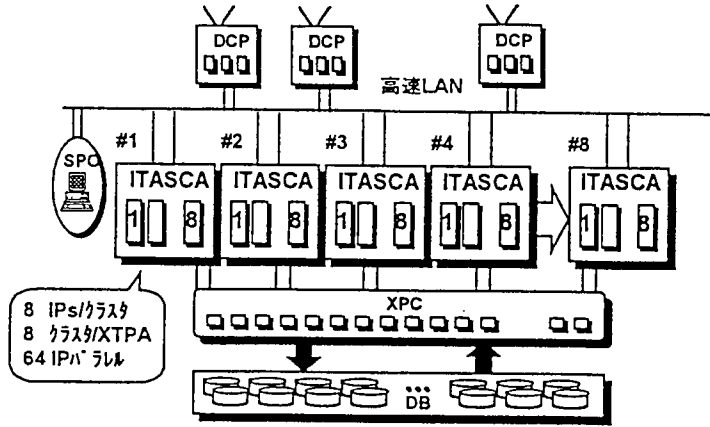


図4 ITASCAの大規模並列処理

- 1) モダナイズとは、既存のAPに対してAPの変更は行わず、CUI(Character User Interface)のGUI化(Graphical User Interface)などである。これにより、PC、UNIXの導入、市販ソフトウェアとの連携などトランザクション処理のデスクトップ環境への統合が可能となる。
- 2) インテグレートとは、異なるシステムを一つの環境に統合することである。ユーザは既存の基幹系システムとは別に目的別に独立した個別システムや部門ごとのC/Sシステムを構築している。そのためDBや運用環境などが必ずしも統一されていない。システムの運用効率改善のために基幹系のサーバと融合し連動して稼働するシステムに作り変えていく必要がある。
- 3) リエンジニアリングとは、ビジネスに最も合った新しいAP環境の構築を行うことである。企業のAPは、ビジネス環境変化に伴い有効性が低下するため、また、新しいサービスに対応したAPの開発が欠かせない。これは企業のリエンジニアリングであり、同時に情報システムのリエンジニアリングを意味する。それには現在のシステムから脱してエンタプライズC/Sシステムへ移行していくことが必要となる。例えばRDBへの移行により、各種の市販ツールが使用できる。また分散オンライン・トランザクション処理の使用により、複数のプラットフォームやDBにまたがるトランザクション処理を可能とする。

4. ITASCA 3800における並列処理技術

ユニシスではいち早く並列処理システムの有用性に着眼し並列処理システムの開発を継続してきた。1988年には、並列処理アーキテクチャ(XTPA) eXtended Transaction Processing Architectureを確立し、並列処理基盤を支えるソフトウェア群を整備した。1989年には世界に先駆けてエアライン業務の並列処理システムを稼働させた。その後も実稼働を通じての経験を生かしソフトウェアの機能拡充や高性能CMOSを採用した価格性能比やTCOの改善、一層の付加価値を高める拡張データ処理装置[XPC]などの開発を継続してきた。現在では、エアライン、金融、社会公共等の業務分野にまで広く並列処理システムの適用分野が拡大している(図4)。

● トータルシステムとしての並列処理技術

並列処理システムの要求される背景には、TCOの削減、柔軟なスケラビリティの要求、そしてシステムのトータルスループットの拡大がある。ユニシスの並列処理システムはこれらの要求に応えるものである。とくに、システム全体(プロセッサ処理、入出力処理、通信処理)に対して並列処理技術を採用しているところにユニシスの特徴がある。

トランザクションの滞留時間の内訳は、通信処理、資源のロック、DBのアクセス、業務処理、DBの更新、履歴更新処理、そして通信処理に分けられる。したがって並列処理はこの時間要素の全てを短縮するように設計される必要がある。例えば、滞留時間のうち中央処理装置(CPU)の業務処理時間の割合は全体の3分の1程度である。

CPU処理時間はCPU処理能力の向上や大容量のメモリの活用で短縮される、仮にCPUの速度が2倍となってもシステム全体の時間短縮への有効度は15%である。DBのアクセス・更新や履歴更新書き込みなどの入出力処理に要する時間が大きな割合を占め、かつファイルの共有のための排他制御時間(レコードロック制御)の割合が大きい。これを解決するプロダクトが「XPC」である。並列処理システム全体を単一システムとして運用管理することも必要である。システム間に制御LANを張り、複数システムの同期や制御・操作を一元管理する必要があり、これを実現したプロダクトが「SPO」(Single Point Operation)である。

5. ITASCA 3800と意思決定支援システム

企業情報システムモデルで定義した、情報系モデルを実現するDSS(Decision Support System)について紹介する。

意思決定支援システム(DSS)とは、「ビジネス上で意思決定をするユーザが、極めて大量のデータに対してタイムリにアクセスし、操作し分析できる新しい情報システム」であり、これまでに企業の基幹システムの上で蓄積された膨大な情報を活用し、企業の競争優位を実現するためにこれからの企業戦略の要となる。規制緩和に伴い、市場はメガコンペティションの時代に突入する。その中で最も期待されているのが、市場環境の変化に対応し、顧客の個性化や価値観の多様化に応えるターゲット・マーケティング・アプローチである。市場のセグメントを明確にして効率の良い、より収益性の高いビジネスを展開することがその狙いである。DSSは1970年代にも話題になった情報技術であるが、長期間にわたる大量蓄積データの活用が、処理速度とコスト面などから実現できなかった。1990年代に入り、並列処理技術の発展など大量のデータを扱える情報技術が経済的に利用可能となった。DSSは、この情報技術の発展を受けて、大規模なデータを中心に、企業内の全レベルでビジネス上の意思決定を支えるシステムに発展している。

● DSSを支える情報技術

DSSを可能にした情報技術として、

- ・高性能CMOSによる価格性能比の大幅な向上
- ・低価格で大容量な記憶装置の普及

- ・並列プロセッサ・アーキテクチャ
- ・並列データベース・アーキテクチャ
- ・新しいデータ収集技術
- ・高度な情報分析技術

・使いやすいツールを活用したC/Sシステムなどが挙げられる。これらにより、企業がこれまで長期間にわたり基幹業務を通じて蓄積してきた膨大なデータを意思決定支援のために活用することが可能となった。

● 「OPUS 2000」と「DataCentral 4000」

企業情報システムモデルにおいて情報系モデルの中に二つのプラットフォームを定義している。すなわち、エンタプライズ・デシジョン・サポート・サーバとエンタプライズ・データマネージメント・サーバである。その定義は、要約すると、前者はデータウェアハウスに代表される「超大量情報の検索機能を重点」にした製品であり、後者は基幹系と連動し「大量の情報のリアルタイム更新と検索」を可能とする製品である。アーキテクチャ的には前者は徹底したオープン技術を基盤として開発され、後者は実績のある堅牢性の高い汎用機の技術を基盤として開発されたものである。

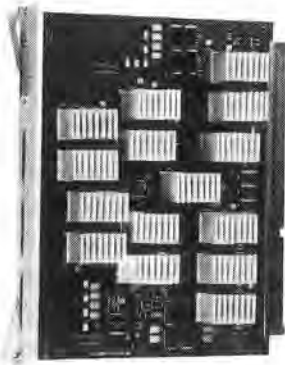
6. オープン・エンタプライズ・サーバ

ITASCA 3800シリーズ

● 大規模なスケラビリティ

ITASCA 3800はオープン・エンタプライズ・サーバと位置付け、2200の基本アーキテクチャである2200/XPA (eXtended Processing Architecture)を踏襲した「UNISYS 2200/900シリーズ」の後継機である。「UNISYS 2200/900」相当の処理能力を高性能なCMOS技術で実現し、価格性能比を大きく改善した。

ITASCA 3800では、並列処理、スケラビリティ、シングルシステム、ノンストップ機能の強化を行った。さらに、トランザクション処理の高速化やバッチ処理の高速化も実現している。ユーザは汎用機の上で、固有の付加価値財産を蓄積してきた。とくに大量データに対する高速処理、集中処理、その信頼性の高さは大きな特徴である。こうした、ユーザの持つDBとAPを保証しつつ、並列処理の環境に円滑に移行できるようにする。また新しいC/Sシステム環境への対応のため、データ検索処理の高速化を図った。



IP: インストラクション・プロセッサ
 SC: ストレージ・コントローラ
 写真2 IP/SCボード実装技術

● ITASCA 3800のハードウェア

すでに実績のある並列処理機能のさらなる強化を図った。

1) 64台のCPUによる並列処理

ユーザの業務処理量に応じてシステムが自由に拡張できる機能である。「UNISYS 2200/900シリーズ」のXTPAでは4ホスト/32台のCPUまでの並列処理が可能であったが、ITASCA 3800では8ホスト/64台のCPUにまで拡大する。これに対応して、拡張データ処理装置「XPC」はキャッシュの容量を2倍に拡大し、64CPUからの要求を並列処理できるようにする。

2) ロードバランス機能

自動的に処理量の平均化を図り並列処理コンピュータの資源を有効活用するためのロードバランス機能を提供する。

3) ノンストップ機能

365日/24時間連続無停止運転を可能にする機能をさらに向上する。光ファイバ・チャンネル(SBCON)を使い、システム全体を停止せず、入出力装置の増設や障害保守を可能とする。また、電源を供給したまま、保守を可能とするホットリプレースメント機能を「DCP/600FT」およびホスト装置にまで拡大する。

4) トランザクション処理

ITASCA 3800ではトランザクション処理の高速化をはかった。高速CMOSベース

のCPUの開発、「XPC」キャッシュメモリの拡大による能力強化、高速チャンネル「BMC」、「SBCON」、「SCSI-2W」、「FDDI」、「ATM」の提供、「DCP/600FT」の性能強化、高速通信機能等を提供する。

5) バッチ処理の強化

また、基幹系のシステムでは膨大なバッチ処理を抱えており、バッチ処理時間の短縮が強く要求されている。バッチ処理の高速化のために、「UNISYS 2200/900」相当の性能をもつ高速CMOS CPUを開発し、主記憶容量を「UNISYS 2200/900」の2倍である4GBまで拡大したので、プログラムの処理時間や「サイババッチ」の効果がさらに向上する。また、「XPC」のキャッシュ容量を2倍に拡大したことにより一層の入出力時間の短縮を実現した。

6) 大規模処理への対応(スケーリング)

ユーザのプログラム・アドレス空間と主記憶を媒介する仮想記憶空間(サイバ空間という)を「UNISYS2200/900」では560テラバイト(10の12乗)であったが、これをペタバイト(10の15乗)まで拡大し、大規模処理に対応する。

● ITASCA 3800のハードウェア・テクノロジー

ITASCA 3800では0.5ミクロンのデザインルールの100万ゲートレベルの高性能CMOS ASICを全面的に採用した(写真2)。

中央処理装置(CPU)は、2個のASICチップで構成され、「UNISYS 2200/900」相当の性能を実現する。一つのキャビネットには8 CPUが実装されるので、キャビネット単位の比較では「UNISYS 2200/900」の8倍の能力を実現することになる。CMOSの採用により価格性能比・設置スペース・消費電力も大幅に改善され、TCOの削減に大きく貢献する。同等性能の「UNISYS 2200/900」と比較すると、設置面積で20数分の1に、消費電力量と発熱量は10数分の1となる。

空調方式は空冷方式となり、電源も50/60 HZの200-240V単相電源となる。

入出力チャンネルは192チャンネルがサポートされる。提供されるチャンネルの種類は、BMC、SBCON、SCSI-2W、イーサネットLAN、FDDI LAN等であり、将来的にはATMの提供も計画している。

● ITASCA 3800のソフトウェア

ITASCA 3800ではオープン環境との共存・協調を強化するためのソフトウェア群を提供する。とくにPC、UNIX、汎用機のクロスプラットフォームでの開発・実行環境、データベースへのアクセス環境、分散トランザクション環境および分散ファイル・アクセス環境の分野でオープン化対応を行う。

開発実行環境の面では、UNIX SVR 4対応を行う。ITASCA 3800上でUNIXアプリケーションの開発・実行環境の提供、また、流通UNIXアプリケーションの移植・実行が可能となる。分散設置されている複数のUNIXシステムをITASCA 3800上で統合することも可能となる。汎用機上にUNIXを搭載することにより、UNIXのアプリケーションが汎用機の持つ強固なインフラ機能を付加価値として享受することができるようになる。

データベースへのアクセス環境のオープン化は、Windows PCの業界標準アプリケーションであるODBC、InfoAccessを採用し、PCからRDMSやDMS DBのアクセスを可能とする。分散トランザクション環境では、X/OpenDTP準拠のAPIを提供する。これによりWindows PC、PCサーバ、UNIXサーバなどと連携したC/Sシステムを構築できる。分散ファイル・アクセス環境では、TCP/IPのFTPプロトコルに基づいた高速ファイル転送、業界標準の分散ファイル・アクセス・システムであるNFSファイル・サーバ機能、分散プリント機能などを提供する。

7. 将来に向けて

ユニシスの提案する企業情報システムモデルとそれを実現する最新のITASCA 3800について説明した。まとめとして、その先をにらんだ製品動向について概観する。

高性能のCMOS技術を採用した並列処理システムが、将来の汎用機の基盤技術として主流にな

る。ユニシスでは1989年以来並列処理を市場に提供し業界では最も多くの実績を誇っている。今後ともユーザの実際のアプリケーション構築、運用等の経験を反映しつつ製品の機能、性能、使い易さ、TCOの改善を図り、その上で「XPC」のような一層の付加価値製品を提供し差別化を図っていく。ITASCA 3800では0.5ミクロンのCMOSチップを使用するが、将来システムにおいてはさらに高度な0.3ミクロンレベルの技術を採用する予定である。CMOSの全面採用に加えて、ユニシスの製品ラインである「2200シリーズ」と「Aシリーズ」さらに「UNIXシリーズ」の間で業界標準の技術の導入等により可能な限りのハードウェアの共通化を進める。これにより、継続的な開発・製造コストの低減を図る。

ソフトウェアの分野では、オブジェクト指向技術が今後の重要な技術となる。オブジェクト指向技術の良さは、既存ソフトウェア資産の再利用を可能とし、ソフトウェアの開発や保守の生産性を大きく向上できることである。ユニシスは、これまでにトランザクション処理分野において「LINC」という「統合CASE」を提供している。「LINC」は非常に生産性が高く、「LINC」で作成したユーザのAPは、そのまま並列環境で動作する。「LINC」に対してオブジェクト指向技術を全面的に採用した新しいLINCであるOO-LINC(Object Oriented LINC)による開発環境を実現する。「LINC」のオープン化をすすめ、プラットフォームに依存しないシステム構築を可能とする。

さらに、将来システムでは「2200シリーズ」や「Aシリーズ」といったエンタプライズ・サーバがUNIXに加えてWindows/NTとも連携して動作するシステムを実現する。並列処理システムによるスケラビリティを活用し、従来の汎用機で蓄積された情報資産を活用しつつ、技術革新の著しいオープンな情報技術との融合を積極的に行いバランスのとれた柔軟なシステムを実現する。

Open Parallel MAPPER

1. はじめに

日本ユニシスは、DSS（デジジョン・サポート・システム—意思決定支援システム）市場への本格的な参入を目指し、新しいプラットフォームとして並列オープンサーバOPUS 2000を平成7年7月に発表した。OPUS 2000では、並列処理化されたリレーショナル・データベースORACLEが提供されているが、これに加えて従来より高い評価を得ているMAPPERが、Open Parallel MAPPERとして平成7年12月に発表された。ここでは、Open Parallel MAPPERの概要を紹介する。

2. OPUS 2000の概要

Open Parallel MAPPERの概要を説明する前に、このMAPPERが稼働するプラットフォームについて少し説明する。

2.1 OPUS 2000 出現の背景

ビジネス環境は規制の緩和、顧客の要求の変化、ビジネスの国際化、円高、アジア諸国の台頭などの要因により大きく変化している。企業は、好むと好まざるにかかわらず、生き残るための施策を立て、それを実行する事を強いられている。このような事は、昨今始まったのではなく、程度の差こそあれ今までずっと続いてきた。最近の変化で大きなものは、画一的なマス・マーケットを対象にしたマーケティングからターゲットを絞り込んだマーケティングへの移行である。これは、ターゲット・マーケティング、マイクロ・マーケティングなどと呼ばれるものである。ターゲット・マーケティングでは、過去のデータを分析し絞り込んだターゲットの傾向を推測し、それに合った施策を取る事が重要になる。そこでDSSが、新しい情報系システムとして望まれるようになった。

DSSで取り扱うデータ量は数百万件単位の膨大なものである。このような膨大なデータを利用したいという要求は、前々からあったが、短時間の内に膨大なデータを試行錯誤の過程を経ながら分析するには、技術が十分に成熟していなかった。しかしながら、昨今の技術の進歩、即ち、低価格で高性能なプロセッサの普及、低価格で大容量・高密度な記憶装置の普及、パラレル処理を可能にする技術、高速ネットワークなどにより、技術的にDSSを行う事が可能になってきた。OPUS 2000は、このような状況を背景に登場したDSS構築のためのプラットフォームである。

2.2 OPUS 2000の特徴

OPUS 2000の大きな特徴は、並列アーキテクチャの採用である。ソフトウェア的には、標準UNIXに準拠したOS (SVR4/MK) を採用している。また、ハードウェア的には、メッシュ構造による高速ノード間通信 (175 MB/S) が可能になっている。ノードは、一つのコンピュータと考えられ、CPU、メモリそしてディスクが装備されている (図1)。OPUS2000の場合は、8から32のノードがメッシュ構造によって結合している。これらのソフトウェアとハードウェアにより大量データベースの高速処理を実現している。

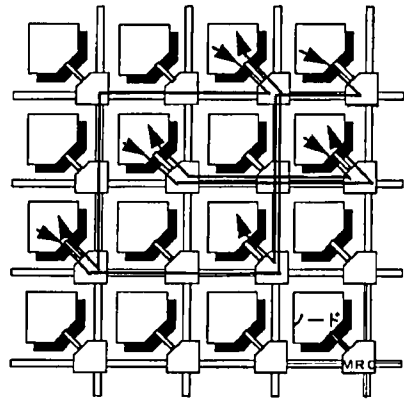


図1 OPUS 2000 概要

3. Open Parallel MAPPERの概要

次にOpen Parallel MAPPERの概要を理解して貰うために特徴と機能を説明する。

3.1 Open Parallel MAPPERの特徴

1) シングル・システム・イメージ

Open Parallel MAPPERは、OPUS 2000の各ノードで稼働し、それぞれのMAPPERはメッシュ機構により接続されている。このように複数のMAPPERが稼働するにもかかわらず、ユーザまたは管理者の眼からは、単一のMAPPERとしか見えない。このことは、使用はもちろん、運用上も一つのMAPPERを想定すれば良く、煩わしい運用管理が不要になる。

2) 並列機唯一の4GL

Open Parallel MAPPERは、PCから汎用機そして並列機までをカバーする唯一の4 GLソフトウェアである。Open Parallel MAPPERをOPUS 2000に導入する事により、ユーザは直ちに並列機を自由自在に使いこなす事が可能になる。

3) Open MAPPERとの完全互換性

Open Parallel MAPPERは、Open MAPPER 5R2と機能的に完全に互換性がある。5R2のユーザは、まったく違和感無くMAPPERを使用する事ができる。またOpen MAPPERとの接続も、LANを経由してシームレスに行える。

4) OPUS2000アーキテクチャとの融合

Open Parallel MAPPERの特徴で特に強調したいものは、OPUS 2000の並列・アーキテクチャとMAPPERのアーキテクチャの融合である。つまりOPUS 2000のハードウェアの特徴を有効利用するようにMAPPERが作られている。この点は、機能のところで詳しく説明するが、一言で言うと、大量データの高速処理を可能にする工夫が施されている。

5) 柔軟なシステム拡張

Open Parallel MAPPERでは、データ量の増加やシステム規模拡大に伴いMAPPERを容易に拡張できる。つまり、OPUS 2000のノードを増やし、それにMAPPERを載せれば、効率の劣化を防ぐ事ができる。最初は、小さいシステムからはじめ規模の拡張にしたがいシステムを拡張すれば、無駄の無い経済的なシステムの構築が可能になる。

6) Visual Basicクライアントとのインタフェース

Open Parallel MAPPERは、Visual Basicで作られたクライアントとのインタフェースを簡単に取るためのライブラリを用意している。これを利用する事により、GUIの充実したアプリケーションの作成ができる。

3.2 Open Parallel MAPPERの機能

機能は大きく四つに分ける事ができる。1) アウトボーデッド化された機能、2) 並列機能、3) 並列機能を支援するための機能、そして4) 他のプラットフォームとの柔軟な接続機能である。

1) アウトボーデッド化された機能

Open Parallel MAPPERは、OPUS S2000の複数のノードの上で動き、各ノードにはCPU、メモリ、ディスクが装着されていることは既に述べた。Open Parallel MAPPERでは、MAPPERのデータベースも各ノードのディスクに分散配置する。アウトボーデッド化された機能とは、データの配置されているノードで実行されるように作られたMAPPERの機能のことである。たとえば、顧客データがノード1のディスクにあったとき、顧客データを処理するランをノード5で実行したとする。アウトボーデッド化されていないときには、ノード1のディスクにあるデータをメッシュ構造を利用してノード5に送りノード5で処理を行う事になる。確かにメッシュ構造は高速ではあるが、大量のデータを転送する時のオーバーヘッドは無視できない。Open Parallel MAPPERではほとんどの機能（ソート、ファインドなど）をアウトボーデッド化し、どのノードでランが実行されてもラン命令はデータのあるノード（この例では、ノード1）で実行される。そして、処理結果をランを実行したノードに返す。このことにより、データ操作の際に生じるメッシュでのデータ転送の負荷を軽減し、効率の向上に寄与している。また、データをアプリケーションにあわせて配置することにより、システム全体で均一にMAPPERを実行し、スループットを上げられるようになっている。

2) 並列機能

並列機能には、並列・サーチ、バ

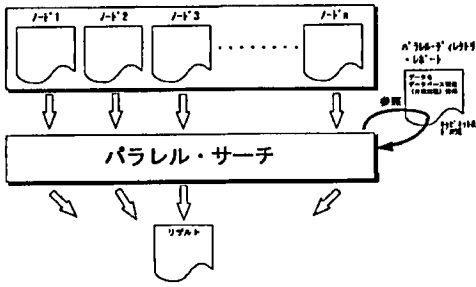


図2 パラレル・リサーチ

ラレル・ロケート、パラレル・バックグラウンド・ランがある。

① パラレル・サーチ

パラレル・サーチは、複数のノードに分散配置されている MAPPER データのサーチを並列に処理し、サーチ時間を飛躍的に短縮することを実現している。また、あらかじめサーチするフィールドのインデックスを作りそれを利用する事により、さらに効率を向上する事ができる。図2が、パラレル・サーチの概念図である。たとえば、150万件のデータがあったと仮定する。このデータを10万件ずつ15に分割し、それを15のノードが持っているディスクに分散配置する。この配置状況をパラレル・ディレクトリ・レポートとよばれるレポートに登録しサーチを行うと、MAPPERは、15のノードで同時にサーチを行い、その結果をリザルトとしてまとめる。したがってサーチの効率は、約15分の1になる。つまり、効率的には150万件で無く、10万件サーチしたのと同じ結果になる。ノードを多くすればするほど高速のサーチが可能になる。一般的なアプリケーションの場合、多く使われるラン命令は、サーチ、計算(トータライズ)、ソート、マッチである。しかもサーチの場合、その処理時間の割合も多く、サーチの効率を上げる事がアプリケーション自身の実行効率をあげる事につながる。したがって、パラレル・サーチはアプリケーションにとって非常に有効な機能である。

② パラレル・ロケート

パラレル・ロケートは、指定された

MAPPER データで文字列の全件検索を行う機能である。パラレル・ロケートもパラレル・サーチと同様にデータを複数のノードに分散配置することにより、パラレル処理が行われ、効率を飛躍的に向上させる事ができる。この機能は、膨大な文書データの中からあるキーワードを見つけだすのに有効な機能である。

③ パラレル・バックグラウンド・ラン

パラレル・バックグラウンド・ランは、実行するノードを指定してバックグラウンド・ランを起動する機能である。一つのアプリケーションから複数のバックグラウンド・ランを起動する事により、アプリケーションをパラレルで処理する事ができる。この機能により、必ずしも順番に処理しなくても良い処理は、複数のノードでパラレルに処理し、アプリケーションの実行時の効率を向上させる事ができる。

④ パラレル運用管理機能

パラレル運用管理機能には、PURGE、SYSTEM、Verify、LOGL、LOGS、MAPSTARTがある。これらの機能は各ノードごとに並行して処理される。たとえばPURGEの場合、各ノードにバージの装置があれば、並行でバージ処理が行われ、MAPPER データベースのバックアップを高速で採取する事ができる。

3) パラレル支援機能

パラレル支援機能には、パラレル・ディレクトリ・レポート、レポート・キャッシュ、レポートの複製がある。

① パラレル・ディレクトリ・レポート

パラレル・ディレクトリ・レポートは、パラレル・サーチで使用されるレポートで、MAPPER データの分散状態が登録されている。MAPPERは、サーチ時にこのレポートを参照し、データの分散状況、たとえば、何ノードに分散されていてどこのノードにデータがあるのかなど、を認識しパラレル・サーチを行う。

② レポート・キャッシュ

レポート・キャッシュはキャッシュ・コントロール・レポートに登録されているレポートをキャッシュ化する。これによりレ

ポートはメモリ上に常駐し、ディスクへのIOが極端に少なくなり処理効率向上に寄与する。

③ レポートの複製

レポート複製は、MAPPERのレポートの複製を複数のノード上に作る機能である。この機能を利用して、MAPPERのラン・コントロール・レポート（ランのコーディングがあるレポート）の複製を複数のノードに置くと、ランを実行する時に@x xのようなランのコーディング・イメージを他のノードに転送する必要が無いため、効率が向上する。また、一つのノードで障害が発生しても、システムを中断せずに他のノードで処理を実行でき、致命的なトラブルを防ぐ事が可能である。

- 4) 他のプラットフォームとの柔軟な接続機能
Open Parallel MAPPERは、TCP/IPで当社の他のMAPPER、2200シリーズMAPPER、

AシリーズMAPPER、UNIX MAPPERそしてMFW (Mapper for Windows)と接続することができる。これらのネットワークを介して、既にメインフレームに蓄えられているデータを活用したり、MFWやVisual Basicを使って、クライアント/サーバ型のアプリケーションを簡単に構築する事ができる。

4. おわりに

日本ユニシスは、今回Open Parallel MAPPERを並列機市場に投入した。既に説明したように、MAPPERは並列機の分野には未だないプロダクトである。MAPPERを導入する事により直ちにエンドユーザも並列機を自在に使用する事ができるようになる。また、Open MAPPER自身も今後、NT対応やWindows 95対応が計画されている。Open Parallel MAPPERとこれらの組み合わせにより、MAPPERはさらに進化を継続し強力なツールになる。

技報編集事務局からのお知らせ

平素はユニシス「技報」をご愛読いただきありがとうございます。ユニシス「技報」では、平成8年3月より日本ユニシスのホームページ (<http://www.unisys.co.jp>) にバックナンバーのご案内をいたします。巻頭言および論文の要約がご覧になれます。

是非、本誌同様ご利用頂ければ幸いです。

お問い合わせ/ご意見を「gihou-info@unisys.co.jp」にてお待ちしております。

安全性確保が重要な課題となっている。日本ユニシス信金バックアップセンタは、信金業界のシステムの安全性を低コストで保証する事を目的として設立された。関俊夫の信金バックアップセンタの紹介は、お客様がバックアップセンタをより理解し有効に利用して頂けるよう、設立の背景・目的、業務内容、今後の展望について紹介している。

昨今の消費動向の変化から、価格破壊が流通業界のキーワードとなっており、低価格な商品の提供が、小売業にとって生き残りの条件になっている。物流の効率化が経営上の重要課題として取り上げられて久しいが、消費者の低価格指向に 대응するため、企業単独の物流改善のみならず、川上から川下までの連携した物流改善が模索されている。平野静夫は食品物流センタにおけるIND・IEの適用の中で、特定のスーパーマーケットチェーンとの連携による日本型ECRの実現を目指した食品物流センタのシステム化事例を紹介している。

コンピュータなど直流を電源として動作する半導体素子を組み込んだ電子機器において交流100/200Vの商用電源を半導体素子が必要とする低電圧の直流に交換する電源部が(電源)高調波を発生する。この高調波が電子機器から配電線を逆に伝播して問題を起こす。蛭本邦夫はコンピュータの電源高調波問題と対策の中で、高調波の発生する仕組み、経験した問題、測定方法、対策、規制などについて具体例を交えて述べている。

8003グラフィックサブシステムの技術的課題はポリゴンアンチ・エリアシングと透過描画をシングルパスで実現し、さらにシェーディング、テクスチャ・マッピングと統合し高速なハードウェア機構とすることであった。これは世界で最初の試みである。渡辺啓は8003の高速・高品位レンダリング機能の中で、採用されたアルゴリズムとASICによる実装のあらましを紹介し、解決されたレンダリング上の問題点を明らかにしている。

▶ 技報編集委員会

委員長 吉村賢一
副委員長 小林 允
委員 青柳幸久, 佐々木健夫, 原 潔
古村哲也, 松倉 司, 松木宏子
佐口 功, 馬場正存, 長島 毅
増山義三, 平山道彦, 萩田勝正

▶ 編集制作担当

インフォメーションサービス
事業推進部 標準企画室
駒崎洋介, 丹野敬子
総合マーケティング部
熊谷 貴

● Editorial Board

K. Yoshimura (Chairman)
M. Kobayashi (Vice Chairman)
Y. Aoyagi, T. Sasaki, K. Hara
T. Komura, T. Matsukura, H. Matsuki
I. Saguchi, M. Baba, T. Nagashima
Y. Masuyama, M. Hirayama, K. Hagita

● Editorial Staff

Y. Komazaki, K. Tanno
(Information Services Planning & Support)
T. Kumagai
(Corporate Planning & Marketing)

ISSN 0914-9996

技 報

UNISYS TECHNOLOGY REVIEW

Vol. 15 No. 4 (No. 48)

発行日 平成8年2月29日
編集発行人 吉村 賢一
発行所 日本ユニシス株式会社
東京都江東区豊洲1-1-1 〒135
TEL(03)5546-4111 (大代表)
印刷所 三美印刷株式会社

禁無断複製転載

© Nihon Unisys, Ltd. 1996

UNISYS

日本ユニシス株式会社
本社 東京都江東区豊洲 1-1-1 千 135 電話 03-5546-4111 (大代表)

ビジネスを革命するサーバが、やってくる。



ETERNAL STREAM
from

ITASCA

類をみない大規模処理能力。将来を見つめ、拡充するオープン機能。傑出したトータル・コスト・パフォーマンス。可能性を広げる、情報系・大量データベース検索能力。それが、メインフレームの新しい流れ「イタスカ」。

既存のシステム資産を継承しつつ、エンタープライズ・クライアント/サーバ・システムを実現するサービスとプロダクト「ClearPath」を提供。超大型機の性能をわずか2個のCMOSプロセッサ・チップで実現。超大型機8台分の能力をシングル・キャビネットに実装。実績のある2200/XPAを継承・発展させた最新の並列処理アーキテクチャを採用。拡張性と365日/24時間無停止処理を実現。

**新世代オープン・エンタープライズ・サーバ
ITASCA3800シリーズ、いよいよ誕生。**

ユニシスの主力機種の開発・製造部門は米国ミネソタ州にあります。「ITASCA」とは、そのミネソタ州を貫流する米国最大の河川ミシシッピの支流である湖の名前です。