

TM-0747

設計問題向き知識獲得支援システムの構想
(1988年度 KSA-KAD-SWG報告書)

溝口理一郎(大阪大学), 瀧 寛和

July, 1989

©1989, ICOT

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456-3191~5
Telex ICOT J32964

Institute for New Generation Computer Technology

I C O T T e c h n i c a l M e m o r a n d u m

設計型問題向き知識獲得支援システムの構想

－ 1988年度 知識システムソフトウェアアーキテクチャワーキンググループ・
知識獲得&設計サブワーキンググループ報告書 －

KSAワーキンググループ・KAD-SWG
(溝口理一郎[大阪大学]主査)
(担当、ICOT 第五研究室)

I C O T

1989年 6月22日

目次

序文	p. 2
設計例の理解に基づく知識獲得インタビューシステム 溝口理一郎主査(大阪大学 産業科学研究所)	p. 3
設計エキスパートシステムにおける知識獲得 小林康弘委員(日立製作所 エネルギー研究所)	p. 7
フルカスタム論理合成エキスパートシステム(EXLOG)と設計知識獲得支援システム(VILLA) 渡辺正信委員(日本電気 C&Cシステム研究所)	p. 10
事例ベース・テンプレートによるルールベース構築・設計問題への構造的アプローチ 石田好輝委員(京都大学 工学部応用システム科学)	p. 12
計装・計算機のシステム設計支援と知識獲得 萩原賢一委員(富士ファコム システム本部)	p. 15
報告書 篠原靖志委員(電力中央研究所 経済研究所情報システム部)	p. 18
報告書 元川 浩委員(日立製作所 基礎研究所)	p. 20
報告書 大塚尚宏委員(アンリツ 厚木事業所)	p. 22
報告書 森下太朗委員(シャープ 情報システム研究所)	p. 24
報告書 薄 隆委員(ソニー 総合研究所)	p. 27
報告書 末田直道委員(東芝 システム・ソフトウェア技術研究所)	p. 29
報告書 長澤 勲委員(九州大学 中央計数施設)	p. 34
設計診断知識の獲得 山口高平委員(大阪大学 産業科学研究所)	p. 37
報告書 安部憲広委員(大阪大学 産業科学研究所)	p. 40
報告書 伊藤公俊委員(東京工業大学 総合理工研究科)	p. 42
深層のおよび連合的領域知識を用いた一般化手法による概念設計知識獲得 片井修委員(京都大学 工学部 精密工学教室)	p. 44

(序文)

本報告は、1989年度 KSA-KAD-SWG(知識ベースソフトウェアアーキテクチャワーキンググループ/設計問題の為の知識獲得サブワーキンググループ)の検討結果を各委員ごとに下記の項目からまとめたものである。

- ・注目する設計問題とそのAI的特徴・モデル
- ・利用可能な技術と理由
- ・知識獲得の方法
- ・知識獲得の問題点
- ・設計向き知識獲得システムのイメージ

このワーキンググループの討論結果から次の研究課題を抽出できた。

- (1) 設計問題では、初期知識ベースの構築だけでなく、設計過程を通じての知識獲得が必要である。設計過程は、そのものが設計者の学習過程である。
具体的には、E B Lなどの学習システムと知識獲得システムの融合が必要。
- (2) 獲得すべき知識として、制約を表す知識の獲得が必要。また、その制約知識を扱うメタ知識の獲得も必要。これには、知的バックトラックの戻り先や制約緩和の知識も含まれる。

なお、本報告書の作成にあたり、執筆に御協力いただいた主査並びに各委員の方々に感謝いたします。

設計例の理解に基づく知識獲得インタビューシステム

大阪大学
溝口理一郎 松田勝志

1. 注目する設計問題とそのAI的特徴

油圧回路の設計知識の獲得を考察する。

油圧回路の特徴：形状の要素が少ない。

要素機器（部品）の概念が確立している。

知識獲得の状況：Domain theoryの概念が確立している。

設計知識の整備のための獲得支援が要求されている。

2. 利用可能な技術と理由（問題解決としての視点）

本稿は知識獲得に重点を置いているので、省略する。

3. 知識獲得の方法

（獲得対象）

油圧回路の概念設計を含む、要素機器の選定や回路構成の知識を獲得する。

（方法）

油圧回路の例を与えられたとき、ある要素機器がある場所に存在する理由をDomain theoryを用いて理解を試み、成功すればその結果を一般化して知識ベースに蓄える。理解に失敗すればインタビューを行い、不足するDomain theoryか、あるいは、設計知識を獲得する。

（理由）

豊富な例題を有効に使う。

既知の知識を有効に利用する。

インタビューの負荷の軽減。

Domain theoryが万能ではないのでインタビューが必要。

4. 知識獲得の問題点

（1）獲得に有効な方法論の活用

（2）例題からの学習とインタビューによる獲得の融合

（3）学習技術の強化

（4）インタビューの戦略

（5）インタビューの際に与える刺激、補助的な情報の整備

5. 設計向き知識獲得システムのイメージ

現在我々が開発中の油圧回路を対象にした例題からの獲得とインタビューを融合した設計知識獲得システムの概要について述べる。

5.1 システムの概要

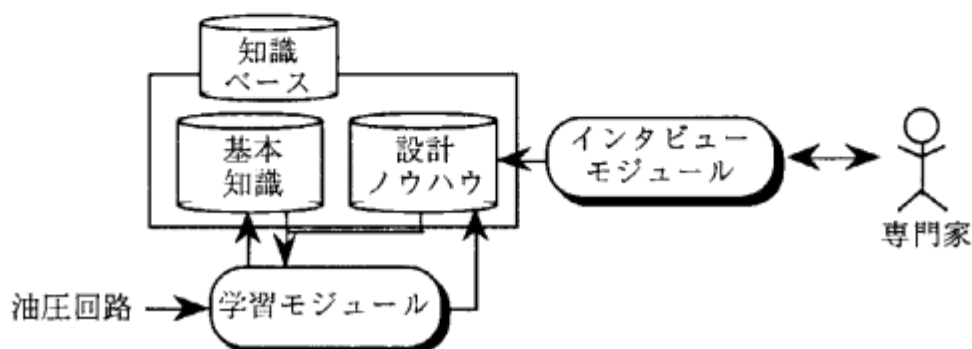
本研究は、油圧回路においてある要素機器がある場所に存在する理由を説明し、その説明からどのようなときにどこにどのような要素機器を接続するかという設計知識を学習とインタビューを用いて獲得することを目的としている。理解に成功した場合は、その理解の過程を設計知識に変換し、失敗した場合には、インタビューによって知識を獲得する。

入力：油圧回路図

出力：ノウハウ知識を蓄積した知識ベース

基本知識：油圧回路を理解するための必要な教科書レベルの知識、要素機器の動作を説明するために必要な力学知識

設計ノウハウ：基本知識では説明することが難しい知識



新たに蓄積される知識の種類

設計ノウハウ：専門家から直接獲得される基本知識では説明することが難しい知識

- ・マクロ基本回路：基本知識や設計ノウハウを用いて理解モジュールで学習したある動作、機能、目的を持った回路のパターン
- ・マクロ基本回路A：基本知識のみを用いる
- ・マクロ基本回路B：基本知識と設計ノウハウを用いる
- ・マクロ基本回路C：設計ノウハウのみを用いる

5.2 学習モジュール

以下に学習モジュールの処理過程を記す。

- 1.「与えられた回路例が要求仕様を満たしている」というゴールが与えられる。
- 2.このゴールにはいくつかのサブゴールがあり、そのサブゴールそれぞれをEBLのGoal Conceptとする。
- 3.Training Example（回路例）がGoal Conceptを満たすことを証明する。その証明は回路のシミュレーション結果を解釈することによって行われる。
- 4.この証明結果からGoal Conceptに対して大きな役割を果たしている要素機器または構造を取り出す。
- 5.これらの要素機器または構造が要求仕様の一部を満たしているということを説明する。
- 6.このようなアプローチを要求仕様すべてについて、更に隠れた仕様についても行う。

5.2.1 EBLとUBLとの違い

UBL (Understanding Based Learning)

UBLでは、学習システムにまず次のものが与えられる。

- ・ Goal Concept: 要求仕様に現れるパラメータが満たすべき条件の記述。
- ・ Training Example: 入力された回路例。
- ・ Domain Theory: 学習する油圧回路についてのルールと事実。Training ExampleがGoal Conceptの例になっていることを説明するために用いられている。
- ・ Operability Criterion: EBLに素直に適用すると、回路が与えられたら、それを効率よく理解するための知識を作っているが、このシステムでは、その知識の条件文と実行文とを入れ換えることによって設計知識を獲得することが目的であるから学習された記述により設計が効率よく認識されるように定めておく。

これらを用いて、

「ある要素機器または構造の回路内における機能と役割の記述」を得る。

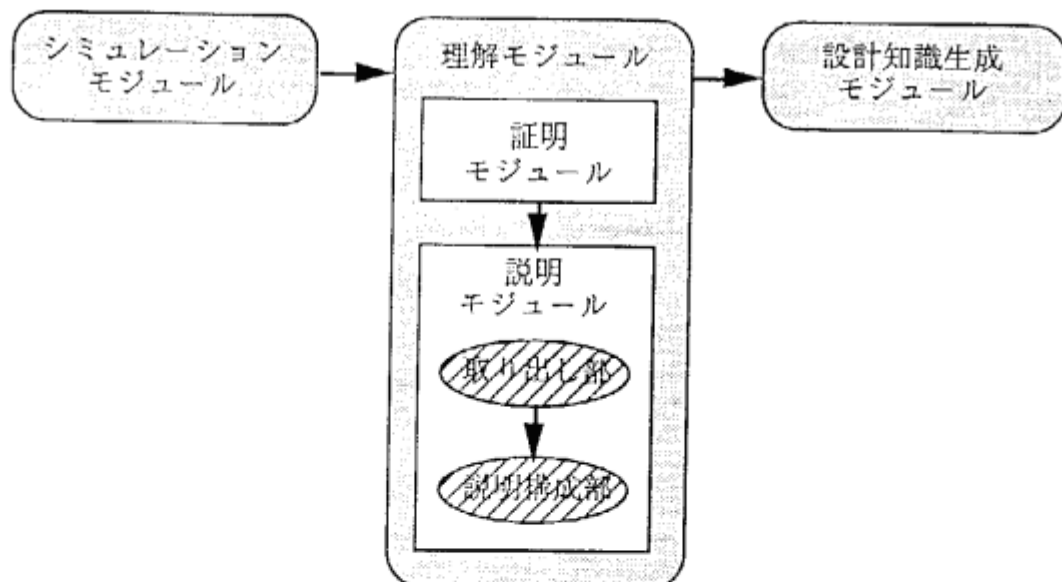
その方法は次の3ステップよりなる。

シミュレーション：与えられた回路図全体に対してDomain Theoryによって、油圧パラメータの状態をシミュレートする。

理解：シミュレーション結果がGoal Conceptを満たすことを証明する。そしてその証明からGoal Conceptに対して大きな役割を果たしている要素機器または構造を取り出す。取り出した要素機器または構造がその回路に存在する理由を証明結果などを用いて説明する。

設計知識生成：理解ステップで構成した説明を条件部と実行部を入れ換えることによって設計知識を生成する。

このステップを図にしたものが下の図である。



5.2.2 UBLの各部の働き

シミュレーションのステップ

シミュレーションは与えられた回路 (Training Example) 全体に対してバスカルの法則や各要素機器間での油圧パラメータの因果の連鎖など (Domain Theory) を考慮しながら行われる。

ここでは、圧力、流量の油圧二大パラメータについてシミュレートしている。

証明モジュール（理解モジュール）

証明モジュールではシミュレーション結果がGoal Conceptを満たすことを証明する。

実際には、与えられたGoal Conceptから証明を始める要素機器（デフォルトはアクチュエータ）と重要パラメータを見つけ、証明を開始する。ある要素機器からあるパラメータについてシミュレーション結果をトレースして証明して行く。

取り出し部（証明モジュール（理解モジュール））

構成された証明パスからその証明木のパラメータの値を変化させる原因を持っている要素機器または構造を見つける。実際には、その証明にどれが貢献しているかということを生成検査法によって調べてみて、見つける。

パラメータの値を変化させる原因には次の2種類のものがある。

- ・ リリーフバルブの設定圧のしきい値やフロコンの設定量の範囲などのパラメータを直接変化させるもの
- ・ 複数のパラメータによって成立する数式によってそのパラメータの値を変化させるもの

説明構成部（説明モジュール（理解モジュール））

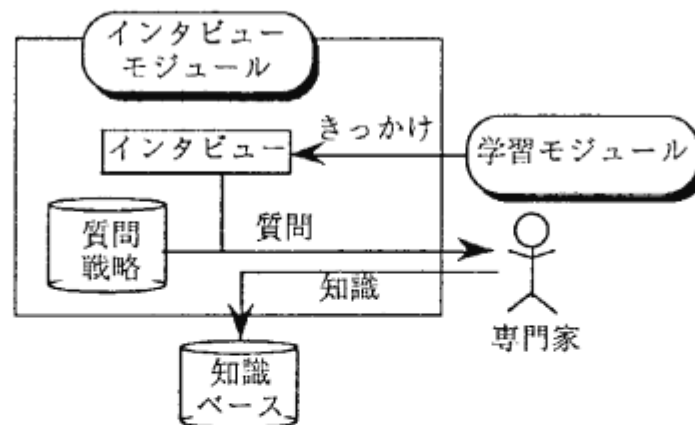
取り出し部によって取り出された要素機器または構造が、なぜその場所にあるかの証明に成功したものについて証明モジュールで証明されたものなどを用いて説明する。

設計知識生成モジュール

説明構成部によって生成された説明の条件部と実行部を入れ換えることによって設計時の効率を上げる

5.3. インタビューモジュール

本システムの特徴は、Domain Theoryを用いて例題の理解を試みて、理解できる範囲であればそれをもとに設計知識を獲得し、理解できなければ専門家にインタビューして直接設計知識、或は理解に必要なDomain Theoryを獲得するという、学習とインタビューを融合した知識獲得システムとなっているところにある。



インタビューにおいて重要な3つの課題については次のように対処する。

きっかけ--->理解の失敗

与える刺激--->理解過程の提示

事前に持つ知識--->Domain Theory

設計エキスパートシステムにおける知識獲得

1. 注目する設計問題とそのAI的特徴／モデル

プラントの論理制御系の設計支援

次の2つのステップから成る。

- (a) 論理の詳細化と実装レベルへの展開
- (b) 図面の構成変換と図面要素の配置

特徴：(a) 論理の基本パターンをより詳細なパターンに置き換える。
(b) 機能を表わした図面から詳細な制御論理を表わした図面を作る。

モデル：(a) 置き換えの手順を階層化する。
(b) 制約条件を抽出し、図面配置に反映させる。

2. 利用可能な技術と理由

- (a) 階層的推論、ルールの先読み
[置き換えの手順を制御し、整合の取れた制御論理を得るため]
- (b) 制約条件の宣言的記述の手続き的記述への変換
[図面のレベルで汎用な作画アルゴリズムを実現するため]

3. 知識獲得の方法

- (a) 既設プラントの図面(事例)を用い、設計者にインタビューする。
機能を表わした図面と、詳細な制御論理を表わした図面の対応する部分に注目して、どのような条件で、どのように置き換えていくのかをルールとして対話的に獲得する。
同時に、どのような置き換えを先にやり、どのような置き換えを後にするのかについてのメタルールを対話的に獲得する。
このステップに関しては、専門家の問題解決のプロセスを記号レベルで模擬する形であり、専門家の知識が利用可能である。
- (b) 基本的には、ドキュメントから獲得する。
ドキュメントとは、既設プラントの図面(事例)に他ならない。
設計者にもインタビューするが、図面作成のガイドラインや

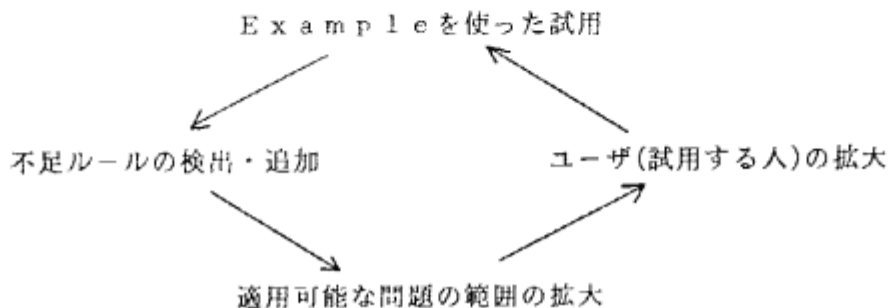
見やすさについてのコメントをもらうだけである。

このステップに関しては、基本的には、専門家の知識に期待していない。
知識処理を用いるのは、ソフトウェアのメンテナンスしやすさに
着目してのことである。

4. 知識獲得の問題点

- (1) 実用レベルの知識ベースに至るまでに時間がかかる。

専門家とのインタビューによるにしても、ドキュメントから読み取る
にしても、Example Driven な知識獲得を通して、
知識ベースは、次のようなサイクルを経てリファインされていく。



このサイクルを早く進めるには、使ってもらえるレベルに早くもって
いく必要がある。そのためには、開発の早い段階で、ユーザの問いかけ
に対して、適切とは言えないにしても何らかの反応を返すようなシステム
を作るべきである。

- (2) 知識ベースの守備範囲を越えると急激に無力化してしまう。

エキスパートシステムは、知識ベースの守備範囲を越えた問題に
対しては、オンオフ的に無力化する。人間のエキスパートの場合、
「当たらずとも遠からず」の答を出すことができる。

5. 設計向き知識獲得システムのイメージ

上記の問題点に対して「エキスパートシステムが設計者並みの常識を持てば良い」と
いうのでは、当面の答にならない。常識にかわる何らかの仕掛け(「常識もどき」)
によって、知識システムが直ぐには馬脚を現わさないようにするしかない。知識獲得
システムが、このような点においてもシステム作成を支援しようとするならば、
今までの知識獲得システムが相手にしていた知識よりも一回り外側の「知識」に
もっと目を向ける必要がある。

(i) 範囲を広げても知識がある場合

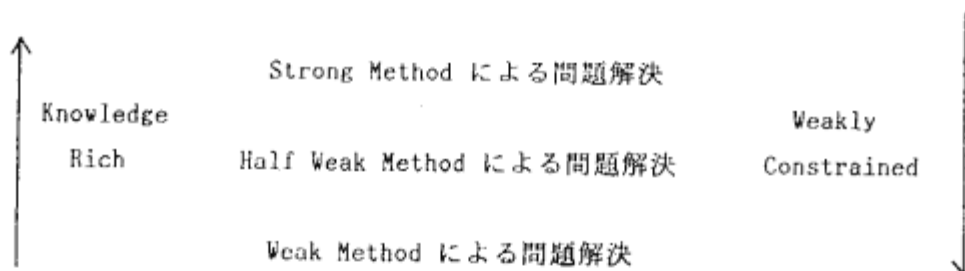
知識の多少の欠落があっても、オンオフ的に無力化しないような仕掛けをいれておくことが重要である。

(ii) 範囲を広げると知識がない場合

知識が直接使える部分に近い領域では、一般的な理論による「知識のパラシュート」(E.A.Feigenbaumの表現)が必要である。

設計向き知識獲得システムのイメージの前に、上記の問題点を緩和した設計向き知識システムのイメージをはっきりさせたい。ここでは、知識システム作成支援＝知識獲得と解して、知識システムのイメージから知識獲得システムのイメージを連想することにした。

上記(i)(ii)のいずれにせよ、「知識」が入っていないことによる無力化に対して抵抗力を持たせるためには、次に示すように、問題解決の方法の性格に注目し、ある方法での処理が難しい場合には、問題解決のパワーが低くてもより汎用的な方法でカバーすることが一策である。



ドメインの知識による Strong Method だけでカバーしきれない場合、Weak Method や Half Weak Method (J.J.McDermotの表現)を動員する。知識システム作成支援＝知識獲得を容易とするためには、ドメインを超えてタスクに注目したレベルで、すなわち、この Half Weak Method のレベルで知識システムを作成すべきである。Generic Task は、Half Weak Method の典型的な例である。もちろん、問題解決に、Half Weak Method が使えない状況では、Hill-Climbing のようなドメインやタスクを超えた Weak Method にも頼る場合もあるだろうが、とりあえずは、タスクのレベルでの「一般的な理論」として Half Weak Method を知識獲得の枠組みの中心に置きくべきであろう。

したがって、上記の問題点を緩和した設計向き知識システムのイメージは、基本的には、Half Weak Method による問題解決システムであり、適切な Half Weak Method が見つからない「知識」の穴の所では、その背景に Weak Method による問題解決の手段が見えているという構成をもつことになる。

フルカスタムVLSI論理合成エキスパートシステム（EXLOG）と 設計知識獲得支援システム（VILLA）

日本電気株式会社システム研究所
渡辺正信

1. 注目する設計問題とそのAI的特徴／モデル

注目する設計問題：フルカスタムVLSI論理合成

AI的特徴：論理合成知識／ノウハウが断片的であるためそのルール表現が基本
但し、複雑な構造を対象とするため高レベルなルール表現が必要

モデル：ノードバスモデルと呼ぶ設計対象表現モデルをフレームベースに構築した。
また、ルールセットに分類されたルールによる前向き推論を基本とする問題解決モデル。ルールセット内のルール適用制御とアジェンダ／タスクによるルールセットの実行制御からなる階層的推論制御モデル。

2. 利用可能な技術と理由

フレーム表現：設計対象やルール等を表現するのに柔軟で有効

ルール表現：設計知識を表現するのに有効。但し、16個の設計作用素を提供し、
ルール記述の簡便化を図った。例えば、モジュールの分解／併合用、
マクロのインスタンスシート用、マクロの動的生成用等の作用素がある。

3. 知識獲得の方法

論理合成ルール

<方法> 設計作用素毎のルール作成の自動化／半自動化手法

例えば、

①変換前／変換後の回路図を入力して論理変換ルールを自動生成して、
そのルールの説明文（疑似自然言語）を生成する。

②マクロ利用ルールの類似ベース獲得支援。

<理由> 設計作用素毎にルール獲得手法がパターン化できるため。

4. 設計問題の問題点／知識獲得の問題点

<設計問題の問題点>

設計制御過程のモデル化：特に、種々の基準を満たす最適設計制御

<知識獲得の問題点>

論理合成基本ルールの獲得手法

制御戦略知識の獲得手法

5. 設計向き知識獲得システムのイメージ

<短期>

設計向けルール言語を提供し、ルールのパターン毎にルール獲得手法／テンプレートを提供する。

<中長期>

設計プロセスをモニターして有効なルール知識／戦略知識を獲得する。

事例ベース推論+テンプレートによるルールベース構築：設計問題への 構造的アプローチ

石田好輝

応用システム科学

京都大学、京都市左京区吉田本町

1. 注目する設計問題とそのAI的特徴／モデル

(例) コンピュータ、自動車、飛行機、自動倉庫など多くのコンポーネントから構成されるシステム。

特徴：システム構築のためのコンポーネント、パラメータ選択を非常に多くの可能な候補から、他のコンポーネントの関係を考慮しながら行わなければならない。

モデル：設計の全タスクを順序づけ、階層化された問題空間に分割し、その各問題空間内では、データベースにアクセスしながらゴール指向探索を行う（提案-適用法, propose-apply method）。

2. 利用可能な技術と理由

1. 事例ベース推論

＜理由＞ 設計問題を構成的方法 (constructive method) で解く場合、初期デザインをどのように決定するかは困難な問題であり、過去の関連のある事例を目的指向でインデックスリングしておいて、関連のある設計事例を初期デザインとして用いることが出来る。

2. 設計問題専用ルールベース構築のための普遍問題解決アーキテクチャとそれを実現するためのルールテンプレート集合

＜理由＞ 組み合わせ爆発に陥りそうな設計問題専門家を、ルールベースで構築するとき、その構造化（ルールベースだけでなく、問題解決機構も含めて）が問題となる。そこで、その構造化を行うと同時に、ルールのコーディングも効率化するルールテンプレート集合が役立つ。

3. 制約伝播法

＜理由＞ 初期デザインとして提案された事例を、現在の設計に変形するために多くのパラメータおよびコンポーネントに改訂が加えられる。このとき、この改訂と他のパラメータ、コンポーネントと

の整合性をとるために制約伝播法が用いられる。この制約伝播ルールも上記のルールテンプレートライブラリからテンプレートをとって構築する。

3. 知識獲得の方法

(例) 以下の知識を獲得する。

1. 初期デザインとして用いる事例の獲得

<方法> 事例記述フォーマットを決定し、その項目ごとに埋めて行く作業を計算機と会話的に行う。

<理由> 事例のパラッキを制御する必要があり、また決められたボキャブラリのみを使う必要があり自動で行うのは困難。

2. 事例変形のための領域固有ヒューリスティックルールおよび制約伝播ルールを獲得

<方法> ルールベースをよび問題解決機構を構造化するために、問題解決法の各ステップでルールのテンプレートを用意し、そのライブラリ化したテンプレートを知的インターフェイス上で埋めて行くことによりルール構築する。

<理由> ある知識が問題解決法のどのステップで、どういう目的のために使われるかが明らかになり、知識の整理と知識獲得のガイド(どの目的の知識が無いとか、足りないとかが明確になる)が行える。

4. 設計問題の問題点/知識獲得の問題点以上の事例ベース推論とルールベース推論を融合した設計システム構築手法は、次のような問題点を考慮したものである。

1. 構成的方法で設計を行う場合、次のような問題点がある。

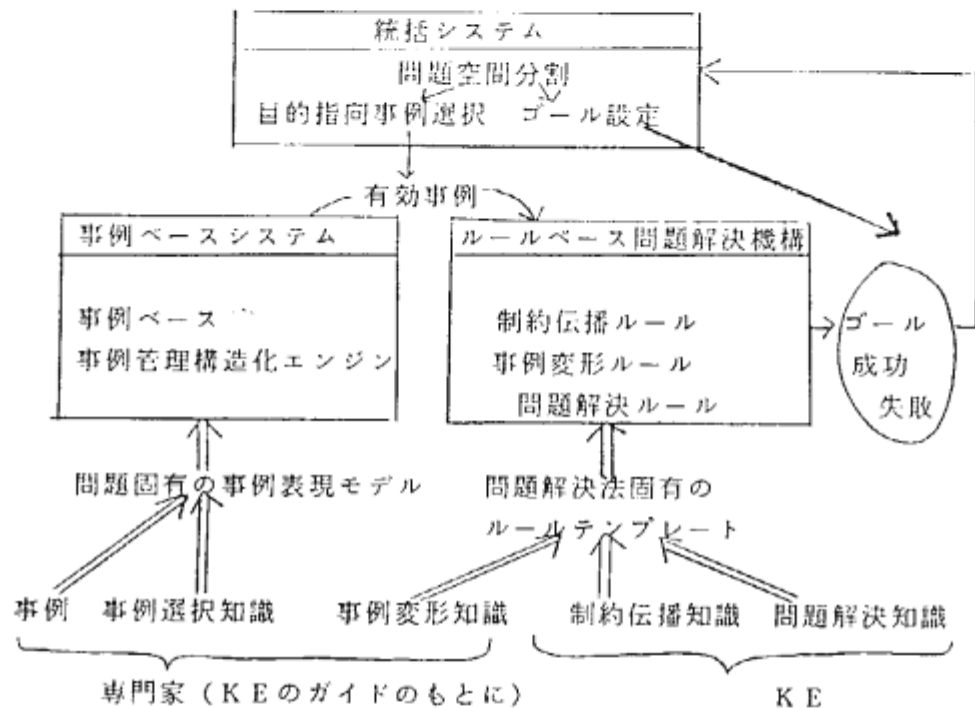
- (a) 初期デザインを選択する方法が確立されていない。
- (b) 設計プロセスが複雑なものには対処できない。
- (c) 過去の設計例を有効に生かした推論ができない。

2. 多くのコンポーネントからなる複雑なシステムを設計する場合、多くのデータベースにアクセスしながら設計を行わねばならない。

3. 設計問題をルールベースで構築する際、それをそのルールベースおよび問題解決機構を構造化する必要がある（ルールベースの管理、拡張性。また多くの人があるルールベースの構築に関与するとき、ルールのフォーマットを統一する意味でも）。

5. 設計向き知識獲得システムの構成

このこの事例ベース推論と SEAR の構造化の考えを用いた推論システム及び知識獲得の枠組は下図のようになる。



知識の流れ $\Longrightarrow$ 推論の流れ $\longrightarrow$

図：事例ベース+ルールテンプレートによるシステム構築の効率化

事例ベース設計のための知識獲得の課題

- (1) 事例表現モデル(事例距離も含む)の開発
- (2) 事例獲得戦略の獲得(事例をサンプリングする分布)
- (3) 事例変形ルールの獲得
- (4) 事例ベースをつねに管理、構造化するためのエンジンが必要である。

1. 注目する設計問題とそのAI的特徴／モデル

計装・計算機のシステム設計

特徴：

知識情報が、数値（プロセス入出力点数・コストなど）、文章（カタログ・機能説明書など）、図（システム構成図）など多様である。

また、未確定な仕様から概略設計したり、仕様を確定していきながら詳細設計を進めるというように処理すべきあいまいさの巾が広い。

モデル：

概略設計では、過去の事例の中から制約（適用分野、プロセス入出力点数やコストなどの概算）を満たし要求機能に近いものを検索し、それを修正してゆくことが多い。

詳細設計では、要求された機能や規模に合わせて性能とコスト上の制約を考慮しながら概略設計の結果を詳細化してゆく。そこでは、あらかじめ用意されたシステム構成部品の選択・組合せと制約に対する検査のサイクルが繰り返される。

2. 利用可能な技術と理由（問題解決としての視点）

事例推論

理由：

システム構成部品が、目的指向で作られているため、要求された機能・性能・規模・コストに対するシステム構成は類型化できる。そこで、類似事例の検索と変形に基づく事例推論で問題を解決できることが多い。

3. 知識獲得の方法（何を獲得するのか、方法は、有効な理由）

獲得する知識：

1) 類似事例検索知識

要求仕様（適用分野、機能、性能、規模、コストなど）から、適用可能なシステム構成事例を検索するための知識。特に、類似度判定知識。

2) 事例変形知識

制約を満たしながら、要求仕様に合せて事例を変形するための知識。特に、変形の仕方が複数ある場合の優先度判定知識。

方法：

過去の事例を分類して、典型例を抽出する。

事例を提示して、適用された要求仕様・仕様項目間の優先度・変形許容範囲を対話的に獲得する。

既に蓄積されている事例との関連も提示し、類似度判定知識の詳細化・典型例の取捨選択をする。

なお、システム構成部品の機能・性能・コストなどの制約事実は、カタログなどから別途データベースとして生成する。

理由：

要求仕様とシステム構成の事例とから、自動的に類似度判定知識や優先度判定知識を生成することは難しい。これらは、専門家から引き出す方が、実現しやすい。

また、過去の膨大な事例や新規事例を有効に活用できる。

そして、典型事例を合理的に蓄積するためには、追加した事例と既に蓄積されている事例との関連を専門家に提示することにより、事例の無駄な重複蓄積を避ける必要がある。

4. 設計問題の問題点／知識獲得の問題点

設計問題の問題点：

要求仕様を同時に満足できなくなったとき、どの項目を優先してシステム設計を進めるかは、人によって異なる。

また、仕様があいまいな段階で、概略設計をする必要がある。

このため、問題を論理的、画一的に規定することが難しい。

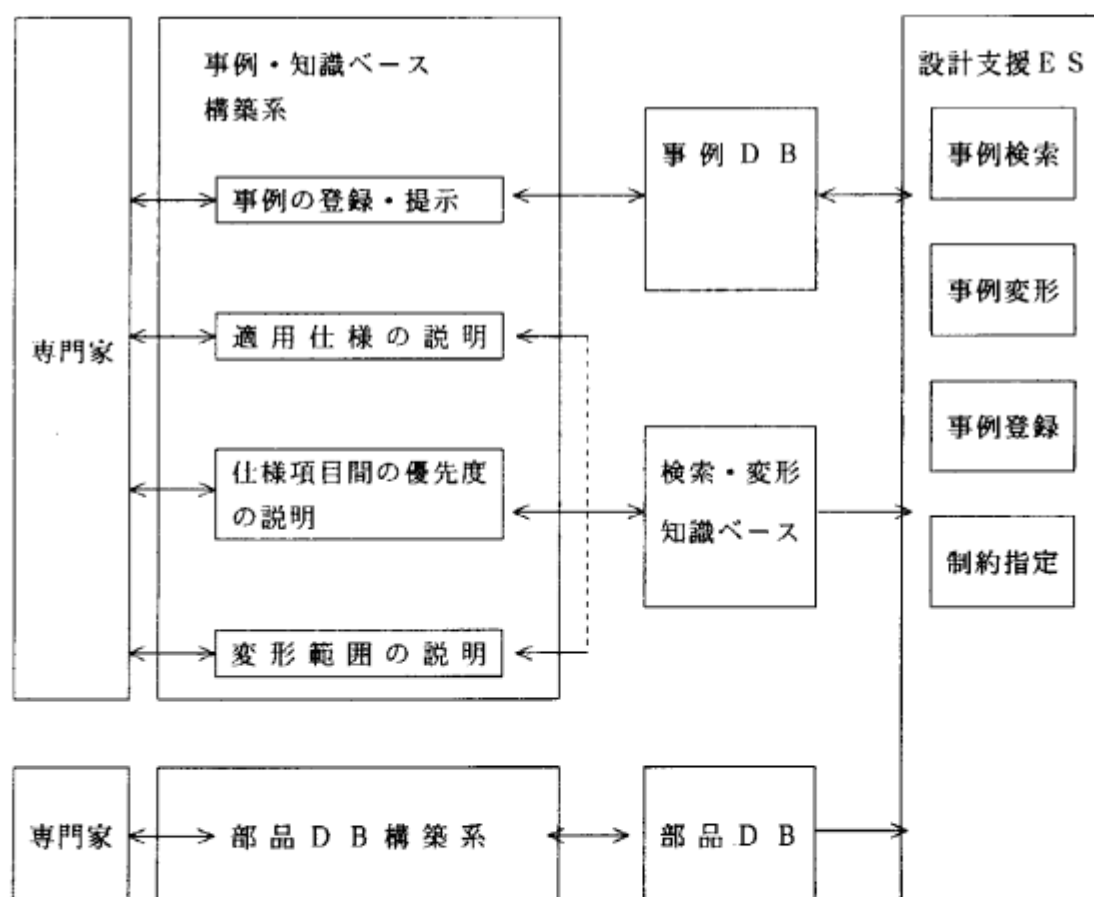
知識獲得の問題点：

システム構成部品に関する個々の知識だけでは不十分であり、システム全体に対して機能・性能・コストのバランスを判定する知識の獲得が必要である。

このため、システム構成部品に関する個々の制約事実は、なるべく機械的に収集する。そして、それとは分離して全体的な判定知識を事例に基づいて獲得しなければならない。

また、知識提供者の間で知識の矛盾が発生する可能性もあるので、各々の知識ベースを分離しておく必要がある。

5. 設計向き知識獲得システムのイメージ



1988 年度 KAS-KAD-SWG 報告書

篠原靖志

電力中央研究所

経済研究所情報システム部

知識処理研究室

1 注目する設計問題とその特徴 / モデル

1.1 対象問題

事務処理（ファイル処理）プログラミング

1.2 特徴：

仕様からファイル処理レベルへ情報を補いながら変換する問題としてとらえることができる。

仕様は、概念レベルでの入力項目から出力項目へのマッピング（導出関係制約）ととらえる（具体的計算式は未知でも良い。ただし、計算式が満たす性質は、利用者が与えられる）。これに、計算時間が短い、メモリが少ないなどの制約が加わる。

参考とできる情報としては、データモデルに関する情報、キー属性やアクセス方式などファイルレベルの情報、計算式の満たす属性がある。これらは、ファイル処理レベルへの変換の際の制約ととらえられる。

1.3 モデル：

基本的には、制約解析と生成検査モデルが使えると考えられる。

概念レベルの導出関係を元に、データモデル、ファイルレベルの制約情報を使った制約解析を行いながら、ファイル処理レベルの導出関係にマッピングする。マッピングの仕方は複数あり、その仕方は、ファイル属性や計算式の属性により変化する。

2 利用可能な技術

利用可能な技術としては、

- ・制約解析
- ・書換規則
- ・部品合成

の3つが挙げられる。

3 知識獲得の方法

処理フローへの変換知識の獲得を考えると、以下のような獲得方法が考えられる。
可能な処理フローを提示して、利用者に提示し、利用者に訂正してもらう。
利用者のフローが制約を満たすことを再検証する。
正当性が検証できない場合は、利用者に確認して、書換え規則等を付加する。
適切な場合には、EBLによって、処理フローへの書換えルールを得る。

4 設計問題の問題点 / 知識獲得の問題点

自明の制約だけでは解の自由度が大きすぎるものか、逆に、多面的で排反する制約を共にある程度満たさなければいけない問題が多いように思われる。

解の自由度が大きい場合には、利用者の選好が重要となるが、これは獲得しにくい。

また、排反する制約をある程度満たす必要がある場合には、解生成の途中段階で、最終的満足度を見積もることが難しく、効率の良い探索を行うことが行いにくい。

5 設計向き知識獲得システムのイメージ

設計問題の多くは、制約を充足する解を得る問題で、その手順を求める問題となる。

大きく2つの問題がある、1つは制約の獲得で、1つが手順の獲得である。

制約の獲得については、最も単純な方法が利用者からの直接入力である。さらに、一般的に、または、分野固有に良く使われる制約は、ライブラリ化しておくことで、定型的な制約を効率よく入力できる。さらに、詳細な制約については、実問題にぶつかって初めて気付くので、1つの可能解を生成して利用者に提示して、または、実際に問題を解いてみて、解けない理由の説明をしてもらうことによって、制約の追加を行う。

また、複数の可能解を提示して、好ましくないもの（利用者の選好）を選択してもらうことで、（満たした方が好ましいが、満たされなくても良い）弱い禁止制約が獲得できると考えられる。好ましいものとして選択されたものについては、EBL流の導出条件を見出しとして、解、その導出過程を解プールとして蓄える。

手順を求める問題については、利用者に入力してもらう方法や、制約情報から生成検査などの方法で求める方法に加えて、解プール中から解を検索する方法がある。解の効率的な検索が行えるようにすることで、早く解手順を提示できる。解プール中の解は、類似問題での基本解としても適用できると考えられる。

1. 注目する設計問題とそのAI的特徴／モデル

電子回路など機能レベルで定性的に仕様が記述されるもの

特徴：機能が定性方程式で記述され、構造との対応が比較的明確。

機能の階層化ならびにそれに対応した定性記述が容易。

モデル：仕様記述からの階層的検索、選択

2. 利用可能な技術と理由（問題解決としての視点）

階層的定性推論：設計目的に適合した代表的な挙動のみを説明できる定性記述がトップダウン的に利用可能。

事例からの一般化：具体的設計事例から抽出した代表的挙動記述を類似の部分目的に利用可能。

3. 知識獲得の方法（何を獲得するのか、方法は、有効な範囲）

具体的な設計事例から、部分構造に対し代表的な挙動を示す一般的な定性方程式を獲得する。これは特定の目的を実現するためにはこのように構成すればよいという知識を機能レベルで記述したヒューリスティックスである。

方法：図1にて説明。

- 1) 具体的な設計事例よりデバイスの動作原理を記述する定性方程式をレベル1の知識（深い知識）として与える。
- 2) 挙動予測において式の解釈が一意に決まらない場合にエキスパートから教示を受ける。
- 3) 教示内容を説明するための条件を探索（これこれしかじかの条件が成立するときは、この項は無視してよいなど）する。
- 4) 説明を拡張してフィードバック構造を検知（一般化しすぎを抑制）する。
- 5) 拡張された説明を一般化し、代表的挙動（レベル2）を生成する。この知識は「意味のある機能を持った固まり」の代表的挙動を表わすヒューリスティックスである。
- 6) 新しく生成された知識を併用し予測を繰り返す。この過程で説明に使用される知識のレベルが*i*であれば生成される知識のレベルは*i+1*と定義され、階層的に知識が獲得される。
- 7) 式の変換、値の評価の方法（定性推論による挙動予測の方法）はInterpretation Rule と Translation Rule で指定する。
- 8) 個別の事例から一般化された機能知識を蓄積し、機能レベルで記述した部品ライブラリーを構成する。

理由：設計用の知識としては目的に対応した知識が重要であるが、挙動から目的を自動的に抽出することは不可能である。従って、エキスパートの助けを借りて既存の事例から説明を生成することにより、将来他の設計にも利用可能な知識を獲得する。

4. 設計問題の問題点／知識獲得の問題点

- 1) 設計では帰納と演繹が混在しており、一般的に設計方法を記述することが難しい。
- 2) 過去の蓄積は事例ごとの個別的な構造記述が多く、機能的な記述で整理されていない。そのため機能的に類似な他の領域の設計事例の検索が困難であり、多分野の知識を有効に活用できない。
- 3) 知識獲得にもいろいろなレベルがある。原理を表わす一般的な基礎理論から具体的な構造に

いたるまでの距離が長すぎる。機能と構造を短絡的に結びつけるのは比較的容易であるが、類型設計の域を出ない。合目的な演繹推論技術を高度化する努力が必要である。

5. 設計向き知識獲得システムのイメージ

類型設計のためのライブラリーとしてのヒューリスティックスの獲得システムを図1にそれを用いた類型設計の概念図を図2に示す。

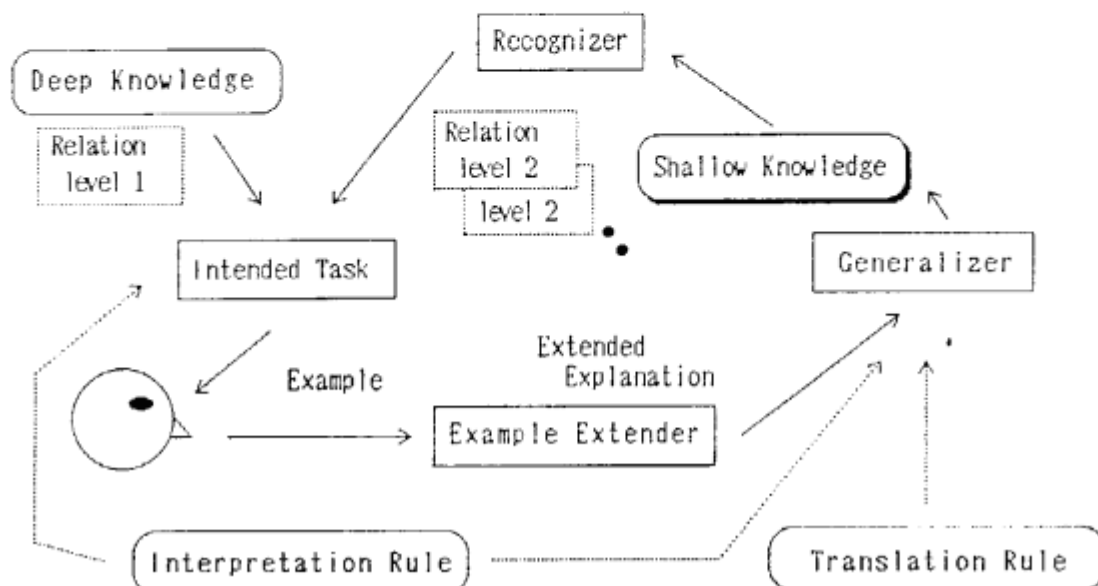


図1 階層的定性推論による設計ヒューリスティックスの獲得

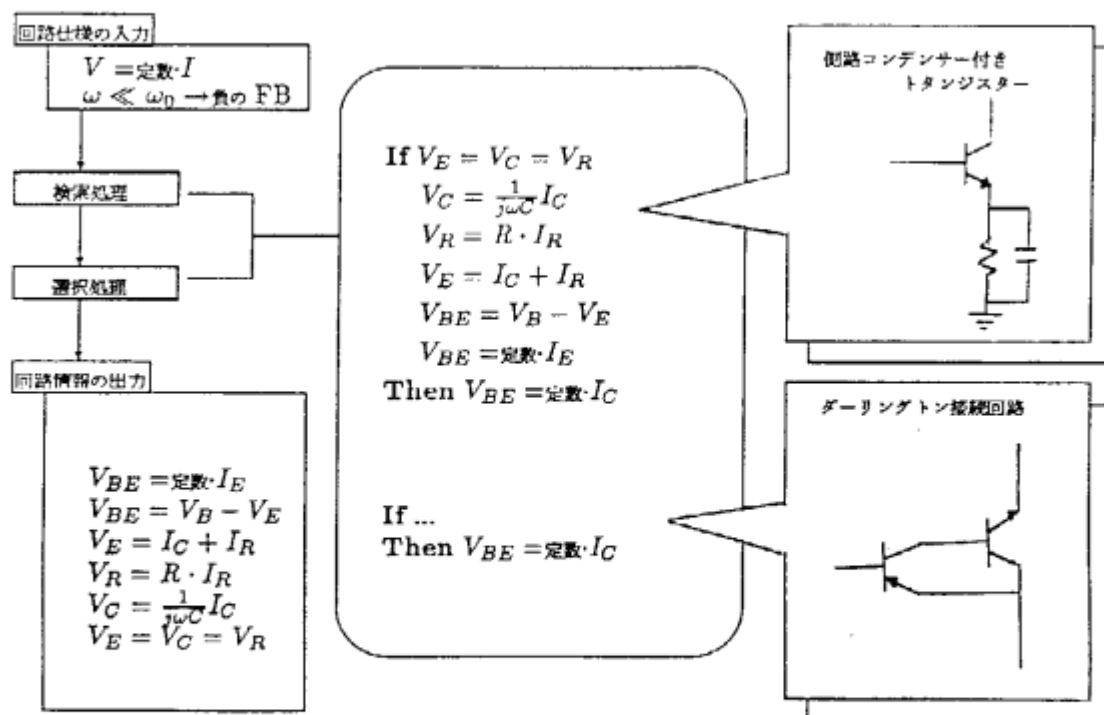


図2 階層的定性推論の類型設計への応用（概念図）

大塚尚宏(アンリツ)

1. 注目する設計問題とそのAI的特徴/モデル

回転機械の異常監視システムの設計

センサの選択および配置の最適化を行う

特徴: 異常の検出の精度および異常の弁別能力を制約条件として、コストの最小化(センサ数の最小化等)を行う。設備の種類ごとの制約条件の満足度(制約緩和の優先度に対応)を定める知識がシステムの構造を決める知識に対応し、診断知識がシステムの構成要素の特性を与える知識に対応する。

モデル: 生成検査モデル

2. 利用可能な技術と理由

原因結果分析

経験的知識あるいは対象モデルを用いて、異常原因と事象との因果関係を表す推論木を生成する。この推論木を用いて、各異常の発生を帰結する事象の組み合わせを求める。ある異常の検出の精度を上げたい場合には、推論木において異常のノードに近い側の事象のノードを選択することにより、検知する事象数を最小化する。また、異常の弁別能力を上げたい場合には、逆に異常のノードから遠い側の事象のノードを選択する。

ATMS

複雑な構成の設備の場合には、推論木が大きくなり、事象の選択の組み合わせが膨大になる。そこで、ATMSにより事象の組み合わせに対する制約条件の成立の状態を管理することにより、効率化を計る。

3. 知識獲得の方法

- 診断知識の獲得

方法: 診断システムのために獲得された知識を設計問題に利用する。

異常原因と事象との相関および事象間の相関に関する知識を診断の専門家より獲得する。

- 制約緩和に関する知識の獲得

制約条件の優先度に関する知識は、設備の種類ごとに異なり、異常の発生確率、異常の重要性などに基づいて決められる。ここでは、この知識を制約条件の満たすべき度合い(満足度)として定量化し、満足度に対する条件(不等式)を制約条件とする。

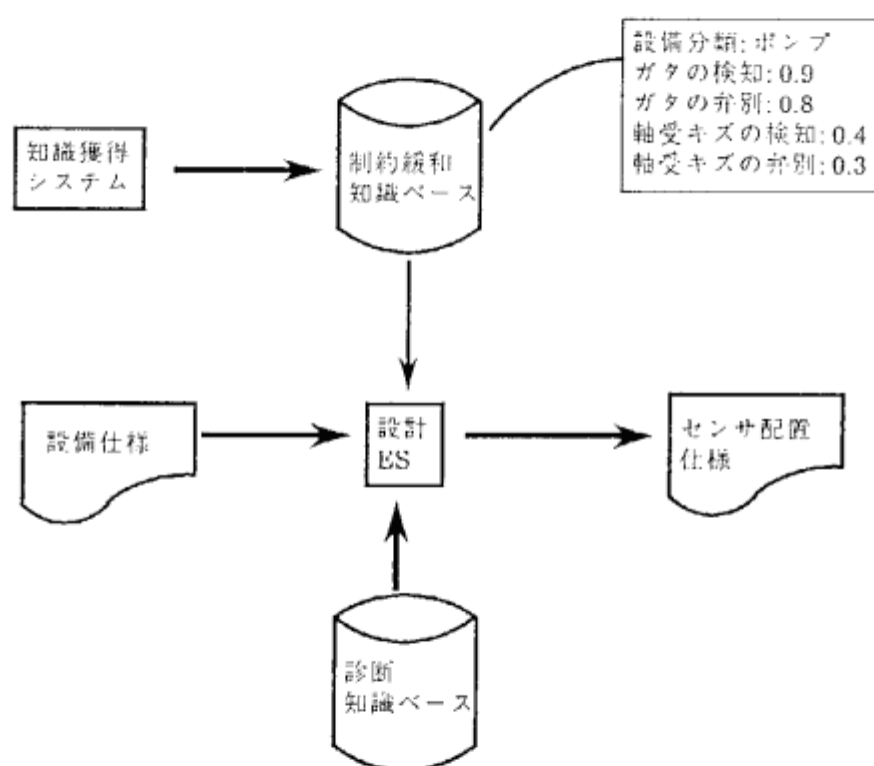
方法: 診断対象の種類に応じて、各制約条件の満足度(0以上1以下の実数値)の範囲を指定する。

4. 設計問題の問題点/知識獲得の問題点

制約緩和に関する知識は定量化されるため、知識獲得自体は容易である。しかし、知識が専門家の意図したような効果を持つという保証はなく、数多くの事例を実行して、その結果をもとに微調整する必要がある。また、知識が設備の種類ごとに異なるため、専門家の負荷が増大する。したがって、過去の事例から知識を自動的に生成することが望ましい。

5. 設計向き知識獲得システムのイメージ

専門家からの対話型入力あるいは事例からの自動生成により制約条件の満足度を診断対象の種類ごとに設定する。



【1】注目する設計問題とそのA I 的特徴/モデル

2. 5次元の金型設計

(特徴)

- ・設計の上流(機能設計)から下流(パラメータ設計)までの全ての要素を含んでいる。
- ・扱う範囲が広く、事例も多いことが期待される。
- ・3次元形状ではないため、C A Dの世界に偏り過ぎない

(モデル)

- ・事例ベースモデル
- ・等、不等式の制約問題解決モデル

【2】利用可能な技術と理由

○事例ベース推論

- 1から設計するのは非効率で現実ともマッチしないため。
- リデザインや設計変更の際に過去の設計事例を目安とするため。

○E B L

- 設計手順をチャック化するため。
- 設計効率化のための知識を獲得するため。

○制約問題解決

- 宣言的に記述された等、不等式から設計手順を決めるため。

○仮説推論

- 修正・変更の際の、パラメタ間の論理的な依存関係を効率的に管理するため。

【3】知識獲得の方法

第一に、事例ベース推論の方法を取り入れることを考える。

事例ベース推論には、現問題に、他の似たような問題解決事例を当て嵌めて問題解決を図る類推的側面と、探索ヒューリスティックスの代わりに事例を利用する推論効率化の側面、さらに、過去の変更事例を用いて目標内に収まるようにパラメタを変更する分析的側面があると考えられる。これらのいずれのタイプの場合でも、知識ベース上に抽象化されたルールが無い状況を問題設定としている。過去の類似経験に基づいて新しい問題にアプローチし、その結果がまた、経験として蓄積されるという事例ベース推論の方法論はそのまま知識獲得の1つの方法論であると言える。

事例ベース推論を用いる場合、問題構造の記述とインデクシングをいかに有効に行えるか、現問題と過去の問題とのマッピングをいかに取るかが重要である。特に類推的側面を強調した事例ベースシステムにおいては、参照される事例側には十分な解決

方法に関する記述がなくてはならない。ここでは、まず最初に何か問題点があり、それをいかに解決したかについての事例が十分に記述できる状況を仮定する。知識獲得の側から考えると、白紙の状態より（上手に解けてはいない）解の例を用いて専門家に解決方法をインタビューするほうが容易であると考えられるのでこのような状況設定には意味があると考えられる。この場合、

- ・ 提示した解の問題点は何か（不都合な状態の定義）
- ・ 問題点の原因は何か（不都合な状態を引き起こすものの因果ツリー）
- ・ 解決案

を表現することが大切である。この表現を得るためには、部品とその機能に立脚した原因・結果に関するインタビュー技術がキーとなる。設計対象品の機能系統図はそのための有力な情報を提供してくれる。このような状況設定ができれば、CYCLOPSで用いられているマッチング手法であるdemandposting法の有効利用ができる。問題点の原因によるマッチングを行う場合、原因の一般化、現問題と過去の問題との対象物のマッピング、解決方法のマッピングを行わねばならない。このための基礎となる理論としては構造写像理論が上げられる。

第二に、知識ベース上の知識の充実を図るためにE B Lを利用することを考える。

事例ベース推論モジュールを構築するだけでは不足すると考えられる事柄は、事例を用いて別の問題が解けたとしてもその論理的裏付けができないことである。論理的裏付けを得るために、基本的な領域知識とE B Lの方法によって、インタビューを交えながら解決事例の説明構造を作成し知識ベースに蓄えていく。

2. 5次元金型設計の場合、以下のような基本設計段階を事例ベース推論とE B Lで処理する。

- ・ 要求仕様からデフォルトの金型モールドを選定する部分
- ・ デフォルトの金型から個別の金型に修正してラフデザインを行う部分。

選定や修正が失敗した場合を元にして、解決事例を作成する。

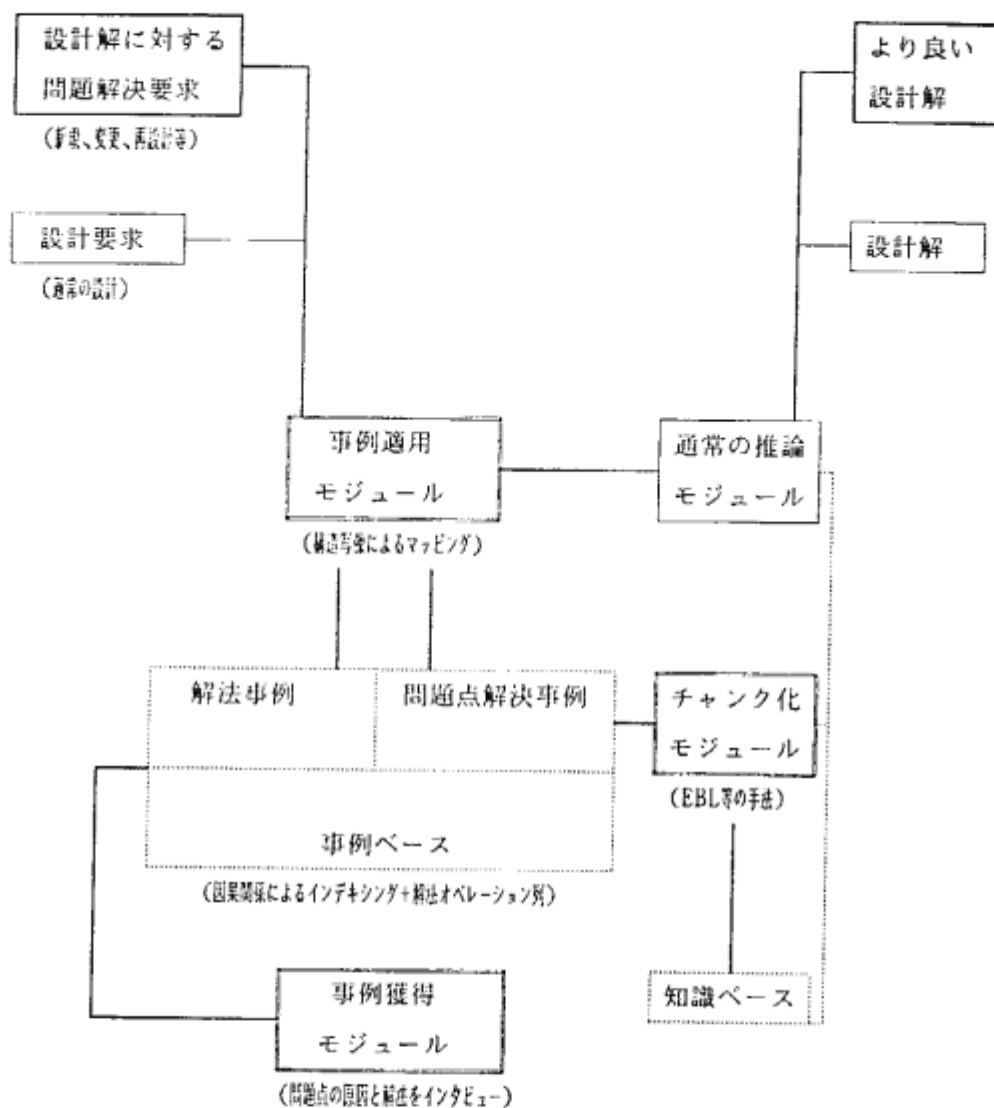
【4】設計問題の問題点/知識獲得の問題点

- ①シミュレーション、記号知識化の困難な領域が多く定式化が難しい。
- ②設計過程、特に試行錯誤過程を系統的に事例として残していない。

同様に、要求が厳しくなった場合の変更設計、新製品の開発設計の系統的な事例が残っていない。修正変更は制約緩和の問題と結び付いており、制約問題解決の視点からはメタな問題となるため知識表現が難しい。

- ③設計者はイメージ情報を有効に活用していると考えられるが、現在の記号処理のアプローチがイメージ処理とA I的に結合されていないので知識獲得が困難である。
（物を見ながらインタビューを進められない）

【5】設計向き知識獲得システムのイメージ



1. 注目する設計問題とそのAI的特徴／モデル

注目する設計問題：

アナログ系半導体のパターン及びプロセス設計

特徴：

- ・シミュレーションベースの設計作業である。入力された設計データ（パターンデータ／プロセスデータ）から、シミュレーション（半導体モデルの作成とその解析）によってはじめて制約充足を判断できる情報が得られる。
- ・しかも、一回のシミュレーションだけでは制約の充足を判断できず、目的毎のシミュレーション結果間に定性的制約関係がある。
- ・大局的な制約充足のための経験的定性的知識が存在する。
- ・形状（パターン）と数値（プロセス制御）をペアで扱うための知識が必要である。
- ・パラメータの数が膨大で、すぐに組合せの爆発を起こす。

モデル：

生成検査モデル

2. 利用可能な技術と理由

技術：

- ・制約解析
シミュレーション結果の統計的分析から得られる、制約と入力パラメータ間の依存関係を利用する。
- ・事例推論
- ・定性的推論

理由：

- ・制約をあらかじめ関係式の形で表現できない。
- ・基本プロセス毎に制約とパラメータの依存関係が変化するが、一度基本タイプが設計できると、その変形で次の設計が可能になる。

3. 知識獲得の方法

獲得の対象：

解がない時に、制約を満たすために変更すべきパラメータに関する知識。最終的に、シミュレーションのパターンと制約条件、そしてパラメータの依存関係を獲得する。

獲得方法：

設計者がシミュレーションの結果に満足できない時（制約条件を満足していない時）に取る改良案とその理由を対話的に獲得する。

さらに、それに基づいてシミュレーションを行ない、結果に対する影響の強さ・効果などを専門家から求める。

理由：

- ・物理モデルだけからでは、制約とパラメータの依存関係を求めることが困難である。設計者から経験に基づく知識を獲得し、効率を上げる必要がある。

4．設計問題の問題点 / 知識獲得の問題点

- ・改良案の取捨選択

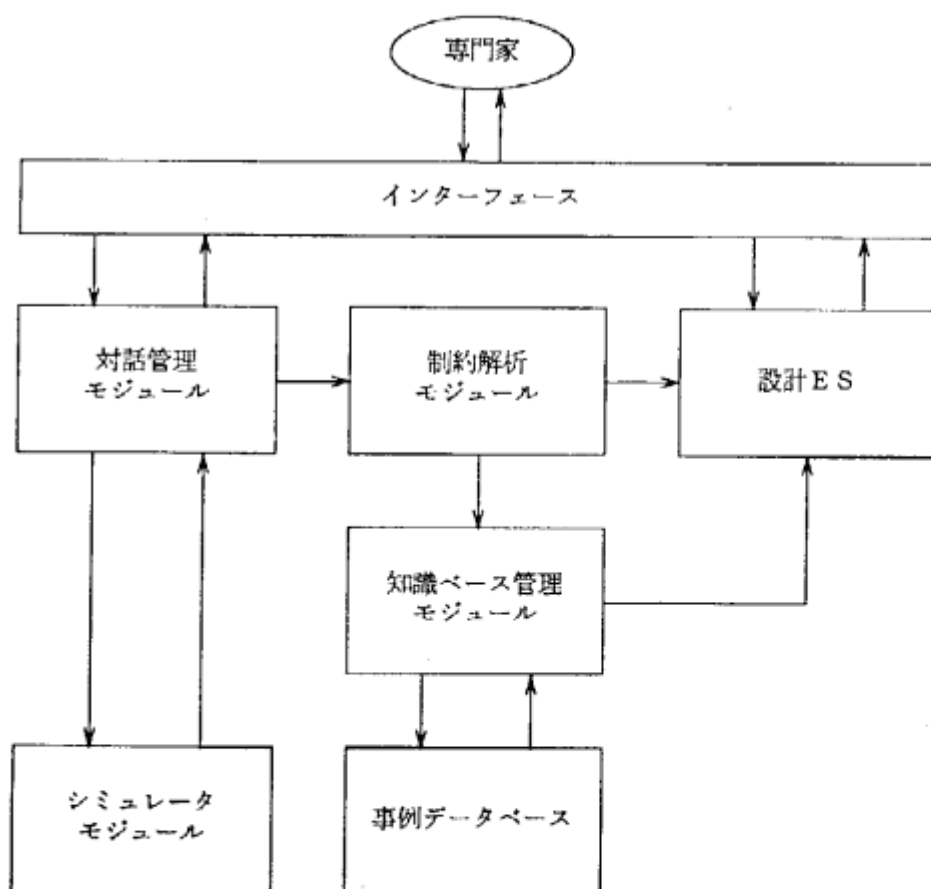
設計者は試行錯誤の過程でバックトラックすることはほとんどなく、常に新しい試みを行なう。このため、個々の試行過程よりも、一連の過程をマクロなレベルで捉える必要がある。

- ・知識の一般化をどのようにして行なうか

変更できるパラメータの多さゆえ、設計者により、問題解決のアプローチが異なる。これらの知識をどのように一般化するか、あるいは複数あった場合にどれを優先するか。

5．設計向き知識獲得システムのイメージ

知識獲得のために必要な機能としては、一連のシミュレーションを管理し、依存関係を抽出するための対話管理モジュールと、制約生成するモジュール、また設計データを事例として蓄積する事例データベース、事例を検索するモジュール等が必要になる。



1. 注目する設計問題とその A I 的特徴 / モデル

1) 注目する設計問題

設計問題には機械設計問題、回路設計問題、レイアウト設計問題等、色々なドメインが存在する。ここでは、プログラム設計問題に注目して論じることとする。

2) プログラム設計問題の A I 的特徴

プログラム設計問題は、ある要求仕様を入力として与えることにより、計算機が、その仕様を満足する動作を行える様なプログラムコードを生成する問題である。この問題においても、いわゆる発想的設計と、ルーチン設計とが存在する。発想的設計とは、設計者がプログラム言語の文法に従い要求仕様を満足させるべくアルゴリズムを考えながら設計していくものである。一方、ルーチン設計とは事前に用意されたプログラム部品を選びだし、それらを適切に結びつけ、要求仕様を満足させようとする方法である。ここでは、このルーチン設計問題について述べる。

< 特徴 >

- (1) 要求仕様が定性的に与えられることが多い
- (2) 部品の機能記述も定性的表現が多い
- (3) 部品の選択において大別して次の二種類がある
 - a) 各部品の入出力関係の整合性を保ちながら適切な部品を結合していく
 - b) 要求仕様に近い類似例があり、これを基にブレイクダウンしていくここでは b) のアプローチをとる
- (4) 選択された部品が妥当である事を検証するなんらかの方法が必要である

2. 利用可能な技術とその理由

(1) C B R (Case based reasoning) の類似例の検索技術

要求仕様とマッチした類似の事例があるかを検索する。この時妥当な類似度の判断と効率の良い検索能力がポイント

(2)階層化設計（トップダウン精密化法）技術

抽象的な要求仕様（ゴール）から、より具体的な要求仕様（サブゴール）列への展開

(3)制約緩和技術

類似判定条件、部品適用条件、パラメータ設定条件、パラメータ結合条件等の制約緩和の戦略と、緩和した時の推論制御（知的バックトラック）

(4)仮説推論技術

ある部品選択を一つの仮説とみなし、それに基づいて詳細に展開した部品列を生成し検査し、その仮説の妥当性を、検証するという生成検査法の枠組みの中で利用される。

(5)実行タスク（シミュレータ）の評価技術

(4)における検査に実行イメージの部品が実行され想定した結果（そのレベルでのゴール）が得られたか否かを評価する。この時のポイントは以下の通りである。

- ・各部品は、単体で実行可能な事
- ・実行結果をいかに推論系が判断できる情報にするか（人間が介入するか）

(6)推論探索技術

通常の探索技術に加え制約の枝切りが発生する前に、その枝に対して見切りをつける技術（状態評価における先読み機能 ??）

3. 知識獲得の方法（トップダウン精密化における部品選択ルールの獲得）

1) 編集型知識獲得

- ・システムが推論して部品展開を行ってきた過程を保持し、推論に失敗した事例に対して、その過程を辿ることにより、どのルールが起因しているかを専門家に尋ね、そのルールの部品選択条件部を修正する。
- ・同じ部品（部品列）を選択しているルールを集めその選択条件部を比較しその整合性を専門家にチェックさせる。

以上のように診断型の獲得ツールである TEIRESIAS や MORE の枠組みが利用できる。また、ソフトウェアのデバッガ的観点からすると上記の操作が

break-pointから任意に行え（推論の中断）、知識修正した後、任意の Break-pointからの推論開始（推論の再実行）が行える環境が必要である。

2) 学習型知識獲得

a) EBLの利用

あるゴールを達成するためのブレークダウンされた処理列を得るために EBL が有効である。

- ・学習概念：設計仕様
- ・領域知識：断片的な処理の知識
- ・操作規範：実行可能な処理名
- ・学習事例：ある問題を解いた実行列
- ・一般化：事例の処理列から余分な処理を除く

b) ABL（類推による学習）の利用

上記 a) においては、部品選択ルールの "t h e n 文" が獲得されたことになる。そこでこのルールが選択されるための条件文を獲得する必要がある。既に得られているルールの t h e n 文と EBL で獲得された処理列とを処理の意味ネットを利用し類比をとり、最も類似度の高いルールを求め、その類比でそのルールの i f 文を変換し、EBL で求めた処理列の条件文とする。

4. 設計問題及び知識獲得の問題点

<設計問題の問題点>

- 1) 仕様の記述が抽象的である。
- 2) 類似例の検索キーの設定が難しい。
- 3) 実行タスク（シミュレータ）の結果評価の仕方が難しい。
- 4) 制約緩和方式（優先度、タイミング、上位への知的バックトラック）
- 5) ブレークダウン打ち切りの指針（設計途中結果の評価方法）

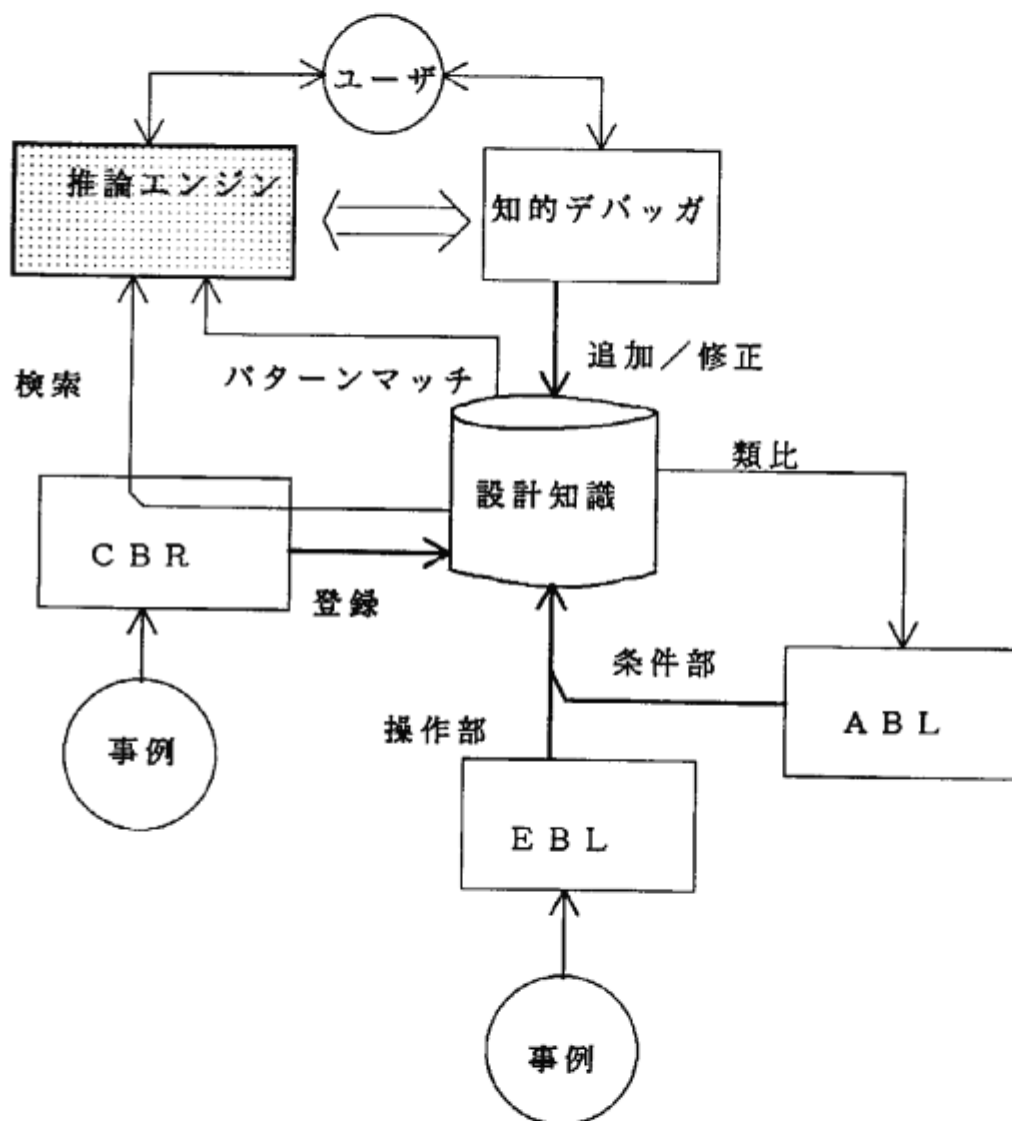
<知識獲得の問題点>

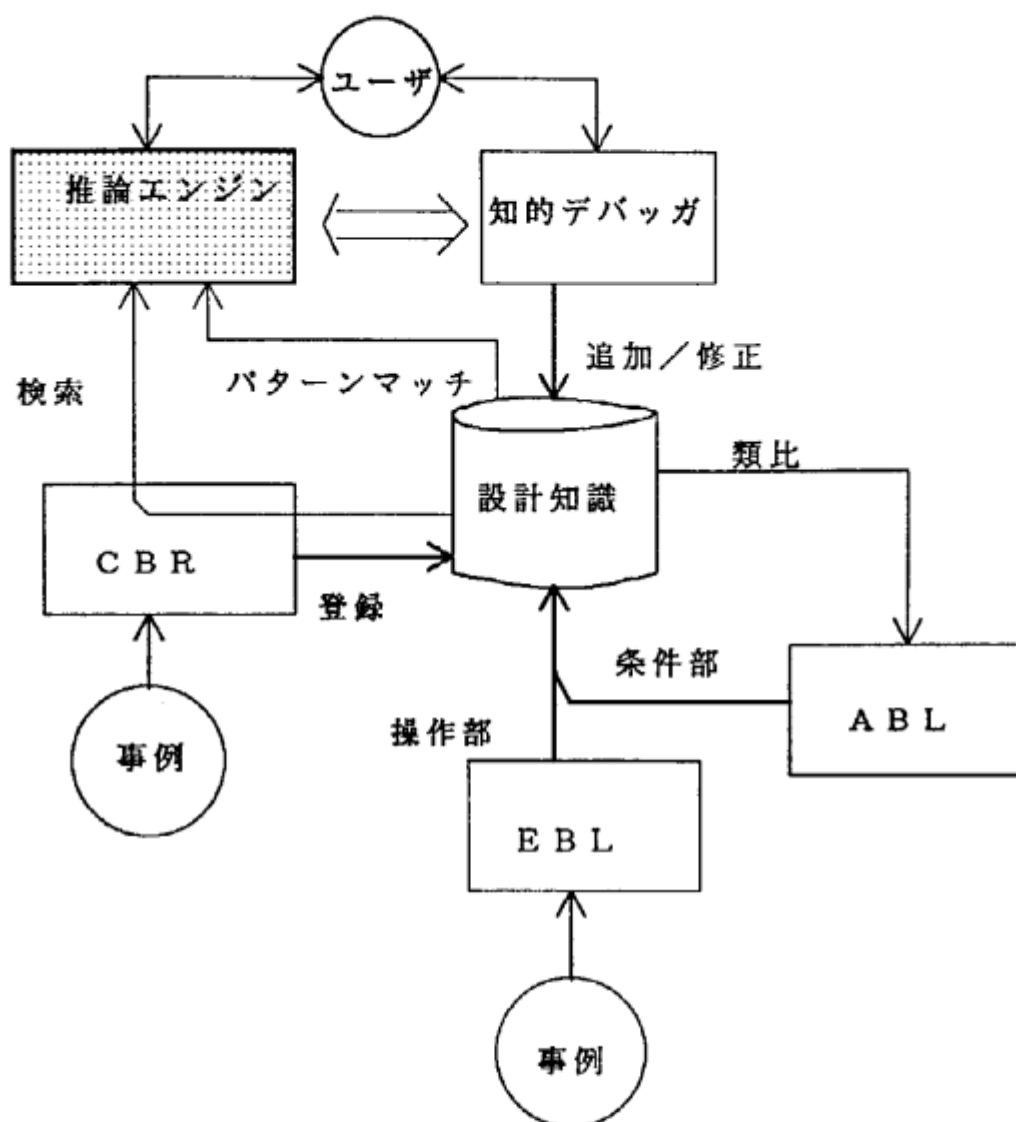
- 1) 学習概念（設計仕様）の記述レベル、表現の仕方の設定が難しい。
- 2) 設計事例を説明する領域知識の獲得を如何にするか。

3) 獲得した知識の検証を如何にするか。

5. 設計向き知識獲得システムのイメージ

設計向き知識獲得システムのイメージを以下に示す。これは、前記した知識獲得方法に、類似設計事例を得るために、C B R 機構を加えたものである。





設計問題へのアプローチ

九州大学中央計数施設 長澤 勲

1. 知的CADに期待する機能

- ① 従来のシステムに比べて設計知識を明示的に取扱える（設計知識の陽な表現，深い知識の表現，知識の管理）。
- ② 情報処理技術者の手を煩わすことなく，設計者自らが設計知識を維持・管理できる（設計知識のデバッグ，設計知識のコンパイル，設計知識の説明，カスタマイズ，CAD開発の分業化）。
- ③ 設計者が，設計知識を整理・獲得する過程を支援できる（企画，品質管理，実験室支援）。
- ④ 設計対象を取り扱う観点を管理する（視座制御，設計検証）。

2. 設計者の知識獲得の支援 … 企画・設計・生産・IEの共同作業（OA的）

(1) 概念的情報の収集，整理

設計作業の上流（企画，許容設計，概念設計） … 設計者の概念獲得支援

- ① 企画の支援 … （例）岸本の創造的設計
- ② 改良設計の支援 … （例）品質機能展開

(2) 実験的情報の収集，整理

製品の試作，シミュレーション → 設計の改良

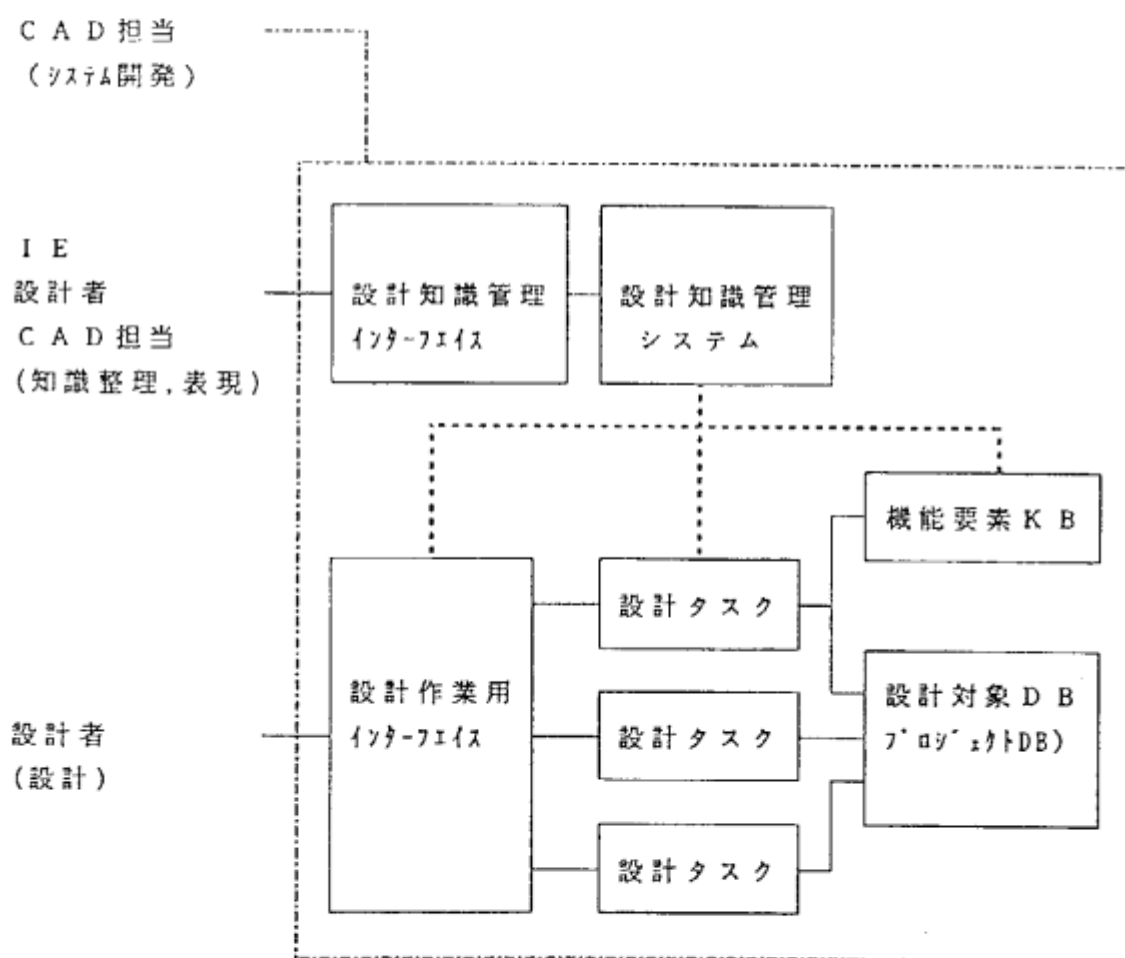
- ① 実験室支援（データ解析，工学モデル自動生成）
- ② 設計情報の蓄積管理（IE的方法） … （例）遠藤の類似設計

3. 設計作業の合理化

設計作業の中流（基本設計，詳細設計，生産設計）

定型設計 → 省力化，迅速化，最適化（C A D，設計E S）

新規設計 → 設計の多様化，高度化（設計方法をまとめる過程の支援）



知的C A Dの枠組

- (1) 機能要素 K B … 設計対象の構成要素
 - ・対象領域（建築，板金，製缶等）ごとに，どのような要素と属性を設定すれば良いか。
 - ・開発・改良設計では，構造や属性が設計途上で変更されることが多く，対象の型が不定である．この場合のモデル化手法は？
- (2) 設計対象 D B … 設計途上の対象の表現と管理
 - ・設計対象の一元管理（多目的性，ビュー変換，連動修正）
 - ・D B 機能（アクセス管理，バージョン管理，透明性，隠蔽性など）
- (3) 設計タスク … 設計作業ごとの支援システム
 - ・自動化の程度
 - ① O A … 設計の意志決定・設計案の解析などすべて設計者が行なう。
（例）製図支援，設計資料検索，設計図面 D B
 - ② 設計案の記述表示 … 形状モデル + C G
（例）住宅の意匠，プラントレイアウト（A⁺-A⁺ドラフト，プラスチックモデルの置換
 - ③ 工学的解析を中心にしたシミュレーション
 - ④ 半自動設計 … 設計作業（設計案の提案・解析・評価・修正）の代行，結果を設計者が評価・修正
（例）自動設計・最適設計システム，E S
 - ⑤ 自動設計 … 完全自動化
（例）自動設計・最適設計システム，E S
 - ・設計の流れ
 - ① 上流（企画，許容，概念設計） … O A 的支援（人が主）
 - ② 中流（基本設計） … コスト／性能のための探索
 - ③ 下流（詳細） … 取り合い，加工，法規など詳細情報の付加
- (4) 設計作業用インターフェイス … 設計作業用言語
 - ・設計図による設計作業（スケッチパッド）
 - ・計算書による設計作業（スプレッドシート）
 - ・設計情報検索（実績，カタログ，法規，設計式など）
- (5) 設計知識の管理
 - ・記述言語，虫取り，説明
 - ・知識獲得技術
 - ・C A D の C A D（設計手順の整理の支援）

設計診断知識の獲得

大阪大学産業科学研究所 山口 高平

1. 注目する設計問題とそのAI的特徴／モデル

設計診断（設計時にその対象物の診断も行うこと）

特徴：設計ミスは、故障が起こってからでないと分からない

モデル：引き出される「設計ミスを調べる知識」に依存するが、

物理知識では、定性値伝播や定性推論技術の利用が考えられ、

化学知識では通常のルールベースの処理となるが、これらの知識を連続的に利用する統合化された推論技術が重要となる。

2. 利用可能な技術と理由（問題解決としての視点）

・モデル構築技術

理由：物理式上で以下の技術を用いて、対象の過去（or未来）の状態を調べようとするとき、まず解析モデルを作成しなければならない。

・定性値伝播

理由：モデル構築が成され、かつ物理パラメータが一度変化したらその後変化しないと仮定すれば、物理式のパラメータの因果関係に沿って、対象の過去および未来の物理状態を大まかに解析することができる。

・定性推論

理由：モデル構築が成されれば、propagationとpredictionを繰り返すことにより、対象の将来の動的挙動を予測できる。ただし、現実的に無意味な状態を刈り込む戦略を整理する必要がある。

・IF-THENルールの処理

理由：化学関係の知識は、変化前の状態と変化後の状態が因果関係としてIF-THENルールで書ける。

3. 知識獲得の方法（何を獲得するのか、方法は、有効な理由）

設計診断知識の獲得

- <方法①> 過去の設計ミス事例をドメイン理論を修正しながらあるいは既に整理された故障モデルを適用しながら再理解し、結果を一般化およびマクロ化して、 I F feature（構造上の特徴）
T H E N 可能性のある故障 という設計診断知識を得る

- <理由①> 設計のためのドメイン理論だけでは、設計ミス事例を理解できないことがあるため、修正されたドメイン理論を利用する必要がある。さらに、この修正されたドメイン理論と故障モデルの相互作用により、設計ミス事例を総合的に理解できると考えられる。

- <方法②> 対象モデルの制約を故障の起こり易いように緩和して、定性値伝播あるいは定性推論によりWhat happen if ～の予測を行い、設計者と対話しながら、 I F feature（構造上の特徴）
T H E N 可能性のある故障、という設計診断知識を得る

- <理由②> 設計者の先入観が誤っていて、設計ミスが起こる事例が頻繁にある。そこで、対象に課せられた制約を緩和することを通して、この種の誤りを見つける。

4. 設計問題の問題点／知識獲得の問題点

- ・基本的に、設計ミスは故障が起こった時点で初めてミスと分かる場合がほとんどである。
そのため、過去の設計ミス事例を一般化して、広く設計ミスに備える知識を得ることが最初に考えられる。
- ・しかしながら、上記のアプローチでは、未経験のミスに対処しきれない。
そこで、設計上課せられている制約を緩和して、対話を通して設計者の援助を得ながら、未経験の設計ミスを設計段階で見つける必要性が出てくる。

5. 設計向き知識獲得システムのイメージ

図1のような中ロール昇降シリンダを設計したとする。そして、方法①によって、過去の設計ミス事例を一般化したレベルの設計診断知識では、設計ミスは捉えられなかったとする。対象に課せられた制約はいくつも存在するが、「中ロールは下降した時に下ロールと接触して制止する」という制約を緩和して、「跳ね返りが起こる」と変更する。そうすると、定性推論により、「跳ね返り発生→跳ね返りの阻止→シリンダ内の高圧水が加圧される→強度の弱い部品が破損」という故障シーケンスが作成され、設計者に提示することにより、同意が得られれば、「IF 中ロールは下降した時に下ロールと接触, THEN 強度の弱い部品が破損」という設計診断知識が得られる。さらに、設計者との対話を通して、図2のように2つの流量調整弁を付加することによって、設計を変更したならば、「IF 中ロールは下降した時に下ロールと接触, THEN 2つの流量調整弁を付加する」という設計知識も得られる。

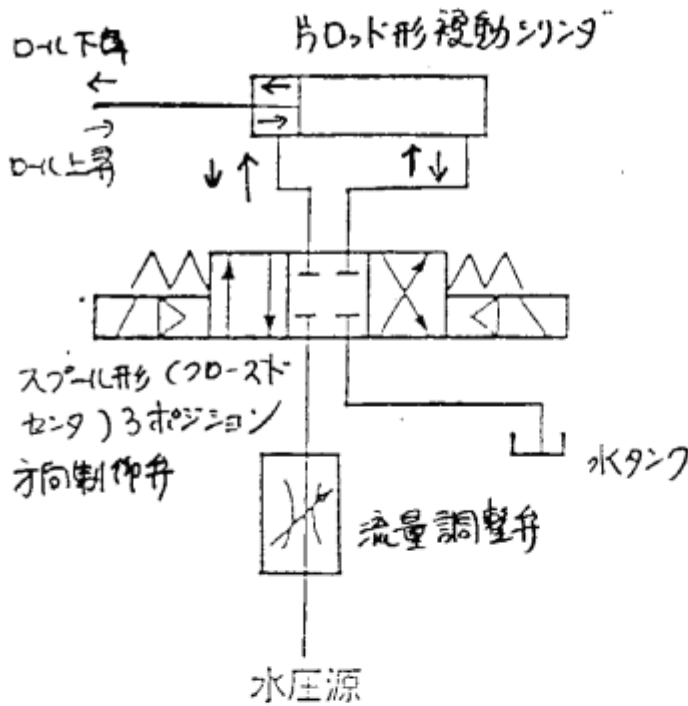


図1 改造前

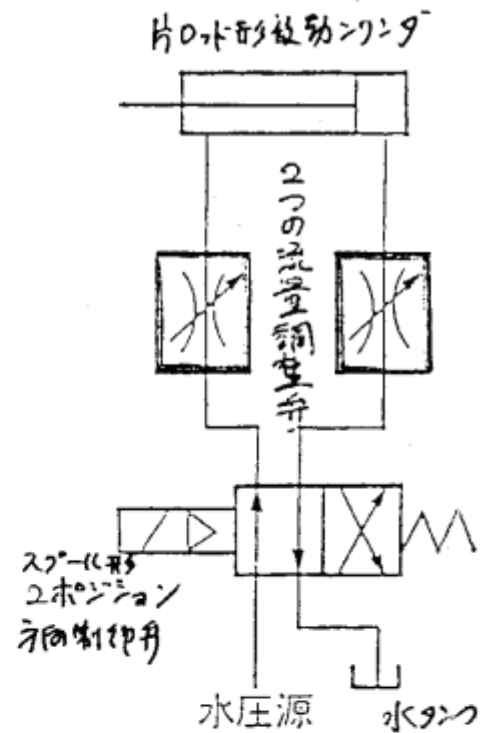


図2 改造後

大阪大学産業科学研究所
安部憲広

1. 注目する設計問題とそのAI的特徴／モデル

機械構造の決定

特徴：既知部品集合を用いて目標機能を達成する部品間結合を探索する

モデル：要素機能達成知識と機械要素機能を用いた組合せ問題解決及び事例推論

2. 利用可能な技術と理由（問題解決としての視点）

部品集合が既知であるから、部品同士の結合可能性は単純な組合せ問題となるが、要素機能の考え方を導入すると組合せの可能性は減少する。可能な候補の組み立てシミュレーションの成否を調べることにより、実際の組み立ての制約（手順）が得られる。成功、失敗事例のインデックスを、機械要素のクラス知識及びパージョンスペースの考え方により改訂していく。また具体例の機械要素の一般化、類似例の知識ベースを利用することにより、部品集合既知という制限も緩和可能となろう。

本手法を用いることにより、ある機能を達成可能な部品集合とその組み立て関係の知識が獲得可能となるが、複雑な機能はより単純な機能の組合せから作られるよう事例をうまく選択してゆかなくてはならない。

実際に適用する技術要素はGTD（組み立て生成、機能達成検証、正しい組み立て生成）及び事例推論である。

3. 知識獲得の方法

<対象>

要素機能の組合せにより、実現されうる機能に関する知識ベースの整備。要素機能に着目した機械要素の分類知識の獲得。

<方法>

多分機械の専門家及び教科書から抽出可能。

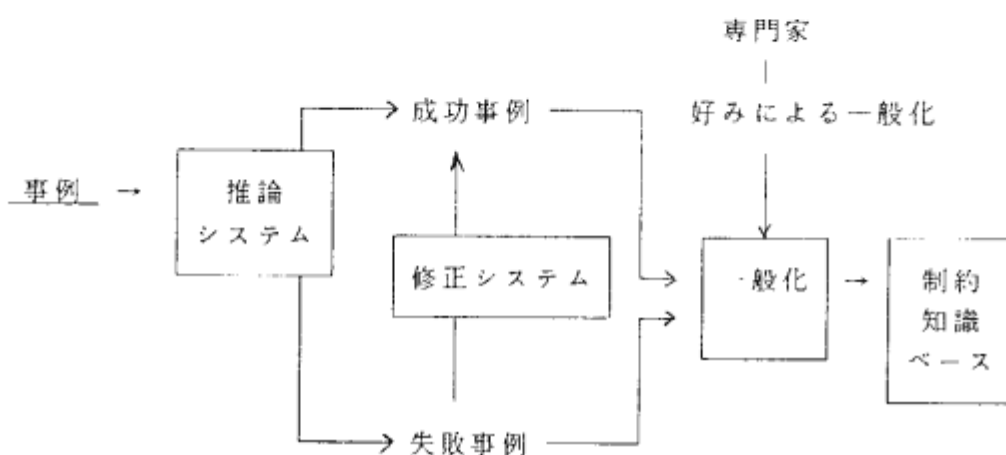
4. 設計問題の問題点／知識獲得の問題点

機能が与えられたとき、部品の選択には設計者の好みなどが大きく影響し、そのために構造も変化する。そこで本報告ではまず部品を固定し、構造を決めることを考えた。部品はその後で制約を緩和していくことにより適用可能性の枠を拡げてゆけばよいのではないか。

5. 設計向き知識獲得システムのイメージ

具体例を処理できる事例が1つ得られたら、自動的一般化を行なうべきであるが、現在の技術レベルでは、あらかじめ何らかのバイアスを与えておかざるを得ない。しかし、それでは設計者が持っている自由な発想によるのとは異なった概念をシステムが学習する可能性がある。

そこで設計者は成功事例や失敗事例を各人の考えに従って修正し、システムがそれらと同化できるようにすることが望ましいのではないか。そうした知識がいつも成功しない場合、システムはそれを無視すればよい。



0. 日本の機械工業をとりまく環境

・並級以下の機械製品の競争力は全くない。

――設計手順だけは確立されている（知識・根拠は陽的でない）。

・高度な（高精度、・・・）機械製品の設計への要求。

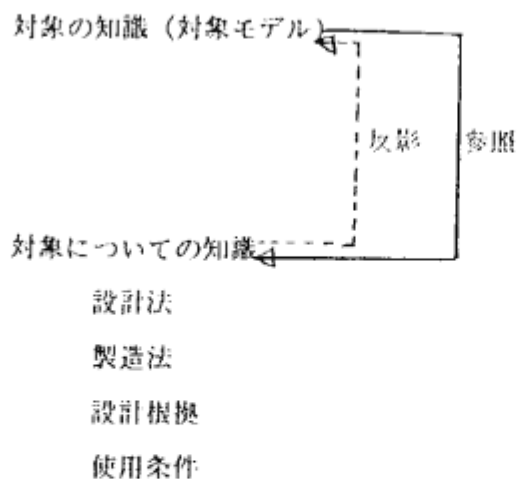
1. 知識獲得の目的

獲得された知識の再利用

人間の機械設計エキスパートは7～10年かかって一人前になる。

（通常OJT）

2. 設計一般における知識

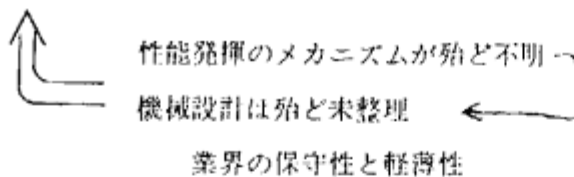


3. 機械設計の知識の特徴

- 特定製品依存で個別性が高い
ある製品のどこを変えたらどういう結果が出るか（殆どデータ）
（設計者個人の記録）
- 比較的一般的な「べからず」ルール（評価型知識）になる
（設計上しない方がよいことを具体例について表現するルール）

4. 機械設計の特徴

- 設計解の任意性が非常に高い
- 汎用的な設計規則（生成型知識）は存在しない
- 全体性・全体についての感覚的バランスが重視される
（ベテランの感覚はなぜか正しい）
- 機械要素は確定的であるが、設計のポイントはそれ以外の部分にある



5. 技術的課題

⑤ 獲得されるべき知識表現のための枠組み

対象表現の技術

-
- A vertical double-headed arrow with two text items in the middle: '・ Featureの利用' and '・ CONMOTO型設計整理'.

設計対象表現のための対象分野知識の獲得

「深層的および連合的領域知識を用いた一般化手法による概念設計知識獲得」

1. 注目する設計問題とそのA I的特徴／モデル

概念設計・機能設計問題、但し 1st Look（新規設計）と 2nd Look（改良設計）両方の場合について注目する。

特徴： 計算機としての支援の範囲が限定されており、設計問題解決の主体は設計者の側にある。したがって、人間と機械の協働（協調）的作業環境の整備が最重要な課題となる。また、過去に行われた設計事例、設計物に関するデータ、経験が重要な参照資料であり、推論を制御する重要なファクターになる。したがって、これらを参照しやすく、かつ活用しやすい形で documentation, indexing し、保存する必要が生じる。

モデル： これらの設計過程のモデルを明確な形で与えることは困難である。一つの考え方としては、吉川らによる「一般設計過程モデル」¹⁾がある。これは、設計物あるいはそれを構成する機能・属性の成す空間に対して位相概念を導入することによって、設計過程の意味づけを与えようとするものである。いま一つの考え方として注目すべきものに、Suhらによる「公理的設計法」²⁾がある。これは、設計過程を場当りの決定的積み重ねとして捉えるのではなく、設計の際に守られるべき公準（設計公理）を予め設定しておき、これから演繹的に導かれる定理群によって設計過程における推論をガイドしようとする立場に立つものである。また、2nd Look の場合には、価値工学（Value Engineering）における機能分析（Functional Analysis）を組み込んだ設計案改良法がよく用いられる。これは、設計物そのものよりもその背後にある考え方に注目することによって、より本質的な改良案の考案を計るものであり、具体的には、設計案に含まれる機能構成に注目し、これと実現に要したコストの対比を行うことによって、改良すべき箇所を特定を行う。勿論、この考え方は、1st Look の場合にも適用可能である。

以上を総括して考えると、設計物をその機能に注目して分析することが、概念設計支援に際しては重要な課題であることが分かる。

2. 利用可能な技術と理由（問題解決としての視点）

人間－機械システムによる設計事例情報の活用と事例ベース推論

理由： 1にも述べたように、事例情報が重要な情報源であり、その活用には人間と機械の協働が必要不可欠である。さらに、2nd Look の場合については、事例ベース推論が一つの基礎を与えるものと考えられる。

3. 知識獲得の方法（何を獲得するのか、方法は、有効な理由）

＜獲得対象＞

(i) 設計物に与えられた目的（基本機能）達成のために必要となる一般化された機能構成（目的の下位機能群へのブレイクダウンの形態）、機能それぞれを達成するための一般的構造・属性の形態、を抽出獲得するとともに、

(ii) 事例ベース推論の際に必要な一般的な知識を得るためには、設計物を規定する属性群から構成される空間上に何らかの位相構造が導入されている場合について、一つの事例の成立（設計物がある性質を満足すること）から、一般に成立が保証される性質につい

て、推論することが必要となる。

<方法>

(i)に対しては、図1に示すように、目的論、因果論の立場から見た設計物の捉え方を基礎として、EBL手法を用いて獲得するのが有効であると考えられる^{3),4)}。

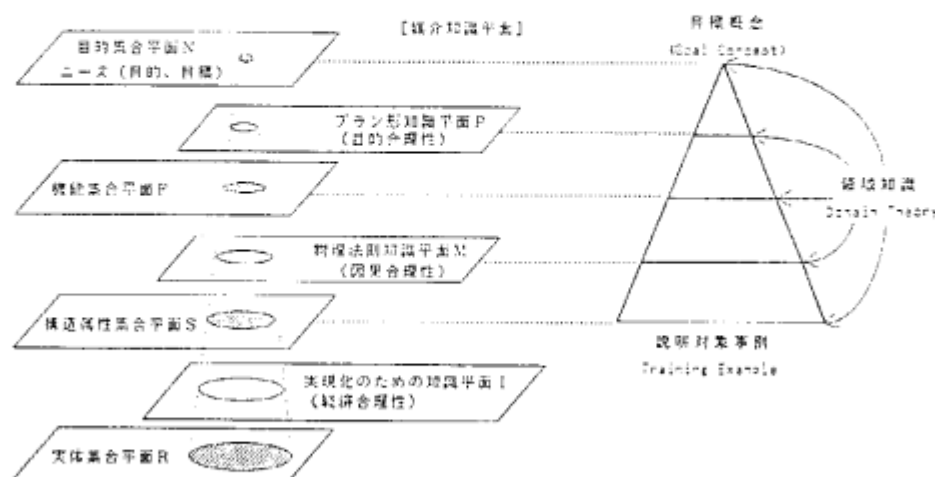


図1 設計物解釈のための階層的モデルと EBL

(ii)に対しては、属性空間上での位相が何らかの順序構造として与えられている場合については、図2に示すようなより広い推論・学習の枠組み⁵⁾を導入することによって、対応することが考えられる。図中、楕円で囲まれた領域はそれぞれ一つの順序構造によって規定された属性空間であり、これらの間の単調性・連続性関係を通して推論が展開される。

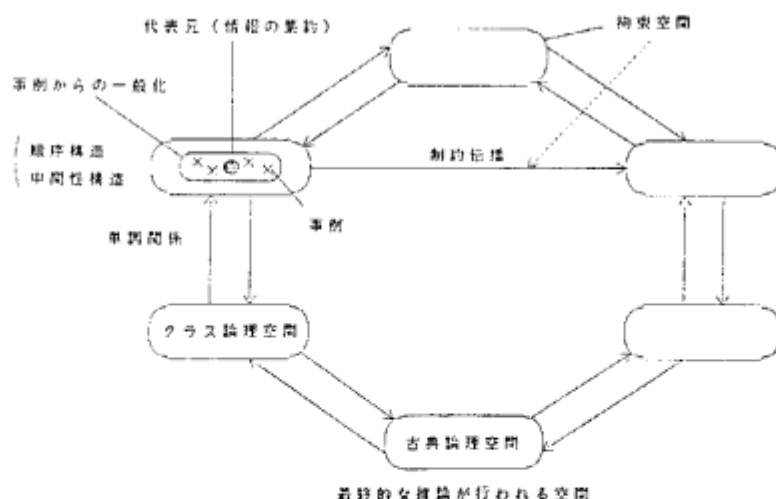


図2 順序構造に基づいた推論・学習の枠組み

事例の一般化・集約化は、背後にある順序構造に呼応した形で実行される。上の EBL が、

”深層的”領域知識に基づくものであるのに対して、”連合的”領域知識を用いた手法であると言える。

<理由>

(i): ”目的—プラン—機能—物理因果法則—構造・属性”といった形の深層的かつ多段階レベルからなる領域知識を用い、目標概念、操作性規範を様々なレベルに設定することによって、プランモジュール、機能モジュール、構造・属性モジュールといった単位型知識の抽出およびこれらの統合化に関する様々な設計知識を獲得することができ、またこのような説明木を設計事例に対する index として付与することによって、柔軟かつ効率的な設計事例参照と(ii)の事例ベース推論の際に最も適切な参照事例の特定が可能となる。

(ii): 順序構造あるいはあるものが他の二つのものの間に位置するという中間性関係といった、極めて基本的かつ普遍的な関係をベースとしていること、およびこれらを組み合わせた形で推論することが可能であることから、多方面に適用可能であると考えられる。

4. 設計問題の問題点/知識獲得の問題点

知識獲得問題を考える上での設計問題のむずかしさは、基本的には設計に関連する知識の空間が開いており、その範囲を限定することが一般に困難であること、さらに人間のもつ様々な常識推論・判断が暗黙のうちに活用されていることであると考えられる。したがって、これらを抽出・獲得することならびにその際に用いるべき知識を設定することも大変困難な課題となる。上に述べた EBL システムにおいては、用意する知識(領域知識)はできるだけ一般化・抽象化した形で与えておき、事例を活用することによって、これを特殊化し、現実性・操作性を帯びた知識に変換する方針を採用している。また、順序構造に基づく推論・学習システムにおいては、個々の順序それぞれ別個に空間を設定することによって、位相構造に関する知識をモジュール化し、これを柔軟に組み合わせることによって、知識の表現と推論を簡略化し、上の困難さに対処する方針を採用している。

5. 設計向き知識獲得システムのイメージ

概念設計レベルの支援を目的とした場合、1にも述べたように、人間—機械システムとしての協働性の高いシステム構築のための知識獲得が必要となる。この場合、このようなシステムを運用しながら随時知識を抽出・獲得してゆく方法が考えられる³⁾。獲得される知識の中には、設計問題固有なものではなく、人間—機械のコミュニケーションの際に必要な知識、例えば語用論的な対話の構成規則なども含まれることになる。

参考文献

- 1) 吉川: 一般設計学序説、精密機械、45, 1979
- 2) Suh, N.P., Bell, A.C. & Gossard, D.C.: On an Axiomatic Approach to Manufacturing and Manufacturing System, Trans. ASME, J. Engg. Ind., 100, 1978
- 3) 片井: 設計事例からの問題解決知識の獲得、ICOT-TM-577 「問題解決モデルと知識表現」、1988
- 4) 片井 他: 設計物からの EBL による階層的設計知識獲得、第9回知識工学シンポジウム資料、1989年3月(計測自動制御学会)
- 5) 片井 他: 順序構造に基づいた各種推論形態の定式化、第9回知識工学シンポジウム資料
- 6) 吉川 他: 実験設計学、精密機械、47, 1981