

人工知能の技術問題への応用

大須賀 節 雄*

Application of AI to Engineering Problems

by

Setsuo OHSUGA

*Research Center for Advanced Science and Technology
The University of Tokyo*

ABSTRACT

Computer has been developed and shown its effectiveness as a support tool to solve deterministic problem, which accepts computing procedures and numerical quantities written in advance and processes these rapidly. There are some problems, however, that are not suited to be solved in this framework of computer. A typical example of such types of problems is design problem. AI (Artificial Intelligence) is expected to realize a new method of information processing and expand the scope of applications over the current computers involving non-deterministic problems such as design. Recently Expert Systems have been developed as an implementation of AI concept. But these systems are not yet powerful enough for being applied to the problems in the real world. In this paper, the author analyzes the reason why current Expert Systems are not powerful enough and propose a new framework of AI systems. A system named KAUS developed by the author's group is presented with some applications to design problems.

1. はじめに

現行コンピュータは、数理的表現を機械化し、前もって与えられた処理手順を高速に実行することによって、決定型の問題、すなわち入力に対して解が一意に定まる型の問題の解決支援手段として効果を発揮してきた。例えば構造物の応力計算や、複雑な物体回りの流体の流れの解析、物体内での温度分布などは、大量の繰り返し計算を主とし、今日のコンピュータが最も得意とする分野である。これらの分野では、現在でも FORTRAN のような数値計算向き言語でプログラムを書くことがコンピュータの利用法と考えている研究者も

多い。しかしその機構上、現行コンピュータでは扱い難い問題が多数存在する。このような問題の代表的な例が創造的な問題、例えば設計問題である。人工知能から発した知識処理は数理的表現のみでなく論理的表現の処理を可能にすることによって、処理対象としての問題表現ばかりでなく問題解決過程あるいは問題解決の方法論を表現することができ、結果的に設計・開発を含む多様な問題に対して自律的問題解決システムの実現を可能にする。このような目的でエキスパートシステムが数多く開発されたが、未だ十分に目的を達してはいない。エキスパートシステムは幾つかの欠点を持つが、そのうち、

(1) これまで蓄積されてきた問題解決の方法論を容易に組み込めない、

* 東京大学先端科学技術研究センター

(2) 創造的活動の支援には必ずしも適さない、
 (2) 大規模問題や実時間問題への配慮の欠如、
 は実用性から特に重要である。この欠点を克服することによって次世代の人工知能システム概念が確立されねばならない。

以下では筆者らが開発してきた知識処理システムKAUS (Knowledge Acquisition and Utilization System)を引用しながら人工知能の技術問題への応用に関する諸問題を論じる。KAUSは設計・開発を含む汎用の問題解決支援知識処理システムであり、その基本的方針は、あらゆる型の問題解決を支援することができることと同時に、人間の持つ柔軟性、適応性、成長性など、問題解決に際して重要な特性を保持すること、大規模問題を扱うことができることなど、従来のシステムでは必ずしも充分でなかった特徴を考慮している。

2. 知識処理システムの設計目標

如何なる方式であれ、コンピュータは必要な情報が明示的に表現できる時にのみ用いることができるが、設計のような創造的な問題解決には処理過程の一部分をコンピュータ化するという方式では効果が薄く、作業全体を機械化の範囲に含めることが重要である。それには設計の過程そのものをコンピュータ内に明示的に表現することが前提であり、さらにそのためにまず設計の過程を明確にせねばならない。今日、最も普遍的な問題解決能力を保証するのは科学的方法論であり、これをコンピュータ内で忠実に実現することを試みる。

科学的方法論は、

1. 状況を明示的に記述する、
 2. 状況に適用される一般的あるいは個別的なルールを準備する、
 3. 状況にルールを適用して結果を導く、
- という基本的な方法で問題解決を図る。これを具体的に示すと図1の様になる。ここで状況を表すものがモデルであり、これはモデル中心の方法である。これまでのエキスパートシステムはこの開発目標が明確でなかったため、本来、明示化が可能であり、処理可能である筈の領域の僅かな一部を扱うに過ぎなかった。KAUSは科学的方法論を忠実に実現することを目指して、それに必要な技術を開発してきた。

3. 知識処理の方式

新しいものを一度で創造することは困難であり、通常は仮のモデルから出発して試行錯誤によって目指すものに到達しようとする。知識処理の目標は図1のプロセスを明示的に表現することである。もしこれが実現したなら、設計の方式は今後大きく変化するであろう。これを示したのが図2である。図2(a)は従来の設計のあり方で、計画段階から始まって設計プロセスの基本部分は全て人間の頭の中に置かれている。人間は細部にいたるまで設計のプロセスに留意しながら設計を進める必要があるが、この中でコンピュータその他の機械的手段が使えるものがあれば、部分的にそれを用いる。この方式では設計におけるコンピュータの

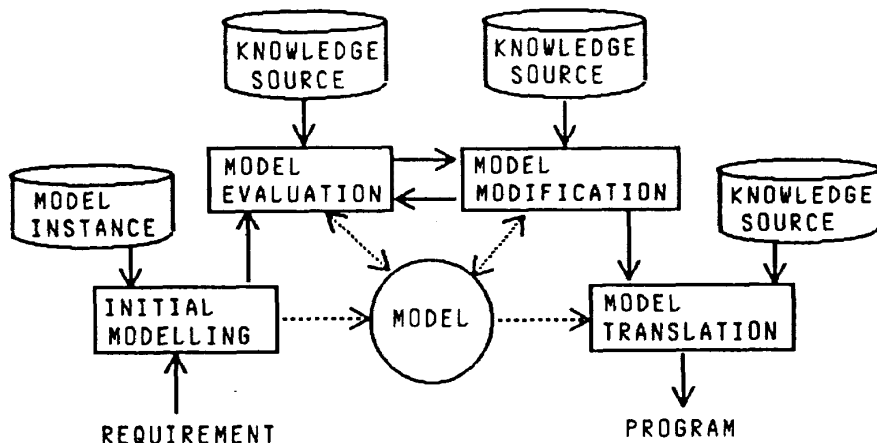
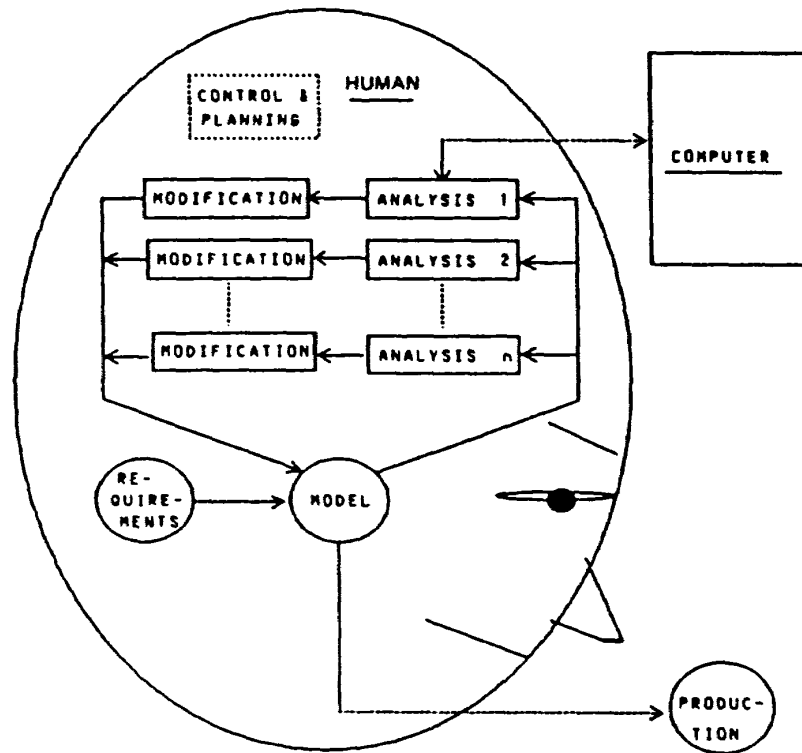
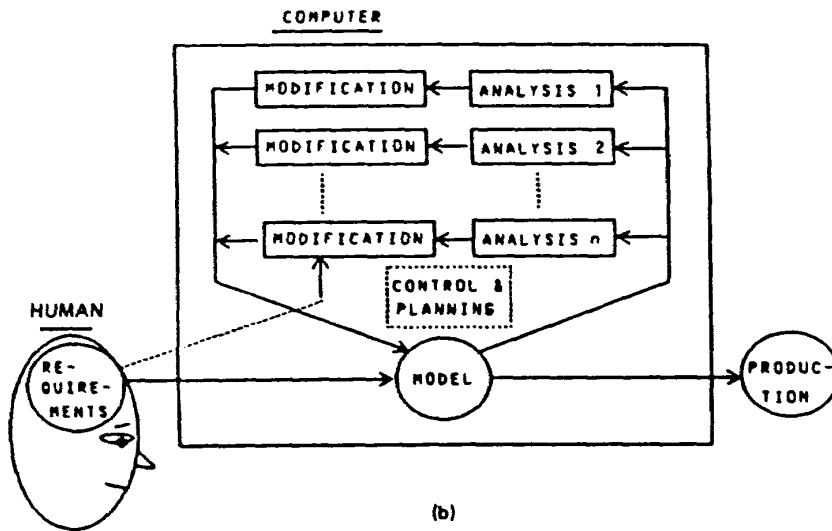


図1 設計プロセス



(a)



(b)

図2 設計方式の変遷

利用比率は余り大きなものにならないばかりか、コンピュータを利用するための手間がかかる。これにたいし、プロセス自身を明示的に表現することができれば、図2(b)のように、設計プロセスの基本部分はすべてコンピュータ内に表現し、従ってコンピュータ内で設計を進めることができる。これが知識処理の目標である。これまでのエキスパートシステムはこの開発目標が明確でなかったため、本来、明示化が可能であり、処理可能であ

る筈の領域の僅かな一部を扱うに過ぎなかった。もちろんいかに人工知能が進んでも、人間の能力と比べれば知的機能は低く、設計のように高度の知能を要する作業全体をコンピュータが行うこと、すなわち自動設計は困難である。設計過程で、高度の判断による決定が必要なときはいつでも人間が介入できるインタラクティブシステムと成ることは必然である。

知識処理によってこれを実現するには、(1)モデ

ル、(2)モデル変換規則、(3)問題解決過程、を表現することが必要である。この詳細を述べる余裕はないが、この条件を満たすシステムの設計は可能であり、KAUSはほぼその条件を満たしてきた。[OHS89]

モデル及びその変換の表現には知識表現言語が重要な役割を演じるが、同時にそれを処理するには、知識の処理とデータベースやプログラム等、従来のコンピュータ技術との統合が不可欠になる。知識処理システムにおける言語は知識表現言語と呼ばれる。知識処理システムにおける知識表現言語は従来のコンピュータにおけるプログラム言語に相当する。知識表現言語とプログラム言語の相違は前者が宣言的な記述方式を、後者が手続き的方式を取る点にあるということができる。詳細は省くが、この言語の相違は応用面で大きな相違と成って現れる。具体的には宣言的言語は設計を始めとして創造的な仕事の支援が可能であるが、手続き的言語はそれに適していない。なお、従来のコンピュータにおける処理の機構はプログラムから命令を逐次的に取り出して、内蔵している演算処理の機構によってそれを実行することであるが、知識処理ではパターンマッチを基本とする推論機構がこれに対応する。

モデルの表現は問題分野によって異なるが、個々のモデルの相違を越えて、モデルは構造と性質等の記述からなるという共通性を持つ。性質等とは対象の固有の性質、与えられた環境での機能や振舞い、他の実体との関係等をまとめて表す。例えば航空機設計の場合、モデルとして形状や構造の様に図で表される情報と共に重量や材質等、言語的に表現されるものの組合せとして対象のモデルが定義される。人の場合、図面や言語が使われるが、コンピュータ内では情報の構造と、述語と呼ばれる言語が用いられる。従って、知識表現言語としては基本的にデータ構造を含む性質等の宣言的記述が可能なるものであることが条件になる。モデルの変換知識は「(構造と性質の複合で表されている)モデル-1の構造(の一部)もしくは性質を変えてモデル-2を生成する」という形で表される。

一方、問題解決過程は処理の過程において利用される知識ソースの適用順序を指定する等によって実現されるが、この表現は知識に関する知識である。通常、知識に関する知識をメタ知識と呼んでいる。これに対し、上記の変換規則のように、対象であるモデルに直接作用する知識を対象知識と呼ぶ。メタレベルでは対象知識をデータ構造によってグループ化し、メタ知識によって適用順序の指定などが出来るようにしている。これらは実現にはそれに適したシステムアーキテクチャの設計を必要とする。柔軟性、適応性、成長性を損なわないように注意しながら、これらの諸要求を満たす知識処理システムを設計せねばならない。一例としてKAUSの基本機能を表1に示す。

表1 KAUSの基本機能

- | |
|--|
| 1. 構造と性質等の記述からなるモデル表現に適した言語 (MLL) を持つ。 |
| 2. その言語のための推論機能を持つ。 |
| 3. 知識処理とプログラム処理の統合が可能。 |
| 4. 知識処理とデータベース処理の統合が可能。 |
| 5. 多層のメタレベル記述が可能。 |

KAUSではモデルを表すために、任意のデータ構造を含む述語の体系を知識の表現に用いる。これを多層論理(MLL-Multi-Layer Logic)と呼んでいる [OHS85]。設計のための各種理論や方式、経験、ノウハウ等は、通常人間が用いている表現に近い形式で表現され、知識ベースに蓄えられ、推論機構によって問題解決に利用される。この方式は従来の情報処理の方式とはまったく異なるものである。モデル中心のシステムでは知識処理における推論と既存プログラムの結合及び知識ベースとデータベースの統合は特に重要であり、KAUSでは前者は推論の中で組み込みプログラム (PTAと呼ぶ) を利用すること、しかも任意のプログラムをPTAとして登録することができること、さらに特別のPTAを通して既存プログラムとの結合ができることによって実現している。後者にたいしては、KAUSでは弱結合、強結合の2通りの方法で知識ベースとデータベースの統合が可能

に成っている。前者は外部の既存データベースの利用方式、後者はKAUSシステム内にあって、知識ベースと一体で利用されるデータベース利用方式である。後方で利用されるデータベースは構造の表現が可能な、オブジェクト指向データベースになっていて、これはデータからの知識の抽出による知識獲得のためにも用いられる。

一方、設計過程の表現は図1の様なプロセスの構造を表すことであるが、図1にも示されているように、これは知識ベースを分割し、処理の過程

において利用される知識ソースのモデルへの適用順序を指定することによって表現される。これは基本的には宣言的な表現に手続き的な表現を導入することに他ならない。このことは、問題の一般的な表現は宣言的記述と手続き的記述の混合であり、その最適な表現を実現することが重要であることを示す。この表現は設計対象によって異なり、対象に関する知見が増すほど、分割も適用順序も詳細になる。すなわち手順化が進む。図3は、フィードバックコントロールシステムの設計プロセス

ブロック図- シグナル- 局所ループ伝達関数集合- 全系伝達- 計算機
フロー図 及び開経路伝達関数集合 関数 プログラム

図3 フィードバックコントロールシステムの解析手順

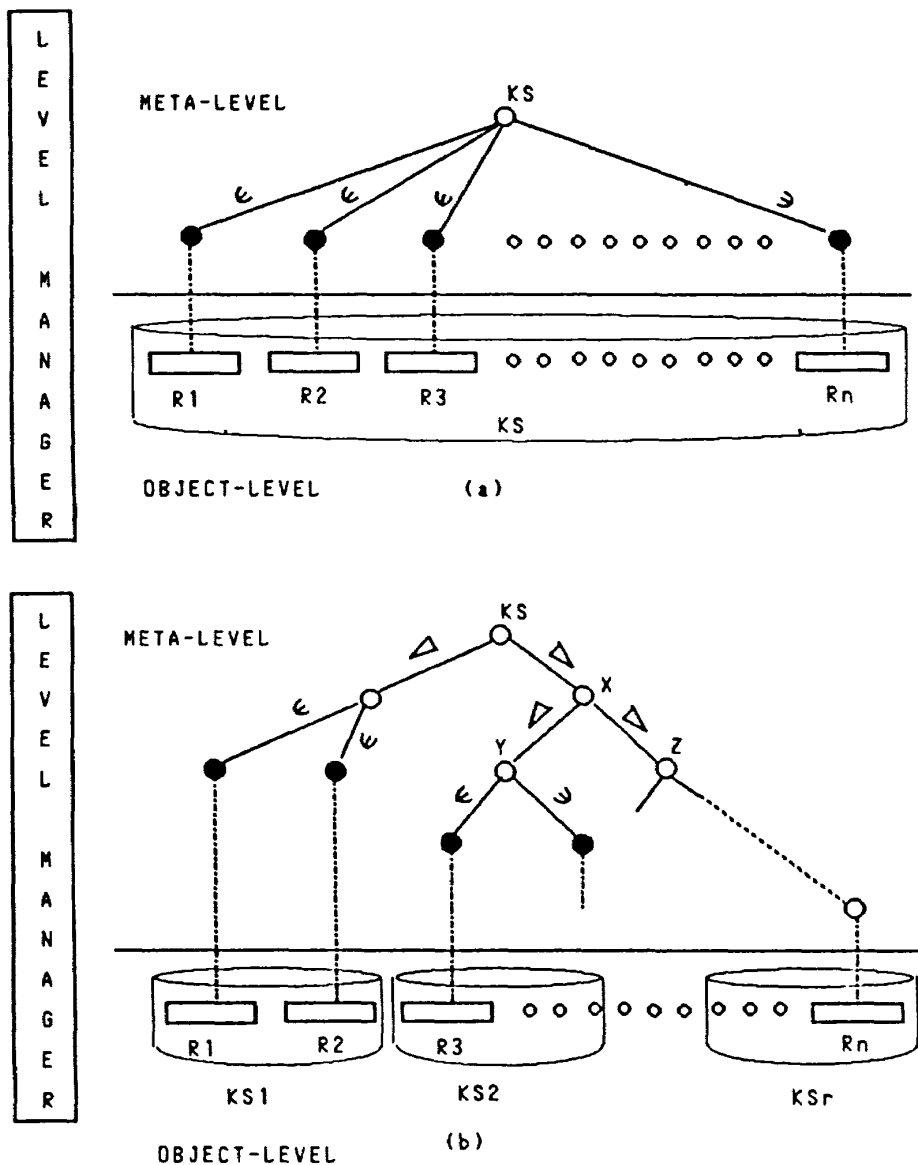


図4 メタレベルでの表現例

内の解析部分の手順化の例である。

設計プロセスの表現はメタ知識を必要とする。これらの実現にはそれに適したシステムアーキテクチャの設計を必要とする。KAUSでは対象知識のグループ化をメタレベルでデータ構造によって表し、適用順序の指定などが出来るようにしている。図4はこの例を示す。オブジェクトレベルにおける知識はメタレベルにおいてそれぞれ固有のIDを持ち、メタレベルでそれが構造化される。この構造によって知識ベースが分割され、必要に応じてこの構造に、個々の知識ソースの適用順序や制御戦略等を示す記述が与えられる。

この表現はオブジェクトレベルにおけるモデルの表現と同じ形式であり、従って同じ知識表現言語によって記述することができる。この構成は重要である。と言うのは、メタレベルの上にさらにメタレベルを定義し、メタレベルにおける設計プロセスや制御戦略自体を変更するシステムをつくることができるからである。これは上位に幾らでも発展が可能であり、しかもシステムは著しく複雑化することはない。システムとして付加するのは、システムの動作がどのレベルで行われているかを管理する部分—これをレベルマネージャと呼ぶ—のみである。KAUSでは個々のレベルを構成する基本部分を基本パッケージと呼ぶ。またKAUSは基本パッケージをレベルマネージャのもとで

必要なレベルまで多層に重ねられる構成に成っている。

図5は表現上の便宜からメタレベルシステムは除いて、KAUSのシステム構成を示したものである。KAUSではこれが基本パッケージとなっている。メタレベルはこれをレベルマネージャのもとで多層に重ね合わせた構成に成っている。

現実にはKAUSは多層論理のサブセットを用いているが、これまでに様々な応用例に適用して、その機能をテストし、ほぼ所用の機能を持つものと考えられている。本稿ではその中で分子構造設計へのKAUSの利用を一例として示す。

5. 分子構造設計への応用

このような基本的な設計支援の枠組みが与えられていれば、後はこれに分野ごとの固有の設計情報を与えてやることにより、それぞれの分野向きの設計支援システムが構築される。KAUSはこれまでに、機械設計、フィードバックコントロールシステムの自動設計、航空機の翼の設計 [TAK-89]、化学物質合成支援などの実問題に適用されてきた。以下ではその中で分子構造設計に関して述べる。[AKU87, AKU88, SUZ90]

この目的は医薬品等の開発に際して、候補となる分子構造を速やかに生成することである。これは設計問題であり、目的は与えられた要求、例え

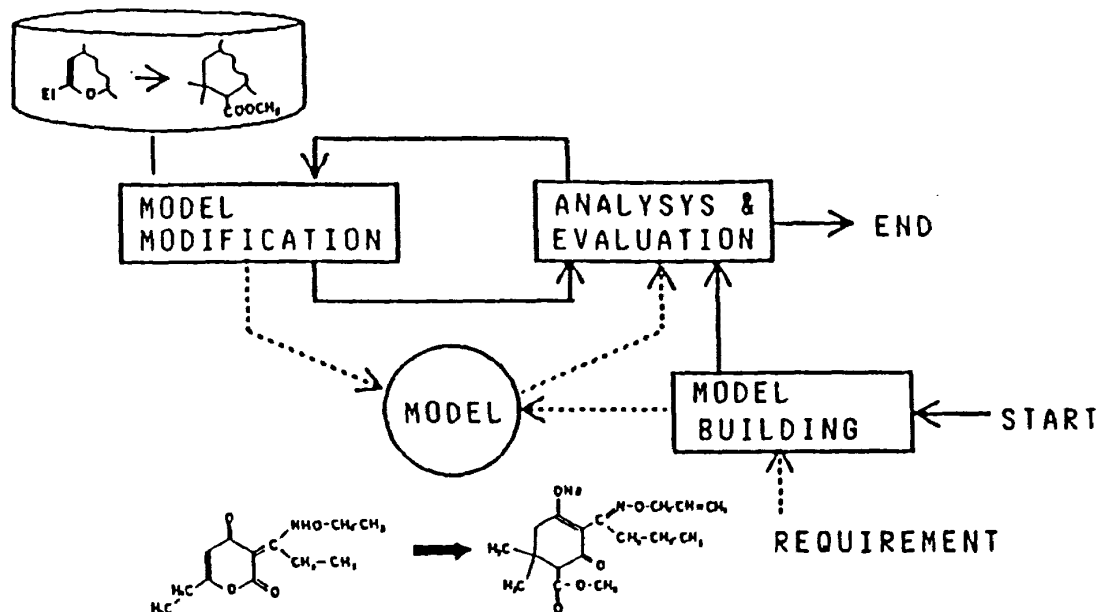


図5 分子構造設計プロセス

ば所要の生理活性を持ち、副作用が少なく、原料が容易に入手可能である等、を満たす新しい構造を持ったモデルの生成である。図1をここに適用すると、モデルは目指す新分子、モデル解析・評価はその性質例えば毒性の評価、モデル修正は分子構造の変換に対応する。一般に、設計問題ではモデル評価は多面的に行われる。

新しい構造を持った分子の生成には少なくとも2通りの異なったアプローチがある。一つは単純な構造を持った分子の複合として、より複雑な構造の分子を定義してゆくもので、これを仮に生成法と呼ぶ。もう一つはこれまでに非常に多くの分子が作られて来たが、その反応の経過から変換則を導き、それを利用して新しい分子を生成するもので、仮に変換法と呼ぶ。以下で述べるのはこの後者のアプローチに対してKAUSを適用した例で

ある。KAUSは設計支援手段であるから、分子構造設計に際しては必要な情報は化学の分野から提供されねばならないが、この応用に際しては情報は京大藤田教授のグループから頂いている。

図5は図1をこの例に適用した場合の構成を示す。モデル変換知識はこれまでに実現された過去の反応例から、まず(反応前の構造-機能, 反応後の構造-機能)の組と、同時にこの変換によって生じる機能変化の特徴を抽出してKAUS言語で表現している。

図6はこのシステムの実験例である。図7はメタレベルの機能の説明図である。オブジェクトレベルには多数の変換則が作られ、これは利用開始後も増え続ける。無作意の変換則の利用は組合せ的爆発を招く。このためメタレベルにおいて変換則をその特徴によって分類している。例えば有る

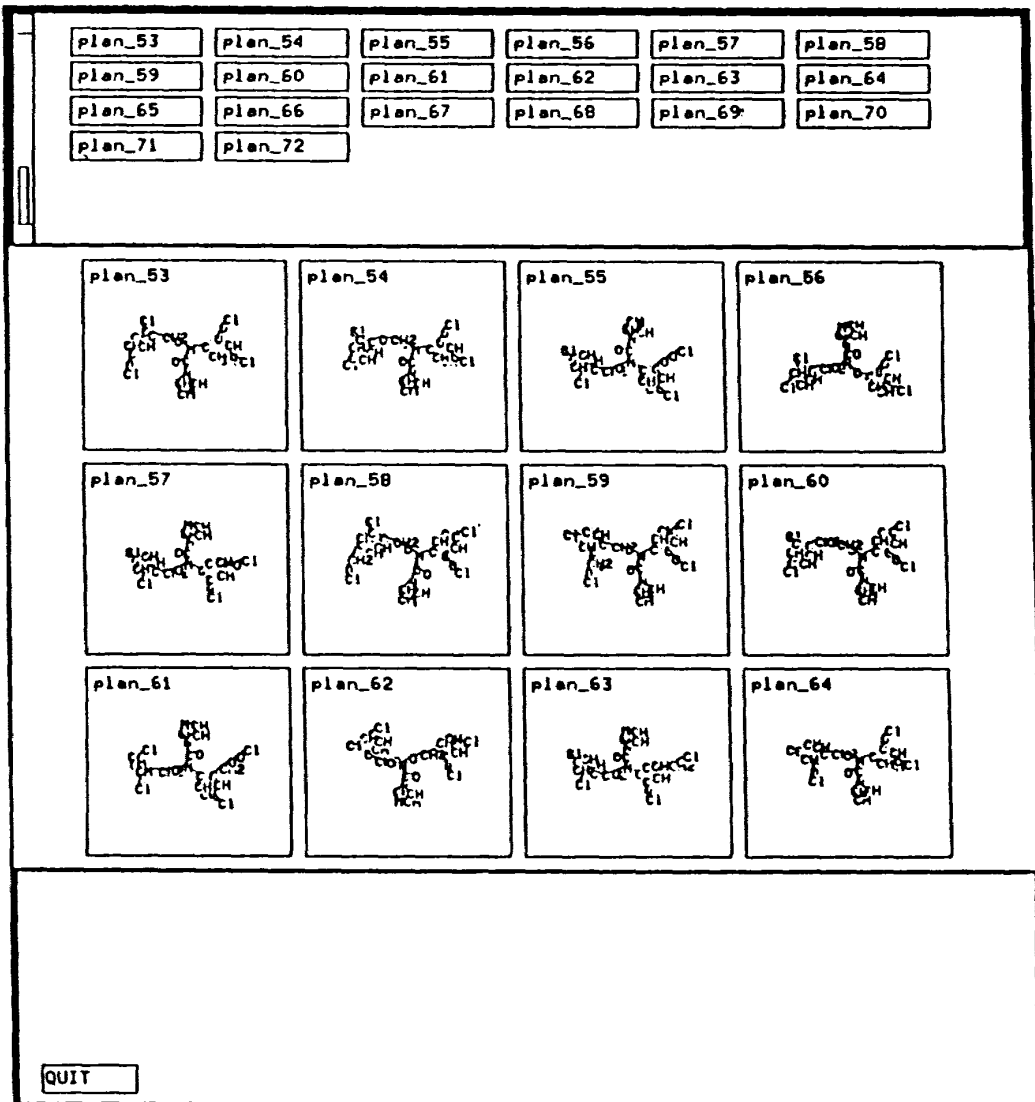
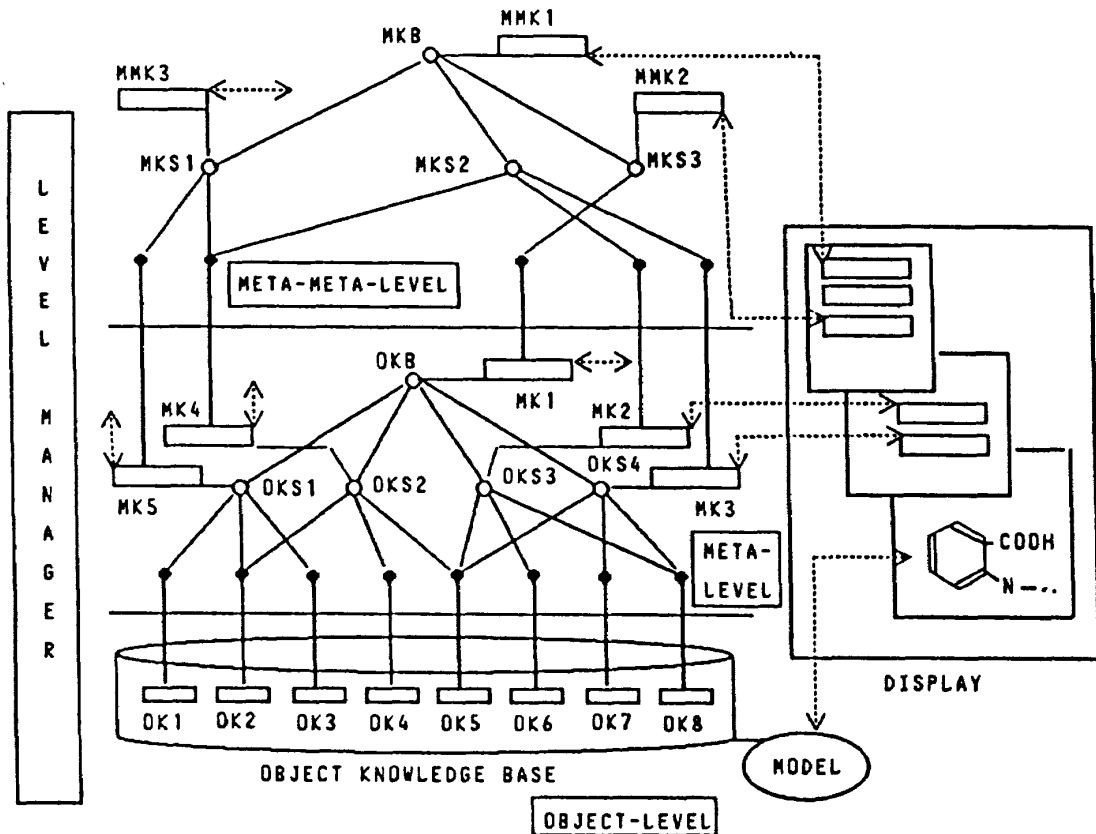


図6 分子構造設計実験例



MMKi; Meta-Meta-Level Knowledge
 MKi; Meta-Level Knowledge
 OJi; Object-Level Knowledge (Conversion Rule)
 MKSi; Meta-Level Knowledge Source
 OKSi; Object-Level Knowledge Source
 MKB; Meta-Level Knowledge Base
 OKB; Object-Level Knowledge Base

図7 分子構造設計プロセス制御例

モデルの生理的あるいは物理化学的な機能に注目し、その機能に一定の変化をもたらす変換則をまとめるといったものである。このまとめ方は普遍的なものである必要はなく、自由な分類が可能である。試行的な分類を行ってみることも自由に行える。この分類をディスプレイ上に表示して、利用者の主導権のもとに設計を進めることが可能であると共に、さらにこの変換則利用順序を定義しておいて、長い変換を一度に行うことも出来る。知識処理方式によればシステムがこのような利用上の制約は最小にすることが可能である。

6. むすび

KAUSによる設計支援の方式について述べた。

設計支援について重要なことは、対象のモデル、その変換則、それによって構成される設計過程をすべて同一の言語で表現し、自動処理可能な方式とすることである。これは設計支援のベースとなる知識処理システムへの要求であり、知識表現、推論機構、システム制御などに関わる。KAUSはこれら要求を満たすものとして開発され、現在、対象レベルについていくつかの実験によって考え方の正しさは実証されている。なおこの研究の一部は科学技術庁の科学技術振興調整費に基づく研究プロジェクト[化学物質設計等支援のための知識ベースシステムに関する研究]において行われたものである。

参 考 文 献

- [AKU88a] 阿久津達也他；化学構造推定エキスパートシステムの構造生成方式に関する考察, 第2回人工知能学会全国大会論文集1987
- [AKU88b] Akutsu, T. & Ohsuga, S.; CHEMLOG-A Logic Programming Language/System for Chemical Information Processing, Proc. 3rd Int. Conf. on FGCS, 1988
- [KAL87] Kaley, Y.E., Swerdloff, L.M. & Harfmann, A.C.; A Knowledge-Based Computable Model of Design, Working Conf. IFIP W.G. 5.2 Working Conf. on 'Expert Systems in Computer-Aided Design, 1987
- [OHS85] Ohsuga, S. & Yamauchi, H.; Multi-Layer Logic-A Predicate Logic Including Data Structure as Knowledge Representation Language, New Generation Computing, Vol.4, 1985
- [OHS89] Ohsuga, S.; Toward Intelligent CAD Systems, CAD, Vol.21, No.5, 1989
- [SUZ90] 鈴木英之進他；メタ・レベル制御を用いたドラッグ・デザイン支援システム, 第4回人工知能学会全国大会論文集, 1990
- [TAK89] Takasu, A., Itoh, Y., Futatsugi, S., Ohsuga, S.; Intelligent Wing Design Support System, Proc. of 2nd Scandinavian Conference on Artificial intelligence, 1989

