

設計過程の計算可能モデルに基づく設計シミュレーション

Design Simulation Based on a Computable Model of Design Processes

武田英明 河合浩之 富山哲男 吉川弘之

Hideaki TAKEDA Hiroyuki KAWAI Tetsuo TOMIYAMA Hiroyuki YOSHIKAWA

東京大学 工学部 精密機械工学科

Department of Precision Machinery Eng., Faculty of Eng., The University of Tokyo

This paper discusses a logical formalization of design processes to establish a computable design process model that is a crucial problem to realize intelligent CAD systems. First we show a cognitive model which is obtained by observing design processes with protocol analysis method. Then we establish a logical and computable framework that can explain the cognitive model. It is realized as a combined reasoning of *abduction*, *deduction* and *circumscription* that are executed repeatedly. Abduction is used to expand the designer's thought, deduction is used when the designer want to get all obtainable facts, and circumscription is applied to solve an inconsistency found during deductive reasoning. We implemented this reasoning as *design simulator* that revises design specifications and the design solutions progressively as the designer does in the design process in order to show that this framework is computable and explains design processes.

1 はじめに

知的な支援を行うCADシステムにおいては、設計対象の表現の支援のみならず、設計過程も支援することが必要である。このためには設計過程の理論およびモデル化が必要である。CAD構築という意味から、そのモデルは単に認知的(cognitive)・記述的(descriptive)モデルだけでなく、計算可能な(computable or computational)モデルである必要がある[1]。本研究では、認知的なモデルをもとにして設計過程の計算可能なモデルを提案する。

2 設計の認知的モデル

設計過程を研究する方法として心理学的手法（プロトコル解析）を用いた設計実験による解析がある[3]。その結果をもとに設計サイクルという認知的モデルを提案した[4]。これは「問題提起」・「提案」・「展開」・「評価」・「決定」という5段階からなるもので、設計過程において、設計解の各部分が決っていく過程を示したものである（図1参照）。設計過程全体はこのサイクルの連鎖としてみることが出来る。

また設計過程における思考には対象レベルと行為

レベルという2つのレベルがあり、設計サイクルはこの2つのレベルにまたがって行なわれている。

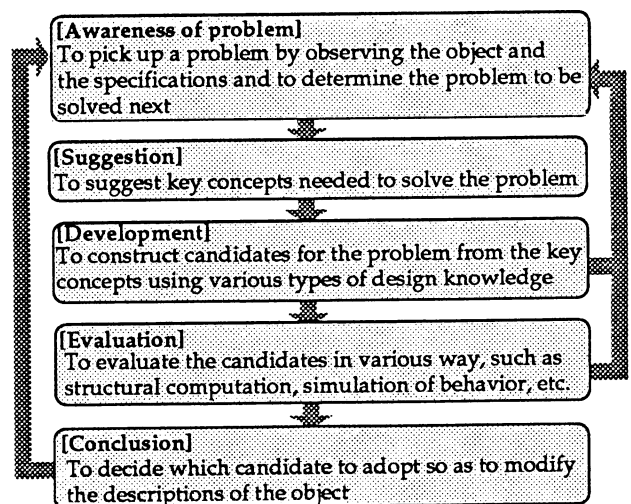


図1 設計サイクル

3 論理を用いた推論モデル

3.1. 論理的枠組み

ここでは設計過程を論理を用いてモデル化することを議論する。仕様から解を導出するという立場からは次のような方法がある。すなわち、S、K、Dsを各々論理式の集合とすると、

SUK-Ds

ただし、

S: 要求仕様, K: 設計知識, Ds: 設計解

とすることができる。すなわち、仕様と関係する知識を与えると、設計解を導出することができるわけである。しかし、この方法には以下の問題がある。

①設計は常に完全な情報に基づいて行なわれるわけでない。機械設計等においては仕様自身も設計過程において変化する。むしろ、仕様の詳細化も設計のひとつの役割であり、対象の詳細化と仕様の詳細化は交互に行なわれる。したがって、1方向だけの推論(仕様から対象へ)だけでなく、逆方向の推論も必要である。

②上記のモデルでは知識にあたるものは「いかに設計をするか」という設計方法知識である。しかし、このような知識だけでは柔軟な設計をするには不十分である。むしろ、設計対象の性質・挙動に関する知識が必要である。設計方法知識はこのような設計対象知識をもとに記述されるべきものである。

③知識は矛盾したり、不十分であることがある。しかし、設計における矛盾とは、本当に矛盾しているのではなく、多くの場合、知識の不完全性によって矛盾になっている。このような場合、知識の変更が必要になる。このような知識の変更も設計における重要な作業である。

そこで、ここでは次のような両方向の推論を行なうモデルを提案する。

DsU Ko-P

ただし、Ds: 設計解, Ko: 対象に関する知識

P: 設計解の性質・挙動

ここで、要求仕様はPに含まれる。設計はDsとKoからPを導く演繹推論とPとKoからDsを導くabduction[5]を交互に行なうことにより進む。まず、要求仕様と設計知識からabductionにより設計解の候補Dsを導く。次にそのDsを用いて、演繹推論を行い、性質Pを導き、矛盾がないかをみる。矛盾があった場合は、他の解候補の検討、仕様の変化、あるいは知識の変更などを行なう。仕様や知識に変化があった場合は再び、abductionと演繹を繰り返す。このようにして、設計解と仕様は順次詳細化される。

さらに、3番目の問題のために、circumscription[6]を導入する。設計における矛盾の多くの場合、知識自身の不完全性から生じている。すなわち、本来用いられるべき状況以外の場面で知識が用いられた、あるいは本来あるべき条件が書かれていなかったことから、見かけ上矛盾になると考える。そこでここではこの矛盾の場合とは例外の場合であるとする。すなわち、矛盾が検知された場合、例外項を各々に付加してcircumscriptionを行い、例外項を求める。これにより知識の更新を行い、矛盾を回避する。

3. 2. 論理を用いた推論モデル

ここでは2節で挙げた認知的設計過程モデルを論理的枠組みで解釈する。以下ではDs、Ko、Pのその時点、次の時点での内容を順にDs_c、Ko_c、P_c、Ds_n、Ko_n、P_nとする。

「提案」では設計者は求められている性質・挙動からもっともらしい解の候補を導きだす。しかし、それが本当に求めている解であるかどうかまでは決定していない。これは一種の仮説形成の推論であり、abductionとして解釈することができる。結果として、

$$Ds_n = abduction(P_c, Ko_c)$$

である。

他方、「展開」と「評価」の段階は演繹として解釈する。これらの段階では設計者は自分の持つ知識を適用して、わかりうることを全て導くことにより、解候補が求めている性質を持っているかを調べている。すなわち先の「提案」段階とは逆の過程であり、演繹で解釈する。2つの段階の違いは、「展開」段階では主に解の候補が持つ性質等を導く知識が使われるのに対して、「評価」段階では性質を比較する知識が使われる。すなわち、以下のように示される。

$$P_n = Deduction(Ds_c, Ko_c)$$

「展開」段階と「評価」段階において、設計者は解に問題点・困難な点を見つけると、新たな問題提起を行なう。この過程は矛盾を起こした知識を対象にしたcircumscriptionで解釈する。例えば、

$$\{A \rightarrow B, A \rightarrow \neg C, \neg C \rightarrow \neg B, A\}$$

という状態は矛盾である。そこで、

$$\{A \wedge \neg ab_1 \rightarrow B, A \wedge \neg ab_2 \rightarrow \neg C,$$

$$\neg C \wedge \neg ab_3 \rightarrow \neg B, A\}$$

というように abnormal 述語 ab_i を付加して、circumscribe を行なう。一つの解は、

$$ab_1 = \neg C, ab_2 = \text{false}, ab_3 = \text{false}$$

である。代入を行うと、

$$\{A \wedge C \rightarrow B, A \rightarrow \neg C, \neg C \rightarrow \neg B, A\}$$

矛盾はなくなり、もし $\neg B$ が必要なものであればこれで解決される。しかし B が必要なものであった場合は新たな問題提起が必要になる。一つの方法は C を導くことである。これが新しい問題になる (この場合、また矛盾が生じる)。結果は次のようになる。

$$Ko_n = \text{Circumscription}(Ds_n, P_n, Ko_n)$$

「展開」や「評価」段階で生じる問題提起は上で述べたように論理的枠組みの中で解釈可能であるが、一般的に「問題提起」自身は論理的な推論の外にあると思われる。この段階は用いる知識、対象、必要な性質などを準備する段階、すなわち、 Ds や Ko 、 P を準備する段階である。そこでこの段階は論理的枠組みに対する操作、メタレベルの操作として解釈する。ここでは論理的枠組みに対するメタレベルの操作としては、以下の6つを用意する。

- Ds の操作、 Ko の操作、 P の操作
- Ko と P からの abduction による Ds の更新、
- Ds と Ko からの演繹による P の更新、
- Ds, P, Ko からの circumscription による Ko の更新

設計サイクルは以上の3つの推論とそれらの制御である上記の6つの操作を導き出す行為レベルの推論によって解釈される (図2参照)。

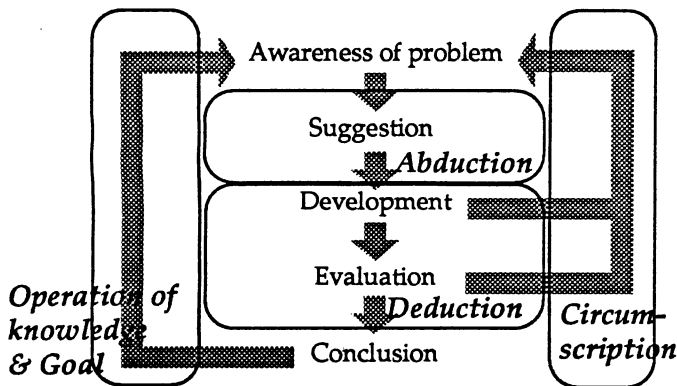


図2 設計サイクルと推論

4 設計シミュレータ

計算可能なモデルを利用して実際の設計過程を再現すると、モデルの有効性、妥当性を検証することができる (設計シミュレーション) [3]。ここでは3節で述べたモデルに基づいたシステムとして、設計シミュレータのプロトタイプを構築した。これを用いて、設計実験により得られたプロトコル・データをもとにして推論を行なわせて、その有効性をみる。

4.1. 構成

図3にその構成を示す。このシステムは主に行為レベル推論部、対象レベル推論部、知識ベースからなり、さらに対象レベル推論部は Ds 、 P 、 Ko のワークスペースと abduction、deduction (演繹)、circumscription の3つの推論部からなる。

行為レベル推論はルールベースシステムである。知識は設計方法に関する知識であり、例えば対象レベルの知識ベースの選択知識、あるいはスケジューリング知識などである。行為レベルの推論は対象レベルからその状態を受け、3節で述べたようなメタレベル操作列を結果として出す。

対象レベル推論は Ds 、 P 、 Ko の現在での状況を保持する各ワークスペースに対する、abduction、deduction、circumscription 推論である。 Ko の内容は Horn 節を基本とする論理式であり、 Ds 、 P はそれぞれアトム式の集合である。知識は知識ベースとして管理・操作される。

abductive 推論はいくつかの方法が提案されているが (例えば [7])、ここでは次のようにして実現した。

abduction とはある G と知識 K ($K \vdash G$)、および可能な仮説集合 A が与えられたとき、

$$K \cup A_1 \vdash G$$

なる $A_1 \subset A$ を見つけることであると定義することができる。これは $K \cup A$ を Prolog 定義節として G を解いたときに用いられた節として求めることができる。この方法は Poole の default reasoning [8] の方法と基本的には同じであり、 K と A に入るものが異なる (K が事実であり、 A がデフォルト) だけである。ただし、より「深い」仮説を得るために節の適用の仕方に順序 (同じリテラルにはまず K の中の節、次

にAの中の節の順に適用する)がある。ここで仮説集合A1がA2より深い仮説である ($A1 \geq A2$) とは、
 $A1 \geq A2 \Leftrightarrow A1 \cup K \vdash A2$
 である。

circumscriptionは中川ら[9]らによって示された方法に従って行なった。これは論理型言語における prioritized circumscription[10]の実現である。

このシステムは以下のように推論を進める。対象レベルの推論はDsやPやKoの内容を更新する。そして場合によっては矛盾を起こす。このような対象レベルの状態は行為レベルに報告される。行為レベルはこの状態を基に推論を行ない、対象レベルへの操作(ワークスペースへの操作または推論の実行)を指示する。このようにシステムでは対象レベルの推論と行為レベルの推論が交互に行われ、また対象レベルでは3つの推論のうち適切なものが実行される。この繰り返しにより、DsおよびPは段階的に詳細化が行われていく。

システムはSun-3/4上でAllegro Common Lisp、X11、CLX(Common Lisp X Interface)を用いてインプリメントされた。

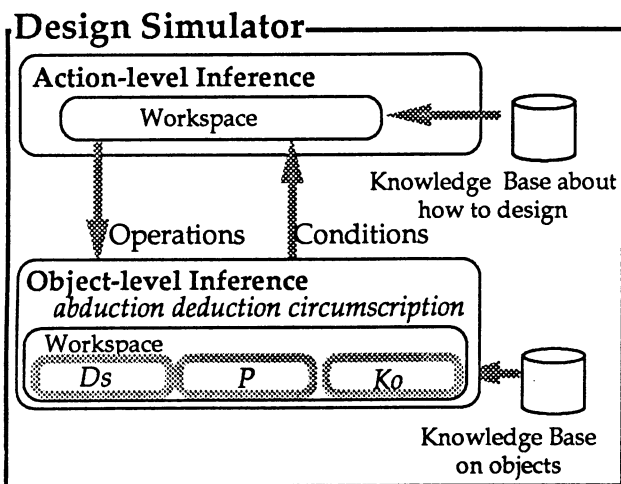


図3 設計シミュレータの基本構成

5. 設計シミュレーション

設計実験[3]から得られたプロトコルデータを利用して13の知識ベースを作り、簡単な設計過程のシミュレーションを行った。その結果、abduction 10回、演繹22回、circumscription 3回を行ない、実際のプロトコル類似した手順・結果の設計過程を作ることができた。図4にその途中の様子を示す。

List of Knowledge Bases	List of Figure	
WEIGHTING_MACHINE CAN_MEASURE SUPPORT TRANSLATE INDICATOR HAS MECHANISM IS_UPWARD IS_EASY_TO_SEE	FIGURE1 FIGURE2 FIGURE3	
List of Assumptions	List of Facts	
(TRANSLATE *wM WEIGHT) (HAS *wM INDICATOR1) (SUPPORT *wM WEIGHT) (TRANSLATE *wM 5MM 10)	(TRANSLATE *wM WEIGHT) (HAS *wM INDICATOR1) (SUPPORT *wM WEIGHT) (TRANSLATE *wM 5MM 10) (WEIGHTING_MACHINE *wM) (CAN_MEASURE *wM WEIGH)	HAS CAN_MEASURE SUPPORT TRANSLATE MECHANISM WEIGHTING_MACHINE IS_EASY_TO_SEE IS_UPWARD MECHANISM
History of used Rules		
AS ((WEIGHTING_MACHINE *wM) (CAN_MEASURE *wM WEIGHT) (SUPPORT *wM) (CAN_MEASURE *wM WEIGHT) (TRANSLATE *wM WEIGHT DISPLACEMENT))		

図4 設計シミュレータの表示

6 結論と展望

本研究では設計過程の計算可能モデルを論理的枠組みによって構築した。これは設計実験から得られた認知的モデルの多くの部分を解釈することができる。さらにこのモデルの基づいたシステムを構築することにより、その計算可能性と設計過程のモデルとしての妥当性を示した。

今後はさらにモデルとシステムの改良を行うと共に、このシステムを用いることによりインテリジェントCADの持つべき機能を検討を行う予定である。

参考文献

- [1] J.R.Dixon: On research methodology towards a scientific theory of engineering design, (AIEDAM), 1(3), (1987) pp. 145-157
- [2] 武田他: 知的CADのための設計シミュレーション, 第7回設計シンポジウム論文集, 1989, 34-36.
- [3] 武田他: プロトコル解析を用いた設計過程の分析の方法について, 認知科学学会第7回大会, (1990), (予定).
- [4] 武田他: 知的CAD開発のための設計過程の論理による定式化, 人工知能学会全国大会(第2回)論文集(1988), pp161-134
- [5] パース, C.S. (内田種臣編訳): パース著作集, 勁草書房(1986)
- [6] J. McCarthy: Circumscription - a form of non-monotonic reasoning, Artificial Intelligence 13(1980), pp.27-39.
- [7] J.J.Finger and M.R. Genesereth, RESIDUE A Deductive Approach to Design Synthesis, Technical Report Stan-CS-85-1035, Stanford University, (1985)
- [8] D. Poole: A logical framework for default reasoning, Artificial Intelligence 36, (1988), pp. 27-47.
- [9] 中川他: 論理型言語におけるCircumscription, 情報処理学会論文誌, (1987), 28(4), pp.330-338
- [10] V. Lifschitz, Computing Circumscription, Proc. 9th IJCAI, (1985), pp.121-127.