

機械設計におけるモデル作成過程の計算機支援手法に関する研究

A study of computer assistance of product model construction in machine design

代表研究者 茨城大学工学部システム工学科助教授 乾 正 知
Assoc. Prof., Department of System Engng., Ibaraki Univ.
Masatomo INUI

Assistance of product model construction is a fundamental task of computer-aided design systems. A general mechanism for recording and manipulating model construction process in computers is proposed. The model construction process is considered as a series of model state transitions realized by modeling operations. This process can only be properly recorded with dependencies between the operations. The dependency information is automatically detected on the basis of the conditions necessary for the execution of an operation. An assumption-based truth maintenance system is used as a tool for recording and manipulating the dependency. Two basic functions for assisting with model construction activities, an endo function and a model maintenance function, are realized by using the dependency in a uniform manner. The applicability of these functions is verified using interactive solid modeling and machining process planning examples.

研究の目的

機械設計における最も基本的な作業は、設計対象のモデルを作成することである。設計者は、製品に対する機能的な要求と設計に関するさまざまな知識に基づき、製品のアイデアをまずスケッチ(=モデル)の形にまとめる。このモデルは、設計者の製品に関する理解が進むにつれて詳細に詳細化され、機械図面へ姿を変えていく。機械設計でのCADの利用が一般化するにつれて、図面による機械製品の表現では不都合を生じることが多くなってきた。図面に代わる機械製品の新しい計算機内表現が、プロダクトモデルである。プロダクトモデリング技術にはまだ研究途上の部分が多いが、その柔軟性と汎用性から、形状情報を境界表現法と呼ばれる立体モデリング手法で処理し、技術情報を立体モデルの面や辺、頂点などの要素間の制約として処理する手法が注目されている。

機械設計とは本質的に試行錯誤を伴う作業であり、設計者は満足できる結果が得られるまで、プロダクトモデルの修正を繰り返す。このようなモ

デル作成作業を支援するためには、実行済みのモデル操作を自由に取り消したり、取り消した操作を再実行したりする機能をCADシステムに用意する必要がある。作成されたプロダクトモデルは、工程設計やNC加工命令生成、ロボット動作命令生成などの、さまざまな生産準備作業で利用される。近年、機械設計と生産準備作業を並行して行うコンカレントエンジニアリングが注目されている。この技術を実現するためには、設計側でモデルに何らかの変更を加えたとき、その影響を受ける工程計画や加工命令を自動的に検出し、可能ならば修正する機能が不可欠である。

このようなモデル作成作業の支援を実現するためには、モデルの作成過程、具体的にはモデル操作間の関係や、操作と操作によって変更されたモデル情報間の関係、そしてそのモデル情報に基づいて導出されたさまざまな生産準備情報間の関係を、計算機内に適切に表現し管理する技術が必要である。本研究の目的は、モデル操作間の依存情報に基づいてモデル作成過程を管理する手法を開

発し、さらにこの手法を利用して幾つかの応用プログラムを作成し実験することで、本手法の有効性を評価することである。

研究の経過

本研究は、以下に示す3段階に分けて進められた。

(1) モデル作成過程の管理手法の決定

一般にモデル操作は、「操作を実行するための前提条件」と「操作によるモデル記述の変更」の二つからなる。このことを簡単な積木モデルの操作を例に説明する。このモデルでは、ある積木 A が別な積木 B の上に存在することを $\text{On}(A, B)$ と述語表現する。また積木 A の上に何も載っていないことを $\text{Clear}(A)$ と表現する。このとき、積木 $?x$ を積木 $?y$ の上から取り除きテーブルに置く $\text{Unstack}(?x, ?y)$ 操作と、積木 $?x$ をテーブルから取り上げブロック $?y$ の上に載せる $\text{Stack}(?x, ?y)$ 操作は、それぞれ以下のように定義できる。ただし preconditions は、操作を実行するための前提条件を表し、delete list と add formula は、操作の実行によりモデルから削除される述語とモデルに付加される述語を表す。

$\text{Unstack}(?x, ?y)$
preconditions: $\text{On}(?x, ?y) \ \& \ \text{Clear}(?x)$
delete list: $\text{On}(?x, ?y)$
add formula: $\text{On}(?x, \text{Table}), \text{Clear}(?y)$
 $\text{Stack}(?x, ?y)$
preconditions: $\text{On}(?x, \text{Table}) \ \& \ \text{Clear}(?x) \ \& \ \text{Clear}(?y)$
delete list: $\text{On}(?x, \text{Table}), \text{Clear}(?y)$
add formula: $\text{On}(?x, ?y)$

これらの操作を用いて、Fig. 1 に示すような積木の塔を作ることを考える。モデルの初期状態 (0) に、順に $\text{Unstack}(A, B)$, $\text{Stack}(C, D)$, $\text{Stack}(B, C)$, $\text{Stack}(A, B)$ を適用した結果得られたモデルの状態を、図中(1)から(4)に示した。これまでの研究では、これらの操作の実行順序に基づいて、モデル作成過程を管理する手法が提案されてきた。この手法では、例えば $\text{Stack}(C, D)$ 操作を取り消すと、モデルは $\text{Stack}(C, D)$ を実行する直前の(1)の状態に変化する。独立性の高い数百の

操作によってモデルを作成する CAD 作業の支援を考えた場合、この手法は実際的ではない。例えば初期に実行された小さな穴明け操作を取り消しただけで、そのモデルに関する他の大部分の操作の影響も消えてしまう。

本研究では、これまでの手法とは異なり、モデル操作間の依存関係に基づいてモデル作成過程を管理する手法を提案する。あるモデル操作が実行されたとき、その操作とそれ以前に実行された操作の間の依存関係を検出することができる。Fig. 1 の例では、 $\text{Stack}(B, C)$ 操作を実行することで、モデルの状態は(2)から(3)に変わる。この $\text{Stack}(B, C)$ 操作の前提条件は、 $\text{On}(B, \text{Table})$, $\text{Clear}(B)$, $\text{Clear}(C)$ だが、この内 $\text{On}(B, \text{Table})$ と $\text{Clear}(C)$ は初期状態で定義されており、 $\text{Clear}(B)$ は $\text{Unstack}(B, C)$ 操作により実現されている。したがって $\text{Stack}(B, C)$ 操作は、モデルの初期状態と $\text{Unstack}(B, C)$ 操作に依存していると考えられる。同様の手法で、全操作間に図中に黒い矢印で示した依存関係を検出することができる。

(2) モデル作成過程の管理機構の実現

本研究では、依存関係を管理する機構として、ATMS: Assumption-based Truth Maintenance System (仮定に基づく真理値管理機構)を採用した。モデル作成過程を管理するために、以下の依存関係が ATMS により記録・管理される。

- 1) モデル操作間の依存関係 前述したように、モデル操作間の依存関係は、ある操作の前提条件にあたるモデル情報を定義したモデル操作を調べることで検出される。
- 2) 操作により付加もしくは削除されたモデル情報の依存関係 ある操作によりモデルに付加された情報は、その操作に依存すると考えられる。逆にある操作によりモデルから削除された情報は、その操作と矛盾すると考えられる。以上の考えに基づいて、操作とその操作により付加もしくは削除された情報の間に依存関係を定義できる。
- 3) モデル情報から導出された生産準備情報の依存関係 あるプロダクトモデルの情報に基

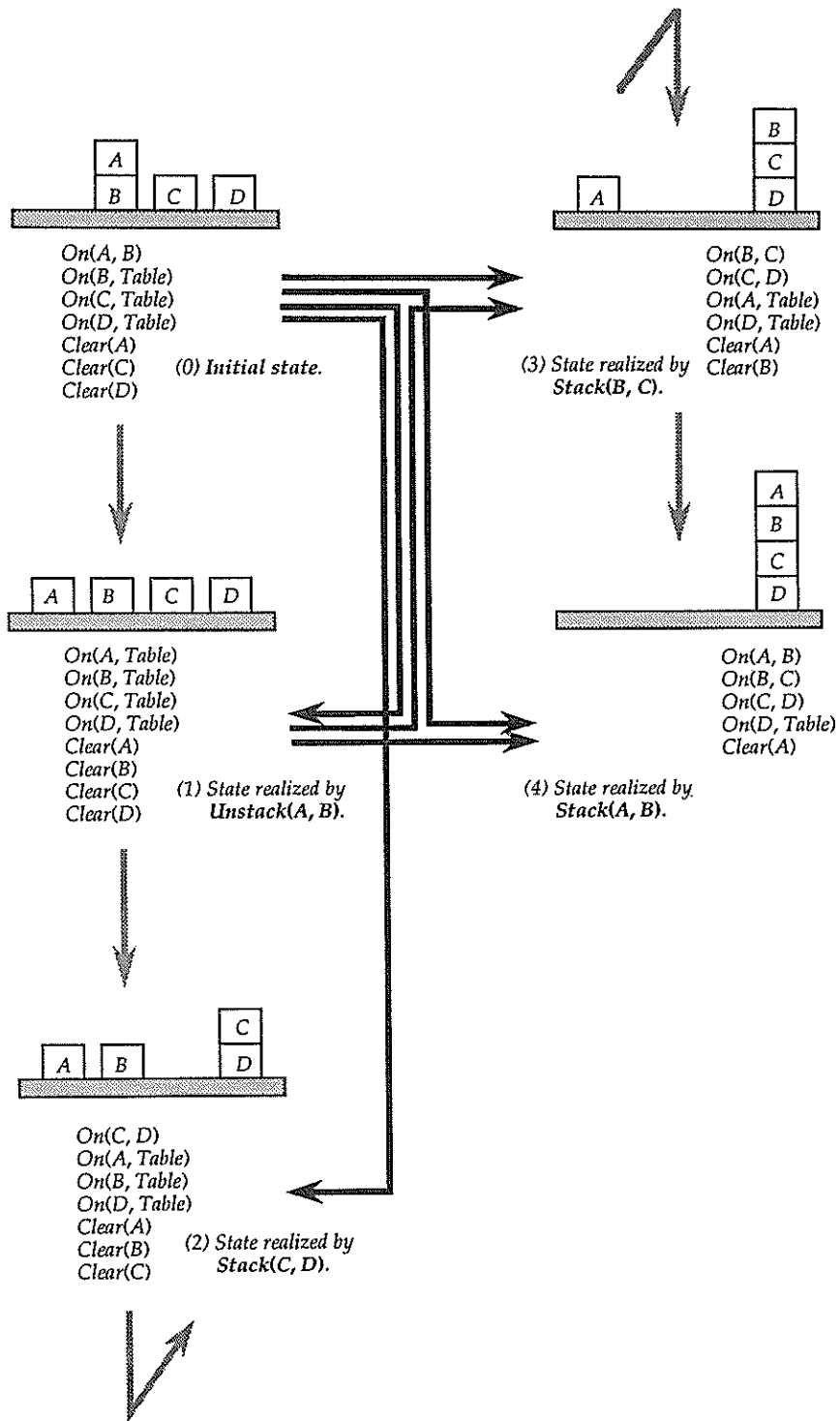


Fig. 1. An example block manipulation process.

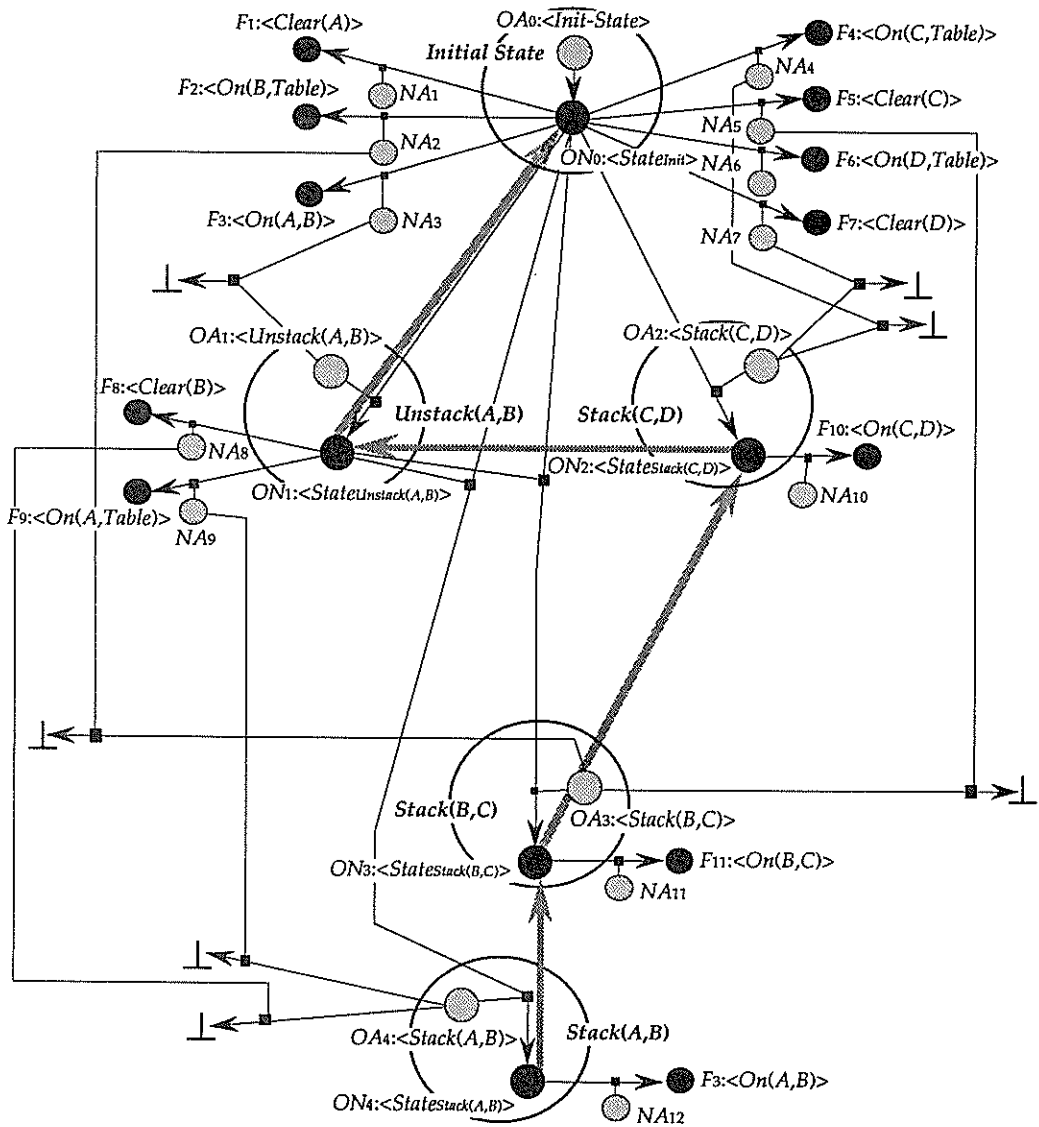


Fig. 2. Dependencies representing the block tower construction process shown in Fig. 1.

づいて例えば工程計画が生成されたとき、この工程計画はそのモデル情報に依存すると考えられる。

これらの依存関係以外に、モデル操作の実行順序を記録した。Fig. 2 には、前述の積木の塔のモデル作成に関して記録された依存関係を図示した。図中黒色の円はモデル操作やモデル情報を表し、それらを結ぶ矢印が依存関係を示す。ATMS で

は依存関係を識別する目的で、「假定」と呼ばれる特殊な情報を用いる。図中灰色の円が假定を表す。また逆 T 字の記号は、この記号に接続している情報が互いに矛盾することを表す。

あるモデル操作が取り消されたとき、その操作に依存するモデル情報などを適切に操作し、操作取り消しに対応するモデルを再構成する必要がある。またある情報がモデルから削除された時に

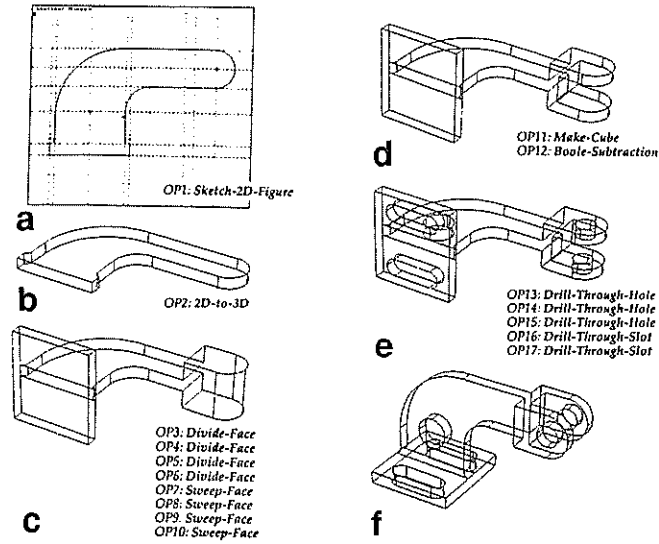


Fig. 3. An example solid model construction process.

OP1: Sketch-2D-Figure
 OP2: 2D-to-3D
 OP3: Divide-Face
 OP4: Divide-Face
 OP5: Divide-Face
 OP6: Divide-Face
 OP7: Sweep-Face
 OP8: Sweep-Face
 OP9: Sweep-Face
 OP10: Sweep-Face
 OP11: Make-Cube
 OP12: Boole-Subtraction <-- undo
 OP13: Drill-Through-Hole
 OP14: Drill-Through-Hole
 OP15: Drill-Through-Hole
 OP16: Drill-Through-Slot
 OP17: Drill-Through-Slot

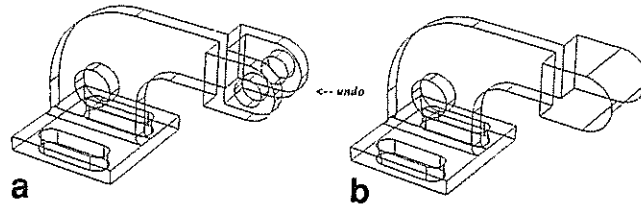


Fig. 4. An undoing example.

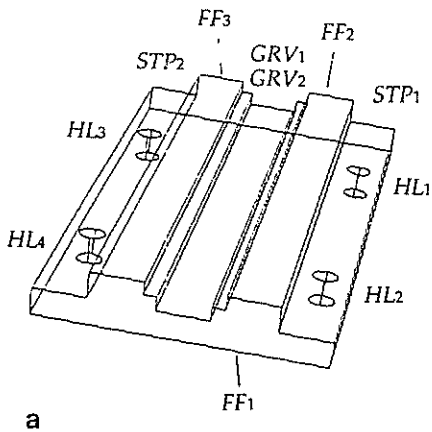
は、その情報に依存する他の生産準備情報を適切に更新する必要が生じる。これらの処理は、ATMSにより管理されている依存関係と、モデル操作の実行順序の記録を用いることで、実行済みのモデル操作の数に比例する時間で実現できる。

(3) 応用プログラムの作成と評価

提案したモデル作成過程の管理手法の有効性を評価するために、本手法に基づくいくつかの

CAD用プログラムを実現した。

1) 立体モデリングへの応用 本手法を用いて、立体モデルの作成過程を管理するシステムを実現した。立体モデリングでは、複数の立体モデルの空間的な集合和、差、積をとる集合演算と、局所形状変形操作を用いながらモデルを作成する。本研究では、これらの操作中に参照されるモデル情報が、操作を実行するための前提条件にあたることを考え、操作間



FIXING WORKPIECE Z+ DIRECTION
RESTING FACE FF₁

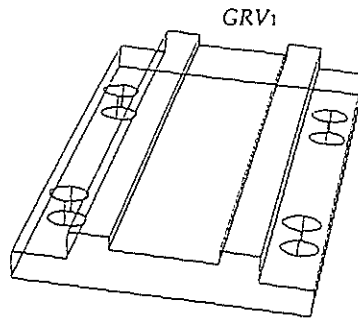
1. ROUGH FACE-MILLING F-FACE (FF₂ FF₃)
2. FIN FACE-MILLING F-FACE (FF₂ FF₃)
3. ROUGH END-MILLING STEP (STP₁ STP₂)
4. FIN END-MILLING STEP (STP₁ STP₂)
5. ROUGH END-MILLING GROOVE (GRV₁)
6. ROUGH END-MILLING GROOVE (GRV₂)
7. FIN END-MILLING GROOVE (GRV₁)
8. CENTER-DRILLING HOLE (HL₁ HL₂ HL₃ HL₄)
9. DRILLING HOLE (HL₁ HL₂ HL₃ HL₄)
10. FIN END-MILLING HOLE (HL₁ HL₂ HL₃ HL₄)
11. FINE END-MILLING GROOVE (GRV₂)

b

Fig. 5. Initial product model and its machining process plan.

の依存関係を検出した。Fig. 3には、簡単な機械部品の立体モデルを作成するプロセスを示した。モデルが完成した後、先端部の溝作成に用いた集合演算を取り消すと、システムは依存関係に基づいて Fig. 4 に示す新しい立体モデルを再構成する。

- 2) 工程設計への応用 本手法は、形体、寸法、公差などの技術情報のモデリングでも利用可能である。そこで本手法に基づくプロダクトモデリングシステムを試作した。このシステムを用いて機械部品のプロダクトモデルを作成し、これを以前開発した工程設計システムに入力することで、機械加工のための簡単な工程計画を生成した。このときモデルの情報と生成された工程計画の間に、自動的に依存



a

FIXING WORKPIECE Z+ DIRECTION
RESTING FACE FF₁

1. ROUGH FACE-MILLING F-FACE (FF₂ FF₃)
2. FIN FACE-MILLING F-FACE (FF₂ FF₃)
3. ROUGH END-MILLING STEP (STP₁ STP₂)
4. FIN END-MILLING STEP (STP₁ STP₂)
5. ROUGH END-MILLING GROOVE (GRV₁)
6. CENTER-DRILLING HOLE (HL₁ HL₂ HL₃ HL₄)
7. DRILLING HOLE (HL₁ HL₂ HL₃ HL₄)
8. FIN END-MILLING HOLE (HL₁ HL₂ HL₃ HL₄)
9. FIN END-MILLING GROOVE (GRV₁)

b

Fig. 6. Modified product model and regenerated process plan.

関係が定義される。したがって工程計画作成後にモデルの一部を修正すると、その部分に依存する工程計画が自動的に更新される。例えば Fig. 5 に示した部品の工程計画を作成した後で、部品の溝の一つを消去し穴の径に修正を加えると、自動的にそれらに依存する工程計画の一部が消去され、Fig. 6 に示す新しい計画が作成される。

本研究の成果

本研究により、以下の成果が得られた。

- (1) CADにおけるモデル作成過程を、モデル操作間の依存関係に基づいて管理する手法を提案した。
- (2) 本手法に基づくモデル作成過程管理システムを、ATMSを用いて試作した。
- (3) 試作したシステムを用いて立体モデリングシステムや工程設計システムを実現し、操作取り消しにもなうモデルの再構成や、工程計画の自動更新などを実験することで、本手

法の有効性を評価した。

今後の課題と発展

本手法をより有効なものにするためには、以下の二つの問題を解決する必要がある。

- (1) 操作の前提条件の検出 モデル操作が、「操作の前提条件」と「モデル記述の変更」の二つからなることは広く認められているが、これら二つは操作手続きの中に埋め込まれ、区別することがむずかしい場合が多い。本手法では、操作間の依存関係は操作の前提条件に基づいて検出されるため、これを正しく識別する手法の開発が必要である。
- (2) 依存関係の管理 本研究では、操作間の依存情報を管理するために ATMS を用いた。ATMS は汎用的なツールとして広く用いら

れているが、本手法のようにいくつかの特別な依存関係だけを扱う場合には不十分な機能が多く、処理速度低下の原因になっている。モデル作成過程管理に目的を限定した、より効率的な依存関係の処理手法を実現する必要がある。

発表論文リスト

(ただし学会誌または国際会議に発表した論文のうち、本研究と直接関係のあるもののみ記載)

- 1) M. Inui and F. Kimura: Using a truth-maintenance system to assist product-model construction for design and process planning. *Computer-Aided Design*, 25(1), 59-70 (1993).
- 2) M. Ranta, M. Inui, F. Kimura and M. Mantyla: Cut and Paste Based Modeling with Boundary Features. Proc. 2nd Symp. on Solid Modeling and Applications, ACM, 1993, pp. 303-312.