

インテリジェント CAD システムの開発の試み An Experience of Developing Intelligent CAD Systems

薛德急 Deyi Xue	桐山孝司 Takashi Kiriyama	武田英明 Hideaki Takeda
富山哲男 Tetsuo Tomiyama	吉川弘之 Hiroyuki Yoshikawa	

東京大学工学部
Faculty of Engineering, The University of Tokyo

Abstract: This paper describes the development of a new generation intelligent CAD system named IIICAD (Intelligent Integrated Interactive CAD), which is currently being conducted at the University of Tokyo. First, based on the results of design experiments, we build a cognitive design process model that can be formalized as a computable model. Second, we discuss that knowledge about the physical world plays an important role in handling design object models. We propose a frame work called metamodel mechanism that integrates models of design object using knowledge about the physical world. Based on theoretical studies about both design process and design object, the architecture of the IIICAD system is proposed. In this system, a design process is described by the computable model and a design object is organised by the metamodel mechanism.

1. はじめに

インテリジェント CAD (ICAD) の概念が提唱されて以来、様々なシステムが開発されつつある [1][2]。第一世代の ICAD システムは主にエキスパート・システム技術に基づいて構築されており、多くの場合設計問題は既知の問題空間内での探索に帰着し、ルーチン設計などの自動化が実現可能となった。第二世代の ICAD システムは第一世代のシステムの考え方に対する拘束処理やオブジェクト指向プログラミングを導入したものと考えられる。これらのシステムは、属性や寸法間に成立する制約を制約処理法によって解く。第一世代と第二世代の ICAD システムでは主に設計対象物、特に幾何情報の表現を中心として開発されたが、本来 ICAD では、単なる設計対象物に関する情報の表現だけでなく、設計を行うための「設計知識の表現と利用」も重要である。我々は ICAD による設計を「設計者の設

計意図を計算機に理解させ、計算機の上に記述されている設計知識を利用することによって、設計者の設計意図を満足する設計対象を獲得する」過程と考え、ICAD システムは「設計者」、「設計知識」及び「設計対象」を「計算機」という媒介を使って構築されるという立場をとる。そこで ICAD は設計に際して便利であるように、設計知識をより「自然に」表現し、それを設計者により「柔軟に」利用できる「環境」を提供することを目的として、第三世代の ICAD (IIICAD (Intelligent Integrated Interactive CAD) システムを提唱する [3])。

本研究では、まず理論的に設計過程と設計対象に関する研究を行ない、これに基づいて IIICAD システムの構成を提案する。IIICAD システムでは、設計過程は計算可能なモデル [1] で表現し、設計対象はメタモデル機構 [5] で管理する。またシステムにおける設計知識

は設計知識表現言語 IDDL (Integrated Data Description Language) [6] で記述する。

2. 設計過程に関する研究

2.1. 設計過程の進化モデル

一般設計学 [7] によれば、設計は機能空間から属性空間への写像と見なすことができる。すなわち、設計仕様は抽象的な機能で記述され、設計結果は具体的な属性で記述される。このような写像過程は設計対象の進化過程によって実現できる（図1）。ここで設計対象

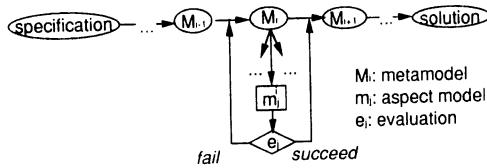


図 1: 設計過程の進化モデル

の集合をメタモデルと呼び、設計者のメンタル・モデルを反映する。設計のある段階では、設計者が設計対象の状態を評価し、また様々な操作を行い、設計対象の進化、つまり詳細化を行う。

2.2. 設計実験及び設計過程の認知的モデル

設計過程がどのように進むかを解明するために、設計実験を行った [4]。これは、まず設計問題を設計者に与えて、設計者は考えていたことを話しながら設計を行うのをビデオで記録し、そのプロトコルを分析する。その結果、設計は設計サイクルと呼ばれる問題解決過程の繰り返しとしてモデル化することができた。一つの設計サイクルは次の五つの段階からなる（図2）。

- (1) 問題提起：解決すべき問題の発見や指摘をする段階
- (2) 提案：問題提起に対して、解決の鍵になる概念を提案する段階
- (3) 展開：設計者が有している知識を用いて、提案されたものを具体化する段階
- (4) 評価：展開された解を評価基準で評価

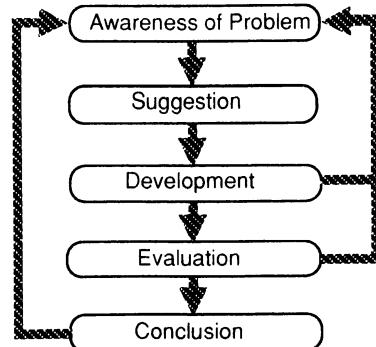


図 2: 設計サイクル

する段階

(5) 決定：評価を基にその解を採用するか、あるいはどの解をとるかを決定する段階

2.3. 計算可能な設計過程モデル

2.2節で述べた設計過程の認知的モデルは論理を用いて、計算可能なモデルとして表現することができる。まず設計知識を次の論理式で記述する：

$$D_s \cup K_0 \vdash P$$

ここで、 D_s は設計対象、 K_0 は設計対象と設計対象の性質の関係に関する知識、 P は設計対象が有する性質であり、設計仕様は P の中に記述される。この式を用いて、設計対象性質 P と知識 K_0 から、設計対象 D_s を求めるアブダクション (abduction) 及び設計対象 D_s と知識 K_0 から設計対象性質 P を求める演繹 (deduction) を行うことができる。

認知的な設計過程モデルは次の計算可能なモデルで表現される。提案段階は設計仕様 P と知識 K_0 から、設計対象候補の D_s を求めるアブダクションで表現され、展開と評価段階は提案された設計対象のほかの性質を求める演繹で表現される。また展開と評価段階において、知識の不完全性のために、矛盾が生じる場合もある。ここでは、サーカムスクリプション (circumscription) [8] を使って、知識の修正を行い、矛盾の解消を行う。

また設計過程に関する知識は二つのレベルに分けられる(図3)。行為レベルの設計知識

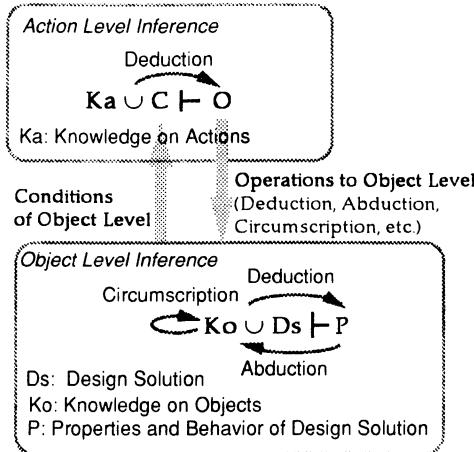


図3: 二つのレベルの設計過程に関する知識

は設計対象の状態とその時可能な操作という各設計に共通な知識を表現し、設計サイクルの五段階を実現する。対象レベルの設計知識は特定の設計対象に関する知識を表現し、具体的な設計対象の進化を行う。すなわち、設計者は行為レベルの設計知識を用いて、対象レベルの設計知識を選択し、適当な操作を行うことによって、設計対象の進化を実現する。

3. 設計対象に関する研究

3.1. 定性物理とフィジカル・フィーチャー

ICADには物理世界に関する知識が必要である。ここでは定性物理[9]を使って、物理世界に関する知識の記述を行う。

定性物理には対象世界のモデルを作る「モデリング」の侧面とそのモデルの上で挙動に関する推論を行う「推論」の侧面とがある。ここでは、Forbusの定性プロセス理論[10]を用いて、物理世界の記述を行う。この理論では、対象世界における実体は個体(individual)で、物理現象はプロセス(process)で記述される。プロセスにはそのプロセスの成立条件と

プロセスの成立によって系に与える影響が記述され、個体間の関係や個体の持つ属性の変化は全て生起するプロセスによって引き起こされると考えられる。

定性プロセス理論に基づいて、設計対象をモデリングするための単位であるフィジカル・フィーチャーを記述する。フィジカル・フィーチャーとは因果関係的に関係する物理現象と機械の構成要素の集まりとして定義される。図4にいくつかの例を示す。

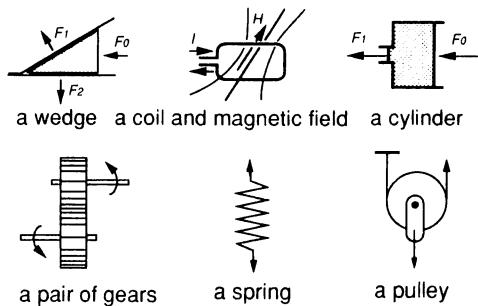


図4: フィジカル・フィーチャーの例

3.2. メタモデル機構

設計対象物は設計の進行に伴い、徐々に詳細化され、またその結果が仕様と整合するかどうかの評価をうけて進化していく(図1)。この評価のために、設計対象は注目する観点の違いによって、複数のモデルで表現される。例えば、運動伝達関係に注目する運動モデルや幾何形状に注目する幾何モデルがある。このような観点に依存する設計対象モデルのことをアスペクト・モデル(aspect model)と呼ぶ。各アスペクト・モデルの背後には背景理論と呼ばれるそのモデルに関する知識が存在する。例えば、運動モデルの背後には運動学、幾何モデルの背後には代数幾何学がある。背景理論を定性プロセス理論で記述することによって、アスペクト・モデル間の統合を背景理論レベルでの統合として実現することができる。このようなモデルを管理するメカニズム

ムはメタモデル機構と呼ぶ[5]。図5はある機構のメタモデルを示す。

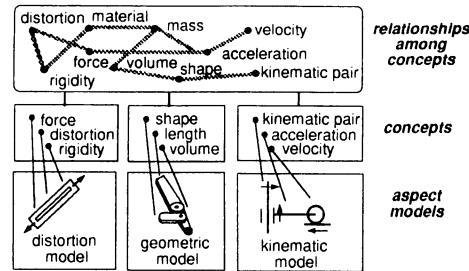


図5: メタモデル機構の例

- メタモデル機構は次の三つの要素からなる。
- (1) Model builder は設計仕様を実現するための設計対象の基本要素、基本構造を定性物理のモデリングの側面を利用して、フィジカル・フィーチャーなどを用いて構築する。構築されたモデルは一次モデルと呼ぶ。また定性物理の推論の側面を利用して、すべての可能な物理現象をエンビジョンングにより推論し、これを表現する概念ネットワークの生成を行う。この概念ネットワークは狭い意味でのメタモデルである。
 - (2) Model generator はメタモデルから、適当な背景理論を用いて、アスペクト・モデルの生成を行う。
 - (3) Consistency manager は一つのアスペクト・モデルにおける変化をメタモデルを通じて、ほかのアスペクト・モデルに伝播し、アスペクト・モデル間の整合性を保持する。
- メタモデルには従って次の3つの役割がある。
- (1) 設計者がもつ設計対象物の定性的なモデル、つまり設計者のメンタル・モデルの反映としての役割。
 - (2) 設計対象物のアスペクト・モデル間の整合性管理。
 - (3) 設計の進行に伴い進化する設計対象物

を記述するワークスペース。

4.IICAD システムの構築

4.1.IICAD システムの構成

本研究では次のような三つの要素からなるIICADシステムの構成を提案する(図6)。

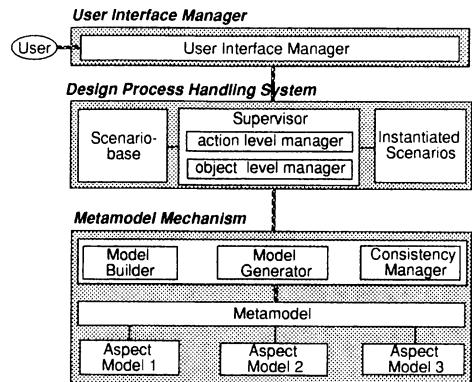


図6: IICAD システムの構成

Design process handling system は設計過程に関する知識を用いて設計過程の進化を管理する。Metamodel mechanism は設計対象に関する情報の管理を行う。User interface manager はシステムと設計者とのインターフェースを管理する。

設計過程に関する設計知識は複数のシナリオ(scenario)で記述され、二つのレベルに分けられる。行為レベルのシナリオはデータベースの状態とその時点での可能な操作との関係という各設計に共通な知識を記述し、対象レベルのシナリオは特定の設計対象に関する設計知識を記述する。スーパバイザにも二つの独立な推論システムがある。行為レベルの推論システムはデータベースの状態を監視し、必要な対象レベルのシナリオの選択、対象レベル推論の実行など可能な操作をユーザーに提供する。ユーザーはこれらの操作を選択することによって、シナリオ・ベースに格納されている汎化シナリオから適当なものを選択し、特定の設計対象についてのみ利用可能なインス

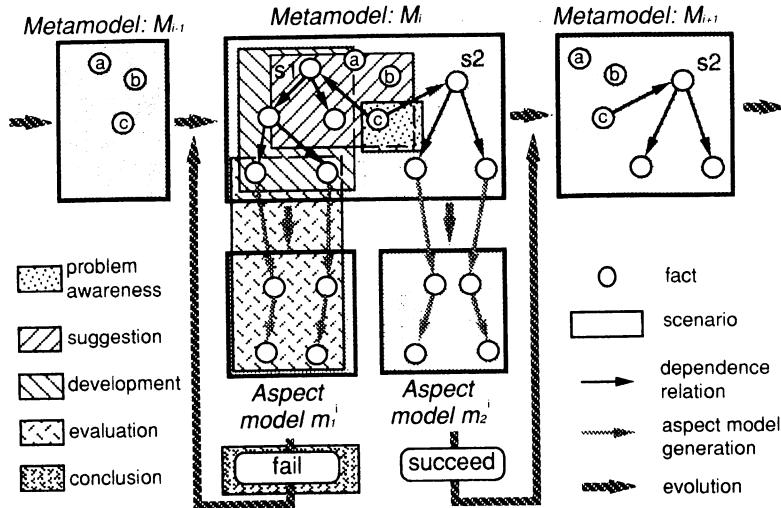


図 7 : システムにおける設計の実現

タンシェーティッド・シナリオ (instantiated scenario) を生成し、適当な推論メカニズムを用いて、設計対象の進化を行う。

設計対象は視点に依存しないメタモデルとこれから導出される視点に依存するアスペクト・モデルとで表現される。各アспект・モデルの背後には定性プロセスで記述される背景理論が存在し、各アспект・モデル間の関係を記述する。またメタモデル機構には、一次モデルの構築及び一次モデルから概念ネットワークの生成を行う model builder、メタモデルからアспект・モデルの生成を行う model generator とアспект・モデル間の整合性の保持を行う consistency manager が存在する。

IICAD を用いた設計は以下のようにして行う (図 7)。例えば、設計のある時点では、メタモデル M_{i-1} から進化したメタモデル M_i がある。まずユーザの指定によって、 c を問題として注目する。シナリオ・ベースにおいて

て、解を提案可能なシナリオが存在するので、このようなシナリオを選択し、 s_1 という解を提案する。Model builder はこれを仮定としてデータベースに付加する。同じようにシナリオ・ベースにおけるシナリオを選択し、提案の展開を行う。この提案を評価するとき、アспект・モデルが必要であるため、model generator が起動される。もしこの提案に関する評価は失敗であれば、ATMS (Assumption-based Truth Maintenance System) [11] を使って、この提案に関するすべてのデータを削除する。同じように提案 s_2 を生成し、もしこの提案に関する評価が成功であれば、メタモデルを M_{i+1} に進化させる。

4.2. 設計過程知識の表現

設計過程に関する設計知識は設計知識表現言語 IIDL[6] のシナリオとして記述される。一つのシナリオはプロダクション・ルールの集合からなる。一つのルールは条件部と結果部があり、演繹のための条件部から結果部を

求める前向き推論とアブダクションのための結果部から条件部を求める後向き推論とに使われる。またサーカムスクリプション [8] を使って、ルールの修正を行い、矛盾の解消を行うこともできる。ルールの条件部と結果部は一階述語で記述される。また行為レベルのシナリオは予約述語の `use` によるほかの行為レベルのシナリオの実行、予約述語の `select` による対象レベルシナリオの選択と予約述語の `do` による対象レベルの操作を行うこともできる。

図 8 に行為レベルのトップレベルシナリオを示す。このシナリオは次の行為レベルの設計知識を記述する：

(1) 設計仕様はいつでも入力することができる。(2) 問題提起はいつでも行うことができる。(3) 問題があれば、その問題に対して提案を行うべきである。(4) 提案があれば、展開を行うことができる。(5) 提案があれば、提案に関する評価を行うことができる。(6) 提案があれば、提案を採用するかしないかという決定を行うことができる。(7) 矛盾が生じるとき、矛盾の解消を行るべきである。(8) メタモデルが存在すれば、アスペクト・モデルの生成を行うことができる。(9) メタモデルが存在すれば、モデル間の整合性を検査することができる。等々。

次に設計過程の実現について詳しく説明する。

(1) 問題提起：設計仕様は IDEF1X のオブジェクト、オブジェクト間の関係としての fact、オブジェクトの属性によって記述される。またユーザの指定によって、ある fact を問題として注目することができる。

(2) 提案：提起した問題に対して、もしシナリオ・ベースに問題解決可能なシナリオが存在すれば、これらのシナリオを選択し、鍵となる概念の提案を行う。提案された概念は仮定となる。システムは ATMS[11] を使ってデータ間の依存関係を記述する。図 9 は提案を行うためのシナリオを示す。例えは、

topActionLevelScenario

"Top level operation"

```
BEGIN
IF TRUE THEN use(addSpecification),
  "(1) Add specification"
IF TRUE THEN use(problemAwareness),
  "(2) Try to set a problem"
IF problem(X) THEN use(suggestion),
  "(3) Try to make suggestion"
IF suggestion(X) THEN use(development),
  "(4) Try to make development"
IF suggestion(X) THEN use(evaluation),
  "(5) Try to make evaluation"
IF suggestion(X) THEN use(conclusion),
  "(6) Try to make conclusion"
IF contradiction(X)
  THEN use(contradictionResolution),
  "(7) Try to solve contradiction"
IF suggestion(X) & metamodel(X)
  THEN do(metaToAspectGeneration),
  "(8) Make aspect model from metamodel"
IF suggestion(X) & metamodel(X)
  THEN do(consistencyInspection),
  "(9) Inspect integration of aspect models"
....
```

図 8: topActionLevelScenario シナリオ

$rotationPair(r_1, r_2)$ という問題に対して、シナリオ・ベースにおいて二つのこの問題を解決する可能なシナリオが存在する（図 10）。これらのシナリオを選択し、後向き推論を使えば、 $beltGearDrive(r_1, r_2)$ 、あるいは $gearPair(r_1, r_2)$ という解候補を提案することができる。

(3) 展開：同じように、適当なシナリオを選択し、演繹推論を使って、提案された解に関して展開を行うことができる。図 10 に示される二つのシナリオは展開にも使える。また一次モデルから概念ネットワークの生成も展開として考えることができる。

(4) 評価：属性に関する拘束条件などの仕様は評価の基準として使うことができる。この段階では、提案に関するデータに対して、すべての評価基準の評価を行う。すべての評価が成功すれば、この提案に関する評価は成功となる。もし失敗の評価があれば、この提

```

suggestion
"Making suggestion to problem"

BEGIN
IF problem(X) & scenario(Y)
& canSuggestByAbduction(Y,X)
THEN select(Y)
& do(suggestionByAbduction)
& ~problem(X),
"Find suggestion by abduction"
IF problem(X) & scenario(Y)
& canSuggestByFunctionDescription(Y,X)
THEN select(Y)
& do(suggestionByFunctionDescription)
& ~problem(X),
"Find suggestion by function description"
IF problem(X) & canNotSuggested(X)
THEN use(makingNewKnowledge),
"Make new knowledge"
IF TRUE THEN succeed,
"END"
END

```

```

gearPairDescription
BEGIN
IF gearPair(X,Y)
THEN rotationPair(X,Y),
IF gearPair(X,Y)
THEN sameDirection(X,Y),
.....

```

(a) gearPairDescription scenario

```

beltGearDriveDescription
BEGIN
IF beltGearDrive(X,Y)
THEN rotationPair(X,Y),
IF beltGearDrive(X,Y)
THEN reverseDirection(X,Y),
.....

```

(b) beltGearDriveDescription scenario

図 10: 提案を行う対象レベルのシナリオ

案に関する評価は失敗となる。また情報量の不足によって、評価ができない場合、演繹やアスペクト・モデルの生成による必要な情報の生成を行う場合もある。

(5) 決定：もし提案に関する評価の結果が失敗すれば、この提案に関するすべてのデータを ATMS で削除する。また未解決の問題を新しい問題としてもう一度注目し、新しい問題解決サイクルに入る必要もある。提案に関する評価の結果が成功すれば、様相論理などを使って、複数の提案に適当なものを選択することもできる。

(6) 矛盾の解消：展開と評価段階では、矛盾が生じる場合がある。本研究ではこれは知識の不完全性によって発生すると考え、サーラムスクリプションを使って、不完全な知識を例外を含むようにルールの修正を行う。これによって、要求仕様が導けなくなる場合、新たな問題の提起となる。4.5 節ではサーラムスクリプションによる矛盾の解消を例によつて説明する。

4.3. 設計対象の表現

HICAD システムにおいて設計対象はオブジェクト、述語で表現されるオブジェクト間の関係の fact、オブジェクトの属性とオブジェクト間の依存関係によって記述される。オブジェクトはプロトタイプによって生成され、設計対象、物理現象などを記述する。プロトタイプは階層的な構造で構築され、定性的な情報と定量的な情報が記述される。プロトタイプの定性的な情報は定性プロセス理論に基づいて記述される。すなわち、一つのプロトタイプは一つの個体、あるいはプロセスに相当し、中には自分自身を成立させるための個体、個体間関係などの前提条件と自分自身の成立によって引き起こす物理現象が記述される。プロトタイプの定量的な情報はオブジェクトの属性や属性間の量的関係である関数として記述する。設計対象は多重世界機構におけるワールド(world)内に保存される。一つのワールドはデータベースの一部であり、ある設計段階での設計対象に関する情報を保存する。また多重世界機構はいくつかの提案が

あるとき、様相論理による適当な提案の選択にも使われる。

4.4. システムのインプリメンテーション

IICAD システムはオブジェクト指向言語 Smalltalk-80 上に開発している。システムには行為レベルシナリオ作成環境、対象レベルシナリオ作成環境、プロトタイプ作成環境、フィジカル・フィーチャー作成環境、設計対象データベース管理環境、設計実行環境など様々なインターフェース環境がある。IICAD による設計は設計実行環境の Design browser を用いて行う。行為レベルの推論システムは行為レベルのシナリオを使って、可能な操作をメニューの形でユーザーに提供する。ユーザーはこれらの操作を選択することによって設計対象の進化を行う。またこのブラウザにおいて設計対象のデータも表示できる。

4.5 設計の実行例

この節では、例によって IICAD における設計の実現を説明する。まず設計問題は次のように記述される（図 11）。ここでは r_1 と r_2

specifications

```
clockwise(r1)
counterclockwise(r2)
rotationPair(r1,r2)
=(n[r1],300)
<(distance[r1,r2],150)
>(n[r2],500)
```

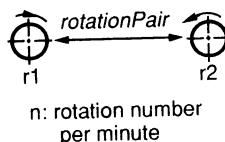


図 11: 設計仕様の記述

は二つのオブジェクトで、別々時計回り方向と反時計回り方向の回転運動を行う。またこの機構に関する拘束条件も設計仕様に記述される。4.2 節で述べたようにこの問題を解決する提案は二つある（図 12）。ここではまず $beltGearDrive(r_1, r_2)$ という提案を生成する。しかし、展開を行うとき、 $sameDirection(r_1, r_2)$ と $\sim sameDirection(r_1, r_2)$ という矛盾が生じるため、サーラムスクリプションを使って、ルールの修正を行う（図 13）。しかし、シ

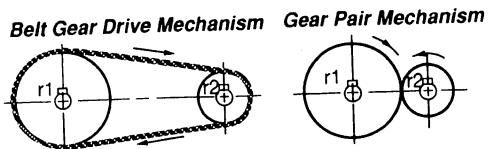


図 12: 二つの可能な提案

```
IF beltGearDrive(X,Y)
  THEN sameDirection(X,Y), ... ... (1)
IF clockwise(X) & counterclockwise(Y)
  THEN ~sameDirection(X,Y), ... ... (2)
```

(a) Two Rules with a Contradiction

```
IF beltGearDrive(X,Y) & ~ab1
  THEN sameDirection(X,Y), ... ... (1')
IF clockwise(X) & counterclockwise(Y) & ~ab2
  THEN ~sameDirection(X,Y), ... ... (2')
```

(b) Adding Abnormal Predicates to Rules

```
(priority: ab2 > ab1)
  ab1 = clockwise(X) & counterclockwise(Y)
  ab2 = false
```

(c) Solutions of Circumscription

```
IF beltGearDrive(X,Y)
  & ~(clockwise(X) & counterclockwise(Y))
  THEN sameDirection(X,Y), ... ... (1")
IF clockwise(X) & counterclockwise(Y)
  THEN ~sameDirection(X,Y), ... ... (2")
```

(d) Contradiction Resolution of Circumscription

図 13: ルールにおける矛盾の解消

```
...
=(n[r1],300)
=(n[r2],600)
=(z[r1],80)
=(z[r2],40)
=(i[r1],2)           n: rotation number
=(i[r2],0.5)         per minute
=(d[r1],80)          z: teeth number
=(d[r2],80)          i: ratio of n
=(m[r1],2)           d: nominal diameter
=(m[r2],2)           m: module
...
...
```

図 14: 設計対象データベース

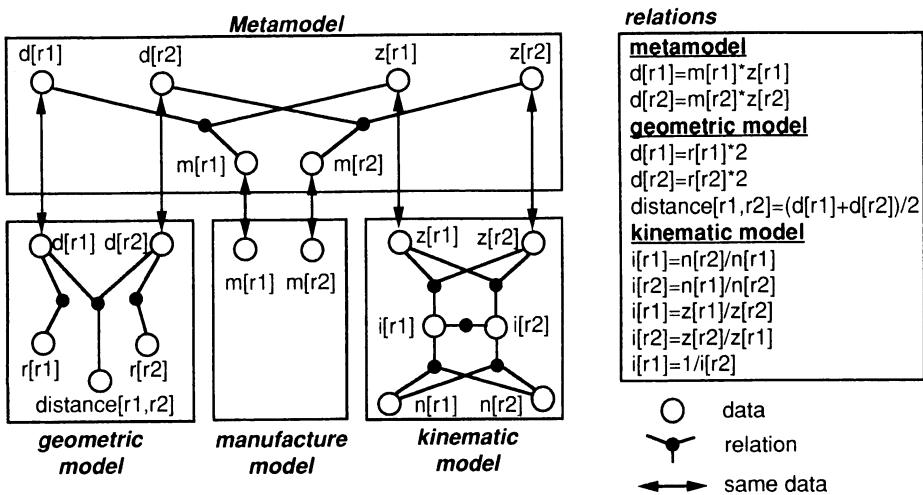


図 15：ギア・ペアのメタモデル機構

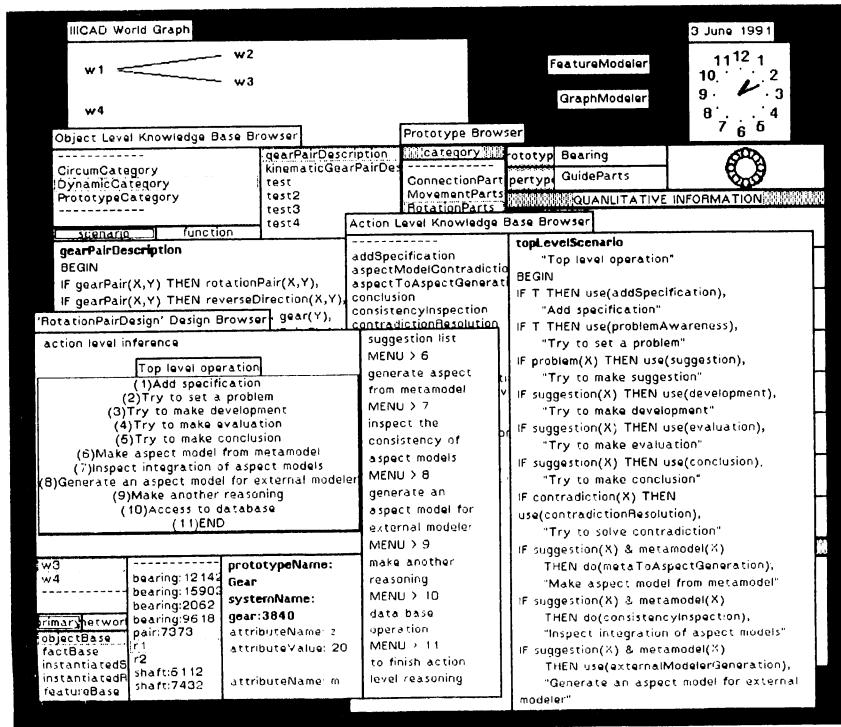


図 16：III CAD システムのハートセンター

ステムにはこの修正されたルールが表現する新しい機構に関する知識が存在しないので、この提案に関するデータを ATMS で削除する。そこで次に $gearPair(r_1, r_2)$ を提案する。評価を行うとき、運動関係と幾何関係に関する情報が必要であるため、この二つのアスペクト・モデルが生成される（図 14、図 15）。しかし、メタモデルにおけるオブジェクト r_1 には $d[r_1] = m[r_1] * z[r_1]$ という関係があるので、二つのアスペクト・モデルにおける矛盾が発見される。ユーザがアスペクト・モデル優先順位を指定することによって、 r_1 の幾何モデルにおけるデータ $d[r_1]$ を 160 に修正する。IICAD システムにおけるこの設計の実行例を図 16 に示す。

5. 結論

本研究では設計過程と設計対象に関する研究に基づいて、IICAD システムの構成を提案した。このシステムにおいては

(1) 計算可能なモデルは設計対象の基本要素、基本構造の決定など設計対象物の進化を基本とする設計過程を計算機上に実現するために有効である。

(2) メタモデル機構はモデル間の変換、モデル間整合性の保持など設計対象物に関する情報を管理するために有効である。

IICAD システムの構成は設計過程と設計対象物の管理を容易に統合的に行うこと可能にした。現在はシステムの開発を引き続き行なっているほか、インテリジェント・インターフェース・システムの開発、設計知識の自動獲得メカニズムの開発、大規模設計知識ベースの構築などに関する研究を進めている。

参考文献

- [1] 古川弘之・富山哲男（編著）：インテリジェント CAD（上） 理念とパラダイム、朝倉書店、(1989).
- [2] 古川弘之・富山哲男（編著）：インテリジェント CAD（下） テクノロジーと展望、朝倉書店、(1991).

- [3] T. Tomiyama and P.J.W. ten Hagen, "The Concept of Intelligent Integrated Interactive CAD Systems", CWI Report No. CS R8717, Centre for Mathematics and Computer Science, Amsterdam, 1987.
- [4] H. Takeda, P.J. Veerkamp, T. Tomiyama, and H. Yoshikawa, "Modeling Design Processes", in *AI Magazine*, 11(4), 1990, pp. 37-48.
- [5] 桐山孝司：定性物理を用いる設計対象表現論、博士論文、東京大学工学部(1990).
- [6] T. Tomiyama, D. Xue, and Y. Ishida, "An Experience with Developing a Design Knowledge Representation Language", in *Intelligent CAD Systems III: Practical Experience and Evaluation*, P.J.W. ten Hagen and P.J. Veerkamp (eds.), Springer Verlag, Berlin, Forthcoming.
- [7] 富山哲男、吉川弘之：一般設計学の展開（第 1 報）—概念空間のコンパクト化—、精密機械、51(4), 1985, pp. 809-815.
- [8] J. McCarthy, "Circumscription - A Form of Non-Monotonic Reasoning", *Artificial Intelligence*, 13, 1980, pp. 27-39.
- [9] D.G. Bobrow (ed): *Qualitative Reasoning about Physical Systems*, The MIT Press, Cambridge, MA, USA, (1985).
- [10] K.D. Forbus, "Qualitative Process Theory", *Artificial Intelligence*, 24(3), 1984, pp. 85-168.
- [11] J. de Kleer, "An Assumption-based TMS", *Artificial Intelligence*, 28, 1986, pp. 127-162.