

知識ベース技術のCADへの応用

| | |
|------|--------|
| 東大.工 | 韓 圭 東 |
| 東大.工 | 大須賀 節雄 |
| 東大.工 | 山内 平行 |

1. はじめに

設計のプロセスにコンピュータ技術を導入し、作業の効率向上を目差すCAD(Computer Aided Design)は、航空機、自動車等の巨大産業分野では確かに効果を上げている[文献1]。しかしながら創造的設計の支援を目的とするCADの本来の広い意味での発展は未だ不十分であり、CADesignがCADrawing, 又はCADraftingのように、単なる製図支援という狭い意味で理解され、使われているのも現実である[文献2,3]。20年位の歴史を持つCADの発展の遅れと意味の縮退に関しては種々の原因が指摘されているが[文献4]、一言でいえば従来のコンピュータ技術—ハード, ソフト両方共が、極めて複雑な知的活動である設計作業、特にトータル設計作業に対処できるまでには開発されていないからである。言い換えればCAD技術を向上させるためには新しい情報処理技術の開発が必要である。最近、高度に知的な活動であり、高度に専門知識を要する設計作業の性質と、計算機に直接に知識を使わせて人間の知的活動を支援させることを目的とする知識工学の性質から、CADへの知識工学の結びつきについて色々議論されている[文献5]。我々も知識工学のCADへの導入は非常に有望だと思う。無論知識工学の実態が未だはつきり理解されていない現在、どのようにこの両技術を結びつけるかという結論を出すには早い。我々はその方法の一つとしては、計算機内に設計対象のモデルを構築して、モデルからの情報抽出、それらの評価、評価結果に基づいてのモデルの修正等をすべて計算機にやらせることと考え、そのために知識ベースシステムKAUSを利用して、直接に計算機内に3次元object・モデルを構築する実験を行っている。

我々の構築しようとするモデルは、当面は機械や建築等を対象とし、取り合えず多面体に限定する。

2. 設計とモデル

設計とは、ある目的を持った対象を造りあげることである。そのための実際上の手順として、まず、要求された性能を持ちそうな対象の形状や構造を考案し、それが所要の性能を有しているか否かをチェックする。もし性能要求を満たさなければ形状や構造を修正する。このプロセスを繰り返し、対象が与えられた要求を満たしていると判定できればそれを更に詳細化して行く。上記の過程で対象についての人のイメージ—形状、構造、材料、性能等に関する情報の集りが設計対象のモデルである。もっと一般的に、モデルとは、それを用いて人が原問題に

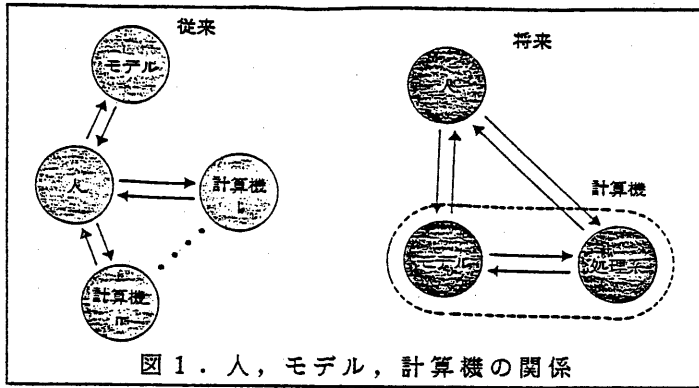


図1. 人, モデル, 計算機の関係

ついて関心を持つ特性について解を与えるだけの十分な情報を含む, 抽象化され, 且つ体系化された情報群であるともいえる.

設計とは, モデルを構築するプロセスであり, 目標に達するまでモデルの修正や性能チェックを繰り返すプロセスで, す

べてがモデルを中心に行われる. 今までのCAD技術では, モデルは図面又は他の形式で与えられ, 人間がモデルから必要な情報を抽出し, 又それを計算機言語化してコンピュータに処理させる. そして, その結果によってモデルを処理するのである. 人とモデルと計算機とのこのような関係は図1のように表現できる. このような状態では人の作業が繁雑になるし, 計算機の機能も十分に発揮し難いため作業の効率が低いのは言うまでもない.

設計の全プロセスにおいてより多くの繁雑な仕事を計算機に渡し, 計算機とモデルの関係をより密接にさせる方法の一つはモデルを直接に計算機内に構築することである. そうなると, 今までは, 人間がモデルから抽出して計算機に入力した情報を, 計算機が直接にモデルから抽出して評価できるようになり, 作業効果の向上が期待できる [文献6].

3. KAUSと知識の表現

KAUS(Knowledge Acquisition and Utilization System)は, 大須賀研究室で今開発中である知識・獲得・利用システムで, 集合-要素関係, 集合-集合関係, 一階述語論理を改造した多層論理等が導入され, 構造記述手段, 属性記述手段等モデル構築に必要な条件が基本的に備えられている. KAUSを3次元モデル構築に利用するためには, まずその中に空間解析幾何学の知識の入力が必要である. KAUSの中では, 知識を表現するために一階述語論理を改造した多層論理というものを使って

```
[APa/*all][APb/*all][APc/*all]
[AS1/str][AS2/str][AS/str]
(! (po3_plane Pa Pb Pc S)
  ~ (ne Pa Pb)
  ~ (popoline Pa Pb S1 S2)
  ~ (po_line_plane Pc S1 S2 S)).
```

(a). an example of predicate definition

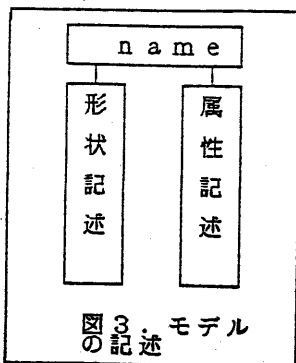
```
[ES1?/str][ES2?/str]
(popoline #p1 #p2 S1 S2)?
S1=z-90 S2=y-100
YES.
(print #p1)?
(-p1:(44.000 100.000 90.000))
YES.
(print #p2)?
(-p2:(150.000 100.000 90.000))
YES.
[ES?/str]
(distance_po_plane S #p1 "x+y+z-77")?
S=90.641
YES.
```

(b). examples of KAUS's answer

図2. KAUSの知識表現例

いる。知識の入力というのはKAUSの中で述語を定義するという意味である。

KAUSの述語は直接にプログラム言語C (KAUSもC言語でコーディングされている)でプログラミングしたり,又はすでに定義されている述語を使って多層論理の論理式で定義する。算術の四則演算等常用のものは元々KAUSの中に用意されている。直接C言語で定義された術語はPTA(Procedural Type Atom)と呼ぶ。手続型述語(PTA)は,連続量の処理に便利で,実行が速い等の利点があり,非手続型述語は定義が簡単等の利点がある。図2の(a)は述語の定義例であるが,その論理式の中で, popoline, npo_on_line, po_line_plane, neはPTAで,それらによって定義されている po3_planeは非手続型述語である。ここで, po3_planeは三つの点Pa, Pb, Pcを通る平面の方程式Sを求める述語である。論理式の意味は, "点PaとPbが等しくなく, Pa, Pbを通る直線の方程式がS1, S2(直線を二つの平面の交わりで表す)で, S1, S2が点Pcを通らなければ, 点Pcと直線S1, S2が決める平面Sは三つの点Pa, Pb, Pcが決める平面である。"である。論理式の初めに '[' と ']' で囲まれている部分は変数のdomain宣言で, Aは全称量詞でEは存在量詞である。例えば [APa/forall]は, 有らゆるPaが集合callの要素であるということの意味している。変数のdomainをこのように具体的に指定するのは一階述語論理と異なる点である。今KAUSの中には空間解析幾何知識の一部が入力されていて, それを使ってKAUSは, 3次元空間内の点や, 直線や, 平面の間の色々な質問に正確に答えられるようになっている。図2の(b)は, その簡単な例であるが, ここでは二つの点#p1, #p2を通る直線の方程式と, 点#p1から平面 $x+y+z-77=0$ (平面の方程式で "=0" 部分は省略して表



わす)までの距離を答えている。

3. モデルの構造とその記述

モデルはKAUSの中でtree構造で記述する。モデルの記述は形状記述に止まらず, 図3のように属性情報も含まれていて, いわゆるobjectモデルである[文献6]。モデルの形状情報にはモデルに対応する実物のtopological表現とgeometrical表現との両方があるが, 物体の形状を正確に表現するためにはこの二つのいずれも不可欠である。KAUSの構造定義コマンドはモデルのtopological構造を定義するのに非常に有力である。それを利

```

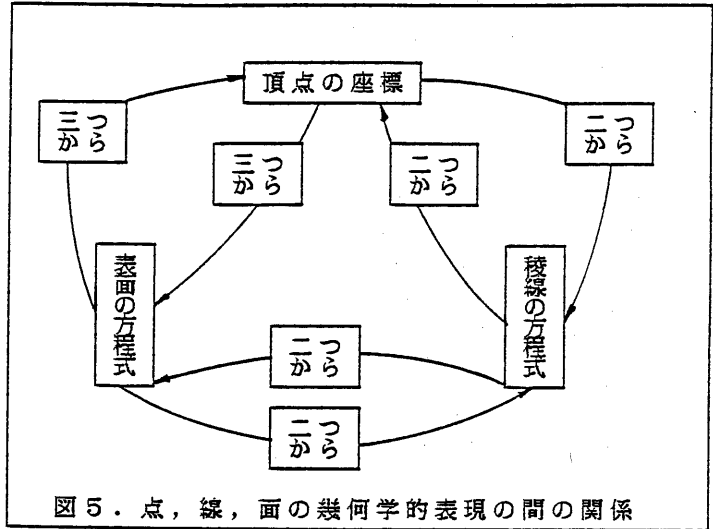
(cube :
  (surfi :
    (ed1 : pa pb 11 )
    (ed2 : pb pc 12 )
    (ed3 : pc pa 13 )
    (ed4 : pa pb 14 )sf1a )
  (surf2 :
    (ed5 : pb pc 12 )
    (ed6 : pb pb 119 )
    (ed7 : pb pb 14 )
    (ed8 : pc pb 111 )sf2a )
  (surf3 :
    (ed9 : pb pb 14 )
    (ed10 : pb pb 13 )
    (ed11 : pb pb 18 )
    (ed12 : pb pb 17 )sf3a )
  (surf4 :
    (ed13 : pb pa 14 )
    (ed14 : pb pb 112 )
    (ed15 : pb pb 18 )
    (ed16 : pb pa 17 )sf4a )
  (surf5 :
    (ed17 : pb pa 17 )
    (ed18 : pb pb 13 )
    (ed19 : pb pb 110 )
    (ed20 : pa pb 11 )sf5a )
  (surf6 :
    (ed21 : pb pb 13 )
    (ed22 : pc pb 111 )
    (ed23 : pb pb 17 )
    (ed24 : pb pb 112 )sf6a )sf6a )
  YES.

```

図4. 立体の内面記述

valはcubeの体積を, surf_i(1 ≤ i ≤ 6)は表面を, ed_i(1 ≤ i ≤ 24)は辺線を, liはed_iの長さを, sf_iはsurf_iの面積をそれぞれ表わす。

用して例えば多面体の場合，多面体は表面の集合で，表面は稜線の集合で，稜線は端点の集合でそれぞれ定義できる．図4はその例である．モデルのgeometrical表現においては，点は3次元座標で，線と面はそれぞれそれらの3次元空間内の方程式で表現できる．そして多面体の三つの幾何学表現の間には図5に示されているように，表面の方程式から稜線の方程式



と頂点の座標を求めたり，頂点の座標から表面の方程式と稜線の方程式を求めたり，稜線の方程式から頂点の座標と表面の方程式を求めたりすることができる関係があり，一種だけ使って十分である．我々は，多面体の場合は頂点の座標だけを記述して稜線とか表面の方程式が必要な場合は頂点，稜線，表面の間の関係に関する知識を使って求めるようにしている．

形状モデルの上述の記述はいわゆるソリッドモデルであるが，ソリッドモデルは設計でよく使う三面図，断面図等の作成とか，モデルの変換とか，干渉チェックとか，属性評価等に要する情報が含まれている等の利点がある．

4. モデルの変換

複雑な物は直接に定義するより単純な物で生成するのが便利である．従って，複雑なモデルを構築するためにモデルの変換は不可欠である．

変換にはモデルのtopologica構造は変化させない線形変換と，モデルのtopological構造も変化させる非線形変換と2種類がある．線形変換はただモデルの各頂点の座標に対する算術演算で割合簡単である．線形変換は次のようなものが出てきている．

- 線型変換：
1. モデルの拡大（縮小）；
 2. モデルの移動；
 3. モデルの回転；
 4. モデルの投影．

図6（次のページ）は回転して移動したモデルのディスプレイを取った写真である．変換を通じて簡単なものから複雑なもの得られる条件はモデルが連続的に変換できることである．即ち，変換操作がモデルの構造の正確性を保つことである．非線型変換の場合これは手数がかかることである．できている非線型変換は次のようなものである．

- 非線型変換：
1. モデルのUNION（二つのものを一体にする）；

2. モデルのDIFFERENCE (あるものの一部を除く) ;
 3. モデルのCUT (平面で切られ、二つに分れた一方を選ぶ) ; 図8は非線型変換の例の写真である。次はモデルのCUT変換に使う知識の概略である。

以下ではモデルをMで、Mの表面集合をM_sursetで、Mの稜線集合をM_edsetで、Mの頂点集合をM_versetでそれぞれ表わす。そしてMが平面Pに切られ、

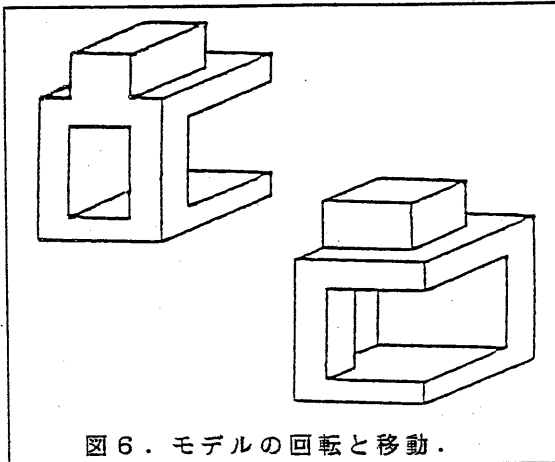


図6. モデルの回転と移動.

二つに分けられた時、Mの頂点Vを含む部分を求めると仮定する。即ち、モデルM、平面P、Mのある頂点Vを引き数とする述語cutを定義する。

C1 : MとPが交わってこそCUTができる。

C2 : M_sursetの要素毎についてCUT操作が必要である。

C3 : 新しい表面を作るべきである。

これを量詞だけは省略してKAUSの

論理式で表わすと

$(\exists (cut \ M \ P \ V)$

$\sim C1 \ \sim C2 \ \sim C3)$. になる。これは論理式

$C1 \wedge C2 \wedge C3 \rightarrow cut$

を表わす形式で、C1,C2,C3の内どれが Falseでもcutは Falseになる。

以上はトップ・レベルの知識であるが、次はそれらの詳しい解釈で、C1.JはC1の解釈である。

C1. 1 : M_versetのすべての要素が P の同じ側にあるとMとPは交じあわない。

C2. 1 : 表面全体がPに対してVの反対側にあるその表面はM_sursetからdeleteする。

C2. 2 : 表面のすべての要素についてCUT操作を行なう。

C2. 3 : C2. 2で新しくできた頂点で新しい稜線を造り、それをM_edsetの要素と、処理中の表面の要素にする。

C3. 1 : C2で新しくできた稜線全体を要素にする表面を造り、M_sursetの新しい要素にする。

次は上のレベルの知識のもっと詳しい解釈でC1.J.KはC1.Jに関する解釈である

C2. 2. 1 : 稜線全体がPに対してVの反対側にあるとその稜線は所属している表面からと、M_edsetからdeleteする。

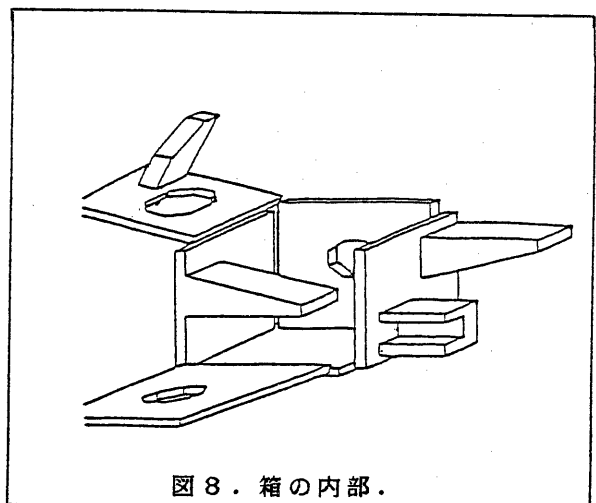
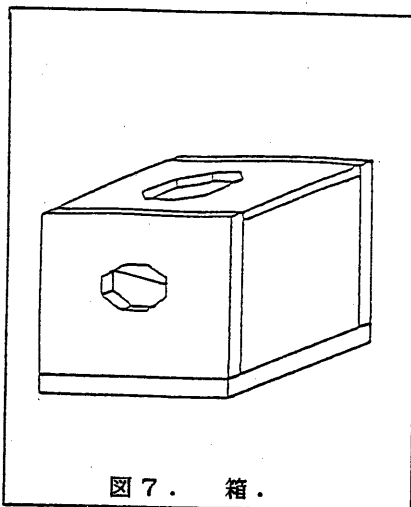
C 2 . 2 . 2 : P に対して V と同じ側にある稜線については何もしない

C 2 . 2 . 3 : P と交わる稜線については交叉点を求め、稜線の二つの頂点のうち、P に対して V の反対側にある頂点と、求められた交叉点とを交換する。

平面と稜線との関係（交わるか否か）チェックや、交叉点の求めや、集合の要素の取り扱い等に使う述語は PTA として定義されているが、論理式での表現は省略する。ここで示されているように、知識は何回にもわけて PTA で解釈できるまで段々詳しく解釈していつている。

5. モデルの組立

機械等の製品は普通多種の部品でできている。そしてそれらは普通異なる場所で異なる人によって別々に設計され、製造されて組み立てられる。それで簡単な部品から複雑な部品ができ、複雑な部品からはもっと複雑な製品ができる。もし情報レベルで組立のシミュレーションができ、設計上の錯誤が未然に防止できるとこれはとても意義がある事である。我々のモデルの組立は部品の組立に対応するものである。従来のソリッドモデリングは、モデルが主には単独加工できる基本部品を表現するもので、組合せでできた複雑な製品の内部の構造まで表現するのは難しい。我々は、モデルの属性の中に基本部品は単体、組合せでできたものは複体とそれぞれ標記して、基本部品だけでなく組合せでできたものも取り扱う予定である。図 7 は複体の例であるが六個の単体で組み合わされている一つの箱である。単体もほとんどは union とか、difference の演算結果得られたものである。図 8 は箱を開いてみた内部の様子であるがかなり複雑である。もしこの箱が普段は図 7 のように置かれるが、時には図 8 のように壁を廻す必要があると仮定すると設計時に必ず突出部分の間の衝突を考えねばならない。これは 2 次元 CAD でチェックし難い問題であるがここではごく簡単に解決できる。



6. おわりに

従来のCADは、設計のプロセスの流れの中で、従来の情報処理技術で計算機化できる部分（例えば図形処理）はすでに実用化、商用化時代を迎えようとしているが、残りの部分は未だあまり開発されていない。しかも、開発されている部分も分野によって異なる方法を取っているのが実情である。このような状況を改善し、CAD技術を一段と向上させる期待が最近は知識工学に集中されつつある。今まで我々の実験はソリッド・モデリングに過ぎないし、簡単なモデル処理に過ぎないが、このモデルは計算機の中に構築されていることと、知識工学的手段で実現されたことにその意義があると思う。これからは、知識表現で与えられた要求に合うモデルの構築支援もしくは局所ときには自動合成を行って、従来の方法では実現しにくいことが、知識工学の手段では可能になることを示してみたい。

参 考 文 献

- [1].大須賀節雄,CAD/CAM技術の現状と将来,CSK技術通信.1983,Vol.3-3,No.6.
- [2].池田 嘉彦,CAD/CAMユーザーのための形状モデリング入門,PIXEL,83.7-8.No.13.
- [3].中塚久世,パーソナルCAD,PIXEL.84.4.No.19.
- [4].長井俊朗,期待されるオート・デジタイザ・システムの応用,
- [5].松島克守,知識工学とCAD/CAM-エキスパート・システムによるプロセス・プランニング,PIXEL.84.2.No.17.
- [6].A Baer,C Eastman and M Henrion,Geometric modelling:a survey,Computer Aided Design,Vol.11.number 5,september 1979.