

1993 年 5 月 発刊

Vol. 13 No. 1

特集：CAD/CAM

巻頭言

特集「CAD/CAM」の発刊によせて……………稲葉 聡 1

論 文

生産の革新を実現する計算機支援技術……………木村文彦 3

次世代 CAD/CAM システム「CADCEUS」の概要……………大高哲彦 26

CADCEUS のデータ構造……………白川忠彦 42

有理 B スプライン曲線・曲面のパラメタ変換と

ウェイト操作……………中邨博之 57

形状モデリングにおける多重構造表現……………松木則夫 75

鋳鍛造金型用形状モデラ……………谷本茂樹 87

CAD のためのユーザ・インタフェースの提案……………鈴木良太郎 117

最新ビジュアルライゼーション技術の探求と

CADCEUS/CG……………大平昌明 146

境界要素法による金型冷却解析システム……………菅沼正之, 平林 繁 166

住宅 CAD 図面の色づけに関する一手法……………前原 要 180

プレス金型の形状設計 CAD……………沢田晃二, 井上博順, 前田幸久 193

パッケージ紹介…………… 210

新製品紹介…………… 221

掲載論文梗概……………表 2, 3

種々の変化に対する高度な柔軟性や人と機械との共生、地球環境を考慮した最適生産等の生産に対する新しい要求が生じ、生産システムやその基礎となる生産技術は急速に変貌しつつある。木村文彦は、生産の革新を実現する計算機支援技術の中で、仮想生産の考え方を中心として、生産革新を実現する計算機支援技術を概説し、仮想生産の基礎となる生産に関わる各種のモデルの中でも重要で研究開発が進んでいる生産対象物モデルの内容・実現方法を論ずるとともに、同モデルデータを中心として、STEP等の標準化について考察している。

CADCEUSはメカニカル分野全般を対象とした、初期設計から生産準備までの製品開発工程を総合的に支援するCAD・CAM・CAE、およびCGの統合システムであり、製品モデル・形状特徴・非多様体・幾何拘束/推論等の先端技術を先取りして実装することにより、設計・生産の大幅な品質向上、工期短縮を狙いとしている。大高哲彦は、次世代CAD/CAMシステム「CADCEUS」の概要の中で、CADCEUSのシステム構成、主要な概念、特徴的な機能について概説している。

次世代CAD/CAM実現のためのデータベースは、製品モデルの実現、データの一元化、ハイブリッドモデルの実現、設計・製図自動化の支援、並行設計の支援等の機能を備えている必要がある。CADCEUSは、これらの要請を踏まえて、ワークスペース、オブジェクト、幾何要素・位相要素等形状を表現する要素、製品属性等の要素、形状特徴・パラメトリック設計のための要素、CAM・CAE・CG等個々の応用独自の要素、及びこれらの要素間の関係等により実現している。白川忠彦は、CADCEUSのデータ構造の中で、これらのデータ構造と特徴、利点等について説明している。

NURBS(有理B-spline)曲線・曲面は、多項式B-spline曲線・曲面にはないウェイトと呼ばれる自由度を持っている。制御点を動かさずにウェイトを変化させると、一般には曲線・曲面の形状は変化する。しかし、NURBSを用いた曲面創成

に際しては、稜線の形状を変化させずにウェイトを操作する必要が生じる。中邨博之の有理Bスプライン曲線・曲面のパラメタ変換とウェイト操作は、曲線・曲面の形状を保つようなウェイト操作の方法を調べ、その結果としてある条件を満たすパラメタの区分的一次分数変換を施してもNURBS曲線・曲面の制御点は変化しない事を示し、その際のウェイトの変換式を与えている。

設計・製造過程を支援するCADシステムは広く普及しているが、依然として多くの問題が残っている。松木則夫の形状モデリングにおける多重構造表現は、これらの問題が形状の形状モデル表現の単一性に由来していると考え、多重表現の必要性を論じるとともに、この対策として、形状を部分形状である擬立体のネットワークで表現し、製品のversion管理、design variationの表現に効果的で、形状特徴モデリングへの適用が可能な多重化したデータ構造QNS(quasi-solid network structure)を提案している。

鋳鍛造金型用の形状モデラには、曲面と立体を同一のデータ構造で表現できることや、分割結合演算、フィレット掛け、形状復元等の機能を持つ事が要求される。筆者らが開発したモデラは、これらの要求を満たし実用システムとして十分な安定性と効率の良さを実現した。谷本茂樹は、鋳鍛造金型用形状モデラの中で、鋳鍛造金型の形状と形状生成手順の特徴、鋳鍛造金型の形状を計算機上に生成するための形状モデラに対する要求、この要求を実現する上での問題点、開発した形状モデラのデータ構造と機能について述べている。

鈴木良太郎のCADのためのユーザ・インタフェースの提案は、最近のユーザ・インタフェースの動向の分析とともに、その根底をなすコミュニケーションのあり方について考察し、それを元に、CADシステムにおけるユーザ・インタフェースのあり方について検討を行っている。さらに、検討結果の部分的検証モデルとして考案したICANモデルをCADCEUSのスケッチ機能にインプリメントし、同モデルの利点を検証している。

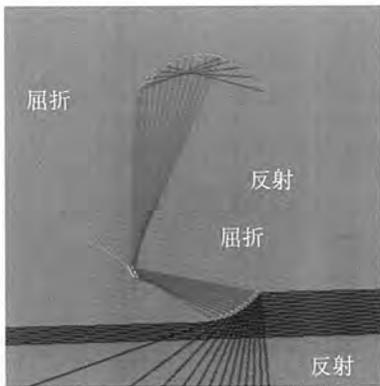


口絵1 面塗りの家



口絵2 レンダリングした家

詳細は、本文「最新ビジュアルライゼーション技術の探求とCADCEUS/CG」に掲載。



口絵3 反射と屈折する光線



口絵4 容器



(a)アンチエイリアシング(ギザギザが目立たなくする方法)をしていない。 (b)アンチエイリアシングをしている。

口絵5 電話機

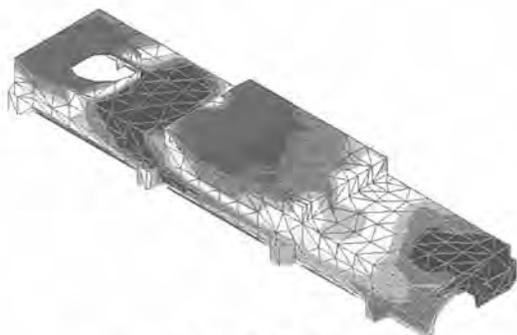


口絵6 反射

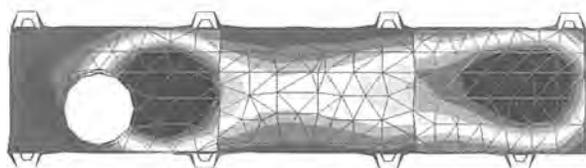


口絵7 屈折

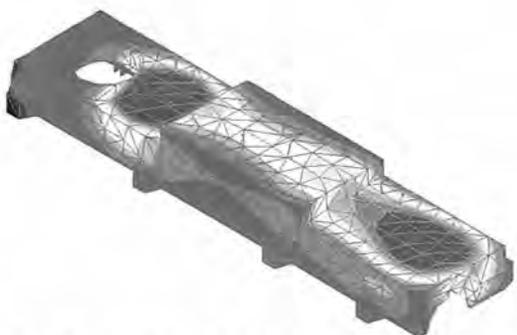
詳細は、本文「最新ビジュアライゼーション技術の探求とCADCEUS/CG」に掲載。



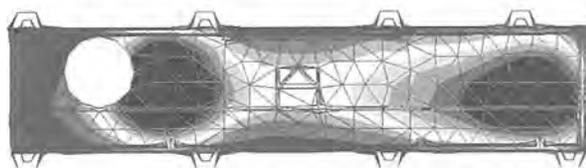
口絵 1 流動解析温度分布 鳥瞰図



口絵 2 冷却解析温度分布(冷却管 6 本)上面図



口絵 3 冷却解析温度分布(冷却管 6 本)鳥瞰図

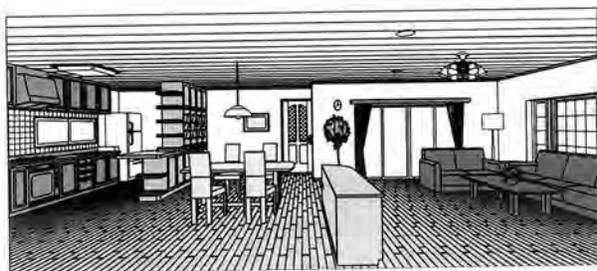


口絵 4 冷却解析温度分布(冷却管 6 本)下面図

詳細は、本文「境界要素法による金型冷却解析システム」に掲載。

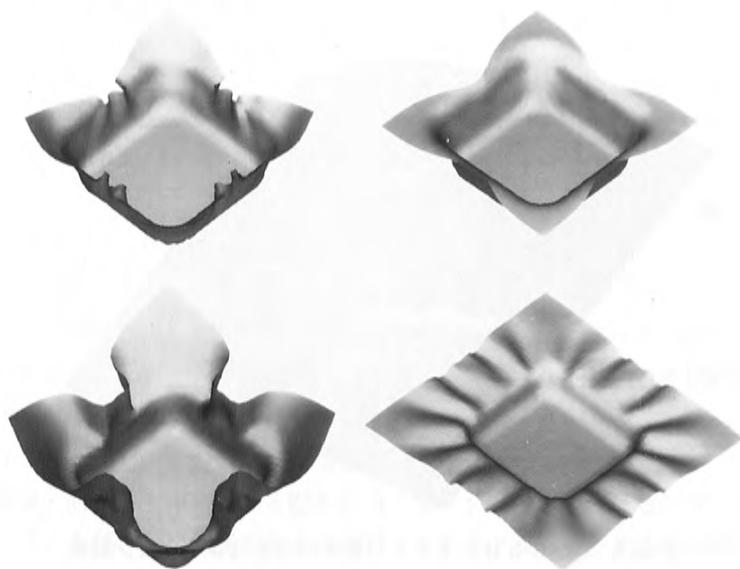


口絵 1 視点の変更

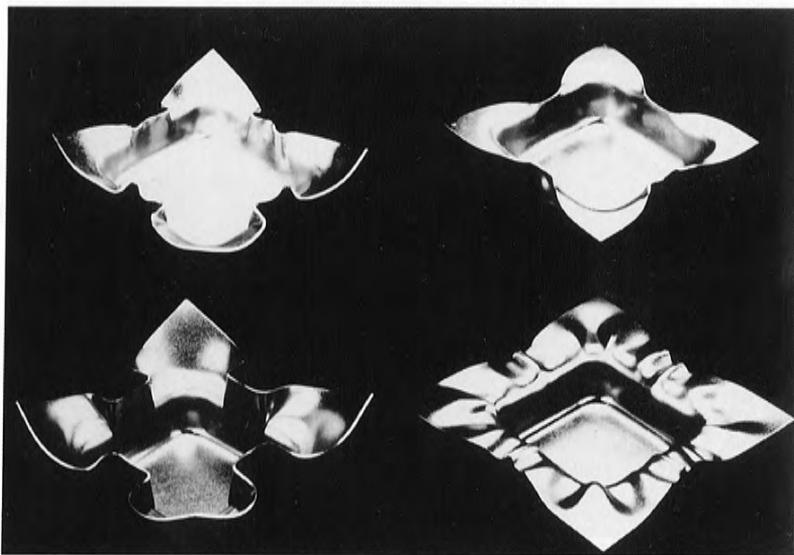


口絵 2 色調の変更

詳細は、本文「住宅CAD図面の色づけに関する一手法」に掲載。

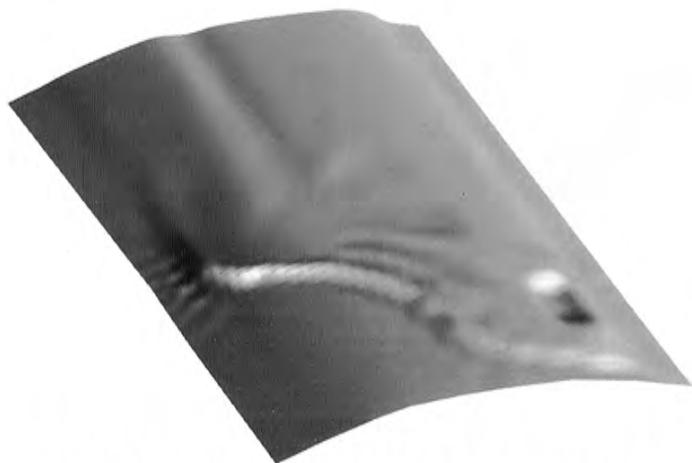


口絵1 ITAS-3Dによる四角筒成形のシミュレーション結果

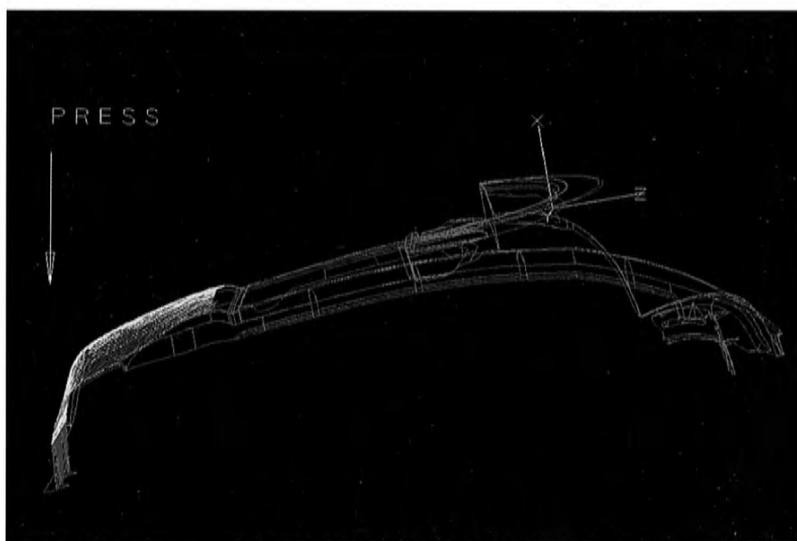


口絵2 四角筒成形の実験結果（口絵1と同一条件での実験）

詳細は、本文「プレス金型の形状設計CAD」に掲載。



口絵3 ITAS-3Dによるシワ発生シミュレーション結果



口絵4 自動インパース・チェック

詳細は、本文「プレス金型の形状設計CAD」に掲載。

特集「CAD/CAM」の発刊によせて

稲 葉 聡

製造業において CAD/CAM システムは、すでに定着したコンピュータの利用技術であるといえる。日本ユニシスにおいても、汎用 CAD/CAM システム・UNICAD を出荷開始してから 10 年余りになり、多くの機械系のアプリケーションに活用されている。

この 10 年間に、CAD/CAM の要素技術は目覚ましく進展してきた。

たとえば、曲線・曲面・立体に関する図形処理とデータベースとによるモデリング技術や、グラフィック・ディスプレイに代表されるコンピュータ・グラフィックスの技術等が挙げられる。この間、EWS (Engineering Work Station) に見られるように、ハードウェアの進歩も目覚ましく、費用対性能比も大幅に向上し、複雑な図形処理計算を高速に処理できるようになってきた。

しかしながら、製造業の大命題である製品開発期間の短縮、品質の向上に十分対応できているとはいえない。物の製造には、形状情報に代表される各種の技術情報や、情報の流れを管理する各種の管理情報等、まだまだシステム化すべき情報が多くある。また、設計・生産の活動においても CIM やサイマルテイニアス・マニュファクチャリング (Simultaneous Manufacturing) やコンカレント・エンジニアリング (Concurrent Engineering) が唱えられて久しいが、実現には課題も多いし、まだまだ時間が必要と思われる。製造活動の革新を支援するには、CAD/CAM システムも従来の路線の延長ではなく、次世代を見据えた斬新さが要請されている。

このような状況を背景に、日本ユニシスは、約 3 年前から新しい CAD/CAM システム「CAD-CEUS」を開発してきた。未だその全体の開発は完了しているわけではないが、本特集でその技術の一端を紹介する。

特集にあたり、まず、東京大学 工学部 精密工学科 木村教授に、今後の製造のあり方について展望して頂き、つぎに「CADCEUS」を紹介する。CADCEUS は日本ユニシスの総力を挙げて開発中であり、一部機能はすでに出荷されている。ここでは、CADCEUS の概要と、中核となる「プロダクト・モデリング」機能を実現するデータ構造や図形処理について紹介する。また、日本ユニシスのもう一方の CAD 商品である住宅 CAD、および CAE、CG の分野における新しい試みについて紹介すると共に、CAD/CAM の要素技術としての、図形処理やモデリング技術、ユーザ・インタフェースについて述べる。

また、日本ユニシスは多くのお客様と CAD/CAM システムの開発を行っており、そこでの成果の一部についてもここに紹介する。

製造業のニーズは多種多様であると共に、最先端技術の取り組みが常に要求されるが、こ

2 (2)

に紹介する新しい技術は、その対応の成果の一端である。日本ユニシスは、今後もお客様の革新を支援できるようなシステム開発に最大限の努力を続けてお役に立ちたい。

最後に、執筆頂いた木村先生に厚くお礼を述べると共に、本特集が少しでも読者のお役に立つことを願うものである。

(製造工業システム本部 本部長)

生産の革新を実現する計算機支援技術

Computer Aided Technology for Realizing Manufacturing Innovation

木村文彦

要約 種々の変化に対する高度な柔軟性や人と機械との共生, 地球環境を考慮した最適生産等の生産に対する新しい要求が生じ, 生産システムやその基礎となる生産技術は急速に変貌しつつある。本稿は, 仮想生産の考え方を中心として, 生産革新を実現する計算機支援技術を概説するものである。まず, 将来の生産革新の動向を概観してから, 計算機支援技術の現状と問題点を整理し, コンカレント・エンジニアリングを中心にこれからの研究開発の方向を探る。生産の技術的活動を徹底的に計算機内にモデル化し, 生産過程をシミュレーションして製品機能や生産性を事前評価しようとする仮想生産の基本概念を明らかにし, その実現のために必要となるモデル群を考察する。生産活動モデルや生産対象物モデルが重要であるが, とくに研究の進展している後者についてその内容や実現方法を論ずる。最後に生産対象物モデルデータを中心として, STEP等の標準化について考察する。

Abstract Manufacturing systems and technology are rapidly changing due to new requirements for higher manufacturing flexibility with respect to changes of social and human resources, human-machine symbiosis and global manufacturing optimization. In this paper, computer aided technology for realizing such manufacturing innovation is discussed. Technological perspective for future manufacturing innovation is briefly explored, and state of the art and issues about computer aided technology are summarized. Concurrent engineering is investigated as an example of future technological developments. A concept of virtual manufacturing is introduced, which models every necessary manufacturing activity in computer, simulates manufacturing behaviour and processes, and evaluates machine functionality and manufacturability as much as possible before actual production. A set of models are considered for virtual manufacturing, among which activity models and object models are important. Some of the details of object modelling are discussed. Standardization activities, such as STEP, are considered for object model data representation.

1. はじめに

工業生産において高生産性や高品質性をより一層向上させることが企業存続のために必要なことは言うまでもないが, われわれの自然や社会環境に対する影響を考慮に入れて総合的に人類の利益になるように生産の最適化を図っていくことも, 将来の生産技術に強く期待されている。工業生産と自然や社会環境との係わり合いについては, 表層的な議論は多くあるものの, 科学技術的に深く考察されておらず, われわれの知識は未だに不十分である。本稿では, 上記のような課題に対応する一つの方法として, 計算機支援技術, とくに生産活動や生産対象物を包括的にモデル化し, 実生産の前に徹底的に製品機能や生産性をシミュレーションして生産の不具合を撲滅する手法を考える。このような技術を総称して「仮想生産」と呼ぶことにする。なお, 以下では生産技術のソフトウェア的側面のみを考察の対象とする。また, 主に機械生産を念頭に

おいている。

われわれは工業生産を一定の枠にはめて見がちであるが、過去の生産技術の発展を振り返ってみると、いくつかの大きな変化を経てきたことがわかる。前工業的な時代を過ぎて、蒸気動力と分業による産業革命、作業の科学管理と大量生産、機械的自動化の時代、戦後の日本を代表とする柔軟で効率の良い生産方式（Just-In-Time とか Lean Manufacturing 等と呼ばれる。）等が思い浮かぶ。各々の時代においては当たり前前の技術もその前の時代においては夢であった。現代においても、上記のような新しい問題があり、次世代に向けての生産システム構築の新しい方式が求められている。想像力をたくましくして、夢のような生産の形態を考え実現していくことが、近年語られる生産の衰退を乗り越え、生産を魅力ある課題として復興させていくために必須のことである。

具体的には生産システムに対する要求として以下のような事柄が論じられている。

- ・変化に対する柔軟性：市場要求、製品・生産技術、人や社会等は急速に時には予期せぬ方向に変化している。
- ・人と機械との共生：人を排除する自動化ではなく、人の能力を増幅し、人と計算機の共生を実現するような仕組みが必要である。
- ・地球環境を考慮した最適生産：生産活動が地球環境に及ぼす副作用を過小評価することはできない。

上記のような要求は、従来から深く考察されてきたとは言えず、永年にわたって蓄積してきた工業生産の知識は十分とは言えない。体系的な工学知識に基づいた理論的な接近法が有用な所以であり、計算機支援技術はこれらの要求に答える中核技術として期待されている。

工業生産の分野で、計算機支援技術はすでに良く普及し先端技術の競争力維持のために不可欠な技術となっているが、上述の要求に答えるためには十分ではない。生産対象物や生産過程のモデリングには制約が多く、新規の製品や製造プロセスに対して有効な予測ができない。対応する計算機ソフトウェアは複雑であり、環境や技術の変化に応じて発展させることが困難である。さらに、計算機処理が人に透明ではなく、何か不都合なことが発生した時に人と計算機との協力が困難である。

上記のような問題点を克服するために、以下のような技術が望まれている。

- ・合理的な理論や知識に基づいた設計生産知識の体系化
- ・生産対象物や設計生産活動のモデル化
- ・製品機能や生産性の正確な事前評価
- ・生産上の不具合の事前除去

本稿では、詳細な技術的分析は行わず、問題点の指摘と可能な技術を論ずるに止める。以下、2章において将来の生産の姿を論じ、3章で技術の問題点を考え、とくに4章では日本の現状を分析する。5章で今後の技術の展開の方向を展望し、その一例として6章でコンカレント・エンジニアリングについて考える。7章で仮想生産の概念を導入し、その実現方法として8章でモデルベースの考え方を論ずる。9章では仮想生産の基礎となる生産に関わる各種のモデルを論じ、その中でも重要で研究開発が進展している対象物モデルについて10章で詳述する。11章では生産に関係する標準

化の問題を検討する。

2. 生産の革新

工学・工業の本分は「もの作り」であり、われわれの経済・社会に大きな影響を与えている。ややもすると工業生産は同じことの繰り返しのように考えられがちであるが、1章に述べたように、時代と共に大きく変貌し、また社会を改革してきた。人や地球環境とのつながりがより重視される中で、今後どのように技術を発展させていけば良いのか、興味は尽きない。包括的ではないが、思い付く範囲で生産に関わる未来社会のシナリオを展望してみたい¹⁾。

現在姿が見えている技術が実用化されるだけで、生産はどのように改革されるだろうか。

- 1) 企業活動の統合化とグローバルイゼーション……仕入先や顧客とも直結した、企業全体として統合された一つのアーキテクチャに基づいた生産情報システムのインフラストラクチャが構成される。少数の管理者に意思決定のための情報を集中させることができるようになり、企業形態が変革され、地球規模の活動が加速される。
- 2) 生産技術の標準化……製品記述情報や生産準備・管理情報等の標準化が進展し、企業内でのシステム統合化が進むだけでなく、企業間、あるいは国際間で生産情報が自由にやりとりされるようになり、国際共同開発や技術の平準化が急速に進む。生産技術が標準化されることにより、公共知識として社会において共有されるために研究開発の無駄な重複が避けられる。
- 3) インテリジェント CAD/CAM/CAE……熟練した技術者の知識の一部を知識ベースとしてシステム内に蓄積することにより、定型的な設計生産準備作業を高度に自動化することができるようになる。将来減少すると予想される熟練技術者の技術を若手の技術者に継承し、生産技術を維持、発展させるために不可欠の道具となる。
- 4) 仮想現実感の利用……製品開発におけるシミュレーション、顧客に対するプレゼンテーション、工場内作業の試行、訓練等に、仮想現実感の技術が有効に利用される。工場内設備としてだけでなく、大衆消費者向けの一般的なプレゼンテーション向けに普及すると思われ、大衆消費財の製品動向に大きな変化をもたらす。
- 5) 加工機械システムのインテリジェント化……種々の生産機械に自律化のための機能を付与し、インテリジェントな無人化を可能とする。とくにインテリジェントセンサによる加工の監視あるいは異常検知技術は重要であり、ニューロコンピュータ等の技術が導入される。同種の技術はその他の生産設備へも拡大適用される。
- 6) 注文情報のダイレクトイン……一般消費者の注文が工場へ直接入るようになり、仕様や納期の柔軟性が増し、生産が効率化される。流通業の役割は、金銭授受や商品配送から、商品企画や評価、消費者教育へと変貌していく。商品の多様化が進展し、個別生産へ近づいていく。半世紀から1世紀先の生産形態を予測することはむずかしい。われわれが現在抱いている生産の基本概念は大きく変わり、

われわれの生活を変貌させるだろう。

- 7) 仮想工場と仮想生産……製品の企画，設計から生産，保守に至る生産のすべての活動を計算機モデルにより事前評価して高品質，高効率な生産を実現する。試行が容易なため，新技術の導入や環境変化等に対する柔軟性が高い。これは，以下に述べる未来生産のための汎用的な基礎技術でもある。
- 8) パッケージ化された生産……定型化された生産技術はモジュール化され，パッケージ化されて，世界中のどこへでも転移しやすくなる。変動する生産環境に応じて，常に生産を最適に維持することができる。
- 9) 自己修復工場……徹底的に無人化された工場では，掃除，点検，修理等の機能を担うインテリジェントロボットが配置され，工場の操業を自律的に維持する。生産要求の変動に伴い自己を再構成して，普段に生産設備の形態を変貌させる。
- 10) 個別的ニーズに対応する生産……顧客が欲しい製品を個別に設計し，その設計に基づいて個別に製品を生産する。製品設計に関して素人の顧客を支援するための計算機システム，標準カタログ等の整備が不可欠である。直接的な生産の効率は落ちるが，個々の製品の使用効率は向上するので，社会的な活動の効率は向上する。
- 11) 社会基盤を再生する生産……地球資源の有限性を考慮して，必ずしも個別生産の欲求の高くないものについては，公共的な設備として社会基盤の一環として整備維持するようになる。いわゆる耐久消費財が主なものであるが，生産の役割は，社会要求に従ってこれらの社会基盤製品を普段に維持，管理し，徐々に世代交代させていくことにある。

多くの困難な技術課題が残されてはいるものの，上に論じたような事柄は計算機支援技術の立場からは技術的に実現可能なものであり，もはや夢とは言えない。より革新的なアイデアが続々と生まれてくることが期待される。

当たり前のことであるが，工業生産は，地球という有限な環境の中で持続 (Sustainability) 的に発展できなければならない。その作り出す製品と共に緊密に社会の中に埋め込まれ，直接従事する人の減少もあって，表層的にはその姿が見えにくくなっていく(農業の発展の歴史に類似性を見ることができる)。このことは，工業が社会のインフラストラクチャとして組み込まれたことを意味するのであって，工業の重要性が増しこそすれ，決して脱工業化のようにはならないことに注意せねばならない。

3. 計算機支援技術の現状と問題点

生産活動をシステム化して情報処理技術により自動化・高度化しようとする研究開発は，過去数十年にわたって活発に行われ，大きな成果を挙げてきた。現在では，その浸透の度合いに程度の差はあるものの，あらゆる産業において不可決の基幹技術となっており，将来の生産システムを考える際のインフラストラクチャとして開発を支援し，また制約する存在となっている。

生産に関する計算機支援技術の発展の経緯を振り返って見ると，現状を革新する概念の創出，その実践のための基礎技術の開発，実践結果の評価と問題点の摘出，というサイクルを長い時間をかけて回っていることがわかる。生産の技術活動支援につい

て考えてみると、現在想定されている最も進化したCAD/CAMの姿はかなり具体的に30年以上前に考えられていたし、CIMの本質を指摘した文献は25年前には存在していた。しかし、概念の理解不足と実践技術の未熟さのために技術が効果を生むには至らず、試行を繰り返してきた。それらの経験が蓄積してようやく環境技術も整い、実践の効果が出てきた、と言えるであろう。

しかしながら、CAD/CAMシステム等の先進ユーザは現在利用可能なシステムに存在する多くの欠陥にすでに気付いており、今後の長期の技術発展に対応できるようにシステムの革新に乗り出している。日本の製造業においては、技術者や現場作業者が優秀であるがゆえに、欧米に比して比較的簡単な計算機支援ツールにより非常に高品質の仕事をこなしてきた、としばしば言われている。しかし、そのような人的資源や技術的な環境は急激に変化している。現状よりも高機能な計算機支援システムが強く必要とされている。世界的に見ても傾向は同様である。1章で述べたように計算機支援技術の満たすべき目標として次の3点がよく議論されている。

- ・技術、人、社会（市場その他）等の変化に対応できる柔軟性
- ・高品質で高付加価値な製品設計生産を支援できる技術支援
- ・全プロダクトライフサイクルを考慮した生産の最適化

しかし、現在のCAD/CAM、FA、CIM等のシステムを概観してみると、大きな問題点が未解決のまま残されていることがわかる。具体的には、次のようなことが指摘される。まず第1に、手の付けやすいところや即時的な効果の上がることから導入されてきたため、生産システム全体として付加価値の高い統合化されたシステムとなっていない点が挙げられる。たとえば、製品コストに大きな影響を与える決定は製品設計の初期段階でなされるのにもかかわらず、この段階に対応する計算機支援技術はほとんどないこと等がよく議論されており、これが最近の設計重視の研究開発の動向となって現われている。

第2に、システムやソフトウェアの構成が整理されていないため、製品、技術、人、等の変化に追従して、機能を増強し、成長させていくことが困難である。往々にして、システムやソフトウェアの全面的な作り替えが必要となり、過去の蓄積が失われていく。いわゆる世代替わりとして、このようなことが行われてきたが、生産システムが高度化するにつれ、生産知識の蓄積は大きな問題となっており、世代替わりでなく、成長できるシステムが強く求められるようになってきている。

第3に、人と生産システムとの共生、融合の問題がある。単純作業の自動化は言うまでもないが、生産が高度化するにつれ、人を排除することなく、人主導で人をシステムの重要な構成要素とするシステム構成が必要となる。上記の二つの問題点とも関連しているが、従来はこの点の考察が極めて不十分であった。

以上のような問題点に対応して生産システムのソフトウェアを改革していくためには、生産活動を分析して統合的にモデル化し、共通の基盤となる知識や手法を計算機上のソフトウェアツールとして体系化し、生産システム構築のための情報処理のインフラストラクチャを整備することが重要である。このような目標を満たすために追求されているいくつかの研究開発の新しい方向が認められるようになってきた。その主要なものは以下のようである。

の CAD/CAM システムや汎用の二次元 CAD/CAM システムが広く導入され、また特有の技術要求を持つ先進企業は、固有の要求に対応できる自社製のシステム開発に多くの努力を投入した。造船業界と自動車業界が自社開発グループの代表である。1980年代の初めまでには、主要な自動車会社はボディ外板設計から型加工に至る一貫設計生産支援システムを稼働させていた。それらのシステムは現在に至るまで使い込まれて、極めて高品質な製品設計生産を可能としている。

一方、前章に述べたように生産に対する要求が厳しくなるにつれて、CAD/CAM システムの先進ユーザは、現用システムに対する多くの欠陥を感じるようになり、システムの根本的改革を考え始めた。次のような点が改革の目標として考えられている。

- ・統合されブラックボックスのようになったシステムからモジュラ化され開放されたシステムへ
- ・単純業務支援から付加価値の高い複雑な業務支援へ
- ・自社開発から Buy-and-Make(利用可能なものは買い、個有のものを作る)式の開発へ
- ・メインフレームベースから分散ワークステーションベースへ

たとえば形状モデリングについては、すべてを基礎から開発するよりは、一般的には市販の形状モデリングパッケージを利用することが有利になりつつあり、自社開発の努力は、プロダクトやプロセスモデルのように各社固有の要素に集中させることが可能になりつつある。

日本の企業の現状は一般的に見ると、やはり比較的単純な計算機支援システムを用いながらも、固有の高度な技術ノウハウを個別に計算機化して既存システムに付加したり、熟練した技術者や作業者の助けにより、極めて高水準の成果を上げている、とすることができる。このように多くを人に依存した体質においては、いかにして体系化されていない技術ノウハウを次世代の人やシステムに伝えていくか、ということが大きな問題となってくる。

計算機支援技術が普及している代表的な業界として、日本の自動車工業は、高度な三次元曲面モデリングに基づく CAD/CAM システムを実用化することにより、急激に仕事の仕方を変革し、製品開発のリードタイムを減少させてきた。たとえば、自動車ボディの意匠設計プロセスやプレス型設計製作プロセスは革新的に改善された。この両者の例において、従来は物理的実体であったマスタモデルは計算機内のモデルに置き換えられ、すべての仕事はより迅速にかつ正確に行われるようになった。この手法が、日本の自動車工業界が効率良く新車を開発できる要因の一つとなっている。その中核となる形状モデリングシステムを初めとして、これらのシステムは完全に国内の技術で開発されたもので、同様の海外のシステムと比べても遜色のないものである。これらの他にもいくつかの優れた CAD/CAM 要素システムが開発、実用化されている。

1980年代の後半になって、日本の企業、とくに電子工業やメカトロニクス分野では、CIM の概念が一般的となり、真に実用的な取組みがなされるようになってきた。しかしながら、すでに実用化された多くの例では、管理業務の効率化や製販一体化に力が注がれており、製品自体やその製造技術は比較的単純なものに止まっている。製造技

術をきちんと取り込んだ CIM は依然として未開拓であり、今後の大きな課題である。

5. 研究開発の新展開

前章で触れたように、自動車産業のような CAD/CAM システムの先進ユーザは現状のシステムに対して重要な問題点を認識し始めており、次世代へ向けてシステムの革新を模索し始めた。技術的な要求のいくつかは以下のようにまとめられる。

- ・コンカレント・エンジニアリングに代表されるような、製品生産の全過程を通じた技術活動の柔軟な統合化
- ・全技術情報の透過的な管理と容易で有効な利用の手法
- ・計算機シミュレーション等による、実物によらない、強力な技術検証
- ・多人数の協調作業を効率的に実現する計算機支援の協調作業環境
- ・生産性や機能の事前評価を十分に行った高品質で頑健な製品設計

前章であげたシステム改革の目標は、以上のような技術革新を実現するための条件をシステム構成の水準から表現したことになる。次世代システムをユーザに対する機能から見た時の特徴として、次のような点が最も重要であろう。

- ・モジュラ化され、分散、開放構成で再編成可能なシステム
- ・利用者によるカスタム化
- ・プロダクト、プロセス、ファクトリ等のような、ユーザの水準での高機能モデリング

最近では、モジュラ化された計算機支援システムの実用化例もいくつか見られるようになってきた。プロダクトモデリングは、この構成の中核をなす基本機能である。このような新しいシステム構築に対応できる要素技術が開発されつつある。非多様体位相構造を可能とする形状の集合演算に関する新機能等がその一例である。定型化された設計等、一定手順の仕事効率良く実行できることは大切であるが、そのような機能をユーザが自在に構築できるようなカスタマイゼーションの道具も整備されつつある。計算機による技術検証も重要であるが、その一例として自動車ボディーパネル成形への適用等を目的とした薄板の深絞り成形シミュレーション等が試みられている。実験と併用することにより、自動車の高品質パネルの製作に不可欠の技術となりつつある。このような新しいシステムの構築を、基本から行うか、既存のシステムを段階的に拡張するかは、むずかしい技術的決定である。

6. コンカレント・エンジニアリング

コンカレント・エンジニアリングは、日本の工業界では良く知られた重要な手法で、計算機支援技術とは結び付いてはいないが、生産性向上のために日常的に実践されている。製品設計情報を未確定のまま後工程の仕事に引き渡し仕事の流れを加速したり、先行的な生産準備作業により生産上で起こりうる不都合を迅速に製品設計へ反映させたりすることが、その一例である。将来的には、これらの人に依存した方法を計算機支援の方法に改変していくことが強く望まれている。コンカレント・エンジニアリングは、上記のような新世代のシステムへの要求条件を考察する際の重要なガイドラインとなるが、また逆に新世代システムは計算機支援されたコンカレント・エンジニア

リング実現のための重要な基礎技術となる。

コンカレント・エンジニアリングは、最近良く議論され、一般的になってきた言葉であるが、はっきりした技術を指す言葉ではなく、その定義は判然とはしない。しかしおおよそ次のようなことを指すと理解されている^[2]。前章で述べたような生産システムに対する要求に答えて、適切なコストと品質を持った製品をできる限り早く生産するために、生産活動全体の管理方式を、従来のような厳密な分業方式による順次実行方式から、より柔軟な仕事の区分に基づく並列実行方式へ変換しようとするのである。

従来の生産方式では、製品の企画から始まって設計、生産準備、検査、保守、等の仕事ははっきり区別され、専門の要員によって独立に効率良く実行されるように考えられていた。しかし、ごく定型的な製品を扱う場合を除いて、実際にはこれらの仕事の間には密接な関連があり、相互に柔軟に通信し合いながら仕事を進める必要が出てきた。設計変更が製造方式に重大な影響を与えることが設計者には良く評価できないとか、製造方式の変更が製品機能を損なう恐れがあることが製造現場ではわからない、等はその一例である。製品競争力を高めるべく、製品機能や製造方式に技術の極限を求めると、従来予期できなかった設計と製造の双方に密接に関連する技術的問題が頻発する可能性がある。逆に、設計と生産準備が隔離されていて、生産の最適化が十分に行われていないと製品競争力が失われる。

上記のような背景から、最近コンカレント・エンジニアリングが強く推進されるようになってきた。具体的な実践の形態は以下に述べるが、その概念的な形態は次のようになるであろう。まず、従来の仕事区分を保ちながらその相互間の通信を柔軟にして部分的に並列実行できるようにすることが考えられる。設計が完了する前から生産準備へ情報を渡して仕事を先行させること等がある。もっと進んだ形態としては、生産に必要な諸活動を見直し、製品や製造技術に関する情報の流通を良くして、個々の専門家が自分の仕事を独自に進めながら、同時に他人の仕事との不整合を起こさずに、生産の最適化を達成できるような仕事の管理の枠組みが考えられる。このような新しい枠組みについての情報処理的な検討は十分になされていないが、要求条件が文献^[3]に議論されている。たとえば、生産に関する種々の「変化」の管理、情報の共有、リスクと不確実さの管理、人的資源の養成、等が挙げられている。

コンカレント・エンジニアリングは、最近良く話題にのぼるが、実際にはすでに実践されてきた使い込まれた技術であるとも言えるであろう。製品開発に際して営業、研究開発、技術、生産等の各部門から人材を集めて開発チームを構成し、製品が市場に出るまで一貫してチームに開発の責任を負わせる方式はわが国では珍しくない。しかし、ここであえてコンカレント・エンジニアリングと呼ぶ意義は、従来は人と人との柔軟なコミュニケーションに頼っていたものを、計算機支援方式により画期的に合理化しようとするのであろう。従来の技術情報の表現や流通を単純に計算機情報により置き換えてしまうと、従来は明示的には意識されていなかったが人の柔軟な情報交換能力によりやりとりされていた情報が流通しなくなり、計算機導入によりかえって情報処理が硬直化し、不便になる恐れがある。現場での仕事の実態をよく把握して慎重に情報化を進めることが肝要である。

コンカレント・エンジニアリングの実践例を見てみよう。まず関係する部門から人を集めて開発チームを構成する。仕事の進展に合わせてチームリーダーや仕事の場所、やり方を変えていく。開発に際してボトルネックになるところを迅速に検出し、チーム内ですべての決断を下してボトルネックを解消していく。必要に応じて設計技術者や生産技術者が相手の部所に駐在して不足な情報を補い、仕事を加速する。具体的には次のようなやり方で仕事が効率化される。

- ・設計や生産準備等の各種の仕事を並列的に進める。
- ・たとえ完全ではなくても、次のステップの仕事へできるだけ早く情報を引き渡す。
- ・後流の仕事で何か不都合が起きないか、できるだけ事前に検討して後戻りを防ぐ。
- ・個々の仕事の効率をできるかぎり向上させる。
- ・顧客や下請けとの連絡を緊密にして不都合を迅速に解決する。

上記のような事柄を実現するためには、技術上の問題点の他に組織、管理上の問題点を解決する必要がある。たとえば以下のものである。

- ・責任の所在や承認の手続き
- ・部門間の文化やモラルの相違
- ・情報の一元化と共有
- ・環境変化に対する対応

従来は、これらの問題は人と人との問題として解決されてきた。しかし、計算機導入により一層コンカレント・エンジニアリングの実を上げようとするならば、包括的な情報処理の枠組みを基に、上記の問題を情報処理の問題としてきちんと定式化してその解決方法を探っていかなければならない^[4]。システム管理、情報共有、ヒューマンインタフェース等、いろいろな課題があるが、以下では、仮想生産に絞って技術的課題を検討する。

7. 仮想生産環境

前章までに議論したように、生産活動の徹底的なモデル化は高機能な計算機支援システムを構築するための手法として極めて有力である。本章では生産活動のモデル化に基づく仮想生産の概念を追求する^[5]。

以下の議論のためにいくつかの術語を導入しよう。与えられた製品要求を物理的に実現可能な製品に変換するために必要となるすべての活動をプロダクトリアリゼーションと呼ぶことにする。これはほとんど生産活動と同じ意味であるが、製品に関する曖昧な情報を確定した製品に仕上げるための全活動、すなわち製品の実体化の観点から関連するすべての活動を体系的に把握しようとすることを強調している。プロダクトリアリゼーションは、技術的な事柄以外にも多くの事柄を含み、さらにソフトウェア的な事柄だけでなく物理プロセス等もその範疇に入るが、ここでは技術的なソフトウェアに限定して考察を進める。プロダクトリアリゼーションの研究のためには、生産に係わる技術活動を形式的に徹底的にモデル化してみることが一つの有力な方法である。これをモデルベースの方法と呼ぶ。

プロダクトリアリゼーションに関わる多くの活動を計算機可処理な形式でモデル化できれば、実世界の処理に頼ることなく、計算機内のモデルの世界だけで生産工程のシミュレーションが可能となる。このような概念を仮想生産と呼ぶことにする。むしろ現実の生産においてはすべての活動を計算機ベースのモデルで置き換えることは実際的ではなく、何らかの試作作業は必要となる。仮想生産を背景とすれば、この試作作業を画期的に加速することが可能となり、これを迅速試作と呼ぶ。

以上を要約すると以下ようになる。プロダクトリアリゼーションのプロセスを注意深く分析することにより、それらのプロセスを計算機ベースの仮想生産プロセスに置き換えることができ、それにより製品の機能や生産性が実生産に先立って効率良く事前評価できることになる。しばしば、仮想生産は計算機支援された迅速試作により補完され、この両者は不可分のものである。プロダクトモデルやプロセスモデル等は仮想生産を実現するためのソフトウェアツールと位置付けられる。コンカレント・エンジニアリングは、生産活動の流れの体系化の一手法であり、上記のツール群を用いて効果的に実現される。仮想生産の概念を図2に示す^[6]。

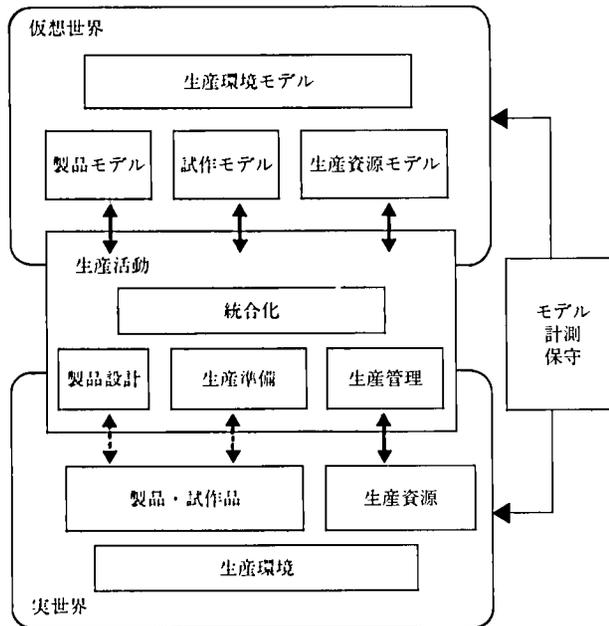


図 2 仮想生産の基本概念

Fig.2 A concept of virtual manufacturing

プロダクトリアリゼーションの精神に則って、生産活動を詳細に分析し、過去の技術蓄積を計算機可処理な形式に体系化することにより、仮想生産や迅速試作のためのソフトウェアツールを実現していくことができる。徒に一般化することは避け、現実の作業を分析し、明示化されていない膨大な技術ノウハウや知識、データ等を可視化して、そこから中核となる基本技術の体系を抽出してることが望まれている。

中核的な技術知識とは、特定の企業や製品に依存せず、誰でも自然に利用しているような重要で基本的な知識である。従来はこのような一見明示的な技術知識が体系的

に記述されていないがために知識の取得や伝承が困難になることがあった。もしこのような中核知識が計算機可処理な形式で多くの社会で共有され利用されるようになれば、個々の企業は自社固有技術の開発に専念できることになり、高度な生産が高効率に実現できることになる。

上記のような中核技術を体系化することにより、先進技術の効率的な開発や成熟した技術の健全な拡散が可能となる。以下では、そのような中核技術を構成する種々のモデルに焦点を当てて検討を進めよう。まず、以下のような点に注目することが必要である。

- 1) 広範囲な技術情報の表現と処理……表層的な技術的手続きや処理を計算機的に体系化しようとする、すぐに種々の技術知識が必要となることがわかる。そのような技術知識は種々雑多であり、教科書的なものから先端的なものまで、また製品技術から生産技術まで、極めて広範である。このような技術知識を十分に蓄積し、利用できる仕組みが必要であるが、現在の人工知能や知識工学が提供してくれる技術は十分ではない。
- 2) 環境変化に対する適応性……生産システムを取り巻く環境条件は不断に変化する。技術の発展、人や社会の変化、その他の生産資源の変化、等さまざまであり、これらの変化に追従できない生産システムは競争力を失って、消滅していく。中核ツール群がこのような環境変化に対応して自己を成長させていくことができることは、ことのほか大切である。
- 3) 人と生産システムとの共生・融合……単純作業の自動化は言うまでもないが、生産が高度化するにつれ、人を排除することなく、人主導で人をシステムの重要な構成要素とするシステム構成が必要となる。人と計算機の高度な協力を可能とするためには、計算機の処理やデータが人に透過的に良く見え、理解できることが必須であり、このことは上記二つのテーマと密接に係わっている。

8. モデルベースの考え方

生産に係わるあらゆる事柄を計算機内にモデル化し、そのモデルに基づいて人と計算機とが協力しながら仕事を進める、というモデルベースの考え方を基礎として、仮想生産の技術課題を検討していくことにする。

生産情報として考えるべき情報は極めて多岐にわたるが、ここでは技術的、管理的情報を主体として考え、経営的、その他の情報は考えない。技術的、管理的情報を包括的にモデル化するために、天下りだが、モデルをいくつかは大分類して考察することにする。このような分類はもとより考察の便宜のためで、明確に分類できるものではない。技術の見方やその利用の方法により、同一の技術も異なった形式にモデル化されることがある。むしろ、それを前提として柔軟で変換可能な技術知識表現形式を考察することが重要である。

具体的にモデルを考える前にモデル化の意義と問題点を検討しておこう。

モデル化の意義として、大きく次の二つを考えることができる。

- 1) 現用技術の高精度なモデル化……ボトムアップ的に現用技術を整理、分析して、それらを高精度にモデル化する。これはある程度確立された技術であるから、技

術自体としての新規開発は必要ではなく、体系化、標準化の努力が主体となる。これを行うことにより、現用技術をより効率的に運用でき、その改良も容易になると共に、これを基盤技術として他へ技術転移していくことも可能となる。技術の様相として、Post-Competitiveなものとも見ることができる。従来のモデル化はこの視点が大部分であった。

- 2) モデル主導で革新的な生産システムの構築……トップダウン的に革新的な生産システムの構成を考え、それを実現するために必要な、製品技術や生産技術をモデル主導で開発していく。既存の技術をモデル化するのではなく、新規技術開発のツールとしてモデルを有効活用する。従来経験のない、超精密、超微細、超高速等の世界では人の直感や経験は余り役に立たず、理論に基づいたモデルベースの接近法が極めて有効である。これは、基盤技術の上に立つ先端技術の開発に相当し、実用化に先立つ先導的な段階では、Pre-Competitiveなものとも見ることができる。従来このようなことは余り行われていなかったが、近年注目されている方向である。

モデル化は困難な問題であるが、主要な問題点をまとめて見よう。

- 1) モデルの精度……生産に関するモデルは、大きく二つに分けて考えることができる。一つは論理世界のモデルであり、他方は物理世界のモデルである。論理世界のモデルは、設計過程における人の思考プロセスやその結果としての設計手順等であり、何が制約条件で何が基本法則であるかが明確でなく、体系化が極めてむずかしい世界である。したがって、そのモデル精度も精密に定義することがむずかしい。物理世界は、物理法則に支配されているわれわれの現実世界であり、必要に応じてモデル精度を設定し、現実を模擬していくが、現実的な複雑度のモデルを構成し操作していくことは困難な課題である。いずれにせよ、現実は無限大の情報を含んでいる。要求される精度のモデルを構成し、維持していくことは困難な課題である。
- 2) モデルの変化……環境条件が変化することにより、対応するモデルも変化させなければならない。人が積極的に変化させるだけでなく、環境情報を取り込んで環境に適合していく仕組みも重要である。
- 3) 現実との対応……日常的な道具として使いこなしていくためには、汎用的なモデルの枠組みを用いて、個々のモデルや生産環境を容易にモデル化でき、そのモデルの中身が理解しやすく、利用しやすくなっている必要がある。たとえば、モデルパラメータの設定が容易に行えるような工夫が実用化に際しては重要である。

製品やその生産環境を包括的にモデル化することは困難で長期的な研究開発課題であり、一気にすべてを解決することはできない。短期的には、ボトムアップに現実のデータを収集しつつモデルの精度を向上させ、長期的には、トップダウンにあるべき生産システムの形態を提案して、必要なモデルを構成していくことになる。理論的な研究と合わせて、現場との対応を重視して、現場と遊離しない技術開発が大切である。

9. 仮想生産実現のためのモデル

生産活動全般の流れは多種多様であるが、必要な情報をモデル化し、できるだけオフラインで製品設計、設備計画、工場設計等ができるようにして、処理の正確さと効率を向上させることが望まれている。オフラインで計画されたものを、オンライン操作に反映して、そこから発生した問題点をまたオフライン処理へフィードバックできるような体系が重要である。

仮想生産で考察する情報モデルの構成を図3のように考える⁶⁾。これは、目的システムとして生産を実行する体系ではなく、目的システムを構成するために必要なツール群の構成を示している。図4に示すようにして、個別の生産形態や製品に対応する目的生産システムが構成されることになる。図3に示したように、モデルを大きく3層に分けて考えることにする。

目的システム層は、われわれが構成して使用したいと考える生産システムモデルを表現している。汎用で高機能のシステムを考えることは困難であるから、生産システムは個別の目的に良くあった専用のシステムを利用者自身が構成していけることが望ましい。そのために、図4に示したような構成で、図3の仮想生産モデル層のモデル群を用いて、システム利用者が容易に自己のシステムを構築できるようにする。

仮想生産モデル層には、目的システム構築に有用な多様なモデル群が用意される。大別すると、生産活動モデルと生産対象物モデルとがある。生産活動モデルは、生産

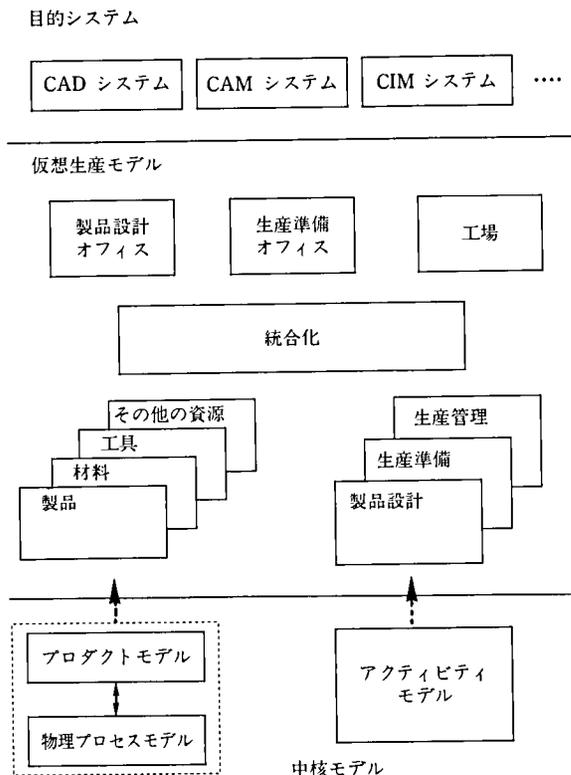


図 3 仮想生産のためのモデル群

Fig. 3 Modeling hierarchy for virtual manufacturing

メタシステム

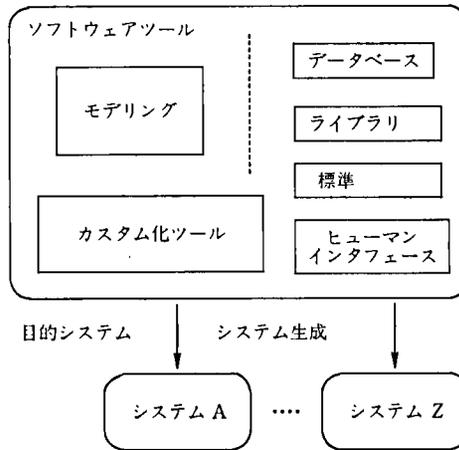


図 4 目的システムの構成

Fig. 4 Configuration of dedicated systems

に関わるあらゆる活動をモデル化したもので、計算機的に自動化されるものもあるが、多くのものは人と計算機との共同作業のモデルとなる。生産対象物モデルは、製品のみでなく、生産過程で利用されるあらゆる物的な生産資源を表現するモデルである。これらのモデルを利用して、一般的な製品設計や生産準備のモデルが構成される。これらのモデルに特定のデータや制約条件を付与することにより、目的システムが生成されることになる。

仮想生産モデル群を構成するための一般的なモデル構成要素からなる中核モデル層がある。いろいろな仕事の手順を一般的に表現するアクティビティモデルと、人工物を一般的に表すプロダクトモデルと物理プロセスモデルからなる。この中核モデル層は、仮想生産のみならず、一般的に現実世界を計算機内にモデル化する時に有効なものである。情報処理的に興味深い技術課題が沢山ある。形状モデリングや CSCW (計算機支援協同作業) 等はその一例である。

以上のモデル群についての詳しい技術的内容は明らかでなく、研究が緒に着いたばかりである。以下、仮想生産モデル層について、その内容を概観してみよう。

生産活動モデルに表現されるものは、極めて多岐にわたり複雑である。これを以下の三つのモデルに分類する。

- 1) 統合化モデル……目的とする生産システム全体の動作を規定するモデルであり、システムの柔軟性や成長可能性を規定する。大量生産から変種変量生産等の体系がその一面である。コンカレント・エンジニアリングもこのモデルにより具現化される。これらのシステムの動作を規定する一般的なモデルが考えられると便利であり、これらも含めて考える。この一般的なモデルは、以下に示す二つのモデルを記述する時にも用いられる。
- 2) 設計・生産準備モデル……設計や生産準備等で現われるあらゆる技術活動のモデルを指す。具体的には、製品企画、詳細設計、工程計画等、製品情報や生産準

備情報の生成に係わるあらゆるプロセスを表す。主に製造プロセスから見るとオフライン的な手続きで、生産情報を生成するためのプランニング的なプロセスである。物理的制約だけからは十分に制約されず、種々の条件を考えて最適化せねばならないため、一般的なモデルを考えることはむずかしく、場合に応じてモデルを考え、記述していくことになる。

- 3) 生産管理モデル……生産システムのモデルに基づき、設計・生産準備モデルにより生成された情報を利用して、生産計画を策定し、詳細な作業計画を立てて、効果的な生産管理を実現する。主に製造プロセスを制御するオンライン的な手続きを表現する。実際の物の流れは、飛び込みや故障により頻繁に変化するが、それらを十分に反映できるモデルを考える必要がある。

生産対象物モデルは、生産活動を通じて取り扱われる物理的に存在する対象物（主に人工物）を表現する。ソフトウェア生産等を考えると対象物は理論的な情報であるが、一般的には対象物はわれわれの物理世界に存在する「もの」であり、物理的な制約条件（たとえば、物体間の衝突とか、重力による落下等）により強く拘束されているのが特徴である。これを以下の二つのモデルに分類して考察してみる。

- 1) 製品モデル……広く人工物を表すモデルであり、目的製品ばかりでなく、生産資源としての工作機械やその他の設備、生産環境に存在するすべての人工物等を表す。他のモデルと比較すると、形状モデルやプロダクトモデル等として割合に研究が進んでいるモデルであるが、物理世界の制約と性質を十分に表すと言う点で、まだ多くの困難な課題が残されている。
- 2) 生産資源モデル……工場内にあるすべての生産資源のモデルを表し、生産機械、その他の設備、材料、人的資源、種々の生産技術データ等の生産知識等が含まれる。生産に係わる資産としての静的なモデルと、生産機械が利用され、稼働している状態を表す動的なモデルとがある。製品情報や生産準備情報を基に生産管理情報の指示により生産資源モデルを駆動することにより、計算機内で仮想的な生産評価が行われ、いわゆる仮想生産が実現される。内容が広範で膨大なモデルであり、未検討の課題が数多くある。

上記のような人工物のモデルの基礎としては、図3に示すように物理プロセスのモデルが重要である。

物理プロセスモデルは、製品の機能や生産加工性評価の基礎となる物理プロセスのモデルである。対象ごとに専用化された機能プロセスや生産加工プロセスは製品モデル等の属性情報となる。実用に耐えうる程度に精度の良いプロセスモデルを構築することは容易ではなく、物理プロセスシミュレーション等、解析主体の研究が多く行われているが、設計の機能評価や生産加工評価に用いるという観点からは、未解決の課題が多い。

上記のような各種のモデルは、数学や物理学のような非常に汎用的で基礎的な知識から、生産や加工のノウハウのような専門的で適用限界が明確でなく、制限されているが強力な知識まで、多様な工学知識あるいは生産知識を含んでいる。上記のモデルは、これらの知識を利用の仕方を中心としてまとめたもので、これらの知識ベースを構築する時には、またそれに適したモデルの構成を考える必要がある。上記のモデル

が計算機上の具体的なモデルとして表される時には、一元化された基礎的な知識の上に、順次適用性に応じて知識がモデル化され、応用面からはそれらが上記のように括られることになる。これが広い意味での一元化された統合的エンジニアリング・データベースあるいは工学知識ベースを構成し、設計生産作業の進行と共に現用の一時的なモデルやデータが抽象化され、データや知識が蓄積され、再利用可能になっていく。このような一元化された工学や生産知識データベースの構築法は未解決の課題であり、仮想生産全体を通じての大きな研究課題である。

10. 生産対象物モデル

生産対象物モデルは、生産活動を通じて取り扱われる物理的に存在する対象物（主に人工物）を表現する。目的製品ばかりでなく、生産資源としての工作機械やその他の設備、生産環境に存在するすべての人工物等を表す。

設計製作しようとする実物が存在しない段階において、あたかもそのものが存在するかのような仮想環境を実現して、設計評価や生産性評価を高精度高能率に遂行しようとするのが主要な目的である。人が主体で処理を行う場合の図面と同等の役割を果たすが、ここで考察しているように、計算機処理を前提として対象人工物をきちんと表現しようすると、図面に陽に記述されているよりはるかに多くの情報を記述せねばならないことがわかる。

モデルの目的により、いろいろなレベルでの表現が考えられる。定型化されたパラメトリック設計の場合には、設計パラメタのみで良く、対応する形状のモデル等、何もいらぬことがある。あるいは、加工情報生成における工作機械情報も一般的には主要なマシンパラメタだけでそれ以上は不要である。しかし、パラメトリック設計でも、図面を書こうとしたり、対象製品の重量を算出しようとする、詳細な形状モデル情報が必要になる。加工情報生成においても、工具との衝突をきちんと検出しようとする、工作機械の詳細な形状モデルが必要となる。最近では、設計や生産展開支援の要求が高度化してきており、対象形状のみではなく、物理的な機械として保持している多くの物理的属性情報をきちんとモデル化することが要求されてきている。いわゆるプロダクトモデリングである。たとえば高度な仮想生産環境を実現しようすれば、対象製品や材料、その加工や組立環境、関連する生産機械等を製品モデルとしてできるだけ完全に表現し、製品機能や生産性を忠実に計算機シミュレーションにより事前評価して、実物が存在しないのに、あたかも実物があるかのごとく、効率良く仕事を進めていける環境の実現が望まれる。

このような目的のためには、単に対象品を定義する情報だけでは不足であり、その定義に至った元になる対象品に対する要求・仕様等、定義（設計）途上で必要となるさまざまな技術的決定、対象品の物理的その他の性質・振る舞い、機能等、多様な情報を表現する必要がある。実物の代わりとして、計算機内モデルだけでできるだけ必要な技術的評価ができるためには、要求に応じて無限に多くの対象品情報を表現することが必要になるが、これは実際的には困難である。基本情報を表現しておき、あとは要求に応じて随時モデル情報を増強できるようなモデリングの枠組みを用意することが現実的である。

このようなモデルが実現できれば、単に設計生産問題だけでなく、一般の科学技術問題の解決にも極めて有用であり、また物理世界と係わりのあるような日常的な計算機課題を解く際にも、問題解決のインフラストラクチャの要素として不可欠である。波及効果の高い技術であると言える。

未だ現実に存在しない新規なものを創出することが設計生産作業の目的であるので、そのために仮想世界構築の基本要素となる製品モデルの必要性はほとんど自明であるように思われる。その重要性のゆえに、過去に多くの研究が行われてきたが、十分ではなかった。その理由として、対象分野の技術や計算機化の技術の未成熟をあげることができるが、それと共に製品モデル概念が十分に良く理解されていなかったこともあげられる。近年の設計生産自動化に関する膨大な研究の結果、高性能な自動化システムを構築するにせよ、有能な技術者を支援するシステムを構築するにせよ、製品モデル技術が一つの重要な基幹技術であることが良く認識されてきた。

製品モデルに要求される事柄を良く検討すると、広範な科学・工学知識を計算機化することが結局必要となり、結果として製品モデリングの研究は、設計生産自動化問題のみでなく、物理世界に関連を持つあらゆる計算機応用システムのための基幹技術として、極めて必要性の高いものであることがわかる。広い意味での工学知識ベースの構築へとつながる波及性の高い技術であると考えられる。

上記のような観点からの研究開発は、目新しいことではなく、過去 20 年程にわたって数多くの研究開発が続けられてきたが、実用的に十分なレベルには未だ遠いと言わざるを得ない。

なかでも、形状モデルは、機械製品等のように、主に形が最終的に機能を発揮する製品では重要であり、古くから研究開発が行われてきた。現在では、純粹に計算機科学的な研究分野として確立しており、多くの研究者が質の高い研究を行っている。しかし、製品記述のための形状モデリングには特有の困難さがあり、依然として多くの問題点が残されている。その主なものは、次のようである。

- 1) 形状規定の仕方が人の思考過程と良く整合していないので、形状入力の手間が大変である。
- 2) とりあえず未定義、あるいは曖昧なままにしておいて、後できちんと形状決定する、というようなことがやりにくい。
- 3) 機能形状や意匠形状は、それを決めた意図があるが、その意図がモデル中に保持されないため、後の変更や各種の処理が困難になる。
- 4) 隅のフィレットや繋ぎのように、重要でない形状でも一応定義せねばならない。

その他、いろいろあるが、これらの問題点はすべて形状が機械製品記述において持つ意味を表現できていないことから発生している。これらは形状モデル単独では解決しにくい問題であり、製品モデルの中に形状の意味を記述できるような仕組みが是非とも必要である。最近良く話題にのぼる形状特徴モデリングは、まさにこのような要求に答えるモデルである。より一般的には特徴モデリングと呼ばれ、形状の意味以外に、機械要素の特徴的な情報を明示的に記述するための枠組みとして利用される。特徴モデルの表現、処理方式は未だに決定的なものがなく、重要な研究課題となっている。

形状以外の製品情報モデルについて、重要な問題点を列挙してみよう。

- 1) 対象物属性表現評価……機械製品について言うと、寸法・公差、組立、運動、表面性状、材料等の通常図面上に記述されるような情報である。単にモデル中に表現できるだけでなく、与えられた評価基準に対して評価できなければならない。そのためには、これらの属性情報の持つ意味を十分に表現できる必要がある。
- 2) 複雑対象物の扱い……現在まで多くの製品モデルの研究があるが、そのほとんどは「トイワールド」を扱ってきたと言えるであろう。現場の要求は、たとえば自動車のエンジン丸ごとモデル化してみたい、というようなものであり、このギャップを埋めることは容易ではない。
- 3) 対象物モデルデータベース……上記のような複雑モデルは、データベースに蓄積され、いろいろな形式で検索され、利用される。単体の製品モデルに加えて、それらの組立構造やその他の機能構造も表現されるので、極めて複雑度の高いモデルを取り扱うことになり、従来のデータベース技術では対処しにくいモデルとして、新しいデータベース技術の開発が要求されている。

上記のようなモデルを一般的に表せるようなモデリングの枠組みについても、研究がなされてきたが、未だに決定的なものはない。形式的な一階述語論理により対象間の関係を精密に記述し、対象指向的な方法により、大局的な構造化を行い、制約処理により、関係を維持、処理していく、というのが現在のところ標準的な接近法である。しかし、この方法にも制限が多く、さらに研究が望まれている。重要な問題点を列挙してみると以下のようである。

- 1) (物理) 制約の表現と処理：とくに非線形数式等を含む複雑な制約の扱い
- 2) 複雑なシミュレーションによる属性評価：定性評価と定量評価の関連
- 3) 動的なモデル構成変更、多視点からのモデル情報抽出：見方の違いによるモデル構成変化
- 4) モデルの整合性管理：整合性をあえて破る操作が存在する

これらの問題点についてはすでに多くの議論がなされているが、有力な解決方向は見つかっていない。とくに複雑で精度の高いモデルを扱う時に問題は深刻となる。

すでに上に述べてきたように、現実の物理世界をシミュレーションできることは、現実にまだ存在しないものを造り出そうとする設計生産作業にとって、極めて基礎的で重要な中核技術である。それゆえに、過去に多くの関連する研究が行われてきたが、数理的基礎知識の未成熟や計算機環境の脆弱さにより、実際に役立つ技術とはならなかった。

その一部として、対象品の形状を表現しようとする形状モデリングの技術はその広い適用性のゆえに過去に膨大な研究が行われてきた。その成果の一端はコンピュータグラフィックス等により可視化されて、テレビ番組等で一般にも良く知られるようになってきた。また、有限要素法等に基づく解析技術の発展もめざましく、計算機環境の急速な向上と相俟って、従来の学問分野に「計算」を冠する新しい学問分野（たとえば、計算力学、計算流体力学、等）を創出しつつ、実際の設計生産準備作業に大きな影響を与えるようになってきている。

このように、製品モデルに関する一部の技術は現実的なものとなってきたが、対象

品を包括的にモデル化できるような技術は依然として基礎研究のレベルであり、今後の多くの研究が要求されている。

11. 標準化と STEP

モデルベースの生産システム構築ツールと、それを使用した生産知識に基づく環境適応性の高い生産システムを具体的に構成していくためには、標準化の概念は極めて重要である。このようなシステム構築に関する標準は、すでに存在する既存システムや技術を標準化するのではなく、先進的なシステム構築のガイドラインとなるような、いわゆる「先取り」型の標準にならざるをえないことが多い。既存技術と先取り標準の関係は単純ではなく、学術的にも興味深い技術課題を提供する。その例として、古くはコンピュータグラフィックス標準に関する CORE と GKS を、最近では CAD/CAM データ交換標準に関する IGES と STEP とを挙げることができる。

CAD/CAM や CIM に関する標準としては、以下に列挙するようにさまざまなものが考えられる。

- ・生産システムソフトウェアの参照モデル
- ・システム機能モジュール間のインタフェース（モデリングやデータベース等について形式的なレベルと意味的なレベルがある）
- ・CAD/CAM-CIM データの表現（技術データや管理データ等について形式的なレベルと意味的なレベルとがある）
- ・計算機上への実装に関するもの（システムやデータ記述言語等）
- ・生産知識の体系的表現（設計基準、標準部品、加工データ、物性データ等、さまざまな分野依存データ）

ここでは、プロダクトモデルデータ交換に着目して、現在活発に標準化活動が展開されている STEP について、その意味付けと概要を述べる^[7]。データ交換の有用性と必要性については良く議論されているので詳述しないが、以下のような視点は重要である。

- ・システムから独立したデータの安定管理
- ・国際分業をも考慮したデータを介するシステム統合
- ・生産知識の管理、蓄積、共有

STEP とは、上記のような要件を十分に考慮に入れて、ISO/TC 184/SC 4 において審議中の、プロダクトモデルデータ交換規格案の通称である。STEP の審議は、正に零から始めて、1983 年以来米国や欧州を中心とする国際協同開発の形態をとり、現在 2000 頁にもならんとするような第 1 版 (Initial Release) の文書ができ上がろうとしている。ISO として種々の手続きを経なければならないが、実質的にはようやく実用可能な段階に入りつつある、と言える。

STEP の目的は、「製品のライフサイクルすべてを通じて、特定のシステムに依存しない独立な形式でのプロダクトモデルデータを記述するメカニズムの提供」であり、データベースの問題等も含まれて、単にデータファイルの交換に止まらない。しかし、第 1 版は静的なデータ交換規格に限定されている。

STEP の一般的な特徴は以下のように要約される。

- ・参照モデルの利用による体系化された規格
- ・形式言語の利用による厳密な記述
- ・データの応用・論理表現と実装表現の分離

STEP の ISO 文書の名称は「Product Data Representation and Exchange」であり、分冊形式で構成されており、各分冊ごとに ISO 10303-xx (Part xx) という番号が付いている^[8]。

STEP の概要を記述している Part 1 : Overview and Fundamental Principles によれば、各分冊 Part は、その役割によりクラス (Class) としてまとめられている。現在次のようなクラスがある。

- ・ Description Methods (Part 1x)
- ・ Implementation Methods (Part 2x)
- ・ Testing Methodology (Part 3x)
- ・ Integrated Resources (Part 4x, Part 1xx)
- ・ Application Protocols (Part 2xx)
- ・ Abstract Test Suites (Part 12xx)

Part 1x は、STEP のデータ要素を記述する方法や言語を定義する。Part 2x は、論理的な STEP の定義をファイル形式のような実データへ写像する方法を規定する。Part 3x は、STEP 交換処理システムが、標準規格に適合していることを検証する手法を定める。Part 4x および Part 1xx が規格の主要部であって、各々適用分野に共通の基本データおよび分野依存のデータを定義する。これらは利用可能なデータの集合であって、実際にデータ交換でそのまま用いられるものではない。これらの一種のサブセットとして適用分野ごとに Part 2xx において規定されている Application Protocol が実際に交換されるデータ要素の組を指定している。

STEP は、ようやくその第 1 版が確定した状況であるが、実用上の要請からその先の拡張もすでに話題に上っている。たとえば、標準部品や製造管理データ、ソフトウェアプロダクト等の扱い、データモデルの統一的扱いやデータアクセスインタフェース等があげられる。今後の継続的な開発の努力が要請されている。

現在あるいは将来の STEP、さらにより広く生産システム一般について考えられうる標準化インタフェースについてさらに考察し、利用の立場からの STEP の位置づけを明確にしたい。一般的に考えると広範にすぎるので、生産システムで扱われるデータについて考えていこう。データについては、その「意味」と「表現」を分けて考えることができる。STEP については、扱う「プロダクトモデルデータ」の意味について、ISO で議論している人々に大体の共通した理解はあるが、厳密に議論されたことはない。この議論のためには、以下のようなことを留意する必要があるであろう。

- ・ CAD/CAM や CIM システムでは、どのようなデータを必要とするのか。(Necessity)
- ・ そのようなデータを形式的に (計算機的に) 扱う技術が存在しているのか。(Feasibility)
- ・ 標準化を議論するに足る知見が得られているか。(Stability)

とくに、ただ必要である、というだけではあらゆるデータが必要になるので、必要

性を考察する基準が重要になる。このあたりの議論は純粹に技術的に行うことがむずかしい面もあるが、利用者から見て有用な規格を制定していくためには、最も重要なことである。STEP では、だいたい次のような理解であろう。

- ・製品の生産に必要な技術的データ、製品の表現に重点を置く。
- ・製造プロセスのデータは製品表現に関連する範囲で考える。
- ・管理的データも製品表現に関連する範囲で考える。

上で述べたような Resource Model は、必ずしもこのような議論をきちんとして出てきたものではない。重要性について議論の余地のないものは良いが、一部のものについては、Necessity, Feasibility, Stability の観点からまだ多くの議論の余地を残している。また、極めて重要であると思われるにもかかわらず、積極的な Volunteer (国, 人) がいないために手付かずのものも沢山ある。これからは、利用者主導により真に有用性の高い Model, とくに Application Protocol を考察していく必要がある。

データの「表現」に関する問題は、はるかに厳密に議論されている。大雑把に以下のような区分を考えることができる。

- ① 静的なデータ表現 (インスタンスのレベル)
- ② 静的なデータ表現 (メタレベル)
- ③ 動的なデータ操作 (形式的)
- ④ 動的なデータ操作 (意味的)
- ⑤ データ主導による表現と操作

現状の STEP は、①のレベルで、あらかじめ想定されたスキーマに従いデータが表現され、交換される。しかし、この枠組みは素直に②のレベルに拡張できて、現在検討が進められており、標準部品の扱い等と同様の枠組みになるものと思われる。ファイルやデータベースアクセスについては、形状モデル等に見られるような応用インタフェースは④のレベルに相当する。STEP の範囲では、データの形式的なアクセスだけを取り扱うのが適当であり (これを越えると Modeling の問題になる。), これが SDAI 等として検討されているものであり、レベル③に相当する。この間に、種々のインタフェースを考えることができると思われるが、標準化の観点からは、「意味」との切り分けを明確にして、統一性のある方法論を確立することが重要である。STEP の作業グループは、この面で大きな技術的貢献を果たしているが、問題の困難さに鑑み情報処理分野との連携を強化することが望まれている。たとえば⑤のレベルは、必要に応じて操作の意味付け情報を与えてやり、意味的なデータ操作を形式的に実現すること等を指しているが、これは困難な課題であろう。

生産システムソフトウェア等を考える時には、その技術開発の過程を体系的に捉えておくことが重要である。先行的な研究開発と実用化やその評価が繰り返され、技術水準が順次向上していく。標準化は、この過程の重要な要素として位置付けられる。体系的に解明された技術は標準化されて、次の開発の重要な基礎知識となる。言い換えれば、標準化は次世代開発の先行開発と位置付けられる。従って、標準化は単純な整理作業ではなく、困難な基礎技術開発と考えるべきである。標準文書を作成する程度までに整理されてくれば標準化の専門家に委ねるべきであるが、それ以前の標準化研究開発は、企業の研究開発の一環として現場の技術者が担当すべき課題なのである。

12. おわりに

既存の製品技術や生産技術が成熟すると共に、引き続き新世代の生産システムのあるべき姿を見出すことが重要になってきている。急激な技術開発の進展、人的資源の減少あるいは枯渇、さらには全プロダクトライフサイクルに代表される生産活動の再考等、挑戦すべき課題は山積しており、今後の強力な研究開発が期待されている。

短期的には、次世代生産システムは、高付加価値製品を柔軟で効率良く作り出し、また技術や人、環境の変化に迅速に追隨できることが求められている。これらの目標を達成するために、以下のようなことが要求されている。

- ・生産システムソフトウェアの柔軟な再編成
- ・ソフトウェアシステムアーキテクチャの成長性
- ・技術知識の蓄積と伝承
- ・人に親和性の良い作業環境の実現

これらの課題は、仮想生産の基礎の上に解決されるものであり、一般のソフトウェア工学の技術課題としても魅力に富むものである。

- 参考文献
- [1] 生産システムと未来の社会, FA 部門報告, 日本機械学会誌, 1993年5月号掲載予定.
 - [2] G. Sohlenius, Concurrent Engineering, Annals of CIRP, 1992, Vol. 42, No. 2, pp. 645~655.
 - [3] Report of the First CIRP Workshop on Concurrent Engineering for Product Realization, Annals of CIRP, 1992, No. 2, pp. 743~746.
 - [4] Special Issue on Computer Support for Concurrent Engineering, IEEE Computer, 1993, Vol. 26, No. 1.
 - [5] 木村文彦, コンカレント・エンジニアリングと仮想生産, 電気学会誌, 113巻, 1993年3月号, pp. 183~188.
 - [6] F. Kimura, Product and Process Modelling as a Kernel for Virtual Manufacturing Environment, Annals of CIRP, Vol. 43, No. 1 (to appear).
 - [7] STEP セミナ資料, 日本コンピュータグラフィックス協会, 1987年以来毎年開催.
 - [8] ISO CD 10303-xx, Product Data Representation and Exchange, 各分冊ごとに日本コンピュータグラフィックス協会より入手可能.

執筆者紹介 木村文彦 (Fumihiko Kimura)

1945年8月5日生, 1968年3月東京大学工学部航空学科卒業, 1974年3月東京大学大学院博士課程修了, 工学博士, 同年4月通産省工業技術院電子技術総合研究所研究員, 1979年4月東京大学工学部精密機械工学科助教授, 1986年同教授, 1987年より1993年の間同大学先端科学技術研究センター教授. 主として, 形状モデリング, CAD/CAM, 生産システムに関する研究に従事, IFIP WG 5.2/5.3 委員, CIRP 副会員.



次世代 CAD/CAM システム「CADCEUS」の概要

An Overview of the Next Generation CAD/CAM System「CADCEUS」

大 高 哲 彦

要 約 CADCEUS はメカニカル分野全般を対象とした、初期設計から生産準備までの製品開発工程を総合的に支援する CAD, CAM, CAE そして CG の統合システムである。CADCEUS は製品モデル、形状特徴、非多様体、幾何拘束/推論等の先端技術を取って実装することにより、設計・生産の大幅な品質向上、工期短縮を狙いとしている。本稿では CADCEUS のシステム構成、主要な概念、特徴的な機能について概説する。

Abstract CADCEUS is an integrated CAD/CAM/CAE/CG system designed to totally support all the product development processes from early design to manufacturing preparation. Target products are all types of mechanical products. CADCEUS aims at drastic quality improvement of design and manufacturing and extensive reduction of leadtime for product development by implementing advanced technologies such as product model, form feature, non-manifold and geometric constraint and reasoning. This paper briefly describes its system configuration, major concepts and characteristics.

1. はじめに

CADCEUS は、自動車、機械、精密、電機分野の製造業の設計から生産準備までの製品開発工程を総合的に支援する CAD, CAM, CAE そして CG の統合システムである。

CADCEUS (Computer Aided Design, manufacturing and engineering for Concurrent Engineering by Unisys System) の語源は、ローマ神話の神々の使者 Mercury が持っている杖という意味の英語：Caduceus であり、設計や生産に携わる技術者の強い味方になるという願いが込められている。

製造業において従来以上に品質の向上、コストの削減、さらに製品開発期間(リードタイム)の短縮の必要性が高まり、新たな方式や概念として CIM (Computer Aided Manufacturing) やコンカレントエンジニアリングの導入が要請されている。CADCEUS はこれらの傾向や要請に対応するために「製品モデル」、「非多様体モデリング」、「形状特徴」さらに「幾何拘束」等の最新技術を取り込み、日本ユニシスが独自に開発した次世代統合システムである。

本稿では 2 章で CAD/CAM システムの現状と CADCEUS の狙いについて述べ、3 章では CADCEUS の特徴を、とくに幾何的製品モデル (geometric product model) の実装に焦点を当てて概説する。以降、4 章ではシステム構成、5 章では機能概要について述べる。

2. CAD/CAM システムの現状と CADCEUS の狙い

製造業は 21 世紀に向けて設計・生産方式の画期的な革新を迫られている。市場におけるユーザーズ多様化への対応はますます重要な課題とされてきた。開発・生産・

販売という企業活動全体にわたる情報を、統合化し一元管理することにより企業運営の効率向上を図るためのCIM構築と、開発工程の同時並列進行による開発期間の大幅短縮のためのコンカレントエンジニアリングの実現が主要な課題である。見込生産における企画、意匠設計、製品設計、生産準備という製品開発工程において、各工程のリードタイム短縮を目的にCAD/CAM化が進められてきた。

しかし従来のCAD/CAMシステムの能力は、各工程個々の自動化や合理化が限度であり、このままの延長線上での改善を進めても大幅なリードタイムの短縮は望めないのが現状である。そこで各工程のCAD/CAMシステムを改良していくのは当然のこととして、今後はそれら個別に導入されたために生じた自動化の島を有機的に結合すること、つまり「統合化」と各工程のシステム間で伝達される「情報の充実」が必要となる。前工程からの情報が不十分であると、後工程は前工程で決められた事項でも再検討しシステムへの再入力が必要となり、「重複作業」が発生してしまう。このことは、生産性を悪化させるばかりか、図面やモデルの削減が実現できず、作業の前倒しによる並列化が見込めなくなる。CAD/CAMシステム導入以前は、図面や作業指示書等の設計図書や、人と人とのコミュニケーションにより前工程の意図を伝達できていた。現状のCAD/CAMシステムはこれらの情報のうち、形状だけを主体に伝えるのみで、形状に込められた意図や機能的特徴や属性は伝えることができない。

一方、形状の表現はワイヤフレーム、サーフェス、ソリッドと現実の物に近付ける進展はしてきた。しかし、あくまでも形状のみで図面の注記や設計図書に盛り込まれている情報はモデリングされたわけではないために、あいかわらず図面等が必要となるのは以前と同じである。これらを解決するために、形状の他に製品が持つべき情報や、設計から生産までに製品が変遷する間の情報、つまり設計・解析・加工に関する情報も併せ持つ「製品モデル」が提唱され、研究界やISOで議論されている。CADCEUSはこの「製品モデル」を実装することにより、モデルを中核として、統合化、データの一元化、自動化の推進等の要請に応えることを大きな狙いとする。さらに形状特徴、幾何拘束、非多様体等の新技術を組み合わせることによって、モデリング環境の革新、設計意図の継承、自動化の推進、設計変更への対応力の強化を計る。

3. CADCEUSの特徴

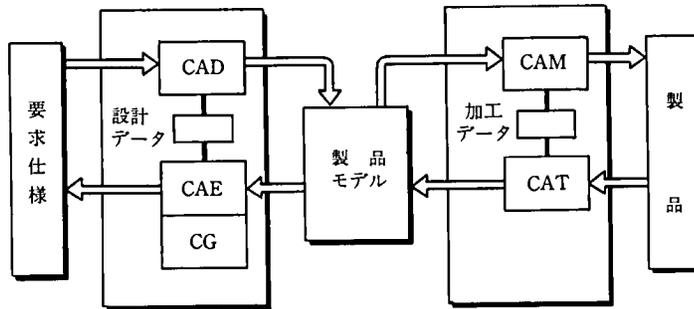
CADCEUSの主たる特徴は、統合データベースと統一ヒューマンインタフェースを実現した「CAD, CAM, CAE, CG統合システム」であり、設計から生産準備までの全工程を総合的に支援するとともに、管理系システムとインタフェースをとることにより、CIMの構築やコンカレントエンジニアリングを実現し、リードタイムの大幅短縮を支援する。

また、設計意図を他工程へ正しく伝達できるようにすること、技術者の言葉遣いでシステムの操作ができること、変更に対し柔軟に適応できるようにすることによりモデリングの生産性を大幅に向上させるために、以下に示すような新技術を取り込んでいる。

3.1 製品モデル

設計・生産の対象である製品形状だけでなく製品の持つ特性、つまり機能や属性に関する情報を体系的にシステム内に表現したモデルを製品モデル(Product Model)という。製品モデルデータは設計から生産準備までの全工程にかかわる情報、つまり、製品設計情報、解析情報、生産技術・加工情報から構成される。

図1はCAD/CAE/CG/CAM/CATシステムと製品モデルの関係を示す。図の左側のサイクルは設計サイクルである。設計者は設計知識や過去の経験を基に、CADシステムを用いて要求仕様を満たす製品モデルを構築する。作り込まれるデータは、設計の段階によって異なるが、最終的には形状、製品構成、設計意図、公差、図面表現、材料、板厚等である。必要に応じてFEM解析、機構解析等のCAEシステムを用いて、製品モデルが強度、振動特性、運動機能等の観点で要求仕様を満足するか否かを確認し、必要な補正を行う。このサイクルは設計の各段階で回され、段階の進みに応じ製品モデルは具体化・精密化される。CAEシステム使用時の解析条件や解析結果も設計意図の一部と考えられるので、製品モデル内に蓄積される。CGシステムは製品モデルがプレゼンテーションに関する要求仕様を満たすか否かを確認する道具という位置付けである。



(北海道大学工学部教授 岸浪健史, 日経コンピュータ
1990年10月22日号 CAD/CAMの技術動向より)

図1 製品モデルの位置付け

Fig.1 The situation of a product model

図の右側のサイクルは製造サイクルである。製造技術者は、加工知識や設備に関する情報を基に、製品モデルを正としてそれを製造するために必要な加工工程を設計し、加工データを作成する。この時用いられるのがCAMシステムである。加工後の製品はCATシステムを用いて製品モデルと照合され、CAMシステムを用いた必要な補正が行われる。

CADCEUSで扱う製品モデルは以下の情報から構成される。

- ① 部品の形状表現(幾何要素, 位相要素で表現。形状特徴, 幾何拘束による表現も可能。)
- ② 部品の形状以外の属性情報(材質, 表面粗さ, 公差, 板厚)
- ③ 製品を構成する中間組立品, 部品間の階層関係, 位置関係, 公差に関する情報

- ④ 部品, 中間組立品間の接続関係に関する情報
- ⑤ 製品, 中間組立品, 部品の図面表現に関する情報
- ⑥ 部品の種々の場(応力, 熱, 電/磁)での評価情報(FEM 解析モデル・解析条件・解析結果)
- ⑦ 製品の運動機能に関する情報(機構解析モデル・解析条件・解析結果)
- ⑧ 部品の加工情報
- ⑨ 管理情報

上述のうち①のみが必須であり, ②~⑨が任意である。②~⑨は用いる機能に応じて生成され①と組み合わせられる。図 2 に製品モデルの体系を示す。

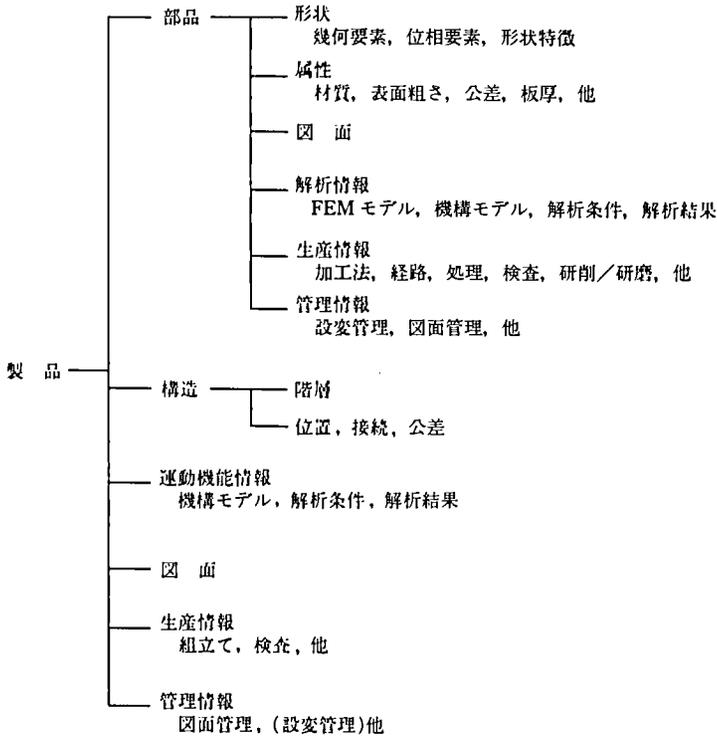


図 2 製品モデルの体系

Fig. 2 Configuration of a product model

本システムのデータベースは, 設計作業の適当なまとまり(仕事)によって分割される。このまとまりを作業空間(ワークスペース: WS)と呼ぶ。WS は階層を持たず, 一つ一つの WS が独立である。

WS は, 一つの部品(Part), 中間組立品(Sub Assembly), 製品(Assembly)あるいは, 1 枚の図面, 図面内の投影図ごとに一つのまとまりを構成する。このまとまりをオブジェクト(Object)と呼ぶ。一つの WS 内でオブジェクトは階層関係や接続関係を持つことができる。

オブジェクトはその性質と用途から次の 5 種類に分類される。

- ① パート : 一つの「部品」に対応するオブジェクト
- ② アセンブリ : 「中間組立品」や製品に対応するオブジェクト
- ③ シート : 1枚の「図面」に対応するオブジェクト
- ④ プロジェクション : 1枚の図面の中の一つの「投影図」に対応するオブジェクト
- ⑤ バリエーション : パラメトリックな部品に対応するオブジェクト

オブジェクトに含まれるデータを集合化する概念として、グループがある。オブジェクトの構成要素を任意に組み合わせてグループ化し、可視性制御、表示色制御、図形処理に利用できる。グループは階層を持たないが、一つの構成要素が複数のグループに所属することが許される。

オブジェクトやオブジェクト内のグループとは別に、これらとはまったく独立した見方(ビュー)でデータを集合化する概念として“レイヤ”がある。レイヤはグループとは異なり、複数のオブジェクトにまたがるのが許される。一つの図形に対して複数の見方(ビュー)が許され、見方ごとの属性を持つことができる。レイヤも可視性制御、表示色制御、図形処理に利用できる。

3.2 非多様体

非多様体とは、二次元の例でいうと任意の点の近傍が二次元円盤と同相にならない形状(図3)をいう。

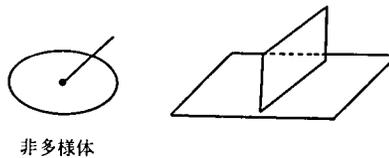


図3 非多様体の例 ①

Fig.3 Examples of non-manifold ①

CADCEUSでは、対象形状が非多様体の場合多様体分割し、二つの多様体の隣接状態をパーテックス、エッジ、フェース3レベルの隣接位相要素で表現するという立場を取る。対象モデルを非多様体まで拡張したモデラは、従来の多様体のみを対象とするモデラの機能を内包するだけでなく、以下のような種々の実用上の長所がある。

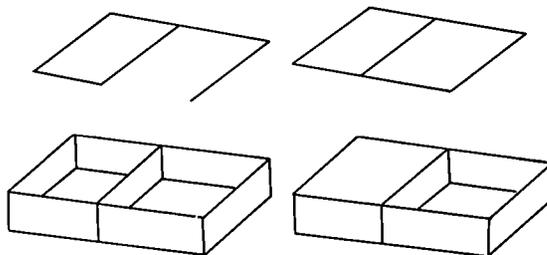


図4 非多様体の例 ②

Fig.4 Examples of non-manifold ②

- ① ワイヤフレーム、サーフェス、ソリッド・モデルの混在が可能である。

ワイヤフレーム/サーフェス/ソリッド・モデルが単一のデータ構造で表現できるので生成過程にあるモデル(図4)を忠実に表現できる。また、ソリッドとサーフェス、サーフェスとサーフェス、の組み合わせの集合演算や複合面演算も可能である。FEM 解析では、異なる次元の形状を一体で解析するニーズは従来から高いが、それも可能になる。

- ② 多重表現・代替表現が可能である。

プレス型の上型/下型共存のモデル(図5)のように、多重表現が可能である。この例はプレス型のピン形状とフィレット形状を共存させ、成形上の考慮から上型、下型で用いる形状を変えた例である。図6の例は穴形状について、穴ありの状態と穴なしの状態を共存させ、粗取り加工では穴なし形状を対象にし、仕上げ加工では穴ありの形状を対象にしている例である。FEM 解析でも解析条件を考慮した FEM 用の理想形状作成にこの多重表現は効果を発揮する。

またソリッドモデルのある部分を簡略表現できる(FEM のモデリングに有効である)。たとえば、図7のモデルも中立面をサーフェス・モデル化した時には非多様体でなければ表現できない。

- ③ 接触形状(図8)等、自由な集合演算の結果を表現できる。

多様体のみを扱うと、集合演算が閉じた系とならないが、非多様体まで拡張することにより、それが閉じた系となる。

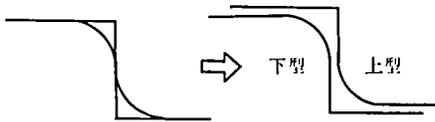


図5 多重表現 ①

Fig.5 Multiple representation ①

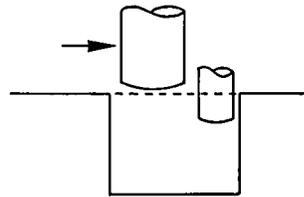


図6 多重表現 ②

Fig.6 Multiple representation ②

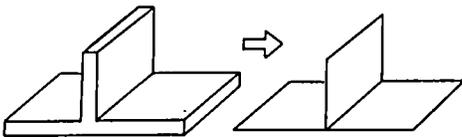


図7 簡略表現

Fig.7 Simplified representation

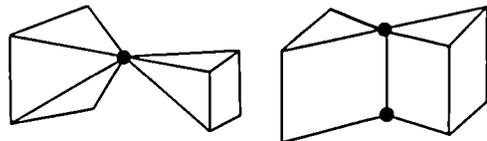


図8 非多様体の例 ③

Fig.8 Examples of non-manifold ③

3.3 幾何拘束モデリング

対象形状の表面を許された形状要素の集まりとして直接表現する従来の手法とは異なり、対象形状が満たすべき幾何学的な拘束条件(幾何拘束)、および長さ、角度等の可変なパラメタとパラメタ間の種々の関係をシステムが記憶しておき、可変パラメタ

に実現値を与えると、拘束条件およびパラメタ間の関係を満足する形状をシステムが自動的に生成する手法を幾何拘束モデリングと呼ぶ。扱う拘束条件は、水平、鉛直、平行、直交、接、長さ、角度等であり、パラメタ間の関係としては、算術式で展開できる範囲で通常の応用では十分である。

幾何拘束モデリングを支える技術は大別すると次の二点になる。

- 1) 拘束付けされたモデル(G.C.M.: Geometrically Constrained Model)の構築法
- 2) 形状の自動生成の手法

1)に関しては、CADCEUSではモデリング時に使用者が用いたコマンドの意味から拘束条件を自動生成する方式を採用している。すなわち、‘水平線コマンド’で描いた線には水平という拘束が、‘鉛直線コマンド’で描いた線には鉛直という拘束が、それらが交っていれば直交という拘束が付与されるという具合である。マンマシン・インタフェースの平易化および使用者の意図の正確な把握の観点からモデリング時の入力装置(たとえばマウス)の動きに応じてシステムが自動認識した拘束を動的に画面に反映し(ダイナミック・ナビゲーション)モデリングを支援する機能を備えている。G.C.M.を完成するためには、2)の形状自動生成が可能な必要十分な拘束条件が蓄積されたか否かの検証が必要である。CADCEUSでは、後述の幾何推論機構を働かせて形状再生シミュレーションを内部的に実行し、拘束条件の過不足を検出する。

2)のG.C.M.および変更パラメタを入力として形状を自動生成する手法については、これまでに種々の方式が提案されている。その主なものは、

- ① 拘束条件を代数式に展開し、非線形連立方程式を解くことによって形状諸元を確定する方式
- ② IF-THEN型の前向き推論によって、与えられた拘束条件から形状諸元を算出する方式
- ③ 自由度解析方式。つまり与えられた拘束条件を満たしながら系の自由度を徐々に低減し、最終的に形状諸元を確定する方式

等である。CADCEUSでは、技術的な見通しの良さおよび拡張性の高さから②の方式を現状では採用している。図9に幾何拘束ソルバの全体図を示す。図中のスケッチャが上記1)の機能をなすモジュールである。推論処理は、推論ルールのIF節をパターンマッチングしてすべてのIF節を満足した時にTHEN節を導き出す(一般的には“仮説から事実の導出”と呼ぶ)。前向き推論の効率向上のために、複数のルールの共通なIF節をまとめてパターンマッチング回数を減らす手法がReteアルゴリズムであり、その結果をネットワーク形式で表したものがReteネットワークである。その例を図10に示す。Reteネットワーク・インストラクションは、ネットワークの内容をインタプリティブに解釈できるように置き換えたものである。Reteネットワーク・インストラクションは推論ルールの変更・追加に応じて作り直す。

推論処理実行時は、図9中のReteネットワーク・インタプリタが、スケッチャが蓄積した拘束述語およびReteネットワーク・インストラクションを参照して推論を行い、事実の導出、導出事実とその手続きおよび適用されたルールとの関係、幾何要素と導出仮説(拘束)の関係等の情報を蓄積する。これらの情報を管理するのがATMS(Assumption based Truth Maintenance System)である。この時点で拘束条件の過

□ 推論エンジン全体構成図

□ Reteネットワーク構築のプロセス



□ 幾何推論プロセス

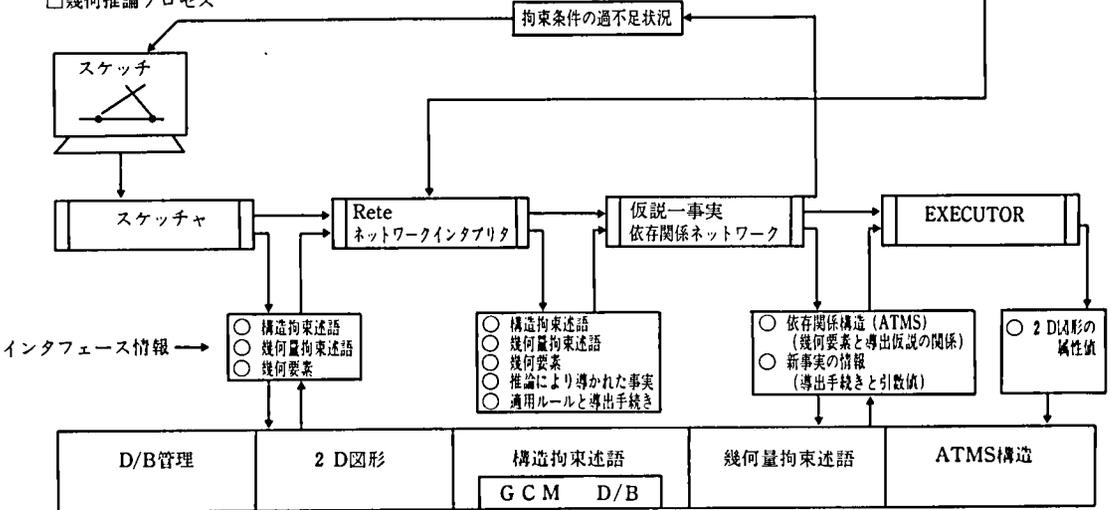


図 9 幾何拘束ソルバ全体図

Fig. 9 Overall configuration of geometric constraint solver

不足が検出される。最後の EXECUTOR は ATMS に蓄積された手順に従って形状を再計算する。

幾何拘束モデリングは、従来の手法に較べて形状設計者の意図を直接データベースに記憶できるという利点があり、CADCEUS では以下の应用到に適用する。

- ① 機能部品の類似設計
- ② 標準部品の登録・実体化
- ③ パラメトリック製図
- ④ 形状特徴プリミティブ(の断面)の定義

3.4 形状特徴

形状特徴とは設計者や生産準備者が意識する機能単位に対応する形状群であり、たとえば、ボス・リップ・フィレット・ヒード等が典型である。機能単位に対応することから当然設計変更を受ける単位とも考えられる。形状特徴をデータ単位としてシステムが認識し管理することにより種々の応用の道が拓けるが、一般的に次の二つに分類されている。

- ① 形状モデリング環境の改善 (Modeling by Feature)
- ② 工程間インタフェースの改善, 自動化の推進 (Automation by Feature)

①の観点では、ボスの取付け、ボスの位置や大きさの変更、ボスの削除等、技術者の用語でモデリングできることが第一の利点である。このとき図 11 に示すように、ボスという形状特徴要素の定義に前述の幾何拘束が使われ、ボスの取付けに際しては、

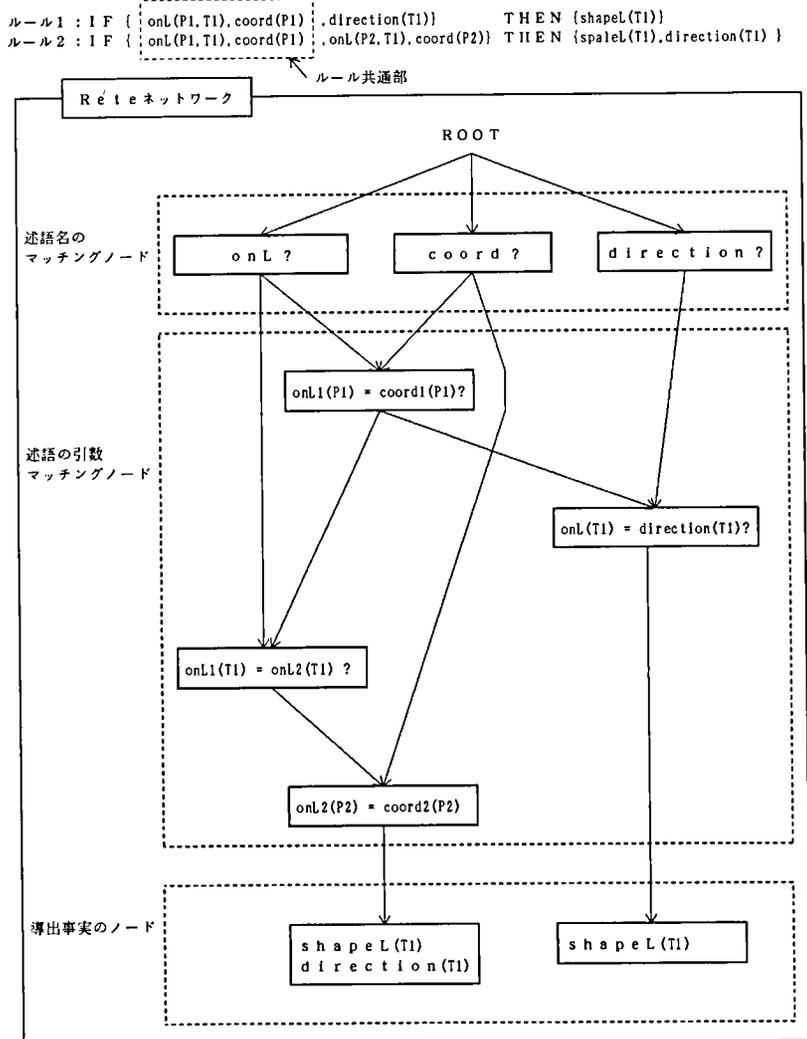


図 10 Rete ネットワーク

Fig. 10 Rete network

図 12 に示すように、ボスの削除、ボスの位置や大きさの変更等の場合に簡単な論理的データ操作で迅速に元の形状が復元できるように非多様体表現を用いる。さらにボスの変形時は、図 13 に示すように幾何推論を用いて変形ボス形状を自動的に生成し取り付け直す。以上に示したように、簡単な形状特徴モデリングであっても、幾何拘束/推論や非多様体表現を組み合わせる必要がある。

第二の利点は、形状特徴の持つ意味(成立条件と言っても良い)を扱えるということである。図 14 に示すように、ある物体の表面から深さ h_1 であけられた穴特徴は、物体の高さが変更されても表面から深さ h_1 の穴であるという成立条件を失なわない。このように形状特徴の成立条件をシステムが認識し管理することによって、成立条件を犯すような操作、たとえば非貫通穴が取り付けられている物体の高さを穴の深さ以下の高さに変更するような操作を禁止することができる。形状特徴モデリングの実装技術に関しては、これまで述べた幾何拘束/推論や非多様体表現だけでなく、

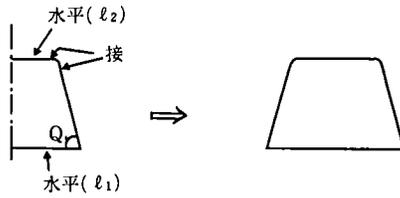


図 11 ポスの定義

Fig. 11 Definition of a protorution

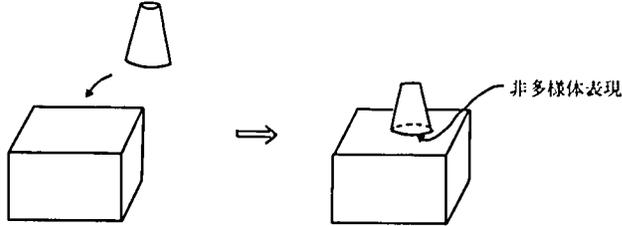


図 12 ポスの取付け

Fig. 12 Addition of a protorution

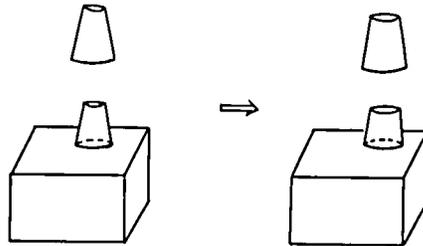


図 13 ポスの変形

Fig. 13 Deformation of a protorution

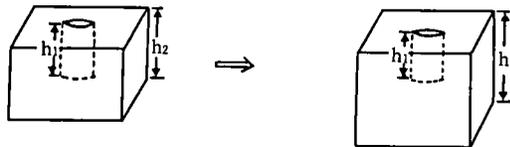


図 14 形状特徴の性格の維持

Fig. 14 Maintenance of form feature characteristics

- 形状特徴操作時に使用者が付与した制約条件，たとえばある物の端から x_{mm} の位置に角度 θ で取り付けるというような条件，をどう記憶し管理するか
- 形状特徴が複雑に干渉し合う状態をどう表現し管理するか

等の課題がある。

形状特徴モデリングに関するこれ以外の課題としては，ユーザ定義形状特徴の問題がある。形状特徴は本質的に応用依存である。なぜなら組立て構造設計者と加工の専門家は同じ対象に対しても別な形状特徴認識をする。この課題に対しては，CAD CEUS は，幾何拘束の項で述べたスケッチャを使って自由に断面形状を定義し，それを挿引や回転させてユーザ独自の形状特徴を定義する手段と，フィレットやビード等

のように形状を生成する手順のひとまとまりをユーザ定義形状特徴とする手段を備えている。

以上 CADCEUS の形状特徴モデリング技術を概観したが、CADCEUS の特徴をまとめると以下の通りである。

- 1) ワイヤフレーム、サーフェス、ソリッドいずれの表現も形状特徴として扱われる。したがって、機能部品だけではなく、自由曲面が支配的な金型の設計等にも適用可能である。
- 2) 非形状特徴と形状特徴の混在が許される。これは一部の市販システムのようにすべての形状要素を形状特徴として捉えるのは不自然で無理があるという認識に基づく。現に車の外板や金型の設計者は基準形状と特徴形状(キャラクタ・ラインやビード等)は明らかに区別しているはずである。
- 3) 形状特徴のユーザ定義機能が豊富である。

②に関しては、CAD と FEM のインタフェースの改善、CAD と CAM のインタフェースの改善、一部の自動化を当面の狙いとしている。FEM の理想形状をどう効率的に作成するかは従来からの課題であるが、CADCEUS では、形状特徴の標準機能として備えている「抑止」機能を用いることにより、解析結果には寄与しない微小な穴や径の小さいフィレット等を簡単なデータ操作で一時削除し、FEM 用の理想形状を導出することができる。また、形状特徴と製品モデルの項で述べた材質、表面性状、公差等の情報や加工技術情報を組み合わせることによって、加工工程設計への応用や NC 加工のための経路計算の自動化への道が拓ける。

4. システム構成

CADCEUS は GWS で稼働する分散システムで、ホストコンピュータやサーバとネットワークを組むことでデータの集中管理を実現する。基本的な集中分散システムの構成は図 15 のとおりである。

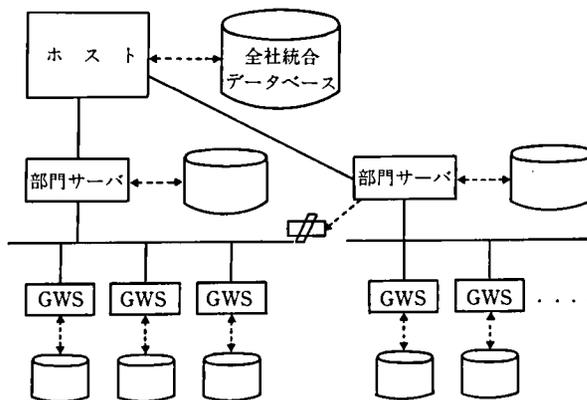


図 15 システム構成

Fig.15 System configuration

また、ソフトウェアは「製品モデラ」を中核として図 16 のように以下のサブシステムから構成される。

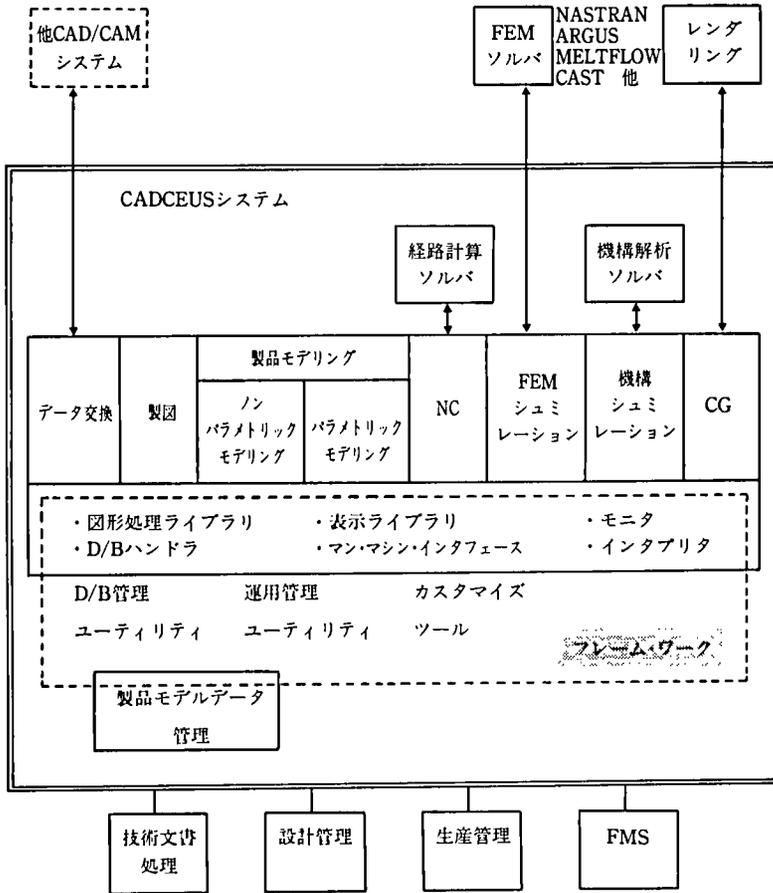


図 16 ソフトウェア構成
Fig. 16 Software configuration

- ① 製図サブシステム
- ② NC サブシステム
- ③ FEM プリ・ポストプロセッサ
- ④ 機構シミュレーションサブシステム
- ⑤ 画像処理(CG)サブシステム
- ⑥ データ交換サブシステム

図中二重線の内部が CADCEUS である。設計対象の形状定義，製品構成定義(部品の階層，位置，接続関係)，製品属性定義(材質，板厚，公差等の属性と形状の関係付け)を行うための製品モデラがその中核である。

製品モデルを中核とした応用機能としては、

- 形状と製図要素の連動，各種規格対応等の特徴を持つ製図サブシステム
- 2軸から多軸に至る広範な加工を支援する NC サブシステム
- 構築した製品モデルの各種の場(応用場，熱場，電場，磁場)での FEM 解析を行うための FEM プリ・ポストプロセッサ
- 構築した製品モデルの運動機能を評価する機構サブシステム

- 製品モデルのプレゼンテーションのためのCGサブシステム
- 他CAD/CAMシステムとのデータ・インタフェースとしてIGES V5.0対応機能他を持つデータ交換サブシステム

が配されている。なお国際規格STEP対応については、第一リリースが確定次第順次対応機能が付加される。

これらの設計・生産準備活動を直接的に支援する機能の他に、

- データベース管理のための各種ユーティリティ
- 本システムの運用管理のための各種ユーティリティ
- マンマシン・インタフェースを使用者独自の形にしたり、機能追加を行うためのカスタマイズツール

等が用意されている。

製品モデルデータ管理サブシステムは、本システムの周辺の技術文書処理、設計管理、生産管理、FMS等の分野の各種システムとのデータ・インタフェースとして用意されているものであり、本システムを用いて構築した製品モデルデータベースを要求キーに沿って検索し、検索結果の情報を相手システムに提供する。

5. 機能概要

CADCEUSには、「製品モデル」を中核に設計・生産準備を支援するための各種応用機能がある。本章では、CADCEUSの特徴的機能を説明する。

5.1 製品モデリング

5.1.1 形状モデリング

本システムの形状モデリング機能には以下の特徴がある。

- 1) 対象形状を目的に合わせた抽象度でモデリングすることが可能である。このためにワイヤフレーム・モデル、サーフェス・モデル、ソリッド・モデルのいずれのモデルも構築できる機能が用意されている。また、これらの混在、一方から他方への変換も可能である。これらのモデルの主な用途は以下のとおりである。
 - ① ワイヤフレーム・モデル : 二次元あるいは三次元の設計・製図
2・1/2軸制御NC加工
FEM解析, 機構解析
 - ② サーフェス・モデル : 意匠設計, 曲面NC加工
FEM解析, 機構解析
 - ③ ソリッド・モデル : 意匠設計, 機械部品設計, NC加工
FEM解析, 機構解析
プレゼンテーション
- 2) 形状モデリングの途中でのモデルの状態を正確に表現することを主目的として「非多様体」を表現できるデータ構造を採用している。
- 3) マクロ的形狀操作, 形状の持つ意味の内包, 変更に強い形状操作等を目的として「形状特徴」を導入している。形状特徴はこの他に, FEM解析における対象形状モデルの簡素化, NC機能との連動等にも使われる。
- 4) 類似部品の形状設計の省力化を目的として「幾何拘束/推論モデリング」機能が

ある。

5.1.2 属性モデリング

本システムで扱う形状以外の製品属性は、材質、表面粗さ、公差(幾何公差、寸法公差)、板厚であり、すべて関連する形状と関連付けて記述される。材質は、重量の算出、FEM解析、材質感を表現したレンダリング等に使用される。表面粗さは、加工条件(使用工具、計算精度)の自動算出等の応用のために利用される。公差は、当面の応用としては、形状モデルと与えられた公差の間の整合性の自動検証を考える。組立工程シミュレーション等の応用でも公差の考慮が必要となる。板厚は、FEM解析で使用する。

5.2 FEM プリ・ポストプロセッサ

プリプロセッサの特徴としては、FEM解析用理想形状創成のための形状特徴の活用があげられる。すなわち、解析において不必要な微小な穴やフィレット等を簡単な指示で一時的に除去して理想形状を創成できる。解析モデル作成については、サーフェス、ソリッド共に自動分割機能が中心であり、適合化メッシュ機能も追加される。

解析プログラム(ソルバ)としては、NASTRAN、MARC、ARGUS(パネルの非線形解析)、Applied Structure(感度解析、最適解析)、JOH-DYNA(衝撃解析)等の他に日本ユニシス独自開発のMELTFLOW(樹脂の流動解析)、CAST(樹脂の凝固解析)、METALFILL(湯流れ解析)等、目的に合わせて最適のものを選択できるように豊富に用意されている。

ポストプロセッサは、変形図、等高線図、矢印図、XYグラフ等の通常の出力形態に加え、広範なアニメーション機能が特徴である。

5.3 機構解析

本システムの機構解析機能は、部品形状の詳細が未定の設計初期段階での製品全体の運動機能の検討、および部品形状が確定した設計の段階での干渉チェック等を主目的とした詳細検討等の用途に使用可能である。

5.4 製 図

特徴は次のとおりである。

- ① 形状と製図要素が関係を持ち、形状の変更に対して製図要素も自動的に追隨する。また、幾何拘束/推論の応用として、寸法を制御パラメタとし、寸法を変更することによって形状を自動変更する寸法駆動の製図(パラメトリック製図)も可能である。
- ② JIS、ANSI、ISOいずれの規格に準拠した製図も可能である。
- ③ 三次元形状モデルに対する製図では、モデルの変更が図面に自動的に反映される。プロジェクション図面の線種定義機能により、モデルの形状を変更せずに、各ビューでのモデルの“見え方”を線種で制御可能である。

5.5 NC

曲面加工と穴や領域等の機械加工とを一体化し種々の加工法に対応する機能を持つ。加工タイプの分類は図17のとおりである。

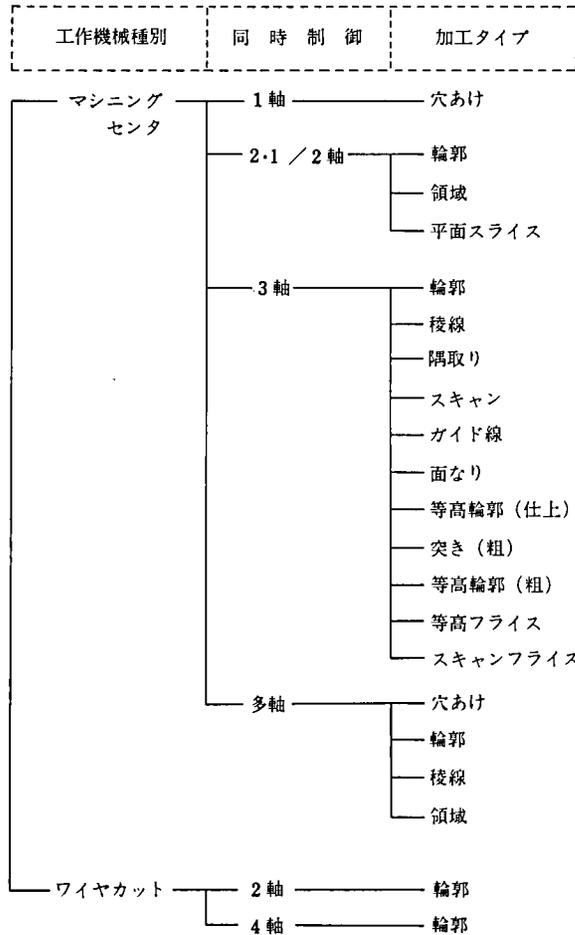


図 17 加工タイプの分類

Fig. 17 Category of machining type

5.6 CG

製品開発における意匠デザインを支援するサブシステムで、レイトレースやマッピング等のレンダリング機能や、画像の合成等のペイント機能がある。さらに、画像のファイリングや各種メディアへの入出力も可能である。

5.7 カスタマイズ

CADCEUS の標準的に提供される機能をもとに、ユーザ固有の操作環境の設定、機能要求の取り組みおよび自動化推進のために、各種カスタマイズを可能とする次のような仕組みが用意されている。

- ① 画面レイアウトの変更
- ② メニューレイアウト、内容、構成の変更
- ③ メッセージの変更
- ④ ユーザ・コマンドの追加登録
- ⑤ ユーザ・サブシステムの追加登録

6. おわりに

CADCEUS の特徴的な概念や機能を概説した。本稿に示したような新技術が設計環境の改善や自動化の推進の有力な武器になり得ることは明らかであるが、世界的にもこれらを総合的に実装したシステムがほとんどないため、応用機能の拡がり、深さあるいは限界について不明な部分も多い。今後、実用的な視点から有効な応用機能を積み上げることによってこれらを明らかにしたい。なお CADCEUS の要素技術に関しては、本誌にデータベースを採り上げて解説している。それ以外については、次号以降で順次解説する予定である。

- 参考文献 [1] Pratt, Synthesis of An Optimal Approach to Feature Modeling, ASME Computers in Engineering, 1988.
- [2] J. R. Dixon et al., Unresolved Research Issues in Development of Design with Features Systems, Geometric Modeling for Product Modeling, Elsevier, 1990.
- [3] P. R. Wilson and M. J. Pratt, A Taxionomy of Features for Solid Modeling, Geometric Modeling for CAD Applications, Elsevier, pp. 125~136, 1988.
- [4] K. Weiler, The Radial Edge Structure: A Topological Representation for Non-Manifold Geometric Boundary Modelling, Geometric Modeling for CAD Applications, Wozny, M J et al, (Eds), 1986.
- [5] E. L. Gursoz et al., Boolean Set Operations on Non-Manifold Boundary Representation Objects, Computer-Aided Design Vol. 23, No. 1, 1991.
- [6] M. Masuda et al., A Mathematical Theory and Applications of Non-Manifold Geometric Modeling Advanced Geometric Modeling for Engineering Applications, North-Holland, F.-L. Krause, H. Jansen (Eds), 1990.
- [7] B. Alfeld, Variation of Geometries Based on a Geometric-Reasoning Method, Computer Aided Design, Vol. 20, No. 3, 1988.
- [8] C. L. Forgy, Rete, A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem, Artif. Intell., Vol. 19, pp. 17~37, 1982.
- [9] D. Roller, An Approach to Computer-Aided Parametric Design, Computer-Aided Design Vol. 23, No. 5, 1991.
- [10] 鈴木宏正, 木村文彦, 佐田登志夫, プロダクトモデルに基づく幾何学的拘束関係の記述と寸法処理への応用, 精密工学会誌, Vol. 52, No. 6, 1986.
- [11] 乾正和, 木村文彦, 依存情報を用いたモデル作成過程の表現とその可変ソリッドモデリングへの応用, 情報処理学会論文誌, Vol. 31, No. 7, 1990.

執筆者紹介 大高 哲彦 (Akihiko Ohtaka)

昭和 20 年生, 45 年東京工業大学大学院 理工学研究科修了, 同年日本ユニシス(株)入社。以来, CAE システムおよび CAD/CAM システムの開発, 適用サービスに従事し現在に至る。現在, 製造工業システム本部 CAD/CAM システム部部长, ISO TC184/SC4 PMAG 代表。



CADCEUS のデータ構造

The Data Structure of CADCEUS

白 川 忠 彦

要 約 次世代 CAD/CAM を実現するためのデータベースは、製品モデルの実現、データの一元化、ハイブリッドモデルの実現、設計・製図の自動化の支援、並行設計の支援等の機能を備えている必要がある。

CADCEUS は、これらの要請をふまえて、

- ワークスペース (作業空間)
- オブジェクト (製品, 部品, 図面等)
- 幾何要素, 位相要素等, 形状を表現する要素
- 製品属性等の要素
- 形状特徴, パラメトリック設計のための要素
- CAM, CAE, CG 等, 個々の応用独自の要素
- およびこれらの要素間の関係

等により実現している。本稿では、これらのデータ構造とその特徴, 利点等について説明する。

Abstract The database management system leading to a next-generation CAD/CAM system is required to functionally support product modeling, the flow of consistent data from early designing to manufacturing and hybrid modeling in addition to automated designing/drafting and concurrent design process. In response to these requirements, CADCEUS is so created as to provide the following :

- work space
- objects (including assembly, parts and drafts)
- geometric and topological entities to represent product shapes product attributes
- entities to define the features of shapes and to allow parametric design
- application-oriented entities for CAM, CAE and CG
- relationships between these entities.

This paper presents an overview of the data structure of CADCEUS and its characteristics as well as its advantages.

1. はじめに

次世代 CAD/CAM が持つべき特徴は『CADCEUS の概要』で触れられているが、それらを支えるデータモデルに関しては次のようなことが要請されている。

- ① データモデルを充実し、形状データのみならず各種の応用特有の情報およびその周辺の管理情報を合わせたモデルを表現できること (製品モデル)。
- ② 設計から生産準備, 解析等の全工程, 全応用分野で, 同一のデータを重複させないこと (データの一元化)。
- ③ 複数のモデリング形態で混在できること (ハイブリッド・モデル)。
- ④ 自動設計・自動製図へのフレームワークが整っていること (自動化支援)。

- ⑤ 複数の作業者が同一の設計作業を分割し並行して実施できること（並行設計）。

このような要請に応えるために開発された CADCEUS のデータモデルの構造と特徴を紹介する。

2. 特徴

CADCEUS のデータモデルは次のような特徴を備えている。

- 1) 製品モデルを構築している。

製品モデルの意義・体系については、本技報『次世代 CAD/CAM システム「CADCEUS」の概要』を参照されたい。

製品モデルにより、

- ・設計・製図の自動化の支援
- ・製品の構造のより厳密な定義
- ・管理情報の充実
- ・形状と非形状情報の整合性の維持

を図ることができる。そのため、設計・生産の場でのデータ交換がスムーズに行われ、生産管理等、他分野との情報授受を容易に行うことができる。

- 2) 設計から生産準備までのすべての応用分野を一貫した同一のデータモデルで実現している。

従来のシステムでは CAD, CAM, CAE, CG 等のそれぞれの分野での局所最適化が行われているため、個々のデータ表現が異なることが多く、たとえば設計から生産準備への段階でデータ変換が必要になり、データモデルとしての一貫性がなかった(自動化の島)。CADCEUS では、CAD と CAM に限らず、CAE や CG 等のすべての応用分野のデータを一貫した同一のデータモデルで表現している(図 1)。

CADCEUSデータ

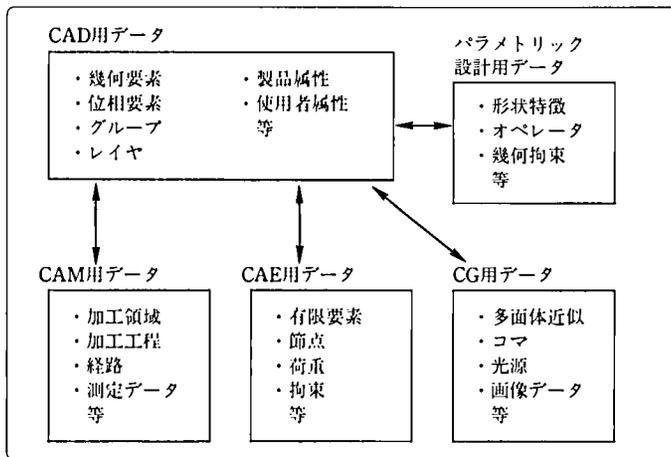


図 1 応用分野とデータの関連

Fig.1 Relation among applications and pertaining data

このデータの一元化による利点は次のような点である。

- ・ 工程間のデータ変換が不要になる。
- ・ 上流工程の意図を下流工程に正確に伝達できる。
- ・ データ変換による情報のロスや時間的なずれがない。

3) 種々のモデリング形態で共通したデータモデルである。

CADCEUS では、二次元または三次元のワイヤフレーム・モデリング、サーフェス・モデリングおよびソリッド・モデリングが可能であるが、これらはすべて共通したデータモデルで表現している。かつまた、これらの混在モデル（いわゆるハイブリッド・モデル）も可能である。

従来のモデリング技術では、混在モデルといってもそれは多様体に限られていたが、『次世代 CAD/CAM システム「CADCEUS」の概要』でも説明したように、非多様体が必要とされる場面がある。CADCEUS の位相構造は、これらの非多様体を容易に表現している。

4) 形状特徴によるモデリングを実現している。

形状特徴の意義と利点については『次世代 CAD/CAM システム「CADCEUS」の概要』で説明したが、CADCEUS の形状特徴のデータ構造上の特徴は、

- ・ 通常の形状特徴が解析図形に限られることが多いのに対して、CADCEUS の形状特徴では、自由曲面・自由曲線も対象としている。
- ・ 形状特徴モデルとそうでないモデル（従来型モデル）との共存が可能である。
- ・ 形状特徴を、単にモデリングの援助とするだけでなく、その『穴』、『フィレット』等の形状の持つ意味を他の応用に利用することができる。

等である。

5) 並行設計作業を支援している。

一つの製品または部品の設計を複数の作業者が同時並行して進めるため、製品を個々の部品に分割すること、分割された部品を組み立てること、他の作業者の中間結果を参照すること等ができる（図 2）。

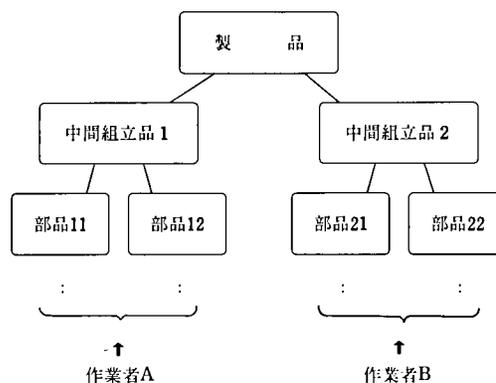


図 2 並行設計作業の例

Fig. 2 An example of concurrent design

3. データ構造

CADCEUS の最も基本的なデータ概念は、

- ・操作者の作業空間＝ワークスペース (WS)
 - ・WS 内の個々の部品または図面＝オブジェクト
 - ・オブジェクト内の個々の要素図形＝幾何要素等
- の 3 種である (図 3)。

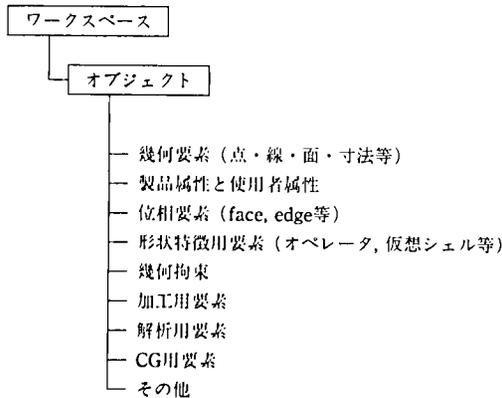


図 3 CADCEUS データ構造の概略

Fig. 3 Outline of CADCEUS data structure

ワークスペース、オブジェクトおよび幾何要素がどの応用分野にとっても必要なのに対して、その他の要素はそれぞれの応用分野にのみ必要である。

3.1 ワークスペース

CADCEUS のデータベースは、設計作業の適当なまとまり (仕事) によって分割される。このまとまりの個々をワークスペース (WS) と呼ぶ。ワークスペースはデータベースの排他制御の最小単位である。

一人の作業者が同時にいくつかのワークスペースを作業対象にしてもよい。同一のワークスペースを同時に複数の作業者が参照してもよいが、更新できるのは一人に限られる。一つのワークスペースを複数の作業者が同時利用することにより、並行設計が実現できる。

ワークスペースは、ドメインと呼ばれる論理的なワークステーションの集まりの中で管理される。ドメインは階層を持つことができ、これによって分散ネットワークによるサーバ・クライアントの階層構造を実現している。

ワークスペースはオブジェクトと呼ばれる部品、製品または図面の集まりである。

ワークスペースは作業者の作業空間としても使用できる他、全社で共通に利用される標準部品を貯える場所としても利用できる。

3.2 オブジェクト

オブジェクトは、ワークスペースの下位概念であり、形状等の意味を持った要素の集合である。

オブジェクトは、その性質と用途から次の 5 種類に分類される。

- ① パート : 一つの部品に対応する。
- ② アセンブリ : 中間組立品や製品に対応する。
- ③ シート : 1枚の図面に対応する。
- ④ プロジェクション : 図面中の一つの投影図(正面図, 側面図等)に対応する。
- ⑤ バリエント : パラメトリックな部品に対応する。パラメタに実現値を与えることにより, パート・オブジェクトになる。

1) オブジェクトの配置……オブジェクトには, 任意の位置に(必要なら回転, ミラー反転, スケール付けを行って)他のオブジェクトを配置することができる。図4はオブジェクト(A)に別のオブジェクト(B)を2箇所配置した例である。

配置された場合も, データはそれぞれに持つわけではなく, すべて一元管理されている。たとえば上図で, オブジェクトBの実データは一つのため, Bを変更するとAに配置されている二つのBも同時に変更されたことになる。

オブジェクトの配置は何階層になってもよい。

配置は, 組立品とその構成部品, 図面と部品等の構造を定義するために使用される。図5は, 製品とその部品の配置の例である。

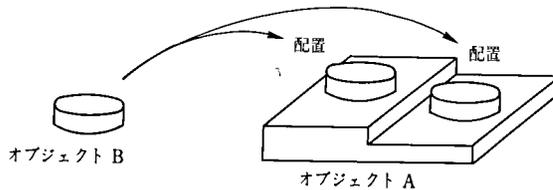


図4 オブジェクトの配置例

Fig. 4 An example of placement of an object

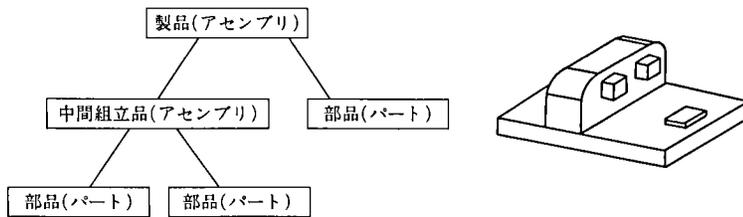


図5 配置の例1 (製品とその部品)

Fig. 5 An example of placement (product structure representation)

図6は図面と部品の配置例である。部品オブジェクトのモデル形状を変更すると, それはそのまま図面上にも反映され, 図面上の寸法も連動して変更される。

2) オブジェクト間の接続……配置関係のあるオブジェクト間に接続の関係を持つことができる(図7)。

この接続関係は, 公差の定義や機構解析で使用することができる。

3.3 幾何要素

幾何要素は, 形状を表す数値モデルを表現する上で中核をなす最も基本的な要素である。

1) 幾何要素の種類……幾何要素には次の種類がある。

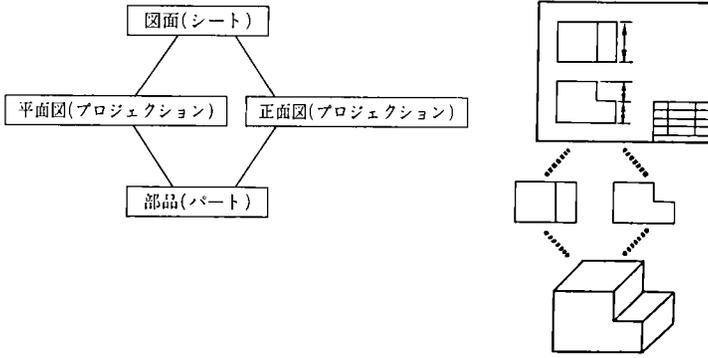


図 6 配置の例 2 (図面と部品)

Fig. 6 Another example of placement (associative drawing)

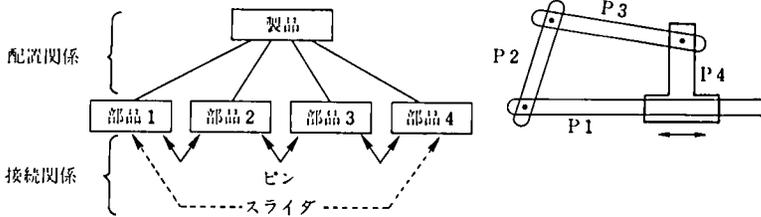


図 7 接続の例

Fig. 7 An example of connections

- ① 点 : 座標値 (X, Y, Z) を持つ点
- ② 線 : 線分, 無限直線, 半無限直線, 円, 円弧, 楕円, 楕円弧, 自由曲線の総称を線と呼ぶ。
- ③ 面 : 平面, 掃引面, 回転面, 円柱面, 円錐面, 球面, 自由曲面, またはこれらの一部分 (トリム面) の総称を面と呼ぶ。
- ④ 立体 : 直方体, 円錐体, 円柱体, 球体, トーラス体, 平行掃引体, 回転体, およびこれらのどれにもあてはまらない一般立体の総称である。
- ⑤ 集合形状 : 複数の線, 面, および立体の意味ある集まりとして, 次の概念がある。個々の構成要素も幾何要素であるが, これらの集まったものもまた別の幾何要素である (図 8)。

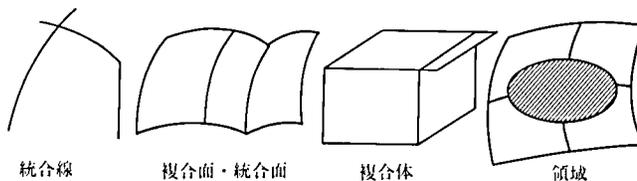


図 8 集合形状の例

Fig. 8 Examples of composite geometries

- ・統合線——連続したひとつながりの線の集まり。
- ・複合面——互いに境界線を介して隣接している面の集まり。
- ・統合面——複合面の部分集合であり、使用者が意図をもって集合化したもの。
- ・複合体——面、複合面および立体の連結した集まり。
- ・領域——統合面とその一部を特定する面上の統合線の集まり。複合面、複合体はシステムが自動的に構築するものであり、その他は使用者の意図によって任意に定義されるものである。

⑥ 製図要素：製図用の要素であり、次のものからなる。

- ・寸法——2点間寸法、半径・直径寸法、角度寸法等。
- ・記号——注記、幾公差、仕上げ記号等。
- ・表——表題欄、部品表等に用いられる。
- ・ハッチング——断面図の切り口の表現等に用いられる。

幾何要素は、二次元形状または三次元形状である。二次元形状の場合、創成時の座標系のビュー（ $+X+Y$, $+X-Y$ 等、24種類）ごとに定義される。三次元形状の場合は常に（ X , Y , Z ）の3軸の値を持つ。一つのオブジェクトの中に二次元形状と三次元形状が混在していてもよい。

2) グループとレイヤ……幾何要素の集合の概念にグループとレイヤがある(図9)。

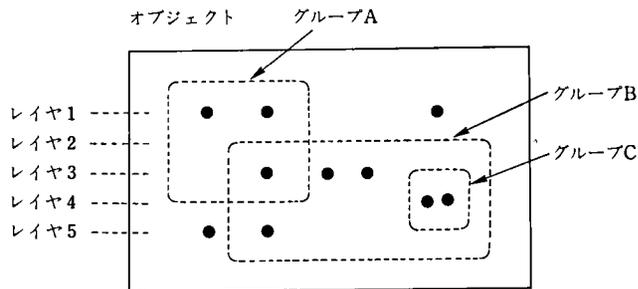


図9 グループとレイヤ
Fig.9 Groups and layers

一つのグループまたはレイヤに含まれるすべての構成要素に対し、一括した下記の操作が可能である。

- ・表示制御（表示・非表示）
- ・表示色制御
- ・図形処理（複写、移動、削除等）

各幾何要素は、複数のグループに所属できる。またどのグループにも属さなくてもよい。グループとは別に、要素を意味や性格によって分類するための概念がレイヤである。複数の見方での分類が許される。この見方（レイヤビュー）ごとに、各要素はどこかのレイヤに属している。

これらは、

- ・幾何要素群をまとめて操作する時の補助手段

- ・幾何要素群の意味あるまとまり（部品の部分集合の定義等）
 - ・個々の幾何要素の意味・性格（意匠用形状と加工用形状の区別等）の定義に利用することができる。
- 3) 幾何要素間の関係……幾何要素間には、システムによりいくつかの関係が保持されている。次はその一例である。
- ・面上線関係、相貫線関係：面の上の線，二つの面の相貫線の関係である。
 - ・評価形状関係：面とその面を評価するための線（ハイライト線等）の関係である。
 - ・一致区間関係：二つの線の一致する区間を認識する。
 - ・製図要素と形状の関係：製図要素とその測定の対象形状（形体）との関係である。
- これらの関係により、
- ・要素間の意味ある関係をシステムが認識し、整合性を保証する。
 - ・形状の変更により寸法，注記等の製図要素を自動的に変更する。
- 等が実現できている。
- 4) 幾何要素と他の要素との関係……幾何要素は、これから述べる他の種類の要素と密接な関係を持つとともに、CADCEUS データモデルの中心として、すべてのアプリケーションの中核的役割を果たしている。たとえば、形状特徴モデリングによる形状特徴の変更は、幾何要素の変更を引き起こし、それは、他の応用にとってのデータ（加工用要素とか、解析用要素等）への変更を促すことになる（図 10）。

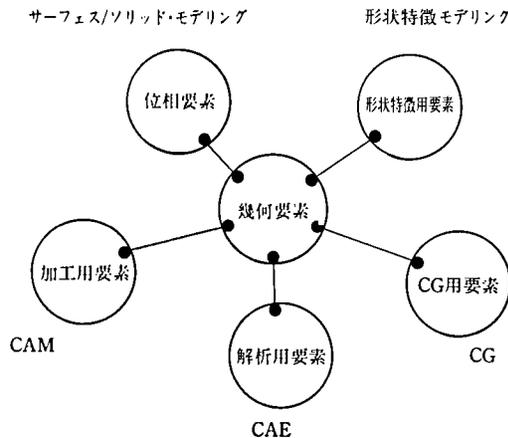


図 10 幾何要素と他の要素との関係

Fig.10 Relations among geometric entities and other entities

3.4 製品属性と使用者属性

- 1) 製品属性……幾何要素が製品の形状を表現しているのに対して、製品の持つ性質を表現する次のような属性がある。

① 材質

- ② 板厚
- ③ 表面粗さ
- ④ 公差

これらは、個々のオブジェクト、幾何要素、グループに対してまたはこれらの間の関係に対して付与することができる。たとえば、部品オブジェクトに材質を、二つの幾何要素間に公差を定義できる。

これらの製品属性は次のような場面で利用することができる。

- ・ 図面作成（寸法線の公差、部品表の材質等）
- ・ CAM 用情報（加工条件、工具、ワークサイクル等）
- ・ FEM 用情報（板厚、材質等）
- ・ CG 用情報（材質感の表現等）

- 2) 使用者属性……オブジェクト、配置、レイヤ、グループ、座標系および幾何要素に対して、使用者が任意の属性値を付与することができる(図 11)。

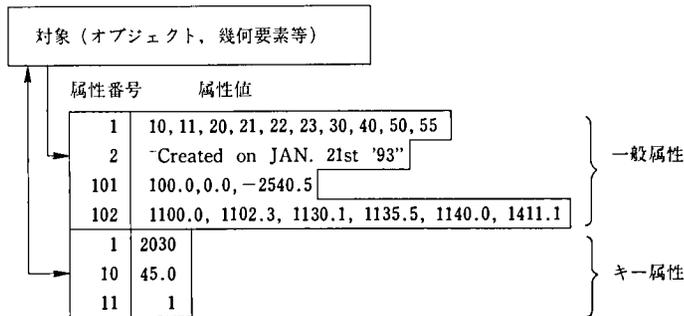


図 11 使用者属性の例

Fig. 11 Examples of user defined attributes

属性値には次の 2 種類がある。

- ① 長さや型が自由な一般属性
大量の属性を付与するのに適している。
- ② 一つの整数値または実数値であるが、属性値をキーにしての検索が可能なキー属性
属性値をキーにしての検索をするのに適している。

どちらの型の属性にも個々の要素に対して複数の属性値を付けることができる。これら複数の属性値は属性番号によって識別できる。

これらの属性は

- ・ ユーザ固有の管理情報、加工情報、ノウハウの蓄積
- ・ 大量の部品中の検索情報（たとえば、呼び径をキーにして該当標準部品を検索する）

等に利用できる。

3.5 位 相 要 素

サーフェス/ソリッド・モデリングにおいて、

面および立体の構造

面および立体の隣接関係

面の有効部分 (トリム面)

等表現するために、位相要素がある。

位相要素には次の種類がある。

- ① vertex…………線の両端点に対応する。
- ② vertex_loop…円錐体の頂点に対応する。
- ③ edge…………線に対応する。
- ④ edge_loop……1 本以上の edge の閉じたループ。
- ⑤ face…………面に対応する。一つの外周の edge_loop と、穴のための 0 個以上の内周の edge_loop からなる。
- ⑥ shell…………連結した face の集まり。
- ⑦ body…………立体に対応する。一つの外周の shell と、空洞のための 0 個以上の内周の shell からなる。
- ⑧ complex_body…連結した body の集まり。

位相要素と幾何要素との対応関係は図 12 のとおりである。

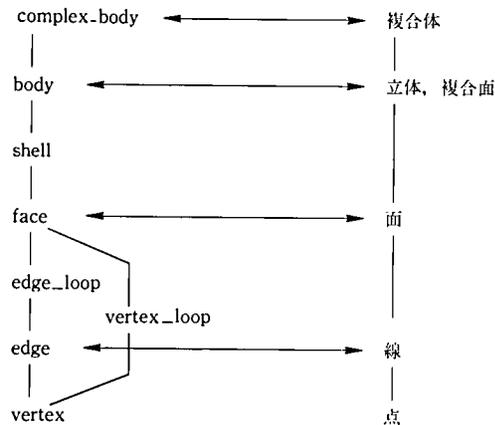


図 12 位相要素と幾何要素の関係

Fig. 12 Relations between topological entities and geometric entities

たとえば、面とその四つの境界線の構造は 4 種類の位相要素 (face, edge_loop, edge および vertex) で図 13 のように表現されている。

また、二つの面の隣接関係は図 14 のような位相関係により表現されている。

このように、境界線のうちの隣接部分は位相が分割され、それらにメイト関係が付けられる。ところで、このメイト関係は通常は二つの edge 間に付いている。メイト関係が三つ以上の edge 間にあると、それは非多様体ということになる。

面をトリムした場合、トリムされた側の部分は捨てられずデータベース内には『無効面』として残される。これはトリム戻しを可能にするためである。無効面は通常表示されていないが、トリム戻しの指示により元の面に併合される。このトリム面と無効面との隣接関係もここで説明したような位相構造で表現されている (図 15)。

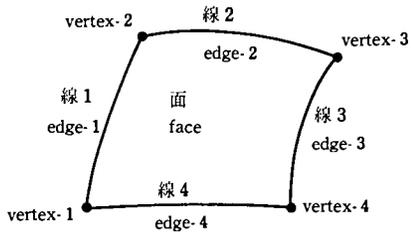
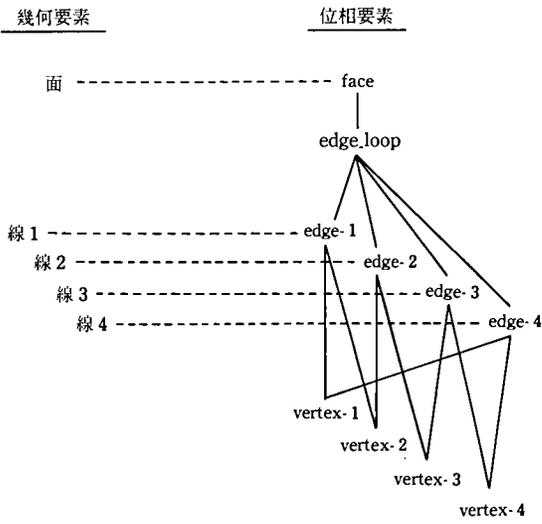


図 13 位相要素による面の表現

Fig. 13 Representation of a surface by topological entities

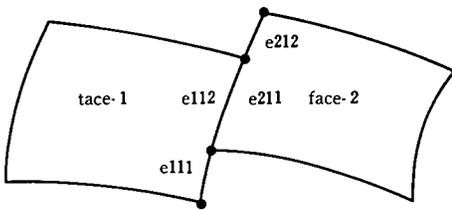
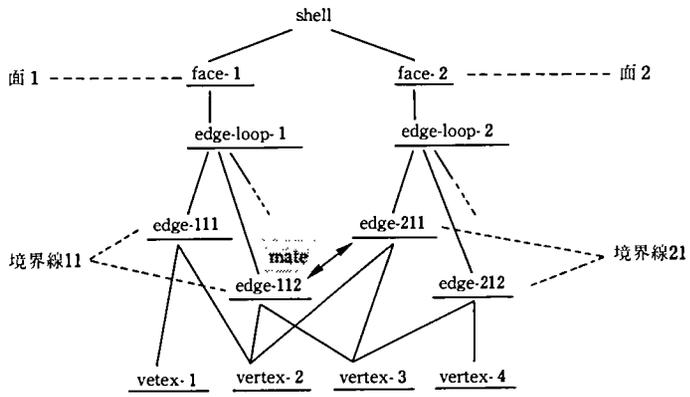


図 14 位相要素による面の隣接関係の表現
(直接関係のない要素は省略した)

Fig. 14 Representation of connection of two surfaces

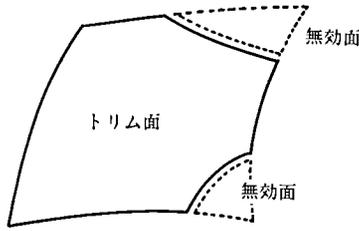


図 15 位相要素によるトリム面の表現

Fig. 15 Representation of a trimmed surface

3.6 形状特徴

形状特徴のデータ構造を、「穴」を例にとって説明すると概略次のようになっている (図 16).

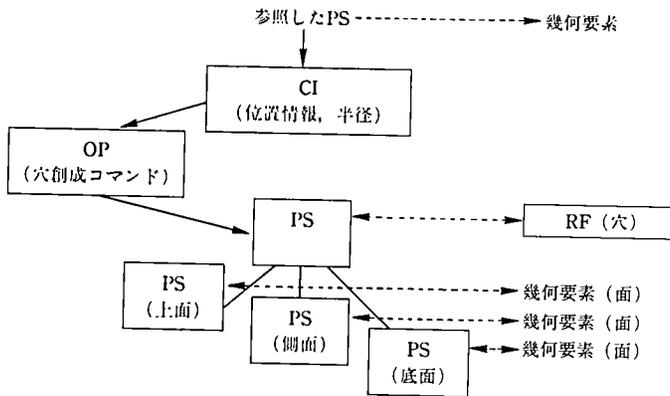


図 16 形状特徴用要素の例

Fig. 16 Entities for form feature

① オペレータ (OP; Operator)

個々のコマンド (ここでは, 『穴創成コマンド』) を識別するものである。

② コマンド・インスタンス (CI; Command Instance)

穴を創成するための必要情報。位置情報, 穴径等のパラメータ, その他環境情報。参照した図形との関係は幾何要素とではなく, 仮想シェルとの間に持つ。

③ 仮想シェル (PS; Pseudo Shell)

穴を構成する幾何要素群の意味構造を保持する要素群。穴を創成すると, 穴の側面が創成され, 逆に穴によって取り除かれた面ができるが, これらの個々の面 (=幾何要素) の構造を保持する要素群である。

④ 形状特徴の実体 (RF; Realized Feature)

生成した穴の実体を代表する要素。

オペレータ間には, 次のような関係がある。

時系列関係: コマンドを実行した順序関係

参照関係: コマンド操作時に参照した要素を生成したコマンドとの関係

干渉関係 : コマンド実行結果により影響を及ぼした要素を生成したコマンドとの関係

穴の半径を変更しようとする時、それはおよそ次の手順で実行される。

- ① その穴に干渉していたすべての形状特徴を、干渉がなかった状態に戻す。
- ② 次の形状特徴を削除する。
 - ・その穴自身
 - ・その穴に干渉していた形状特徴
 - ・その穴を参照して生成していた形状特徴

ただし、ここで削除されるのは、その形状特徴と関係する幾何要素だけであり、オペレータ、コマンド・インスタンスおよび仮想シェルはそのまま残される。

- ③ ②で幾何要素が削除されたオペレータをその時系列の順に再実行する。このとき生成された幾何要素は最初に生成されたときの仮想シェルと関係付けられる。このことにより、他の要素との関連が穴の半径の変更前と同様に保たれる。

3.7 幾何拘束

パラメトリック設計のために、幾何要素間に拘束を付けることができる。図 17 は単純な長方形の拘束の例である。パラメータ (図では、X, Y, Z, L1, L2) の実現値を指示することにより、これらの拘束を維持した形状を生成することができる。

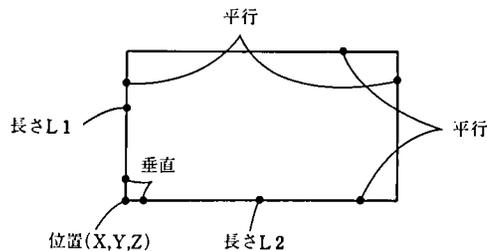


図 17 幾何拘束の例

Fig. 17 An example of geometric constraint

幾何拘束には要素間の関連を表す構造拘束と、距離等の量を規定する幾何量拘束の2種類がある。表 1 はその一例である。

これらの拘束は、パラメトリックモデリングの時にシステムが自動的に生成する他、使用者が定義、変更、削除することもできる。

3.8 その他の応用データ

その他、個々の応用ごとに専用のデータがある。

CAM 用 : 加工領域, 基本経路等

FEM 用 : 有限要素, 節点, 荷重, 拘束等

CG 用 : 多面体近似データ, 材質, 質感, 光源等

これらのデータは、対応する幾何要素との関係を保持している。一例として、図 18 に CAM 用のデータ構造をあげる。

表 1 幾何拘束の一覧
Table 1 List of geometric constraints

	拘束名	引数	属性値	意味
構造拘束	onL	点, 線	true	点は線の上にある
	onC	点, 円	true	点は円周上にある
	parallel	線, 線	同 逆	2線は平行
	perpen	線, 線	+ -	2線は垂直
	symmetricP	点, 線, 線	true	2点は線に対して対称
	symmetricL	線, 線, 線	+ -	2線は線に対して対称
	symmetricC	円, 円, 線	+ -	2円は線に対して対称
幾何量拘束	tangentLC	線, 円	同 逆	線は円に接する
	tangentCC	円, 円	同 逆	2円は接する
	angle	線, 線	角度	2線間の角度
	distancePP	点, 点	距離	2点間の距離
	distancePL	点, 線	距離	点と線間の距離
	distanceLL	線, 線	距離	2線間の距離
	radius	円	半径	円の半径
	direction	線	角度	線の水平線に対する角度
	coord	点	座標値	点の座標値
	shapeL	線	座標値	線の端点の座標値
	shapeC	円	座標値	円の中心の座標値

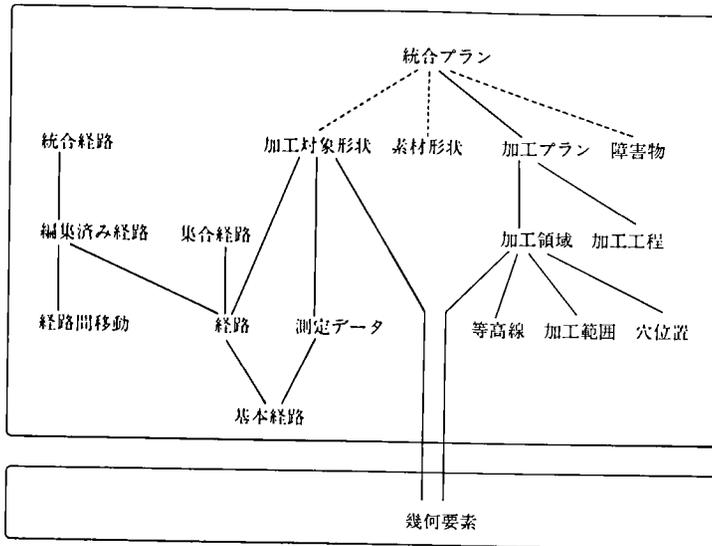


図 18 CAM 用データ構造
Fig. 18 Structure of CAM data

4. おわりに

ここまで、CADCEUS が実現しているデータ構造とそれがどのような製品モデルやデータの一元化を果たしているかを説明してきた。

しかし次のように、現在の技術、環境ではまだなお実現できていない課題も多い。

- ・設計変更やそれに伴うバージョンの管理がシステム化されていない。

- ・設計意図やノウハウの蓄積は使用者の管理に任されている。
 - ・段階的詳細化または曖昧な表現に関するデータモデルが構築されていない。
- これらは、今後の課題としてとらえたい。
また、データベースハンドラについての紹介は次の機会に譲りたい。

-
- 参考文献 [1] 鈴木宏正, 「プロダクトモデルを用いた情報処理技術」, ユニシス研究会, 第14回 UNICAD 分科会例会資料, 平3.3.11.
[2] 小西惣一郎, 「製品モデルを本格的に採用した次世代統合 CAD/CAM システム」, 日経 CG, 1992.3月号.
[3] 日本ユニシス, 「CADCEUS における新技術の概要—モデリング関連」, 伊豆エグゼクティブセミナー資料.
[4] 日本ユニシス, 「CADCEUS 解説書—基本編」.

執筆者紹介 白川 忠彦 (Tadahiko Shirakawa)
昭和22年生, 44年東京大学理学部数学科卒業, 同年日本ユニシス(株)入社, 58年よりCAD/CAMシステムの開発に従事, 現在CAD/CAMシステム部に所属,



有理 B スプライン曲線・曲面のパラメタ変換とウェイト操作

On Reparametrization and Weight Manipulation of Rational B-spline Curves and Surfaces

中 邨 博 之

要 約 NURBS (有理 B-spline) 曲線・曲面は多項式 B-spline 曲線・曲面にはないウェイトと呼ばれる自由度をもっている。制御点を動かさずにウェイトを変化させると、一般には曲線・曲面の形状は変化する。しかし、NURBS を用いた曲面創成に際しては、枠線の形状を変化させずにウェイトを操作する必要が生じる。本稿では、曲線・曲面の形状を保つようなウェイト操作の方法を調べる。その結果として、ある条件を満たすパラメタの区分的一次分数変換を施しても NURBS 曲線・曲面の制御点は変化しないことを示し、その際のウェイトの変換式を与える。

Abstract Compared with polynomial B-spline curves and surfaces, NURBS (Non Uniform Rational B-spline) curves and surfaces have additional degrees of freedom called weights. If weights are changed while fixing control points, the shape of the relevant curve or surface generally changes. But, when NURBS representation is used for practical design problems such as surface interpolation with given curves, manipulation of weights which leaves the shapes of the given curves unchanged becomes necessary. The main purpose of this paper is to investigate the means of weight manipulation which preserves relevant geometry. It is shown under certain conditions that piecewise linear fractional transformation of parameters does not change relevant geometry, and the transformation formulae for weights under such reparametrization is given.

1. はじめに

NURBS (有理 B-spline) 曲線・曲面は非有理 (多項式) B-spline 曲線・曲面に比べて、円・楕円や球面・楕円体面等より広い範囲の曲線・曲面を敲密に表現できるといふ利点を持つ反面、ウェイトと呼ばれる新しい自由度が付加されているために、その制御という新しい問題が生ずる。

たとえば、曲面創成の基本的な方法の一つに、四辺の枠線から内挿して曲面を創成する方法がある。その枠線が NURBS 曲線として与えられた場合に、既知の方法を適用して曲面を張るためには、隣接する枠線の共有する端点におけるウェイトが等しくなっていなければならない。もしウェイトが異なっていた場合には、与えられた枠線の形状を変えないでウェイトを変更する必要が生じる。

NURBS 曲線・曲面で、形状を変えずにウェイトを変化させる方法として、パラメタの一次分数変換がある。Patterson は文献^[1]で有理 Bézier 曲線について、また Lee と Lucian は、文献^[2]で有理 B-spline 曲線についてパラメタの一次分数変換の結果ウェイトがどのように変化するかという公式を与えた。

有理 B-spline 曲線の場合、また複数セグメントの有理 Bézier 曲線の場合にはパラメタ空間全体を一つの変換式で変換することよりも、各セグメントごとに別々の一次

分数変換を施す方がより大きなウェイト変更の自由度を持つ。ただし、各セグメントごとにまったく独立にパラメタ変換をしたのでは曲線のパラメタ連続性が失われてしまう。そこで、曲線のパラメタ連続性(とくに C^1 連続性)を保つような、セグメントごとの一次分数変換(区分的一次分数変換)を考える必要がある。

本稿では、2章で定義、記号の準備をし、3章、4章で文献^{[1][2]}の主な結果を紹介したあと、5章で有理 Bézier, B-spline 曲線のパラメタの区分的一次分数変換を考察する。6章で曲面の場合の結果について言及し、7章では5章の結果を曲面創成の問題に応用する。

2. 準備

2.1 Bézier 曲線・曲面

n 次の Bernstein 基底関数 $B_{i,n}(t)(i=0, \dots, n)$ は、次の式により定義される。

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

ここで、 $\binom{n}{i}$ は 2 項係数。

これを用いて、 n 次の Bézier 曲線 $C(t)$ は、次の式により定義される。

$$C(t) = \sum_{i=0}^n B_{i,n}(t) P_i \quad (2-2)$$

ここで、 $P_i(i=0, \dots, n)$ は制御点と呼ばれる平面上または空間内の点(の位置ベクトル)である。 t はこの曲線のパラメタで、 $0 \leq t \leq 1$ の範囲を動く。

また、 u 方向 m 次、 v 方向 n 次の Bézier 曲面は次の式により定義される。

$$S(u, v) = \sum_{i=0}^m \sum_{j=0}^n B_{i,m}(u) B_{j,n}(v) P_{i,j} \quad (2-3)$$

$P_{i,j}(i=0, \dots, m, j=0, \dots, n)$ は空間内の点で、曲線の場合と同様に制御点と呼ばれる。 u, v はこの曲面のパラメタで、 $0 \leq u, v \leq 1$ の範囲を動く。

2.2 有理 Bézier 曲線・曲面

式(2-2)の右辺はパラメタ t に関する多項式であるが、これを有理式へと拡張することができる。

各制御点 P_i に対応して実数 w_i が与えられたとき、次の式によって曲線 $C(t)$ を定義する。

$$C(t) = \frac{\sum_{i=0}^n B_{i,n}(t) w_i P_i}{\sum_{i=0}^n B_{i,n}(t) w_i} \quad (2-4)$$

この曲線 $C(t)$ を有理 Bézier 曲線と呼ぶ。 $w_i(i=0, \dots, n)$ はウェイト (weight) と呼ばれ、通常 $w_i > 0$ が仮定される。 $w_0 = w_1 = \dots = w_n = 1$ の場合には式(2-4)は式(2-2)と一致する。ウェイト w_i を一斉に定数倍しても式(2-4)は変わらない。すなわち、ウェイトはその比のみに意味があることに注意する。

有理 Bézier 曲線は非有理(多項式) Bézier 曲線に比べて、ウェイトの自由度の分、より広い範囲の曲線を表現することができる。たとえば円弧や楕円弧を非有理 Bézier

曲線では厳密に表現することはできないが、有理 Bézier 曲線を使えばできる。

曲面の場合にも同様に、各制御点 $P_{i,j}$ に対応してウェイト $w_{i,j}$ が与えられたとき、

$$S(u, v) = \frac{\sum_{i=0}^m \sum_{j=0}^n B_{i,m}(u) B_{j,n}(v) w_{i,j} P_{i,j}}{\sum_{i=0}^m \sum_{j=0}^n B_{i,m}(u) B_{j,n}(v) w_{i,j}} \quad (2-5)$$

により、有理 Bézier 曲面を定義することができる。

2.3 B-spline 曲線・曲面

k を自然数とし、 $T = \{t_0, t_1, \dots, t_p\} (p \geq k)$ を実数の非減少列で次の条件を満たすものとする。

・各 t_i の T 中に現れる回数 (t_i の多重度と呼ぶ) は k を越えない。

このとき、オーダー k の (i. e. 次数 $k-1$ の) B-spline 基底関数 $N_{i,k,T}(t) (i=0, \dots, N; N=p-k)$ は次の式により定義される^[3]。

$$N_{i,k,T}(t) = (t_{i+k} - t_i) [t_i, \dots, t_{i+k}] (\cdot - t)_{+}^{k-1} \quad (2-6)$$

ここで、一般に関数 $f(x)$ に対し、 $[t_i, \dots, t_{i+k}]f$ は $f(x)$ の k 階差分商と呼ばれるもので、 $f(x)$ と $x = t_i, \dots, t_{i+k}$ で一致するような k 次以下の多項式 (これは唯一つ存在する) の最高次係数 (x^k の係数) を表す。 ($T_i = \{t_i, \dots, t_{i+k}\}$ が真増大列でなく、重複して現れる項がある場合には、“ $f(x)$ と $g(x)$ が $x = t_i, \dots, t_{i+k}$ で一致する”とは次のことを意味する： t_j の T_i 中の多重度を m_j とするとき、 $f(x)$ と $g(x)$ は $x = t_j$ における第 $0, \dots, m_j - 1$ 階微分係数が一致する ($j = i, \dots, i+k$).)

また、 $(x-t)_{+}^{k-1}$ は切断べき関数と呼ばれ、

$$(x-t)_{+}^{k-1} = \begin{cases} (x-t)^{k-1}, & \text{for } x > t \\ 0, & \text{for } x \leq t \end{cases} \quad (2-7)$$

で定義される。式(2-6)の右辺の“ \cdot ”は「その場所の変数」(名前は任意、たとえば x) について差分商をとることを示している。

$N_{i,k,T}(t)$ は t に関する区分的多項式であり、その区分点のパラメタは t_i, \dots, t_{i+k} である。 T はノット列(knot sequence)と呼ばれる。まぎれる恐れのない時には $N_{i,k,T}(t)$ の T を省略して $N_{i,k}(t)$ と書くこともある。

Bernstein 基底関数を用いて Bézier 曲線・曲面が定義されたのと同様にして、B-spline 基底関数を用いて B-spline 曲線・曲面が定義される。

$$C(t) = \sum_{i=0}^N N_{i,k,T}(t) P_i \quad (2-8)$$

$$S(u, v) = \sum_{i=0}^M \sum_{j=0}^N N_{i,k,U}(u) N_{j,k,V}(v) P_{i,j} \quad (2-9)$$

ノット列 T が $\underbrace{\{0, \dots, 0\}}_{k \text{ 個}}, \underbrace{\{1, \dots, 1\}}_{k \text{ 個}}$ のとき、B-spline 基底関数 $N_{i,k,T}(t)$ は Bernstein

基底関数 $B_{i,k-1}(t)$ と一致する。したがって、B-spline 曲線・曲面は特別の場合として

Bézier 曲線・曲面を含む。また、ノット列が $\underbrace{\{\tau_0, \dots, \tau_0\}}_{k \text{ 個}}, \underbrace{\{\tau_1, \dots, \tau_1\}}_{k-1 \text{ 個}}, \dots, \underbrace{\{\tau_{r-1}, \dots, \tau_{r-1}\}}_{k-1 \text{ 個}}$

$\underbrace{\{\tau_r, \dots, \tau_r\}}_{k \text{ 個}}$ のように、両端のノットの多重度が k で、中間のノットの多重度がすべて

$k-1$ (次数) である場合には、B-spline 曲線は Bézier 曲線を複数個つなぎ合わせたも

の (複数セグメント Bézier 曲線) となる。

2.4 有理 B-spline 曲線・曲面

有理 Bézier 曲線・曲面の定義と同様に、有理 B-spline 曲線・曲面が定義される。

$$C(t) = \frac{\sum_{i=0}^N N_{i,k,\tau}(t) w_i P_i}{\sum_{i=0}^N N_{i,k,\tau}(t) w_i} \quad (2-10)$$

$$S(u, v) = \frac{\sum_{i=0}^M \sum_{j=0}^N N_{i,k,u}(u) N_{j,l,v}(v) w_{i,j} P_{i,j}}{\sum_{i=0}^M \sum_{j=0}^N N_{i,k,u}(u) N_{j,l,v}(v) w_{i,j}} \quad (2-11)$$

$w_i, w_{i,j}$ は制御点 $P_i, P_{i,j}$ のウェイトと呼ばれ、Bézier の場合と同様、通常正の実数が用いられる。

有理 Bézier 曲線(2-4)や有理 B-spline 曲線(2-10)において制御点 P_i は固定してウェイト w_i を変化させると、一般には曲線形状が変化する。しかし、曲線形状を保ちながらウェイトを変化させる方法がある。その方法がパラメタの一次分数変換である。

3. 有理 Bézier 曲線のパラメタ変換

本章では文献^[1]の主な内容を紹介する。

有理 Bézier 曲線(2-4)にパラメタの一次分数変換

$$t = \frac{as + b}{cs + d} \quad (3-1)$$

を施すことを考える。問題を単純にするために、パラメタ t が $0 \leq t \leq 1$ の範囲を (0 から 1 の方向に) 動く時にパラメタ s も $0 \leq s \leq 1$ の範囲を (0 から 1 の方向に) 動くとして仮定する。一次分数変換(3-1)でこの条件を満たすものは

$$t = \phi_K(s) = \frac{Ks}{(K-1)s + 1} \quad (3-2)$$

と表される。ここで、 K は正の実数 ($K = a/d$)。

そこで問題は、有理 Bézier 曲線(2-4)にパラメタの一次分数変換(3-2)を施して得られる曲線 $C(\phi_K(s))$ はどのようなものかということである。Patterson^[1]はこの問題に次の解答を与えた。

定理 1 (Patterson) 有理 Bézier 曲線(2-4)にパラメタの一次分数変換(3-2)を施して得られる曲線は、同じ制御点 P_i をもち、ウェイトが $w_0, Kw_1, \dots, K^n w_n$ の有理 Bézier 曲線である。すなわち、

$$C(\phi_K(s)) = \frac{\sum_{i=0}^n B_{i,n}(s) K^i w_i P_i}{\sum_{i=0}^n B_{i,n}(s) K^i w_i} \quad (3-3)$$

Patterson^[1]では、この定理を行列群の表現の視点から証明しているが、ここではより直接的な証明を与えておく。

証明 Bernstein 基底関数(2-1)がパラメタ変換(3-2)によりどのように変換されるかを計算すると、

$$\begin{aligned}
 B_{i,n}(\phi_K(s)) &= \binom{n}{i} \{\phi_K(s)\}^i \{1-\phi_K(s)\}^{n-i} \\
 &= \binom{n}{i} \left\{ \frac{Ks}{(K-1)s+1} \right\}^i \left\{ \frac{1-s}{(K-1)s+1} \right\}^{n-i} \\
 &= \frac{K^i}{\{(K-1)s+1\}^n} \binom{n}{i} s^i (1-s)^{n-i} \\
 &= \frac{K^i}{\{(K-1)s+1\}^n} B_{i,n}(s)
 \end{aligned} \tag{3-4}$$

したがって,

$$C(\phi_K(s)) = \frac{\sum_{i=0}^n B_{i,n}(\phi_K(s)) w_i P_i}{\sum_{i=0}^n B_{i,n}(\phi_K(s)) w_i} = \frac{\sum_{i=0}^n B_{i,n}(s) K^i w_i P_i}{\sum_{i=0}^n B_{i,n}(s) K^i w_i}$$

すなわち、 $C(\phi_K(s))$ は $C(t)$ と同じ制御点 P_i を持ち、対応するウェイトが $K^i w_i$ の有理 Bézier 曲線である。(証明終わり)

パラメタの変換を行っても、曲線の形状は変化しない。したがって、定理1は有理 Bézier 曲線の曲線形状を変えずにウェイトを変化させる方法を与えているものと見ることが出来る。すなわち、次のことが成り立つ。

系1 任意の正の実数 K が与えられたとき、有理 Bézier 曲線(2-4)において制御点 P_i は変えずにウェイト w_0, w_1, \dots, w_n を $w_0, Kw_1, \dots, K^n w_n$ で置き換えても曲線の形状は変化しない。

4. 有理 B-spline 曲線のパラメタ変換

本章では文献^[2]の主な内容を紹介する。

$I=[\tau_0, \tau_1]$ をすべてのノット (t_i) を含む有限区間とし、 $J=[\sigma_0, \sigma_1]$ をもう一つの有限区間とする。 J を I の上に写すような一次分数変換

$$t = \phi(s) = \frac{as+b}{cs+d}, \quad s \in J \tag{4-1}$$

を考える。($\Delta = ad - bc$ は 0 でないと仮定する。) その逆変換は、

$$s = \psi(t) = \frac{-dt+b}{ct-a}, \quad t \in I \tag{4-2}$$

である。(4-1), (4-2)の分母

$$\lambda(s) = cs + d, \quad \mu(t) = ct - a \tag{4-3}$$

はそれぞれ区間 J, I の上で 0 にならないと仮定する。

問題は有理 B-spline 曲線(2-10)にパラメタ変換(4-1)を施して得られる曲線 $C(\phi(s))$ はどのようなものか、ということである。

$$S = \{s_0, \dots, s_r\}, \quad s_i = \phi^{-1}(t_i) \quad (i=0, \dots, r) \tag{4-4}$$

とおくと、解答は次の定理で与えられる。

定理2 (Lee, Lucian) 有理 B-spline 曲線(2-10)にパラメタの一次分数変換(4-1)を施して得られる曲線 $\tilde{C}(s) = C(\phi(s))$ は、次の形の有理 B-spline 曲線である。

$$\tilde{C}(s) = \frac{\sum_{i=0}^N N_{i,k,S}(s) \tilde{w}_i P_i}{\sum_{i=0}^N N_{i,k,S}(s) \tilde{w}_i} \quad (4-5)$$

ここで、

$$\tilde{w}_i = w_i \prod_{j=1}^{k-1} \lambda(s_{i+j}) = \frac{w_i}{\prod_{j=1}^{k-1} \mu(t_{i+j})} \quad (i=0, \dots, N) \quad (4-6)$$

ただし、 \tilde{w}_i を一斉に定数倍する不定性を除く。

この定理の証明は付録で与える。

ノット列 T が $\{ \underbrace{0, \dots, 0}_{k \text{ 個}}, \underbrace{1, \dots, 1}_{k \text{ 個}} \}$ のとき、有理 B-spline 曲線 (2-10) は有理 Bézier 曲

線 (2-4) になるが、この場合 (3-2) の変換 ϕ_K に対して、

$$\lambda(s) = (K-1)s + 1$$

であり、(4-4) の S は

$$S = \{ \underbrace{0, \dots, 0}_{k \text{ 個}}, \underbrace{1, \dots, 1}_{k \text{ 個}} \}$$

であるから、ウェイトの変換式 (4-6) は、

$$\tilde{w}_i = K^i w_i \quad (i=0, \dots, k-1) \quad (4-7)$$

となり、定理 1 の結果と一致する。

5. 区分的一次分数変換

3, 4 章では、パラメタの一次分数変換の結果、有理 Bézier 曲線、有理 B-spline 曲線がどのように変換されるかという結果を紹介した。とくに、パラメタの一次分数変換は制御点は動かさず、ウェイトのみの変換を引き起こすということを見た。その結果、定理 1, 2 は曲線形状を変えずにウェイトを操作する自由度を記述しているものと解釈することができる。

本章では、さらに大きなウェイト操作の自由度を求めて区分的一次分数変換を考察する。

5.1 区分的一次分数変換と C^1 連続条件

3 章では区間 $[0, 1]$ を $[0, 1]$ へ写すような一次分数変換の一般形 (3-2) を求めたが、これを一般化して、区間 $[\sigma_0, \sigma_1]$ を区間 $[\tau_0, \tau_1]$ へ (σ_0 を τ_0 へ、 σ_1 を τ_1 へ) 写すような一次分数変換の一般形を求めると、次の形になる。

$$t = \phi_K^{[\sigma_0, \sigma_1; \tau_0, \tau_1]}(s) = \frac{(K\tau_1 - \tau_0)(s - \sigma_0) + \tau_0(\sigma_1 - \sigma_0)}{(K-1)(s - \sigma_0) + (\sigma_1 - \sigma_0)} \quad (5-1)$$

ここで K は正の実数。

その導関数および $s = \sigma_0, \sigma_1$ における微分係数を求めると、

$$\frac{d}{ds} \phi_K^{[\sigma_0, \sigma_1; \tau_0, \tau_1]}(s) = \frac{K(\tau_1 - \tau_0)(\sigma_1 - \sigma_0)}{\{(K-1)(s - \sigma_0) + (\sigma_1 - \sigma_0)\}^2} \quad (5-2)$$

$$\frac{d}{ds} \phi_K^{[\sigma_0, \sigma_1; \tau_0, \tau_1]}(\sigma_0) = \frac{K(\tau_1 - \tau_0)}{\sigma_1 - \sigma_0} \quad (5-3)$$

$$\frac{d}{ds} \phi_K^{[\sigma_0, \sigma_1; \tau_0, \tau_1]}(\sigma_1) = \frac{(\tau_1 - \tau_0)}{K(\sigma_1 - \sigma_0)} \quad (5-4)$$

となる。

K_0, K_1, \dots, K_{r-1} を正の実数, $\sigma_0 < \sigma_1 < \dots < \sigma_r, \tau_0 < \tau_1 < \dots < \tau_r$ として, 次の式で定義される変換を考える。これを区分的一次分数変換と呼ぶ。

$$t = \phi(s) = \begin{cases} \phi_{K_0, \sigma_1; \tau_1, \tau_2}^{[\sigma_0, \sigma_1]}(s), & \text{for } \sigma_0 \leq s \leq \sigma_1 \\ \phi_{K_1, \sigma_2; \tau_2, \tau_3}^{[\sigma_1, \sigma_2]}(s), & \text{for } \sigma_1 \leq s \leq \sigma_2 \\ \vdots \\ \phi_{K_{r-1}, \sigma_r; \tau_{r-1}, \tau_r}^{[\sigma_{r-1}, \sigma_r]}(s), & \text{for } \sigma_{r-1} \leq s \leq \sigma_r \end{cases} \quad (5-5)$$

この変換が区分的点 $\sigma_0, \sigma_1, \dots, \sigma_r$ で連続一階微分可能 (C^1 連続) であるための条件は, (5-3), (5-4) より,

$$\begin{aligned} \frac{\tau_1 - \tau_0}{K_0(\sigma_1 - \sigma_0)} &= \frac{K_1(\tau_2 - \tau_1)}{\sigma_2 - \sigma_1} \\ \frac{\tau_2 - \tau_1}{K_1(\sigma_2 - \sigma_1)} &= \frac{K_2(\tau_3 - \tau_2)}{\sigma_3 - \sigma_2} \\ &\vdots \\ \frac{\tau_{r-1} - \tau_{r-2}}{K_{r-2}(\sigma_{r-1} - \sigma_{r-2})} &= \frac{K_{r-1}(\tau_r - \tau_{r-1})}{\sigma_r - \sigma_{r-1}} \end{aligned} \quad (5-6)$$

である。

5.2 有理 Bézier 曲線のパラメタの区分的一次分数変換

本節では, 5.1 節で定義したパラメタの区分的一次分数変換を複数セグメント有理 Bézier 曲線に施した時にその表現式はどう変わるか, とくにウェイトはどう変換されるかという問題を考える。

まず, 2 章で定義した Bézier 曲線は区間 $[0, 1]$ 上で定義されたものであったが, これを任意の区間 $[\tau_0, \tau_1]$ 上で定義された Bézier 曲線に拡張する。

Bernstein 基底関数(2-1)において, t を $\frac{t - \tau_0}{\tau_1 - \tau_0}$ で置き換えることにより, 区間 $[\tau_0, \tau_1]$ 上の n 次の Bernstein 基底関数 $B_i^{[n, \tau_0, \tau_1]}(t)$ ($i=0, \dots, n$) が定義される。

$$B_i^{[n, \tau_0, \tau_1]}(t) = \frac{1}{(\tau_1 - \tau_0)^n} \binom{n}{i} (t - \tau_0)^i (\tau_1 - t)^{n-i} \quad (5-7)$$

これを用いて, 区間 $[\tau_0, \tau_1]$ 上で定義された有理 Bézier 曲線 $C^{[\tau_0, \tau_1]}(t)$ を,

$$C^{[\tau_0, \tau_1]}(t) = \frac{\sum_{i=0}^n B_i^{[n, \tau_0, \tau_1]}(t) w_i P_i}{\sum_{i=0}^n B_i^{[n, \tau_0, \tau_1]}(t) w_i} \quad (5-8)$$

と定義する。

次に, $\tau_0 < \tau_1 < \dots < \tau_r$ とし, 制御点 P_0, \dots, P_m および対応するウェイト w_0, \dots, w_m が与えられたとき, r 個のセグメントからなる n 次有理 Bézier 曲線を次の式で定義する。

$$C(t) = \begin{cases} C_0(t), & \text{for } \tau_0 \leq t \leq \tau_1 \\ C_1(t), & \text{for } \tau_1 \leq t \leq \tau_2 \\ \vdots \\ C_{r-1}(t), & \text{for } \tau_{r-1} \leq t \leq \tau_r \end{cases} \quad (5-9)$$

ここで, $j=0, \dots, r-1$ に対し, 曲線 $C_j(t)$ は制御点 $P_{jn}, P_{j+1}, \dots, P_{(j+1)n}$ と, 対応す

るウェイト $w_{jn}, w_{j+1}, \dots, w_{(j+1)n}$ から定まる区間 $[\tau_j, \tau_{j+1}]$ 上で定義された有理 Bézier 曲線とする。すなわち、

$$C_j(t) = \frac{\sum_{i=0}^n B_{i,n}^{\tau_j, \tau_{j+1}}(t) w_{jn+i} P_{jn+i}}{\sum_{i=0}^n B_{i,n}^{\tau_j, \tau_{j+1}}(t) w_{jn+i}} \quad (5-10)$$

j 番目のセグメント $C_j(t)$ と $j+1$ 番目のセグメント $C_{j+1}(t)$ で制御点 $P_{(j+1)n}$ を共有していることに注意する。

ここで定義した複数セグメント有理 Bézier 曲線は、ノット列 T を

$$T = \{ \underbrace{\tau_0, \dots, \tau_0}_{n+1 \text{ 個}}, \underbrace{\tau_1, \dots, \tau_1}_n, \dots, \underbrace{\tau_{r-1}, \dots, \tau_{r-1}}_n, \underbrace{\tau_r, \dots, \tau_r}_{n+1 \text{ 個}} \} \quad (5-11)$$

とした時の $n+1$ 階の有理 B-spline 曲線に他ならない。すなわち、このノット列 T によって定まる $n+1$ 階の B-spline 基底関数を $N_{i,n+1,\tau}(t)$ ($i=0, \dots, rn$) とするとき、(5-9) の $C(t)$ は、

$$C(t) = \frac{\sum_{i=0}^{rn} N_{i,n+1,\tau}(t) w_i P_i}{\sum_{i=0}^{rn} N_{i,n+1,\tau}(t) w_i} \quad (5-12)$$

と書ける。

さて、ここで考えたい問題は、この複数セグメント有理 Bézier 曲線(5-12)に対してパラメタの区分的一次分数変換(5-5)を施せば曲線の表現式はどのようになるか、とくにウェイトはどう変換されるかということである。

$$S = \{ \underbrace{\sigma_0, \dots, \sigma_0}_{n+1 \text{ 個}}, \underbrace{\sigma_1, \dots, \sigma_1}_n, \dots, \underbrace{\sigma_{r-1}, \dots, \sigma_{r-1}}_n, \underbrace{\sigma_r, \dots, \sigma_r}_{n+1 \text{ 個}} \} \quad (5-13)$$

とおくとき、その結果は次のようになる。

定理 3 式(5-12)の曲線 $C(t)$ にパラメタ変換(5-5)を施して得られる曲線 $\tilde{C}(s) = C(\phi(s))$ は次の式で表される。

$$\tilde{C}(s) = \frac{\sum_{i=0}^{rn} N_{i,n+1,s}(s) \tilde{w}_i P_i}{\sum_{i=0}^{rn} N_{i,n+1,s}(s) \tilde{w}_i} \quad (5-14)$$

ここで、定数倍の不定性を除いて、

$$\tilde{w}_i = \begin{cases} K_0^i w_i & (0 \leq i \leq n) \\ K_0^n K_1^{i-n} w_i & (n+1 \leq i \leq 2n) \\ \vdots \\ K_0^n \cdots K_{r-2}^{i-(r-1)n} K_{r-1}^{i-(r-1)n} w_i & ((r-1)n+1 \leq i \leq rn) \end{cases} \quad (5-15)$$

すなわち、結果はやはり r 個のセグメントからなる有理 Bézier 曲線であり、その制御点 P_i は変化せず、ウェイトは式(5-15)の変換を受ける。

この定理の証明は付録で与える。

パラメタ変換(5-5)が C^1 連続条件(5-6)を満たす場合には、もとの曲線(5-9)が区分点 $t = \tau_1, \dots, \tau_{r-1}$ において C^1 連続のとき変換後の曲線も C^1 連続となることに注意する。

5.3 有理 B-spline 曲線のパラメタの区分的一次分数変換

本節では、前節の結果を有理 B-spline 曲線の場合へ一般化する。

k を自然数, n を 0 以上の整数とし, $T = \{t_0, \dots, t_{n+k}\}$ をノット列とする B-spline 基底関数 $N_{i,k,T}(t)$, $i=0, \dots, n$ を考える。

T は次の条件を満たすと仮定する。

- 両端のノットの多重度は k 。
- 実数 $\tau_1, \tau_2, \dots, \tau_{r-1} (\tau_1 < \tau_2 < \dots < \tau_{r-1})$ が T の中に多重度 $k-1$ のノットとして含まれる。

さらに, $\tau_0 = t_0 (= \dots = t_{k-1})$, $\tau_r = t_n (= \dots = t_{n+k})$ とおく。

ノット列 T に関する有理 B-spline 曲線

$$C(t) = \frac{\sum_{i=0}^n N_{i,k,T}(t) w_i P_i}{\sum_{i=0}^n N_{i,k,T}(t) w_i} \tag{5-16}$$

に対して, $\tau_0, \dots, \tau_{r-1}$ を区分点とするようなパラメタの区分的一次分数変換を施すことを考える。すなわち, K_0, K_1, \dots, K_{r-1} を正の実数, $\sigma_0 < \sigma_1 < \dots < \sigma_r$ として, パラメタ変換

$$t = \phi(s) = \begin{cases} \phi_{K_0}^{\{\sigma_0, \sigma_1; \tau_0, \tau_1\}}(s), & \text{for } \sigma_0 \leq s \leq \sigma_1 \\ \phi_{K_1}^{\{\sigma_1, \sigma_2; \tau_1, \tau_2\}}(s), & \text{for } \sigma_1 \leq s \leq \sigma_2 \\ \vdots \\ \phi_{K_{r-1}}^{\{\sigma_{r-1}, \sigma_r; \tau_{r-1}, \tau_r\}}(s), & \text{for } \sigma_{r-1} \leq s \leq \sigma_r \end{cases} \tag{5-17}$$

を考える。

$i=0, \dots, r-1$ に対し, $\phi_{K_i}^{\{\sigma_i, \sigma_{i+1}; \tau_i, \tau_{i+1}\}}(s)$ の分母を $\lambda_i(s)$ とおく。すなわち,

$$\lambda_i(s) = (K_i - 1)(s - \sigma_i) + (\sigma_{i+1} - \sigma_i) \tag{5-18}$$

また,

$$n_i = \max_{i \leq p < \tau_{i+1}} p \quad (i=0, \dots, r-1) \tag{5-19}$$

とおく。さらに,

$$S = \{s_0, \dots, s_{n+k}\} \quad s_i = \phi^{-1}(t_i) \quad (i=0, \dots, n+k)$$

とおくとき, パラメタ変換の結果は次のようになる。

定理 4 有理 B-spline 曲線 (5-16) に対してパラメタ変換 (5-17) を施して得られる曲線 $\tilde{C}(s) = C(\phi(s))$ は次の式で表される曲線である。

$$\tilde{C}(s) = \frac{\sum_{i=0}^n N_{i,k,S}(s) \tilde{w}_i P_i}{\sum_{i=0}^n N_{i,k,S}(s) \tilde{w}_i} \tag{5-20}$$

ここで, 定数倍の不定性を除いて,

$$\tilde{w}_i = \begin{cases} \frac{\prod_{j=1}^{k-1} \lambda_0(S_{i+j})}{(\sigma_1 - \sigma_0)^{k-1}} & (i=0, \dots, n_0) \\ w_i K_0^{k-1} \frac{\prod_{j=1}^{k-1} \lambda_1(S_{i+j})}{(\sigma_2 - \sigma_1)^{k-1}} & (i=n_0+1, \dots, n_1) \\ \vdots \\ w_i K_0^{k-1} \dots K_{r-2}^{k-1} \frac{\prod_{j=1}^{k-1} \lambda_{r-1}(S_{i+j})}{(\sigma_r - \sigma_{r-1})^{k-1}} & (i=n_{r-2}+1, \dots, n_{r-1}(=n)) \end{cases} \quad (5-21)$$

すなわち、変換後の曲線はノット列 S に関する有理 B -spline 曲線であり、その制御点¹は変換前の曲線と同じでウェイトは式 (5-21) の変換を受ける。

証明は付録で与える。

6. 有理 Bézier, B-spline 曲面のパラメタ変換

本章では、有理 Bézier 曲面、有理 B-spline 曲面にパラメタの区分的一次分数変換を施した時の結果について簡単に述べる。

まず、 K, L を正の実数として、有理 Bézier 曲面 (2-5) に対してパラメタ変換

$$\begin{aligned} u &= \phi_K(\tilde{u}) = \frac{K\tilde{u}}{(K-1)\tilde{u}+1} \\ v &= \phi_L(\tilde{v}) = \frac{L\tilde{v}}{(L-1)\tilde{v}+1} \end{aligned} \quad (6-1)$$

を施すことを考えると、その結果は次のようになる。

定理5 有理 Bézier 曲面 (2-5) に対してパラメタの一次分数変換 (6-1) を施して得られる曲面 $\tilde{S}(\tilde{u}, \tilde{v}) = S(\phi_K(\tilde{u}), \phi_L(\tilde{v}))$ は、次の形の有理 Bézier 曲面である。

$$\tilde{S}(\tilde{u}, \tilde{v}) = \frac{\sum_{i=0}^m \sum_{j=0}^n B_{i,m}(\tilde{u}) B_{j,n}(\tilde{v}) \tilde{w}_{i,j} P_{i,j}}{\sum_{i=0}^m \sum_{j=0}^n B_{i,m}(\tilde{u}) B_{j,n}(\tilde{v}) \tilde{w}_{i,j}} \quad (6-2)$$

ここで、定数倍の不定性を除いて、

$$\tilde{w}_{i,j} = K^i L^j w_{i,j} \quad (i=0, \dots, m; j=0, \dots, n) \quad (6-3)$$

証明は (3-4) を用いれば定理1の証明と同様。

次に、複数パッチ有理 Bézier 曲面のパラメタの区分的一次分数変換について述べる。

複数パッチ有理 Bézier 曲面とは、2章で定義した(1パッチの)有理 Bézier 曲面をその境界線にそって複数個張り合わせたものであるが、有理 B-spline 曲面として表現すれば次のようになる。

$u_0 < u_1 < \dots < u_r, v_0 < v_1 < \dots < v_s$ として、パラメタ u 方向のノット列を

$$U = \underbrace{\{u_0, \dots, u_0\}}_{m+1 \text{ 個}}, \underbrace{\{u_1, \dots, u_1\}}_m, \dots, \underbrace{\{u_{r-1}, \dots, u_{r-1}\}}_m, \underbrace{\{u_r, \dots, u_r\}}_{m+1 \text{ 個}}$$

とし、パラメタ v 方向のノット列を

$$V = \underbrace{\{v_0, \dots, v_0\}}_{n+1 \text{ 個}}, \underbrace{\{v_1, \dots, v_1\}}_n, \dots, \underbrace{\{v_{s-1}, \dots, v_{s-1}\}}_n, \underbrace{\{v_s, \dots, v_s\}}_{n+1 \text{ 個}}$$

と定め、これに関する B-spline 基底関数を $N_{i,m+1,u}(u) (i=0, \dots, rm), N_{j,n+1,v}(v) (j=$

$0, \dots, sn)$ とする. さらに制御点 $P_{i,j}$ および対応するウェイト $w_{i,j}(i=0, \dots, rm; j=0, \dots, sn)$ を与えられたとき,

$$S(u, v) = \frac{\sum_{i=0}^{rm} \sum_{j=0}^{sn} N_{i,k,u}(u) N_{j,l,v}(v) w_{i,j} P_{i,j}}{\sum_{i=0}^{rm} \sum_{j=0}^{sn} N_{i,k,u}(u) N_{j,l,v}(v) w_{i,j}} \quad (6-4)$$

さて, $\bar{u}_0 < \bar{u}_1 < \dots < \bar{u}_r, \bar{v}_0 < \bar{v}_1 < \dots < \bar{v}_s$ とし, $K_0, \dots, K_{r-1}, L_0, \dots, L_{s-1}$ を正の実数として, パラメタの区分的一次分数変換

$$u = \phi(\bar{u}) = \begin{cases} \phi_{K_0}^{[u_0, u_1; \bar{u}_0, \bar{u}_1]}(\bar{u}), & \text{for } \bar{u}_0 \leq \bar{u} \leq \bar{u}_1 \\ \phi_{K_1}^{[u_1, u_2; \bar{u}_1, \bar{u}_2]}(\bar{u}), & \text{for } \bar{u}_1 \leq \bar{u} \leq \bar{u}_2 \\ \vdots \\ \phi_{K_{r-1}}^{[u_{r-1}, u_r; \bar{u}_{r-1}, \bar{u}_r]}(\bar{u}), & \text{for } \bar{u}_{r-1} \leq \bar{u} \leq \bar{u}_r \end{cases} \quad (6-5)$$

$$v = \phi(\bar{v}) = \begin{cases} \phi_{L_0}^{[v_0, v_1; \bar{v}_0, \bar{v}_1]}(\bar{v}), & \text{for } \bar{v}_0 \leq \bar{v} \leq \bar{v}_1 \\ \phi_{L_1}^{[v_1, v_2; \bar{v}_1, \bar{v}_2]}(\bar{v}), & \text{for } \bar{v}_1 \leq \bar{v} \leq \bar{v}_2 \\ \vdots \\ \phi_{L_{s-1}}^{[v_{s-1}, v_s; \bar{v}_{s-1}, \bar{v}_s]}(\bar{v}), & \text{for } \bar{v}_{s-1} \leq \bar{v} \leq \bar{v}_s \end{cases}$$

を曲面 $S(u, v)$ に施すことを考える.

$$\tilde{U} = \{ \underbrace{\bar{u}_0, \dots, \bar{u}_0}_{m+1 \text{ 個}}, \underbrace{\bar{u}_1, \dots, \bar{u}_1}_m, \dots, \underbrace{\bar{u}_{r-1}, \dots, \bar{u}_{r-1}}_m, \underbrace{\bar{u}_r, \dots, \bar{u}_r}_{m+1 \text{ 個}} \}$$

$$\bar{u}_i = \phi^{-1}(u_i) \quad (i=0, \dots, r)$$

$$\tilde{V} = \{ \underbrace{\bar{v}_0, \dots, \bar{v}_0}_{n+1 \text{ 個}}, \underbrace{\bar{v}_1, \dots, \bar{v}_1}_n, \dots, \underbrace{\bar{v}_{s-1}, \dots, \bar{v}_{s-1}}_n, \underbrace{\bar{v}_s, \dots, \bar{v}_s}_{n+1 \text{ 個}} \}$$

$$\bar{v}_i = \phi^{-1}(v_i) \quad (i=0, \dots, s)$$

とおくと, 結果は次のようになる.

定理 6 複数パッチ有理 Bézier 曲面 (6-4) に対してパラメタの区分的一次分数変換 (6-5) を施して得られる曲面 $\tilde{S}(\bar{u}, \bar{v}) = S(\phi(\bar{u}), \phi(\bar{v}))$ は次の形の複数パッチ有理 Bézier 曲面である.

$$\tilde{S}(\bar{u}, \bar{v}) = \frac{\sum_{i=0}^{rm} \sum_{j=0}^{sn} N_{i,m+1}(\bar{u}) N_{j,n+1}(\bar{v}) \tilde{w}_{i,j} P_{i,j}}{\sum_{i=0}^{rm} \sum_{j=0}^{sn} N_{i,m+1}(\bar{u}) N_{j,n+1}(\bar{v}) \tilde{w}_{i,j}} \quad (6-6)$$

ここで, $i=0, \dots, rm; j=0, \dots, sn$ に対し, $p = \left[\frac{i}{m} \right], q = \left[\frac{j}{n} \right]$ ($[x]$ は x を越えない最大の整数) とおくと,

$$\tilde{w}_{i,j} = K_0^m \cdots K_{p-1}^m K_p^{i-pm} L_0^n \cdots L_{q-1}^n L_q^{j-qn} w_{i,j} \quad (6-7)$$

ただし, $\tilde{w}_{i,j}$ を一斉に定数倍する不定性を除く.

証明は定理 3 の証明と同様であり, 省略する.

また, より一般に定理 4 の曲面版も証明できるが, ここでは省略する.

7. 応 用

四辺の枠線が NURBS 曲線と与えられたとき, その内部を内挿して曲面を創成する問題を考える. このとき, 既知の方法を適用して曲面を張るためには, 隣接する枠線

の共有する端点におけるウェイトが等しくなっていないなければならない。もしウェイトが異なっていた場合には、与えられた枠線の形状を変えないでウェイトを変更することが必要となる。

枠線が複数セグメントの有理 Bézier 曲線で与えられた場合、四辺の枠線のセグメント分割点のパラメタを変更することなく、端点におけるウェイトの値を合わせられることが望ましい。

セグメント分割点のパラメタを変えないという条件を満たす区分的一次分数変換の一般形を求めてみる。区分的一次分数変換の C^1 連続条件(5-6)において、 $\sigma_i = \tau_i$ ($i = 0, \dots, r-1$) とおくと、

$$\begin{aligned} \frac{1}{K_0} &= K_1 \\ \frac{1}{K_1} &= K_2 \\ &\vdots \\ \frac{1}{K_{r-2}} &= K_{r-1} \end{aligned} \quad (7-1)$$

すなわち、 $K = K_0$ とおくと、

$$K_i = \begin{cases} K, & i \text{ が偶数のとき} \\ \frac{1}{K}, & i \text{ が奇数のとき} \end{cases} \quad (7-2)$$

となる。

この区分的一次分数変換による複数セグメント有理 Bézier 曲線のウェイトの変換式を求める。(7-2)を式(5-15)に代入することにより、 r 個のセグメントからなる n 次有理 Bézier 曲線の終端点のウェイト w_m は

$$\bar{w}_m = \begin{cases} w_m, & r \text{ が偶数のとき} \\ K^n w_m, & r \text{ が奇数のとき} \end{cases} \quad (7-3)$$

と変換されることがわかる。

ここで、 K は任意の正の実数であるから、次のことがわかる。(ウェイトを一斉に定数倍する自由度は常にあることに注意)

定理7 奇数個のセグメントからなる C^1 連続な複数セグメント Bézier 曲線に対して、曲線の形状および C^1 連続性を保ちつつ、かつセグメント分割点のパラメタを変えずに、両端点のウェイトの値を任意の値に変更することができる。

この結果と、ウェイトを一斉に定数倍する操作を組み合わせることにより、面張りの問題において四本の枠線のうち少なくとも一本が奇数個のセグメントからなる場合には、各枠線の形状、 C^1 連続性、セグメント分割点のパラメタ値を保存しつつ、端点におけるウェイトの値を合わせることができる。

8. 残された課題

本稿では、複数セグメント (複数パッチ) の有理 Bézier 曲線 (曲面) や、より一般の B-spline 曲線・曲面に対して、その形状を保ちながらウェイトを変更する自由度を、

パラメタの区分的一次分数変換という視点から考察した。その結果、Bézier の場合にはセグメント分割点を区分点とするような区分的一次分数変換、B-spline の場合には多重度が次数と一致するようなノットを区分点とするような区分的一次分数変換を施すことにより、制御点を動かさないうエイトのみを変更することができることを示した。

B-spline の場合に“区分点のノットの多重度が次数と一致する”という条件は、かなり強い制限であり、本質的に“Bézier 的”なケースのみを扱ったものである。そこで、残された課題としては、

- ・区分点のノットの多重度に関する制限をゆるめて、区分的一次分数変換の適用範囲を広げることはいか
- ということが考えられる。とくに、 C^1 連続な B-spline 曲線・曲面を扱う立場からすると、多重度が“次数-1”に等しいようなノットを区分点とできることが望ましい。
- また、本稿ではパラメタの区分的一次分数変換によるウェイト操作について考察したが、
- ・曲線・曲面形状を不変にするようなウェイト変更の方法として、パラメタの区分的一次分数変換によるもの以外の有効な方法は有り得るか
- ということも興味ある問題である。

9. おわりに

本稿の内容の考察、執筆にあたり、CAD/CAM システム部第二課、CADCEUS 図形処理チーム清水リーダーには大変お世話になった。文献^{[1][2]}に注目し、その内容を区分的一次分数変換へ拡張するというアイデアは同氏によるものである。同氏および CAD/CAM システム部大高部長、システム三部システム開発課松木課員には原稿を何度も読んで頂き、多くの有益な助言を頂いた。深く感謝する次第である。

A 付 録

A.1 定理 2 の証明

まず、B-spline 基底関数 $N_{i,k,r}(t)$ にパラメタ変換(4-1)を施したとき、どのような形になるかを考える。簡単のため、ノット (t_i) がすべて異なる場合をまず考える。この場合には、差分商は簡単な形に表される。

補題 1 (差分商の Lagrange 形式) $t_0 < t_1 < \dots < t_k$ のとき、任意の関数 $f(x)$ に対し、

$$[t_0, \dots, t_k]f = \sum_{i=0}^k \frac{f(t_i)}{\prod_{0 \leq j \leq k, j \neq i} (t_i - t_j)} \quad (\text{A-1})$$

が成り立つ。

証明

$$\Delta_i(t) = \prod_{0 \leq j \leq k, j \neq i} (t - t_j) \quad (\text{A-2})$$

$$L_i(t) = \frac{\Delta_i(t)}{\Delta_i(t_i)} \quad (\text{A-3})$$

とおくと、 $L_i(t)$ は t の k 次多項式であり、

$$L_i(t_j) = \delta_{ij} = \begin{cases} 1, & \text{for } j=i \\ 0, & \text{for } j \neq i \end{cases}$$

をみます。したがって、

$$F(t) = \sum_{i=0}^k f(t_i) L_i(t)$$

とおくと、 $F(t)$ は t の k 次多項式であり、

$$F(t_i) = f(t_i) \quad \text{for } i=0, \dots, k$$

をみます。したがって、差分商の定義により、

$$\begin{aligned} [t_0, \dots, t_k] f &= F(t) \text{ の最高次係数} \\ &= \sum_{i=0}^k \frac{f(t_i)}{\prod_{0 \leq j \leq k, j \neq i} (t_i - t_j)} \end{aligned}$$

(補題 1 の証明終わり)

補題 1 より、

$$N_{0,k,s}(t) = (t_k - t_0) \sum_{i=0}^k \frac{f(t_i)}{\prod_{0 \leq j \leq k, j \neq i} (t_i - t_j)} \quad (\text{A-4})$$

と書ける。簡単のため、 $\Delta > 0$ (つまり、 ϕ と ψ は増加関数) と仮定する。

$$t_i - t = \frac{\Delta(s_i - s)}{\lambda(s_i)\lambda(s)} \quad (\text{A-5})$$

であり、 λ が J で符号を変えないことより $\lambda(s_i)\lambda(s) > 0$ に注意すると、

$$(t_i - t)_+^{k-1} = \frac{\Delta^{k-1}(s_i - s)_+^{k-1}}{(\lambda(s_i)\lambda(s))^{k-1}} \quad (\text{A-6})$$

となる。(A-4)の分母は、

$$\prod_{j \neq i} \frac{\Delta(s_i - s_j)}{\lambda(s_i)\lambda(s_j)} = \frac{\Delta^k \prod_{j \neq i} (s_i - s_j)}{\lambda(s_i)^{k-1} \prod_{j=0}^k \lambda(s_j)}$$

となるから、結局、

$$N_{0,k,\tau}(t) = \frac{\prod_{j=1}^{k-1} \lambda(s_j)}{\lambda(s)^{k-1}} (s_k - s_0) \sum_{i=0}^k \frac{(s_i - s)_+^{k-1}}{\prod_{j \neq i} (s_i - s_j)}$$

が得られる。したがって、一般に、

$$N_{i,k,\tau}(t) = \frac{\prod_{j=1}^{k-1} \lambda(s_{i+j})}{\lambda(s)^{k-1}} N_{i,k,s}(s) \quad (\text{A-7})$$

が成り立つ。 s と t の役割を入れ換えると、

$$N_{i,k,\tau}(t) = \frac{\mu(t)^{k-1}}{\prod_{j=1}^{k-1} \mu(t_{i+j})} N_{i,k,s}(s) \quad (\text{A-8})$$

が得られる。式(A-7)、(A-8)はノット t_i がすべて異なるという仮定のもとで導いたが、ノットどうしを近づけた時の極限をとることにより、ノットが重複している場合にもこれらの式は成立する。

式(A-7)または式(A-8)を式(2-10)に代入することにより、定理が得られる。

A.2 定理3の証明

簡単のため、 $r=2$ として証明する。一般の場合も同様である。

まず、B-spline 基底関数 $N_i(t) = N_{i,n+1,r}(t)$ と Bernstein 基底関数 $B_{i,n}^{[t_0, t_1]}(t)$ との関係を書くと、

$$\begin{aligned}
 N_0(t) &= \begin{cases} B_{0,n}^{[t_0, t_1]}(t), & \text{for } t_0 \leq t \leq t_1 \\ 0, & \text{for } t_1 \leq t \leq t_2 \end{cases} \\
 &\vdots \\
 N_{n-1}(t) &= \begin{cases} B_{n-1,n}^{[t_0, t_1]}(t), & \text{for } t_0 \leq t \leq t_1 \\ 0, & \text{for } t_1 \leq t \leq t_2 \end{cases} \\
 N_n(t) &= \begin{cases} B_{n,n}^{[t_0, t_1]}(t), & \text{for } t_0 \leq t \leq t_1 \\ B_{0,n}^{[t_1, t_2]}(t), & \text{for } t_1 \leq t \leq t_2 \end{cases} \\
 N_{n+1}(t) &= \begin{cases} 0, & \text{for } t_0 \leq t \leq t_1 \\ B_{1,n}^{[t_1, t_2]}(t), & \text{for } t_1 \leq t \leq t_2 \end{cases} \\
 &\vdots \\
 N_{2n}(t) &= \begin{cases} 0, & \text{for } t_0 \leq t \leq t_1 \\ B_{n,n}^{[t_1, t_2]}(t), & \text{for } t_1 \leq t \leq t_2 \end{cases}
 \end{aligned}$$

となる。 $N_n(t)$ については、

$$\begin{aligned}
 N_n^{(0)}(t) &= \begin{cases} B_{n,n}^{[t_0, t_1]}(t), & \text{for } t_0 \leq t \leq t_1 \\ 0, & \text{for } t_1 \leq t \leq t_2 \end{cases} \\
 N_n^{(1)}(t) &= \begin{cases} 0, & \text{for } t_0 \leq t \leq t_1 \\ B_{0,n}^{[t_1, t_2]}(t), & \text{for } t_1 \leq t \leq t_2 \end{cases}
 \end{aligned}$$

と定義すると、

$$N_n(t) = N_n^{(0)}(t) + N_n^{(1)}(t)$$

と書ける。また、曲線 $C^{(0)}(t)$, $C^{(1)}(t)$ を

$$\begin{aligned}
 C^{(0)}(t) &= \frac{\sum_{i=0}^{n-1} N_i(t) w_i P_i + N_n^{(0)}(t) w_n P_n}{\sum_{i=0}^{n-1} N_i(t) w_i + N_n^{(0)}(t) w_n} \\
 C^{(1)}(t) &= \frac{N_n^{(1)}(t) w_n P_n + \sum_{i=n+1}^{2n} N_i(t) w_i P_i}{N_n^{(1)}(t) w_n + \sum_{i=n+1}^{2n} N_i(t) w_i}
 \end{aligned}$$

と定義すると、曲線 $C(t)$ は

$$C(t) = C^{(0)}(t) + C^{(1)}(t)$$

と表される。問題は曲線 $C(t)$ にパラメタ変換

$$t = \phi(s) = \begin{cases} \phi_{K_0}^{[s_0, s_1; t_0, t_1]}(s), & \text{for } s_0 \leq s \leq s_1 \\ \phi_{K_1}^{[s_1, s_2; t_1, t_2]}(s), & \text{for } s_1 \leq s \leq s_2 \end{cases}$$

を施した結果を求めることであるが、 $C^{(0)}(t)$ は $[t_0, t_1]$ 上で 0, $C^{(1)}(t)$ は $[t_1, t_2]$ 上で 0 となることより、

$$C(\phi(s)) = C^{(0)}(\phi_{K_0}^{[s_0, s_1; t_0, t_1]}(s)) + C^{(1)}(\phi_{K_1}^{[s_1, s_2; t_1, t_2]}(s))$$

が成り立つ。

さて、Bernstein 基底関数 $B_{i,n}^{[t_0, t_1]}(t)$ にパラメタ変換 $\phi_{K_0}^{[s_0, s_1; t_0, t_1]}(s)$ を施した結果を

求めると,

$$\begin{aligned}
 B_{i,n}^{[t_0, t_1]}(\phi_{K_0}^{[s_0, s_1; t_0, t_1]}(s)) &= \frac{1}{(t_1 - t_0)^n} \binom{n}{i} \{ \phi_{K_0}^{[s_0, s_1; t_0, t_1]}(s) - t_0 \}^i \{ t_1 - \phi_{K_0}^{[s_0, s_1; t_0, t_1]}(s) \}^{n-i} \\
 &= \frac{1}{(t_1 - t_0)^n} \binom{n}{i} \left\{ \frac{K_0(t_1 - t_0)(s - s_0)}{(K_0 - 1)(s - s_0) + (s_1 - s_0)} \right\}^i \\
 &\quad \times \left\{ \frac{(t_1 - t_0)(s_1 - s)}{(K_0 - 1)(s - s_0) + (s_1 - s_0)} \right\}^{n-i} \\
 &= \frac{K_0^i}{\{(K_0 - 1)(s - s_0) + (s_1 - s_0)\}^n} \binom{n}{i} (s - s_0)^i (s_1 - s)^{n-i} \\
 &= \frac{(s_1 - s_0)^n K_0^i}{\{(K_0 - 1)(s - s_0) + (s_1 - s_0)\}^n} B_{i,n}^{[s_0, s_1]}(s)
 \end{aligned}$$

となる。したがって,

$$C^{(0)}(\phi_{K_0}^{[s_0, s_1; t_0, t_1]}(s)) = \frac{\sum_{i=0}^{n-1} N_i(s) K_0^i w_i P_i + N_n^{(0)}(s) K_0^n w_n P_n}{\sum_{i=0}^{n-1} N_i(s) K_0^i w_i + N_n^{(0)}(s) K_0^n w_n} \quad (\text{A-9})$$

同様に,

$$C^{(1)}(\phi_{K_1}^{[s_1, s_2; t_1, t_2]}(s)) = \frac{N_n^{(1)}(s) w_n P_n + \sum_{i=n+1}^{2n} N_i(s) K_1^{i-n} w_i P_i}{N_n^{(1)}(s) w_n + \sum_{i=n+1}^{2n} N_i(s) K_1^{i-n} w_i} \quad (\text{A-10})$$

が成り立つ。

式(A-10)の分母・分子に K_0^n をかけて式(A-9)に加えると, 次を得る。

$$C(\phi(s)) = \frac{\sum_{i=0}^{n-1} N_i(s) K_0^i w_i P_i + N_n(s) K_0^n w_n P_n + \sum_{i=n+1}^{2n} N_i(s) K_0^n K_1^{i-n} w_i P_i}{\sum_{i=0}^{n-1} N_i(s) K_0^i w_i + N_n(s) K_0^n w_n + \sum_{i=n+1}^{2n} N_i(s) K_0^n K_1^{i-n} w_i} \quad (\text{A-11})$$

これで証明された。

A.3 定理4の証明

簡単のため, $r=2$ の場合に証明を行う。一般の場合も同様である。

$\tau_1 = t_{n_0+1}$ が多重度 $k-1$ のノットであることから,

$$N_{n_0, k, r}(t) = \begin{cases} B_{k-1, k-1}^{[t_{n_0}, t_{n_0+1}]}(t), & \text{for } t_{n_0} \leq t \leq \tau_1 \\ B_{0, k-1}^{[t_{n_0+1}, t_{n_0+2}]}(t), & \text{or } \tau_1 \leq t \leq t_{n_0+k} \end{cases}$$

が成り立つ。

$$\begin{aligned}
 N_{n_0}^{(0)}(t) &= \begin{cases} B_{k-1, k-1}^{[t_{n_0}, t_{n_0+1}]}(t), & \text{for } t_{n_0} \leq t \leq \tau_1 \\ 0, & \text{for } \tau_1 \leq t \leq t_{n_0+k} \end{cases} \\
 N_{n_0}^{(1)}(t) &= \begin{cases} 0, & \text{for } t_{n_0} \leq t \leq \tau_1 \\ B_{0, k-1}^{[t_{n_0+1}, t_{n_0+2}]}(t), & \text{for } \tau_1 \leq t \leq t_{n_0+k} \end{cases}
 \end{aligned}$$

とおくと,

$$N_{n_0, k, r}(t) = N_{n_0}^{(0)}(t) + N_{n_0}^{(1)}(t)$$

と書ける。そこで, 曲線 $C^{(0)}(t)$, $C^{(1)}(t)$ を

$$C^{(0)}(t) = \frac{\sum_{i=0}^{n_0-1} N_{i,k,T}(t) w_i P_i + N_{n_0}^{(0)}(t) w_{n_0} P_{n_0}}{\sum_{i=0}^{n_0-1} N_{i,k,T}(t) w_i + N_{n_0}^{(0)}(t) w_{n_0}}$$

$$C^{(1)}(t) = \frac{N_{n_0}^{(1)}(t) w_{n_0} P_{n_0} + \sum_{i=n_0+1}^n N_{i,k,T}(t) w_i P_i}{N_{n_0}^{(1)}(t) w_{n_0} + \sum_{i=n_0+1}^n N_{i,k,T}(t) w_i}$$

と定義すると、曲線 $C(t)$ は

$$C(t) = C^{(0)}(t) + C^{(1)}(t)$$

と表される。

$$C(\phi(s)) = C^{(0)}(\phi_{K_0}^{[\sigma_0, \sigma_1; \tau_0, \tau_1]}(s)) + C^{(1)}(\phi_{K_1}^{[\sigma_1, \sigma_2; \tau_1, \tau_2]}(s))$$

となるが、式(A-7)により $(\lambda_0(\sigma_1) = K_0(\sigma_1 - \sigma_0))$ に注意すると、

$$C^{(0)}(\phi_{K_0}^{[\sigma_0, \sigma_1; \tau_0, \tau_1]}(s)) = \frac{\sum_{i=0}^{n_0-1} \frac{\prod_{j=1}^{k-1} \lambda_0(S_{i+j})}{\lambda_0(s)^{k-1}} N_{i,k,S}(s) w_i P_i + \frac{\{K_0(\sigma_1 - \sigma_0)\}^{k-1}}{\lambda_0(s)^{k-1}} N_{n_0}^{(0)}(s) w_{n_0} P_{n_0}}{\sum_{i=0}^{n_0-1} \frac{\prod_{j=1}^{k-1} \lambda_0(S_{i+j})}{\lambda_0(s)^{k-1}} N_{i,k,S}(s) w_i + \frac{\{K_0(\sigma_1 - \sigma_0)\}^{k-1}}{\lambda_0(s)^{k-1}} N_{n_0}^{(0)}(s) w_{n_0}}$$

$$= \frac{\sum_{i=0}^{n_0-1} N_{i,k,S}(s) \frac{\prod_{j=1}^{k-1} \lambda_0(S_{i+j})}{(\sigma_1 - \sigma_0)^{k-1}} w_i P_i + N_{n_0}^{(0)}(s) K_0^{k-1} w_{n_0} P_{n_0}}{\sum_{i=0}^{n_0-1} N_{i,k,S}(s) \frac{\prod_{j=1}^{k-1} \lambda_0(S_{i+j})}{(\sigma_1 - \sigma_0)^{k-1}} w_i + N_{n_0}^{(0)}(s) K_0^{k-1} w_{n_0}}$$

(A-12)

同様に、 $(\lambda_1(\sigma_2) = K_1(\sigma_2 - \sigma_1))$ に注意すると、

$$C^{(1)}(\phi_{K_1}^{[\sigma_1, \sigma_2; \tau_1, \tau_2]}(s)) = \frac{\frac{(\sigma_2 - \sigma_1)^{k-1}}{\lambda_1(s)^{k-1}} N_{n_0}^{(1)}(s) w_{n_0} P_{n_0} + \sum_{i=n_0+1}^n \frac{\prod_{j=1}^{k-1} \lambda_1(S_{i+j})}{\lambda_1(s)^{k-1}} N_{i,k,S}(s) w_i P_i}{\frac{(\sigma_2 - \sigma_1)^{k-1}}{\lambda_1(s)^{k-1}} N_{n_0}^{(1)}(s) w_{n_0} + \sum_{i=n_0+1}^n \frac{\prod_{j=1}^{k-1} \lambda_1(S_{i+j})}{\lambda_1(s)^{k-1}} N_{i,k,S}(s) w_i}$$

$$= \frac{N_{n_0}^{(1)}(s) w_{n_0} P_{n_0} + \sum_{i=n_0+1}^n N_{i,k,S}(s) \frac{\prod_{j=1}^{k-1} \lambda_1(S_{i+j})}{(\sigma_2 - \sigma_1)^{k-1}} w_i P_i}{N_{n_0}^{(1)}(s) w_{n_0} + \sum_{i=n_0+1}^n N_{i,k,S}(s) \frac{\prod_{j=1}^{k-1} \lambda_1(S_{i+j})}{(\sigma_2 - \sigma_1)^{k-1}} w_i}$$

(A-13)

式(A-13)の分母・分子に K_0^k をかけて式(A-12)に加えると、次を得る。

$$C(\phi(s)) = \frac{\sum_{i=0}^{n_0} N_{i,k,S}(s) \frac{\prod_{j=1}^{k-1} \lambda_0(S_{i+j})}{(\sigma_1 - \sigma_0)^{k-1}} w_i P_i + \sum_{i=n_0+1}^n N_{i,k,S}(s) K_0^{k-1} \frac{\prod_{j=1}^{k-1} \lambda_1(S_{i+j})}{(\sigma_2 - \sigma_1)^{k-1}} w_i P_i}{\sum_{i=0}^{n_0} N_{i,k,S}(s) \frac{\prod_{j=1}^{k-1} \lambda_0(S_{i+j})}{(\sigma_1 - \sigma_0)^{k-1}} w_i + \sum_{i=n_0+1}^n N_{i,k,S}(s) K_0^{k-1} \frac{\prod_{j=1}^{k-1} \lambda_1(S_{i+j})}{(\sigma_2 - \sigma_1)^{k-1}} w_i}$$

これで証明された。

参考文献 [1] R. R. Patterson, "Projective Transformation of the Parameter of a Bernstein-Bézier Curve", ACM Transactions on Graphics, 1985, Vol. 4, No. 4, pp. 276~290.
 [2] E. T. Y. Lee and M. L. Lucian, "Möbius reparametrizations of rational B-

splines”, Computer Aided Geometric Design 8, 1991, pp.213~215.

[3] de Boor, Carl, *A Practical Guide to Splines*, Springer-Verlag, 1978.

執筆者紹介 中 邨 博 之 (Hiroyuki Nakamura)

1962年生。1987年3月東京大学大学院 理学系研究科数学専攻 修士課程修了。1991年8月東京大学大学院 理学系研究科数学専攻博士課程退学。同年9月日本ユニシス(株)入社。現在、製造工業システム本部 CAD/CAM システム部第二課所属、CADCEUS 図形処理ライブラリの開発に従事。



形状モデリングにおける多重構造表現

A Multiplied Structure for Geometric Modeling

松 木 則 夫

要 約 設計・製造過程を支援する CAD システムは現在、広く普及しているが、依然として多くの問題が残っている。とくに、並行作業、設計変更データの操作等に問題が多い。本稿では、これらの問題が現状の現状モデル表現の単一性に由来していると考え、多重表現の必要性を論じる。そして、この対策として多重化したデータ構造 QNS を提案する。これは、形状を部分形状である擬立体のネットワークで表現するもので、製品の version 管理、design variation の表現に効果的であり、さらに形状特徴モデリングへの適用が可能であることを示す。

Abstract CAD systems are widely spread and become effective tools for the design and production process, however, there are still some drawbacks. Among other things, simultaneous engineering and design modification are difficult to be handled. In this paper, we show that these difficulties arise from the assumption of current geometric modelers, that is, single and flat representation scheme of geometric modelers, and discuss the need of the multiplied adjacency representation. We propose here the multiplied structure, called QNS (quasi-solid network structure), to facilitate the version manipulation and design variation, and show the QNS can be applicable to the feature based modeling.

1. はじめに

現代の設計・製造過程は巨大化し複雑化している。このために、データやシステムの統合化等さまざまな努力がなされているが、依然として多くの問題が残っている。本稿では、並行作業、類似性データ表現等の問題の本質的な部分が現状の形状モデルの表現に関係していることを指摘し、対策として多重化したデータ構造を提案する。

2. 設計製造過程表現としての形状モデラの問題

2.1 並行作業

2.1.1 並行作業に必要な機能

一つの製品を作るために、会社の複数の部門だけでなく、複数の会社にまたがって製品のデータが行き来している。各部門、各社は直列ではなく、並列に作業をしている。いわゆるコンカレント・エンジニアリングである。しかし、並行作業が行われているのは事実であるが、省力化、自動化等 CAD システムが果たすべき役割が並行作業に対して十分機能しているとは言いがたい。

並行作業にはいくつかの並行性が内在している。まず製品の並行性がある。設計の履歴管理だけでなく、複数候補の設計が並行して行われ、いくつかの評価項目を比較検討して最終製品が決められることがある。つまり、ある時点では複数の製品の版 (version) が意味ある形状として処理される。つぎに工程内の並行性がある。複雑な部品では同一部品に対しても、複数作業者が同時に形状の生成、変形作業を行うため

ある。そして工程間の並行性がある。製品の設計がある程度進行すると、製造に必要なさまざまな設備の設計を開始することが製品のリードタイム短縮のために不可欠であるが、製品の設計が停止しているわけではない。つまり下工程と上工程で製品形状が異なる状態になる。このように並行作業には性質の違った並行性があり、それらが複雑に関連している。

では、一体どのような機能がCADシステムに求められているのか。まず製品の並行性では、本来は一つの製品形状が複数の version で処理されていることが表現でき、version 間の形状の差異を効率的に判断し操作できること、ではないだろうか。これを「version 管理機能」と呼ぶことにする。つぎに工程内の並行性では同一部品に対し、同時に形状を生成、変更できることであろう。これを「同時形状操作機能」と呼ぶ。そして、工程間での並行性では「形状データの統一管理機能」と「version 管理機能」が共に求められていると思われる。この中で「形状データの統一管理機能」は統合化されたデータベース (OB) により可能となる部分もあるが、「version 管理機能」と「同時形状操作機能」については問題が残る。

さらに機密保護も並行作業では重要である。すべてのデータを外部に見せられないことも多い。このため、データ発行のために一体化した形状表現の製品モデルをわざわざ分割する作業も発生する。しかし発行後も形状は変更されていくため、同じ製品のモデルがいくつも発生することになる。つまりアクセス管理のできる「version 管理機能」が必要となる。

2.1.2 現状の対応策

以上の並行作業に求められる機能について、現状の形状モデラの対応策を検討する。まず「同時形状操作機能」であるが、立体 (Solid) モデラ^[1]を例にすると、一つの body が更新の最小単位である。たとえどんなに複雑な形状でも、ひとたび一体化され一つの body となれば、別々の作業者が同時に修正することは基本的にできない。ところが共同設計の環境では、一つの部品の異なる場所を同時に生成・変更できることが望ましく、効率もよい。したがって、作業の最小単位に部品を分割して別々に作業を進め、後で合成することになる。しかし、これは並列処理とは言えない。分割された部分は

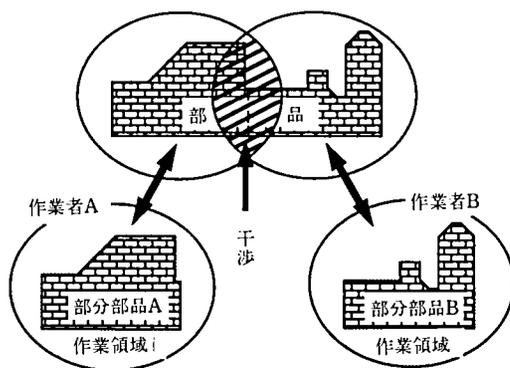


図 1 作業領域の干渉

Fig. 1 Collision of work areas

相互に深く関係しており、作業は図1のように重なり合うため、他の部分の形状と何等かの関連付けが求められる。つまり、単品の問題をより複雑なアセンブリの問題に変えたことに等しい。部品の境界で問題が発生しやすく、不要な境界面が発生する等の不都合がある。

そこで、一つの部品データベース(DB)に複数の作業者がアクセスすることが考えられる。「形状データの統一管理機能」の下での「同時形状操作機能」である。NFS(Network File System)等の普及から、ネットワークを介しての並行作業のハードウェア環境は整っている。しかし事務処理のDBに比較して、形状データに関するDB管理には困難な問題がある。たとえば一つのbodyは、そのbodyデータ全体で意味を保っている。つまり、更新処理で意味の整合性を維持するために参照すべき範囲がほとんどbodyデータ全体に及ぶ。いくら形が巨大でも、一つのbody全体が操作の対象とならざるを得ない。このため、DBアクセス時に適当な作業単位でRead/Writeを指定することが行われる。Write Openできるのを一人だけに限定し、この作業者が終了するまで他の作業者は形状を見ることしかできないようにする。このRead/Write管理は、仕組みは簡単だが並行作業とは言えない。時分割処理のように、Read/Writeのタイミングを形状モデルの整合性を保つ最小作業単位(たとえばコマンド)まで細分化すると、あたかも並行作業が実現するようになるが見えるが、このとき作業者には非常に奇妙な事態が起こる。コマンドの切れ目ごとに形状がどう変わるか、各作業者に予測のつかない状態が発生するのである。他の作業者との区分はあるにしても、作業者の作る幾何データは一時的な図形も多く、これらが他の作業者の作業領域に侵入することは大いにあり得るためである。このように、他の作業者の作る製品形状の評価が実時間でできる状況は望ましいが、今作業している形状に他人の操作が常に直接影響することが好ましいとは考えられない。さらに、各作業者ごとの操作取消し(undo)等の困難な問題も発生する。

このように、素朴に形状を修正する機能では「同時形状操作機能」を実現することは困難である。したがって、設計履歴を作業者ごとに独立に管理し、設計終了時点で置き換える手法^[2]等が必要となる。

次に、「version管理機能」に対する現状のCAD/CAMソフトウェアの対応を検討する。まず、よく行われるのが、幾何要素の属性にさまざまな情報を書き込めるようにすることである。このとき情報を書き込むことはできるが、これらを図形演算に反映させ、形状モデラとして一貫したものにすることは困難である。たとえ、できたとしてもアプリケーションと形状モデルとがあまりにも密接に関連し、システムの見通しが悪くなる。演算の構造と属性の内容を分離させると、極端に言えば属性の記述内容が作業者に対するコメントとしての意味しかなくなる。これは、形状モデルを三次元の図面として扱うことに等しくversion管理の省力化には程遠い。

CSGとB-repのHybrid表現には、幾何形状xが立体Aの一部でもあり、同時に別の立体Bの一部でもあるという状態を表現できるものがある^[3]。これは現在の形状に至るまでの操作、および入力された元の形状を記憶しておく方法である。つまり、断片化され細分化された入力形状群を保持しておき、断片の組み合わせを変えることで、さまざまな形を作ることができる。しかし、これは無駄な部分で困難な問題を引き起

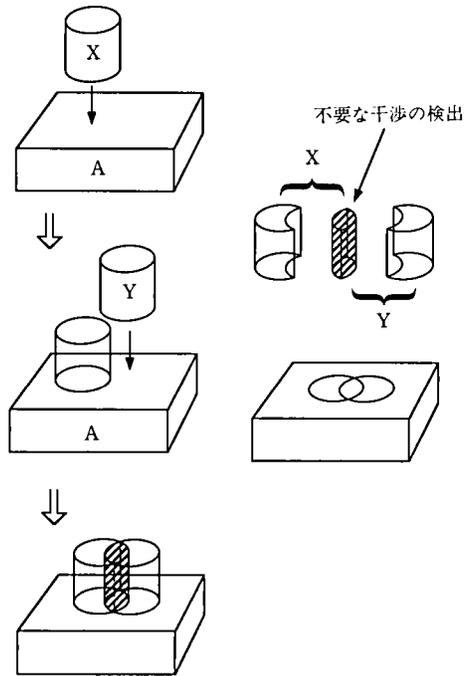


図 2 設計変更形状に対する Hybrid 表現の問題

Fig. 2 Drawback of the hybrid representation for design modification

こすことになる。たとえば、図2のように円筒 x を立体 A に加える作業があったとする。これが設計変更で、円筒 x の代わりに円筒 y を加えることになったとする。両方の形状の保持が必要なとき、実際には不要な斜線部の立体の表現が必要となる。この干渉部分が非多様体形状^[4]となることもある。一般に、設計変更前と、後の形状は似通っている。もし、同筒が自由曲面で形状の変化が微妙であるとき、集合演算で数値的に問題が発生しやすい。そして問題なのは、このようなことが表現する必要のない部分で起こることである。また、この表現では並行作業の問題も依然として残ったままである。

2.2 製品群の類似性の表現

設計対象は一般に、いままで設計したものの延長上にあり、まったく新規に設計することは多くない。製品の種類は多くても、個々の部門あるいは会社では、設計、製造する製品群は似通っている。ここでは、これらの製品群を一つの基本製品の design variation と呼ぶことにする。これら似通った製品部分をいかに効率良く設計、製造するかは重大な関心事であり、さまざまな試みがなされている。たとえば、頻繁に使われる部分を標準部品として登録し、設計や製造の省力化を図っている。ここで、「似通っている」という類似性についてももう少し詳しく見る。まず、形状の寸法は多少異なっているが、位相構造が等しい、いわゆる「作り」が同じものがある。可変ソリッドの考えに近く、パラメトリック設計が有効な生成手法である。さらに進んで、一度生成された後も一括して修正が可能となるモデルがある。設計の省力化のために、同一形状の部分は設計の変更時には同期して修正したいという要請は自然である。これは、

形状特徴モデリング機能のあるシステム等で実現されている。

ところが、これらのシステムが対象とするのは同一の部品に対する類似性の管理である。ほとんどのシステムは、一つの部品に対する操作の利便性を追及するが、複数の製品に共通な形状部分の管理を行わない。これらのシステムが提供するの、簡易作成の手段であって、同一性をそのままで表現する手段に乏しい。つまり、同じ形状のコピーを数多く簡単に作る機能は充実しても、それらが同じ形である、という情報が管理されていない。とくに、B-repの立体表現における和集合演算等では、演算前の個々の形状の情報が失われてしまう。しかし、実際は同一形状の部位は複数の製品に共通であることが多く、これらの同一性をCADシステムが効率的に活用できることが望ましいのは明らかである。たとえば、自動車、家電製品等多様化した製品のvariation表現に有効と考える。また、製造過程でNC (Numerical Control) データを複数製品にわたり利用できれば大変効率が良い。このように、複数製品にわたるデータの共用化を「design variation 管理機能」と呼ぶことにする。

2.3 問題の共通性と多重表現

複数製品にわたる類似性の問題は、並行設計における形状モデルの問題と同じ性質であると筆者は考える。すなわち、ある幾何形状は唯一の立体の一部であるという通常の形状モデルの前提に深く関係している。一般に、ある幾何形状 x が立体 A の一部でもあり、同時に別の立体 B の一部でもあるということは許されない。しかし、これが可能ならば「version 管理機能」と「design variation 管理機能」が実現できることがわかる。つまり、ある部位 x が製品 A と製品 B に形状表現として共有することができる。「design variation 管理機能」であり、製品 A を version 1 に、製品 B を version 2 としたものが「version 管理機能」に他ならないからである。そして、設計変更が履歴情報という形ではなく実際の形状として表現できることになる。

しかも、付随的な情報をもとにアプリケーションの仕組みとして実現するのではなく、形状モデラの機能として実現できることが大切である。アプリケーションの仕組みとして実現する時は、別々の形状データを同一のものとして扱うために、図形処理だけでなく、表示、データベース等、数多くの処理に特別な考慮を必要とする。これは、システムの負荷を増大させるとともにモジュールやデータの独立性を損なう場合が多い。一方、形状モデラの機能として実現すれば、単にデータ量が削減できるだけでなく、形状の自己同一性を幾何形状自身で表現できる。これは、システムの構成を単純化するだけでなく、作業者の感覚とも合致すると考える。

3. 形状の多重構造表現

本稿で提案する多重化表現は、複数の立体について、その形の異なった部分と同一の部分に分離し、同一の部分に共有して表現する枠組みである。分岐する部分では、隣接位相構造が多重化される。分離した部分はツリー構造、あるいはネットワーク構造を形成する。この表現では、共有部分を指定してもどの立体に属しているか特定することができない。これが、通常の形状モデラと大きく異なる点である。そして、集合演算、表示等、一つの立体として表現する必要がある時に、全体あるいは部分的に一体化し、通常の立体表現と操作が可能となる。一体化には、部分の情報を保持した

まま、一時的に合併するだけの merge 処理と、実際に融合して元の部分が消滅する melt 処理がある。Merge や melt 実現のためには、ネットワーク内のパス群を指定する仕組みが必要である。このパスに沿って立体を組み合わせることで立体を形成する。また、操作の時間的な順序はパスのノードの順序により表現されている。

この多重表現は通常の B-rep 表現だけではなく、非多様体も表現可能である。つまり、いわゆる「位相」で「幾何」の接続状態を表現する立体表現に適用可能である。そして、merge された立体はそのモデリングドメインにおいて意味のある立体表現となり、melt すれば完全に立体表現となる。Melt では、矛盾や自己交叉を検出する正当性検査処理が必要である。

形式的には、Hybrid 表現に近いが、意図するところはかなり違っている。どちらかと言えば、undo/redo の仕組みを一般化したと言える。

3.1 多重位相モデラ

本稿では多重化表現の一つとして、擬立体によるネットワーク構造 (Quasi-Solid Network Structure: QNS) による多重化構造を提案する。QNS は、B-rep 表現の立体モデラを基にした形状表現の枠組みである。ひとことで言えば、face 間の関係をアークとし、擬立体をノードとするネットワークで形状を表現するものである。擬立体は参考文献^[5]で提案したものであり、後で詳しく説明するが、実体のない face を含んだ B-rep 立体表現である。

本稿では、多重位相モデラの表現構造を解説するにとどめ、演算系については別の機会に行う。

3.2 多重位相モデラ表現の特徴

簡単な例で QNS の特徴を説明する。図 3 のように、まず立体 A にボス形状 X を付け、立体 B を作る操作を行う。その後、設計変更で立体 A に別のボス形状 Y の取り付けを行い、立体 C を作成したとする。立体 A、B と C を同時に保持したいとき、元の形状 A が $A = a \& b$ 、取り付け後の立体 B が $B = a \& x$ 、立体 C が $C = a \& y$ となるように、擬立体 a, b, x, y に分割する。ここに & は merge を意味する。各立体はそれらの組み合わせで表現する (図 4)。これが QNS による立体の表現である。a と b, x,

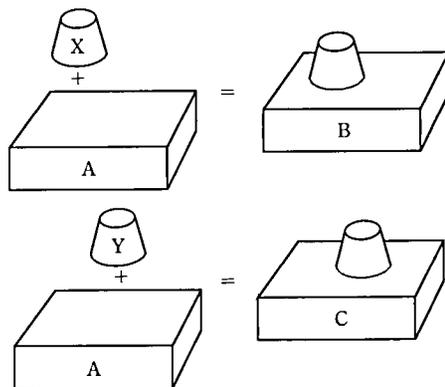


図 3 同一の立体に同時に異なる操作

Fig. 3 Different operations for the same object

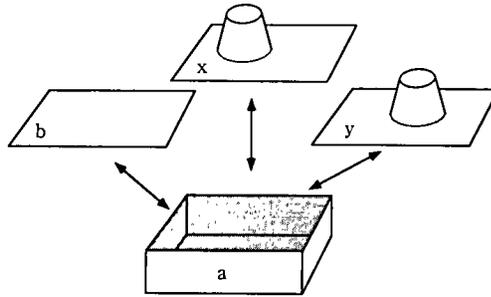


図 4 多重構造表現

Fig. 4 Multiplied structure representation

y の接合部分の面分には幾何実体としての面は存在しない。

QNS による表現は Hybrid 表現とは異なり、ボス形状 x と y の干渉形状を求める必要はない。形状 x と y は擬立体 a を物理的に共有してはいるが、別々に立体表現されているからである。設計変更履歴を QNS で保持することで、履歴管理の手間はかなり減少すると思われる。それは、各設計段階での完全な形状を保持することが可能であるのみならず、その同一性と違いが形状データ自身で表現されているからである。

「同時形状操作機能」も、QNS 形式ならば可能である。作業開始時点で立体 A は共有 DB にあるとする。別々の作業者が同時に X と Y を取り付ける変更を行う。このとき、共有 DB の立体 A は face b が変更されていることだけを知っていればよく、A の位相構造を変更する必要はない。Melt しない限り立体の位相構造は不変だからである。そして、通常は、その日の作業が終了した時点で melt する。このときの正当性検査により、作業者間の干渉を検査する。無論 melt せず、そのまま作業を続行してもよい。

立体 B と立体 C を別な製品として扱うとき、つまり B と C が製品 A の design variation となる時もこの表現が使える。製品群に共通な変更は a に対する操作となり、design variation 個別の操作とは区別される。立体 B と立体 C の集合演算では、同一な部分 a がデータ形式上区別できるため演算の頑健性が向上する。

その他、さまざまな利用法が考えられる。たとえば、各々の擬立体に対してアクセス権を設定することで機密性の高い部分を分離し、外部の作業者から代替形状のみ参照させることも可能である。また NC 計算では、製品形状と実際に加工するワークをこの方法で表現することで、一段と精度のよい加工シミュレーションが実現できる。直接加工する部位以外を簡易形状で代替することで、干渉計算の効率化も図れる。これらが、共通部分を実際に共有しつつ作業できることの意味は大きいと考えている。

実際には、初期段階からの形状をすべて QNS 構造のまま表現し続けることは可能であるが実用的ではない。不要なデータが多くなり過ぎるためである。適切な段階ごとに分岐を解消することが望ましい。

3.3 擬立体ネットワーク(QNS)

QNS について用語の定義、図式表現等について、いくつかの例題を挙げて解説する。

3.3.1 擬 立 体

擬立体 (quasi-solid: qs) は位相的には dummy face^{[6][7]}を持つ B-rep 立体表現に近く、二次元多様体と同相であるものを対象とする。ただし、三次元空間に自己交差なしで埋め込めるとは限らない。幾何データが定義されていない face を仮想面分 (virtual face: vf) と呼ぶ。特徴として、擬立体に定義座標系 Q が付随しており、これと独立な座標系 U で面データ (surface: s) を定義することができる。面データがない場合も座標系 U と、変換 $\phi: U \rightarrow Q$ だけは定義されていることが前提である (図 5)。擬立体はさらに製品座標系 P に配置 ψ することにより実座標値が定まる。一般に $\psi: Q \rightarrow P$ により、 $\psi(qs)$ が擬立体の製品における位置を表現する。そして面の座標は $\psi(\phi(s))$ で決まる。しかし、後で述べる形状特徴モデリング等の特別な場合を除いて、これらの変換は恒等変換、つまり製品座標系に面データが定義されているのと同じ座標値となっている。

合成された立体は内部に仕切りが表現できる。三次元セルモデルと言われるものに近い表現だが、ほとんどの場合、区切り面の幾何データは無いため、結果として通常の B-rep 立体表現とほぼ同等の表現となる。

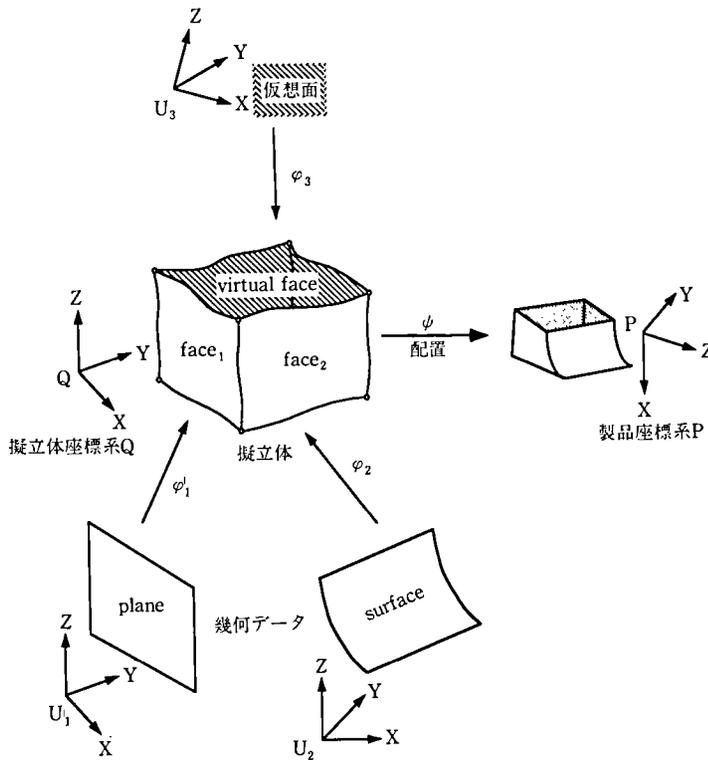
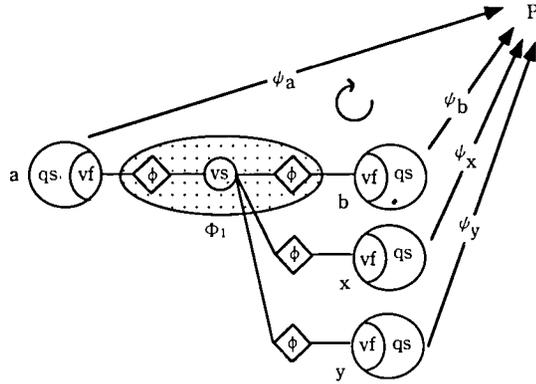


図 5 擬立体の定義

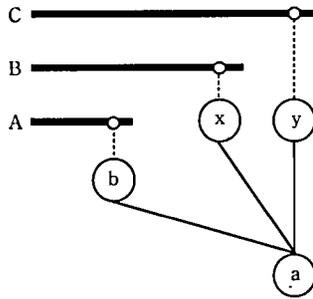
Fig. 5 Definition of quasi-solid

3.3.2 擬立体の接続とネットワーク

擬立体と擬立体のアークによる接続は、面間で面定義座標系を共有することで決まる。つまり、擬立体間の対応する面の間の座標変換がアークに付随している。この



(a) 図式表現



(b) V-linkによるQNS表現

図 6 擬立体ネットワーク構造 QNS の図式表現

Fig. 6 Diagram representation of quasi-solid structure QNS

変換は二つの ϕ の合成であり簡単のため Φ と書く。この Φ を使って図 4 を図式表現すると、図 6(a) のようになる。しかしこのままでは、どれとどれを組み合わせると立体とするかが決まらない。このため、ツリーの葉に相当する部分を束ねるものが必要となる。これを V-link と呼ぶ。つまり各々の組み合わせごとに V-link が対応する。以降、簡単のため Φ , ψ も省略する (図 6(b))。

3.3.3 形状の分割とハンドル分解

実際に多重構造化するためには、立体あるいは擬立体を分割する操作が必要である。これは円に同相な、いくつかのループで面分群を切り取ることで可能となる。図 4 の例では変更範囲に含まれる面分の最外周で切り取っている。もちろん、分割方法にはさまざまな方法があるが、本稿では単純に面分単位での処理例を述べている。

貫通穴をあける等、分割後に操作によって数学的な意味での位相が変化することがある。これは、いわゆるハンドル分解^[9]に対応している。つまり、変形前の立体に、変形のハンドルを加える形となる。これについては次節で述べる。

3.4 多重表現の例

まず、位相が変化する例を説明する。図 7(a) のように立体 A に非貫通穴を開け立体 B とする。同時に、立体 A に貫通穴を開け立体 C とする。このとき、擬立体 a と元の立体 A の隣接位相構造は同じで、表現は図 7(b) のようになる。擬立体 c は変な形のよ

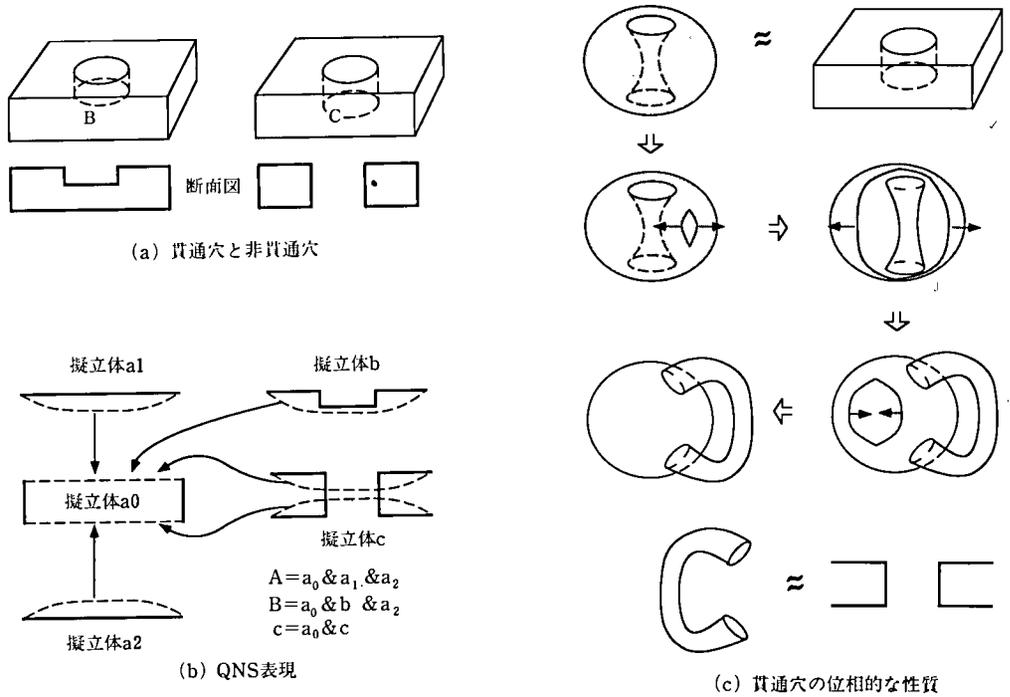


図 7 位相が変化する場合の QNS 表現
 Fig.7 QNS for the change of topology

うに見えるが、三次元空間にうまく埋め込めないだけで、球面と同相である。これは、取手を付けたことと同じであることがわかる (図 7 (c))。

また、図 8 (a)のような三重の部品を QNS 表現すると、図 8(b)のようなツリー構造で表現できる。このとき、立体 C を図 8(c)のように変形して立体 D を作ると、ツリー構造は図 8(d)のように変化する。

3.5 形状特徴モデリングのための拡張

図 9(a)のような台座の設計の例では、三つの足は同一形状であるため、図 9(b)の標準部品を 120 度の角度で回しながら配置したいと考えるのは自然である。配置後に本体と一体化するならば、今までの議論の範囲である。しかし、配置後も標準部品として操作するためには、たとえば、一つの穴の径を変更した時に、他の二つも同期して変わるためには、今までのようにツリー構造では表現できない。このとき、一般的に定義した擬立体が有効となり、図 9(c)のような QNS 構造で表現すれば可能である。こうすると、標準部品への変更はすべての配置された標準部品への操作と見なされる。同一の擬立体において幾何データが共有されているためである。ただし、標準部品の製品空間への変換は未定義である。

さらに、配置後の標準部品の design variation 表現も必要となることがある。図 9 の例で、足の一つの穴の半径を変更するが、他の部分は標準部品として三つ同時に操作したいものとする。これは、いままでの構造では表現できないが、V-link にポインタ

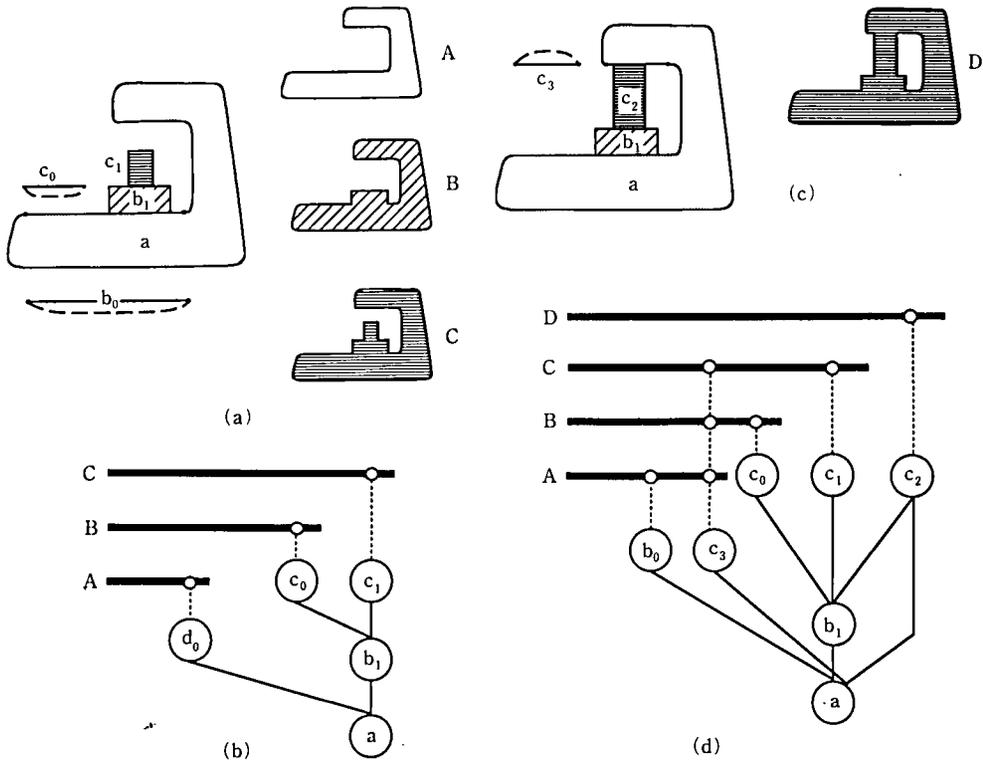


図 8 三重構造の部品の QNS 表現例
Fig. 8 QNS for the layered part

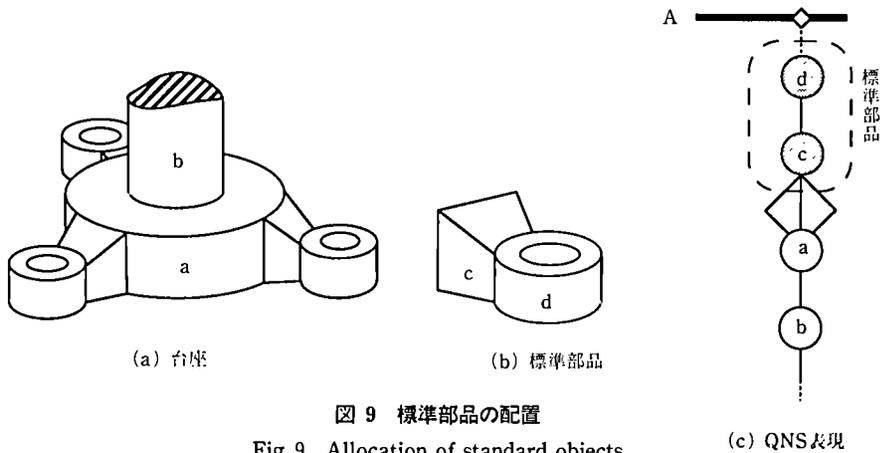


図 9 標準部品の配置
Fig. 9 Allocation of standard objects

を追加する等の変更を加えることで可能となる。しかしながら、紙面の関係から本稿では省略する。

4. おわりに

並行設計や design variation の表現のためには、何等かの多重表現が必要であることを論じた。解決策の一つとして、多重位相構造を提案し、使用例を解説した。

本稿は、筆者が受託研究員として東京大学工学部で行った研究にいくつかの工夫を加えたものであり、当社のCADシステムにそのすべてが実装されているものではない。なお、担当の東京大学工学部教授の木村文彦先生、乾正知先生に多くの示唆、助言をいただいた。ここに深く感謝の意を表する。

- 参考文献 [1] Mäntylä, "An Introduction to Solid Modeling", Computer Science Press, 1988.
 [2] 鳥谷浩志 他, "設計履歴の共有による並行モデリング", PIXEL no. 124, 1993.
 [3] G. A. Crocker, F. R. William, "An Editable Nonmanifold Boundary Representation", IEEE CG & A, Vol. 11, No. 3, 1991, pp. 39~51.
 [4] K. Weiler, "The Radial Edge Structure: A Topological Representation for Non-Manifold Geometric Boundary Modeling", Geometric Modeling for CAD Applications, Sorth-Holland, 1988, pp. 3~36.
 [5] 松本則夫, 木村文彦, "隣接位相構造の多重化による製品モデルの表現", 1991年精密工学会秋期全国大会学術講演会講演論文集, 1991, pp.817~818.
 [6] P. R. Wilson, "Euler Formulas and Geometric Modelling", IEEE CG&A, Vol. 5, No. 8, 1985, pp.24~36.
 [7] 谷本茂樹, "サーフェスマデラとソリッドモデラの統合", ユニシス技報, Vol. 11, No. 4, 1992, pp. 179~201.
 [8] 岩波数学辞典第3版, 岩波書店, 1985.

執筆者紹介 松木 則夫 (Norio Matsuki)

昭和55年早稲田大学大学院数学専攻修了。同年日本ユニシス(株)入社, CAD/CAMアプリケーションの開発業務を担当。現在, 製造工業システム本部 システム三部に所属。



鑄鍛造金型用形状モデラ

A Geometric Modeler for forging/casting dies

谷本茂樹

要約 筆者らは、某自動車会社で、自動車ユニット部品の鑄鍛造金型用形状モデラを開発した。鑄鍛造金型用の形状モデラには、曲面と立体を同一のデータ構造で表現できることや、分割結合演算、フィレット掛け、形状復元等の機能を持つことが要求される。開発したモデラは、これらの要求を満たし、実用システムとして十分な、安定性と効率の良さを実現した。

本稿では、① 鑄鍛造金型の形状と形状生成手順の特徴、② 鑄鍛造金型の形状を計算機上に生成するための形状モデラに対する要求、③ この要求を実現する上での問題点、④ 開発した形状モデラのデータ構造と機能について述べる。

Abstract We have developed a geometric modeler for the die of forging and casting in an automobile company. In this development, we analyzed the die shape and the modeling process, and we extracted the requirements of data representation and functions for the modeler. It is required to represent surfaces and solids in a unified data structure and to store the shapes deleted by operations. We realized the requirements by following methods; a dummy face is added outside to a surface, and a model is represented as a non-manifold shape composed by shells, which store the deleted shape by each operation. And as modeling functions we supply a divide-and-join (such as a set operation between solids and surfaces), a filletting and a restoration operation (which restores a shape modified by a divide-and-join or a filletting operation). The modeler realizes sufficient functions, stability and high performance to apply to the practical use.

1. はじめに

自動車の部品は、車体系部品とユニット系部品に大別される。車体系部品には、車体外板や内装部品があり、その形状には意匠性が要求される。一方、ユニット系部品には、エンジン、トランスミッション等の部品があり、その形状には自動車としての機能を満足することが要求される。ユニット部品の代表的な素材材（鑄鍛造等により製造された後、機械加工されるまでの部品）製造工程に鑄造と鍛造がある。

自動車のCAD/CAM化は、車体系の形状モデラを中心に発展してきた。鑄鍛造金型のCAD/CAM利用では、二次元の製図システムが中心であり、三次元の形状モデラの普及率は低い。このことは、鑄鍛造金型では、車体に比べCAD/CAM化する利点と要求が少ないため、また表現すべき形状が複雑なためである^[4]といわれている。しかし、ユニット部品の高品質化、軽量化、金型設計製作の工数と期間の短縮等が要求され、また将来、木型作成者の不足が予想されることから、鑄鍛造金型の設計、形状生成にも三次元形状モデラが浸透し始めている。

鑄鍛造金型の形状生成に三次元形状モデラが使われ始めたころは、某自動車会社では、車体外板用に開発された形状モデラに鑄鍛造用の機能拡張を施し使用していた^[9]。しかし、車体外板と鑄鍛造等のユニット部品では、図面の描き方、形状生成手順等の

点で、形状モデラに対する要求が異なる^[13]。鑄鍛造金型用の形状モデラには、車体外板のような意匠形状を作ることは要求されず、少ない期間と工数で図面に忠実な形状を生成することが要求される^[4]。外板用の形状モデラを鑄鍛造金型の形状生成に使うのでは、作業効率が悪く、適用部品も一部に限定される^[19]。その後、EUKLID, GRADE, U-Graph 等^[27]、鑄鍛造金型の形状生成に利用できる形状モデラが市販され、鑄鍛造用の形状モデラを自社開発する自動車会社も現れた^[15]。

筆者らは、某自動車会社で鑄鍛造金型用の形状モデラ(本稿では、F モデラと呼ぶ)を開発した。F モデラは、鑄鍛造金型形状が描かれている図面をもとに、金型の三次元形状を計算機内に形状データとして構築するシステムである。F モデラが生成した形状データは、工具軌跡計算システム^[6]により、切削データ計算用の形状データに変換され、工具軌跡が計算される^{[10][11]}。

F モデラに関して、学会等にすでに発表しているが^{[2][7]~[13][25][26][37]}、本稿は計算機会社のシステム・エンジニアとしてシステム構築の観点で既発表内容を整理し、補足したものである。本稿では、以下のことについて述べる。

- ① 鑄鍛造金型形状の特徴と形状生成手順の特徴
- ② 鑄鍛造金型の形状モデラに対する要求
- ③ 要求を実現するための問題点
- ④ F モデラでの問題点の解決方法

①については2章で述べる、②③④については、データ表現(3章)と機能(4章)に分けて述べる。

2. 鑄鍛造金型の特徴

本章では、鑄鍛造金型の形状を分析し、CADにより鑄鍛造金型形状を生成する場合の、形状生成手順と形状認識方法を分析する。分析の結果、鑄鍛造金型用のモデラに対する要求を明確にし、システム開発のために必要な技術課題を抽出する。

2.1 鑄鍛造金型形状の特徴

鑄鍛造金型形状には、次の特徴がある。

- 1) 基本形状には解析曲面が多い…鑄鍛造金型形状を決める線は、円弧と線分である。三面図中の円弧と線分は、二次元の線を意味するだけでなく、線の描かれている視線方向への掃引により生成される円柱面、平面や、軸回りの回転により生成される円柱面、円錐面、球面、円環面を意味する。なお、部品の基本形状に解析曲面が多く使われるのは、自動車の機械部品が平行移動や回転運動により力を伝えるため、他の部品と接続する部分を正確な寸法で表現し、機械加工を施すため等の理由による。
- 2) 鑄鍛造金型形状には自由曲面もある……鑄鍛造金型には、次のような自由曲面が現れる。なお、鑄鍛造金型に現れる自由曲線、自由曲面は、自己交差や製造上不都合な歪がなければ、図形の微分連続性や、意匠性は通常問題にされない。

① 基本形状に現れる自由曲面(図1(a))

基本形状には、1)で述べたように解析曲面が多いが、自由曲面も現れる。基本形状に現れる自由曲面には、2組の曲線列上の点を線分で繋ぐ線織面や、複

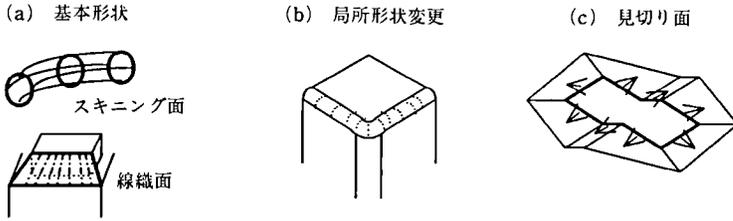


図 1 金型に現れる自由曲面

Fig.1 Free form surfaces representing forging/casting dies

数の断面線を滑らかに繋ぐ（マニホールド等のチューブ形状に使われる）スキニング面等がある。

② 局所的な形状変更により作られる自由曲面（図 1 (b)）

基本形状に、フィレット掛け、抜き匂配掛け等の製造上必要な局所形状変更が加えられる。この形状変更により作られる曲面の多くは自由曲面である。

③ 見切り面を構成する自由曲面（図 1 (c)）

素形材側面の型の分割線を見切り線といい、見切り線の外側の金型の合わせ面を見切り面という（図 2）。見切り面の形状は平面のこともあるが、見切り線が一つの平面内になく、三次元的に変化する場合は、見切り線の区間ごとに異なる方向に掃引される（掃引方向が徐変することもある）自由曲面になる^{[3][4]}。

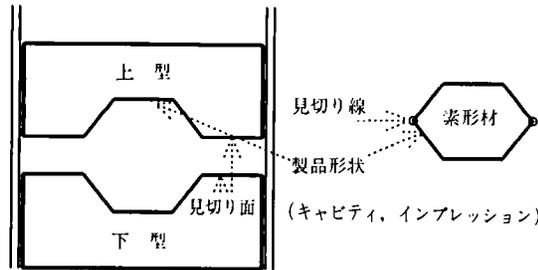


図 2 見切り線と見切り面

Fig.2 Parting line and parting surface

- 3) 面数が多い……鑄鍛造金型形状を構成する個々の曲面は、解析曲面等の形状の単純な面が多い。しかし、金型全体の形状は、面数が多いので複雑になる。したがって、サーフェスマデリングのように曲面を個々に生成し、個々にトリム（部分面化）するのでは作業能率が悪い。ソリッドモデリングのように、曲面群を一括して処理する方法が適している。
- 4) 多様体である……鑄鍛造金型の表面形状は、多様体^[16]の図形（面上の微小な部分が平面または半平面に変形可能な図形）としてモデル化される。図面をもとに金型形状を生成する過程で、操作者は多様体の範囲内で形状を認識し、形状生成手順を考える。

操作者は頭の中で思い描いている形状を計算機内に生成したいのであり、現実には存在しない非多様体形状（多様体でない形状）を生成することは、多様体形

状を生成しようとする操作者の思考を混乱させることになると思う。鑄鍛造金型の形状生成に限れば、形状は常に多様体であることが求められ、生成過程で一次的にでも非多様体形状を作る必要はない。たとえば、立体間の集合演算の結果、非多様体を生成する場合は、操作者の指示誤りであると考え、非多様体形状は積極的に排除したい^[8]とも言われている。

- 5) フィレットが掛かる……鑄鍛造金型形状には、部品機能、部品剛性、製造性(溶けた金属を流れやすくする等)の実現のためにフィレットを掛ける。鑄鍛造金型形状に現れるフィレットには、面と面の間に丸みが付いていけばよい部分と、部品や金型の強度上の理由で丸み付けの順序が重要な部分がある。とくに、鍛造部品には強度と剛性が、鍛造金型には耐久性が要求される。図3に示すように、フィレットを掛ける順序が異なると作られる形状も異なる。そのため、フィレット掛けの順序が強度に大きく影響する^[13]。

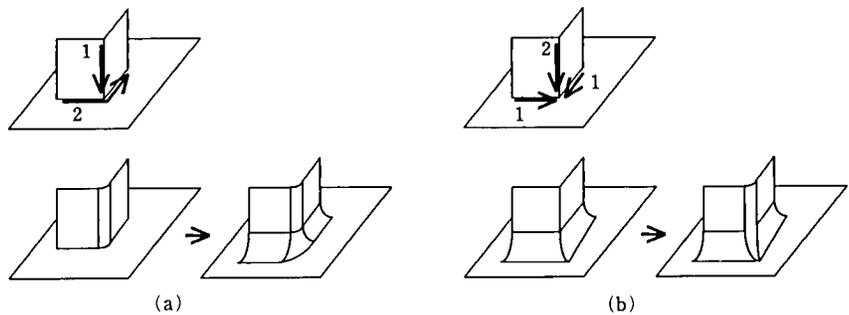


図3 フィレット掛けの順序

Fig. 3 Sequence of filletting operations

鍛造の木型作成者は、接平面連続でない部分を一つ一つ順番にフィレット掛けする。フィレット面の重なる部分では、形状をばかすのではなく、先に掛けたフィレット面の上に次のフィレットを掛ける。CADシステムを使用する形状生成過程が、木型の形状生成過程と同一でない、金型の耐久性や素形材の製造性に関する過去の経験的な情報を活用できない。その意味でも、フィレット掛けの順序は重要である。

- 6) 抜き勾配が掛かる……たとえば、お菓子のプリンやゼリーの形状は、円柱でなく円錐台の形をしている。プリンの形状が円柱だと、プリンをプリンの金型である容器から取り出しにくい。プリンを容器から抜き取りやすくするために、円柱の縦壁に傾斜を付け、円錐台にしている。

同じことが素形材の製造についても言える。素形材を金型から抜き取る際の抵抗力を下げ、製品の破壊や変形を防ぐために、製品面と抜き方向(素形材が金型から離れる方向)のなす角度が抜き角度以内の形状部分に勾配を付ける(図4)。この勾配を抜き勾配といい、抜き勾配の付加された面を抜き勾配面という。抜き勾配が掛かることは、鑄鍛造金型形状の重要な特徴である。

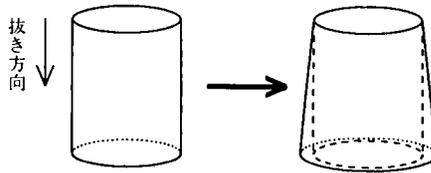


図 4 抜き勾配掛け
Fig. 4 Drafting

2.2 鋳鍛造金型の形状生成の特徴

CAD システムを使用して鋳鍛造金型形状を生成する場合の、形状生成手順と、操作者の形状認識の特徴について述べる。

2.2.1 形状生成手順

鋳鍛造金型形状の生成手順の特徴について述べる。

- 1) 基本的な形状生成手順は、プリミティブ生成、抜き勾配掛け、フィレット掛け、集合演算により構成される。

鋳鍛造金型は、基本的に次の手順で生成される (図 5)。

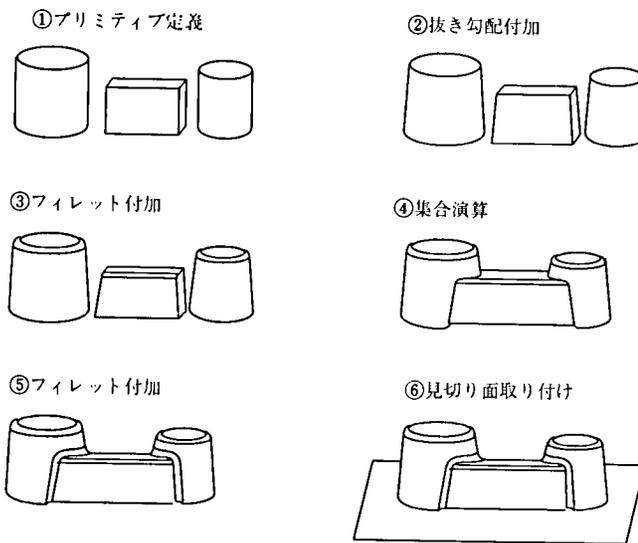


図 5 鋳鍛造金型におけるモデリング手順
Fig. 5 Modeling process in forgings/castings

- ① ブロック、フランジ、ボス、リブ、穴等のプリミティブを定義する。
- ② プリミティブに抜き勾配を掛ける。
- ③ プリミティブにフィレットを掛ける。
- ④ 集合演算をする。
- ⑤ 集合演算の結合部にフィレットを掛ける。
- ①～⑤を繰り返す。
- ⑥ 見切り面を付加する。

鋳鍛造金型の形状生成に使われる基本的な演算は、プリミティブ生成、抜き勾配掛け、フィレット掛け、集合演算である。これらの演算の繰り返しにより、形状生成が進められる。

2) 形状の変更が頻繁に行われる。

鋳鍛造部品では、既存の部品と一部形状の異なる類似部品を作ることや、設計変更のため一部の形状を変えることがある。そのため、集合演算やフィレット掛け等により変更された形状を、他の部分とは独立に、演算前の形状に戻す必要がある。

3) 曲線と曲面の生成順序が、外板の形状生成とは異なる。

車体外板では、形状の特徴を表現する曲線が先に生成され、曲面は曲線から補間して生成される。一方、ユニット部品では、部品機能を実現するための曲面が先に生成され、曲線は曲面の交線として生成される。三面図に描かれている線は、三次元の曲線を意味するのではなく、線を掃引や回転して得られる曲面を意味する。(ただし、これは基本的な生成順序であり、鋳鍛造でも、2.1節で述べたように、二組の曲線列を結んで線織面を作ることもある。)

4) 設計過程や木型作成過程と同じ作業手順を計算機を使用して実現する。

設計者でも、複雑な三次元形状を頭の中で想像するのは困難である。ユニット部品の設計過程では、二次元空間で設計検討を行い、その形状が三次元空間で成立するか検証することが多い。すなわち、二次元の円弧と線分をもとに、三次元の曲面や立体を生成する。その形状をグラフィック表示し、干渉検査、断面計算、距離計算、マスプロパティ計算等を行う。

また、鋳鍛造金型形状を生成する操作者は、木型を作るのと同じ手順と方法で、計算機内に形状を生成したい。木型を成形するように、形状モデルを変形する機能が必要である。

2.2.2 形状認識の方法

鋳鍛造金型形状の生成過程での、操作者の形状認識方法には次の特徴がある。

1) 特徴を持つ形状ごとに、操作者は形状を認識する。

サーフェスマデラでは、曲面と曲線を個々に定義し、曲面を曲線でトリムする。しかし、ユニット部品の設計者は、個々の形状をこのように曲面単位に分解しては考えない。

フランジ、ボス、リブ等の形状に特徴を持つプリミティブの単位で、操作者は形状を認識する。プリミティブを集め、交差させ、不要部分を削除して、基本形状を作る。

2) 複合曲面や立体の単位で操作する。

車体外板の形状を生成する操作者は、個々の曲面を操作単位とし、パネル全体の形状を、トリムされた曲面が粗に結合した面の集合であると考ええる。

一方、ユニット部品の形状を生成する操作者は、部品形状全体を曲面が密に結合した複合曲面や立体として捕える。曲面は、複合曲面や立体の構成要素と考える。ユニット部品の形状生成の操作単位は、複合曲面や立体という形状全体、または、1)で述べたように操作者にとって意味のある(プリミティブ等の形状に特

徴を持つ) 曲面群である。

3) 次のように、複合曲面と立体を区別せずに扱う^[16]。

- 操作者は表面形状に注目し、立体と複合曲面とを同等に扱うことが多い。操作対象の形状が立体であるか否かは、操作者には重要でない。
- 金型は立体であるが、操作者は金型の一部分である製品形状部(キャビティ、インプレッション) や見切り面だけを複合曲面として生成する。しかし、この形状生成過程でも、操作者は複合曲面を立体の一部であるかのように考える。
- 形状生成の途中で、立体から一部の面を取り除き、複合曲面として扱うことも、複合曲面を貼り合わせて立体を作ることもある。また、一部の面の形状が未定義な状態の、不完全な立体のまま、立体中の他の部分の形状生成を進めることもある。
- 曲面が部分的に一致する立体間の集合演算や、平行に近い曲面間部分のフィレット掛け、幾重にもフィレット面が重なった部分のフィレット掛けを行うと、従来の形状モデラでは、処理が正常に行われないことや、操作者の意図する形状が得られないことがある。このような場合、サーフェスマデリングのように曲面を切り貼りして、形状生成作業を進める必要がある。

形状の特徴、形状生成過程の特徴を以上のように分析し、F モデラの開発のために必要な技術課題を以下のように抽出した。この中の主な項目について、次章以降で述べる。

- ・解析曲面と自由曲面を扱う (3.1 節)。
- ・複合曲面と立体を区別せずに表現し、統一的に扱う (3.2 節)。
- ・集合演算 (4.1 節)、フィレット掛け (4.2 節) を高速で安定的に実行する。
- ・集合演算等により変更された形状を、演算順序に関係なく複元する (4.3 節)。

3. データ表現

形状モデラは、理想化された実世界の形状を計算機で扱える形式に変換して表現する。この計算機内で表現される形状をモデルという。鋳鍛造金型のモデルには、2.1 節で述べた形状を表現することが要求される。

F モデラのモデル表現方法は、立体の表面形状を隣接関係を持つ図形要素の連結体として表現する境界表現法(B-rep)である。この章では、F モデラのモデル表現方法を、図形要素の幾何形状表現(幾何表現)と図形要素間の隣接関係表現(位相表現)とに分け、次の順序で検討する。

- ① データ構造に対する要求
- ② 要求を満たすための問題点
- ③ F モデラでの要求の実現方法

3.1 幾何表現

曲面の表現方法を例に、幾何表現を説明する。

[要 求] 鋳鍛造金型の基本形状には解析曲面が、フィレット面、抜き勾配面等には自由曲面が多く使われる。そのため、解析曲面と自由曲面の両方を表現することが要求される。

形状を構成する面が解析曲面であると、操作者は面が解析曲面であることを意識し、面の生成情報を知りたいことがある。たとえば、穴の側面が円柱面であることを操作者は意識し、円柱面からその軸方向や半径を抽出したい。

[問題点] 解析曲面と自由曲面の両方を扱うために、次の二通りの実現方法を検討した。それぞれの利点と問題点を述べる。

- 1) 解析曲面を自由曲面で近似する方法……解析曲面を自由曲面で近似すると、曲面表現式を1種類に統一できる。しかし、解析曲面を用いて直接計算する場合に比べると、次の問題がある。
 - ① 近似による誤差が発生する
 - ② 計算時間が掛かる
 - ③ 計算が不安定である
 - ④ 近似後の自由曲面から、もとの解析曲面の定義情報を、誤差なく抽出するのは困難である。

なお解析曲面を有理式自由曲面で表現すると、解析曲面を近似誤差なく（計算機の丸め誤差は残るが）表現できる。しかし、有理式を扱い、ウェイトという形状を制御する独立変数が増えるため、効率、安定性がさらに劣化する^{[31][4]}。

- 2) 自由曲面と解析曲面とを併用する方法……図形処理は、解析曲面のまま扱うと、近似誤差がなく高速で安定する。また、解析曲面を直接扱うため、解析曲面の定義情報を曲面式から抽出でき、演算の結果、生成される図形も解析図形として求めることができる。たとえば、円柱面と、その円柱の軸に垂直な平面との交線は、円として求めることができる。しかし、すべての曲面種類に対応する図形処理ルーチンを用意する必要があり、図形種類に応じて図形処理ルーチンを使い分ける必要もある。

[実現] 以上の検討から、2)の方法を採用し、高速で頑健な図形処理を実現した。

図形種類に応じて、図形処理ルーチンを使い分けるアプリケーション・インタフェースを用意している。

3.2 位相表現

[要求] 2種類の要求がある。

- 1) 複合曲面と立体を同一のデータ構造で表現したい……操作者は、複合曲面と立体の区別を意識せずに形状を認識し、操作の対象としたい。そのため、複合曲面と立体を同一のデータ構造で表現したい。
- 2) 演算により消滅する形状を保存したい……後述する形状復元機能の実現のために、演算により消滅する形状をモデル中に保存する必要がある。

[問題点] この二つの要求を実現するためのそれぞれの問題点を述べる。

- 1) 複合曲面と立体の同一データ構造による表現……数年前から、サーフェスマデラとソリッドモデラを統合する形状モデラが研究されている。形状モデラの統合に関する技術動向を前回報告^[16]したので、本稿では統合の問題点の詳細については述べない。統合を実現するモデルとして、相沢の胞複体鎖モデル^[1]、Weilerのラジアルエッジ構造^[39]等、複合曲面と立体だけでなく、非多様体と呼ばれる曲線、曲面、立体の連結体まで統一的に表現するデータ構造が提案されている。最近で

は、市販の形状モデラにも、複合曲面と立体を同一のデータ構造で表現するものがある。発表されているデータ構造には問題もある。たとえば、ラジアルエッジ構造は、ソリッドモデラで多く用いられているウイングドエッジ構造に比べ、位相要素と位相要素間の関係の種類が多い。そのため、データ量と計算時間が増え、位相処理が複雑になる。

- 2) 演算により消滅する形状の保存……演算により消滅する形状を、モデルの中に保存し、形状復元に利用する方法として、増田らが、CSG/非多様体 B-rep のハイブリッド表現を提案している^[32]。消滅する形状を隣接関係を保持したまま保存するには非多様体を表現する必要がある。

[実 現] まず、F モデラのモデリング空間(表現可能な範囲)について説明する。次に、シェル内部の位相表現方法とシェル間の位相表現方法とに分けて、F モデラの位相表現方法を説明する。

- 1) モデリング空間……F モデラが使用する図形要素の位相的意味を示す(図 6)。なお、図形要素の中で、ボディ、シェル、ループは位相情報だけを、フェイス、エッジ、バーテクスは、位相情報と幾何情報の両方を持つ。
 - ボディ：一つの立体や一つの複合曲面を示す形状全体を表現する単位である。複数のシェルを連結して、非多様体を表現することができる。
 - シェル：立体の表面や複合曲面を示す。面・線・点の連結体であり、向き付けられた(表裏の区別のある)多様体である。シェルには、ライブシェルとデッドシェルの 2 種類がある。ライブシェルは、操作者が形状を認識し、操作の対象とするシェルである。デッドシェルは、集合演算等の演算により消滅する部分の形状を、将来復元するために保存するシェルである。ボディは、1 個のライブシェルと 0 個以上のデッドシェルからなる。デッドシェルには連結性の条件を適用しない。
 - フェイス：一つの面を表現する二次元の図形要素である。フェイスの幾何形状は、ベース曲面(フェイスを包含する曲面)をループを境界としてトリムし

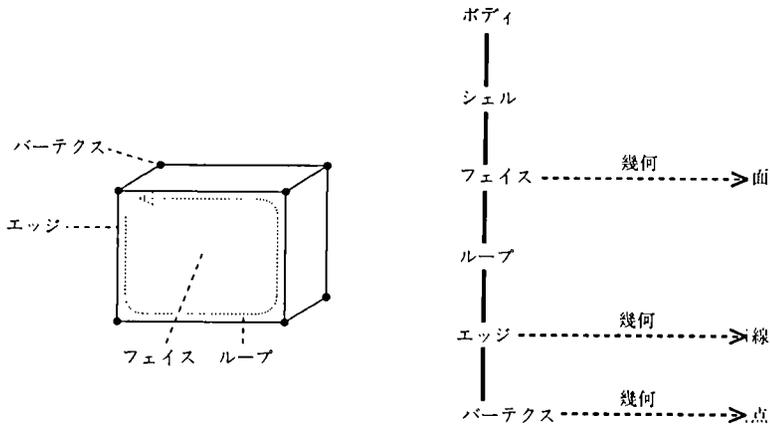


図 6 F モデラの図形要素

Fig. 6 Shape elements in F modeler

たものである。フェイスは向き付けられている。面形状を表現する幾何情報を持つ。

- ループ：フェイスの境界を示すエッジとバーテックスの列である。フェイスは、1個以上のループを持つ。ループは円と同相である（同じエッジやバーテックスを複数回通らない）。また、ループは向き付けられている（方向がある）。
- エッジ：一つの辺を表現する一次元の図形要素である。エッジは線分と同相である（両端は異なるバーテックスである）。エッジは向き付けられている。線形状を表現する幾何情報を持つ。
- バーテックス：一つの点を表現する零次元の図形要素である。点を表現する幾何情報（座標値）を持つ。

ループが円と同相であるため、図7に示すフェイス内のぶら下がりエッジや、孤立エッジ、孤立バーテックスの表現を許さない。また、エッジが線分と同相であるため、両端が同一バーテックスであるエッジの表現を許さない。（演算結果として、このような位相を生成する場合、その演算をエラーとする。）表現範囲を限定することで、位相処理やトリムされた曲面を対象とする図形処理のアルゴリズムを単純にし、処理速度を高速にした。鋳鍛造金型の形状表現では、ぶら下がりエッジや孤立エッジ、孤立バーテックスを持つフェイスを表現する必要はない。さらに、このようなフェイスは設計者を混乱させるので、積極的に排除すべきだという報告もある¹⁸⁾。

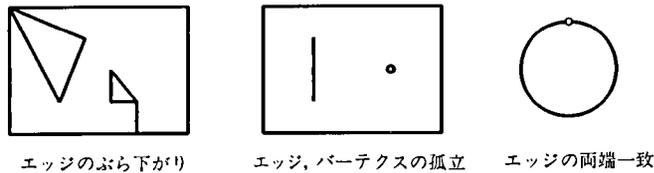


図7 許容しない位相

Fig. 7 Prohibited topology

F モデラのモデリング空間には、操作者のモデリング空間として多様体に限定することが要求され、消滅形状の保存のために非多様体が要求される。この異なる二つの要求を満たすために、外部モデリング空間と内部モデリング空間の概念を導入する。

- ・外部モデリング空間：操作者が生成可能な形状の範囲
- ・内部モデリング空間：システムが内部的に表現している形状の範囲

操作対象となる形状は一つのシェル（ライブシェル）であるため、外部モデリング空間は、向き付け可能な二次元多様体である。一方、内部モデリング空間は、演算により消滅した形状を、形状復元のためにシステムが内部的に保存するので、一つ以上の多様体シェルがエッジを介して接続する非多様体になる。すなわち、操作者が形状生成するのは多様体の範囲であるが、内部的には非多様体表現をする。

なお、F モデラは、これまで述べてきた位相を持つ図形とは別に、無限平面等、

位相のない図形も扱う。無限平面を扱うのは、操作者の思考過程で無限の広がりを持つ平面を必要とするからである。

- 2) シェル内部の位相表現方法……立体表面や閉じた複合曲面を閉シェル、開いた複合曲面を開シェルという。閉シェルのエッジは二つのフェイスに隣接する。一方、開シェルも内部のエッジは二つのフェイスに隣接するが、外周のエッジは一つのフェイスにだけ隣接する。閉シェルと開シェルを同一のデータ構造で表現するためには、開シェルの外周エッジの隣接関係の表現方法が問題になる。開シェルの外側に、幾何実体を持たないフェイス(ダミーフェイス)を補うことにより、位相的には開シェルを閉シェルと同等に扱うことができる(図8)^[40]。たとえば図9の箱は、蓋フェイスの有無により閉シェルと開シェルになるが、この二つの箱は同一の位相を持つ。

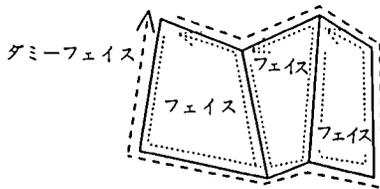


図 8 開シェルの表現方法

Fig. 8 Representation of an open shell

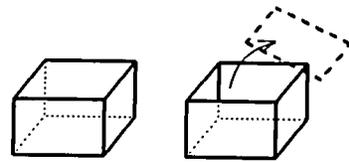


図 9 蓋を持つ箱と持たない箱

Fig. 9 A closed box and an open box

F モデラは、このような同一のデータ構造で曲面と立体を表現する。要求を満たすなら、データ量、計算速度と、開発工数、保守工数の点で、データ構造は単純なほど良いと思う。

- 3) シェル間の位相表現方法……ライブシェルとデッドシェルの間や、デッドシェル同士の間には、図10のように、演算前には繋がっていたエッジを介して接続関係が定義される。この接続関係を、非多様体接続関係と呼ぶ。一つのエッジは、隣接する二つのフェイスごとに、それぞれのフェイスと演算前に繋がっていたデッドシェルのエッジに対し、非多様体接続関係を持つことができる。シェル間の位相表現方法の詳細は、4.3節の形状復元で述べる。なお、形状復元のために内部的に非多様体を表現するが、操作者が形状生成作業を行うのは、あくまでも多様体の範囲内である。

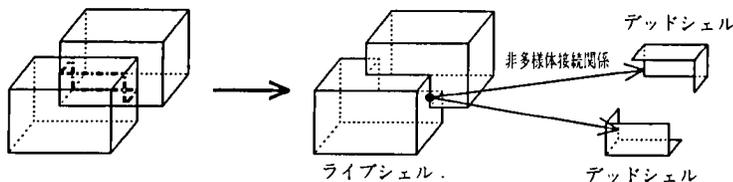


図 10 デッドシェル

Fig. 10 Dead shell

4. 機能

前章で述べたデータ構造は、図形要素間の隣接関係を表現するため、サーフェスモデルやワイヤフレームモデルに比べると複雑である。F モデラは、この隣接関係を利用して、操作工数を低減させ、要求される形状生成機能を実現している。

本章では、鋳鍛造金型用の形状モデラに要求される機能の中から、分割結合演算(集合演算)、フィレット掛け、形状復元を取りあげる。データ構造と同様に、機能に関しても次の順序で検討する。

- ① 機能に対する要求
- ② 要求を満たすための問題点
- ③ F モデラでの要求の実現方法

4.1 分割結合演算⁹⁾

本節では、集合演算の対象を複合曲面にまで拡張する演算について述べる。

[要 求]

- 1) 曲面立体間の演算……ユニット部品の形状生成では、部品機能を満たすプリミティブを集め、その中から必要な部分を切り取り繋ぎ合わせるにより目的の形状を生成する。必要な部分を集める操作は、ソリッドモデラでは集合演算により実現されている。しかし、鋳鍛造金型形状には、立体だけでなく複合曲面のプリミティブもある。そのため、立体と複合曲面を区別せず、必要な表面部分を集め不要な部分を削除する(集合演算のような)演算が必要になる。

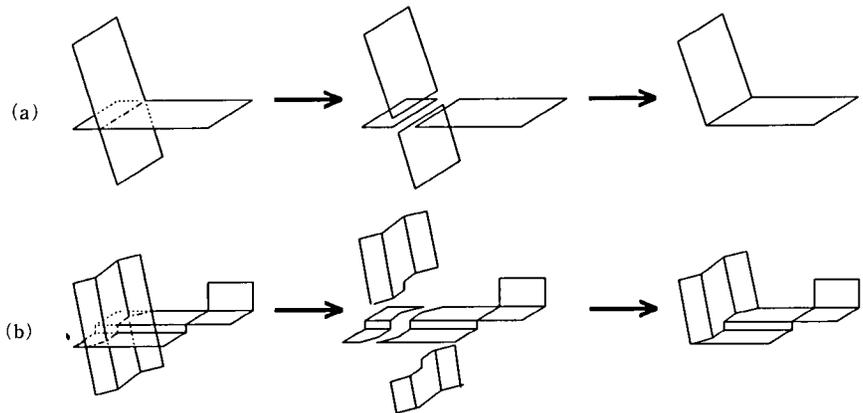


図 11 曲面の交線による分割・結合

Fig. 11 A Divide-and-join operation of surfaces at an intersectional curve

サーフェスモデラでも、二つの曲面をその交線で互いにトリムしたいことがある(図 11(a))。複合曲面の形状生成についても同様で、二つの複合曲面をそれぞれ交線で分割し、必要な部分を繋ぎ合わせる事が要求される(図 11(b))。同じ機能の演算を立体と曲面を対象に施した例を図 12 に示す。

- 2) 高速で頑健な集合演算……集合演算は、概念としても、実際の形状生成過程においても、ソリッドモデラの最も重要な演算である。しかし、従来のソリッドモデラの集合演算は、計算時間を多く要し、安定して解を得ることが困難な演算で

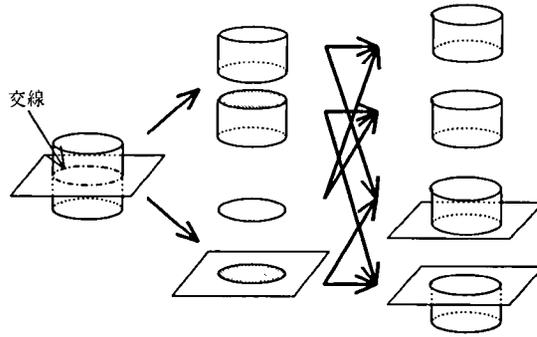


図 12 曲面立体間の分割結合演算

Fig. 12 A divide-and-join operation between a surface and a solid

あった。とくに、鋳鍛造金型形状の生成は次の特徴を持つため、集合演算（に相当する演算）の高速性と頑健性が他の適用分野以上に強く要求される。

- 鋳鍛造金型形状は、フィレット掛けや抜き勾配掛け等により生成される自由曲面を含む。そのため、自由曲面を形状の中に含む立体や複合曲面の集合演算を実現する必要がある。自由曲面を扱うと、不安定さと計算時間が増加する。
- 鋳鍛造金型形状は、面数が多く複雑であるため、1回の集合演算で計算対象となるフェイスの数が多い。

[問題点] (詳細は、文献^{[9][16]}参照)

- 1) 複合曲面立体間の集合演算に相当する演算の概念が未定義であるという問題点
 ……境界表現法によるソリッドモデラの集合演算は、立体間の交線でそれぞれの立体を分割し、必要な分割部分を集める演算である。要求される演算も、集合演算と同様に複合曲面や立体を交線で分割し、必要部分を集める演算である。しかし、要求される演算は集合演算ではない。

ソリッドモデラにおける立体間集合演算は、三次元空間を点の集合と考え、立体に属す点集合の間の集合和、集合差、集合積を求める演算であると定義される。立体と複合曲面の間の集合演算を、集合演算本来の定義に従い実行すると、図 13 のようになる。これは要求される図 12 の演算とは異なる。(要求される形状変更

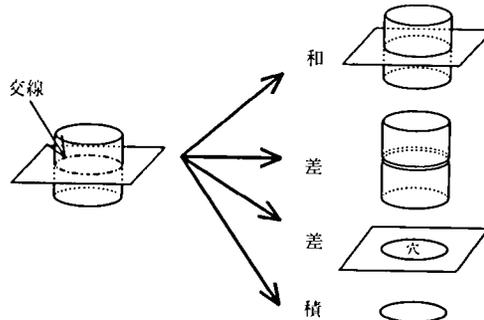


図 13 曲面立体間の集合演算

Fig. 13 A set operation between a surface and a solid

は、集合演算を組み合わせれば実現できるが、1回の集合演算では実現できない。) 要求される演算と集合演算は、処理内容は類似しているが、演算の概念は異なる。要求される演算は、従来の形状モデラにはない演算であり、演算の概念を定義する必要がある。

要求される演算と同様の演算を、佐藤らが提案している。これは、複合曲面を、ダミーフェイスに幾何実体を与える時にできる立体であると考え、立体間の集合演算と同一の考え方で複合曲面間の集合演算を実行する方法である^[36]。しかし、この演算の概念が定義されていない。すなわち、形状が曖昧な立体同士の集合和(差・積)とは何であるのか不明である。また、この方法で、集合演算に和・差・積の指示をするには、操作者は複合曲面の表裏の区別を意識する必要があり、操作者の頭の中で複合曲面を立体化する作業が必要になるという別の問題もある。

- 2) 集合演算の高速化と頑健化における問題点…… 鋳鍛造金型形状の生成には、自由曲面を形状の中に含む立体の間の集合演算を実現することが必要である。このような集合演算を実現しているソリッドモデラは少ない。実現しているソリッドモデラでも、自由曲面を多面体近似する等の回避方法を用いて、集合演算を行うものが多い^[20]。鋳鍛造に現れる複雑な曲面形状を、ソリッドモデラでは効率よく扱えない^[33]といわれている。

自由曲面を扱うと次の問題が生じる。

- 自由曲面を対象とする図形処理は、解析曲面を対象とする図形処理に比べ、計算時間を多く要し、解法が不安定になりやすい。たとえば、自由曲面間の交線計算の解を安定的に得ることは困難である^[28]。
- 自由曲面間の交線は、両方の曲面に誤差なく乗るのではなく、許容誤差の範囲内で離れる^[16]。フェイスの幾何形状が平面だけのソリッドモデラや、エッジからフェイスの幾何形状を内挿するソリッドモデラ^[17]では、計算機の丸め誤差を無視すれば、エッジとフェイスの離れはない。しかし、フェイスのベース曲面が自由曲面であると、許容範囲内のエッジとフェイスの離れや、エッジとパーテックスの離れ等、形状内に誤差(曖昧さ)を持つ。そのため、さまざまな演算が不安定になる。不安定になる例を図14に示す。エッジとフェイスの離れを d とし、フェイスと角度 θ で交わる平面が存在する時に、フェイスと平面の交線と、エッジと平面の交点との間のフェイス沿いの離れ l は、 $l=d/\tan(\theta)$ になる。エッジとフェイスの離れの誤差が許容範囲内であっても、交差角度によっては交線と交点が許容範囲以上に離れることがある。たとえば、 $d=.001$, $\theta=.1^\circ$

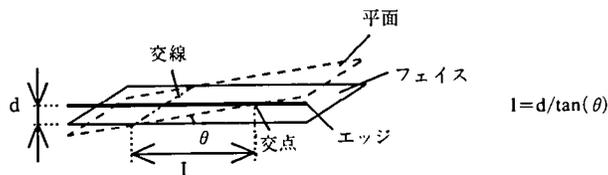


図 14 形状の持つ誤差

Fig. 14 Tolerance in a model

だと、0.57 離れる。この形状に含まれる誤差は、形状生成を進めると累積する。集合演算では、交線計算等の図形処理が多く繰り返されるため、計算時間が多く必要となり、対話的な設計環境に対応できる応答時間は得られていない^[17]といわれている。また、繰り返される図形処理の中で、1回でも図形処理が正常に終了しなければ、集合演算全体がエラーになる。そのため、自由曲面を形状に含む立体間の集合演算には、図形処理の頑健性が求められる。ソリッドモデラでは、図形処理の頑健性や効率の良さが、サーフェスマデラ以上に重要である。

[実 現]

1) 分割結合演算の概念……境界表現によるソリッドモデラの立体間集合演算は、二つの立体を交線で分割し、分割された立体片を、和・差・積の指示に従い選択結合するという手順で実行される。この手順の処理を、立体間だけでなく、複合曲面と立体間、複合曲面間にも適用する。F モデラは複合曲面と立体を区別せず、ともにシェルとして扱うので、演算対象が立体(閉シェル)であるか複合曲面(開シェル)であるかに関わりなく、要求される演算をシェル間の交線による分割・結合の演算として考えることができる。この演算を分割結合演算と名付ける。

すでに述べたように、集合演算は点集合の空間で定義される演算である。しかし、境界表現のソリッドモデラは、形状データを点集合ではなく、図形要素と図形要素間の隣接関係により表現する。そのため、集合演算の定義される空間と、形状データの定義される空間が異なる。一方、分割結合演算は境界表現のモデリング空間で定義される演算であり、演算と形状データの定義される空間が一致する。したがって、境界表現の形状モデラにとっては、集合演算より分割結合演算の方が、データ表現に、より即した演算であると考ええる。

2) 分割結合演算の処理手順…分割結合演算の処理手順を以下に示す(図 15)。

- ① シェル間の交線を求める。
- ② 交線をエッジ(列)としてシェルに付加する。
- ③ 付加エッジを境界としてシェルを分割する。
- ④ 分割後のシェルの中から必要なシェルを選ぶ。

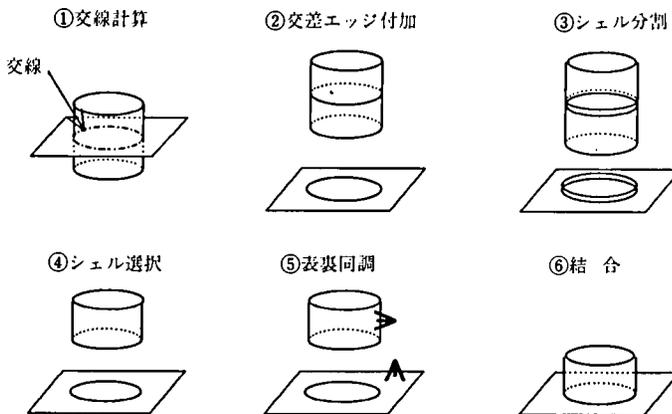


図 15 分割結合演算の処理手順

Fig. 15 Algorithm of a divide-and-join operation

- ⑤ 選択されたシェルの表裏を同調させる（向きを合わせる）。
- ⑥ 選択されたシェルを結合する。

分割結合演算の機能仕様を決める上で、④の分割後のシェルの選択方法が問題になる。立体間集合演算では和・差・積の指示に従い、分割後のシェルが相手立体の内部に位置するか、外部に位置するかを判定し、必要なシェルを選択するが、分割結合演算では、分割後のシェルの中から残すべきシェルを操作者が指示することにより選択する。

シェルの幾何形状が計算に関わるのは①の交線計算であり、②～⑥は位相を変更する処理である。ただし、③のエッジ付加では、エッジの分割、内部ループの帰属判定等に幾何情報を使う。交線計算では、単に交線の幾何形状を求めるだけでなく、交線の乗るフェイス、エッジとの関係も求める。一般に、ソリッドモデラにおける形状を扱うコマンドの不安定さは、位相処理ではなく幾何の計算に起因する。これは、分割結合演算の問題点として挙げた形状の持つ誤差や、図形処理の解法の誤差、計算機の丸め誤差のためである。①で、交線の各部分をエッジとしてどのフェイスに付加すべきかが決まれば、②以降は位相処理を優先し、位相的に矛盾のない形状を作ることができる。このように、幾何処理と位相処理を分離することにより、アルゴリズムを簡潔にしている。

他の形状モデラの集合演算でも、F モデラと同様の処理手順を採用の多い。モデリング空間が多様体である形状モデラは、F モデラのように、シェルを分割した後で結合する^[31]が、モデリング空間が非多様体である形状モデラの中には、シェルを結合して一旦非多様体を作り、その後に分割を行う^{[24][29]}ものもある。

また、F モデラで、形状の一部だけを分割結合演算の対象とすることができる。図 16 は肉厚の薄い部品の上部にボスを取り付ける例である。(b)のように、和集合をとると下側にまでボスが付加される。これを避けるために、部品下側の面と交差しないように、ボスの下部をあらかじめ切除する必要がある。この操作は、操作者にとって煩わしい。F モデラでは、演算に使用する交線を操作者が選択できるようにし、(a)のように上側にだけボスを取り付けることもできる。なお、点集合空間の演算である集合演算でこの機能の実現を考えると、上側にだけ和集合をとるといった概念の説明が困難である。

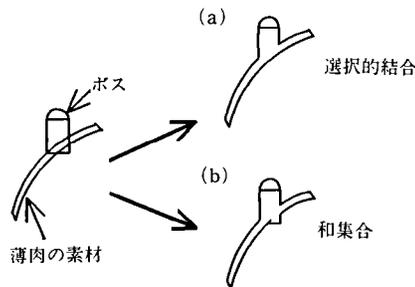


図 16 選択的結合

Fig. 16 Selective join

次に分割結合演算の制限について述べる。F モデラの外部モデリング空間は、向き付け可能な二次元多様体である。分割結合演算の結果、モデリング空間の範囲外の形状を生成する演算は実行できない。たとえば、図 17(a)は演算結果の形状が非多様体になり、(b)はメビウスの帯のような向き付け不能なシェルを作るため実行できない¹⁹⁾。(a)は交線計算処理で、(b)はシェルの表裏同調処理で、F モデラが誤りを検出する。)

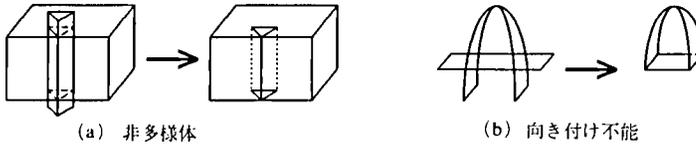


図 17 分割結合演算の制限

Fig. 17 Restriction of a divide-and-join operation

また、複合曲面間の分割結合演算では、交線が複合曲面の途中で止まることがある(図 18(a))。このままでは、交線を使って複合曲面を分割することができないので、交線を複合曲面端まで延長する。交線の延長方法にはいくつかの方法が考えられるが、F モデラでは図 18(b)のように、不足している複合曲面を交線端点における接平面で補い複合曲面端までの交線を求めている。

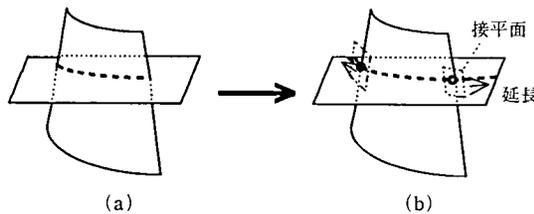


図 18 交線の延長

Fig. 18 Extension of an intersectional curve

- 3) 高速な集合演算……F モデラの開発にあたり、従来使用していた図形処理ライブラリを全面的に改訂し、頑健性と高速性の向上に努めた。また、データ構造を単純にし、基本位相演算子に実用的なものを用意する¹⁸⁾ことにより、位相操作の効率化に努めた。
- 4) 頑健な集合演算……すでに述べたように、集合演算の内部処理では、幾何処理と位相処理を分離している。シェルを分割する交線が得られれば、位相主導で形状を変更することにより、安定的に位相変更を行う。

シェル間の交線列を求める場合、フェイスやエッジが接する部分や一致する部分で、とくに計算が不安定になりやすい。そのため二つのフェイスの交線を求める時に、フェイスとフェイスの一致、エッジとエッジの一致、エッジがフェイス上に乗る等の特殊な位置関係を自動的に検出し、それぞれの場合に適する処理を行うことで交線計算の安定性を高めている。それでも交線列が計算できない場合、計算可能な交線を生成し、操作者が手作業で分割結合演算と同一の形状生成を行

える回避機能も用意している。

4.2 フィレット掛け

〔要求〕 フィレット掛けは、形状生成の全工程の中で最も煩雑な作業であり、工数面でも形状生成作業の総時間の半分以上を占める^[5]。鋳鍛造金型形状は起状が激しく複雑である。このような形状を対象として、多くのフェイスにフィレットを一度に掛けるため、鋳鍛造金型用の形状モデラには、他の適用分野以上にフィレット掛けの安定性と高速性が要求される。

1) フィレット掛けの操作手順……F モデラでは、フィレット掛けを次の二つの独立な操作に分けた。(もちろん一度の操作で①②を実行することも要求される。)

① フィレット曲面の生成

② フィレット曲面の埋め込み(フィレット曲面と、フィレットの掛かるフェイスのトリムと隣接関係付け)

フィレット掛けを、二つの操作に分ける理由を説明する。

フィレット曲面を埋め込むと、隣接関係のためにフィレット曲面は周りのエッジ、フェイスから拘束を受ける。そのためフィレット曲面に自己交差や歪がある場合には、埋め込み後に整形するより、埋め込み前に整形の方が容易である。フィレット曲面の形状を、操作者が目視確認し、必要なら整形を行い、その後埋め込みたいという要求がある。

また、フィレット曲面を自動的に埋め込めない場合、複数の操作(複合曲面や立体の一部を切り取る、フィレット曲面と貼り合わせる等)を組み合わせて、フィレット曲面を埋め込みたいという要求もある。このような操作は、不要になることが望ましい。しかし現在の図形処理技術は操作者の意図するフィレットの曲面形状が確実に得られ、確実に埋め込める段階には達していない。フィレットが掛からないために、形状生成作業が進まないという事態は避けたい。操作工数は増えるが、操作者の介入による手作業で、形状生成作業を進める方がよい。フィレット掛けの信頼性を高めるとともに、操作者の意図する形状が得られない場合に、フィレット曲面を整形する機能と、切り貼り作業用の機能を用意することも、実用システムには必要である。

2) フィレット曲面生成への要求……鋳鍛造金型形状の生成では、次の条件でもフィレットを掛けることが要求される(図19)。

① 複数のフェイスにわたるフィレットが掛けられる。

② 案内エッジ列(丸めるエッジ列)に隣接しないフェイスを通過するフィレ

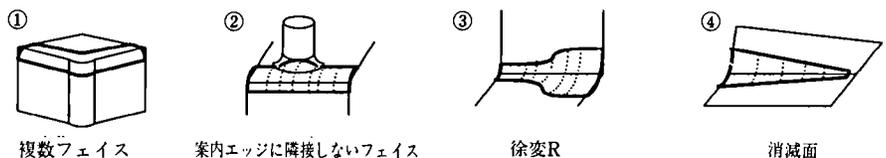


図 19 さまざまなフィレット掛け条件

Fig. 19 Various condition in filletting

トが掛けられる。

- ③ フィレットの径を徐変できる。
- ④ フィレットの通過する面がしだいに平行に近くなると、フィレット曲面の幅が狭くなり、ついには消滅する。このようなフィレットが掛けられる。

前述のように、鋳鍛造金型形状の生成では、製造性、部品剛性の保持、木型との互換性等の理由により、フィレットを掛ける順序が重要なことがある。フィレット曲面が重なる部分では、フィレット掛けの順序に関係なく、フィレット曲面をブレンドした曲面を作る方法^[17]もある。しかしこのような方法ではなく、先に掛けたフィレット曲面の上に、あとのフィレットを掛けるように形状を生成する必要がある。

図形精度の観点から考えると、鋳鍛造金型のフィレット曲面にとって、面形状の意匠性や、指定半径値を正確に保つことは要求度が低い。しかしフィレット曲面端の接触部が、接続フェイスと許容誤差以内で一致することは、フィレット埋め込みやその後の形状生成のために強く要求される。また、接触部は接平面連続であることが要求される。接平面連続でないと、中子形状から主型形状を（中空部表面の形状から外側表面の形状を）オフセットして作る時に、フィレットと周りのフェイスとが離れる。

[問題点] 従来のソリッドモデラのフィレット掛けは、機能が貧弱で、しかも処理が不安定である。実際には、フィレット掛けのできる形状はごく単純なものに限られ、ソリッドモデラで生成した形状でも、フィレット掛けはサーフェスモデラで行うこともある^[5]。

- 1) フィレット曲面生成における問題点……従来の形状モデラでのフィレット曲面生成の主な問題点を挙げる。
 - ① 計算時間を多く要する。
 - ② フィレット掛けの条件の入力操作が煩雑である。
 - ③ フィレットの掛かるフェイスの曲率半径が、フィレットの半径より小さい場合、フィレット曲面の形状に不具合が発生する。不具合の種類は、フィレット曲面を生成する方法により異なる。球回転法^[30]（ローリングボール法：二つのフェイスに接する球を求め、この球を案内エッジに沿って移動する時にできる球の包絡面をフィレット曲面とする方法）では、曲率半径の小さな部分に球が入らないため、フィレット曲面が対象フェイスから浮く。また、円弧移動法^{[18][22]}（案内エッジ列の法平面内で、二つのフェイスの間に円弧を掛ける。法平面を案内エッジ列に沿い移動させ、各々の位置で円弧を作り、円弧を繋いでフィレット曲面とする方法）では、フィレット曲面が自己交差することがある。
- 2) フィレット曲面埋め込み処理の問題点……フィレット曲面を生成する図形処理技術は、サーフェスモデラでも研究され^[22]、1)の問題はあるものの実用レベルに達している。しかし、フィレット曲面をもとのシェルに自動的に埋め込む処理は、フィレット曲面生成処理ほどには研究されていない。フィレット曲面埋め込み処理は以下の問題を持つ。
 - フィレット曲面埋め込みの位相変更アルゴリズムが定式化されていない。発

表されているアルゴリズムには、中間状態が非多様体になる^[23]、位相処理と図形処理が分離されていない^[21]等の問題がある。(中間体が非多様体であることは、必ずしも問題ではないが、データ構造に制約を与えることになる。)

- フィレットの埋め込み処理を、サーフェスマデラの場合と比較する。サーフェスマデラでは、フィレット曲面が操作者の意図する形状と異なる場合でも、フィレットの曲面形状を修正した後に周りの面をトリムし、埋め込むことができる。しかしソリッドモデラは一般に曲面を扱わないので、フィレット曲面の形状を修正することが困難である。
- 案内エッジ列端のパーテックスの回りのフェイス列の中で、案内エッジに隣接しないフェイス列のことを、停止フェイスと呼ぶ。通常フィレット曲面と停止フェイスは、交差し、交線によりそれぞれトリムされる。トリムすることにより停止フェイスの面積が、フィレットを掛ける前より大きくなることもある(図20)。この場合は、特別な考慮が必要である。停止フェイスのベース曲面が十分に大きいと、位相を変更するだけでよい(図20(a))。ベース曲面が十分には大きくないと、曲面を拡張する必要がある(図20(b))。さらに、拡張すべき部分に複数の停止フェイスが存在すると、拡張方法が複雑になる(図20(c))。

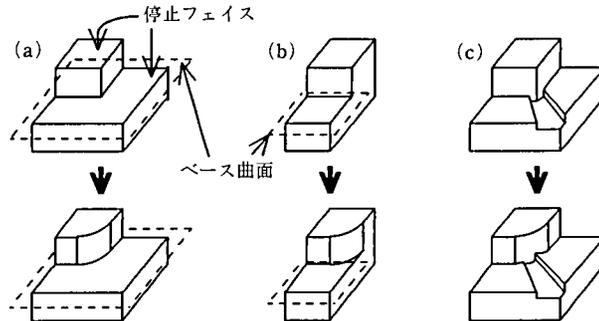


図 20 停止フェイスの面積が増加する例

Fig. 20 Examples of filleting as increasing a region of stopping faces

[実 現] 立体にフィレットを掛ける操作を、エッジを丸めるという局所形状変更とは考えず、フェイスを滑らかに接続する曲面を生成し、その曲面を立体に埋め込む処理であると考える。

フィレット掛けの操作をフィレット曲面生成とフィレット埋め込みという2段階の操作に分け、フィレット曲面生成直後の状態を、演算途中の一時的な中間状態ではなく、安定な状態として操作者に解放する。このことにより、フィレット掛け操作の自由度を増し、形状生成を柔軟なものにした。立体のみを扱うソリッドモデラでは、フィレット曲面を独立に取り扱うことはできない。このような操作を可能にするのは、曲面と立体を同等に扱うデータ構造によるものである。

- 1) フィレット曲面生成方法……開発したフィレット曲面生成演算は、図19のさまざまなフィレットの掛け方を実現した。また、前述の問題点を以下のように解決

した。

- ① 高速な処理：フェイス間の隣接関係やフェイスの向き付けを、処理の効率化に利用した。また、従来使用していた図形処理ライブラリを、効率と頑健性の観点から見直し、再作成した。
- ② 操作工数削減：図形要素間の隣接関係を有効に利用して、操作者の入力回数を削減した。
- ③ 自己交差の除去：鋳鍛造金型形状の生成では、通常接線連続なエッジ列を案内エッジ列としてフィレットを掛ける。すなわち、案内エッジ列が折れるパーティクスを経由して、フィレットを一度に掛けるような操作は行わない。そのため、案内エッジ列が接線不連続なことによるフィレット曲面の自己交差は発生しない。しかし、フィレットの掛かるフェイスの曲率半径や案内エッジ列の曲率半径が小さいこと等により、フィレット曲面に自己交差が生じることがある。この場合、システムが自動的に自己交差を検出し除去する。

円弧移動法により生成されたフィレット曲面に歪みが生じる場合には、歪みのない曲面を生成するように、案内エッジ列の幾何形状を変えることが求められる。しかし、案内エッジ列は周りのフェイスから拘束されていて、自由に形状を変えることができない。そのため、案内エッジ列の（どのエッジにフィレットを掛けるかという）位相情報だけを使い、（法平面を決める曲線としての）幾何情報は使用せず、操作者の指示する別の曲線の幾何形状を使用することもできるようにした。

2) フィレット曲面埋め込み方法……フィレット曲面埋め込みは、フィレットのシェルと、フィレットの掛けられるシェルとの、交差・分割・結合の演算である。フィレット埋め込みの処理手順を以下に示す（図 21）。

- ① フィレット曲面と停止面との交線を求める。
- ② ①の交線をエッジとしてフィレット・シェルに付加する。
- ③ ②のエッジ列を境界として、フィレット・シェルの端部を除去する。
- ④ ③のフィレット・シェルの外周エッジ列を、フィレットの掛けられるシェルに付加する。

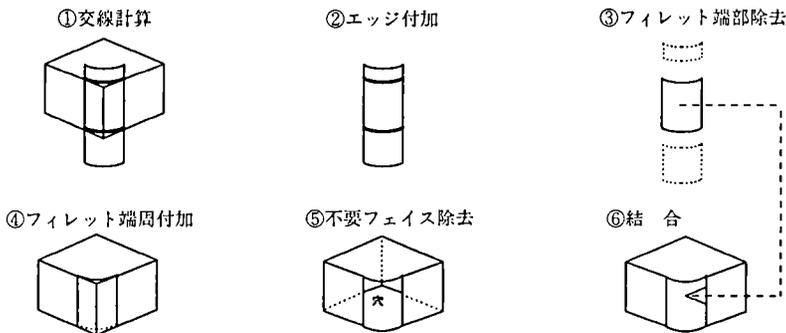


図 21 フィレット曲面埋め込み方法

Fig. 21 Algorithm of embedding a fillet surface

- ⑤ ④のエッジ列を境界として、案内エッジ列を含む側のフェイス群を取り除く。
(フェイス群が一つのダミーフェイスに置き換わる。)
- ⑥ ③と⑤のシェルを結合する。

この処理手順は分割結合演算と似ている。分割結合演算と同様、幾何の処理は①だけであり、②～⑥は位相変更の処理である。幾何処理と位相処理を分離することにより、アルゴリズムを明快にし、位相変更を安定的に処理している。

なお、フィレットを付加することにより、停止フェイスが大きくなる場合(図20)には、自動的に停止フェイスと停止フェイスのベース曲面を拡張する。しかし、自動延長処理には限界がある。たとえば図20(c)より複雑な形状になると、延長することにより停止フェイス群が自己交差することもある。また図22の太線のエッジにフィレットを掛ける場合、(a)のように停止面を自動延長するのではなく、(b)のように他の面を補い形状生成したいこともある。そのため、拡張部分の停止フェイスを別に指示する方法も用意した。

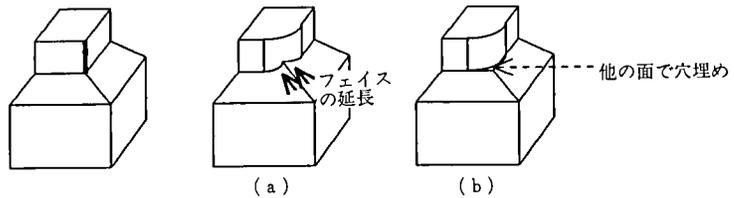


図 22 フィレット停止面の隙間の埋め方
Fig. 22 Embedding a hole in stopping faces

4.3 形状復元

[要求] 鋳鍛造金型形状の生成では、主として分割結合演算とフィレット掛けを繰り返して形状を作る。また形状生成後、設計変更や類似形状の生成が行われる。そのため、形状生成過程の任意の分割結合演算やフィレット掛けを対象に、その演算により変更された形状部分だけを、形状生成の演算順序に依存せず、演算前の形状に戻すことが要求される。この要求を実現する演算を、形状復元と名付ける。

分割結合演算とフィレット掛けにより作られた形状を復元する例をそれぞれ示す。

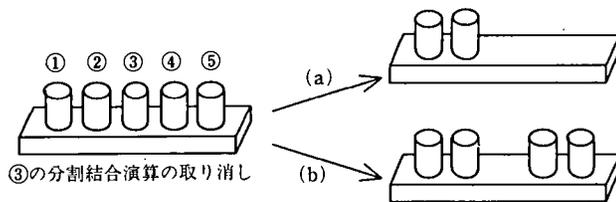


図 23 分割結合演算により生成された形状の復元
Fig. 23 Restoration of the shape created by divide-and-join operations

- 1) 分割結合演算により変更された形状の復元 (図 23)……角柱に五つの円柱を①～⑤の順に分割結合演算したあと、③の円柱のみを取り除く場合を考える。要求される演算は、図 23(a)のように③の演算を実行する直前の形状に戻すのではなく、図 23(b)のように③の演算により変更された部分のみを戻す演算である。
- 2) フィレット掛けにより変更された形状の復元 (図 24)……角柱に①～⑤の順序で、①～④は角柱側面に、⑤は角柱上面一周にフィレットを掛ける。ここで、①のフィレットを復元する場合を考える。図 24(a)のように、①のフィレットを掛ける直前の状態に戻すのは要求に反する。分割結合演算と同様に考え、図 24(b)のように①だけを戻すと、①の復元形状と⑤のフィレットの間に隙間があき、その後の形状操作が困難になる。鋳鍛造金型形状は複雑なため、図 24(b)のような異常な形状が生成されても操作者が気付かない可能性もある。要求されるのは、図 24(c)のように、①とそこに掛かる⑤を戻す処理である。⑤も戻すのは、⑤のフィレット曲面形状が①のフィレット曲面形状に依存しており、①の演算が取り消されれば⑤の演算も無効であると考えられるからである。

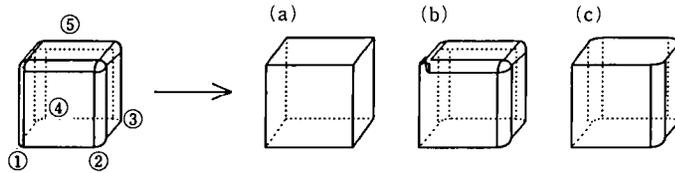


図 24 フィレット掛けにより変更された形状の復元

Fig. 24 Restoration of the shape created by filleting operations

このように、形状復元には、復元対象の演算を無効にするだけでなく、その演算が生成した図形を参照する演算も無効にすることが要求される。このことを、形状復元の局所的順序依存性と呼ぶ。

図 23 の分割結合演算だけの形状復元の例では、局所的順序依存性の要求は出てこない。フィレット掛けは一つのシェルの中の局所的な形状変形であるが、分割結合演算は独立な二つのシェルの間の演算だからである。しかし、実際の形状生成過程では、分割結合演算とフィレット掛けが混存するため、分割結合演算の形状復元でも局所的順序依存性を考慮する必要がある。たとえば、図 25 のように、角柱に円柱の穴を分割結合演算であけた後、その交差部分にフィレットを掛けた

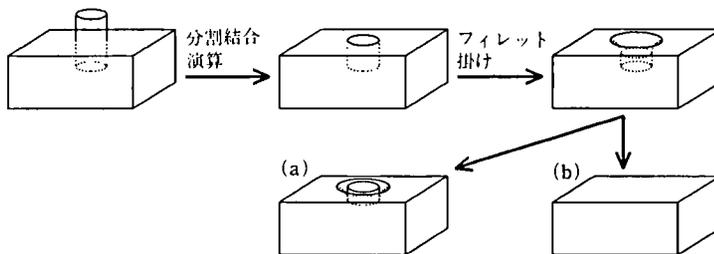


図 25 分割結合演算とフィレット掛けを組み合わせた形状復元

Fig. 25 Restoration for the shape created by a divide-and-join operation and filleting

形状に対し、分割結合演算だけを取り消すと図 25(a)のように操作者の意図しない形状になる。フィレット曲面の形状は、分割結合演算により付加された円柱部分の形状に依存するため、図 25(b)のようにフィレットも取り除く必要がある。

[問題点] 従来の形状モデラは、設計者の意図や図面の記述内容に従い、形状を計算機内に生成することを目的としていた。そのため、設計変更等に対し、形状を復元しにくい。形状の一部を復元、変更する方法として、従来の形状モデラで採られているいくつかの方法を述べ、それらの問題点を検討する。

1) 操作者の手作業により形状の一部を作り直す。

形状を復元、変更したい領域のフェイス群を取り除き、取り除いた部分に、別途生成した形状をはめ込む方法である(図 26)。従来この方法が、実務で一般的に使われていた^[21]。

しかし、この方法は操作者が部分的に形状生成作業を行うため作業工数が多くなる。たとえば、図 23 で⑤のフィレットが掛からない形状に、手作業で復元する場合は、角柱から作り直した方が操作工数が少なくてすむ。

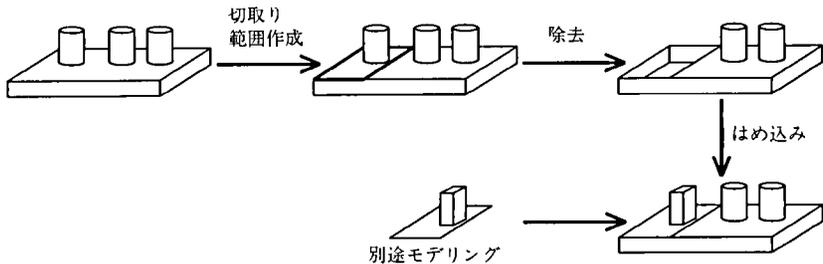


図 26 操作者による修正作業
Fig. 26 Modification by user

2) 任意の時点の形状に戻す。

入力操作・コマンド実行の記録や、データベースの保存により、過去のデータベースの状態を再現することができる。再現するための 3 種類の方法について述べる。

- ① 過去のデータベースを保存すること(チェックポイントファイル)により、記録した時点のデータベースの状態を再現する。
- ② 入力操作やコマンド実行の記録を保存し、再実行する。
- ③ UNDO(1コマンドごとの実行取り消し)により、1コマンドずつ順に過去の状態に戻る。この方法には、データベースの更新前の状態を保存し(ピフオアルック)、データベースの状態を順に戻す方法と、オイラ演算子の逆操作により戻す方法がある。

これら 3 種類の方法には、共通して次の問題がある。

- 戻したい時点を探す作業が煩雑である。操作者が形状生成の順序を記憶しているとは期待できない。とくに、形状生成期間の長い大物部品は、複数の操作者により形状生成することがある。そのため、特定の演算を実行した時点を探すのは困難である。

- 戻す対象の演算より時間的に後に実行されたすべての演算が無効となるため、復元したい部分以外の形状も図 23(a)や図 24(a)のように戻ってしまう。また戻す対象の図形だけでなく、無関係な図形までその時点の状態に戻る。この復元方法は、要求される形状復元とは異なる。この状態から、入力操作の記録を再実行する等の方法により、図 23(b)や図 24(c)等の要求する形状を生成することも考えられる。しかし、フェイス識別子やエッジ識別子等の位相に関わる図形要素識別子が入力操作の記録に含まれていると、はじめの実行と再実行で対象図形の位相が異なり、再実行できないことがある^[21]。

さらに、①の方法はチェックポイントを採取した状態にしか戻ることができない。②の方法は再実行のための演算時間が掛かる。③の方法は近い過去の状態に戻るのに適しているが、遠い過去の状態に戻るのには適していない。数コマンド前の状態に戻るのが、実用上の限界であると思う。少なくとも UNDO を繰り返して、数日分の作業を遡ることを実務で行うとは思えない。とくにオイラ演算子の逆操作は、形状を戻す単位が小さいため実用的でない。

3) 演算により消滅する形状を保存し、任意の演算を取り消す。

増田らは集合演算で消滅する形状をモデルの中に保存することにより、実行順序によらない集合演算の取り消しや、和集合を積集合に変える等の演算の変更を実現している^[32]。この方法では、集合演算が行われるごとに空間を細分し、細分された空間の中から、和・差・積の指示に従い必要な空間を取り出す。そのため、特定の集合演算を無効にすることや、集合和を集合積に変更すること等は容易にできる。

しかし、この方法には次の問題がある。

- 集合演算により消滅した部分の形状も、その後の集合演算の交線計算の対象となる。そのためデータ量と計算時間が多い。
- 図 25 の例で、この方法の演算取り消しを行うと、(a)の形状を生成する。これは局所的順序依存性の要求と異なる。

その他に、形状フィーチャの取り付けにより消滅する形状を保存しておき、形状フィーチャを取り除く(サプレス)際の、形状の復元に利用する方法も提案されている^[35]。

[実現]

- 1) デッドシェル……まず、デッドシェルを作る処理について説明する。すでに述べたように、分割結合演算やフィレット埋め込みは、二つのシェルを、それぞれ

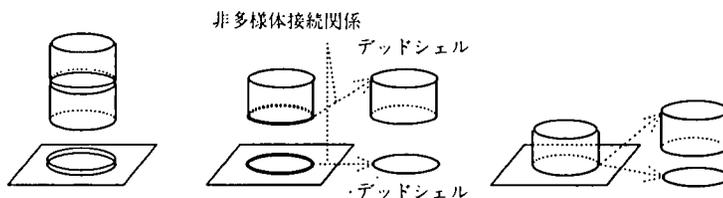


図 27 分割結合演算のデッドシェル作成手順

Fig. 27 Algorithm of the creating dead shells in a divide-and-join operation

エッジ列で分割し、分割後のシェルの中から必要なシェルを選び、結合する演算である。分割後選択されなかったシェルを削除せず、演算前の形状に戻すために、デッドシェルという形で保存する。分割結合演算では、演算対象の二つのボディの不要部分をそれぞれデッドシェルにする (図 27)。

フィレット埋め込みでは、埋め込まれるシェルのフィレット掛けにより消滅する部分が、一つのデッドシェルになる (図 28)。

次に、シェル間の非多様体接続関係について説明する。分割により二つに別れたエッジの間に、非多様体接続関係を持つ。一つのエッジはフェイスへの接続関係と同様に、非多様体接続関係を左右一つずつ持つことができる (図 29)。一つのエッジから三つ以上のデッドシェルが繋がる場合でも、図 30 のように、二つのデッドシェルのエッジの間に非多様体接続関係を持つことにより、エッジごとに二つだけの非多様体接続関係で、一つのエッジに関わる三つ以上のシェルの接続関係を表現する。さらに、この表現方法は、一つのエッジに関わるシェルの時間的な結合順序も表現する。たとえば、図 30 の太線のエッジ e には、A, B, C, D の

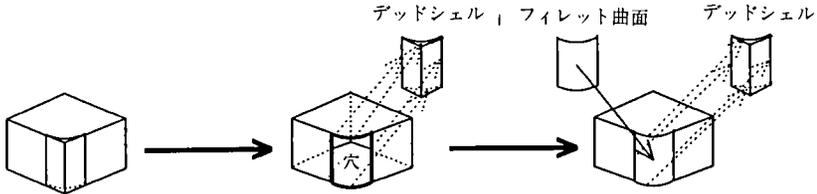


図 28 フィレット曲面埋め込みのデッドシェル作成手順

Fig. 28 Algorithm of creating a dead shell in embedding a fillet surface

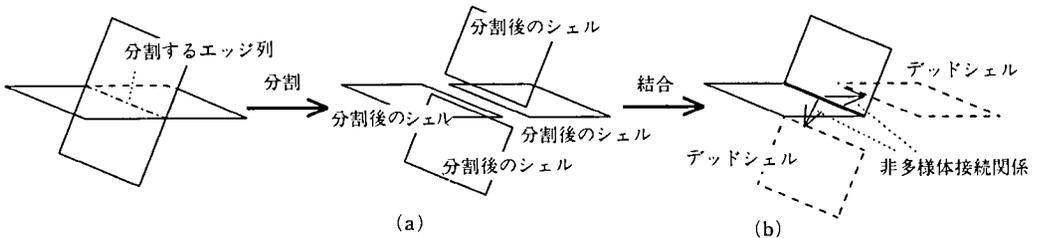


図 29 非多様体接続関係

Fig. 29 The relationship of a non-manifold

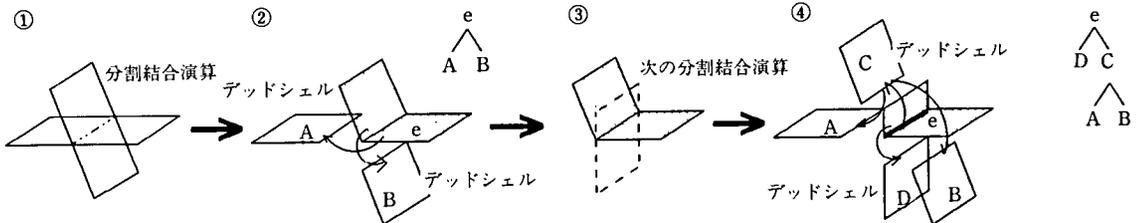


図 30 三つ以上のデッドシェルが繋がるエッジ

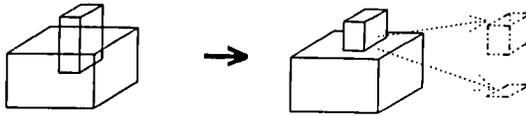
Fig. 30 Edge connecting with more than two dead shells

四つのデッドシェルが接続する。このエッジ e と直接接続するのは、③の演算で作られた C, D のシェルである。C のシェルの e に対応するエッジは、①の演算で作られた A, B のシェルと接続する。一つのエッジに関わる非多様体接続関係の階層を調べれば、そのエッジと接続するデッドシェルの作られた時間的順序がわかる。この順序情報を、形状復元の局所的順序依存性の実現に使用する。

デッドシェルは、操作者の行う通常の形状生成演算では参照、変形されず、形状復元の時だけ使用される。デッドシェルは生成された状態のまま保存され、その後の演算により変形されることはない。たとえば分割結合演算実行時に、演算対象の形状の持つデッドシェルと、演算相手の形状とが干渉していても、デッドシェルは交差計算の対象にならない。そのためデータ量が膨大になることはない。F モデラでは、形状復元に局所的順序依存の制約を加えているため、デッドシェルを変形させる必要がない。

2) 形状復元方法……復元すべきデッドシェルが指示されると、以下の処理手順で形状を復元する (図 31)。

[分割結合演算]



[復元]

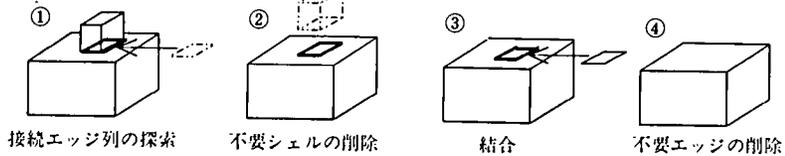


図 31 形状復元演算の処理手順

Fig.31 Algorithm of a restoration operation

- ① デッドシェルに接続するエッジ列を捜す。
 - ② ①のエッジ列を境界としてライブシェルを分割し、指示デッドシェルと接続するフェイスを含む側を取り出す。
 - ③ ②のシェルと、指示されたデッドシェルとを結合する。
 - ④ 結合部のエッジの中で、二つの隣接フェイスの曲面幾何実体が同一であるエッジ (分割結合演算の交線エッジ、フィレット接触部のエッジ等) を削除する。
- 形状復元が演算として成立するためには、デッドシェルの外周エッジ列と、それに接続するライブシェルのエッジ列が対応している必要がある。たとえば、

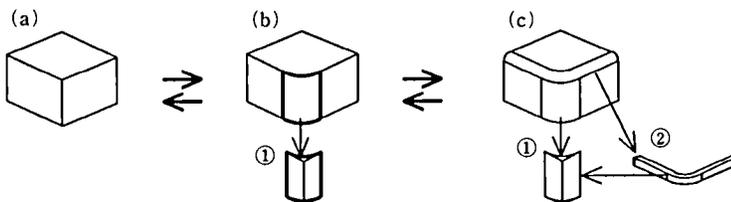


図 32 実行順序に依存する形状復元の例

Fig.32 Restoration depending on the operation sequence

図 32(c)で②のデッドシェルは、接続するエッジがすべてライブシェル中にあるため復元できる。しかし、①のデッドシェルは、接続するエッジがライブシェルと②のデッドシェルに別れる。そのため、②のデッドシェルを復元した後でなければ、①のデッドシェルを復元できない。このように、F モデラの形状復元方法は局所的順序依存性の要求を満たす。

5. 成 果

F モデラ開発の成果をまとめる。

- 1) 鋳鍛造金型形状と形状生成手順を分析し、鋳鍛造金型用の形状モデラに対する要求をまとめた。
- 2) 1)の要求を満たす F モデラを設計し、計算機システムとして実現した。
- 3) 以下のように形状モデラのいくつかの先端的技術課題に取り組み、その解決方法を考案し、実用システムの中で実現した。
 - ① サーフェスマデラとソリッドモデラを統合する（曲面と立体を同一のデータ構造で表現し、曲面と立体を統一的に扱う）新しい形状モデラを実現した。
 - ② 形状復元のために、演算により消滅する形状を非多様体表現を利用して、モデル中に保存する方法を考案し実現した。
 - ③ 曲面と立体の間の集合演算に相当する分割結合演算の概念を考案し実現した。
 - ④ フィレット埋め込み処理を定式化し、実現した。
 - ⑤ 実行順序によらず、分割結合演算、フィレット掛けにより変化した形状を復元する形状復元の演算の概念を考案し実現した。
 - ⑥ 分割結合演算、フィレット掛け、抜き勾配掛けという難易度の高い図形処理を、高速で頑健に処理した。

6. お わ り に

F モデラは、平成 4 年度の型技術者会議技術賞を授賞した^[11]。また、某自動車会社において、実業務に適用され顕著な効果を発揮していると報告されている^{[2][10][11][25]}。

鋳鍛造金型形状の生成に特化した形状モデラの開発を通して、今後の形状モデラ開発のために有益な情報を記録することを目的として、本稿を作成した。今後のシステム開発の参考になれば幸いである。

-
- 参考文献 [1] 相沢民王, “CAE データベースのためのデータの胞複体鎖モデル, 精密工学会誌, Vol. 55, No. 2, 1989, pp. 311~316.
- [2] 江口誠治, “曲面立体共存モデラの鋳造型への適用” 鋳物, Vol. 62, No. 12, 1990, pp. 1037~1040.
- [3] 倉賀野哲造他, “高品位形状の射出成形品に対するパーティング面の設計とその工具経路生成”, 精密工学会誌, Vol. 57, No. 7, 1990, pp. 1256~1261.
- [4] 桑原利幸他, “自由曲面を持つ金型の CAD/CAM 複雑形状を高精度に設計, 加工”, 日経メカニカル別冊 CAD/CAM/CAE, 1985, pp. 158~166.
- [5] 後藤玲爾, “3次元 CAD によるモデリングを体験してみる”, 日経 CG, No. 52, 1991,

- pp. 154~160.
- [6] 近藤幹夫, “ソリッドモデルを用いた金型加工システム”, 第 6 回自動化工学講演会講演論文集, 1989, pp. 208~210.
 - [7] 鈴木建彦, 谷本茂樹, “ソリッドモデラの問題点と「曲面立体共存モデラ」の開発”, 精密工学会誌, Vol. 56, No. 11, 1990, pp. 2015~2020.
 - [8] 鈴木建彦, 松林毅他, “曲面立体共存モデルの位相数学モデルと「集合演算」(第 1 報) — 胞複体とセルオペレータ”, 精密工学会誌, Vol. 57, No. 8, 1991, pp. 1393~1398.
 - [9] 鈴木建彦, 谷本茂樹, “曲面立体共存モデルの位相数学モデルと「集合演算」(第 2 報)”, 精密工学会誌, Vol. 58, No. 7, 1992, pp. 1245~1250.
 - [10] 鈴木建彦他, “鋳鍛造金型用「曲面立体共存モデラ」の開発と実用化”, 日産技報, No. 26, 1989, pp. 205~213.
 - [11] 鈴木建彦他, “鋳鍛造金型用曲面立体共存モデラの開発と実用化”, 型技術者会議 '92 講演論文集, 1992, pp. 22~25.
 - [12] 鈴木建彦他, “階層型非多様体モデルによる形状復元機能の実現”, 1992 年度精密工学会秋季大会学術講演会講演論文集, 1992, pp. 445~446.
 - [13] 鈴木建彦, “胞複体モデルにおける曲面立体共存モデラの理論と実用化に関する研究”, 東京大学工学部博士論文, 1990.
 - [14] 末永浩章他, “ユニット試作鋳造部品への CAD/CAM 適用の取り組み”, 日産技報, No. 28, 1990, pp. 176~181.
 - [15] 高橋昭八郎, “マツダの CAD/CAM”, 工業調査会, 1985.
 - [16] 谷本茂樹, “サーフェスモデラとソリッドモデラの統合”, ユニシス技報, No. 32, 1992, pp. 179~201.
 - [17] 千代倉弘明, “ソリッドモデリング”, 工業調査会, 1989.
 - [18] 津田順司, 上西博文, 菊地純男, “3 次元図形処理システム MDM-I の開発”, 情報処理学会コンピュータ・グラフィクス研究会資料, Vol. 5, No. 3, 1982, pp. 1~9.
 - [19] 東畑透他, “鋳造・鍛造における CAD/CAM の現状”, 日産技報, No. 23, 1988, pp. 101~105.
 - [20] 烏谷浩志, “自由曲面のモデリングの実際”, 日経 CG, No. 56, 1991, pp. 151~160.
 - [21] 烏谷浩志, 千代倉弘明, “3 次元 CAD の基礎と応用”, 共立出版, 1991.
 - [22] 橋本可輝, “複数の曲面を接続するフィレット曲面創成法”, ユニシス技報, No. 11, 1986, pp. 1~26.
 - [23] 東正毅, 水野誠治, 彌富英樹, “非多様体サーフェスモデリングによるフィレット形状生成”, 精密工学会誌, Vol. 57, No. 11, 1991, pp. 1971~1976.
 - [24] 東正毅, 水谷好宏, 中倉清, 小川真智, “非多様体シェル演算による統一的形状モデリング(第 1 報) — 立体の集合演算と閉シェル抽出による立体化 —”, 1992 年度精密工学会秋季大会学術講演会講演論文集, 1992, pp. 443~444.
 - [25] 本多正人他, “曲面立体共存モデラの鍛造金型への適用”, 塑性加工学会講演会, 1989, pp. 567~570.
 - [26] 松林毅, “CAD/CAM システムにおける UIM の実現”, ユニシス技報, Vol. 10, No. 4, 1991, pp. 103~113.
 - [27] “「金型設計・製作特集号」”, 鋳物, Vol. 62, No. 12, 1990.
 - [28] R. P. Markot and R. L. Magedson, “Solutions of Tangential Surface and Curve Intersections”, CAD, Vol. 21, No. 7, 1989, pp. 421~429.
 - [29] E. L. Gursoz et al, “Boolean set operations on non-manifold boundary representation objects”, CAD, Vol. 23, No. 1, 1991, pp. 33~39.
 - [30] L. Holmstrom and T. Laako, “Rounding facility for solid modelling of mechanical parts”, CAD, Vol. 20, No. 10, 1988, pp. 605~614.
 - [31] M. Mantila, “An Introduction to Solid modeling”, Computer Science Press, 1988.
 - [32] H. Masuda et al, “A Mathematical Theory and Applications of Non-Manifold Geometric Modeling”, In Advanced Geometric Modeling for Engineering Applications, North Holland, 1990, pp. 78~92.
 - [33] J.R. Miller, “Sculptured Surfaces in Solid Models, Issues and Alternative Approaches”, IEEE CG & A, Vol. 6, No. 12, 1986, pp. 37~48.
 - [34] L. Piegel, “On NURBS: A Survey”, IEEE CG & A, Vol. 11, No. 1, 1991, pp. 55~71.
 - [35] M. J. Pratt, “A Hybrid Feature-based modelling system, In Advanced Geometric Modeling for Engineering Applications”, North-Holland, 1990, pp. 189~201.
 - [36] T. Satoh, H. Chiyokura, “Boolean Operations on Sets Using Surface Data,

Symposium on Solid Modeling Foundations and CAD/CAM applications", ACM press, 1991, pp. 119~128.

- [37] T. Suzuki et al, "CAD/CAM in the Automotive Foundry-Development and Applications of a New Type of Solid Modeler-", Transaction of the Japan Foundrymen's Society, Vol. 10, Oct, 1991.
- [38] T. Varady, "Rolling Ball Blends in Solid Modelling", Computer Applications in Production and Engineering, North-Holland, 1989, pp. 295~308.
- [39] K. Weiler, "Topological Structure for Geometric Modeling", PhD. Thesis, RPI, 1986.
- [40] P. Wilson, "Euler Formulas and Geometric Modeling", IEEE CG & A, Vol. 5, No. 8, 1985, pp. 24~36.

執筆者紹介 谷本 茂樹 (Shigeki Tanimoto)

昭和 27 年生. 52 年大阪大学理学研究科修士課程修了.
同年日本ユニシス(株)入社. 主に形状モデラを中心とする
CAD/CAM システムの開発に従事. 平成 3 年日産エア
ール・テクノロジー(株)に出向. 精密工学会会員. 情報処
理学会会員.



CAD のためのユーザ・インタフェースの提案

A Consideration of User Interfaces for CAD Systems

鈴木 良太郎

要 約 イベント駆動型 GUI, GUI ビルダー, ビジュアライゼーションソフト, CASE ツール, 状態遷移表エディタ, 等に見られるユーザ・インタフェースの最近の動向について分析すると共に, その根底をなすコミュニケーションのあり方について考察し, それを元に CAD システムにおけるユーザ・インタフェースのあり方についての検討を行った. その結果, 今後のユーザ・インタフェースの課題として, 以下の条件を満足するシステムが有効であると考えられる.

- 1) 並列処理への対応
- 2) システム全体の部品化の推進
- 3) ユーザモデルの設定
- 4) モジュール間のインタフェースと入出力インタフェースの記述
- 5) システムの状態とユーザの状態の記述
- 6) 状態に対する処理の記述
- 7) これらを実現するためのデータ構造

検討結果を部分的に検証するモデルとして, ICAN モデル (Input-Condition-Action-Notice Model) を考案した. ICAN モデルは, 表形式データ構造とシステムの状態表現に基づいたイベント駆動によるユーザ・インタフェースの対話制御のモデルである. CADCEUS のスケッチ機能に ICAN モデルをインプリメントし, イベント駆動型処理との適合性を初めとする同モデルの利点を検証した.

Abstract Besides looking through the current trends of user interfaces such as event-driven GUIs, GUI builders, visualization software items, CASE tools, network transition table editors, and so forth, this paper discusses how data communications (which provide basics for user interfaces) ought to be, followed by the way user interfaces for CAD systems ought to be accordingly.

When future ideal user interfaces are envisioned, systems considered to be the most effective would have to be capable of making available the functionalities listed below :

- 1) concurrent processing
- 2) segmentation as software components
- 3) introduction of "user models"
- 4) description for a module-to-module interface and an I/O interface
- 5) description for system and user conditions
- 6) description for the processing of these conditions
- 7) data structure suited for the implementation of these features.

The author has devised a model called the "ICAN model" (input-condition-action-notice model) as a partial verifier of the above-listed capabilities. The ICAN model is designed to control dialogs in event-driven user interfaces based on table type data structure and system condition expressions. This paper depicts the advantages of the model, which have been verified through its adoption to the sketch module

of the CADCEUS system, including its applicability to event-driven procedures.

1. はじめに

1.1 背景

ユーザ・インタフェースというもののシステムにおける位置づけを考えると、狭義には、今日一般に GUI と呼ばれるところのシステムの入出力に直結した部分を指すことが多いが、広義には、汎用的な対話制御のレベルまでを含めた、個々のアプリケーションとは独立したシステムの機能をユーザ・インタフェースとして考える。

このようなユーザ・インタフェース全体を統括的に管理するモジュールを UIMS (User Interface Management System) と呼ぶ。

UIMS のモデルとして今日考えられるものに、SeeHeim Model^{[1]~[5]} がある。Seeheim Model は、1983 年にドイツの Seeheim で開催された UIMS の会議で提唱されたモデルである。

この他のユーザ・インタフェースに関するモデルとしては、Smalltalk* の MVC (Model-View-Controller) Model, 認知理論と予測モデル的アプローチによる GOMS (Goal-Operation-Method-Selection of rules) Model (Card, Moran & Newell, 1983), Sheridan の監視制御モデル (1984) から派生した Triple Agent Model 等が挙げられる^{[14][15]}。

Seeheim Model は、以下の 3 要素からなる (図 1)。

- ① Presentation Component: ユーザの入出力に直結した部分であり、GUI に相当する。
- ② Dialogue Management Component: 対話処理の制御を行う部分。
- ③ Application Interface Model: ユーザ・インタフェースと AP とのインタフェース部 (API)。

Seeheim Model において、システムにおけるユーザ・インタフェースの位置づけと対象範囲、ユーザ・インタフェースの構成が明確なものとして定義付けられた。また、UIMS の各モジュールの独立性・汎用性をいかに高め、またその自動化をいかに進めるかということが、大きな課題として認識された。

その後、マルチウィンドウ型 EWS が急速に普及し、Xwindow** を中心とするウィ

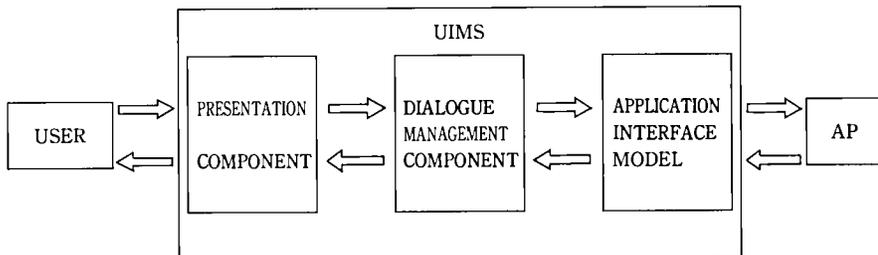


図 1 シーハイムモデル

Fig.1 Seeheim model

* Smalltalk は米国 Xerox 社の登録商標である。
 ** Xwindow は米国 M. I. T の登録商標である。

ンドウシステムとその上の GUI としての OSF/Motif*, OPEN LOOK** 等が CAD の世界でも今日普及しつつある。

このような動向の中で、Seeheim Model の Presentation Model に相当する部分については、それらの GUI の導入により、独立性・汎用性が高まり、またその構築の自動化を助けるツールとしても、各種の GUI ビルダ－(インタフェースビルダ－)が手に入るような状況になってきた。

しかしながら、UIMS の GUI 以外の部分、とくに対話制御の方式に関しては、CAD の分野では、この部分のあり方が重要であるにも関わらず、GUI と同レベルの標準化は進んでおらず、個々の CAD システムで個別的な開発が繰り返されている。

Seeheim Model では、Dialogue Management のモデルを状態遷移ネットワークモデル・文法モデル・イベントモデルの 3 種類に類別している。現在の多くの CAD システムの対話制御の方式は、文法モデルに基づいた逐次処理制御である。

一方、前述のマルチウィンドウ処理を前提とした GUI における今日の一般的な制御方式はイベント駆動型制御である。したがって、CAD の従来の対話制御とイベント駆動型の GUI との間の整合性、さらにはイベント駆動型処理をユーザ・インタフェース全体にどのように拡張していくかということが、今後の課題になるものと思われる。

本稿では、このような問題を中心に、CAD におけるユーザ・インタフェースのあり方について検討する。

1.2 本稿における立場

前節にも述べたように、ここではユーザ・インタフェースを対話処理制御も含めたものとして考えるが、さらに、時系列的な範囲としてどのように捉えるか、またそれらの前提として、ユーザ・インタフェースというものをどのような概念で捉えているかが問題となる。

とくに、CAD システムのユーザ・インタフェースというものを考えるとき、それが固定的な完成形態を長年維持することはまず有り得ない。CAD システムは半年ないし 1 年単位で常に改造されていくのが普通である。それに伴って、ユーザ・インタフェースに関しても改造が加えられていく。

したがって、ユーザ・インタフェースというもののあり方は、このようなシステムの発展過程の中で捉えられなければならないと考える。

本稿では、以下の立場のもとで考察を進める。

- 1) システムとは、ある究極の静的な完成状態を持ち得るものではなく、より望ましい状態に向かって常に動的に進化していくものである。
- 2) ユーザ・インタフェースとは、人間がシステムに指示を与え、システムがそれに応答するための窓口であると同時に、上記のようなシステムの環境適応行為がユーザ・インタフェースを媒介とする人間とのコミュニケーションをもとに行われていく。
- 3) コミュニケーションとは何かということが、ユーザ・インタフェースのあり方を規定する。

* OSF/Motif は米国 Open Software Foundation 社の登録商標である。

** OPEN Look は米国 AT & T 社の登録商標である。

4) 環境適応性の高いシステムのあり方が、結果的にユーザ・インタフェースの向上をもたらす。

このように、ユーザ・インタフェースの問題とは、システムの初期開発段階から、運用維持管理、改造、移植までにわたるユーザと情報処理システムとの間の総合的なコミュニケーションの問題であると考えられる。

1.3 本稿の構成

本稿の2章以下では、次の構成でユーザ・インタフェースに関する考察と提案を行う。

まず、2章ではユーザ・インタフェースの最近の動向事例として、イベント駆動型GUI、GUIビルダー、CASEツール、ビジュアライゼーションソフト、状態遷移表エディタを取り上げ、それらにおけるユーザ・インタフェースの実例を紹介するとともに、ユーザ・インタフェースとしての特徴を分析し、今後のユーザ・インタフェースの方向を探る。

次に、3章では、ユーザ・インタフェースの根底をなすのはコミュニケーションであるという観点から、基本的な人間のコミュニケーションの成立の有り方について考察し、コミュニケーションモデルの提案を行う。

さらに、4章では、2章と3章の考察結果に基づいて、今後のユーザ・インタフェースが満たすべき諸条件についてのまとめを行う。

5章では、4章における結論に基づき、それを検証するモデルとして、ICANモデル (Input-Condition-Action-Notice Model) を提案するとともに、その具体的なシステムへの導入事例を紹介する。

最後に、6章で、本稿では取り上げることのできなかったユーザ・インタフェースの問題について補足する。

2. ユーザ・インタフェースに関する最近の動向

2.1 ユーザ・インタフェースの動向事例

2.1.1 GUIとGUIビルダー

1970年代にゼロックス・パロアルト研究所でSmalltalkが開発され、それが一つの礎となって、1980年代になってからマルチウィンドウ・アイコン・マウスを用いたGUIとその開発言語としてのオブジェクト指向言語が徐々に普及していった。現在では、Xwindow, OSF/Motif, C++等がウィンドウシステム、GUI、オブジェクト指向言語の標準と成りつつある。

今日までの一連の動向の中でとくに注目すべきポイントは、システムにおけるGUIの重要性が広がり、それと並行して、それまでのプログラム駆動型処理ないしデータ駆動型処理に替わって、イベント駆動型処理が重視されるようになったという点である。

イベント駆動とは言い替えばユーザ駆動であり、すなわちそれは対話処理におけるユーザの指示に動的に対応する処理形態である。Direct Manipulation (直接操作) にも対応する処理形態と言える。

もう一つの最近の動向として、OSF/Motif, OPEN LOOK等のGUIの普及に伴い、

さらにそれらのメニューやアイコンの作成とそのレイアウト等を対話処理で簡単に行えるようにした GUI ビルダ（インタフェースビルダ）が使われるようになってきたということが挙げられる。ソフトウェア環境として、あらかじめ GUI ビルダまで含めて用意されているものは、たとえば、NextStep* 等、まだ例が少ないが、その一方で、OSF/Motif ベースの GUI ビルダはすでに各種のものが市販されている。これらの中には Syclops II** のように、DB システム用 GUI として、RDB との相性の良さから Prolog を言語処理系の核としたツールも見られる。GUI ビルダの機能を持った最近のソフトウェアの中で、とくに興味深いものに HyperCard***^[8] と Garnet^{[9][16]} がある。

HyperCard は、一見いわゆるカード型データベースであるが、さまざまな機能がボタンに組み込まれており、それらをユーザが自由にレイアウトすることができ、また自作のプログラムをボタンに組み込むこともできる。

HyperCard のソフトウェア部品は“スタック”と呼ばれる。

スタックの作成には、あらかじめ用意された部品を画面上で編集することにより、コマンドスクリプトを自動生成する方式が用いられているため、プログラムのわからないユーザにも簡単にソフトウェアを構築することが可能である。

任意の部品をコピー・ペーストすることが可能であり、HyperTalk を利用することにより、スクリプトの修正や新規作成も可能である。HyperTalk は、スクリプトを作成・修正するための言語である。HyperCard の制御はイベント駆動型のメッセージハンドラによって行われ、マウス入力に対して何を行うかをスクリプトに記述する（図 2）。

HyperCard の特徴は、システム全体が基本的に DB システムとして位置づけられな

```

nn mouseUp

  answer "Sort all cards of this stack according to:"
  with "First Name" or "Last Name" or "Cancel"

  if it is "First Name"

    then sort by first word of first line of field "Name and Address"

  else if it is "Last Name"

    then sort by last word of first line of field "Name and Address"

end mouseUp

```

図 2 HyperTalk によるスクリプトの実例

Fig. 2 A sample script by HyperTalk

* NextStep は米国 NEXT 社の登録商標である。

** Syclops II は株式会社アイザックの登録商標である。

*** HyperCard は米国 Apple Computer 社の登録商標である。

がら、親和性の高い GUI の元で、自由にアプリケーションプログラムを組み込めるようにすることにより、これまでになかった新しいソフトウェア環境のスタイルを提示している点にある。今後は、OS に近いソフトウェア環境のレベルで HyperCard ライクなユーザ・インタフェースの方式が普及していくことも考えられる。

一方、Garnet (Generating an Amalgam of Real-time Novel Editors and Toolkits: 実時間エディタとツールキットの集合体の生成) は、既存のシステムの GUI を構築するためのツールキットであるが、それまでの GUI ビルダーにはない多くの特徴を持っている。

Garnet のグラフィックスシステムは、オブジェクト指向型の KR (Knowledge Representation: 知識表現) オブジェクトシステムによるグラフィックスオブジェクトが処理の単位となっている。

Garnet のソフトウェアは CommonLisp で記述されており、Lisp の動的で柔軟な特性が KR オブジェクトシステムに生かされている。

KR オブジェクトシステムでは、通常のオブジェクト指向言語におけるクラス/インスタンスモデルではなく、プロトタイプ/インスタンスモデルを用いてオブジェクトを表現する。

プロトタイプ/インスタンスモデルでは、インスタンスとクラスを区別しない。どのインスタンスも他のインスタンスのプロトタイプになることができる。すべてのデータとメソッドは「スロット」に記憶する。上書きしなかったスロットは、プロトタイプのスロットの値をインスタンスのスロットが継承する。

データスロットとメソッドスロットを区別せず、どのスロットにも任意の型のデータを格納できる (LISP の考え方に基づく)。

インスタンスには任意の数だけ新しいスロットを追加して良い。スロット数は動的に変えられる。新しいメソッドを適切なスロットに代入すれば、メソッドが動的に変更される。

プロトタイプ/インスタンスモデルはクラス/インスタンスモデルに比べて動的で柔軟性があり、画面上でプロトタイプをユーザが編集すると、すべてのインスタンスにその結果が自動的に反映される。たとえば、メニュープロトタイプの look-and-feel を変えると、そのインスタンスであるすべてのメニューが即時に変化する (図 3)。

KR オブジェクトシステムには、制約システムの機能が組み込まれている。オブジェ

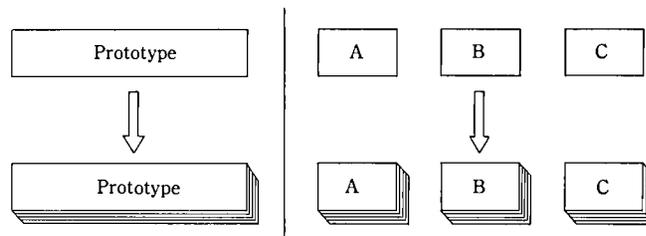


図 3 プロトタイプの属性の継承

Fig. 3 Inheritance of prototype attributes

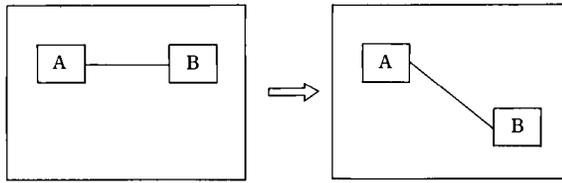


図 4 制約の効果

Fig. 4 Effect of constraint

クト間の関係を制約型プログラムで定義することにより、オブジェクトのレイアウトやその変更に伴う他のオブジェクトの振舞いを自動化することができる。制約は一方方向であり、制約式により参照される側のオブジェクトが変化しても元のオブジェクトは再評価されないが、このような処理方式を採用することにより、高速な再評価が可能となっている（図4）。

GarnetのGUIビルダは、Lapidaryインタフェースビルダと呼ばれる。Lapidaryインタフェースビルダにより、メニュー等のツールキットの構成要素を対話処理でプログラムレスに作成することができる。さらに、これらの構成要素のオブジェクトとしての実行時の振舞いを、実際にデモンストレーションを行ってみることを通して、定義することができる。

Garnetの最大の特徴は、徹底したオブジェクト指向方式の導入と、Lisp言語の採用による柔軟性の高さにある。Garnetのグラフィックスオブジェクトはデータとグラフィックス表示とがオブジェクトとして一体となっており、オブジェクトの属性の変更が動的にグラフィックス表示に反映される。また、画面上に表示されたオブジェクトをユーザが変更すれば、それが直接データに反映される。このような機能が、アプリケーションプログラムによる制御に依存しない形で、システムとして実現されている。

2.1.2 CASE ツール

CASE ツールは、ソフトウェア生産の合理化を進めるために、80年代になってから急速に発展し、各社から各種のツールが発表されているが、とくに最近の傾向としては、以下の二つが挙げられると思われる。

- 1) 上流から下流までを一貫して支援し、かつ、ソフトウェアの運用・維持・管理・改造までをも対象とする統合CASEツールが注目され、主流となりつつある。
- 2) システムのオープン化を強調し、共通のフレームワークのもとで、各社の既存の様々なツールをユーザが自由に組み合わせて利用できるようなあり方が重視されてきた。

とくに、後者の傾向は注目に値し、後述のビジュアルゼーションソフトとの共通点も見出せる。ユーザ・インタフェースの向上という問題を考えるとき、非常に重要な考え方であると思われる。

オープン化の傾向を代表するCASEツールとして、SoftBench^[12]がある。SoftBenchは、各種のソフトウェア開発ツールに対する統合環境のフレームワークであり、満たすべき仕様として、以下のものが設定されている。

* SoftBenchは米国Hewlett-Packard社の登録商標である。

- 統合されたツールを連動させたタスク指向の環境
- 既存ツールの活用
- 分散型コンピュータ環境の支援
- 多様なチーム編成，作業スタイル，ライフサイクルに対応

このようなフレームワークを実現するためのモデルとして，トースターモデル（図5）が考案された。

トースターモデルでは，仮想的なスロットに挿入されたさまざまな既存のツールやアプリケーションに対して，共通的なユーザ・インタフェースと，データ統合，制御統合の機能が提供される。

SofBench のトースターモデルでは，共通ユーザ・インタフェースの GUI として，OSF/Motif を採用している。GUI を持たない既存のツールを Encapsulator によりカプセル化することにより，共通的な OSF/Motif の GUI を利用することが可能となる。

SoftBench の各ツールは独立したプロセスとして実行され，ツール間の通信は BMS (Broadcast Message Server) によるプロセス間通信として実現される。各ツールの上には，より抽象度の高い，EDIT (各種エディタのツールクラス)，BUILD (make 等のツールクラス) 等のツールクラスが設定されており，クラス単位でのメッセージ通信が可能となっている。このような仕組みはオブジェクト指向の考え方に基づいている。トースターモデルでは，個々のツールがオブジェクトとしてカプセル化されており，一つのクラスとして認識される。

たとえば，新しいエディタを追加するような場合には，EDIT クラスの内容を変更すれば，全体の構成を変更しなくても，他のクラスとの通信を継続することができる。

ツールのカプセル化は，Encapsulator により行う。標準入出力に基づいたコマンドベースのツールをカプセル化する場合には，ツールのソースコードにはまったく手を加えずに，EDL (Encapsulator 記述言語) によるカプセル化が可能である。

EDL は，C 言語ライクな会話形式のプログラム言語であり，widget のようなウィン

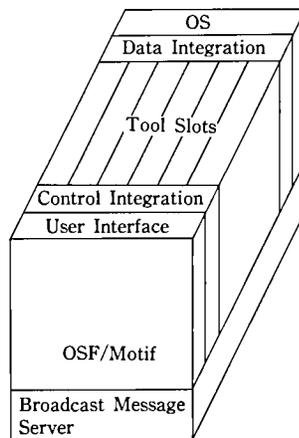


図5 トースターモデル

Fig.5 Toaster model

ドウインタフェースと BMS とのメッセージ通信のための組み込み関数を持っている。EDL のプログラミングはイベント駆動型であり、EDL で記述されたプログラムが既存 AP の標準入出力に対応することにより、カプセル化が実現される。さらに、既存 AP のソースコードを変更すれば、既存の Xwindow で動くツールに BMS を介したメッセージ通信の機能を付加できる。

このように、既存のソフトウェアをプロセスに対応させて、それらの間でメッセージ通信を行うことにより、ソフトウェア部品を自由に組み合わせて使用することを可能にするという考え方は、今後の統合的なソフトウェア環境におけるユーザ・インタフェースを考える上で、極めて重要である。

2.1.3 ビジュアライゼーションソフト

ビジュアライゼーションソフトと呼ばれるソフトウェアが世の中で注目されるようになったのは、まだごく最近のことであると思われる。ビジュアライゼーションソフトは、既存の各種解析プログラムを自由に組み合わせ、その計算結果をさまざまな形で可視化するツールである。

ビジュアライゼーションソフトの特徴は、既存のソフトウェアとユーザとの間に汎用的で操作性の高いユーザ・インタフェースを提供する点にある。

具体例としては、AVS*、IRIS Explorer** 等が挙げられる。

AVS では、データや手続き等をアイコンの形で表し、これらのアイコンを組み合わせることでプログラミングを行う。AVS3 では 120 個のモジュールが標準で用意されており、これらをアイコンとして画面に表示し、それらの入出力をマウスで指示して接続することにより、各モジュール間の関係を定義し、プログラミングを完成させる。

各モジュールは、あらかじめ入出力インタフェースが用意され、受け渡されるデータのデータ構造ごとに色分けされている。画面上で同じ色の出力と入力を選択することにより、モジュール間のインタフェースが確立する。

既存のアプリケーションプログラムをもとに、新規モジュールを作成することも可能である。この場合は、AVS 関数を使って C 言語または FORTRAN でプログラムを記述する。AVS 関数により、GUI 部との関連づけを容易に実現できる。

アプリケーションを実行すると、ネットワーク内で現在実行中のモジュールが点滅するため、実行状況とデータの流れを視覚的に把握することができる。

また、モジュールごとに画面上でそのパラメータ値を変更し、その結果をリアルタイムで presentation にフィードバックさせることができる。

モジュールの配置と接続およびプログラムの実行は、ネットワークエディタにより行う。モジュールパレットから必要なモジュールのアイコンをマウスにより引き出して、配置する。各モジュールの計算に必要なパラメータは、左のスペースにダイヤル、スライダ等の形で表示され、これを操作することにより、AP を実行する。

AVS の処理構造は、各モジュールがプロセスに対応し、プロセス間通信により、ネットワーク化されたアプリケーションが実行される。モジュール間の通信に共有メモリを使うことで、メモリの節約や転送時間の短縮を図ることができる (図 6)。

* AVS は米国 AVS 社の登録商標である。

** IRIS Explorer は米国 Silicon Graphics 社の登録商標である。

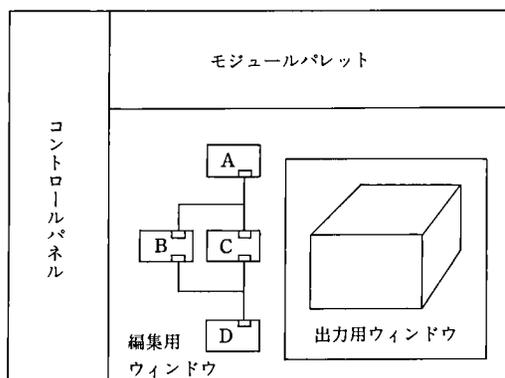


図 6 AVS のユーザ・インタフェース

Fig. 6 AVS user interface

IRIS Explorer は、AVS の開発者がシリコングラフィックス社に移籍して開発したビジュアライゼーション・ソフトであるため、内容的には非常に類似している。AVSの方がより完成度が高い一方で、IRIS Explorer は AVS にはない特徴を持っている。

モジュールを組み合わせて作成したプログラムを AVS ではネットワークと言うが、IRIS Explorer ではマップと言う。IRIS Explorer においてとくに改善が加えられている点の一つとして、このマップのユーザ・インタフェースが挙げられる。

IRIS Explorer では、画面に表示されたモジュールの左が入力、右が出力である。データ構造による色分けはない。AVS におけるパラメータ操作が画面左側のコントロールパネルにまとめられた形で行われるのに対し、IRIS Explorer では、モジュールの表示サイズを変更することができ、大きくすると、パラメータ操作のボタン類が表示される仕組みになっている。

IRIS Explorer におけるもう一つの大きな改善点は、カスタマイズに関する機能である。ユーザの手持ちのデータとプログラムをシステムに組み込むためのツールとして、データスクライプとモジュールビルダーが用意されている。データスクライプは、入力データのデータ構造を定義することにより、それを IRIS Explorer のデータ構造に変換して、システムに取り込むことができる。モジュールビルダーは、AP の入出力インタフェースを構築し、Explorer のモジュールとする。これらの処理は、画面上のグラフィカルな対話処理として実現され、AVS にはない強力な機能である。

ビジュアライゼーションソフトは、既存のソフトウェアをソフトウェア部品として自由に組み合わせて使用するという点で、前述の SoftBench に極めて類似している。また、その構成操作を画面上での対話的なグラフィックス処理として行えるという点においては、さらに進化していると言える。

これらのビジュアライゼーションソフトに見られるような方式は、今後、CAD を含む多くの分野において、ソフトウェア環境の一般的な方式として普及していくものと思われる。

2.1.4 状態遷移表エディタ

状態遷移表は、通信制御系のプログラム開発でよく用いられるが、最近それを画面

上に表示し、対話形式で状態遷移に基づいたプログラムを自動生成するツールが各種出回ってきた。

このようなツールの例としては、NECのSOFTEX/SEM^[13]がある。SOFTEX/SEMは、より汎用的なソフトウェア自動合成システム構築シェルであるSOFTEX/Sの適用事例として開発された。

SOFTEX/SEMの状態遷移表エディタでは、イベント名の行と状態名の列から構成される二次元マトリックスの表を画面上に表示し、その各セルに状態条件式(どのような状態の時にアクションを起こすか)、アクション名(その状態の時にどのようなアクションを起こすか)、次状態名(そのアクションの結果として、次にどのような状態になるか)を記入する(図7)。

エディタにより作成された状態遷移表を元に、C++の拡張言語であるDSL/C++のソースコードは自動生成される。

エディタが仕様検証機能とシミュレーション機能を持ち、入力された次状態名や状態変数名の有無のチェック、空フィールドのチェック、到達可能性のチェック、無限ループのチェック等を行うことができる。

SOFTEX/SEMは1レベルの複数の状態間の遷移を対象にするが、ある状態に対応するセルを階層的に細分することにより、複数レベルにわたる状態の組み合わせに対応できるものもある。しかしながら、このような組み合わせ方法の場合“ある状態下において、さらにある状態であれば”，という状態表現であるため任意の状態の組み合わせに対する制御はしづらい。

一般に、状態遷移表エディタの欠点としては、以下の点があげられる。

- ① 複数の任意の状態の組合せを扱えずらい。
- ② アクションに対する入出力引数が固定的なものになりがちである。
(これは、イベント処理全般の問題である。)
- ③ 表の内容が希薄(空セルが多い)になりやすいため、データ効率が悪い。

いずれにしても、現在のこの種のツールは、状態遷移のあり方が限定された小規模プログラム向けであるが、それでも特定の対象分野に十分役立つものと思われる。(数十倍の開発効率の向上が期待できると言われる。)

さらに、現在のGUIを中心とするユーザ・インタフェースの方向がイベント駆動型

状態名 イベント名	状態 1	状態 2	状態 3	状態 4	
イベント 1	状態条件式 アクション名 次状態名				
イベント 2					

図 7 状態遷移表エディタ

Fig. 7 Transition network table editor

に向かっていることを考えると、この種のツールに対して必要な改善が加えられれば、それがユーザ・インタフェースのソフトウェア開発に役立つ可能性は大きい。

また、観点を変えて見れば、状態遷移表エディタというソフトウェアのあり方そのものが、ユーザ・インタフェースとしての優れた特性を持っている。すなわち、表形式というユーザとの親和性の高い画面との対話処理で、必要最小限の入力操作により、自動化度の高い生産行為が実現でき、また設定内容の修正・変更も容易である。

2.2 今日のユーザ・インタフェースの特徴

前章で述べたユーザ・インタフェース動向事例より、今日のユーザ・インタフェースの特徴を以下のようにまとめることができる。

- 1) マルチウィンドウを用いた並列処理……マルチウィンドウ、アイコン、マウスを用いたビットマップ型グラフィックワークステーションによる対話型のユーザ・インタフェースである。

マルチウィンドウやアイコンを並列的に扱うため、並列処理化が進んでいる。

- 2) わかりやすい表現 (Look and Feel) と直接操作 (Direct Manipulation)……操作全体を簡単でわかり易いものにするのを重視するのは当然であるが、その具体的な手段として、アイコンその他の表現に身近な物の形を利用する。

また、操作方法を日常の動作を象徴化したものにする。

たとえば、「クリップ」でウィンドウを固定する、「ごみ箱」にファイルを捨てる、等である。

このような操作方式により、ユーザの画面上でのオペレーションをシステムの内部処理に直接反映するようにし、操作性の向上を図る。

- 3) イベント駆動型制御……並列処理を実現するため、制御方式はマウスクリックを主体とするイベント駆動型の制御である。
- 4) ユーザプログラムとユーザ・インタフェースの独立……ユーザプログラムとそのインタフェース部を独立させることにより、ユーザ・インタフェースを意識しないでプログラムを作成することを可能とする。ユーザ・インタフェースに変更があっても、プログラムは変更しなくて良い。各種のアプリケーション間で、ユーザ・インタフェースを統一することが容易となり、使いやすくなる。
- 5) ユーザ・インタフェースの統一……とくに、ユーザ・インタフェースの統一という点に関しては、各アプリケーション間に限らず、OS、各種のユーティリティを含めて、そのマシン上のすべてのソフトウェアのユーザ・インタフェースの統一を図る。
- 6) ソフトウェアの部品化……ユーザ・インタフェースの構成要素（メニュー、アイコン、等）を部品化することにより、ユーザが自由にそれらを組み合わせてユーザ・インタフェースを作成することが可能となる。

さらに、統一的なユーザ・インタフェースのもとで、既存のソフトウェアを部品化し、自由に組み合わせて使用することができる。

- 7) プログラムレス化 (コードレス化)……どこにどのようなメニューを表示するか等、すべてのユーザ・インタフェースを、部品を利用した対話処理で作成することにより、プログラムの作成を不要とし、変更もユーザ自身が簡単に行えるよう

にする。

さらに、ソフトウェアの部品化や組み込みも画面上でのグラフィカルな対話処理によりプログラムレス化また、ソフトウェアの実際の振舞いについても、デモンストレーション等の手段により、プログラムレス化する。

(一般的なプログラミングそのものについても、ビジュアルなユーザ・インタフェースのもとでプログラムレス化する傾向が、CASE ツール、ビジュアルプログラミング等に見られる。)

- 8) マルチメディア化……今までのテキスト、図形情報だけでなく、音声、ビデオ、等の多様な情報を扱えるようにする。

また、これに伴い、各メディアにおける情報の仕様を統一し、その品質を向上させる。たとえば、画面に表示される文字・図形がそのままの形でプリンタに出力されるようにする。(WYSIWYG: What You See Is What You Get)

- 9) ネットワーク化……Xwindow ベースの GUI に代表されるように、ネットワーク上で使用されることを前提とする傾向にある。
- 10) オブジェクト指向の導入……イベント駆動型の処理を自然な形で記述し、また、ソフトウェアの部品化を容易に実現するために、オブジェクト指向言語を使用する。

この他、カード型データベースやスプレッドシートのように操作性の高いソフトウェアツールを一般的なソフトウェア環境のユーザ・インタフェースとして位置づけようとする傾向も見られる。

以上の特徴を一言でまとめるならば、統一的なユーザ・インタフェースのもとでのビジュアル化・並列処理化・部品化ということが言えよう。また、それらの特徴の根底にあるものは、人間の思考・認識形態および行動様式に対する直接的で動的な対応であると考えられる。

3. コミュニケーションモデル

1章で述べたように、ユーザ・インタフェースの問題はより一般的なコミュニケーションの問題に帰着すると考えられる。2章に挙げたさまざまな事例からも、ユーザとシステムとの間のコミュニケーションには多様な場面とスタイルがあることがわかる。

本来、コミュニケーションとは人間相互間の言語を中心とする情報交換行為を指す。本章では、コミュニケーションとは何かという問題に立ち帰り、コミュニケーションの共通概念に基づくコミュニケーションモデルを設定することにより、ユーザ・インタフェースの多様な問題に対する核となる緒を探ることとしたい。

これまで、言語学、心理学、社会学、哲学等、さまざまな分野でコミュニケーションのモデルが考えられてきた。コミュニケーションのモデルは、おおよそ次のような枠組を持っている。

- 1) 同じ世界に、同じ概念、同じ言語(表現・理解のルール)を持った2者が存在する。

(ここで言う同じ世界、同じ概念、同じ言語とは、まったく同じという意味では

なく、共有可能な部分を持つということである。そして、それらがどのように形成されたかは問わず、核心的な部分はアプリアリに存在すると考える。）

- 2) コミュニケーションの送り手は、伝えたい内容（2者間の共通概念の上で形成された何等かの意味のあるもの）を言語等によるメッセージとして、コミュニケーションの受け手に送る。
- 3) コミュニケーションの受け手は、送り手からのメッセージを自分の言語で理解し、メッセージの意味を自分の概念の上に位置づける。
- 4) 最初のメッセージ授受の段階では、メッセージの解釈が間違っている可能性があるが、その後のメッセージのやり取りの中で、最終的には正しくメッセージが理解される。

またその一方で、人間のコミュニケーション能力とそれに基づく共有言語を前提とする時、明確な意味が正しく表現できれば、それは正しく理解できると考える。

以上が、一般常識的なレベルでのコミュニケーションのモデルの概要であるが、一方、言語表現に厳密さを求めれば、多くの曖昧さを伴っている。

ここで、最も原始的なレベルにおけるコミュニケーションの成立について考察するため、私の考えるコミュニケーションモデルを提示したい。

日常の人間の社会生活において、われわれはそれぞれがまったく異なった主観を持っている（言い換えれば、別個の世界に生きている）にも関わらず、互いにコミュニケーションが成立していると信じ合って生活している。このような世界を簡明に記述できるモデルが存在すると考えられる。

3.1 存 在 論

まず、本質的に、そもそもコミュニケーションとは成立し得るものなのかという問題に対して、コミュニケーションとは主体相互の認識により成立するものであるという仮定のもとに、コミュニケーションモデルを考える。

以下の記述において、「認識主体」とは、認識能力を持った存在を指す。一般には、通常の間を想定しているが、必ずしも人間に限定はしない。「(被認識)実体」とは、認識主体により認識され得るところのあらゆる対象を指す。「認識」とは、日常の言語において使用される「認識」という言葉の持つ意味の枠組みの中で、以下の記述を成立させ得るところの「認識」を指す。(必ずしも一義的ではなく、コミュニケーションとの関連の中で多義性を持つ。)

なお、各々の記述に対応して、様相述語論理式による記号化を行った。論理式の公理体系は、体系 K+BF (Barcan Formula) の範囲内にある^[17]。

3.1.1 認識のあり方

- 1) 認識主体の集合が存在する。

$$U = \{a_1, a_2, a_3, \dots\}$$

- 2) 被認識実体（認識される実体）の集合が存在する。

$$\Omega = \{b_1, b_2, b_3, \dots\}$$

- 3) 被認識実体の集合には、認識主体そのもの、それ以外の被認識実体、それらに対する認識が含まれる。認識が被認識実体であるためには、それを認識する認識主体が存在することを前提とする。認識主体の集合は被認識実体の集合の部分集

合である。

$$U \subset \Omega$$

- 4) すべての被認識実体に対して、その被認識実体を認識する認識主体が存在する。
($\forall y/\Omega)(\exists x/U)R(x, y)$
- 5) すべての認識主体は自分自身を認識している。
($\forall x/U)R(x, x)$
- 6) すべての認識主体は、自らの被認識実体に対する認識を認識できる。
($\forall x/U)(\forall y/\Omega)(R(x, y) \supset \diamond R(x, R(x, y)))$)

3.1.2 客 観 論

- 1) ある認識主体のある被認識実体に対する認識が真であるならば、すべての認識主体のその被認識実体に対する認識が可能である。
(すべての主体が同じ認識能力を持ち、かつある実体が本当に存在するならば、ある主体がその実体を認識できれば、他の主体も同じようにその実体を認識できるはずである。)

$$(\forall y/\Omega)((\exists x/U)R(x, y) \supset (\forall z/U)\diamond R(z, y))$$

- 2) すべての認識主体のすべての被認識実体に対する認識が可能である。

$$(\forall x/U)(\forall y/\Omega)\diamond R(x, y)$$

- 3) したがって、被認識実体である任意の認識に対する認識が可能である。

3.1.3 主 観 論

ある認識主体のある被認識実体に対する認識が真であっても、他の認識主体のその被認識実体に対する認識が真であるとは限らない。

(互いに自分自身の認識に対してしか確信は持てない、)

$$(\forall y/\Omega)\diamond((\exists x/U)R(x, y) \wedge (\exists z/U)\neg R(z, y))$$

3.1.4 コミュニケーション

以上の前提条件のもとで、次の仮説を立てる。

- 1) 認識主体 x の認識主体 y に対する認識、認識主体 y の認識主体 x に対する認識、 x の y に対する認識に対する y の認識、 y の x に対する認識に対する x の認識が成立するとき、コミュニケーションが成立する。

(最後の2条件は、一つが成り立てば1方向の、二つが成り立てば2方向のコミュニケーションを成立させる。)

$$(\forall x/U)(\forall y/U)(R(x, y) \wedge R(y, x) \wedge R(y, R(x, y)) \supset C(x, y))$$

$$(\forall x/U)(\forall y/U)(R(x, y) \wedge R(y, x) \wedge R(x, R(y, x)) \supset C(y, x))$$

$$(\forall x/U)(\forall y/U)(R(x, y) \wedge R(y, x) \wedge R(y, R(x, y))$$

$$\wedge R(x, R(y, x)) \supset C(x, y) \wedge C(y, x))$$

- 2) 認識主体 x が認識主体 y を認識しているとき、ある被認識実体としての x から y へのメッセージ m が存在するならば、認識主体 y がメッセージ m を認識することにより、 x から y へのコミュニケーションが成立する。

メッセージが存在しなければ、メッセージは認識されない。 x が y を認識していなければ、 x から y へのメッセージは存在しない。

$$(\forall x/U)(\forall y/U)((R(x, y) \wedge (\exists m/\Omega)(M(m, x, y) \wedge R(y, m))) \supset C(x, y))$$

認識は常にある認識媒体を通して行われる。さらに、認識に対する認識はより困難な行為である。メッセージの存在が、認識主体 x 、認識主体 x のメッセージに対する認識、認識主体 x の認識主体 y に対する認識を認識主体 y が認識することを可能とする。メッセージの認識媒体としての特性は、そのメッセージがそのメッセージの発信者である認識主体のコミュニケーションに対する明確な意図のもとに成立するという点にある。

上記の仮説から以下の結論が導き出される。

客観論の立場に立っても、主観論の立場に立っても、コミュニケーションは成立し得る。

客観論の立場では、「任意の認識に対する認識が可能である」という前提から、コミュニケーションが可能であることは、明らかである。より具体的には、共通言語に基づくメッセージを媒介として可能となる。

主観論の立場では、 x の認識と y の認識はあくまでも異なるが、メッセージに対して自分が相手の意図を認識したと判断することにより、コミュニケーションが成立する。

メッセージが認識されるならば、メッセージは存在し、 x から y へのメッセージが存在すれば x は y を認識している。したがってメッセージ認識されるならば、 x は y を認識している。メッセージは主観内でのコミュニケーションの成立を保証する。

主観論におけるコミュニケーションは、それが外側の世界で正しく成立していることを保証するものではないが、メッセージ交換の繰り返しにより、客観的世界との整合に近づく。ただし、それが本質的に主観論的コミュニケーションであることに変わりはない。

3.1.5 主観と客観

本コミュニケーションモデルは、コミュニケーションの成立の前に立ちほだかる主観と客観の壁がどのように解消されるかということの問題としている。

コミュニケーションは共通の言語に対する共通であるという意識に基づいて行われ、それは実際に成立しうるが、主観的には、コミュニケーションが成立しているという自分の思い込みでしかない。

コミュニケーションを健全に成立させるもととなるのは、本来その前提条件となっているはずの共通基盤に対する絶対的信頼である。

すなわちそれは、「互いに同じ世界に生き、同じように考え、同じ言葉を話す、同じ人間である」という共通意識である。

3.2 機 構 論

存在論レベルでは、個々の人間が互いに独立した主観と認識能力を持っているにしても、それらが生物学的、進化論的、構造的に同一なものであるということを一緒にコミュニケーションの成立について考えた。

しかしながら、存在論レベルでのコミュニケーションモデルは、それをいくらつきつめたとしても、可能性の世界に留まるといった限界を持つ。コミュニケーションの成

立を事実として保証するものは何もない。様相論理学の可能世界意味論においても、可能世界というもののあり方の捉え方はさまざまであり、現代科学が単なる因果的必然性と確率的偶然性を越えた概念に迫りつつあるものの、究極的には「生き方」の問題に帰結すると言わざるを得ない。

ここで重要なことは、そのようにコミュニケーションの成立が不確かであるにもかかわらず、現にわれわれはコミュニケーションをしながら（そう信じながら）生きているということである。そのような信頼関係が成立すること、そのようにして生活が成り立つこと、社会が成り立つことが即ちコミュニケーションが成立することであるとも言える。少なくとも、コミュニケーションの問題と関わりながら、そのような生活が成立していることは事実であり、その事実を成り立たせている機構の方が、現実の応用レベルではむしろ重要であるという捉え方もできる。

そのような機構を考える上での緒は、存在論レベルのコミュニケーションモデルにおいては、認識の認識という概念とメッセージという概念により与えられている。ここでは、それをさらに発展させて考えてみたい。

3.2.1 メッセージ

言語あるいはそれに替わるものによるメッセージの存在により、コミュニケーションが成立する。メッセージには、以下のことが重要である。

- ① 主体相互間において成立するメッセージの共通基盤（たとえば共通言語）
- ② メッセージの表現のわかり易さ
- ③ メッセージ交換の繰り返しによる確認行為

メッセージのあり方は、以下のすべての問題と関連を持つ。

3.2.2 状態表現

コミュニケーションが成立するためには、相手の自分に対する認識に対する自分の認識が必要であり、それはメッセージを媒介として行われる。

これを円滑にするためには、自分と相手との関係において、自分がどういうものであるか、相手がどういうものであるか、自分がどういう状態にあるか、相手がどういう状態にあるかがうまく表現されることが重要である。

3.2.3 コミュニケーション対象のモデル化

そのような状態表現は、実際には、コミュニケーション主体が互いにその対象をモデル化し、メッセージ交換を通してそのモデルの情報を更新していくことによって成り立っていると考えられる。

すなわち、コミュニケーションは、直接相手と行われるのではなく、自分の頭の中の相手のモデルとの間で行われる。このようなモデルは固定的なものではなく、さまざまなメッセージにより、その情報が更新されると共に、モデルそのものが徐々に確立し、さらに発展していく。

モデルの構造はメッセージにより規定される。（複数のコミュニケーション主体、モデルの間で唯一直接的に共通なものはメッセージである。）

3.2.4 モデルによるコミュニケーションと思考

自分の頭の中には、自分自身のモデルもあり、またそれは複数である。言語は人間の論理的思考の中刻を成すということはよく言われるが、これは言い換えれば、思考

そのものがコミュニケーションによっているものであると考えられる。コミュニケーション主体の状態とコミュニケーションとは不可分なものであるが、さらに、思考とコミュニケーションとが不可分なものである。コミュニケーションによる思考のスタイルには、以下のようなものがある。

- ① 自分の頭の中の複数の自分のモデルの間でのコミュニケーション
- ② 自分の頭の中の一つの自分のモデルの中での自己反復的なコミュニケーション
- ③ 自分の頭の中の自分のモデルと相手のモデルの間でのコミュニケーション
- ④ 相手との実際のコミュニケーションを通してのモデル間のコミュニケーション、またその外延としての社会的思考行為

3.2.5 コミュニケーションにおける同期

末端的なコミュニケーションが自分の中のモデル間で行われるとしても、コミュニケーションの本質は、それが総体としては外界との間で行われるということである。すなわち、主体相互におけるメッセージ交換に適切な同期が取られないと、主体内部での思考活動としてのコミュニケーションもうまくいかない。

逆に言えば、適切な同期が取られれば、自分が相手の頭で考える、相手が自分の頭で考える、相手の考えを誘導する、複数の人間が同一の思考を進める、複数の人間の間で個々には思いもよらない考えを思いつく、等のさまざまなことが可能となる。また物理的に極めて限られたメッセージの交換で、正確に多量な情報を伝達することが可能となる。(構造的に可能であり、かつ主観的にそのようなイメージを共有する事が可能である。)

3.2.6 コミュニケーションと認識

3章の初めで、認識という概念はコミュニケーションとの関連において多義的であるということ述べた。コミュニケーションと認識とは、相互依存的に他を規定し合っている。唯一絶対の正しい認識というものはなく、認識は常に何らかの幅を持っており、その幅は、コミュニケーションの過程において、常に変化する。既に形成されている概念と実際に獲得される情報との間には、常に差異が存在する。このような「差異の中でのパターンマッチング」が、人間の思考とコミュニケーションのダイナミズムを生み出すものと思われる。

3.2.7 コミュニケーション主体

機構論は存在論の限界を抜け出す緒となるものではあるが、現象だけに左右されると、そもそもコミュニケーションを成立させ制御しているのは誰であるのかという主体性を見失うことになりかねない。

自己の主体性の先に何があるにしろ、主体間の相互関係により成り立つ存在論レベルでのコミュニケーションが基本であり、主体性に対する認識が重要である。「暗示」という概念もコミュニケーションを考える上で興味深いテーマである。それは言語を中心とするメッセージを媒介として、「イメージ」、「反射」、「同期」により成立する。「暗示」において重要なポイントは、暗示を受ける側がその暗示の内容に対して主体的な意図を持たなければ、暗示は成立し得ないということである。)

4. ユーザ・インタフェースの課題

2章では、最近のユーザ・インタフェースが、人間とのコミュニケーションにおいて、より直接的かつ動的に対応する傾向にあることを示した。また、3章では、コミュニケーションにおいて、認識、メッセージ、状態表現、対象のモデル化、同期、等の概念が重要であることを示した。

ユーザ・インタフェースの理想像として、システムを限りなく人間に近づけるべきか、あるいは、あくまで道具として割り切るべきかについては、大きく意見の分かれるところである。筆者は、今までの考察を通しての結論として、システムの構造を人間の思考構造と行動様式に則ったものにする事により、人間とシステムとの間のコミュニケーションの円滑化を図ることが、ユーザ・インタフェースの向上をもたらすと考える。

具体的な項目としては、近い将来のユーザ・インタフェースが必要とする機能として、以下のもの考える。

- ① 並列処理への対応
- ② システム全体の部品化の推進
- ③ ユーザモデルの設定
- ④ インタフェースの記述
- ⑤ 状態の記述
- ⑥ 状態に対する処理の記述
- ⑦ これらを実現するためのデータ構造

4.1 並列処理への対応

コミュニケーションモデルで示したように、われわれは自分の頭の中に設定した複数のモデルの間で並列的な思考を行っていると考えられる。

したがって、ビジュアライゼーションソフトの事例に示したように、GUIレベルに留まらず、システム全体として並列処理化することが、人間の思考形態との親和性を高めることになると考えられる。

すなわち、イベント駆動型処理に対する本稿における筆者の結論は、それをシステム全体に拡張し、さらにそれをマルチプロセス化、マルチプロセッサ化、ニューロプロセス化というような段階を経て、コンピュータシステムとしてのより純粋な並列処理形態にまで広げていくべきであるということである。

並列処理においては、各処理間の同期の制御をどのように行うかが重要である。また特に、ユーザ・インタフェースにおいては、それを処理システム内のクローズドな処理系としてとらえずに、対話対象であるユーザも含めた処理系全体の同期制御を考えなければならない。人間の生体としての時系列的リズムに対する配慮は、今後のユーザ・インタフェースを考える上で極めて重要なテーマであると考えられる。

4.2 システム全体の部品化の推進

ビジュアライゼーションソフトや SoftBench の事例に示したようなマルチプロセス化を進めることは、同時にソフトウェアの部品化をより純粋な形で進めることになる。部品化の究極的な形式は、既存のソフトウェアがまったくそのままの状態ですべて自由に組み込み、利用できることである。

CAD システムの場合で言えば、モデラ、レンダラ、DBMS 等をソフトウェア部品化し、ユーザが使いたい部品を統一的なユーザ・インタフェースのもとで自由に組み合わせ、利用することが考えられる。

このような部品化により、ユーザにとっての利便性が向上すると共に、システムの開発効率、改造のしやすさ等も大きく向上することが期待できる。

4.3 ユーザモデルの設定

コミュニケーションモデルで示したように、われわれはコミュニケーションの相手のモデルを頭の中に設定し、そのモデルとの間でコミュニケーションを行っているように考えられる。

ユーザ・インタフェースにおいても、インタフェースシステム内にユーザモデルを設定することにより、コミュニケーションの円滑化が期待できる。

ユーザモデルには、初心者、熟練者、システム開発者、設計技術者、オペレータ等の様々なクラスがあり、またそのインスタンスとしての個々のユーザがいる。ユーザモデルのクラスは、通常のオブジェクト指向言語におけるクラスのように固定的なものではなく、Garnet のプロトタイプのように動的に拡張されていくものである。ユーザ・インタフェースに学習機能を持たせることにより、ユーザとの対話処理を通して、ユーザモデルが自動的に拡張される。

ユーザモデルが設定されることにより、異なるユーザに個別に対応する形で、メニューの表示、処理対象範囲の設定、ツールの選択等が行われる。それらは UIMS の機能の枠の中で実現される。

(ユーザモデルという概念は、ステレオタイプモデルによる推薦図書選定システム GRUNDY (Rich, 1979)、ユーザモデルによる文章生成システム (Hovy 1984, Wilensky 1984)、対話ヒストリによる知的 CAD のユーザモデル (Bannon 1984, 浜川 1986) 等に導入されているが、本格的に実用レベルのものはまだ存在しないと考えられる。また、人間の認知プロセスを情報処理的にとらえたモデルとして、メンタルモデルや認知モデルがある (Card 1983, Norman 1983)^{[14][15]}。

4.4 インタフェースの記述

これらのことが実現されるためには、汎用的・効率的でかつ正確に各種のインタフェースが記述される必要がある。そのためには、各種のツール・サブシステムの間で共通的な言語が存在することが必要になる。

それぞれに対して encapsulator 的なものを用意し、その外側のインタフェース仕様レベルでは同一の共通言語を共有するような形式になるものと考えられる。

インタフェースの記述には、どのようなデータを扱うかという書式のレベルと、入出力と状態との関係の記述のレベル (どのような入力に対して状態がどのように更新され、どのような状態にどのような出力が対応付けられるか) とがある。後者は次に示す状態の記述との関連を持つ。

4.5 状態の記述

インタフェースの記述と並んで重要なのが、状態の記述である。状態の記述とは、前述のユーザモデルとシステムを構成する各サブシステムのモデルの状態の具体的な記述である。

従来の状態遷移ネットワークモデルでは、通常、単一な状態あるいは階層化された状態の組み合わせを扱うが、CADシステムを対象とする場合には、並列的な状態のさまざまな組み合わせを記述する必要があると考えられる。イベントモデルによるUIMSとして知られる *Sassafra*^[6] 等ではこのような状態を記述することも可能であるが、これまでイベントモデルは状態遷移ネットワークモデルに対比的なモデルとして、イベントハンドラによるイベントに対する直接的な処理のみが注目され、状態表現というものの位置づけが十分に整理されていないように思われる。

単にどのような出力が与えられたかだけでなく、現在システムがどのような状態にあるかを単純明解に把握できることが、ユーザが安心してシステムを利用できることにつながる。また、システム側からは、ユーザが現在どのような状態にあるかを把握できることにより、ユーザに対する適切な対応が可能となる。

このような機能を実現するためには、システムの各構成要素の状態の記述とユーザモデルの状態の記述に統一的な書式が用意され、それに対する統一的なユーザ・インタフェース (look-and-feel は変更可能) でその参照 (および必要があれば直接的な更新) が可能となる必要がある。

4.6 処理の記述

UIMS は対話処理も含めて制御するため、状態の記述と共に、状態に対応する処理の記述も必要となる。(ここでの処理はユーザ・インタフェースとユーザの間の入出力からは独立したシステムの内部処理を指す。)処理の記述に基づき、システムとユーザの各種の状態に対応する処理が呼び出され、実行される。

従来のCADシステムでは、処理の記述は単にこのメニューが選択されたらこのコマンドを実行するというレベルのものであったが、より複雑に組み合わせ的に表現される状態の記述にきめ細かく対応するものとして、状態と処理との関係が記述される必要がある。

システムの状態は、前述の入力処理 (場合によっては出力処理) によって更新されると共に、システムの内部処理によっても更新される。

イベント駆動型処理は、イベントから直接処理が呼び出されるのではなく、UIMSの管理する状態情報と処理情報を介して、間接的に実現される。このような方式により、APとUIMS (およびUIMSに対するイベントそのもの) との独立性が確保される。

4.7 データ構造

インタフェースの記述、状態の記述、処理の記述が有効に処理されるためのデータ構造が必要となる。

前述の *Seeheim Model* では、対話処理のモデルが状態遷移ネットワークモデル、文法モデル、イベントモデルに分類され、またこれらに対する *M. Green* の検討^[5] では、その中でイベントモデルが並列処理・取り消し処理・エラー処理等に対して最も処理能力が高いとされている。その一方で、イベントモデルは記述しやすく、プログラムがわかりにくいものになりがちであるという別の指摘もある。*M. Green* の論文でもイベントモデルで記述することを薦めている訳ではなく、対象に最も合ったモデルで記述し、それをイベントモデルのソースコードに自動変換すれば良いとしている。

既存の多くのシステムでは、状態遷移ネットワークモデルによるものも、イベントモデルによるものも、CADシステムが対象にするような複雑なシステムの状態をうまく記述できるものはないように思われる。また、状態表現が明示化されずにイベント処理の中に隠ぺいされてしまうと、プログラムの処理がわかりづらいものとなる。状態の記述とその処理との関係の記述を適切に行うことにより、個別的なプログラムの記述に依存しない汎用的な処理を実現することが重要である。

並列処理との適合性が高く、かつシンプルなデータ構造として、表形式のデータ構造が考えられる。各種の書式による表現を並列的に並べ、それらの情報(知識)を汎用的な制御機構により制御・処理する。DBとの相性が良く、また、データ編集のための統一的なインタフェースを構築することが容易である。表の構成や要素の型をダイナミックに変更できることが前提条件となる。表や要素の階層構造や参照ネットワーク関係のあり方については、今後の考察を必要とする。(たとえば、スプレッドシートのセルに他のスプレッドシートとの参照関係を持たせ、ネットワーク化するようなことも考えられる。)また、より図形的なビジュアルなオペレーションに基づくプログラムレス化(逐次的あるいはネットワーク的な構造となる)との対応関係をどのように考えるかも今後の課題である。

5. ICAN モデル

CADシステムのユーザ・インタフェースにおいて、4章における検討結果を具体的に検証するモデルとして、CADCEUSに導入されたICANモデルを紹介する。ICANモデルでは、並列処理化・部品化・状態表現を重視したイベント処理、等に関する独自の試みが成されている。

5.1 概 要

最近のCADシステムではパラメトリック処理が重視されるようになり、それと並行してその基本形状(パラメトリック断面の二次形状)を定義するための手段として、ラフにスケッチを描き、後から必要な寸法を定義して清書するようなツールが目されるようになった。

CADCEUSでは、その形状特徴モジュールの一部として、スケッチ機能が提供されている(図8)。スケッチ機能のユーザ・インタフェースは、従来のワイヤコマンド(ワイヤフレームモデリングのためのコマンド)が逐次的な対話処理によっているのに対して、よりイベント駆動型の処理仕様である。

たとえば、円を描く時、3点指示、2点1接線指示、1点2接線指示、3接線指示等のさまざまな入力パターンがある。ワイヤコマンドでは入力パターンによってコマンドが分けられ、それぞれに入力手順が決められている。これに対して、スケッチ機能では、同じ一つのコマンドの中で、任意の順番で通過点または接線を指示することにより円を定義することができる。

このような機能をできるだけ汎用的で効率的な形で実現するため、ICANモデル(Input-Condition-Action-Noticeモデル)を考案し、それをスケッチ機能にインプリメントした。

ICANモデルは、表形式データ構造とシステムの状態表現に基づき、ユーザ・イン

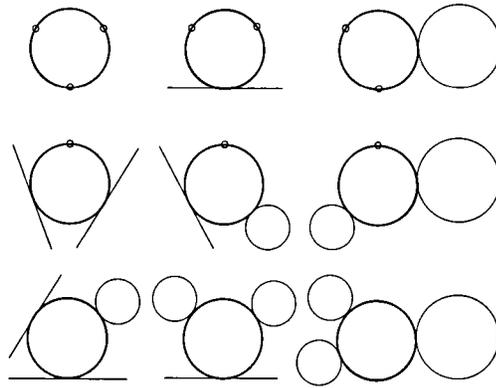


図 8 CADCEUS のスケッチ機能によるさまざまな円の描画

Fig. 8 Circle patterns by "Sketch Circle" command in "CADCEUS"

タフェースにおける対話処理制御をイベント駆動（状態駆動）で行う。CADCEUS では、コマンドスクリプト処理*と GUI の中間に位置付けられるものとしてインプリメントされた。すべて C 言語で記述され、UNIX** 上の CADCEUS のプロセスの一部として動作する。

5.2 構成

ICAN モデルは以下のモジュールから構成される。

- Input : システムの入力部
- Condition : システムの状態表現部
- Action : システムの内部処理部
- Notice : システムの出力部

各モジュールは、それぞれの制御部とそれぞれの処理に対応したテーブルを持ち、その制御部がその処理系に対する入力とテーブルの内容を参照することにより、各モジュールの処理が行われる（図 9）。

5.2.1 入力部 (Input)

Input モジュールは、Input Table と、その内容に基づいて入力イベントの内容に対応して Condition Table を更新する制御部からなる。

インプリメントで考慮した入力イベントは、マウスによるロケート座標とポップアップメニュー（拘束メニュー）の選択情報であり、どのような座標指示あるいは拘束指示が行われた時に Condition Table をどのように更新するかが、以下に示す Input Table に記述される。

制御部は、入力イベントが発生した時に Input Table を参照し、その内容に従って Condition Table を更新する。

```
/* Input Table 構造体 */
typedef struct_inpsub {
    UINT    type ;    /* Condition Table のデータタイプ
```

* CADCEUS におけるスクリプト言語によるカスタマイズプログラムの逐次処理をコマンドスクリプト処理という。

** UNIX オペレーティング・システムは UNIX System Laboratories, Inc.が開発し、ライセンスしている。

```

VSGPNT : 点
VSLIN  : 直線
VSGCIR  : 円
0       : セットしない */
UINT    flag ; /* 確定条件フラグ */
} VsInpSub ;
typedef struct_inptbl {
UINT    command ; /* 入力コマンド番号 */
UINT    dummy ;
VsInpSub subSA [VSINPDATMAX] ;
} VsInpTbl ;

```

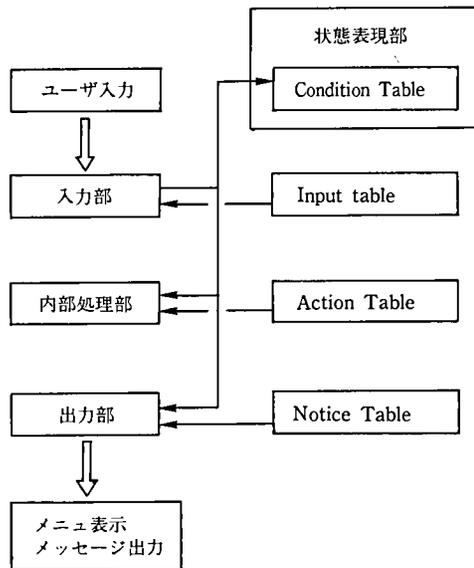


図 9 ICAN モデルのモジュール構成
Fig. 9 Module structure of ICAN model

5.2.2 状態表現部 (Condition)

Condition モジュールは、次に示すような Condition Table のみにより構成される。Condition Table には、処理中の幾何要素（線分・円・円弧・点）の確定状態が記述される。

Condition Table の内容は、フラグ部とインデックス部から構成される。フラグ部は整数表現されたフラグの各ビットが、平行線・垂直線・接線等の拘束状態および端点や幾何要素全体の確定状態を表す。フラグの状態表現は組み合わせ的であり、たとえば、円の 1 通過点と 1 接線のみが確定しているような状態を表現することができる。

インデックス部には、フラグ部のビットに 1 対 1 対応する形で幾何要素のインデックスがセットされる。

```

/* 点 Condition Table 構造体 */
typedef struct _vscndpnt {
UINT    fig ; /* 確定条件フラグ */

```

```

UINT      pid ;          /* 点 ID          */
DOUBLE    tpA[ 3 ] ;     /* 確定座標      */
DOUBLE    npA[ 3 ] ;     /* 近傍点座標    */
UINT      rpid ;         /* 参照点 ID     */
UINT      dummy ;

UINT      lid1 ;         /* 線分 1ID      */
UINT      lid2 ;         /* 線分 2ID      */
UINT      cid1 ;         /* 円・円弧 1ID  */
UINT      cid2 ;         /* 円・円弧 2ID  */
UINT      hpid ;         /* 水平拘束点 ID */
UINT      vpid ;         /* 垂直拘束点 ID */

} VsCndPnt ;

/* 直線 Condition Table 構造体 */
typedef struct _vscndlin {
    UINT      flg ;          /* 確定条件フラグ */
    UINT      lid ;          /* 直線 ID          */
    UINT      acflg ;        /* 暗黙拘束フラグ */
    UINT      dummy1 ;

    DOUBLE    bpA[ 3 ] ;     /* 基準点座標      */
    DOUBLE    dirA[ 3 ] ;    /* 単位方向ベクトル */
    DOUBLE    pp1A[ 3 ] ;    /* 通過点 1 座標    */
    DOUBLE    pp2A[ 3 ] ;    /* 通過点 2 座標    */
    DOUBLE    np1A[ 3 ] ;    /* 近傍点 1 座標    */
    DOUBLE    np2A[ 3 ] ;    /* 近傍点 2 座標    */
    UINT      rlid ;         /* 参照線分 ID     */
    UINT      dummy2 ;

    UINT      cid1 ;         /* 接円・円弧 1ID  */
    UINT      cid2 ;         /* 接円・円弧 2ID  */
    UINT      pcid1 ;        /* 中心通過円 ID1  */
    UINT      pcid2 ;        /* 中心通過円 ID2  */
    UINT      ppid1 ;        /* 通過点 ID1      */
    UINT      ppid2 ;        /* 通過点 ID2      */
} VsCndLin ;

/* 円 Condition Table 構造体 */
typedef struct _vscndcir {
    UINT      flg ;          /* 確定条件フラグ */
    UINT      cid ;          /* 円 ID            */
    UINT      cpid ;         /* 中心点 ID       */
    UINT      dummy ;

    DOUBLE    rds ;          /* 半径             */
    DOUBLE    pp1A[ 3 ] ;    /* 通過点 1 座標    */
    DOUBLE    pp2A[ 3 ] ;    /* 通過点 2 座標    */
    DOUBLE    pp3A[ 3 ] ;    /* 通過点 3 座標    */
    DOUBLE    np1A[ 3 ] ;    /* 近傍点 1 座標    */
    DOUBLE    np2A[ 3 ] ;    /* 近傍点 2 座標    */
    DOUBLE    np3A[ 3 ] ;    /* 近傍点 3 座標    */

```

```

UINT    tlid1 ;        /* 接線 ID1          */
UINT    tlid2 ;        /* 接線 ID2          */
UINT    tlid3 ;        /* 接線 ID3          */
UINT    tcid1 ;        /* 接円 ID1          */
UINT    tcid2 ;        /* 接円 ID2          */
UINT    tcid3 ;        /* 接円 ID3          */
} VscndCir ;

```

5.2.3 内部処理 (Action)

Action モジュールは、以下に示す Action Table とその内容に基づいて Condition Table の状態に対して実際の内部処理を呼び出す制御部からなる。

Action Table には、Condition Table のどのフラグが立っている時にどの処理を起動するかが記述される。Action Table のフラグ表現も組み合わせ的 (AND 表現) に記述され、たとえば、ある線分が他のある線分に平行でかつある点上にある時に、その直線を確定する処理ルーチンを呼び出すような記述が行われる。

制御部は、Action Table をサーチして、各テーブル要素のフラグと Condition Table のフラグとの間でビット演算を行い、条件を満足する処理ルーチンを呼び出す。ある Condition に対して複数の処理ルーチンが呼び出されることもある。また、処理ルーチンは処理結果として Condition Table を更新する。

制御部の処理は、処理中の幾何要素が確定するまで、あるいは Condition Table にそれ以上の変化がなくなるまで繰り返行われる。

```

/* Action Table 構造体 */
typedef struct _functbl {
    UINT    flag ;      /* 確定条件フラグ    */
    INT     (* func)() ; /* 計算メソッドポインタ */
} VsFcTbl ;

```

5.2.4 出力部 (Notice)

Notice モジュールは、Notice Table とその内容に基づいて Condition Table の状態に対してメニューやメッセージの表示制御を行う制御部からなる。

Notice Table は、複数のテーブルから構成される。あるテーブルではポップアップテーブルの初期表示内容が定義され、他のテーブルで Condition に対応するメニュー表示の抑制内容が定義される。たとえば、Condition Table 上で直線確定フラグが立っている場合には、平行線・垂直線・接線等を指示する拘束メニューは表示しない。

CADCEUS へのインプリメントでは、初期表示メニューの定義のみをテーブル化し、メッセージ表示はスクリプトで、メニューの抑制は表示制御のプログラム処理で実現した。

5.3 検 証

ICAN モデルのインプリメントにより、以下の点を検証することができた。

- 1) イベント駆動型処理を対話制御レベルで簡単に実現できる。
- 2) 組み合わせ的な状態表現を用いることにより、必要最小限の状態の組み合わせで複雑多様なシステムの処理状態を的確に記述できる。
- 3) 状態に対応する処理をイベントから独立した汎用的な形式で定義できるため、

用意する処理ルーチンの種類を重複のない必要最小限のコンパクトな構成に整理することができる。

- 4) 各種のテーブル化により、プログラムの構成がわかりやすくなり、開発効率が向上し、維持管理、改造もしやすくなる。
- 5) モジュールの汎用性、独立性が高いため、既存システムへのインプリメントが容易である。

一方、以下の問題点も確認された。

- 1) ユーザモデルが導入されていない。現在の ICAN モデルの枠組みの延長上でユーザモデルを扱えるかどうかについては、検討を要する。
- 2) インプリメントレベルの ICAN モデルでは、対話処理制御のすべてをカバーすることはできない。

(たとえば中心点タイプの円の作図は、1点目で中心点を指示し、2点目で円周上の通過点を指示する。その制御はコマンドスクリプトによる固定的な逐次処理によっている。今回のインプリメントでは、スケッチ機能には CADCEUS の一括入力(スクリプト処理において、複数の入力を一まとめに受け付ける機能)を利用していないが、個別のパラメータ入力には一括入力が並列処理的な有効性を持っている。両者の利点を組み合わせられるようなインプリメントがより望ましかった。)

- 3) Action の処理ルーチンが Condition Table を直接更新するため、状態表現と内部処理との独立性が弱く、処理内容がわかりづらい。

(処理内容の性質により、状態の更新をテーブル表現することが困難である。今後の大きな課題と言える。)

- 4) Action の制御部で複数の処理ルーチンが呼び出されるような場合、その呼び出し順序の違いが処理結果や処理効率に影響を及ぼすような場合に対する配慮がなされていない。

(並列処理の一般的な問題と考えられる。)

- 5) テーブルの書式の変更が必要になった場合には、テーブル全体の修正に余分な手間がかかる。

(テーブルの既存データを自動更新および編集できるツールが必要である。またテーブルに階層概念を導入することも必要である。)

- 6) Notice の機能の位置付けが十分に整理されておらず、インプリメントも部分的にしに行えなかった。UIMS におけるメニュー処理という観点を中心に、さらに整理する必要がある。

- 7) 取り消し処理、エラー処理について、汎用的で有効な処理機能が ICAN モデルの枠組みの中で用意されていない。

(ラバーバンドエコーに関しては、ICAN モデルとは別機能として用意され、インプリメントされたが、ラバーバンド処理のルーチンから ICAN モデルの機能の一部が参照されている。)

6. おわりに

本稿では、各種のユーザ・インタフェースの事例とコミュニケーションモデルに関する検討をもとに、システムとユーザの状態表現に基づいた並列処理によるユーザ・インタフェースのあり方を提案した。これはまだごく概念的なレベルのものであり、ICAN モデルはそのごく一部の機能を実現したものに過ぎない。またその一方で、取り消し処理やエラー処理のあり方、ラバーバンド処理の位置づけ、PHIGS レベルのグラフィックパッケージと GUI との関係等のさまざまな問題があり、それらは別個の問題というよりも、ユーザ・インタフェース全体の問題の一部として、互いに関連を持ちながら解決していくべき問題である。

今後は、これらの問題も考慮に入れながら、より総合的な UIMS のレベルでプロトタイプ化とその検証を進めていくことが必要であると考えられる。

-
- 参考文献 [1] 守屋慎次, “ユーザーインターフェース管理システムの基本概念および対話の方式”, 情報処理学会誌, Vol. 33, No. 11, 1992, pp. 1285~1294.
- [2] 宮崎一哉, “ユーザーインターフェース管理システムと対話制御”, 情報処理学誌, Vol. 33, No. 11, 1992, pp. 1295~1303.
- [3] 橋本治, “ユーザーインターフェース管理システムの研究動向と将来”, 情報処理学会誌, Vol. 33, No. 11, 1992, pp. 1331~1339.
- [4] G. Pfaff, Ed., “User Interface Management Systems”, Springer-Verlag, Berlin, 1985.
- [5] M. Green, “A Survey of Three Dialog Models”, ACM Transactions on Graphics, Vol. 5, No. 3, 1986, pp. 244~275.
- [6] R. D. Hill, “Supporting Concurrency, Communication, and Synchronization in Human-Computer Interaction-The sassafras User Interface Management Systems”, ACM Transaction on Graphics, Vol. 5, No. 3, 1986, pp. 179~210.
- [7] 石田晴久編, “X ウィンドウとその仲間たち”, bit 別冊, 共立出版, 1992.5.
- [8] C. Kaehler, “HyperCard その偉大なパワー”, Addison-Wesley, トッパン, 1989.
- [9] B. A. Myers, et al., “Garnet: 対話性すぐれ, 柔軟な GUI の作成を支援するツール”, 日経エレクトロニクス, No. 522, 1991.3, pp. 187~203.
- [10] 吉川慈人他, “ビジュアル・プログラミング技術を使った可視化ツール AVS”, 日経コンピュータグラフィックス, 1991.4, pp. 193~203.
- [11] 木崎健太郎, “解析データ統合能力を強化するビジュアルセッションソフト”, 日経コンピュータグラフィックス, 1992.5, pp. 97~106.
- [12] 山野修, “ツール間統合を実現したフレームワーク SoftBench”, SuperASCII, Vol. 3, No. 9, 1992.9.
- [13] 渡辺正信, “ソフトウェア自動合成システム構築シェル”, (社)人工知能学会, 第17回人工知能セミナー講演テキスト, 1992.7.8, pp. 23~34.
- [14] 人工知能学会編, “人工知能ハンドブック”, オーム社, 1990.1.
- [15] P. A. Hancock, M. H. Chignell, Ed., “知的インタフェイス”, Elsevier Science Publishers B. V., 海文堂出版, 1991.5.
- [16] B. A. Myers, Ed., “Languages for Developing User Interfaces”, Jones and Bartlett Publishers, 1992.
- [17] 遠藤弘, 白石光男, “現代の論理学”, 南窓社, 1990.3.

執筆者紹介 鈴木 良太郎 (Ryotaro Suzuki)

昭和 29 年生，55 年東京大学大学院工学系研究科修士課程修了。主に建築系の CAD システムの研究開発に従事。平成 4 年日本ユニシス(株)入社。現在，製造工業システム本部システム三部に所属。情報処理学会会員。



最新ビジュアルライゼーション技術の探求と CADCEUS/CG

A Quest for New Visualization Technique and CADCEUS/CG

大 平 昌 明

要 約 コンピュータ・グラフィックス (Computer Graphics) は変化と多様性に富んだ分野である。つい最近まで、高価なディスプレイを必要とし、コンピュータ資源を消費し、一部の専門家しかわからない分野だった。しかし、この数年間でハードウェアの価格性能比が良くなった。それによって、コンピュータ・グラフィックスは私達に絵画的なコミュニケーションを提供し、マンマシン・インタラクション (人と機械の相互作用) の促進に役立っている。

本稿では、照明とシェーディング (shading) 手法について説明する。はじめに基本的な内容を次に模様張り (texture mapping)、影付け (shadows)、透明 (transparency)、反射 (reflection)、屈折 (refraction)、照明モデル (illumination model) の説明をする。最後に CADCEUS/CG の紹介をする。

Abstract Computer graphics can be called a field rich in changes and variety. Until recently it has been an esoteric specialty involving expensive display hardware, the use of substantial computer resources and idiosyncratic software. However, improved hardware price performance as seen in the last few years has helped computer graphics a great deal to provide pictorial communications whereby it has established its proven position of being a major facilitator for man-machine interaction.

This paper is intended to discuss illumination and shading algorithms. The early part refers to algorithms most commonly adopted for today's hardware, followed by texture mapping, shadowing, transparency, reflection, refraction, illumination models and ray tracing. CADCEUS/CG is highlighted at the end.

1. はじめに

コンピュータ・グラフィックス (CG: Computer Graphics) はここ数年にわたるコンピュータの急激な性能向上と相俟って進歩している。そして、コンピュータ・グラフィックスの分野は変化と多様性に富んでいる。大学、研究所、建築・自動車等の業界、パッケージング業界、コマーシャルや映画等のアニメーション業界で、よりフォト・リアリスティック (写真のように写実的な) な表現をする努力がされ、次々と新しい手法が提案されている。しかし、このように多種多様な分野の人々が関係しているため、コンピュータ・グラフィックスという言葉はいろいろな使われ方がされ、手法についてもあまり真剣な整理がされていない。そこで、本稿ではコンピュータ・グラフィックスについて、現時点で発表されている代表的な手法を年代順に整理し、解説を加え、最後に CADCEUS/CG の特徴を簡単に紹介する。

ここで、本稿で使用する用語のモデル (model) の意味について説明する。モデルは模型 (scale model) の意味で、多数の面 (surface) から構成されている物である。もう一つのモデルは数学的モデル (mathematical model) で物体の動きを支配してい

る物理的法則を記述するものである。たとえば、人を例にとると、人は模型として捉えられる。そして、人の歩き方は数学的モデルとして方程式の集合で記述できる。

また、本稿で使用する記号の意味を次に示す。

- ＋：加算を意号する記号である。
- －：減算を意味する記号である。
- ・：乗算を意味する記号である。
- ／：除算を意味する記号である。
- K：普通文字は変数を意味する記号である。
- N：強調普通文字はベクトルを意味する記号である。

2. コンピュータ・グラフィックスとは

コンピュータ・グラフィックスの誕生は 1963 年にアメリカの MIT (マサチューセッツ工科大学) の I. E. サザランドによって発表されたスケッチパッド (skeepad) による。このスケッチパッドは CAD/CAM (Computer Aided Design/Computer Aided Manufacturing) システムの前身となった。そして、コンピュータ・グラフィックスが考えられた背景は次のように分類できる。

- 1) 科学的なシミュレーションを視覚的に見せる。この中には毎日、テレビで見ている天気予報も入る。また、細胞等の微小世界や宇宙等の超巨大空間をわかりやすく見せているテレビ番組等もこの分野である。
- 2) 製品のデザインに利用する。自動車や家電等の工業デザイン、飲料品や各種のパッケージングが代表的である。われわれが利用する GUI (Graphical User Interface) もこの分野である。
- 3) 商業アートに利用する。テレビのコマーシャルやアニメーション、映画が代表的である。最近では映画の特殊効果を実現するのに必要な技術になっている。
- 4) 芸術にコンピュータ・グラフィックスの分野が追加されようである。日本でも作品の展覧会が行われている。

このように、コンピュータ・グラフィックスは創造的な分野や情報伝達の新しい道具として鉛筆や紙と同じような役割を演じつつある。

2.1 コンピュータ・グラフィックスの要求

次のような観点から視覚に訴える情報伝達方法が求められていると考えている。

- 1) 人間と機械の対話として……人間は外部からの情報を五感 (五官) で感じる。五感とは視る、聴く、嗅ぐ、味わう、触るである。人間はこの五感を五官で感じる。五官は目、耳、鼻、舌、皮膚である。この五官のうちで目が一番理解力がある。たとえば、耳は音を聴くことで一次元的 (距離がわからないでもないが) な情報を受取る器官と言える。片目では二次元 (左右、上下がわかる) で見え、両目では三次元 (奥行きまでわかる) で見える。両目に時間の経過を加えると四次元 (アニメーション) で見える。このように、人間と機械の対話をする器官の能力から情報は今後ますます視覚化されると予想される。
- 2) 文化の伝承媒体として……過去、人類は自分達の文化を伝えるために、文字を誕生させた。そして、伝承を記号化し、言葉を考え出した。しかし、言葉で説明

しても情報は正確に伝えられないと悟り、細部を伝えることをあきらめている。詩という言葉でかろうじて、頭の中に映像をつくることを試みているが、同じ体験をしていない人に伝えられないでいる。言葉の本質的な限界と言える。そして、映像や画像で言葉の足りない情報を補うようになってきた。つまり、言葉の代わりに視覚化された情報がコミュニケーションや文化の伝承媒体に追加されたと言える。

- 3) 情報量の大爆発の回避策として……現代文明と情報量の増加について、「すでに人間の適応能力を越えている」^[1]との指摘もある。物質社会から情報社会への移行、情報社会における人間と情報の問題、情報伝達におけるビクトリアル・メディア（漫画本、写真、テレビ等の絵を主体にして情報を伝えるもの）の持つ機能とその特性、情報の受容・伝達媒体に限界がきていると言える。米国のエール大学の教授がエール大学にある図書館の蔵書量と科学論文雑誌の数が、「18世紀の初めに約1000冊しかなかった蔵書がほぼ16年ごとに倍増している。そして、今後このまま増え続けると、2040年には2億冊になり、毎年新規に入ってくる1200万冊のカード作りのために、6000人の司書が必要になる」^[2]との予想をしている。さらに、知識の総量も生態的に大爆発したように急激に増えていることを、「1930年から1965年にかけて就労人口が50パーセントしか増えないのに、エンジニアと科学者の数が4倍と10倍とに増えている」^[3]で示している。このように、増加を続けている情報を上手に処理するにはメディアの情報を絵にし、電子化することになる。つまり、情報社会は「絵は言葉よりも多くを語ることができる」や、「百聞は一見にしかず」の諺のように、見せる（魅せる）情報が増えることになる。

2.2 コンピュータ・グラフィックスとレンダリング

コンピュータ・グラフィックスは頭（心）に描いたイメージを具現化する道具である。絵の具や鉛筆で画く絵がアナログな表現なら、コンピュータ・グラフィックスで作る絵はデジタル表現である。さらに、コンピュータ・グラフィックスは絵筆や鉛筆で描くように、マウスやスタイラスペン等の入力装置を用いて絵を画く二次元のペイント・システムと三次元の形状モデル（模型）を作り、光をあてた状態をシミュレーションして絵を作る方法がある。本稿では三次元の形状モデルに光を当て、リアルな二次元の絵（画像）を作ることをレンダリングと言う。レンダリングは隠線・隠面処理をし、物体の色付け処理を行う。この色付け処理をシェーディングと言う。

2.3 コンピュータ・グラフィックスの基礎理論上の位置付け

コンピュータ・グラフィックスは三次元の形状を表現するために、いくつかの基礎理論から成り立っている。まず、形状を創生、修正、参照するために、データの階層構造、位相構造、線・面演算、集合演算、局所変形制御等の幾何学的形状生成の知識が必要である。次に、それらを表示するために、展開処理、干渉処理、透視変換、レンダリング（隠線・隠面処理、シェーディング、マッピング）等の射影幾何学や熱伝導の知識も必要である。そして、これらの知識は古典幾何学、線形代数学、解析幾何学、位相幾何学、射影幾何学、さらにシェーディングはエネルギーを扱うためニュートン力学、量子力学、リーマン(Riemann)幾何学、放射熱伝導理論、超弦理論(Super string theory)、微積分学、確率統計学、情報科学に裏付けられる。そして、ガウス・ザイデ

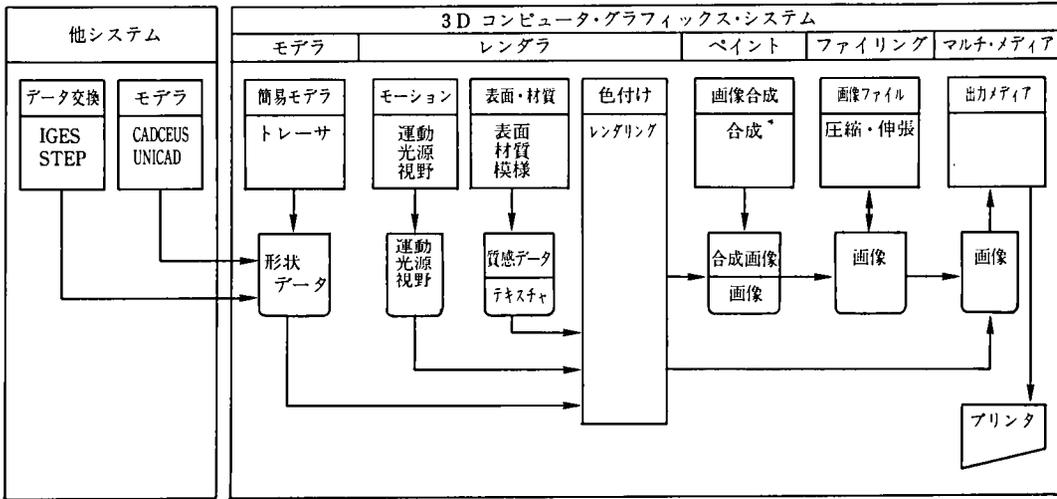


図 1 レンダリングの位置付け
Fig.1 The Location of rendering module

ル法 (Gauss-Siedel method) や漸進法 (Progressiv refinement approach) 等の数値計算手法が多用される。

三次元の形状を表現するコンピュータ・グラフィックス・システム (図1) を構成する機能には以下のものがある。

- ① 物体の幾何形状を作るモデラ
- ② 物体の運動や変化, 光源の大きさや性質, 視点の位置や視線の方向と角度, 物体の色や質感等の情報を参照してレンダリング (色付け) するレンダラ
- ③ レンダリングされた絵 (画像) と背景画像等と合成するペイント機能
- ④ でき上がった画像を後で利用するために, 検索情報を付加した画像のファイリング機能
- ⑤ 画像や文字情報を各種のメディアに出力する機能

本稿ではこの中のレンダリング機能, さらにその中のシェーディング理論について述べる。

2.4 コンピュータ・グラフィックスの代表的な理論

三次元のコンピュータ・グラフィックスに関わる代表的な理論を年代順に整理したのが表1である。

ワイヤ・フレーム, サーフェス, ソリッドの各形状モデラは隠線・隠面処理をして画像を表示する。隠面処理はシェーディング処理より, 物体の表面の色を得る。また, 隠面処理は大域照明モデル (Global illumination model) に反射, 屈折方向に物体があるかどうかを知らせる。ボリューム・レンダリングはシェーディングにより物体の色を得る。

表1 コンピュータ・グラフィックスの代表的理論
Table 1 The typical theory of the computer graphics

年代	形状モデラ				レンダリング						画像出力	
					隠線面処理		シェーディング					
	ワイヤ・フレーム	サーフェス	ソリッド	ポリウム	隠線・隠面	局所照明モデル	大域照明モデル	スムーズシェーディング	アンチエイリアシング	マッピング	合成	圧縮伸張
1963	ワイヤ・フレーム				隠線処理							
1965									ドレッセンハム			
1967		ポリゴン			スキャンライン	ランバード						
1969					優先順位							
1971						グロー		グローシェーディング				
1974		パラメトリック Bezier B-spline			Zバッファ					カラーマッピング		
1975			ソリッド			フォン		フォン・シェーディング				
1976										リフラクション リフレクション		
1977						プリン						
1978										バンパ		
1980		算術曲面 メタボール			レイ・トレーシング		レイ・トレーシング		ピットウエイ			
1982		手続き型モデル フラクタル パーティクル(霧霞)				クック・トランス						
1984					A バッファ		分配レイ トレーシング ラジオシティ				画像合成	
1985										ソリッド・テクスチャー		
1986							順方向レイ トレーシング				モンタージュ 法	G 3, G 4, JPEG
1988				ポリウム						ポリウム		

G 3 : Group 3, Group 4 ファクシミリで使用される圧縮伸張方法である。

JPEG : Joint Photographic Experts Group

3. レンダリングとシェーディング

レンダリングとシェーディングをことさら分ける必要もないが、レンダリングは精密描画を意味し、シェーディングは陰影付けを意味する。しかし、本稿ではレンダリングを図2のように三次元のモデルを二次元化、つまり絵(画像)にする処理であるとする。

そして、レンダリングから隠線・隠面処理を除いた残りをシェーディングと定義する(表1, 図2)。

3.1 シェーディングと色付け

最近、虹を見る機会があまりなくなった。虹は空中の雨粒が一つ一つのプリズムになってできた太陽スペクトルによって発生する。太陽の光は空中の雨粒に入り、雨粒の中で反射して出てくる。また、太陽の光は雨粒に入る時と出る時に雨粒の表面で光が屈折する。太陽の光は光の色によって屈折の角度が異なるから、色別に分解される。

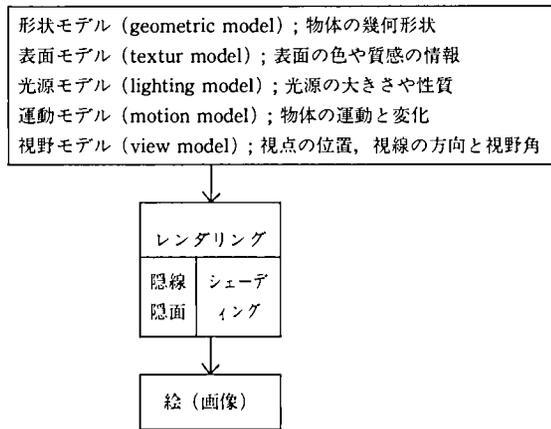


図 2 レンダリングの機能

Fig.2 The function of rendering

それがスペクトルである。「スペクトルの色は光の波長の違いで発生する」との認識は自然科学の常識になっている。しかし、色は光の周波数が異なる信号を捉え、その強弱を区別して感じる側の条件で決まると言える。つまり、色は光の波長による感覚である。このことから、物体がそこに存在しているか、いないか、の判断は可視光線スペクトルの範囲内の周波数で感知すれば存在していることになる。私達、人間の可視光線の波長は約 380-700 nm (単位：十億分の一メートル) である。物体が存在しているように見せるためには可視光線の波長で物体を通る光の反射、屈折をシミュレーションをすれば良いことに気がつくのにそう時間はかからなかった。

また、物質の根源を考えていた人々が原子 (atom) を見つけたのは 19 世紀に入ってからである。その原子からさらに素粒子へ進み、さらに基本的なクォーク (quark) で構成されていることがマリー・ゲルマン (ノーベル物理学賞を授賞) によって仮定された。クォークはアップ (up), ダウン (down), チャーム (charm), ストレンジ (strange), トップ (top), ボトム (bottom) の六つの種類がある。たとえば、陽子はアップ・クォークとダウン・クォークの組み合わせからできている。そして、このクォークには赤 (Red), 緑 (Green), 青 (Blue) の三色があり、観測される物質の色は白くなる。これが光の三原色である^[2]。

したがって、光の可視光線が赤、緑、青の波長から成り立っていて、その反射光によって物体の存在を示していることがわかる。

3.2 シェーディング理論

シェーディング理論は局所照明モデルから大域照明モデルへの進歩として捉えられる。局所照明モデルは、光源から物体に直接作用する現象だけを扱う世界で、スキャンライン法が代表的である。大域照明モデルは反射光だけでなく、透過光も考慮するレイトレーシング法や物体の相互反射を考慮するラジオシティ法が代表的である。

ここではシェーディング理論の中から、表 1 コンピュータ・グラフィックスの代表的理論に示した順で紹介する。初めに局所照明モデルの中から代表的なモデルとして、ランバート、フォン、プリン、クック・トーランス (全部人物名である) を紹介する。

次に大域照明モデルの中から代表的なモデルとして、レイ・トレーシング法、分配レイ・トレーシング法、順方向レイ・トレーシング法、ラジオシティ法（全部手法の名前である）を紹介する。そして、マッピング手法の中から代表的なテクスチャ、バンブ、ソリッド、リフレクション、リフラクション、マスク、属性の各マッピング手法を紹介する。最後に、ボリューム・レンダリングと最近注目されているプレビ面を紹介する。

シェーディング理論は、物体から目に入る光の成分が次の三種から成っていると仮定している。

- ① 鏡面反射光（スペキュラ：specular）：物体の表面から跳ね返された光でハイライトとも呼ばれる。
- ② 拡散反射光（デフューズ：diffuse）：光源からやってくる光が物体の表面や内部で乱反射されて目に入る。
- ③ 環境光（アンビエント：ambient）：物体の回りにある物体の照り返しや空気による散乱光である。太陽、白熱電球、発光体等がこれに入る。

光の強度（エネルギー）を照度（illumination）と言う。

照度 I は次の式になる^[9]。

$$I = K_s \cdot P_s + K_d \cdot P_d + K_a \cdot P_a \quad (3-1)$$

P_s ：鏡面反射光

P_d ：拡散反射光

P_a ：環境光

K_s ：鏡面反射係数

K_d ：拡散反射係数

K_a ：環境反射係数

($0.0 \leq K_s, K_d, K_a \leq 1.0$ の定数で物質によって異なる)

この基本式を各種の仮説に従って修正した式が発表されている。

3.2.1 局所照明モデル (Local illumination model)

スキャンライン法が扱うような、光源が物体に直接作用する現象だけを扱うので、局所照明モデルと呼ばれる。

- 1) ランバート (Lambert) のモデル……ぼんやりした光沢のない表面はすべての方向に等しく光を拡散反射させる。したがって、その表面はすべての視角から同じ明るさに見える。自然界に存在するほとんどの材質がこのような性質を持っている。紙の反射が代表的である。

ランバートは1967年に反射光の総量と光源の方向 L と表面からの法線 N の間の角度 θ の余弦 (cosine) を関係付けた。つまり、物体表面の任意の点で単位面積あたりに入射する光の量は面と光がなす角度に比例する。

余弦が負の時は光源と反対方向だから光量は0(零)になる(図3)。

光線ベクトル L と表面のベクトル N を正規化して内積で表現できる。

$$L \cdot N = \cos(\theta)$$

したがって、ランバートの式^[9]は式(3-1)の係数を次のように解釈したと言える。

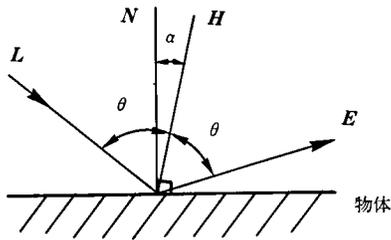


図 3 拡散反射

Fig. 3 Diffuse reflection

$$Kd = \max(0, L \cdot N)$$

$$Ks = 0$$

ランバートは鏡面反射を省略した。

光量は光源と面上の反射点との距離 R の二乗に反比例して弱まる。

$$Kd = \max(0, L \cdot N / R^2)$$

しかし、平行投影の無限遠にある光源では光源と反射点との距離 R も無限大になる。また、視点が物体に近づく場合は $1/R^2$ が零に近づく。この急激な値の変化はあまりにも極端な濃淡が付きすぎ、どぎつく見え、違和感がある。ちょうど夜中に懐中電灯を物体に当てたようになる。そこで、 R^2 を $R+k$ に置き換えた。 k は定数である。つまり、次のように改良された。

$$Kd = \max(0, L \cdot N / (R+k))$$

さらに、色の付いた物体からの拡散反射や照明光に色が付いているなら、三原色(赤・緑・青や水色・青紫色・黄色)要素の各色について、反射定数を代えて計算する。現在はリアルさに欠けるため使われない。

- 2) フォン (Phong) のモデル……フォン (Phong Bui-Tuong) は 1975 年に鏡面反射について経験的な事実を基に近似式を導き出した^[3]。つまり、鏡面反射光の分布は材質が違えば変わるため、材質感により鏡面反射モデルを使い分けることにより色々な材質を表現できる。

フォンのモデルは計算が簡単で、ハイライトがはっきり表現できるため、物体の表面のザラツキが少ないプラスチックのような質感を表現できる (図 4)。

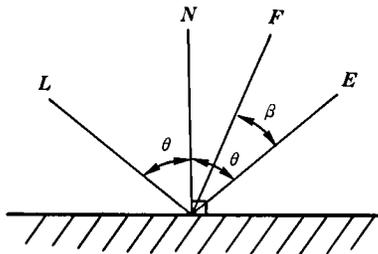


図 4 フォンのモデル

Fig. 4 Phong model

E : 視線ベクトル

F : 反射ベクトル

フォンのモデルは式(3-1)の係数を次のように表現できる。

$$Kd = \max(0, L \cdot N)$$

$$Ks = (E \cdot F)^n \quad 1 \leq n \leq 70 \quad n \text{ は } 50 \text{ から } 60 \text{ の間が多い。}$$

n はハイライトの広がりを変化させる係数である。整数であるため細かな調整はできない。

輝く物体は鏡面反射がハイライトを作り、白く見える。その色は入射光の色である。厳密な理論でないためか、金属感が表現しにくい弱点がある。

- 3) ブリン (Blinn) のモデル……ブリンは 1977 年にフォンの式を理論的に整えた。照明技術者が 1966 年に開発したトーランス・スパロウ (Torrance-Sparrow) の式と呼ばれるものである。ブリンは「物体の表面はランダムな方向を向いた微小平面の集合でできている」^[3] と考えた。つまり、物体の表面は完全な平面でなく、ざらついている (図 5)。それゆえ、反射の方向は広がりを持つ。鏡面反射光は本来の反射の方向とのずれ角度 α が小さいほど大きくなる。鏡面反射光の強さはこのずれを割合で決める (図 6)。

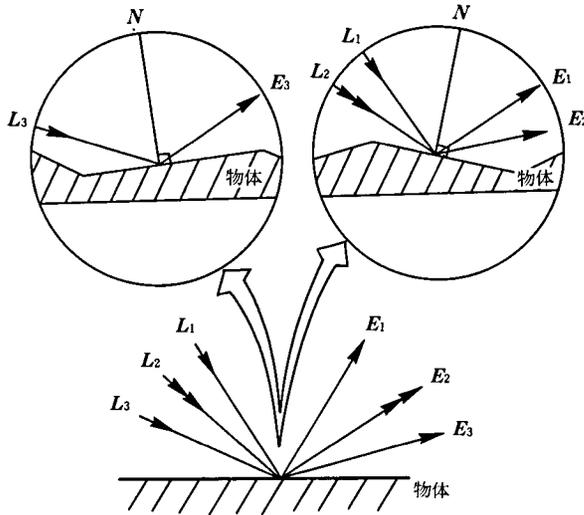


図 5 微小平面から反射する光線

Fig. 5 Light ray reflecting from surface microfaces

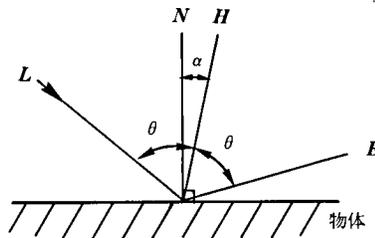


図 6 ブリンのモデル

Fig. 6 Blinn model

H : 光源の向きベクトル L と視点の向きベクトル E の二等分のベクトルである。ベクトル E は光源の反射方向で 1984 年にヘックバート (Heckbert) が提案した。

$$H = (E + L) / (|E + L|)$$

ブリンのモデルは式(3-1)の係数を次のように表現できる。

$$K_s = (D \cdot G \cdot F) / (N \cdot E)$$

ここで、 D は微小平面の向きを表現する散乱関数で、 H の向きを向いている微小平面の占めている割合である。散乱関数として

余弦関数..... $D = \cos^n \alpha$

ガウス確率密度分布関数... $D = a \cdot \exp(-(b \cdot a))^2$

回転楕円体関数..... $D = (c^2 / (\cos^2 \alpha (c^2 - 1) + 1))^2$

を使っている。

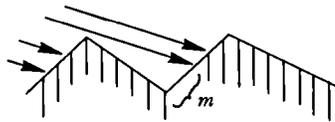
α は法線 N と微小平面の法線 H とのなす角度 $\alpha = \cos^{-1}(N \cdot H)$ 、 a は定数、 b は分布の標準偏差、 $c = H \cdot L$ である。

G は幾何減衰係数で微小平面が互いに隠される割合である。入射光が遮られる場合と反射光が遮られる場合がある。幾何減衰係数は次のように決める。

① ($N \cdot E \geq N \cdot L$) で

$((2(N \cdot H)(N \cdot L)) < (E \cdot H))$ なら入射光が遮られる (図 7)。

$$G = (2(N \cdot H)(N \cdot L)) / (E \cdot H)$$



m は入射光が遮られる範囲

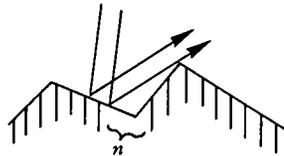
図 7 入射光が遮られる場合

Fig. 7 Partial intersection of incident light

② ($N \cdot E < N \cdot L$) で、

$((2(N \cdot H)(N \cdot E)) < (E \cdot H))$ なら反射光が遮られる (図 8)。

$$G = (2(N \cdot H)(N \cdot E)) / (E \cdot H)$$



n は反射光が遮られる範囲

図 8 反射光が遮られる場合

Fig. 8 Partial intersection of reflection light

③ その他なら, すべての反射光が目が届く (図9).

$$G=1$$

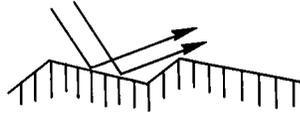


図9 遮られない場合

Fig.9 No interference

F はフレネル (Fresnel) 項で, 入射光と反射光の比の定数である。フレネル項は, 反射率が入射角や光の波長で変化する様子を表す式である。クック (Cook) が後に改良している (参照 3.2.1 の 4)。

- 4) クック・トーランス (Cook-Torrance) のモデル……クック・トーランスのモデルは 1982 年にプリンのモデルを改良した^[3]。プリンの散乱関数 D をベックマン (Beckmann) 関数にし, フレネルの定数をクックの式に改良した。

$$D=1/(m^2 \cdot \cos^4 \alpha)(\exp(-(\tan(\alpha)/m)^2))$$

m の値が小さい時はプリンの正規分布とほぼ同じである。大きいと反射分布が広がった状態を表す。 $0.1 \leq m \leq 0.7$ が良く使われる。さらに, フレネルの項を光の入射角で変化するようにした。

$$F=1/2 \cdot (g-c)^2 / (g+c)^2 \cdot (1+(c(g+c)-1)^2 / (c(g-c)+1)^2)$$

ここで,

$$c = \mathbf{E} \cdot \mathbf{H}$$

$$c^2 = n^2 + c^2 - 1$$

n は屈折率

散乱関数 D はこれ以外にもたくさん考えられている。たとえば, 凹凸のある金属表面やブラシ処理された指向性表面, 布地等の繊維表面を持った場合である。今後も, 関数 D は色々な特性を決めるために提案されるであろう。

3.2.2 大域照明モデル (Global illumination model)

局所照明モデルは光源が物体に作用する現象を扱ってきた。レイ・トレーシングは反射・透過レイを考え, 鏡面反射成分を大域照明としてシミュレーションする。分配レイ・トレーシングは新しい手法で, 動いている物体をブレたりボケた状態を表現する手法である。ラジオシティは物体表面の相互拡散反射を考慮した手法である。

- 1) レイ・トレーシング (Ray tracing) 法……レイ・トレーシングは光線追跡法と言われ, 視点から各画素に視線ベクトルを延ばし, ぶつかった物体の表面から反射と屈折方向に光線 (レイ) を延長する。そして, 次につつかった物体の表面でも同じことを繰り返す。この単純なことを繰り返して色を決めていく (図 10)。単純な正統派の手法であるが処理時間がかかるのが難点と言える。そのためレイ・トレーシングの歴史は計算の高速化手法を開発する歴史であった。高速化の手法は次のような方法に分類できる。

- ① 光線と物体の交差判定処理計算の高速化
- ② 光線と物体の交差判定回数の減少化による高速化
- ③ 光線の本数の減少化による高速化
- ④ その他

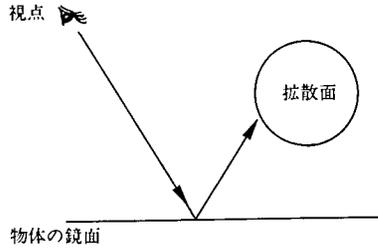


図 10 レイ・トレーシング法
Fig. 10 Ray tracing method

その他の代表的なものに次の二つがある。

- ビーム・トレーシング (Beam tracing)：ヘックバート (Heckbert) とハンラハン (Hanrahan) が 1984 年に発表した^[3]。視点を頂点とした四角錐を仮定してクリッピング処理を高速化した。ポリゴンのみ可能である。屈折の表現はできない。
 - 鉛筆トレーシング (Pencil tracing)：高橋、内藤が 1987 年に発表した^[3]。複数本の光線をまとめて交差判定を行う高速化を行っている。中心の一本の光線だけ計算し、その回りを相関関数を利用して求める手法である。
- しかし、次に示すように、正統派の計算手法も最近の速い計算機の出現で再評価されている。

$$\begin{aligned}
 I(x) = & ka(x)Ia(x) \\
 & + Kd(x) \sum (N \cdot Li) li(x) \\
 & + Ks(x)(Is(x) + \sum Fr(V, Li, N, a)li(x)) \\
 & + Kt(x)(It(x) + \sum Ft(V, Li, N, a)li(x))
 \end{aligned}$$

ここで、

$I(x)$ ：光の強度
 $Ka(x)$ ：周囲光係数
 $Kd(x)$ ：拡散光係数
 $Ks(x)$ ：反射光係数
 $Kt(x)$ ：透過光係数
 $Ia(x)$ ：周囲光強度
 $Id(x)$ ：拡散光強度
 $Is(x)$ ：反射光強度
 $It(x)$ ：透過光強度
 N ：面の法線

L_i : 光源ベクトル i

V : 視線ベクトル

a : 面の材質

$Fr(V, L_i, N, a)$: 鏡面反射関数

$Ft(V, L_i, N, a)$: 鏡面透過関数

x : 光の周波数

である。

- 2) 分散レイ・トレーシング (Distributed ray tracing) 法……レイ・トレーシングにぼかし表現を入れる手法である。1984年にクック (Cook) が発表した^[3]。ぼかしの表現はアンチエイリアシングを使う場合が多い。分散レイ・トレーシング法はぼかしを平均化と捉え、モンテカルロ法をレイ・トレーシングに適用し、物体の色の平均を近似して求める手法である。モンテカルロ法は乱数を使って積分値を求める方法である (図 11)。

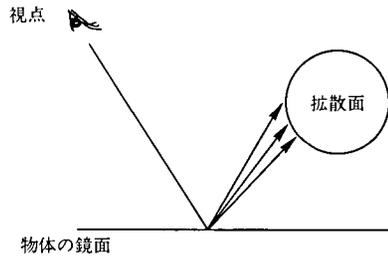


図 11 分散レイ・トレーシング法

Fig. 11 Distributed ray tracing method

- 3) 順方向レイ・トレーシング (Forward ray tracing) 法……つい最近まで逆方向レイ・トレーシング (Backward ray tracing) と呼ばれていた手法で、1986年にアーボ (Arvo) によって発表された。光源からの光線探索を順方向 (forward)、視点からの光線探索を逆方向 (Backward) とする呼び方に統一されるようで、最近では順方向レイ・トレーシングとして紹介されている。この手法の特徴は光源からの鏡面反射の影響を輝度計算に考慮する点にある (図 12)。

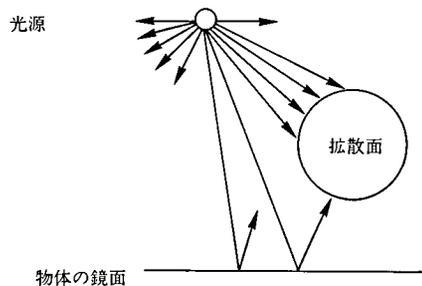


図 12 順方向レイ・トレーシング法

Fig. 12 Forward ray tracing method

- 4) ラジオシティ (Radiosity) 法……ラジオシティ法は放射熱伝導理論 (Thermal theory for radiative heat transfer: 物体相互の熱伝導作用の理論) を基にし、光のエネルギーの相互作用を仮定している。この理論を正確に理解するのはそれなりの専門知識が必要である。1981年にシーゲル (Siegel) によって色の決め方で提案され、1984年にゴラル (Goral), 1985年に中前, 1986年に鍛冶屋 (Kajiya) によってコンピュータ・グラフィックスで使われ、1988年にミチエール (Michael), シェンチェング (Shenchang), ジョーン (John), ドナルド (Donald) 等によって計算時間を削減した実用的な方法が発表された (図 13)。

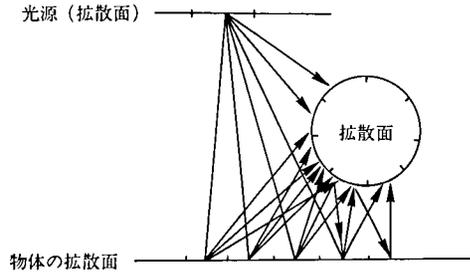


図 13 ラジオシティ法
Fig. 13 Radiosity method

この手法の特徴は光源と物体の区別がなく、観察者が光源と物体を見ることができる。そして、落ちついた色調の穏やかな表情の画像が表現できる。

次の式(3-2)は、エネルギー平衡状態にあるパッチ i (ポリゴン平面として良い) のラジオシティ・エネルギーを表す。エネルギーは単位面積、単位時間あたりの量である。式の第一項は自分自身の放射エネルギーで、第二項は自分自身以外のパッチから受けるラジオシティ・エネルギーの総量である。

$$B_i \cdot A_i = E_i \cdot A_i + p_i \sum_{j=1}^n B_j \cdot A_j \cdot F_{ji} \quad (3-2)$$

B_i : パッチ i のラジオシティ・エネルギー

E_i : パッチ i の放射エネルギー

A_i : パッチ i の面積

F_{ji} : パッチ j からパッチ i に達するエネルギーの割合

p_i : パッチ i の反射率

n : パッチ数

これをパッチ i とパッチ j の対称性を考慮して、

$$A_i \cdot F_{ij} = A_j \cdot F_{ji}$$

これを上記の式(3-2)に代入して、

$$B_i = E_i + p_i \sum_{j=1}^n B_j \cdot F_{ij} \quad (3-3)$$

さらに、パッチ i からパッチ j に向かって放射されるラジオシティ・エネルギーは

$$B_j = p_j \cdot B_i \cdot F_{ji}$$

であるので、さらに次のように改良できる。

$$B_j = p_j \cdot B_i \cdot F_{ij} \cdot A_i / A_j \quad (3-4)$$

この式の特徴は F_{ij} がなくなったことで、計算の作業領域が少なくて済むことである。

3.2.3 マッピング (Mapping)

マッピングの対象になる物体の任意の点とマッピング・データの対応点を求め、その点の色を処理することである。対応点はポリゴンの頂点を使い、頂点と頂点の間は補間で求める場合が多い。この手法は物体表面のざらつき (梨地等) や、反射率、屈折率、透明度等のデータをマッピングすることができる。

- 1) テクスチャ・マッピング (Texture mapping)……物体の表面に色や、模様の色を張り付ける。しかし、通常的手法では張り付けるデータ (画像データが多い) が物体座標系と一致しない。そのため、それらしく見せるだけの処理が多い (図 14)。

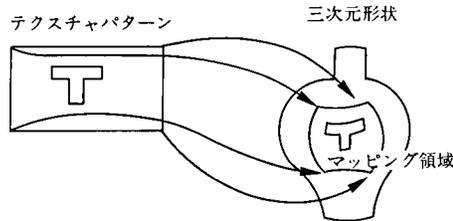


図 14 テクスチャ・マッピング

Fig. 14 Texture mapping

- 2) バンプ・マッピング (Bump mapping)……1981年に発表された。物体の表面の属性は色や反射率だけでなく、法線ベクトルを張り付けることで細かな凹凸を表現する手法である^[3]。しかし、表面に細かい凹凸は表現できるが物体の輪郭部分に凹凸は現れない。波 (wave) や波紋 (ripple) 等も表現できる (図 15)。

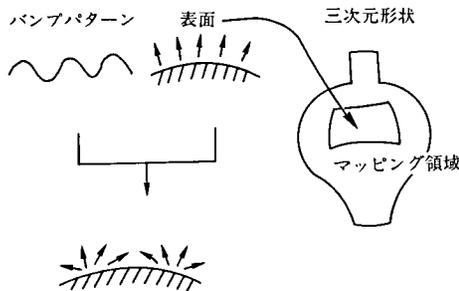


図 15 バンプ・マッピング

Fig. 15 Bump mapping

- 3) ソリッド・テクスチャ (Solid texture)……ピアチェ (Peachey) が 1985年に発表した^[3]。マッピング・データを三次元で持ち、任意の三次元の位置で色を求める関数で表現する。たとえば、バームクーヘン等の輪切りの途中の断面模様を表現する時に使われる。関数で表現できる物体に向いている。

- 4) リフレクション・マッピング (Reflection mapping)……金属の表面等の鏡面反射を疑似的に表現するためにブリン(Blinn)によって1976年に発表された手法である^[3]。対象となる物体を包含する一つの大きな仮想の球や立方体を考え、それに風景等をマッピングして置く。そして、視線ベクトルが当たった物体の表面からの反射ベクトルを仮想の球や立方体に延ばし、交点を求め、その点の色を求め、画素の色にする。この手法によってレイ・トレーシングの質感を短時間で疑似的に求められる(図16)。

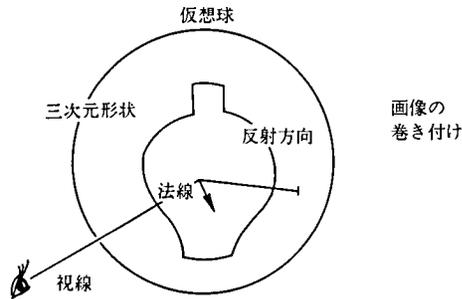


図16 リフレクション・マッピング

Fig.16 Reflection mapping

- 5) リフラクション・マッピング (Refraction mapping)……透明な物体を光が屈折して起きる歪を簡易的に表現するためにブリン(Blinn)によって1976年に発表された手法である^[3]。対象となる透明物体を包含する一つの大きな仮想の球や立方体を考え、それに風景等をマッピングして置く。そして、視線ベクトルが当たった物体の表面から屈折ベクトルを仮想の球や立方体に延ばし、交点を求め、その点の色を求め、画素の色にする。この手法によってレイ・トレーシングの質感を短時間で疑似的に求められる(図17)。

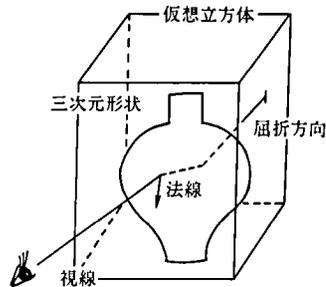


図17 リフラクション・マッピング

Fig.17 Refraction mapping

- 6) マスク・マッピング (Mask mapping)……物体の表面に、模様を領域で指定した範囲だけを使用して張り付ける手法である。この領域で指定した範囲の模様をマスクパターンと言う。この手法によって容器のラベルのマッピング等を行う(図18)。

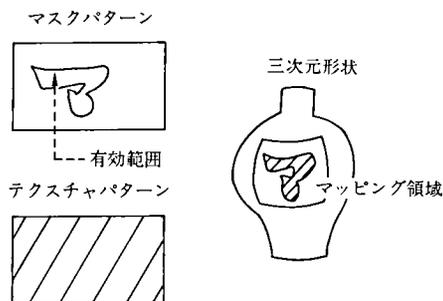


図 18 マスク・マッピング

Fig. 18 Mask mapping

- 7) 属性マッピング (Attribute mapping)……物体の表面に色情報や凹凸情報、光学情報の属性を張り付ける処理である。物体の表面の領域ごとにさまざまな材質感を表現できる。使用する画像をインデックス・パターンと言う。格子戸や欄間、石垣等の複雑で込み入った形状の表現に使われる (図 19)。

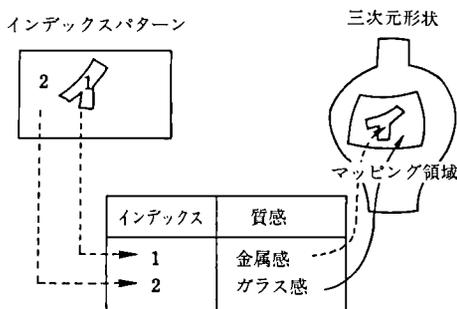


図 19 属性マッピング

Fig. 19 Attribute mapping

3.3 ポリューム・レンダリング (Volume rendering)

アップソン (Upson) によって 1988 年に発表された^[3]。雲や霧のような微粒子 (Particle) の集合体からできている物に光が入射し、微粒子層で反射、散乱が起きる。このようなモデルを表現する手法として発展した。たとえば、CT (cell thrapy) スキャンのデータを層状に持ち、三次元の任意の位置で切断した状態を表現する時に使う。

3.4 ブレビ (Blobby) 面

ブリン (Blinn) によって 1982 年に発表された。ブリンが発表した時はブラブ (Blob: 小さな球体) と呼んだが、メタボールの方が知れわたっている。メタボールはメタモルフォーゼ (変形する) という言葉を起源にしている。濃度球とも呼ばれる。1985 年に西村が発表したものと同じ方法である。メタボールは等ポテンシャル曲面集合である。1990 年に村木が距離の分布関数モデルとして発表している^[6]等ポテンシャル曲面とはポテンシャル (電位) を持った関数で定義され、ポテンシャルの値が等しい曲面である。最近ではモデラとしても評価が高く、この面データからカットパスを出

力し、加工まで行うシステムもある。テレビ番組の人体のアニメーション等で盛んに用いられている。しかし、モデリング中の表現が一般的にむずかしいので、利用されにくい原因になっている。

$$W(x, y, z) = \sum_{i=0}^n W_i(x, y, z) - T = 0 \quad (3-5)$$

$W(x, y, z)$: 等ポテンシャル曲面

$W_i(x, y, z)$: 分布関数 i

T : ポテンシャル面を決める値

通常、任意の点のポテンシャルはガウス分布の形で表される。

4. CADCEUS/CG の特徴 (レンダリングを中心に)

CADCEUS/CG は四つの機能から構成されている。三次元の形状モデルに色付け処理を行い、実物のように見せるレンダリング機能、レンダリングしてできた画像の修正やスキャナから読み込んだ背景との合成を行うペイント機能がある。また、このようにして作った画像の管理や、必要になった時に検索する画像ファイリング機能、画像の入力や出力を行う画像入出力機器を制御するマルチメディア・インタフェース機能である。

各機能を家を例に説明する。三次元形状モデラで作られた家は、ワイヤ・フレームの状態グラフィック・ディスプレイに表示される。また、面に色を付けて表示することもできる(口絵1)。しかし、さらに家らしく見せるためには屋根には瓦があり、壁には外装材が、窓枠や扉は木目模様で見えなければならない。これを実現するのがレンダリング機能の中のテクスチャ・マッピング(模様張り付け)である。そのため、瓦や外装材、木目のある板を写真に撮り、カラー・スキャナで読み込み、画像ファイルに画像として保存しておく。

つぎに、CADで作った家モデルの屋根部品の属性として瓦を登録し、壁の属性として外装材を、窓枠や扉の属性として板を指定して、レンダリング処理を行うと家らしくなる。属性して登録された瓦や外装材、板は実寸で家に張り付けられる。軒先の影も付く。そして、この家を建てる予定の空き地の写真をあらかじめ撮っておいて、レンダリング後の画像と重ね合わせ、立木や垣根をペインティング機能で植えれば、より一層の現実感が出る(口絵2)。空き地や立木、垣根の画像を数多く保存しておいて必要な画像を検索できれば、もっと良い効果が出せる。画像データは一画面分の画像で約4メガバイトになる。したがって、画像の圧縮・伸張の技術が必要である。さらに、それを必要な検索項目で検索できることが必要である。

CADCEUS/CG の代表的な特徴を次に示す。

4.1 CAD データと CG データの融合

データベースに物体と画像を混在して取り込むことができる。口絵2のレンダリングした家(CAD)とタイル(画像)や外装材(画像)、口絵4の瓶(CAD)とラベル(画像)、口絵5の電話機(CAD)と木目模様(画像)。

4.2 画像データと実寸

テクスチャ(模様)に実寸の概念がある。家の外装に張り付ける画像がタイルなら、

タイルのサイズや目地の実寸をデータベースに指示することで正確な枚数のタイルが張られる。口絵2のレンダリングした家 (CAD) とタイル (画像+実寸) や外装材 (画像+実寸), 口絵4の瓶 (CAD) とラベル (画像+実寸), 口絵5の電話機 (CAD) と木目模様 (画像+実寸)。

また、実寸でプリントアウトすることができる。

4.3 正確で高品質なレンダリング

画像が実寸で利用できるため、正確で高品質な画像ができる。口絵3は光線がガラスの物体 (楕円の厚みのあるガラス) にぶつかり、反射と屈折を繰り返す様子を三次元の線で表示した例である。光線が反射と屈折を繰り返して進む様子が良くわかる。レンズやランプ等の設計に役立つ。

4.4 高速な計算

高速に計算処理をして反射や屈折した様子を表現するために、一回だけ反射や屈折処理をさせ、その画像を物体に張り付ける方法がある。それがリフレクション・マッピングとリフラクション・マッピングである。その例を示す。口絵6のリフレクション・マッピングの例は、球に空と花畑が反射している場合である。口絵7のリフラクション・マッピングの例は、球が空と花畑を屈折させている場合である。正確な計算より約1/10の時間で処理できる。

4.5 ペイント (レイアウト, レタッチ) と画像合成

シェーディングで作られた画像の修正や背景との合成や文字による説明文も入れることができる (レイアウト)。色替えや加筆もできる (レタッチ)。仮想出力紙面は A1 (594 mm×840 mm) サイズまで可能である。

4.6 画像ファイリングと画像圧縮・伸張

画像ファイリングは SQL (構造化問い合わせ言語: Structured Query Language) コマンドを作り出し、画像データを検索するようになっている。個々の画像サイズを小さくするために各種のデータ圧縮・伸張をする機能を用意している。検索のための豆絵画像も利用できる。

4.7 各メディアと WYSIWYG

各種のメディア (プリンタ類) に最高の品質で出力するために、出力はすべてスクリプト言語でファイルし、各メディアがそのメディアに最もふさわしい分解能で出力している。WYSIWYG (What you see is What you get) を実現している。

4.8 プレビ面

プレビ面が取り扱える。CAD とは区別されていたがこのシステムでは同じように取り扱える。

5. 今後の課題

レイ・トレーシングのところでも述べたように、この分野の歴史は高速化手法の開発の歴史であった。その状況は今でも本質的には変わっていない。今、最も注目している手法は物体の表面の特徴線^[5]を利用する方法である。新しい方法で現在試行中である。まもなく結果が出る予定である。もう一つは並列コンピュータの利用である。これはネットワーク上のワークステーションに分散して処理させる粗結合型の並列処

理、並列コンピュータに分散して処理する密結合型の並列処理、のどちらでも選択できる仕組みを作り、それにより高速化を実現することである。これは現在一部の客先で試行中である。本物らしい表現をするための色合いの評価や写真のようなリアルな結果を出すために画像の質を再評価している。そして、入出力装置の色質の再現性を色度計で評価を行っている。この結果も近々出る予定である。もちろん、良い色調を出すためのフィルタリングの値も決まることになる。

これらの実験に検討を重ね、高速で良質な画像処理を実現することが課題である。

6. おわりに

ここで取り上げたコンピュータ・グラフィックスの技術は完成されたとは言えない。コンピュータ・グラフィックスは進歩の激しい分野であり、絶えず新しい技術が開発・発表されている。そして筆者等も、より、フォトリアリスティックな画像を作り出すために新しい手法を考案し続けたいと思う。本稿により、コンピュータ・グラフィックスの分野を知る機会を得、新しい技術や手法を相像できるようになっていただければ幸いである。

- 参考文献
- [1] 立花隆, 「文明の逆説」, 講談社文庫.
 - [2] B. マックスカー著, 宮崎忠訳, 「クオークをさがす」, ブルーバックス.
 - [3] J. D. Foley, Computer Graphics (SECOND EDITION), Addison-Wesley Publishing Company.
 - [4] 日経エレクトロニクス, ビジュアライゼーション特集, 1989.3.6 (no. 468) pp. 105~138.
 - [5] A. S. グラスナー著, 白田耕作監訳, 「3D コンピュータ・グラフィックス」, アスキー出版局.
 - [6] 村本茂, 「距離データの分布関数モデル表現に関する研究(1990.8.17)」, 電子技術総合研究所, プラフィックスとCAD.
 - [7] 東正毅, 穂坂衛, 「レンダリングにおける特徴線の定式化と表示」, 1990年度精密工学会春期大会論文集, 1043.

執筆者紹介 大平昌明 (Masaaki Oohira)

昭和45年日本大学文理学部応用数学科卒業, 同年日本ユニシス(株)入社. CAD/CAMシステムの開発, 適用に従事, 容器デザイン・システム, 自動車設計システム, UNICAD/SOLID, CGシステムの開発業務に従事, 現在に至る.



境界要素法による金型冷却解析システム

A Mold Cooling Analysis System Using the Boundary Element Method

菅 沼 正 之, 平 林 繁

要 約 プラスチックの射出成形や軽合金のダイカスト鑄造における金型への充填解析（流動解析）に続き、成形品の効率的かつ均一な冷却を目ざして金型冷却解析システムを試作した。金型の設計段階ではさまざまな冷却管路の配置が検討されるが、管路の変更による金型内部の要素再分割作業を避けるために、離散化手法として境界要素法を採用した。

また、冷却解析用に新たにモデルを作成せず、充填解析用のシェルモデルを流用する目的で、システム内でシェルモデルから簡易的なソリッドモデルを作成するサブシステムを開発した。

従来からの要素分割の煩雑な有限要素法による冷却解析システムや、冷却管への一次元熱伝導計算による簡略システムと比較して、今回開発したシステムは解析の精度を保ちつつ、要素再分割作業を必要としない、使用者にとって使いやすい実用化システムである。

Abstract As a continued effort by the authors after creation of a mold filling analysis system (which falls under the category of flow analysis) for plastic mold injection and light metal alloy diecasting, a prototype version of mold cooling analysis system has been developed in an attempt to make it possible to cool products in an effective, uniform way. Although a variety of cooling pipe layouts are studied at the stage of mold base designing, the desire to avoid regenerating a finite element model for rearranging a cooling pipe layout encouraged the adoption of the boundary element method (BEM) as a discretization scheme.

Instead of creating a new model for cooling analysis, and for the purpose of applying a shell model used for filling analysis to a cooling analysis model, the authors have developed a subsystem by which to convert a shell model to a simple solid model.

Compared with traditional cooling analysis systems based on the finite element method (FEM), which require remaking a finite element model by rearranging of cooling pipe layouts and with a simplified method for which a 1D thermal transfer model is used, the mold cooling analysis system developed this time has an advantage of being easy-to-use and retaining the same level of analysis accuracy as an applicable system which does not require the regeneration of a finite element model.

1. はじめに

高機能樹脂の開発や、部品の軽量化の要請からプラスチック射出成形は、新素材の重要な成形加工技術の一つである。射出成形の工程は、概略成形機シリンダ内での樹脂の熔融、熔融された樹脂の金型内への射出・充填、金型による樹脂の冷却・固化に分かれる。その中で、冷却・固化の過程は製品の品質を左右し、射出成形プロセス全体の生産性に大きな影響を及ぼす。通常プラスチック射出成形では金型内に冷却用の管路を配置して、高温の樹脂から、金型を経由して、熱を管内流体へ伝達し、金型外へ排出している。冷却管路の配置によっては、樹脂と接する金型表面の温度が製品全体でほぼ均一とはならず、温度分布を持つことになる。不均一な温度分布は成形品の

そり・ひけの変形，寸法精度の不安定性，外観不良の原因となる。また生産性の面からは，成形サイクルの中で冷却過程の占める割合が一番大きく，効率的な冷却は生産性向上にとって必須条件でもある。従来冷却管の設計は，長年培われた経験と勘により行われてきたが，冷却の前段階である充填過程において，コンピュータによるプラスチック流動解析^[1]が大きな成果を挙げている現在，冷却過程にも CAE 技術を適用しようという試みが，内外の大学，研究機関で活発に行われている。

一方，金属の成形加工技術の一つである鋳造の分野では，プラスチック射出成形と同様に，軽金属（アルミニウム，マグネシウム，亜鉛）のダイカスト鋳造で，金型内に冷却管路を配置して，熱を金型から奪う冷却方法が採られている。

金型冷却解析を数値解析の立場から眺めると，金型の熱伝導，熱伝達解析であり，コンピュータによる各種解析の中でも成熟した技術の一つであるが，従来有限要素法 (FEM) による熱伝導解析を金型冷却へ適用した例は少ない。その最大の理由は，FEM を使用した場合，解析領域（金型全体）を三次元要素（ソリッド要素）に分割しなければならないが，とくに管路の設計では，冷却管の配置を変更することに要素分割をやり直す必要があり，実用の面から適用がむずかしかった。この欠点を克服したのが境界要素法 (BEM) による金型冷却解析である。周知のように，境界要素法では解析領域の内部を要素分割する必要はなく，境界表面の分割のみで計算される。したがって三次元解析は二次元解析へ次元を下げることができる。金型冷却の場合は大気と接する金型表面，製品と接する金型表面，および冷却管と接する金型表面のみを要素分割する。したがって冷却管の配置を変更しても，管路表面のみの要素分割のみで再計算が可能となる。

過去において開発された金型冷却解析システムには，Moldflow Pty 社（オーストラリア）による樹脂表面から，冷却管への一次元熱伝導を解く簡略解析システム (MOLD COOL)^[2]，境界要素法では，射出成形プロセスの研究で著名なコーネル大学（アメリカ）の CIMP プロジェクトで試作された例^[3]，国内では豊田中央研究所が BEM を用いたシステムを発表しているが，金型を固定側と可動側に分けて冷却用モデルを作成している^[4]。

今回試作した金型冷却解析システムは，流動解析用に作成した製品モデルを利用して，金型の外形形状と冷却管モデルを追加することで解析できるように工夫した。また，流動解析モデルを利用する時に問題となるのが，通常プラスチック流動解析は二次元的な板要素（シェル要素）でモデリングされる点である。BEM を適用するには，三次元の外形形状が必要であるため，板要素モデルからシステム内で自動的に簡易的なソリッドモデルを作成することにした。これによって，金型冷却のために，特別なモデルを作成する必要はなく，使用者にとって使い勝手のよいシステムとなった。

本稿では，境界要素法の基本概念，金型冷却解析への適用，および解析例について詳述する。

2. 境界要素法による金型冷却解析

射出成形の全工程を伝熱の観点から見ると，成形開始後に金型は過渡的な温度上昇を示し，その後金型全体は一定の温度に収束していく。詳細に見ると，成形品に近い

金型部分は、平衡的な温度のまわりで、成形の各工程により微少な温度上昇、下降を繰り返している。したがって金型全体の温度分布は平衡的な温度分布とその周りの摂動分布に二分できる。

今回試作したシステムでは、まず充填完了時の成形品温度分布と冷却管の配置、温度と金型外表面の温度から、金型全体の平衡的な温度分布を計算する。したがって熱伝導解析としては、金型の定常的な温度分布を求めることになり、ラプラス方程式によって記述されるポテンシャル問題へ帰着する。

2.1 問題の定式化

射出成形金型は内部に樹脂が充填される製品形状をしたキャビティと呼ばれる空洞と、冷媒を流すための冷却管と呼ばれる管状の空洞を内部に持った直方体のブロックと考えることができる。図1は金型とその内部のキャビティおよび冷却管を二次元的に模式図で示したものである。あとの説明のために金型内部の領域を Ω 、金型のキャビティ側の境界を Γ_c および冷却管側の境界を Γ_p 、金型の雰囲気側の境界を Γ_m とする。境界 Γ_c 、 Γ_p および Γ_m は互いに重なり合わないことを仮定している。

定常状態の金型の温度 u は (2-1) のラプラス方程式および境界条件 (2-2)、(2-3)、(2-4) で記述される。

$$\nabla(k\nabla u(P))=0 \quad P \text{ in } \Omega \quad (2-1)$$

$$u(P)=\bar{u}_p \quad P \text{ on } \Gamma_p \quad (2-2)$$

$$u(P)=\bar{u}_m \quad P \text{ on } \Gamma_m \quad (2-3)$$

$$-k \frac{\partial u}{\partial n}(P)=\bar{q}_c \quad P \text{ on } \Gamma_c \quad (2-4)$$

ここで(2-1)の ∇ はナブラ演算子である。また、(2-1)、(2-4)の k は金型の熱伝導率を示している。境界条件は、冷却管と金型の雰囲気側ではそれぞれ(2-2)および(2-3)で示されるように温度が指定される。また、キャビティ側の境界では(2-4)で示されるように熱流束が指定される。境界条件(2-2)および(2-3)のように、境界上での未知関数の値が指定される場合をディリクレ境界条件という。また、(2-4)のような未知関数の法線方向の勾配が指定される場合をノイマン境界条件という。金型冷却解析のよう

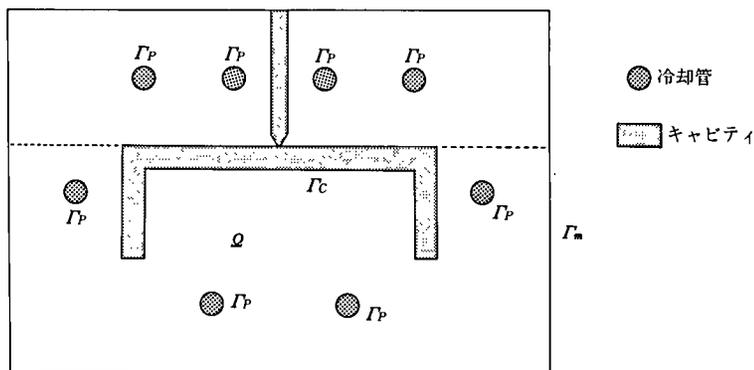


図1 射出成形金型の単純化モデル

Fig. 1 Simplified model of mold base for mold injection

に、ディリクレ境界条件とノイマン境界条件が同時に系に課せられている場合を混合境界条件という。

2.2 境界積分方程式^[5]

境界要素法で金型冷却問題を定式化するためには、与えられたポテンシャル問題を境界積分方程式に変換することが必要となる。境界積分方程式に変換する第一歩として(2-1)で与えられたラプラスの方程式の随伴方程式(2-5)とその解である基本解 $u^*(P)$ を考える必要がある。随伴方程式(2-5)は三次元空間全域で定義されている。

$$\nabla^2 u^*(P) + \delta(P, Q) = 0 \quad (P, Q \text{ in } \Omega = R^3) \tag{2-5}$$

ここで、 $\delta(P, Q)$ はディラックのデルタ関数と呼ばれ、次のような性質を持つ。

$$\int_{\Omega} f(x) \delta(x, y) dx = \begin{cases} f(y) & y \in \Omega \\ 0 & y \notin \Omega \end{cases} \tag{2-6}$$

三次元空間での(2-5)の解 $u^*(P)$ は具体的には(2-7)のようになる。ここで、 $\|PQ\|$ は点 P と点 Q 間のユークリッドの距離を意味している。

$$u^*(P, Q) = \frac{1}{4\pi\|PQ\|} \tag{2-7}$$

2.2.1 内点の場合の境界積分方程式

Q が Ω の内点である場合の点 Q での温度 $u(Q)$ に対する境界積分方程式を求める。簡単のために、方程式の定義される領域 Ω を単純化し、図 2 (a) のような、なめらかな境界 Γ を持つと考える。領域 Ω の中に点 Q をとり、 Q を中心とする半径 ϵ の球を考えこれを S_ϵ とする。また、領域 Ω から球 S_ϵ を取り除いた領域を Ω_ϵ とする(図 2 (b))。次に、 Ω_ϵ の中で Green の公式(2-8)を考える。(2-8)において u をポテンシャル問題の解、 v を基本解 u^* と見ると、(2-9)が得られる。

$$\int_{\Omega} \{ \nabla(k\nabla u)v - ku\nabla^2 v \} d\Omega = \int_{\Gamma} \left\{ k \frac{\partial u}{\partial n} v - ku \frac{\partial v}{\partial n} \right\} d\Gamma \tag{2-8}$$

$$\int_{\Gamma + S_\epsilon} k \frac{\partial u}{\partial n} u^* d\Gamma - \int_{\Gamma + S_\epsilon} ku \frac{\partial u^*}{\partial n} d\Gamma = 0 \tag{2-9}$$

(2-9)で、 $\epsilon \rightarrow 0$ の極限をとると、 Q が内点である場合の温度 $u(Q)$ の評価式(2-10)を

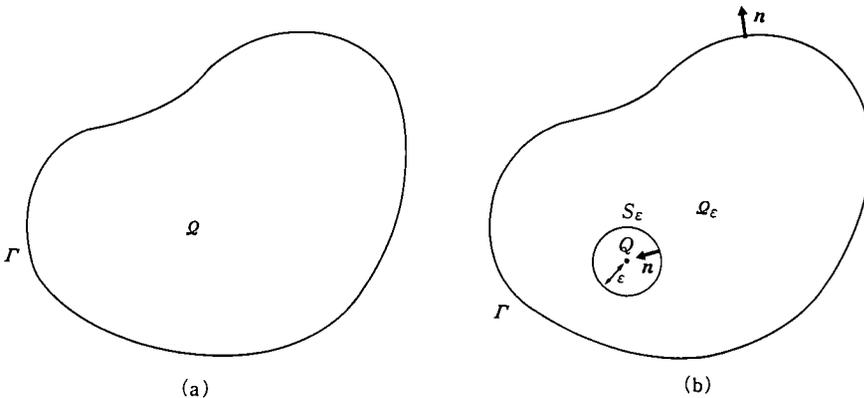


図 2 内点の積分形式を導くための領域の変形

Fig.2 Modification of region for potential evaluation at an inner point

得ることができる。

$$ku(Q) + \int_{\Gamma} ku \frac{\partial u^*}{\partial n} d\Gamma = \int_{\Gamma} k \frac{\partial u}{\partial n} u^* d\Gamma \quad (2-10)$$

ここで、点 P についての u^* の法線方向微分を(2-11)のおく、境界上の温度 u および熱流束 q と内点の温度 u の関係を表す境界積分方程式(2-12)を得る。(2-12)は境界 Γ 上のすべての点で温度 u と熱流束 q の値が既知であれば、領域 Ω の任意の内点 Q での温度 $u(Q)$ を計算できることを意味している。

$$q^*(P, Q) = -\frac{\partial}{\partial n} u^*(P, Q) \quad (2-11)$$

$$ku(Q) - \int_{\Gamma} ku(P)q^*(P, Q)d\Gamma = -\int_{\Gamma} q(P)u^*(P, Q)d\Gamma \quad (2-12)$$

2.2.2 境界上の点に対する境界積分方程式

点 Q が境界上にある場合の関係を導く。図3で示すように、点 Q を境界 Γ 上にとり点 Q を中心とする半径 ϵ の半球を追加することによって領域 Ω とその境界 Γ を変形する。半球上の表面を Γ_{se} で表し、それ以外の境界部分を Γ_{ϵ} で表す。また、半球が追加された領域全体を $\Omega_{\epsilon+}$ で表すことにする。

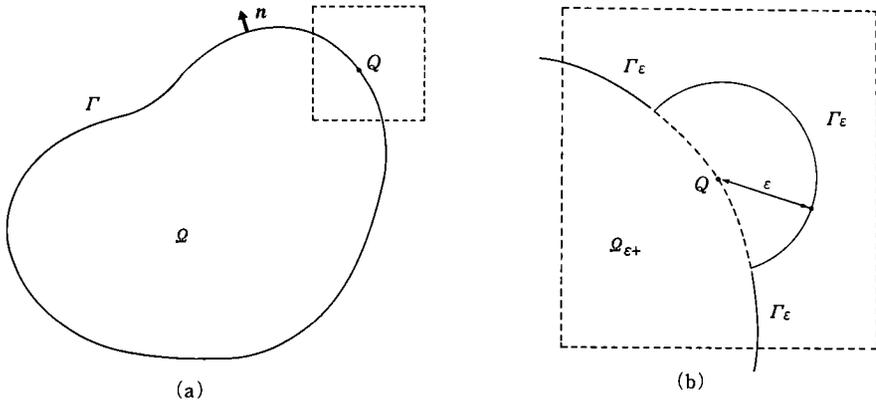


図3 境界上の点の積分形式を導くための領域の変形

Fig.3 Modification of region for potential evaluation at a boundary point

領域 $\Omega_{\epsilon+}$ においては、点 Q は内点であるから(2-12)が成り立つことになり、すなわち(2-13)のように書くことができる。

$$ku(Q) - \int_{\Gamma_{\epsilon} + \Gamma_{se}} ku(P)q^*(P, Q)d\Gamma = -\int_{\Gamma_{\epsilon} + \Gamma_{se}} q(P)u^*(P, Q)d\Gamma \quad (2-13)$$

$\epsilon \rightarrow 0$ の極限をとると、 Q が境界上の点である時の温度 $u(Q)$ の評価式(2-14)が得られる。ただし右辺の積分はコーシーの主値の意味で解釈する。

$$\frac{1}{2}ku(Q) - \int_{\Gamma} ku(P)q^*(P, Q)d\Gamma = -\int_{\Gamma} q(P)u^*(P, Q)d\Gamma \quad (2-14)$$

2.3 境界積分方程式の離散化

金型冷却解析では、金型のキャビティ側の境界での温度分布を知ることが大きな課題となる。このため、(2-14)の離散化を行い数値的に近似解を得ることを考える。

金型の雰囲気側の境界 Γ_m 、キャビティ側の境界 Γ_c および冷却管側の境界 Γ_p の和

を Γ とおく。離散化のために境界 Γ を小さな要素に分割する。ここで、この時のキャビティ上の節点の数を l 、冷却管上の節点の数を m および金型の雰囲気側境界上での節点の数を n とする。形状関数 $N_i(P)$ を適当に選び、形状関数の張る関数空間の中で $u(P)$ および $q(P)$ を評価する。すなわち、形状関数 $N_i(P)$ の線型結合で $u(P)$ および $q(P)$ を表現すればよい。

$$u(P) = \sum_{i=1}^N N_i(P) u_i \quad (2-15)$$

$$q(P) = \sum_{i=1}^N N_i(P) q_i \quad (\text{ここで、} N=l+m+n) \quad (2-16)$$

ここで、 u_i および q_i は境界点 P_i における解 $u(P_i)$ および $q(P_i)$ の近似値である。(2-15) および (2-16) を (2-14) に使用すると境界上の N 個の節点 $Q(=P_i)$ について (2-17) を得る。

$$\frac{1}{2} k u_i - \int_{\Gamma} k N_j(P) q^*(P, P_i) d\Gamma \cdot u_j = - \int_{\Gamma} N_j(P) \cdot u^*(P, P_i) d\Gamma \cdot q_j \quad (2-17)$$

各係数を (2-18) のようにおくと離散化された境界積分方程式 (2-19) が得られる。(2-18) の δ_{ij} はクロネッカーのデルタであり、 h_{ij} および g_{ij} は「 H 」行列および「 G 」行列の成分を示す。

$$\left. \begin{aligned} h_{ij} &= \frac{1}{2} k \delta_{ij} - \int_{\Gamma} k N_j(P) q^*(P, P_i) d\Gamma \\ g_{ij} &= - \int_{\Gamma} N_j(P) u^*(P, P_i) d\Gamma \end{aligned} \right\} \quad (2-18)$$

$$\begin{aligned} [H]\{u\} &= [G]\{q\} \\ \{u\} &= \{u_1, \dots, u_l, u_{l+1}, \dots, u_{l+m}, u_{l+m+1}, \dots, u_{l+m+n}\} \\ \{q\} &= \{q_1, \dots, q_l, q_{l+1}, \dots, q_{l+m}, q_{l+m+1}, \dots, q_{l+m+n}\} \end{aligned} \quad (2-19)$$

金型冷却解析のような混合型境界条件の問題では、すべての境界上の節点に対して温度 u_i または熱流束 q_i のどちらか一方の値が既知である。この既知の u_i および q_i の成分を (2-19) のベクトル $\{u\}$ および $\{q\}$ に反映させ、左辺に未知成分が右辺に既知成分が集まるように変形することによって一次方程式 (2-20) を得る。具体的には、ベクトル $\{q\}$ とベクトル $\{u\}$ の未知の q_i と既知の u_i を入れ替え、対応する $[H]$ 行列と $[G]$ 行列の i 列を符号を反転させて入れ替えればよいことになる。金型冷却解析では、冷却管の境界と金型の雰囲気側境界では q_i は未知であり u_i は既知である。このため、冷却管と金型の雰囲気側境界の成分に対してのみこの処置を行えばよい。このようにして作成された係数行列 $[A]$ は非対称な密行列となる。

$$\begin{aligned} [A]\{x\} &= [B]\{b\} \\ \{x\} &= \{u_1, \dots, u_l, q_{l+1}, \dots, q_{l+m}, q_{l+m+1}, \dots, q_{l+m+n}\} \\ \{b\} &= \{\bar{q}_1, \dots, \bar{q}_l, \bar{u}_{l+1}, \dots, \bar{u}_{l+m}, \bar{u}_{l+m+1}, \dots, \bar{u}_{l+m+n}\} \\ [A] &= [h_1, \dots, h_l, -g_{l+1}, \dots, -g_{l+m}, -g_{l+m+1}, \dots, -g_{l+m+n}] \\ [B] &= [g_1, \dots, g_l, -h_{l+1}, \dots, -h_{l+m}, -h_{l+m+1}, \dots, -g_{l+m+n}] \\ &(\text{ここで、} h_i, g_i \text{ は各々 } [H] \text{ 行列, } [G] \text{ 行列の第 } i \text{ 列を示す。}) \end{aligned} \quad (2-20)$$

3. システムの概要

金型冷却解析の処理流れを図 4 で示す。解析システムは大きく分類すると、金型境

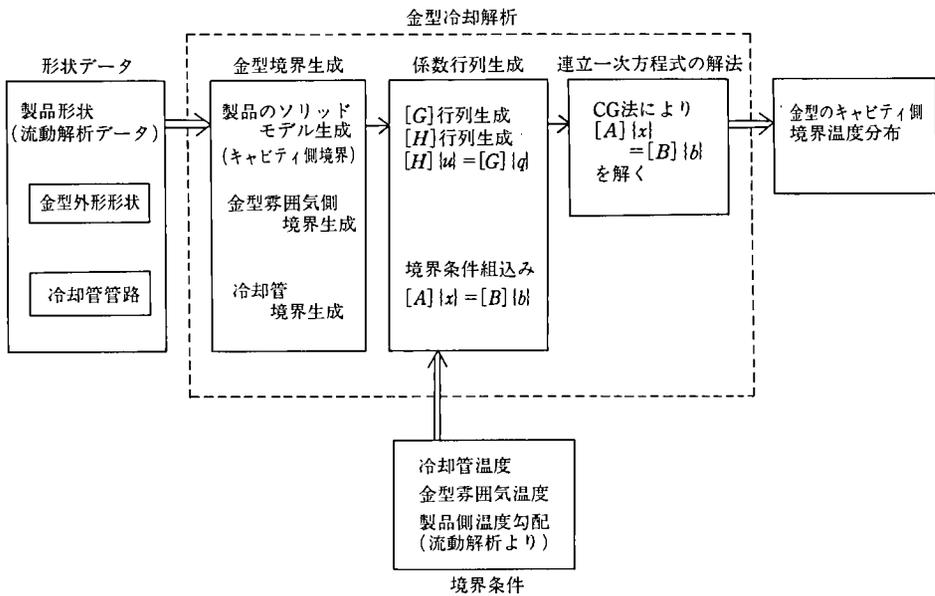


図 4 金型冷却解析の処理の流れ

Fig. 4 Flow chart of mold cooling analysis

界生成系、係数行列生成系および連立一次方程式の解法の三つの部分から構成されている。まず始めに、形状データから金型境界の生成を行う。次に、生成された金型境界の要素上での数値積分により [G] 行列, [H] 行列の要素 g_{ij} , h_{ij} を計算する。これらの行列に対して、使用者が指定した金型の製品側境界の温度勾配, 冷却管および金型の雰囲気側の温度を使用して境界条件を組み込み連立一次方程式を得ることができる。このとき、流動解析の計算結果から得られる温度勾配を冷却解析に反映させることもできる。この連立一次方程式を解くことによって、金型のキャビティ側境界の温度分布と冷却管の熱流束が得られる。

3.1 金型境界の生成

形状データは製品形状, 金型外形および冷却管管路の三つの部分から構成されている。このうち製品形状に関しては、流動解析で使用したシェルモデルを流用する。金型外形は六面体要素を一つ追加することで定義され, 冷却管管路に関しては線要素を追加することで定義される。冷却管の直径は線要素の属性として指定する。

キャビティの境界要素は、流動解析で使用した製品形状をもとに生成する。金型冷却解析ではソリッドの製品形状を必要とするため, 内部でシェルモデルを簡易ソリッドモデルへ変換しキャビティの境界要素を生成している。金型の雰囲気側の境界要素は、使用者が追加した六面体要素の表面を分割することで生成する。また, 冷却管の境界要素は使用者が追加した線要素と冷却管の直径をもとに, 線要素の周囲に生成している。

3.2 係数行列の生成

本システムでは, 冷却管と金型の雰囲気側の境界上では有限要素として四角形を使

用し、形状関数として一定要素を使用している。また、キャビティの境界上では有限要素として三角形を使用し、形状関数として面積座標を用いた線形補間関数を使用している⁶⁾。

積分公式は三角形要素上では頂点对称性を有する三角形の数値積分公式⁷⁾を使用しており、四角形要素上では通常のガウスの積分公式を使用している。[H] 行列、[G] 行列双方とも対角項の要素の積分は特異点を含む積分となり、高精度の積分公式の使用が原則となる。ただし、[H] 行列の対角項に関しては直接数値積分を行うことなしに(3-1)の関係を使用することで求めることができる⁸⁾。これは、(2-19)で境界上の温度の値がすべて等しければ境界上の熱流束の値は0になることに由来している。

$$h_{ii} = - \sum_{i \neq j}^N h_{ij} \quad (3-1)$$

境界要素法は取り扱う次元が一つ下がるため、有限要素法と比較すると行列生成時のメモリ使用量、計算時間等の点で有利である。しかし、モデルが大規模になると高次数値積分による密行列の生成にはかなりの計算時間を要する。通常、適切な管路の配置を得るために何種類かの案が試みられることになる。[G] 行列や [H] 行列の要素の中には、冷却管の配置や金型の外形には依存しない部分もある。このため、最初の計算の時にこの部分をファイルに保存しておき、冷却管の配置や金型の大きさを変更した2回目以降の計算では、このファイルを使用することで行列生成に必要な計算時間を減らしている。

3.3 非対称行列への共役勾配法の適用⁹⁾

連立一次方程式の解法は直接法と反復法に大別される。モデルの規模が大きくなった場合を考慮して、ここでは反復法の一つである共役勾配法を解法として選択した。共役勾配法は係数行列が正定値対称行列である時に限って適用ができる。しかし、境界積分方程式の離散化で生じる行列 [A] は非対称であるので、原則としては共役勾配法は適用できないことになる。ところが、行列 [A] が非対称であってもその転置行列 [A]^t との積 [A][A]^t または [A]^t[A] は対称である。そこで、(2-20)の両辺に左側から [A]^t を乗じた(3-2)をつくり、これを解くことにする。

$$[\tilde{A}]\{x\} = \{\tilde{b}\} \quad \text{ここで} \quad (3-2)$$

$$[\tilde{A}] = [A]^t[A], \{\tilde{b}\} = [A]^t[B]\{b\}$$

ここで、[A]^t を先に計算して保存しておき [A]^t[A]{x} を n 回計算する場合と、[A]^t([A]{x}) を n 回計算する場合の計算量を比較してみる。n 次元正方行列に関する演算量を評価する場合の単位として n 次元ベクトルの内積の演算量を選択する。まず、[A]^t を先に計算する場合を考える。[A]^t の要素の計算は1回の n 次元ベクトルの内積の演算によって計算される。[A]^t の要素は n² 個あるため、[A]^t を作るのに必要な計算量は n² と評価される。行列とベクトルの積 [A]^t{x} は n 回の内積を計算することが必要となる。このため、[A]^t{x} を n 回評価するのに必要な演算量は n² である。よって全体では 2n² の演算量が必要である。次に、[A]^t([A]{x}) を n 回評価する場合の演算量を考える。[A]^t([A]{x}) は1回の計算あたり、行列とベクトルの積を2回計算する必要があるので必要な演算量は 2n である。n 回評価するためには、全体として

```

k=0;
{x}_0を設定する (初期値);
{r}_0=[A]^t ([B]{b}-[A]{x}_0);
{p}_0={r}_0;
while 収束条件が満足されない do
begin {y}_k=[A]^t ([A]{p}_k);
      
$$\alpha_k = \frac{({r}_k, {p}_k)}{({p}_k, {y}_k)};$$

      {x}_{k+1}={x}_k +  $\alpha_k$ {p}_k;
      {r}_{k+1}={r}_k -  $\alpha_k$ {y}_k;
      
$$\beta_{k+1} = -\frac{({r}_{k+1}, {y}_k)}{({p}_k, {y}_k)};$$

      {p}_{k+1}={r}_{k+1} +  $\beta_k$ {p}_k;
      k=k+1;
end;
```

図 5 非対称行列のための共役勾配法

Fig. 5 Conjugate gradient method for unsymmetric matrix

$2n^2$ の演算量が必要である。 n 回の繰り返しを仮定した場合両者の演算量は等しいことになるが、繰り返しが n よりも小さい場合は $[A]^t([A]\{x\})$ を n 回計算する方が演算量が少ない。このため、計算中に $[\tilde{A}]\{x\}$ が必要になった場合、 $[A]^t([A]\{x\})$ という形式で計算する。具体的な手順は図5のようになる。

収束を早めることは反復法の重要な課題である。スケーリングは収束を早めるための有効な手段である。 $[G]$ 行列の要素は基本的には r^{-1} を積分することによって計算され、 $[H]$ 行列の要素は基本的には r^{-2} を積分することによって得られる。このため、 r が大きい領域では行列の要素のオーダーがかなり異なることになる。 $[A]$ 行列は $[G]$ 行列と $[H]$ 行列の列の交換をすることによって生成される。このため、 $[A]$ 行列の要素の絶対値は大きく異なることになり、 $[\tilde{A}]$ 行列の絶対値はさらに大きく異なる。ここでは、 $[\tilde{A}]$ 行列を直接計算していないため、 $[A]$ 行列のスケーリングを行うことで間接的に $[\tilde{A}]$ の行列要素の絶対値をそろえる努力をした。スケーリングを組み込むとスケーリングなしの場合に比較して 1/2 から 1/3 程度の反復回数で収束する。

4. 解析事例

解析事例としては単純な箱状のテストモデルと実際の成形品であるシリンダブロックのヘッドカバーを取り上げる。

4.1 三点ゲートで形成された箱

本モデルは図6で示すような長方形の開口部に縁を持った箱である。金型は $400.0\text{ mm} \times 400.0\text{ mm} \times 400.0\text{ mm}$ の大きさを持つ立方体で原点は中心にあるものとする。樹脂を注入するゲートは開口部の縁の三箇所を設定されている。冷却管による箱の底面の冷却効果を見るために、箱の下に冷却管を一本と二本配管した例を解析する。図7に冷却管が一本の場合の冷却管とキャビティの位置関係図を、図8に冷却管が二本の場合の冷却管とキャビティの位置関係図を示す。

この例では、キャビティ側境界の熱流束は樹脂流動解析で計算した結果を使用している。流動解析時の条件を表1に示す。また、金型冷却解析での条件を表2に示す。

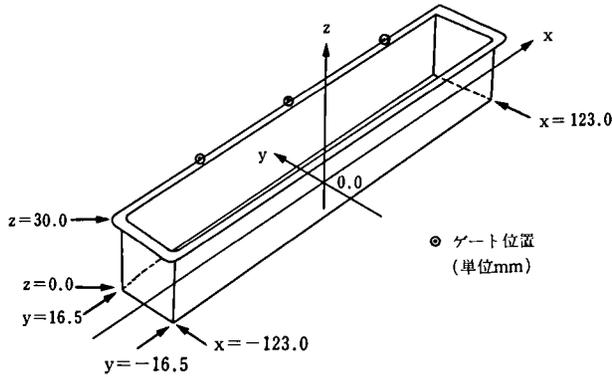


図 6 三点ゲートで形成された箱
Fig. 6 Simple box model with 3 gates

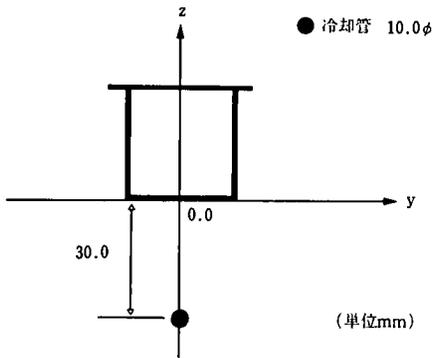


図 7 冷却管とキャビティの位置関係 (冷却管 1 本)
Fig. 7 Geometrical relation between cavity and cooling pipe (one cooling pipe)

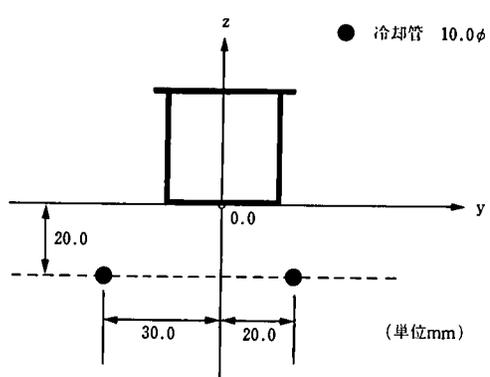


図 8 冷却管とキャビティの位置関係 (冷却管 2 本)
Fig. 8 Geometrical relation between cavity and cooling pipes (two cooling pipes)

表 1 流動解析条件 (三点ゲートで形成された箱)

Table 1 Filling analysis condition of simple box with 3 gates

樹脂材質	: ナイロン 6
樹脂温度	: 250.0°C
金型温度	: 80.0°C
樹脂熱伝導率	: 0.25 W/m°C

表 2 冷却解析条件 (三点ゲートで形成された箱)

Table 2 Cooling analysis condition of simple box with 3 gates

冷却管温度	: 40.0°C
金型雰囲気温度	: 30.0°C
金型熱伝導率	: 46.0 W/m°C
冷却管直径	: 10.0 mm
製品側温度勾配	: 流動解析の結果を使用

図 9, 図 10 および図 11 は, 冷却管が一本の場合の温度分布を示した例である。全体的な傾向として三点あるゲート側の面が温度が高い傾向にあり, 底面の温度分布も箱の内側, 外側ともゲート側の温度が高いような分布をしている。

図 12, 図 13 および図 14 は冷却管を二本に増やし, さらに距離を近づけた解析例である。冷却管が一本の場合と比較すると, 箱の内側外側とも底面の温度は下がり, 分布は均一化されている。



図 9 冷却解析温度分布 上面図 (冷却管 1 本)
Fig. 9 Temperature distribution of inner bottom part (one cooling pipe)



図 10 冷却解析温度分布 上面鳥瞰図 (冷却管 1 本)
Fig. 10 Temperature distribution of side part (one cooling pipe)



図 11 冷却解析温度分布 下面図 (冷却管 1 本)
Fig. 11 Temperature distribution of outer bottom part (one cooling pipe)



図 12 冷却解析温度分布 上面図 (冷却管 2 本)
Fig. 12 Temperature distribution of inner bottom part (two cooling pipes)



図 13 冷却解析温度分布 上面鳥瞰図 (冷却管 2 本)
Fig. 13 Temperature distribution of side part (two cooling pipes)



図 14 冷却解析温度分布 下面図 (冷却管 2 本)
Fig. 14 Temperature distribution of outer bottom part (two cooling pipes)

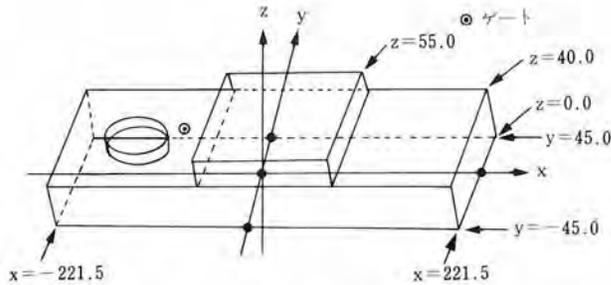


図 15 シリンダブロックのヘッドカバー
Fig. 15 Head cover of cylinder block

表 3 流動解析条件(シリンダブロックのヘッドカバー)

Table 3 Filling analysis condition of head cover of cylinder block

樹脂材質	: ナイロン 6
樹脂温度	: 301.0°C
金型温度	: 105.0°C
樹脂熱伝導率	: 0.35 W/m°C

表 4 冷却解析条件(シリンダブロックのヘッドカバー)

Table 4 Cooling analysis condition of head cover of cylinder block

冷却管温度	: 40.0°C
金型雰囲気温度	: 30.0°C
金型熱伝導率	: 46.0 W/m°C
冷却管直径	: 10.0 mm
製品側温度勾配: 流動解析の結果を使用	

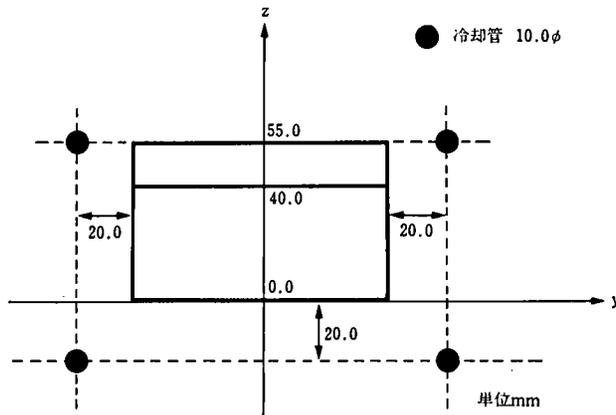


図 16 冷却管とキャビティの位置関係 (冷却管 4 本)

Fig. 16 Geometrical relation between cavity and cooling pipes (four cooling pipes)

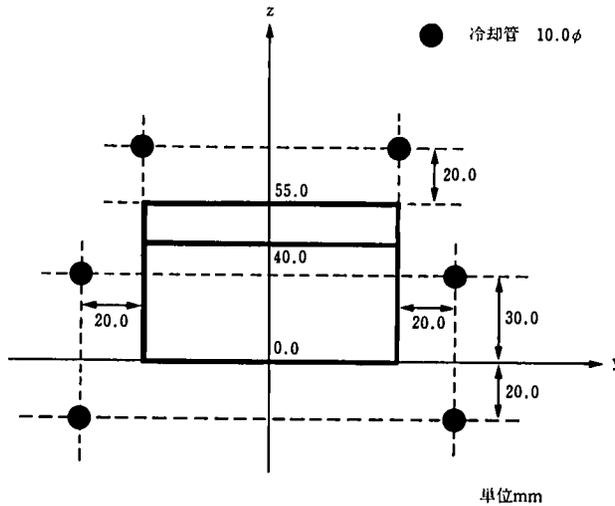


図 17 冷却管とキャビティの位置関係 (冷却管 6 本)

Fig. 17 Geometrical relation between cavity and cooling pipes (six cooling pipes)

4.2 シリンダブロックのヘッドカバー

図 15 に解析対象全体の概略とその大きさを示す。また、図 16 に 4 本の冷却管を使用した際の冷却管とキャビティの位置関係図を、図 17 に 6 本の冷却管を使用した場合の冷却管とキャビティの位置関係図を示す。金型は 1000.0 mm×1000.0 mm×1000.0 mm の立方体で中心に原点を持つ。

この解析例でも、金型のキャビティ側の境界での熱流束は樹脂流動解析で得られた結果を使用する。流動解析の条件を表 3 に示し、金型冷却解析の条件を表 4 に示す。

流動解析を行うことによって、充填直後の製品温度分布の情報を得ることができる。充填直後の製品の温度分布を図 18 と口絵 1 に示す。流動解析はシェルモデルを使用し



図 18 流動解析温度分布 上面図
Fig. 18 Result of filling analysis
(temperature distribution)

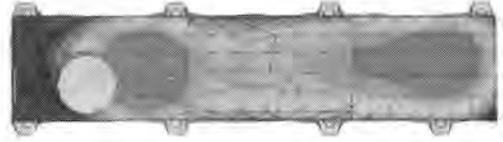


図 19 冷却解析温度分布 上面図 (冷却管 4 本)
Fig. 19 Temperature distribution of outer
part (four cooling pipes)



図 20 冷却解析温度分布 上面鳥瞰図 (冷却管 4 本)
Fig. 20 Temperature distribution of side
part (four cooling pipes)

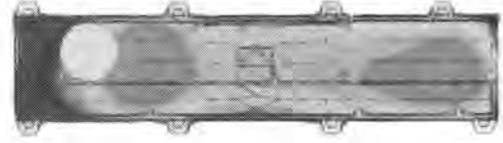


図 21 冷却解析温度分布 下面図 (冷却管 4 本)
Fig. 21 Temperature distribution of inner
part (four cooling pipes)

ているため、製品の内側、外側の温度の違いは考慮されていない。高温部分は製品左側の丸い穴の右側にあるゲートの周辺部分と製品右側の最終充填の場所付近にある。製品左端の部分は比較的初期に充填がされてしまうため温度が下がっている。中央部にも温度の低い部分が見受けられる。この部分は比較的薄肉のため、温度が下がりやすいことに由来する。厚肉の側面部分を伝って樹脂が流れていくため、側面の温度はかなり高いことがわかる。

冷却管 4 本を使って冷却した場合の解析例を図 19 から図 21 に示す。図 19 は製品の上面から見た製品の温度分布である。ゲート付近と最終充填の場所付近では、温度の高い部分が残っているものの、全体的な温度はかなり下がっていると見ることができる。図 20 は製品上面の温度分布を斜め上から見たものである。冷却管の効果によって側面の温度もかなり低下している。図 21 は下から製品内部を見たものである。温度分布は上面から見た場合と似かよっているが、箱の内側の熱は逃げにくいので温度は多少高めに出ていることが高温部分の面積の比較からわかる。

冷却管 6 本を使って冷却した場合の解析例を口絵 2 から口絵 4 に示す。全体の温度は冷却管 4 本の場合と比較するとさらに下がっている。口絵 2 は上面の温度分布であり、新たに製品上側に設けた 2 本の冷却管の効果があることを示している。口絵 3 は上面の温度分布を斜め上から見たものである。側面の温度も冷却管が 4 本の場合と比較して低くなっている。口絵 4 は製品の下面から内部を見た際の温度分布である。上面の温度分布と比較すると、温度が高めの部分が多少多いことを除けば大きな分布の差はない。

5. お わ り に

従来からの金型冷却解析システムの欠点であった冷却管路の再配置による有限要素の再分割作業を必要としない境界要素法に基づく冷却解析システムを試作した。システムによる簡易ソリッドモデルの自動創成により、流動解析用に作成された有限要素モデルを利用して、金型の外形形状と冷却管路の追加だけで冷却解析が可能となった。

今回開発した部分は、成形工程全体における金型の平衡的な温度分布を求めるものであるが、冷却過程における金型の過渡的な温度変化を計算する非定常熱伝導解析の部分を開発して本システムを商品化する予定である。

- 参考文献 [1] 平林繁, “射出成形における溶融樹脂の金型内流動シミュレーション”, ユニシス技報, Vol. 8, No. 3, 1988, pp. 108~121.
- [2] 鬼木和典, “コンピュータによる金型冷却解析”, 型技術, Vol. 7, No. 12, 1992, pp. 54~58.
- [3] T. H. Kwon, “Mold Cooling System Design Using Boundary Element Method”, Trans. of the ASME, Vol. 110, 1988, pp. 384~394.
- [4] 松岡考明, 他, “境界要素法による射出成形金型の非定常熱解析”, 第2回高分子加工技術討論会講演要旨集, 1990, pp. 5~6.
- [5] 神谷紀生・大西和榮, “境界要素法による計算力学”, 森北出版株式会社, 1985.
- [6] A. R. Mitchell, R. Wait, “The Finite Element Method in Partial Differential Equation”, Jhon Wiley & Sons Ltd., 1977. (森正武訳, “偏微分方程式の有限要素法”, 科学技術出版, 1978).
- [7] 菊池文雄・岡部政之, “有限要素システム入門”, 日科技連, 1986.
- [8] 境界要素法研究会編, “境界要素法の理論と応用”, コロナ社, 1986.
- [9] 戸川隼人, “共役勾配法”, 教育出版株式会社, 1977.

執筆者紹介 菅 沼 正 之 (Masayuki Suganuma)

昭和56年東海大学大学院工学研究科応用理学専攻修了。平成元年日本ユニシス(株)入社。CAE関連プログラムの開発, サポート業務を担当。情報処理学会会員。現在CAEシステム部に所属。



平 林 繁 (Shigeru Hirabayashi)

昭和48年東京大学理学部物理学科卒業。同年日本ユニシス(株)入社。CAE関連プログラムの開発, サポート業務を担当。溶融樹脂流動解析(MELT FLOW), 金属溶融流動解析(METAL FILL)を開発。日本鋳物協会, プラスチック成形加工学会会員。現在CAEシステム部に所属。



住宅 CAD 図面の色づけに関する一手法

A Coloring Method for Housing CAD Drawings

前 原 要

要 約 住宅 CAD システム HCAD (Housing CAD system) は、システム内に構築した三次元家モデルをもとに各種の図面を作成するシステムである。HCAD で出力した線図図面は、主に確認申請や施工用に使用されている。近年、これらの図面を顧客に対するプレゼンテーションの場で利用したいというニーズが高まってきた。しかし、プレゼンテーションに不可欠な図面の彩色は、デザイナーによる手描きに頼らざるを得ず、すべての顧客に対応しきれない状況にあった。

今回、線図図面に対して 256 色で彩色を施す、色づけ機能 (彩色機能) を開発した。本機能により、初心者でも十数分の作業時間でプレゼンテーション用の彩色図面を作成することが可能となった。彩色機能は、

- 1) 三次元家モデルの持つ属性情報を継承し、彩色の自動化を実現している、
 - 2) 256 色表示のラスター画像ファイルを使用するという条件のもとで、なめらかな線品質を実現している、
 - 3) 色の微調整に欠かせない画像の色替えと再表示を高速に行っている、
- を特徴としている。

Abstract The HCAD (housing CAD) system helps produce a full variety of drawings by using a 3-D housing model database created in a computer system. The drawings produced by the HCAD system are mainly used for confirmation requests and construction work purposes. And needs have become noticeably stronger these days for the use of these drawings as a tool of presentation to customers. The fact, however, was that the coloring of drawings (because colored drawings are indispensable to any presentations) had to depend on designer's handcraft, making it hard for housing companies to respond to all prospective buyers. Most recently, the author has succeeded in developing a new functionality which allows CAD drawings to be automatically colored in no less than 256 different hues.

The newly developed function now enables even novice designers to spend only ten-odd minutes getting ready colored drawings for presentation use. Its primary features include the following:

- 1) Automatic coloring through the use of attributes stored in the 3-D housing model database
- 2) Displaying of smoother, higher-quality lines through the use of 256-color raster files
- 3) High-speed color change and redisplaying for effective color finetuning.

1. はじめに

住宅 CAD システム HCAD (Housing CAD system) は、システム内に三次元の家モデルを構築し、これをもとに、製図図面の作成、構造、プレカット (施工に先立って、あらかじめ工場内で木材を加工しておくこと)、積算 (建築物の数量拾いに基づいて建築工事費を計算し予測すること) 等を行う一貫システムである。

従来、HCAD の図面は主に確認申請 (建築物の新築等を行う場合に、建築主が建築

主事に対して行う申請) や施工を目的として作成, 利用されてきた。図面はペン・プロッタで描く線図図面である。

近年, これらの図面を顧客に対するプレゼンテーション (折衝) の場で利用したい, という動きが増えてきた。しかしながら, HCAD で作成した図面をプレゼンテーションに使用するためには, デザイナーに依頼し, 手作業で彩色を施す必要があった。この場合, 費用, 作業日数ともに大であり, すべての顧客に適用することができない, という問題があった。このような状況から, ユーザより「受注に結び付くような説得力のある彩色図面を作成する機能」のシステム化が強く望まれていた。

今回, この要望に応え廉価で手早く図面の彩色ができるシステムを開発した。

2. 住宅設計業務におけるプレゼンテーション

本章では, 住宅設計業務の中で行われるプレゼンテーションとは, どのようなものであるか, また HCAD システムとしてどのようなプレゼンテーション支援機能が求められているか, について述べる。

1) プレゼンテーションの目的……新しい住宅を設計する段階で, 住宅メーカーの設計担当者は顧客との間でさまざまな折衝 (以降, プレゼンテーションと呼ぶ) を行う。プレゼンテーションの目的は, 広義にとらえれば顧客に対してその住宅メーカーの信頼性や実績, さらには行き届いたアフタ・サービス等を印象づけることである。また狭義にとらえれば, 施工前のプランを何らかの形で視覚化することにより, 顧客に対して,

- ① 住宅の外観や間取りが居住者の意図に合致しているか,
- ② 室内の広さ感や使い勝手に問題がないか,
- ③ 設計上の不備がないか,

等を検討してもらうことにある。

自社の設計プランを, 他社の商品と差別化し, 顧客の意志を決定づける手段としては, CAD/CG システムを活用した図面や動画表現等が効果的である。

2) プレゼンテーションを支援する CAD/CG システム……プレゼンテーションを支援する CAD/CG システムとして, 現在, 表 1 のようなシステムが実用化されている。しかしながら, 線図表現からフルカラー表現へ, 図面から動画へと, 現実感を追求すればするほど, 処理時間やシステム価格は増大していく。

3) HCAD におけるプレゼンテーション支援機能の位置付け……パース図による表現は, すでに HCAD システムで実現済みである。一方で, アニメーションや Virtual Reality は高度な現実感を表現できるが, 反面, コストの高いプレゼンテーションとなり, 個人住宅を建築しようとしている顧客向きではない。

このような観点から, HCAD におけるプレゼンテーション支援機能の位置付けを, 「二次元の線図図面に対する彩色」と設定し, その上で次の 2 項目を課題とした。

- ① 従来のペイント・システムの弱点, すなわち,
 - ・モデラとのデータ連動が弱い点
 - ・手作業主体のため作業時間が増大してしまう点
 - ・設計変更に従従しにくい点

を克服すること

- ② システムの出力は、デザイナーによる手書きの彩色図面に迫ること

表1 プレゼンテーションを支援する CAD/CG システム

Table 1 CAD/CG systems for presentation

分類	機能と特徴
建築向け 製図CADシステム	パース図(線図)により、外観、室内を表現する。 平面図、立面図と比較し、現実感が増す。
ペイント・システム	平面図、パース図等、二次元の線図をスキャナから入力し、フルカラーで彩色する。 モデラとの連動が弱い為、手作業が多い。
レンダリング・システム	三次元モデルを使用し、シェーディング、陰影付け、テクスチャ・マッピングができる。 個々の画像をVTRに収録し、室内を歩きまわる等のアニメーション表現が可能である。
Virtual Reality ^{III)}	手や体の動きに合わせ、頭部に搭載したディスプレイに、映像を表示させる。ユーザは、システムが作り出した仮想空間の中で、臨場感を持って活動することが可能。 将来的なシステムである。

3. 彩色機能の概要

今回開発した住宅CAD図面に対して色づけを行う機能を彩色機能と呼ぶ。本章では、彩色機能のデータの流れ、および設計上のポイントについて説明する。

- 1) 彩色機能のデータの流れ……彩色機能は、設計サブシステム、製図サブシステム、および彩色サブシステムから構成される。

設計サブシステムではラフ・スケッチをもとに三次元家モデルを構築し、製図サブシステムでは、三次元家モデルから二次元の図面を作成する。最後に、彩色サブシステムで図面をラスタ化し色づけを行う(図1)。

- 2) 設計上のポイント……彩色機能の設計上、ポイントとなった点を以下に示す。

- ① きめ細かい色の塗り分けを可能とする：HCADでは、部屋、ドア、家具、等の一つ一つを部材として扱っている。しかし、一つの部材を1色で彩色するわけではない。一例として、部屋の場合、床・壁・天井は別々の色で彩色できる必要がある。

このため、個々の部材を、もう一段階細分化して識別できるようにした。部材各部の識別方法については4章で説明する。

- ② 彩色の過程を自動化する：市販のペイント・システムのように、手作業で彩色を進めるようでは困る。HCADの持っている豊富な属性情報を継承し、彩色の過程を自動化する必要がある。

面の塗りつぶしを行うためには、図面上でその部材を閉領域として認識する必要がある。そこで、三次元家モデルを二次元の図面に加工する際、従来の線分データに加え、彩色のための領域データも一緒に作成するようにした。領域データの作成については5章で説明する。

- ③ 線の品質を保つ：透視図では、図面を構成する線のほとんどが斜線である。

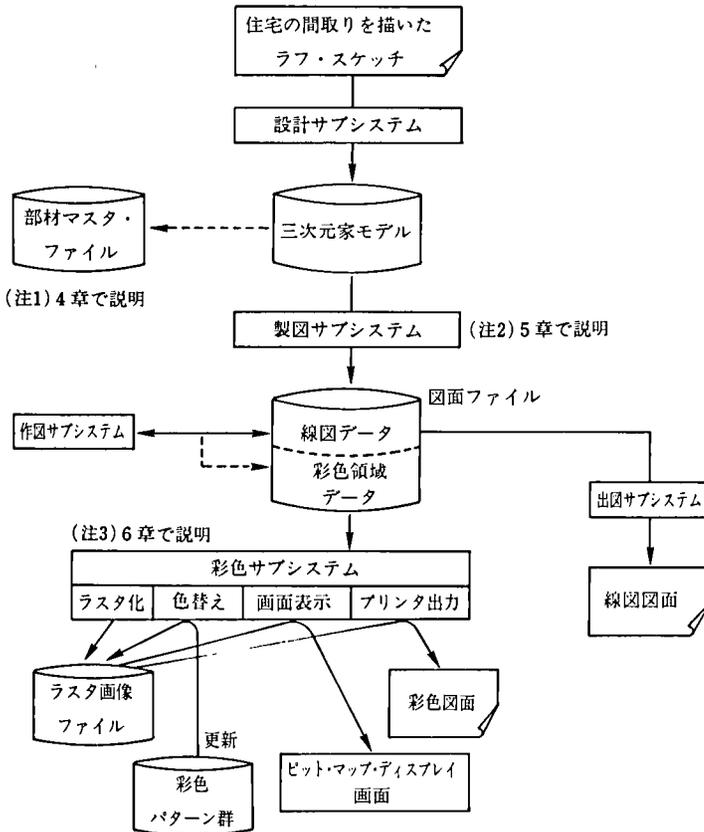


図1 彩色機能のデータの流れ

Fig.1 Data flow of coloring system

水平線・垂直線はもちろんのこと、斜線についても、ビット・マップ・ディスプレイ画面やフルカラー・プリンタに出力した際、ジャギ（斜めの直線を表示したとき、階段状の線となること）のない、なめらかな線品質を実現する必要がある。

そこで、彩色図面はアンチ・エリアシングを施したラスタ画像ファイルとした。

- ④ 画像の色替えと再表示を高速に行う：一度彩色した結果がユーザの意図に合わない場合、ただちに部材指示による色の変更ができる必要がある。

このため、複数の色変更手段を提供すると共に、ラスタ画像ファイルの更新方法を工夫した。色替えと再表示の方法については6章で説明する。

- ⑤ ハードウェアは、HCAD システムの標準構成 (US ファミリの GX モデル*) のままで稼働するようにした。

4. 色の塗り分けのための部材各部の識別方法

個々の部材を彩色する際、一つの部材の構成要素をどこまで細分化して認識できる

* GX モデル：1677 万色の中から選んだ 256 色を同時表示することが可能なビット・マップ・ディスプレイ

かにより、彩色図面のきめ細かさが決まる。本章ではこれらの部材の識別方法について説明する。

4.1 部位物データの分類

HCAD では、部材を部位物と配置物に分け、それぞれ異なった取り扱いをしている。部屋（洋室、主和室、寝室、……）・屋根・屋外造作（ポーチ、テラス、……）等、家ごとに異なった形をとる部材を部位物と呼ぶ。部位物の図形データは、家ごとに三次元家モデル内に持っている。

一つの部位物データ、たとえば洋室 A は床・壁・天井といった細分類の下に床面・壁面等の平面データを持っている。このため、洋室 A を構成する個々の面に対して、「大分類：部屋の中分類：洋室 A の小分類：床面」といった名札を付け、面データと名札を後工程まで伝達させるようにした。後工程では、小分類を単位とした色の塗り分けが可能となる（図 2）。

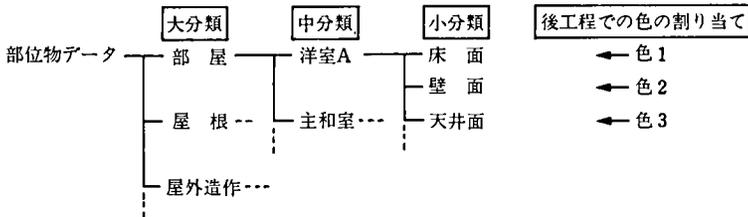


図 2 部位物データの分類

Fig. 2 Classify portion data into detail categories

4.2 配置物データの分類

ドア・窓・家具等、あらかじめ形状が決まっている部材を配置物と呼ぶ。配置物の図形データは部材マスタ・ファイル内に持ち、三次元家モデル側にはその配置情報を持っている。

一つの配置物データ、たとえば、テーブル A は、スチール製の脚と枠、ガラス製の天板から構成されている。脚・枠・天板はいくつかの平面の集まりである。従来、これらの平面群には識別情報がなかったため、個々の面に対して、小分類となる「材質コード」を付与するようにした。そして、個々の面に対して「大分類：家具の中分類：テーブル A の小分類：ガラス面」といった名札を付け、面データと名札を後工程まで伝達させるようにした。後工程では、面データと名札により、部位物・配置物の区別なく、面の識別を行うことが可能となる（図 3）。

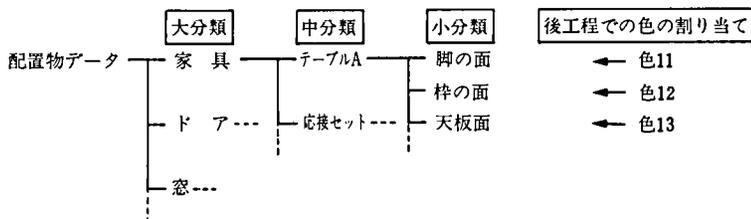


図 3 配置物データの分類

Fig. 3 Classify placement data into detail categories

5. 製図サブシステムにおける彩色領域データの作成

製図サブシステムでは、三次元家モデルを二次元化して図面とする際、従来の線分データに加え、彩色のための領域データも一緒に作成し、出力するようにしている。

1枚の図面は一つの図面ファイルに格納する。このとき、線分データと領域データはレイヤ（層・図面内のデータを階層状の構造に分類する際の単位）を分けて格納している。

5.1 平面図系の彩色領域データ作成

平面図は、建物を標準の高さの位置で水平に切断し、間取りを上方向から水平投影したものである。平面図を彩色する場合、各部材の彩色の順序はあらかじめ規定できる。一例として、LDK内にテーブルが置いてある場合、テーブルの領域は、LDKの床の領域に対して常に優先して彩色する。言い換えれば、テーブルがLDKの床の領域によって隠されることはない。

そこで、領域データを格納するために10層のレイヤを用意し、色の塗り重ねの順に、最初に塗りつぶす領域は第1層に、最後に塗りつぶす領域は第10層に格納するようにした(図4)。後工程では、第1層に所属する領域から、順次、領域の塗りつぶしを行っていく。こうすることにより、領域を作成する際に、煩雑な領域演算(二つの領域の一部が重なるとき、隠される領域の形状を隠す領域の形状で切り取っておく)を避けることが可能となった。

5.2 外観透視図・室内透視図の彩色領域データ作成

外観透視図・室内透視図は、建物の外観や室内を任意の視点から透視投影した図である。透視図では、各部材の彩色の順序をあらかじめ決定しておくことはできない。このため、領域データを格納する層は1層のみとし、すべての面対面の隠面消去を行い、互いに重なりのない彩色領域データを作成している。

製図サブシステムで作成した外観透視図の線分データを図5に、同じ図の彩色領域データを図6に示す。

6. 彩色サブシステムにおけるラスタ画像ファイルの作成と色替え

製図サブシステムは図面を彩色するために必要な材料を提供した。彩色サブシステムでは、図面をラスタ化して彩色し、ラスタ画像ファイルを画面表示し、色合いを微調整し、フルカラー・プリンタへ出力する。1枚の図面を彩色するための操作手順は図7のとおりである。

本章では、ラスタ画像ファイルの形式(6.1節)、ラスタ化の際のアンチ・エイリアシングの手法(6.2節～6.3節)、および色の微調整の方法(6.4節)について説明する。

6.1 ラスタ画像ファイルの形式

ラスタ画像ファイルは、USファミリのモデルGXの標準ラスタ・ファイル形式を採用している。この形式では、1677万色の中から選んだ256色を使用することができる。標準ラスタ・ファイル形式の構造を図8に示す。

6.2 アンチ・エイリアシングの実現方法

ラスタ・データの上で線をなめらかに表現する手法として、アンチ・エイリアシングという技術が広く使われている。彩色サブシステムは、面積法によるアンチ・エリア

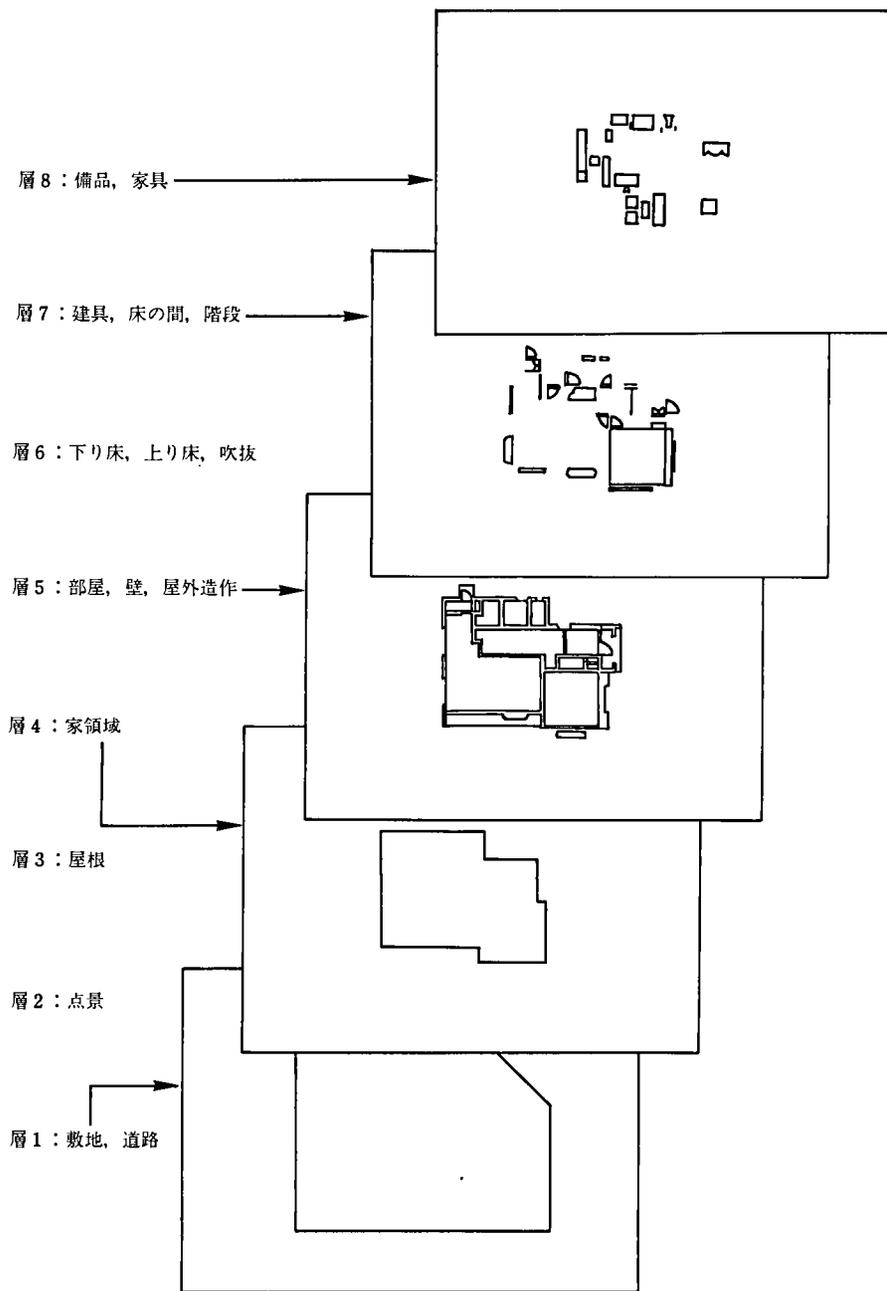


図 4 平面図における彩色領域データの階層構造

Fig. 4 Layer structure of surface data



図5 外観透視図の線分データ

Fig. 5 The exterior drawing of a model (vector data)



図6 外観透視図の彩色領域データ

Fig. 6 The exterior drawing of a model (surface data)

シング[2]を採用した。図9に面積法によるアンチ・エアシングの考え方を示す。

すなわち、ラスタ・データ上に表現する線分は、ある一定の幅を持った長方形であると考え、その長方形が通過する画素について、画素と長方形が重なった面積（太線の部分）に応じて、各画素の輝度（つまり、その画素の黒さ）を決めている。

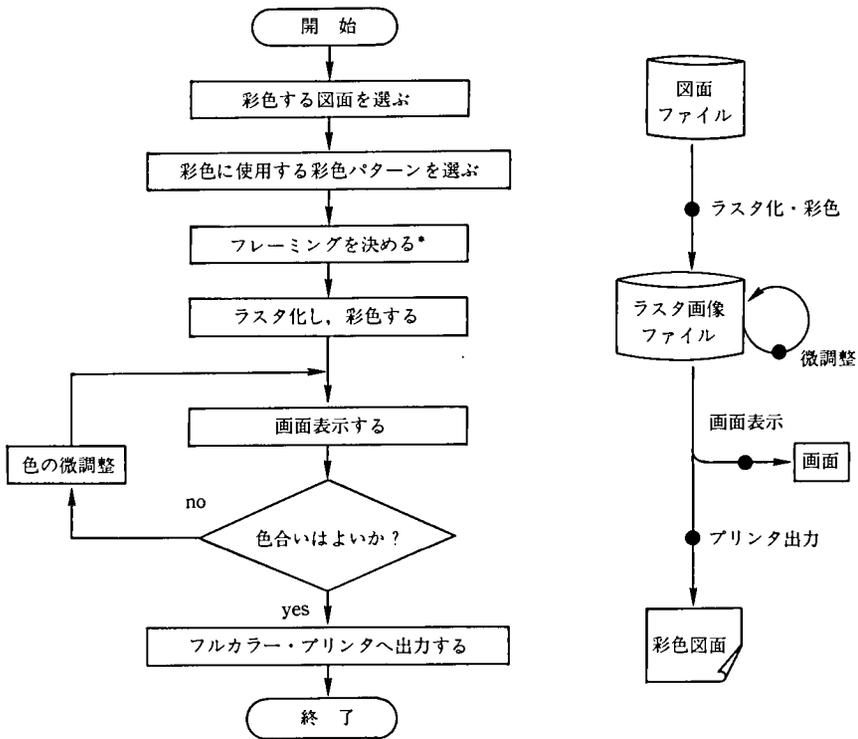
画素の輝度は、256階調としている。また、面積の算出はその都度計算を行わず、

- ① 画素の中心と線分の中心線との距離 (dy),
- ② 線幅 (W),
- ③ 線分の傾き (S)

をパラメータとしたテーブル参照方式とし、処理速度を上げている。

6.3 色と明暗を256色に丸める

彩色図面の上では、部材を構成する一つ一つの領域の境界に黒色の輪郭線を描く。



* 1枚の図面に、1階平面図と2階平面図が配置してある場合等に、実際に1枚の彩色図面にする範囲を決定する。

図7 1枚の図面を彩色する操作

Fig.7 Process flow to make a colored drawing

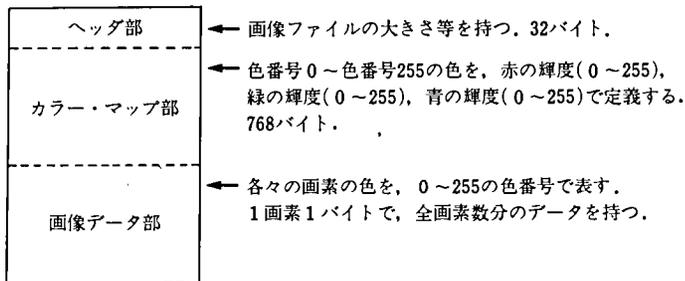


図8 標準ラスタ・ファイル形式

Fig.8 Standard raster file format

また、床模様のように領域の内部に黒色の模様線を描く場合もある。領域の色はさまざまであり、領域の色が線の色(すなわち黒色)へと変わっていく過程は、アンチ・エイリアシングにより、なめらかに変化させなければならない。これを256色の範囲内で表現する必要がある。

このため、ユーザに開放する色数を限定し、残りの色数で領域の色が線の色(黒色)に移行する過程の中間の色を定義するようにしている(表2, 図10)。

表3 ラスタ画像ファイルの更新方法の比較

Table 3 Modification method of raster file

ラスタ画像ファイルの更新方法	特 徴
【方法1】 その領域を彩色している色番号を変更する。	①その領域の色のみの変更が可能である。領域Aと領域Bを同一の色番号で彩色している場合、Bを変更せずにAのみを変更することができる。 ②画像ファイル内の画像データ部(1152×900画素の場合、約1メガ・バイト)の書き換えが必要である。 ③画素と領域との関係を保持しておく必要がある。
【方法2】 色番号に対応する色の定義(赤・緑・青の輝度)を変更する。	①同一の色番号を使用している領域は、すべて新しい色に変わる。 ②画像ファイル内のカラー・マップ部(768バイト)の書き換えのみで実現できる。 ③画素と領域との関係を保持しておく必要がない。

すような2通りの方法が考えられる。

彩色サブシステムでは、「色の変更を波及させたくない部材どうしには、別々の色番号を付与してもらおう」という条件のもとで方法2を採用した。この結果、一度ラスタ画像ファイルを作成しておけば、彩色パターンの変更に合わせて高速にラスタ画像ファイルを更新させることが可能となった。

ラスタ画像ファイルの画面表示は、USファミリの標準機能である screenload コマンドを使用している。画面表示の所要時間は、1152×900画素の場合で約1秒である。

彩色機能を使用して作成した彩色図面を口絵1～2に示す。

口絵1は、外観透視図を自動彩色したもので、64色を定義した彩色パターンを使用したためアンチ・エリアシングの階調は、4階調である。口絵2は、室内透視図を自動彩色したもので、64色を定義した彩色パターンを使用したためアンチ・エリアシングの階調は、4階調である。

7. おわりに

彩色機能は、デザイナーが線図図面を彩色する過程を自動化し、システム化したものである。したがって、いわゆるペイント・システムの一つとみなすこともできる。ここで、彩色機能と市販のペイント・システムとの比較をまとめてみる。

ペイント・システムとの機能比較表を、表4に示す。彩色機能は、テキスト・マッピング、画像合成、マスク操作といった、画像データを汎用的に加工する機能を持っていない。反面、前工程との密結合により、属性情報を活かした彩色の自動化を実現している。彩色機能の特徴をまとめると、次のようになる。

- 1) モデル上の属性データの継承……ペイント・システムが扱うラスタ画像データは、個々の画素の明暗を持っているに過ぎない。これに対し、彩色機能では図面ファイル上に、線分データと共に領域情報、生い立ち情報を保持することにより、部材指定による色の変更要求に柔軟に対応できる。
- 2) 彩色の自動化……図面上のある部分を彩色しようとしたとき、ペイント・システムでは、①領域の定義、②色の決定、③領域の塗りつぶし、という手順を踏んで彩色を行わなければならない。これに対し、彩色機能では彩色パターンの選択

表 4 ペイント・システムとの機能比較表
 Table 4 Compare coloring system with paint system

比較項目	ペイント・システム	彩色機能
使用できる色数	1677 万色	1677 万色中の 256 色
データの形態	ラスタ画像	三次元モデル→二次元線図→ラスタ画像
彩色の対象	任意データ	HCAD の製図図面に限定
任意の色作り	○	○
テクスチャ・マッピング	○	×
画像の合成	○	×
マスク操作	○	×
モデル上の属性データの継承	×	○
彩色の自動化	×	○
モデルの設計変更への対応	×	○
1 図面の彩色に要する時間	2~3 時間	図面作成：5 分、彩色：5 分
専用ハードウェア	フレーム・バッファが必要	不 要

のみで彩色を行うことができる。

- 3) モデルの設計変更への対応……いったん彩色図面を作成した後、モデルを設計変更する場合を考える。ペイント・システムでは、①新たな図面を用いて最初から彩色作業をやり直す、あるいは②モデルとの同期を断念し彩色図面上で形状を手直しする、という対応が必要であった。反面、彩色機能では再創成機能を利用し、ただちに図面を作成し直すことが可能である。

最後に、彩色機能の導入により、プレゼンテーション用の彩色図面を初心者でも十数分の作業時間で作成することが可能となり、2 章の 3) HCAD におけるプレゼンテーション支援機能の位置付け、で述べた課題①を達成することができた。しかしながら、課題②については、彩色機能で作成した画像と、デザイナーが彩色した図面を比較すると、いくつかの点でまだ機械的な硬さが感じられる。今後の課題として、

- 1) 太陽の位置を設定し、外観透視図に対する陰影付けを行う、
- 2) 水彩画風のぼかした彩色を可能にする、
- 3) 背景画像との合成を行う、

のような機能追加を行い、彩色図面の完成度を高めていく必要がある。

- 参考文献 [1] 野村淳二, “仮想現実感の住環境設計への応用”, PIXEL, No. 113, 1992, pp. 122~131.
 [2] 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, Vol. 9, No. 4, 1990, pp. 18~30.

執筆者紹介 前 原 要 (You Maehara)

昭和 34 年生, 56 年京都コンピュータ学院情報工学科卒業, 同年日本ユニシス(株)入社, 以来, CAD/CAM システムの開発に従事, 現在, 製造工業システム本部システム四部に所属.



プレス金型の形状設計 CAD

Stamping Die CAD System

沢田晃二, 井上博順, 前田幸久

要 約 自動車の車体部品用の鋼板プレス金型の設計・製作に要する費用の削減, 期間短縮のためには, プレス成形性を保証した良質の設計作業が不可欠である。

本稿は, 日産自動車(株)にて開発・業務適用されている「プレス金型形状設計用 CAD システム」を事例として, いわゆるダイフェース CAD システムとして満たすべき機能要件と, それにそっての該システムの特徴を解説する。

前半部で, 「プレス金型形状設計用 CAD システム」は形状定義と検討・評価機能が統合化された三次元 CAD システムであるべきだという観点から, その具体的な機能要件を, DB/操作等のフレームワーク面, モデリング/成形性評価等のコマンド機能面の各面から考察する。

後半部では, 日産自動車(株)におけるプレス金型形状設計用 CAD 「KISS-PUNCH」のシステム概要, 特徴的機能について報告する。

Abstract The design process of stamping die for parts of automobiles that assures good sheet metal forming is indispensable to reduce production cost and to shorten lead time.

In this paper we assert that, in the CAD to design shape of stamping die, modeling functions and evaluation functions should be combined into design functions and inevitably the CAD must be integrated with CAE and CAM systems.

We describe requisite functions to the CAD and present a CAD system 「KISS-PUNCH」 which was developed and is applied now in NISSAN MOTOR CO., LTD. Machine & Tool Plant.

The first half of this paper is devoted to functional requirements for stamping die CAD systems from aspects of

- 1) database and human-machine interface and,
- 2) user commands for modeling and sheet metal forming simulations.

The second half covers the outline of 「KISS-PUNCH」 system and explains characteristic functions from above-mentioned aspects.

1. はじめに

プレス加工は自動車をはじめとして製造業界で広く活用されているが, 成形形状が複雑な薄鋼板のプレス成形の場合, 成形後の材料の弾塑性変化による形状の戻りやシワ・ワレ等の発生により意図通りの最終形状を得ることがむずかしい。このために, 自動車の車体部品用プレス金型の設計・製作に要する費用・工数の多くは, 仕上げ・トライアル補正段階で費やされている。

この費用の削減, 期間短縮のためには, 形状不具合の少ないプレス成形性の保証された良質の設計作業が不可欠である(図 1)。

日産自動車(株)においては, 旧来, プレス金型の成形形状の設計・検討は主として熟練設計者が図面やモデル上でを行い, NC 加工用のデータ化を専任オペレータが行っ

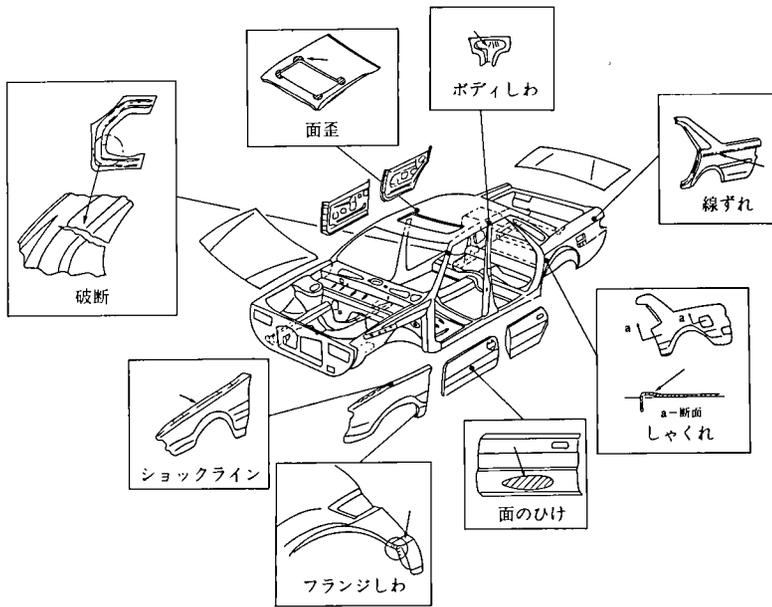
図1 成形不具合の例¹⁾

Fig.1 Failures in sheet metal forming

ていた。しかし、この方式は、品質保証という点で設計者個人の経験・勘に依存する側面が強く、設計段階で成形性に関する定量的・客観的な評価・検討がしにくい、また工数的に見ても重複作業が多い、という問題点を持っていた。これらを解決するために、金型設計・製作の基盤技術としてのCAD/CAE分野において「作業性の良いデータ形式と形状モデラ」および「成形性評価を容易にする解析技術」ならびに「これらを有機的に結合する技術」を新たに確立し、マンマシンインタフェースに優れた生産性の高いCADシステム「KISS-PUNCH」を開発した。本システムにより金型設計者が容易に使いこなせるCADシステムが初めて実用に供され、現在、プレス金型設計・製作費の削減、金型の高品質化等に大きな成果を上げている。

以降、筆者らの上記システム開発の経験をふまえて、自動車車体部品用プレス金型形状部の設計用CADシステムとして満たすべき機能要件と、それにそつての上記システムの特徴を述べる。なお本稿において、CADはとくに断りのない限り、成形性検討・解析機能という意味でのCAEを包含した概念で使用する。

2. プレス金型形状設計業務とCAD化における機能要件

本章では、自動車車体部品を主対象としたプレス金型形状部の設計を中心とする工程設計業務の概要と、これをCAD化する際のシステム要件について一般的な観点から論ずる。

2.1 プレス金型形状設計・製作業務のシステム化

プレス金型形状設計・製作業務をシステム化の観点から見ると、典型的には以下の方式があり、歴史的にもほぼこのSTEP1 → STEP2 → STEP3の順で変遷しているとみられる(図2)。

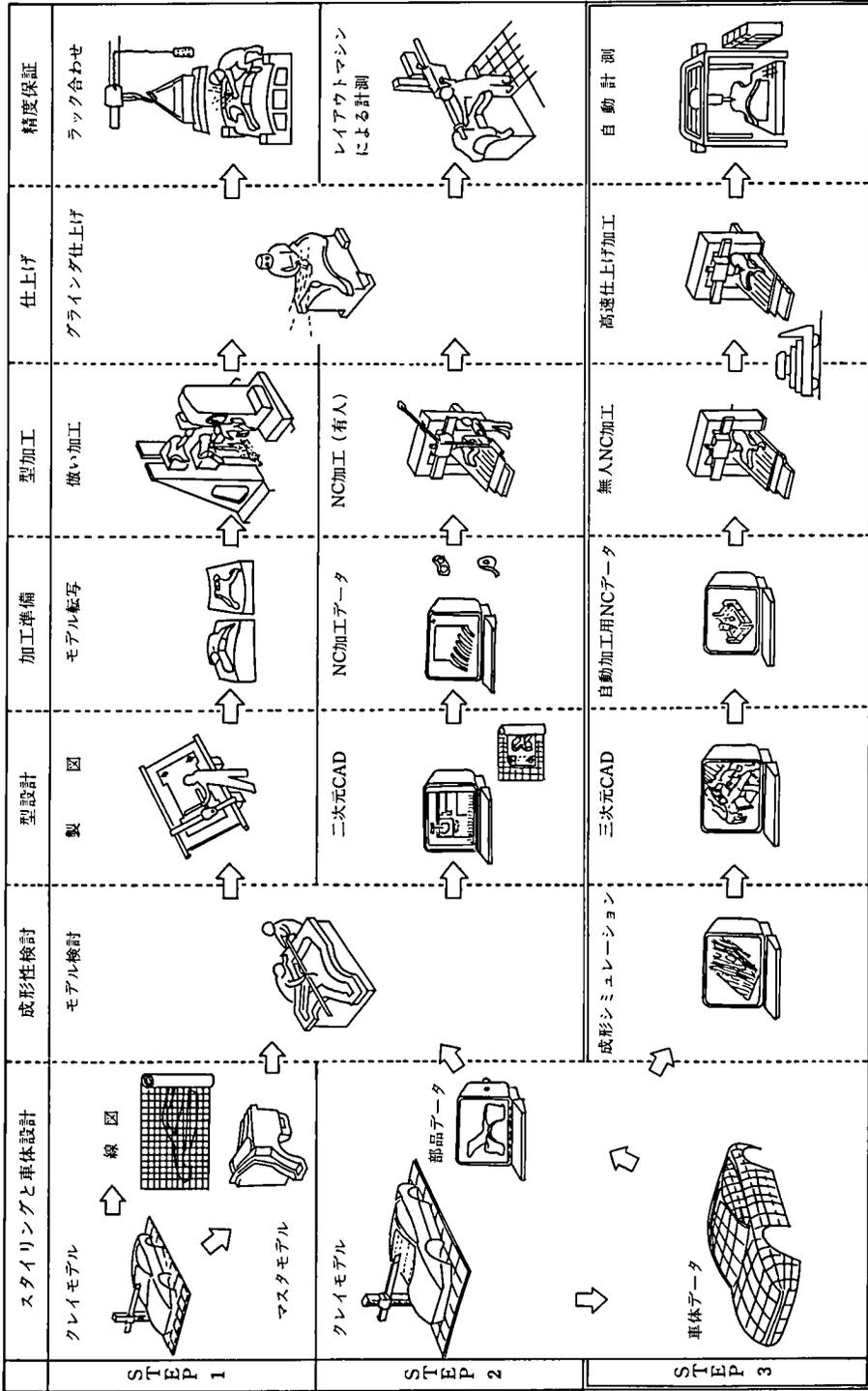


図2 プレス金型設計・製作のプロセス

Fig. 2 Design and production process of stamping die

- STEP1 (非 CAD, 非 CAM 方式)
 - 設計：製図 (手書き), 物理モデル検討
 - 製作：倣い加工
 - 特徴：石膏や樹脂等の物理モデル基準による製作
- STEP2 (CAM 方式)
 - 設計：製図 (二次元 CAD または手書き), 物理モデル検討
 - 製作：NC 加工
 - 特徴：物理モデルと数値モデルの併用による製作
NC 加工用の 3D データ作成 (数値モデリング作業) が必要
- STEP3 (CAD/CAM 方式)
 - 設計：三次元 CAD (含：二次元 CAD)
 - 製作：NC 加工
 - 特徴：数値モデル基準による製作
設計時点で以下の作業を同時並行的にすすめる。
 - ① 二次元データの定義・作成
 - ② 二次元での検討・評価
 - ③ 三次元データの定義・作成
 - ④ 三次元での検討・評価・解析
 - ⑤ 図面作成

STEP2 の特徴は製作業務の NC 化であるが、これは現在の大手自動車関連会社ではほぼ 100% 実現化されている。

STEP3 の本質は設計業務 (単なる作図行為だけでなく、試行錯誤しながら形状を決めていくプロセス) の CAD 化にあるが、この部分は、プレス金型形状部が複雑な三次元曲面のため、従来技術では完全なシステム化が困難な領域であり、本稿の中心テーマである。

プレス加工における工程は製品部品の種類や形状により多岐にわたるが、多くは

- 1) 材料取り (ブランク：blank)
 - 2) 絞り (ドロー：draw)
 - 3) 縁切り (トリム：trim), 縁まげ (フランジ：frange), 穴あけ (ピアス：pierce)
- 等の二次成形

からなり、この中でも絞り工程が成形性の面から見て最も難易度が高くかつ成形品質に影響を及ぼす。このため、プレス金型工程設計において、絞り工程における形状部の定義が重要な設計ポイントとなっている。

図 3, 図 4 に絞り形状の概略図を示す。

製品部 ：ドア、フェンダ等、最終的に車体の一部として使用される部分。
車体設計部門にて設計される。

ダイフェース：プレス金型のブランク・ホルダー部分の形状 (しわ押さえ面*)

ビード ：ブランク材に張力を与えるためのダイフェース上の溝形状

余肉 ：プレス成形上必要であるが、最終的に製品部とならないダイフェ

* 「ダイフェース」に余肉を含める場合もあるが、本稿では余肉以外のしわ押さえ面という意味で使用する。

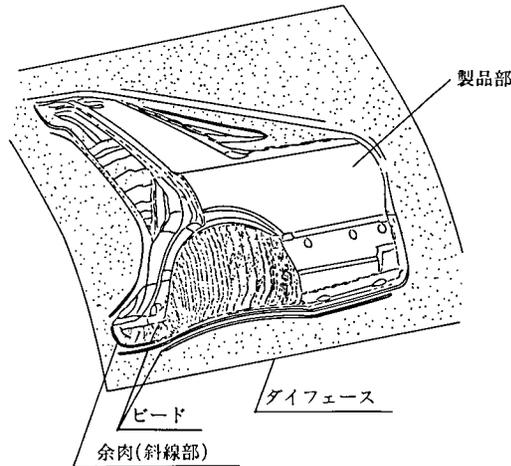


図3 プレス金型の絞り形状

Fig. 3 Shape of draw stamping die

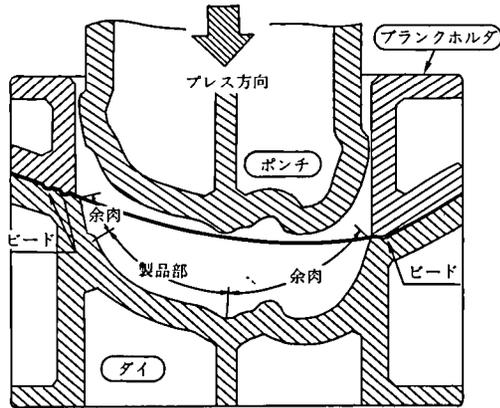


図4 プレス金型の絞り形状(断面図)

Fig. 4 Shape of draw stamping die (cross section)

ース以外の形状部分

プレス金型形状の設計作業の主は、製品部を元にしてダイフェース、余肉、ビードを定義することであるが、これはほぼ図5の手順により行われる。

また、以降において(金型の)形状とは、とくに断りのない限り、ダイフェース、ビード、余肉からなる絞り金型の製品部を除いた部分を示すこととする。

STEP2における形状の設計は二次元CAD(または手書き)により行ってきたが、検討・評価作業が十分システム化されていなかった。あるいは、システム化されていても形状定義と独立したシステムであった。

これを解決するためには、形状設計時に成形性に関する品質保証を行う必要があり、形状定義と検討・評価機能が一体化された三次元CADシステムが不可欠である。また、このことにより初めてCADシステムとCAMシステムの統合化が実現される。

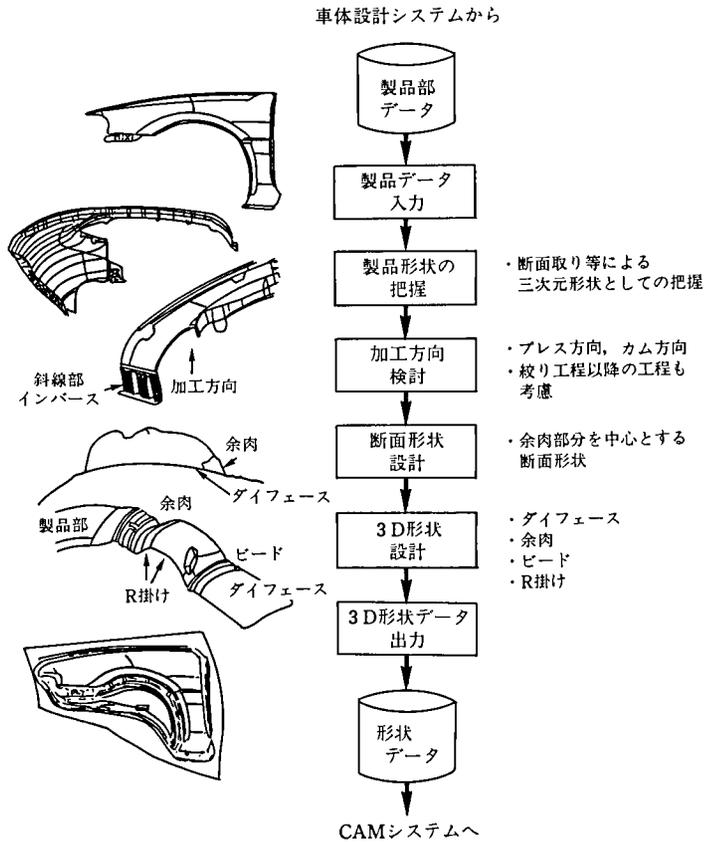


図5 絞り工程における金型形状設計手順
Fig. 5 Design process of draw stamping die

2.2 CAD化における機能要件

プレス金型形状設計業務のCAD化の目的は、根源的に

- 1) 型設計品質の向上
- 2) 型製作コストの低減
- 3) 開発リードタイムの短縮

のQCDにつぎすが、本節ではこれを実現するためのシステム機能要件をいくつかの観点から挙げる。

2.2.1 コマンド体系と操作

コマンド体系は、汎用CADシステムのような出力図形タイプごとのコマンドではなく、プレス金型の形状設計業務専用システムとして、その作業手順単位に設定すべきである。

- ・製品データ入力
- ・製品形状の認識・検討
- ・プレス方向検討
- ・余肉設計
- ・ダイフェース設計

- ・ビード設計
- ・成形性評価・シミュレーション
- ・余肉・ダイフェース形状等出力

また、設計作業の本質が、試行錯誤、すなわち

- 1) 形状の仮り決め
- 2) 検討・評価
- 3) 1) 2)の繰り返しの後に、形状の本決め

にあるので、形状定義機能の他に

- ・検討・評価機能の充実（含：コマンド割り込み機能）
- ・UNDO 機能（入力 UNDO, 実行 UNDO, 形状復元機能）

が必要不可欠な機能要件となる。

2.2.2 設 計 DB

曲面・曲線等の三次元形状データ、断面線等の二次元データ、寸法等の図面データのみならず、以下のような設計・製作作業に関わる諸々の情報を保持・管理する必要がある。その意味では近年よく話題にのぼる製品モデル*を指向すべきであろう。

- ・加工方向（プレス方向、カム方向）
- ・鋼板材質、板厚情報、板内/板外区分
- ・形状データの部位別管理
 - …………… 製品部、ダイフェース、余肉、ビード等
 - （定義時のパラメータも保持）
- ・設計変更情報
- ・プレス成形性に関する
 - 評価用基準データ
 - トライアル結果のデータ

これらの中でもとくにプレス成形性に関する評価用基準情報、検討/測定結果のデータは「成形性に関するノウハウ」そのものであり、

- データ収集/蓄積方法
- 利用/システムへの組み込み
- 蓄積データの管理

等に関して十分な配慮・工夫がなければならない。

2.2.3 二次元設計と三次元モデリングの同期

前述 STEP2 におけるベテラン設計者達は、その設計ノウハウを長年の手書き図面作成・モデル検討という作業を通して培ってきており、それが企業としての技術レベルをささえてきた。STEP3 への移行に際しても従来のノウハウの継承は必須であり、システムとしても二次元的な設計機能と三次元形状定義・検討が設計者から見て自然な形で統合化されている必要がある。

具体例として、寸法に基づく主要な設計は断面等での二次元作業となり、断面間をつなぐ形状モデリングは三次元処理となるが、こうした一連の作業は交互に行う必要

* 製品モデル：設計生産の対象である製品の形状だけでなく、製品の持つ各種技術情報を体系的にシステム内に表現したデータモデル

がある。このためには、二次元データと三次元データ間の相互変換・参照、二次元設計と三次元モデリングの並行作業、が自由に行えることが重要である。

2.2.4 設計者とモデリング作業

STEP2での設計者は図面作成が主作業であり、確認用モデルは図面作成後に製作し設計形状を事後的に確認・検討する方法をとっている。

STEP3においては、形状定義時に数値モデルによる評価・検討を並行的に行う。このため、余肉・ダイフェース・ビード等、プレス金型固有の断面形状をパラメータ設計的に容易に定義するコマンドの他に、断面形状を「適当につなぐ・ほかす」、あるいは面と面の間を「面なりに埋める」といった、STEP2における物理モデル作成時に造形感覚で行っていた機能が非常に重要になる。

図6でビードを例に、パラメータ設計による断面形状定義、ほかし・縮退のある曲面モデリングを示す。また、図7は曲面の隙間を周辺の曲面なりにうめる図である。

一方、設計者のためのシステムである以上、本質的に設計と無関係なモデリング作業（曲面のトリミング等）は自動化機能等によりその負荷を極力減らすべきである。図8はR掛け（フィレット曲面）による自動トリミングおよびR削除時の自動復帰を示している。

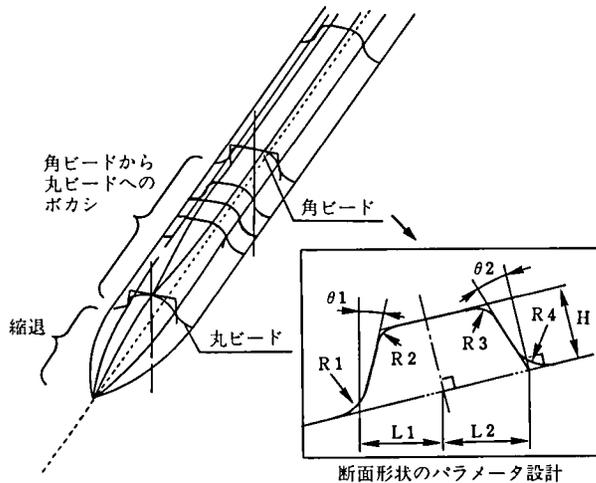


図6 ビード曲面

Fig. 6 Bead surface

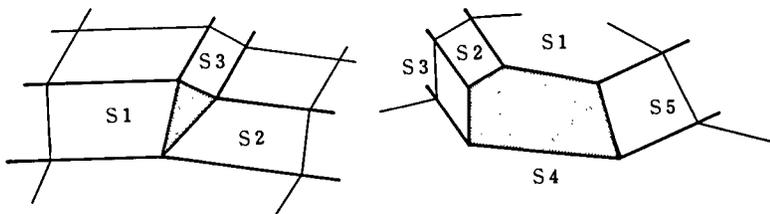


図7 面間のすきま埋め

Fig. 7 To cover gap between surfaces

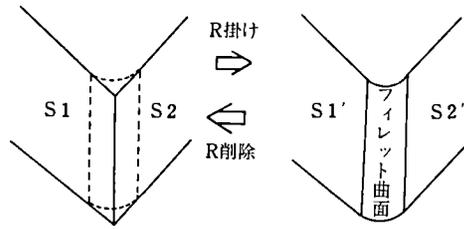


図8 R 掛けによる自動トリミングおよび自動復帰

Fig.8 Automatic trimming by fillet surface and automatic restoring

2.2.5 成形性検討機能

CAD システムにおける検討・解析機能の中心は成形性に関するものである。

成形性に関しては、成形形状が極めて複雑な自由曲面であることから学問的にも未解明な部分が多く、従来から理論よりも実験データや過去の経験をベースに判断・評価を行っていた。しかしながら近年、ITAS-3D 等のような FEM による 3D 板成形シミュレーション技術の研究が、質・量ともに大きな進歩をとげつつあり、EWS (エンジニアリング・ワーク・ステーション) の高性能化と低価格化とあいまって、かなり実用化のレベルに近づいたといえよう。

口絵 1, 口絵 2 はそれぞれ、ITAS-3D によるシミュレーション例と、同条件による実験結果を示している。口絵 3 はドア部品の成形におけるシワ発生シミュレーション例である^[2]。

鋼板の成形シミュレーションでは、小さい R 形状部の摩擦が重要な意味を持つ。このため FEM による解析を考えた場合、解析用要素数を少なくできず、一般に長い計算時間を要するが、実際の設計作業では、種々のパラメータ値に対する短時間の解析結果が必要となる場合が多い。その意味で、実験に裏付けられた簡易シミュレーション機能を工夫・充実させ会話型の成形性検討機能としてこれを多用する一方で、弾塑性力学を考慮した本格的な FEM 解析機能をシステムにリンクさせ、設計作業での節目節目で解析結果を検証することが望ましい。表 1 に CAD/CAE による解析手段と成形性不具合内容の対応を示す。

2.2.6 設計ノウハウの柔軟な折込

設計者個人個人によるバラツキの少ない高品質な設計を行うには、ベテラン設計者の定量化されたノウハウを盛り込んだ標準設計手順の確立が前提となる。そのシステムへの実装にあたっては

- ・技術の変革やニューモデルへの対応に伴う設計ルールの追加・変更
- ・成形性評価基準データの変更
- ・成形ノウハウの蓄積・検索

等が容易にできるだけの柔軟なシステム構造が重要となる。

これは、一般的に「AI システムの設計業務への適用」「設計知識ベース」等の題で論じられている内容とオーバラップするものであるが、本格業務適用という意味でのシステム化のためには、深い業務分析/システム構築の工夫/先端技術の検討/組織としての熱意等、地道でかつ着実な努力が欠かせない。この部分がどこまで実現できるかが、

表1 成形性検討内容与方法^[3]

Table 1 Failures and estimates of sheet metal forming

成形性 検討内容	CAD/CAE による成形性 検討手段	バ	絞	ダイ	部	断	分	材	フ	曲	F
		ネ	り	フェ	品	面	布	料	ラン	弾	E
		ラ	深	ース	形	線	の	移	ジ	性	M
		ン	さ	ス	状	長	計	動	伸	回	解
		ク	等	の	状	変	算	の	び	復	析
		ホ	高	の	線	化	ひ	ず	率	解	析
		ール	線	成	の		み		の		
		ド		成	形				計		
		時		挙	成				算		
		の		動	挙						
		挙			動						
		動									
破	断	○	○				○		○		○
し	わ	○	○	○	○	○	○		○	○	○
面	ひ	○	○	○	○	○	○			○	○
ず	み						○	○			○
す	べ						○	○			○
り	り						○	○			○
き	き						○	○			○
ず	ず						○	○			○
((○	○			○
シ	シ						○	○			○
ョ	ョ						○	○			○
ク	ク						○	○			○
マ	マ						○	○			○
ーク	ーク						○	○			○
))						○	○			○
線	ず	○		○			○	○			○
ず	れ						○	○			○
れ							○	○			○
ス							○	○			○
プ							○	○			○
リ							○	○			○
ン							○	○			○
グ							○	○			○
バ							○	○			○
ッ							○	○			○
ク							○	○			○
ハ							○	○			○
ッ							○	○			○
ク							○	○			○
ハ							○	○			○
ッ							○	○			○
ク							○	○			○
ハ							○	○			○
ッ							○	○			○
ク							○	○			○
ハ							○	○			○
ッ							○	○			○
ク							○	○			○
ハ							○	○			○
ッ							○	○			○
ク							○	○			○
ハ							○	○			○
ッ							○	○			○
ク							○	○			○
ハ							○	○			○
ッ							○	○			○
ク							○	○			○
ハ							○	○			○
ッ							○	○			○
ク							○	○			○
ハ							○	○			○
ッ							○	○			○
ク							○	○			○
ハ							○	○			○
ッ							○	○			○
ク							○	○			○
ハ							○	○			○
ッ							○	○			○
ク							○	○			○
ハ							○	○			○
ッ							○	○			○
ク							○	○			○
ハ							○	○			○
ッ							○	○			○
ク							○	○			○
ハ							○	○			○
ッ							○	○			○
ク							○	○			○
ハ							○	○			○
ッ							○	○			○
ク							○	○			○
ハ							○	○			○
ッ							○	○			○
ク							○	○			○
ハ							○	○			○
ッ							○	○			○
ク							○	○			○
ハ							○	○			○
ッ							○	○			○
ク							○	○			○
ハ							○	○			○
ッ							○	○			○
ク							○	○			○
ハ							○	○			○
ッ							○	○			○
ク							○	○			○
ハ							○	○			○
ッ							○	○			○
ク							○	○			○
ハ							○	○			○
ッ							○	○			○
ク							○	○			○
ハ							○	○			○
ッ							○	○			○
ク							○	○			○
ハ							○	○			○
ッ							○	○			○
ク							○	○			○
ハ							○	○			○
ッ							○	○			○
ク							○	○			○
ハ							○	○			○
ッ							○	○			○
ク							○	○			○
ハ							○	○			○
ッ							○	○			○
ク							○	○			○
ハ							○	○			○
ッ							○	○			○
ク							○	○			○
ハ							○	○			○
ッ							○	○			○
ク							○	○			○
ハ							○	○			○
ッ							○	○			○
ク							○	○			○
ハ							○	○			○
ッ							○	○			○
ク							○	○			○
ハ							○	○			○
ッ							○	○			○
ク							○	○			○
ハ							○	○			○
ッ							○	○			○
ク							○	○			○
ハ							○	○			○
ッ							○	○			○
ク							○	○			○
ハ							○	○			○
ッ							○	○			○
ク							○	○			○
ハ							○	○			○
ッ							○	○			○
ク							○	○			○
ハ							○	○			○
ッ							○	○			○
ク							○	○			○
ハ							○	○			○
ッ							○	○			○
ク							○	○			○
ハ							○	○			○
ッ							○	○			○
ク							○	○			○
ハ							○	○			○
ッ							○	○			○
ク							○	○			○
ハ							○	○			○
ッ							○	○			○
ク							○	○			○

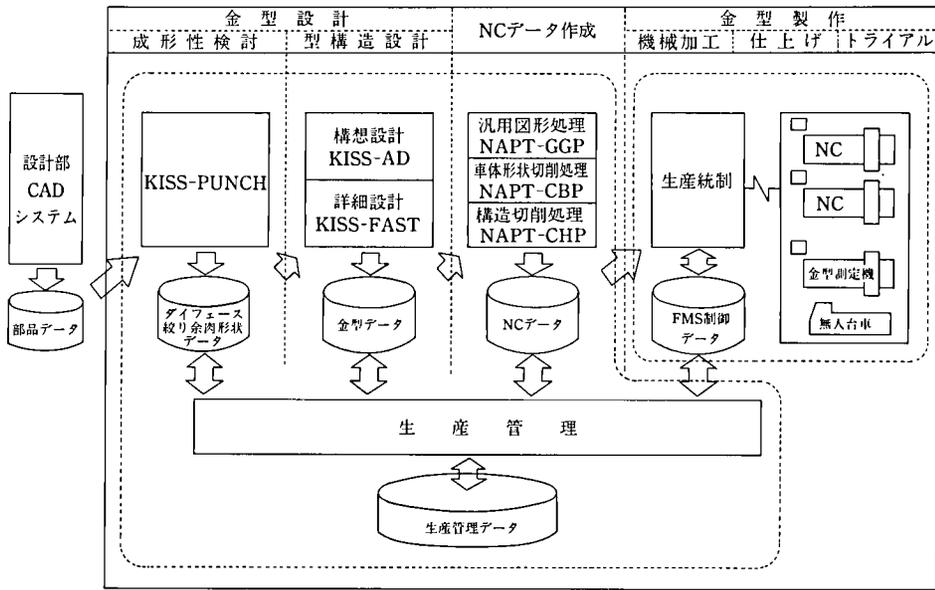


図9 INCAM システム概要図
Fig.9 Outline of INCAM system

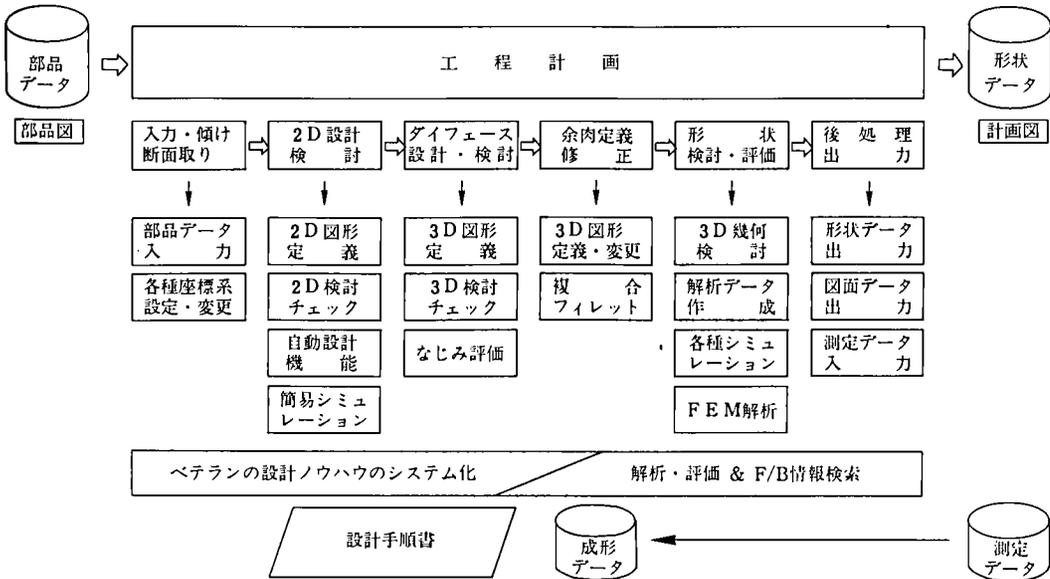


図10 KISS-PUNCH システム機能構成図
Fig.10 Functional components of KISS-PUNCH system

部, 余肉部), 板厚, 板厚側等, 形状データ以外の設計情報を多く持ち, とくに成形性の評価・解析に使用している。

- 2) 設計・解析向けのデータ形式……図 11 のような形状を表現するデータ形式として直線の交点と円弧情報を合わせ持つ曲線データとしているため, 断面形状の設計・変更が簡単に行えるようになっている。また, 後述するように, この形式は

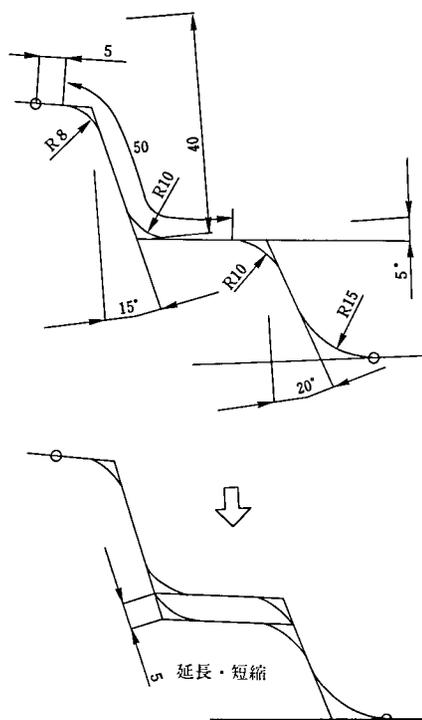


図 11 二次元設計形状

Fig. 11 Design of 2-dimensional curve

二次元の成形シミュレーションにも適している。

- 3) パラメータ設計によるモデリング……三次元形状部においても、とくに余肉、ビード・フィレットに関しては、豊富なパラメータ設計機能による設計工数の低減および品質の安定をはかるとともに、定義時のパラメータを保持・管理することにより、設計変更への対応を容易にしている(図 6)。
- 4) 自動トリミング機能と形状自動復元機能……図 12 に示すように、従来のサーフェスモデルとソリッドモデルの長所を合わせ持つ新しいデータ構造の実現により、操作性・視認性の向上および自動化をはかっている。すなわち、ソリッドモデルにおける自動トリミングや形状復元機能を持つとともに、ソリッドとして不完全な状態でも思った二次元断面が簡単にとれるデータ構造とすることにより、任意時点での形状評価が可能となっている。

3.2.3 二次元と三次元の並行作業環境

2.2.3 項で二次元設計と三次元モデリングの同期の必要性を論じた。

本システムでは、単図モード、モデルモード、図面モードの 3 種類の画面作業環境が用意されている(図 13)。単図モードは、平面図、側面図、断面図、矢視図を表現する画面モードであり、二次元の設計作業は主としてこのモードで行う。モデルモードは三次元操作を行う画面であり、ダイヤルによるローカルビューイング機能がフルに利用される。図面モードは工程計画図に相当する画面モードであり、断面図や矢視図を任意に配置することができる。また単図モードやモデルモードによる図形定義の結

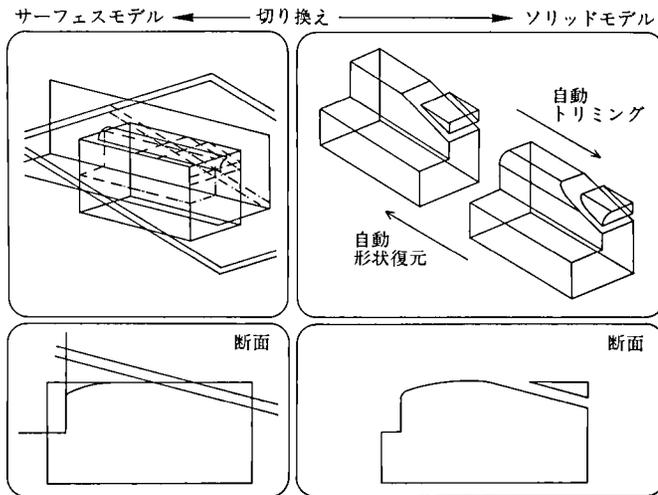


図 12 データ構造
Fig. 12 Data structure

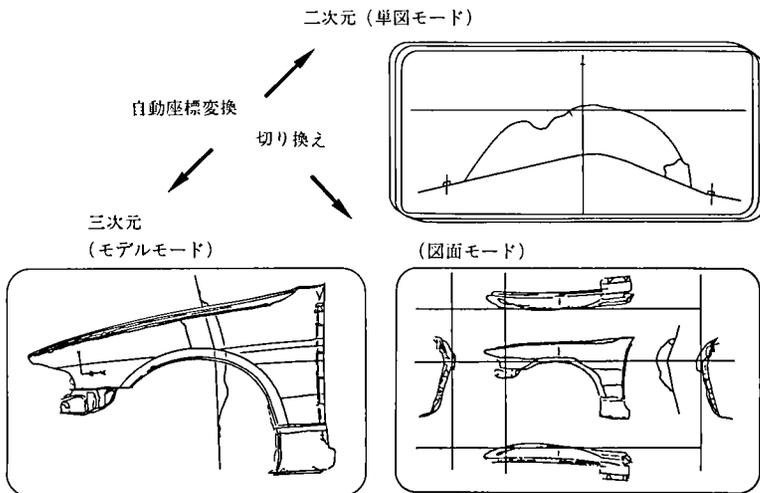


図 13 作業環境切換え
Fig. 13 Exchange of views

果は即座かつ自動的に図面モードに反映される。

設計者はこれらの各モードを瞬時に切り換えることにより、断面線等の二次元設計、曲面等の三次元設計、工程計画図面作成の並行作業を実現している。単図モードで二次元設計された断面形状は、三次元へ自動座標変換され曲面作成に使用される一方、三次元で作った形状線は断面図へ自動投影されるため、二次元と、三次元の間の移行作業をなくすことができている。また図面モードでは、互いに異なる断面図に属する断面線・座標軸線間の交点計算等、従来感覚での図面作業も可能となっている。

3.2.4 成形形状特有のモデリング機能とチェック機能

プレス金型の成形形状は、上流 CAD システムから渡される製品部にダイフェース面、余肉面、ビード面、フィレット面等を順次付加していくことにより構築される。

本システムではこれらの面のタイプごとにコマンドが用意され、入力の手易性、正確な設計意図の反映、面形状変更作業の容易さを実現している。

また、寸法指示とボカシ指示を組み合わせた形状定義機能の実現による作業性の向上もはかっている。たとえば、余肉面に関してはガイド線の数/断面形状のパターン/掃引方式等の組み合わせによる種々の定義バリエーションを持っている。

モデリングでもっとも工数のかかる複合フィレット処理に関しても、コーナ部のガウジ自動除去や面が不足している端末の自動延長機能等の実現により、トリミング作業を含めて煩わしい処理を省いている。

図 14 にモデリングに関する特徴的機能の例を示す。

- ① 円錐台、角錐台、回転体の一括作成を示す。
- ② 境界曲線といくつかの面内断面線からダイフェース曲面を求めている。
- ③ 余肉作成時に隣接曲面間のボカシ、ガウジ取りを行っている。
- ④、⑤ それぞれビード曲面、フィレット曲面の例である。
- ⑥ ボカシ接続と自動トリミングによる曲面の部分変更を示す。

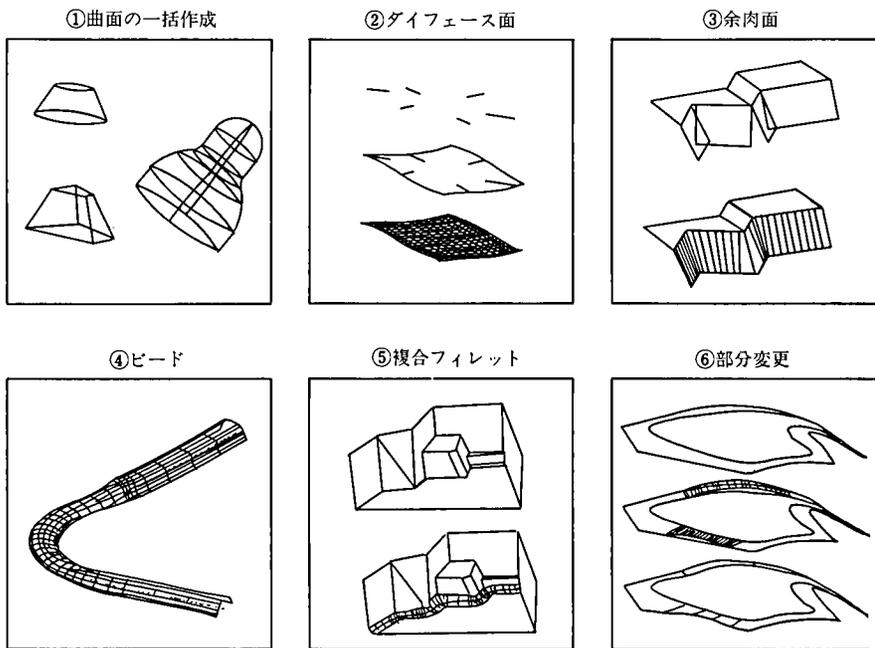


図 14 モデリング機能

Fig.14 Modeling functions

一方、加工方向検討やインパースチェックを始めとする成形形状特有のチェック機能に関しては、できる限り自動化することにより、バラツキのない定量的な設計・評価を可能としている（口絵 4）。

3.2.5 実用的な成形シミュレーション

前述したように、本システムの曲線データ形式の一つは直線と円弧情報の組み合わせで実現されている。このため、図 15 に示す解析式と組み合わせることにより、実用

$$T_{i+1} = (T_i + M/r) e^{\mu \alpha} + M/r$$

M; 曲げモーメント
 r; 曲げ半径
 α; 曲げ角
 μ; 摩擦係数

$$M = 2 \int_0^{t/2} \sigma h dh$$

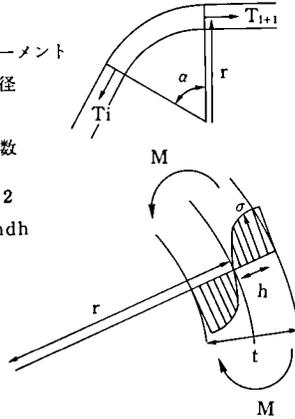


図 15 解析式

Fig. 15 Analytical equation

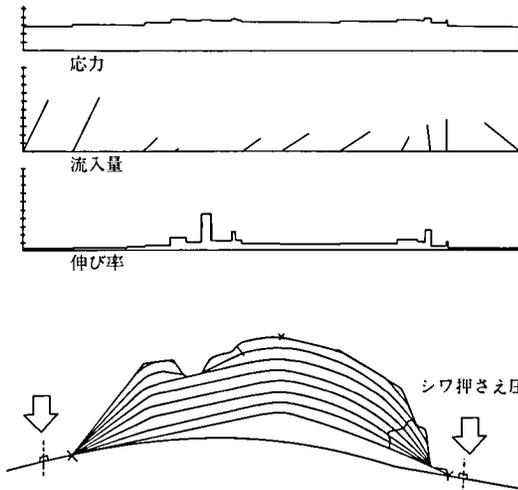


図 16 伸び率・流入量

Fig. 16 Elongation percentage and influx

的な二次元成形シミュレーションを可能としている。すなわち、断面形状を直線と円弧で近似した解析モデルとすることにより解析要素数を少なくできるうえ、成形過程における型とパネルの接触状態を正確に求められるため、図 16 のような成形各部の伸びや流入量（擦り傷位置）を短時間（秒以下）で精度よく求めることが可能となっている。本データ形式は、前述のようにパラメトリックな形状変更も容易であり、将来的には最適形状の自動探索も可能である。

設計した断面形状からブランクホールド時のパネル形状を短時間でシミュレーションする FEM 解析機能も実現しており、主要な断面形状のみから、自然な形のしわ押さえ面を簡単に作成することができている。

図 17 に部位別の成形シミュレーション機能の例を掲げる。

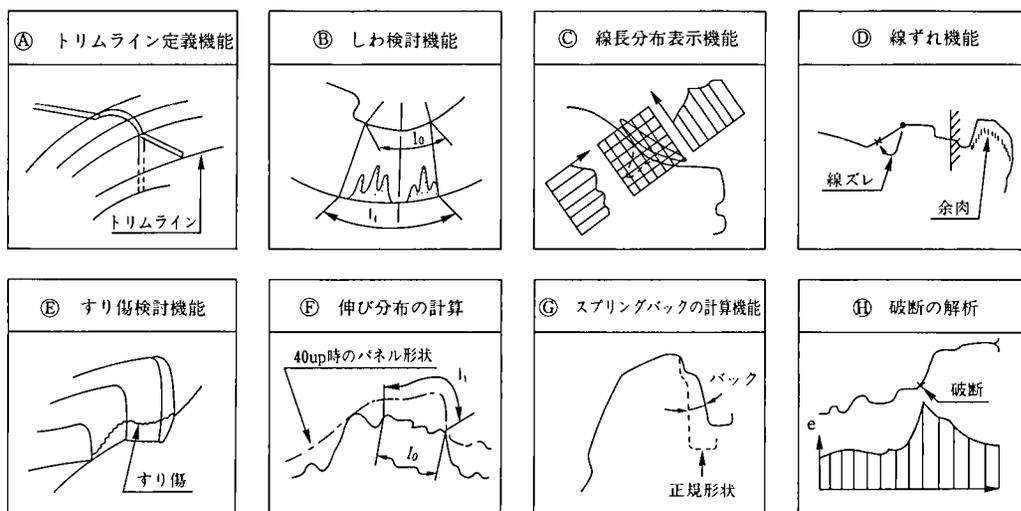


図17 成形シミュレーション機能

Fig. 17 Simulation functions of sheet metal forming

4. おわりに

筆者らの「KISS-PUNCH」システム開発の経験をふまえて、

- ・プレス金型形状設計CADとして必要な機能要件
- ・該システムの機能特徴

について解説した。

本システムの業務適用の結果、旧来に比べて設計・データ化の総工数が約半減された。同時に、定量的な評価に基づく形状設計のための標準手順が確立されたことにより、旧来ベテラン設計者に付随していた各種ノウハウの共有化・統一化、結果としての品質の向上、リードタイムの短縮が実現できた。

今後は、2.2.6項「設計ノウハウの柔軟な折込」でも述べているように、工程設計業務/成形形状設計業務における設計ノウハウのシステム化のレベルをより高めることが大きな課題である。

本稿が、プレス金型設計・製作関係者にとって参考になれば幸いである。

- 参考文献 [1] 薄鋼板成形技術研究会編、「プレス成形難易ハンドブック」, 日刊工業新聞, 1987年3月20日.
- [2] M. Kawka, A. Makinouchi, "Finite element simulation of sheet metal forming process by simulation use of membrane, shell and solid elements", UNIFORM92.
- [3] 氏原 新, 「自動車車体用金型および設備における三次元シミュレーション」, 塑性と加工 第31巻第350号 (1990-3).

執筆者紹介 沢田 晃二 (Kouji Sawada)

昭和53年京都大学大学院 修士課程(精密工学科)修了。
同年日産自動車(株)入社。現在工機設計部所属。



井上 博順 (Hironori Inoue)

昭和60年早稲田大学大学院 修士課程(機械工学科)修了。
同年日産自動車(株)入社。現在工機設計部所属、日本
塑性加工学会員。



前田 幸久 (Yukihisa Maeda)

昭和47年東京大学理系大学院 修士課程(数学科)修了。
同年日本ユニシス(株)入社。昭和53年よりCAD/CAMシ
ステム開発に従事。現在製造工業システム本部システム三
部所属。



統合ドキュメント・マネジメント・システム
Interleaf 5

1. はじめに

Interleaf 5はInterleaf社により開発された統合ドキュメント・マネジメント・システムである。Interleaf社は、1981年に設立、3年間におよぶ文書処理システムの研究開発を経て最初の製品TPS (Technical Publishing System) を発表し、UNIXをベースにした電子出版の分野において成功をおさめた。このTPSをベースに、アクティブ・ドキュメント技術やオブジェクト指向等の最新のテクノロジーを取入れ、1990年新たに発表されたのがInterleaf 5である。全世界で13万セット以上の実績を誇るこの商品は13か国語に対応し、1992年に14番目の言語として日本語バージョンを発表した。

各企業がビジネススピードの向上・品質の向上・コストの低減を図るためには、各種システム化の推進が重要な課題である。その一つの方策として、既存システムと連動する効果的なドキュメント処理システムの構築をあげることができる。Interleaf 5は、単にWS(ワークステーション)を使用し会話操作で美しくドキュメントを作成する道具だけではなく、既存システムとの融合を可能にする各種ツールを提供している。Interleaf 5は、DTP (Desk Top Publishing) としての基本機能を提供するとともに、プログラム開発環境および他システムと融合した実行環境を提供するドキュメント・システム構築時のSI (System Integration) ツールと位置付けることができる。

それゆえ、Interleaf 5が活用される分野は広く、製造・流通・金融・政府/官公庁等のあらゆる業界で使用されている。特に、製造業においては、

- ・自動車産業
- ・電機、機械産業
- ・化学、製薬業界
- ・建設、建築業界
- ・旅客航空、航空機製造業

等で多数活用されている。

製造業においても、OA化によるビジネススピ

ードの向上・品質の向上・製造コストの低減は、重要な課題といえる。とくに、生産・技術・販売の各分野を対象にトータルなOA化を実現するため、製造業特有の業務(生産管理・CAD/CAM/CAE・図面管理等)に関するEOA (Engineering Office Automation) システムを整備・構築することが必要不可欠である。Interleaf 5は、生産・技術・販売の各分野で稼働する既存システムおよびD/B (Data Base) と柔軟に連携し、ドキュメント(技術文書・作業指示書等)の作成・編集/管理/検索・配布の処理を具現化できるEOAシステム構築の要となるソフトウェアである。

Interleaf プロダクトは次の三つの製品から構成される。

- ・Interleaf 5 : 文書作成・編集
- ・RDM (Relational Document Manager)
: 文書管理
- ・WorldView : 検索・配布

これらの製品を適切に組み合わせて活用することにより、OA化の効果を一層高めることができる。

今回は、Interleaf 5(日本語バージョン)の特徴と機能について適用例・用途・操作イメージを中心に紹介する。なお、RDMとWorldViewの各日本語対応版は現在開発中(年内出荷の予定)であり、英語版で提供されている機能概要の紹介にとどめる。

Interleaf 5は、日本ユニシスが取り扱うUSファミリー上で稼働するほか、IBM社・HP社のUNIXプラットフォーム上でも稼働するソフトウェアであり、オープンシステムの推進役として必要な要件を備えている。

2. Interleaf 5の特徴

1) パワフルなパブリッシング・システム……

Interleaf 5は、基本的なDTP機能に加え大量ドキュメントのフォーマット・ページ番号・参照図番等を自動的に調整する編集機能を提供する。また、目次・索引の自動作成機能を提供し、締切間際のドキュメントに修正が発生しても、ページ番号・参照図番・目次・索引等の再編集をパワフルに処理することができる。言語は、日本語を含め世界14か国語

に対応しており、多国語対応のドキュメント作成が可能となる。フォーマットが同一であれば対応する言語にテキスト部だけを作成すれば多国語対応が可能となる。

グラフィックスは、500種以上のクリップアートを含め豊富なグラフィックス・ライブラリ（地図・化学式・矢印等）が提供され、グラフィックス作成時の部品として活用ができる。

2) ネットワークをフル活用したグループワークを支援……膨大なドキュメントの作成に際し、複数の作成者が（たとえば各章を）同時に異なる場所で作業を同時並行して実施することができる。この場合、個々に作成された複数のドキュメントは各々固有の章立て・図番・ページ番号等を持つことになる。しかし、Interleaf 5は、ネットワーク（LAN環境）を活用し複数のドキュメントを1冊の本（BOOK）として取り扱うことができる。このブック機能は、対象となるすべてのドキュメントの章立て・図番・ページ番号等を自動的に調整し、全体（複数のドキュメント）を1冊の本として仕上げることができる。

3) 各種標準に対応……ユーザ・インタフェース、ネットワーク、テキスト、グラフィックス等の各分野において標準と呼ばれる（代表的な）仕様に数多く対応している（表1）。

ISO9000シリーズ（国際標準化機構がまとめた品質管理の指針・品質保証についての要求項目とその使用のためのガイド・ライン）は、現在世界50か国以上で採用されており、国内企業にて注目されている国際標準規格である。SGML（Standard Generalized Markup Language）は、文書の論理構造や意味構造を簡単なマークで記述する汎用のマークア

ップ言語であり、文書交換の標準規格としてISO 9000シリーズで採用されている。Interleaf 5は、Interleaf 5 <SGML>（欧米版）をすでに製品化し出荷しており、海外企業で多数採用され実用化に至っている。先進の業界は、ISO 9000シリーズのガイドラインにそってSGMLに準拠した業界独自の文書規約を定義する段階に入っている。SGMLを採用した代表的例として自動車業界のJ2008や旅客航空業界のATA-100等を挙げることができる。Interleaf 5は、各業界で定義されるこれらの規約に対し、リーダーシップをもって今後とも積極的に支援していく予定である。

4) カスタマイズ可能なソフトウェア……アクティブ・ドキュメントは、ドキュメントを一つの媒体として捉え、他システム上のアプリケーションを起動したり、データの連携を実現するための概念である。Interleaf 5は、このアクティブ・ドキュメントを実現するため、言語としてInterleaf Lispを提供し、オブジェクト指向によるプログラミング開発環境（Developers Tool Kit：以下DTKと呼ぶ）を提供している。このDTKにより、ドキュメントからCAD/CAM/CAE, CASEツール、表計算ソフトウェア等の豊富な第三者ソフトウェアを含む各種アプリケーションを起動したりデータ連携を実現させることが可能である。

DTKは、ユーザニーズにそったカスタマイズを効率良く実現することを支援するツール群で構成されている。

5) 先進性と協調性……Interleaf 5の特徴は、アクティブ・ドキュメント・テクノロジー/オブジェクト指向/ワークグループ作業の支援等を取り込んだ先進性と、各種標準仕様の対応/

表1 標準について
（注：欧米版にて対応中/日本語対応は計画中）

テキスト	グラフィック	アウトプット	ネットワーク
JIS	Calcomp 925/960	Kanji PostScript PostScript	イーサネット トークンリング NFS
シフト JIS	TIFF		
EUC	CGM		
ASCII	PostScript/EPS		
nroff/troff	HPGL		
SGML	IGES		
英語・日本語ワープロの 有力ソフトのフォーマット	PICT/PICT 2 等		

マルチプラットフォーム対応/他のアプリケーション（第三者ソフトウェアを含む）との連携を実現する協調性にあり、その実績は類似ソフトウェアの比ではない。

3. 機能概要

ここでは、個々の詳細を割愛し、とくに Interleaf 5らしさを象徴している機能について紹介する。

- 1) 基本機能……Interleaf 5 の基本機能は、テキスト入力、グラフィックツール、作表ツール、数式作成ツールテキストおよびグラフィックスに対応する各種フィルタ（欧米版）で構成される。

Interleaf 5 を特徴付ける大きな要素は、ドキュメントが構造化されていることである。Interleaf 5 は、ドキュメント内の各構造（たとえば文書構造として、章・節・項・本文…）ごとにプロパティ（属性）が定義できる。同一属性を持つテキストの塊をコンポーネントと呼び、個々に名前をつけることができる。このコンポーネントに付けられた名前が、膨大なドキュメントを対象にした追加・修正・削除等の編集作業において、非常に大きな意味を持つ。

一つの例は、あるドキュメントの章を記述する場合、たとえば「1章 はじめに」と記述された文字の並びを一つの塊として捉え、

この塊に対し『章』と言う名前を付ける（図 1）。ここで付けられた『章』はコンポーネント名であり、この『章』に各種の属性が定義できる。属性として、文字の大きさ・字体・左右の詰め・センタリング・上下左右のマージン等が指定できる。定義されたコンポーネントはドキュメントの任意の場所で使用することができる。各章（2章・3章…）の記述がコンポーネント『章』を使用し記述されていれば、任意の『章』コンポーネントの属性を変更するだけで、ドキュメント内のすべての『章』コンポーネントに適用し属性を統一することができる。この適用により、各章のテキスト（文字）の表示も自動的に統一される。本文用のコンポーネント『本文』を用意しておき、各章の中で記述される本文が『本文』コンポーネントを使用していれば本文に対しても同様に機能させることができる。この機能はドキュメントの表示属性（字の大きさ・字体等）の変更の際に非常に有効な機能である。膨大な文書に対し一つの操作で関連するテキストの属性が統一され表示形式のすべてが変更（更新）される。この更新される様を見たとき Interleaf 5 のパワフルさを実感することができる。

二つ目の例としてグラフィックスを挙げることができる。各グラフィックス（イメージを含む）もプロパティを持っており、属性の

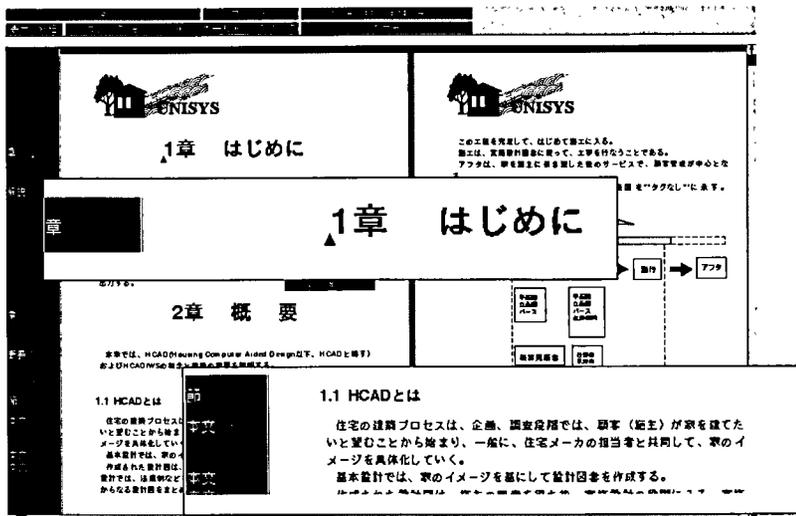


図 1 ドキュメント例

一つとして名前を付けることができる。便宜上、グラフィックス機能により機械部品の図を描画したとする（実務ではCADからの図面ファイルを取り込むことになる）。描かれた機械部品に対しグラフィックスの名称として『部品 A』と名付ける。名称が付けられたグラフィックスを「名称付きオブジェクト」と呼ぶ。名称付きオブジェクトは、ポップアップメニューに自動的に追加登録され、作成したグラフィックスのメニュー化が図られる。このメニューによりドキュメント内に簡単に複製をペーストできるようになる。ドキュメントにペーストされた各グラフィックスは同一の名称

『部品 A』を持つことになる。名称付きオブジェクトの特徴は、グラフィックス『部品 A』の形状が変更された場合でも、任意場所にペーストされた『部品 A』の内容（形状そのもの）を変更（更新）すれば、グラフィックス『部品 A』がどの章のどのページにペーストされているにもかかわらずすべてのグラフィックス『部品 A』を対象に、変更後の形状へ一括して更新できることである。

また基本機能の一つとして、ドキュメント内で割り付けられた章番号・節番号等の連番を自動的に調整するオートナンバと呼ぶ機能を提供している（図2）。たとえば第1章の第

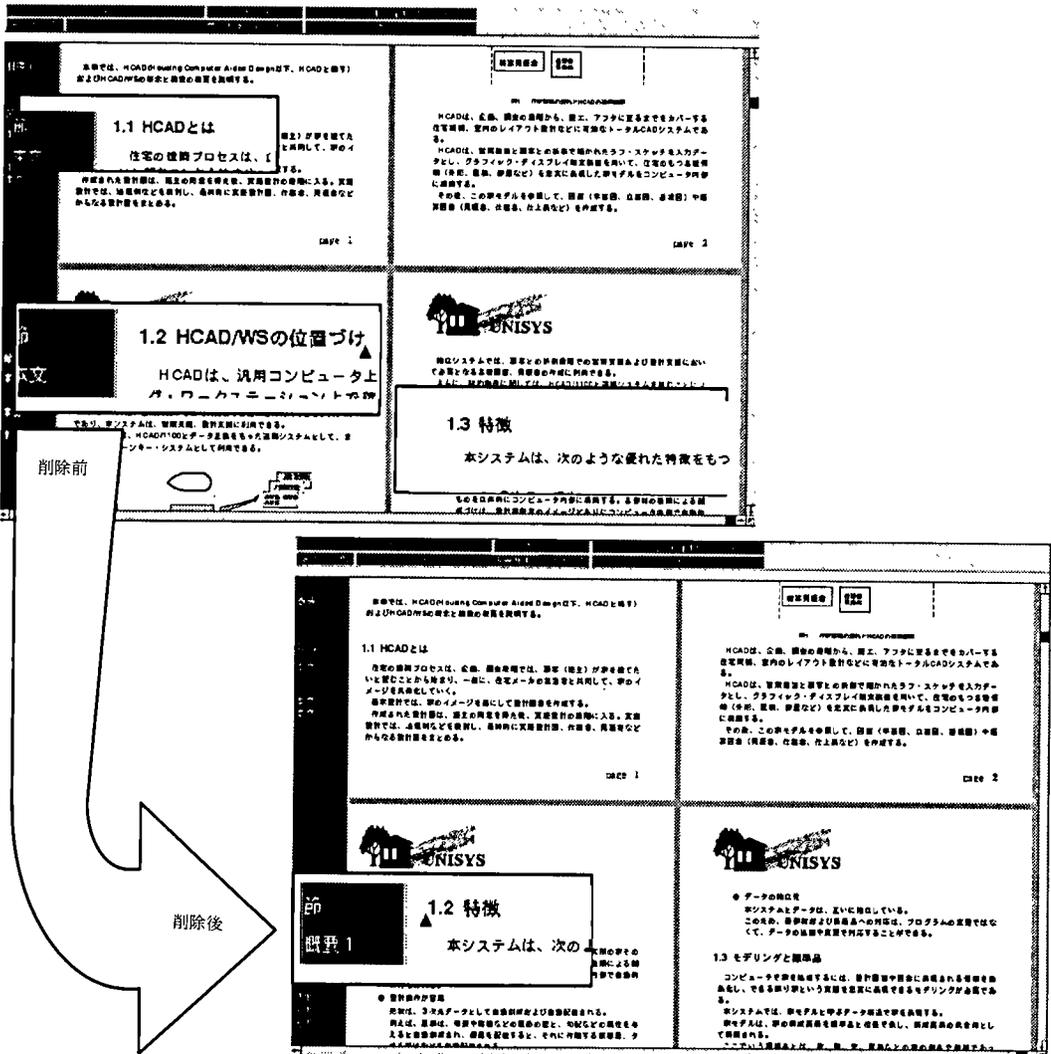


図2 オートナンバの例

2 節 (1.2 HCAD/WS の位置付け) をすべて削除したとき、その後の第 1 章第 3 節 (1.3 特徴) が第 1 章第 2 節に繰り上がる。当然、その節以降もすべて繰り上がり、各章・節の連番としての整合性を自動的に保持する。仮りにその削除した節を異なる章に挿入しても各章・節の連番は同様に整合性を保つ。参照されるグラフィックス (図形・イメージ) に付けられた番号がオートナンバであるなら、そのグラフィックスの位置 (表示されている章やページ) を移動しても、自動的に最初のグラフィックスから番号を割付けることができる。

本文内でその移動された図を参照する記述があった場合、オートナンバにより移動後に割り付けられた図番を自動的に引用し、本文内の記述と参照図に割り付けられた番号の整合性を保証するオートリファレンスと呼ぶ機能も提供している。少量のドキュメントであれば手作業にて片付くかも知れないが、数千ページにおよぶメンテナンスマニュアル等の膨大なドキュメントの図番が変更されたとき、手作業でその図番を引用している本文との整合性を完全に保証することは不可能に近い。この機能も Interleaf 5 のパワフルな一面と言える。

つぎに、基本機能の中で提供されるブック機能について紹介をする。文字通り、ブック機能は、複数のドキュメントを 1 冊の本としてメイクアップする機能である。操作上ブックは、本の形をしたアイコンで表示される。アイコンを開くとブック機能を提供するウィンドウが表示される。ブック機能の中でも前述したオートナンバおよびオートリファレンスの機能は有効である。開かれたブックウィンドウ内に点在した複数のドキュメントアイコンをカット&ペーストで取り込むだけでよい。ドキュメントアイコンの並び (左から右、上から下の順) に従って、オートナンバ機能により章・節・ページの各番号がすべて通し番号となる。つまり、ドキュメントアイコンの移動により、複数のドキュメントを 1 冊の本として編集し仕上げる作業が自動化できる。この機能を応用し、各ワークステーションを LAN で結び複数のメンバが膨大なドク

ュメントを分担し各自が同時に作業を手掛けることにより、大幅な作成時間の短縮を図ることができる。

Interleaf 5 は、前述の機能の他、数式入力・グラフ作成・目次/索引の自動生成・表作成等数多くの機能を基本機能の中で提供している。

Interleaf 5 の基本機能は Interleaf 5 の用途を示唆しており、テキストやイラストを組み合わせてデザインすることが目標となる一般的な DTP システムの機能に加え、ドキュメントの構造 (各種プロパティを含め) をテンプレートとして定義しておくことにより、大規模ドキュメントの作成編集作業を柔軟に処理できる機能を提供している。

2) オプション機能

① カタログ機能

カタログ機能 (オプション) は、ブック機能 (基本機能) を拡張しパワフルなフォーマット機能を提供する。カタログ機能は通常の操作で作成されたドキュメントをカタログと定義 (宣言) するだけで、その機能を実行させることができる。カタログと宣言されたドキュメントはスーパー・スタイルシートと呼ばれる。

カタログ機能は、ブックウィンドウ内に取り込まれた複数のドキュメントのヘッダ/フッタ等の表示が異なっても、すべてのドキュメントを対象にスーパー・スタイルシートで記述した内容に統一することができる。

同様に、コンポーネントおよびテキスト部をスーパー・スタイルシートに登録しておけば、ブック内の複数ドキュメントを対象として、同一のコンポーネント名を使用しているすべてのドキュメントに対しコンポーネントの属性およびテキストをすべて統一できる。文字の大きさ等の変更の場合は、スーパー・スタイルシート内の同一名称を持つコンポーネントの属性を変更するだけでよい。適用されるドキュメント内の変更は不要である。

グラフィックスに対しても同様で、スーパー・スタイルシートに登録されたグラフィックスの名称と適用される全てのドク

メント内で定義されたグラフィックスの名称が同一であれば、その内容をスーパー・スタイルシートの内容に統一する事ができる。

カタログ機能を実行させるには、ブックウィンドウ内の左上にカタログアイコンを置き、その右に編集したいドキュメントアイコンを配置すればよい。複数のドキュメントを対象とする場合は、左から右、上から下とブックウィンドウ内にドキュメントアイコンを並べて行けばよい。オートナンバ/ページナンバは、カタログアイコンに近いドキュメントから順に割り付けられ、各章・節番号等を自動編集する。

膨大なドキュメントのフォーマット（カタログ）を複数作成しておけば適用したいフォーマットのカタログアイコンをブックウィンドウの左上に移動するだけで、各ドキュメントのフォーマットを統一してくれる。

カタログ機能は、スーパー・スタイルシートで定義されたコンポーネント・名称付オブジェクト・オートナンバ・ページナンバヘッダ/フッタ・表等の属性および内容をブックウィンドウ内の複数のドキュメントに送り出す機能を提供している。

② 履歴管理機能

履歴管理機能は、ドキュメントに対し履歴管理オンと設定する操作により開始される。

自動的に記録される情報は、

- ・変更時の改訂レベル
- ・変更を開始した日時と担当者
- ・一番最近に変更した日時と担当者
- ・追加データか削除データかを示すフラグ（＋で表示）

等であり追加・変更・削除されたデータはその差分だけを記録しているため改訂におけるディスクスペースを大幅に節約することができる。

履歴管理機能は、改訂作業を進める中で任意のバージョンを基にバージョンの直列並列管理が可能であり、各バージョンの変遷をグラフィカルなツリー構造で確認できる。また、任意のバージョンに遡り表示させることができる。表示中のドキュメント

に対して、過去に削除されたデータをすべて含めてドキュメント内に表示することも可能である。

③ 開発支援ツール DTK

DTKは、ユーザーニーズに合わせたカスタマイズを担当するプログラマを支援するツールキットであり、Interleaf Lispを言語とするプログラム開発環境を提供する。

DTKはドキュメントシステム構築において、ユーザインタフェースの開発だけでなく数多くの第3者ソフトウェアやレイヤード・アプリケーション（ここでは、Interleaf Lispにより作成された固有のアプリケーションの意）との統合により、ユーザーにそったアクティブ・ドキュメントを実現する。

DTKが提供するツール群は、以下の通りである。

- ・Interleaf Lispで作成されたソースコードエディタを提供（プログラマの環境に合わせた機能拡張も可能）
- ・Interleaf Lisp式の評価を対話型で行うリスナを提供
- ・ハングアップやLispコード内のバグ追跡をプログラマが正しく処理できるよう支援するデバッグ用ツールを提供
- ・Interleaf 5のプラットホーム間で、LAN (TCP/IP) を用いストリームベースのデータのやり取りを可能にするIPC (Inter Process Communications) ファシリティを提供
- ・Leafwareと呼ぶサンプルプログラムをオンラインツールとして提供
- ・各カテゴリ（シンボル・機能・クラス・キー・ポップアップ・内部変数・Interleaf Lisp）に分けられたオンラインヘルプ機能を提供
→カテゴリが不明の場合でも一般ヘルプ機能を用いることができる。

プログラム開発環境の特徴は、各種のユーザーニーズに柔軟に対応できる環境を提供し適用分野を拡大するとともに、オブジェクト指向の採用によりプログラム（システム）のメンテナビリティを大きく向上させることである。

4. アクティブ・ドキュメントの適用イメージ

アクティブ・ドキュメントを実現するためのツールとしてアクティブ・リンクツール（日本語対応版検討中）を使用し、他のアプリケーションによって作成されたファイルの最新のデータを取り込むことができる。

また、ドキュメントをハイパーテキスト（メディア）として活用することもできる。これはドキュメント内のテキストやグラフィックスをマウスクリックすることにより任意の処理（たとえば、特定ドキュメントの特定ページを開く処理等）を起動し実行する。

これらの機能を組み合わせることにより、取り込んだデータを基に評価し加工処理した結果を反映させたドキュメントを作成することができる。このとき実行されるプログラム（またはアプリケーション）は仮に、第3者ソフトウェアの統計パッケージであっても、独自開発されたエキスパートシステムであっても良い。アクティブドキュメントは、他のアプリケーションと連動するドキュメントシステムを含む業務アプリケーションを実現する。

5. 日本語対応計画中の主な製品

- 1) RDM (Relational Document Manager) ……現在欧米版にて提供されている RDM は Interleaf プロダクト・ファミリの主要な製品の一つである。ドキュメントに関連する各種属性情報を含めて RDB (Relational Data Base) に格納し、ドキュメント・システム全体を管理する機能を提供する。ドキュメントのアクセスは、すべての RDM 経由で行なわれ、アクセス権の管理および関連情報を記録管理することができる。

たとえば、あるプロジェクトのセキュリティを管理するため、ドキュメントの属性として、テーマ・期間・グループ・メンバ等を定義しておき、表示・編集を含むドキュメントの制御やドキュメントの参照履歴を管理することが可能である。また、プロジェクトの予定・実績に関する情報により、進捗状況表等の生成も可能になる。

RDM は、このようにプロジェクト管理等を支援するグループウェア製品とも位置付けできる。

- 2) WorldView……WorldView は、データを一元的に管理し各ユーザのドキュメントの参照（改変は不可）を WYSIWYG に実現する検索システムである。ネットワークは LAN および公衆回線等の利用が可能であり、ハードウェアは、PC（パーソナル・コンピュータ）を含め多機種の活用が可能となっている。検索機能は、キーワード登録が不要なフルテキスト検索が可能で、検索時、本文中のテキストに合致するドキュメントを探すことができる。ドキュメントの参照セクションにとっては、必須の機能と言える。

6. おわりに

ドキュメント・システム構築ツールとして Interleaf プロダクト・ファミリーへの期待は大きい。製造業における EOA システム構築の大きな要であり、また出版業界の熱い視線も感じられる。しかし、現状における日本語化の対応は Interleaf 5 であり、続いて RDM と WorldView が日本語対応され年内出荷の見込みである。この3製品が揃うことにより、今以上に Interleaf 5 の真価が発揮できるようになる。

その意味からも、日本における Interleaf プロダクト・ファミリーの歴史はこれからはじまる。Interleaf 5 ができあいの DTP でないことは前述したが、エンドユーザが活用を始めるに当たり、これを支援する基本的な操作教育やカスタマイズにいたる教育支援体制の確立が重要といえる。弊社においては、この製品をすべての分野に適用できる全社プロダクトとして位置付け、支援体制を含む総合的な体制整備を強力に進めている。

- UNIX オペレーティングシステムは、UNIX System Laboratories, Inc.が開発し、ライセンスしている。
- Interleaf, WorldView, RDM, Activelink Tool は、米国 Interleaf 社の登録商標である。
- HPGL は、米国 Hewlett Packard 社の登録商標である。
- NFS は、米国 Sun Microsystems 社の登録商標である。
- その他、本文中の個有名詞は、一般に各メーカーの商標、登録商標、製品名である。

**統合図面管理・出図システム
EDMICS**

1. はじめに

EDMICS (Engineering Drawing Management and Information Control System) は、従来の手書き図面/CAD 図面/ホスト内属性データといった、複数の独立したデータファイルとして運用されている図面/技術文章情報を整合性のあるデータベースとして管理・コントロールするためのシステムであり、技術部門における図面管理業務および設計情報管理業務の包括的な改善を可能にするものである (図1)。

2. 特徴

- 1) 図面情報を中心とする技術情報データベースの統合化
 - ① 従来ばらばらに保管されていた図面、関連情報を一括同期管理
 - ・図面実体情報 (手書き図面, CAD 図面, マイクロフィルム, …)

- ・図面属性情報 (図面付加情報)
 - ・技術資料 (テスト報告書, …)
 - ・仕様書類 (見積設計書, …)
 - ・関連資料 (カタログ, …)
- ② 保管メディアを意識せず、図面とその属性情報が一体に見え、かつ取り扱うことができる。
- 2) マルチメディア
 - ラスター (イメージ), ベクター (図形), テキスト (文書) 情報を統合して処理・管理することができる。
 - 3) 汎用ワークステーション/サーバ/LAN による分散処理
 - ① オープン・システム
 - UNIX, Ethernet, TCP/IP, CCITT/MMR, Indexed-RLC, EUC, RDBMS 等の業界標準の採用
 - ② ネットワークと分散処理
 - LAN, TCP/IP, RPC によるサーバ/クライアント方式により、ネットワーク環境の多様なニーズに対応することができる。また分散環境により拡張性、柔軟性に優れている。
 - 4) オブジェクト・オリエンテッド・ユーザインタフェース
 - デスクトップとアイコンの採用により、コ

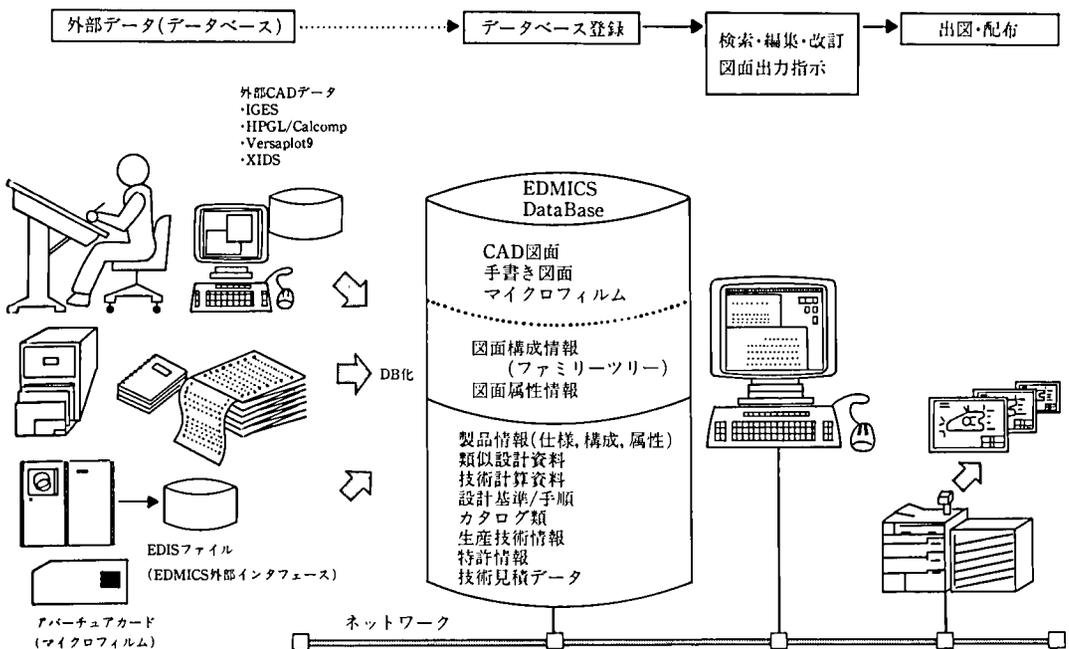


図1 統合図面管理・出図システムのイメージ

ンピュータやワープロのプロでなくても容易に使える、使い勝手のよいユーザインタフェースを提供している。

3. EDMICS システム構成

EDMICS は「作図、図面管理、検索指示等の作業用ステーション」「図面入力ステーション」「大容量データベースサーバ」「大容量光ディスクサーバ」「プロットングサーバ」等から構成される。

各ステーションの機能、および各サーバの機能は、独立したソフトウェア・コンポーネント・モジュールとして存在するので、同一のワークステーションに複数のコンポーネント・モジュールを混在して設置することもできる (図2)。

4. EDMICS ソフトウェア構成

EDMICS のソフトウェア構成は、「図面入力/作図・編集/図面管理/検索/出図指示等の作業を行うデスクトップアプリケーション・ソフトウェア」, 「プロットングサービス/光ファイル管理サービス/データベースサービスを行うサーバ・ソフトウェア」および「外部データを EDMICS で活用できるデータに変換するコンバージョン・ソフトウェア」から構成されている (図3)。

1) デスクトップアプリケーション・ソフトウェア

・ED-Desktop (User Interface)

ユーザは、デスクトップ上にシンボル化されたアイコンをマウスで操作することにより作業を進めることができる。

・ED-Manager

ネットワーク上に接続されたサーバとコネクションを張り、図面/属性/構成データの格納・管理・検索および出図管理を行う。

・ED-DRAW

二次元製図機能の提供。ベクタ図面の作成/編集, ラスタ図面の編集およびベクタ図面とラスタ図面の合成編集を行うことができる。

・ED-Scan

A0 スキャナをコントロールして、図面イメージを取り込む。

・ED-Scan 12

A3 スキャナをコントロールして、図面イメージを取り込む。

・ED-Scan M

マイクロフィルムリーダ装置をコントロールして、アパーチャカードより図面イ

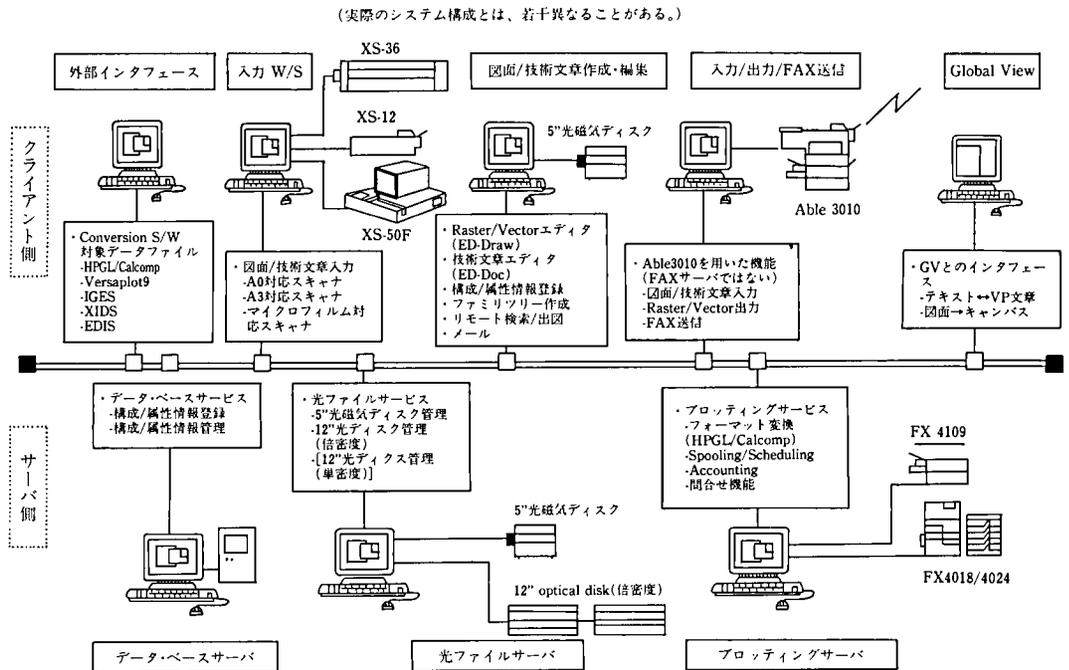


図2 EDMICS システム構成図 (例)

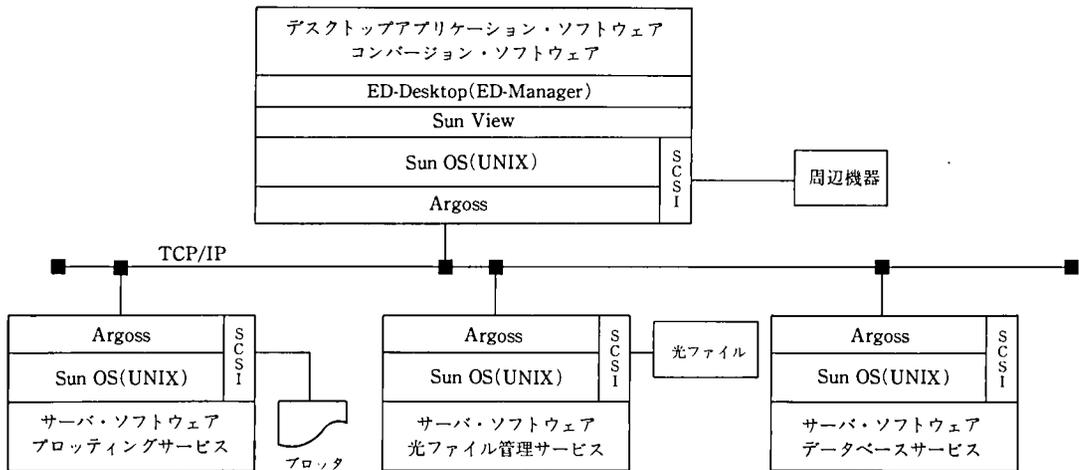


図3 EDMICS ソフトウェア構成

メージおよびパンチデータを取り込む。

- ED-Tree
図面の構成情報（ファミリーツリー）をビジュアルに表示/作成/編集することが可能。
- ED-Mail
デスクトップ上のメール送信/受信アイコンを用いて、EDMICS のデスクトップ間でデータのやり取りを行うことができる。
- ED-XNS
EDMICS と 8000 INS (J-star) および Argoss (Gloval View) との間で、データ転送を行う。
- ED-Storage
5 インチ光磁気ディスクに図面を保管することができる。またネットワーク上の他のワークステーションに接続された 5 インチ光磁気ディスクへのアクセスも可能。
- ED-Fax
Able 3010(富士 Xerox 社製品)を制御することにより、スキャン入力（イメージデータの取込み）、プロット出力および遠隔地 Fax への送信を行うことができる。
- ED-Doc
デスクトップ上で、技術文章、帳票類、マニュアル等のさまざまなドキュメントを作成・編集することができる。
- ED-Shell
アプリケーションとユーザとの対話をと

りもつインタフェースを提供。ユーザインタフェースの変更や再構築が可能である。

- ED-Doc Process Interface
ネットワークにある EDMICS と異なるシステムのデータを ED-Doc のドキュメントに取り入れるためのインタフェースを提供する。
- 2) サーバ・ソフトウェア
- Plotting Service (PLS)
クライアントの指示に基づき、出図処理を行う。
 - Optical Filing Management Service (OFMS)
図面データを格納する光ディスク (12 インチ光ディスク/5 インチ光磁気ディスク) の管理を行う。
 - Relational Data Management Service (RDMS)
図面の属性情報、構成情報の管理を行い、クライアントの指示に基づき、図面および構成情報の登録、検索、図面管理のサービスを提供する。
 - RDMS Batch Registration
E. D. I. S フォーマットの属性データ、構成データ、図面データを EDMICS のデータベースへ直接登録する機能を提供する。
- 3) コンバージョン・ソフトウェア
- PLS/Format Conversion
-PLS/Forant Conversion (HPGL 用)

- PLS/Format Conversion (Calcomp 用)
- PLS/Format Conversion (Postscript 用)
- Raster Conversion
 - EDMICS/Raster Conversion (HPGL 用)
 - EDMICS/Raster Conversion (Calcomp 用)
 - EDMICS/Raster Conversion (Versaplot 9 用)
- Vector Conversion
 - EDMICS/Vector Conversion (IGES 用)
 - EDMICS/Vector Conversion (DXF 用)
 - EDMICS/Vector Conversion (BMI 用)
- Data Conversion
 - EDMICS/Data Conversion (E. D. I. S 用)

5. 適用分野

- 製造業一般
- 大規模プラント, 設備
- 建築, エンジニアリング, 建設
- 官公庁
- 通信, 航空会社
- 流通, サービス業

EDMICS は種々の業種に適用することができる。とくに次のようなニーズに対して効果的に適用できる。

- 種々のサイズの図面を多数保管・管理している。また図面を複数の形式で持っている。
- 図面の保管場所を統合したい, また図面の管理コスト, 複写コストを減少させたい。

- 複数の部門間で正確にかつタイムリにドキュメントのアクセスをする必要がある。

6. おわりに

EDMICS は 1989 年に富士ゼロックス社において国内開発され販売開始した。その年の日刊工業「10 大新製品賞」を受賞し、現在までに 400 システム以上、2000 ノードの実績を持っている。

EDMICS は現在も富士ゼロックス社においてエンハンス開発が続けられており統合図面管理・出図システムとして常に業界標準の最新技術を取り入れながら進化を続けている。

EDMICS は技術部門のさまざまな技術情報を統合化し効率的に運用するためのシステム・インテグレーションツールとして今後ますます期待されるシステムである。

EDMICS および Able は富士ゼロックス株式会社の登録商標です。

次の名称は富士ゼロックス株式会社の商品名です。

Versaplot 9, XIDS, EDIA, ED- , 8000INS, Argoss (商標登録出願中), XS-36, XS-12, XS-50F, FX4109, FX4018, FX4024

GlobalView は米国 Xerox Corporation の登録商標です。

HPGL はヒューレット・パッカード社の商標です。

CalComp は米国 CalComp 社の商標です。

- 当社は、EDMICS を富士ゼロックス株式会社と販売取引基本契約を締結し販売している。
- 本文中の個有名詞は、一般に各メーカーの商標、登録商標、製品名である。

フォルトトレラント・ネットワーク・プロセッサ
DCP/600 シリーズ

DCP/600 シリーズは、米国ユニシス社のネットワーク体系である DCA を実現する通信制御装置であり、またユニシス社が提唱するユニシス・アーキテクチャの XTPA 思想に基づいて開発したフォルトトレラント機能の通信制御装置を母体とする。シリーズ 2200 ホストにフロント・エンド・プロセッサ(FEP)、ノード・プロセッサ(NP)、リモート・コンセントレータ (RC) として接続可能であり、中・大規模ネットワーク構築用として 3 シリーズ 6 モデルがある。

DCP/610 SP シリーズは冗長性を持たないシングル・プレックス・タイプ、DCP/620 FT シリーズは障害検知機能を強化したホット・スタンバイ・タイプ、DCP/620 FTX シリーズは障害検知機能の強化に加え、セッションの維持およびメッセージの保持機能を有するフォルトトレラント・タイプの 3 シリーズである。各シリーズごとに制御部当り 2 CPU モデルと 4 CPU モデルがあり、合計 6 モデルから構成する。

DCP/600 FT・FTX シリーズはハードウェア、ソフトウェアともに完全二重化構造を採用しサ

ブ・システムとしての高信頼性と耐障害性を追及するとともに、マトリックス・スイッチを接続することにより回線の自動切換え機能を提供する。また、イーサネット*LAN 接続機能に加え新たにトークンリング LAN, FDDI, ISDN (一次群インタフェース) に対応する回線モジュールを提供する。



DCP/600 シリーズ基本キャビネット

1. DCP/600 シリーズの特徴

DCP/600 シリーズは中・大規模ネットワーク構築用として新 DCP ファミリの最上位に位置付け、基本キャビネット内に搭載する制御部の数とそのプロセッサ・モジュール数、使用する回線モジュールの種類により 3 シリーズ 6 モデルに分類する (表 1)。

表 1 DCP/600 シリーズの構成と特徴

総 称		DCP/600 シリーズ		
シリーズ名 (名称)		DCP/610 SP (シングルプレックス DCP)	DCP/620 FT (ホットスタンバイ DCP)	DCP/620 FTX (フォルトトレラント DCP)
モデル名	2 CPU タイプ	DCP/612 SP	DCP/622 FT	DCP/622 FTX
	4 CPU タイプ	DCP/614 SP	DCP/624 FT	DCP/624 FTX
特 徴	制御部数	1	2	2
	CPU 数/制御部	2/4	2/4	2/4
	主に使用する 回線モジュール	SB LM 1)	DB LM 2) DB ILM 3)	DB ILM
	主要機能	シングルプレックス仕様	ホットスタンバイ仕様	フォルトトレラント仕様

1) シングル・バスタイプ回線モジュール 2) デュアル・バスタイプ回線モジュール
3) デュアル・バスタイプ・インテリジェント回線モジュール

* イーサネット (Ethernet) は米国ゼロックス社の登録商標である。

表2 DCP/600 シリーズの主な仕様

シリーズ名	DCP/610 SP シリーズ		DCP/620 FT シリーズ		DCP/620 FTX シリーズ	
モデル名	DCP/612 SP	DCP/614 SP	DCP/622 FT	DCP/624 FT	DCP/622 FTX	DCP/624 FTX
制御部数	1		2		2	
CP 数/制御部	2	4	2	4	2	4
相対性能比	1:1.7		1:1.7		1:1.7	
主記憶容量	16 MB/制御部					
補助記憶容量	80 MB/制御部					
回線収容ポート数	28 (基本) — 228 (最大)					
サポート回線種類	Ethernet LAN, トークン・リング LAN, FDDI, 高速デジタル回線, 専用回線等					
ホスト接続 I/F	Ethernet LAN, FDDI, BMC (最大 16 チャンネル)					
キャビネット数	1 (基本) — 6 (最大)					
キャビネット寸法 (cm) (基本, 拡張共通)	560(幅)×920(奥)×1,770(高)					
TELCON ソフトウェア・レベル	9 R 1-3 C	9 R 3-1 A	9 R 1-3 C	9 R 3-1 A	9 R 3-1 A	9 R 3-1 A
重量 (基本, 拡張共通)	キャビネット当たり 295 Kg (最大)					
国内発表時期	1993 年 2 月 15 日					
国内出荷時期	1993 年 4 月 (DCP/612 SP, DCP/622 FT) 1994 年 5 月 (DCP/614 SP, DCP/624 FT, DCP/622 FTX, DCP/624 FTX)					

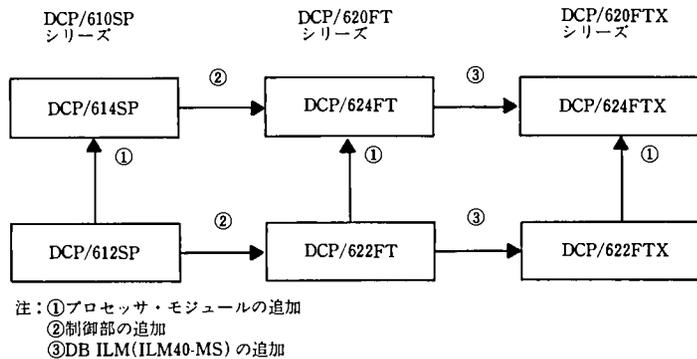


図1 DCP/600 シリーズのアップグレード・パス

また、DCP/600 シリーズの仕様を表2に示す。

- 1) シリーズ上位モデルである DCP/620 FT・FTX シリーズは、回線モジュールを除き完全二重化構造を採用し、通信システムとしての耐障害性と高信頼性を実現している。
- 2) DCP/600 シリーズは TELCON 9 R 1-3 C 以降のレベルで稼働し、イーサネット LAN, トークンリング LAN, FDDI, ISDN (1次群インタフェース) に対応し、LAN 接続はもとより新しい通信インフラに積極的に対応する。
- 3) ユーザの使用環境の変化に合わせて、DCP/600 シリーズの中から最適なモデルを選択できるように、下位モデルに制御部、プロセッ

サ・モジュールもしくは回線モジュール (ILM 40-MS) を追加することによりそれぞれ上位モデルへアップ・グレードができるように設計されている (図1)。

- 4) DCP/600 シリーズは米国ユニシス社の並列処理技術を駆使し、1 制御部当たり 2 CPU, 4 CPU を搭載したものである。近い将来には、システム当たり 16 CPU を搭載した高性能の DCP/618, 628 (8 X シリーズ) の商品化を計画しており、名実共に世界最高性能の通信制御装置を提供する予定である。

1.1 フォルトトレラント機能

DCP/620 FTX シリーズでは、万一通信制御装置に障害が発生した場合でも、障害発生箇所の如

何に拘らずユーザの業務に支障を与えないよう、障害回復を高速かつ自動的に行うことができる。

1.2 高速障害検知機能

DCP/600 シリーズの制御部内のすべてのハードウェア・モジュールを、新規に開発した MCF (Maintenance Control Facility) が 1 秒間に数回の割合で状態監視を実行。また、ソフトウェア障害、回線モジュール、回線状態の監視は常時 DCP/OS が実行しており、これら二つの監視機能により DCP 内のあらゆる障害を瞬時に検知する。

1.3 マトリックス・スイッチの提供

DCP/600 FT・FTX シリーズの基本思想である「高速回復・自動運転」を実現する構成機器であるマトリックス・スイッチは、DCP/600 FT・FTX シリーズ専用の回線切換装置として提供する。従来のハードウェア・インタフェースの他に、ISDN 回線および V. 35 (2.0 MBPS) 等の高速回線用のインタフェースも提供する。

1.4 遠隔制御

DCP/600 シリーズから「リモート・メンテナンス」思想を当社の標準として採用した。「NUL サポート・センタ」には、DCP/600 シリーズのコン

トロール・ステーションから DCP の障害情報等のステータス情報が自動的に伝送され、高度な技術要員により即座に分析・対処できるシステムを採用した。

2. DCP/600 シリーズのハードウェアおよびソフトウェア

シンプレックス仕様の DCP/610 SP シリーズは基本キャビネット内に 1 基の制御部を搭載した冗長性のない通信制御装置でシングルバス回線モジュールを主として使用する。ホットスタンバイ仕様の DCP/620 FT シリーズおよびフォルトトレラント仕様の DCP/620 FTX シリーズは、基本キャビネット内に 2 基の制御部を搭載した完全二重化構造 (回線モジュールを除く) の通信制御装置で、デュアルバス回線モジュールおよびデュアルバス・インテリジェント回線モジュールを使用する (図 2)。

2.1 ハードウェア

DCP/600 シリーズの各モデルはプロセッサ・モジュール、主記憶装置、補助記憶装置、通信プロセッサ、電源制御装置、MCF、コントロール・ス

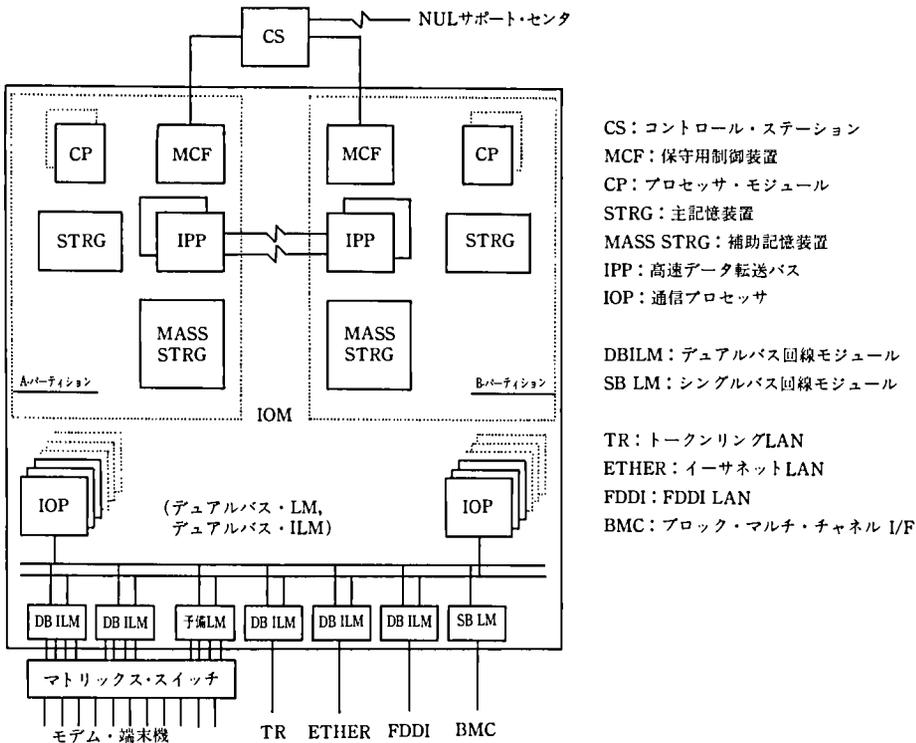


図2 DCP/620FTX シリーズ構成概念図

ーションを共通して用いる。基本キャビネット、拡張キャビネット、IPP、回線ユニット、回線モジュールは各シリーズごとに定めたものを用いる。キャビネット・サイズ、外観、回線収容能力、電源仕様等は各モデルについて同じであり、すべてのハードウェア・モジュールはホットカード挿抜機能（電源を入れた状態で、モジュールの挿抜ができる機能）を有する。回線モジュールは DCP/610 SP シリーズ用、DCP/620 FT シリーズ用および DCP/620 FTX シリーズ用とに大別し、その基本的な使用方法を定めている。

DCP/620 FTX シリーズのハードウェア基本構成は図 2 に示すように回線モジュール部分を除き、完全二重化構造をしている。制御部の制御範囲をパーティションと呼ぶ。A-パーティションおよび B-パーティションには、それぞれプロセッサ・モジュール (CP)、主記憶装置 (STRG)、補助記憶装置 (MASS STRG)、MCF、IPP 等が配置されており両パーティション間は IPP (高速データ転送バス) で接続されている。MCF はパーティション内のすべてのハードウェア・モジュールの障害検知を司り、コントロール・ステーション (CS) に接続して DCP のステータスを表示する。また、すべての通信プロセッサ (IOP)、回線モジュール (デュアルバス LM, デュアルバス ILM) は回線ユニット (IOM) と呼ぶ内部筐体に収容する。デュアルバス回線モジュールは A-パーティションおよび B-パーティション双方の IOP の制御下にあり間接的に両 CP の制御を受ける。回線モジュールの DCE 側は通常マトリックス・スイッチを経由してモデムもしくは端末機に接続する。

DCP/620 FTX シリーズが TELCON O/S のホット・スタンバイ・モードで稼働している場合を例にあげ、その障害回復のメカニズムを簡単に説明する。業務遂行中の A-パーティション(本番

系)に主記憶装置等、パーティション全体の制御に係わる障害が発生した場合、瞬時に MCF がそれを検知し、IPP 経由で B-パーティション(待機系)に通知するとともに CS 上へステータス表示する (リモート・センタにも自動通知)。障害通知を受信した B-パーティションは即座に制御権を得て本番系として稼働し、個々の回線モジュール (ILM 40-MS) が保持している端末セッション情報を交換、仕掛り中の端末メッセージを処理することによりユーザ業務を再開する。また、回線モジュール (ILM 40-MS) に障害 (部分障害) が発生した場合には DCP/OS がこれを検知し、TELCON 経由でマトリックス・スイッチにあらかじめ指定された予備回線モジュールへの切換えコマンドを送信する。予備回線モジュール (実際には端末機) とパーティションとの間で新規にセッション情報の確立を行い、仕掛り中の端末メッセージがある場合には端末からメッセージの再送を受けて、ユーザ業務を再開する。

2.2 ソフトウェア

DCP/600 シリーズの各モデルごとに指定したソフトウェア (DCP/xx TELCON O/S) が必須である。DCP/620 シリーズ用の TELCON O/S には、①ホット・スタンバイ・モード、②ロード・シェア・モード、③ロード・シェア&バック・アップ・モードの三つのモードがあり、ユーザは DCP の利用目的に応じて選択することができる。

TELCON O/S はシリーズ 2200 システム側の通信ソフトウェア「CMS 2200」と対応して稼働する。他に、端末と通信を行う「ターミナル・ハンドラ」が必要であり、異機種間接続を可能とする通信ソフトウェアとして、OSI ソフトウェア、SNA (IBM 社のネットワーク・アーキテクチャ) 接続用ソフトウェア、TCP/IP 接続用ソフトウェア等を準備している。

CGは変化と多様性に富んだ分野であるが、高価なディスプレイを必要とし、コンピュータ資源を消費し、一部の専門家しかわからない分野であった。しかし、この数年のハードウェアの価格性能比の向上によりCGはマンマシン・インタラクションの促進に役立っている。大平昌明は、最新ビジュアライゼーション技術の探求とCAD-CEUS/CGの中で、照明とシェーディング手法について説明するとともに、CADCEUS/CGの紹介を行っている。

プラスチックの射出成形や軽合金のダイカスト鋳造における金型への充填解析に続き、成形品の効率的かつ均一な冷却を目指し金型冷却解析システムを試作した。金型の設計段階では冷却管路の配置が検討されるが、管路の変更による金型内部の要素再分割を避けるため離散化手法として境界要素法を採用した。菅沼正之・平林繁の境界要素法による金型冷却解析システムは、境界要素法の基本概念、金型冷却解析への適用、および解析例について詳述している。

住宅CADシステムHCADで出力した線図図面は主に確認申請や施工用に使用されるが、近年この図面を顧客に対するプレゼンテーションの場で使用したい、というニーズが高まってきた。前原要は、住宅CAD図面の色づけに関する一手法の中で、プレゼンテーションに不可欠な、新開発の線図図面に対し256色で彩色を施す色づけ機能(彩色機能)の解説を行っている。

自動車の車体部品用の鋼板プレス金型の設計・製作に要する費用の削減、期間短縮のためには、プレス成形性を保証した良質の設計作業が不可欠である。沢田晃二・井上博順・前田幸久は、プレス金型の形状設計CADの中で、日産自動車(株)にて開発・業務適用されている「プレス金型形状設計用CADシステム」を事例として、いわゆるダイフェースCADシステムとして満たすべき機能要件と、それに沿った当該システムの特徴を解説している。

▶ 技報編集委員会

委員長 柳生孝昭
副委員長 小林 允, 米口 肇
委員 朝倉文敏, 岩佐宏一, 岩澤慶次
村岡俊彦, 木村修三, 久保田俊雄
佐藤 博, 佐藤政俊, 高畑和夫
内藤 聰, 永田利地, 馬場正存
深堀年弘, 古谷雄一, 青柳幸久
渡辺 寛, 古村哲也

▶ 編集制作担当

研究開発部 駒崎洋介, 丹野敬子
業務本部 熊谷 貴

● Editorial Board

T. Yagi (Chairman)
M. Kobayashi (Vice Chairman)
H. Yoneguchi (Vice Chairman)
F. Asakura, K. Iwasa, K. Iwasawa
T. Muraoka, S. Kimura, T. Kubota
H. Sato, M. Sato, K. Takahata
S. Naito, T. Nagata, M. Baba
T. Fukabori, Y. Furuya, Y. Aoyagi
H. Watanabe, T. Komura

● Editorial Staff

Y. Komazaki, K. Tanno
(Research and Development)
T. Kumagai
(Corporate Planning)

ISSN 0914-9996

技 報

UNISYS TECHNOLOGY REVIEW

Vol. 13 No. 1 (No. 37)

発行日 平成5年5月31日
編集人 柳生孝昭
発行人 富田和夫
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. 1993

UNISYS

いま使っているNECのパソコンのWindows™に

IBMのホストからデータを読み込んで

Excelで表計算して、

営業課の山本君のSUNのワークステーションに

電子メールで送りたいんだけど。



…ムリよね。

無理な注文、実現します。ユニシスのオープン・システム・テクノロジー。

例えばこの方のようなご要望。どなたも、その実現の難しさがおわかりになるはず。でも、専門知識がなくても、さまざまなメーカーの優れたものを自由に、しかも快適に使えてこそ、ほんとうのオープン・システム。日本ユニシスはつきつぎと「ムリ」を「ホント」に変えています。上の例は、ユニシスの「U6000シリーズ」をサーバに、各社のプロダクトを組み合わせたクライアント/サーバ・システム。その上、電子メールに画像データを同封して送るということも可能なのです。それができるのも、柔軟な発想でメーカーの枠を超え、システムをインテグレーションできるユニシスのオープン・システム・テクノロジーがあればこそ。今まで蓄積してきたいろいろなノウハウや豊富な知識。それらをシステムの中で適材適所で生かしてゆくのが私たちのオープン・システム。「ムリかな？」と思ったときこそユニシスにご相談ください。

人に、オープン。

UNIX & UNISYS

●Windowsは米国マイクロソフト社の商標です。●NECは日本電気株式会社。IBMは日本アイ・ビー・エム株式会社。SUNは日本サン・マイクロシステムズ株式会社。●UNIXオペレーティングシステムはUNIX System Laboratories, Inc.が開発し、ライセンスしています。

日本ユニシス株式会社 本社 東京都江東区豊洲1-1-1 〒135 電話03-5546-4111(大代表)