

## 多重世界を用いた設計過程の論理による形式化

18-12

Formalization of Design Processes by Logic with Multiworlds

武田 英明

吉岡 真治

富山 哲男

吉川 弘之

Hideaki TAKEDA Masaharu YOSHIOKA Tetsuo TOMIYAMA Hiroyuki YOSHIKAWA

東京大学工学部精密機械工学科

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

## ABSTRACT

This paper discusses building a logical design process model, which is a prerequisite for realizing intelligent computer-aided design systems. This model consists of abduction, deduction, circumscription and meta-level reasoning in the logical framework, and it represents human designing processes as logical processes. Furthermore, we introduce data logic to represent design objects that are gradually detailed in design processes. Data logic is modal logic with partial semantics, i.e., there are three truth values ( $t$ ,  $f$  and  $u$ ) and possible worlds. In our model, every possible world contains descriptions of the design objects at a certain moment in design processes, and an accessibility relation represents evolution of the design objects. Formulae including necessity and possibility modals can be used to represent knowledge about how to proceed design. Based on the logical model, we implemented a design simulation system by which we show the model is appropriate for representing design processes.

## 1はじめに

近年多くの設計において CAD(Computer Aided Design)が使われるようになってきたが、その多くは製図作業の支援にとどまっており、設計全体を計算機で支援するシステム(インテリジェント CAD)が必要とされている。ここではインテリジェント CAD システムの実現にあたって重要な問題である設計過程のモデリングについて論じる。システムが設計過程のモデルを用意することにより、設計者の行為を理解して、適当な支援を行なうことができると考えられる。

本研究では前報[1]で設計過程を論理的枠組みの中で形式化し、推論が可能であること示した。本報では、そこで議論をさらに発展させ、多重世界の枠組みを利用して、設計過程中での対象と対象の性質記述を管理する方法について述べる。

## 2 設計過程の論理的枠組みによるモデル

ここでは設計を abduction と演繹の繰り返しによる対象の詳細化過程としてモデル化する。すなわち、abduction により現在の対象記述を説明するより詳細な仮説を生成し、演繹によりその対象がもつ性質を導出して、要求仕様に合

致するかなどを調べる。そして要求仕様や設計の方針に満足していれば、対象記述をさらに詳細化する。

これを論理的枠組みで解釈すると、以下のよう形式化が可能である。対象に関する知識  $Ko$  とは、設計対象の記述  $Ds$  からその対象のもつ性質  $P$  を導出するものであるとする。すなわち、

$$Ds \cup Ko \vdash P \quad (1)$$

である。このとき、設計とは対象に関する知識  $Ko$  のもとで、要求仕様  $R$  が示されたとき、

$$Ds \cup Ko \vdash R \quad (2)$$

を満たす設計対象記述  $Ds$  を求めるとなる。当然、この設計対象の性質の記述  $P$  は  $R$  を含むはずである。

$$P \supseteq R \quad (3)$$

ある時点での対象の記述  $Ds_i$ 、対象の性質の記述  $P_i$ 、対象知識を  $Ko_i$ 、次の時点での対象の記述  $Ds_{i+1}$ 、性質の記述  $P_{i+1}$ 、対象知識  $Ko_{i+1}$  とする。ただし、 $Ko_{i+1} \supseteq Ko_i$  とする。

Abduction による対象の詳細化とは

$$Ds_{i+1} \cup Ko_{i+1} \vdash D_i \quad (4)$$

を満たすような  $Ds_{i+1}$  を見い出すことである。そして、

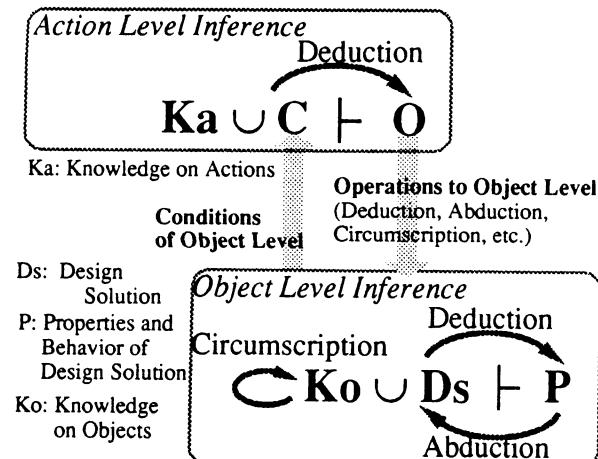


図 1: 設計過程の論理によるモデル化

演繹による対象性質の導出とは

$$P_{i+1} = \{p : Ds_{i+1} \cup Ko_{i+1} \vdash p\} \quad (5)$$

なる  $P$  を求めることである。

このようにして、順次  $Ds$  を変化させていき、その内容が全て製造が可能な情報になったとき、設計は終了する。

演繹によって矛盾が生じた場合、2通りの解釈がある。まず、そのそもその対象記述が存在し得ないものである場合であり、この場合は別の解を探す。あるいは、知識の不十分性から見かけ上の矛盾である場合であり、この場合は知識自身の変更がなされる。前者は *backtrack* により、後者は *circumscription* [2] により例外事項を計算して、論理式に付加することにより解決する。

さらに、利用する知識の変化についてはメタレベル推論という形で実現している。対象記述とその対象記述に関連する知識という関係をメタレベルの論理式として記述して、メタレベルの推論を行なうことで対象レベルで利用する知識を変化させる。今、対象レベルを  $L_o$ 、メタレベルを  $L_m$  とするとき、

$Ds \cup K \vdash_{L_o} P \leftrightarrow \vdash_{L_m} demonstrate(Ds, K, P) \quad (6)$

と定義すると、ある対象記述  $ds$  と対象レベルの論理式集合  $K_{ds}$  が関連していることを、メタレベル述語を *demonstrate* を用いて下に示すように記述する。

$$\begin{aligned} & demonstrate(Ds_i, K_i, P_i) \wedge ds \in Ds_i \\ \rightarrow & demonstrate(Ds_i, K_i \cup K_{ds}, P_i) \end{aligned} \quad (7)$$

推論の全体としては図 1 のようになる。これらの推論の詳細は [3] で議論している。

### 3 設計過程における対象の表現モデル

ここでは、段階的変化する対象記述や対象の性質記述をどのように管理するかことについて述べる。

設計での段階的に変化において各段階を設計状態と呼ぶことになると、前節で述べたように、各状態  $S_i$  において、そこでの対象記述  $Ds_i$ 、対象の性質記述  $P_i$ 、利用可能な知識  $Ko_i$  間において、

$$Ds_i \cup Ko_i \vdash P_i \quad (8)$$

の関係がある。次の状態  $S_{i+1}$  に進むと、式 (8) を満たすように新たな  $Ds_{i+1}$  を求める。Abduction の解が複数あるときは、次の状態は複数存在する。また演繹で矛盾を発見した、あるいは要求仕様が満たされなくなった場合、適切な過去の状態まで逆戻り、その状態からやり直すことがある。したがって、全体としては状態の関係は木構造になる。

そこで、このような複数の状態を様相論理の可能世界を利用して管理する。ここではとくにデータ論理 (data logic) [5] という論理体系を利用する。この論理は直感的には、様相のある部分論理 (partial logic) である。

データ論理では真理値は  $t$  (真)、 $f$  (偽)、 $u$  (未知) という3値がある。この  $u$  は直感的には「未定」という意味であり、いずれは真か偽になるというが期待される真理値である。この真理値において、 $t \sqsupseteq u$ 、 $f \sqsupseteq u$  という半順序をつけ、可能世界間の到達可能性関係はそこでのすべての基本命題において  $\sqsupseteq$  が満たされるときのみつけられる。すなわち、ある可能世界から到達可能な世界では、もとの世界で  $u$  である基本命題は  $t$  または  $f$  に変わっていてよいが、 $t$  や  $f$  であるものは変化していないということであり、直感的には「より真偽が定まった状態」である。

このデータ論理における可能世界を設計状態に対応させる。前節で述べたような推論において、 $Ds_i$  と  $Ds_{i+1}$  の論理的帰結の集合である  $P_i$  と  $P_{i+1}$  は、

$$P_{i+1} \supseteq P_i \quad (9)$$

すなわち、abduction による対象記述を詳細化する場合、性質の記述は単調増加する (これまで導かれていたことが覆ることはない)。したがって、性質の記述を命題とする可能世界を考えれば、データ論理での可能世界となる。

Abduction により新しい状態が順次できていく場合、順につながる可能世界の枝を作っていくことになる (図 2 参照)。Abduction 以外により性質の記述が非単調に変化する場合 (*circumscription* による知識の変更など)、単調変化になるような適切な位置まで可能世界の枝を戻って新しい可能世界をつくる (図 2 参照)。

### 3.2 様相を用いた知識表現

このとき様相記号を含む論理式による記述は次のような意味をもつ。

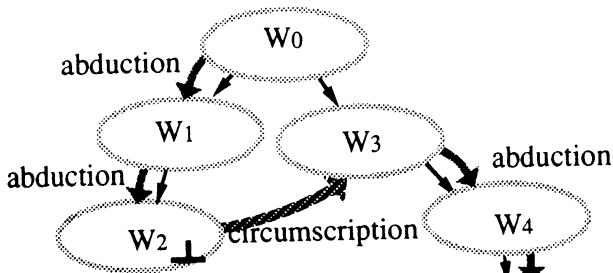


図2: 可能世界における設計の進行

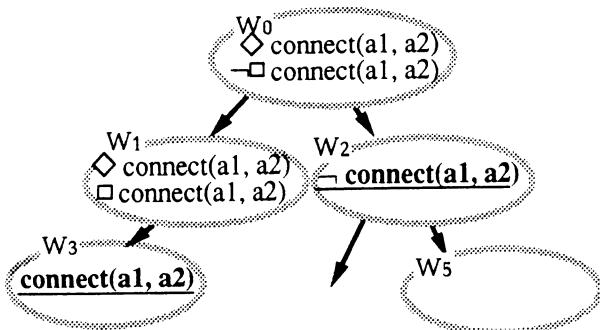


図3: データ論理における様相

この論理において、ある命題に必然記号をつけた場合、その命題の否定が到達可能な世界に存在しないときに真である（図3参照）。したがって、必然記号をつけた命題の言明(assertion)はその命題の否定の言明は許さないという意味（真である必要はない）であり、要求仕様のようにいずれ満たされなければならないことを表現する。

節形式で書いたとき、後件部が必然命題であるような式は、仕様を詳細化する知識を記述することができる。前件部も必然様相のみならば、仕様自身を設計する知識であり、そうでなければ、そこでの設計対象の状態を利用して新たな仕様をつけ加える知識である。例えば、「変速機を設計することはギアボックスを設計すればよい」という仕様詳細化知識は、

$$\square \text{transmission}(x) \leftarrow \square \text{gearbox}(x) \quad (10)$$

と記述することができる。

ある命題に可能記号をつけた場合、その命題が到達可能な世界のうちでひとつの世界で真でもあれば可能記号をつけた命題は真である（図3参照）。したがって、ひとつの解として設計されたものはそれより以前の世界ではその解を示す命題に可能記号をつけたものが真となる。

以降の段階で何をすればよいかということについての知識は可能様相を含んだ式で記述することができる。例えば、「変速機としてはギアボックスが可能である」とき、

$$\text{transmission}(x) \leftarrow \diamond \text{gearbox}(x) \quad (11)$$

と記述することができる。可能様相をつけた命題を宣言す

とを要請する。したがって、以降の設計過程ですべきことを規定している。先の例では、 $\diamond \text{gearbox}(a1)$ が宣言された場合、それ以降の世界のどこかで、 $\text{gearbox}(a1)$ が満たされなければならない。

ここでの様相は設計過程を考えたとき、意味を持つ。すなわち、設計過程でどう使うかに依存して、様相は付けられている。例えば、式(10)、式(11)はどちらも「ギアボックスは変速機である」こと、すなわち、

$$\text{transmission}(x) \leftarrow \text{gearbox}(x) \quad (12)$$

を示している。しかし、変速機やギアボックスであるということを仕様と扱うか、変速機の詳細化はいつ行なうか、といったことを考慮して、異なる式として記述される。

設計における知識とは、単に「もの」や、「もの」と「もの」の関係に関する情報だけでなく、設計でどう用いるのかも含めて知識となっている。したがって、このような様相を使った記述はその一部を表現している。

## 4 多重世界を用いた設計シミュレータ

### 4.1 システムの構成

設計過程のモデルを利用して設計過程を再現することを設計シミュレーションと呼ぶ。ここでは、以上に述べたような推論を行う設計シミュレータを作成した。このシステムは前報[1]のシステムを多重世界を扱うように拡張したものである。

システムの構成を図4に示す。Abductionによる対象記述の変化と世界間の関係は Morris[6] の方法を利用した多重世界における ATMS を作成して管理している。Abductionを形式的に行なう方法についてはいくつか提案されているが、ここでは Poole の default 推論[7]の利用により実現している。すなわち、Poole の fact が利用可能な知識であり、選ばれた default が新しい仮説である。

様相記号は可能世界の選択のときに利用することができる。すなわち、様相記号を含む式を記述することにより、その式を満たす可能世界の集合を選択することができる。なお現在のところ、様相を含む論理式は観察のときのみ可能で、知識として記述することはできない。

システムは Sun-4 上で Allegro Common Lisp、X11、CLX(Common Lisp X Interface)を用いてインプリメントを行なっている。

### 4.2 設計シミュレーションの例

設計実験[8]の設計のプロトコルの一部を利用して知識ベースを作成して、このシステムを用いて設計シミュレーションを行なった。ここで課題は体重計の試作であり、

## Design Simulator

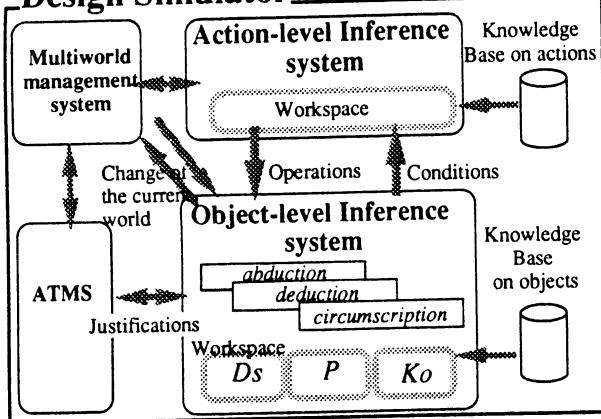


図 4: システムの構成

<b>Current</b>	<b>"(W-1)"</b>	<b>"(W-12)"</b>	<b>"(W-2)"</b>	<b>"(W-1)"</b>
Assumptions (DISPLACEMENT 5 (WEIGHT 100KG) (INDICATOR 129) (HAS SCI 129) (HAS SCI 129 132) (MANY GEARS 133) (HAS 129 133) HELICAL_GEAR_H (HAS SCI SP88) (PULL SCI SP38) (SPRING SP38)	Assumptions (DISPLACEMENT 5 (WEIGHT 100KG) (TRANSLATE SCI) (SUPPORT SCI 10) (INDICATOR 129) (HAS SCI 129) (HAS 129 132) MANY_GEAR_133 (HAS 129 133) HELICAL_GEAR_H Derived facts (SCALE SCI) (CAN_MEASURE SCI) (IS_EASY_TO_SEE) (IS_UPWARD 129) (NOT (IS_EASY_M) not expand (SUPPORT SCI 10) (TRANSLATE SCI) good property (IS_EASY_TO_SEE) bad property	Assumptions (DISPLACEMENT 5 (WEIGHT 100KG) (TRANSLATE SCI) (SUPPORT SCI 10) (CAN_MEASURE SCI) (IS_EASY_TO_SEE) (HAS SCI 129) (IS_EASY_TO_SEE) Derived facts (SCALE SCI) not expand (CAN_MEASURE SCI) (SUPPORT SCI 10) (TRANSLATE SCI) good property (IS_EASY_TO_SEE) bad property Ancestor world #W-1 #W-2 #W-12	Assumptions (DISPLACEMENT 5 (WEIGHT 100KG) (TRANSLATE SCI) (SCALE SCI) Derived facts not expand (SCALE SCI) (TRANSLATE SCI) good property bad property Ancestor world #W-1 #W-2	Assumptions (DISPLACEMENT 5 (WEIGHT 100KG) (TRANSLATE SCI) (SCALE SCI) Derived facts not expand (SCALE SCI) (TRANSLATE SCI) good property bad property Ancestor world #W-1
Derived facts (SCALE SCI) (TRANSLATE SCI) (CAN_MEASURE SCI) (SUPPORT SCI 10) (IS_IN_PROPRTY) (IS_EASY_TO_SEE) (IS_UPWARD 129) (NOT (IS_EASY_M)	Derived facts (SCALE SCI) (CAN_MEASURE SCI) (IS_EASY_TO_SEE) (IS_UPWARD 129) (NOT (IS_EASY_M) not expand (SUPPORT SCI 10) (TRANSLATE SCI) good property (IS_EASY_TO_SEE) bad property	Derived facts (SCALE SCI) not expand (CAN_MEASURE SCI) (SUPPORT SCI 10) (TRANSLATE SCI) good property (IS_EASY_TO_SEE) bad property Ancestor world #W-1 #W-2 #W-12	Derived facts not expand (SCALE SCI) (TRANSLATE SCI) good property bad property Ancestor world #W-1 #W-2	Derived facts not expand (SCALE SCI) (TRANSLATE SCI) good property bad property Ancestor world #W-1

図 6: 複数の可能世界の比較

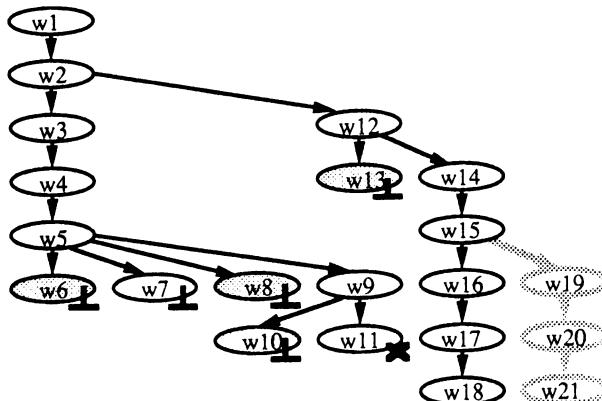


図 5: 生成された可能世界

利用したプロトコルの部分では仕様から基本構造(バネ、表示部などの部分への展開)を行なっている。このプロトコルを abduction と演繹と circumscription の繰り返しで実現したところ、図 5 に示す可能世界を生成した。この図において上は矛盾が生じたことを、×は仕様が満たされなかつたことを示している。また、灰色の世界はその世界で circumscription が行なわれたことを示す。世界 17 がこのプロトコルでの最終結果である。また、世界 20 はプロトコルにはないが、知識から導出可能な結果である。世界 17 と世界 20 の違いはバネを圧縮方向に用いるか、引っ張り方向に用いるかの違いである。

この状態で様相を含む式を利用して可能世界を比較することできる。その例を図6に示す。この例では、

$$\diamond (push(X, Y) \wedge pull(X, Z)) \quad (13)$$

なる式を満たす可能世界を表示している。これはバネを圧縮方向に使う設計と引っ張り方向に使う設計に分岐する前の状態を意味しており、世界1、2、12、14がその条件を満たしていることを示している。

5 まとめ

本研究では設計過程をの論理的な枠組みの中でモデル化を行い、そのモデルに基づくシステムを構築することで設計過程を計算機上で表現することができた。今後はモデル化をさらに展開すると共に、対象表現の理論と組み合わせることでインテリジェント CAD を構築する予定である。

参考文献

- [1] 武田英明, 河合浩之, 富山哲男, 吉川弘之. 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. 1990年度人工知能学会(第5回)論文集, pp. 583-586, 1990.
  - [2] J. McCarthy. Circumscription — a form of non-monotonic reasoning. *Artificial Intelligence*, Vol. 33, pp. 27-39, 1980.
  - [3] H. Takeda, P. Veerkamp, T. Tomiyama, and H. Yoshikawa. Modeling design processes. *AI Magazine*, Vol. 11, No. 4, pp. 37-48, 1990.
  - [4] F. Veltman. Data semantics. In J.A.G. Groenendijk, T.M.V. Janssen, and M.B.J. Stokhof, editors, *Formal methods in the study of language*, pp. 541-565. Mathematisch Centrum, Amsterdam, 1981.
  - [5] F. Landman. *Towards a Theory of Information: the Status of Partial Objects in Semantics*. Foris, Dordrecht, 1986.
  - [6] P. Morris and R. Nado. Representing actions with an assumption-based truth maintenance system. In *AAAI-86*, pp. 13-17. Philadelphia, 1986.
  - [7] D. Poole. A logical framework for default reasoning. *Artificial Intelligence*, Vol. 36, pp. 27-47, 1988.
  - [8] 武田英明, 富山哲男, 吉川弘之. インテリジェントCADのための設計過程の分析と論理による形式化. 精密工学会誌, Vol. 57, No. 6, 1991. (印刷中).