

## 開放的分散オペレーティングシステム XERO の設計†

加藤和彦<sup>††</sup> 猪原茂和<sup>††</sup> 成田篤信<sup>††</sup>  
千葉 滋<sup>††</sup> 益田隆司<sup>††</sup>

現在広く普及しつつある分散計算環境においては、単一のハードウェアアーキテクチャの計算機群のみでシステムを構築することはむしろまれであり、さまざまなアーキテクチャをもつ各計算機の特徴を生かし、それらを有機的に結合することが求められている。次の4つの分散性管理機能を備えた分散計算環境を開放的な分散計算環境と呼ぶ。

1. **地理的分散性.** 異なる地点に置かれたサイト上のプログラム間で情報の交換と共有を可能とする機能。これまでの分散オペレーティング・システム研究において言われていた**分散透明性**がこれにあたる。プログラム間の通信により実現される。
2. **ハードウェア・アーキテクチャ的分散性.** 異なるハードウェア・アーキテクチャをもつ計算機の間で情報の交換と共有を可能とする機能。ハードウェア・アーキテクチャ間のデータ表現の変換（例えば、バイト順序の変換や浮動小数点フォーマットの変換等）により実現される。
3. **ソフトウェア・アーキテクチャ的分散性.** 異なるプログラム開発環境を用いて開発されたプログラム間で情報の交換と共有を可能とする機能。ソフトウェア・アーキテクチャ間でのデータ表現の変換（例えば、Pascal のレコード型、C の Struct 型、Lisp の S 式の間での変換）により実現される。
4. **時間的分散性.** 相異なる時点のプログラム間で情報の交換と共有を可能とする機能。情報を永

続的に管理すること（すなわち、システムの動作中/停止中にかかわらずに情報を保持すること）により実現される。

XERO はこのような開放的な分散計算環境を実現することを目的として設計と実現が進められている分散オペレーティングシステムである。そこでの根幹的な研究課題は上述の4つの分散性による「壁」を乗り越える統一的な枠組を、いかにして提供できるかにある。

本論文では、開放的な分散計算環境の実現を目的として開発を進めている分散オペレーティングシステム XERO のプログラミングモデルと、それを実現するための内部機構について述べる。XERO のプログラミングモデルでは、計算は、タスク、スレッド、コンテキスト、そしてコンテキスト間の**遠隔手続き呼び出し**という概念で抽象化されている。タスクは仮想アドレス空間、スレッドは仮想 CPU、そしてコンテキストはメモリオブジェクトの抽象化である。このモデルでは、仮想記憶空間上のコンテキストも永続記憶空間上のコンテキストも統一的に扱われており、さらに、コンテキストのタスク間の同期 (synchronization) と移動 (migration) という概念までもが統一的に扱われている。このプログラミングモデルを実現するための内部構造として、マルチコンテキスト、マルチスレッドを効率よく実現する方法も述べる。

また本論文では、開放的な分散計算環境の中で永続オブジェクトの管理を行うために設計した、**複合オブジェクト**の概念に基づいたファイルシステムについても述べる。前述の永続記憶空間上のコンテキストは、複合オブジェクトの1つとして位置付けられている。XERO の複合オブジェクト管理機構では、著者らが既に提案している永続キャッシング技法を用いることにより複合オブジェクトの効率的な管理を行う。

† Design of the XERO Open Distributed Operating System by KAZUHIKO KATO, SHIGEKAZU INOHARA, ATSUNOBU NARITA, SHIGERU CHIBA and TAKASHI MASUDA (Department of Information Science, Faculty of Science, University of Tokyo).

*J. of Info. Processing*, Vol. 14, No. 4 (1991), pp. 384-397  
†† 東京大学理学部情報科学科

&lt;論 文&gt;

## クライアント・サーバ構成を指向したオペレーティング・システムの設計†

田 胡 和 哉\*\* 根 岸 康\*\* 干 場 美 佳 子\*\*

複数の計算機を通信回線を用いて結合したサーバを、クラスタ・サーバとよぶ。クラスタ・サーバを対象とした、効率のよいオペレーティング・システムを実現する基本方式について提案する。クラスタ構成では、既存の計算機を結合することにより、低い価格で、広い範囲にわたる性能を実現することができる。また、光通信技術の発展により、高い転送効率を持つ通信回線が、安価に実現できるようになっている。LAN を用いて結合されたクライアント・サーバ・システム用のクラスタ・サーバを実現することを考える。これは、ファイル管理、高いスループットを持つジョブ処理、および、数値計算等の並列処理を実現する。これらの実現に必要なシステム機能である、ファイル・システム、分散共有メモリ、および、プロセス間通信機構は、クラスタのノード間でページ、および、制御情報を転送する機構を利用して実現される。その性能について検討する。

分散処理システムにおいて、ノード間でページを転送する機構として、NFS 等の、分散ファイル・システムが用いられている。しかしながら、それらは、種々の通信媒体、および、プロトコルに対応できるように実現されている一方、オーバーヘッドが大きくなりやすく、クラスタ内部の通信機構として十分なものとは言い難い。実際に、NFS のプログラムを検討してみると、通信制御のためのプログラム・ステップが大きく、また、通信パケットを構築、分解するために、ソフトウェアによって、転送すべきデータ量の 4 倍の量のデータがコピーされていることが判明した。

クラスタ内部の通信では、アクセス権のチェックや

データ表現の変換を行う必要がない。また、単一のプロトコルを用いることができる。この性質を利用するとともに、ハードウェアによる補助機構である Pion と、システム核内の転送機構 CPI を新たに設計することにより、転送効率の改善を試みた。送信者となるソフトウェアは、転送すべき領域の番地、および、サイズを Pion に与える。Pion は、複数の領域から、ノード間のデータ転送ハードウェアが直接利用することができる、ひとまとまりのデータ・ストリームを作成する。受け手の Pion は、受信のためのデータ領域を保持しており、受信したデータストリームに自動的に領域を割り当てる。割り当てる領域の大きさは、送信者が指定することができる。CPI は、ファイル・システム等の上位階層からの要求により、ノード間で論理ページを転送する。要求側の CPI は、ページ要求のための制御ブロックを、ページが存在するノード上の CPI に転送する。要求を受けた CPI は、割り込みハンドラのレベルで、ページ・プールを探索し、ページを返送する。ページを要求した側の Pion は、ページ枠を割り当て、上位階層に返答する。これにより、ソフトウェアは、パケットを構築せずにノード間でデータを転送することができる。

現在、IBM PS/55 ワークステーションを用いてシステムの実現を行っている。基礎となるオペレーティング・システムとして、OSF/1 システムを用いる。これまでに、Pion 機能をエミュレートする機能、CPI、および、それらを用いたファイル・システムのコーディングが完了した。作成したプログラムのステップ数を計算することによりオーバーヘッドの推定を行ったところ、NFS に比べて、ページ転送に要するコストが 80% 程度削減できることが判明した。ファイル・システムの実現方式、および、性能に関する検討について述べる。

† A Proposal for an Operating System Designed for Cluster Servers by KAZUYA TAGO, YASUSHI NEGISHI and MIKAKO HOSHIBA (Tokyo Research Laboratory, IBM).

*J. of Info. Processing*, Vol. 14, No. 4 (1991), pp. 398-404

\*\* 日本アイ・ピー・エム(株)東京基礎研究所

## 知的分散 OS†

—放送型オブジェクト指向分散 OS—

関 俊 文\*\* 長谷川 哲夫\*\*  
 岡 宅 泰 邦\*\* 田 村 信 介\*\*

技術革新に伴ってユーザーニーズが多様化し、システムには機能だけでなく拡張性・信頼性・保守性・操作性などの向上が求められている。知的分散システムはこれら要求を実現すべく集中管理部を完全に排した、宣言型の分散計算機システムのアーキテクチャである。つまり、システム全体は各々独立に宣言されたシステム要素の単なる集合であり、システムに投入されるジョブはそれら要素群の協力と協調によって処理される。したがって、集中ボトルネックがなくなり、高い適応性・拡張性・信頼性・保守性が実現できる。知的分散 OS は、この知的分散システムを実現するためのオペレーティングシステムである。

既に多くの分散オペレーティングシステム (OS) が研究、開発されており、これらの分散 OS では個々のワークステーションに高い自律性を持たせると共に、システム全体を単一の複合計算機として機能させるようにしている。しかし、従来の分散 OS では各システム要素への機能や負荷の割当等、システム全体に渡る管理は何らかの要素が集中管理しているので、個々の要素の独立性が損なわれ宣言型システムの構築は難しかった。V-kernel, Sprite, Mach, Amoeba 等でも OS の主要機能を核の部分からアプリケーションレベルに移し機能の分散化を図っているが、OS の個々の機能は特定の要素が担当しており、システムワ

イドな管理を要素群の動的な結合によって行っているわけではない。したがって、集中部の故障が全体の停止を招いたり、負荷の増大が集中管理機構を複雑、大規模にすることになる。

知的分散 OS の特徴は、システム要素への機能や負荷の割当て、あるいはシステム要素間の同期や排他機構を各システム要素に分散させることによって集中機構を排し、システム要素の完全な独立性を実現したことである。アプリケーションだけでなく、システムプログラムのモジュールもどこに存在しても、また何重に配置しても全く同じ動作をする。したがって、管理負荷の特定計算機 (サイト) への集中がなくなり、処理速度や信頼性の問題が解決される。また OS も含めたプログラムの変更や追加、サイト間移動、さらにサイトの追加、削除が容易になる。プログラムやサイトの変更や追加要求には、稼働中の非変更部分を止めることなく対応することができる。

個々のプログラムの独立性を実現するため、知的分散 OS ではオブジェクトモデルと放送通信を採用している。すなわち、アプリケーションやシステムプログラムを、オブジェクトモデルにおけるオブジェクトとして表現し、OS の諸機能は複数サイトに分散したオブジェクト群の放送通信による動的な結合によって実現する。

本論文では知的分散 OS が、オブジェクトの位置・存在・多重度からの独立 (透過) 性をどのように実現しているかを、オブジェクト構造、オブジェクト間通信機構、およびシステム高信頼化機構の面から述べる。さらに知的分散 OS を、複数の計算機を多重化されたローカルエリアネットワークで接続した環境上に実装し、測定した性能結果についても述べる。

† An Operating System for the Intellectual Distributed Processing System —An Object Oriented Approach Based on Broadcast Communication— by TOSHIBUMI SEKI, TETSUO HASEGAWA, YASUKUNI OKATAKU and SHINSUKE TAMURA (Systems & Software Engineering Laboratory, TOSHIBA Corporation).

*J. of Info. Processing*, Vol. 14, No. 4 (1991), pp. 405-413

\*\* (株)東芝システム・ソフトウェア技術研究所

## A-NET 並列オブジェクト指向計算機の オペレーティングシステム†

吉 永 努<sup>††</sup> 馬 場 敬 信<sup>††</sup>

A-NET 計算機のオペレーティングシステム (OS) は、並列オブジェクト指向言語 A-NETL の効率的な実行を支援することを目的として設計、試作された。その設計方針は、次の5点に要約される。

- メッセージを受信してからできるだけ高速にユーザメソッドを起動する。
- コンテキストチェンジのオーバーヘッドを軽減する。
- 並列オブジェクト指向実行モデルを良く反映したコンテキスト管理を行う。
- メッセージ受信履歴をとるなどのデバッグ支援機能を提供する。
- 各ノードプロセッサ上に割り付けられたオブジェクトのみを管理し、グローバルな情報は持たない。

各ノードプロセッサ上に同一のコードを持つ局所的な OS が置かれることから、ローカル OS と呼ぶ。ローカル OS の主な機能は、自ノード宛メッセージの受理、コンテキスト管理、およびオブジェクト管理などである。

メッセージ受理処理は大きく2つに分けられる。すなわち、メッセージセレクタで示されるメソッドを起動するためのものと返答のリターンである。メソッド起動の高速化のための工夫は次の4点である。

- メッセージの宛先となるオブジェクト ID に、オブジェクトポインタを登録した表のオフセットを持たせ、オブジェクトの存在確認を容易にする。
- 静的なオブジェクトについては、メッセージ辞書検索を行わなくてもいいよう、セレクタをアドレ

ス化する。

- メッセージ引数のメッセージ内での並べ方は、メッセージサイズよりもむしろ、受信時の処理軽減を考慮した構造とする。
- メソッドの実行は、専用のレシーブ命令によってコンテキストを生成せずに開始する。  
コンテキストは、受理したメッセージが他のメッセージとの待ち合わせを必要とする時、または、メソッドの実行が返答待ちで進めなくなった時 (future trap) に生成される。コンテキストチェンジのオーバーヘッド軽減のため、次のようにしている。
- 各データのタグ内に同期を表すためのフラグを設け、そのアクセスチェックをハードウェアサポートする。
- 2セットの特殊レジスタを用いて、メッセージ割り込み時のユーザの実行イメージの退避/復帰回数を減らす。
- 実行イメージの退避と復帰にブロック転送を用いるため、コンテキストと特殊レジスタ、メッセージの構造を対応させている。

記憶管理に関しては、ファームウェアによりノード内で世代別ガベージコレクションが行われる。実行時に使用される一時的なコンテキストやリストなどのデータは、コピー型の GC を行う2分割したヒープ領域を使用し、ユーザ定義オブジェクトと寿命の長いデータセルは、コピーによって移動されない領域に割り付ける。OS は、表を用いてユーザ定義オブジェクトの割り付けと動的消去を管理する。また、メソッドの実行を行う要素プロセッサとメッセージ通信を行うルータとの共有メモリ上に置かれた入出力メッセージキューなどの管理を行う。

ローカル OS は、A-NETL を用いて単一のオブジェクトとして記述されており、メソッド総数 63、約 16.5 KB である。シミュレータ上でいくつかのサ

† A Local Operating System for the A-NET Parallel Object-Oriented Computer by TSUTOMU YOSHINAGA and TAKANOBU BABA (Department of Information Science, Faculty of Engineering, Utsunomiya University).  
*J. of Info. Processing*, Vol. 14, No. 4 (1991), pp. 414-422  
†† 宇都宮大学工学部情報工学科

ンプルプログラムを実行した実験結果から、

- ユーザとシステムの実行時間の比は 1:10 である。
- 最大並列度は、使用ノード数に近い値が得られる。

- アドレス化セクタにより平均 2.9 回の辞書検索が不要になる。
- ノード当りのメモリ容量は十分である。などがわかった。

## 欧文誌掲載論文要約

## 並列・分散オペレーティングシステム特集

<論 文>

### 可変構造型並列計算機のメッセージ・プール指向型 並列オペレーティング・システム†

—複数スレッドの並列生成—

恒 富 邦 彦<sup>††</sup>      福 田 晃<sup>†††</sup>  
村 上 和 彰<sup>‡</sup>      富 田 眞 治<sup>‡‡</sup>

われわれが現在開発中の「可変構造型並列計算機」は、128 台のプロセッシング・エレメントをクロスバ網で結合したマルチプロセッサ・システムである。本システムではメモリ・アーキテクチャにダイナミック・アーキテクチャを採用している。これにより、システムは、密結合型/疎結合型マルチプロセッサ、またはそのハイブリッド形態を実現できる。

本システム上のオペレーティング・システム (OS) として、現在、密結合型マルチプロセッサ用のものを

開発中である。本 OS は、OS 内の種々の並列性を引出し、OS の高速処理の達成を目的としている。これを達成するため、本 OS はメッセージ・プール機構をベースとする。並列処理の典型例として、複数スレッドの並列生成がある。本論文では、並列生成方式について論じる。並列生成の性能は、並列生成自体の効果と、それによって生じるオーバーヘッドのトレードオフで決まる。まず、最も素直な実現方式である並列簡易方式を取り上げる。この方式は、複数スレッド生成・システム・コールを受け取ったカーネルがその生成要求をメッセージ・プールに登録し、アイドル・プロセッサが並列生成を実行する方式である。このとき、各プロセッサはメッセージ・アクセス当り 1 つのスレッドを生成する。シミュレーション結果により、共有資源へのアクセス競合によって、並列簡易方式は性能が出ないことを示す。さらに、これを改善する方式として、並列テンプレート方式、チャンク方式、併用方式の 3 つを提案し、シミュレーションによって評価する。その結果、チャンク方式が適切であることが判明した。

† A Message-Pool-Based Parallel Operating System for the Kyushu University Reconfigurable Parallel Processor—Parallel Creation of Multiple Threads— by KUNIHICO TSUNEDOMI (Hitachi Research Laboratory, Hitachi Ltd.), AKIRA FUKUDA (Department of Computer Science and Communication Engineering, Faculty of Engineering, Kyushu University), KAZUAKI MURAKAMI (School of Engineering Sciences, Kyushu University) and SHINJI TOMITA (Department of Information Science, Faculty of Engineering, Kyoto University).

*J. of Info. Processing*, Vol. 14, No. 4 (1991), pp. 423-432

†† (株) 日立製作所日立研究所

††† 九州大学情報処理教育センター

‡ 九州大学情報システム学専攻

‡‡ 京都大学工学部情報工学教室

&lt;論 文&gt;

## 分散型コンピュータシステムにおける静的最適負荷分散方式の パラメータ解析について†

張 勇 兵<sup>††</sup> 亀 田 壽 夫<sup>††</sup> 清 水 謙 多 郎<sup>††</sup>

分散型コンピュータシステムにおける静的最適負荷分散問題において、全体最適化方式<sup>1)</sup>と個別最適化方式<sup>2)</sup>が考えられる。この二つの負荷分散方式はそれぞれシステム全体と各ユーザにとって最適となる負荷分散を行うもので、全く異なる性質を持つ。本論文では、まず、個別最適化方式の解が満足する条件を新たに示し、その解の存在と唯一性を示した。次に、システムパラメータの変化の、両方式におけるシステム性能指標に対する影響を調べた。

全体最適化方式は、システム全体の平均応答時間を最小とするように各ノードに到着したジョブを分散する方式である。個別最適化方式は、各ジョブに対して、そのジョブ以外のすべてのジョブの処理ノードが決まったという条件の下で、そのジョブの応答時間の期待値が最小になるようなノードをそのジョブの処理ノードとするように、負荷分散を行う方式である。

全体最適化方式と個別最適化方式は、異なる方針で負荷分散を行うにもかかわらず、それらの解の満たすべき条件は類似している形をしている。それによって、個別最適化方式の解を全体最適化方式の解を求めると同じ方法で求めることができた。

本論文では、パラメータ解析、すなわち、システムパラメータの変化によってシステム性能指標がどう変わるかを調べることによって、両方式の性質を明らかにした。システムパラメータとして通信回線の通信所要時間、各ノードの処理能力、各ノードの外部到着率を考えた。パラメータ解析の結果により、全体最適化方式と個別最適化方式は極めて似た性質を持つことを

示した。しかし、下記のような一見異常に見える現象があった。

1) 個別最適化方式の場合、通信回線の通信所要時間が大きくなっても、システム全体の平均応答時間が逆により小さくなる場合がある。

2) 全体最適化方式と個別最適化方式とも、外部到着率すなわち負荷の総和が増大しても、システム全体の平均応答時間が逆により小さくなる場合がある。

図1は数値実験の一例であり、通信回線の通信所要時間を変化させた場合、全体最適化方式(OOP)と個別最適化方式(IOP)のシステム全体の平均応答時間の様子を示したものである。図の最右端は負荷分散が行われていないときのシステム全体の平均応答時間を

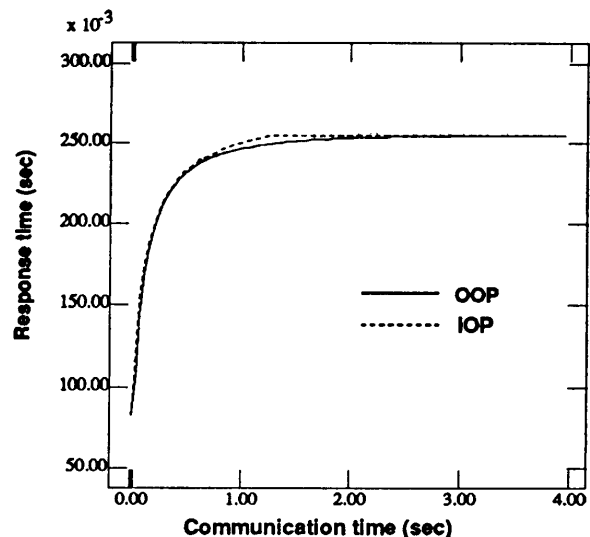


図1 通信回線の通信所要時間を変化させた場合のシステム全体の平均応答時間(全体最適化方式と個別最適化方式)

Fig. 1 The overall mean job response times when changing the communication time under the overall and individually optimal policies.

† Parametric Analysis of Optimal Static Load Balancing in Distributed Computer Systems by YONGBING ZHANG, HISAO KAMEDA and KENTARO SHIMIZU (Department of Computer Science and Information Mathematics, University of Electro-Communications).

*J. of Info. Processing*, Vol. 14, No. 4 (1991), pp. 433-441  
<sup>††</sup> 電気通信大学情報工学科

示している。この図において、個別最適化方式のシステム全体の平均応答時間は全体最適化方式のそれに近いことがわかり、また、負荷分散によってシステムの性能指標が改善されることがわかる。

### 参 考 文 献

- 1) Tantawi, A. N. and Towsley, D.: Optimal Static Load Balancing in Distributed Computer Systems, *J. ACM*, Vol. 32, No. 2, pp. 445-465 (1985).

### 欧文誌掲載論文要約

### 並列・分散オペレーティングシステム特集

<招待論文>

## マイクロカーネルオペレーティングシステム アーキテクチャと Mach†

DAVID L. BLACK    DAVID B. GOLUB    DANIEL P. JULIN  
RICHARD F. RASHID    RICHARD P. DRAVES    RANDALL W. DEAN  
ALESSANDRO FORIN    JOSEPH BARRERA    HIDEYUKI TOKUDA  
GERALD R. MALAN    DAVID BOHMAN

マイクロカーネルに基づくモジュラーアーキテクチャは、オペレーティングシステムの設計と実現に適している。マイクロカーネル・アーキテクチャを使用した

プロトタイプ・システムは、商品に期待され要求される機能性および性能を達成しつつある。カーネギーメロン大学、オープンソフトウェアファウンデーション、その他のサイトで、Mach マイクロカーネルを使用した、いくつかのオペレーティングシステム (Unix, MS-DOS など) の実現が進んでいる。本論文では、Mach マイクロカーネルおよび他のオペレーティングシステムの実現を支援するための Mach マイクロカーネルの利用、これらの試みの現状について述べる。

† Microkernel Operating System Architecture and Mach by DAVID L. BLACK (Research Institute, Open Software Foundation), DAVID B. GOLUB, DANIEL P. JULIN (School of Computer Science, Carnegie Mellon University), RICHARD F. RASHID (Microsoft Corporation, One Microsoft Way), RICHARD P. DRAVES, RANDALL W. DEAN, ALESSANDRO FORIN, JOSEPH BARRERA, HIDEYUKI TOKUDA, GERALD R. MALAN (School of Computer Science, Carnegie Mellon University) and DAVID BOHMAN (Next, Inc.).  
*J. of Info. Processing*, Vol. 14, No. 4 (1991), pp. 442-453

## 欧文誌掲載論文要約

<情報処理からの翻訳>

### エディタを部品としたユーザインタフェース構築基盤：鼎†

暦本純一<sup>††</sup> 垂水浩幸<sup>††</sup> 菅井勝<sup>††</sup>  
 山崎剛<sup>†††</sup> 猪狩錦光<sup>††††</sup> 森岳志<sup>††</sup>  
 杉山高弘<sup>††</sup> 内山厚子<sup>††</sup> 秋口忠三<sup>††</sup>

マルチウインドウ環境が一般化し、テキストだけでなく各種の視覚的メディアを扱うアプリケーションへの要求が非常に高まってきている。ところが、これらのアプリケーションでは、ユーザとの対話部分（以下UI部と略す）の実現に高度なプログラミングを必要とし、UI部の開発がネックになってシステムの構築を困難なものにしていた。

われわれが開発中のUI構築システム鼎は、従来のUI構築に利用されていたツールキット的機能に加え、6種のメディアを操作するためのエディタ部品を備えている。これらのエディタ部品をベースにして、鼎が提供する種々のカスタマイズ機構を利用してアプリケーション向けのUIを構築していく。メディアの選定に際しては、われわれが過去に作成してきたCASEツールからの経験を基にした、各エディタ部品の特徴は以下のとおりである：

**テキスト** 文字列（日本語）を編集するための部品。かな漢字変換機能を含んでいる。他メディア中の文字列（たとえば表のセル中の文字列）を編集するためにも用いる。

**図形** 基本図形の組み合わせによる図、表、階層、ネットワーク以外の構造をもつ図や、一般的な作画に

よる図を入力する手段として使う。

**イメージ** スキャンデータなどのビットマップ。

**グラフ構造** ノードとアークの組み合わせによる図式を編集するための部品。ノードの形状は、スキーマファイルの形で、アプリケーションの分野に合わせて定義することができる。モジュール間の接続や、制御フロー、状態遷移図などの入出力インタフェースを構築するために用いる。

**階層構造** 木構造のデータを編集するための部品。木のノードにはタイトル文字列と、自分の下に所属している部分木のリスト、そしてノード自体の情報をあらかず他メディアを格納している。構造化フローチャートや、操作マニュアルのように章立ての明確な文章を作成するために用いる。

**表** マトリックス上に配置された文字列や他メディアを編集するための部品。データベース検索やフォームシートによる入力など、表の形式でユーザに見せるとわかりやすいものに使う。

鼎を利用したCASEツールに関する利用実績を3例（いずれも開発規模数10Kステップ程度）について調査した（表1）。（A）は各ツールで独自に開発した部分、（B）は鼎のライブラリから各ツールにリンクされた部分、（C）、（D）はXのライブラリから各ツールにリンクされた部分である。数値は（A）-（D）の総和

表1 CASEアプリケーションにおける鼎の使用実績

	例1	例2	例3
(A) 新規開発部分	29.8%	18.9%	23.7%
(B) 鼎ライブラリ	32.3%	45.1%	45.2%
(C) X Toolkit	23.6%	19.4%	17.9%
(D) Xlib	14.3%	16.6%	13.3%
(α) $\frac{B}{A+B}$	52%	70%	66%

† Canae: A Platform for Constructing Graphical User Interfaces with Editors by JUNICHI REKIMOTO, HIROYUKI TARUMI, MASARU SUGAI (Software Architecture Department, Software Engineering Development Laboratory, NEC Corporation), GO YAMAZAKI (NEC Microcomputer Technology, Ltd.), KANEMITSU IGARI (NEC Scientific Information System Development Corporation), TAKESHI MORI, TAKAHIRO SUGIYAMA, ATSUKO UCHIYAMA and CHUZO AKIGUCHI (Software Architecture Department, Software Engineering Development Laboratory, NEC Corporation). *J. of Info. Processing*, Vol. 14, No. 4 (1991), pp. 455-462

†† 日本電気(株)ソフトウェア生産技術開発本部基本方式開発部

††† 日本電気マイコンテクノロジー(株)

†††† 日本電気技術情報システム開発(株)



を100%とした場合の、各部分の割合であり、ソースコードの行数をベースにして計算した。 $(\alpha)$ は新規開発部分と鼎の利用部分の合計に対する後者の割合を示している。すなわち、ツール全体の開発規模に対する

鼎担当部分の割合を示している。この数字が大きいほど、鼎利用による工数削減効果が大きいと言える。鼎を利用することにより、開発工数が約半分に削減されていることが分かる。

## 欧文誌掲載論文要約

<情報処理からの翻訳>

### 自由曲面パッチ接続問題の解決†

穂 坂 衛††

この論文は Computer Aided Geometric Design における自由曲面パッチの生成と接続を取扱う。曲面は曲線のテンソル積として表せるものと表せないものがある。前者では  $C^{(n-1)}$  接続をする  $n$  次ベジェ曲線列の生成と  $C^{(2)}$  接続の条件を曲率連続に緩めた曲線列の生成の問題を解決しそれを曲面に及ぼす。次にテンソル積として表せない曲面領域の境界条件を満たす曲面パッチを生成する方法を示す。近年自由曲面の設計や干渉さらに品質評価の問題において、多数の曲面上の点の計算や微分方程式を解くことの必要性が生じた。しかし再帰的に定義される B スプライン関数を直接用いることは複雑のため、閉じた単純な形式のベジェ曲面の接続に落とすことが必要になった。ベジェ曲面であればその幾何学的性質や図的な理解は容易になる。また制御点のインデックスに対するシフト演算子を用いれば表現式やその解析は非常に単純化される。

始めに  $C^{(n-1)}$  接続をするベジェ多辺形列を定める方法を述べる。2つのベジェ多辺形の接続を支配する接続定義多辺形および曲線の大きさに関係せずに0と1との間に規格化されたパラメータ間の関係を定めるスケール比を導入する。相続く接続定義多辺形を生成管理するものとして、さらに多辺形が定義されこれがスプライン多辺形であることを示す。このことはスプライン多辺形の幾何学的意味を明瞭にしているだけでなく、スプライン多辺形から直接ベジェ点の位置を定める公式を作ることや、また高次のスプライン多辺形

でも急速に低次のスプライン多辺形に落とすアルゴリズムを与えるのに役立つ。スプライン多辺形から図的にもベジェ点を決定できる能率の良い方法を作れることを示す。スプライン制御点の挿入とベジェ多辺形の分割が対応することとその利用法も示す。

さらに  $C^{(2)}$  接続を曲率連続だけに弛めた  $G^{(2)}$  接続の場合も、2次の通常のスプライン多辺形の辺上に定まるベジェ多辺形の第2第3の制御点はその辺の上を移動するだけという簡単な関係になることが分り、このことは従来のベータ・スプライン導出より遙かに単純であるにも拘わらずより一般的で応用の範囲が広いこと、直観的にも性質が理解しやすいことを示す。

曲面はテンソル積の手法によってスプライン・制御点ネットから  $C^{(n-1)}$  や  $G^{(2)}$  接続をするベジェ点ネットが得られる。ベジェ制御点ネットの形状は生成される合成曲面の形状をかなりよく反映しているが、前者は生成される形状を想像するのは困難である。しかしそれからベジェ点ネットは規則的手法で定まる。それからの曲面の幾何学的性質は数式的にも、幾何的にも容易に導出される。最後に、テンソル積型の曲面だけでは広範な応用には耐えられないので、従来困難な問題の1つであった「与えられた境界条件を満足する曲面パッチ」を表現する公式を導く。これは周囲とすべて  $C^{(0)}$  接続をする曲面は容易に作れるため、接続条件の合わない分を各辺独立に補正できる補正パッチを表す式を定義する。これを元の曲面に加えることによって高次の接続条件でも満たすことのできる方法を与える。応用例としてテンソル積の面に囲まれる3角形や5角形領域での曲面パッチ生成への利用法を示す。

† New Solutions of Connection Problems of Free-form Curves and Surfaces by MAMORU HOSAKA (Tokyo Denki University). *J. of Info. Processing*, Vol. 14, No. 4 (1991), pp. 463-472  
†† 東京電機大学

## 欧文誌掲載論文要約

&lt;情報処理からの翻訳&gt;

## 代用電荷法に基づく双方向的な数値等角写像の方法†

天 野 要††

等角写像の数値計算法すなわち数値等角写像の方法は、数値解析の課題のひとつとして古くから研究され、近年、再び注目を集めている。それらは、与えられた問題領域から標準領域への写像を求める方法と、逆に標準領域から問題領域への写像を求める方法とに大別される。いずれの場合にも有効な方法はまだ知られていない。本論文では、代用電荷法に基づいて双方向的な数値等角写像の方法を提案する。

まず、 $z$ 平面上の Jordan 曲線  $C$  で囲まれた Jordan 領域  $D$  から  $w$  平面上の単位円内部  $|w| < 1$  への等角写像を考える。その写像関数  $w=f(z)$  は、 $D$  内に原点をとれば、正規化条件  $f(0)=0$ ,  $f'(0)>0$  の下に一意に定まる。代用電荷法に基づいて、2つの共役な調和関数を  $\bar{D}=D+C$  の外部に配置された  $N$  個の電荷点  $\zeta_i$  を極とする複素対数ポテンシャルの実部と虚部の1次結合で近似して、この等角写像の簡潔な近似写像関数

$$W=F(z)=z e^{G(z)+iH(z)}, \quad z \in \bar{D},$$

$$G(z) = -\sum_{i=1}^N Q_i \log |z-\zeta_i|,$$

$$H(z) = -\sum_{i=1}^N Q_i \arg(z-\zeta_i) + \theta_0$$

$$= -\sum_{i=1}^N Q_i \arg(1-z/\zeta_i),$$

$$\theta_0 = \sum_{i=1}^N Q_i \arg(-\zeta_i)$$

を得ることができる。ここに、 $\theta_0$  は正規化条件から定まる回転の定数である。未定係数である電荷  $Q_i$  は境界上の  $N$  個の拘束点  $z_j$  で境界条件  $|F(z)|=1$ ,  $z \in C$  を満たすように定められる。すなわち、拘束条件と呼ばれる連立1次方程式

$$\sum_{i=1}^N Q_i \log |z_j - \zeta_i| = \log |z_j|, \quad z_j \in C,$$

$$j=1, \dots, N$$

の解である。

次に、 $w$  平面上の単位円内部  $|w| < 1$  から  $z$  平面上の Jordan 領域  $D$  への等角写像を考える。その写像関数  $z=f^*(w)$  は、 $D$  内に原点をとれば、正規化条件  $f^*(0)=0$ ,  $f^{*'}(0)>0$  の下に一意に定まる。この等角写像は前述の等角写像の逆写像にはかならない。ここで、 $f^*(w)$  を  $f^{-1}(w)$  ではなく  $F^{-1}(w)$  で近似することを考える。この  $F^{-1}(w)$  は  $F(z)$  とまったく同一のアルゴリズムで、以下のように計算することができる。

$$Z=F^*(w)=w e^{G^*(w)+iH^*(w)}, \quad |w| \leq 1$$

$$G^*(w) = -\sum_{i=1}^{N^*} Q_i^* \log |w-\zeta_i^*|$$

$$H^*(w) = -\sum_{i=1}^{N^*} Q_i^* \arg(w-\zeta_i^*) + \theta_0^*$$

$$= -\sum_{i=1}^{N^*} Q_i^* \arg(1-w/\zeta_i^*)$$

$$\theta_0^* = \sum_{i=1}^{N^*} Q_i^* \arg(-\zeta_i^*)$$

ここに、 $\zeta_i^*$  は  $w$  平面上の電荷点で、電荷  $Q_i^*$  は拘束条件の連立1次方程式

$$\sum_{i=1}^{N^*} Q_i^* \log |w_j - \zeta_i^*| = \log |w_j| - \log |z_j|,$$

$$j=1, 2, \dots, N^*$$

の解である。拘束点は  $w_j=F(z_j)$ ,  $z_j \in C$  を満たす  $N^*$  個の適当な点の集合である。最も簡単には、 $F(z)$  の計算に用いられた  $z$  平面上の  $N$  個の拘束点の像をそのまま採用することができる。このとき、逆等角写像の計算に新たなプログラミングは不要である。

論文では、さらに、境界上の計算値を用いた簡潔な誤差の評価式を与え、偏心円(凸型)、Cassiniの楕形(凹型)、正方形(角点の存在)、勾玉形(非星形)という典型的な領域に対して数値実験を行って、提案された方法の有効性を検証している。

† A Bidirectional Method for Numerical Conformal Mapping Based on the Charge Simulation Method by KANAME AMANO (Department of Computer Science, Faculty of Engineering, Ehime University).

J. of Info. Processing, Vol. 14, No. 4 (1991), pp. 473-482

†† 愛媛大学工学部情報工学科

## 欧文誌掲載論文要約

<情報処理からの翻訳>

### 3次元ベクトル演算の並列実行に関する考察†

成 瀬 正††

自然界が3次元空間であることから、3次元ベクトル、 $3 \times 3$ マトリクス演算をベースとして記述できる問題はきわめて多い。光線追跡法、ロボティクス、プラズマシミュレーションなどはその例である。これらの応用では、計算時間を多く必要とし、高速処理が求められている。本論文では、上記問題の高速処理に寄与することを目的として、3次元ベクトル、 $3 \times 3$ マトリクスから構成される式の並列実行について集合論的観点から論じる。また、それを実現する並列演算機構について論じる。

まず、3次元定数ベクトル、 $3 \times 3$ 定数マトリクスから構成される集合を考え、その上に基本演算を定義する。ここで、基本演算は、ベクトル要素に対する演算、ベクトル-マトリクス演算、マトリクス要素に対する演算、マトリクス-マトリクス演算である。ついで、3次元ベクトル変数を導入し集合を拡張する。この集合を $G$ と書くと、 $G$ は3次元ベクトル、 $3 \times 3$ マトリクスから構成される式の集合である。さらに、 $G$ 上に微分演算子を導入する。そして、集合 $G$ が持つ基本的性質を論じる。主要な性質として、 $G$ が基本演算に関して閉じていること、および微分演算に関して閉じていること、が得られた。後者は、 $G$ の式を微分した結果の式がまた $G$ の式であり、それは基本演算を用

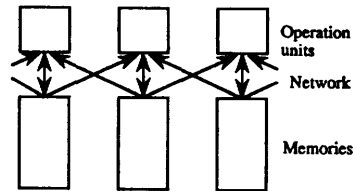


図1 並列演算機構

いて記述できることを意味している。したがって、 $G$ 上の基本演算の並列演算機構を明らかにすることにより、 $G$ の式的高速計算が可能となる。

3個の演算器を用いた場合が演算器の利用効率を最も高められるので、3演算器構成の並列実行機構を考える。具体的な構成は、図1に示すように3個の演算器と3個のメモリをネットワークで結合した構成である。この構成により、メモリアクセスの競合なく基本演算の並列実行ができる。演算並列度を逐次演算器の演算回数と本演算器の演算回数の比として定義すると、 $G$ 上の各基本演算に対して演算並列度3を得る。すなわち、3個の演算器を遊ぶことなく使って並列計算できることがわかる。

ついで、線形代数で用いられる主な演算は、上記基本演算を用いて記述できることを示す。これらの中にはベクトル要素の総和をとるような本質的に逐次演算が含まれる。本質的に逐次演算だけが演算並列度を低下させる要因であり、その部分を除けば $G$ の式は演算並列度3で実行できることを示す。

† Parallel Execution of Three-Dimensional Vector Operations  
by TADASHI NARUSE (NTT Human Interface Laboratories).  
*J. of Info. Processing*, Vol. 14, No. 4 (1991), pp. 483-489  
†† NTT ヒューマンインタフェース研究所

欧文誌掲載論文要約

<情報処理からの翻訳>

ループを用いた大規模分散処理システム†

鶴保 征城†† 木ノ内 康夫††† 星子 隆 幸††††
仲谷 元‡ 宮川 順治††

信頼性の向上, システム処理能力やサービス, 機能の柔軟な拡張, 徹底した運転自動化等をねらいとして, 汎用計算機による分散処理構成を用いた大規模オンラインリアルタイムシステムを開発した.

本システムは図1に示す標準構成をとり, 100メガビット/秒の光ループと48キロビット/秒の同軸ループの二種類のループを介して HOST, FEP, SCP が相互接続される. これら各プロセッサ上には同一のオペレーティングシステムが走行し, 光ループはプロセッサ間のデータ送信のため, 制御ループはシステム内装置(周辺装置を含む)の電源投入切断, 接続切替等のハードウェア制御のために使用する. システムの運転は, SCP 上の管理マネージャがシステム全体の運転を統括し, 個々のプロセッサまたはプロセッサ群に閉じた運転は各 HOST, FEP 上の管理マネージャが分担する階層構成をとる.

本論文では, このシステムで採用した主要な方式技術について, 実現上の問題とその対処法, 導入実績を踏まえた評価について報告する.

本システムで採用した主要な方式とその評価結果は以下のとおりである.

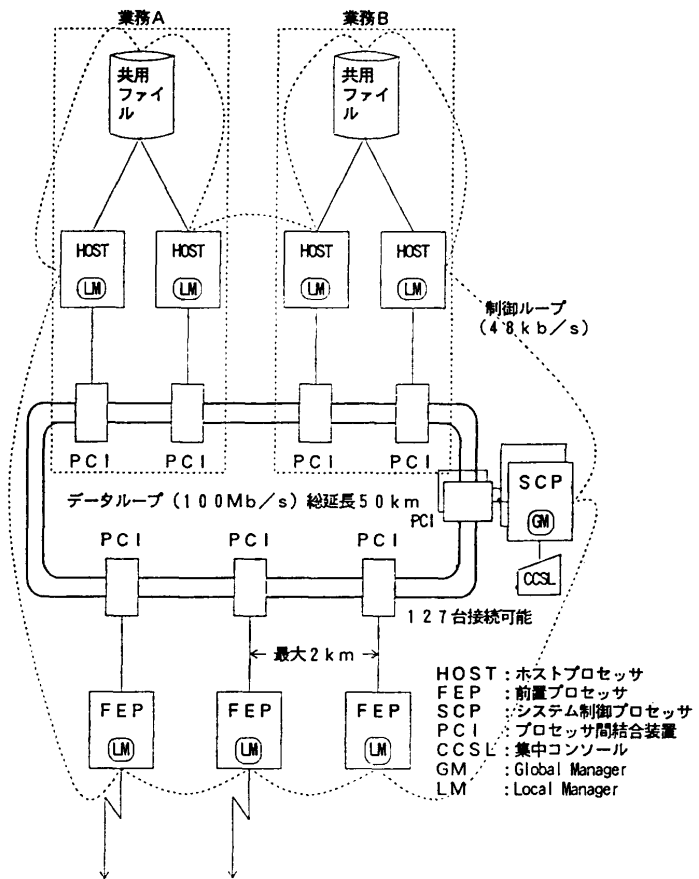


図1 システムの構成概要

† Large-scale Distributed Processing System Using Loop Networks by SEISHIRO TSURUHO (NTT Software Laboratories), YASUO KINOUCHI (NTT Communication Switching Laboratories), TAKAYUKI HOSHIKO (NTT Network Systems Development Center), HAJIME NAKATANI (NTT Network Information Systems Laboratories) and JUNJI MIYAKAWA (NTT Affiliated Companies Department).

J. of Info. Processing, Vol. 14, No. 4 (1991), pp. 490-500

†† NTT ソフトウェア研究所

††† NTT 交換システム研究所

†††† NTT ネットワークシステム開発センター

‡ NTT 情報システム本部

††††† NTT グループ事業推進本部

(1) 高速のプロセッサ間通信方式

①PCI 内に物理的な通信パスを最大 256 個用意し, この物理パスと通信管理プログラムの意識する論理パスとを1対1に対応づける方式とした. これにより, 電文の振り分け, メッセージバッファ間の電文移送等の処理を省略可能とした. さらに, 二つの物理パスを全二重パスとして使用することにより, 送信権制御の簡素化, 通信起動時の割込み処理不要化等を実現した.

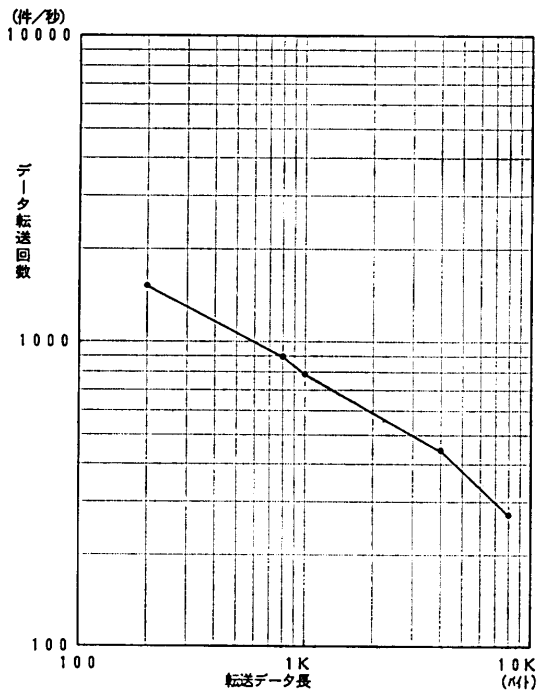


図2 PCI を介したデータ転送回数

②PCI ハードウェアによる 1KB を超える長データの自動分割転送, 1データ転送単位の早期トークン(送信権)解放等により, 沈み込みのない, 高効率ループ伝送制御を実現した。

以上①②により, プロセッサ間のデータ転送制御用ダイナミックステップ数を, 従来方式と比べて約 30% 削減し, 図2に示すデータ転送能力を実現した。

#### (2) システム運転の自動化・省力化方式

①コンソール操作手順を簡易に記述できる言語を開

発し, メッセージや時刻を契機とするコマンドの自動投入を可能とした。

②カートリッジテープを用いた集合型大容量記憶装置を開発し, 媒体操作を自動化した。

以上①②により, 通常運転から定型的な異常処理までの運転操作を自動化した。

#### (3) システム高信頼化方式

①HOST, FEP を  $n+1$  のホット予備構成として SCP から常時監視し, 高速の半導体ファイル装置等の導入により, 障害の検出と HOST 障害時の情報引継ぎを高速化して切替え時間を短縮可能とした。

②FEP に OSI レイヤ7相当の機能まで分担させ, HOST 障害時の端末との通信パス保持とその間の受信電文保留を可能とした。

以上①②により, ハードウェア障害だけでなく, ソフトウェアバグ, 誤操作等に起因する HOST 障害に対してもシステム復旧を1分以内に実現し, ほぼすべての端末利用者に HOST 障害を隠蔽できた。

#### (4) サービス無中断増設方式

①PCI を介した物理的な通信パス情報を SCP で一元管理し, これを各プロセッサ立上げ時に SCP から自動配信することとした。

②上記配信用のパスを複数設け, 各パス対応に論理的に閉じた交信系を構成可能とした。

以上①②と光ループ, SCP 等の二重化により, 増設プロセッサと試験用プロセッサを現用系から論理的に切り離し, サービス無中断のプロセッサ増設が可能であることを確認した。

## 欧文誌掲載論文要約

&lt;情報処理学会論文賞受賞論文からの翻訳&gt;

複合パラダイム言語 TAO におけるオブジェクト指向  
プログラミングとその実現†

大里 延 康†† 竹内 郁 雄†††

本論文では、複合プログラミング・パラダイム言語 TAO におけるオブジェクト指向プログラミングとその処理系の実現および評価について報告する。TAO は、人工知能研究の分野において扱う問題の対象領域の拡大や、問題の性質の多様化に対応しうる、柔軟で高機能なプログラミング環境を提供することを目標としている。

TAO の設計思想の重要な点は、プログラマが自己の問題に応じたプログラミング・パラダイムを自由に選択することを許容する、という点である。これを実現するため、TAO は、論理型プログラミング、オブジェクト指向プログラミングそのほかを、基本となる Lisp プログラミングの中に融合している。このために、Lisp の S 式のシンタクスやセマンティクスのもつ単純さが極めて有効にはたらいっている。また、プログラミング・パラダイムの融合は、Lisp の eval の計算メカニズムそのものに手を入れて、言語の核の部分で達成された。

TAO は、専用の記号処理計算機として開発されたハードウェア ELIS 上に実装された。特に、ELIS の性能を背景として、インタプリタの性能を重視する、という思想で設計され、インタプリタ eval の本体、基本関数、基本メカニズムなど、その中心的な部分は

ELIS のマイクロプログラミングによって実現されている。ELIS は、これらを受容しうる大容量の WCS を備えている。高速なインタプリタは、プログラムのテストやデバッグなどの開発サイクルにおいて極めて有効である。

オブジェクト指向は、TAO において重要な柱となるプログラミング・パラダイムのひとつであり、またもっとも多用されているパラダイムでもある。オブジェクト指向の性能は、(i)メッセージ伝達式の解釈部、(ii)メソッドの探索と起動、(iii)メソッド本体の実行部の三つの要因で決まる。TAO では、専用計算機 ELIS のマイクロプログラミングによって、(i)式の解釈部を eval の中に埋め込み、(ii)メソッド探索を二分探索で実現した。その結果、メソッド本体に至るまでの解釈のステップが高速化できた。また、(iii)メソッドの実行自体はインスタンス変数の参照以外、関数の実行とまったく同一であるため、メソッド固有のオーバヘッドはない。インスタンス変数はハッシュ表によって高速化することができた。これらにより、インタプリタにおける TAO のオブジェクト指向機能は、Lisp 部と比較して遜色のないものとなった。

また、オブジェクト指向のプログラムが必要とする種々の内部的な情報の生成を工夫することにより、メモリ使用量やプログラム開発時のターン・アラウンド時間を縮小することができた。これによって、TAO のオブジェクト指向は、実用的な規模のプログラム開発にも十分に対応しうるシステムとなっている。本論文では、実際の応用プログラムの解析を通じて、その実現法の妥当性についても述べる。

† Object-Oriented Programming in Multiple-Paradigm Language TAO and Its Implementation by NOBUYASU OSATO (NTT Human Interface Laboratories) and IKUO TAKEUCHI (NTT Basic Research Laboratories).

*J. of Info. Processing*, Vol. 14, No. 4 (1991), pp. 501-507

†† NTT ヒューマンインタフェース研究所

††† NTT 基礎研究所

## 欧文誌掲載論文要約

<情報処理学会論文賞受賞論文からの翻訳>

### 高速自動微分法と区間解析とを用いた丸め誤差推定†

久保田 光一†† 伊 理 正 夫†††

高速自動微分法の実用化により、理論的には知られていた丸め誤差の評価式を実際に計算できるようになった。しかし、従

来提案されている評価式は厳密な意味での丸め誤差の絶対値の上界を与えるものではないので、その評価式による上界（の近似値）は真の（無限精度で計算したときの）値の存在範囲を保証することには利用できなかった。一方、数値演算の結果の品質を保証するためには、計算の過程で実行する浮動小数点数演算をいわゆる機械区間演算に置き換えて計算すること以外に、実際的な方法はなく、計算の規模が大きくなると、結果として得られる区間の幅が広くなりすぎてしまうことが多かった。本論文では、これらの問題を解決すべく、Yu. V. Matiyasevich の区間解析手法を丸め誤差推定に応用し、丸め誤差の絶対値の厳密な上界を計算する算法を提案した。この上界と丸め誤差を含んだ計算値とから、真の値の存在範囲を保証することができ

表 1 計算の精度を変えたときの保証区間幅の比較

$\epsilon_M$	従来の丸め誤差評価値から推定される有効桁(下線部)	機械区間演算による保証区間	本算法による保証区間
$2^{-13}$	0.7438964843	—	—
$2^{-14}$	0.7542193532	[-3344.4582519, 3346.1162109]	[0.4637820125, 1.0446566939]
$2^{-16}$	0.7542197853	[0.0575954438, 1.4508441332]	[0.7542197555, 0.7542198151]
$2^{-18}$	0.7542197855	[0.7540463030, 0.7543932679]	[0.754219785475757, 0.7542197854840781]

る。しかも、ある仮定の下では、その上界が丸め誤差の発生し得る上限に漸近することを証明した。

機械区間演算をソフトウェアでシミュレートしたところ、計算が単純な場合には、本算法により得られる区間幅と機械区間演算により得られる区間幅とは大差ないが、演算回数が増えると、それらが比にして  $10^8$  にもなる場合があるという数値実験結果も得ている。表 1 には、その一例として、線形方程式系  $Ax=b$  の解  $x$  の第 1 成分  $x_1$  (約 800 回の演算実行結果) について、従来の丸め誤差評価値から推定される計算値の有効桁数、機械区間演算結果、本算法が与える保証区間を示した。演算の精度が高くなる—マシンエプシロン  $\epsilon_M$  が小さくなる（浮動小数点の仮数部長が長くなる）—に従い、本算法が与える保証区間幅が狭くなっていることがわかる。

計算結果を保証するために区間を反復改良するための方法が種々提案されているが、本算法は、それらに対して最も基本的な情報を提供する手段であり、それらの反復改良手法のより複雑な計算への適用を可能にするものである。

† Estimates of Rounding Errors with Fast Automatic Differentiation and Interval Analysis by KOICHI KUBOTA (Department of Administration Engineering, Faculty of Science and Technology, Keio University) and MASAO IRI (Department of Mathematical Engineering and Information Physics, Faculty of Engineering, University of Tokyo).  
*J. of Info. Processing*, Vol. 14, No. 4 (1991), pp. 508-515

†† 慶応義塾大学理工学部管理工学科

††† 東京大学工学部計数工学科

## 欧文誌掲載論文要約

&lt;情報処理学会論文賞受賞論文からの翻訳&gt;

微分幾何学特徴に基づく距離画像分割のための  
ハイブリッド手法†

横 矢 直 和†† マーチン D. レビン†††

3次元物体のモデル化と認識を目的とした距離画像解析の初期段階において最も重要な処理はセグメンテーションである。本論文では、この問題に対して領域およびエッジに基づくハイブリッドな手法を提案している。

物体認識の観点からは特に、観察方向（物体の移動）に不変なシーン記述の抽出が重要であり、ここでは、距離画像のセグメンテーションを、観察方向に不変な微分幾何学特徴が一樣でかつ、距離と法線ベクトルに関する不連続点を含まないような表面領域への分割と定義している。この分割を実現するために、まず最初に初期分割として、ガウス曲率 ( $K$ ) と平均曲率 ( $H$ ) の符号の組合せに基づく、座標系のとり方と観察方向に不変な以下のような画素分類（曲率符号マップ）を行う。

- (1)  $K > 0$  かつ  $H < 0$ …負楕円面 (peak surface)
- (2)  $K > 0$  かつ  $H > 0$ …正楕円面 (pit surface)
- (3)  $K = 0$  かつ  $H < 0$ …負放物面 (ridge surface)
- (4)  $K = 0$  かつ  $H > 0$ …正放物面 (valley surface)
- (5)  $K = 0$  かつ  $H = 0$ …平坦面 (flat surface)

(6)  $K < 0$  かつ  $H = 0$ …極小面 (minimal surface)(7)  $K < 0$  かつ  $H < 0$ …負双曲面

(saddle ridge surface)

(8)  $K < 0$  かつ  $H > 0$ …正双曲面

(saddle valley surface)

同時に、距離の不連続点（ジャンプエッジマップ）と法線ベクトルの不連続点（ルーフエッジマップ）を抽出する。そして最後に、この3種類の初期分割マップを統合することによって最終的な分割を得る。

なお、本アプローチでは、物体の輪郭や稜線に対応した距離や法線の不連続の近傍でも距離画像の1次および2次の偏微分を正確に計算する必要がある。この問題を解決するために、選択的局所曲面当てはめによる新しい距離画像の数値微分法を開発した。

本論文で提案している領域分割法は、距離画像を物体の部分表面に対応した領域に分割するとともに、各領域を上記8つの基本曲面タイプに分類し、領域間のエッジタイプを判定することができる。そして、最終的な分割結果は、領域内の属性を付加した領域隣接グラフとして、モデルに基づく物体認識において利用可能な形で表現される。本手法は多面体と自由曲面物体が混在するような複雑なシーンに対しても有効である。これは人工データとレーザレンジファインダから得られた実データを用いた実験によって確かめられた。

† A Hybrid Approach to Range Image Segmentation Based on Differential Geometry by NAOKAZU YOKOYA (Machine Understanding Division, Electrotechnical Laboratory) and MARTIN D. LEVINE (McGill Research Centre for Intelligent Machines, McGill University).

*J. of Info. Processing*, Vol. 14, No. 4 (1991), pp. 516-524

†† 電子技術総合研究所知能情報部

††† マギル大学知能機械研究センター



## 欧文誌掲載論文要約

<論 文>

# ミュージックサーバシステム†

青柳龍也†† 平田圭二†††

## 1. はじめに

われわれが現在研究している Music Server システムについて述べる。Music Server システムは、分散環境の下での音楽（実時間）アプリケーションのためのプラットフォームである。本システムはサーバ・クライアントモデルを基に設計されており、音楽アプリケーションに適したいくつかの機能を提供している。それらの機能には、音楽イベントの入出力制御、挿入、削除などの音楽イベントの操作、音楽イベントのスケジューリングなどがある。さらに、アプリケーションクライアントは Music Server システムを通して互いに通信することもできる。

Music Server システムの主な問題点は、(1)音楽イベントの実時間管理、すなわち、イベント実行の正確なアルゴリズムとネットワークの遅延やプロセス切り替えのオーバーヘッドの影響をなくす機構を開発すること、(2)Music Server システムが提供すべきイベント操作機能を明確にすることである。それらの機能は簡単でかつ十分なものでなければならない。本論文では、これらの問題点について議論し、われわれの要求を満たすシステムアーキテクチャとアルゴリズムを提案する。

## 2. システムの構成

図1に Music Server システムのアーキテクチャを示す。Music Server システムは、ローカルエリアネットワーク（イーサネット）で結ばれたワークステーション（SUN 3/280）とパーソナルコンピュータ（NEC PC 9801）上に実装されている。Music Server

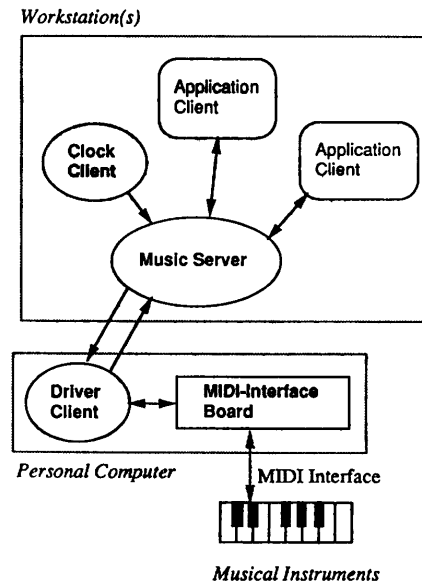


図1 全体構成

システムは複数のプロセスから構成されている。それらのプロセスは、特別なプロトコルでプロセス間通信を行い、同期をとりながら協調して計算を進めていく。

クロッククライアントは、タイマ割り込みを利用した実時間クロックを持つクライアントである。クロッククライアントは、Music Server 内のサーバクロックを進める特別なメッセージを発行する特権を持つ。すなわち、 $N$ ミリ秒ごとにクロッククライアントは Music Server にメッセージを送り、サーバクロックは $N$ ミリ秒進む。このクライアントは、パーソナルコンピュータ上で実現してもいいし、ワークステーション上でも構わない。

ドライバクライアントは、Music Server と電子楽器との間のインタフェースを司る。ここで、電子楽器は抽象化され、一つのクライアントとみなされている。すなわち、ドライバクライアントは他のアプリケーションクライアント等と同様な方法で、Music

† Music Server System—Distributed Music System on Local Area Network— by TATSUYA AOYAGI (The University of Electro-Communications) and KEIJI HIRATA (Institute for New Generation Computer Technology).  
J. of Info. Processing, Vol. 15, No. 1 (1992), pp. 1-9

†† 電気通信大学

††† (財)新世代コンピュータ技術開発機構

Server と通信する。

アプリケーションクライアントは、音楽アプリケーションを実現するために、Music Server と通信し音楽イベント操作等を行う。

このシステムを MIDI インタフェース以外の楽器に拡張することは容易である。また、複数のドライバクライアントを Music Server に接続するように拡張することも容易である。

現在の実装では、Music Server は C 言語で書かれていて、約 2,000 行、ドライバ、クロッククライアントも C 言語で、約 1,800 行である。アプリケーションクライアントの例は Prolog で書かれている。Prolog から Music Server を使うためのライブラリが用意されている。それらは、SICStus Prolog (700 行) および C 言語 (500 行) で書かれている。

### 3. モデル

Music Server モデルは次の abstraction からなる。すなわち、イベント、トラック、タイムマップである。

イベントは、アクションとそれが実行される時間との組として定義される。現在の実装では、イベントには三つのタイプがある。メッセージイベントが実行されると、メッセージがクライアントに送られる。メッセージイベントはクライアント間の通信に使われる。MIDI イベントはメッセージイベントの特殊なものである。MIDI イベントが発火すると、Music Server は MIDI コードを含むパケットを生成し、ドライバクライアントに送信する。コマンドイベントを実行することにより、ループ構造、トラックの生成、タイム

マップの管理、特定のイベントの削除などの操作が可能である。

トラックはイベントを保持し、イベントを正確な時間に発火させる。大まかに言えば、トラックは楽譜の五線やテープのトラックに対応する。それぞれのトラックは自分自身のタイムマップを持つ。タイムマップは、実時間をトラック時間に変換する単調増加関数である。

### 4. 性能評価

ネットワーク遅延とプロセス切り替えのオーバーヘッドが、システムの応答時間を引き延ばすだけでなく、時間的に正確なイベント操作の障害となる可能性がある。この問題を解決し、時間精度を高く保つために、デッドラインスケジューリングとイベントバッファリングを採用した。

この機構は、しかし、バッファ遅延、すなわち、サーバ内の仮想的な時間と実時間とのずれを引き起こす。このずれに影響する主な因子は、(1)周期的に送られる同期メッセージの間隔、(2)イベント先取りの時間幅、(3)要求メッセージを作る時間の三つである。われわれは、さまざまな条件でシステムの性能を測定し、このずれの下限を評価した。さらに、ずれに対する個々の因子の感度を調べた。

### 5. おわりに

今後は、より多くのアプリケーションクライアントを開発し、Music Server のコマンド構成を改良する。さらに、音楽に限らない実時間アプリケーション一般に適した分散アーキテクチャを研究する。

## 欧文誌掲載論文要約

<論 文>

### 中国語漢字のジップ分布の線形化†

KIM TENG LUA††

中国語に対するジップ分布の挙動をより正確に記述する汎用な経験法則をわれわれは求めてきた。われわ

れが対象としたデータ集合は、(i) 32 の漢字数、(ii) 623 個の部首、(iii) 5,584 個の漢字、(iv) 46,520 個の熟語である。

われわれが最小二乗法による適合度を求め、結果を比較するのに使用したのは次の 4 つの数学的モデルで

† Linearization of Zipfian Distribution for Chinese Characters by KIM TENG LUA (National University of Singapore). *J. of Info. Processing*, Vol. 15, No. 1 (1992), pp. 10-16

†† シンガポール国立大学

ある：(i)双曲線型の拡張ジップの法則 ( $f$  対  $1/R$ ),  
 (ii)対数型の拡張ジップの法則 ( $\log f$  対  $\log R$ ),  
 (iii) $f'$  対  $R'$  ( $t$  は定数) の多項式級数, (iv)マンデル  
 ブロー・モデル (すなわち,  $f=a/(R+C)^b$ ).

(iii)に対しては, 観測と実験とによって, 分布が次の形をした直線へと線形化できることが示された.

$$f' = b_0 + b_1 R' \quad (1)$$

ただし,  $b_0, b_1, t$  は定数である.

適合時の  $b_2$  の値を観測することで, 線形化が達成される. それは,  $b_2=0$  か 0 に近いときの  $t$  の値である.

結果を表 1 に示す.

したがって,  $t=0.1975$  すなわち 0.2 に近いときに

表 1 線形化係数とそのエラー

種別	$t$	$b_0$	$b_1$	適合のエラー率
画面数	0.197335	1.2708	-0.50396	20.9%
部首	0.199088	0.6350	-0.13196	7.2%
漢字	0.194546	0.4283	-0.05980	4.8%
熟語	0.080245	1.0108	-0.29729	4.2%

は,  $f$  と  $R$  の依存度は 5 のオーダーであると結論することができる.

(1)式が他の 3 つのモデルと比較して, ジップ分布に対してより適切な表現となっていると考える. この理由は, 3 つのパラメータだけで一番よく適合する結果を与えているからである.

## 欧文誌掲載論文要約

<論 文>

### 流体運動方程式から発生したカオスのグラフィック表現†

荒谷 敏 朗\*\*

非線形常微分方程式の差分近似による数値解法からのバイファーケーション (二分岐) およびカオスの発生に関しては多くの研究が見られる. 本研究では非線形偏微分方程式の代表例である流体運動方程式であるナビエ・ストークス方程式に着目し, 二分岐およびカオスの発生可能な数値解法および発生条件を探索した.

流体運動方程式の数値解法に関しては, 有限差分法・有限要素法・境界要素法等が開発されているが, その基礎となるナビエ・ストークス方程式自体および差分公式等に未解決の問題がある. 差分近似法ではその分割数に限りがあり処理速度と関連してコンピュータの進歩にも大きく依存するテーマである.

従来のナビエ・ストークス方程式に差分公式を適用したのでは, 対称解しか得られないため, 基礎運動方程式を非線形項を強調した形に変換後に有限差分公式

を適用した. その結果非対称の差分形式が得られた. 空間変数に対しては非対称差分を採用することにより基礎偏微分方程式を時間に関する連立の常微分方程式に変換した. カオスの発生は常微分方程式にオイラーの数値解法を採用することにより起きることが分かっているため, 連立常微分方程式をオイラー近似により有限時間差分を用いて差分化した.

従来のナビエ・ストークス方程式は二次元の場合では二方向の速度成分と圧力の三変数に対して三式が成立しており解が定まるように思われるのであるが, 圧力が陰に入っているため初期値の算定等に困難を生じる. 本研究では, この困難を避けるため, 圧力勾配を運動エネルギーの勾配に置き換えるとする大胆な仮定を採用した. この仮定により圧力項を速度項に置き換えることができ, 差分化およびプログラミングが極めて容易となった.

有限要素法では, 有限要素の決定に人為的な方法を採用していること, および積分法に基礎を置いているので非線形性を検討するには適しないので, 本研究では有限差分法を採用することにした. 差分点の数が精度に直接影響するのが有限差分法であるが, 縦 200

† Graphic Presentation of Chaos Generated in Computational Fluid Dynamics by TOSHIRO ARATANI (Department of Management and Information Sciences, Hiroshima Prefectural University).

J. of Info. Processing, Vol. 15, No. 1 (1992), pp. 17-24

\*\* 広島県立大学経営情報学科

点と横 400 点と十分な数の差分点を取り主に時間差分による二分岐およびカオスの発生状況を数値実験により検討することにした。

円筒状の物体を均一な流れの中に突然投入したときの円筒の周りの流れを提案した数値解法により解き、各差分点における運動エネルギーすなわち動圧を色差としてグラフィック表示した。

レイノルズ数が 200 では、時間差分の増分値  $Dt$  が 0.00063 以上で二分岐が発生し後流の外側にゴーストのように数値積分誤差による陰の振動波が発生した。 $Dt$  が 0.000625 では二分岐がわずかに発生はするものの時間の経過と共に消滅していくことが確かめられた。レイノルズ数が 400 では、 $Dt$  が 0.00126 以上で二分岐とカオスが後流中および後流後に発生することが解った。レイノルズ数が 800 では、 $Dt=0.00063$  以上で後流内に二分岐が発生し、 $Dt=0.00075$  以上でランダムなカオス状態に移行することも解った。レイノルズ数が 200 では対流項の係数が拡散項の係数より小さいが、レイノルズ数が 400 では両係数は等しい。レイノルズ数が 800 では拡散項の係数は対流項の係数より小となる。拡散項は差分式の安定化に大きく寄与しており、高いレイノルズ数においては数値解は同じ  $Dt$  に対しては不安定となることが分かる。 $Dt$  を十分小さくとると高いレイノルズ数においても二分岐とカオスの発生がないことから、本研究で得られたカオスは数値積分誤差が主原因で発生したものと考えられる。乱流発生との関連については今後の検討課題である。

ジュコフスキーの翼型を流過する流れからも同じ  $Dt$  値で、二分岐とカオスが発生した。他の形状の物体に対しても同じ結果を得たことから、二分岐とカオスの発生には、物体の形状は影響しないことも明らかとなった。

十分に小さい  $Dt$  により計算した動圧分布から物体の後流における渦列は実測に良く類似した形状のものが得られ提案した数値解法の実用性が確かめられた。

提案した数値解法は、拡張性に富み圧縮性流体や三次元構造にも適用可能な手法である。特に、境界条件

の設定が容易であり、種々な形状の物体を流過する流体に適用できる。

アルゴリズムが単純であることから、並列計算が可能であるので、パラレルプロセッサやアレープロセッサを駆動するプログラミングが可能である。

本研究の計算およびグラフィック表示には、自作したコンパイラを使用した。文法は BASIC 言語に類似しており、人工知能型パーサを採用したのでアセンブラプログラミングに引けをとらないオブジェクトコードを発生可能である。32 ビットアドレッシングが可能であり広大なプロテクトメモリをアクセス可能である。実数演算には数値演算プロセッサを使用し、整数演算には 32 ビット CPU のレジスタ演算を 32 ビット整数で使用する。

PSET 文、LINE 文、CIRCLE 文等のグラフィックインタフェース文も装備しており、しかも超高速の描写が行えるためコンピュータグラフィックスを行うには最適のコンパイラである。演算結果のプロットの必要がない。

本研究により提案した流れのグラフィック表示法は、従来の流線を表示する手法とは異なり時間的に変動していく運動エネルギーすなわち動圧を知ることができる利点がある。流線を表示する方法では、時間的な変化の表示が困難である。数値解法の安定性を検討するには、まず運動エネルギー分布の表示を行うのが良策である。

## 参 考 文 献

- 1) 荒谷敏朗：非線形常微分方程式からのフラクタルの発生，情報処理学会論文誌，Vol. 29, No. 1, pp. 1-6 (1988).
- 2) 荒谷敏朗：非線形制御プロセスからのカオスとフラクタルの発生，システム制御情報誌，Vol. 34, No. 2, pp. 28-35 (1990).
- 3) 荒谷敏朗：パソコンによる数値計算の高速化，153 回講習会教材：数値流体力学の初歩から実用まで，機械学会関西支部，pp. 89-96 (1987).
- 4) 水野明哲：流れの数値解析入門，朝倉書店 (1990).

---

 欧文誌掲載論文要約
 

---

&lt;論 文&gt;

## 単一鍵方式による安全な放送形通信方式の一考察†

JINN-KE JAN CHOUNG-DONG YU

本論文では、放送形通信のための暗号方式について提案している。提案の方式は、RSA 暗号方式、中国剰余定理、単一鍵の考え方に基づくもので、以下に示す特長を有する。

- (1) 暗号文は一つだけである。
- (2) 暗号化の操作は一回だけである。
- (3) 各々の利用者が用いる鍵は、公開鍵方式に基づくものである。

- (4) 暗号文の長さは、従来提案されているものより短い。
- (5) メッセージ送信者は宛先の相手を任意に指定することができる。
- (6) デジタル署名にも容易に適用できる。
- (7) 暗号方式の安全性は RSA 暗号方式と同等である。

この暗号方式を用いることにより、複数の正当な利用者に対して同時に機密文書を送信することが可能になる。本方式は、従来の方式に比べて暗号化・復号化が簡単明瞭で、かつ、付加メッセージの長さも短いことが特長である。

---

† Yet Another Approach for Secure Broadcasting Based upon Single Key Concept by JINN-KE JAN and CHOUNG-DONG YU (Institute of Applied Mathematics, National Chung Hsing University).  
*J. of Info. Processing*, Vol. 15, No. 1 (1992), pp. 25-28

---

 欧文誌掲載論文要約
 

---

&lt;論 文&gt;

## 矛盾を契機とする非単調推論の確率的意味について†

佐 藤 健††

非単調推論は不完全な知識下における人間の推論の定式化を目的としている。不完全な知識の補完の一つの方法は常識を用いることである。常識は普通よく成り立つ情報であるので、常識を用いればたいていの状況で正しい決定をすることができる。つまり、常識はある種の統計的または確率的な性格を持つと考えられる。このような考え方から最近、確率に基づいて非単調推論の意味を考えようとする研究が多くなされている。

しかし、極小限定や非単調論理のような非単調推論の論理的な定式化と確率による定式化との理論的な関係は Lifschitz が指摘しているようにいまだ明らかになっていない。

本論文では非単調推論を前提帰結関係 (consequence relation) としてとらえ、前提帰結関係のあるクラスが確率的意味を持つことを示す。前提帰結関係は、論理式間の 2 項関係であり、前者がある前提を表し後者がある論理の推論規則に基づいて得られる帰結を表す。

非単調推論の論理的定式化は以下のような前提帰結関係でとらえることができる。つまり、前提は今わかっていて知識の状態を表し、帰結はそのときに信じ

---

† A Probabilistic Interpretation for Lazy Nonmonotonic Reasoning by KEN SATOH (Institute for New Generation Computer Technology).  
*J. of Info. Processing*, Vol. 15, No. 1 (1992), pp. 29-39

†† (財) 新世代コンピュータ技術開発機構

ている信念を表すと考える。たとえば、極小限定では、最初に与えられた公理を前提とし、ある述語を極小化した結果得られる定理を帰結として考えればよい。また、Default Logic では、公理を前提として考え Default 規則によって得られた推論結果を帰結として考えればよい。

われわれは Lehmann, Magidor によって研究された合理的前提帰結関係 (rational consequence relation) と呼ばれる、より制限された前提帰結関係のクラスに対して確率的な意味を与える。合理的前提帰結関係においては、ある知識から得られた信念は、その信念に矛盾しない知識が付け加わらない限り保持される。つまり、非単調性が現れるのは、現在保持している信念に矛盾する知識が付け加わるときである。そういう意味でこの性質を持つ推論は矛盾を契機とする非単調性を持つ (lazy nonmonotonic) といえる。

ここで与える確率的な意味は、確率の極限に基づいている。前提を表す論理式のもとでの帰結を表す論理式の条件付き確率をパラメータ  $x$  を持つ確率関数で表し  $x$  を 0 に近づけていったときの確率関数の極限が 1 のときかつそのときにかぎり、その論理式の組は前提帰結関係の中に入っているとす。この前提帰結関係を確率の極限による前提帰結関係と呼ぶ。

このように確率の極限による前提帰結関係を定義すると以下のことが示される。前提帰結関係が合理的 (rational) であるときかつそのときにかぎり、あるパラメータ  $x$  を持つ確率関数が存在し、その確率関数の極限による前提帰結関係がその合理的前提帰結関係に一致する。

さらに、本論文では、この結果を極小限定に対する確率的意味に応用する。上で述べたように選好的前提帰結関係や合理的前提帰結関係は可能状態の順序に基づいた意味論を持っており、極小限定は解釈の順序に基づいた意味論を持っている。この親和性により、極小限定への応用が可能となる。しかし、極小限定によって定義される前提帰結関係は選好的前提帰結関係ではあるが合理的前提帰結関係でない場合がある。とくに定義が変化できない述語がある場合や同時に 3 つ以上の述語を極小化する場合にはいつも合理的前提帰結関係ではない。

しかし、解釈の順序がランク付けされているような極小限定のクラスでは、上で述べたような確率関数を定義することができ、その確率関数の極限による前提帰結関係がその極小限定で定義される前提帰結関係と一致することを示す。

## 欧文誌掲載論文要約

<情報処理学会論文誌からの翻訳>

### 論理回路生成エキスパートシステムにおける推論方式†

戸次 圭介<sup>††</sup> 横田 孝義<sup>††</sup> 永井 徹<sup>††</sup>  
中塚 範子<sup>††</sup> 浜田 亘曼<sup>††</sup>

現状の方法を用いてエキスパートシステムを構築するには、まず専門家の知識をルールの集まりとして記述しなければならない。しかしながら、ルールとして記述できる知識の概念レベルには限界があるため、設計問題のように複雑な問題を扱うエキスパートシステ

ムを構築することは非常に困難で重要な課題となっている。このような課題を解決するには、設計問題をよく分析し、設計問題に適した汎用的な推論方式を確立しなければならない。

本論文では、論理回路設計問題を例題として取り上げ、設計問題に適した推論方式について議論する。図 1 は、大規模な論理回路の代表例であるマイクロプロセッサの回路構成を示したものである。大規模な論理回路は、この図に示されるように必ず階層構造となっている。したがって、論理回路設計とは、多くの部品

† Reasoning for a Logic Circuit Synthesis Expert System by KEISUKE BEKKI, TAKAYOSHI YOKOTA, TOHRU NAGAI, NORIKO NAKATSUKA and NOBUHIRO HAMADA (Hitachi Research Laboratory, Hitachi Ltd.).

*J. of Info. Processing*, Vol. 15, No. 1 (1992), pp. 41-49

†† (株)日立製作所日立研究所

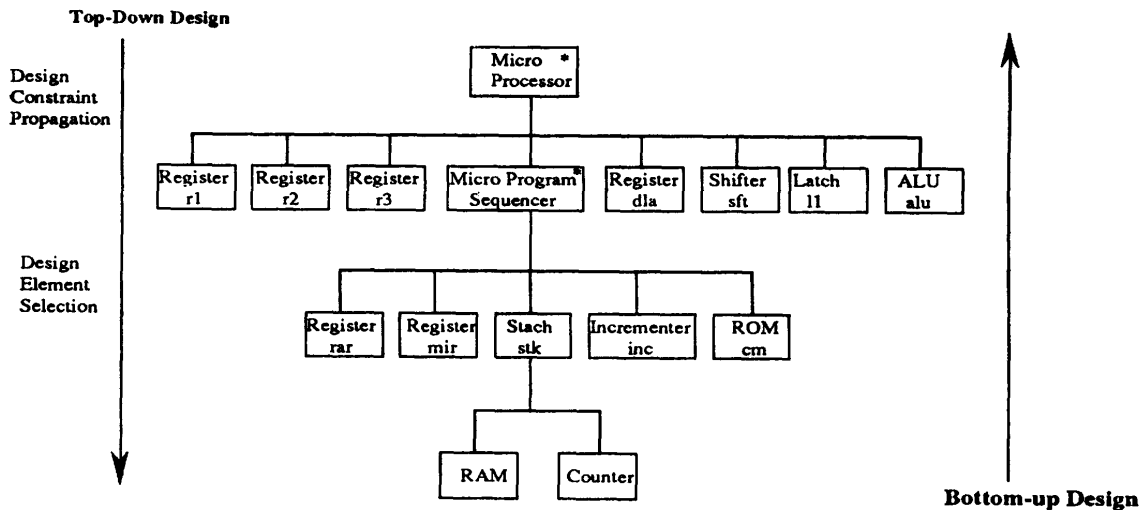


Fig. 1 The components structure of the micro processor.

回路を組み合わせ、つなぎ合わせるにより仕様を満足する回路を作ることである。論理回路の部品は、(1)過去に設計した回路をそのまま利用できるものと、(2)再設計しないと利用できないものとの2つの種類に大別することができる。(1)の種類の論理回路を設計する問題は、過去の経験に基づいて設計要素を詳細化する、トップダウン戦略に基づくルーチン設計を用いて解決することができる。(2)の種類の論理回路を設計する問題は、トップダウン戦略と全体の使用を解析して設計要素をつなぎ合わせるボトムアップ戦略の両方を用いることにより、解決することができる。

したがって、すべての論理回路設計問題は、トップダウン戦略とボトムアップ戦略とを組み合わせた設計モデルでモデル化できると考えられる。

この設計モデルを実現する推論方式は、オブジェクト指向のプログラミング技法の導入により拡張よく実現でき、さらにこれを用いて論理回路設計エキスパートシステム ProLogic が容易に構築できる。ProLogic を用いて論理回路の設計を試みた結果、人手設計による回路とほぼ同等の品質の回路が得られた。これにより、本論文で議論する推論方式が人間の設計者の設計過程をよくモデル化していることが明らかになった。

## 欧文誌掲載論文要約

<情報処理学会論文誌からの翻訳>

### 多属性効用理論を応用した知識獲得方式†

満田 透<sup>††</sup> 小林 康弘<sup>††</sup> 野中 久典<sup>††</sup>

エキスパートシステムを効率良く構築するための技術課題として、専門家から知識獲得を容易とする支援技術の開発が挙げられる。これに対し、インタビュー

を効果的に行い専門家が意識していない知識を意識させるための知識獲得支援手法がいくつか提案されている。一方、専門家が意識はしていても、言葉による表明が困難なためルール化できない場合もある。特に、設計・計画問題では、コスト、信頼性、効率などの複数の指標を統合的に評価して、特定の代替案を選択する機会が多い。エキスパートシステムの構築に際して、この判断基準を従来のルール形式で表現するため

† A Knowledge Acquisition Method Based on a Multi-Attribute Utility Model by TOORU MITSUTA, YASUHIRO KOBAYASHI and HISANORI NONAKA (Energy Research Laboratory, Hitachi, Ltd.).

*J. of Info. Processing*, Vol. 15, No. 1 (1992), pp. 50-59

†† (株)日立製作所エネルギー研究所

には、判断基準を各指標の具体的な値の組み合わせとして明確化する必要があるが、これは専門家自身でも困難であることが少なくない。本論文では、このような場合の知識表現手法および専門家の判断結果（既存の事例）からの知識獲得支援手法を提案する。

本手法では、多目的意志決定問題の一解法である多属性効用理論のモデル（多属性効用関数）を知識表現モデルとして利用し、専門家が判断した代替案の選好順序から、判断の基準となる知識を獲得する。この手法によれば、判断基準を属性値の組み合わせの関係として直接明確化することが困難であっても、考慮すべき属性が明らかであれば、事例からの半自動的な知識獲得、すなわちモデルの同定が可能となる。モデルの同定は、専門家とモデルとからそれぞれ得られる代替案の選好順序を比較し、この結果に基づきモデルを修正することにより行う。修正では、修正候補を絞り込むための4段階の仮説選択モジュールと全体を制御する制御モジュールを用いる。処理を効率化するために、仮説の選択等にヒューリスティクスを利用する。得られたモデルは、メタルールにより、関連するルールと対応させておき、ルール前提部の判定時に利用する。

本手法を建設作業の順序決定に関する知識に試験的に適用した結果、専門家が行った順序付けを再現できる知識の獲得ができた。本手法は、従来のルール形式では獲得が困難である設計案の選択などに関する知

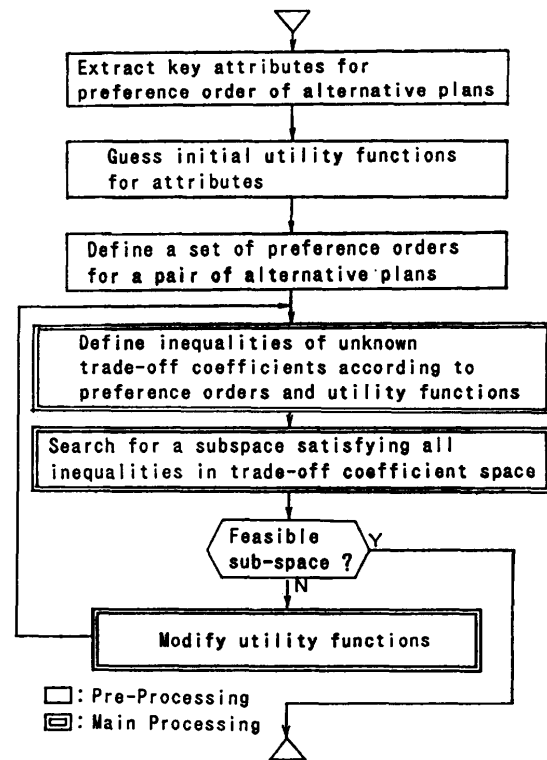


Fig. 2 Procedure for knowledge acquisition.

識、すなわち複数の属性を総合的に評価して、代替案を順序付ける知識に適用することが可能である。（図として、図2知識表現モデルの同定手順を添付）

## 欧文誌掲載論文要約

<情報処理学会論文誌からの翻訳>

### 4 階層プラン認識モデルを使った対話の理解†

飯田 仁†† 有田 英一††\*

日本語の話し言葉による対話は書き言葉と比べて、旧情報や述語の一部などが省略されたり、述語などに

対して代用的な表現が使われるなど断片的な発話が多く現れる。また、様々な言語表現による情報伝達行為がそれらに付加される。このような特徴を持つ対話の理解および翻訳を目指すとき、各断片的な発話を対話進展の流れに沿って解釈することが必要である。断片的な発話の解釈は、対話者が共有している話題領域の知識や対話実現のための言語運用の知識などを援用することによって可能となる。それにより省略の復元が可能

† Natural Language Dialogue Understanding on a Four-layer Plan Recognition Model by HITOSHI IIDA and HIDEKAZU ARITA (ATR Interpreting Telephony Research Laboratories). *J. of Info. Processing*, Vol. 15, No. 1 (1992) pp. 60-71

†† ATR 自動翻訳電話研究所

\* 現在 三菱電機(株)

Mitsubishi Electric Corp.



となる。

本論文では対話を理解するための一つの機構を提案する。このモデルは目標指向型対話を対象としている。対話のコーパスは「国際会議の参加申込みに関する問合せ」として計算機端末間で収集したものを一部修正してある。本モデルでは4階層のプラン認識モデルを使った目標指向型対話の理解の手法を提案する。4階層のプランとは(1)ある情報の交換が話し手と聞き手の間の順序づけられた発話で表現できるという知識であるインタラクションプラン、(2)対話を介した情報伝達のための行為が一連の情報伝達行為で実現で

きるという知識であるコミュニケーションプラン、(3)ある行為が順序づけられた行為の達成で実現できるという知識であるドメインプラン、(4)対話が対話展開のための言語運用行為で実現できるという知識であるダイアログプランである。これらのプランを利用すると対話の進展に伴う各発話と話題領域の知識との関係付けを漸次進めていくことが可能となり、対話全体に渡る構造を作ることができる。対話の構造の中で断片的な発話の解釈をすることにより代用的表現、格要素の省略を復元できる。

## 欧文誌掲載論文要約

<情報処理学会論文誌からの翻訳>

### 命題自己認識論理における拡張世界構成アルゴリズム†

馬場口 登<sup>††</sup> 森馬 純一<sup>††</sup> 手塚 慶一<sup>††</sup>

非単調推論を定式化する論理の1つである自己認識論理  $AL$  (Autoepistemic Logic) は、信念を基に結論に到達する推論 (自己認識推論) をモデル化したものである。 $AL$  は知識表現の基礎となる通常論理体系に信念オペレータ  $L$  を導入して拡張されたものであり、 $L$  を「信じている」と解釈する。

自己認識論理で重要な概念である拡張世界 (extension) は、前提  $A$  から自己認識推論により得られる結論に相当するものであるが、拡張世界を前提から直接的に構成することが難しいという問題点がある。Moore は、この問題点に対処するために、可能世界意味論から  $AL$  を考察し、重要な定理を導いた。本論文では、Moore の考察を基に、拡張世界と可能世界意味論に関する新しい定理を与え、命題  $AL$  の拡張世界構成アルゴリズムを具体化する。

可能世界意味論では、構造  $K$  (可能世界集合とその到達可能性)、および真理値割当  $V$  により解釈が規定されるが、特に命題  $AL$  では、安定な理論  $T$  (自己

認識論理式の集合) が完全  $S5$  構造 (任意の世界から、すべての世界に到達可能) によって表せるという性質を持つ。拡張世界に関して、

[定理]  $K$  を安定な理論  $T$  を表す完全  $S5$  構造とするとき、次の2つの条件は同値である。

- (1)  $T$  の前提の集合  $A$  の拡張世界である。
- (2)  $A$  の各式が真となるどの  $T$  の可能世界解釈  $(K, V)$  についても、 $V$  の真理値の割当に一致する可能世界  $w$  が  $K$  に含まれ (包含条件と呼ぶ)、かつ  $K$  のどの可能世界  $w$  でも  $A$  の各式が真となる (充足条件と呼ぶ)。

が成り立ち、拡張世界を求めることは上述の充足・包含条件を満たす完全  $S5$  構造  $K$  を求めることに帰着される。アルゴリズムの詳細を以下に記述する。

[拡張世界構成アルゴリズム]

- (1) 前提の有限集合  $A$  に現れるすべての  $L$  式 ( $LP$  の形の式、部分式も含む) の集合を  $\Delta_L$  とする。また前提に現れるすべての命題定数の集合を  $\Delta_c$  とする。
- (2)  $\Delta_L$  に含まれる各式に対する真理値の割当  $U_i$  ( $i=1, \dots, i \max$ ) を行う。同様に  $\Delta_c$  に含まれる各命題定数への真理値の割当  $V_j$  ( $j=1, \dots, j \max$ ) を行う。

† An Algorithm for Constructing Extensions of Propositional Autoepistemic Logic by NOBORU BABAGUCHI, JUN-ICHI MORIUMA and YOSHIKAZU TEZUKA (Department of Communication Engineering, Faculty of Engineering, Osaka University).

*J. of Info. Processing*, Vol. 15, No. 1 (1992), pp. 72-78

†† 大阪大学工学部通信工学科

表 1 拡張世界構成用テーブル

$U_i/V_j$	$V_1$	$V_2$	...	$V_{j_{\max}}$	$K$	Label
$U_1$						
$U_2$						
$\vdots$						
$U_{i_{\max}}$						

(3) 以下の手順に従い、表 1 の空欄を埋めていく。

(3a) 真理値の割当  $U_i$ , ( $i=1, \dots, i_{\max}$ ),  $V_j$ , ( $j=1, \dots, j_{\max}$ ) に対して、 $\Delta$  のすべての式が真となるとき、表の  $U_i$  と  $V_j$  の交わる欄に“1”，そうでないとき“0”を記入する。

(3b)  $U_i$  について、 $U_i$  と  $V_j$  の交わる欄を“1”としている  $V_j$  の集合を作り、 $K$  の欄に記入する。

(3c)  $U_i$  について、 $K$  が次の条件を満たすか否かを調べる。

- $\Delta_L$  に含まれる各々の式  $LP_k$  について、
- $U_i$  によって  $LP_k$  に真が割り当てられているとき、 $K$  に含まれるすべての  $V$  で  $P_k$

が真となる。

- $U_i$  によって  $LP_k$  に偽が割り当てられているとき、 $K$  に含まれる少なくとも1つの  $V$  で  $P_k$  が偽となる。

ただし、 $P_k$  がさらに  $LQ$  なる形の式を含むとき、 $LQ$  の真理値は  $U_i$  に従うものとする。

これらの条件を満たすとき、ラベルの欄に“C”，そうでないとき、“I”を記入する。 ■

本アルゴリズムにおいて“C”とラベル付けされる完全  $S5$  構造  $K$  が、充足条件・包含条件を満たし、またこの2条件を満たす  $K$  が、本アルゴリズムで必ず“C”とラベル付けされること、すなわちアルゴリズムの妥当性が証明できる。また本アルゴリズムの計算複雑度は、入力サイズ  $n$  を  $L$  式と命題定数の総和とするとき  $O(n^2)$  となる。

本アルゴリズムは、可能世界を通して式  $p$  と式  $Lp$  との真理値の対応関係が与えられるという理論的根拠に依るもので、拡張世界が直接的に構成でき、しかも拡張世界で成立する式が、直観的に把握できることが利点として挙げられる。

## 欧文誌掲載論文要約

<情報処理学会論文誌からの翻訳>

### 2次元探索共役勾配法を用いた多層神経回路網の高速学習法†

吉 田 利 信\*\*

本論文では、自動微分法および2次元探索共役勾配法の概要が示され、これらの方法を用いた多層神経回路網の高速学習法が提案されている。

従来、偏導関数値を求める方法として、偏導関数を表す式を数式処理によって求め、それを評価する方法、および、数値微分法が用いられてきた。数式処理を用いる方法は、正確な偏導関数値が得られるという長所はあるが、関数が複雑になると偏導関数を表す式

が膨大なものとなり、数式処理システムですらその導出ができなくなるという短所がある。一方、数値微分法では、偏導関数を表す式を求めることなしに偏導関数値が得られるが、差分による近似を用いるため精度の高い偏導関数値を求めることが困難である。また、これらの方法では、スカラ関数の勾配計算の手間が変数の数に比例するという欠点もある。これらの方法に対して、関数の計算過程を順または逆にたどることによって、正確な偏導関数値を効率よく計算する自動微分法がある。スカラ関数に対しては、関数の計算過程を逆にたどることによって、変数の数によらずに、関数計算の高々定数倍の手間で勾配を得ることができ

† Rapid Learning Method for Multilayered Neural Networks Using Two-Dimensional Conjugate Gradient Search by TOSHINOBU YOSHIDA (Department of Computer Science, Faculty of Engineering, Gunma University). *J. of Info. Processing*, Vol. 15, No.1 (1992), pp. 79-86

\*\* 群馬大学工学部情報工学科

Rumelhart らによって提案された誤差逆伝播学習法は、多層神経回路網の結合係数を最急降下法に基づいて学習する方法である。ここで誤差関数の勾配は、誤差関数の計算過程を逆にたどることによって計算されており、この計算法は自動微分法の関数の計算過程を逆にたどる方法そのものである。

逆伝播法では、勾配ベクトルを $-\eta$ （学習パラメータ）倍したベクトルを用いて神経回路網の結合係数を修正しているが、最急降下法であるため一般に収束は遅い。収束を加速する方法として、この修正ベクトルに直前の修正ベクトルを $\alpha$ （慣性パラメータ）倍したベクトルを加える方法が示されている。しかし、これらのパラメータは学習の過程において最適な値が変化しており、固定したパラメータで学習を行う場合、これらのパラメータは何回かの試行によって決定されなければならない。

本論文で用いられている2次元探索共役勾配法は、誤差関数が最急降下方向と直前の修正方向を含む2次元平面上において極小となるように、誤差関数の2次関数近似を用いて二つのパラメータ $\eta$ と $\alpha$ を探索する。本方法にはヘッセ行列の2次形式などの計算が含

まれているため、ヘッセ行列を求めてからこれらの値を計算すると、神経回路網の大きさの2乗に比例した計算領域と計算量が必要となる。しかし、関数の計算過程を順にたどる自動微分法を用いることにより、神経回路網の大きさに比例した計算領域と計算量でこれらの値を計算することが可能となる。

本論文では、神経回路網における誤差関数の勾配ベクトル、ベクトルと勾配ベクトルとの内積、および2次形式の値を計算するための、自動微分法を用いた具体的な計算方法が示されている。また、これらの計算に必要な記憶領域が、神経回路網を表現するのに必要な記憶領域の3倍強であること、および、本方法の繰り返し1回当たりの計算量が逆伝播法の高々6倍の計算量であることが示されている。また、逆伝播法、慣性項を伴う逆伝播法、学習パラメータと慣性パラメータを逐次変化させる逆伝播法および本高速学習法が、数値実験において比較され、本方法を用いることによって、これらの逆伝播学習法よりもかなり少ない反復回数で学習が収束すること、繰り返し1回当たりの計算時間が逆伝播法の約3倍となることが示されている。

## 欧文誌掲載論文要約

<情報処理学会論文誌からの翻訳>

### チェックポイントング濃度を使用した最適チェックポイントング方策†

福本 聡<sup>††</sup> 海生 直人<sup>†††</sup> 尾崎 俊治<sup>††††</sup>

データベースシステムにおけるファイル系の構成および回復技術の研究は、システム運用の上で極めて重要な課題である。システムに起こる何らかの障害により、多くの情報を記憶するファイルが破壊されたり、

誤った情報が付加されたりすると、データベースシステムの信頼性および処理性能に多大な支障をきたすことは明らかである。ここでは、障害発生時のファイル回復技術の代表的な例であるロールバック・リカバリについて考察する。

一般にデータベースシステムでは、処理するトランザクションの要求により主記憶装置のファイルの更新が行われるとき、そのトランザクションの処理に必要な情報をジャーナル（またはログ）と呼ばれる安定な二次記憶装置（磁気ディスク、磁気テープ）に保存する方法が用いられている。これによりファイルの更新に関するトランザクションの取り消し（undo）、やり直

† Optimal Checkpointing Policies Using the Checkpointing Density by SATOSHI FUKUMOTO (Graduate School of Engineering, Hiroshima University), NAOTO KAI (Department of Management Science, Hiroshima Shudo University) and SHUNJI OSAKI (Cluster II (Electrical Engineering), Faculty of Engineering, Hiroshima University).  
*J. of Info. Processing*, Vol. 15, No. 1 (1992), pp. 87-92

†† 広島大学大学院工学研究科

††† 広島大学管理科学科

†††† 広島大学工学部第二類

し (redo) が可能となる。主記憶装置のファイルが何らかの障害により破壊されたとき、あらかじめ正常な時点で主記憶装置から採取されたファイルのバックアップコピーをあらかじめ主記憶装置に導入し、その上にジャーナルから得られたトランザクションの情報の処理を重ねることにより、ファイルを障害の発生する直前の状態までリカバリすることができる。このとき、ファイルの状態を過去のバックアップコピーを採取したときの状態まで戻すことをロールバックと呼ぶ。バックアップコピーが最も古い状態であるとき、ロールバックはシステムの稼働開始時点までさかのぼり、ジャーナルによるファイルリカバリは膨大なものになる可能性がある。そこで、システムに障害が発生するまでにあらかじめ定められた時点ごとに主記憶装置のファイルの情報を、安定な二次記憶装置に保存する。この処理をチェックポイントニング、処理を行う時刻をチェックポイントと呼ぶ。チェックポイントの決定はシステム全体の処理効率、コストなどを考慮して行われなければならない。すなわち、頻繁にチェックポイントニングを行うとジャーナルによるロールバック・リカバリのための時間・費用は小さくてすむが、チェックポイントに要する総合的な時間・費用が大きくなり、逆に少なく行くとチェックポイントニングに関する時間・費用は小さくてすむが、障害発生時のロールバック・リカバリのための時間・費用が大きくなるという問題がある。

本稿では、システムが稼働を開始して、障害発生後にロールバック・リカバ리를完了し、再びシステムが

稼働を開始するまでの一連の動作を1サイクルとする確率モデルを考え、チェックポイントに要する費用とロールバック・リカバリに要する費用とのトレードオフを考えたチェックポイント方策について議論する。評価関数として定常状態における単位時間当りの近似期待費用を導入する。これはシステムが再び稼働開始する時点が再生点となるため、1サイクルにおける単位時間当りの近似期待費用に置き換えることができる。1サイクルにおける近似期待費用は、各チェックポイント間隔によるチェックポイント費用関数と、最も新しいチェックポイントから障害発生までの時間間隔によるロールバック・リカバリ費用関数とによって表す。本稿のモデルでは、障害発生時間の累積時間分布に一般分布を仮定しているため、すなわち故障率が一般に時間の経過とともに変化するため、最適なチェックポイントは間隔が時間とともに変化する時刻列として導かれる。

解析ではまず、単位時間当りのチェックポイントニング回数を表す関数—チェックポイントニング濃度—を考え、上述の費用関数および分布を用いて定常状態における単位時間当りの近似期待費用をチェックポイントニング濃度の汎関数として求める。そして、それを最小にするチェックポイントニング濃度を導く。次にこれらの結果に具体的な費用関数を与え、それらを含めた近似期待費用、チェックポイントニング濃度を表す。最後に、障害発生時間の累積分布関数にワイブル分布を仮定した場合の数値例により、本稿の解析結果の有効性を示す。

## 欧文誌掲載論文要約

<情報処理学会論文誌からの翻訳>

### 準 LL (2) 文法に対する構文解析表の作成アルゴリズム†

吉田 敬一††

竹内 淑子†††

本論文は準 LL ( $k$ ) 文法と称する LL ( $k$ ) 文法の新しいクラスを提案し、 $k=2$  の場合の解析表の作成

アルゴリズムならびにその表による解析法を示すものである。準 LL ( $k$ ) 文法と LL ( $k$ ) 文法、強 LL ( $k$ ) 文法との間には

$$\text{強 LL } (k) \subseteq \text{準 LL } (k) \subseteq \text{LL } (k), \quad (k \geq 1)$$

の関係がある。かつ  $k=2$  の場合には、このクラスに対して実現した解析表作成法の手数は Aho-Ullman の LL (2) 文法に対する場合に比べて、後に示すように圧倒的に少ない。

† An Algorithm for Constructing a Semi-LL (2) Grammar's Parsing Table by KEIICHI YOSHIDA (Department of Computer Science, College of Engineering, Shizuoka University) and YOSHIKO TAKEUCHI (Department of Computer Science, Hamamatsu Polytechnic College).

*J. of Info. Processing*, Vol. 15, No. 1 (1992), pp. 93-107

†† 静岡大学工業短期大学部

††† 浜松職業訓練短期大学校

	<i>a</i>	<i>b</i>	$\$$
<i>S</i>	[ ]1	[ ]2	
<i>A</i>	[ <i>a</i> ]4	[ ]3 [ <i>b</i> ]4	
<i>a</i>	[ ]1 [ <i>a</i> ]4	[ ]1	
<i>b</i>	[ <i>a</i> ]3	[ ]2	
$\$$			

図 1

以下に、実例に従って、実現した準 LL (2) に対する解析表の構造とそれにもとづいた解析法を示す。

1.  $S \rightarrow aAaa$
2.  $S \rightarrow bAba$
3.  $A \rightarrow b$
4.  $A \rightarrow \epsilon$

この文法は強 LL (2) 文法ではないので、一般には Aho-Ullman の解析表作成法に従わなければならない。しかしこの文法は準 LL (2) 文法であるので、本論文で提案するアルゴリズムを適用することができ、その結果、図 1 の解析表  $T$  が得られる。

解析表  $T$  を用いて入力テキスト  $abaa$  を解析する過程を以下に示す。解析の状態を

(テキストの残り, 解析スタック, 解析列)

で表す。初期状態ではテキストの残りは入力テキストに  $\$$  を接続した  $abaa\$\$$  であり、スタックは  $S\$\$$ 、解析列は  $\epsilon$  である。つまり、初期状態は

$(abaa\$\$, S\$\$, \epsilon)$

である。最初は、テキストの先頭二つは  $ab$  で、解析スタックの先頭は  $S$  であるから、以下のようにして、演算子  $\cap$  を用いて、解析表より  $T(S, ab)$  を求める。

$$\begin{aligned} T(S, ab) &= T(S, a) \cap T(a, b) \\ &= [ ]1 \cap [ ]1 \\ &= [ ]1 \end{aligned}$$

この場合、 $T(S, ab)$  の要素はただ一つであり、[ ]1 は無条件で規則 1 を適用することを意味する。したがって、解析状態は

$(abaa\$\$, S\$\$, \epsilon) \vdash (abaa\$\$, aAaa\$\$, 1)$

と進む。この状態ではテキストも解析スタックもとも

に終端記号であるから、pop する。すなわち、

$$\begin{aligned} (abaa\$\$, S\$\$, \epsilon) \vdash (abaa\$\$, aAaa\$\$, 1) \\ \vdash (baa\$\$, Aaa\$\$, 1) \end{aligned}$$

と進む。ここで、 $T(A, ba)$  についても同様に、

$$\begin{aligned} T(A, b) \cap T(b, a) &= [ ]3, [b]4 \cap [a]3, [b]4 \\ &= [a]3, [b]4 \end{aligned}$$

を得るが、この場合は選択すべき生成規則が一意に定まらない。これは文法が強 LL (2) でないことによる。そこで提案する解析アルゴリズムに従い、解析スタックの先頭の  $A$  に右接する文法記号と一致する方を選択する。この場合は、 $[a]3$  が選択される。以下こうして解析を続けていくと

$$\begin{aligned} \vdash (baa\$\$, baa\$\$, 13) \\ \vdash (a\$\$, a\$\$, 13) \\ \vdash (\$\$, \$\$, 13) \end{aligned}$$

となって解析が進行し、最終的に、解析列 13 が得られる。

準 LL ( $k$ ) 文法の  $k=2$  の場合につき ISO PASCAL 文法と PASCAL-文法を用いて、本論文で提案する解析表作成アルゴリズムと Aho-Ullman のアルゴリズムと実験的に比較した結果、以下の結果を得た。

- (1) 解析表に要するメモリ空間の大きさは  $1/120 \sim 1/400$  程度ですむ。
- (2) 解析表作成のための所用時間は約  $1/10$
- (3) 生成規則の数は  $1/8 \sim 1/16$  ですむ。
- (4) この解析表によると解析時間は 2 倍程度遅くなるが、これは解析表の構造を改良することにより、同程度まで向上させることができる。(ただし、このときの解析表は大きくなるが、それでも Aho-Ullman のものに比べると  $1/15 \sim 1/30$  程度である。)

これらの比較結果はアルゴリズムの性質から考えて、他の具体的なプログラム言語に対しても成り立つものと推定できる。さらに、本論文で提案する解析表の作成アルゴリズムは一部の条件をはずすことにより、LL (1) 文法に対する解析表作成アルゴリズムに容易に変更し得る。

## 欧文誌掲載論文要約

&lt;情報処理学会論文誌からの翻訳&gt;

## 順序付けによる点パターン的高速マッチング・アルゴリズム†

張 鴻 賓<sup>††</sup> 美 濃 導 彦<sup>††</sup> 池 田 克 夫<sup>††</sup>

パターン認識におけるマッチング問題において、以下のように記述できる点パターンのマッチングは重要な役割を果たしている。

平面上の二つの点パターンを  $U$ ,  $V$  とする。

$$U = \{u_1, u_2, \dots, u_i, \dots, u_m\} \quad 1 \leq i \leq m$$

$$V = \{v_1, v_2, \dots, v_i, \dots, v_n\} \quad 1 \leq j \leq n$$

ただし,  $u_i = (x_{ui}, y_{ui})$ ,  $v_j = (x_{vj}, y_{vj})$ .  $u_i$  と  $v_j$  はそれぞれ点パターン  $U$  と点パターン  $V$  の点で,  $(x_{ui}, y_{ui})$  と  $(x_{vj}, y_{vj})$  はそれぞれの座標系に対する点の座標値である.  $m$  と  $n$  は点パターン  $U$  と点パターン  $V$  の点の数である. このような点パターン  $U$  と  $V$  との間に位相的な対応関係を見つける問題が点パターンのマッチング問題である.

点パターンのマッチング問題は, 組合せ問題の性質によって計算複雑度のオーダーが非常に高くなる. 数学的に言えば, これはグラフの ISOMORPHISM 問題であり, 補助的な情報がなければ, NP 問題に属する. 原理的には, マッチング状態を表したマッチング状態木を用いて解決できるが, どのような情報を利用して, ヒューリスティック型のサーチをするかが問題である. 従来から, サーチの手法に関する議論は多くあるが, マッチング時の点の組み合わせ方に関してはあまり報告がない. そこで, 本論文では, 順序付けによる点パターンの組み合わせ方を工夫した高速マッチング・アルゴリズムを提案する.

画像処理では, 特徴点が誤って抽出されたり, あるべき特徴点が抽出されていなかったりする. また, 抽出された特徴点の位置にも誤差がある. したがって, 画像から抽出された特徴点を点パターンと考え, それをあらかじめ与えられたモデルとしての点パターンとマッチングさせる問題においては, 二つの点パターン

が全く同一であると判定するのではなく, ある誤差尺度によって二つの点パターン間の誤差を求め, 誤差値が最小となるものを見つけ出さなければならない.

点パターンのマッチングをアフィン変換に不変なものとするために, 両点パターン間で距離の尺度を一致させる比例因子を定義し, 相互角度と点どうしの相互距離を利用する. また, 特徴点の誤抽出や脱落に対処するために, 空の符号  $A$  を用いて, 対応する点が存在しない時には,  $(u_i, A)$  や  $(A, v_j)$  などの符号対をマッチング状態木に導入する.

マッチング・アルゴリズムの原理は,  $U$  パターンと  $V$  パターンに対応する二つの順序付けしたリスト上でマッチングを行うことである. 順序付けしたリストは次のように作成する.  $U$  パターンと  $V$  パターンから一つの基準線対を選ぶ. 基準線以外の各点に対して, 基準線との角度の昇順に順序付けしたリストを構成する. ある基準線対に対して, 順序付けした二つの点集合を  $U$  リストと  $V$  リストとする.

$$Ulist = (u_1, u_2, \dots, u_p, u_{p+1}, \dots, u_{m-2}) \quad 1 \leq p \leq m-2$$

$$Vlist = (v_1, v_2, \dots, v_q, v_{q+1}, \dots, v_{n-2}) \quad 1 \leq q \leq n-2$$

ただし, 基準線の起点と終点は順序リストから除く.

今,  $u_p$  と  $v_q$  が対応したとすると, 次のマッチング点対候補は,  $u_{p+1}$  と  $v_{q+1}$  あるいは  $v_{q+1}$  以後のある範囲内にある.  $u_{p+1}$  と基準線の角度を  $A_u$ , 許容角度誤差を  $A_t$  とすると,  $V$  リスト内の点 (角度を  $A_v$  とする) の中で,  $|A_u - A_v| < A_t$  を満足する点のみが,  $u_{p+1}$  と対応する可能性がある. したがって, 探索の範囲がかなり限定できる. 最大許容角度誤差  $A_t$  に対して,  $U$  リストの各要素と対応可能な  $V$  リストの要素数は, 現実的には, ある定数以下とみなせる. このとき,  $U$  リストと  $V$  リストのマッチングは,  $V$  リストに含まれている要素の数と無関係になるので,  $U$  リストの要素数の定数倍の時間 (線形時間) で完了する. 他に必要なコストは両リストの順序付けコストだけである.

アルゴリズムの複雑度は主として, 基準線対の数と一つの基準線対の中で実行される操作の計算量で決ま

† An Efficient Algorithm for Point Pattern Matching Using Ordered Lists by HONGBIN ZHANG, MICHIIHIKO MINOH and KATSUO IKEDA (Department of Information Science, Faculty of Engineering, Kyoto University).

*J. of Info. Processing*, Vol. 15 No. 1 (1992), pp. 108-115

†† 京都大学工学部情報工学教室

る。本アルゴリズムでは、一つの基準線対に対して、ソーティングとマッチングを行っている。マッチングはUリストの要素数の線形時間内で終了する。ソーティングはU, Vリストそれぞれ  $m \cdot \log m$  と  $n \cdot \log n$  の時間でできる。ただし、 $m$ と $n$ はUパターンとVパターンの点の数である。したがって、一つの基準線対

に対する処理時間は  $p \cdot \log p$  (ただし、 $p = \max(m, n)$ ) のオーダーとなる。基準線対の数は、最悪の場合で  $m^2 \cdot n^2$  本、最良の場合で1本で、対象とする問題で利用できる拘束条件に依存する。

このアルゴリズムを種々の例題に適用し、従来の手法と比較して、その有効性を確認した。

## 欧文誌掲載論文要約

<情報処理学会論文誌からの翻訳>

# ソフトウェア開発環境 dmCASE†

松浦 佐江子<sup>††</sup> 大林 正晴<sup>††</sup>

## 1. はじめに

現在、ソフトウェア開発の新しい技術であるプロトタイプング・オブジェクト指向・形式的仕様記述・AI・自動プログラミング等が研究されているが、ソフトウェア開発における確固たる設計の枠組みはまだ確立されていない。このような新しい技術の上に立ったソフトウェア開発のライフサイクルを考慮した統合的な開発環境の実現が重要な課題である。

## 2. ソフトウェア開発環境

本論文では、Balzer らの提唱するソフトウェア開発のパラダイムにおけるソフトウェア開発環境 dmCASE を提案する。ソフトウェア開発工程の早期において仕様レベルの検証・保守を行うためには、設計方法論と実行可能な形式的仕様記述が必要である。さらに環境として人間の思考過程に即した柔軟なユーザインタフェースを持つ環境が重要な役割を果たすと考える。これらを dmCASE ではつぎのように実現している。

- オブジェクト指向的な設計方法論「概念による設計法 (DMC)」に従った設計。

- 関数型言語 ML を形式的仕様記述言語として採用。さらに、ツールとしての支援に対する基本姿勢はつぎ

のとおりである。

- 人間の思考過程に即したユーザインタフェース
- 本質的に設計に関わる仕事だけに従事できる環境
- 思考を妨げない制約
- 検証

## 3. 設計方法論

与えられた要求の中に現れる具体的なくもの>やそれらを抽象化した<概念>を基にして、システムに対するモデルを構築する。このような<概念>に基づく構成によって問題を整理するオブジェクト指向的な設計方法論が DMC である。設計の手順の概略はつぎのとおりである。まず、上記のくもの>や<概念>を設計の断片として「単語表」に登録する。そしてそれらを使いながら、要求の要旨を「概念構造図」としてモデル化する。このモデルを構成する各<概念>を形式的言語によって詳細に設計する。

さらに、大規模なシステムに対応するための DMC の拡張について、空間設計と平面設計による設計の考え方を述べる。

## 4. 仕様記述言語

形式的仕様記述言語として必要な要素としてつぎのことが考えられる。

- 理解性を高めるための、明確な計算モデルをもつ。
- 静的な検証を行うための、解析および検証可能な形式性をもつ。
- プロトタイプ的な動的な検証のための、実行可

† The Software Development Environment: dmCASE by SAEKO MATSUURA and MASAHARU OHBAYASHI (Kanri Kogaku Kenkyusyo, Ltd.).  
J. of Info. Processing, Vol. 15, No. 1 (1992), pp. 116-128

†† (株)管理工学研究所

能性をもつ。

● 実現と切り離した記述のために抽象度の高い記述が可能である。

これらの観点から関数型言語 ML を仕様記述言語として採用した。

### 5. ツールによる支援

前記の技法と言語の特性をツールとしてどのように支援するかについて、つぎの9つの観点が重要と思われる。検証・可視化・情報整理・テストデータの生成・ガイダンス・ナビゲーション・モニタリング・文書化・柔軟性。

### 6. dmCASE の概要

dmCASE における設計プロセスは、システムモデル構築プロセス・シミュレーションプロセス・変換プロセスの3つのプロセスから構成される。

また、dmCASE はつぎの6つのサブシステムから構成される。

- 記述系：DMC に従った仕様の記述を支援する。
- 静的解析系：記述過程における整合性や一貫性を検証する。

証する。

● モニタ：記述過程における矛盾や、実行状況等の監視を行う。

● 動的解析系：仕様を実行して検証する。

● 変換系：仕様を効率の良いカテゴリカル・コンピレータ・コードに変換し、実行する。

● 知識ベース：さまざまな設計情報を管理する。

本節では、dmCASE を形成する様々な概念<である単語表・概念構造図・インスタンス定義表・構文誘導型エディタ・メモ帳・カード・カードパネル・回路図・スコープ・回路図マップについてその役割を説明する。また、dmCASE における操作はほとんどマウスのみによる一貫した操作である。

このように、ソフトウェア開発においてより知的な作業のみ従事するために、dmCASE は設計方法論・形式的仕様記述・支援ツールの統合化された環境を目指している。

最後に、記述実験による dmCASE の評価および今後の課題について、記述性・仕様の読みやすさ・ユーザインタフェースの観点で議論する。

## 欧文誌掲載論文要約

<情報処理学会論文誌からの翻訳>

### 有限オートマトンと表現等価な正則時相論理とその論理設計検証への応用†

平石 裕 実††  
藤 井 寛†††

浜 口 清 治†††  
矢 島 脩 三†††

超大規模集積回路技術の進展に伴い、設計対象となる論理回路の規模も益々増大化し、設計段階において

誤りの生じる可能性が高くなってきている。設計誤りが生じると、製品のコストや開発期間等に重大な悪影響を及ぼすため、設計の正しさを保証するための論理設計検証手法の確立が緊急かつ重要な課題となっている。

従来、論理設計の正しさを確かめるための一手法として、論理シミュレーションによる方法が広く用いられているが、論理シミュレーションではシミュレーションで用いた入力パターン以外の入力に対する設計の正しさを保証することはできない。このため、設計の正しさを保証するためには、何らかの論理体系を用いて設計の正しさを証明する必要があり、設計対象と

† Regular Temporal Logic Expressively Equivalent to Finite Automata and Its Application to Logic Design Verification by HIROMI HIRAIISHI (Department of Information and Communication Sciences, Faculty of Engineering, Kyoto Sangyo University), KIYOHARU HAMAGUCHI (Department of Information Science, Faculty of Engineering, Kyoto University), HIROSHI FUJII (Information and Communication Processing Research Laboratory, NTT) and SHUZO YAJIMA (Department of Information Science, Faculty of Engineering, Kyoto University).

*J. of Info. Processing*, Vol. 15, No. 1(1992), pp. 129-138

†† 京都産業大学工学部情報通信工学科

††† 京都大学工学部情報工学科

†††† NTT 情報通信処理研究所基本アーキテクチャ研究部



なる論理回路の仕様や設計を記述した設計検証を行うための論理体系が必要である。

従来より、論理回路の一部である組合せ論理回路に対応する論理体系として命題論理が知られているが、一般的な論理回路である順序回路やその数学的モデルである順序機械や有限オートマトンにちょうど対応し有限個の論理記号で構成できる論理体系は明らかにされておらず、このような論理体系の構築が論理設計検証の分野を始めとして並行プロセスやプロトコルの検証、ひいては数理論理学の分野においても一つの重要な研究課題となっていた。

設計検証のための論理体系として、時間の概念を陽に表現することができる時相論理は、近年プロトコルや論理回路の形式的設計検証の分野において注目されており、実用的な設計検証システムの研究開発も試みられているが、これらのシステムで用いられている時相論理は表現能力が低く任意の有限オートマトンの設計仕様を記述することはできない。そこで、時相論理の表現能力を拡張するいくつかの試みが行われ、拡張時相論理 ETL や区間時相論理 ITL などが提案されている。しかしながら、ETL では、ある有限オートマトンの仕様を記述する場合にはそのオートマトンに対応した時相演算子を用いる必要があるので、一般に任意の有限オートマトンの仕様を記述するためには無限個の時相演算子が必要となる。一方、ITL は表現能力が高すぎて充足可能性判定問題が決定不能となる

ため、設計検証のための論理体系として用いることはできない。

そこでわれわれは、このような観点から有限オートマトンにちょうど対応する論理体系として、有限オートマトンの入出力系列（動作系列）の集合と考えることのできる正則集合に着目し、正則集合やその部分クラスである  $\epsilon$  フリー正則集合、また無限長の系列も考慮した  $\infty$  正則集合や  $\omega$  正則集合などに対応するいくつかの時相論理体系（正則時相論理）を提案してきた。正則時相論理は、命題論理に時間を表現するための時相論理記号として、「次の時刻」、「次の区間」、「繰り返し」を意味する三つの時相論理記号を加えて構成されており、比較的単純な構成ではあるが有限オートマトンと等価な表現能力を持っている。したがって、正則時相論理を用いることにより、設計対象である論理回路が満たすべき設計仕様を、その論理回路の入出力の関係として完全に記述することが可能になる。

本論文ではこれらの正則時相論理の体系を整理するとともに、正則時相論理を実際の論理設計検証問題に応用するために、設計された有限オートマトンが正則時相論理で記述された設計仕様を満たすことを証明する方法として、モデルチェックと呼ばれる効率の良いアルゴリズムを示し、このアルゴリズムに基づいた論理設計検証システムを実際に試作し、状態数が数百程度の中規模の順序機械に対して実用的な時間で設計の正しさを証明できることを示す。