

事例ベースアプローチによる設計支援
～設計支援システムSUPPORTにおける事例の再利用～

仲谷 善雄 築山 誠 福田 豊生

三菱電機(株)産業システム研究所社会システム開発G

〒661 兵庫県尼崎市塚口本町8-1-1

e-mail: nakatani@soc.sdl.melco.co.jp

SUPPORT は標準機械製品の設計に事例ベース推論を適用した事例ベース型設計支援システムである。客先の要求を仕様に変換する仕様設計、仕様を動作順序を考慮した機能系統図に展開する機能設計、および機能を実現する部品選択の各段階で、事例ベースに記憶した過去の類似設計事例を再利用できる。本稿では、事例ベース型設計支援における事例の再利用の枠組みを提案するとともに、意味記憶に基づき多視点からの仕様、機能、部品の探索を実現する類似設計事例探索方法を提案する。

Engineering Design Support by Case-based Approach
～Reuse of design cases in SUPPORT～

Yoshio NAKATANI Makoto TSUKIYAMA Toyoo FUKUDA

Industrial Electronics & Systems Lab., Mitsubishi Electric Corp.

SUPPORT is a case-based engineering design support system which reuses past design cases at any steps of design including the specification development, the functional design, and the parts selection. This paper proposes the framework of how to reuse past design cases and the analogical case retrieval method based on the semantic memory.

1. まえがき

本稿では事例ベース型設計支援システムにおける事例の再利用枠組みおよび類似事例の探索方法を提案する。

事例ベース型設計支援は設計業務の中心的支援方法として事例ベース推論を採用したものである。設計者は過去の設計事例を参考にしたり修正／再利用することにより、新規に設計する手間を少なくすることで効率よく客先からの設計要求に対処できる。

事例ベース型設計支援でもっとも重要な機能のひとつは設計者が望む事例を的確に探索できることである。もし設計者が指定した通りの事例がなくても適当な類似事例を探索できる必要がある。本稿では設計者の探索意図を反映した類似事例探索方法を提案する。

以下では、最初に事例ベースアプローチの特徴を整理した後、事例ベース型設計支援システムSUPPORTの概要を述べ、事例の再利用枠組みおよび類似探索方法を提案する。

2. 事例ベース推論

事例ベース推論(Case-based Reasoning)は過去の問題解決事例を修正／再利用することによって新しい問題に対処する方法論である[1]。法律判断[2]や紛争解決[3]などのように、同一ではないが類似の問題が多頻度で発生し、一般的な解法が知られていない悪構造の問題領域で有効である。このような領域の問題には従来のエキスパートシステムの方法論ではうまく対処できないことから最近注目されている。

CBRの特徴としては以下のものがある。

(1)過去の成功例を再利用するため、問題解決効率がよい上に、結果の信頼性が高い。

(2)事例がそのまま問題解決のための知識となるので、問題解決を通じて自動的に知識を獲得できる。

(3)未知の問題に対しても、類似事例を参考にすることによりもっともらしい対応をすることができる。

CBRは非常に類似した事例(近似事例)が存在する場合には非常に強力である。しかし以下のような問題がある。

(1)設計者の指定する属性を持った事例がない場合に、設計者の意図通りの類似事例を探索する有効な方法がない。

(2)探索した事例を新しい要求に適合するように自動的に修正するための知識を獲得することが困難である。

(3)事例ベースアプローチは近似事例が存在しない場合には一部のみが似た事例を設計者自身が大幅に修正しなければならない。このとき修正範囲がわからなかったり修正コストが新規解決より高くなる場合がある。

3. SUPPORTの概要

3.1 概要

SUPPORTは受注生産型の準標準機械製品を対象とし、仕様書作成、特殊仕様の機能への展開および機能の部品への展開をCBRにより支援する対話型設計支援システムである[4]。

準標準機械製品は、標準設計にオプション(あらかじめ用意された特殊設計)や新規の特殊設計を加えたり、場合によっては標準設計部分を修正することによって客先からの要求を実現する製品である。このような準標準製品の設計は設計者の

創造性や過去の設計経験に依存するところが多い。一般的に熟練設計者は過去の類似事例を有効に利用して特殊設計を行っている。しかし膨大な過去の事例からの探索は設計者に大きな負荷をかける。また設計事例に関する知識は設計者間で共有されていないため、経験の少ない非熟練設計者は設計に時間を要する。

本システムはこのような問題を CBR を採用した計算機システムにより解決することを目的としている。類似事例探索により過去の事例を再利用できるため新規設計の手間を少なくできる。また他の設計者の設計した事例を再利用できることから知識の共有化を図れる。できる限り自動化しているが、設計者の有能な助手を目指すという立場から、多くの部分はグラフィックエディタなどを通じてインタラクティブに実行できる。

3.2 システムの構成

図1にシステムのS/W構成を示す。

●問題解決エンジン … 設計型の事例ベース処理方略を整理したものである。

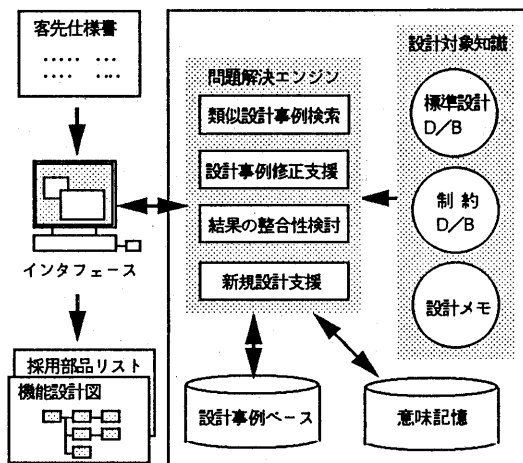


図1 SUPPORTの構成

●設計対象知識 … 標準の仕様・機能・部品、制約、設計メモ(マニュアルや設計ノウハウなどの文章)などの、製品に関する一般的な知識を記憶する。

●設計事例ベース … 過去の特殊仕様、機能系統図、採用部品、事例に固有の設計メモなどを事例名とともに記憶する。一般に設計事例は膨大になるので、年度ごとに事例ベースを分けている。

●意味記憶 … 対象領域を記述する概念を構造化して記憶する。

設計対象知識、設計事例ベース、意味記憶は領域固有の知識で、この部分を交換することにより、容易に対象領域を変更できる。また設計メモや宣言的な制約はオンラインで追加できる。

SUPPORTはEWS上にops83とC言語を用いて実現されている。設計事例ベース、設計対象知識、意味記憶を、現バージョンでは関係データベースにより管理している。

3.3 設計知識の表現

(1)仕様の表現

仕様は客先の要求を手持ちの仕様表現形式に変換したものである。通常は複数の仕様項目の組合せにより要求を表現する。各仕様項目を

$$[\text{仕様属性、値}] \quad (3.1)$$

で表現する。一般に準標準製品の場合には一定数の標準仕様項目があらかじめ用意されている(図2)。標準仕様項目で表現できない要求を特殊仕様として追加する。また標準仕様項目であっても値が特殊であれば特殊仕様として扱う。標準仕様項

正が可能である。

3.4 設計支援ステップ

設計者は仕様設計、機能展開、部品展開というステップを自由に往来しながら試行錯誤的に設計を進めることができる。各ステップの終了時に制約違反のチェックを行なう。制約には仕様／機能／部品の組合せに関する必要条件および排他的条件を表現したる宣言的制約と、数式表現される制約がある。制約違反があれば設計者は先行ステップなどに自由に戻って再設計できる。このとき制約DB内の修正知識が利用できる場合はシステムが自動的に修正する。

設計終了時に設計した機能を標準設計DBに記憶した機能関係構造の中に位置づけ記憶する。この機能関係構造は製品のすべての機能を関係づけて階層化したものである[6]。

4 事例の再利用

SUPPORTでは特殊な要求にどのような特殊仕様で対応するか、特殊仕様の機能への展開、機能の修正、機能の部品への展開、適当な部品がない場合の類似部品の使用、などの決定場面において過去の仕様／機能／部品を参照したり再利用できる。とくに設計の中心となる機能設計においては、任意のレベルの過去の機能列あるいはその一部をカットアンドペーストで再利用できる。

(1) 事例の再利用の枠組み

図4にSUPPORTにおける事例の再利用枠組みを示す。

●仕様名の決定…仕様設計においては、客先からの要求を手持ちのどのような仕様項目で実現するかを過去の事例を参考にして決定する。

●参照事例の決定…特殊仕様を実現する機能を過去の事例から探索するとき、仕様間に相互作用があることを考慮して、できるだけ多くの仕様が一致する事例の機能を再利用することが望ましい。本システムでは、事例間の一致度と相違度をともに考慮した類似度によって類似事例を探索し、適当な事例の機能システムをコピーして再利用する。

●機能の修正…コピーされた機能を新しい要求に適合するように修正するとき、過去の機能システムのすべてあるいは一部をコピーして再利用できる。

●部品の決定／変更…機能を部品に展開するとき、特殊な部品の場合には過去の事例の部品を再利用することができる。過去の機能を修正せずにそのまま再利用する場合には、その事例のすべての

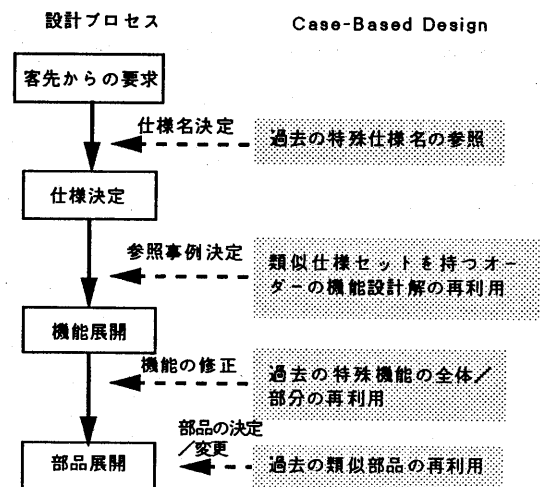


図4 事例の再利用方法

部品を再利用することもできる。

(2)仕様名の決定

過去の仕様を探索するときには、新しい要求の特徴をキーワードとする。過去の仕様にはそれぞれの特徴を表わすインデックスがつけられている。設計者が指定するキーワードと一致するインデックスを持つ仕様が目印の仕様である。

インデックスはフレーム形式で表現される。仕様は大きく3種類(電気関係、機械関係、意匠関係)に分類され、さらにそれぞれが4種類(特殊制御、特殊部品、特殊材料、特殊な値)に分類され、それぞれに固有のフレームが用意されている。設計者が探索時にキーワードを入力する場合にも同様のフレームを用いる。例えば特殊制御として分類される仕様を探索する場合のキーワード入力は図5のようなフレームで行なわれる。

探索結果としては、キーワードを満足する仕様とそれを採用している事例名が表示される。すべてのキーワードを満足する仕様がない場合には、満足するキー

ワードの数が多い順に提示される。

新規の仕様を追加するときには将来の再利用のため、どの仕様分類に属するかを判断し、その分類に固有のフレームに従ってインデックスをつける。ひとつの仕様を複数に分類しても構わない。

(3)参照事例の決定

どの事例の機能を再利用するかを決定するときには、新しい事例と仕様がもっとも類似した過去の事例を探索する。新しい事例C1と過去の事例C2の仕様の類似度 $s(C1, C2)$ を次のように定義する。

$$s(C1, C2) = (C1 \cap C2) / \{(C1 / C2 + 1)(C1 \cup C2)\} \quad (4.1)$$

ここで $C1 \cap C2$ は仕様の一致数、 $C1 / C2$ は相違数 $(C1 \cap C2) - (C1 \cup C2)$ である。このとき設計者は必ず一致しなければならない仕様項目を指定できる。

(4)機能の修正

(3)でコピーされた機能を図6のような機能編集エディタで修正する。画面には機能系統図の任意のレベルが表示され、そのレベルでのみ機能を修正できる。設

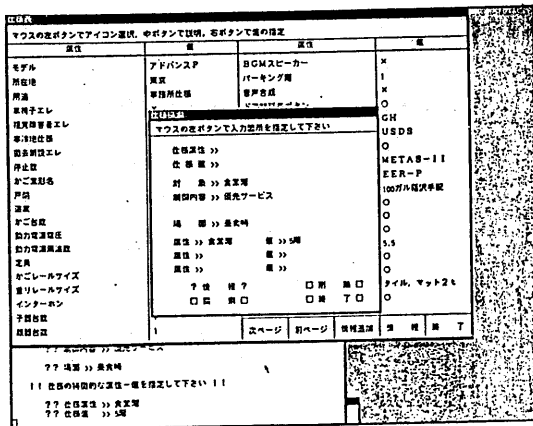


図5 類似仕様探索のためのキーワード入力画面の例

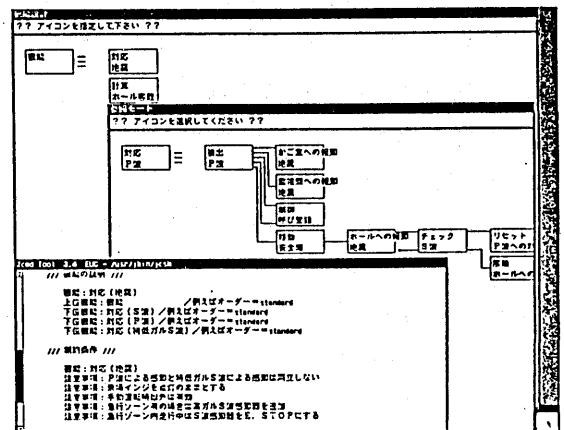


図6 機能編集画面の例

計者は表示されるレベルを自由に変更することで様々なレベルで機能を修正できる。設計対象知識に記憶された制約に基づく修正は自動的にシステムが行なうが、一般的には制約や設計メモを参考にしながら設計者が修正する。図6において背後の画面が編集画面である。編集画面上では機能の追加・削除・移動・名前の変更・属性値の変更・制約や設計メモの追加が可能である。

設計者は機能の修正に際して機能参照画面を通じて過去の機能設計事例を参考にすることができる。機能を参照するためには機能名とそれを採用している事例名を指定しなければならない。参照するための機能の名前あるいはそれを採用している事例名を類似機能探索によって知ることができる。図7は類似機能探索のためのキーワード入力画面の例である。上位機能名、下位機能名、付加キー(仕様を指定する)、機能名、設計月などのキーワードによって機能名および採用事例名を知ることができる。適当な機能名および事

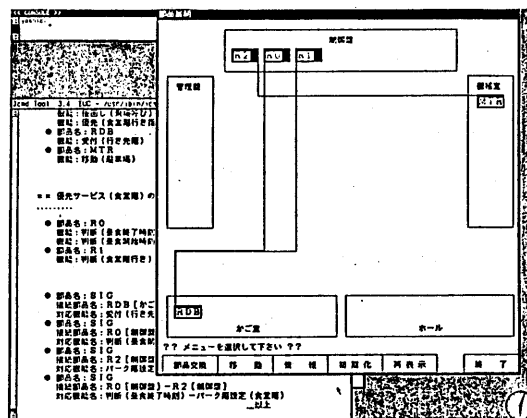


図8 部品レイアウト画面の例

例名を指定すると機能参照画面に表示される(図6において手前の画面)。設計者は機能参照画面上の機能のすべてあるいは一部を機能編集画面にコピーできる。

(5) 部品の決定/変更

部品を探索するときには部品が実現している機能名あるいは属性値をキーワードとする。探索の結果それらを満足する部品名とその採用事例名がリストアップされるので、図8に示すような部品レイアウト画面上で部品の置換などを行なう。

5 類似事例探索

過去の仕様/機能/部品を探索する場合にすべてのキーワードを満足する候補がない場合には、設計者は部分マッチによる探索のほかに、意味記憶に基づいて満足されなかったキーワードを類似概念に置換した類似探索を指示することができる[7]。

本章では意味記憶における概念構造関係を提案した後、それをを用いた類似事例探索方法を提案する。

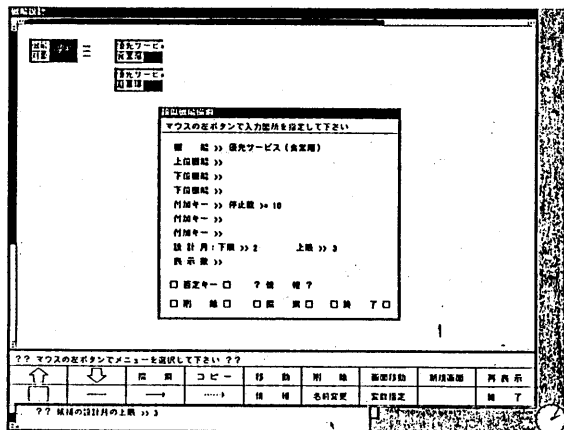


図7 類似機能探索のためのキーワード入力画面の例

5.1 意味記憶

意味記憶は仕様名、仕様のインデックス、機能名などを表現する概念を記憶する。意味記憶は、複数の設計者が使用する用語を統一するためのシソーラスとして、あるいは仕様名、機能名、部品名間の類似性の判断基準として機能するとともに、後述する機能の構造化のための知識としても用いられる。

(1) 概念定義

2つの概念A、Bの関係を特定することを概念定義と呼ぶ。通常、概念定義はノード(概念)とリンク(関係)からなるネットワークで表現される[8]。本研究では関係をさらに構造化し、「関係名」と「視点」からとらえる。すなわち、

【概念A、関係名、視点、概念B】(5.1) という4つ組リストで概念定義する。

「関係名」に関する研究は従来より行なわれてきた。本研究では概念間の関係に関する分類研究[9]、is-a関係の多義性に関する議論[10]、および設計における使用頻度を考えて表1に示す関係名を考慮した。

同じ関係名を持っていても関係としては異なる場合を区別するため「視点」を導入した。例えば「エンジン」という概念は「内燃機関」および「DOHC」という概念の双方

表1 関係名リスト

名称	意味
instance-of	実例
kind-of	上位-下位関係
part-of	全体-部分関係
has-a-value-of	属性
equivalent	同義語

とkind-ofという関係名を持つ。しかし「内燃機関」はエンジンの実現手段であり、「DOHC」は方式のひとつである。関係名だけに基づいて「内燃機関」の類似概念を求めると「DOHC」が得られる場合が考えられるが、これはおかしい。そこで「実現手段」や「方式」という視点を考慮することによりこのような混乱を回避できる。ただし関係名がinstance-ofおよびequivalentの場合には視点は不要とする。

図9に「エンジン」に関する概念定義の例を示す。図中、関係名と視点を「関係名(視点)」と表現した。意味記憶中の概念は本図のような複雑なハイパーネットワークを構成している。このような概念間の関係に基づいて本方式では多視点からの類似事例探索を実現できている。

このような意味記憶の構造や利用方法はSycara & Navinchandra [11]に似るが、is-a関係のみを考慮している文献11に対して、本方式では類似関係をより厳密に定義でき、強力である。

なおSUPPORTでは概念定義をオンラインで支援するためのグラフィックエデ

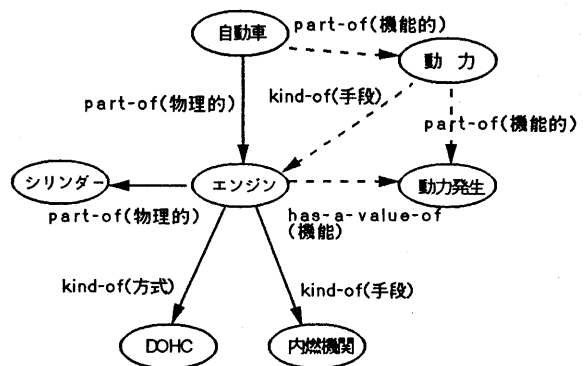


図9 概念定義例

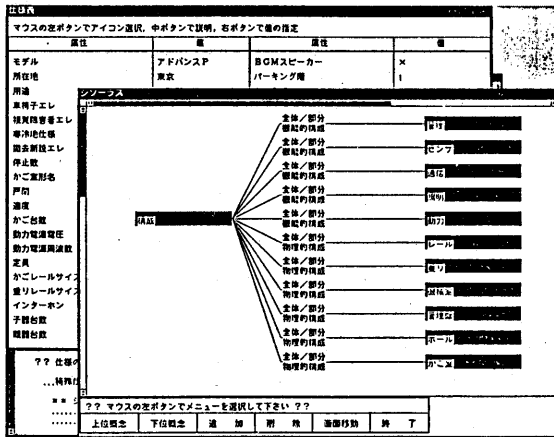


図10 概念定義支援画面例

ィタを提供している(図10)。

(2)類似概念

類似概念を以下のように定義する。

【定義1:類似概念】

概念aと概念bがともに概念cと関係名Rおよび視点Vで概念定義されているとき、概念aと概念bを関係R(V)から見た類似概念と呼ぶ。

5.2 類似事例探索方法

満足されなかったキーワードを置換する場合は以下のような手続きに従う。

(1)複数のキーワードが置換対象となるとき、優先順位の高いものから順に置換候補とされる。優先順位は、例えば

●仕様の探索の場合

仕様項目名と値の間では項目名>値

特殊制御に関する仕様のキーワードの場合には制御内容>制御対象

●機能名による機能および部品の探索の場合

機能の操作名>対象名

などである。

(2)置換候補は意味記憶に基づいて類似概

念に置換され、探索が行なわれる。このとき、以下のような手順で置換が行なわれる(図11参照)。

(case 1)今置換対象であるキーワードをFCとし、FCが概念MCと関係R(V)で概念定義されているとする。もしMCと関係R(V)で概念定義されている別の概念が存在すれば、それを新しいキーワードSCとする。関係R(V)から見た類似概念が複数(C1、C2、...)存在する場合には、設計者が選択しSCとする。SCの選択後探索を実行する。

もし探索に失敗すれば、関係はそのままにしてSCを新たなFCとして同様の手続きを繰り返す。

(case 2)FCが複数の概念MC1、MC2、...と異なる関係R1(V1)、R2(V2)、...で概念定義されており、それぞれに類似概念C1、C2、...が存在する場合には、まず関係名R1、R2、...を比較し、優先順位のもっとも高いものを選択する。関係名の優先順位は

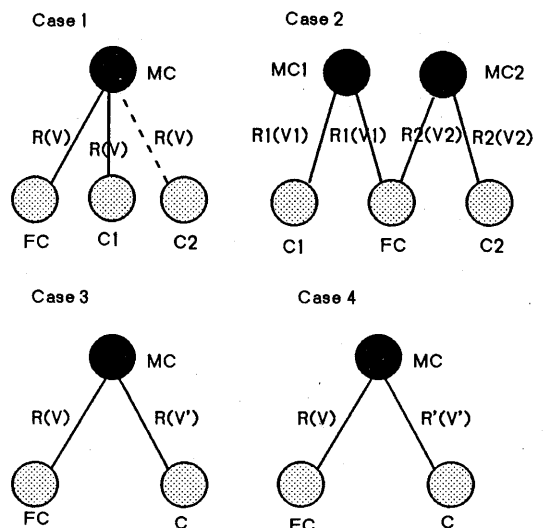


図11 概念の置換パターン

instance-of > kind-of > part-of >
has-a-value-of

である。次に選択された関係名の下に視点が複数存在する場合(例えばR(V)とR(V'))には、設計者が選択する。最後に選択された関係を持つ概念の中から適当なものを設計者が選択する。

もし探索に失敗すれば、関係はそのままにしてSCを新たなFCとして同様の手続きを繰り返す。

(case 3)もしcase 1あるいはcase 2の置換による探索に失敗したなら、関係名をそのままにして視点を別の視点V'に置換した上でSCを変更し探索する。

(case 4)case 3においてもし変更すべき視点がなければ、関係名を変更する。関係名の変更方法はcase 2に準じる。

対象 > 食堂階
制御内容 > 優先サービス

.....

!! 「食堂階」を満足するような仕様は過去にありません !!
?? 類似仕様を探索しますか 1.はい 2.いいえ > 1

... 「食堂階」に関する視点には次のようなものがあります
1. 関係名=kind-of 視点=混雑度 (共通概念=混雑階)
2. 関係名=kind-of 視点=用途 (共通概念=特殊用途階)
?? 視点を指定して下さい > 1

... 対象を「玄関」に変更してみます
.....

!! 候補が見つかりました !!
例えば...
< 類似度=2 >
** オーダー=EA10123 **
仕様項目: 玄関優先
仕様値 : 〇
対 象: 玄関
制御内容: 優先
.....

図 12 類似仕様探索におけるキーワード置換の

図 12 に特殊制御として分類された仕様の場合のキーワード置換の例を示す。この場合は制御対象が置換されている。

6. あとがき

現在 SUPPORT は MOP 理論に基づく機能システムの構造化学習機能の実装中である。今後本稿で述べたような類似探索の能力向上や事例の構造化などを重視してゆきたい。

参考文献

- [1] Hammond, K. J. : Case-Based Planning, Academic Press, San Diego, Ca, 1989.
- [2] Ashley, K. D. and Riisland, E.L.: A Case-Based Approach to Modeling Legal Expertise, IEEE EXPERT, pp. 70-77, FALL 1988.
- [3] Sycara, K.: Resolving Adversarial Conflicts: An Approach Integrating Case-Based and Analytic Methods, PhD dissertation, School of Information and Computer Science, Georgia Institute of Technology, 1987.
- [4] 仲谷・築山・福田: 事例ベース型設計支援における事例利用方法、第15回システムシンポジウム・第10回知識工学シンポジウム合同シンポジウム、pp.405-409、1989.
- [5] Miles, J.D.: Techniques of V. A. & V. E., 2nd ed., Soc. AM. Value. Eng., 1972.
- [6] 仲谷・築山・福田: 事例ベース型設計支援における事例の汎化学習、第16回システムシンポジウム、pp.1-6、1990.
- [7] 仲谷・築山・福田: 事例ベース型設計支援システムSUPPORTにおける類似事例探索方式、情報処理学会第41回全国大会、pp. 4-131-4-132、1990.
- [8] Tulving, E. : Episodic and Semantic Memory, In Tulving, E. and Donaldson, W. (Eds.), Organization of Memory, Academic Press, New York, 1972.
- [9] 人工知能学会(編):人工知能ハンドブック、オーム社、東京、1990.
- [10] Brachman, R. J. et al.: What IS-A is and isn't: An Analysis of Taxonomic Links in Semantic Networks, IEEE COMPUTER, pp.30-36, 1983.
- [11] Sycara, K. P. and Navinchandra, D. : Integrating Case-Based Reasoning and Qualitative Reasoning in Engineering Design, In Gero, J. (Ed.) Artificial Intelligence in Design, Springer-Verlag, New York, July, 1989.