

CAD における知識の利用形態

Mode of Use of Design Knowledge in CAD Systems

宮 地 恵 美

要 約 近年，設計ルールなどの設計知識を利用した CAD 機能の発表が相次いでいる．設計知識とその利用技術には様々なものがあるため，本稿では CAD システムが扱う幾何形状に着目して，形状データを利用するときの前提条件の違いから設計問題を分類する．そして，分類した設計問題の特徴と現状の課題から，問題解決に必要な知識と機能を抽出し，それらの機能を CAD システムで実現する上での要件を明らかにする．試行錯誤を伴う設計作業をインタラクティブに支援する知識の利用機能を実現することが，高品質で効率のよい設計作業を支援する CAD システムには不可欠であることを主張する．

Abstract Some CAD capabilities using design knowledge such as design rules have been launched one after another in these years, and however, the various definitions of design knowledge exist and the design knowledge technologies are wide ranging. In order to clarify the requirements of implementation of CAD capabilities to manage design knowledge, at first this paper classifies the design problems according to the precondition for using geometric data, and then abstracts knowledge and capabilities required for problem solving from classified problems. And next, this paper makes a point that CAD capabilities to support a trial and error design process in conversational mode by using knowledge is indispensable for high-quality and efficiency of design tasks.

1. はじめに

ここ数年，知識ベース CAD システムという言葉に代表される知識を標榜した CAD システムが，いくつか発表され話題となっているが^[1]，CAD システムにおける知識と一口に言っても，その内容と利用方法は千差万別である．例えば，商用のハイエンド 3 次元 CAD システムにおいては，CATIA V5^[1] が，知識利用の一例として，「穴の半径は 10 mm 以上 12 mm 以下でなければならない」という設計制約条件をシステムに設定することで，モデリング時に穴の半径として設定範囲外の値以外は受けつけない，あるいは警告を発するといったことを実現できる機能を紹介し，UG V17^[2] ではエンジニアリングルール定義向きのオブジェクト指向プログラミング言語 UG-KnowledgeFusion^[3] を使って再利用可能な知識ベースを構築できると主張している．

実用システムでは，日本電気(株)の「通信装置設計の知識駆動設計システム」^[2]が報告されている．研究システムでは，部品に共通する知識，例えば「はめ合い」「組み立て」「熱処理」等々をデータベース化して設計に利用しようとしている例がある^{[3][4]}．設計時の発想を支援するシステムとしては，創造的問題解決の方法論 TRIZ^[4] をベースにした，米 Invention Machine 社の技術知識の管理を支援するシステム TechOptimizer が設計者の注目を集めている．

これまでも，多くの CAD ユーザが，それぞれの CAD で利用できるカスタマイズ機能，マクロ言語，パラメトリック機能^[5] などを利用して設計ノウハウの蓄積を行な

ってきた。弊社のCAD/CAMシステムユーザも、UNICAD^{*6}のUDL言語の利用に始まり、CADCEUS^{*7}のスクリプト言語、パラメトリック標準化ライブラリ機能、最適化機能の利用に至る十数年間にわたり設計知識を獲得し利用する工夫を続けてきている。このようなCADシステムの歴史の中で、ここに至り、主要CADベンダが相次いで知識を利用した新たな機能を市場に投入する背景には、次の二つの理由が存在する。

一つめは、フィーチャーパラメトリック設計手法^{*8}が実務に浸透してきて、パラメタによる形状の生成・変更の機能と検査ルールといった知識を結びつけることが、ユーザにとって必然であること。二つめは、これまでの設計のノウハウを生かす様々な取り組みにもかかわらず、いまだに設計者の経験や勘に頼る作業が多く残っているという状況を打破するために、新たな知識獲得や利用技術への期待が高まっていることである。

知識の利用技術には様々なものがあるため、本稿では、現状で何ができて、何ができないのかを明らかにした上で知識を扱うためにCADシステムで必要となる機能要件を列挙する。そのために、まず第2章で、設計問題を分類した上で、各問題の現状と課題について述べる。第3章では、分類した各設計問題の解決に利用している機能と情報を整理して、CADで知識を扱うために必要な機能を抽出する。第4章では、抽出した知識を扱う機能の現状と課題についてまとめた上で、これらの機能をCADシステムで実現する方法を提案する。また実現にあたって解決しなければならない問題について考察する。第5章では、試行錯誤を伴う設計作業を、知識を利用してインタラクティブに支援するCAD機能を実現することが、高品質で効率のよい設計作業には重要であることを主張し、今後の課題について述べる。

2. 形状に関する設計問題の分類

本稿では、CADにおいて重要なデータである幾何形状に注目して、問題に対する幾何形状の状態（幾何形状の前提条件）の違いによって設計問題を分類する(図1)。これは、幾何形状の前提条件が、問題を解決するための知識の内容と利用方法に密接に関係すると考えるからである。

まず、対象とする幾何形状がCADに存在する場合と存在しない場合で分類する。存在する場合を形状検査問題とする。存在しない場合を形状生成問題とする。

形状生成問題は、過去に対象形状と似たものが存在するか否かで分類する。過去に似たような形状が全く存在せず、新たな形状を見つけなければならない問題を発見問題と呼ぶ。一方、過去に似たような形状が存在しているということは、多くの場合、なんらかの幾何生成の手続きやルールやパラメタが存在するものと考えられるが、全く存在しない場合もあり得る。この場合、実物の製品が存在すれば、いわゆる一般的なリバースエンジニアリングの問題とみなすことができる。

幾何生成の手続きやルールやパラメタが存在する場合、これらから目的とする形状を一意に決定できる場合を自動生成問題と呼ぶ。目的形状を一意に決定できない場合には、目的形状の評価方法が存在する場合としない場合で分類する。目的形状の評価方法が何も存在しない場合は、発見的な問題と同種の問題であるとみなす。評価方法

が決まっている場合は、その評価方法が定式化できない場合を目標具現化問題と呼び、定式化できる場合を最適化問題と呼ぶ。

形状検査問題をさらに分類していくことはしない。なぜならば、例えば与えられた形状を検査して検査内容にしたがって形状を変形するような問題は、形状検査問題と自動生成問題の組み合わせ、あるいは形状検査問題と目標具現化問題の組み合わせであると考えられるからである。

形状生成問題の前提条件として実物のみが存在するリバースエンジニアリング問題は、測定方法の問題、測定結果から CAD データへ変換する問題、あるいは測定結果の視覚化や設計へのフィードバック問題に分類できる。これらの問題は、本論文では知識を利用して CAD で解決すべき形状に関する問題の範囲外とする。ただし、測定結果は他の問題において知識として利用可能データである。また測定結果が CAD データに変換されれば、後は形状検査問題と同様に扱える。

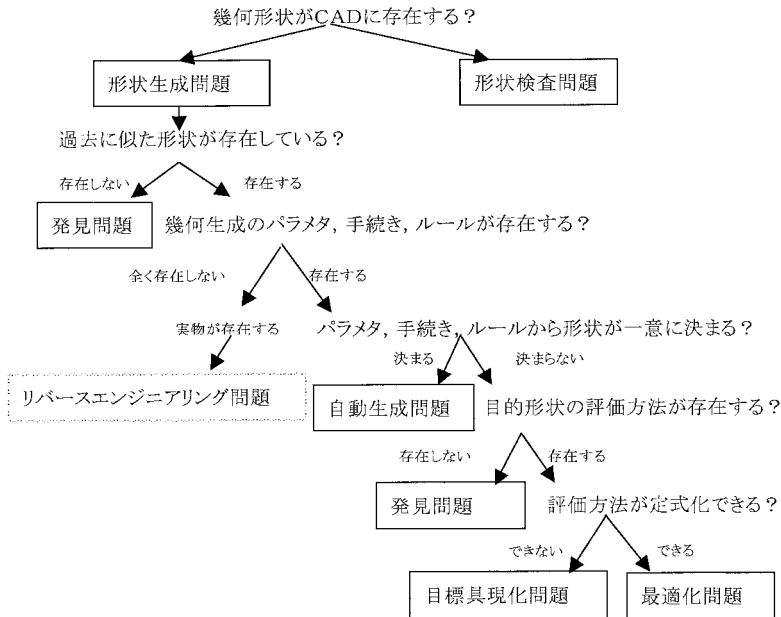


図 1 CAD の形状に関する設計問題の分類

このように問題解決の前提条件から分類した木の葉（図 1）にあたる次の五つの問題を、CAD で扱う形状に関する設計問題であると定義する*9。

- 1) 形状検査問題
- 2) 発見問題
- 3) 自動生成問題
- 4) 目標具現化問題
- 5) 最適化問題

次節から、これらの問題の特徴と、現在これらの問題で利用されている知識とその利用形態および課題について述べる。

2.1 形状検査問題

形状検査問題は、「形状の認識問題」と「認識形状に対する問題」という二つの問題に分解できて、形状認識処理が前処理となり、認識形状に対する問題の処理が後処理となる。形状検査問題を処理するプログラムは、例えば、与えられた形状の検査をして、検査結果のレポートを作成する検査表作成アプリケーションや、ある条件を満たす形状の属性を変更する自動修正アプリケーションなどのように後半の処理内容によって異なるものとなる。また後処理で利用される知識の内容も、例えば、検査項目として法律や社内規格を利用するなど様々なものがある。このように形状検査問題の後半処理はアプリケーションごとに異なるが、前半の形状認識処理は検査問題を解くために共通の処理である。

一般に、形状を認識して後処理で必要とする形状の部分を判断することは困難である。そのため、多くの実用システムでは、形状の入力指示や、あらかじめ形状に部分を識別するための情報を付与しておくなどの方法で対象形状の部分を特定している。

一方、与えられた幾何形状から穴、フィレットといった特徴形状を抽出するフィーチャー認識ソフトウェアが最近登場してきており、ミッドレンジ CAD システムの SolidEdge をはじめ、いくつかの CAD システムがこのフィーチャー認識ソフトウェアを採用している。しかしながら、現状のフィーチャー認識ソフトウェアは、穴、溝などの比較的簡単な形状の特徴認識に適用が限られている。複雑な立体の構造を認識するようなシステム^[6]や設計者の意図する形状を認識するシステムは研究レベルにとどまり、実用システムとして普及しているわけではない。

今後、測定技術の進歩や大量点群処理ソフトウェア出現により、リバースエンジニアリングが身近になること、異なる CAD 間におけるフィーチャーモデルデータの流通の機会が増えていくこと、設計作業の中で手軽に FEM 解析などの解析システムを利用する要求が高まることが予想される。このため幾何形状特徴の自動認識技術は、検査問題に必要であるだけでなく、設計作業を効率化するためのさまざまなアプリケーションを構築する上でも重要な役割を果たすようになる。

2.2 発見問題

与えられた課題を解決するような新たな形状を見つけ出す問題を発見問題と呼ぶ。上流工程の構想設計や新しい機構を決定するような問題である。このような問題では、過去に同じような形状が存在しないので、人間が解決方法を見つけなければならない。したがって、CAD の知識は人間の思考や発想を支援するものでなければならない。具体的には、過去の設計事例、関連事例、特許など多種多様な情報が、人間の思考の助けになるような形で提供されなければならない。

情報提供の技術に対する関心は、設計分野に限らず、ここ数年高まってきていて、データ検索技術と発想方法の手法とを組み合わせた商品や研究が発表されている。設計上流工程を支援するシステムにおいては、このような技術が、今後重要な役割を果たすものと考えられている。

発見問題で利用される情報、すなわち知識は、データの種類が多種多様であるという点においては、検査問題の後半処理で必要とする知識と同様であるが、検査問題後半で扱うデータの所在が比較的局所的で、利用すべき知識を明示的に指示できるのに

対して、自由な発想を助けるためのデータはデータ形式の種類がより多く、その存在範囲が広範囲で、利用したい情報を明示的に指示できないという性質を持つ点が異なる。このため発見問題では、膨大なデータの中から必要なデータを絞り込む技術が必要になる。データ検索機能の商品の一例としては、特許などの技術文書から、“問題解決策”の形式で自動的に知識を抽出する意味解析ソフトウェア Knowledgeist が米 Invention Machine 社から発売されている。このようなソフトウェアをどのように CAD システムと連動させていくかは今後の課題である。

2.3 自動生成問題

目標形状を作り出す手順が明確に決まってい、手順に対するパラメタを指定することで、目標形状を自動的に生成できる問題を自動生成問題と呼ぶ。現在、自動生成問題の解決方法として、次の二つの方法が一般的によく利用されている。

- 1) マクロ言語等でプログラムを記述する方法
- 2) パラメトリックな設計手順を再利用する方法

1) のプログラムを記述する方法は、古くから利用されてきた。プログラム記述言語は大きく三つのグループに分類できる。1 番目は CAD 専用のマクロ言語で、その機能と文法は CAD に依存して他 CAD システムと互換性がない。2 番目は、FORTRAN, C, LISP などの汎用プログラミング言語で、CAD のサブルーチン、関数などのアプリケーションインタフェースを利用する。各 CAD のアプリケーションインタフェースは異なるため、異なる CAD 間でのプログラムの互換性はない。しかしながら CAD の機能に依存しない、例えば文書ファイルからの情報検索プログラムなどは、異なる CAD 間でも利用可能である。3 番目は Intent! のような CAD とは独立の専用言語システムである。CAD の機能は抽象化した専用言語の機能として扱われる。例えば、立方体を生成するとき、汎用言語の場合、ある CAD システムでは立方体定義、他の CAD では平行掃引体定義をするユーザインタフェースを利用することになるが、Intent! では、BOX クラスを実体化するプログラムになり。特定の CAD に依存しない形状定義プログラムの記述が可能である。すなわち抽象化したクラスのレパートリが同じであれば、異なる CAD 間で Intent! プログラムは互換性を保つ。

どの三つの言語グループでも、プログラムの作成、保守作業が必要である。知識はプログラムの中に埋めこまれていて可視性に乏しく、再利用が困難である。

2) の方法は、機械設計、金型構造部設計などの分野で、過去のデータと似たようなものを設計する場合（流用設計）に有効であるため、近年、盛んに利用されるようになってきた。この方法は言語によるプログラミングと比較すると、プログラムを書かなくて済むという点では優れているが、知識が設計手順の中に埋没して可視性に乏しいという点ではプログラムと変わらない。また形状変形によって位相構造が保たれないいかなる場合にも、破綻なく形状を構築できる堅牢な CAD システムは存在しないため、あらかじめ再利用を考慮したモデリング手順を構築することが要求され、すべての設計問題に 2) の方法を適用することはできない。

1) と 2) の方法を組み合わせた方法も考えられる。あらかじめ登録されたパラメタ値の組を検索するプログラムとパラメトリック形状として定義された標準部品を組み合わせるような例である。

2.4 目標具現化問題

目標具現化問題とは、目的とする形状をCAD上で生成する問題である。形状を生成する手続きやルールが存在し、目標形状の評価方法も決まっているが、形状を生成する手順の入力パラメタ値が最初から判っていないことと、評価方法の定式化が困難なため、自動的に形状を生成できない問題である。例えば、成形後の形状が製品形状になるような型を作るという金型設計問題が典型的な問題である。具体的な例としては、鉄板の流入量を制御するために突起や溝形状（ビード形状）をつける、鉄板のもどり（スプリングバック発生）を見込んで形状をあらかじめ変形させておくといったプレス金型設計の問題があげられる。プレス成形後の製品形状を求められている形状にするためのビードやスプリングバックを見こんだ金型形状は正確にはわからない。そこで経験値を利用して型設計を行なう。型設計後はシミュレーションや、試作をして測定をすることで製品の評価を行ない、型の不具合箇所を修正するという作業を、繰り返しおこなっている（図2）。現場では、評価と変更の繰り返し回数を減らすための努力として、次のような二つの方法がとられていることが多い。

一つめの方法は、設計者が経験的に覚えている数値、経験から得られた数式で計算した数値、設計基準書などに記載されている数値を、形状生成のための入力パラメタとして利用する方法、

二つめの方法は、設計者が、過去の「金型設計形状」と「成形後の形状」と「解析評価結果」というデータの組を比較検討し、過去の例から類推して、よりよい型形状を設計するという方法である。どちらの方法も、現状では、知識として利用する情報は紙の資料や設計者の記憶であることが多く、CADシステムと連動して利用するまでには至っていない場合が多い。

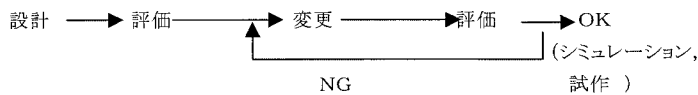


図2 評価、変更のループ

これらの情報をCADと連動して利用するためには、過去のデータを再利用可能な形式で蓄積する機能と、蓄積したデータから形状生成のパラメタを導き出す手法が必要である。過去の有効なデータを数多く蓄えることによって、目的形状の評価方法が定式化でき、その結果として目的形状を生成するパラメタを自動的に導き出す方法が見つかる場合もあるはずである。そうすると、この問題は自動生成問題と同じになる。また、評価、変更ループの中で、試作ではなく、成形シミュレーションプログラムを使って、コストを削減しようとする試みも近年盛んになってきている。評価方法が定式化できれば、成形シミュレーションプログラムを利用することで、目標具現化問題を最適化問題として扱うことが可能になる。

2.5 最適化問題

最適化問題は、ある条件のもとに、パラメトリックに変形できる形状を複数個配置するとき、評価関数の評価結果の良い形状と配置位置を求める問題である。解は通常

一つではない。本稿では、配置形状が2個以上の場合を配置最適化問題と呼び、配置形状が一つの一般の最適化問題とは区別する。例えば、図3の部品の質量を変えずにドアミラーのブラケット部分の振動数を大きくする問題は一般の最適化問題とし、図4の指示カットラインとスクラップカットラインに対して最適な鋼材の形状と配置位置を決定する問題を最適化配置問題とする。

現在、いくつかの商用最適化エンジンが利用でき、対象とする形状の定義が一意に決まり、評価式や条件式が明確に定義できる問題を解くことができる。例えばCAD-CEUSの最適化機能は、米Integral Solution Corporation社の最適化エンジンEMS Design Optimizerを利用して、図3のような問題を解くことができる。この例の場合、対象モデルは、断面形状の断面積を一定にする幅、高さ、板厚の拘束条件から簡単に定義できるドアミラーの立体で、評価関数は一次モーメントを最大にすることである。

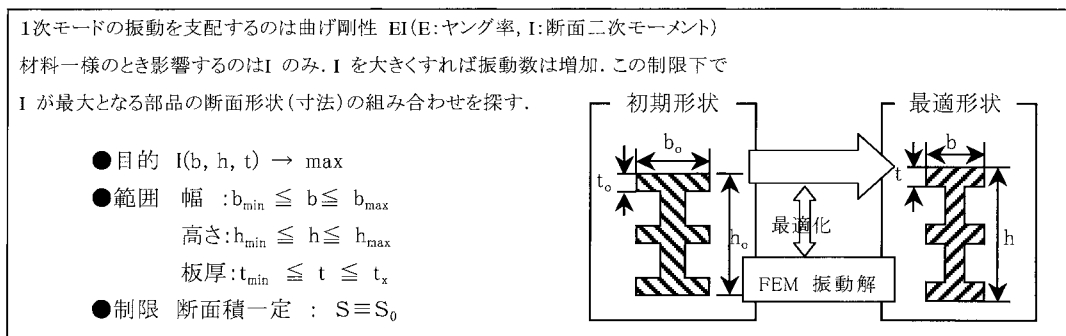


図3 CADCEUSの最適化機能適用例

図3のような拘束条件、評価関数が明確に定義できる問題に対しては、汎用的な最適化エンジンを利用することができるが、図4の鋼材配置問題のような、配置条件や手順は存在するが、必ず一意に決まるものではなく、適用が必須でないルールが存在して、配置部品個数が変化して、かつ部品間の位置関係のルールも存在するような問題に対して、汎用的な最適化エンジンを利用した実用システムの例はないといっている。

3. 知識を利用するために必要な機能

前章で分類した五つのCADの形状に関する問題において、問題解決に利用している機能と情報を整理すると表1のようになる。各問題で利用している知識に印がついている。

表1に挙げた機能を、参照する情報の種類でさらに分類してみる。形状とルール(手順)を参照する三つの機能、「形状と検査ルール対応付け」、「形状の参照」、「形状の変更、削除、創成」は一つのまとまりにして「ルールによる形状定義」機能と呼ぶことにする。

「検査問題」、「自動生成問題」などで利用する情報(テキスト、数値)を、「検索、

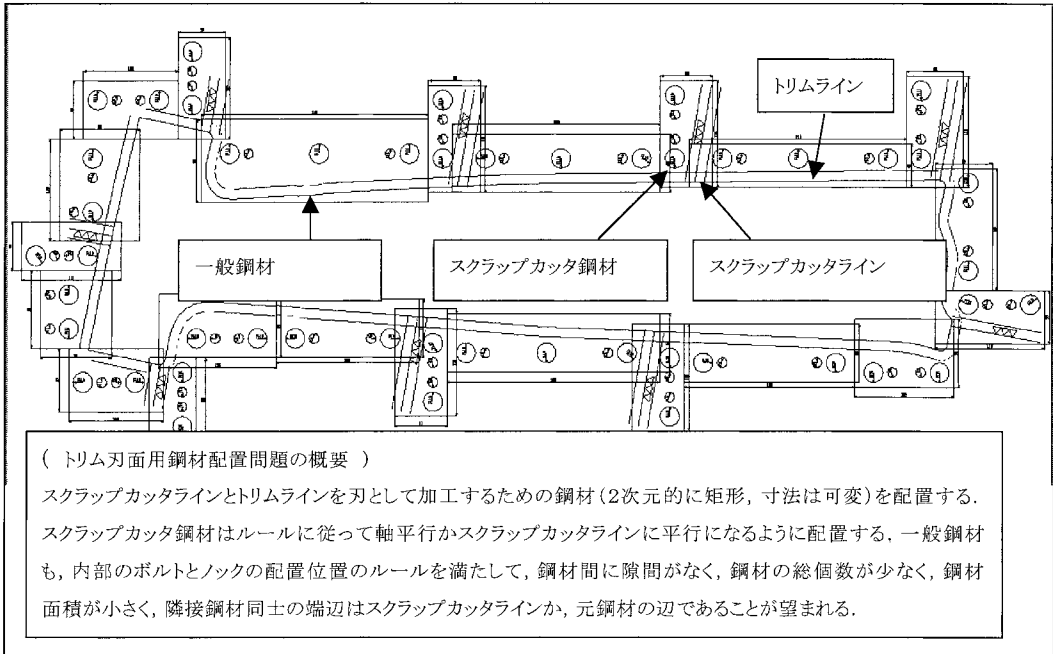


図 4 トリム刃面用鋼材配置の例

表 1 機能とデータによる知識の分類

機能	知識 情報	問題				
		検査	発見	自動生成	目標具現化	最適化
形状特徴認識	形状, 検査ルール	○				
形状とルール対応付	形状, 検査ルール(手順)	○		○	○	○
形状の参照	形状, 検査ルール(手順)	○		○	○	○
形状の変更, 削除, 創成	形状, 変更, 削除, 創成ルール(手順)	○ ^(注1)		○	○	○
情報検索, 提供	テキスト, 数値 ^(注2)		○	○		○
	画像, CADデータなど ^(注3)		○		○	
結果の蓄積, 提供	パラメタ値, CADデータ, 解析結果(数値, 画像 etc) 測定結果(数値, 画像 etc)		○		○	
最適化推論	形状生成, 拘束, 配置のルール, 評価式					○

注1 検査問題とはいっても、例えば半径5の穴径を一斉に変更するような処理も検査問題とする。

注2 検査, 自動生成, 最適化問題で扱う情報は、システムで利用できる形式もの、数値、テキストなどに限る。

注3 発見問題, 目標具現化問題など扱う情報は、テキスト、数値に限らず、画像、CADデータなどあらゆる種類のものを含み、設計者が情報提供対象となる。

提供」する機能は、検索結果を、「ルールによる形状定義」機能へのパラメタとして渡す。一方「発見問題」で利用する情報(テキスト、数値、画像、CADデータなど)を「検索、提供」する機能は、情報を設計者に提示する。前者の機能は、情報をシステムに渡すという役割を果たすものなので「パラメタの検索」機能と呼ぶことにする。後者は設計者に情報を提示する「目標具現化問題」の「結果の蓄積、提供」機能と似

であり、他の「形状特徴認識」や「最適化推論」、「情報検索提供」、「結果蓄積」などの機能は、すべてこの「ルールによる形状定義機能」の上に構築されるアプリケーションだと見なせる。

表 2 知識を扱う機能の現状と課題

機能	現状と課題
形状とルールの対応付	要素のID, 外部名を利用したマクロプログラムが利用されることが多い。要素IDや外部名は、ユーザが意図する形状と対応しないという問題がある。スケッチ拘束寸法と寸法表 (EXCEL表など)との対応づけ機能を持つCADも多い。 ルールから形状への対応付け機能は存在するが、形状の変形がルールに反映される機能をもつシステムは少ない。
パラメタ(ルール)検索	プログラムに埋めこまれる。あるいはマクロ言語の外部ファイルI/O機能で実現されていることが多い。
形状の参照・変更・削除・創成	マクロ言語などで記述されることが多い。ルールを利用した形状のインタラクティブな変形に追従するような参照・評価は実現できていない。 3次元ハイエンドCADでは履歴再実行機能が利用されているものが多い。履歴再実行機能では位相が変わるような変形の頑健性が低い。 マクロ言語、履歴再実行機能のいずれも、プログラムや履歴の保守が困難。
形状特徴認識	加工フィーチャーの自動認識は実用化されているが、ユーザの意図する形状の認識、複雑なアセンブリ部品の認識は研究段階。
最適化推論	評価式と形状生成のルールが簡単なものは実用化されているが、複雑な配置問題や曲面変形の問題は研究段階。
情報検索提供	システム化が進んでいない。WEB検索エンジンは近年増えているがCADとの結合はまだこれから。人間に対してどれだけ役に立つ形式で情報を提供するかが課題。
結果蓄積提供	システム化されていないことが多い。運用に頼っている。設計意図を蓄積利用するためには重要な機能。

この五つの機能、①ルールによる形状定義機能、②形状特徴認識機能、③最適化推論機能、④情報検索、提供機能、⑤結果収集蓄積機能を、CADシステムの機能としたときの機能要件と実現方法は以下のとおりである。

①ルールによる形状定義機能は、利用するルールの形式が多様多様であることから、例えば表形式といった特定の形式で形状定義を表現することもできるし、表現能力が高く、過去の実績も多い専用プログラミング言語を利用することも考えられる。いずれにしても、条件に応じてCAD形状とルールの対応付けができ、ルールによる形状の参照、創成、変更、削除ができる機能が前提となる。このような機能を、汎用プログラミング言語から利用可能なCADのユーザアプリケーションインタフェース(ユーザAPI)にしておくと、CAD専用のプログラミング言語システムも、表形式でルールを扱うようなユーザ専用機能もシステムを実現することが容易である。

②の形状特徴認識機能と③の最適化推論機能は、CAD本体に、ある特定の機能を埋め込むことはやめて、遺伝的アルゴリズム、ニューラルネットワークなどの問題に適合するプログラムを、ユーザAPIを使ってCADと連動させるようにすべきである。

④の情報検索、提供機能については、当面CADデータとは密接に関係しない機能であるため、WEB情報検索機能などのプログラムを利用したほうが良い。必要があれば、検索結果や提供されたデータとCADデータとの対応づけを、ユ

ーザ API を介して行なう。

⑤の結果収集蓄積機能は、目標具現化問題を解決するために必要であると同時に、設計意図を蓄積し、知識として再利用するためにも重要な機能である。この機能はユーザのインタラクティブなCAD操作と関わるため、CADの入力操作から情報を収集して、収集した情報を知識として再利用できるようにCAD形状と関連づけて出力できなければならない。

いずれの機能要件においても、ルールと形状を結びつける機能を実現するためのユーザ API が、重要な役割を担うことになる。以降、CADのユーザ API を利用して各機能を実現する場合に、解決しなければならない問題について考察する。

4.1 形状とルールを対応づけるための機能

設計意図やルールと幾何形状を対応付ける機能がCADのユーザ API には必要である。現行のマクロプログラムや、ユーザ API を利用したプログラムでは、幾何形状はCADの要素 ID や外部名で扱われることが一般的で、この要素 ID を、何の条件もなく、設計意図やルールと対応づけることは困難である。例えば、「部屋の壁の面積は100以下である」というルールで図6(a)の部屋を検査しようとしたとき、「壁」に対応する側面は4面ある。面要素の ID を使ったプログラムで、直方体6面の中の、どの4面を選択してよいのかは、何か条件がなければ判断がつかない。また「壁」という意味のある形状の幾何要素の個数は常に一定とは限らない。たとえば、図6(b)のように、側面の平面の個数が6面になるようなこともある。それでも、壁に関するルールをこの6面の「壁」に対して適用したい。すなわち、幾何要素の ID ではなく「壁」という意味のある名前で形状を指示したい場合がある。そのためには、意味のある幾何要素のまとまりごとの名前付けと参照ができて、集合演算等の操作で幾何形状が変更されても、その名前が引き続き利用可能である機能が必要となる。



図 6 部屋の壁の面はどれ？

4.2 CADのインタラクティブな操作で知識を適用する機能

ルールやアルゴリズムによって完全に最終形状が決まらない目標具現化問題、あるいは最適化配置問題においては、設計者は、形状を変形して、その変更結果が条件に合うかどうか、設計のバランスがとれているかどうか等の評価を繰り返し行なう。既存のCADシステムでは、このような繰り返し作業に非常に手間がかかっている。

例えば、図7の鋼材配置では、鋼材をちょっとずらしたり、大きさを変更したりするたびに、鋼材内部のボルトやナットを、ルールを満たすように作り直さなければならない。あるいは、隣接する鋼材同士の関係がルールを満たすように、干渉部分のモデリングをしておさなければならない。

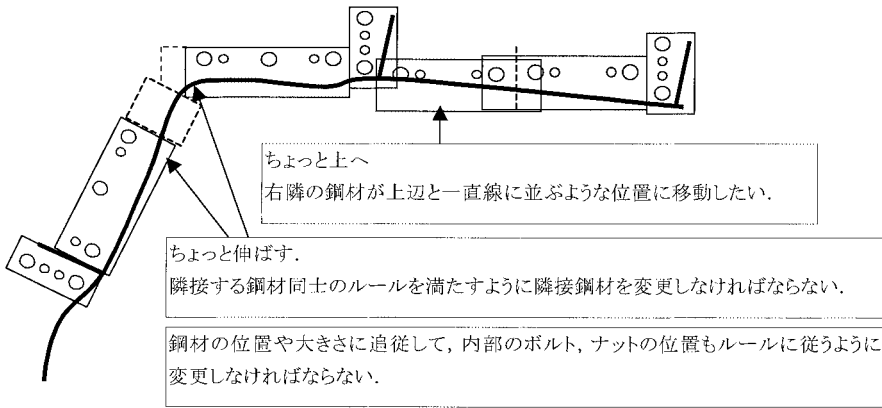


図 7 トリム刃鋼材の配置問題

図 8 のエンジンのバルブレイアウトは、与えられたボア内に、エンジンの要求特性を満たす最適の径のバルブをバランスよく配置する問題であるが、例えば、エンジンの出力を上げたいとき、バルブ径を大きくする、あるいは燃焼室を小さくする (= バルブ角度を小さくする。) といった目標を達成するためのいくつかの方法が考えられる。バルブ径、バルブ角度、ポート形状、燃焼室容積、全体寸法といった各項目は「こちらをたてれば、あちらがたたず」という関係にあるため、総合的な判断が必要である。設計者は、最適解を見つけるために、できるだけ多くの可能性を検討したいが、設計変数を幾何形状の拘束条件にそのまま使用すると拘束条件が過剰^{*10} になり、パラメトリックな形状変更機能が利用できないため、レイアウト図を作り直した上で、クリアランスのチェックや性能を評価しなければならない。

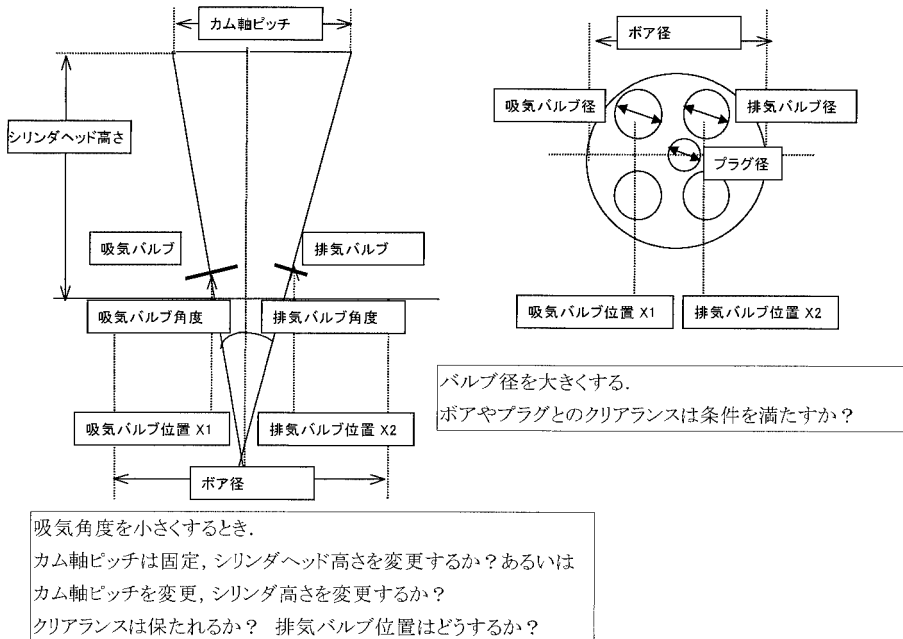


図 8 バルブレイアウトの構想設計問題

このような形状変更と評価の労力を解消するために、CADのインタラクティブな形状変更操作では、操作に追従して知識を適用することで、変形状の評価と、他の形状との関係の整合性維持が自動的に行なわれる必要がある。また、バルブレイアウトのような設計変数を使用した形状変更問題では、すべての設計変数を無条件に形状拘束変数として利用するのではなく、設計者が、設計意図と設計の状況に合わせて、「拘束する変数」と「なりゆきで変更される変数」の組み合わせや、拘束の優先順位を、柔軟かつ容易に選択できる機能が必要である。

4.3 設計意図を保存する機能

通常、設計後の形状には設計意図が明示的に保存されない。例えば、図9のような製品断面形状のプラスチック射出成形金型のパーティング面（型を分割する面）を設計する場合を考えてみる。ここでは、一般的なパーティング面決定に関する次の二つのルールが存在するとする。

①負角にならない場所を選ぶ。②バリのめだたない場所を選ぶ。

図9の例で、この二つのパーティング面決定ルールを同時に満足することはできない。なぜならば①のルールに従ってAの位置をパーティング面を選択すると、バリが製品面の目立つ場所にできてしまい。②のルールに従ってBの位置を選択すると俯角になるので、スライドピン等をつけなければ成形ができない。何らかの判断基準にしたがって、AまたはBの位置を選択しなければならない。例えば、この製品が高級なもので、目立つ個所のバリは許されないという判断であれば、コストは高くてもBの位置に決めるし、何よりもコストが優先されるのであれば、余分な機構のいらぬAを選択する。このように一般的なルールを適用するだけでなく、様々な設計ルールと与えられた条件との兼ね合いを考慮して形状を決定しなければならないことがある。

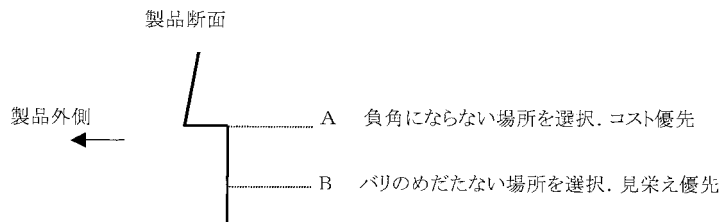


図9 どこがパーティング面として良いのか

このような形状を決定した時の設計者の判断は設計意図として蓄積して、必要に応じて利用できるようにすべきである。設計意図を付与する形状のまとまりと情報を採取するタイミングは、設計者の作業を妨げないように決めなければならない。形状のまとまりと情報収集のタイミングは、システムであらかじめ決めるのではなく、アプリケーションごとにユーザが設定できるようにしておくべきである。

5. おわりに

最近、登場してきているウィザードと呼ばれる業務に特化した知識利用アプリケー

シヨンの多くは、設計手順を標準化し、知識を利用して設計者の判断をできるだけ減らそうとするシステムである。人間の判断を必要としない、試行錯誤のない設計分野では、このようなシステムが設計作業の大幅な効率化に役立つことは異論のないところであろう。

しかしながらウィザードだけは、本当に良いものを作ることができないと思われる。特に当社のCAD/CAMユーザの約8割を占める金型設計分野で、複雑な精度のよい金型を効率よく作るとは困難である。なぜならば、第4章で述べたようなプラスチック成形用金型のパーティング面を決める簡単な例でさえ、良いものを作ろうとすると、ルールを一律に適用することはできず、設計者の判断が必要となるからである。問題が簡単なものは、推論機能などの利用が可能ではあるが、単純な評価関数だけですべてを決定できない実際の設計では、決められたルールだけでは解決できない場面が数多くあり、インタラクティブなCADの形状創成、変更操作に伴う、設計者の評価と判断作業が必要不可欠である。知識は人手がかかり、かつ品質に影響を及ぼすような試行錯誤的なCAD操作を支援するために利用されるべきであると考える。

試行錯誤を伴うCADの操作で、知識を利用するためには、形状変更操作に追従してルールが動的に参照、評価される機能が必要であることを第4章で述べた。この機能を実装する上で解決しなければならない課題の一つに、複数形状間の関係を記述する方法とその実装方法がある。複数種類の複数個の形状間のルールを、すべての形状を含む仮想的な空間内のルールとして記述するのか、個々の形状が相手の形状を指名するようなルールとして記述するのかといったことを検討しなければならない。このとき最も重視すべきことは実行効率である。知識の利用によって操作性を向上させることが目的だからである。

CADで知識を利用するために必要な基本機能は、設計意図やルールを幾何形状と対応付ける機能であり、そのために意味のある幾何形状の部分を表す名前が必要であると述べた。一般にネーミングルールと呼ばれる幾何形状の部分の名前付け方法が、パラメトリック履歴再実行のしくみを頑健にするための重要な課題の一つであり、この課題に関する論文がいくつか発表されている^{[7][8]}。このようなネーミングルールを実用の知識機能の中で、どう利用していくかについては具体的な検討と実験が必要である。

さらに設計意図を蓄積して知識として利用するための機能が必要であると述べたが、その保存の手段、利用方法については、実際の設計作業を分析したうえで検討を進めなければならない。

検討が必要な項目はまだあげられるが、目指すべきは、定形業務に対しては知識を活かした自動設計ウィザードなどの仕掛けが容易に作れることはもちろん、試行錯誤作業においては、設計者の思考を助けて、設計作業が快適に行なえるように知識が利用できるシステムである。

- * 1 仏 Dassult System 社ハイエンド 3 次元 CAD 1999 年出荷最新バージョン .
- * 2 米 Unigraphics Solution (UGS) 社ハイエンド 3 次元 CAD 2000 年出荷最新バージョン
- * 3 米 Heide 社のエンジニアリングアプリケーションを記述するオブジェクト指向言語 Intent ! をもとに UG V 17 に組み込まれた言語 . Intent ! は欧米で AutoCAD 用アプリケーションの実績がある .
- * 4 TRIZ は「発明問題解決の理論 (Theory of Inventive Problem Solving)」のロシア語略称の英文つづり . 1946 年に旧ソ連で着想された . 冷戦終了後に西側諸国の知るところとなり最近では米国の大企業を中心に導入例がある .
- * 5 形状をパラメタで定義する機能 . 例えば , 径をパラメタとしたねじなど標準部品は径の値を与えることで形状を定義することができる .
- * 6 日本ユニシス 3 次元 CAD/CAM システム . CADCEUS の前身システム . 1982 ~
- * 7 日本ユニシスのハイエンド 3 次元 CAD/CAM システム . 現行プロダクト . 1991 ~
- * 8 パラメタで形状を定義するパラメトリックモデリングと穴やボスなどの形状の特徴を持つモデルを利用するモデリングを合わせてフィーチャ パラメトリックモデリングと呼ぶ . 3 次元 CAD システムの多くがこの機能を持つようになってきた .
- * 9 CAD における知識利用の形態を分類した論文⁹⁾では次の四つにシステムを分類をしている . (i) Generative KBE system , (ii) Advisory KBE system , (iii) Inovative KBE system , (iv) Selection KBE system .
本論文の分類では (i) は自動生成 , (ii) は検査 , (iii) は最適化に相当する . (iv) はスマートカタログと呼ぶ例えばユーザの入力した “ 材質 ” から対応する部品を選ぶようなシステムで , 本論文の分類では (i) になる .
- * 10 バルブレイアウト問題を簡略化したモデルを考える . 直角三角形の角度 (バルブ角度) , 高さ (シリンダヘッド高さ) , 底辺 (カム軸ピッチ) が与えられる . 直角三角形は , 角度と高さから底辺 , あるいは底辺と高さから角度 , あるいは角度と底辺から高さが決るので , 三つの拘束条件を直角三角形に与えることは過剰拘束である .

- 参考文献** [1] 3 次元 CAD に知識を組み込める知識ベースモデリング (前 , 後編) 日経コンピュータグラフィックス , 2 月号 pp.128 ~ 133, 3 月号 pp.106 111, 2000.
- [2] 日本電気 通信装置設計の知識駆動型設計システム (KDDD 開発事例) 精密工学会誌 Vol.67 No 3, 2001, pp.515 520.
- [3] 畑村洋太郎 , 中尾政之 , 設計者が欲しい設計支援システムの開発 , 情報処理 , vol.41 No.7, 2000, pp.862 867.
- [4] 畑村洋太郎 , 中尾政之 , 創造的設計のデジタル支援 , 精密工学会誌 , Vol.67 No.5, 2001, pp.715 719.
- [5] J. A. Penoyer, G. Burnett, D. J. Fawcett, S Y Liou, Knowledge based product life cycle system : principles of integration of KBE and C 3 P, CAD, Vol.32, 2000, pp.311 320.
- [6] Hiroshi Sakurai, Parag Dave, Volume decomposition and feature recognition, PartII : Curved objects, CAD, Vol.28, 1996, pp.519 537.
- [7] Xiangping Chen and Christoph M Hoffmann, On editability of feature based design, CAD, Vol.27 No.12, 1995, pp.905 914.
- [8] Vasilis Capoyleas, Xiangping Chen and Christoph M Hofmann, Generic naming in generative constraint based design, CAD, Vol.28 No.1, 1996, pp.17 26.

執筆者紹介 宮 地 恵 美 (Emi Miyachi)

1957 年生 . 1982 年慶應義塾大学工学系研究科修士課程修了 . 同年日本ユニシス (株) 入社 . CAD/CAM システム開発に従事 . 現在エンジニアリングシステム部所属 . 情報処理学会員 .