CAD のための拘束条件モデリング環境†

山口 泰†† 木村文彦†††

機械製品の設計においては，設計者の意図を拘束条件という形で表現することが自然である．そこで，拘束条件を述語論理で統一的に表現し，これに演繹推論を施して，設計対象を具体化するという手法が研究されてきた．しかし，設計者にとって述語は直観的な拘束の表現法ではなく，述語で拘束条件を記述することは容易でない．本研究では，拘束条件記述を作成する作業を拘束条件モデリングと名づけ，設計者の負担軽減と創造性の支援を図る拘束条件モデリングシステムを提案する．すなわち，この拘束条件モデリングシステムでは，ユーザは述語論理を意識せずに具体的なモデルの上で作業を行い，これと並行してシステムが拘束条件記述を自動的に生成する．拘束条件記述の自動生成には，モデル操作の意味記述を利用する．また，具体的なモデリング作業を支援するために，モデル操作の依存関係を表す作業履歴を利用する．作業履歴によってモデルの正当性を保証しつつ，モデルを容易に変更することができる．オブジェクト指向の概念を用いてモデル操作を管理することにより，操作の意味記述，作業履歴の管理，操作の実行制御を統括的に実現する．さらに試作システムによって，この提案の有効性を確認した．

1. はじめに

機械製品の設計から生産までを一貫して支援する体系的な CAD/CAM システム実現の必要性は広く認識されている．しかるに，現行の CAD システムはいまだ設計者の設計意図や生産上の種々の制約を陽に扱うことができず，設計生産活動を有効に支援するものではない．このギャップを埋めるためには，拘束条件を陽に扱う必要があると考えられている[1),2)]．つまり，設計意図や製品の機能，加工性などに関する拘束条件を製品情報として直接取り扱う枠組みが必要とされている．このような製品情報管理を実現する枠組みとして，設計対象にかかわる拘束条件を述語論理で記述し，演繹推論を用いてモデルを詳細化していく手法がある．たとえば，寸法拘束を取り扱うパラメトリックな形状表現法[3)]などが研究され，有効性が認められている．

しかし，従来の研究は与えられた拘束条件をいかに利用するかという点に重きを置いており，設計者が拘束条件記述をいかに作成するかについては何ら触れられていなかった．述語によって拘束条件を記述する作業は，表現のわかりやすさや拘束の整合性などの様々な問題点を抱えている．設計者の負担を軽減し，創造性を向上させるためには，これらの問題を解決する必要がある．本研究では，拘束条件により記述されたモデルを‘拘束条件モデル’，拘束条件モデルを作る作業を‘拘束条件モデリング’と名付け，この拘束条件モデリング支援環境の構築法を提案するものである．拘束条件モデリングを支援するための基礎として，‘具体的なモデル’と操作の依存関係の2点に着目する．さらに，この2つの着想を統括的に実現するシステムの構築法としてオブジェクト指向の概念を利用する．以下，2章では具体的なモデルに対する操作と拘束条件の関わりについて述べる．3章で作業履歴をもとに操作の依存関係を表現し，これを利用する方法を説明する．4章では，オブジェクト指向の概念に基づく拘束条件モデリングシステムの構築法を示す．さらに5章で本研究の提案に基づいて作成した試作システムと実験結果について解説を行う．

2. 拘束条件モデリング

拘束条件モデルの具体的な応用として広く研究されているものに，寸法指定によって形状を変更する‘可変形状モデル’がある．従来，可変形状モデルにおける寸法拘束の表現法としては，手続き的な表現[4),5)]，数式による表現[6),7)]，述語による表現[7),8)]などがあった．一方，計算機で拘束条件を取り扱う手法として，論理型プログラミングが注目を集めている．たとえば，拘束条件の処理を主目的とした制約論理型プログラミング[9)]などの研究がある．CAD においては，多様な拘束条件を統一的に扱う必要があるため，拘束条件を述語論理によって表現することが適当と考えられる．

以下，説明の便宜上，2次元図形の寸法拘束を例に

† Constraint Modeling System for CAD by YASUSHI YAMAGUCHI (Department of Mechanical Engineering, Faculty of Engineering, Tokyo Denki University) and FUMIHIKO KIMURA (Research Center for Advanced Science and Technology, University of Tokyo).

†† 東京電機大学工学部機械工学科
††† 東京大学先端科学技術研究センター

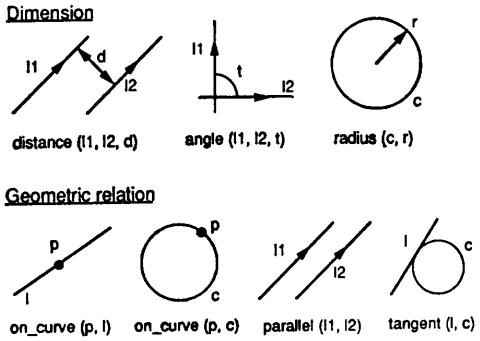


図 1 寸法拘束を表現するための述語例
Fig. 1 Samples of predicates to represent geometric constraints.

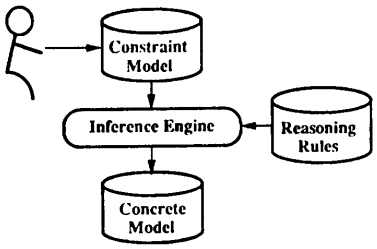


図 2 拘束条件モデルの具体化
Fig. 2 Constraint model evaluation to obtain a concrete model by deductive inference.

話を進めていく．寸法拘束を表現する述語の例を**図 1** に掲げる．述語表現の利点は表現力の豊かさにある．半径や長さなどの寸法値表現だけではなく，平行や接線といった形状要素間の関係も述語によって統一的に表現することができる．

拘束条件モデルは，拘束条件という要素間の関係を表すが，各要素の属性値を陽に表現するものではない．製品の評価や生産の段階では，全体構造とともに各要素の属性値が定まった'具体的なモデル'が必要となる．そこで拘束条件を評価してモデルを具体化しなくてはならない．述語論理を基本とした拘束条件モデルでは，**図 2** のように演繹推論を用いて具体的なモデルを導出する．この推論は推論規則を前向きに適用して進められる．寸法拘束から具体形状を生成する形状決定規則の例を**図 3** に示す．この例は，2本の直線がともに通過する点は，2直線の交点として計算できることを示している．このように形状決定規則は，特定

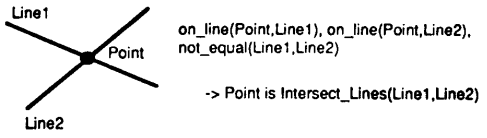


図 3 拘束条件評価のための推論規則の例
Fig. 3 A sample of inference rules for constraint evaluation (Intersection of two lines).

の関係を満たす形状要素群において，既定の形状要素から他の形状要素を決めるものである．形状決定規則を繰り返し適用することにより，形状全体が次第に具体化されていく．

2.1 拘束条件モデリングの問題点

述語論理に基づく拘束条件モデルは，拘束条件を統一的に表現でき，前向き推論によって具体化することができる．しかし，前提となる拘束条件モデルの構築にあたっては，いくつかの問題点がある．ここでは寸法拘束を例に，それらの問題点を明らかにする．

（1） 述語による拘束表現

述語を用いた拘束条件の表現は，手続きや数式等に比べ拘束の意図が読み取りやすいと言える．しかし，拘束の量が増すと，人間が拘束全体を把握することは容易でない．多様な拘束条件を統一的に表現できるのが述語表現の特徴ではあるが，**図 4** のように簡単な形状であっても述語の数が非常に多くなってしまう．これでは述語から形状を想像することは難しい．寸法拘束を扱う場合には，具体的な形状を思い浮かべて，その上で寸法を考えるのが普通である．一般に人間が拘束条件を扱う場合には，述語形式で考えるのではなく，各拘束条件に対応する具体的な対象を想定していると思われる．

（2） 拘束の整合性

拘束条件モデルにおいては，様々な拘束条件を評価することによって対象が具体化していく．この時，拘

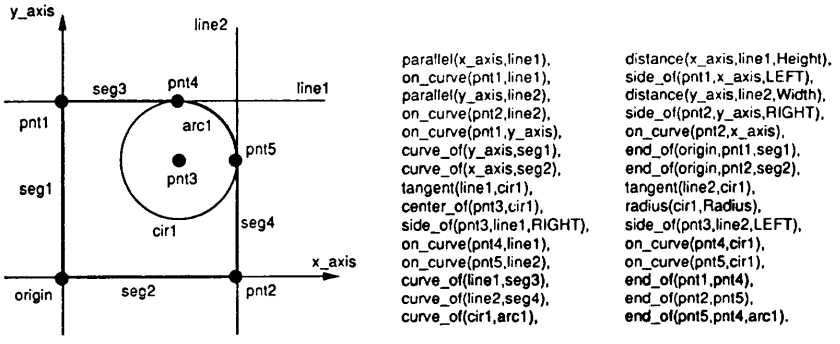


図 4 述語による寸法拘束の記述例
Fig. 4 A sample of geometric constraint description in logical predicates.

束条件には次のような理由から曖昧さや過剰が生じる．

① 暗黙の拘束

拘束条件には，人間の意図に直接結びつく明示的な拘束と，人間は意識しないが拘束条件を解くために必要となる暗黙の拘束がある．たとえば，3角形の3辺の長さは，3角形の形状を規定するのに十分な拘束を与える．しかし，計算機内の形状表現が頂点座標に基づく場合には，3頂点の座標値すなわち6自由度を拘束する必要があり，3つの拘束条件では不足である．これは3角形の位置と方向に対応するものである．このように計算機内モデルの表現形式に応じて，人間の意識しない暗黙の拘束が必要とされることが多い．

② 曖昧な拘束

拘束条件の表現には‘曖昧さ’の介入する余地が多分にある．たとえば，ある直線 l に関して平行で一定距離にある直線 l' を次のように述語で規定すると，

parallel (l, l'), distance (l, l', D)

基準の直線 l の両側に2通りの可能性が生じる．この曖昧さを排除するには，直線に向きをつけて左右を指定する必要がある．このような付加的な拘束をもれなく記述することは非常に困難である．従来の研究[7),8)]では，完全な拘束条件記述を与えるのではなく，参照形状という具体例を与えることによって不足した情報を補っていた．参照形状は位相的に全く等価で，寸法のみが異なる具体的な形状である．つまり平行線の場合，線間の距離は異なるものの左右の位置関係は等しい具体的な2直線が参照形状として与えられる．

③ 過剰拘束

拘束条件が過剰になることもある．これは人間が意図的に過剰指定する場合もあれば，予期せずに過剰になってしまう場合もある．一般に過剰な拘束条件は矛盾の原因となる．矛盾を解消するには，拘束条件を緩和するか，拘束条件に優先順位をつけるかしなくてはならない．したがって，拘束条件が過剰に与えられた場合には，過剰である事実とともに過剰となる拘束条件の組を人間に示すことが望ましい．

以上のように，拘束条件に曖昧さや過剰が生じる可能性は極めて高い．特に，全くの混沌状態から過不足のない拘束条件を得ることはほとんど不可能である．一方，前向き推論によって唯一具体的なモデルを得るためには，拘束条件に過不足があってはならない．まず，最終的なモデルに類似し，しかも過不足のない拘束条件モデルが与えられれば，これに新たな拘束条件を加えたり削除したりして，最終的な拘束条件モデルを得ることは比較的容易である．つまり，原型として曖昧さや過不足のない拘束条件モデルを作成することが肝要である．

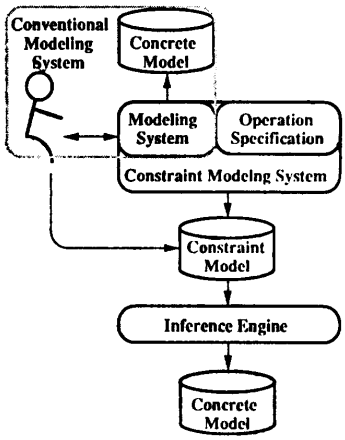


図 5 拘束条件モデリングシステム
Fig. 5 Constraint modeling system based on actual modeling system.

2.2 モデル操作と拘束条件

拘束条件モデリングには，述語という拘束表現の問題と拘束の整合性という問題がある．本研究では，これらの問題を解決するために，具体的なモデルを利用する．すなわち，**図5**のような拘束条件モデリングシステムを提案する．ユーザが従来型の具体的なモデル構築操作を行うと，システムはモデル操作の仕様記述をもとに過不足のない拘束条件記述を自動的に生成する．ユーザは意図に応じて，この拘束条件モデルに拘束条件を付加・削除する．これを推論機構で再評価することで，新たな拘束条件を満たす具体的なモデルが得られる．また，拘束条件の過剰が生じた場合には，過剰という事実を判定するとともに，過剰を導く拘束条件の組を選別することもできる．

（1） 操作の意味に基づく拘束

モデルの構築操作を実行すると，その操作にかかわったデータは特定の関係を満たす．このようなデータ間の関係は，操作の実行に伴って生じる拘束条件と見ることもできるし，より一般的に操作の意味とみなすこともできる．そこで操作ごとに意味記述を用意し，その意味記述に操作の実行結果をあてはめることで，実行に伴って生じるデータ間の拘束条件を導出する．操作の意味はデータの入出力仕様をもとに記述する．たとえば，2直線間の交点生成操作の場合，2つのデータが入力され1つのデータが出力される．したがって，入出力の仕様は次のようになる．

入力：In 1, In 2　出力：Out 1

入力2直線の双方にのる点を生成することが操作の意味であるから，次のようなデータ間の関係記述をもって操作の意味記述とする．

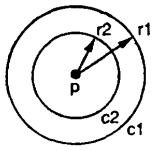
line(In 1), line(In 2), point(Out 1),
on_curve(Out 1, In 1), on_curve(Out 1, In 2)

実行時には具体的な直線データ（例：line 1, line 2）が入力され，点データ（例：point 1）が生成される．意味記述の述語の各項 In 1, In 2, Out 1 をこれらの具体的なデータ line 1, line 2, point 1 で置き換えると，

on_curve(point 1, line 1),
on_curve(point 1, line 2)

となる．これがデータ間に成り立つ拘束条件の記述となる．

モデル操作の意味記述はモデリングシステムの構築時に与える．一般にモデル操作は決定的な手続きであるから，モデル操作の意味も決定的に記述できる．つまり操作の意味記述が適当であれば，過不足のない拘束条件が導出できるはずである．たとえば，ある直線 l から一定距離だけ離れた平行線 l' を生成する場合には2つの可能性があるため，操作の過程でユーザにもとの線 l のどちら側に平行線を生成するかを指定させるのが普通である．この指定も拘束条件の一部として記述されていなくてはならない．新たに生成する直線



center_of(p, c1), center_of(p, c2),
radius(c1, r1), radius(c2, r2),
r1 > r2.

図 6 同心円の大小関係の表現
Fig. 6 Size difference description between concentric circles.

（3） 過剰な拘束条件

具体的なモデルに対する操作として，モデル構築の操作だけでなく，モデルの評価も含めることを考える．具体的なモデルに対する評価操作とは，モデルを何らかの尺度で数量化するものである．モデル構築操作が具体的なモデルを新たに生成・変更する構成的な拘束条件を与えるのに対して，評価操作は既定の部分に過剰な拘束条件を与える．

この拘束の過剰は必ずしも拘束条件全体にわたるものではない．評価対象が局所的であれば拘束の過剰も局所的に起こる．つまり，評価対象を規定する構成的な拘束条件と評価操作に伴う付加的な拘束条件との間で拘束の過剰が生じている．したがって，評価対象部分に対するモデル構築操作をたどることで，過剰な拘束条件の組合せを抽出できる．その他の拘束条件は，拘束の過剰とは無関係であり，競合解決にあたって考慮の対象から外すことができる．

置関係を記述することで，拘束条件の曖昧さを排除できる．

point(p), on_curve(p, l'), side_of(p, l, LEFT)

注意すべき点は，この操作の意味記述をシステム構築者が行うことである．システム構築者は，過不足のない拘束条件が導出できるように，操作の意味を記述しなくてはならない．一方，ユーザは拘束条件の表現を意識せず，具体的なモデリング作業に専念できる．

（2） 数量に関する拘束

拘束条件には，平行線間の距離や円の半径など，数量に関するものがある．具体的なモデルを対象とする場合には，これらの数量は具体的な値をとる．たとえば，円を生成する場合には，具体的な数値が半径として指定される．したがって，意味記述を利用してシステムが生成する拘束条件は，具体的な数値を持ったものになる．ここで具体的な数値に代えて変数を用いれば，等式や不等式によって数量的な拘束が表現できる．たとえば，同心円の大小関係は図6のように表現

3. モデル操作の依存関係

モデル操作の依存関係は，モデリング作業の支援に利用できる．モデルを構築するユーザにとって，モデルを操作の前後関係から捉えることは自然である．丸まった角は，とがった角を丸めたものと見ることもできるし，円に2本の接線を引いて作ったものと考えることもできる．モデル操作の履歴には，結果のモデルからは伺い知ることのできない，ユーザの意図にかかわる情報が含まれている．このモデル操作の前後関係を‘モデルの由来’と呼ぶ．モデルの由来によって，生成，評価，修正というモデル構築のサイクルにおいて，モデルの正当性を保証するとともにモデルを容易に変更することが可能になる．

3.1 作業履歴

モデルの由来を明らかにするために，モデリング作業の履歴を記録する．この作業履歴は，従来からあるコマンドログのような入力文字列ではなく，モデル操作の依存関係を明示的に表現するものである．個々の

べての操作に依存するわけではない．作業履歴は各操作が以前のどの操作に依存するかを示すものである．

作業履歴はモデル操作を単位として記述される．つまり，モデル操作を実行するごとに作業履歴に‘履歴要素’が1つ付加されていく．履歴要素は，inputs と outputs の2つの関係からなる．inputs 関係は操作が引数として受けとった入力データの並びを示す．一方，outputs 関係は操作が実行結果として出力したデータの並びを表す．2直線の交点を算出する操作の場合には，履歴要素は次のようになる．

inputs(intersect_lines$X$, line1, line2)

outputs(intersect_lines$X$, point1, point2)

作業履歴は操作とデータの2種類のノードを持つ有向グラフになる．操作ノードは各操作の実行を表し，リンクは操作の実行にあたってのデータの流れを示している．したがって，リンクの向きは時間経過の順序関係と一致しており，有向グラフはループを持ちえない．モデリング作業は多くの操作を積み重ねて進められるため，有向グラフは**図7**のような複雑なネットワークを構成する（点線矢印は inputs 関係に対応し，実線矢印は outputs 関係に対応する）．このネットワークは実行された操作とデータの間の依存関係を示したものと考えられる．そこで，これを‘依存関係ネットワーク’と呼ぶ．依存関係ネットワークをたどることによって，モデルの由来を明らかにし，モデリング作業を支援する．この時の軌跡はネットワーク上を広がっていくので，図7の矢印方向に時間順でたどることを‘前向き伝播’，またこれと逆にたどることを‘後向き伝播’と呼ぶ．

3.2 前向き伝播

作業履歴を再実行することで，モデリング作業の全体または一部を再現できる．部分的な再実行は作業のマクロ化に相当する．履歴要素を実行順序にたどり，これと同じ操作を実行すればよい．ただし，作業履歴上に表れるデータはあくまで過去のものであり，再実行時には新たな文脈に合わせて，操作を実行しなくてはならない．このために履歴上の過去のデータと再実行時の新しいデータの対応づけが必要となる．操作を実行するごとに結果の出力データと履歴の出力データを比べて対応づけを行う．時間の順序関係を保つため，再実行とともに依存関係ネットワークも再構成する．

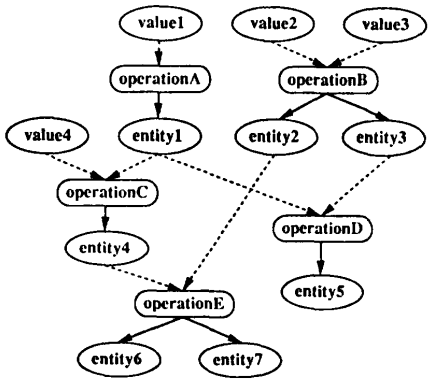


図7 依存関係ネットワーク
Fig. 7 Dependency network of modeling operations.

作業履歴を変更して再実行すれば，結果として変更されたモデルが生成される．変更箇所から依存関係ネットワークを前向き伝播でたどって到達可能な操作のみを再実行することで，変更の影響を受ける部分のみを再実行・修正できる．この時，履歴の変更によっては，再実行結果のモデルがもとの履歴のモデルと対応しなくなる場合がある．たとえば，直線と円の交点計算では，円の半径があまりに小さくなると交点がなくなってしまう．このもとのモデルと変更後のモデルの対応関係をモデルの正当性と呼ぶ．モデリング作業の位相同型性をもとにモデルの正当性を保証する．つまり，操作の実行ごとに結果のデータの個数や型を対応させ，履歴と再実行の位相同型性を保ち，結果としてモデルの正当性を保証するものである．

通常は，モデリング作業の意図を保持するために，再実行と履歴の位相同型性を保持する．しかし，意図を変更したい場合には，この機構を制御する必要がある．たとえば，同じ高さ基準を持つ2つのボスの一方の高さを変更すると，それに伴って他方の高さも変わってしまう．ボスの高さを独立にするには，高さに関する依存関係を変えなくてはならない．つまり，2つの異なった高さ基準が必要であり，作業を位相同型ではなくす必要がある．

3.3 後向き伝播

作業履歴の変更によってモデルを修正するには，ユーザが作業履歴から適当な操作を選んで変更する必要がある．しかし，モデルと実行した操作の間の関連は，モデルを作った本人にさえ明らかでないことがある．そこで，結果のモデルから原因にあたる操作への写像関係をユーザに示すようにする．この写像関係は後向き伝播によって得られる．

一般にモデリングの過程では試行錯誤や後戻りが頻繁に行われる．その結果，途中の操作の中には最終結果のモデルに何の影響も及ぼさないものも多い．後向

き伝播を用いて，これらの操作を作業履歴から枝刈りすることができる．つまり，操作を次のようにアクティブなものとインアクティブなものに分類する．

アクティブな操作：

出力データが，“ユーザの関心のあるデータ”か“アクティブな操作の入力データ”である操作．

インテクティブな操作：

アクティブでない操作で，枝刈りの対象となる．

この分類作業には後向き伝播が利用される．これを実行時に並行して行い，履歴を動的に編集する．インアクティブな操作は最終的な結果に関与しないのであるから，インアクティブな操作に伴って導かれた拘束も不要である．つまり，履歴自体の動的編集とともに不要な拘束条件も編集できる．

4. オブジェクト指向による操作の管理

拘束条件モデリングシステムは，具体的なモデル操作に基づく拘束条件記述の自動生成と操作の依存関係の管理によって，設計者の負担軽減と創造性の支援を図るものである．このようなシステムを実現するために，オブジェクト指向の概念が非常に有用である．ここでは，オブジェクト指向の概念に基づいた拘束条件モデリングシステムの構築法を解説する．

4.1 オブジェクト指向プログラミング

オブジェクト指向プログラミングは，データ抽象を推し進めたプログラミング法である[10]．‘オブジェクト’とは，プログラムの内部構造を隠蔽するために，データに作用する手続き群をデータとともにまとめたもので，カプセル化された内部状態と内部状態を操作する‘メソッド’からなる．

このオブジェクト指向を強力に支えるものとして，‘クラス・インスタンス階層’と‘メッセージパッシング’がある．クラスとは同種のオブジェクトを抽象化した概念であり，クラスに属する個々のオブジェクトをインスタンスと呼ぶ．さらに同種のクラスを抽象化した上位概念をスーパクラスと呼ぶ．この階層構造が，クラス・インスタンス階層であり，上位から下位へと性質が継承される．またメッセージパッシングは，オブジェクト間でメッセージを受け渡すことによってメソッドを起動する機構である．メッセージパッシングはコルーチンの制御に有効であり，オブジェクトを計算プロセスと見立てて並列計算に利用しようという試みもある[11]．

4.2 操作の管理法

拘束条件の導出に用いる操作の意味記述は，操作の型ごとに用意されていればよい．一方，作業履歴表現においては，実行されたすべての操作が区別されなくてはならない．たとえ同じ型の操作であっても，2回実行されたならば，その2回の操作は別の操作として扱われる．この操作の型と実行される操作の関係は，クラスとインスタンスの関係になっている．つまり，操作の型ごとに操作のクラスを定義し，操作の意味をクラスに記述する．実行時には操作のインスタンスが生成され，作業履歴は操作インスタンスとデータをノードとする依存関係ネットワークとして記述される．

また操作をオブジェクトとみなすと，制御の移行にメッセージパッシングが利用できる．操作の実行時には，制御がシステムのトップレベルから操作に移る．つまり，システムのトップレベルと個々の操作はコルーチンの関係にあり，制御をメッセージパッシングによって移すことができる．この制御の流れを示したものが図8である．操作のインスタンスを生成して，トップレベルからメッセージ‘RUN’を送ることで，制御が操作に移り，操作の実行が開始される．操作の実行が完了すると，操作のインスタンスがメッセージ‘COMPLETE’をトップレベルに送り，制御がトップレベルに戻る．操作の実行を一時中断する場合には，操作インスタンスは，その時点での環境を内部に保持してメッセージ‘SUSPEND’を送り，制御を戻す．実行を再開する場合には，操作インスタンスにメッセージ‘RESUME’を送ればよい．これにより操作を一時中断して，別の操作を実行することができる．操作インスタンスは同時に複数作れるので，中断したものと同種の操作を実行することもできる．

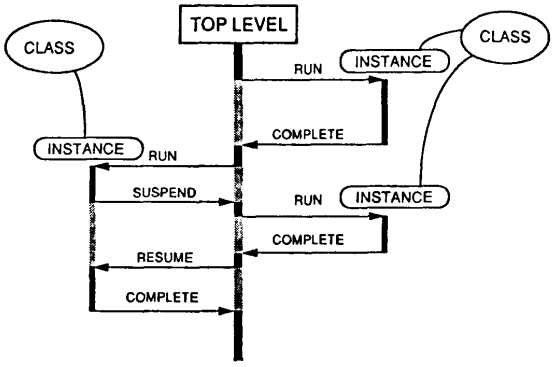


図8 操作の制御
Fig. 8 Flow of operation control.

5. システムの試作

本研究では，SUN ワークステーション上の Lisp システム (KCL)[12] を用いて，拘束条件モデリングシステムを試作し，実験を行った．**図 9** は試作システムの概要である．例題として2次元図形を対象とした．ユーザは一般的な作図作業を行い，システムは作業履歴を管理しつつ，2次元図形の拘束条件記述を自動的に生成する．生成された拘束条件記述を前向き推論機構に入力して，図形が再構成できることを確認した．

図 10 は作図操作の仕様記述例である．操作の入力変数，出力変数，それ以外の変数を定義する．この例では，2直線間の交点生成操作がデータを2つ入力し，1つ出力することを示している．この変数の入出力の関係をもとに作業履歴が構築される．操作の意味は，これらの変数間の関係として定義される．例では，出力されるデータは点であり，入力された線の上にのることが示されている．

5.1 作業履歴の管理

2次元図形の作図システムは，位相情報を持たない非常に簡単なものである．作図作業は原点と X 軸，Y 軸を基準にして進められる．**図 11** は，ある作図過程の最初の段階を示したもので，直交する2本の一点鎖線が X 軸と Y 軸である．ここまでの作図作業は，

1. 原点を中心とし指定された半径を持つ円を描き，
2. X 軸からある距離だけ離れた平行線を引き，
3. 円と直線の交点を求め，
4. この交点を通り Y 軸に平行な線を引き，
5. 同じ交点を中心とした指定半径の円を描いた．

この作業を進めて得られた最終結果が**図 12** である．

履歴管理によって容易に形状を変更できる．まず，**図 13** は作図の過程を照会した様子である．ユーザがフランジ下部の穴をマウスでピックすることで，この部分の作図過程が表示された．これから軸間距離と軸のズレで他方の軸を決め，その軸を中心とする4つの穴の1つであることがわかる．操作時に入力された数値が操作の意味に応じて寸法線形式で表示されている．

このように図の寸法線は作図操作と1対1に対応しており，寸法線をピックすることで作業履歴上の操作インスタンスを選択できる．数値を変えて作業履歴を

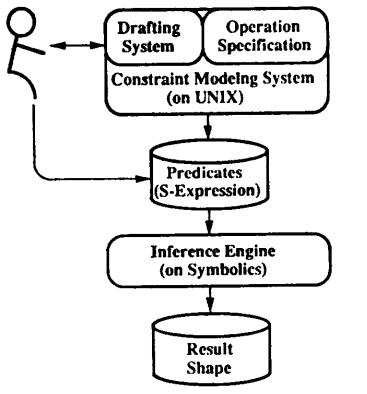


図 9 試作システム
Fig. 9 Prototype system.

```
(defoperation intersect-lines (l1 l2) (pnt) ( )
   :relations ((assert-fact (point pnt))
               (assert-fact (on-curve pnt l1))
               (assert-fact (on-curve pnt l2))))
```

図 10 操作の仕様記述の例
Fig. 10 A sample of operation specification description.

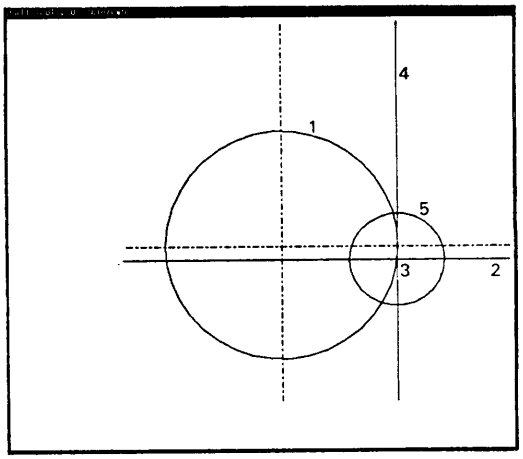


図 11 作図作業の進行
Fig. 11 Early stage of a drawing process.

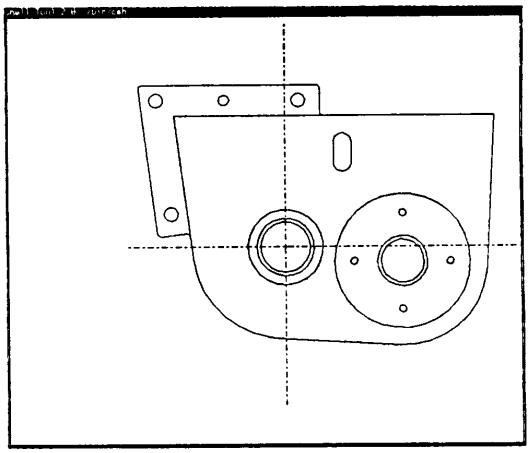


図 12 最終形状
Fig. 12 The result figure.

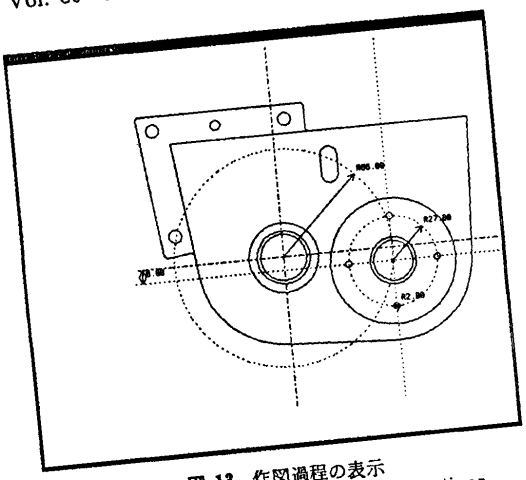


図 13 作図過程の表示
Fig. 13 The display of causing operations.

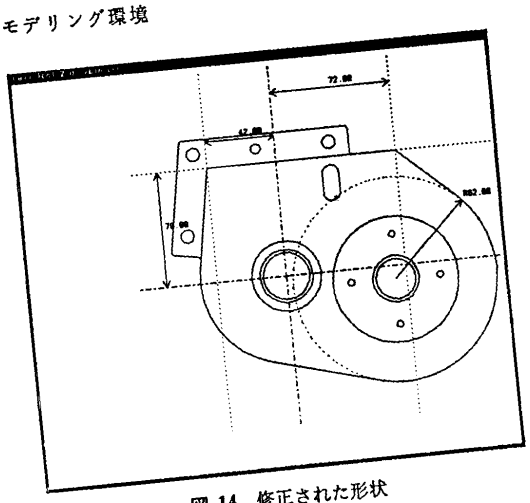


図 14 修正された形状
Fig. 14 The modified figure.

再実行すれば，寸法を変えるようなイメージで形状を変更できる．前向き伝播によって，実際に変化する部分のみが更新される．寸法線表示と数値の変更を繰り返して形状を徐々に変化させ，望みの形状を得ることができる．たとえば図 12 を修正して**図 14** のような形状が得られる．図の寸法線は図 12 から変更した部分を表している．

図 15 は作業意図を変更した例である．一方のボスの高さを変更しようとしたところ(①)，同じ高さ基準を用いていたために両方とも高さが変ってしまった(②)．そこで，変更を望まない部分に対して変更の取消しを指定する(③)．その結果，高さの異なる 2 つのボスが作られ，各々が独自の高さ基準を持つようになった(④)．つまり，同じ高さと指定されていた 2 つのボスの高さを独立にしたことになる．これはボスの高さについての意図を変更したものと考えられる．

5.2 拘束条件モデルの生成

ユーザは具体的な作図を行い，システムは拘束条件記述を自動的に生成する．しかも，直線の向きや左右の区別など，ユーザの意識しない計算機固有の情報を扱うことで，過不足のない述語表現が得られる．ユーザは単に作図に専念すればよく，拘束条件モデリングのために特に行うべき作業や意識すべき事柄はない．図 12 の形状の場合には，システムは 459 個の述語からなる拘束条件記述を生成した．途中の試行錯誤に基づく最終結果に無関係な拘束条件は，作業履歴の動的編集機能によって削除されている．**図 16** は，図 11 の段階までで生成された拘束条件記述である．補助的な述語 (SIDE-OF, ORDER-ON) によって，曖昧さを排除し，過不足のない表現になっている．

この拘束条件記述を Symbolics 3640 上の前向き推論システム[13]に入力することにより，形状を再構成したものが**図 17** である．再構成に要した時間は 10 秒程度であった．操作の意味記述が決定的であるため，参照形状なしで形状を再構成できた．この拘束記述を人間が直接行うと，参照形状を利用しても丸 1 日ほどかかってしまう．参照形状を利用せずに過不足ない拘束条件記述を得ることはほとんど不可能と言える．

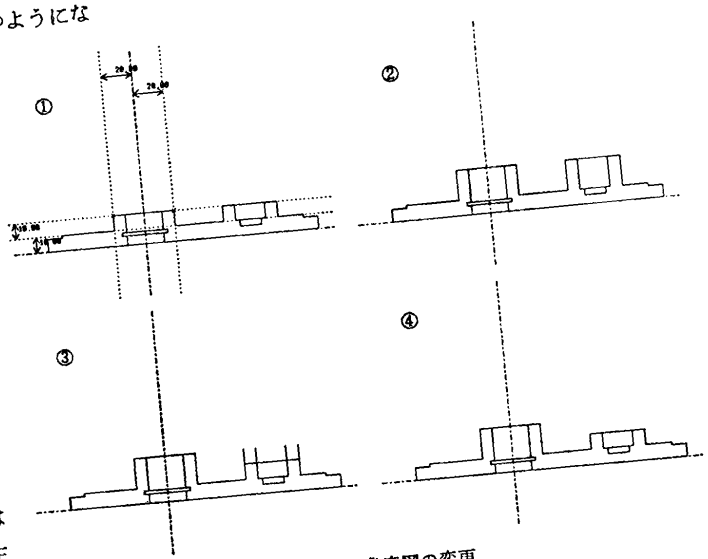


図 15 作業意図の変更
Fig. 15 Modification of modeling intention.

```
(POINT ORIGIN)                                  ┐
(COORDINATE ORIGIN 0 0)                         │
(LINE X-AXIS)                                   │
(ON-CURVE ORIGIN X-AXIS)                        │ 原点
(DIRECTION X-AXIS 1 0)                          │ 座標軸の定義
(LINE Y-AXIS)                                   │
(DIRECTION Y-AXIS 0 1)                          │
(ON-CURVE ORIGIN Y-AXIS)                        ┘
(CIRCLE CIRCLE8)                                ┐
(CENTER-OF ORIGIN CIRCLE8)                      │ 操作1
(RADIUS CIRCLE8 66)                             ┘
(LINE LINE15)                                   ┐
(PARALLEL X-AXIS LINE15)                        │
(DISTANCE X-AXIS LINE15 8)                      │ 操作2
(POINT POINT19)                                 │
(ON-CURVE POINT19 LINE15)                       │
(SIDE-OF POINT19 X-AXIS RIGHT)                  ┘
(POINT POINT21)                                 ┐
(ON-CURVE POINT21 CIRCLE8)                      │
(ON-CURVE POINT21 LINE15)                       │ 操作3
(POINT POINT22)                                 │
(ON-CURVE POINT22 CIRCLE8)                      │
(ON-CURVE POINT22 LINE15)                       │
(ORDER-ON POINT22 POINT21 LINE15)               ┘
(LINE LINE24)                                   ┐
(ON-CURVE POINT21 LINE24)                       │ 操作4
(PARALLEL Y-AXIS LINE24)                        ┘
(CIRCLE CIRCLE28)                               ┐
(CENTER-OF POINT21 CIRCLE28)                    │ 操作5
(RADIUS CIRCLE28 27)                            ┘
```

図 16 生成された拘束条件記述
Fig. 16 The generated constraint description.

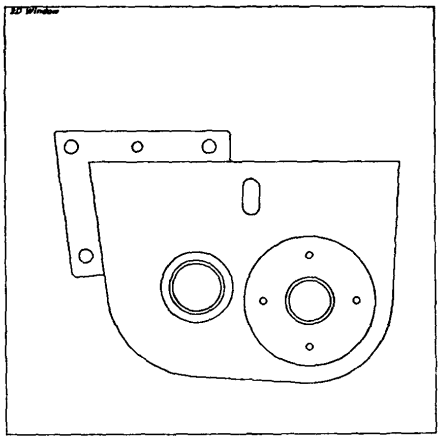


図 17 推論機構で再構成された図形
Fig. 17 The reconstructed figure by the inference engine.

6. おわりに

本研究では，拘束条件モデリングの問題点を明らかにするとともに，この問題点を解決する拘束条件モデリングシステムを提案した．まず，拘束条件モデリングには，述語という拘束表現の問題と拘束の整合性という問題があることを示した．拘束条件モデリングシステムは，ユーザには具体的なモデリング作業を行わせて，過不足のない拘束条件記述を自動的に生成するものである．このためにシステム構築者はモデル操作の意味を記述し，この意味記述をもとに拘束条件記述を導出する．また操作の依存関係を表す作業履歴をもとに，具体的なモデリング作業を支援する．作業履歴の管理によって，モデルの正当性を保証しつつ，モデルを容易に変更することが可能となる．またシステムの構築法として，モデル操作をオブジェクト指向の概念で管理する方法を提案した．これによって，操作の実行制御，操作の意味記述，作業履歴の管理を統括的に実現できる．さらに試作システムによって，この提案の有効性を確認した．

謝辞 本研究の一部は，精密工学会産学協同研究協議会“高度生産自動化のためのプロダクトモデリングシステム開発研究協力分科会”によるものである．

参考文献

1) Kimura, F., Kawabe, S. and Sata, T.: A Study on Product Modelling for Integration of CAD/CAM, *Proc. of the IFIP WG 5.2/WG 5.3 Working Conf. on Integration of CAD/CAM*, pp. 227-252 (1984).
2) Krause, F.-L., Armbrust, P. and Bienert, M.: Methods Banks and Product Models as Basis for Integrated Design and Manufacturing, *Proc. of the 2nd Int. Conf. on the Manufacturing Science and Technologie of the Futures* (1985).
3) 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, pp. 105-110 (1986).
4) 沖野教郎：自動設計の方法論，p. 192，養賢堂 (1982).
5) Brown, M.: PADL-2: A Technical Summary, *IEEE CG & A*, Vol. 2, No. 3, pp. 69-83 (1982).
6) Hillyard, R.C. and Braid, I.C.: Analysis of Dimensions and Tolerances in Computer-Aided Mechanical Design, *Comput. Aided Des.*, Vol. 10, No. 3, pp. 161-166 (1978).
7) Light, R. and Gossard, D.: Modification of Geometric Models through Variational Geometry, *Comput. Aided Des.*, Vol. 14, No. 4, pp. 209-214 (1982).
8) Aldefeld, B.: Variation of Geometries Based on a Geometric-Reasoning Method, *Comput. Aided Des.*, Vol. 20, No. 3, pp. 117-126 (1988).

9) Jaffer, J. and Lassez, J.-L.: Constraint Logic Programming, *Proc. of 14th ACM Principles of Programming Languages Conf.*, pp. 111-119 (1987).
10) 片山卓也：非手続き型計算言語におけるデータ構造，情報処理，Vol. 27, No. 2, pp 151-159 (1986).
11) 石川 裕，所真理雄：オブジェクト指向並行プログラミング言語，情報処理，Vol. 29, No. 4, pp. 325-333 (1988).
12) Yuasa, T. and Hagiya, M.: Kyoto Common Lisp Reference Manual for SUN Workstation, p. 91, Research Institute for Mathematical Sciences (1984).
13) 安藤英俊：機械設計のための幾何拘束処理システム，昭和62年度東京大学修士論文 (1988).

(昭和63年12月 5 日受付)
(平成元年 9 月12日採録)

山口 泰（正会員）

1961年生．1988年東京大学大学院博士課程（情報工学専攻）修了．工学博士．同年東京大学教養学部助手．1989年より東京電機大学工学部機械工学科講師．機械系CAD/CAM，形状モデリング，コンピュータ・グラフィクス，ユーザインタフェース，知識工学などの研究に従事．精密工学会，ACM，日本ソフトウェア科学会などの会員．

木村 文彦（正会員）

昭和20年生．昭和49年東京大学大学院博士課程修了．同年電子技術総合研究所パターン情報部入所．昭和54年より東京大学工学部精密機械工学科助教授．昭和62年より同教授．マン・マシン・システム，コンピュータ・グラフィックス，形状モデリング，CAD/CAMなどの研究に従事．工学博士．IFIP-WG 5.2-5.2-5.3 委員．精密工学会，日本機械学会などの各会員．