

TM-0938

平成元年度 KSA/KAR サブワーキング
グループ報告書事例ベース
推論の研究動向に関する報告書

小林 重信（東工大）他

August, 1990

© 1990, ICOT

ICOT

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

(03)3456-3191~5
Telex ICOT J32964

Institute for New Generation Computer Technology

平成元年度

K S A / K A R サブワーキンググループ報告書

事例ベース推論の研究動向に関する報告書

平成2年6月

新世代コンピュータ技術開発機構

目 次

1. 序 論 (小林重信)	
1.1 調査研究の背景	1
1.2 調査研究の目的及び方法	1
1.3 活動の経過	2
1.4 事例ベース推論に関する問題提起	9
1.5 本報告書の構成	15
2. 事例ベース推論のニーズ分析	
2.1 電力及び原子力分野におけるニーズ分析 (小林康弘)	16
2.2 機械の編集設計におけるニーズ分析 (今村 聡)	23
2.3 計装・計算機システム設計支援におけるニーズ分析 (萩原賢一)	30
2.4 製鉄分野におけるニーズ分析 (久保真哉)	33
2.5 O A 分野におけるニーズ分析 (本山貴巳)	39
2.6 法律分野におけるニーズ分析 (新田克巳)	42
3. 事例ベース推論に関する文献調査	
3.1 HYPO (新田克巳)	47
3.2 PRODIGY (山村雅幸)	54
3.3 CYCLOPS (和田正寛)	59
3.4 Opportunistic Memory (榎本哲夫)	64
3.5 FIRST (前田 茂)	70
3.6 自動車故障診断(Redmond) (打橋知孝)	76
3.7 JUDGE (古関義幸)	79
3.8 DMAP (島津秀雄)	86
4. 事例ベース推論の要素技術	
4.1 事例の柔軟な特徴づけ (打橋知孝)	92
4.2 最小原理に基づく事例の組織化 (石田好輝)	96
4.3 類推について (原口 誠)	105
4.4 事例ベースの整理と一般化 (久保真哉)	123

4.5	事例照合の心理学的尺度について—プロトタイプ理論— (榎本哲夫)	129
4.6	事例ベース推論における領域知識とユーザの協調 (中村孝太郎)	138
4.7	C B RとE B L : Intractable Domainへの適用 (吉田健一)	149
5. 事例ベース推論のモデルおよび適用例		
5.1	モデルベース診断における経験的知識の学習 (古関義幸)	156
5.2	エピソードから一般化シナリオを構成する学習モデル と事例からの知識獲得 (榎本哲夫)	162
5.3	事例に基づく組立経験知識の学習モデル (安部憲広)	172
5.4	事例を用いた自然言語における動詞の意味同定 (山村雅幸, 小林重信)	181
5.5	事例に基づく問題解決過程の制御モデル (小野典彦)	190
5.6	事例に基づくプラント診断システム : C A O S (石田好輝)	200
5.7	事例を用いた対話型機械調整支援システム (中村孝太郎)	210
5.8	事例に基づく知識獲得支援システム (森下太朗, 椿 和弘)	220
5.9	事例に基づく旅行案内システム (末田直道)	228
5.10	事例を用いた法的推論システム (新田克巳)	235
5.11	事例に基づく相互主導的対話システム (小野典彦)	243
6. 結 論 (小林重信)		
6.1	活動成果の要約	252
6.2	提 言	253

1. 序 論

東京工業大学 小林重信

要 旨

本テクニカルレポートは、平成元年度に I C O T 内に設置されたワーキンググループ W G / K S A のサブワーキンググループ S W G / K A R の活動成果をまとめたものである。本 S W G は、次世代知識システムの重要な基幹技術とみられる事例ベース推論を取り上げ、その基本的な枠組みおよびその工学的利用について、多角的な視点からの検討を行った。全 6 章構成の本報告書は、事例ベース推論のニーズ分析、文献調査、要素技術、モデルおよび適用例について、本 S W G の委員およびオブザーバが分担執筆したものからなる。

1. 1 調査研究の背景

知識情報処理技術に対するニーズは、分類・診断・制御などの解析型問題領域から、計画・設計などの合成型問題領域へとシフトしつつある。設計問題とは、要求仕様として与えられるシステム特性を実現するように、構成要素の特性とシステム構造を決定する問題として定式化されるが、その組合わせ的な性質故に、フォーマルなアプローチは確立されておらず、各分野ごとに試行錯誤的にシステム開発が進められているのが現状である。昨年度 K S A ワーキンググループの活動成果のまとめにおいて、設計問題へ接近する基本的な枠組みとして、問題解決と学習の統合化アーキテクチャの確立が必要であり、その中で事例ベース推論(Case-Based Reasoning; C B R)および説明に基づく学習(Explanation-Based Reasoning; E B L)が主要な役割を果たすことが指摘された。これを受けて、本 S W G は、C B R を中心に設計問題へ接近する上での基盤技術を固めるための活動を展開した。

1. 2 調査研究の目的及び方法

C B R を中心にその基本的な枠組みおよびその工学的利用について、多角的な視点から検討を行い、設計問題へ接近する上での重要な要素技術としての基礎を確立することを本 S W G の活動目的として設定した。

上記目的を達成するために、以下の活動を展開した。

- 1) 文献調査による C B R の最新の研究動向の把握

- 2) C B R に対するニーズの調査
- 3) C B R 関連研究者(外部の研究者を含めて)の講演
- 4) 各委員(オブザーバを含めて)が提出したポジションペーパーに基づく討議
- 5) 合宿を通じての総括的な討議

1. 3 活動の経過

第1回委員会(平成元年5月26日)

本 S W G の研究課題と活動方針を中心に討議した。

- 1) 第5研究室の生駒室長・前田研究員より、本 S W G の趣旨説明および活動方針に関する説明が行われた。
- 2) 小林主査より、設計問題に対する C B R 的接近の必要性ならびに C B R の基本的な枠組みとその構成要素、C B R における研究課題などサーベイ的な報告がなされた。
- 3) 各委員より、C B R に対し、どのような観点からどのように接近していくかについての基本的立場と方針の説明が行われた。
- 4) 前田委員(ICOT)より、C B R に関する参考文献リストが提出された。

第2回委員会(平成元年6月23日)

C B R に関する5篇の論文紹介があった。

- 1) 新田委員(当時ETL, 現在ICOT)より、Ashley and Rissland による Trade Secret 法を対象とする C B R システム HYP0 における Compare and Contrast と呼ばれる推論メカニズムの紹介があった。法律の事例を組織化する上で基礎となる特徴づけ(indexing)について、日本は米国に比べ、基盤の整備が遅れていることが指摘された。
- 2) 山村委員(東工大)より、Carbonell and Veloso による誘導型類推を組み込んだ汎用問題解決システム PRODIGY の紹介があった。誘導型類推の利点は justification が一致するときのみ過去の事例が適用され、知識の変換が行われるところにあることが強調された。
- 3) 広沢氏(ICOT)より、Koton and Chase による C B R の default logic(by Reiter) による定式化の紹介があった。C B R における類似性の多様性と default logic における extention の多様性を対応づけたところが興味深いことが指摘された。
- 4) 前田委員(ICOT)より、Bradtke and Lehnaert による Case-Based Search の紹介があった。8 パズル問題を対象として、事例の特徴づけの違いが事例ベース問題解決の効率にどのような影響を与えるかを論じたものであり、事例の特徴づけ問題の重要性が指摘さ

れた。

- 5) 和田オブザーバ(シャープ)より、Navinchandra による景観設計問題を扱う CBR システム CYCLOPS の紹介があった。事例ベース検索において、direct matching, relaxed matching, systematicity matching の3種が使われているところが興味深いが、その汎用性および正当性については疑問視する意見が寄せられた。

第3回委員会(平成元年7月21日)

1 件の文献紹介、2 件の研究発表および1 件の問題提起があった。

- 1) 榎本委員(京大)より、Hammond による Opportunistic Memory の紹介がなされた。動的環境下でのプランニングではプランと実行を1つのプロセスとして統合的にとらえることが必要であり、エピソード記憶をいかに組織化し、想起するかがポイントであることが指摘された。
- 2) 榎本委員より、エピソードから一般的なシナリオを構成する学習モデルに関連する研究発表がなされた。ファジィ理論を導入して、エピソードの一般化されたルールを生成し、トリガー事象を入力すると、前向き推論により関連事象を探索し、特異事象が求まると後向き推論に切り替え、その原因事象を見出し、一般化知識を出力するもので、表層的な類似性に基づく事例検索ではなく、発想論的な事例検索にその特徴がある。
- 3) 中村委員(日本たばこ産業)より、機械調整における事例ベースの利用に関する研究発表がなされた。エキスパートシステムにおけるルールベース推論を補完する目的で、一般化されていない調整事例を CBR の枠組みで柔軟に利用できるようにしたものであり、対話型モデルを提案し、ESP 上にプロトタイプシステムを実装し、その評価を行っている。問題と事例の間の類似性は属性の一致度および属性値の一致度に基づいている。
- 4) 小林主査より、事例ベース推論の研究課題として、事例の定義、事例の獲得、事例の特徴づけ、事例の組織化、事例の検索、検索事例の競合、修正の型、領域知識の役割、領域知識とユーザの協調、推論の正当化、知識ベース推論との関係、探索との関係、統合型問題解決における CBR の役割、学習との関連、CBR のツール化、CBR とファジィ/ニューロとの関係など16項目が挙げられ、各項目の問題の所在の説明がなされた。

第4回委員会(平成元年9月22日)

1 件の研究発表、2 件の文献紹介、3 件のポジションペーパー発表があった。

- 1) 古関委員(日本電気)より、モデルベース診断における経験的知識の学習に関する研究発表がなされた。モデルベース推論とルールベース推論の融合を目的としており、経験した事例を EBL により一般化したルール形式でルールベースに蓄積する方式を提案し、電子交換機の簡易モデルに適用した場合の知識表現、推論制御、テストパターン生成、

計算効率の評価などが紹介された。

- 2) 古関委員、島津オブザーバ(日本電気)より、IJCAI'89におけるC B Rの最近の研究動向の紹介がなされた。古関委員より、テクニカルセッションにおけるC B Rの研究動向と各論文の概要が報告された。島津オブザーバより、Kolodner and RiesbeckによるC B Rのチュートリアルの概要が報告された。
- 3) 今村委員(MEL)より、機械設計と事例ベース推論に関するポジションペーパーの発表がなされた。定型的機械設計の自動化のために、設計対象モデルに設計プロセス知識も扱う機能をもたせることが必要であり、その1つに過去の設計事例を組み込み、事例ベース推論を行わせる研究計画の構想が紹介された。
- 4) 小林康弘委員(日立製作所)より、プラント設計計画問題における事例ベース推論に対するニーズに関するポジションペーパーの発表がなされた。電力および原子力分野での設計計画用知識システムに関連して、知識獲得の方法として、また設計事例の再利用の方法として事例ベース推論の利用可能性を検討しているとの報告がなされた。
- 5) 打橋オブザーバ(NTT)より、事例ベース推論におけるもっとも重要な研究課題として、事例の特徴づけ問題と検索問題の2つを指摘し、特に、事例に付与すべきキーのシステムティックな抽出方法の確立が必要であるとして、いくつかのアイデアを示すとともに、JICSTシソーラスや分野別用語集の利用が考えられるとの考えが示された。

第5回委員会(平成元年10月20日)

1件の研究発表および4件のポジションペーパーの発表があった。

- 1) 石田委員(京大)より、最小原理に基づく事例の動的組織化と診断問題への適用に関する研究発表が行われた。事例の動的記憶は属性を表わすスロット数が全体として最小となるように逐次的に行われる。ただしこれだけでは不十分なので因果関係の知識にによる事例の特徴づけも必要なことを論じた。また診断システムへの応用においては、事例の修正を中心にいくつかの事例オペレーションの方法が報告された。
- 2) 新田委員(ICOT)より、C B Rと判例に関するポジションペーパーの紹介があった。法律分野はルールベース推論とC B Rの併用システムが共存する領域であることを指摘し、判例に関する基本的な見解を述べた上で、判例のC B Rにおける利用形態として、状況判断、命題および論理展開の3種類の事例ベースが考えられることが報告された。
- 3) 吉田オブザーバ(日立製作所)より、設計問題領域におけるC B Rの利用に関する考察がポジションペーパーとして報告された。設計問題に対するC B Rの適用は有望と思われるが、実際には領域知識をどの程度与えるかが問題となりそうであり、階層的な知識ベースの構築が必要であり、そこではE B Lが利用可能であるとの見解を示した。

- 4) 末田委員(東芝)より、画像処理分野における C B R のニーズと適用に関するポジションペーパーが報告された。画像処理システムの各手順における処理の類似性に着目すれば、知識の獲得が可能であるとの考えが示された。また C B R における問題点として類似性だけに着目する変換には限界があり、E B L の利用も併せて検討すべきと指摘された。
- 5) 小野委員(徳島大)より、C B R の分散型 A I への適用可能性に関するポジションペーパーが報告された。Case Retriever としての CYRUS の特徴を論じた上で、ルールベース推論と C B R との比較分析を行い、分散型 A I の観点からみて、C B R は興味深い性質があることを指摘し、C B R の分散型 A I への利用可能性を論じた。

第 6 回委員会(平成元年11月17日)

3 件のポジションペーパー発表、3 件の文献紹介および中間のまとめがなされた。

- 1) 本山オブザーバ(松下電器産業)より、O A における C B R に対するニーズに関するポジションペーパーが報告された。O A 問題領域においては、過去の事例を用いて処理を行っているものとして、会議室の予約とプロジェクトの再編成があることが社内調査から明らかになり、C B R の利用可能性に関する検討結果が報告された。
- 2) 久保オブザーバ(新日本製鉄)より、製鉄所における C B R に対するニーズに関するポジションペーパーが報告された。C B R の適用が有望な問題として、高炉の操業管理などの制御問題、合金の品質設計などの生産設計問題、生産機械設計や設備工事計画などの工事設計問題が挙げられ、ルールベース推論との比較で、C B R の優位性が議論された。
- 3) 萩原委員(富士ファコム制御)より、計装・計算機のシステム設計支援からみた C B R へのニーズに関するポジションペーパーが報告された。計装・計算機のシステム設計において、類似事例の検索と修正に基づく方法を必要とする背景が述べられ、C B R を適用する上で解決すべき技術的課題が論じられた。
- 4) 前田委員(ICOT)より、Daube and Hayes-Roth による C B R に基づく機械再設計システム FIRST の文献紹介がなされた。FIRST は過去の設計プランを利用して、梁の再設計を行うもので、類推によって変換されたプランと解析的知識を 1 つの再設計プランにまとめるところに特徴がある。FIRST は黒板モデル BB1 上に実装されている。
- 5) 打橋オブザーバ(NTT)より、Sycara and Navinchandra による C B R における Index の生成と変形に関する文献紹介がなされた。設計分野を対象に、設計例の検索に有効な情報を検討し、生成については、動作原理および因果関係に基づく方法を、変形については詳細化、抽象化、属性変更に基づく方法を、それぞれ提案している。
- 6) 打橋オブザーバより、Redmon による自動車の故障診断における専門家の問題解決を観測して、後の利用に有用と思われる事例を生成するシステムに関する文献紹介がなされ

た。本システムでは問題解決の過程でなされたアクションの根拠を説明した上で、事象を細分化して、これらを目標を中心に再構成することにより、事例を生成する。

- 7) 小林主査より、本SWGの中間のまとめとして、CBRの研究課題を24項目に分類、整理した結果が報告された。これは第3回委員会で、同主査が示した16の研究課題を見直し、詳細化したものであり、本SWGの今後の活動を進め、収束させていくためのガイドラインを示したものである。

第7回委員会（平成元年12月12日）

ICOT が招聘した Dr. David Poole (Univ. of British Columbia) による講演“Causal and Evidential Reasoning” に引き続き、1件のサーベイペーパーの発表と1件の研究発表があった。

- 1) 原口委員(東工大)より、類推研究の動向に関するサーベイが紹介された。類推の正当性について、Gentener, Winston, Russell などの研究を紹介した上で、Indurkha による Approximate Semantic Transference 理論における Metaphor と Analogy の関係を中心に類推に関連する諸研究を体系的に論じた。
- 2) 仲谷氏(三菱電機)より、CBRに基づく設計支援システムの研究発表が行われた。機械設計の支援を目的とする CBR システム SUPPORT について、システムの概要、設計支援の過程、事例の利用方法、事例の定義と表現方法、事例の特徴づけ、事例の探索方法および事例の修正方法などが具体的に紹介された。

合宿 平成2年2月2日～3日(横須賀市津久井NTT銀鱗荘にて)

本SWG活動の集大成を図ることを目的に2日間の日程で合宿を行った。参加者は全員がこれまでの活動成果をポジションペーパーの形式でまとめたものを持ちより、延べ10時間を超える討議を行った。

- 1) 小林康弘委員(日立製作所)は、「事例ベース推論に対するニーズ」と題し、既存の設計例を再利用する定型的設計を対象とした問題解決システムについてニーズとその特徴をまとめ、CBRの技術的検討課題を指摘した上で、エネルギー・システム(コージェネ・システム)設計を例として、CBRの適用可能性を検討した結果を報告した。
- 2) 森下・椿両氏(シャープ)は、「事例に基づく知識獲得支援システム」と題し、CBRを利用して、異なる領域の問題間で共通に使われている知識を抽出することを支援するシステムの概要を紹介した。森下氏は同システムの事例ベース推論機構について、椿氏は共通知識獲得機構について、それぞれその詳細を説明した。
- 3) 久保オブザーバ(新日本製鉄)は、「事例ベースの整理と一般化」と題し、事例の整理・一般化に必要な手法や機能について報告した。事例はいつどういう状況のもとで必要と

されるかを分析、事例の種類を分類した上で、事例ベースの整理と一般化の手法として、統計的手法、構造化手法、機械学習、ニューラルネットの4つを目的別に整理した。

- 4) 榎木委員(京大)は、「事例照合の心理学的尺度について：プロトタイプ理論」と題し、CBRシステムの一般的構成とそのツール化、参照事例の検索プロセス、人間の概念化(概念形成)プロセスの理論、概念化プロセスにおける専門的知識の重要性、特徴リスト以外の表現による事例の表象について報告した。
- 5) 中村委員(日本たばこ産業)は、「CBRにおける領域知識とユーザの役割」と題し、CBRにおいて利用される領域知識を分類し、各推論過程での領域知識の役割を位置づけることにより、CBRの実用化という観点から、CBRシステムにおける領域知識とユーザの役割分担を明確化するための検討結果を報告した。
- 6) 吉田オブザーバ(日立製作所)は、「CBRとEBL：Intractable Domain への適用」と題し、設計問題の性質を分析し、CBRとEBLの利用について検討した結果を報告した。事例の修正という観点からは、CBRとEBLは本質的な違いはないことを指摘し、対象の階層構造を考慮したモデル化の必要性和CBRとEBLの係わりを論じた。
- 7) 萩原委員(富士ファコム制御)は、「事例推論の技術課題提起～計装・計算機システム設計の視点から～」と題し、類似事例検索知識および事例変形知識の獲得について、獲得の方法および獲得コストを論じた。さらに、事例の取捨選択の必要性和そのメカニズムおよび事例ベースの管理の問題にも言及した。
- 8) 山村委員(東工大)は、「事例ベース推論を用いた自然言語における動詞の意味同定」と題し、動詞の意味同定のモデルの紹介を行った。事例は問題文、文構成要素の格および動詞の意味からなり、事例ベースは名詞の概念階層に基づいて組織化され、問題文の入力に対し、動詞の意味がその根拠とともに出力されるシステムの概要を報告した。
- 9) 打橋オブザーバ(NTT)は、「事例ベース構築について」と題し、事例ベース構築支援ツールに望まれる機能として、観点の違いによる検索にも対応できるように柔軟な特徴づけの機能を持たせることが必要であることを指摘し、EBLに基づく特徴の抽出、観点ごとの多重組織化を支援する機能をもたせることが必要なことなどを論じた。
- 10) 新田委員(ICOT)は、「事例を用いた法的推論とその並列化」と題し、事例を用いた法的推論のモデル化とその並列化の方法を報告した。法的推論の特徴、判決文に現われる説明の構造を分析し、事例を利用した問題解決の例を示し、CBRとRBRを融合した法的推論エンジン HELIC-II の概要を紹介した。
- 11) 古関委員(日本電気)は、犯罪を対象として判決を下すCBRシステム JUDGE を取り上げ、その構成と動作を分析して、紹介した。JUDGE は、解釈・検索・比較分析・適合修正・一般化の5つのステップからなり、各ステップのメカニズムの特徴と限界について

て、例題を通じて明らかにした。

- 12) 末田委員(東芝)は、「事例ベース推論の事例検討」と題し、KBRとCBRを融合した柔軟な問題解決器の開発に向けて、CBRの枠組みを明確にし、技術的課題を抽出することを目的に試作中のCBRシステムについて報告した。旅行案内を題材に、特に事例の修正を柔軟に行うことに重点をおいて開発が進められているとのことであった。
- 13) 島津オブザーバ(日本電気)は、Fridman の DMAP(Direct Memory Accessing Parser)を取り上げ、その概要をプログラムの動作とともに紹介した。DMAP は人間の自然言語理解のシミュレーションを意図したものであり、単語の予測、失敗の認識と格納および既存の意味構造の修正の3点においてCBRが有望な枠組みであることを指摘した。
- 14) 今村委員(MEL)は、「機械の編集設計における事例ベース推論」と題し、設計作業を支援するCBRについて報告した。設計対象について、一般モデルと視点モデルを対応づけることの必要性を指摘し、知識獲得と探索の負荷の軽減および設計実績の重視の観点からCBRは有用であるとし、CBRに関する研究課題を構成要素別に指摘した。

第8回委員会(平成2年2月23日)

2件の研究発表に続いて、本SWG活動の成果をテクニカルレポートとしてまとめるための打ち合せを行った。

- 1) 安部委員(大阪大学)より、事例に基づく組み立て経験知識を学習するためのモデルに関する研究発表があった。
- 2) 小野委員(徳島大学)より、CBRとEBLを融合した問題解決のモデルとその不定積分問題への応用に関する研究発表があった。
- 3) テクニカルレポートの構成と内容について議論し、執筆分担を決定した。

1. 4 事例ベース推論に関する問題提起

1. 4. 1. 事例ベース推論とは

事例ベース推論(Case-Based Reasoning; 以下 C B R)とは、与えられた問題に類似する過去の事例(成功および失敗の経験)を直接利用して問題解決に導くような推論をいう。C B Rは、ルールベース推論のように領域知識に全面的に依存することはしない。C B Rでは、過去の成功例や失敗例を事例としてメモリに記憶しておくこと、必要に応じて適切な事例を検索すること、成功事例に手を加えて問題解決に導くこと、失敗事例を想起して同じ過ちを回避すること、などを扱う。

C B Rの研究は類推の研究に密接に関係する。これまで、類推の研究が理論を指向し、領域独立的であったのに対し、C B Rの研究はシステム化を指向し、領域依存的側面をもつ。C B Rの枠組みは類推の枠組みよりもかなり広いものである。

C B Rは、領域知識への依存度という観点からは、ルールベース推論よりは弱い問題解決方法であるが、探索(生成検査法)よりは強い問題解決方法である。

次世代知識システムの基盤技術としてC B Rが注目されるのは、以下の理由による。

- 1)知識獲得負荷の軽減： 専門家が不在で、探索をガイドする理論や知識が確立されていない問題領域は少なくない。例えば、新しい機能材料や触媒、香料などの研究開発では、勘と経験に基づく場当り的な探索が唯一の方法であり、C B Rに寄せる期待は大きい。
- 2)探索負荷の軽減： 探索に膨大なコストがかかる問題領域では、C B Rの導入により、探索効率を向上させることが期待できる。
- 3)事例が優先する問題領域の存在： 一般的な知識があっても、事例が優先する問題領域は、特に、非工学的分野において多い。裁判においては、判例の拘束力が強く、法律は後づけ的に利用されるに過ぎない。

1. 4. 2. 事例ベース推論の枠組み

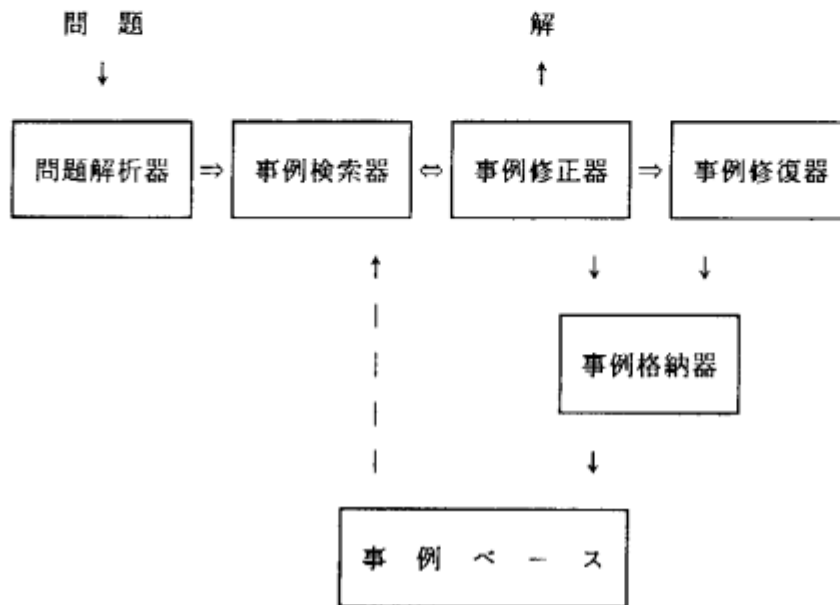
C B Rシステムは、一般に、つぎのような構成要素からなる。

- 1)事例ベース(Case Base)： 特徴づけ(Indexing)された問題解決事例の集りからなり、成功例だけでなく、失敗例も含まれる。
- 2)問題解析器(Problem Analyzer)： 与えられた問題を解析し、その特徴づけを行うとともに、予想される問題点を列挙する。
- 3)事例検索器(Case Retriever)： 与えられた問題の特徴ともっともよく照合する特徴をもつ事例の中で、予想される問題点を回避する事例を事例ベースから検索する。
- 4)事例修正器(Case Modifier)： 検索事例と与えられた問題の間で照合しない部分について、領域知識を使って、修正を施した上で、事例の解法を与えられた問題に変換する。

5)事例修復器(Case Repairer)： 事例の利用に失敗した場合、その原因について、領域知識を使って解析し、事例の特徴づけに反映させ、同じ過ちを回避する。

6)事例格納器(Case Storer)： C B Rによって、与えられた問題が解決されると、新しい事例が生成されたことになるので、その特徴づけを行い、事例ベースに格納する。

C B Rシステムの一般的な構成と推論の流れを下図に示す。



すなわち、C B Rの一般的な流れはつぎのとおりである。

- 1)問題が与えられると、問題解析器により、予想される問題点も考慮した上で、問題の特徴づけが行われる。
- 2)特徴づけられた問題は事例検索器に送られ、事例ベースとの間で照合が行われ、最良照合した事例が事例修正器に送られる。
- 3)事例修正器によって解が見つければ出力され、同時に新しい成功事例を事例格納器を経て事例ベースに登録される。
- 4)事例の修正に失敗した場合には、その事例は事例修復器に送られ、特徴づけが訂正され、事例格納器を経て事例ベースに失敗事例として登録される。
- 5)事例検索器と事例修正器の間では何回かの往復が繰り返される。

C B Rの特徴は、つぎのように要約される。

- 1)事例とは、問題解決努力の集積であり、問題解決コストを節約できる。
- 2)C B Rは、ルールベース推論によって導き出された解の正当化に利用できる。
- 3)失敗事例は、起こり得る問題の予測と回避に利用できる。
- 4)事例だけでは解空間を完全にはカバーできないので、最適解を逃す可能性がある。
- 5)事例間の整合性の維持は困難であり、どの事例を利用するかにより、結果は異なる。

1.4.3. 事例ベース推論の研究課題

C B Rの研究課題を以下に列挙する。

[P01]事例の表現問題

事例の記述は、一般に、問題記述、解記述、その他からなる。問題記述は特徴記述であり、属性一値のペアの集りからなるのが一般的である。解記述は結論、主張、行動、決定、解法などからなる。その他として、事例の実行結果の評価(success or failure)及びその修正または修復の履歴などの情報が付加される場合がある。事例をプロダクションルールと比較すると、特徴記述がルールの前提部に相当し、解記述がルールの結論部に相当する。ルールはパターン記述であるのに対し、事例はインスタンス記述であることが異なる。またルールは小さな知識の断片であるのに対し、事例は大きな事実のチャックである。事例はフレーム形式で表現されることが多いが、まだその表現法は確立されていない。

[P02]事例の特徴づけ問題

事例は問題解決に関連する属性を用いて必要かつ十分な特徴づけがなされることが望ましい。現状では、大半のC B Rシステムにおいて、ユーザによる特徴づけがなされているが、信頼性および効率性において難がある。特徴づけにおけるユーザの負荷を緩和するために、クラスタリング手法の適用も考えられるが、特徴抽出の問題に関係し、A I分野に限らず大きな問題である。領域知識が利用可能な場合、例えば、説明に基づく学習(E B L)利用して、特徴づけを行う方法も考えられ、実際にそのような試みもある。属性の間に従属関係がある場合、属性に関する概念階層の組織化が必要とされる。

[P03]事例の組織化問題

事例の組織化を行わないフラットな事例ベースの構成は、明らかに限界がある。複数事例の間での特徴記述の共通性を利用して事例の汎化階層(分類階層)を形成することは、事例検索の効率化に寄与する。事例集合が事前に所与の場合、汎化階層の最適構成が問題とされる。事例が逐次的に与えられる場合、汎化階層の動的更新が問題とされる。すなわち、事例が入力されるごとに、汎化階層をインクリメンタルに更新するアルゴリズムを確立する必要がある。

[P04]事例の組織化と検索アルゴリズムとの関連

事例ベースがフラットな場合、事例の追加は容易であるが、網羅的探索が必要となり、並列探索のためのハードウェアが必要となるなど、一般に、非経済的である。事例ベースが階層的に組織化されている場合、検索を効率化することが期待できるが、事例ベースの最適構成とその維持に負荷がかかる。Discrimination NetsやPandemonium Netsなどがよく利用されている。

[P05]事例の検索問題

事例ベースより問題に適用可能な事例を見出すためには、問題と事例の間での属性名および属性値の一致および不一致の程度を尺度化することが必要である。属性名および属性値は、その相対的な重要度に応じて、重みづけされる必要がある。

[P06]事例の照合

問題と事例が完全に照合することは稀であり、部分的に照合(Partial Matching)するのが一般的である。陽に表現されている表面的な特徴に基づく照合ではなく、隠された特徴をも考慮した意味的な照合が必要である。意味的な照合を行うためには、概念階層的知識や因果関係的な知識の利用が必要である。

[P07]類似性に関するモデルの必要性

類似性に関しては、例えば、幾何学モデル、集合論的モデルなど数理心理学の分野での伝統的なモデルがあり、それらのC B Rでの利用可能性の検討や見直しが必要であろう。

[P08]検索事例の競合問題

与えられた問題に対して、複数の事例が検索され、しかもそれらが相反する結果を示す場合、競合解消が必要である。HYP0では、比較・対比(compare and contrast)という方法で競合解消を実現しているが、競合を解消するためには、メタな知識が必要である。

[P09]複数事例の利用問題

複数事例の利用には、2とおりの場合がある。第1は、類似の結論をもつ複数事例を利用することにより、推論結果の正当性を高めることであり、第2は、合成型問題領域を対象とする場合、複数事例を相補的に利用して、合成的な問題解決を実現することである。

[P10]修正の型

事例の修正には類推が関係する。類推の基本型として、変形型類推(transformational analogy)と誘導型類推(derivational analogy)の2つがある。一般に、解析型問題領域に対しては変形型類推が、合成型問題領域に対しては誘導型類推が適している。

[P11]推論の正当化問題

C B Rにおける推論の基本原理は類推にあり、類推の本質は、単一事例の一般化(帰納)とその具体化(演繹)に他ならない。C B Rの結果を正当化するためには、別の手段が必要である。背景知識に基づく演繹的推論による正当化、あるいは複数事例からの推論に基づくevidential reasoningとしての正当化が要請される。

[P12] 領域知識の役割

事例の検索において意味的類似度を最適化するためには、概念階層的知識の利用が不可欠である。また、類推によって事例の修正や修復を行なうための知識として、因果的知識や修正知識などの領域知識の整備が必要である。

[P13] システムとユーザの協調問題

領域知識が充実するにつれて、C B Rはルールベース推論に近づく。一方、ユーザの介入が拡大するにつれて、C B Rはデータベース検索に近づく。C B Rは、ルールベース推論とデータベース検索の中間に位置するとの認識の元に、システムとユーザの協調問題を考えることが必要である。

[P14] 事例の獲得問題

知識ベースシステムにおける知識獲得問題と同様に、C B Rシステムにおいても事例獲得問題は重要な問題である。事例は経験が知識化される前の情報であることから、多数存在するとはいえ、その取捨選択には注意を要する。典型的な事例を選択的に獲得し、問題空間を適切にカバーすることが望まれる。事例の獲得には、専門家に対するインタビューに基づく対話的な獲得および探索ベースでの問題解決過程の解析に基づく自動獲得の2つがある。事例ベースの維持管理問題、すなわち事例間の矛盾や冗長性を除去することはC B Rの性能向上に寄与する。ただし、知識ベースの維持管理には厳格に行なう必要はない。

[P15] 学習との関連

C B Rはその問題解決過程自身が新しい事例の獲得につながり、自己増殖的な側面をもつので、その意味で学習システムの1形態とみなすこともできる。また、事例をもっともよく説明する特徴を抽出するために、類似性に基づく学習(S B L)や説明に基づく学習(E B L)が利用可能であり、学習との関連も深い。

[P16] C B Rの利用形態

C B Rの利用形態には、自動化指向および対話指向の2つがある。自動化を指向する場合には、問題解決能力の完全性、すなわち、検索・修正・修復の各過程における完全性が要求される。一方、対話的利用にもいくつかの形態がある。教育訓練が目的の場合、検索・修正・修復の完全性が要請される。オペレータガイダンスが目的の場合、検索と修正の完全性が要請される。領域専門家の利用が目的の場合、修正や修復の完全性は不要で、検索の完全性だけが要請される。

[P17] C B Rの定量的／定性的評価

C B Rシステムと知識ベースシステムの比較は多角的に行う必要がある。システム開発期

間、知識獲得の負荷、問題解決の性能、説明能力、システムの拡張性や保守性などが評価として考慮されるべきである。定量的／定性的評価を行うために、testbed的な問題をいくつか用意することが望まれる。

[P18] CBRのツール化

CBRシステムの研究開発はまだプロトタイプの段階にあり、CBRは概念的枠組みについての完全な合意が得られているとはいえない状況にある。したがって、CBRのツール化もまだプロトタイプの段階にあり、概念的枠組みをツール化するための条件の洗い出しがこれから必要になってくる。

[P19] 知識ベース推論との関係

CBRと知識ベース推論は、互いに、相補的な側面と競合的な側面をもつ。事例ベースから知識ベースへの変換は、知識の自動獲得問題に関係し、説明に基づく学習に密接に関係する。

[P20] 探索との関係

CBRと生成検査をベースとする探索は、互いに、相補的な側面と競合的な側面をもつ。探索データベースから事例ベースへの変換は、事例の自動獲得問題に関係し、今後の大きな研究課題である。

[P21] データベース検索との関係

CBRとデータベース検索は、互いに、相補的な側面と競合的な側面をもつ。データベース検索では、正確な検索条件の入力が要請されるが、事例ベース推論では部分的な照合が許されるので、柔軟な検索が可能である。データベース検索と比べて、CBRは自己組織化的側面に特徴がある。

[P22] 事例ベース推論とファジィ／ニューロとの関係

ファジィ推論やニューラルネットも、CBRの1種とみなすことができる。ファジィ推論の利点は複数の類似事例からの合成（補間）能力にある。一方、ニューラルネットの利点は自己組織化能力にある。

[P23] 統合型問題解決における事例ベース推論の役割

CBRは知識ベース推論と探索型問題解決の中間に位置して、両者の橋渡しの存在であり、統合型問題解決を実現する上での核となる。

[P24] 問題解決と学習の統合化におけるCBRの役割

CBRは問題解決パラダイムの中核的存在であることに加えて、その学習的側面を併せもつことから、問題解決と学習の統合化を実現する上での核ともなる。

1. 5 本報告書の構成

本報告書は、「事例ベース推論に関する調査研究報告書」と題し、全6章からなる。

第1章「序論」では、本研究の背景、目的、方法、活動経過及び研究成果の要約を述べている。

第2章「事例ベース推論のニーズ分析」では、事例ベース推論のニーズについて、電力、機械、製鉄などの工学的分野をはじめとして、O Aや法律の分野まで、幅広く調査し、その必要性を明らかにしている。

第3章「事例ベース推論に関する文献調査」では、主として米国における事例ベース推論の動向を把握するための文献調査の結果をまとめている。

第4章「事例ベース推論の要素技術」では、事例ベース推論の要素技術である事例の特徴づけ、事例の組織化、類推、事例ベースの整理、事例照合の尺度、領域知識とユーザの役割、E B Lとの関連などを個別に議論している。

第5章「事例ベース推論のモデルおよび適用事例」では、わが国で研究開発中の事例ベース推論システムをケーススタディの形で紹介し、事例ベース推論の技術的、具体的な課題を論じている。

第6章「結論」では、本S W Gの活動成果を要約し、得られた知見及び残された研究課題を提言としてまとめている。

2. 事例ベース推論のニーズ分析

2.1 電力及び原子力分野におけるニーズ分析

(株)日立製作所 小林 康弘

要 旨

電力分野、原子力分野において見られるような工学的システムの設計・計画問題を対象とする知識処理システムに焦点を当て、事例ベース推論の実用化に関する当面の課題を検討した。

エネルギー・システム設計支援システムという机上で想定した一つの事例をもとに、事例ベース推論を適用した問題解決アプローチが、このドメインにおいて大きく発展する可能性を指摘した。

2.1.1 はじめに

事例ベース推論は、ルールベース推論に代わる新しい問題解決パラダイムであると言われる。確かに、機器のトラブルシューティングのような診断問題や自然言語処理に関連した解釈問題を想定する場合、両者のギャップは印象的であろう。しかし、電力分野、原子力分野において見られるような工学的システムの設計・計画問題を対象とする知識処理システムに焦点を当てる場合、その新しさはさほど鮮明ではなくなる。それは、工学システムの設計・計画問題を扱う知識処理システムにおいては、他のドメインに先んじて事例ベース推論のコンセプトが浸透しているためであると考えられる。

本稿では、机上で想定した一つの事例をもとに、事例ベース推論を適用した問題解決アプローチが、このドメインにおいて大きく成長する可能性を検討したい。

2.1.2 既存の設計例を再利用する問題解決システム

(1) ルーチン設計と事例ベース推論

ここでは、議論を、工学システムの設計・計画問題を対象とする知識処理システムに限定する。現行のこのようなシステムは、ルールベース推論に基礎を置くものの、その構築に当たっては、設計や計画の事例が既に大きな位置を占めている。その理由として

は、第一に、知識ベースに格納するルールを獲得する段階において、事例から手掛かりを得ることが多いことが挙げられる。次いで、「ルーチン設計」と呼ばれ、既存案を再利用するスタイルを取るアプローチが、このドメインのエキスパートにとってきわめて自然であることを指摘できる。特にルーチン設計は、問題解決方法の上で Selection and Refinement として類型化され、事例ベース推論の中心的部分と重なりが大きい。

〔Chandrasekaran 87〕 エピソード記憶という歴史的な原形とは距離があるが、ルーチン設計は、深いところでつながりを持つもう一つの前形と見なすことができる。

事例ベース推論システムの処理の流れを、ルーチン設計に関連付けて図1に示す。

〔AI センター 88〕 Selection の部分はステップ①、②に、Refinement の部分はステップ③に対応する。現在までいくつかの設計支援システムが、ルーチン設計を扱う知識処理システムとして開発されている。しかし、これらのシステムでは、Refinement の部分を自動化するところに重点が置かれており、Selection の部分はシステム利用者への依存度が高い。この点は、他のドメインにおいて、事例ベース推論を実現したシステムと対照的である。しかも、この理由は、必ずしも、Selection の部分が Remembering という人間の持つ最も高度な情報処理の側面に関連するが故にソフトウェアとして実現が難しいという単純な話でもないのである。ドメインには、ドメインの事情があるのである。

(2) 対象ドメインから見た事例ベース推論

事例ベース推論に関連して、電力分野、原子力分野という「オーソドックスな工学システムを扱うドメインにおける知識処理」という観点から数点コメントする。

- (a) 実際の設計でエキスパートは、既存の事例を用いている。結果として全く異なるような場合でも既存例を使う。事例ベース推論は、この意味で、「設計のカルチャー」と親和性が高いと言える。
- (b) 設計結果の再利用はするが、必ずしも再利用を意識して設計していない。したがって、設計結果のデータベース化、特にインデックス情報を付加したデータベース化は進んでいない。この点から、事例ベース推論、特に検索のステップのインフラストラチャは、現状では高い水準であるとは言えない。
- (c) 設計仕様、設計者の意図は、設計結果（図面）の中に顕在化しているとは限らない。事例として有効に活用するためには、これらを付帯的情報として顕在化させる必要がある。このような情報を取り入れていくためには、当面マンマシン協調が不可欠となる。
- (d) 高信頼化のため、十分実証を経た設計例をベースとすることが積極的に求められる場合がある。このようなケースは、事例ベース推論によって初めて知識処

理による設計支援が可能となる。必ずしも技術的な根拠によるものではないが、ルールベース推論ではできないことが、事例ベース推論でできるようになる問題解決の例である。

(e) 処理の各ステップでの特徴を挙げる。

- ①規模容量、実現方式等既知のインデックス情報による分類で対応できる。
- ②上記の分類に対応して、事例の数は限定できる。
- ③事例の変更を自動化できたとしても、対話的修正の手段が必要である。
- ④得られた解が、妥当であることを検証できることが重要である。

(3) 事例ベース推論の実用化に関する当面の課題

以上の議論より、事例ベース推論に対しては、対象ドメインに特有な次のようなニーズが明らかになった。

- (a) 事例ベース推論については、当面は、設計例の変更のステップを中心とする後半フェーズだけを自動化したシステムの実現を考えれば良い。
- (b) 設計例の検索のステップを中心とする前半フェーズについては、自動化のシーズ、ニーズが不透明であるので、代替の手段を準備する必要がある。例えば、このフェーズを、マンマシン協調により実行することが考えられる。また、前述のルーチン設計システムと同様、システム利用者が対話的に Selection を行なってしまう形の設計支援も現実的である。既存の設計案そのものではなく、それに替わる雛形（テンプレート）を用いることにより、利用可能な事例が乏しいことをカバーし、設計例の変更のプロセスを容易にすることができる。

したがって、課題の形でまとめると、次のようになる。

- (i) ③の段階で変更が容易な雛形（テンプレート）を利用した事例ベース推論の後半フェーズの実現
- (ii) マンマシン協調推論の実現

これらの課題に応えたシステムのイメージとして、机上で想定したシステム事例を 2. 1. 3 に説明する。

2. 1. 3 エネルギー・システム設計を例とした検討

(1) 対象領域

小規模の電力・熱・冷熱供給システムの概念設計を支援するシステムに、事例ベース推論を適用することを考える。システム概念設計とは、機器構成と機器の容量・タイプの選定を行うことである。設計仕様は、①電力・熱・冷熱の設計基準負荷、②自家発電、温水供給等の条件によって与えられる。[早川 85]

(2) 設計対象

図2に、エネルギー・システムの概念設計のためのデータ階層を示した。

与えられた負荷に対応して、システムは発電機、ボイラー等の機器を要素として設計される。エネルギー・システムの設計事例は、システムのレベルで与えられる。図のデータ階層では、システムの事例の内容が、大きく構成と機器に分けられることに注目して、システムの事例を、構成を表わすテンプレートのレベルと機器のレベルとに分けて蓄えている。

小規模のエネルギー・システムでは、機器レベルの設計は、カタログにある製品を選択することである。機器のレベルでは、事例はバラエティに富むが、事例を修正するという必要性は乏しい。一方、構成のレベルは、図3に示すような比較的少数のテンプレートで表わすことができ、このテンプレートを、構成の事例として修正することは容易である。[日本コージェネレーション研究会 89]

実際には、システムの事例から構成と機器の事例が出てくるのではなく、設計の出発点には、あらかじめライブラリー化したテンプレートと、カタログをデータベース化した機器の事例がある。問題が与えられたときには、システムの事例を検索するのではなく、まずテンプレートを選択し、次に機器を選択する。

(3) 設計のフロー

図4は、既存のエネルギー・システム評価プログラムの概略計算フローであり、システムの構成が与えられた段階でシステムの性能評価を行う。[日本コージェネレーション研究会 89] 本設計支援システム場合においても、図4と同様の設計計算を行ないながら、下記の対話的設計プロセスをガイドする。

- (a) 与えられた設計基準負荷、設計条件のもとでテンプレートを選択し、必要に応じ修正する。特別な場合、新たにテンプレートを作成することもある。
- (b) 機器レベルでの設計基準負荷から、台数・タイプ・容量を決定し、製品を選択する。

2. 1. 4 まとめ

設計・計画問題を対象とするドメインでは、現実の問題解決プロセスとの類似性より、事例ベース推論の潜在的ニーズは、大きい。特に、設計例の変更を中心とする事例ベース推論の後半フェーズに重点を置き、マンマシン協調推論と組み合わせることにより、早い段階で実用化されることが期待される。

事例ベース推論のアプローチとしての新しさを指摘するものとして、次のような議論がある。従来のアプローチが C & D (Create and Debug) であるのに対し、このアプローチは A and A (Anticipate and Avoid) であると言う。[Riesbeck 89] しかし、従来より設計・計画問題を扱うプロセスにおいては、Create の段階でも不完全ながら A and A の思想は入ってきている。過去における問題解決の失敗事例は、トラブル事例集という形で伝承され、新しい問題解決に反映される。それでもなお Debug が不可欠であるのは、問題の難しさということなのであろう。

参考文献

- [A I センター 88] ICOT-JIPDEC A I センター : 計画・設計型知識システムの構築方法論, 知的情報処理システムに関する調査研究報告書, Vol. 3, レポート 63-A001 (1988)
- [Chandrasekaran 87] B. Chandrasekaran : Towards Functional Architecture for Intelligence Based on Generic Information Tasks, Invited Talk of IJCAI-87 (1987)
- [早川 85] 早川一也 他 : コージェネレーションシステム, シーエムシー (1985)
- [日本コージェネレーション研究会] 日本コージェネレーション研究会 : コージェネレーション導入意志決定用評価プログラム, ユーザーズ・マニュアル (1989)
- [Riesbeck 89] C. K. Riesbeck and R. C. Schank : Inside Case-Based Reasoning, Lawrence Erlbaum (1989)

- ①問題（事例）の特徴付け
 - ②設計例の検索
 - ③設計例の変更
 - ④設計結果の評価（デザイン・レビュー）… 必要なら③へ
 - ⑤設計例のデータベース化
- } selection
 Refinement
- 成功事例 → CADシステムのデータベース
 失敗事例 → トラブル事例集

図1 事例ベース推論システムの処理の流れ

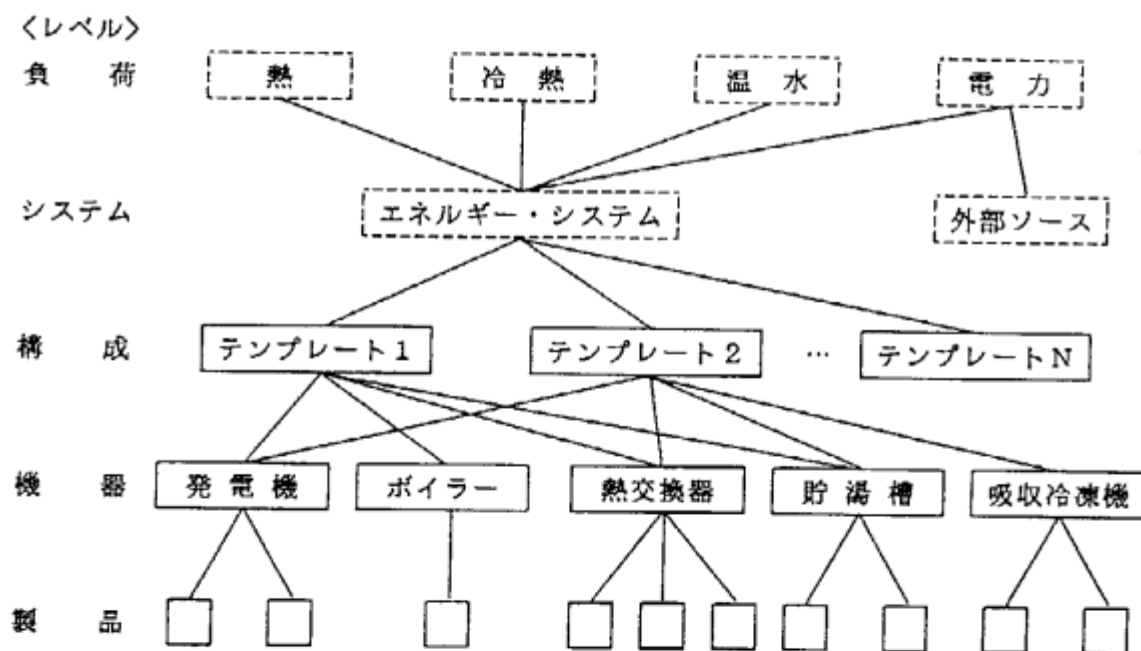


図2 エネルギー・システム概念設計のためのデータ階層

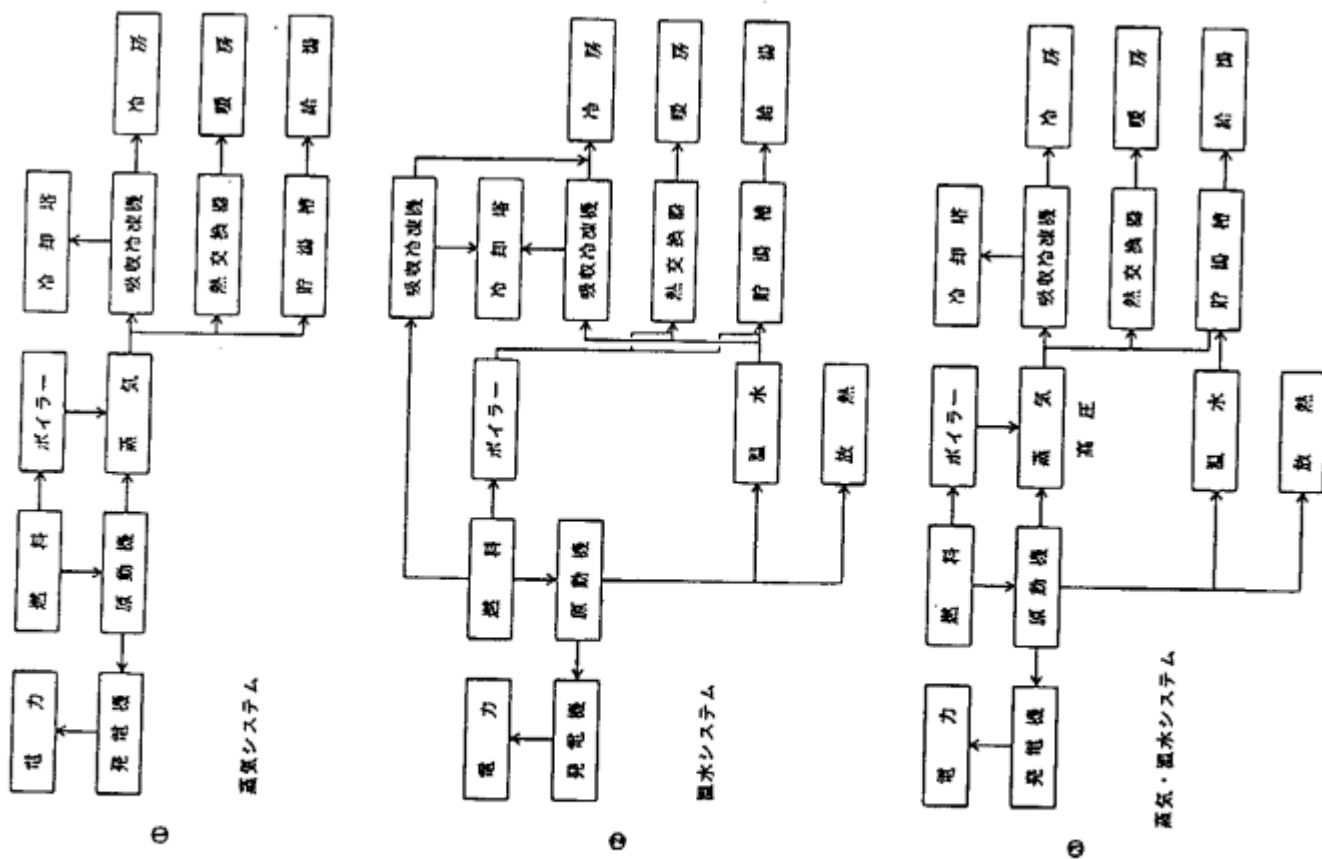


図3 模範構成を改めたテンプレート

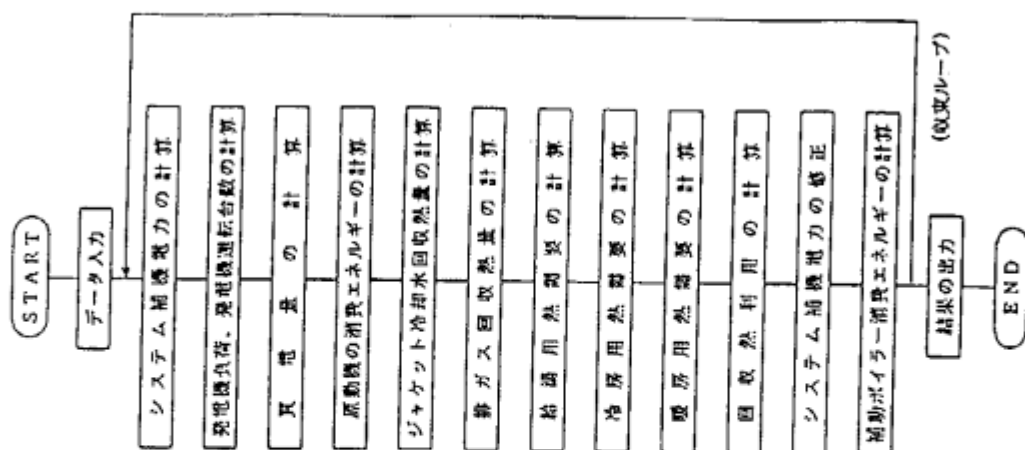


図4 エネルギー・システムの概略計算フロー

2. 2 機械の編集設計におけるニーズ分析

機械技術研究所 今村 聡

要旨

比較的様式化の進んでいる機械設計を対象に、設計対象モデルライブラリを構築し、編集設計、流用設計、類似設計と呼ばれる比較的定型的な設計を支援する方法について考察する。このような設計では既存の設計対象モデルを再利用し、効率的な設計を図る必要があるが、そのような既存のモデルを集積してライブラリ化すれば、広い範囲の設計に対処することができる。モデルはデータ構造とともに、データの取扱方に関する知識を持っており、ライブラリはオブジェクト指向データベースにより構築される。ライブラリには、規格化された機械要素とともに、一般性の保証はできない設計事例というレベルのものも扱う必要があるので、事例ベースとしての性格も持たせるとよい。

2.2.1 設計対象モデル

インテリジェントCADに対してはいろいろな考え方があるが、ここでは編集設計とか流用設計、類似設計と称される、比較的定型的な設計作業を対象に、設計支援の方法について考察する。

設計知識には設計対象に関する知識と設計プロセスに関する知識があるとされる。設計対象に関する知識は、設計対象モデルのデータ構造、データ間の関係、などで表現されるものである。一方設計プロセスに関する知識は設計対象モデルを生成、成長、修正させる手順に関するものである。これらを独立なものとして扱う場合がある。

しかし設計手順に関する知識も多くは個別的で、設計対象に強く依存しており、設計対象と独立に扱うことはできない。むしろ設計対象モデル自身が設計プロセスに関する知識も扱うようにするとよいと考える。すなわち、それぞれに固有の設計プロセス知識と問題解決器も個別に各モデルが持つようにするとよい。但し、制約解決器、RDBなど一般性のあるものは独立に定義し、各モデルから利用できるようにする。設計作業はモデルと設計者のインタラクションを前提としている。

設計対象モデルは、オブジェクト指向データベースにより管理され、図2.2.1に示されるようなカテゴリーに分類される。

<u>形状要素</u>	
標準形状要素	— J I S 化されているもの (ex ネジ穴、キー溝)
準標準形状要素	— 形状特徴 (頻繁に使われる形状要素)
<u>機械部品</u>	
標準部品	— カタログ表から選択できるもの (ex 軸受け、ねじ)
準標準部品	— 形状などの連続的変更を許すもの (ex ブーリ、歯車)
<u>機能ユニット</u>	
— 過去の設計例に基づく機構の属性モデル	
<u>機械</u>	
— 過去の設計例に基づく機械の属性モデル	

図 2. 2. 1 設計対象モデルの分類

形状要素はG-rep (Gluing Representation) と呼ぶソリッドモデルで表現される [Imamura88]。形状はパラメトリックな変更が可能で、標準形状要素では離散的、準標準形状要素では連続的な変更が許される。

機械部品には標準機械部品と準標準機械部品の二つのレベルが考えられる。標準機械部品は J I S、部品カタログなどで扱われているもので、主要データが表形式で表現されるものである。データは離散値からの選択のみが許され、連続的な変更は許されない [Piera88]。準標準機械部品は、しばしば設計に利用されるが、規格化されるまでにはいたっていないもので、一部の手直しにより何度も再利用されるようなものである。寸法などのデータは連続的な変更が可能である。

標準部品で中心となる表形式のデータは関係データベースによる表現が適している。問い合わせのための関係論理式を SQL ライクな言語でサポートし、さらに対象に依存した特殊関数 (例えば応力、寿命などの計算式) による検索もローカルに登録できるとよい。表の集合演算に関しては、直積演算 (m 項の表と n 項の表から m + n 項の属性表を作り出す演算) がサポートされれば十分と思われる。

準標準部品では、属性データは基本的には自由に操作できるが、属性値の範囲、属性間の関係などの制約はありうる。このような属性に関する制約が表現されていることが必要である。

これらの情報は汎化階層 (性質の継承による階層構造) を利用して効果的に整理することが出来る。例として、機械要素のうち特にラジアル軸受けについて詳細化した汎化階層を図 2. 2. 2 に示す。

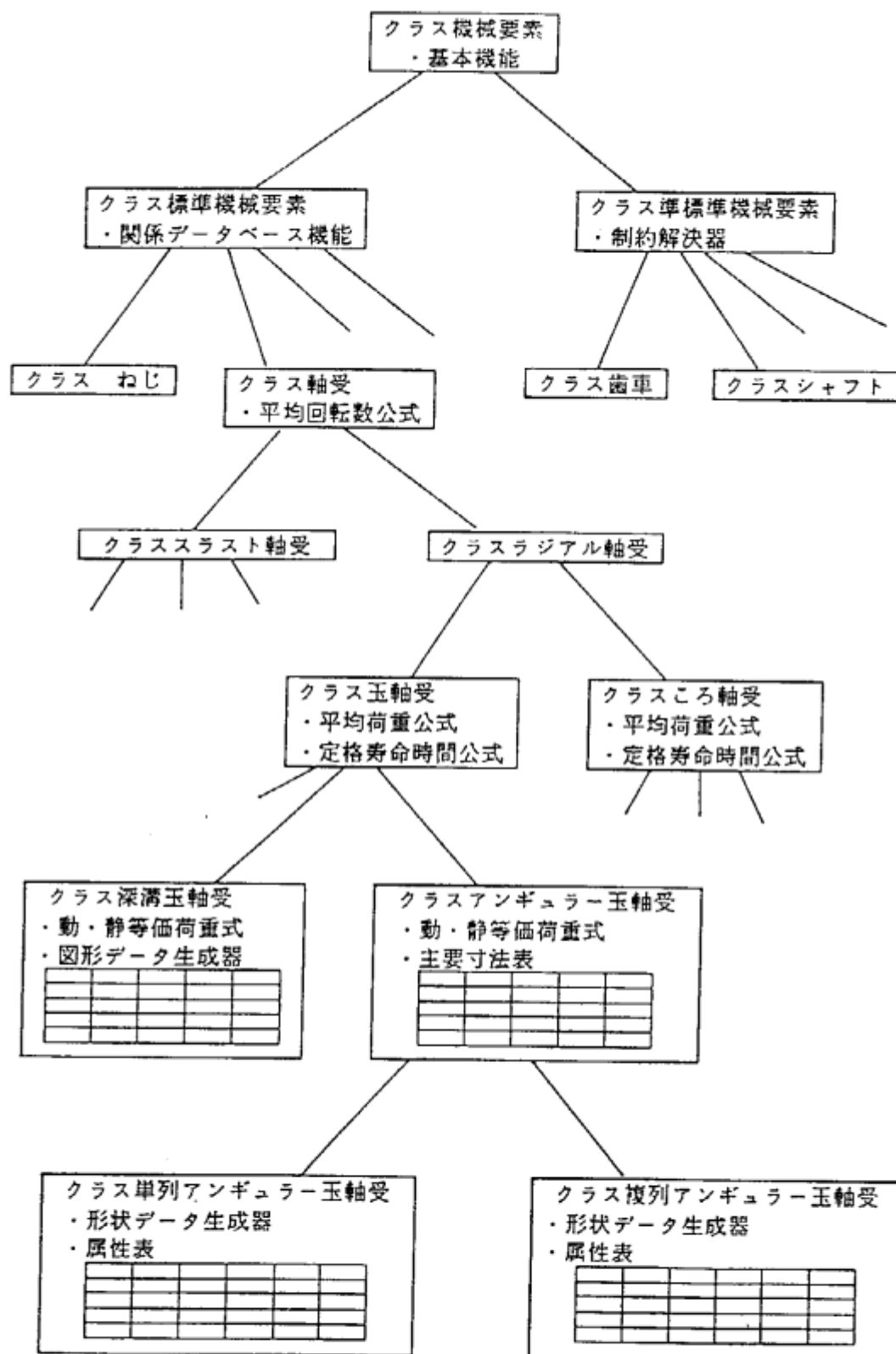


図 2. 2. 2 機械要素の汎化階層

図にある軸受け、玉軸受け、深溝玉軸受けなどは、各々に共通なデータや手続きを記入したクラスとして表現されている。平均荷重公式、定格寿命時間公式は玉軸受けところ軸受けで公式が異なるので、別々に登録される。動等価荷重式や静等価荷重式も同様である。実際の軸受け毎に詳細に規定されるデータ（呼び番号、主要寸法、基本定格荷重、許容回転数など）は最下位のクラスに表形式で登録される。2次元、3次元の形状データは主要寸法データをもとにそれぞれのクラスが持つ形状生成器により生成することができる。設計条件に適合する軸受けは、クラス内またはそのクラスの下位に属するすべてのクラスに登録されている表から、関係論理式を使って選択される。

直積演算は二つの表から一つの表を作成する場合に必要となる。アンギュラー玉軸受けには、単列型と複列型があるが、主要寸法などの共通データは上位のクラス（クラスアンギュラー玉軸受け）に登録し、基本定格荷重、許容回転数などは単列、複列で異なるので、それぞれのクラスに別個に登録することになる。従って、単列、複列それぞれのクラスのデータ表はそれぞれのクラスの表と上位クラスの表との直積演算によって作成されることになる。

機能ユニットは標準部品、準標準部品の複合オブジェクトにより表現される。

図2.2.3に機能ユニットの表現例を示す。機械も標準部品、準標準部品、機能ユニットの複合オブジェクトにより表現される。このモデル表現の特徴は、等値関係をオブジェクトの共有により表す点にある。図2.2.3では、ギアのモジュールの等値関係、ギアの内径とシャフトの外径の等値関係の表現にこの方法が用いられている。その他の制約関係は制約解決器により扱う。

これらの機械（構成要素）に対して編集、選択、修正などの作業を通して設計を進めることになる。また選択、修正などに関わる知識はそれぞれのモデルが持つようにする。

これらの設計対象のモデル表現は次のような特徴を持つ。

- 1) 属性モデルにより対象を表現する。属性モデルとは対象を設計作業で扱う属性の集合で表現したもの。機能設計の範囲が守備範囲であり、詳細設計までを対象としない。詳細設計は、詳細設計用のモデル（製品モデル）で行なう。
- 2) 設計式、検索式、形状モデルなどの非正規形データを扱う。
- 3) 汎化階層による表現。設計式、属性値は汎用性により、属する階層が決まる。
- 4) 複合オブジェクトにより表現する。例えば、機械は機構や（準）標準部品から、（準）標準部品は（準）形状要素から構成される。

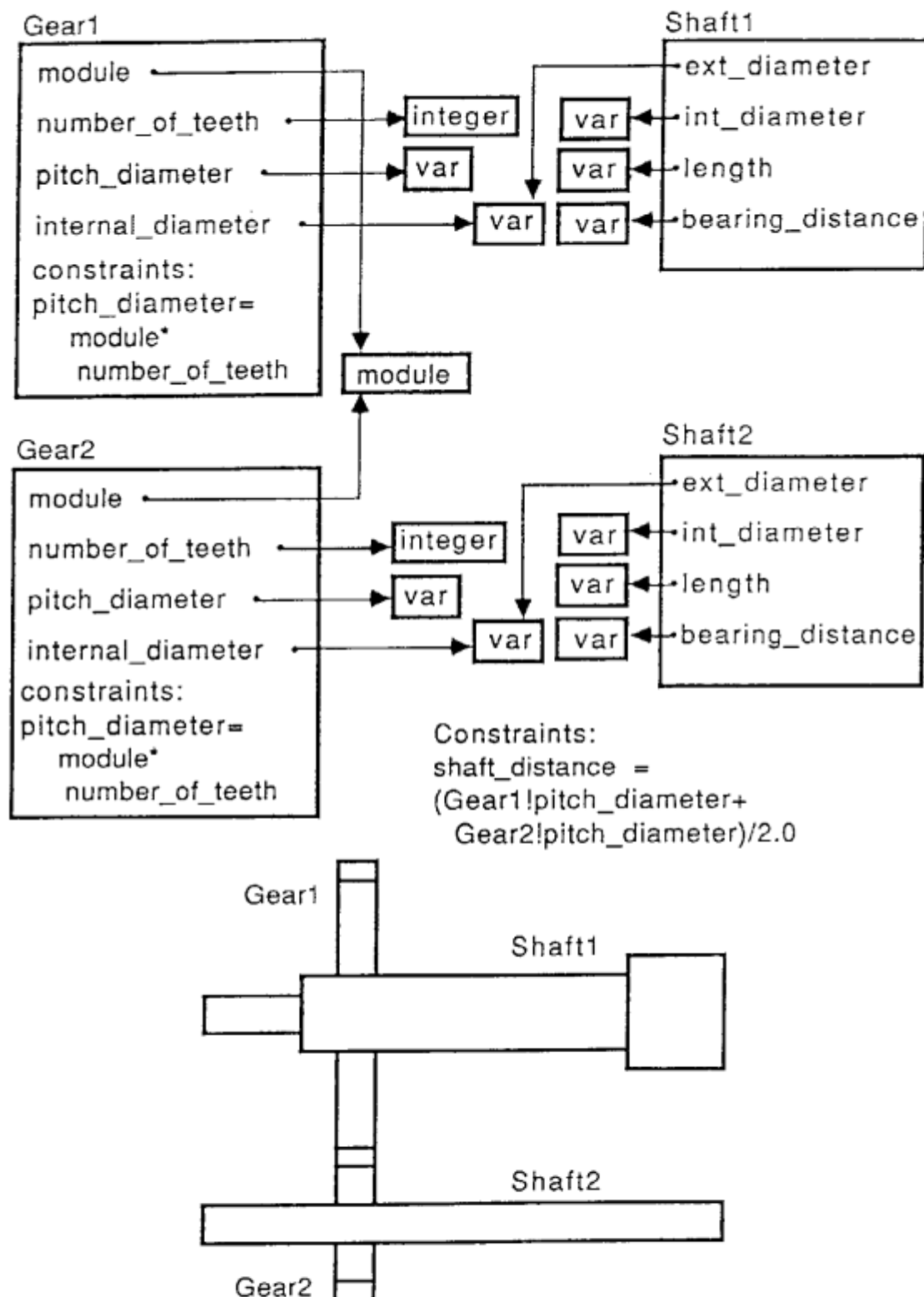


図 2. 2. 3 機能ユニットの複合オブジェクト表現

2.2.2.2 設計モデルライブラリにおける設計事例の扱い

設計事例は設計モデルの実際の利用例を記述する形式で表現される。すなわち設計モデルではデータ構造が決められているだけで値そのものの決定は設計者が行なうことになるが、設計事例には、実際のデータの値とその値に決めた根拠、条件（仕様など）が記述される。設計事例もオブジェクトとして表現され、設計モデルとはis_view_of関係により関係付けられる。この様子を図2.2.4に示す。

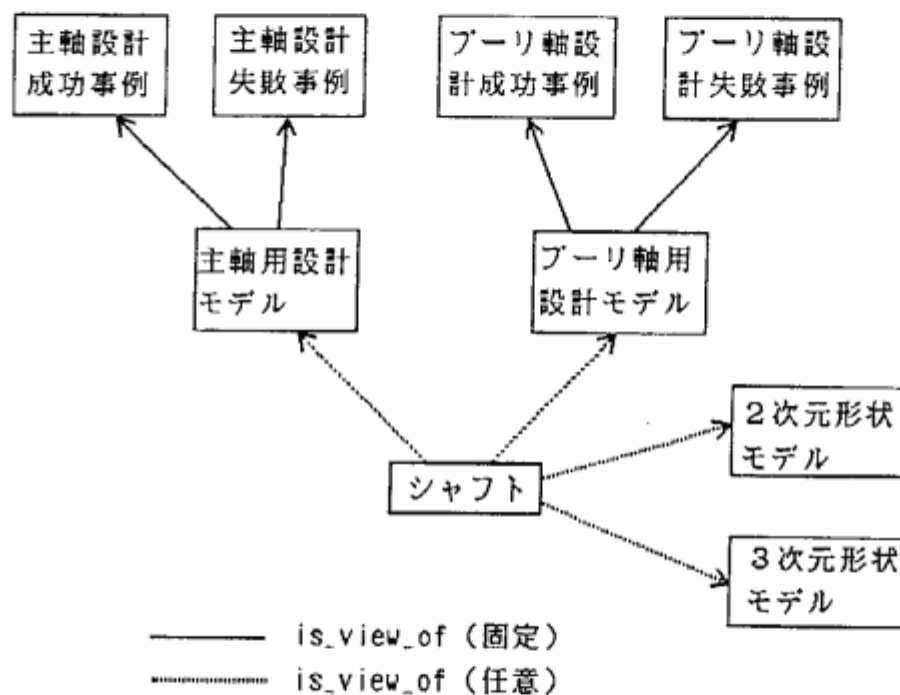


図2.2.4 設計モデルと事例の関係

事例の特徴記述は「仕様属性-値」の集合による。値の範囲、属性間関係などの記述も行なう。設計モデルの選択はその設計モデルを利用した設計事例の検索を介して行なうことができる。すなわち、設計仕様に最も近い設計事例を持つ設計モデルを選択すればよい。事例は基本的には、等しい仕様属性集合を持ち、属性値の近いものを選ぶことになる。事例の仕様属性を決めるにあたっては次の点に注意する必要がある。モデル構造（機構構造）が違っていても似た機能を持つものは、同じ仕様属性集合を持たせる。一方、似たモデル構造を持っていても、機能が大きく異なるものは、必然的に異なる仕様属性集合を持つこ

とになる。

事例は設計モデルとは独立に汎化階層を構成することにより、インデクスによる検索を可能にする。事例の獲得は、設計モデルを使って設計した結果を記録することにより行なわれる。実際に決定された、モデルの構成要素、パラメータ値を仕様と共に記録する。

設計対象に最も近い設計事例が検索されると、その事例が所属する設計モデルを使って設計を進めることになる。設計モデルの修正は、構成要素及びパラメータの変更までは許されるが、データ構造の変化までは許されない。データ構造の変化が必要な場合は、他の設計モデルを利用するか、新たに設計モデルを定義しなおさなければならない。構成要素・パラメータの変更は、設計モデル内に記述されている設計知識（設計式、設計ルールなど）を使って行われる。設計知識は信頼性の高いものを扱っており、制約、関係演算式、論理式の形で表現される。

2.2.3 まとめ

設計モデルライブラリによる編集設計について議論し、事例ベースとの関係についても考察した。データベース機能と事例ベース機能を導入することにより、信頼性の高い知識から低い知識まで適切に整理された形で表現することが可能に成り、設計知識・データの集積にも役立つと期待される。

参考文献

- [Imamura88] S. Imamura, T. Kojima, K. Inoue, H. Sekiguchi: A Study on the Object Oriented Product Model - Representation of Geometry and Dimension, Annals of the CIRP, Vol. 37/1, pp.127-130 (1988)
- [Imamura89] S. Imamura, T. Kojima: Public Knowledge Base for Intelligent Machine Design Systems, Preprints of the Third IFIP WG5.2 Workshop on Intelligent CAD, pp.38-47 (1989)
- [今村89] 今村 聡: 機械要素の属性モデル化、PIXEL'89-12, pp.138-141(1989)
- [Pierra88] G. Pierra: Conceptual Analysis and Functional Constraints for a Standard Concerning an European Standard Part Library, Report of the CAD/LIB /SWG3 Brussels meeting (1988)
- [井上88] 井上克己、永井保夫、藤井裕一、今村 聡、小島俊雄: 工作機械の設計手法の解析－旋盤の回転機能部品の設計、ICOT研究速報TM-0494 (1988)

2. 3 計装・計算機システム設計支援におけるニーズ分析

富士電機 萩原賢一

要旨

計装・計算機システムの概略設計では、過去の事例の中から制約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。また、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。このように、計装・計算機のシステム設計は類似事例の検索と変形に基づく事例推論と形式の上では共通点が多い。ここでは、計装・計算機のシステム設計に事例推論を応用するというニーズ寄りの視点で、知識表現と獲得方法、事例の取捨選択、知識ベースの管理について技術課題を提起する。

2. 3. 1 計装・計算機のシステム設計と事例推論

プラントや設備を監視・制御するためには、センサ・発信器などの計測機器、信号によって対象プラントを制御するアクチュエータ、計測信号や制御信号を光などで伝送する伝送装置、集った情報を処理する計算機、運転員と情報交換するマンマシンインタフェース装置などが必要である。

計装・計算機のシステム設計とは、要求された対象プラントの監視・制御機能を実現するのに必要なこれらの機器や装置について、どのような性能・規模のものをどのようにシステムとして組み立てるかをコストを考慮して決定してゆく作業である。

計装・計算機のシステム設計では、扱う情報が数値（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）など多種多様である。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。このため、従来からシステム設計業務の全体を支援することが困難であった。

一方で、システム構成部品が目的指向で作られているので、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ある程度類型化できる。このため概略設計では、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。そして詳細設計では、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。

ここに、計装・計算機のシステム設計において類似事例の検索と変形に基づく事例推論を応用した支援システムへの期待がある。

しかし、このような支援システムの実現可能性を評価するには、技術的に不透明な点が多い。ここでは、計装・計算機のシステム設計に事例推論を応用するというニーズ寄りの視点で、技術課題を提起したい。

2. 3. 2 技術課題

2. 3. 2. 1 知識の表現と獲得方法

事例推論応用システムでは、下記の2つの知識が重要である。これらの知識をどう表現し、どう獲得したらよいか考えてみる。

i) 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である。特に、類似度を判定するための下記のような知識が必要である。

a) インデクス属性値の関係知識

設計事例の機能、性能、コストなど多様な関係を表現するための知識である。

性能、コストなどは、定量的に表現されるので類似度を評価し易いが、機能は定性表現なので適切な尺度の設定方法を明らかにすることが望まれる。

b) インデクス属性の優先度知識

必ず一致すべきであるという制約やAよりはBが一致している事例を優先すべきであるといった相対比較の条件を表現するための知識である。

知識獲得の観点からは、a) は、システム構成部品のカタログなど既存のデータベースから抽出・合成して得られるので、比較的成本は低いと言えよう。b) は、設計者の意図と密接に関連するので獲得コストはかなり高い。一般に、ある事例を設計したときの意図を明確に述べ事例に付加することは設計者にとって苦痛だからだ。当面は対話型システムを作り、他の事例と比較しながら必要な優先度知識を獲得し詳細化してゆくことになる。

ii) 事例変形知識

制約を満たしながら、要求仕様に合わせて事例を変形するための知識である。特に、事例

変形の方法を選定するための下記のような知識が必要である。

a) 変形の許容範囲に関する知識

設計の意図として絶対に変形してはならない部分を直接表現する知識と、ある部分を変形した場合の影響を評価することで変形の許容範囲を間接的に表現する知識とがある。

b) 変形のコストに関する知識

変形の仕方が複数ある場合に優先度を判定するための知識。

知識獲得の観点からは、a)とb)共に、上記i)のb)と同様に設計者の意図と密接に関連するので獲得コストはかなり高い。やはり対話型システムで、他の事例と比較しながら必要な知識を獲得し詳細化してゆくことになろう。また、変形の影響を評価する上で深い知識(機器の増設コスト、信号伝送方式と処理性能との関係など)の利用が有効であると思われる。

2. 3. 2. 2 事例の取捨選択

一般に事例推論では、事例をなるべく多く蓄積することによって問題解決力の向上を図る。しかし、新しい事例が得られたとき、既に蓄積されている事例との関連を調べ、類似度判定知識を詳細化すると共に、典型例の取捨選択も必要である。

特に、計算機関連ではテクノロジーの変化が激しく、陳腐な事例は積極的に廃棄する必要がある。検索頻度などに応じて事例を整理したり、廃型情報を入力するとその部品を使っている事例を削除又は修正する仕組みが望まれる。

2. 3. 2. 3 知識ベースの管理

要求機能とコストなどのトレードオフを行うとき、どの項目を優先してシステム設計を進めるかは、設計者によって異なる。

そこで、複数の設計者が各人の設計事例を登録してゆくと、知識提供者の間で知識の矛盾が発生する可能性もある。そこで、設計の標準化を促進するという観点からも、矛盾する知識の関係を説明し、事例を分類・蓄積する手助けとなる仕組みが望まれる。

2. 4 製鉄分野におけるニーズ分析

新日本製鐵（株） エレクトロニクス研究所 久保真哉

要旨

事例ベース推論（CBR）はまだ新しい方法であり、確立された手法や使い方は存在しない。従って、どのようなニーズがどういう所にあるのかを調査することは有意義なことである。製鉄分野は、種類の異なるたくさんの小分野から構成される複合分野といえる。そこでこの節では、CBRの利用者という立場で、製鉄分野の中から4つの小分野、すなわち操業管理、生産設計、設備設計、システム開発をCBRの応用分野として取り上げて分析、検討する。次に検討の結果、明らかになったいくつかの研究課題を、事例の検索と問題解決の2つのフェーズに分けて示す。

2.4.1 はじめに

製鉄業は様々な技術の集積・複合化により成り立っている。近年、この分野においてもCIMの重要性、統合化システムの必要性が強く認識され、エキスパートシステムなど、従来システムの枠組みを越えたシステムが、次々に実用化されつつある。しかし、複雑で悪構造なシステムを従来言語やルールで完全に記述することは、一般に困難である。一方、多くの場合において、特にシステム化が困難な場合においては、業務に関わるデータや履歴はなんらかの形で長期に亘って保存されている。

そこで本節では、CBRによるシステム化という観点から、製鉄業の中のいくつかの分野に注目して、CBRのニーズを検討してみた。

2.4.2 CBRの応用分野

製鉄業には、製造分野だけでも、生産管理、製造技術、設備技術、整備技術など多くの業務分野がある。ここでは、CBRを適用可能な（あるいは、適用を望んでいる）小分野としてどのようなものがあるかを、次のような点に基づいて選定した。

- 多くの事例が存在していること
- 実際の業務で事例を参照していること
- システム化の要求が高いこと

以下では、操業管理、生産設計、設備設計、システム開発の4つの小分野を取り上げて、特に前者2つを中心に分析、検討を行う。

2.4.2.1 操業管理〔製造技術〕（制御問題）

操業管理とは、以下の例に示すような、製品の製造を計画的、効率的、安全に行うための業務分野である。

例： 高炉の操業管理

入力情報： 各種センサー情報

溶鉄温度、挿入物（鉄鋼石、コークス etc）落下速度、ガス分布etc.

制御情報

過去の操業アクション

事例： 主に参照するのは溶鉄温度や挿入物落下速度の時系列グラフに操業アクションを書き込んだものである。

出力情報： 操業アクション

特徴： 高炉は、アクションと結果の因果関係がはっきりしない世界のひとつである。

（高温で巨大なため、内部の化学反応を観察できない。）

（化学反応自体も複雑であり、十分なモデル化ができない。）

専門家もヒューリスティックな知識の背後にある理論を説明できない。

（具体的な状況を設定して始めて、推論してアクションを決定できる。）

膨大な量の事例が蓄えられている。

推論過程： 現状（温度変化etc.）の解釈

類似例の検索

検索項目の重み付け

類似例の絞り込み

類似例の修正

（現状と比較できるように、部分的にアクションを削除してみたり、
アクション量を修正したりする）

類似例の解釈

（ある状況下での因果関係の同定）

将来の状況の予測・評価

アクションの決定

知識： 現状（観測される事実）の解釈（大まかな見方）

グラフ情報の解釈（大まかな見方）

因果関係の推定（限定された条件の下）

事例の修正（限定された条件の下）

2.4.2.2 生産設計（生産管理）（設計問題）

生産設計とは、生産設備を有効に計画的に使い、要求された仕様を満たすような製品を製造するための生産方法や生産計画を立案する業務である。以下に示すのは、鋼材の品質を要求に沿うように製造するために、原料の成分や製造方法を設計する例である。

例： 品質設計

入力情報： 要求仕様

規格、強度、サイズ etc.

参照情報： 過去の製造方法と製品の性質のデータ

出力情報： 成分、製造方法、性質（予測値）

推論過程： 現状（要求仕様）の解釈

類似例の検索

（負事例も含む）

類似例の修正

（要求仕様に合うように成分値などを修正してみて品質を予測する）

設計の具体案の提案（複数）

（修正された類似例そのもの、あるいは合成）

設計の具体案の比較・評価

（矛盾するものもある）

特徴： 通常の規格からはずれるような場合に事例を参照する。

多くの要因が絡んでいて（対立していて）一様に設計できない。

少数の要因（成分値etc.）だけの変化の結果や、

多数の要因の少しだけの変化の結果は信頼性が高い。

従って、類似の設計例を修正して、解とすることが可能である。

膨大な量の事例が蓄えられている。

知識： 類似例の検索の観点

類似例の修正（添加成分と強度の関係 etc. 定式化されたものもある）

矛盾する具体案の評価方法、評価基準（生産コストetc.）

2.4.2.3 設備設計〔設備技術〕（設計問題）

設備設計とは、立案された生産計画、要求される製造方法を実現するための生産設備を設計することである。例として以下のようなものが挙げられる

例： 製造機械設計、工事設計（配線、配管システム etc.）

2.4.2.4 システム開発〔製造技術＋設備技術＋生産管理＋．．．〕（設計問題）

ここで言うシステム開発とは、これまでに述べてきたような様々な業務の支援または代行を目的としたソフトウェアシステムの開発をさす。

例：エキスパートシステム開発

C B Rを最近話題になっている上流のC A S Eとして使うことが考えられる。類似のソフトウェアの概念設計をする場合には、ある程度規格化された形式や、方法（ジェネリックタスクなど）を大まかなテンプレートとして使用することにより、設計負荷を軽減できる。

例えば、複数の工場で別々に高炉操業E Sや故障診断E Sを開発する場合には、各工場ごとに、操業方法や設備環境などが違うため、細かい点でシステム構成、知識獲得方法が異なり、全く同じ様にはシステム開発できないものの、大まかな形式や問題解決方法は、少し見方を変えれば同じものとみなすことができるため、この同じ部分を利用して概念設計が可能である。

従って、過去のシステム開発の事例を分析し、類似のシステム部品や開発方法をうまく整理してまとめることにより、C B Rを利用したシステム開発支援が可能になる。

2.4.3 研究課題

上述の応用事例を検討した結果、C B Rの研究・開発を進めるめるにあたって重要だと考えられるいくつかの課題を、C B Rシステムの2つのフェーズ、すなわち事例を検索するフェーズと、検索された事例を利用して直面している問題を解決するフェーズとに分けて、以下に記す。

2.4.3.1 事例の検索のフェーズ

事例を検索することに関しては以下のような研究課題があると考えられる。

- 事例に隠された意味を見出し、検索項目とする手法
 - 知識獲得方法（なぜこの事例が利用できるのかという知識の獲得方法）
- 事例の検索の観点（複数）の設定
 - 場合場合によって見方を変える方法
 - 事例の多面的（マルチビューポイント）な構造化
- 検索項目（インデックス）の重み付けの方法
 - 場合場合での項目の重要度の変化方法
- 参照事例を説明する情報の解釈
 - 事例適用に都合の良いように事例そのものを修正する方法
 - 矛盾した参照事例の競合解消
- 直面している問題状況を説明する情報（入力情報）の解釈
 - 事例の参照に都合の良いように情報を修正あるいは簡略化する方法
- 検索の方法（知識）そのものの事例ベース化

2.4.3.2 問題解決のフェーズ

問題を解決する方法に関連する研究課題は次のようなものがある。

- 参照事例を修正して、それをそのまま問題解決の解とする方法（変形型類推）
 - 数理解析、数理統計などの手法との融合方法
- 参照事例から抽出した情報（問題解決のノウハウ）を利用して、問題解決方法を決定して行く方法（誘導型類推）
 - 参照事例に隠された情報（意図、思惟、ノウハウ）を抽出・利用する方法
 - いつ、どこで、どの様に利用するのかを決定する方法
 - あるいは、この様な知識を獲得する方法

2.4.4 おわりに

最後にまとめとして、今回の製鉄分野におけるニーズ分析を通して気がついたことを以

下に箇条書する。

- CBRは、次のような時に有効である。
 - 問題解決に多くの要素が絡んでいて、ストレートな解法がないとき、あるいは、解法が一般的に定義できないとき。
 - しかし、事例があれば解法を説明できるとき。
 - 事例の修正、デフォルメ、検索ができるとき。
 - 多数の異質な事例があるとき。
- 事例があらかじめ明白に一般化や抽象化、あるいは、事例がきれいに分類、または整理できるようであれば無理にCBRを使う必要がない。
- 関係してそうな事例を検索して、それらに一般化等の操作を加えて有益な情報を得ることがCBRの本質である。
- 事例を参照する必要が生じたときに、その場に応じた事例を参照できなければならない。そのためには、事例を修正できなければならない。
 - 事例をその場に応じてデフォルメできなければならない。
 - 事例を複数の観点から（検索して）評価できなければならない。
- 少数の事例から推論するときは、事例の解釈や知識の抽出が重要である。
 - しかし、推論の裏付け知識がないと正当性が説明できない。
 - 逆に言えば、多くの事例があれば、それは正当性の根拠の一つとなる。
- 事例を参照するときの隠された意図を表現できなければならない。
 - これは難しいが、CBRの正当性も説明できると考えられる。
- 事例の分類方法として古典的な方法（クラスター分析や主座標分析etc.）も利用できる。
 - これらの方法は、事例ベースに対して単独に用いるのではなく、領域専門家（事例利用の知識を持った人）とのインタラクティブな対話の中で利用することによって、有効な事例ベースの構築のための知識や、問題解決の知識が得られる。

2.5 O A分野におけるニーズ分析

松下電器産業（株） 情報通信東京研究所 本山貴巳

要 旨

新しい問題解決パラダイムとして注目されている事例ベース推論 (Case-Based Reasoning, 以下 CBR) に対する、O A分野におけるニーズを調べた。O A分野においては、定式化された事務処理がほとんどで、推論を必要とする処理は少ない。しかし、O Aを経営まで広げると、大量の情報をいろいろな方向から分類し処理しなければならない分野があり、そういう分野では CBR は有効である。

2.5.1 はじめに

O A化された分野は、ほとんどが事務処理であり、定式化された処理手順がある。

しかし、定式化できない問題として、会議室などの共同スペースやワープロなどの共有資源の有効利用がある。また、経営ということまで含めればプロジェクトの発足や機材の購入などの意思決定がある。

前者は、予約システムとして、後者は意思決定支援システム (Decision Support System, 以下 DSS) としてシステム化されているが、ここでは、これらに CBR が利用できないかということを検討する。

2.5.2 予約

共同で利用する機材や、スペースの利用法としては、予約制が一般的である。

普通の予約制では、何らかの事情により、急に機材やスペースを使用したいことが生じた場合には対処できない。

優先順位をつけて、優先順位の高い方に予約を変更することが考えられる。それでも、予約を取り消された方はどうするか、また、優先順位はどのように付けるかなどの問題が生じる。特に、代わりになる機材や

スペースがある場合は、複雑で単純な優先順位だけでは対処できないことがある。

そこで、CBR の利用が考えられる。即ち、予約の変更を事例として蓄えておき、急な予約変更が生じたときにそれを、利用して予約変更を行う。例えば、単純な例として会議室の予約を考える。

推論を行うための会議室のデータとして以下のようなものを入れておく。

- 定員
- 場所
- 主な使用目的

そうして、ルールベースを用いて、予約や変更を行う。そして、ルールを用いても旨く行かないときに

- どの会議室を何人で何のために使用したかなどのデータ
- どのような状態からどのように、なぜ変更したかなどのデータ

などの事例を入れておき、CBR を行うということが考えられる。

2.5.3 意思決定支援

管理者がビジネス上の判断を行う場合に必要な情報を、必要な時に提供するシステムが経営情報システム (Management Information System) である。さらに、管理者が、自らコンピュータと対話しながら必要な加工し、参照しつつビジネス上の意思決定を行うためのサポートシステムが DSS である [森田 89]。

これらのシステムは 1960 年代から 70 年代にかけて盛んに構築されたが、次第に衰退していった。衰退した理由としては、以下のようなことがある [森田 89][石崎 89]。

1. 当時のデータベースの処理能力とデータが足りないため
2. 比較的簡単なモデルしか組むことができなかったため

1. に関しては、ATM(Automatic Teller Machin) や POS(Point of Sales) システムを利用して、情報を収集することである程度解決されているシステムがある。しかし、今度は情報が多くなり過ぎて、その中からどのように情報を取捨選択するかが問題となってきた。

2. に関しては、使用目的をある程度絞り込むことで解決を図っているシステムもある。しかし、根本的解決にはなっていない。

このような問題を解決するためには、同じ情報をいろんな観点から見ることができる方式が必要である。また、意思決定を支援するならば、情報を見せるだけでなくどうすればよいかを示唆して呉れる方が良い。

情報をどのように利用したかを事例として蓄え、それによって、情報を分類し直したり、その事例を使って推論を行って示唆を与えてくれる CBR は有効であると思われる。

2.5.4 おわりに

以上のように、O A の分野においても、CBR は有効であると思われる。経営の分野における意思決定支援においては、特に利用範囲が広いと思われる。

参考文献

- [森田 89] 森田 哲: 戦略情報システム, 講談社, 東京 (1989)
- [石崎 89] 石崎 純夫: 米国著名学者に聞く S I S の意義とその成功・失敗要因, 日経コンピュータ 1989.8.28 ,87-90(1989)

2. 6 法律分野におけるニーズ分析

ICOT 新田 克己

要旨

法律の問題解決を行うときには、法律として与えられた知識をそのまま用いて三段論法を行うのではなく、解釈を伴う法的推論が行われる。解釈を行うときには他の法律や学説や過去の判決例などを参考にすることがあるので、法的推論はある種の事例ベース推論ととらえることが可能である。毎年生み出される膨大な判例をいかに検索し、利用するかは法律分野での大きな問題となっている。ここでは、法律における事例の利用方法について考察し、コンピュータによる事例検索や論理構築などの法律分野のニーズ分析を行う。

2. 6. 1 はじめに

判決の予測、裁判における論理展開、法律にしたがった行動計画など、法律の専門家の活動は多岐にわたっている。専門家の仕事量が増えるにつれ、その一部をコンピュータによって支援してもらい、仕事の効率化を図りたいという要求が生じている。

法律の問題は、条文を論理式で表現し、法的結論をその論理式で証明する定理証明の問題として形式的に取り扱えるようにみえる。しかし、現実の裁判などでは、法律で与えられる知識は具体的な問題を解決するには不完全であり、法の解釈がしばしば問題となる。現実の問題に対して、法を正しく解釈して適用することができるかどうか専門家と素人の相違である。

解釈を行うためには、学説や他の法律や判例などが参照される。特に、判例は判決の予測を行う上での最も重要な情報源であり、法律事務所には判例集が備えられている。しかし、毎年、生成される判例は膨大であり、あまり体系的に整理されていないので意図する判決例を検索することはかなり面倒な作業である。商用の判例データベースを使った検索も、なかなか意図する判例を絞り切れないという不満もある。さらに検索された判例をどのように理解し、どのように利用するかは専門家によって異なっている。

このように法律の分野では、事例は豊富にあるものの、人手による整理と利用法には限界がある。そこで、事例ベース推論による専門家の業務支援の可能性を検討する。

2. 6. 2 法律と解釈

法律において意志決定をする場合、観測される事実に対し、法律を演繹的に適用すれば結論に至るように見える。しかし、現実にはそのような単純な図式では解決できない問題が存在する。

意志決定の場合、結論にいたるまでには、事実認定、法の適用、量刑決定（刑法の場合）などの段階がある。

〔事実認定〕の段階においては、解決したい事件の記述が必ずしも完全ではないという問題がある。観測される事実には見落としがあるし、同じものを観察しても、主観的な判断が入る可能性がある。どの事実があれば、どの法的概念に該当するかという必要十分条件は決定できないことが多いので、他の事件での認定結果などが参考にされる場合がある。

認定された事実に対し、法の〔適用〕を行うときには、法律に現れる用語の多義性や、概念の抽象性や、論理体系としての不完全性などの問題がある〔五十嵐 79〕。それにもかかわらず、法律が社会的規範として機能するのは、推論を行うときに〔解釈〕がなされるからである。

解釈とは、法の意味を具体化して認定された事実にあてはめる作業である。具体的には、用語に多義性があるときには、条文の目的に応じてその意味を個別に考えることに相当し、概念が抽象的なときには、概念を具体化する基準を生成することに相当し、適用できる法律間に矛盾があるときには、全体としての整合がとれるように調整規則を生成したり法律間の優先順位を生成したりすることに相当する。法律が時代遅れになったり、法律が不足したりするときには、既存の規則の修正や新たな規則の生成が行われる。

例えば、図 2.6.1 のように、観測事実が法律の概念 A に該当するかどうかの判断をする必要があるとする。観測事実から直接判断できる場合には問題がないが、A が抽象的な場合には、その判断がむずかしい。そのとき、解釈によってより具体的なルール

IF C THEN A

を臨時に作り、このルールを使って A に該当するかどうかを判断することがある。

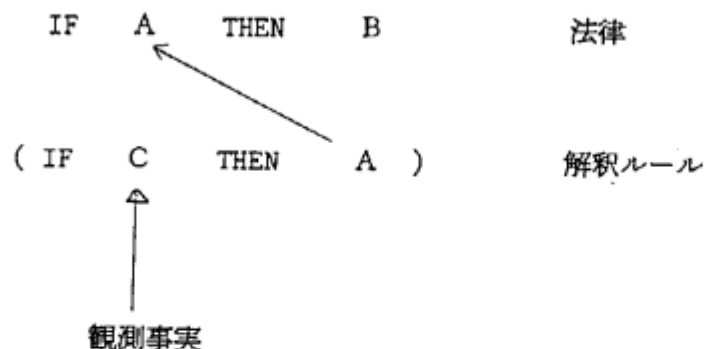


図 2.6.1 法の適用

解釈は、具体的な事件の価値判断という側面があるから、事件が与えられる前に法律を解釈しきれるものではない。解釈は「法の創造」という要素がある法的推論における最も困難な技術である。解釈がどこまで客観的に行えるものであるかは昔からの議論の対象であった[五十嵐 79]。

2. 6. 3 法的推論の種類

法律の専門家は、判決の予測、補償金の算出、財産の運用、論争における論理構築、契約文書作成、文献検索、などのいろいろな業務を行う。そのような業務遂行になされる法的推論を、吉野は正当化の推論と発見の推論に分類した[吉野 87]。正当化の推論とは、結論が前提条件から演繹的に導出されたものとして証明するもので、「意志決定」の機能と「論理構築」の機能がある。

「意志決定」とは、与えられた事件の記述に対して、法律や一般常識のルールを順次適用して、最終的な結論に達するものをいう。判決の予測や行動計画などの問題がこれに該当する。

「論理構築」とは、認定事実が特定の法律状態（ゴール）にある事を主張するため、その理由を生成するものをいう。裁判で争いが生じると、原告も被告も、観測される事実がどうしてその概念に該当するか（または該当しないか）という論理（説明）を展開する。これはゴールを達成するため、自分に有利な事実のみに焦点を当て、自分に有利な法律のみを適用しようとするものである。

判決文には原告/被告の論理とともに、裁判官の論理が記載される。そこに展開されている論理は、観測された事実と法律や判断ルールを適用し、最終結論に至っているように見える。しかし、実際には、裁判官は観測事実からある程度判決を仮定し、それを正当化するための説得の論理を構築していることがある[中野 85]。

判決予測などの意志決定の場合には、観測される事実があらかじめ用意されたどの典型的パターンに該当するかという分類問題として捉えられるのに対し、論理構築の場合には、分類問題としての要素以外に、観測される事実と典型的なパターンとの相違を発見してそのパターンに分類されないことを主張するような発見的な要素がある。

意思決定や論理構築のいずれの場合も問題となるのは解釈の問題である。新たな解釈を行うために発見の推論がなされる。解釈はできるだけ客観的な説明がつくようになされるが、その材料は法律内に限られないので、発想の着眼点を限定することは一般的には困難である。

2. 6. 4 事例を利用した推論

法律における推論の目標は、観測事実と結論をつなぐ論理（説明）を構築することである。

前述したように、論理を構築するには解釈という段階が必要なことがある。裁判では解釈は個々の裁判官の裁量にまかされているが、裁判官は解釈をまったく自由に行うわけ

ではなく、学説や法目的や産業政策や社会常識や他の法律や過去の判例などを考慮し、なるべく客観的な説明がつくように行う。

解釈ではバックに法律体系に関する理解と社会常識などの膨大な知識を必要とし、異なる価値判断の間の調節をしなくてはならないが、その方法は確立したものではない。例えば、民法の解釈には、文理解釈、論理解釈、目的論的解釈、立法者意志解釈などの方法があるとされており[遠藤 88]、どれを採用するかによって異なる結論が導かれる事が少なくない。これらの解釈の調整を行う原理は明確ではないし、それぞれの解釈を行うには、法律体系や個々の条文の趣旨や学説などを知識として、複雑な推論をしなければならない。

しかし、過去に類似の事件があったり、他の法律で類似の条文があったりするときには、そこで用いられた論理を拡張や類推によって利用して、解釈や判断の問題に対処することができる。現に、判決文には(1)過去に生成された命題や判断基準をそのまま利用したもの、(2)過去の命題や判断基準の趣旨を利用したもの、(3)説得するための根拠の補強として、過去の命題や判断基準を引用したものなどがあり[中野 85]、専門家も事例を利用した推論を行っている。(1)は過去に生成されたルールそのままの利用、(2)は過去のルールの拡張/類推、(3)は確信度の強化のための事例の列挙、である。ただし、判決の中の命題や判断基準の中で、どれが他の事件に応用できるかについては専門的な議論がある[中野 85]。

2. 6. 5 事例ベースシステム構築の必要性

最近のように、判例の数が増大すると、その検索にコストと時間がかかり、目的の判例がさがせるかどうかは運と金次第ということになりかねない。民事裁判などでは、訴訟費用は敗訴者が負担するのが原則であるが、弁護士費用はそれぞれの当事者が負担する。従って、裁判に勝っても経済的に打撃を受けることもある。優秀な弁護士を雇うことができ、判例検索に金をかけられるような社会的に強者である者のみが有利になるおそれがある。

また、裁判官もその専門家としての経験は自分の目に触れた範囲にとどまっている。次の世代にその経験が伝えることができないならば、これから多様化する社会的問題にどこまで対処できるか疑問視する声もある[夏井 90]。

判例を検索し、論理を構築するシステムの開発は、このような社会的な問題の解決の1つの方法となりうる。このようなシステムで構築された論理は、専門家の判断を拘束するものではなく、むしろ、専門家の支援として役に立つのではないかと思われる。

2. 6. 6 おわりに

法律分野における事例ベース推論の必要性について考察した。この分野では、実際に専門家による事例ベース推論が行われているので、それを機械化することは専門家の支援につながるものである。

ただし、判例を体系的に整理し、データベースとして満足に使えるようになるには、かなりの労力を要する。現行の判例データベースは特徴のある判例を集めたものであって、

ありふれた判決例はほとんど載っていない（これこそ事例ベースの構築に必要なものであるが）。また、判決例にのっている論理の流れが不十分なものがときどき見られるという問題点もある。

参考文献

[五十嵐 79] 五十嵐：法学入門，一粒社

[吉野 87] 吉野：法律エキスパートシステムに関する調査研究報告書，機械システム振興協会（1987）

[中野 85] 中野：判例とその読み方，有斐閣

[遠藤 88] 遠藤：民法（I），有斐閣双書

[夏井 90] 夏井：電子技術の進歩と司法の将来（上）（下），判例タイムズ No.711, 712(1990)

3. 事例ベース推論に関する文献調査

3. 1 H Y P O

I C O T 新田克己

要旨

K. Ashley and E. Risslandの開発したHYPOシステムについて説明する。これは裁判における原告と被告の論争を過去の判決例を用いて展開するものである。事例はdimensionという法的な性質を表現するデータ構造によって表現されている。新しい事件が与えられると、過去の事例から類似のものをいくつか検索して、近い順に並べ、それらを比較し対比することによって、原告と被告の間になされる論争を予測する。

3.1.1. はじめに

専門家が、問題状況を分析するのと、その状況を類似の状況と比較して何故同じか（違うか）を説明するのは別の仕事である。専門家が前者だけしかできないならば、その能力は疑わしい。後者の状況比較は、説明や議論や計画をするときの重要な要素である。それなくして類推はできない。状況間の類似点に焦点をあてるだけでなく、重要な相違点に焦点をあてることによって、一番よい事例を選んだり、最悪の事例を回避したり、それほどよくない事例から予測をしたりすることができる。このような状況の比較の重要性にもかかわらず、ほとんどのエキスパートシステムでは事例を表現したり、事例を比較する機能を持った制御構造を持ったりすることがなかった。C B R（事例ベース推論）の研究は、この欠点とその解決に焦点を当てるものである。

3.1.2. CBRの基本要素

事例に基づいた推論には3つの基本的な要素がある。

(1) 信用割当(Credit Assignment)

意志決定プログラムは、ある事実特徴があることを理由とし、ある事実特徴があることを無視して適切な事例を決定する。すなわち、事例の事実特徴のうちであるものに信用割当を行う。要するに、事例の決定とは、(a)信用割当をするための重要な特徴を選択し、(

b)選択された特徴をクラスタ化し、(c)それらを重み付けすることである。

(2) 前例による正当化(Precedential Justification)

以前の事例はあるクラスタの特徴を持っている。その中である特定の特徴があることによって意志決定がされる。このことを、類似の特徴を持った将来の事件について同じ結論を導くための正当化の根拠として扱うことができる。

(3) 議論(論理構築)(Arguments)

CBRでただ1つの正しい答えが見つかることはまれである。互いに矛盾する正当理由を生成できる。ある正当理由を他の正当理由よりよいものであるとする基準はある。しかし、よい議論を選んで意志決定をする場合は、CBRの出力は対案を表しているものととらえるべきである。

CBRで事例の探索ができるためには、事例の表現と記録と組織化、すなわち、「インデキシング」が重要である。

3.1.3. HYP0の役割

HYP0はトレードシークレット法を処理する。ユーザから状況を受け取って、それを解析し、事例データベース(CKB)から関連の事例を検索し、事実の重要性の割当を考慮し、現在の事件に関して事例を関係付け、焦点の事例を選択し、仮説を助言し、議論の骨格を提案し、その議論を事例の引用で正当化する。

HYP0の主な知識源はCKBとdimensionのライブラリである。dimensionとは、事実のクラスタとそれによって成り立つ法的結論の間の関係を表わすものである。dimensionは、事例とそれについての解析と議論へのインデックスを提供するだけでなく、その推論に関する事実状況の強さ(弱さ)を判定するメカニズムを提供する。dimensionは、事実を表す述語で表現された「前提条件」と、そのdimensionについて事例を強くしたり弱くしたりする特定の事実を選び出す「焦点スロット」と、焦点スロットの変化がその事例の強さに及ぼす影響を表す領域情報を持つ。

例えば、Disclose-Secrets(秘密の公開)というdimensionでは、焦点スロットは実際に秘密を公開した数である。この数が多いと、原告の立場が弱くなる。

3.1.4. HYP0の推論プロセス

新しい状況(current fact situation: CFS)の推論を例(図3.1.1)を用いて説明する。

1962年から1964年まで、原告（π） Crown Industryは自動ドアのためのパワーバック（小型電源）PX-121を開発した。Crownは被告（δ） Kawneer Co.が開発したPX-125が原告のトレードシークレットを盗用したとして訴えた。PX-121はすでに販売され、5つの公共の販売施設にインストールされている。Crownはサードパーティにそのパワーバックをすでに公開している。そして1963年から1965年にKawneerにも公開した。PX-121は以前の知られていない技術としての特徴は何にもなかった。KawneerはPX-125を開発するのに、1962年から1968年までかかった。

図3.1.1 CFSの例

子の例に対する推論は5段階で行われる。

（1）新しい問題状況(CFS)について、HYP Oはdimensionのライブラリを見て、case-analysis-recordを作る(図3.1.2)。このrecordには、(1)適用可能な事実の述語と、(2)適用可能なdimensionと、(3)ニアミスのdimensionと、(4)可能性のある犯罪（potential claim）と、(5)CKBにある関連事例が含まれる。

Applicable Factual Predicates:
exists-corporate-claimant,
exists-confidential-info, exists-disclosures ...

Applicable Dimensions: *Disclose-Secrets*

Near-Miss Dimensions:
Restricted-Disclose,
Competitive-Advantage,
Vertical-Knowledge

Potential Claims: Trade Secrets Misappropriation

Relevant CKB cites: See claim lattice, Figure 3 (a)

図3.1.2 Case-Analysis-Record

ニアミスdimensionとは、dimensionの前提条件の一部だけが満たされているものをいう。適用可能なdimensionとニアミスdimensionをあわせてD-listという。

（2）case-analysis-recordからclaim latticeを作る(図3.1.3(a))。このlatticeのルートはCFSであり、子供ノードはCFSのD-listのサブセットを共有する事例へのポイントを持つ。

latticeで多くのdimensionを共有するものほどルートに近い。そのD-listがニアミスdimension (図では*がついている) を含まないノードの中で最もルートに近いノードは、最も着目される事例most-on-point-cases(mopc)である。

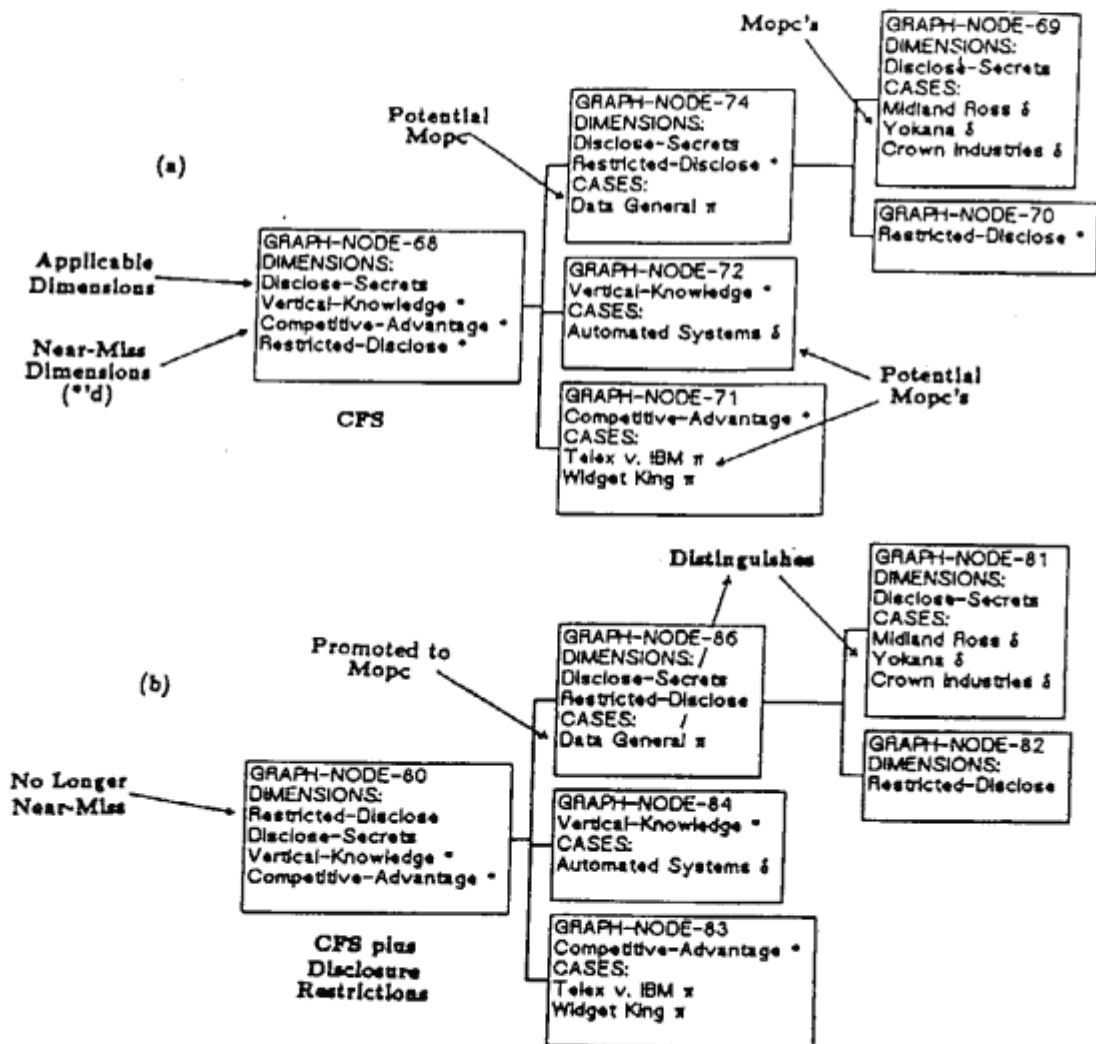


図3.1.3 2つのclaim lattice

(3) このlatticeを使って、双方の当事者のmopcを認識すると、被告有利のmopc (δ がついているもの) は、Midland Ross事件とYokana事件などであり、原告有利のmopc (π がついているもの) はない。Data General事件は π がついているが、Restricted-Discloseがニアミスだからmopcではない。

CFSと似ているがニアミスがあるものをpotential mopcという。Data Generalの事例が原告にとってのpotential mopcである。CFSの場合には「公開を受けた者がそれを秘密にしておくことに同意した」というRestricted-Discloseの前提条件が満たされないため、Restricted-Discloseはニアミスである。もしRestricted-DiscloseがCFSで成立していれば、Data Generalの事例は非常に原告に有利にはたらいていた。

(4) lattice中の事例を使って、CFSについて議論と反論を生成する。latticeの異なった枝別れが異なった議論に相当する。HYP Oの議論は、当事者1の指摘、当事者2の反論、当事者1の反駁という形で生成される。

- [a] $\hookleftarrow$ For Side 1: (Δ 's point)
 - Cite: Midland-Ross, Yokana
(Δ should win because Δ s in cited cases won where Π s disclosed secrets to outsiders.)
- [b] $\hookleftarrow$ For Side 2: (Π 's response to [a])
 - Distinguish: Midland-Ross
(In *Midland-Ross*, Π disclosed to 100 outsiders. Π in cfs disclosed to only 7 outsiders.)
 - Cite: Data General
(Π in *Data General* won eventhough Π disclosed to 6000 outsiders, more than in *Midland-Ross*.)
- [c] $\hookleftarrow$ For Side 1: (Δ 's rebuttal to [b])
 - Distinguish: Data General
(In *Data General* disclosees agreed to keep secrets but not so in cfs.)

図3.1.4 主張、反論、再反論の議論

この議論を説明する。

[a]は被告の主張である。Disclose-Secret (原告が秘密を他人に公開した) というdime

nsionが共通で被告が勝訴した事例(Midland Ross, Yokana)を引用して、被告の無罪を主張している。

HYP Oでは3つの観点から反論の種を探す。

- (i) CFSと引用事例に共通のdimensionの強さを比較する。
- (ii) 共有しないdimensionの強さ(弱さ)を評価する。
- (iii) もっとon pointな事例を探す。

[b]は原告の反論である。反論の観点(i)により、共通のdimensionの焦点スロット(公開した数)の値を比較し、disclose-secretsの領域情報から、Midland-Rossのほうが被告に有利であることを指摘(Crownは7つしか公開していないのに、Midland-Rossは100も公開している)。さらに、Data Generalの事例ではMidland-Rossより多くの公開をしているのに、原告が勝訴したことを指摘する。

[c]は被告の再反論である。上記の反論の観点(ii)により、原告を有利にしているdimensionであるRestricted Disclose(公開を受けた者が秘密の保持に同意)がData Generalには成立していてもCFSではニアミスにすぎないことを指摘する。

(5) 当事者の立場の強さと弱さのテストに有効な仮説を作る。

原告のpotential mopcであるData Generalと被告のmopcであるMidland-Rossの相対的な位置を比較して、原告の公開が制限されていれば(Restricted-Disclose)原告の立場が強くなることがわかる(反論の観点(iii)により、Data Generalが原告の立場を強くする)。図3.1.3(b)がCFSを仮説として変形したときのlatticeを表す。HYP Oの原告のための再反論が以下のように構築される。

[d] ← For Side 2: (II's response to [a])
Cite: Data General
(II should win because in *Data General*, II won where II disclosed secrets and discloseses agreed to keep disclosures secret.)
Distinguish: Midland-Ross, Yokana
(*Data General* is more on point than these cases where discloseses did not agree to keep disclosures secret.)

3.1.5. 事例の比較と対比

HYP Oでは事例を3つのレベルで比較と対比を行っていることになる。

(1) 事実レベル

CFSをlattice中の関連の事例と比較し、共通のdimensionで示された重要な事実に焦点をあてている。

引用事例と区別することで反論するときに対比を行う（反論方法(i)(ii)）。

(2) 正当化レベル

CFSにとってよい先例を探すのに、latticeを使って事例同士を互いに比較し、どれだけルートに近いのか、議論においてどれだけ有効か（(iii)の反論）、議論においてどれだけ潜在的に有効かを評価している。

(3) 議論レベル

提議を支持する議論を評価するのに、有利なmopcと不利なmopcの関係を比較している。

図3.1.3(a)と図3.1.3(b)のmopcの比較が一種の評価関数となっている。

3.1.6 おわりに

CBRの3つの要素（信用割当、正当化、議論生成）について述べ、HYPOについて説明した。

例題でHYPOは裁判所の判断（Midland-Rossを引用して被告の勝訴）と一致した。裁判所もHYPOと同じような比較と対比を実際に行っている。

3. 2 P R O D I G Y

東京工業大学 山村雅幸

要 旨

本節は文献[Carbonell 88]の調査報告である。まず3.2.1～3.2.7節にかけて、本文献の概要を示し、3.2.8節に担当者の考察を示す。本文献の要旨を次に示す。

真の専門知識には、主な精神活動としての知識主導型推論と、領域知識のギャップを橋渡しする、すなわち予期しない状況に対処するための一般問題解決への後退とが必要である。本論文では、誘導型類推(derivational analogy)、すなわち豊富なエピソード的知識によって駆動される事例ベースの再構築的推論と、一般問題解決・計画アーキテクチャとを統合する。主な学習機構は、探索主導型問題解決エピソードの誘導トレースの抽出・組織化から成る。それは成功・失敗条件の明示的な正当化パスによって拡張され、インデクスされる。この新しいエピソード的事例知識の蓄積によって、弁明構造の本質的部分を共有する問題の推論において、誘導型類推機構が駆動される。この手法をPRODIGY問題解決システム上に実装し、線形代数の領域で例示する。

3.2.1 Introduction

学習された知識がどういう形式で蓄えられるかにはいろいろあるが、いずれも現在の領域知識をより効率的な形にコンパイルする。その理想は、教科書的知識を実践的知識に変換することにある。

アナロジーとC B Rはコインの裏表である。いずれも、複雑な問題解決をガイドするためにカプセル化したエピソード的知識を利用する。アナロジーではmodifying, adapting, verifyingに重点がおかれるが、C B Rではorganization, hierarchy indexing, retrievalに重点がおかれる。

ここでは、一般問題解決・計画アーキテクチャPRODIGYに誘導型類推を統合する。

3.2.2 The overall picture

PRODIGYの知識源は次の4種類からなる。第1に、static domain knowledgeとは、基礎的な述語と対象の公理化である。第2に、working domain knowledgeとは、合法的な行動すなわち作用素・ルール・制御則からなる。第3に、solvable classes of problemsとは、上の2つで解ける問題のカテゴリー化である。第4に、episodic knowledge libraryとは、事例の集積であるが、これについては開発途上である。

working domain knowledgeの例を示す。次の式は積木の世界におけるPUT-DOWN作用素の記述である。

```
(PUT-DOWN (params (<ob>))
  (preconds (and (object <ob>)(holding <ob>)))
  (effects ((del (holding <ob>))(add (clear <ob>))(add (on-table <ob>)))))
```

3.2.3 The basic analogical replay method

PRODIGYにおける類推は次のステップによって行われる。ただし4)は未実装である。

- 1) 教師が問題を提示し、解の構築時に代替案の選択を通じてプランナーをガイドする。
教師による初期のガイドは次の3レベルで提供される。
 - 1) 現在のゴールの集合から、次のサブゴールを選択する。
 - 2) ゴールを達成する作用素を選択する。
 - 3) 変数の束縛を選択する。

このステップのキーは、教師がその選択の正当化を提示することにある。

- 2) 新しい問題を解くときに、PRODIGYは過去に解決した事例ライブラリを考慮する。
新しい解の生成は、過去の解の生成に影響した正当化によってガイドされる。
想起した事例の修正がすべて失敗したときに限り、代替案を探索する。
- 3) 新しい解をその正当化とともにエピソード的事例ライブラリに格納する。
- 4) 誘導トレース、すなわち事例の集合に一般化を適用して、一般プランを生成する。

3.2.4 An example from elementary linear algebra

3.2.4.1 The initial state representation

線形代数における初期状態記述の例を示す。

```
(dimension 3 4)
|2 -1 3||x1| | 2| (mat 1 1 2) (mat 1 2 -1) (mat 1 3 3) (mat 1 4 2)
|1 2 -2||x2|=| 1| => (mat 2 1 1) (mat 2 2 2) (mat 2 3 -2) (mat 2 4 1)
|4 1 2||x3| |-3| (mat 3 1 4) (mat 3 2 1) (mat 3 3 2) (mat 3 4 -3)
```

ガウスの消去法における目標状態では述語triangular upperが真となる。

3.2.4.2 The original search tree

事例がないとき、PRODIGYは拡張and-or探索木を生成して、網羅的探索を行う。

3.2.4.3 Static domain knowledge

線形代数におけるstatic domain knowledgeの例を示す。

```
(PRAME matrix (is-a data-structure)
  (definition (def1 (table-of objects))
    (def2 (ordered-set-of vectors))
    (def3 (map (cprod setA setB) objects))))
(characteristics
  (dimensions (number 2) (name row column))
  (diagonal (set (elt (row i column j))(such-that (equal i j))))))
```

3.2.4.4 Working domain knowledge

線形代数におけるworking domain knowledgeの例を示す。

```
(TRIANGULIZE (params ())
  (preconds (and (demension <n> <m>) (pred-f <n1> <n>)
    (forall (<k>) (range-f <k> 1 <n1>)
      (and (workrow <r> <k>)
        (forall (<col>) (range-f <col> <st> <k>)
          (clearredelt <r> <col> 0)))))))
```

```

(effects ((add (triangular upper))))
(justifs ((assumes INTERCHANGE-ROWS)
          (refers-to ((upper-triangular (definition def1))))
          (probl-class ((system-of-equations))))))

```

3.2.4.5 Solvable classes of problems

線形代数における solvable classes は次の図に示す階層構造をもつ。

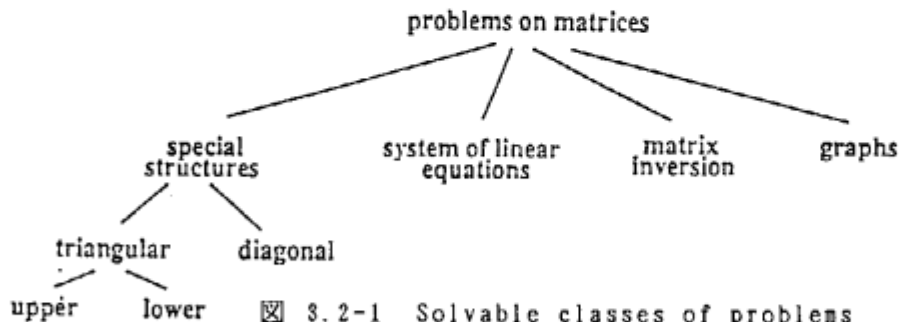


図 3.2-1 Solvable classes of problems

3.2.5 Introducing justifications

3.2.5.1 Justifications at the operator level

PRODIGYの作用素に次の3つの正当化のフィールドを付加した。第1に、assumesとは、この領域で妥当であると仮定される作用素のリストである。第2に、refers-toとは、この作用素の基礎となる静的領域知識のリストである。第3に、prob-classとは、この作用素が適用可能な問題クラスである。

3.2.5.2 Justifications at the goal choice

PRODIGYは線形プランナーと異なり、ゴールを再配置する。そのときのゴールの選択理由は次の5つからなる。第1に、uniqueとは、唯一可能な代替案であったことをさす。第2に、stackとは、最後に配置(post)されたことをさす。stackはデフォルトである。第3に、arbitraryとは、無作為に選択されたことをさす。第4にprotectとは、このサブゴールを解く過程で破壊されないことが仮定できるような、既に達成されたゴールのリストにあったことをさす。第5に、necess-beforeとは、このサブゴールの達成がその達成に必要なようなサブゴールのリストにあったことをさす。

3.2.5.3 Justifications at the choice of alternatives among operators

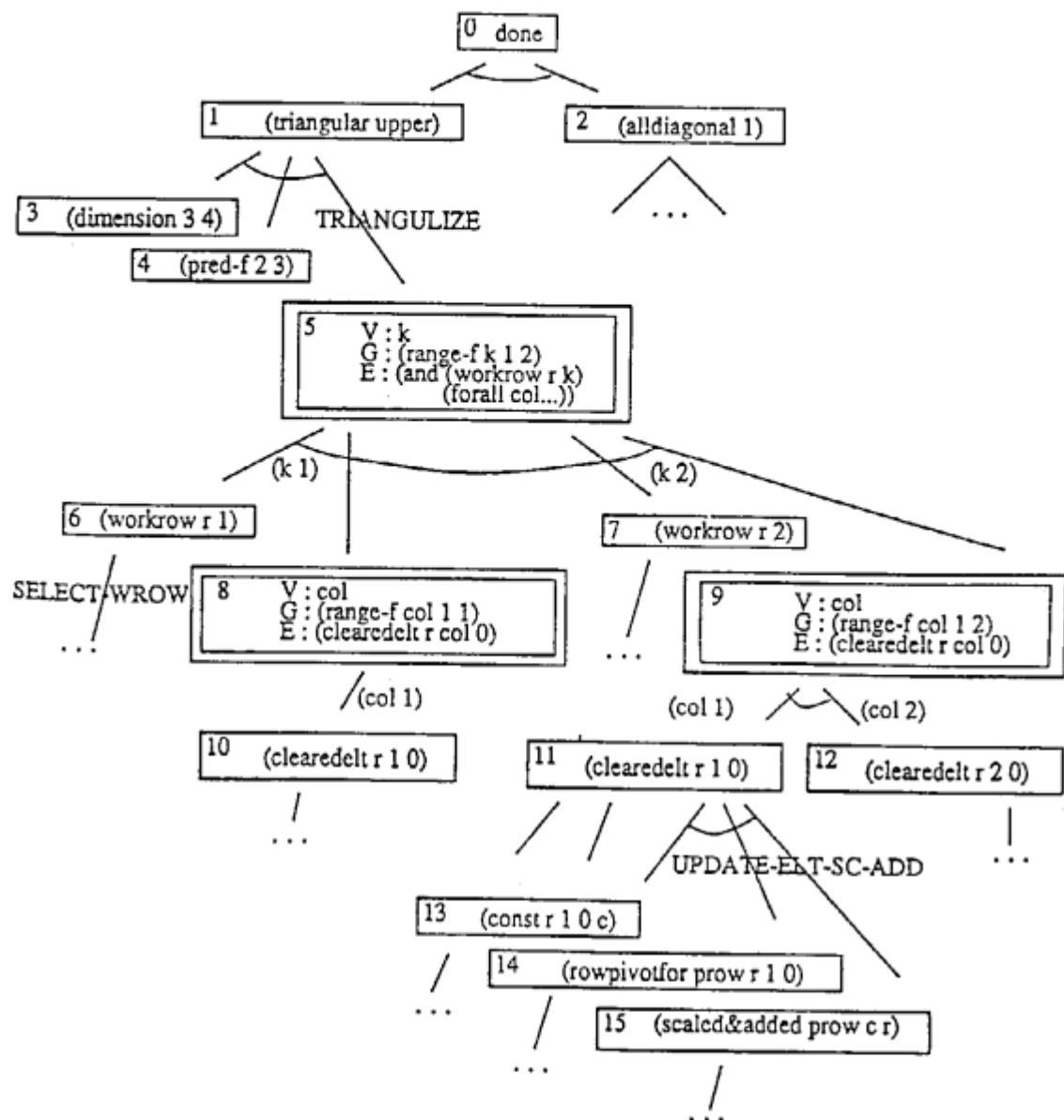
作用素の選択理由は次の3つからなる。第1に、uniqueとは、唯一可能な作用素であったことをさす。uniqueの場合にのみ類推が適用される。第2に、arbitraryとは、無作為に選択したことをさす。第3に、failureとは、作用素とそれがなぜ失敗したかの理由のリストからなる。失敗とは無限ループ等である。failureが成立するとき、類推は適用されない。

3.2.5.4 Justifications for the bindings of variables

失敗した変数束縛のみを記憶する。

3.2.5.5 An example of a justified solution path

ガウスの消去法の例を次の図に示す。



At node	Goal selected	Selected among goals	Justification
0	1	1,2	(necess-bef 2)
1	3	3,4,5,2	(necess-bef 4 binding)
	4	4,5,2	(necess-bef 5 binding)
	5	5,2	(stack)
5	7	6,7,8,9,2	(necess-bef 9 binding)
	9	6,8,9,2	(arbitrary 6,8)
	11	6,8,11,12,2	(arbitrary 6,8,12)
	6	6,8,12,2	(necess-bef 8 binding)
	8	8,12,2	(necess-bef 12)
	10	10,12,2	(necess-bef 12)
...			(protect 11)

図 3.2-2 ガウスの消去法の例

3.2.6 Analogical reconstruction and learning: The algorithm

類推を知的に使うには次の2つの接近がある。第1に、satisficing approachとは、プランニングに必要な努力を最小化することである。第2に、optimizing approachとは、プランの質を最大化することである。現在はsatisficingのみが実装されている。

PRODIGYは類推においてすべての正当化を検証する。正当化が妥当なときは単に解をコピーする。正当化が妥当でないときは次の3つの代替案がある。第1に、失敗した選択点で再計画する。成功したらその選択を解に代入する。第2に、失敗した条件を最優先のゴールとして再達成する。それを解に挿入する。第3に、部分的に正当化されない行動の推敲を何等かの形で実験する。成功したら、それにあわせてシステムの知識を洗練する。

3.2.6.1 Some learning examples

類推による学習の例を示す。

先の線形代数の例題では、ピボット行の選択がある。この選択理由が無作為であったとする。何題かの例題を解くうち、ピボットが0のものを選択するかもしれない。このとき、解は乗算乗数の選択で失敗する。このときシステムは、失敗した前提条件、すなわち除数は0を除くことから誘導パスを遡って、問題の変数が束縛された点に移動する。この場合それはピボット行を選択する作用素である。こうして、新しく正しい版の作用素が学習される。それは失敗した前提条件を付加したCHOOSE-ROW-PIVOT作用素である。獲得された知識は、形式的証明が構成されるまでは推測とみなされる。この場合はスカラー計算から直接証明することができる。

3.2.7 Concluding Remarks

本論文では、汎用問題解決アーキテクチャPRODIGYへの、誘導型類推の完全な統合化について報告した。そこでは、作用素ベースの領域知識のPRODIGY上へのコード化、エピソード的な領域の専門知識の因果的正当化構造によるカプセル化、それらの知識による問題解決行動のガイド、推測された正当化の妥当性の実験的検証、さらにまた正当化が失敗したときあるいはエピソード的専門知識が欠如しているときの汎用問題解決器への後退が議論された。

(以上本文概要)

3.2.8 検討

著者は、類推ではmodifyingに重点がおかれ、CBRではindexingに重点がおかれると位置付けており、本文献は、PRODIGYの上での誘導型類推の実現に力点をおいて述べている。このため、CBRシステムとしては、想起の部分を欠く不完全なものである。しかしながら、現存するCBRシステムでは、極めて単純な修正しか行われなことを考えると、これらの技術は相補的であり、統合化されるべきものと結論できるだろう。このために、誘導型類推は有望な要素技術であり、今後の発展が期待される。

(以上)

参考文献

[Carbonell 88] Carbonell, J. and Veloso, M. "Integrating derivational analogy into a general problem solving architecture." Proc. DARPA CBR Workshop (1988)

3. 3 CYCLOPS

東京工業大学 小林重信

Case Based Reasoning in CYCLOPS, a Design Problem Solver

by D. Navinchandra (CMU)

from Proc. DARPA Workshop on CBR Workshop, pp. 286-301(1988)

要 旨

CYCLOPS は、CMU の D. Navinchandra によってつくられた景観設計問題領域を対象とする CBR システムである。本論文では、潜在的な問題を認識したり、設計問題を修復するために過去の事例を利用する設計問題解決器を提案している。本問題解決器は、①問題を見つけること、②目標および副目標を設定すること、③対象問題に直接照合あるいは類似照合する適当な事例を見つけること、④現在の問題の説明を利用して、事例から関係する知識を抽出すること、⑤すべての副目標が満足されるまでこのプロセスを続けること、の5つのプロセスからなる。

3.3.1. CYCLOPSの概要

CYCLOPSは景観の計画と設計の領域を対象とし、レイアウト問題を取り扱う。CYCLOPSは設計案を生成し、問題を解決する。CYCLOPSで用いている基本的な探索メカニズムは、設計案を生成し、それを事例に基づく推論モジュールにより検査することである。設計案の特徴に基づいて、事例が想起される。

事例の検索は、事例（ベース）の特徴と現在の設計問題（ターゲット）の特徴との間の照合により行われる。照合には3種類ある。

①直接照合(Direct Match)

ターゲットと同じ属性をもつベースが検索される。

②緩和(Relaxed Match)

概念階層を探索して、ターゲットと意味的に照合する属性をもつベースが検索される。

③系統性に基づく照合(Systematicity-based Match)

Gentnerの系統性原理に従い、ターゲットと共通する因果関係をできるだけ多くもつベースが検索される。

問題解決過程は以下のステップから成り立っている。

①問題の発見

②目標と副目標の設定

③適切な事例の発見

④事例の部分の応用による目標と副目標の達成

この過程はすべての副目標に対して、再帰的に行われる。この過程を実現するために、CYCLOPSではDemand Posting（要求掲示）という問題解決技法を提案している。

3.3.2. CYCLOPSにおけるDemand Posting

“傾斜面に家を建てる”という問題を例に、CYCLOPSにおけるDemand Postingの考え方を示す。

- 1) まず“傾斜面に直接家を建てる”ことを考える。ここで、以前に、“斜面に緑のブロックを置いたとき、そのブロックが傾いてしまった”事例を想起する。その事例における状況説明は、“斜面に直接ブロックを置いたから、そのブロックは傾いた”とある。この説明を用いて、問題解決器は“家が傾くであろう”と推論する。
- 2) “家が傾くであろう”ことを認識すると、領域知識を用いることにより、“家が傾くことは問題であり、この問題は除去されなければならない”ことを推論する。
- 3) “家が傾くことを阻止する方法があるか？”という目標を設定する。この目標をインデックスとして、適切な事例を見つけようとする。しかし、これには失敗したとする。
- 4) 問題となった原因を見つけようとする。ブロックの事例に戻り、その説明を調べると、家が傾く原因は、(a)家は地面に建っていること、および(b)その地面は傾斜していること、にあることを知る。
- 5) 2つの原因を、選言的な副目標として設定する。すなわち、(a)“家を直接地面に建てる方法があるか”および(b)“地面を平坦にする方法があるか”を副目標とする。
- 6) 副目標(a)については、タイのある村の家の風景を事例として想起する。その風景には支柱の上に小屋がある。その住民は洪水と地面の湿気の問題をもっていて、その解決策として、家を地面から離し、パイルの上に建てている、という説明がなされている。
- 7) タイの事例の説明に基づいて、問題解決器は家をパイルの上に建てることを決定する。この決定によって、他の新たな問題が発生することがないことを確認する。もし何も問題がなくて、連言的副目標がすべて充足されたならば、問題は解決されたことになる。

このように、Demand Postingは、目標指向の推論を展開する。適用可能な事例を見出し、適切なアクション(ex. 支柱を利用する)を適用すると、システムはすでになされたすべての推論を適切に更新する必要がある。この例では、パイルの上に家を建てることを決めると、家は地面の上にあるという元の事実は真でなくなる。その結果、緑のブロックの事例と傾斜を推論するのに使われた規則は支持理由を失うので棄却される。このような信念維持機能がDemand Postingに取り入れられている。また、システムはすべての選言的副

目標は保持しており、あるパスで解を見つけるのに失敗したら別のパスについて調べるためにバックトラックする。

すなわち、CYCLOPSはDemand Postingの手法によりアナロジーを導きだし、それを利用して問題解決を行う。CYCLOPSが過去の事例と直接照合しないような問題に直面した時は、アナロジーによる推論を試みる。アナロジーを導くためには、過去の事例がターゲット問題の特徴と大きく異なっているにもかかわらず照合できるような能力が要求される。

上の例では、家の傾斜の問題を解決するのに、タイのバイルの上の小屋という事例を用いたが、タイの事例の目的は洪水を防ぐということにあり、家の傾斜の問題とは目的が異なっていることに注目すべきである。それらの間の照合は、表面的な特徴の照合ではなく、もっと深い理解に基づくものである。CYCLOPSはこのような推論を過去の事例の説明に含まれている副目標に対して照合を行うことにより達成している。Demand Postingにおけるこの照合の部分を副目標照合と呼んでいる。

3.3.3. CYCLOPSにおける事例の表現と利用

CYCLOPSでは、事例とは観察事実 (observation) または解法 (solution) である。観察事実は条件 (condition) と効果 (effects) から構成される。事例には説明されない観察事実 (unexplained-observations) と説明される観察事実 (explained-observations) の2つがある。解法は問題状況と解からなり、いずれも説明される観察事実である。

“傾斜面に家を建てる”問題に対して、つぎの事例と規則が想起される。

事例#123

条件： ブロック1は緑色である ∧ ブロック1は傾斜面の上にある

効果： ブロック1は傾いている。

説明： ブロック1は地面の上にある ∧ 地面が傾斜している

→ ブロック1は傾いている。

規則#668

IF： 家が傾いている。

THEN： “傾いた家は好ましくない状況にある”

問題の条件部分と前例が照合し、“傾斜面に家を建てれば、家は傾く”と推論する。照合に際し、“家”と“ブロック1”は、共に直方体であるという概念階層からの知識および事例の説明が利用される。

事例#123の後、規則#668が発火し、“傾いた家は好ましくない状況にある”を推論する。“好ましくない状況にある”が推論されると、この問題を解決するために、Demand Posting手法が適用される。Demand Postingは、依存関係追跡と副目標照合よりなり、以下のように問題解決が展開される。

Q1: "傾いた家は好ましくない状況にある"ことを無効にする事例は存在するか?

もし事例が見つからなければ、問題の原因が新しい質問として生成される。規則#668より、原因は"家が傾いている"ことがわかる。つぎの質問が生成される。

Q2: "家が傾いている"ことを無効にする事例は存在するか?

やはり事例が見つからなければ、もう一度現在の問題の原因が検索される。事例#123の説明より、家が傾いている原因は、"家が地面の上に建てられている"ことおよび"地面が傾斜している"ことにあることがわかる。2つの原因のうちどちらかを否定すれば、問題を解決できるので、つぎの2つの質問が生成される。

Q3a: "家が地面の上に建てられている"ことを無効にする事例は存在するか?

Q3b: "地面が傾斜している"ことを無効にする事例は存在するか?

Q3aに対しては、タイで洪水の対策として家をパイル（長い支持棒）の上に建てているというつぎの事例が見つかったとする。

事例#1002

初期状況

条件: 小屋は地面の上にある ∧ 地面は冠水している

∧ 小屋はタイにある

効果: 小屋は浸水している ∧ 小屋は好ましくない状況にある

説明: (B1) 小屋は浸水している → 小屋は好ましくない状況にある

(B2) 小屋は地面の上にある ∧ 地面は冠水している

→ 小屋は浸水している

解法:

処置: パイルの上に小屋を置く

効果: 小屋は好ましくない状況にはない

説明: (B3) 小屋は浸水しなくなる ∧ (B1) は真である

→ 小屋は好ましくない状況にはない

(B4) 小屋は地面の上にはない ∧ (B2) は真である

→ 小屋は浸水していない

(B5) 小屋は元は地面の上にあった ∧ パイルの上に小屋を置く処置を取った

→ 小屋は地面の上にはない

この事例は説明された解事例 (explained-solution case) である。

事例#1002の副目標B5が質問Q3aと照合して、B5の因果関係はベースからターゲットに変換される。その結果、"家をパイルの上に建てる"という解が導かれる。

このように、Demand Postingプロセスは

- ・問題を発見するのに事例を用いる
- ・新しい事例を発見するのにその問題を用いる。
- ・目標と副目標の依存関係を保持する。
- ・類推的に問題を解決する。

3.3.4. まとめ

CYCLOPSの特徴と限界はつぎのように要約される。

- ・学習

Demand Postingプロセスが問題解決に使われると、結果として得られる状況－解構造が将来の問題に対する事例として再利用することができる。

- ・完全性の仮定

事例に記述されている因果関係は独立かつ完全であること、および完全な文脈をもつことを仮定している。

- ・閉世界仮説

CYCLOPSでは類推の妥当性を検証していない。類推結果が何か新しい問題を招くことを示す事例がなければ、その類推は正しいとみなす。すなわち、事例知識ベースの閉世界仮説を仮定している。

- ・説明構造への依存性

ベースからターゲットへの知識の移行は、ベース中の説明構造により限定される。説明構造を予め決定しておくことは、事例の再利用の方法を予め決定しておくことになり、それは事例の見方を固定する。将来的には事例が用いられる目的に対応して、必要に応じて説明を導く必要がある。それには、purpose-directed analogy[Kedar-Carbeli 85]が有用と思われる。

3.4 Opportunistic Memory

京都大学工学部 榎木 哲夫

Opportunistic Memory : Storing and Recalling Suspended Goals

by K. J. Hammond (Chicago University)

from Proc. of DARPA CBR Workshop. pp. 154-168. 1988.

要旨

opportunism (opportunistic : 臨機応変型・機会主義的) は、プランニングの領域において従来盛んに議論されてきたが、その多くは、プラン生成時の opportunism であり、プラン実行時 (plan execution) の opportunism については、あまり議論の対象にはされてこなかった。本論は、プラン実行時に見出される機会 (opportunities) を、プラン生成時に推論するための opportunistic planning モデルに関する所見をまとめたものである。本モデルは以下の3つの部分から構成される。

- ・ a planning-time mechanism : blocked goal を、後に想起されるべく記憶内に位置づける。
- ・ an understanding system : 対象世界の認識 (parsing the world) 時に生じた「中断された目標 (suspended goal)」を想起する。
- ・ an execution-time process : 機会 (opportunities) の評価や新たに想起された goals の統合 (merging)。

3.4.1. Planning and Acting

従来のプランニングは実行前に、複数の goal を遂行するプランを前もって作成するがこれは非現実的 (a closed world and complete knowledge of operators). これに対し、近年のプランニング研究では、プラン及びその実行を1つのプロセスに統合するのが主流。

ex. ' Situated activity " : reflexive response to external cues.

実行時のプラン修正、プラン失敗からの復旧 (repair)

- ・ Adaptive Planning (Alterman)
- ・ RAP (Reactive Action Packets) (Firby)
- ・ Case-Based Planning (Hammond)
- ・ ' active planners ' : 変化する対象世界の中で、プランを生成するとともに、そのプランを実行する。
 - ・ TRUCKER : pick-up and delivery scheduling
 - ・ RUNNER : "お違い" のプランナー

3.4.2. Opportunistic Memory

opportunities の組織化・認識・掘りおこしにエピソード記憶を使う。エピソード記憶はプラン生成・実行時の一種の共有黑板 (blackboard) の役割を担うことになる。

プランナーの opportunities の認識は、エピソード記憶の良悪により決まることになる。

3.4.3. An Example

RUNNER からの例題

「John は朝、朝食の用意をしていると、オレンジジュースをきらしていることに気付いた。仕事に遅れるので買いに行くことはできなかった。仕事からの帰り、seven-eleven を通りかかったところ、John はオレンジジュースをきらしていることに付き、立ち寄って買い求め、家路についた」

Points !

- ・プラン実行中に新たなプラン目標が生じ preplanning が不可能。
- ・ある目標達成を成し遂げる前に、プランを停止することができる。
- ・プランが停止されると同時に、その目標達成のための前提条件を認識する。例えば目標を成就できなくても、そのためのプランが必要とするものを推し量ることができる。このことが、プラン実行時に opportunity を見出すトリガーとなる。

3.4.4. Opportunistic Planning

プランニング時の opportunism に関する 2 つの視点。

① Hayes-Roth & Hayes-Roth の視点。(opportunistic planning)

プラン生成時の opportunism,

opportunities に基づくプランニング戦略のシフト

黑板 + specialists (知識源)

- ・ domain level plan developer.
- ・ strategic operators (目標のクラスタリング etc.) 用件 (errands) を、場所でグループ化したり、優先度でグループ化したり。

プラン実行時の opportunism を扱ったものではない。

② Birubbaum and Collins (1984) の視点。

プラン実行時の opportunism にふれている。

各目標は、各々自律的に現況の中に自らの充足の機会をうかがう処理ユニット。

中断された目標をその都度 store して、activate する方法では、予期せぬ opportunities を見逃すことになる。

プラン中断時に、その充足条件をインデックスとして store しておくのでは、インデックス自体大変複雑なものになるし、刻々現況との照合を繰り返さねばならず、真の意味での opportunism とはいえない。

③ Our Stance :

execution-time opportunistic behavior like Beinbaum、しかし Beinbaum の考え方では

a memory of goals.

the planner's memory and understanding of the world.

の両者の分離を余儀なくするが、その必要はなく、両者は同一の memory 上で事足りる。

例 家を建てかえているときに新しいハンマー・くぎ・軍手やしっくいなどが必要になった。

・ Beinbaum's view : 対象世界の中で金物屋があるか否かを逐一調べる。

・ Our views : 「金物屋の近くにいるプランナー」に対応する表現に対応づけて memory 内に目標を位置づける。

3.4.5. The Assumptions

Case-Based Planning の考え方の一環としての opportunistic memory

past success → prototype for new plans.

past failunes → to avoid repeating the failunes.

past plan modification → tailor old plans to new situations.

↓

従来手法(primitive actions → large plans)との相違

CBP における仮定

- ・ 閉世界仮説の棄却 → preplanning・projection を制約
- ・ incomplete and imperfect model of the world.
- ・ プランをシミュレーションしてみてもプランを構成する各スラップ間での相互干渉にともなう問題を調べ上げることは出来ない。(不完全な知識に基づく限りこのようなシミュレーションが現実からかい離することは不可避)

実際、複雑状況下で人間がプランニングを行う際にも、自らの行為の結果を完全にモデル化することもできなければ、他者 agent の行為を完全に予測することもできない。

↓

- ・ mechanism that will allow it to react to the world as well as act in it.
- ・ mechanism for parsing or understanding the world that provides the planner with enough information to make execution-time decisions.

→ 実行しながら実行結果をモニターするメカニズム

↓

a planner must understand and interact with the world in which it executes plans

プランの実行を通じて対象世界との相互干渉をとりながら対象を理解するプランナー

3.4.6. TRUCKER and RUNNER

3.4.6.1 TRUCKER

pick-up and delivery task

プラン：注文処理のスケジューリングと配送トラックの routing

複数目標の最適化について

case が memory にある限り行わない。

実行中に最適化の opportunity を見出したときのみ行う。

プランの統合 (merge) と新たなプラン事例の格納 goals goals の充足条件を対象世界に認識する際に用いられる memory structure.

例 シアズタワーでの pick-up の目標

シアズタワーを認識するのに必要となるタワー 内部の表現

3.4.6.2 RUNNER

a single agent to control, but a wider range of activity

a richer set of goal interactions

↓

opportunism and execution-time failure recovery

wait for the world to inform it about positive plan

goal 充足の opportunity についてかなり複雑な解析を要する。

precondition のチェック

opportunity となるべき precondition の特定

opportunity となりうるか否かの評価

non-normative な feature 程 評価は高くなる。

例 John の朝食の例題

precondition : having money → normative

being at a grocery store → non-normative

3.4.7. Opportunism in TRUCKER and RUNNER

to halt planning on a goal.

to suspend it.

to recall it when opportunities for execution present themselves.

そのメカニズム

① the blocked goal とその充足の opportunities を表す memory structure との association

different types of planning problems

e.g. resource limitations

time constraints

planner's limitations

② plan execution + monitoring the effects of its plan

(world parsing)

↑

blocked goals の associate されている memory structure を用いて parsing
understanding of the world → immediately activates goals associated with that
situation.

③ activated goals の進行中のプランへの統合とそのプランの実行

例 John の朝食

1. 「オレンジジュースの入手」の goal block
planning problem : lack of time
2. being at a store → an opportunity to get the orange juice
goal: being near a store.
3. Seven-Eleven の認識 → goal activation.
4. precondition のチェックと実行

3.4.7.1 Suspending blocked goals

TRUCKER :

receiving a new request for a pick-up and delivery.

1. check all active requests that have positive interactions with the new request.
2. idle truck to take up the order.
3. search for its "desktop " for a suspended request that might be usefully combined with the
new order.
4. order の待行列へ < goal blocked/suspended >

RUNNER :

reason far more about the condition that might be constitutes opportunitives to satisfy a blocke d
goal.

there are more constraints on " opportunitives " than on simple preconditions

ease fo recognition.

likelihood of occurrence.

predictiveness.

taxonomy of opportunity types to derive the conditions that will serve as opportunities to satisf y
the plan

- non-normative resource.
- special tool
- special location
- special agent or skill
- time constraint

3.4.7.2 Recalling suspended goals

TRUCKER :

parse the objects at its current location and responds to
any changes that the tokens it has recognized suggests.

the 5700 block of Woodlawn → turn

also checks the token for any annotation of a goal

この作業は TRUCKER の認識する各 object の特定のタイプのリンクをたどることで事
足りる。

RUNNER : requires general-purpose recognition system

pick up orange juice at any grocery store

DMAP parser (Riesbeck & Martin, 1986)

to represent concept sequences to recognize concepts by recognition of their parts

〈関連文献〉

・ M. Marks, K. J. Hammond and T. Converse.

“ Planning in an Open World : A Pluralistic Approach. ”

Proc. of CBR DARPA Workshop pp. 271-285, 1988.

3.5 FIRST

(財)新世代コンピュータ技術開発機構 前田 茂

A Case-Based Mechanical Redesign System

by Francois Daube, Shlunberger Laboratory for Computer Science
and Barbara Hayes-Roth, Stanford University
from IJCAI-89, pp.1402-1407

概要

過去の設計プランを利用して梁の再設計を行う FIRST というシステムを示す。FIRST は現存する設計を物理法則を使って解析し、制約が満たされない時は、類似の問題状況を検索し、その状況に用いられたプランを得る。そして、その各プランを類推を使って新しい状況に合うようにし、変換されたプランと解析知識を一つの再設計プランにまとめる。FIRST は BB1 という黒板システムの上に実現されている。

3.5.1 Introduction

機械設計は first principle から始めることは少なく、過去の設計を用いることが多い。設計を変更する知識は伝統的にルールかプランの形で表される。

別の可能性は類似の事例から修正を導き出すことである。この再設計知識は普通プランによって表される。

ここでの仕事は事例ベースのアプローチの機械設計への可能性を試すことにある。梁の設計を選んだ理由は、理論が枯れていて、first principle からの推論に適しているため、他の研究と比べ易いこと。一つの物体か物理的な部分システムに注目すればいいこと。そして、この領域がまだ充分機械設計分野で重要なことなどが上げられる。

3.5.2 Overview of the System

FIRST は梁を記述する入力変数の集合と、設計制約の集合によって定義される問題を解く。

最初、元となる設計が与えられ、制約を満たすのが目標である。システムは評価関数を持たないので、設計は制約を満たすだけで良い。

FIRST はまず、制約を評価するための解析フェーズから始まる。これは基礎物理の一般的な知識からなり、修正の効果を評価する。

ある制約が満たされないと、過去のプランから再設計プランを作る。これはある類似性の基準を用いる。

そして、この再設計プランを新しい問題に適用できるように変換する。

変換された個々のプランをまとめて全体の再設計プランを作る。

ここで、現在の問題状況に無関係な行動は除かれる。これは、behavioral equations から作られる dependency graph と領域概念の分類法中にあるヒューリスティックスによる。

システムは全ての制約が満たされた時再設計プランの適用を終える。

図1参照。

3.5.3 Mesuring Similarity

システムは領域の概念を分類木の形で覚えておく。それは、梁や、荷重や材料、問題の型、制約の型などを記述する単語である。また、ある特定の属性も含む。ここでは、この概念木を事例ベース推論の能力の基礎であると考ええる。

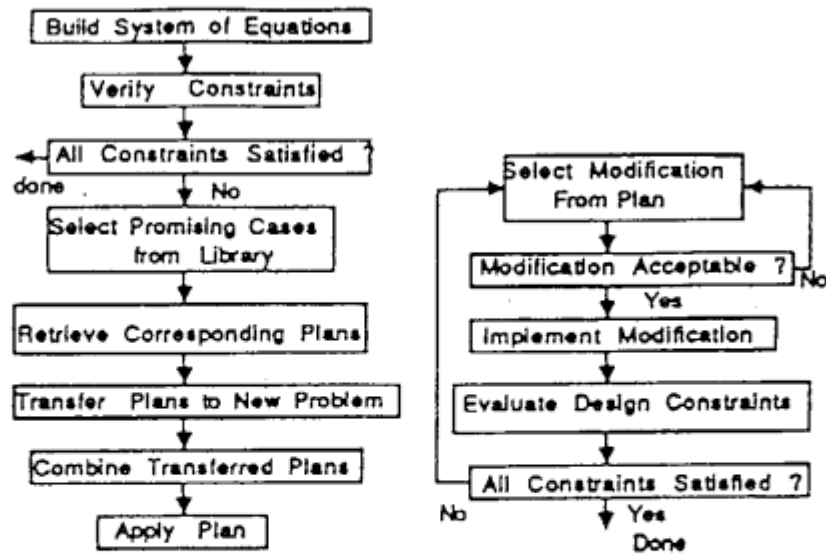


図 1: Flow of Control

Plan Application

ここでは、概念間の類似度を計る距離を *is-a* のリンクの数で定義する。この方式が適当でなかったり、充分でなかったりする時は、その属性間の距離で値を修正する。二つの概念のリスト間の距離は個々の要素間の最小の距離の和で定義する。(図 2)

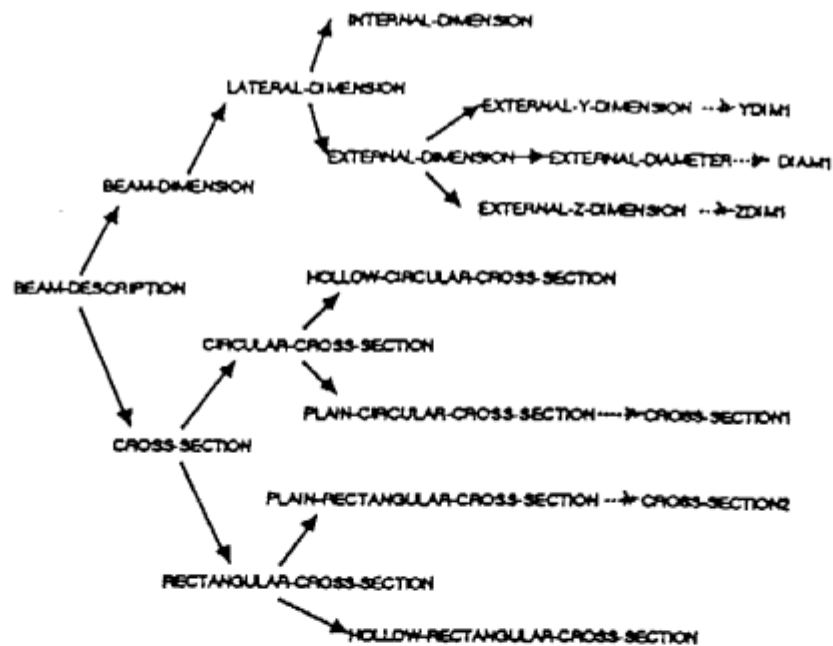


図 2: Simplified taxonomic hierarchy

3.5.4 Analysis

システムの初期目標は制約のチェックである。その正当性は構造の挙動を記述する等式によって計算された変数による。解析モジュールは物理法則に関する知識を用いてこの等式を導く。また、これは設計の修正効果を挙動等式に伝播し、特定の領域特有の制約を具体化する。

図3で示される例では、梁の記述が設計変数として与えられる。また、制約も与えられる。最終的には挙動モデルが等式として得られる。

解析が終わると、システムは制約が満たされるかどうかをチェックする。例では、*maximum equivalent stress* が *yield strength* を越えた事と、材質が鉄では *corrosive environment* に耐えられない事が分かった。それで、システムは構造を変更するために過去の問題を検索する事に決める。

3.5.5 Redesign Options

3.5.5.1 Symbolic Analysis 制約の破綻が単純な等式の時、*symbolic analysis* を用いて構造を変更する。非競合制約では、定性的修正を行う。競合制約の時は事例記憶を用いた修正に頼る。

3.5.5.2 Redesign Plans - Redesign Language ここでは、過去の解に注目するのではなく、過去のプランに注目する。

事例ベース推論を用いて再設計プランを作る時にはいくつかのステージがある。

access フェーズは現在の問題に対して有効な類似事例を認識する。

transfer フェーズは選択された事例を現在の問題状況に合うように類推を用いて変換する。

mapping ステージは事例と問題状況の類似性と差異を認識する。

プランを高次の宣言的でかつ単純な形式で表すために、*language framework* という設計行動の階層を作り、高次の行動から LISP への変換を作った。

3.5.6 Case Access

事例記憶は問題の物理的特徴と関連する制約の型と制約の状態（満足／破綻）とでインデックスされる。類似の事例を選ぶ基準は現在の問題の破綻した、または満足した制約と事例の破綻した、または満足した制約との間の類似度による。有効な事例のリストが現在の問題で破綻した制約と一緒に結果として得られる。

3.5.7 Case Transfer and Mapping

再設計プロセスの始まる前に変換が起きると、記憶中のプランによって参照された石器得変数の幾つかは、新しい問題中では等価にならない事がある。システムはこの問題を設計変数の変換をできるだけ遅らせる事で対処する。

3.5.7.1 Plan Transfer プランの変換はプランの記述の解析と、記憶中の問題の設計変数から新しい問題の類似変数への変換によって行われる。

このアプローチは問題独立な概念の修正を行わず、例えば、インスタンスをクラスに結び付ける *change-type instance-A to type-B* のような記述には不十分である。それで、システムはタイプの階層を通して概念 A から B へ導くような変換を見つけ、この変換を各リンクにそって属性の付加と削除のリストで特徴付ける。新しい問題の中の行動記述の類似はそれから、*instance-A* に最も近い *instance-A'* を *type-B'* に変換する。これは、両方の問題の各ステップで付加、削除された属性のリストの距離が最小のものである。

記述: *Change-type cross-section2 to hollow-rectangular-cross-section*
は: *change-type cross-section1 to hollow-circular-cross-section* に変換される。それは、*cross-section2* から *hollow-rectangular-cross-section* へ

の変換だと認識され、cross-section1 から始まって新しい問題のタイプ階層で最も近いものを見つけ出す。

3.5.7.2 Delayed Transfer and Action Elimination 事例記憶の評価は解決過程の最初に起こる。その時、過去のプランの概念のいくつか、たぶん解決過程で作られたものは新しい問題では等価ではなくなるかも知れない。そこで、変換機構では、概念の変換はそれを参照する行動が実行に移される時にだけ行われるようにする。この遅延評価戦略は変数が変換される時に類似候補の数を最大にする。

過去のプランの全てが新しいプランにも有効ではないので、システムは各行動の現在のゴールへの関連度を評価する。事例は制約と共に記憶されているので、行動記述中で参照される設計変数と破綻した制約の間の影響を評価し、破綻した制約に含まれる変数を見つけだし、解析フェーズで作られる挙動等式に基づいて *dependency graph* を作る。行動中の変数が *dependency graph* に現れる時は、その行動は事例に関係あると見なす。つまり、*dependency graph* はある変数が破綻した制約に影響を持つかどうかを調べるために用いられる。変数と破綻した制約に関係がないときは、その行動は除かれる。

このアプローチは等式に現れない変数、例えば、cross-section K には用いる事ができない。底で、システムは領域概念間のリンクを用いて影響を評価する。cross-section は 'affects' リンクで cross-section-properties にリンクされている。これで、cross-section の変更は cross-sectional properties に影響を及ぼす事が分かる。行動間の関連度を評価するのは *mapping* ステージである。ここでは事例と新しい問題の間の類似度の総量を求める。

3.5.8 Combining Plans: The Blackboard Model

FIRST は BB1 という黑板システムの上に実現されている。システムではプランは知識源の集合として表され、各知識源は特定の行動を実現する。

黑板モデルは *cooperative problem solving* を提供し、symbolic analysis と事例から得られた知識を統合するのに用いられる。システムは大域的な制御戦略にしたがってどちらかを選択する。戦略は、選択された事例の制御戦略によって提供された重要度の和によって実現される。

3.5.9 Plan Application Example

最初の行動は round rod(rr1) と rectangular beam(rb1) プランからの計画を現在の問題状況に変換する事と、symbolic analysis からの行動である。round rod からの行動は *dependency graph* により無関係であり、除かれた。たとえば、制御変数 *diam1-problem2* と破綻した environment の間の関係が見つからなかったとする。

symbolic analysis で提案される行動が単純なルーチン修正だったので、高い優先度を持つ。したがって、これが選ばれ、梁の最大重量に達するまで梁の直径を増やす。次に、rr2 もルーチン修正だったので選ばれ、材質を steel から *nickel_alloy* に変更する。ここで、maximum stress と maximum weight の競合する制約が残り、ルーチン修正はなくなった。そこで、システムは rb1 からの非ルーチン修正を選び、circular beam を hollow circular に変換し、hollow beam の製造制約である外径の制限を付け加える。この構造の修正により、解析モジュールは構造の挙動等式を更新する。次の行動は rb2 で、内径と外径を増やす。このステップは製造制約が破られるまで行われる。次に rb3 が選ばれ、circular beam を I-beam に変換する。ここで挙動モデルも修正される。ここで、実行すべき行動がなくなり、解決過程は終了する。(図3)

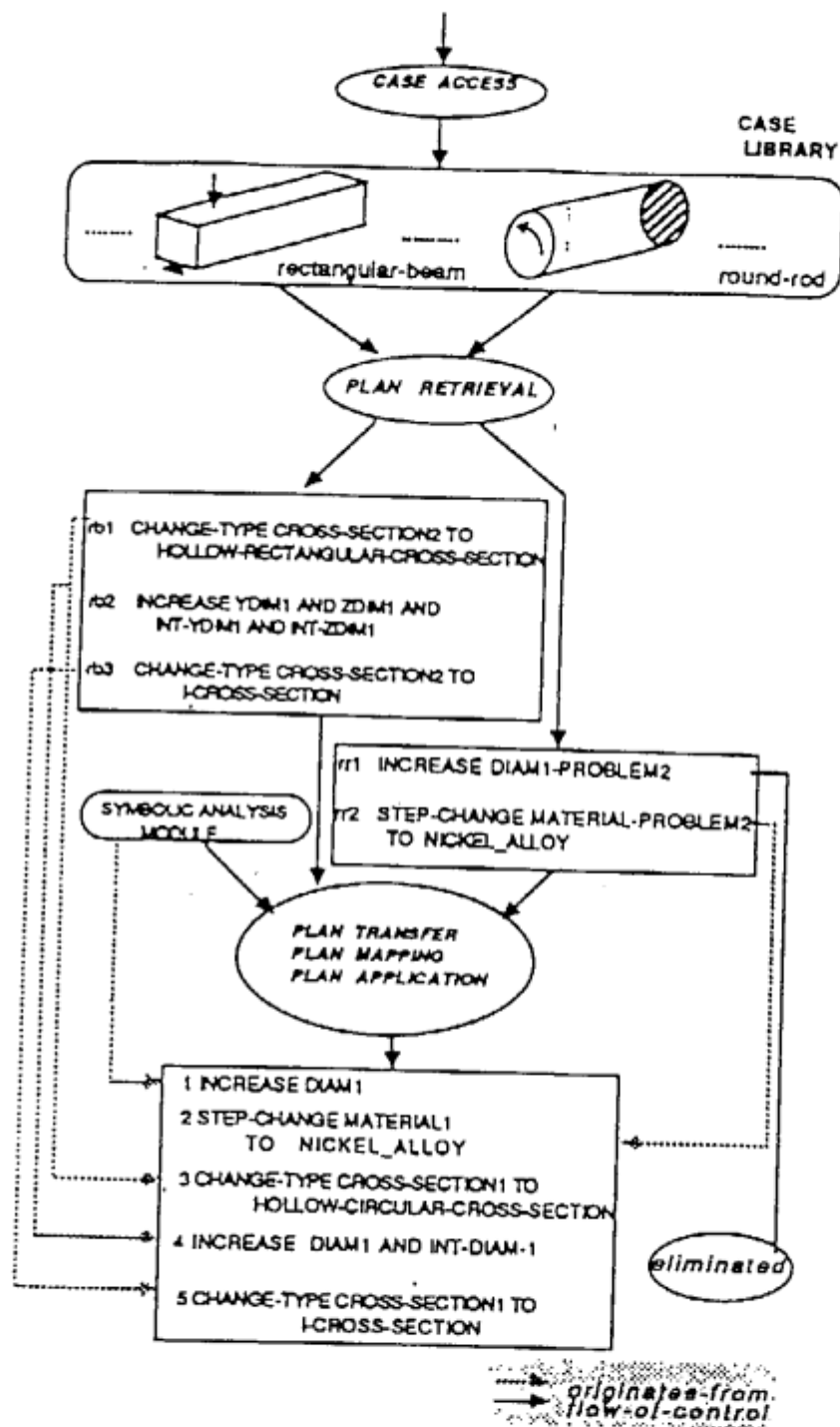


图 3: Redesign Summary

3.5.10 Limitations

ここでは transfer と mapping に重点を置いた。事例の検索も重要であり、multiple indexing は大きな事例記憶を扱う時に有効であるが、ここでは簡単なものとした。

一つの問題の概念から他の問題への変換はタイプ階層の距離に基づいたものに制限されている。これは、問題に対して詳細なタイプ階層が用意できる時は有効であるが、そうでない時はいくつかの解を導き出す。その時は概念の属性の情報による距離を用いれば良い。

挙動方程式からの修正とその適用のメカニズムは他の方式と比べて単純である。それらは、定量的評価を行うが、ここでは定性的解析の繰り返しによって行う。

正確には行動の列として表されるプランはその行動の論理的根拠を示す事ができないので、dependency graph のような weak method に頼る事になる。

3.5.11 Conclusion

さらに他の領域の設計への適用を試す事によって評価する必要がある。このアプローチは非ルーチン修正を行える。過去のプランから新しいプランを作るので、革新的な設計は行えないが、first principle とヒューリスティックスを用いたものより効率的である。タイプ階層と言語を与えると一般的な設計に用いる事ができる。ここでは、有効な事例の検索と、プランの変換、関連するプランの一部の大域的な再設計プランへのマッピングの一般的なメカニズムを示した。

3. 6 自動車故障診断 (Redmond)

N T T情報通信処理研究所 打橋 知孝

要 旨

事例ベース推論 (Case-Based Reasoning, 以下 CBR) では一般的に専門家自身が行った問題解決事例やシステムのプログラマが投入した事例を利用して推論を行っている。本論文で筆者 Redmond は、自動車の故障診断分野において観測された故障箇所同定に至るまでのアクション例から事例を獲得するシステムを報告している。このシステムではとられた一連のアクションを事例とするのではなく、各アクションの目的を推論してその目的を INDEX とした事例として獲得している。

3.6.1 論文名・筆者

LEARNING FROM OTHERS' EXPERIENCE:

CREATING CASES FROM EXAMPLES (DARPA'89 pp.309-312)

Michael Redmond (Georgia Institute of Technology)

3.6.2 研究の背景

一般的に事例を用いて推論を行う際には、過去に自分自身が解決し経験した事例を利用する。そのため、その事例中で行われるアクションが何を目的として行われるのかを理解しており、特に明示する必要はなかった。しかし CBR システムへの投入等他人が使う際にはその目的の説明を行う必要があり、後に利用することを考慮して事例を作成する必要がある。

3.6.3 内容

3.6.3.1 INTRODUCTION

人間の教育の現場において事例は重要である。生徒は事例によってドメインの原理の適用法を学習する。この事例は生徒自身が経験する必要は無いが、なぜ行われるかが説明されなくてはならない。

そのため事例を獲得する際には、着目するアクションが何を目的に行われているかを計必要がある (その目的を INDEX とすることができる)。問題解決事例はそれらの事例間にリンクをはることで表され保存される。

問題解決事例から事例を獲得するためには次の処理が必要となる。

1. 一連のアクション内に隠された目的の推論を行い、複数の事例に分ける。
2. 各事例間の関係付けを行う。
3. 各アクションの主たる問題解決への影響をチェックし、1. で設定した目的の妥当性を確かめる。

3.6.3.2 INFERRING INSTRUCTOR'S GOAL

自身の経験していないアクションの目的は明示されないので、推論する必要がある。

診断分野において考えられるアクションの目的は、“仮説生成、仮説検証、検証に必要な処理、誤りの訂正、症状の評価”である。

図3.6-1 は専門家の行ったアクション（部分）を示す。症状は＜エンジン停止＞である。まず、専門家はその原因として、“アイドリングスピードが低い(low fast-idle-speed)”という仮説をたてテストする。このテストの際に

＜手を使って＞＜エアクリーナを外し＞,
＜レンチを使って＞＜ラジエタファンを外し＞,
＜タコメータとエンジンをつなぎ＞,
.....

＜仕様書のタコメータの値と比較する＞
というアクションが行われる。

これら一連のアクションは(low fast-idle-speed)という仮説の検証を目的としたアクションの事例として仮説と共に蓄えられる。

```
(test (low fast-idle-speed))
(use (hands))
(do (open hood))
(do (remove air-cleaner-casing-top))
(do (remove air-filter))
(do (remove air-cleaner-casing-bottom))
(do (set (position gear-shift neutral)))
(use (socket-wrench))
(do (disconnect radiator-fan))
(do (connect tachometer engine))
(do (start engine-system))
(do (run engine-system) until (temperature engine-system warm))
(do (disconnect vacuum-advance-hose distributor))
(do (plug vacuum-advance-hose))
(do (small (open throttle)))
(use (c-4812-2c))
(do (connect c-4812-2c choke-cam-follower-pin))
(do (release throttle-lever))
(ask (desired-fast-idle-rpm nil) hood-sticker (reply 2400))
(ask ((rpm engine-system) nil) tachometer (reply 1600))
```

図3.6-1 Instructor's Actions to be explained

3.6.3.3 INFERRING PLACE IN CURRENT DIAGNOSIS

問題解決事例として観測できるアクションは時間を追った一本の線で表される。しかし、一般的に診断においては、“ある仮説が他の仮説を導く”ことや“ある仮説が否定された際に他の候補が存在する”ことから図3.6-2に示される木構造で表現される階層的な診断が行われている。

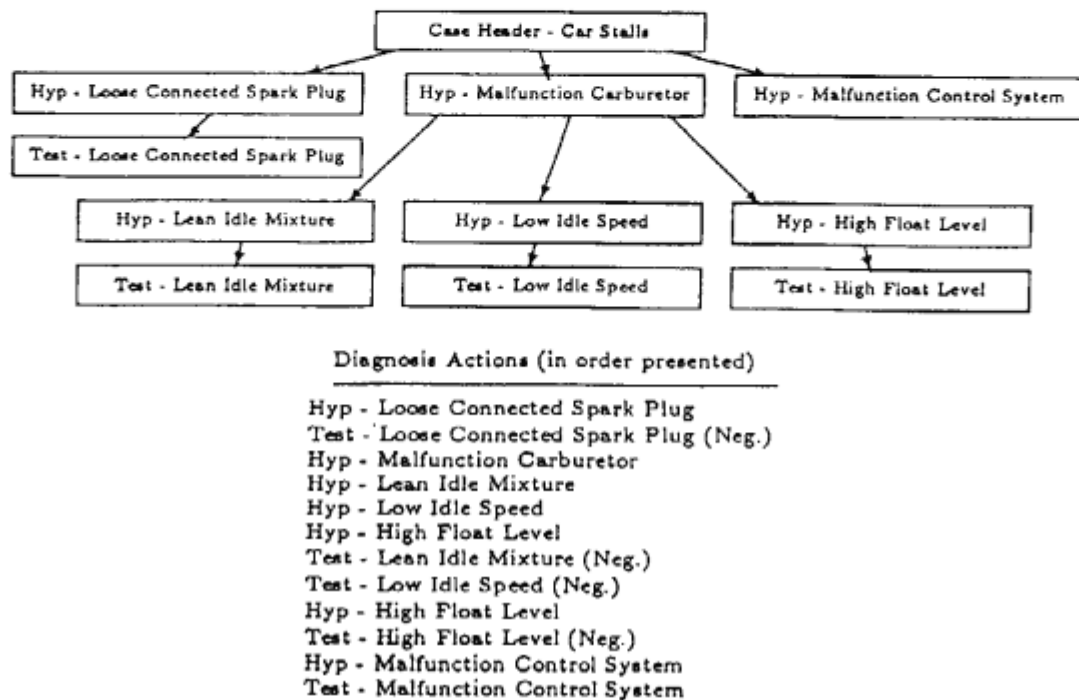


図3.6-2 Inferred Diagnosis Structure

3.6.3.4 UPDATING THE CURRENT CONTEXT

問題解決は初期記述、その変更、問題解決に関係するアクションを含んでいる。変更時の影響等も考慮し、最適な事例の組合せをつくる。

3.6.3.5 CONCLUSION

一般の教育で行われているものを参考に診断分野での問題解決事例からの事例の獲得するために必要な処理を示した。今後、設計などの分野への適用を予定している。

3.6.4 研究の到達点および今後の課題

筆者らは自動車の故障診断における問題解決時の専門家のアクションを観測するシステムを提案している。このシステムは後にCBRにおいて利用される事例を、観測された一連のアクションをそのまま事例にするのではなく、各アクションの組合せの形で作成する。システムは各アクションがなぜ行われたのかを説明することで事象を細分化する。これにより柔軟性が高くなったといえる。

3.7 JUDGE

日本電気(株) C & Cシステム研究所 古関 義幸

要旨

本節では、事例ベース推論技術を応用して、過去の判決を基にして新たな犯罪に対して判決を下すシステムの研究 [Bain 86, Riesbeck 89] を紹介する。

3.7.1 はじめに

JUDGE は、過去の判決を利用して、犯罪の凶悪性、行為者の正当性を比較・考慮して、直面した事件に対する判決を下すシステムである。このシステムは下記の5つのフェーズに分けて処理を行うことにより、判決を下す。

1. 解釈 (Interpretation) フェーズ:
行為者の動機 (motive) を推論。各人の暴力行為の正当性 (justification) を決定。
2. 事例検索 (Retrieval) フェーズ:
過去の事例を検索。
3. 差の分析 (Differential Analysis) フェーズ:
検索された事例と入力事例を比較。
4. 修正 (Application and Modification) フェーズ:
判決を下す。(過去の事例と入力事例における犯罪の凶悪性 (heinousness) の差異が、判決の重さに反映するようにする。)
5. 一般化 (Generalization) フェーズ:
判決ルールを作る。

3.7.2 解釈フェーズ

解釈フェーズでは、犯罪の当事者の動機についての解釈を行う。

入力情報 解釈のために使用する情報は以下の3種類。

- 行動 (例: A が B を叩いた)
- 結果 (例: B は殴り倒された)
- 間接的な結果 (例: B は頭を叩かれて、意識を失った)

解釈の種類

- 初期の動機の解釈
なぜケンカを始めたのか。
- 意志の解釈
(escalation) 相手に反撃した時に、相手の攻撃の程度を越えた反撃をしたかどうか。
(motive) 行為の理由
- 結果の解釈
行為の結果が意図したもの (intended) か、事故的なもの (accidental) か

意志の解釈のルール

- Unprovoked-Violation:
他者の生存目標 (Presevation goal: 以下 P-goal) を侵害する者は unjustified。
ここで攻撃者 (aggressor) を A、防御者 (defender) を D とする。
- Parity Retaliation 1:
何の脅威もないのに、current violent level(以下 c.v.l.) で D が A の P-goal を侵害したとき、D は unjustified。
- Parity Retaliation 2:
c.v.l. で A が D の P-goal を侵害したとき、A は unjustified。
- Self Defense 1:
D が脅威に直面して、A にされたのとおなじ v.l. で A を侵害したとき、D は justified。
- Self Defense 2:
A が D にされたのと同じレベルで復讐したとき、A は unjustified。
- Escalation Rule 1:
c.v.l. を越えるレベルで侵害したとき、A でも D でも unjustified。
- Escalation Rule 2:
D は、相手の侵害を防ぐ他の選択がない場合に限り、c.v.l. を越えたレベルで侵害しても justified。(但し、c.v.l. を越えた最低のレベルに限る。)
(注)v.l. とは各暴力行為の凶暴性を数値化したものである。例えば strike は 2、slash は 4、stab は 5 である。c.v.l. を越える行為で反撃すると escalate と解釈される。

例題 1つの事例はイベント列で表現される。イベントとは行為とその結果のペアで記述される。システムはイベントの起こった順に解釈を行い、motiveを見つけ、escalationの値を決める。CRIME1の例：

First, Randy struck Chuck with his fists several times. Chuck was only slightly hurt. Next, Chuck struck Randy with his fists several times. Randy was only slightly hurt. Then, Randy slashed at Chuck with a knife one time. Chuck's skin was cut. Next, Chuck slashed at Randy with a knife one time. Randy's skin was cut. Finally, Randy stabbed Chuck with a knife several times. Chuck died.

この例 (CRIME1) は次のように整理出来る。

[CRIME 1]

イベント	解釈	理由
Randy Chuck		
strike → hurt	unprovoked-violation	(最初に暴力をふるった)
hurt ← strike	retaliation	(脅威が無いのに c.v.l. で対抗した)
slash → cut	escalation	(c.v.l. を越えて侵害した)
cut ← slash	self defence	(脅威に直面して c.v.l. で対抗した)
stab → dead	escalation	(c.v.l. を越えて侵害した)

ここで、各行が1つのイベントに相当する。例えば、最初のイベントは unprovoked-violation であると解釈される。理由は Randy が最初の行為者だから。

3.7.3 事例検索フェーズ

顕著な特徴に焦点を当てて検索する。

- 違反している法令は？
- 誰がケンカを始めたか？ (被告か被害者か)

システムは最後のイベントの結果、その結果を引き起こした行為、その行為の動機 (motive)、最後から2番目のイベントの結果、... と逆順に解釈結果を比較して、最も類似した過去の事例を検索する。

CRIME 0(既に事例ベース中にあったもの)：

First, Ted slashed at Al with a knife one time. Al's skin was slightly cut. Next, Al slashed at Ted with a knife one time. Ted's skin was slightly cut. Finally, Ted stabbed Al with a knife several times. Al died.

```

[CRIME 0]
Ted    Al
slash  → cut
cut    ← slash
stab   → dead

```

システムは CRIME 1 と CRIME 0 を比較する。

最終のイベントに対する比較結果

```

final result: kill (同じ)
final action: stab knife (同じ)
final intention: escalate retaliation (同じ)

```

以降、一つずつ前のイベントを比較していく。複数の事例がマッチした場合は、より最近の事例か、または特徴に重み付けをしてトータルの得点の高いものを選んで1つに絞る。

3.7.4 差の分析フェーズ

システムは検索事例と入力事例を突き合わせて、Causal Features の中で顕著な差異が見つかるか、一方の事例が吟味され尽くすまで、最後のイベントから逆順に各イベントを比較していく。(in-depth comparison)

最後から3番目のイベントで、次の差異を発見。

Causal Features	CRIME 1	CRIME 0
amount of harm		同じ
extreme force used		同じ
motive	response to be hit	unprovoked-violation
justification		同じ
escalated level	+2 (slash - strike)	+4 (slash - zero)

検索事例に判決ルールが付いている場合、判決ルールで考慮している特徴を比較する。その結果、差異が見つからなければ、検索事例の判決をそのまま適用する。差異が見つかった場合は、修正フェーズへ。

3.7.5 修正フェーズ

事例の比較で発見された特徴の差異に基づき、過去の判決の修正を行う。

判決の修正

- conflict:
一方の事例が、動機は justified なのに、結果が悪い場合。
- agreement:
一方の事例が、動機も結果もともに悪い場合。
- similar-result:
一方の事例の動機がより悪いこと以外は差異がない場合。
- extreme-force:
どちらの動機も unjustified だが、一方の事例に於て extreme force が使用された。
- closeness:
最後から 2 つのイベントのまでの間に差異がみつかったかどうか。見つければ immediately、見つからなければ not immediately である。

判決修正ルール

	not immediately	immediately

conflict	± 0%	± 25%
agreement	± 25%	± 50%
similar-result	± 25%	± 50%
extreme-force	± 50%	± 75%

3.7.6 一般化フェーズ

次の 2 つの条件が整ったとき一般化した判決ルールを作成する。

1. 入力事例と検索事例が十分な数の特徴を共有している。
2. 2 つの事例が causal feature に関して類似している。この類似性の度合は、in-depth comparison を行った深さに依存する。（経験的に最後の 2 つのイベントの中の特徴が一般化ルールを作るのに役立つ。）作られたルールは入力事例にリンクされて蓄えられる。

CRIME 1 は CRIME 0 と大変よく類似しているので、CRIME 1 の被告に対し CRIME 0 と同じ判決（懲役 40 年）を言い渡す。

FORMING GENERAL SENTENCING RULE:

--- FOR VIOLATION OF MURDER,
--- FOR CAUSING RESULT OF KILL,
--- FOR USING ACTION OF STAB-KNIFE,
--- FOR HARMING INITIAL-DEFENDER,
--- FOR RESPONDING TO A SLASH-KNIFE LEVEL OF HARM,
--- FOR USING ESCALATED FORCE AND ACTING OUT OF RETALIATION,
--- FOR INTENDING TO CAUSE THE RESULT.

THE SENTENCE FOR THIS VIOLATION WILL BE THE SAME AS GIVEN
FOR CRIME1 ABOVE.

3.7.7 事例の修正

判決ルールの適用と修正 CRIME2 の例 :

First, David slapped at Tim with his hands several times. Tim was pushed. Next, Tim struck David with his fists several times. Daavid was only slightly hurt. Then, David hit Tim very hard one time. Tim fell down. Finally, Tim stabbed David with a knife several times. David died.

[CRIME2]

Tim David
pushed ← slap
strike → hurt
down ← hit
stab → dead

類似の事例として CRIME 1(判決は懲役 40 年) が検索された。CRIME 1 には判決ルールがついているのでこれを使おうとした。ところが 2 カ所でミスマッチが見つかった。

CRIME 2	CRIME 1
Harming INITIAL-AGGRESSOR	Harming INITIAL-DEFENDER
Respond to KNOCK-DOWN	Respond to SLASH-KNIFE

このミスマッチを埋めるために、修正ルールを適用する。いま、修正ルールが、誰がケンカを始めたかよりも暴行のエスカレートの度合いを重く見るような場合、一つ目のミスマッチに対して CRIME 2 の判決をほんの少し軽くし、二つ目のミスマッチに

対して CRIME 2 の判決をやや重くする。この結果 CRIME 2 のほうがより凶悪であることになり、45 年の判決を言い渡す。

一貫性の保存 CRIME 2(45 年) は CRIME 1(40 年) より悪い。CRIME 0(40 年) も CRIME 1(40 年) より悪い。CRIME 2 は CRIME 0 と比べてどうか。in-depth comparison をおこなってやはり CRIME 2 が CRIME 0 より悪いことを確認する。この結果、CRIME 2 と CRIME 1 の差を新しい判決ルールとして CRIME 2 につけて保存する。

差異が大きすぎる場合 検索事例の持つルールに記述された特徴の値と、入力事例の特徴の値の差異が大きすぎて修正ができない場合 (Modification がない場合) は、ルールを完全に忘れて、検索事例と入力事例の in-depth comparison を行って、その結果から判決を下す。

参考文献

- [Bain 86] W. M. Bain, *Case-Based Reasoning: A Computer Model of Subjective Assessment*, PhD thesis, Yale University, 1986.
- [Riesbeck 89] C. K. Riesbeck and R. C. Schank, *Inside Case-based Reasoning*, pp. 93-139, Lawrence Erlbaum Associates, New Jersey, 1976.

3.8 DMAP

日本電気(株) 島津秀雄

要旨

DMAP (Direct Memory Accessing Parser) は、事例ベース推論の考えを最初に提案したエール大学の Schank 教授の研究室から生まれた、事例ベース推論を内含する、自然言語理解および記憶システムである。DMAP は、MOPs (Memory Organization Packets) 上に構築されている。MOPs は記憶の表現モデルでありそれ自身記憶の一般化能力を持つ。DMAP は、MOPs で表現された記憶への検索・格納を行なうための処理機構としてマークパッシングによる推論機構を提案している。また、自然言語理解部分は、pattern-concept 対による Phrase-based Parser の機構に基づいている。DMAP では、自然言語理解処理と記憶処理が不可分に結合されている為、自然言語理解過程においては、記憶システムからの予測情報を有効に使うことが出来るし、自然言語理解の結果抽出された情報は、自然に記憶システムに組み込まれるという利点がある。また、予測が失敗した時にはその失敗を解析し、2度と同じ失敗を繰り返さないように記憶システムの修正を行なう。これらの特徴を有しているため、DMAP の設計者たちは、DMAP を Case-based Parser の一種であると称している。

3.8.1 はじめに

Direct Memory Accessing Parser (略して DMAP)[Riesbeck 86][Martin 89a] [Martin 89b] は、エール大学の C. Riesbeck と C.E. Martin が中心になって作成した自然言語理解のためのモデルである。DMAP 自身は内部に自然言語を解釈するためのパーザを持っているが、これまでの自然言語理解システムとは異なり、自然言語理解の処理が記憶システムと不可分につながっていることが大きな特徴である。自然言語処理をした結果新しい事実を抽出すると、それを記憶システムに組み込むことをする。記憶システム側で有している知識は、自然言語理解過程での予測処理に使われる。また、予測が失敗した時にはその失敗を解析し、2度と同じ失敗を繰り返さないように記憶システムの修正を行なう。この節では、まず、DMAP のモデルを記述するのに使われている知識表現言語の一種である MOPs について述べ、次に DMAP 自身についてそのモデルと処理について述べ、最後に DMAP モデルの基本機能を実現した Micro-DMAP につい

て説明する。

3.8.2 MOPs

MOPs(Memory Organization Packets) [Schank 82] は、知識表現言語の一種であると言ったが、正確には MOPs 自身が人間の記憶システムをシミュレートするモデルである。MOPs は、ユール大学の Schank とその生徒たちによって提案された。Schank は、それ以前に、人間の言語理解過程に生ずる推論機構を実現するモデルとしてスクリプトモデル [Schank 77] を提案している。スクリプトとは、日常生活における典型的な事象列を 1 つの枠の中にまとめたものである。スクリプトの理論によれば、人間は、非常に多くのスクリプトを持っており、ある状況に直面すると、その場に適したスクリプトを取り上げ、そこに書かれている事象に従って、物事が進行するであろうと予測をしながら状況をすすめていく。スクリプトを使うと、例えば、入力中で陽に指定されなかったり明言されなかった事象でも、スクリプトに記述されていれば、そのことは発生したのだろうという常識を表現することができるし、単語のあいまい性の解決にも使うことが出来る。例えば、order という単語は、レストランに関する文脈の中では「メニューを指定する」という意味になるし、軍隊の文脈では「上官が部下に命令する」という意味になる。従って、それぞれの order の意味を「レストラン・スクリプト」と「軍隊・スクリプト」の中に格納しておけばこのような単語のあいまい性は解決できる。

その後、スクリプトには 幾つもの問題点が指摘された。それらのうちの幾つかは、どの時点でどのスクリプトが選択されるか、あるいは使われなくなるか、という問題や、進行していくの話題がスクリプトから外れたらどうするのか、という問題や、記憶の階層についてサポートしていないのは不自然である、という等である。

MOPs は簡単化して言うと、構造的にはスクリプトの階層構造を可能にしたものである。抽象的なレベルで記述されている MOP が上位にあり、詳細な記述がされている MOP がその下位に存在する。階層化が可能になったせいで、新しい MOP の組み込みやある状況の予測処理が自然に実現されるようになった。すなわち、新しい事象列が提示されると、たとえそれと全く同一の事象でなくても似たものがあつた場合には、より抽象的な記述の MOP の配下に納めることが出来る。また、似ている MOP がある程度集まると自動的に帰納推論をさせて、それらの共通する部分を抽出した抽象的な MOP を自動的に構築する機構も盛り込める。

MOPs は、記憶のモデルであり、Schank の多くの生徒たちが MOPs モデルを基にその拡張を試みたり、効率的な実現方法を提案したりしている。Dyer [Dyer 83] は、BORIS システムの中の 22 種類の知識構造の 1 つとして MOP を採用しているし、Lebowitz は、IPP [Lebowitz 80] システムの中で、MOPs に帰納推論機構を付加して、最初の学習をする自然言語理解プログラムを作成した。また、Kolodner は Cyrus

[Kolodner 84] の中で、并別ネットワークを使って MOPs の記憶の効率的な検索・格納を実現した。DMAP は、MOPs に基づいた、また別の実行機構を持つ自然言語理解モデルである。

3.8.3 DMAP

DMAP の自然言語理解部は、言語パターンとそれに対応する意味表現の対を単位とした事例ベースからなっている。幾つかの例をあげる。

1. (actor), says, (content) $\Rightarrow$ M-MTRANS
2. Milton, Friedman $\Rightarrow$ M-ACTOR
3. an argument $\Rightarrow$ M-MOBJECT
4. (文1) ならば (文2) $\Rightarrow$ M-CAUSAL

左辺は、自然言語のパターンであり、右辺はそれに MOP で表した対応する意味表現である。例えば 2 番目の例は、“Milton” という単語が出現しそれに “Friedman” という単語が続いた時には、MOP で表した ACTOR(動作者) 構造を構築することができる、と読むことが出来る。DMAP による自然言語文解析の大きい流れを言うと、入力文と事例ベースを比較し、入力文に完全にあるいは部分的にマッチするものを見つけ出す。左辺のパターンがすべてマッチしたパターンが見つかったら、右辺の概念を構築する。左辺の定義では、既に出来上がった概念と別の概念を組み合わせて更に大きな概念を構築する、と言うことを定義しているものもある。その例が 4 番目の定義である。ここでは、2 つの文が「ならば」という言葉をはさんで存在した時には、両方の文の意味構造間には因果関係が存在するとして、因果関係を表現する MOP 構造を構築する。

このような入力文と言語パターンを比較していき、適当なマッチングが発生すると、対応する概念を生成していく処理は、マーカバッシング機構を使って実現されている。マーカバッシングの定義は、一般にはノードとリンクで構成されるモデルにおいて特定のノードが励起されると、そのノードからマーカをそのノードに接続する周囲のノードに伝達していき、あるノードで別のソースから来た 2 つのマーカが出会うと、前もって用意しておいたヒューリスティックルールに従ってその解釈を行なう。DMAP では、2 種類のマーカを用意している。1 つは、activation marker で、これは入力テキストと現在参照されている記憶構造に関する情報を取り出すのに使われる。あるノードが励起されると、その上位の概念に向かって activation marker が渡されていく。このマーカを受けとった記憶構造は、励起される概念候補とみなされる。Prediction marker は、どの記憶構造が取り出されるかの予測をする。あるノードが activation marker を受けとると、そのノードに続く可能性のあるノードにたいして prediction marker が送られる。幾つかのノードを経て Prediction marker が最終的にある概念に達したら、その概

念を呼び起こすことができる全ての言語パターン構造の先頭要素に prediction marker が渡される。Prediction marker が到達しているノードに activation pattern が到達したと言うことは、出現が予測された概念が入力から支持されたということなので、その概念が励起される。

予測が失敗した時の失敗構造を構築する機構について図 3.8-1 を使って詳しく説明する。図 3.8-1 は、ある入力に対して DMAP がした予測とそれに続く入力を解析した結果システムの予測が失敗したことがわかって、それで生成された記憶構造が両方書かれている MOP による記憶システムの一部を表した図である。入力文は、

Donald Regan: On a long term basis rates are headed down.

である。上の入力が順々に解釈され headed まで来たところで DMAP は、その次にくる概念として、これまでの Donald Regan の発言内容から予想して、利息率が上がると予測して IR-Increase という構造を作った。ところが、実際の入力では、up ではなく down が明記された。それで、DMAP は IR-Decrease という構造を作った。次に DMAP は、それまでの状況と今回の発言の状況の差を調べ、今回の発言では “on a long term basis”(これは、DMAP 内部では「True-trend (本質的な傾向)」を表すフレーズであると解釈している)、と明記されていることを見出し、それがいつもの発言との差の原因になっていると判断した。その結果、Repair.7 という DMAP が予測したもの (IR-Increase) と予測が失敗して実際に生成されたもの (IR-Decrease)、その差の原因となったこと (True-trend) の 3 つ組を保持した構造を作って記憶システムに組み込む。この構造があるために、次に DMAP が IR-Increase を予測した時に、それと同時に True-trend が表明されていればこの Repair.7 が励起され、2 度と同じ予測の失敗を繰り返さないようにする。

まとめると、DMAP が事例ベースシステムの 1 つであるという理由は、次の 3 つが考えられる。1 つは、DMAP が使う言語知識が、実際の事例である例文とその意味を表す概念の対の構造であるということである。2 つめは、DMAP は入力を受けると、次に何が記述されるか予測をたてていくが、予測が失敗した時にはそれを失敗として認識し、その失敗の構造を記憶システム中に格納し、2 度と同じ間違いをしないようにする。これは新しい問題が与えられると、それに近い事例を取り出し、それが間違えているとそれに修正を加える、という一般的な事例ベース推論の処理の機構の中の大きな流れと一致する。3 つめは、DMAP が、意味構造を新しく生成するタイプのパーザではなく、既存の意味構造を修正して与えられた入力の意味構造を生成するタイプ、すなわち Recognize and Modify 型、のパーザであるということである。このことも事例

ベースの定義を満たしている。

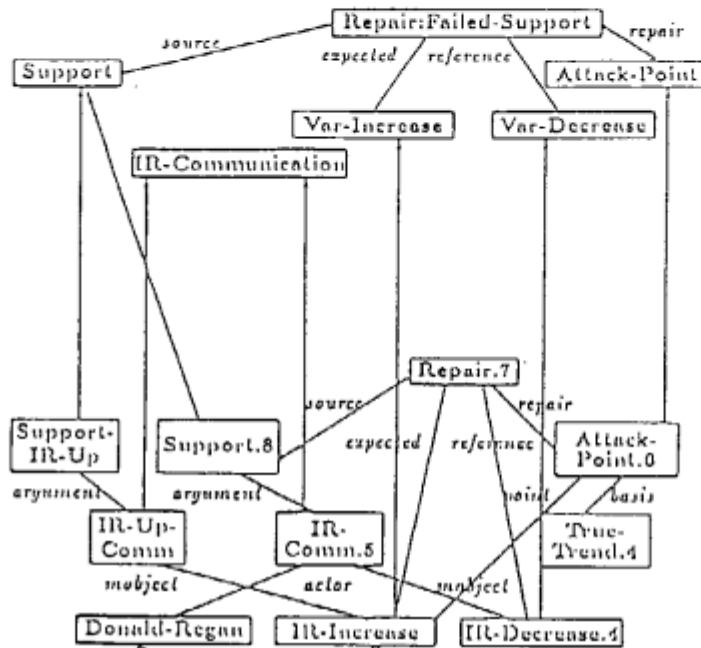


Figure 10.3: Case-based Parsing Output

図 3.8-1 DMAP に予測失敗の修正構造が作られた図

3.8.4 Micro-DMAP

Micro-DMAP は、[Martin 89a] 中で DMAP の提唱者の一人である Martin が作成した DMAP の小型版である。Micro-DMAP は、次の 2 つの文だけを解釈できるだけの事例ベースしか持っていないと言うことと、制御機構としてマーカバッシング機構を使わない、と言うことが、DMAP より制限されている主な点である。

1. Milton Friedman says interest rates will rise because of the monetary explosion.
2. Arthur Laffer says interest rates will drop.

Micro-DMAP のプログラムは、MOPs の部分を加えても 2000 行にならない小さいプログラムであるが、DMAP における本質的な部分は備えている。Micro-DMAP の事例ベースの量的な制限は、事例ベースを加えることで解決される。また、マーカバッシ

ング機構を使わないでより簡単な機構を採用していることも、DMAPの本質的な機能を損なっていることにはならない。Activation marker はないので、入力で明記された概念だけが励起される。また、prediction marker はないが、それに対応する機能を持つ「予測構造」というものが作られる。

3.8.5 おわりに

本稿では、DMAP(Direct Memory Accessing Parser) について簡単な説明を行なった。DMAP では、自然言語理解部と記憶システム部が融合しているため、自然言語理解過程においては記憶システムからの予測情報を有効に使うことが出来るし、自然言語理解の結果抽出された情報は自然に記憶システムに組み込まれるという利点がある。DMAP が事例ベースのパーザであると呼ばれるゆえんは、DMAP パーザが使う知識が事例(例文)ベースな構造であること、予測が失敗した時にはそれを修復して2度と同じ間違いをしないようにする機構を備えていること、等の特徴を持つためである。

参考文献

- [Dyer 83] M. G. Dyer, *In-depth Understanding*, The MIT Press, Cambridge, MA, 1983.
- [Kolodner 84] Kolodner, J.L., *Retrieval and Organizational Strategies in Conceptual Memory*, Lawrence Erlbaum Associates, Hillsdale, N.J., 1984.
- [Lebowitz 80] Lebowitz, M., *Generalization and Memory in an Integrated Understanding System*, Research Report 186, Yale University, 1980.
- [Martin 89a] Martin, C.E., "Case-based Parsing" and "Micro DMAP" in *Inside Case-based Reasoning*, Lawrence Erlbaum Associates, Hillsdale, N.J., 1989.
- [Martin 89b] Martin, C.E., "Pragmatic Interpretation and Ambiguity", the 11th Annual Conference of the Cognitive Science Society, 1989.
- [Riesbeck 86] Riesbeck, C.K. and Martin, C.E., "Direct Memory Access Parsing", YALEU/DCS/RR 354, Yale University, 1985.
- [Riesbeck 89] Riesbeck, C.K. and Schank, R.C., *Inside Case-based Reasoning*, Lawrence Erlbaum Associates, Hillsdale, N.J., 1989.
- [Schank 77] Schank, R.C. and Abelson R., *Scripts, Plans, Goals, and Understanding*, Lawrence Erlbaum Associates, Hillsdale, N.J., 1977.
- [Schank 82] Schank, R.C., *Dynamic Memory: A Theory of Learning in Computers and People*, Cambridge University Press, 1982.

4. 事例ベース推論の要素技術

4. 1 事例の柔軟な特徴づけ

N T T 情報通信処理研究所 打橋 知孝

要 旨

設計などの合成型問題分野における問題解決手法として事例ベース推論(Case-Based Reasoning,以下 C B R)が提案されている。本節では、C B Rの部分処理である事例検索及び事例ベースの構築に着目、事例ベースの使われ方を検討し、検索タグとなる事例の特徴付け(INDEXING)について考察した。

4.1.1 はじめに

診断問題を対象にしたエキスパートシステムではルールベース推論が行われている。このルールベース推論を行うためには知識がルールとして記述できるまでに整理・分類されている必要がある。しかし、特に設計問題においては知識を獲得・整理することが困難であり、エキスパートシステム開発におけるいわゆる知識獲得ボトルネックが存在する。C B Rはこの知識獲得ボトルネックの解決の一案となる。

本節では、事例ベースの構築及び事例の検索処理に注目し、検索時に有効な事例へのINDEXINGについて検討する。

4.1.2 事例ベース構築支援

4.1.2.1 C B R 基礎技術

C B Rシステムは一般的に、事例ベース、問題解析器、事例検索器、事例修正器、事例修復器、事例格納器の6個の構成要素からなる。各構成要素実現に必要な基礎技術には次に示すものがある。

事例の特徴付け：C B Rでは事例の検索が行われる。多数の事例からの効率良い検索が行えるよう適切な検索タグの設定が必要

事例の検索：最適な事例を検索するために評価規範の設定が必要

事例の修正：検索された事例及び修正箇所の妥当性の判定や修正のための知識の導入が必要

事例の修復：修復するための知識の設定が必要

事例の格納：C B Rによる問題解決後、問題事例は参照事例として事例ベースに格納できる。新規事例格納に伴うINDEX変更など事例ベース管理が必要

事例ベース：検索に有効な情報の抽出やデータ構造等の設定が必要

4.1.2.2 事例ベースに望まれるもの

事例とは、ある解が存在するときに確認できる事実である。しかし、その全事実を事例として事例ベースに格納することは無駄である。

例えば機械設計において、ある機能を実現する装置が存在する場合に、この装置の構造や構成（いかにして機能を実現しているかを示す）の他に、サイズ、材質、色、作成者等が事実として確認できる。この機械設計の例にある“装置の色”という情報は“機能”には関係無いため、問題事例と類似の“機能”を持つ事例を検索する際には注目する必要の無い情報である。そのため、事例ベースに事例を収めるには、検索時に有効な情報の抽出・整理を行う必要がある。

4.1.3 INDEXING

前記のように事例ベースに事例を収めるには、検索時に有効な情報の抽出・整理を行う必要がある。これらの情報は検索タグとして利用できる。

4.1.3.1 INDEXINGの必要性

多量に存在する検索対象の内から問題事例と最も類似の事例を短時間に検索しなければならない。しかも、検索結果が的確であることが望まれる。これを満足する検索は雑に造られたデータ（事例）ベースを利用しては実現できない。データベース中に十分に吟味された検索のためのタグが用意されていることが必要となる。

4.1.3.2 INDEXINGへの要求

検索されることを前提としてINDEXを吟味・選出しなければならない。4.1.2.2では検索時に有効な情報の抽出・整理を行う必要があると述べた。こうすることによって“余計なものを含まず、必要なものは漏れなく”検索できることをねらったのだが、事例中の情報の有効性はその使われ方に依存する。

前記の機械設計の例で“装置の色”という情報は問題事例と類似の機能を持つ事例を検索する場合には不要な情報であるとみなした。しかし、問題事例と同様の売れすじの事例を検索する場合には“装置の色”は有効かも知れない。このように同一の事例に対しても解決する問題の性質に応じて有効な情報が変化する。つまり、観点（検索の目的）の変更によって変化する有効な情報をINDEXにする必要がある。

CBRで行われる検索は類似検索であるため完全なマッチングは行われない。また、問題解決において事例の部分修正が行われる。この修正によって、その事例の持つ本来の特徴が失われ問題解決に無効になる可能性もある。このような自体を防ぐために、必ずマッチングしなくてはならないもの、修正をしてはならないものを指定する必要がある。

る。

また、INDEXに言葉を使うと検索者によって同一の意味に対して異なった言葉を設定（“住所”と“アドレス”や勝手な略語など）し、雑音が増加することが考えられる。言葉の規制を行う必要がある。

4.1.3.3 実現案または各種手法の適用

観点の変更に対応できる検索を行うために、事例のINDEXINGにあたって、検索の目的を明らかにして観点の設定を行ってから、問題解決の観点に応じたINDEXINGを観点毎に行うことが考えられる。観点を設定した後のINDEXに利用する特徴の抽出に関してはE B L（Explanation-Based Learning：説明に基づく学習）を利用し、説明木を作成していく過程でINDEXとして不要なものを排除する手法[Barletta 88]がある。また新規事例の追加や希望の検索結果が得られない場合のINDEXの変更などに対しては、INDEXの一般化・詳細化を行うことで新たな事例への対処を行った手法[Sycara 89]がある。

必ずマッチングしなくてはならないもの、修正をしてはならないものを指定するためにINDEXに重要度を導入することが考えられる。INDEXの重要度付与に関してはA H P [Saaty 80]の利用が考えられる。

検索時の雑音を軽減するための言葉の規制には、シソーラスのようなINDEX用語辞書を用いる。またシソーラスでは用語が階層化されているのでその構造を上下することによる類似検索が可能になると考えられる。

この他にもK J法によるINDEX整理やI D 3を利用した有効情報抽出など知識獲得手法とC B Rとの融合が考えられる。

4.1.4 まとめ

以上のようにINDEXINGにあたっては、

- ①観点の設定、
- ②各観点で必要な特徴の抽出、
- ③各観点毎のINDEXING、
- ④その事例における各特徴の重要度設定、
- ⑤INDEXINGに用いた用語の類似語定義（用語辞書）

が必要であり、これらにはE B LやA H P等の知識獲得手法を適用できることを報告した。また、初期事例ベースのINDEXINGの結果は固定ではなく、新しい事例の入手や期待の検索結果が得られなかった場合のINDEXINGの変形・生成もサポート出来なければならない。

INDEXINGの問題点としては、検索のされ方やその事例の使われ方のパターンをできる限り洗いだして把握しておき、それを前提にして特徴を抽出、INDEXINGする必要があると思われる。

<参考文献>

[Barletta 88]Ralph Barletta:Explanation-Based Indexing of Cases,

DARPA'88 pp.50-60(1989)

[Saaty 80]T.L.Saaty:The Analytic Hierarchy Process,McGraw-Hill(1980)

[Sycara 88]Katia P.Sycara:Index Transformation and Generation for Retrieval,

DARPA'89 pp.324-328(1989)

4.2 最小原理に基づく事例の組織化¹

石田好輝 京都大学, 工学部 応用システム科学教室

要旨

ここでは、ダイナミックメモリを実現するための手法であるスロット最小化による事例の組織化を提案する。スロット最小化による組織化は、NP 完全であるので、これをより簡便に実現するアルゴリズムについても述べる。さらにこれを診断事例の組織化に用いたときの問題点を指摘し、因果性に基づくより深い類似性を導入する。このより深い類似性がスロット最小化メモリでどのように実現されるかも論じる。なおこのスロット最小化メモリじょうでのインプリメントされた診断システムとして CAOS があるが、それについては 5.6 で述べる。

4.2.1 はじめに

ルールベース推論がほとんどの知識ベースシステムで用いられてきた。しかしながら、人間の問題解決では他の多くの推論方法も用いられる。そのような推論方法のうちルールベース推論と対照的とも言える推論方法に事例ベース推論がある。比較のため、裁判における裁判官の問題解決過程を例にとろう。裁判官は大まかに次の二つの方法によって判決を決定するものと考えられる。(1) 現在の事例に適用可能な法律を適用して判決をだす。(2) 過去の類似の事例を参照し、その事例を変形して判決をだす。大ざっぱにいうと、前者がルールベース推論に対応し、後者は事例ベース推論に対応する。事例ベース推論 [Schank 86] は、過去の事例を現在の問題となっている事例にマッピングするのに、類推を用いる。さまざまな類推 [Winston 82, Carbonell 85] のなかで、事例ベース推論のみが動的記憶との関連で議論されてきた。

IPP [Lebowitz 82] は新聞からテロリズムに関する記事を抽出し一般化する。CYRUS [Kolondner 86] は、アメリカの国務長官に関する事実を異なった特徴で索引することにより記憶を組織化する。JUDGE [Bain 86] は事例ベース推論により裁判における宣告を行う。CHEF [Hammond 86] は事例ベース計画システムであり、四川料理に应用されている。その動的記憶は、過去の料理の経験事例を、それが満たすべきゴールとそれが避けるべき失敗により索引している。それらの索引は、現事例についてそのゴールを構成するために必要な類似事例の想起に用いられる。

我々の提案する知識の動的表現もこの方向に沿うものである。人間の知識の構造は、新しい事例が記憶される度に動的に変えられるものである。この時の構造的変更は、新しく入ってきた事例の抽象性のレベルや、その事例が既に記憶されている事例とどのような関係にあるか、すなわち現在の知識構造に整合するものであるか、矛盾するものであるかなどに依存する。新しい事例²を記憶するときには、いつもその事例を、既に記憶されている事例になんらかの形で関連づける。記憶はこのような関係付けのみにより行われると

¹この節は、[石田 87] を詳細化した報告であり、[Ishida 89] からの抜粋である。

²ここで事例は、あるイベントを記述する属性-属性値ペアの集合のことである。

仮定する。このような動的記憶の特徴としては、(1) その記憶における想起、検索はその事例の内容自体と関係を用いて行われる。(2) その構造は、記憶される事例の順序に大きく依存する。

このような動的記憶を実現するのに、われわれはそのスロットの数を出来るだけ少なくするように管理されている動的なフレーム表現を導入した。すなわちこの知識表現では、事例間で共通なスロット値をもつスロットを共有してスロットの総数を最小に維持するのである。

4.2.2 では、ここで採用された記憶の組織化法と一般化メカニズムについて述べる。よー一般化を行う問題は、組み合わせ最適化の問題として定式化される。4.2.3 では、この記憶構造を用いた事例ベース診断の枠組を提案する。

4.2.2 事例の動的構造化

スロット共有による一般化

[最小スロット問題 (MSP)]

それぞれの事例 S_i は属性-属性値のペアで記述されるとし、事例の集合 $\{S_i\}_{i=1..n}$ が与えられたとき、 $\sum_{i=1}^n s_i - |S_0|$ ($|S|$ は集合 S の要素数を意味する。) を最大にするような 属性-属性値ペアの集合 S_0 を求めよ。ただし

$$s_i = \begin{cases} |S_0| & \text{if } S_i \supseteq S_0 \\ 0 & \text{otherwise} \end{cases}$$

この問題は NP-完全である。³

事例の構造化はこの MSP を繰り返し解くことにより求められる。すなわち、まず与えられた事例の集合 $\{S_i\}_{i=1..n}$ に対して S_0 を求め、 $S_i \subseteq S_0$ となる全ての S_i を S_0 のサブクラスとする。すなわち S_i のうち S_0 にあるものは全て S_0 から継承されるとして省略する。次に同様の操作をこの構造化された事例ベースに対して行う。以下同様の操作を繰り返していく。

```
([element-name] name=[case-name]
      [attribute-value-pair1]
      [attribute-value-pair2]...)
```

ただし [case-name] は事例 (要素) 名であり、その後に任意個の属性-属性値ペアが続く。属性-属性値ペアは、属性名、述語、属性値で記述される。例として、様々な 4 角形のクラスの構造化を考えよう。

³ グラフ理論的に述べると、この MSP は 2 部グラフにおいて最大なエッジ数をもつ完全部分グラフを求めることに対応し、従って NP-完全である。ただしこの 2 部グラフは、事例を表すノードの集合と、属性-属性値ペアを表すノードの集合を用意し、ある事実が、ある属性-属性値ペアを持つ時のみ対応するそれらのノード間をエッジを結んでできるものである。

```

(fig name=square          corner=4 side=4)
(fig name=lozenge         corner=4 side=4
                        EqualSide=4)
(fig name=trapezoid       corner=4 side=4
                        ParallelSide1=1)
(fig name=parallelogram   corner=4 side=4
                        ParallelSide1=1 ParallelSide2=1)
(fig name=rectangle       corner=4 side=4
                        ParallelSide1=1 ParallelSide2=1
                        EqualAngle=4)
(fig name=PerfectSquare   corner=4 side=4
                        ParallelSide1=1 ParallelSide2=1
                        EqualAngle=4 EqualSide=4)

```

まずはじめに $S_0 = (\text{fig name=square corner=4 square=4})$ がその他のクラスの上位クラスとして選ばれる。なぜなら、他のクラスをこのクラス S_0 の下位クラスとすることにより $2 \times 6 - 2 = 10$ 個のスロットが省略できるからである。このようにして次々に構造化を行った階層構造が図1に示されている。上位クラスに共有されているスロット数が示されている。

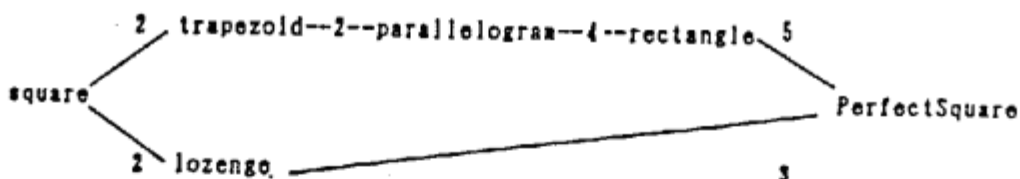


図1 スロット最小化による階層構造

この再構成プロセスは非常に大きな計算量を必要とし、新しい事例が記憶される度にこのような再構成を行うのは現実的ではない。代りに我々は次のインクリメンタル(incremental)アルゴリズムを用いる。これは前のアルゴリズムより効率的であるが、スロット最小化による構造化と一致する保証はない。

このアルゴリズムを述べる前に、事例 C_1, C_2 間に次の様な距離を定義する。

$$d(C_1, C_2) = |C_1/C_2|/|C_1 \cup C_2|$$

ここで C_1/C_2 は集合 C_1, C_2 の対象差 (symmetric difference) である。すなわち、 $C_1/C_2 = (C_1 - C_1 \cap C_2) \cup (C_2 - C_1 \cap C_2)$ 。事例 A, B, C が図2のように関連している

時、A は B, C の上位事例 (*supercase*)、B, C は A の下位事例 (*subcase*) と呼ぶ。B は C の近傍事例 (*close case*) と呼ぶ。もし事例間に何の関係も無いときは、距離は 1 となる。逆に事例が全く一致するときは 0 となる。



図 2 上位事例と下位事例

5.6.1 で議論される推論において、この距離は事例変形作用素を選択するのに重要な役割を果たす。

[インクリメンタルアルゴリズム]

ある与えられた事例について、

1. 想起：事例ベースのなかから、上に定義する距離で最も近い事例を見つける。
2. 記憶：与えられた事例をその最も近い事例に次のように関連づけることにより記憶する。
 - もし与えられた事例が、最類似事例の上位事例ならば、その与えられた事例を最類似事例の上位の適当な位置に挿入する。
 - もし与えられた事例が、最類似事例の近傍事例ならば、その与えられた事例を最類似事例と共通な上位事例の下位事例とする。その様な事例が無ければ、最類似事例との共通な属性-属性値ペアを一般化した上位クラスを生成し、与えられた事例と最類似事例をその下位事例とする。

動的記憶機構の診断事例への応用

Fig.3 [Belenblut and Whitehouse 73] に示されるような簡単なプロセス診断の例を考えよう。

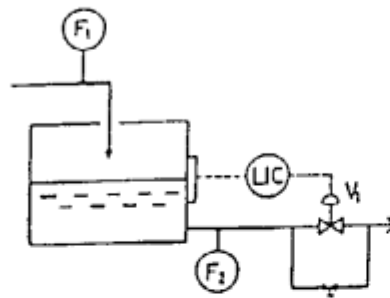


図 3 簡単なバッファータンクの例

プロセスはインフロー $F1$ 、アウトフロー $F2$ 、レベル $L1$ 、制御バルブ $V1$ をもつ簡単なバッファータンクである。タンクのレベル $L1$ はバルブ $V1$ の開度を調節することにより一定のレベルに保たれている。正味のフロー DF は $F1 - F2$ で定義される。表 1 は診断のためのディシジョンテーブルである。各々の列は、症状とそれに対応するイベント（最下段）を示している。ここでの我々の問題は、多くの診断事例が与えられとき、それらを表 1 のように事例ベース推論が行い易いように構造化することである。

表 1 診断のためのディシジョンテーブル

Rule	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
ΔF	-1	-1	N																	
$F1$				L	H	H														
$L1$							L	L	H	H	*									
$V1$							O	H	H	C	*	O	O	O	H	H	C	C	C	H
$F2$												L	H	H	L	H	L	H	H	H
Event	5	A	1	6l	6h	1	A	A	A	A	1	4	4	6h	4	2	5	2	2	1
Flow Balance Supply Control Loop Leaks and Blockage																				

Events:

- (1) Normal operation
- (2) Pipe leakage between $F2$ and $V1$
- (3) $V1$ by-pass open in error
- (4) Blockage in exit line
- (5) Leak in tank
- (6h) Abnormal throughput high
- (6l) Abnormal throughput low

(A) Anomaly (defined as a physically impossible combination of measure outputs from the point of view of the plan)

診断事例ベースにおける一般化

スロットを共有して、スロット数を少なくすることによる一般化機構は、診断事例ベースにおいて次のような特長がある。

- (1) 一般化機構は診断事例から、無関係なあるいは誤った症状をフィルターすることができる。
- (2) 一般化機構は、いくつかの複数故障の事例から単一故障のクラスを抽出しうる。一般化機構がどのように働くかを見るために、次の様な事例が最初に与えられているとする。

(event name=BlockInF2&V1_15

```

        fault=block location=F2&V1
        F1=high V1=open F2=low)
(event name=BlockInF2&V1_24
        fault=block location=F2&V1
        F1=low V1=open F2=low)
(event name=BlockInF2&V1_6
        fault=block location=F2&V1
        F1=normal V1=open F2=low)

```

一般化機構は、次の様な上位事例をこれらから生成する。

```

(event name=GEN1
        fault=block location=F2&V1
        V1=open F2=low)

```

ここで、診断に無関係な属性 F1 がフィルターされたのがわかる。さらに、次のような複数故障の事例が与えられたとき、

```

(event name=BlockInF2&V1&AnomalyInControl_31
        fault=anomaly location=control
        fault2=block location2=F2&V1
        F1=normal V1=normal F2=low L1=high)
(event name=LeakInTank&AnomalyInControl_31
        fault=anomaly location=control
        fault2=block location2=F2&V1
        F1=high V1=normal F2=normal L1=high)

```

次の、単一故障の事例が一般化により求められる。

```

(event name=GEN2
        fault=anomaly location=control
        V1=normal L1=high)

```

索引における因果関係の導入

診断において、二つの事例の症状は類似しているがその故障は全く異なるということはしばしば起こる。従って、索引として症状の類似性のみを用いると誤った事例ベース診断に陥り易い。そこで、我々は症状間の因果関係をも索引に用いる。症状間の因果関係とは、それらの属性間の作用関係である。

類推 [Winston 82] において議論されたように、因果関係は事例から事例への類推写像に置いて保存されなければならない。このヒューリスティックは普遍的なものでありほとんどの領域で成り立つ。診断において、我々は事例の索引づけにこの因果関係を用いた。

このような症状間の因果関係を生成するには、基になるモデル（ベースモデル）が必要となる。図4に示されるように、ここで用いるベースモデルは潜在的に因果関係のありそうな全ての属性間に因果の方向づけをしたものである。ベースモデルはコンポーネントごとに付けられる知識であり、診断対象が代らない限り同一のものをを用いる。⁴

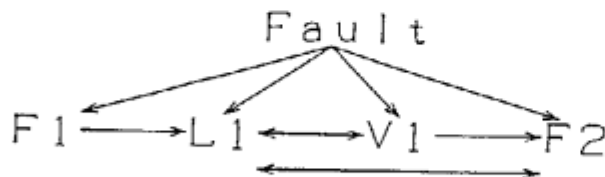


図4 バッファータンクのベースモデル

ベースモデルに置いて、F1 から L1, V1, F2 への因果パスは存在する。しかし、F2 から F1 への因果パスは存在しない。ベースモデルは前もって与えられる必要がある。ベースモデルが与えられると、症状間の因果関係は次のようにして与えられる。もし事例が F1=high と V1=close をその症状としてもつなら、ベースモデルにおいて F1 から V1 へ因果パスがあるため、因果関係 F1=positive_V1 が生成される (positive, negative は属性の値が正方向であるか反対方向であるかにより定義される)。例えば、次の二つはその症状において全く異なった事例である。

```

(event name=ThroughputHigh_14 type=data
  fault=high location=throughput
  V1=open F2=high)
(event name=ThroughputLow_27 type=data
  fault=low location=throughput
  V1=close F2=low)
  
```

しかし、誘導された因果関係は次のように同じものである。

```

(event name=ThroughputLow_16 type=effect
  throughput=positive_V1
  V1=positive_F2)
  
```

⁴あるコンポーネントの診断事例を別のコンポーネントの診断に用いるための索引にはこのベースモデルの類似性により、類似のコンポーネントを探してくることになる。

```
(event name=ThroughputHigh_18 type=effect
  throughput=positive_V1
  V1=positive_F2)
```

各々の事例の症状をデータタイプ、症状間の因果関係をエフェクトタイプと呼ぶ。

4.2.3 おわりに

このようにスロット最小化ダイナミックメモリに、ドメイン独自のより深い因果関係を導入することでより深い類似性が定義できる事が分かった。類似性をより深くすることは、後の事例操作にも大きく影響し、事例ベース推論が成功するか否かを決定する。

参考文献

- [石田 87] 石田好輝: 知識ベース故障診断システムの動向、知的診断を目指して、信学技報、DE-87-12,1-8 (1987).
- [Ishida 89] Y. Ishida: A Framework for Dynamic Representation of Knowledge: A Minimum Principle in Organizing Knowledge, Proc. KR89 ,pp.170-179, (1989).
- [Schank 86] Roger C. Schank: *Explanation Patterns*. LEA Publishers, London, (1986).
- [Winston 82] Patric H. Winston: Learning Principles from Precedents and Exercises. *Artificial Intelligence*,19(3), (1982).
- [Carbonell 85] Jaime Carbonell: Derivational Analogy: A Theory of Reconstructive Problem Solving and Expertise Acquisition. Carnegie-Mellon University, cmu-cs-85-115, (1985).
- [Schank 82] Roger C. Schank: *Dynamic Memory*. Cambridge University Press, (1982).
- [Lebowitz 82] Michael Lebowitz: Correcting Errorneous Generalizations, *Cognition and Brain Theory* 5, :367-381 (1982).
- [Kolondner 86] Janet L. Kolondner: A Process Model of Case-based Reasoning in problem Solving, *Proc. of AAAI-86* :284-290, (1986).
- [Bain 86] William M. Bain: A Case-based Reasoning System in Subjective Assessment, *Proc. of AAAI-86* :523-527, (1986).
- [Hammond 86] Kristian J. Hammond: SHEF: A Model of Case-baed Planning, *Proc. of AAAI-86* :267-271, (1986).

[Belenblut and Whitehouse 73] B.J. Belenblut and M. B. Whitehouse: A Method for Monitoring Process Plant Based on Decision Table Analysis. *chem. Eng.*, :175-181, (1973).

4. 3 類推について

東京工業大学 原口 誠

要 旨

筆者ら類推の機械化を目指して、特に類推と演繹推論との関係に重点をおき、類推の形式化とその機械化の可能性を追及するという目標のもとに、類比の理論および類推の理論について新しい概念を導入し、その定式化を試みてきた。

一方、Russellは、既存の類似性に基づいて新たな類似性を推論する方法を演繹的な規則として記述し、これを類推を行うための演繹的な知識として用いるDBAR(Determination-based Analogical Reasoning)を提唱している。

類推の理論とDBARでは、類推の形式化の方法や正当性の保証に関して異なっている。計算機による類推を考えると、どのような対象知識に対して、いかなる類推規則を適用するのか、また類推規則が参照することができる類推のための背景知識としてどのようなものが可能であるかが、キーポイントであるという問題意識のもとにこれまでの類推システムを概観し、妥当な背景知識としてどのようなものが考えられるかを考察する。

4.3.1 はじめに

近年、計算機による様々な推論の研究が盛んである。従来からその機械化が研究されてきた演繹推論を始め、帰納推論、非単調論理に基づく推論などが考察されている。ここで考える類推もそのような推論の一種として捉えることができるだろう。推論とは一般に、与えられた前提からある論理に基づいて結論を引き出す操作・過程を意味している。従って、推論研究においてはどのような前提からいかなる結論を出したいのかをまず明らかにする必要がある。演繹推論は通常、現在関心を持っている対象領域に対して十分な前提が与えられていることを仮定し、そこから論理的に正しい結論を引き出す。本稿で焦点をあてる類推においては、現在与えられている知識の体系が十分既知ではないときに、他の問題領域の知識を類似性に基づいて変換し、一つの試みの推論を行うことと考えられる。

筆者等[Haraguchi85-87]は、このような類推の機械化を目指して、特に類推と演繹推論との関係に重点をおき、類推の形式化とその機械化の可能性を探ってきた。すなわち、類比の理論[haraguchi85a, 85b]では対象が事実の有限集合で与えられたときに、類比の順序に基づいて極大類比を求める方法を論じた。ここで極大類比とは異なる対象間で成り立つ極大な類似性を表していた。また[Haraguchi86-87]（以下類推の理論という）では、類推においては特に演繹との関係が重要になるとの観点から、ホーン論理の枠組みの中で類推

を考え、これをルールの具体化、ルールの変換、及び三段論法の三つの推論規則をもつシステムとして定式化した。さらに論理プログラムの類推和の概念を導入し、類推を演繹の延長線上に捉えることができることを示した。

一方、既存の類似性に基づいて新たな類似性を推論する方法を演繹的な規則として記述し、こうした規則を類推を行うための演繹的な知識として用いる D B A R (Determination-based Analogical Reasoning)の研究もある[Russell86,87]。D B A Rではその推論の定義から、推論結果の正当性を常に主張できるという大きな長所を持つ。一方、Winston の因果関係に基づく類推[Winston80,83]から出発した類推の理論では、類似性に基づく推論の方法それ自体は演繹システムの外部で規定しており、推論結果の正当性について直接的にはその保証を与えない。

このように類推の理論と D B A Rでは、類推の形式化の方法や正当性の保証に関して全く異なっている。その本質的な相違点は、類推の対象となる知識と、類推を行うため用いる規則（以下、類推規則と呼ぶ）を明確に区別するかどうかという相違点であると言える。ここで類推の対象となる知識とは、類推規則が推論の対象とする知識で、従来の類推システムが採用していた因果関係等の高階の関係や含意規則を含んでいる。また類推規則としては、ルールの変換規則や類似性に関するヒューリスティックスも含むと考える。

計算機による類推を考える時、どのような対象知識に対して、いかなる類推規則を適用するのか、また類推規則が参照することができる類推のための背景知識としてどのようなものが可能であるかが、今後の類推研究のキーポイントになるように思える。そこでこの問題意識のもとにこれまでの類推システムを概観し、妥当な背景知識としてどのようなものが考えられるかを考察してみたい。

4.3.2 類似性推論のための高階の関係

まずここで論じる類推をいわゆる予測推定型の類推に限定して考えてみる。予測推定型の類推では、推論の前提である対象が陰に持っている類似性を抽出し、新たな類似性を導き出すことによって、ある種の事実やルールの成立を予測推定する。すると類推の前と後では、類推システムによって認識された類似性は増大している。このような予測推定型の類推においては、類推規則とは類似性を増大させるために用いる知識であると述べることができるだろう。そうした類推規則の典型的な例として、構造写象理論[Gentner82]でいう高階の関係に基づく類推を先ず挙げることができる：『高階の関係が（類推において）より重要であり、重要な関係を保存させることにより類推を行う。』

筆者らが類推の理論を作るさいにその出発点とした Winston の類推原理も、基本的には上記の思想を因果関係という一つの高階の関係に設定したものとなっている：『類推とは因果関係を保存させる推論である。』

こうした因果関係等の対象知識は、類推システムに事前に与えられていると仮定することはできず、他の推論系の力を借りて類推の対象記述から抽出することが必要になる。実

際 Winston のシステムでは与えられた対象記述（簡単な英語風の文章）に対して、デモンによる演繹推論を行うことにより、できるだけ多くの因果関係を含む関係情報を読み取り、その後に類似性を決定するためのマッチングを行っている。

このような類推システムが賢く振舞う事ができるための条件は、類推の対象記述から関係情報を獲得する推論手続きが豊富な知識をもって演繹を行う能力を持つことである。また素朴な疑問として、そうした関係情報をどの程度事前に獲得すればよいのか、関係情報の獲得とそれに基づいた類推を全く独立に行ってよいのか、という問題も起きる。ルールの変換に基づく類推の理論[Haraguchi86-87]は、こうした観点から、論理的な含意関係（実質的含意）を類推における高階の関係として採用し、類推と演繹を一つの演繹システムのなかで統一的に実現することを試みたものである。

4.3.3 類似性の演繹推論

ルールの変換に基づく類推の理論では類推規則とはルールの変換であり、Siのルールの具体化が対象知識となっている。一方、既存の類似性に基づいて新たな類似性を推論するための経験則を演繹的な規則として記述し、その規則を用いる演繹推論で類推を実現するDBAR (Determination-based Analogical Reasoning)の研究もある[Russell86, 87]。

DBARの基本的な考え方は、類似性を推論するときに用いる経験則を演繹的な知識として整理し、演繹推論の論理性（推論の正しさ）をそのまま類推に適用しようとする。この類推の論理性問題は正確には、類推が論理的に正当化されうる条件を求める問題として与えることができる。Russellはこの問題に対する一つの解答として、決定規則に基づく推論方式を与えた。

定義 4.3.1 決定規則(total determination rule)

$P(x, y)$ と $Q(x, z)$ を x, y, z を自由変数（のベクトル）として含む論理式とする。このとき、 $\forall w \forall y \forall z [P(w, y) \wedge Q(w, z) \rightarrow x[P(x, y) \rightarrow Q(x, z)]]$ が成立するとき、 $P(x, y)$ は $Q(x, z)$ を決定するといい、これを $P(x, y) > Q(x, z)$ と書く。決定規則 $P(x, y) > Q(x, z)$ を直感的に述べれば、『任意の x と y に対して、 x と w が P に関して同じ値 y をもつなら、 Q に関して同じ値 z を持たなければならない』ことを意味し、関係データベースにおける従属性に相当した概念になっている。この決定規則を用いた場合の類推の推論図式は以下のように表すことができる。

$$\frac{P(T, y) \wedge P(S, y) \wedge Q(S, z)}{Q(T, z)}$$

ここで、 S と T は類推のソースとターゲットを表す記号であり、 S と T が P に関して同じ性質を有するという類似性から Q に関する類似性を決定規則を用いた演繹推論で導き出す推論になっている。したがって類推は、決定規則によって演繹的に導かれ、それ故に前述した類推の論理性問題に答えていることは明らかである。また、Winston 及び Gentner の類推や類推の理論においては、類推規則の対象となる高階の關係に類推規則を適用する

形で推論が行われた。それに対して、決定規則を用いた類推の演繹推論においては、対象知識を類推規則に適応する方式それ自体を決定規則の形であらかじめ準備しておき、そうして演繹規則化された論理式を用いた演繹推論を行う。

決定規則 = 対象知識 + 類推規則の適用結果

このことから、演繹推論を推論エンジンとする知識の処理系、特に論理プログラムで決定規則を用いた類似性の演繹推論が可能になることも明らかであろう。以下で述べる類推手続きは、 $P(x, y) \supset Q(x, z)$ の Q に関する z の値を求める質問から、 P の y 値と同じ値を持つソースを探索し、そのソースの Q の z 値を求める手続きになっている。

決定規則を用いた類推手続き

入力：質問式 $QT(z)$ ；但し z は質問式中の自由変数、

出力： z の解代入

手続き本体：

begin

決定規則 $P(x, y) \supset Q(x, z)$ で

$Q(T, z)$ と $QT(z)$ が単一化できるものを探しなさい；

$P(T, y)$ なる y の値をもとめなさい；

$P(s, y) \wedge s \neq T$ なる s を求めなさい；

$Q(s, z)$ なる z を求めこれを z の束縛値としなさい

end

例 4.3.1

$\text{diet}(X, D, T), \text{exercise}(X, E, T), \text{metabolism}(X, M) \supset \text{weight_gain}(X, W, T)$

なる決定規則と

$\leftarrow \text{weight_gain}(\text{bert}, Y, \text{today})$

なる質問式を考えてみる。まず、決定規則の右辺 $\text{weight_gain}(X, W, T)$ と

質問 $\text{weight_gain}(\text{bert}, y, \text{today})$ は単一化できるから

$T = [\text{bert}, \text{today}]$ と単一化 $\theta = \{X := \text{bert}, W := y, T := \text{today}\}$

を得る。 θ を決定規則の左辺に適応して次の“サブゴール”

$\leftarrow \text{diet}(\text{bert}, D, \text{today}), \text{exercise}(\text{bert}, E, \text{today}), \text{metabolism}(\text{bert}, M)$

を生成し、 bert に関する D, E, M をそれぞれ得る（これを α, β, γ とする）。

次に $T = [\text{bert}, \text{today}]$ に関する $\text{diet}, \text{exercise}, \text{metabolism}$ の値、 α, β, γ を同一にするソース $S = [\text{Someone}, \text{Date}]$ を求めるために

$\leftarrow \text{diet}(\text{Someone}, \alpha, \text{Date}), \text{exercise}(\text{Someone}, \beta, \text{Date}),$
 $\text{metabolism}(\text{Someone}, \gamma)$

なるゴールを発生し $\text{Someone} := \text{john}, \text{Date} = 11/19/82$ を得たとする。すると初期ゴールの変数 Y に対する答えは

weight_gain(john, W, 11/19/82)

の W の値で決定できることになる。

この例から容易に分かるように、決定則 $P(x, y) > Q(x, z)$ を用いた類推では少なくとも一つ $P(w, y_0) \wedge Q(w, z_0)$ なる w の実例が存在する場合は、このことから、 $\forall x [P(x, y_0) \rightarrow Q(x, z_0)]$ なるルールを使用できることを意味している。例えば 国籍が言語を決定する規則 $\text{nationality}(x, y) > \text{speaks}(x, z)$ は 一人の日本人が日本語を話すと分かった以降は、全ての日本人は日本語を話すと推論してよい保証を与えることに他ならない。本稿の冒頭で述べたように、類推とはある時点での知識の体系が完全ではないときに、あるいはそのような知識の体系を仮定できない問題領域において、知識の類似性に基づく変換により一つの試みの推論を行うことである。決定規則を用いた類推では、『全ての日本人が日本語を話す』といった強い含意知識を仮定できないときに、決定規則と唯一つの事例から別の事例の結論を導くことができるという利点があるが、結果的には強い含意知識を用いることを陰に期待しており、この意味で決定規則の定式化には不満が残る。また、一つの決定則によって、何に関する類似性からどのような類似性が帰結できるかが完全に（演繹的に）規定されるという意味で類似性推論の枠組みとしては強すぎると言えるだろう。

4.3.4 類似性の拡張推論

計算機による類推を考える際に最も重要な問題点として

(1) アナロジーの検出方法、

(2) 検出された類比（アナロジー）に基づいて知識を変換する方法

をあげることができる。また、一般に類推はある問題を解決するために、またはある質問に答えるために起動されると考えてよいだろう。ルールの変換に基づく類推の理論に基づく類推手続きも、与えられたゴール（質問）に対して、類推の対応付けを求めながらルールの変換と適用を同時に行う一種の反駁手続きとして実現されている[Haraguchi87]。この反駁過程において類推の対応付けは段階的に、類似性を定義する以前に求めた対応付けを拡張する形で次の対応付けが求まる過程にもなっている。このように、類推を行う前と後では類似性に関する情報量が増大すると考えることができ、別の表現をするならば、類似性の拡張推論こそが類似性検出プロセスにおいてもっとも基本的な問題となるとも言えるだろう。この類推における

(1) 質問駆動型、(2) 類似性の拡張・増大

という二つの性質に関しては前節で述べた決定規則を用いた類推に関しても全く同じであるが、拡張・増大の仕方が決定規則によって一意的に決ってしまうという意味で強すぎると言った。一方ルールの変換に基づく類推では、対応付けが全く構文的に行われるために、以下の例のように類似性を旨く推論出来ない場合がある。

例 4.3.2 S1とS2をそれぞれ

```

S1= { F1: speak(a, english),
      F2: nationality(a, united_kingdom),
      F3: official_language(english, united_kingdom),
      F4: born_at(a, united_kingdom),
      F5: native_language(english, united_kingdom) },
E1: F1 ← F2,
E2: F2 ← F4 }
S2= { F6: born_at(b, japan),
      F7: nationality(b, japan),
      F8: official_language(japanese, japan),
      F9: native_language(japanese, japan) }.

```

として、類推システムに $\leftarrow \text{speak}(b, X)$ なるゴールが与えられたとする。するとルールの変換

$$\frac{\text{speak}(a, \text{english}) \leftarrow \text{born_at}(a, \text{united_kingdom})}{\text{speak}(b, X) \leftarrow \text{born_at}(b, Y)}$$

が試みられる。ここで変数 X, Y はそれぞれ `united_kingdom` と `english` に対応・束縛されるべき物を表している。変数 Y の値を求めるためにサブゴール $\leftarrow \text{born_at}(b, Y)$ が生成され $Y = \text{japan}$ によって反駁は終了する。このとき得られる対応付けは

$$\Phi = \Phi(X) = \{ \text{english} \sim X, a \sim b, \text{united_kingdom} \sim \text{japan} \}.$$

類推和の理論に従うと $\Phi(X)$ が 1 対 1 である任意の X に対し $\text{speak}(a, X)$ は $S1 \Phi S2$ からの論理的帰結となってしまふ。 X が特定されずにルールの変換を用いた反駁が終了するのは X に対応付けられた定数記号 `english` がルールの定義部に現れないことに直接的な原因があり、`english` に相当する S の個体定数を、既存の類似性

$$A = (\{ \text{born_at}(A, B) \}, \{ A \sim \langle a, b \rangle, B \sim \langle \text{united_kingdom}, \text{japan} \rangle \})$$

から推論して求める必要がある。このような個体定数が求まった時点で類似性 A を拡張した類似性 B が推論できたことになる。

このような類似性の推論を論じるために本節では、類推の対象を事実のデータベース、すなわち、事実と事実に関する確定節の有限集合とし、 $D = (D, F)$ と記する。ただし、 D は事実の有限集合、 F は $\alpha \leftarrow \beta_1 \dots, \beta_n$ の形をした節の有限集合で、 α, β_j は事実を表すとする。類推の対象 D_1 と D_2 の間の類似性としては、類比の理論という部分同一の考え方を踏襲する。

定義 4.3.2 $D_i = (D_i, F_i)$ を事実のデータベースとする。このとき、

- (1) $D_1 \times D_2$ の部分集合 A は、 A の要素 $\langle \alpha, \beta \rangle$ に対して、 α と β が同じ述語記号を持つとき、選択 (selection) と呼ばれる。
- (2) 選択 A に対して $\Phi(A)$ を

$\Phi(A) = \{ \langle s, t \rangle \mid \langle p(s_1, \dots, s_n), p(t_1, \dots, t_n) \rangle \in A \text{ なる事実の組と} \\ \langle s, t \rangle = \langle s_i, t_i \rangle \text{ なる添字の組が存在する} \}$

で定め、 A の対応付けと呼ぶ。(3) $D_1 \times D_2$ の部分集合 A は、選択でありかつ $\Phi(A)$ が 1 対 1 性の条件を満たすとき、 D_1 と D_2 の間の類似性と呼ばれる。

定義 4.3.3 D_1, D_2 を事実のデータベース、 A を D_1 と D_2 の間の類似性とする。さらに D_1 の事実 $\alpha = p(t_1, \dots, t_n)$ と変数を含む $\beta = p(s_1, \dots, s_n)$ を考える。このとき $\langle \alpha, \beta \rangle$ に関する A の拡張 (extension) とは

(EC: 拡張性) $A \subseteq B \subseteq D_1 \times D_2$

(AC: 応答性) $\{\langle \alpha, \beta \rangle\} \cup \Phi(B)$ が EPIC を満たす。

上記の定義において、応答性 AC は、集合としての A の拡張 B が $\langle \alpha, \beta \rangle$ と EPIC の意味で両立することを要求し、したがって、ある基礎代入 θ によって、 $\{\langle \alpha, \beta \theta \rangle\} \cup \Phi(B)$ が D_1 と D_2 の類似性になることを意味する。またこの代入が β に現れる変数の解代入となる。 A に対する拡張 B を求める過程は、 p_1, p_2 を射影として、 $D_1 - p_1(A)$ の事実と $D_2 - p_2(A)$ の事実の対応付ける過程である。この過程を図 4.3.1 の非決定的な手続きで記述しておく。

```

Procedure make_extension(A,  $\alpha$ ,  $\beta$ ), where
    A is an analogy between D1 and D2,
     $\alpha$  is a ground atom  $p(t_1, \dots, t_n) \in D_1$ ,
     $\beta$  is a atom  $p(s_1, \dots, s_n)$  containing some variable.
begin
    find a fact  $\gamma \in D_1 - p_1(A)$  such that
         $\gamma$  contains  $t_j$  or its subterms;
    let  $\gamma = q(u_1, \dots, u_m)$ ;
    find  $\sigma \in D_2 - p_2(A)$  with the predicate symbol q;
    check if  $B = \Phi(A) \cup \{\langle \gamma, \sigma \rangle\}$  satisfies EPIC or not
        if B violates EPIC then fails
        else if B satisfies the condition AC
            then return(B)
            else make_extension(B,  $\alpha$ ,  $\beta$ )
        endif
    endif
end

```

図 4.3.1 類似性の拡張推論を行う非決定的手続き

さて、以下で示される例の場合、english に対応する D_2 の個体定数を特定できる二つ

の可能なAの拡張 B1, B2 が存在する.

D1 = { F1: speak(a, english),
F2: nationality(a, united_kingdom),
F3: official_language(english, united_kingdom),
F4: born_at(a, united_kingdom),
F5: native_language(english, united_kingdom) },

D2 = { F6: born_at(b, y),
F7: nationality(b, y),
F8: official_language(english, y),
F9: native_language(ylang, y) }.

A = { <F2, F7>, <F4, F6> },

α = speak(a, english),

β = speak(b, X),

B1 = A $\cup$ { <native_language(english, united_kingdom),
native_language(ylang, y)> },

B2 = A $\cup$ { <official_language(english, united_kingdom),
official_language(english, y)>

ここでB1は born_at, nationality, native_language に関する類似性に着目して english に対応する ylang を特定する. 一方B2では, born_at, nationality, official_language に関する類似性によって, english に対応するものは english であると推論できる. B1, B2ともに定義4.3.3の要請を満たすが, 異なる推論結果を導く. 形式的に述べると文献[Haraguchi85a, 85b]の意味でB1とB2は極大類比であり, 比較不能になっている.

4.3.1節で, 類推規則がどのような知識を用いることができるかが問題になると言った. 極大類比の選択基準も, どのような類似性をより好ましいものとするかという基準を与えろという意味で, 類推規則が参照する背景知識だと言えることができる. 一般的に述べるならば, そのような多用な類似性の中から妥当な類似性を選択するための背景知識は, 推論の対象領域や推論自体の目的に依存する. 逆の言い方をすれば, ルールの変換や, 類似性の拡張機構をなどの推論部に外付で与えてやることにより, 問題に応じた類推機構を構築できる可能性がある. ちょうど, 情報検索においてシソーラス等の語の関係情報データを活用するのと同じ考え方である.

この問題意識の下で, 先の2つの極大類比の中から一つを選び出すことを考えてみよう. 類似性の拡張推論の前提Aは born_at と nationality に関する類似性に着目している. 問題はAの類似性が妥当であると判断されるとき, Aに加えて native_languageの類似性に注目すべきか, あるいは official_language に注目すべきかの選択の問題である. この問題に対する一つの自然な解答として, 『類似性の推論の前提たるAが注目している類似

性とできるだけ関係のある類似性を選択する』。

そうすると、類似性を選択するために用いることができる背景知識とは、語あるいは概念間の関連情報(relevancy)に他ならないことがわかる。すなわち、native_langauge なる概念の方が officail_language という概念よりも、Aで扱っている born_at, nationality により関係があるといった推論を行うことになる。

このような概念間の関連情報を扱うための技法として、例えば情報検索や自然言語処理におけるシソーラスにおいて共通の上位概念を持つものは関連していると言える。また例えば、CYC [Lenat86] において採用しているように、概念フレームが共通した定義スロットあるいはスロット値を持つことでも関連しているということもできよう。あるいは、情報検索の手法それ自体を(簡易的に)関連性決定手段として用いることもできよう。すなわち、概念名、制御されたキーワード集合、及び概念を説明する自然言語の抄録を基本単位とする概念データベースを考える。情報検索システムは与えられたキーワードを持つ概念を、キーワード集合や抄録から切り出されたキーワード集合からその関連度を決定し、質問式に関係のあるデータとして出力する。我々の類似性推論の場合はデータベース中の全ての概念データに当たる必要はなく、類似性推論の前提に現れる概念名に対し、可能な拡張に新しく現れる概念が検索手続きによりヒットすればそれで十分である。

こうした関連度決定手続きが類似性推論において有効であることを見るためには、今後の実験が必要であるが、このことに関しては別の機会に論じてみたい。いずれにせよ、そうした関連概念の検索手法を用いることにより、与えられた概念名、すなわち述語記号 p に対して、 $R(p) = \{q : p \text{ と } q \text{ は関連する}\}$ なる R を決定できることになる。

定義 4.3.4 A を類似性、 p と q を A には現れない述語記号とする。また $\Pi(A)$ を A に現れる述語記号の集合とする。このとき、 p は

$$q \cap \Pi(A) \subseteq p \cap \Pi(A)$$

であるとき、 A に関して q より関連度が高いと呼び、 $q \leq p$ と記す。

この類似性推論の前提 A に関する述語記号の順序関係を用いて、応答性と拡張性を満たす新たな類似性を求める手続きを、類比的理論[Haraguchi85a, 85b]で与えた極大類比の下降型の探索手続きをそのまま用いて実現できる。

4.3.5 類推と無矛盾性

既存の類似性に基づいて新たな類似性を推論する方法を演繹的な規則として記述し、こうした規則を類推を行うための演繹的な知識として用いる DBAR ではその推論の定義から、類推は決定則と他の知識からの演繹推論となり、したがって推論結果の論理的正当性を常に主張できるという大きな特徴を持った。また類似性推論の方法それ自体が演繹的な規則で与えられているので、与えられたターゲットに関する質問から関連するソースを選択することが可能である。

ここでは、類推和の理論で示した類似性を与える写像を求めながら同時にターゲットで

の推論を行う技法と、D B A Rのソースを自動的に求める長所を兼ね備えた類推の再定式化を与える。この再定式化においては、いくつかの確定節で書かれた制約と多くの正あるいは負の個別的なデータ(atomic fact)から構成されたデータベースを想定している。類推和の理論とは対象的に、データベースは唯一つ与えられる。そうしたデータベースに対する推論の出発点ではソース領域に相当するものは不定であるが、推論が進行するに従い部分データベースとして特定化される。また負のデータは、帰納推論において過度の一般化を避けるために用いるのと全く同様に、過度の類推を防ぐために用いる。すなわち、得られた類推は少なくとも現在のデータとは矛盾しないという意味での妥当性基準を満たすものとなっている。この妥当性基準は、前節で述べた類似性の拡大推論における可能な選択枝を狭めるものとして考えることも可能である。

4.3.5.1 類推の再定式化

単一の対象領域に対する類推を考えるために、まず類推の再定式化を与える。Lを有限の述語記号と定数記号を持った一階言語、Lcを定数記号だけからなるLの部分言語とする。このとき推論の対象領域Dを組(P, N)で与える。但し、PはLの確定節の有限集合、NはLの基礎アトムの否定の有限集合で、P ∪ Nは無矛盾であることを仮定する。Nの要素を否定的事実、Pの基礎アトムを肯定的事実と呼ぶ。こうした領域での類推を考えるために部分写像 $\Phi: Lc \rightarrow Lc$ を考える。 $\Phi(\sigma)$ が定義されていることを $\Phi(\sigma)$ で、また定義されていないことを $\Phi(\sigma)$ で表す。

こうした定数記号の対応を与える部分写像 Φ は論理式の変換を定める構文的な準同型写像 Φ' に拡張できる。すなわち、

$$\begin{aligned} \Phi'(\sigma) &= \Phi(\sigma) && \text{if } \sigma \in Lc \\ \Phi'(X) &= X && \text{if } X \text{ は変数} \\ \Phi'(p(t_1, \dots, t_n)) &= p(\Phi(t_1), \dots, \Phi(t_n)) \\ &&& \text{if } p \in L - Lc \text{ かつ } t_j \text{ は項} \\ \Phi'(A \leftarrow B_1, \dots, B_n) &= \Phi(A) \leftarrow \Phi(B_1), \dots, \Phi(B_n) \end{aligned}$$

Φ' を簡単のために Φ 、また項または節Cに対し $\Phi(C)$ が定義されていることを $\Phi(C) \downarrow$ と記す。節の集合Xに対して $\Phi(X) = \{\Phi(C) \mid C \in X, \Phi(C) \downarrow\}$ と定める。前述したように、領域 $D = (P, N)$ における否定的事実 $\sim\beta$ は、 Φ による確定節の変換による“過度”の類推を防ぐためにのみ使用するので $\sim\beta \in N$ に対する変換 $\Phi(\sim\beta)$ を、ここでは定義する必要がないことを注意したい。無矛盾性により不適節な類比・類推を棄却する考え方は[Indurkha87]において述べられている。

さて、以上の準備の下にここで考察する類似性と類推原理を述べておく。

類似性: $D = (P, N)$ を領域として、Pの節Cに対し $\Phi(C) \downarrow$ かつ $\Phi(C) \in P$ であるとき、対 $\langle C, \Phi(C) \rangle$ を Φ の証拠(evidence)と呼ぶ。このとき Φ が与える類似性とは Φ の証拠からなる $P \times P$ の部分集合 $E(\Phi) = \{\langle C, \Phi(C) \rangle \mid C \in P, \Phi(C) \downarrow \text{ かつ } \Phi(C) \in P\}$ である。

$\Phi(C) \downarrow, \Phi(C) \in P$ のことを意味する。特に $E(\Phi)$ が空集合でない Φ を許容的写像という。

この写像 Φ による類似性の概念は、類比的理論[Haraguchi85a, 85b]という事実の有限集合の間の部分同一の考え方を、節の集合 (P の部分集合)

$$S1(\Phi) = \{C \in P \mid \Phi(C) \downarrow\}, S2(\Phi) = \{\Phi(C) \in P \mid C \in S1\}$$

の間の部分同一のそれに拡大解釈したものである。したがって、記号の書換えによる構文的な部分同一の概念になっている。但し、類比的理論では Φ の一対一性を要求していたがここでは特に仮定しない。

類推原理： Φ を許容的写像とする。このとき、 $\Phi(P)$ が D のデータと矛盾しないならば、すなわち、 $\Phi(P) \cup P \cup N$ が無矛盾なとき、 $\Phi(P)$ の節を P の新たな節として仮定し、 D の拡張領域 $D' = (\Phi(P) \cup P, N)$ での演繹推論を行う。特に、このような Φ を T 写像とぶ。

一般に $\Phi(C) \downarrow$ かつ $\Phi(C) \in P$ なる P の節 C に対し、 $P \vdash \Phi(C)$ なこともうる。こうした場合は、類推が新たな情報を P にもたらさないことを意味するが、 $\Phi(C)$ は論理的に正しい節なので、推論する際に有害なものとはならないことに注意したい。

類推のタイミングと推論問題： 与えられた領域 $D = (P, N)$ に対する類推は D での演繹推論が有限失敗した時、類推を試みるとする。すなわち、 L のゴール節 $\leftarrow B1, \dots, Bn$ に対し、 $P \vdash \exists B1 \wedge \dots \wedge Bn$ でないことが判明したとき $\Phi(P) \cup P \vdash \exists B1 \wedge \dots \wedge Bn$ なる T 写像 Φ を求める。ただし、 $\exists B1 \wedge \dots \wedge Bn$ は $B1 \wedge \dots \wedge Bn$ に出現する全ての自由変数を存在限定してできる論理式を表す。

この推論問題の設定の仕方は、類推和の理論と全く同じであり、演繹推論に失敗したときに、類推により新たな情報を取り込むことで推論を続行させるとの考え方に基づく。

4.3.5.2 類推と不完全情報の推論

T -射像 Φ に基づく推論は、不完全情報の推論を行っているとも考えることも可能である。ここで情報が不完全であるとは、ある完全な情報が仮定でき、その情報の一部が欠落したために不完全となったことを意味している。また不完全情報の推論問題とは、完全な情報の一部である欠落した情報を、残りの（欠落しなかった）情報から推論する問題である。こうした不完全情報の推論を行う一つの方法として、例えばデータ従属性などの制約を仮定し、演繹的に欠落情報を復旧することも挙げることができる[Miyano82]。しかしながら、制約のもとに演繹推論で結論できる情報は、現在の情報から必然的に導かれるという意味でそもそも欠落した情報と言いがたい。

一方、類推による不完全情報の推論とは、類似性を与える対応関係が現在の情報と矛盾がない限り、対応関係に基づく情報の変換を行い（演繹的には導かれない）情報を推論・推定する。

前節において類推の対象 D を確定節の集合 P と否定的事実の集合 N の組で与えた。本節

では、不完全性を与える欠落した情報を、個別的な事実を与える P の肯定的事実 に制限して考えるために、 P を (確定節で書かれた) 制約の集合 C と肯定的事実の集合 F の和集合として考える。すなわち、類推の対象領域 D とは $D = (C, F, N)$ とする。ただし、 $C \cup F \cup N$ は無矛盾であることを仮定する。また 2 節では C の節も $\Phi: L_c \rightarrow L_c$ によって変換されるとしたが、この節に限り、 C の節は変換の対象とはならず、 F の事実の変換 $\subseteq$ によって $\Phi(F) \cup F \cup C \cup N$ が無矛盾となる T 写像 Φ を求める問題を考える。例として次のような領域を考えてみよう。

C : $X = X$,
 $p(Y_1, \dots, Y_n) \leftarrow p(X_1, \dots, X_n), X_1=Y_1, \dots, X_n=Y_n$
for each n -ary predicate symbol p
 $Y = Z \leftarrow \text{nationality}(X, Y), \text{nationality}(X, Z)$
 $Y = Z \leftarrow \text{born_at}(X, Y), \text{born_at}(X, Z)$
 $\text{person}(X) \leftarrow \text{speak}(X, Y), \text{language}(Y) \leftarrow \text{speak}(X, Y)$
 $\text{nation}(X) \leftarrow \text{nationality}(Y, X), \text{person}(X) \leftarrow \text{nationality}(X, Y)$
 $\text{person}(X) \leftarrow \text{born_at}(X, Y), \text{nation}(X) \leftarrow \text{born_at}(Y, Z)$
 $\text{nation}(X) \leftarrow \text{n_language}(X, Y), \text{language}(Y) \leftarrow \text{n_language}(X, Y)$
 $Y = \text{japanese} \leftarrow \text{n_language}(\text{japan}, Y)$
 N : $\sigma \neq \tau$ for each distinct constant symbols σ, τ in L_c
 $\sim \text{nation}(\sigma), \sim \text{language}(\sigma)$ for $\sigma = a, b, c$.
 $\sim \text{person}(\sigma), \sim \text{nation}(\sigma)$ for $\sigma = \text{english}, \text{japanese}, \text{french}$
 $\sim \text{language}(\sigma), \sim \text{person}(\sigma)$ for $\sigma = \text{united_kingdom}, \text{japan}, \text{france}$
 F : $\text{speak}(a, \text{english}), \text{nationality}(a, \text{united_kingdom}),$
 $\text{born_at}(b, \text{japan}), \text{native_language}(\text{english}, \text{united_kingdom})$
 $\text{nationality}(b, \text{japan}), \text{born_at}(a, \text{united_kingdom})$
 $\text{speak}(c, \text{french}), \text{n_language}(\text{japan}, \text{japanese}),$
 $\text{nationality}(c, \text{france}), \text{n_language}(\text{france}, \text{french})$

ここで、 $C \cup N$ の一部は等式理論であり、また C の nationality と born_at に関する制約式は、 nationality と born_at が関数関係であることを制約している。この例では関数従属のような制約は仮定されていないが、一般にはそうした制約が存在してもよい。 F は現在成立することが判っている事実の集合である。また $\text{language}, \text{person}, \text{nation}$ に関する C と N の節は定数記号の型を定めている。

この現在与えられた領域 D に対し、 $C \cup F \models \exists \text{ speak}(b, X)$ である。この変数 X の値を推論するために、次の仮定をおく：

仮定：『ある L_c 上の領域 $D' = (C, F', N)$ と L_c の要素 σ で
 $C \cup F' \models \text{ speak}(b, \sigma)$ かつ $F \subseteq F'$ なるものが存在する』。

すると類推による不完全情報の推論とはこの場合で $C \cup \Phi(F) \cup F \vdash \text{ speak}(b, \sigma)$ なる T 写像 Φ と定数記号 $\sigma \in L_c$ を求める問題である。すなわち、欠落した $F' - F$ のデータを $F' - F \cup \Phi(F)$ なる T 写像 Φ で補うことに他ならない。

ではどのような Φ がこの場合考えられるだろうか？ まず定数記号の型に関する制約と否定的事実より、型を保存しない写像 Φ は現在のデータと矛盾することから考察の対象外である。次に $\text{ speak}(b, X)$ なる X の値を直接的に F のデータから得るために、例えば

$$\Phi 1 = \{ (c \rightarrow b), (\text{french} \rightarrow \text{english}) \}$$

なる写像を考えることが可能だが、これは証拠を持たないので T 写像にはならない。証拠を $\Phi 1$ に持たせるために拡張した

$$\Phi 2 = \{ (c \rightarrow b), (\text{french} \rightarrow \text{english}), (\text{france} \rightarrow \text{japan}) \}$$

は $n_language(\text{japanese}, \text{japan})$ と矛盾するデータ $n_language(\text{english}, \text{japan})$ を導くので T 写像ではない。すなわち $\Phi(F) \cup C \cup N$ が矛盾であるものは妥当性に欠ける許容的写像として棄却される。一方、

$$\Phi 3 = \{ (a \rightarrow b), (\text{uk} \rightarrow \text{japan}), (\text{english} \rightarrow \text{japanese}) \},$$

$$\Phi 4 = \{ (c \rightarrow b), (\text{france} \rightarrow \text{japan}), (\text{french} \rightarrow \text{japanese}) \}$$

は共に T 写像であり、 D からは推論することが不可能な $\text{ speak}(b, \text{japanese})$ が推論される。

このようにデータベースは単に事実の集合ではなく、肯定的事実は互いにそれを記述する定数記号により関連付けられている。この例を見ても分かるように、T 写像を求めるプロセスはそうした事実の関連を定数記号を介して辿り、妥当なものを求めるプロセスである。得られた T 写像は少なくとも現在のデータと制約には矛盾しないという意味で、類推の妥当性基準を満たしていると言える。

4.3.5.3 推論手続き

この節では与えられた領域 $D = (P, N)$ と $P \vdash \exists B_1, \dots, B_n$ なるゴール節 $\leftarrow B_1, \dots, B_n$ に対し、 $\Phi(P) \cup P \vdash \exists B_1, \dots, B_n$ なる T 写像 Φ を求める手続きを与える。この手続きは類推和とよく似た性質を持っている。すなわち、SLD 導出の拡張であり、ある T 写像 Φ を事前に求めて次に $\Phi(P) \cup P$ からの演繹推論を行うのではなく、 Φ を求めながら同時に $\Phi(P) \cup P$ からの演繹推論も行う。また類推和の理論では、1 対 1 の条件、すなわち、EPIC による（1 対 1 性の意味での）無矛盾性の検証を行う推論になっていたが、ここで与える手続きにおける無矛盾性とは、以前の節で論じてきたように、 Φ による変換で得られた節が D の節と矛盾しないことを意味している。ある時点でのゴール節に対し、どの P の節を Φ により変換するかは本質的に非決定的な問題なので非決定的な手続きとして与える。

手続きを記述する前に Φ を求めながら同時に導出を行う“T-導出”の概念を示しておこう。与えられたゴール節 $\text{Goal}: \leftarrow B_1, \dots, B_n$ と P の節 $C: A \leftarrow D_1, \dots, D_m$ に対して T-導出とは、

- (1) 通常のSLD導出か、または
 (2) $A \leftarrow D_1, \dots, D_m$ に Φ を適用して得られる節とのSLD導出
 のことを指すとする。ここで (1) は明かであるから、(2) を正確に述べよう。

$\sqsubseteq$ を L_c の定数に L_c の定数または変数を対応させる部分写像 (簡単のために v -写像と呼ぶ), $\text{dom}(\Phi)$ を Φ の定義域とする。また θ を代入とする。このとき節 $A \leftarrow D_1, \dots, D_m$ の定数記号で $\text{dom}(\Phi)$ に属さない σ に対し、それぞれ新たな変数 $V[\sigma]$ を導入し

$$\Phi_0 = \Phi \cup \{(\sigma \mapsto V[\sigma]) \mid \sigma \in \text{dom}(\Phi), \sigma \text{ appears in } C\}$$

なる v -写像を考える。このとき、上記の (2) で言う導出とは、 $\leftarrow B_1, \dots, B_n$ と $\Phi_0 (A \leftarrow D_1, \dots, D_m)$ とのSLD導出を指す。導出が可能な場合は、 B_1 と $\Phi_0 (A)$ の mgu を τ として、 T -導出は 次のゴール節

$$\text{Goal}_1: (\Phi_0(D_1), \dots, \Phi_0(D_m), B_2, \dots, B_n) \tau,$$

次の合成 $\theta_1 = \theta \tau$, 次の v -写像 $\Phi_1 = \Phi_0 \tau$ を作りだす。

このような T -導出を行う手続きを $\text{resolve}(C, \text{Goal}, \theta, \Phi, \text{Goal}_1, \theta_1, \Phi_1)$ とする。ここで T -導出が通常のSLD導出の場合は v -写像は変化せず、したがって、 Φ と同じ Φ_1 を resolve は返す。

推論手続き $\text{reason}(\text{Goal})$ は入力ゴール節 Goal に対し、 T -導出によって空節を導出することを試みる。以下 ε, ϕ は、それぞれ空代入、空集合を表し、 reason に呼び出される手続き refute はゴール節 Goal , 代入 θ , v -写像 Φ , および節の集合 S を入力として受け取る。節集合 S は反駁において使用された P の節の集まりである。

```

procedure reason(Goal)
begin
  call refute(Goal,  $\varepsilon$ , [],  $\phi$ )
end

procedure refute(Goal,  $\theta$ ,  $\Phi$ ,  $S$ )
begin
  if Goal is empty then call check( $\theta$ ,  $\Phi$ ,  $P-S$ );
  choose  $C \in P$ ;
  call resolve( $C$ ,  $\text{Goal}$ ,  $\theta$ ,  $\Phi$ ,  $\text{Goal}_1$ ,  $\theta_1$ ,  $\Phi_1$ );
  call refute( $\text{Goal}_1$ ,  $\theta_1$ ,  $\Phi_1$ ,  $S \cup \{C\}$ );
end
  
```

上記の手続きの中の check は v -写像の拡張と無矛盾性の検証を行う重要なものだが、その説明は後で行う。

check を呼び出すまでの過程を例示するために、次のような領域 $D = (P, N)$ と初期ゴール節 $\leftarrow \text{speck}(b, X)$ を考える。

```

P : nation(y).    nation(uk).
    nat(X, uk) ← born(X, uk).
    speak(X, eng) ← nat(X, uk).
    nat(X, y) ← born(X, y)
    ed(X) ← born(X, y). speak(X, eng)
    off(eng, y).    n_language(y, lang, y).    born(b, y).    nat(b, y)
    off(eng, uk).    n_language(eng, uk).

N : ~edu(b)

```

但し、nat, born, ed, off, n_language, eng, uk, ylang はそれぞれ、nationality, born_at, educated, official_language, native_language, english, united_kingdom, language_spoken_in_country_y の略記である。

まず $C0 : \text{speak}(X, \text{eng}) \leftarrow \text{nat}(X, \text{uk})$ を選択して T-導出を行う。eng, uk にそれぞれ新たな変数 $V[\text{eng}]$, $V[\text{uk}]$ を対応させて節 $C1 : \text{speak}(X, V[\text{eng}]) \leftarrow \text{nat}(X, V[\text{uk}])$ を得る。

$$\frac{C0 : \quad \text{speak}(X, \text{eng}) \leftarrow \text{nat}(X, \text{uk})}{C1 : \quad \text{speak}(X, V[\text{eng}]) \leftarrow \text{nat}(X, V[\text{uk}])}$$

このとき用いた v -写像 Φ は、集合表現して $\{(eng \rightarrow V[\text{eng}]), (uk \rightarrow V[\text{uk}])\}$ である。C1 とゴール節 $\leftarrow \text{speak}(b, X)$ の SLD 導出行い、次のゴール節 $\leftarrow \text{nat}(b, V[\text{uk}])$ を得る。これと P の事実 $\text{nat}(b, y)$ の SLD 導出を行って、空節が導き出される。解代入 θ は $\theta = \{X := V[\text{eng}], V[\text{uk}] := y\}$ 、また v -写像 Φ は $\{(eng \rightarrow V[\text{eng}]), (uk \rightarrow y)\}$ となることに注意したい。

こうして空節が導出され、得られた v -写像 Φ が変数を含まないときは、 Φ が証拠をもつかどうかのチェックを行い推論を終了する。 Φ が変数を含むときは、その変数の値を求めるために Φ をさらに拡張することを試みる。この過程を先の例で示そう。得られた Φ は変数 $V[\text{eng}]$ を含むので、その値を P が持つ情報から引き出すことにより拡張を行う。すなわち、P の節で eng を定数記号として含む C を考え、それを P の他の節 D に対応させることで $V[\text{eng}]$ の値を推論する。推論が成功終了した時点で $\langle C, D \rangle$ は Φ の一つの証拠となることに注意したい。この場合、 $C = \text{off}(eng, uk)$ を Φ で写像してできた節 $\text{off}(V[\text{eng}], y)$ とマッチする P の節 D の存在を問う。一般には $\Phi \sigma \in \text{dom}(\Phi)$ かつ σ が C に出現するような定数記号が存在しうるので、T-導出のときと全く同様にして、新たな変数を導入しさらに v -写像 Φ を拡張する必要がある。 $\Phi(C)$ と D のマッチングは、単に、 $\Phi(C) \tau \in P - \{C\}$ なる代入 τ を求めることにより行う。この例の場合 $\tau = \{V[\text{eng}] := eng\}$ によって 新たな $\Phi' = \{(eng \rightarrow eng), (uk \rightarrow y)\}$ を得る。

$$\Phi'(\text{off}(eng, uk)) = \text{off}(eng, y) \in P$$

結局、初期ゴール $\leftarrow \text{speak}(b, X)$ から空節を導出するために用いた C1 は

$$\Phi'0(C0) : [\text{ speak}(X, \text{eng}) \leftarrow \text{nat}(X, y)] \in P$$

であることが判明する。しかるに

$$\{\Phi'0(C0)\} \cup P \vdash \text{ed}(b)$$

となり、否定的事実 $\sim \text{ed}(b) \in N$ と矛盾する。したがってこの $\Phi'0$ は棄却される。この例の場合は $\Phi 0$ に対して、eng を含む別の節 $\text{nl}(\text{eng}, \text{uk})$ に対する $\text{nl}(V[\text{eng}], \text{uk})$ と $\text{nl}(\text{ylang}, y)$ をマッチさせて、矛盾をおこさない T 写像

$$\{ (\text{eng} \rightarrow \text{ylang}, (\text{uk} \rightarrow y)) \}$$

を得ることができる。T-反駁の解代入 $\theta = \{ X := V[\text{eng}], V[\text{uk}] := y \}$ とこのマッチングによって得られた代入 $V[\text{eng}] := \text{ylang}$ を合成させて、初期ゴールに対する最終的な解答 $\text{speak}(b, \text{ylang})$ を得る。以下の手続きのうち、extend が上記の T-反駁で得られた v-写像から目的の T 写像を求める手続きとなっている。

procedure check(θ, Φ, S)

begin

if $\subseteq$ contains no variables then call evidence(Φ)

else call extend(θ, Φ, S)

end

procedure extend(θ, Φ, S)

begin

if $\subseteq$ contains no variables

then call consistent(Φ, θ);

choose $(\sigma \rightarrow \text{Var}) \in \subseteq \Phi$, where Var is a variable in Φ ;

choose $C \in P$ containing σ as its constant symbol;

while there exists constant symbol τ in C such that $\Phi(\tau) \downarrow$

introduce new variable $V[\tau]$ and let $\Phi = \Phi \cup \{(\tau \rightarrow V[\tau])\}$;

find a substitution τ such that

$$\Phi(C)\tau \in S - \{C\};$$

call extend($\theta\tau, \Phi\tau, S - \{C\}$)

end

procedure consistent(Φ, θ)

begin

check if there exists some $\sim \beta \in N$ such that

$$\Phi(P) \cup P \vdash \beta;$$

if such a negative fact β exists then fails

else return(θ)

end

4.3.6 おわりに

類推とはそもそも多様なものであり、推論の目的や意図によってその定義自体が変化するべきものであろう。すると期すべき類推システムとはその目的と意図の変化に柔軟に対応できることが要求される。類推の目的と意図は類推の推論としての定義に拘るだけでなく、それを背景知識と結合させることにより、推論の実際的な制御を可能にする機能を与えることにもなるだろう。実際、類似性の概念それ自体が組合せ的なものであり、したがってある種の意味情報や制約条件を背景知識として使わざるを得ないことは、推論の妥当性と計算量の両方の観点から必要であろう。

特に最後の節では類推におけるソース領域の選択問題を解決するために、類推の再定式化を行い、これを不完全情報の推論として用いる一つの可能性を示した。最後の節で示した推論手続きは、類推和の理論と同様に、可能な類比の対応付けを求めながら反駁を行い、その後で対応付けの拡張と無矛盾性の検証を行うものであった。基本的には許容写像を枚举し、次にその無矛盾性を検証する生成テスト法の形をしている。したがってこのままでは大量の肯定的事実を持つデータベースに対しては、計算量の爆発をまねいてしまうが、T-導出で得られた部分写像の証拠を求める過程において、段階的に無矛盾性を検証し、たその環境を保存することにより、ある程度の効率化は可能であると考えている。

参考文献

- [Gentner82] D.Gentner(1982): Are Scientific Analogies Mataphors?, in Metaphor: Problems and Perspectives (D.S.Miall,Ed.), The Harvester Press, 106-132.
- [Haraguchi85a] Haraguchi,M.(1985) : Towards a Mathematical Theory of Analogy, Bull. Inform. Cybernetics, Vol.21, 29-56.
- [Haraguchi85b] 原口 誠 (1985): 類推の機械化, 創造性研究 No.3, 247-270.
- [Haraguchi86a] Haraguchi,M. and Arikawa,S. (1986): A Formulation of Analogical Reasoning and its Realization, J. of JSAI, 1, 1, 132-139.
- [Haraguchi86b] Haraguchi,M. and Arikawa,S.(1986): A Foundation of Reasoning by Analogy -Analogical Union of Logic Programs, Proc. LPC'86, 103-109.
- [Haraguchi87] Haraguchi,M. and Arikawa,S: Reasoning by Analogy as a Partial Identity between Models, Proc.AII'86, Springer-LNCS 265, 61-87.
- [Indurkha87] Indurkha, B. (1986): Approximate Semantic Transference: A Computational Theory of Metaphor and Analogies, Cognitive Science 11,445-480.
- [Miyano82] Miyano, S. and Haraguchi, M. (1982): Recovery of Incomplete Tables under Functional Dependencies, Bull. Informatics and Cybernetics, 20, 25-41.
- [Lenat86] Lenat,D. et.al. (1986) : CYC: Using Common Sense Knowledge to Overcome Brittleness and Knowledge Acquisition Bottlenecks, AI Magazine, 6, 4, 65-85.
- [Russell86] Russell,S.J. (1986): Analogical and Inductive Reasoning, STAN-CS-87-

1150, Dept. Computer Science, Stanford Univ.

[Russell87] Russell, S. J. (1987): Analogy and Single-Instance Generalization, Proc. 4th International Workshop on Machine Learning, 390-397.

[Winston80] Winston, P. H. (1980): Learning and Reasoning by Analogy, CACM, Vol. 23, 689-703.

[Winston83] Winston, P. H. (1983): Learning New Principles from Precedents and Exercises, AI, Vol. 19, 321-350.

4. 4 事例ベースの整理と一般化

新日本製鐵（株） エレクトロニクス研究所 久保真哉

要旨

事例ベース推論（CBR）が有効に使われるかどうかの鍵の一つは事例ベースの整理・一般化にある。ここではまず、事例がどういう場面で必要とされるかを、問題解決の種類に関連づけながら分析し、さらに事例ベースの整理・一般化の種類を分類した上で、CBRの重要な要素技術である事例ベースの整理・一般化に必要な手法として、統計的手法、構造化手法、機械学習、ニューラルネットの4つに大別して、各々の具体的機能について整理、考察する。最後に、問題解決の種類と、事例ベースの整理・一般化の手法との関連等について述べる。

4.4.1 はじめに

ESでは、問題や分野の型に応じた知識表現や問題解決法を用いることが、有効なシステム構築のために必要だとされている。同様に、CBRにもいくつかの型があると考えられ、それにふさわしい事例の特徴付けや検索や推論を行なうことが、より適切なCBRシステムを容易に開発することにつながる。

そのためには、まず、どういうふうに事例が利用されているかを分析する必要がある。そして、それぞれの場合にふさわしい事例の構造化、参照・検索の方法を検討しなければならない。

そこで、今回は、①事例の参照・検索がどういうタイミングで、どの部分を対象として行なわれているか（4.4.2）、また、②“事例の整理・一般化”の種類（4.4.3）と、③それを行なう（支援する）機能（4.4.4）について検討した。

上記の3つ、すなわち、参照の対象と、一般化の種類、それを実現する機能、との関係の検討は、今回の検討の更なる分析以後の重要な課題である。

4.4.2 事例参照の対象となる部分

問題解決するときに、どういう場面で事例が必要とされるか、つまり、いつ事例が参照されているかを、問題記述の部分に着目して考察した。

その結果、分析や合成などの問題解決の種類とも関連のあることが分かった。事例検索の場面を以下に示すような4つの場面に分類し、各々に関連する問題解決の種類と具体例を挙げる。

<状況>

問題が提示された状態を検索の対象とする。

「分析問題」

- 高炉E S （溶鉱炉の状態）
- スケジューリング （ex.過密時間帯という状況での運行スケジュールの例は？）
- 株式予測
- 異常の伝播の予測
- 故障診断 （症状）

<制約>

問題解決の際の制約条件を検索の対象とする。

「計画問題」

- 品質設計 （ex.この制約条件を満足するためにとった措置は何だったかな？）
- スケジューリング

<仕様>

問題の解の概略を検索の対象とする。

「設計問題」

- 品質設計 （ex.こういう強度の鋼材を製造した例は？）
- 機械設計

<問題の解決過程>

問題の解決過程（行為列、考え方）そのものを検索の対象とする。

「合成問題」

- 高炉E S （アクションを仮定して検索し、将来の高炉の状態を予想する）
- 方程式の解法 （行為列の部分マッチングの検索による次の行為の推定、予測）
（サブゴールを仮定して、サブゴールまでの過程を検索して求める）

4.4.3 事例ベースの整理・一般化の種類

事例が、項目（属性）と値（属性値）で表現できるとすれば、事例の整理・一般化は次に示すような種類が考えられる。ただし、“属性”は、事例の特徴の表現の場合も、事例の部分構造（例えば、自動車とエンジンの関係）の場合もあるが、現段階では厳密な区別はしない。

<事例の分類>

- 事例集合が形成する空間はその属性により、いくつかの小空間に分類できる。

<一般化>

- 事例そのものの一般化
 - 複数の具体事例から共通項目を選び、新たな一般化事例を作成（概念形成）する。
（ex. {人、犬、鯨} → ほ乳類）
 - ある特定の項目に着目して、一般化事例を作成する。
（ex. {計算機、蛍光灯、電車} → 電気を必要とするもの）
（特定の項目の選び方は、例えば直交成分や、主要部品などが考えられる。）
- 項目の一般化による間接的な事例の一般化
 - 特徴等の項目を一般化表現する。
（ex. ハードディスク容量 → 外部記憶容量）
 - 項目（事例の特徴）の解釈や値を変化（拡大、縮小、限定）させる。
（ex. 法律問題における解釈の変形、設計問題におけるパラメータの変化など）
- 事例も項目も一般化
 - 項目の一般化により、事例間の共通項目を増やし、同時に一般化事例を作成する。

<典型例（典型度）の抽出>

- 事例集合の空間の中心（典型度が高い）にあるものを典型例（代表例）とする。
- 正事例の典型例と負事例の典型例が考えられる。
- 各事例と典型例との相違を明確にして、例外処理、修正などの知識（ルール）を得る。
（ex. 部品Aの大きさがこの程度違っているから、部品Bの形がこうなる）

4.4.4 事例の整理・一般化の方法

次に、このような事例の一般化等を有効に行うために必要な手法や機能について考察する。

現在、私の所属する研究グループでは、知識獲得に使用するために、各種のツールや機能を開発中である。また同時に、ニューラルネットやE Sシェルの研究・開発も行なっている。そこで、これらの機能がCBRにおける事例ベースの整理・一般化にどう利用できるかに主眼をおいて検討する。

事例ベースの整理・一般化の手法は、次の4つ、すなわち統計的手法、構造化手法、機械学習、ニューラルネットに整理することができる。以下に、各々についての具体的な手法と、それを使うことによって実現できるCBRの部分機能を列挙する。

<統計的手法>

「主座標分析」

- 重要項目の選定、それを基準にした一般化
- 典型例の抽出

「多次元尺度構成法」

- インデックス間の尺度構成
 - 統一的な尺度表現 → 既存の統計解析法の利用
- 事例間の尺度構成

＜構造化手法＞

「クラスター分析[Anderberg 73]」

(分類対象の属性を基に自動的に構造化分類する)

- 事例の分類（事例空間の作成）
- 事例の一般化
- 欠落した項目（インデックス）の発見の促進
- 事例間の距離の算出

「AHP (Analytic Hierarchy Process) [Saaty 80]」

(競合する案件を比較するための評価基準の定量的重みづけを、1対比較を基に行う)

- 状況に応じた事例検索
- インデックスの重み付け

「ISM (Interpretive Structural Modeling) [赤木 88]」

(2つの概念間の関係記述をもとに構造化された概念を順次形成する方法)

- AHPの階層構造の決定支援
- 領域の因果構造知識の整理、グループ化
- 領域の因果構造知識の表示による知識の修正支援
- 問題解決過程の追跡や階層化、チャंक

「IDS [Michalski 83]」

(いくつかの正、負の事例から、正、負を分類する最適な木構造を生成する)

- 分類（正例、負例）ルールの抽出

「KJ法」

(ブレンストーミングの1手法。思いついた断片知識を順次まとめながら関連づけてゆく)

- 隠されたインデックスの発見

- インデックスの整理

「ルール表現、フレーム表現」

- 領域知識の表現
- 問題対象の表現

<機械学習>

「SBL (Similarity-Based Learning) [Mitchell 81]」

- 類推
- 事例の特徴抽出

「EBL (Explanation-Based Learning) [Dejong 86]」

- 事例の階層構造化
- 問題解決過程の一般化

<ニューラルネット>

「学習機能」

- イメージデータ、パターンデータの判別、類推

「連想記憶」

- シソーラスの検索

4.4.5 終わりに

最後に、今回の考察で気づいたいくつかの知見、及び今後の課題を以下に述べる。

- 問題解決の種類、事例参照の対象、事例ベースの整理・一般化の種類、整理・一般化の手法の間にある関係を明確にできれば、それを利用した有効なCBRシステムが開発できる。問題解決の種類と事例の整理・一般化の手法には図4.4-1に示すような大まかな関係があると考えられる。
- 既存の一般化（構造化）された知識の利用方法の検討や、自動的な構造化手法であるクラスター分析等の手法とCBRシステムとを結合させる方法の検討が必要である。
- 問題解決過程の分析では人間の行為の目的を明白にすることが大切である。
- インデックスの生成や変形には（知的な）用語辞書が不可欠である。

- 一般化の逆の具体化（特殊化）も重要である。

つまり、一般化された事例によって得られた問題解決の方法や回答は、あくまでも基本的なものであるから、個々の問題に適合するように具体化しなければならない。それには、どう具体化するかでいくつかの選択や発案が必要になろう。具体事例の違いを説明することによって、選択基準や発案の知識が得られるかも知れない。

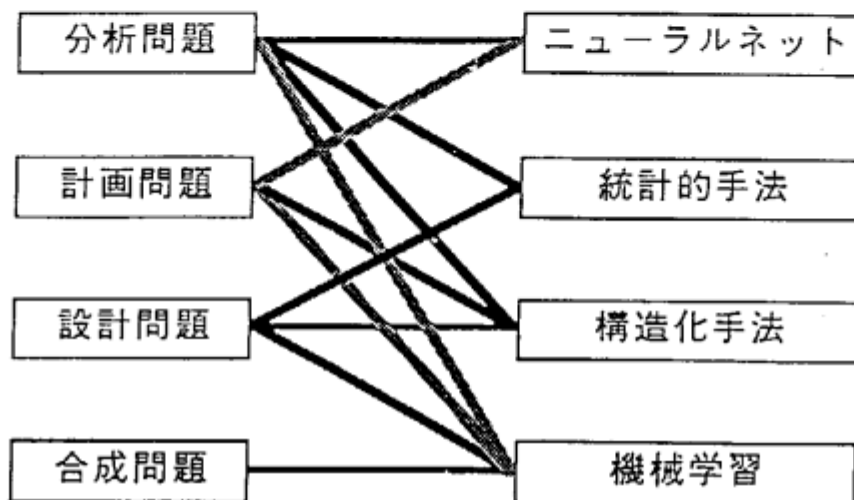


図4.4-1 問題解決の種類と事例の整理・一般化の手法

〔参考文献〕

- [Anderberg 73] M.R. Anderberg : CLUSTER ANALYSIS FOR APPLICATIONS, ACADEMIC PRESS, New York(1973)
- [DeJong 86] G. DeJong, R. Mooney : Explanation-Based Learning: An Alternative View, Machine Learning, Vol.1, No.2, 145-176(1986)
- [Michalski 83] R.S. Michalski, J.G. Carbonell, T.M. Mitchell : MACHINE LEARNING Morgan Kaufmann, California(1983)
- [Mitchell 81] M.T. Mitchell : Generalization as Search, READINGS IN ARTIFICIAL INTELLIGENCE, Tioga pub., 517-542(1981)
- [Satty 80] T. L. Satty : The Analytic Hierarchy Process, McGraw-Hill, New York(1980)
- [赤木 88] 赤木 新介 : AI技術によるシステム設計論, コロナ, 東京(1988)

4.5 事例照合の心理学的尺度について---プロトタイプ理論---

京都大学工学部 榎木哲夫

4.5.1 事例ベース推論における事例の照合過程に見られる問題点

事例ベース推論は、問題解決のなされた過去の既知事例の変形・修正により新規事例の問題解決を図るための枠組みであり、従来のルールベースに代表されるような問題解決のための一般則を予め用意することが困難な ill-defined な領域において、知識システム構築のための有効かつ現実的な方法論といえる。本来、事例ベース推論システムにおいては、事例ベースに格納される事例は、それからマイナーな修正・変形によって未知の事例の問題解決に当てうる (adaptation phase) ような「典型的」な事例集合を想定することになるが、実際に事例ベースを構築する際には、このような典型的な事例のみを取捨選択して格納することは困難であり、「個々の経験の集積=事例ベース」として集積されているのが現状である。従って、このような無目的に集積された事例ベースの中から、新規事例における問題解決の各フェーズで、タイムリーに典型的な既知事例を照合・決定するためのメカニズムならびにそのための事例の組織化・構造化が事例ベース推論システムに要請されることになる。

現在までに開発の進められている事例ベース推論システムにおいては、このような事例照合のプロセスは、「典型的な事例=新規事例に最類似の事例」という構図に基づいてなされている。そこでの個々の事例は [属性-値] の特徴集合として表現されており、このような特徴集合間の属性の重複の大小により、既知事例 (INSold) - 新規事例 (INSnew) 間の照合を行なっている。通常、多用されている照合度の定量化は、以下に一般的に示すような Kahneman の類似性尺度、

$$\text{similarity}(\text{INSold}, \text{INSnew}) = af(\{(\text{INSnew} \cap \text{INSold})^a\}) - bf(\{(\text{INSold} - \text{INSnew})^b\}) \\ - cf(\{(\text{INSnew} - \text{INSold})^c\})$$

によってなされている。ここで、右辺第一項は、既知事例・新規事例の両者に共有されるような積集合としての特徴集合の要素数、第2・3項は、いずれか一方にしか存在しないような差集合としての特徴集合の要素数の関数として決まる量を意味しており、重み a,b,c のもとで、前者から後者を差し引いた値として2つの事例間の類似性を評価している。各属性毎に重みを付けて評価している場合も、本質的には上式の類似性評価に類するものである。

しかしながら、上式のような類似性評価に基づくかぎり、真の意味で「典型的」な既知事例を決定できるものかについて、以下のような問題点が考えられる。

まず第一に、類似性に基づく学習 (Similarity-Based Learning) の抱える問題点と共通の課題、すなわち、特徴集合として用意されている属性が、必ずしも事例を的確に本質を押さえて表現しているとは限らないことから、noisy な属性が特徴集合に混入する場合や、事例の持つ概念とは無関連な属性の偶然の一致によって、その評価は大きく歪められることになる。

第二に、通常我々が、典型的な事例を検索する場合には、「どのような概念のもとで典型的

か」といった視点設定が不可欠となる。従来の事例ベース推論システムでは、特にこのような視点設定がなされないまま、事例ベース全体の中から最類似のものを検索するものであるが、実際には、新規事例のおかれている問題解決の局面やニーズに呼応して、適宜、内包している視点を明らかにした上で、事例表現の中のどの属性が本質的であり、何が非本質的なものかを見極め、顕在化しうえで照合を図るメカニズム、いわば purpose-driven な検索が必要になる。

第三に、特徴集合を形成する属性の間に、因果性などの依存関係が存在する場合には、たった一つの属性値の変更が連鎖的に他の属性値の変更を引き起こすような場合、上式の類似性評価では、真の意味での最類似の事例を見逃す危険性がある。

本報告では、以上のような考察に基づき、まず第一、第二の問題点を克服するため、心理学分野における概念形成の理論であるプロトタイプ理論 [Rosch 76][Rosch 78]、ならびに演繹学習の手法として知られる説明に基づく学習手法 [Mitchell, et al., 86]を融合した手法を提案する。本手法の背景となる概念図を図4.5.1に示す。同図に示すように、本手法は、与えられた新規事例をどのような概念に対応する視点を設定するかによって、同一の事例に対する典型度評価が変わるものであり、参照すべき既知事例の決定は、従来のような最類似の事例から、設定された視点下での最も典型的な事例をもって行なうというものである。そのために、次節では、プロトタイプ理論についてサーベイし、次々節で本手法の詳細について例題とともに述べる。さらに、最終節では、上記第三の問題点を克服するための方法論として、定性推論分野における比較解析手法を紹介し、その適用可能性についての考察を行なう。

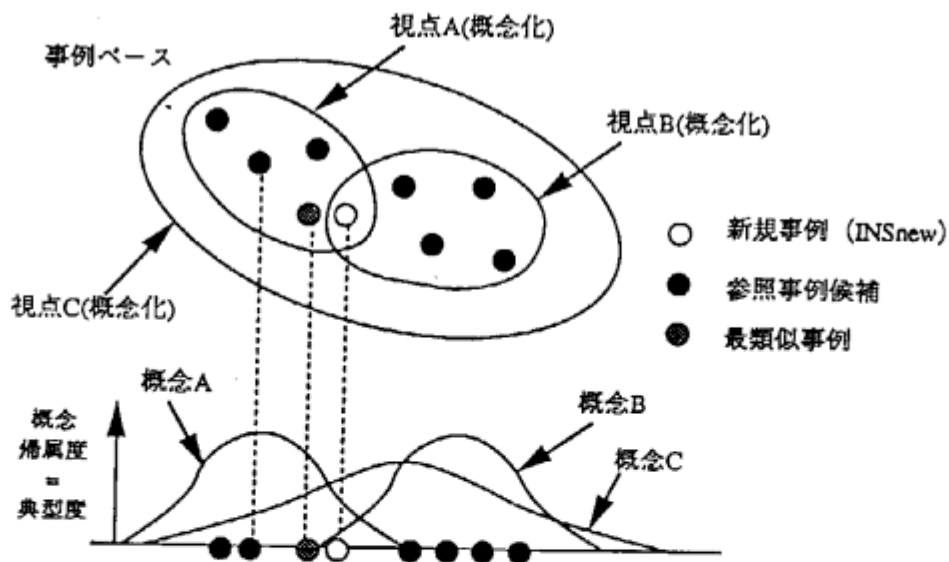


図4.5.1 視点のもとでの典型的事例の検索

4.5.2 プロトタイプ理論

Rosch の研究を契機とするプロトタイプ理論 (prototype theory) は、人間の概念・カテゴリー形成に関する一連の研究である。Rosch は、カテゴリーは、ある中心的な成員、すなわち「原型あるいは典型 (prototype)」を中心に構築されており、典型はそのカテゴリーの代表例であって、カテゴリーの成員とほとんどの特徴を共有する一方、カテゴリー外のものとはその特徴をほとんど共有することはない、としている。認知対象となる事例 (成員) と最も類似する典型を見いだすことにより、各成員のカテゴリーへの帰属度が決定されるというものである。Rosch は、カテゴリー構造の他の側面についても同じく注意を喚起しており、一般的な経験カテゴリーの多くにある水準があって、そこでは人が最も容易に名称を覚えたり、想起したりできることを見だし、それを基礎水準 (basic level) と呼んだ。この水準にある対象物はそれぞれ数多くの知覚的類似性や機能的特徴を共有しており、上位水準 (superordinate level) と呼ばれるより高い水準と対照をなし、また下位水準 (subordinate level) とも対比されるとしている。また、カテゴリー間の強固な、あるいは固定された境界の存在を主張した古典的な見解に疑問を投げ掛け、現実世界のカテゴリーは、曖昧な境界を持ち、互いに融合する傾向にあるとし、カテゴリーは基準的な定義特徴を持つものではなく、典型を内に含むものであるとしている。この典型こそが、我々がカテゴリーを概念化している際の中核概念となるべき表象 (あるいは内包) であり、この表象への類似性に基づいてその成員性が連続的に変化するという考え方は、従来、概念の外延的 (extensional) な捉え方、すなわち要素事例の集合に注目するファジィ集合論の考え方に対し、概念の持つ意味、特徴、属性に注目する内包的 (intensional) な捉え方として一線を画するものである [Sawaragi, et al., 89][Sawaragi, et al., 90]。

Osherson and Smith は、「ある基礎的な手持ちの概念から思考や複雑な概念を構成する能力が人間の精神活動の中心的位置を占める」という考え方に立って、ファジィ集合論、あるいはさらに広く外延的なアプローチに対して批判的な見解を表している [Osherson et al. 81]。彼らは、基本的概念から論理的な関係子 (AND・OR・NOT) や含意 (implication) によって定義づけられる複合的な概念のもとで事例のもつ概念帰属度は、ファジィ演算で定めるように、その構成要素概念のもとでの帰属度からの形式的な演算操作で算出評価したのでは、実際の判断と大きくずれてしまうことを例証している。例えば、観賞魚 (pet fish) といった複合概念は愛玩動物 (pet) と魚 (fish) という基本概念から構成されていると思われるが、外延的な捉え方では、観賞魚の集合は、愛玩動物の集合、魚の集合のいずれにも真に含まれるものとなり、従って、ある成員のこのような複合概念の下での帰属度は、各々の基本概念下での帰属度を上回ることはないことになる。しかしながら、実際、例えば「グッピー」のような成員を考えると、観賞魚のカテゴリー下での帰属度が各基本概念のもとでの値より大きく上回ることに、誰も異存を表さない筈である。類似のことは、修飾された概念のもとでの典型性評価の変化についても言える。今ここに、赤い色をしたりんご (apple1: 成員) と茶色い色をしたりんご (apple2: 成員) があるとする。これらの成員を、「赤いりんご」(

RED-APPLE: 概念)「茶色いりんご」(BROWN-APPLE: 概念) また修飾子のつかない「りんご」(APPLE: 概念) の概念のもとでの帰属度を比較してみることにする。まず apple1 についての典型性は、RED-APPLE > APPLE > BROWN-APPLE となる筈であり、apple2 については逆に、BROWN-APPLE > APPLE > RED-APPLE となる筈である。外延的に見ると、RED-APPLE、BROWN-APPLE のいずれも APPLE に真に含まれる集合となることから、矛盾をきたすことになる。

Smith らは、上記のような現象を説明するためのモデルとして、選択的修正モデル (selective modification model) を提案している [Smith et al., 88]。その意味するところは、複合的な概念が与えられた時点で、成員とは無関係に、表象レベルで基本概念からの概念の再構成・再編成がなされ、新たにその複合概念の原型が造りだされて、各成員の典型性評価は、この原型との類似性評価として行なわれるという考え方である。ここでの事例 (成員) 表現は、図4.5.2 に示すように、属性ならびにその値 (確かさの度合いに応じて重み付けされている) のペアとして表現されている。一方、概念の原型 (プロトタイプ) は同様に、属性 (各々の重要度が重み付けされている) ならびにその値 (複数候補が典型性の大小に応じて重み付けされている) として表現されており、各事例と原型との間の類似性評価は、前節で述べた Kahnemann の類似性尺度に従って算出している。ここでの、さまざまな数値情報は定義や決め方など、不明な点が多すぎるものであるが、重要なのは、事例相互の間の類似性を算出するのではなく、概念の表象としての原型と事例との間の類似性を算出している点である。いまここで、「赤いりんご」という修飾された概念を考える。選択的修正モデルにおいては、このような修飾子「赤い」は、もとの「りんご」の概念の原型に対して、図4.5.3 に示すような2つの作用をもたらすとしている。一つは、特定の属性を選択し、その重要度を高める作用、いま一つは、その属性の値の典型度を高め、背反する値のそれを低めるという作用である。このような変形修正された原型に対して、前述の類似性尺度に従って照合が図られることになる (図4.5.4)。

以上のような概念の再編成は、画一的に手続き化できるものではなく、どのような概念と概念が結びつくかにより、各々のもつ固有の意味に依存して、ある特徴のみの実在性が顕在化されたあらたな表象が生成されるものである。そして、このような表象間の干渉を定めるのは、とりもなおさず評価者のもつ領域に固有な背景知識であり、課題固有の内容が、換言すれば各カテゴリーがもつ固有の構造が、評価者の分類ストラテジーを規定しているという意味において、表層的な特徴集合のみによる類似性判断に警鐘を与えるものといえる。

4.5.3 概念化プロセスにおける専門的知識の重要性

概念の中には、大まかな判断の基準として役立つようなあいまいな表象もあれば、明確な定義が要求されるときに用いられる表象もある。しかし、必ずしも、専門家の持つ表象が後者で、素人の持つ表象が前者であるとは限らない。むしろ逆転している場合が多い。後者の場合、当然のことながら、専門家が概念的カテゴリーを表象するやり方は、何を知っている

プロトタイプ Apple (A)	事例 1 I_1	事例 2 I_2
1 color { red 25 green 5 brown =	color { red 30 green brown =	color { red green brown 30 =
0.50 shape { round 15 square cylindrical 5 =	shape { round 20 square cylindrical =	shape { round 20 square cylindrical =
0.25 texture { smooth 25 rough 5 bumpy =	texture { smooth 30 rough bumpy =	texture { smooth 30 rough bumpy =
	$\begin{aligned} \text{Sim}(A, I_1) &= 1(25-5-5) \\ &\quad + 0.50(15-5-5) \\ &\quad + 0.25(25-5-5) \\ &= 15 + 2.5 + 3.75 \\ &= 21 \end{aligned}$	$\begin{aligned} \text{Sim}(A, I_2) &= 1(0-30-30) \\ &\quad + 0.50(15-5-5) \\ &\quad + 0.25(25-5-5) \\ &= -60 + 2.50 + 3.75 \\ &= -54 \end{aligned}$

図4.5.2 特徴リストによる原型・事例表現

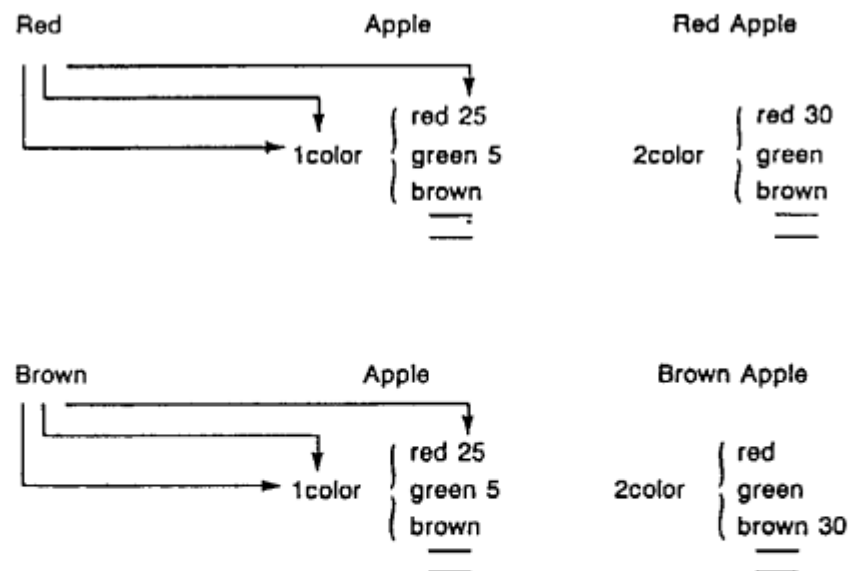


図4.5.3 Selective modification model

Red Apple (RA)		I_1	I_2
2 color	$\begin{cases} \text{red 30} \\ \text{green} \\ \text{brown} \\ \hline \hline \end{cases}$	color $\begin{cases} \text{red 30} \\ \text{green} \\ \text{brown} \\ \hline \hline \end{cases}$	color $\begin{cases} \text{red} \\ \text{green} \\ \text{brown 30} \\ \hline \hline \end{cases}$
0.50 shape	$\begin{cases} \text{round 15} \\ \text{square} \\ \text{cylindrical 5} \\ \hline \hline \end{cases}$	shape $\begin{cases} \text{round 20} \\ \text{square} \\ \text{cylindrical} \\ \hline \hline \end{cases}$	shape $\begin{cases} \text{round 20} \\ \text{square} \\ \text{cylindrical} \\ \hline \hline \end{cases}$
0.25 texture	$\begin{cases} \text{smooth 25} \\ \text{rough 5} \\ \text{bumpy} \\ \hline \hline \end{cases}$	texture $\begin{cases} \text{smooth 30} \\ \text{rough} \\ \text{bumpy} \\ \hline \hline \end{cases}$	texture $\begin{cases} \text{smooth 30} \\ \text{rough} \\ \text{bumpy} \\ \hline \hline \end{cases}$
		$\begin{aligned} \text{Sim (RA, } I_1) &= 2 (30-0-0) \\ &+ 0.50 (15-5-5) \\ &+ 2.25 (25-5-5) \\ &= 60 + 2.50 + 3.75 \\ &= 66 \end{aligned}$	$\begin{aligned} \text{Sim (RA, } I_2) &= 2 (0-30-30) \\ &+ 0.50 (15-5-5) \\ &+ 0.25 (25-5-5) \\ &= -120 + 2.50 + 3.75 \\ &= -114 \end{aligned}$

Brown Apple (BA)		I_1	I_2
2 color	$\begin{cases} \text{red} \\ \text{green} \\ \text{brown 30} \\ \hline \hline \end{cases}$	color $\begin{cases} \text{red 30} \\ \text{green} \\ \text{brown} \\ \hline \hline \end{cases}$	color $\begin{cases} \text{red} \\ \text{green} \\ \text{brown 30} \\ \hline \hline \end{cases}$
0.50 shape	$\begin{cases} \text{round 15} \\ \text{square} \\ \text{cylindrical 5} \\ \hline \hline \end{cases}$	shape $\begin{cases} \text{round 20} \\ \text{square} \\ \text{cylindrical} \\ \hline \hline \end{cases}$	shape $\begin{cases} \text{round 20} \\ \text{square} \\ \text{cylindrical} \\ \hline \hline \end{cases}$
0.25 texture	$\begin{cases} \text{smooth 25} \\ \text{rough 5} \\ \text{bumpy} \\ \hline \hline \end{cases}$	texture $\begin{cases} \text{smooth 30} \\ \text{rough} \\ \text{bumpy} \\ \hline \hline \end{cases}$	texture $\begin{cases} \text{smooth 30} \\ \text{rough} \\ \text{bumpy} \\ \hline \hline \end{cases}$
		$\begin{aligned} \text{Sim (BA, } I_1) &= 2 (0-30-30) \\ &+ 0.50 (15-5-5) \\ &+ 0.25 (25-5-5) \\ &= -120 + 2.50 + 3.75 \\ &= -114 \end{aligned}$	$\begin{aligned} \text{Sim (BA, } I_2) &= 2 (30-0-0) \\ &+ 0.50 (15-5-5) \\ &+ 0.25 (25-5-5) \\ &= 60 + 2.50 + 3.75 \\ &= 66 \end{aligned}$

図4.5.4 異なる視点のもとでの典型度評定

か、すなわち、背景にある専門的知識（領域知識）の影響を、以下の2つの側面で受けると考えられる。

- 原型の特徴リストの中にどのような特徴を与えるか（本質的な特徴の選択）。
- 複合あるいは修飾された概念に対する原型の組み替え。

本節では、前節で提示した選択的修正モデルの考え方に説明に基づく学習（EBL）の枠組みを適用し、上記のような原型の組み替えを、背景知識を用いて系統的に進めるための手法を提案する。

いま、図4.5.5に示すような物体の事例Xを考える。このような新規事例が与えられたもとで、事例ベース（さまざまなカップの事例が集積されているとする）内から、以下の3つの文脈のもとで最も典型的な既知事例を決定することを考える。

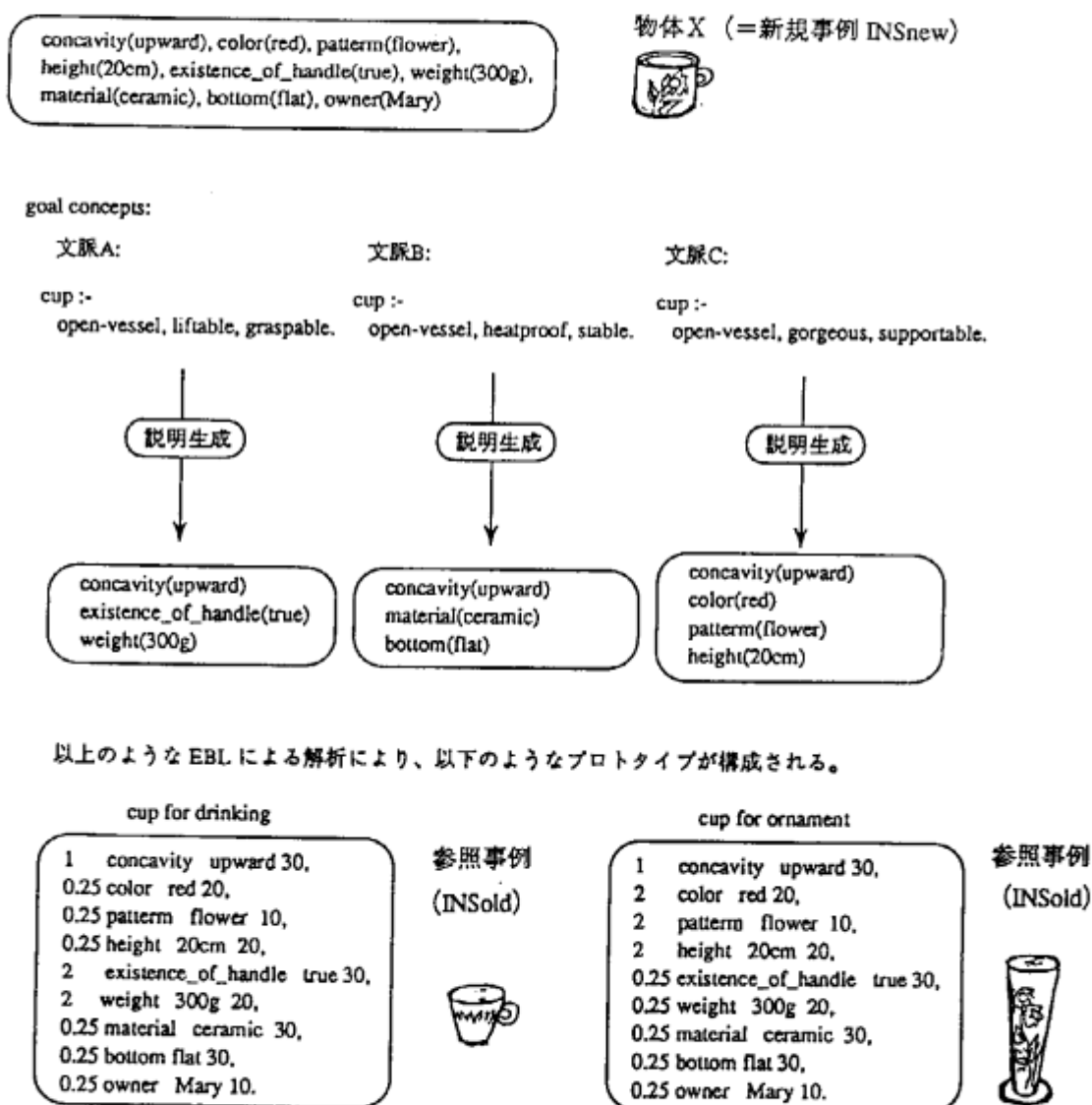


図4.5.5 文脈による参照事例の相違

文脈A：物体Xを手にもち珈琲をのんでいる。

文脈B：物体Xにはマッシュポテトが入れられ食卓の上におかれている。

文脈C：物体Xには花が飾られ棚の上におかれている。

各々の文脈で想起されるカップの目標概念 (goal concept) は、図4.5.5 に示すように定義され、各々の目標概念のもとで事例Xの説明生成がなされる (図に示すような目標概念は、上記の文脈記述から自動的に導出する手法も提案されている)。この結果、事例Xから説明生成に寄与した[属性—値]群が各々の文脈に合わせて決定されるが、これらの属性群について、もとのカップの原型記述に対して前節と同様の修正を施すことにより、各々の文脈に合わせたカップの原型定義が作り出される。そして、この原型との類似性評価により、各々の視点から見た場合に最も典型的な事例が、事例ベース中から参照すべき既知事例として検索されてくることになる。

4.5.4 特徴リストの属性間に依存関係が存在する場合の事例照合について

現在までに開発されている事例ベース推論の多くは、すべて事例が互いに独立で依存関係を持たないことを想定している。従って、特徴リストとしての静的な事例表現が有効になるわけであるが、逆に属性相互の間に因果性などの依存関係が存在する場合、換言すれば、事例表現の背後になにがしかの対象モデルが想定できる場合には、特別の注意を要する。このような場合に対処するための方法論として、定性推論の分野で興味深い考え方が提唱されている。

D. Weld により提唱された比較解析法 (Comparative Analysis) は、いくつかの定性方程式 (微分、単調増加、単調減少、積制約MULT、和制約ADD) で表された対象系の定性挙動 (qualitative behavior) が既知のもとで、その中の系パラメータが増減の変動 (perturbation) を受けたとき、その挙動にどのような影響が現われるかを推論するための手法である [Weld, 88]。その特徴は、挙動として得られている既知事例から、その背後にある対象モデル (定性方程式) に基づいて、新規事例における挙動を演繹的に導出するための推論手法である。著者らは、この考え方に基づき、大型船舶操舵のためのファジィ制御システムにおいて、新規の制御対象における制御則のチューニング・プロセスを、既知の制御則のチューニング結果を有効に活用しながら行なうための事例ベース推論システムを構築している [榎木他、89][Sawaragi, et al., 89][榎木他、90a][榎木他、90b]。

【参考文献】

- [Rosch 78] Rosch, E.: Principles of categorization. In E. Rosch and B.B. Lloyd (Eds.). Cognition and categorization, Lawrence Erlbaum Assoc. Pub., Hillsdale, NJ., 27-48, 1978.
- [Rosch 76] Rosch, E., and C.B. Mervis, W.D. Gray, D.M. Johnson and P. Boyes-Bream, : Basic objects in natural categories. Cognitive Psychology, 8, 382-439, 1976.
- [Osherson et al. 81] Osherson, D.N., and E.E. Smith: On the adequacy of prototype theory as a theory

- of concepts. *Cognition*, 9, 35-58, 1981.
- [Smith 88] E.E. Smith et al.: Combining Prototypes: A Selective Modification Model, *Cognitive Science*, 12, pp.485-527, 1988.
- [Mitchell, et al., 86] Mitchell, T.M. and R.M. Keller and S.T. Kedar-Cabelli (1986). Explanation-based generalizations: a unifying view, *Machine Learning*, 1, 47-80.
- [Sawaragi, et al., 89] T. Sawaragi, S. Iwai and O. Katai : Conceptual Clustering: Discovering Patterns from Chaotic Reality and Pattern-Directed Decisionmaking, in *Analysis, Design and Evaluation of Man-Machine Systems 1988. Selected Papers from the Third IFAC/IFIP/IEA/IFORS Conference*, Pergamon Press, 409-414, 1989.
- [Sawaragi, et al., 90] T. Sawaragi, S. Iwai and O. Katai : Self-Organization of Conceptual Generalities and Pattern-Directed Learning, to be appeared in *AUTOMATICA*, Vol.26, No.6, 1990.
- [Weld, 88] D.Weld: Comparative Analysis, *Artificial Intelligence*, 36, 333-373, 1988.[榎木他、89] 榎木、小川、片井、岩井：オペレータの操作履歴の解析に基づく知的ファジィ制御、第5回ファジィシステムシンポジウム、139-144、1989.
- [Sawaragi, et al., 89] T.Sawaragi, K.Ogawa, O.Katai and S.Iwai: Skill Acquisition for Intelligent Fuzzy Controller by Analyzing Operator's Control Action Sequences, *Proc. SICE'89 in Matsuyama*, 1369-1372, 1989.
- [榎木他、90] 榎木、片井、岩井、坂本：説明学習に基づく事例からのファジィ制御則の自動チューニング、計測自動制御学会第11回知識・知能工学シンポジウム、93-98, 1990.
- [榎木他、90] 榎木、片井、岩井、坂本：定性推論による操作履歴の比較解析に基づく知的ファジィ制御、第34回システム制御情報学会研究発表会発表予定、1990.

日本たばこ(株)生産技術研究所 中村孝太郎

機械調整問題を例にとり、事例ベース推論で利用される領域知識を分類し、各推論過程での役割を分析し、ユーザと領域知識間の、および異なった領域知識間の参照関係や相互補完的關係をまとめた。本検討は、領域知識とユーザの位置付けおよび協調関係のモデルを明確化したことにより、対話型事例ベース推論システムの実用化に利用できるものと考えられる。

対話的な運用を想定した事例ベース推論モデルの枠組みを説明するとともに、機械調整領域に適用するための事例の形式化および獲得方法について述べる。

事例ベース推論の一般的なモデルは、次の①～⑥から構成される。

- この中で特に、①問題の特徴づけ、④事例の修正、および⑥事例の修復については、これらを遂行する上で問題領域固有の知識（領域知識と呼ぶ）を必要とする。領域知識が完全であれば、上記の①、④、および⑥は自動化することが可能である。しかし、領域知識が不完全な場合には、これを何らかの方法で補完することが必要になる。ここでは、事例ベース推論の利用形態として対話の利用を指向する。すなわち、領域知識の支援の下に、ユーザが①、④および⑥のステップに介入する方式である対話型事例ベース推論モデル〔中村89〕を想定し、実用的見地から領域知識とユーザの協調を検討する。本モデルの説明図を図4.6-1に示す。ただし、②の事例の検索において、
-
- ```

graph TD
 User([ユーザ])
 Problem[問題]
 Solution[解]
 General[通用]
 Modify[事例の修正]
 Repair[事例の修復]
 Search[事例の検索]
 Eval[事例の評価]
 Base[事例ベース]
 Domain([領域知識])

 User --> Problem
 User --> Solution
 User --> General
 User --> Repair
 User --> Domain

 Problem --> Search
 Search --> Eval
 Eval --> Modify
 Modify --> Solution
 Solution --> General
 General --> Repair
 Repair --> Base
 Base --> Search
 Base --> Eval
 Base --> Modify
 Base --> Repair
 Base --> Domain
 Domain --> Search
 Domain --> Eval
 Domain --> Modify
 Domain --> Repair
 Domain --> General
 Domain --> User

```
- 図4.6-1 対話型事例ベース推論モデル

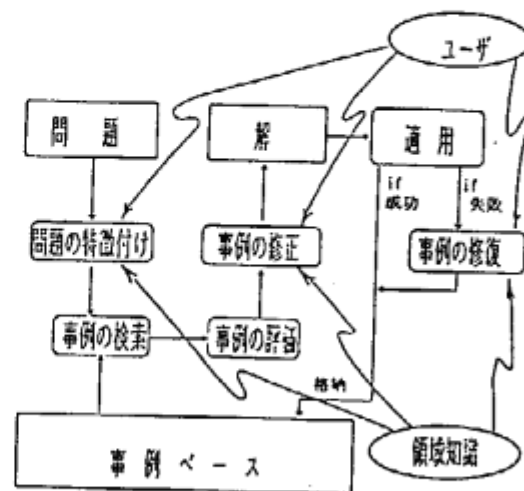


図4.6-1 対話型事例ベース推論モデル

照合を行うために領域知識が関連するが、ここでは②は一次検索的なものと考え、④で詳細な照合を行うものとする。

#### 4.6.1.2 適用対象と事例

機械調整では、機械稼働上や製品品質上の、異常や不具合を解消することを目標として原因の同定や調整方法（対処方法）の選択が行われる。機械調整における事例とは、「ある異常や不具合が認められたという調整条件のもとで、調整目標が設定され、これに基づき調整方法が実行され、その結果と評価が得られた場合の具体例」である。ここで、調整条件とは、対象に関して異常が認められ、兆候等も用いて、診断を行った結果、原因が同定された状況下のことである。すなわち、事例は図4.6-2 に示されるような5つの属性から表現され、さらに調整条件は3つの属性から表現されるものとする。

なお、兆候とは、異常と原因を結ぶ因果関係の過程における（因果的パス）観測可能な特性項目のことをいう。兆候として肯定的兆候と否定的兆候の2種を考える。例えば、「のり付け位置が振れる」は肯定的兆候であり、「ヒータは特に問題はない」は、否定的兆候である。

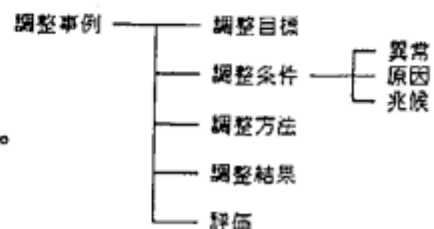


図4.6-2 調整事例の表現形式

事例の獲得方法を以下に示す。①文書化された調整事例として、生産管理上の記録や標準作業マニュアルの調整基準データ等が利用可能である。②文書化されていない場合については、専門家からのヒヤリングを行い、事例を獲得する作業が必要である。専門家からの事例獲得は図4.6-2 の形式の事例を“1つのエピソード”として抽出すればよいことから、いわゆる知識獲得に比べて容易な作業と考えられる。

このようにして獲得された調整事例の例を図4.6-3 に示す。

|       |                                     |
|-------|-------------------------------------|
| 事例-55 |                                     |
| 調整目標: | フィードインが正常                           |
| 調整条件: |                                     |
| 異常:   | 機械始動時、棒巻がフィードインしない                  |
| 原因:   | ロッドヒータ1の高さ調整ストッパの不良                 |
| 兆候:   | ヒータ2が下降時に棒巻が変形した                    |
| 調整方法: | ロッドヒータ1の高さ調整ストッパの調整                 |
| 調整結果: | 始動時の棒巻の圧縮による変形がなくなり、スムーズにフィードインができた |
| 評価:   | 良好(成功)                              |

図4.6-3 調整事例の例

以下、4.6.2 節では、機械調整領域を例にとり、領域知識の種類別に内容や獲得方法等を述べ、4.6.3 節では、事例ベース推論の枠組みにおける領域知識の利用方法を例示する。次に、4.6.4 節では、これらの領域知識とユーザとの協調の方式について、事例ベース構築管理時、事例ベース推論実行時、および事例の一般化時の各々についてまとめる。

#### 4.6.2. 領域知識の種類（機械調整領域を例として）

事例ベース推論で使用する領域知識には、概念階層的知識、因果的モデルおよび修正知識等がある。各々の領域知識について機械調整領域を例として説明する。領域知識を獲得するためのコストが高いと領域知識を完備することが困難となり、対話型事例ベース推論過程におけるユーザへの依存性が高くなる。したがって領域知識の獲得方法とその獲得コストについてもふれる。

##### 4.6.2.1 概念階層的知識

概念階層的知識とは、与えられた事例や問題を構造化するための知識および事例の修正と修復において利用される知識であり、主として概念間の階層的な関係を表す知識からなる。概念階層的知識の例を図4.6-4に示す。図のa)は機械調整事例に関する知識であり、b)は機械の部品階層に関する知識である。なお、a)には、事例の各属性がもつことが可能な属性値の候補に関する知識も含まれる。

このような概念階層的知識の獲得に関しては、上記の知識a)は適用領域の専門家モデルの分析と専門家からのヒヤリングから得ることができる。知識b)は機械の構造および機能に関する知識であり、設計仕様書、機械の説明書等から得ることが可能である。

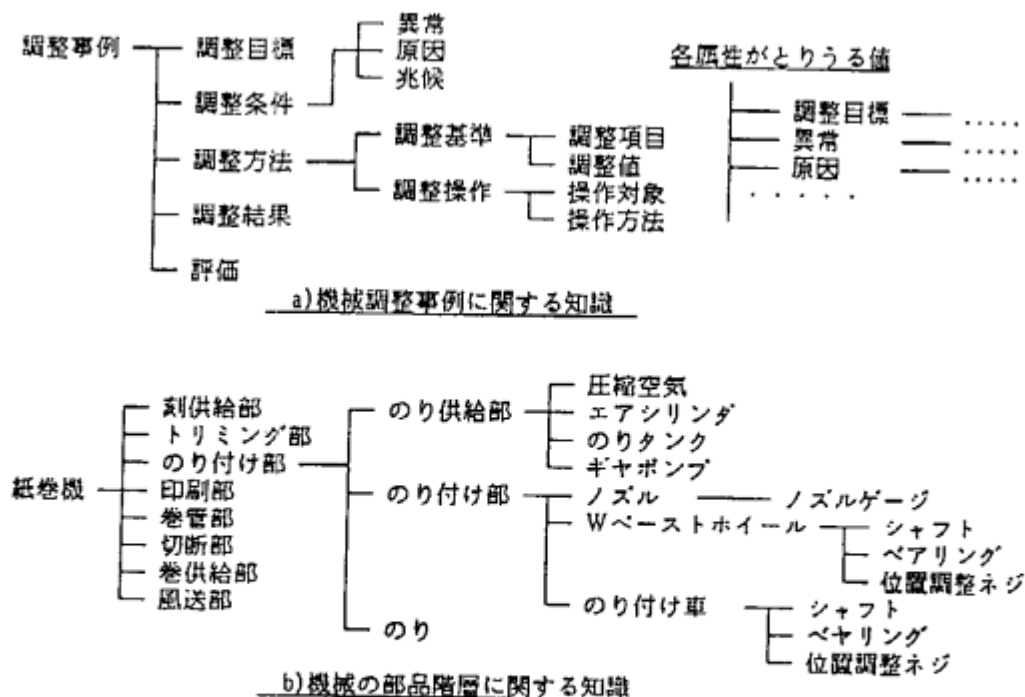


図4.6-4 概念階層的知識の例

#### 4.6.2.2 因果的モデル

因果的モデルとは、因果的解析（入力伝播）により事例の正当性を証明する（例えば挙動を原因に関連付ける等）ための知識であり、機械の構成要素間の因果的な関係を表す知識からなる。

すなわち、機械の各構成要素は、それぞれ複数の特性項目により特徴付けられている。ある構成要素の特性項目の変化が、構成要素間のつながりを通じて、別の構成要素の特性項目に影響を及ぼす時、両者間に因果的関係があると考え、このような因果的関係の集まりを、因果モデルと考える。機械調整の対象となる紙巻機の因果モデルの一部を図4.6-5に示す。

なお、因果的モデルには、各特性要素に対応して、点検観測に関する知識が付加される。本知識は特性要素の観測容易性および診断戦略上の重要性を示し、後述する事例の一般化〔中村90〕時に因果的パスから操作的知識を抽出する場合に利用される。調整専門家が実際に利用していると推定される因果的モデルは次の方法で同定される。

- ①対象とする機械のシステム構造および機能に関する説明書
- ②専門家からの機械に関する因果的フローに関するヒヤリング

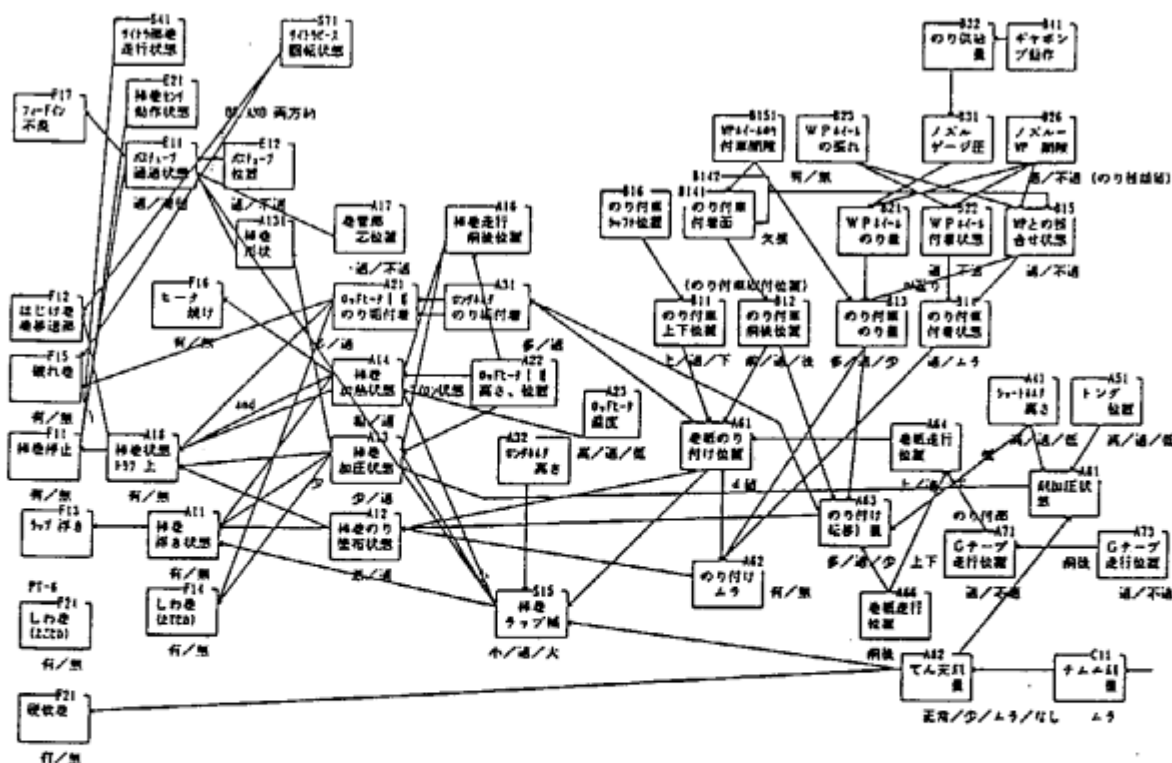


図4.6-5 紙巻機に関する因果的モデル（一部）

シンボルは、図 4.6-4 を基本とする

#### 4.6.2.3 修正知識

修正知識とは、概念階層的知識または因果的モデルを利用して、問題と事例間の部分的照合を行った上で、事例の変更を行うために使われる知識であり、主として経験的な知識からなる。

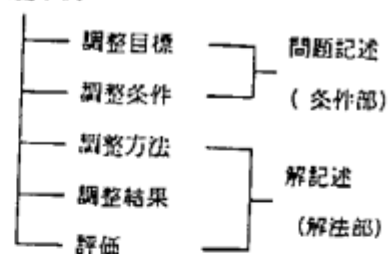
修正知識は、a)修正の推論制御に関する知識、およびb)領域依存的な修正方法に関する知識からなる。本知識の例を図4.6-6に示す。知識a)は、問題と事例の対比により、属性構造(事例の条件部、解法部等の区分)の検出、および概念階層的知識あるいは因果的モデルにより、照合処理およびこれに基づく修正処理を行うための制御的知識である。知識b)は、修正処理の基本ループで表現できない領域依存的な知識であり、例えば照合の対象となる属性と修正の対象となる属性の関連性(ある特定の原因の場合には特定の調整基準を用いる)等を示す。これらの修正知識により属性値または属性構造の修正が行われる。

修正知識は、概念階層的知識や因果的モデルのように、最初から組織的な知識獲得を行うことは困難を伴うため、他の領域知識の獲得および事例ベース推論の実行とともに漸進的に行なうことが必要である。上記の知識a)は、適用領域の専門家モデルの分析に基づいて獲得される。また、知識b)は、属性間の因果的または時間的な制約条件の分析等から得られる。

- ・事例(C), 事例種類(CK, C) — 属性構造(CK, [CP, SP, EP])
- ・事例(C), 属性構造(CK, [CP, SP, EP])
  - 照合(P, CP, MR), 修正(MR, C)
- ・照合(P, CP, MR) — 概念階層的知識による照合(P, CP, MR)
- ・照合(P, CP, MR) — 因果的モデルによる照合(P, CP, MR)
- ・修正(MR, C) — 概念階層的知識による修正(MR, C, MDR)
- ・修正(MR, C) — 因果的モデルによる修正(MR, CP, MDR)
- ・修正(MR, C) — 領域依存的修正(MR, CP, MDR)

a) 修正の推論制御に関する知識 (一部)

調整事例



b) 領域依存的知識 (一部)

図4.6-6 修正知識の例

#### 4.6.3 領域知識の利用例

前節で述べた領域知識の質と量、およびこれに基づく処理が完備されるほど、ユーザの関与は少なくなる。

そこで本節では、有効な事例処理を行うための適度な知識の獲得が可能な実用的見地から、各領域知識の表現および利用機能を下記のように仮定する。

①概念階層的知識 ⇒ a)機械調整事例に関する知識の表現と参照機能

b)部品階層的知識の表現と部分照合機能

②因果的モデル ⇒ 各構成要素はディスクリートに表現可能、構成要素は複数の特

性項目からなり、特性項目間の因果的連関が表現される。

因果解析機能および説明生成機能

③修正知識

⇒ a)修正の推論制御に関する知識

①、②およびユーザによる照合の手順の制御

①、②による修正および領域依存的修正の手順の制御

b)領域依存的修正知識

事例属性項目の区分（条件部、解法部等）

以上のような領域知識が事例ベース推論システムで利用可能なこと前提とした場合  
その推論過程における各領域知識の利用方法を次節以降で例示する。

#### 4.6.3.1 概念階層的知識の利用

概念階層的知識の利用例について事例ベース構築時、問題の特徴付け時、および事例の修正時について利用例を示す。

##### ①事例ベース構築時

専門家による事例の形式化（概念階層的知識に含まれる属性項目に基づいて事例データの再構成を補助入力を伴って行なうこと）および事例の詳細化（概念階層的知識に基づいて徐々に属性項目ごとの特徴を明確にし展開していくこと）をガイドする。

例えば、図4.6-7に示すように事例情報の形式化を行なう場合、該当する属性項目の選択項目を行なうこと、および各属性ごとに入力可能な属性値の候補提示等に機械調整事例に関する知識が利用される。

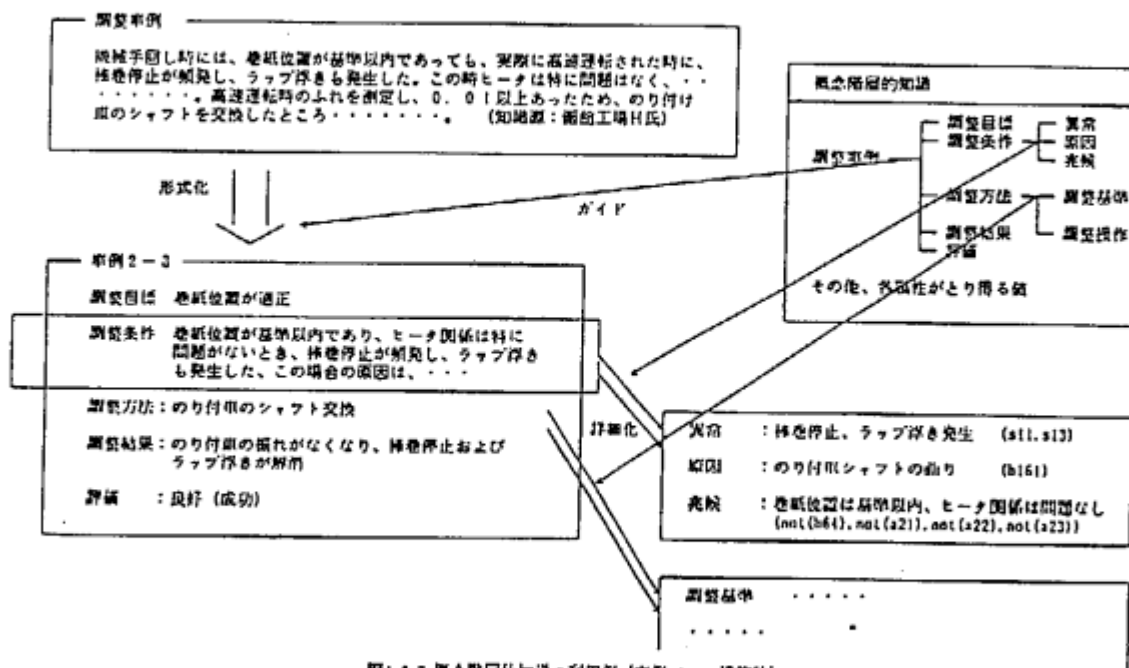


図4.6-7 概念階層的知識の利用例（事例ベース構築時）

## ②問題の特徴付け時

与えられた問題の特徴分析のガイドを行なう場合も と同様な形式化や詳細化が必要に応じて行なわれるため概念階層的知識が利用される。

## ③事例の修正時

問題と事例間で詳細な照合条件のチェックを行なう場合および照合操作に基づいて行なわれる解法の変換を行なう場合に参照される。

例えば、図4.6-8 に示すように、問題と事例の条件部の間での部分的照合の可能性を提示したり、事例が含む解法への変換を行なうときに、属性値の修正のための候補データを提示する際に、機械の部品階層に関する知識が利用される。

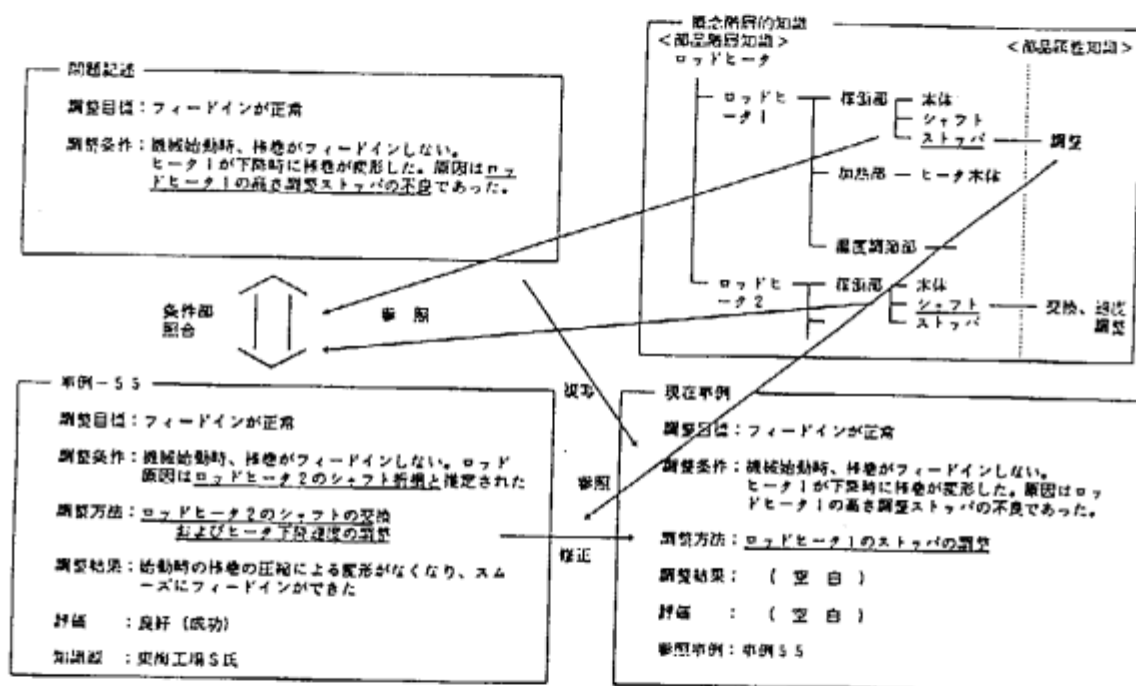


図4.6-8 概念階層的知識の利用例 (事例の修正時)

### 4.6.3.2 因果的モデルの利用例

因果的モデルの利用例について事例ベース構築時、問題の特徴付け時および事例の修正時について利用例を示す。

#### ①事例ベース構築、現在事例の格納時

事例格納時のインデックスの設定は、事例に含まれる情報をもとに行なわれる。この場合、事例の情報をインデックスデータとして直接利用するのではなく、一般化や操作化（観測容易性や重要性の高い情報として抽出）を行なった上でインデックスに利用する方が、検索効率の観点から効果的である。

例えば、図4.6-9 に示すように、EBL（説明に基づく学習）を用いて調整事例（特に調整条件情報）の因果的モデルによる説明を行ない、一般的かつ操作的なインデックスを同定する場合に、因果的モデルが利用される。



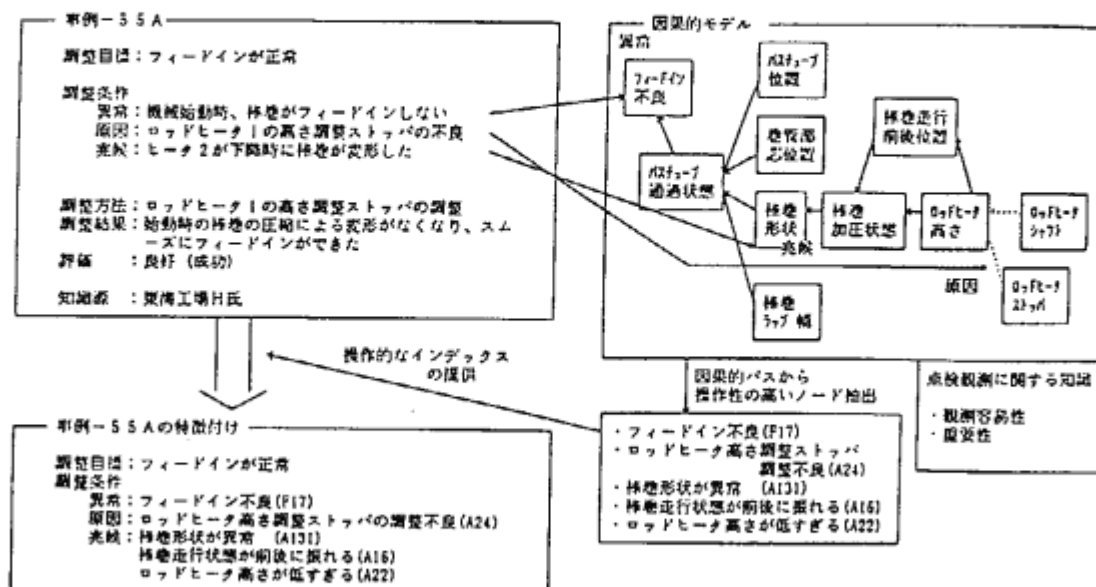


図4.6-9 因果的モデルの利用例 (事例ベースの構築、現在事例の格納時)

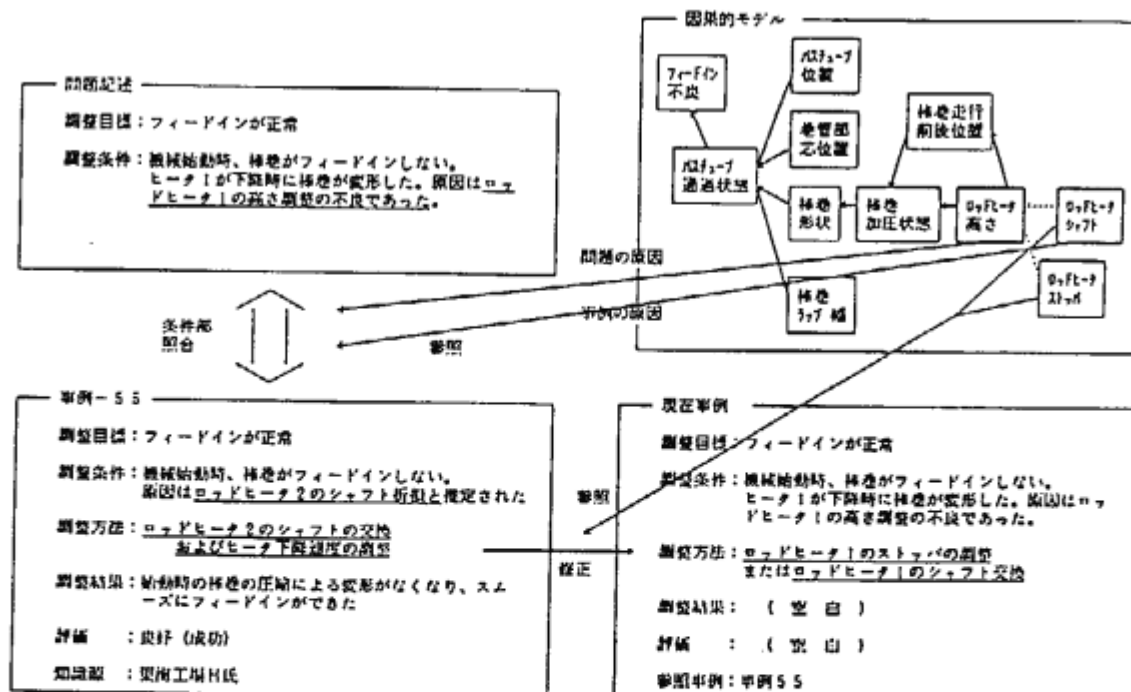


図4.6-10 因果的モデルの利用例 (事例の修正時)

## ②問題の特徴付け時

問題の特徴付けに際しても形式化後の属性値入力時には①と同様なニーズが存在する。一般性や操作性が保証された特徴付けを行なうことは、類似事例の検索負荷の軽減につながる。

## ③事例の修正時

問題と類似事例の照合に基づく修正を行なうために、因果的な包含関係を検出することにより詳細な照合条件のチェックおよび照合操作に基づいて行なわれる解法の変換を因果的な前後関係を参照して行われる。

例えば、図4.6-10に示すように調整条件の原因に関して、問題に対し類似事例がより特殊な原因をもつものとして照合操作を行い、これに基づき可能性のある調整方法が提示される。

### 4.6.3.3 修正知識の利用例

修正知識は事例の修正時に利用される。例えば、事例の修正時に図4.6-8では概念階層的知識が、図4.6-10では因果的モデルが利用されたが、これらはいずれも図4.6-6で示した修正の推論制御に関する知識に基づき照合順序等が制御されている。

さらに問題と事例の条件部間の差異分析により、因果的モデル上の前後のノードあるいは概念階層上の上下のノードをたどること（局所的な探索）により、事例の解法の修正を行う場合も本知識によりガイドされる。例えば、図4.6-8では、「ロッドヒータ」の部品階層上の曾孫ノード同士を対比することにより、調整方法の修正が行われている。

また、領域依存的な修正知識で処理可能な属性値がある場合はこれを利用して、属性ごとに修正方法の候補を提示する。例えば、調整方法として「シャフトの交換」が選択された場合、これを詳細化する際に「シャフト」に関する調整基準の候補を提示する。

### 4.6.4 領域知識とユーザとの協調の方式

前節において、各領域知識ごとに、その利用方式についての検討を行った。本節では、事例ベース推論の各過程における領域知識間および領域知識とユーザ間の協調の関係をまとめる。以下、事例ベース構築時、事例ベース推論実行時、および事例の一般化〔中村90〕時の各々について述べる。

なお、事例の一般化とは、診断知識の獲得や洗練化のために、事例ベース推論システムにより蓄積された事例を正当化し知識に変換することである。一般化の手順は、因果的モデルによる説明構造の生成、操作性規範（観測容易性および診断戦略的重要性）による説明構造の縮約、および適切なルールへの変換からなる。本研究では事例の一般化機能も事例ベース推論の枠組みの必要な拡張機能と考え、本節においてまとめた。

#### 4.6.4.1 事例ベース構築・管理時

事例ベース構築・管理時の領域知識とユーザの協調関係を図4.6-11に示す。図中の矢印のうち一方向矢印はユーザによる参照関係を、双方向矢印は相互補完的关系を示す。

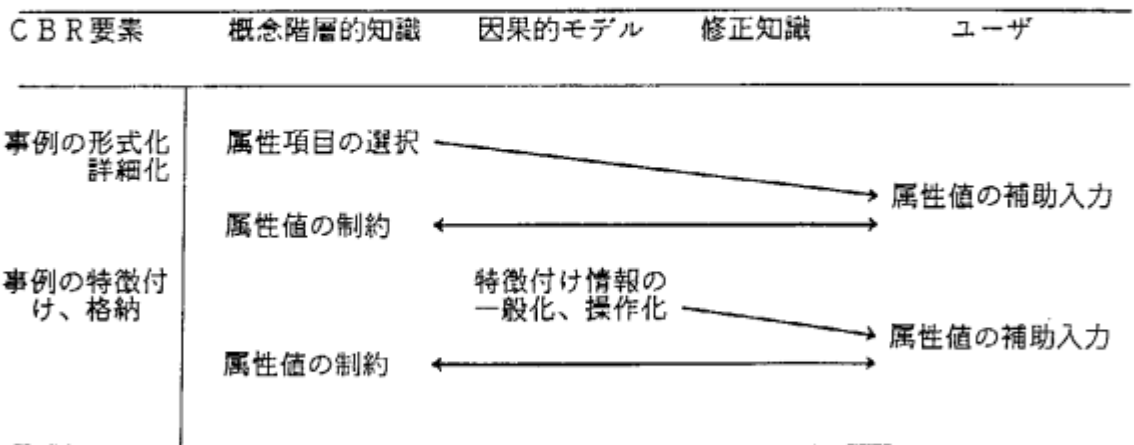


図4.6-11 事例ベース構築・管理時の協調関係

#### 4.6.4.2 事例ベース推論実行時

事例ベース推論実行時の領域知識とユーザの協調関係を図4.6-12に示す。矢印の意味は前節と同様であるが、修正知識は他の領域知識をガイドする。

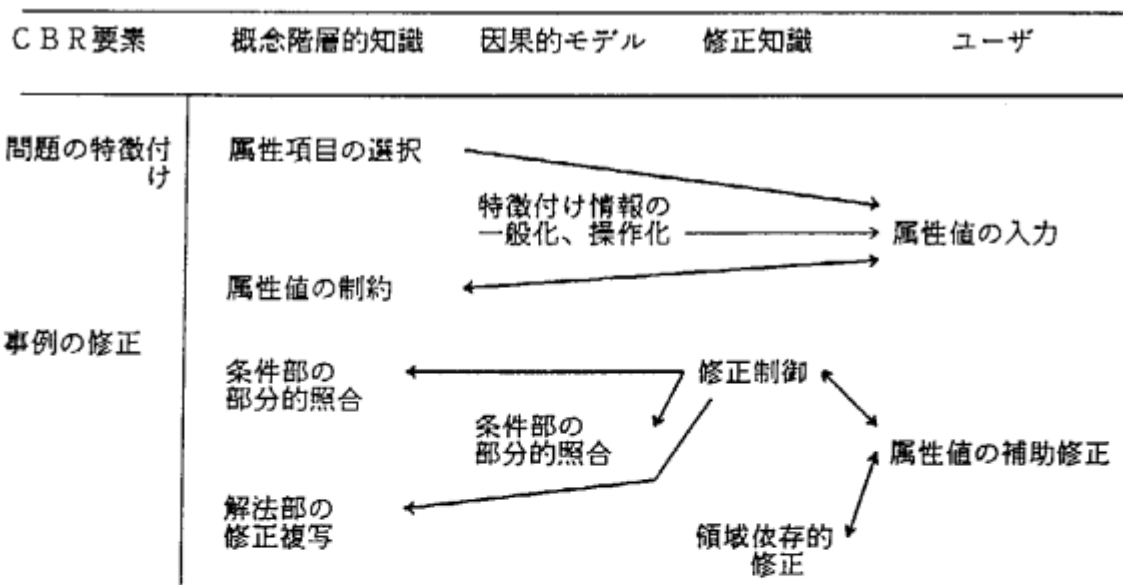


図4.6-12 事例ベース推論実行時の協調関係

#### 4.6.4.3 事例の一般化時

事例ベース推論実行時の領域知識とユーザの協調関係を図4.6-13に示す。矢印の意味は今までと同様であるが、修正知識に相当するのは、汎化変換規則でありルールへの変換操作をガイドする。

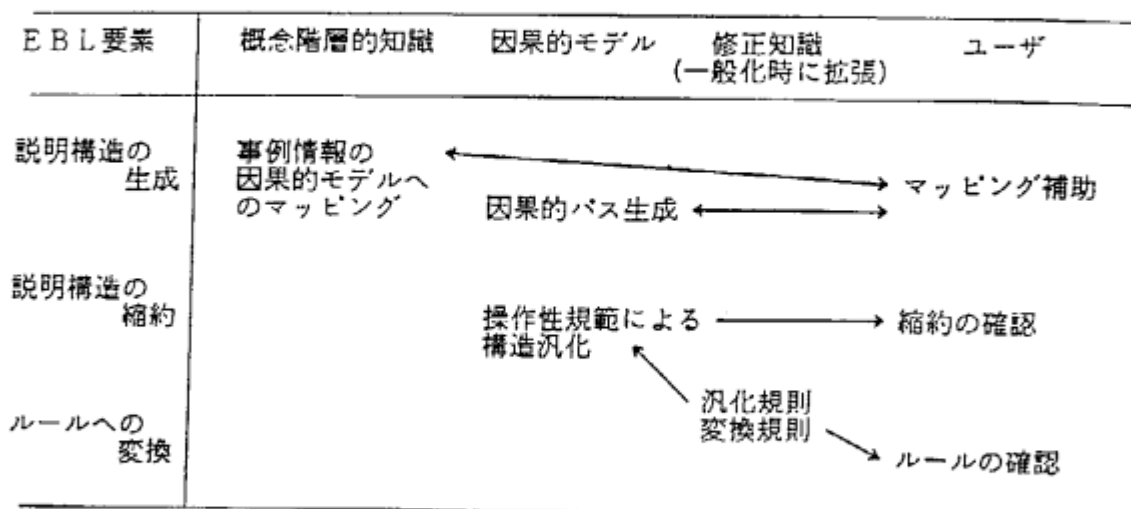


図4.6-13 事例の一般化時の協調関係

#### 4.6.5 まとめ

機械調整領域に事例ベース推論を適用した場合について、必要な領域知識とその利用方法について検討し、領域知識の表現や利用機能を実用的観点から仮定した上で、ユーザとの協調および領域知識間の協調関係についてまとめた。

領域知識の役割の特徴として以下が指摘される。①事例の特徴付けや照合のために、概念階層的知識および因果的モデルが主に利用され、事例の修正のために、概念階層的知識と修正知識が主に利用される。②因果的モデルと概念階層的知識は、事例や問題の特徴付け時および事例の修正時に相補的な関係にある。

今後の課題として次の事項が挙げられる。①他の代表的な事例ベース推論システムについても検討を行いより妥当性のある検討を行う。②本検討において、SBL（類比に基づく学習）の実用的枠組みを明確化することにより、SBLに必要となる背景知識の観点から、領域知識への依存度の検討を進める。

#### 参考文献

- [小林 89] 小林：事例ベース推論の研究課題（その1、その2）ICOT/KAR-SWG資料
- [中村 89] 中村：事例ベース推論の対話型モデルとその機械調整支援への適用、人工知能学会誌 Vol.4, No.6, pp83-91 (1989)
- [中村 90] 中村：因果的モデルと操作性規範に基づく機械調整事例の一般化、計測自動制御学会第11回知識・知能システムシンポジウム予稿集, pp99-104(1990)
- [Kolodner 89] Kolodner, J. et.al. : Case-Based Reasoning, Tutorial, IJCAI-89 (1989)

## 4.7 CBR と EBL : Intractable Domain への適用

日立製作所 基礎研究所 吉田健一

### 要旨

本節では機械学習の1手法である EBL [Mitchell 86, DeJong 86] と CBR の関係について検討し、両者が極めて密接な関係にあることを指摘する。

#### 4.7.1 設計問題と CBR

##### 4.7.1.1 設計問題の性質

CBR と EBL の関係を分析する準備として、本ワーキング・グループの持つもう一つのキーワードである設計問題の性質について考えてみる。

一般に、設計問題は診断等に比較して解の探索スペースが大きい。このことは、ワーキング・グループの第1会目の会合で小林主査が「設計問題の多くは OR 問題として定式化した場合、NP 完全となるため、計算量の壁を克服するため、新しいアプローチの開発が望まれている。」[小林 89] と指摘した通りである。すなわち、設計問題は、対象の動作原理等は既知だが、必要な情報量が扱い切れない、Intractable な問題領域に属していると言える。

このような問題には、断片的知識 (Domain Model または If Then Rule で表現した Heuristics) を組合わせて利用しようとする Expert System より、CBR または EBL 的考えを利用した System が有利に思える。すなわち、事前に用意した chunk の利用により、初期設計時の探索 (探索スペースが大きい場合問題) が不要になり、計算時間などの短縮が期待される。<sup>1</sup>

断片的知識を利用した Expert System と、CBR (または EBL) 的考えを利用した System の対比を図 4.7-1 に示す。

---

<sup>1</sup>階層性を利用して探索スペースを削減するアプローチについては、後述する。

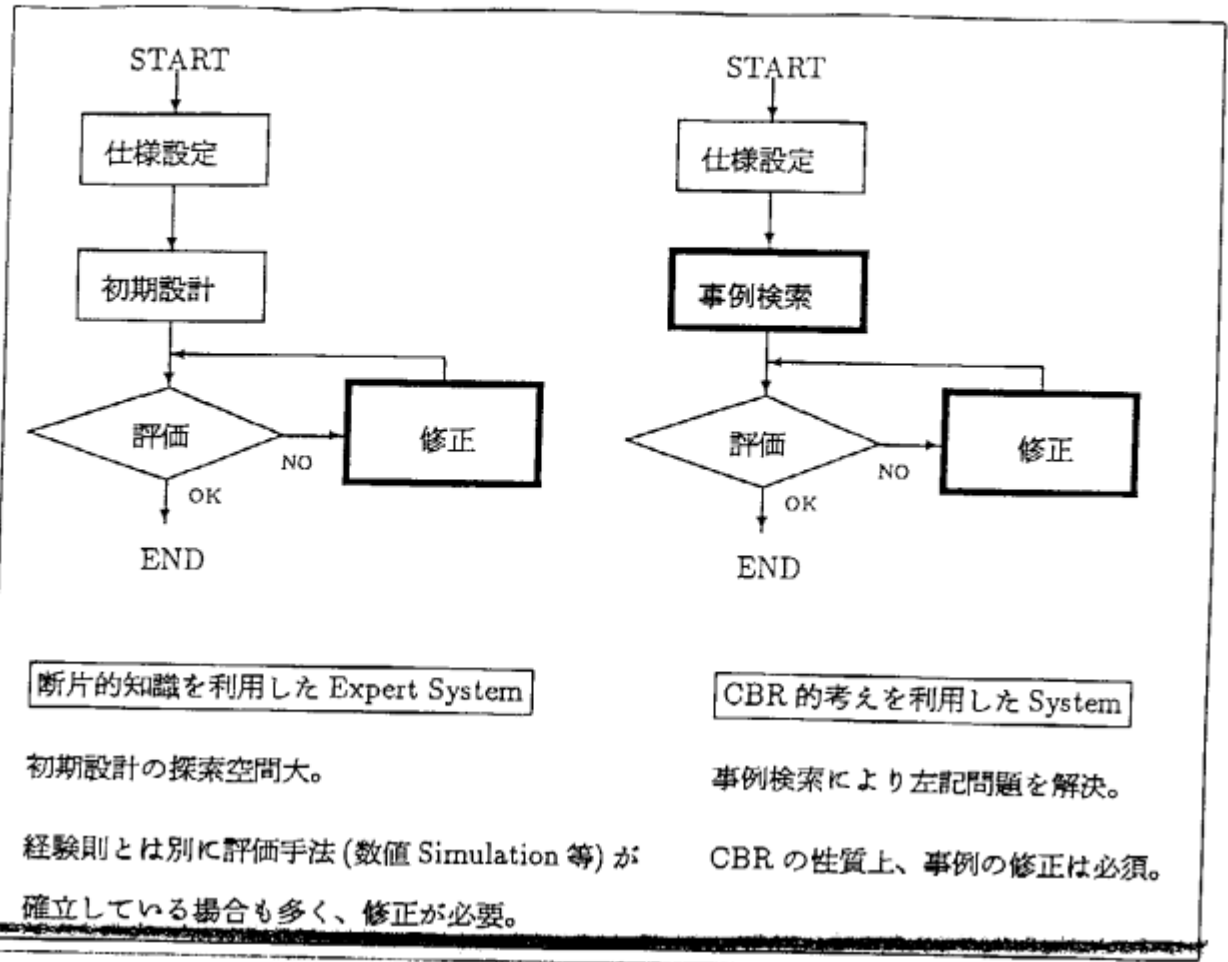


図 4.7-1

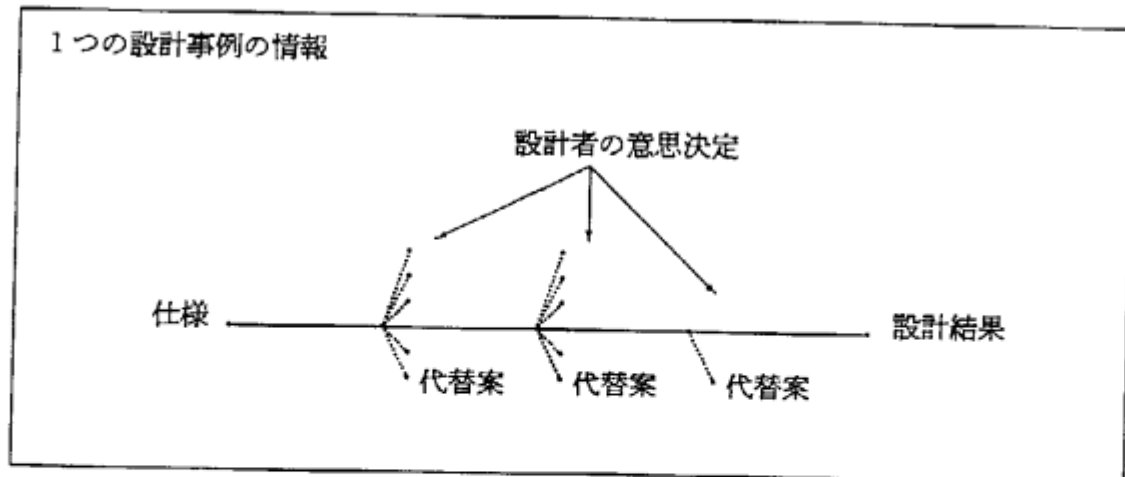


図 4.7-2

#### 4.7.1.2 設計変更を考慮した CBR

設計問題にとり結果の修正は重要である。以下、この設計変更の扱い方と CBR の関係を考察する。

断片的知識を利用した Expert System にしろ、CBR(または EBL) 的思考を利用した System にしろ、初期設計の結果を評価し修正を加えるステップは重要である (図 4.7-2)。この時、修正を自動化しようとする、何故そのような設計がおこなわれたかの理由 (設計者の意図) の情報が必要になる [Mitchell 8?]。すなわち、必要な修正がされないことを防止するためには、領域知識により、予め設計事例を分析しておき、「どこを修正したら、どこも修正しなければならない。」といった、データを事前に用意しておく必要がある。

この時、領域知識による説明構造を、そのまま indexing に利用するのなら EBL そのものとなってしまう、CBR との違いはない。しかしながら、設計対象の属性を、代替案の多いものと、小さいものに分類できれば、代替案の多いもののみ key として利用し、小さいものは修正で対処するという CBR の方針がうまく適用できる可能性がある。

すなわち、EBL では代替案の情報は消えている (図 4.7-2 の直線部分の情報のみ) が、CBR 的思考方を適用すれば、探索スペース縮小の観点から、代替案の多い場所での意思決定の情報のみ記憶し、少ない場所は、事例の修正で対処することが可能である。

このような考えに基づいた設計例のケース・ベース作成の特徴は次のようになる。

1. 代替案の多い属性のみ indexing の key として利用。
2. 代替案の少ない属性は、後で修正。
3. EBL 利用の System と同じく、強力な domain knowledge を利用。  
代替案に関する情報を利用して key を選択する分だけ、より domain knowledge に依存。
4. 1. により、EBL 利用の System より検索時間が短い。
5. 代替案が少ない属性が多ければ、それらの属性に関する修正コストは無視できる。

#### 4.7.2 対象の階層構造と EBL

この節では Intractable Domain への EBL の適用研究 [Bennett 87, Doyle 86, Mozetic 87, 吉田 89, 元田 89] の例として、吉田らによる研究 [吉田 89, 元田 89] を紹介し、前節で述べた、設計例のケース・ベース作成の特徴との比較を試みる。

吉田らは、複雑な対象を階層的に表現し、階層的知識ベース作成の構築を試みている。この中では、対象を階層的に表現し、上位レベルの浅い知識は、下位レベルの深い知識の情報を一部無視して作成されるものとして扱っている。その中で、「下位の深い知識のうち、どの情報を残して上位の知識とするのか」という判断を EBL の枠の外で行なう、EBL を利用した浅い知識の合成法を提案している。図 4.7-3 に深い知識と浅い知識の関係を示す。ここでは従来の「task に依存した浅い知識と、対象の機能・構造に関する深い知識」の区別だけでなく、「対象の機能・構造に関する知識の記述の詳細度に依存した浅い・深い知識」も区別されている。

図 4.7-4,5 に電気回路の「対象の機能・構造に関する知識の記述の詳細度に依存した浅い・深い知識」を例示する。

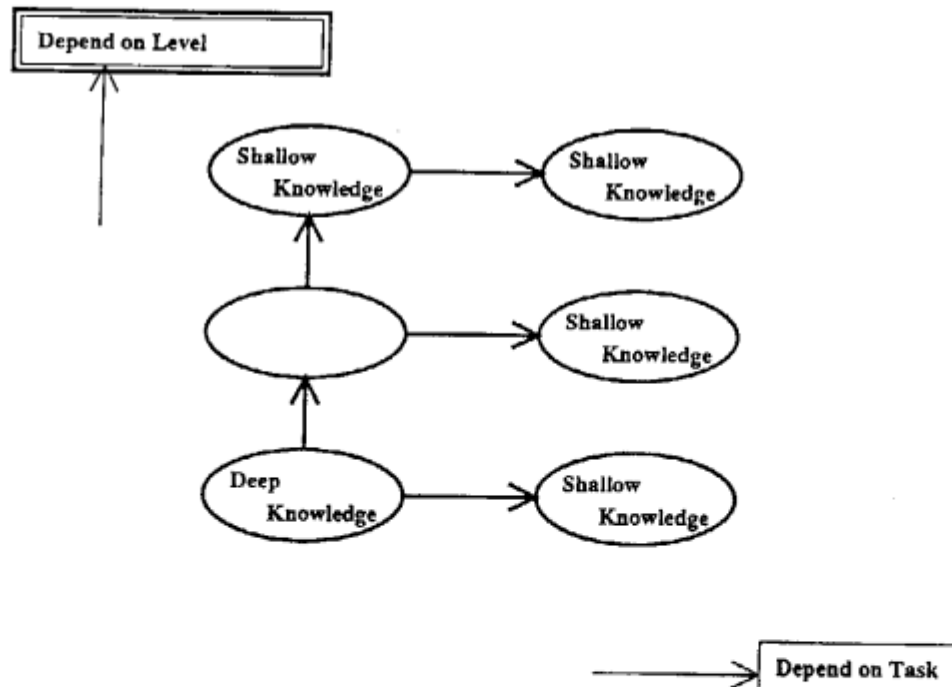


図 4.7-3



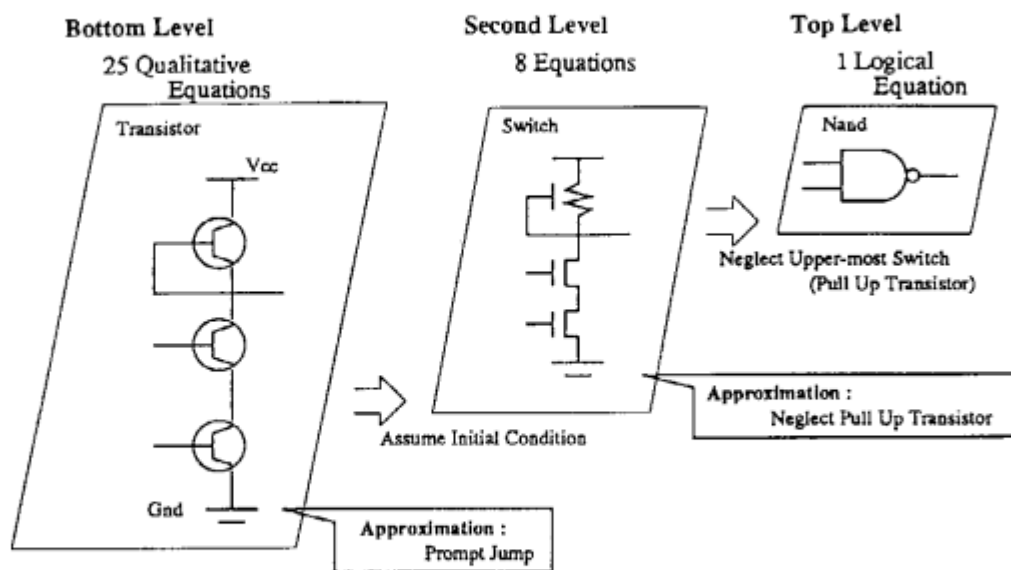
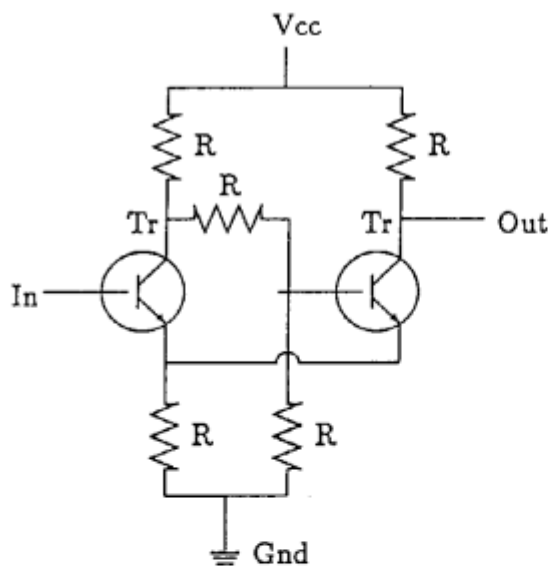


図 4.7-4



全体の代表的な挙動

2 方程式

- アンプとして動作
- 正の FB あり

↑ 挙動予測の結果

部品の代表的な挙動

28 方程式

- ベース電圧を変化させコレクタ電流を操作
- ベース電圧を変化させエミッタ電流を操作
- 抵抗の端子に流れる電流を変化させ電圧を操作
- 直列に接続した抵抗の端子電圧を変化させ中点の電圧を操作

↑ 例題の説明

回路方程式

34 方程式

図 4.7-5

4.7.1 で述べた設計変更を考慮した CBR が

- 方針は CBR : 代替案の多いもののみ key として利用し、小さいものは修正で対処する。
- Case Base の作成は EBL を利用 : EBL を利用して、Indexing 等の操作を実施。

のように、「何を記憶しておくか」という制御を EBL に取り込んだのに対して、吉田らの研究は「何を無視して詳細度を変化させるか」という制御を EBL に取り込んでいる。chunk を作成するのに mgu を使用する点ではどちらも同じ操作を行なっていることになる。

#### 4.7.3 Intractable Domain における EBL と CBR

以上、述べてきたように Intractable Domain における EBL と CBR は極めて密接な関係を持っており、立場によっては同じものとみなすこともできる。しかし、設計問題など intractable な領域に CBR 的思考を持ち込む上では、EBL の indexing 技術だけではなく、下記の項目の考慮も重要である。

- 探索スペース削減の為、CBR を使う場合、修正を考慮すると、EBL の外に、「代替案の多いもののみ key として利用し、小さいものは修正で対処する。」等 CBR 的立場のコントロールを付けるのが有効に思える。
- 特に、代替案が 2 つ以下の属性が多いと、それらの属性は後で修正しても、探索コストは無視できる。

#### 参考文献

- [Mitchell 86] Tom M. Mitchell, Richard M. Keller, and Smadar T. Kedar-Cabelli. Explanation-Based Generalization : A Unifying View. *Machine Learning*, 47-80, 1986.
- [DeJong 86] Gerald DeJong and Raymond Mooney. Explanation-Based Learning : An Alternative View. *Machine Learning*, 145-176, 1986.
- [小林 89] 小林. 設計問題へのアプローチ. ICOT KSA KAR WG 資料, KAR-1-3 (1989).
- [Mitchell 8?] T.Mitchell. An Intelligent Aid for Circuit Redesign. *IJCAI* 274-278 (198?).

- [Bennett 87] Scott W. Bennett. Approximation in Mathematical Domains. In *IJCAI-87*, 239-241 (1987).
- [Doyle 86] Richard J. Doyle. Constructing and Refining Causal Explanations from an Inconsistent Domain Theory. In *AAAI-86*, 538-544 (1986).
- [Mozetic 87] Igor Mozetic. The Role of Abstractions in Learning Qualitative Models. In *Fourth International Workshop on MACHINE LEARNING*, 242-255 (1987).
- [吉田 89] 吉田、元田. 階層的定性推論のための浅い知識の合成法. 人工知能学会誌, Vol. 4, No. 4, (1989).
- [元田 89] 元田、吉田. 定性推論と深い推論. 人工知能学会誌, Vol. 4, No. 5, (1989).

## 5. 事例ベース推論のモデルおよび適用例

### 5.1 モデルベース診断における経験的知識の学習

日本電気(株) C&Cシステム研究所 古関 義幸

#### 要旨

本節では、経験事例からの学習機能を持つ故障診断システムの研究[古関 89]を紹介する。このシステムは、深い知識(対象装置の構造及び動作に関する知識:設計知識)と浅い知識(経験的知識)を併用して柔軟な診断を行う。そして、経験した事例から、深い知識を利用しながら浅い知識の学習を行う。学習の方式として説明に基づく学習方式(EBL: Explanation Based Learning)を用いている。実験による評価の結果、経験した事例と似たような症例については、短い処理時間で解を求めることが可能であることが明らかになった。

#### 5.1.1 はじめに

従来、診断システムの方式として、大別して二種類の方式が提案されている。第一はMYCIN [Shortliffe 76]に代表されるような、熟練者の経験的知識(浅い知識)をルールベースに記述し利用する、いわゆるエキスパートシステムの方式である。第二は、対象装置の構造と本来の動作の記述(深い知識)を利用して診断を行うモデルベース方式である。

経験的知識を用いる方式では、知識ベースに記述されている症状に対しては能率的な診断が可能である反面、記述されていない症状に対しては対処が出来ないために、適用範囲が狭いという欠点がある。また、事前に熟練者のインタビュー等を通して経験的知識を整理し知識ベースに記述することは、大変困難な作業であり、かつ、対象問題と知識ベース構築技術の双方に精通した知識エンジニアを必要とし開発コストが高い。このように従来のエキスパートシステムの方式には、「適用範囲が狭いこと」と「知識獲得が困難であること」という二種類の問題がある。

これに対してモデルベース方式は、論理回路に対して古くから研究が行われ、テストパターン生成などで利用されている[Roth 67]。しかし、対象装置が複雑になり、論理回路レベルでの診断が不可能であるような装置レベルでの診断には応用が不可能であった。これを克服するために、DART [Genesereth 84]のように任意の記述レベルで診断を可能にする方式が提案されている。これらの方式には、装置の構成部品の動作を基に診断を行うために、計算時間がかかるという欠点がある。また、人間の持つようなヒューリスティックを持たずに論理的な判断だけで診断結果を絞り込むために、必要以上に多くのテストを必要とする場合があるという欠点もある。

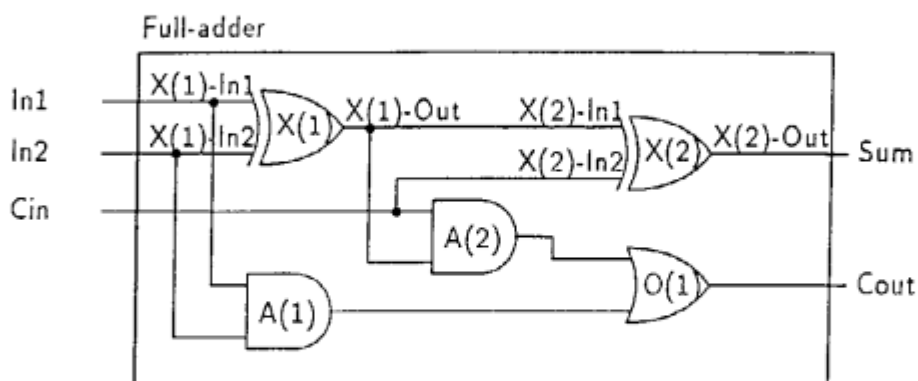


図 1: 例題

本報告では、経験した事例を一般化したルールの形式で経験知識ベースに蓄積しておくことにより、過去の経験に類似した症例については、複雑な計算をせずに診断を行うことのできる方式について述べる。知識の記述方式、推論方式、および実験プログラムの結果について述べた後、今後の課題について述べる。

### 5.1.2 知識ベースの構成

本方式では、装置の構造と動作を記述した論理的知識ベースと、経験的知識を蓄えた経験知識ベースとの2種類の知識ベースを利用して診断を行う。経験知識ベースで診断可能な症例に対しては、過去の経験に基づいて能率の良い診断を行い、経験知識ベースに無いような症例に対しては、論理的診断方式を用いて診断を行う。

症例が与えられると、まず、経験知識ベースを参照し、これによる診断を試みる。これが成功した場合には、論理的知識ベースを参照しないために高速に解が求まる。経験知識ベースに無い症状の場合には、論理的知識ベースを使用して、診断を行う。

論理的知識ベースは、次章で述べるように対象装置の設計情報を記述したものであり、これは対象装置を構成する部品の接続関係を記述した構造知識と、各部品の正常な動作を記述した動作知識からなる。論理的診断方式は、対象装置の入出力のペアとして与えられた症状について、装置の本来の動作を考慮して、その症状が現れる原因となった可能性のある部品をリストアップする。また、これと同時に、経験知識ベースに蓄積するためのルールを生成する。このルールは、経験した症例について症状と障害箇所の一般的関係を記述したものである。論理的知識の範囲内で一般化して生成される。

### 5.1.3 設計知識の表現

設計知識の表現である論理的知識は、一階の述語論理式で表す。ここでは、図 1 に示す半加算器の例題を用いて説明する。

構造知識は、対象とする装置の構成部品とその接続関係を、以下のように表した

ものである。

Conn(In1, X(1)-In1).  
Conn(In2, X(1)-In2).  
Conn(X(1)-Out, X(2)-In1).  
Conn(Cin, X(2)-In2).  
Conn(X(2)-Out, Sum).  
...

部品の動作の知識は部品の入出力の関係を論理式で表したものである。例えば、ディレイの無い接続点の動作は、

$Val(q,v,t) \Leftarrow Conn(p,q) \wedge Val(p,v,t).$

のように表す。また、排他的論理和ゲートの動作は、

$Val(X(i)-Out,z,t) \Leftarrow$   
 $Val(X(i)-In1,x,t) \wedge Val(X(i)-In2,y,t)$   
 $\wedge XOR(x,y,z) \wedge Working(X(i),[x,y],z).$

XOR(0,0,0).

XOR(0,1,1).

XOR(1,0,1).

XOR(1,1,0).

のように表す。他の構成部品についても同様に表す。

#### 5.1.4 推論方式

モデルベース診断 モデルベース診断の推論方式として、仮説推論の一方式である Residue [Finger 85] 方式を用いている。この方式は、DART [Genesereth 84] に用いられている方式である。これは、ドメイン知識、ゴールおよび Assumable と指定された事実の集合の中から、ゴールを証明できる事実の集合 (Residue) を生成する方式である。

モデルベース診断の入力情報は、前述の構造知識と部品の動作の知識からなる設計知識  $D$  と、症状記述  $\langle I, \neg O \rangle$  である。ここで、 $I$  は入力ポートの値、 $O$  は装置が正常である時に出力ポートに出力されるべき値である。求める診断結果は、故障している可能性がある部品の動作状態を表す記述の集合  $W$  である。

Residue 推論方式を使用して、 $W$  を計算するためには、ドメイン知識に  $D$  と  $I$  を使用し、Assumable に構成部品の機能が正常に動作していることを示す記述 ( $Working(\dots)$ ) を指定し、ゴールを  $O$  とする。Residue 推論は、 $I \wedge D \wedge W \Rightarrow O$  が成り立つような  $W$  を計算する。これは、 $I \wedge \neg O \wedge D \Rightarrow \neg W$  に等価である。従って、設計知識  $D$  を正しいと仮定すると  $I \wedge \neg O \Rightarrow \neg W$  が成り立つ。

先の例題において、例えば、時刻 10 において入力 In1 が 0、In2 が 1、Cin が 1 で

ある時に、出力 Sum が正常出力 0 で無かったとする。この場合、症例記述の構成要素である  $I$  と  $O$  はそれぞれ、 $I: \text{Val}(\text{In1}, 0, 10) \wedge \text{Val}(\text{In2}, 1, 10) \wedge \text{Val}(\text{Cin}, 1, 10)$

$O: \text{Val}(\text{Sum}, 0, 10)$

である。Assumable は、

$\text{Working}(\langle \text{部品} \rangle, [\langle \text{入力値} \rangle, \dots], \langle \text{出力値} \rangle)$  の形式で与える。Residue 推論は、Assumable で指定された事実のうちで、ドメイン知識  $D$  と  $I$  と共にゴール  $O$  を証明することが出来る事実の集合  $W$  を求める。作成した実験プログラムでは、Prolog の深さ優先の探索戦略およびバックトラック機構をそのまま用いている。結果として、この例では、

$W: \text{Working}(X(1), [0, 1], 1) \wedge \text{Working}(X(2), [1, 1], 0)$  が求まる。

**経験知識の生成** 事例の一般化の実現方式として、本システムでは REBG (Residue with Explanation Based Generalization) という方式を開発して利用している。本方式は、Residue 方式に、演えき推論学習方式の EBG (Explanation Based Generalization) [Mitchell 86] 方式を融合したものである。

EBG 方式は、ゴール概念 (Goal Concept)、訓練事例 (Training Example)、ドメイン知識 (Domain Theory) および Operational Criteria と呼ばれる語彙が与えられた時に、まず訓練事例がゴール概念の一例であることをドメイン知識を用いて証明する。次にその証明木を用いて、ゴール概念を Operational Criteria で与えられた語彙を用いて書き替え、かつその時に変数の一般化を行う。このようにして、EBG は、ゴール概念を Operational Criteria を語彙に用いた一般的なルールとして書き替える。

前述の Residue 処理は、ゴールを説明できる様な Residue の集合を求める際に、説明木を作成している。従って、EBG 機能の融合のためには、Operational Criteria による書き替え処理と、変数の一般化の処理を追加すればよい。このようにして EBG 機能を追加した Residue 処理である REBG は、Residue を求めながら、同時に一般化されたルールを生成することができる。

前述の例題を REBG を用いて計算すると、同様の診断結果が求まると同時に、以下のようなルールが求まる。但し、Operational Criteria として入力値を表す記述 ( $\text{Val}(\text{In1}, \dots)$  など)、各部品の機能を表す記述 ( $\text{XOR}(\dots)$  など)、および正常性を表す記述 ( $\text{Working}(\dots)$ ) を指定した。

$\text{Val}(\text{Sum}, v1, t) \leftarrow$   
 $\text{Val}(\text{In1}, v4, t) \wedge \text{Val}(\text{In2}, v5, t) \wedge \text{Val}(\text{Cin}, v3, t) \wedge$   
 $\text{XOR}(v4, v5, v2) \wedge \text{XOR}(v2, v3, v1) \wedge$   
 $\text{Working}(X(1), [v4, v5], v2) \wedge$   
 $\text{Working}(X(2), [v2, v3], v1).$

| 例題 | モデルベース方式<br>による推論 (msec) |      | 学習したルールに<br>よる推論 (msec) |
|----|--------------------------|------|-------------------------|
|    | Residue                  | REBG |                         |
| A  | 27                       | 141  | 3                       |
| B  | 49                       | 243  | 5                       |
| C  | 34                       | 137  | 4                       |
| D  | 500                      | 1990 | 10                      |

表 1: 学習したルールによる実行速度の改善

このルールを、参照し易い形式で経験知識ベースに蓄積しておく。これにより、後にこのルールにマッチする症例があった場合には、モデルベース推論をせずに、簡単に解を求めることが出来る。

#### 5.1.5 実験結果

提案した方式に基づき、DEC-20 上の Prolog により実験プログラムを作成し、いくつかの例題について評価を行った。実行速度について評価を行った結果、表 1 に示すような良好な評価が得られた。Residue および REBG プログラムは、Prolog のメタインタプリタとして実現されており、実現方式は [Kedar-Cabelli 87] に述べられた方法を参考にしている。一列目の数字は EBG 機能の無い Residue 処理の実行に要した CPU 時間を示す。二列目は、Residue 処理を行うと同時に EBG 機能によりルール生成を行う REBG プログラムの実行時間を示す。三列目が学習したルールによる実行時間である。このように、この例題の場合は、EBG 機能を入れることによるオーバーヘッドは約 5 倍程度であり、ルールを蓄積することによる実行時間の短縮は約十分の一程度であった。

#### 5.1.6 おわりに

診断経験を一般化して蓄積することによって、同様の事例に対する処理を高速化する方式について報告した。この方式によって、経験した事例と似たような症例については、論理的診断を省略して短い処理時間で解を求めることが出来ることを実験結果を使って示した。

本方式では、経験知識は単に論理的知識ベースの内容の一部を言い替えただけのものであるために、高速化には貢献するものの、診断の質的向上には貢献しない。筆者らは本報告で述べた考え方を拡張し、経験知識をテストパターン生成の際に利用することにより、診断の処理の高速化だけでなく、故障箇所絞り込みのためのテスト数を少なくするなどの質的向上ができる方式を研究中である。これについては、テスト



ボタン生成を含んだ全体システムの構成について述べた報告 [Koseki 89] で述べる。

## 謝辞

本研究は、筆者が1987年度にStanford大学Computer Science学科に留学中に行われたものである。本研究の機会を与えて下さった、日本電気(株)石黒辰雄 所長、後藤敏 所長代理、永井義裕 部長、森啓 課長および、ご指導頂いたStanford大学Michael Genesereth 教授に感謝します。

## 参考文献

- [Finger 85] J. J. Finger and M. R. Genesereth, "A Deductive Approach to Design Synthesis," HPP-85-1, Stanford University Heuristic Programming Project, January, 1985.
- [Genesereth 84] M. R. Genesereth, "The Use of Design Descriptions in Automated Diagnosis," *Artificial Intelligence* 24, 1984, pp. 411-436.
- [Kedar-Cabelli 87] S. T. Kedar-Cabelli and L. T. McCarty, "Explanation-Based Generalization as Resolution Theorem Proving," *Proc. Fourth International Workshop on Machine Learning*, 1987, pp. 383-389.
- [Koseki 89] Y. Koseki, "Experience Learning in Model-Based Diagnostic Systems," *Proc. IJCAI-89*, 1989, pp. 1356-1362.
- [古関 89] 古関義幸, "モデルベース診断における経験的知識の学習," 第3回人工知能学会全国大会, 1989, pp. 235-238.
- [Mitchell 86] T. M. Mitchell, R. M. Keller, and S. T. Kedar-Cabelli, "Explanation-Based Generalization: A Unifying View," *Machine Learning* 1:1, 1986, pp. 47-80.
- [Roth 67] J. P. Roth, W. G. Buricuis, and P. R. Schneider, "Programmed algorithms to compute tests to detect and distinguish between failures in logic circuits," *IEEE Trans. Electron. Comput.*, vol. EC-16, no. 10, 1967, pp. 567-580.
- [Shortliffe 76] E. J. Shortliffe, *Computer Based Medical Consultations: MYCIN*, Elsevier, New York, 1976.
- [Singh 87] N. P. Singh, *An Artificial Intelligence Approach to Test Generation*, Norwell, MA: Kluwer Academic, 1987.

## 5.2 エピソードから一般化シナリオを構成する学習モデルと 事例からの知識獲得

京都大学工学部 榎木 哲夫

### 要旨

事例ベース内の組織化に関する所見として、Schank による「失敗による想起 (failure-driven reminding)」がある。この考え方を計算機に実装するためには、個々の事例について、なにが常識的で一般的であるのかについての世界モデルの構築機能、ならびにこのもとでの説明生成の機能が必要になる。本節では、事例内に含まれる部分的な情報から過去の既知事例を想起するための事例表現、事例からの一般化による対象モデル (=一般化シナリオ) の構成法、このモデル下での事例の評価法と事例からの知識獲得手法を提案する。

### 5.2.1 まえがき

事例ベース推論を進める上で、事例のインデキシング、組織化が重要な役割を果たすことはいうまでもない。従来の CBR システムの多くは、事例を [属性—値] の集合で記述しており、類似性評価は、これら特徴集合の重複の大小としてなされてきた。一方、事例の組織化の観点からは、Schank による failure-driven reminding の考え方、すなわち、事例は常識的な予測や期待感から逸脱する項目について組織化されており、このような特異項目を手掛かりに想起される、という考え方、が提起されており、特徴集合として表された事例間の類似性に基づく組織化 (generalization-based memory) と対峙する興味深いものである [Schank, 82]。このような事例想起のメカニズムを計算機に実装するためには、従来のように事例をつぎつぎに蓄積しておけば良いというものではなく、個々の事例を、なにが常識的で何が特異的であるのかを、演繹的に説明付けて解析し、その結果を手掛かりとして組織化していくことが必要になる。

このような事例の演繹の説明としては、事例からの概念学習の手法である「説明に基づく学習 (Explanation-Based Learning)」があるが、領域定説とでもいうべき domain theory の位置づけや境界を明確に定義できるものではない。また、ここでの事例の果たす役割は、あくまで目標概念 (goal concept) が与えたもとで、領域定説による説明付けをまっとう受動的な意味しかもたないが、本来は、事例に含まれる情報の一部が事例背後の存在する対象モデルを喚起し、このモデルに基づく解析を通じて、事例の持つ固有性や特異性が認識されて、既存の知識集合に組み入れられていくものである。このような対象モデルは、事例のおかれた世界において、何が常識的であり一般的であるのかを規定する世界モデルと考えることができるが、従来の領域定説のように、予め一般化された形で用意されているというよりは、何か問題が与えられたときに、その問題に関連する既知事例だけを集めてきて一般化して利用する学習形態 (Case-Base

d Learning) により、適宜自律的に生成されるものとするのが妥当である。

本節では、事例内に含まれる部分的な情報から過去の既知事例を想起するための事例表現、事例からの一般化による対象モデル（＝一般化シナリオ）の構成法、このモデル下での事例の評価法について、本報告書 4.5 で述べた心理学的所見に基づく実現法について報告、事例からの知識獲得手法を提案する[橋本他、89]。

### 5.2.2 エピソード記憶の適応的組織化による説明学習

学習の定義としては、これまでさまざまになされてきているが、本研究では、Doyle の [Doyle, 88] に従い、

「学習者がメンタルに捉えるところの状態や操作を適応的かつ合理的に変更しながら経験を解釈していくための認知的技能」

と定義する。このような認知システムとして「学習」を捉えることにより、学習システムには以下のような要件が必要となる。

#### (1) 学習目標概念の自己生成

人間の学習プロセスをモデル化する上で、学習者のおかれたコンテキストや学習ニーズ・目標を抜きに考えることはできない。ある特定の事例が専門家により訓練事例として与えられる背景には、そこに既存の事例知識へのダイレクトな同化では説明づけられないような新たな知識が具体化されて潜在しているという判断が存在する。このような専門家に意図された学習目標を、EBLのように外部から陽に教えられることなく、学習システム自らがその知識に基づいて見いだすことが不可欠となる。

#### (2) モデルに駆動された能動的な事例解釈

従来のEBLは、訓練事例を一方向的に受け取って適用可能な領域知識を当てはめ分析するといった受動的な事例処理である。これに対し認知システムでは、与えられた事例記述の一部をトリガーとして、過去の経験の蓄積からの帰納的に導出される「世界モデル」を、事例背後に存在するコンテキストとして自身の内に構築する。そしてこのモデルからのトップダウンの事例解釈により、与えられた事例に含まれる文脈依存要素を的確に認識して切り出し、その間の因果性を説明づけるといった、双方向かつ能動的な解釈プロセスが必要になる。

#### (2) 注視点の変更によるモデルの適応的組織化

このようなモデル駆動の解釈は、同時に、常套的な事象進展から逸脱するような特異事象の認識をも可能にする。このとき、何故にこのような特異事象が生じたのかについての新たな学習目標を見だし、適応的に過去事例の検索時の注視点を切り替えてモデルの再編を行う。そしてこのモデルに基づいて事例を再解釈し合理的な説明づけを行った上で、後に利用可能な形で一般的な知識として抽出・格納することになる。

一方、マネジメントの分野においては、企業トップの政策レベルでの決定判断を支援する知識システム、意思決定支援システム(DSS)の開発が要請されている。このような問題領域においては、従来のエキスパートシステムのように、与えられた問題状況に対し厳密な対象モ

デルや普遍的な一般則からの演繹による分析のみを行っているわけではなく、「過去に類似した前例を熟知しているから」という経験的模倣に従った直観的な理解がよりどころとなっている。またこのような実例は通常、状況のさまざまな局面や有意性の判然としない情報を含むものであるが、このような中から専門家は固有の意味・ノウハウを抽出している。

本研究では、このような観点から、図5.2.1. に示すように、事象スキーマの系列として表現されたエピソード群を領域知識として有し、これらから帰納的に生成される世界モデルであるスクリプトに基づいて、新たに与えられた訓練事例としての学習エピソードを説明づけ、新たな学習概念を獲得学習するシステムについて報告する。ここでのモデル駆動の解釈プロセスはファジィ推論により定式化され、獲得される知識はメンバーシップ値を伴ったファジィプロダクションルールとして出力されることになる。

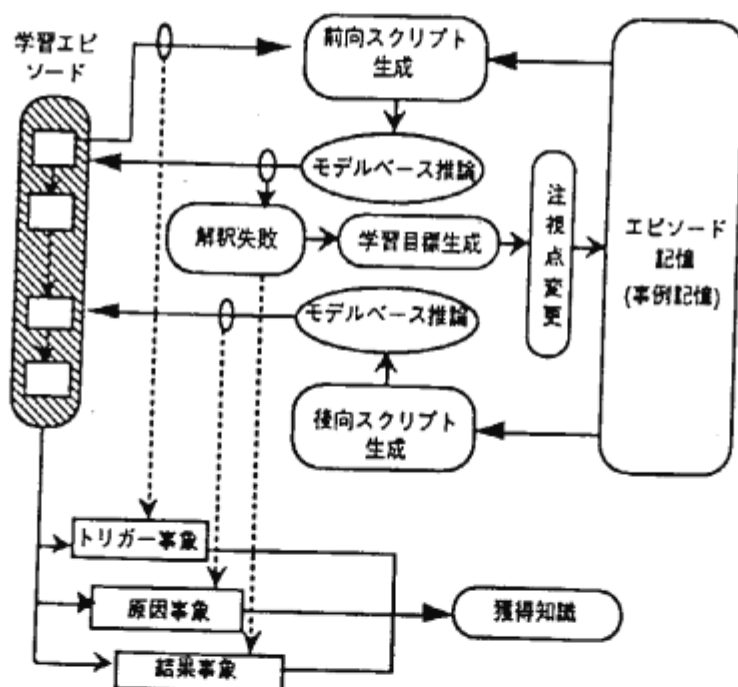


図5.2.1 事例記憶に基づく説明学習システム

### 5.2.3. エピソードの事象スキーマ系列表現とスクリプトの組織化

我々は既に、オビニオン・リーダーや政策決定者の論説記事に記載されたエピソード記述を、これに関与する各種団体個人等のアクターの決定・価値効用を最小構文単位とする因果関係ネットワークとしてコーディングする表現手法を提案している。またこのような構文単位の組合せからあらゆる普遍的な社会行動・事象概念を記述するための少数のプリミティブ（基本事象）が定義でき、さらにこれらの定型化された基本事象の系列からより複合された事象概念（複合事象）が階層的に集約・構造化されるような「閉じた表現系」が構成できることを示した[榎木他、87; 榎木、88; Sawaragi et al., 87]。同システムでは、個々のエピソードが事象スキーマの系列として組織化されてエピソード記憶を構成している。図5.2.2. に、事象スキーマ系列に変換された「日米経済摩擦問題」に関するエピソードの例を示す。

本システムは、記憶内の類似したエピソードから、これらの原型パターンを、一般化された事象概念の系列、すなわち世界モデル＝スクリプトとして活性化する機能を有している。

一般化事象スキーマの生成は、類似するエピソード群から、それらを包含するような一般的な記述を創り出すことを意味する。各事象スキーマは、図5.2.2. に示すようにスロット：event-type、actor、content の値によって諸属性が与えられている。現実世界で例えば図5.2.3.(a) に示すような事象  $e_1$ （トリガー事象）が観測されたとき、システムはこれに類似するエピソード記憶内の事象スキーマ（ここでは同じ event-type をもつもの）の集合  $\Omega(0)$  が検索される。システムは同図(b) に示すように、スロットの値に一致するものについてはその値を一般化事象スキーマにそのまま伝達し、相違するものについてのみシステム内に内蔵する各属性値の上位下位関係知識を参照することにより、両者の上位概念で置き換える最小一般化を繰り返し行う。ただし、いたずらに未観測の数多くの事象を包含してしまうような過剰な一般化を防ぐための制約が設けられているため、構成される一般化スキーマ階層は、ボトムレベルの具体事象群の意味的な類同性に基づいて決定される。すなわち、例外的な事象は周囲の事例と結び付くことなく具体レベルに留まるが、周囲に類似する事象の多いスキーマほど比較的意味的にかけ離れたようなスキーマをも吸収統合して上位複数レベルで一般化スキーマを構成しながら成長することになる。

同図(c) に示すように、 $E_{1,1}^{(0)}$  に一般化された各基本事象  $e_1, e_2, e_3$  に後続するエピソード記憶内の事象集合  $\Omega(0)$  のうちその過半数が  $E_{1,1}^{(0)}$  と同じレベルにおいて  $E_{1,1}^{(1)}$  あるいは  $E_{2,1}^{(1)}$  に一般化されるとき、この一般化レベルにおける事象推移パターン： $E_{1,1}^{(0)} \rightarrow E_{1,1}^{(1)}$  および  $E_{1,1}^{(0)} \rightarrow E_{2,1}^{(1)}$  が、定型的パターンとして高い再現性をもつような原型スクリプトとして生成される。上記一般化スクリプトは、さらに後続の事象についても同様の手続きに従って構成され、 $e_1$  のもとで予期されるシナリオが途切れるまで、すなわち後続の事象集合上に過半数を包含するような一般化スキーマが生成されなくなるまで続けられる。また上記一般化アルゴリズムによるスクリプトは、 $e_1$  上に構成された全ての一般化事象スキーマの各一般化レベル（ $E_{1,1}^{(0)}$  および  $E_{1,2}^{(0)}$ ）において並列的に活性化される。

またこのような事象スキーマの一般化を、同図(d)に示すように  $\Omega(0)$  に先行する事象スキーマ

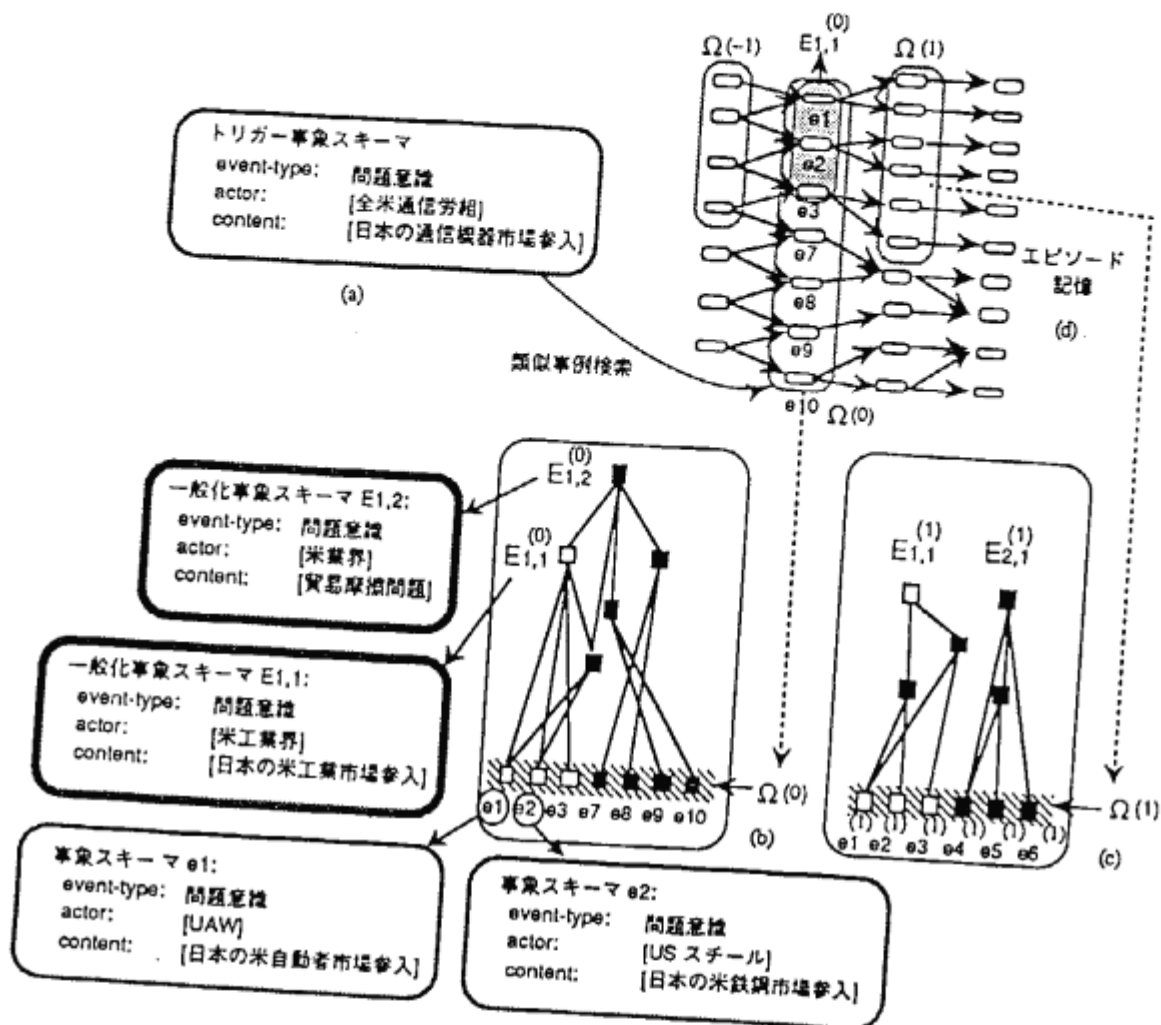


図5.2.3 エピソード記憶の活性化と一般化事象スキーマの組織化

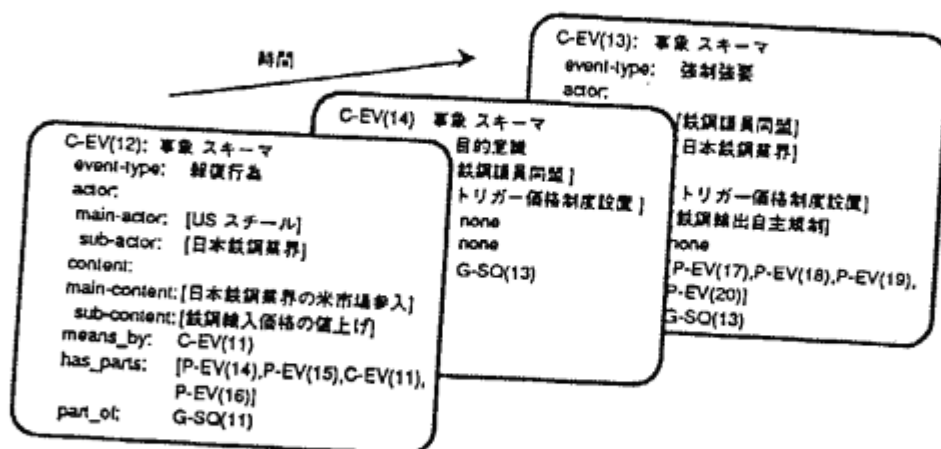


図5.2.2 事象スキーマ系列としてのエピソード例

マ集合  $\Omega(-1)$  に対して行うことにより、 $e_i$  の生起を理由づけるシナリオ（後向きスクリプト）が構成される。

#### 5.2.4. スクリプトに基づくモデル・ベース推論

前章で構成されたスクリプトは、部分的な観測情報が得られた際に常套的な事象の展開を述べたの一般化された世界モデルである。しかしながら、このモデルから演繹的に導出される具体事例の解釈や予測における真理性は、モデルがエピソードから帰納的に導出されたものであるため必ずしも保証されるものではない。本システムでは、このようなモデル駆動推論のプロセスをファジィ推論により以下のように定式化している。

このような演繹における conformity の度合は、それがベースになっている具体エピソード集合の意味的な相関の強さ・冗長性に依存して決定される。そこで、各々の一般化事象スキーマの原型（プロトタイプ）を、これを構成する事象スキーマから以下のように定義する。

〔定義1〕事象スキーマ  $e_i$  のスロットに埋められた値ならびにその上位概念の値を含めた全属性の集合を  $e_i$  の属性リスト  $FD(e_i)$  とする。

〔定義2〕一般化事象スキーマ  $E_{j,k}^{(n)}$  を構成する全事象スキーマ  $e_i$  ( $e_i \in E_{j,k}^{(n)}$ ) の過半数以上に共有される属性リスト要素の集合を  $PD(E_{j,k}^{(n)})$  とする。

〔定義3〕事象スキーマ  $e_i$  の一般化事象スキーマ  $E_{j,k}^{(n)}$  のもとの典型性  $typ(e_i | E_{j,k}^{(n)})$  を、

$$typ(e_i | E_{j,k}^{(n)}) = \frac{|FD(e_i) \cap PD(E_{j,k}^{(n)})|}{|FD(e_i) \cup PD(E_{j,k}^{(n)})|} \quad \#$$

として定義する ( $0 < typ \leq 1.0$ )。■

すなわち任意の事象スキーマ  $e_i$  を一般化事象スキーマ  $E_{j,k}^{(n)}$  から演繹する際の conformity は、定義される概念帰属度（典型性、typicality）として評価される。同様の考え方は、コネクションニズムの観点からもインプリメンテーションが可能である[Sawaragi, et al., 90]。

図5.2.4. は図5.2.3.(b) の7つの事象スキーマについて、その上位に構成された2つの一般化事象スキーマ下での典型性を算出、表示したものである。同図が示すように、互いに意味的に類似した事例同士から構成されている概念ほど、その帰属度分布はクリスプ集合に近いものになるのに対し、より冗長度の弱い上位概念のもとではファジィ性が強くなることになる。以上のように一般化事象スキーマを、事例集合上でのファジィ集合として定義すると、これらの因果系列としてのスクリプト下での演繹は、図5.2.5に示すようなファジィ推論の連鎖として定式化される。すなわち、トリガー事象  $e_t$  の後に生起すると予測される事象概念は同図斜線部のように活性化されることから、ある具体事象  $e_p^{(1)}$  の生起を予測する際の conformity は、

$$\min \{typ(e_t | E_{j,k}^{(0)}), typ(e_p^{(1)} | E_{j,k}^{(1)})\}$$

で与えられる（[Mamdani, 1974] のファジィ推論則による）。さらに、連鎖的な事象（シナリオ） $H(q) (= e_t \rightarrow e_p^{(1)} \rightarrow \dots \rightarrow e_p^{(n)})$  の スクリプト  $S_{j,k} (= E_{j,k}^{(0)} \rightarrow E_{j,k}^{(1)} \rightarrow \dots \rightarrow E_{j,k}^{(n)})$  の下での評価は、

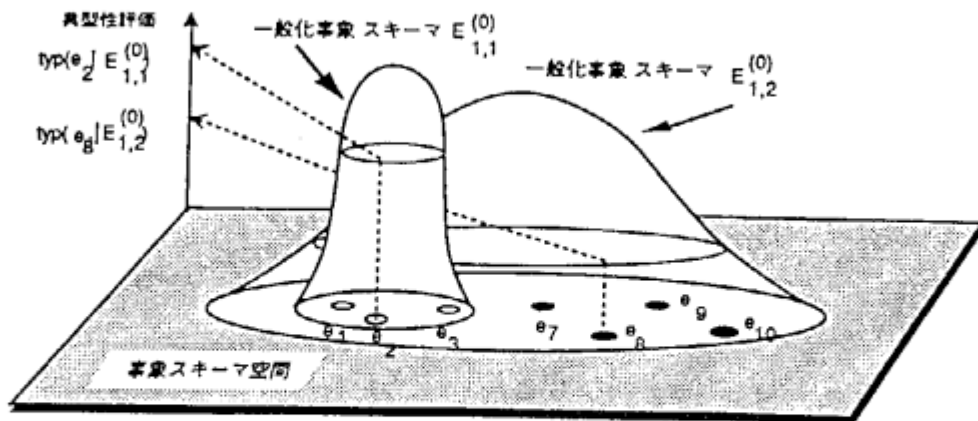


図5.2.4 一般化事象スキーマ下での事象の典型性評価

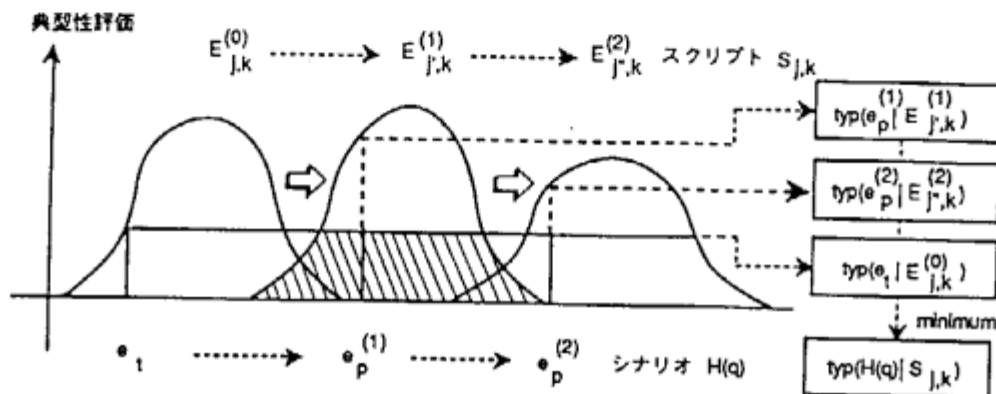


図5.2.5 スクリプト下でのファジィ推論

$$\text{typ}(H(q) | S_{j,k}) = \min \{ \text{typ}(e_i | E_{j,k}^{(0)}), \text{typ}(e_p^{(1)} | E_{j,k}^{(1)}), \dots, \text{typ}(e_p^{(q)} | E_{j,k}^{(q)}) \}$$

となり、シナリオ全体の評価は最も活性レベルの弱い部分に拘束されることになる。また並列スクリプトのもとでの評価  $\text{typ}(H(q))$  は、

$\text{typ}(H(q)) = \max \text{typ}(H(q) | S_{j,k})$  (但し  $S_{j,k}$  は  $e_i$  の下で活性化されるすべてのスクリプト) で最も高い照合を呈するスクリプト下での評価が当てられることになる。

以上のアルゴリズムでは  $e_i$  が与えられた時点で活性化されるスクリプトによりその後の事象解釈における文脈 (コンテキスト) ・視焦点が拘束されるため、文脈抜きのルールの連鎖を扱う際に生起する通常のファジィ推論における「曖昧さの爆発」は回避されることになる。

### 5.2.5. エピソード記憶からの学習目標の自己生成と説明に基づく知識学習

専門家がある学習事例を訓練事例として与える状況を考えると、その事例が内包している学習目標概念が漠然とは専門家の意識下にはあるものの、それを言語表現したりルール記述するまでには具体化されておらず、むしろ実例を通してしかそのような概念を伝えられない場合が多いと考えられる。このような状況下での知識獲得システムの果たす役割は、何故にその実



例が専門家により訓練事例として認識されたかを説明づけることによって、事例に含まれる個々の状況依存要素を的確に抽出し、専門家の漠然たる直感を、後の知識システムに直ちに実装可能な操作性の高い記述でかつ他の類似事例にも適用可能な十分一般的な記述に変換処理することにある。

本研究では、このような専門家に意図された学習目標を、訓練事例を構成する事象の中で、過去事例とのアナロジーで常套的に予測される事象展開の流れから逸脱するような特異事象との関連で捉える。すなわち、前章で述べたスクリプトの組織化ならびにそのもとで定式化したモデルベース推論を訓練事例に対して適用することによって、典型性評価が極端に落ちるような事象を同定することが可能になるが、この特異事象生起の理由づけとなるような事象が該当事例の内部に見いだすことができれば、専門家の訓練事例に託された学習目標が説明づけられたと考える。

図5.2.6.(a) は、訓練事例として与えられた以下に示すようなエピソード：

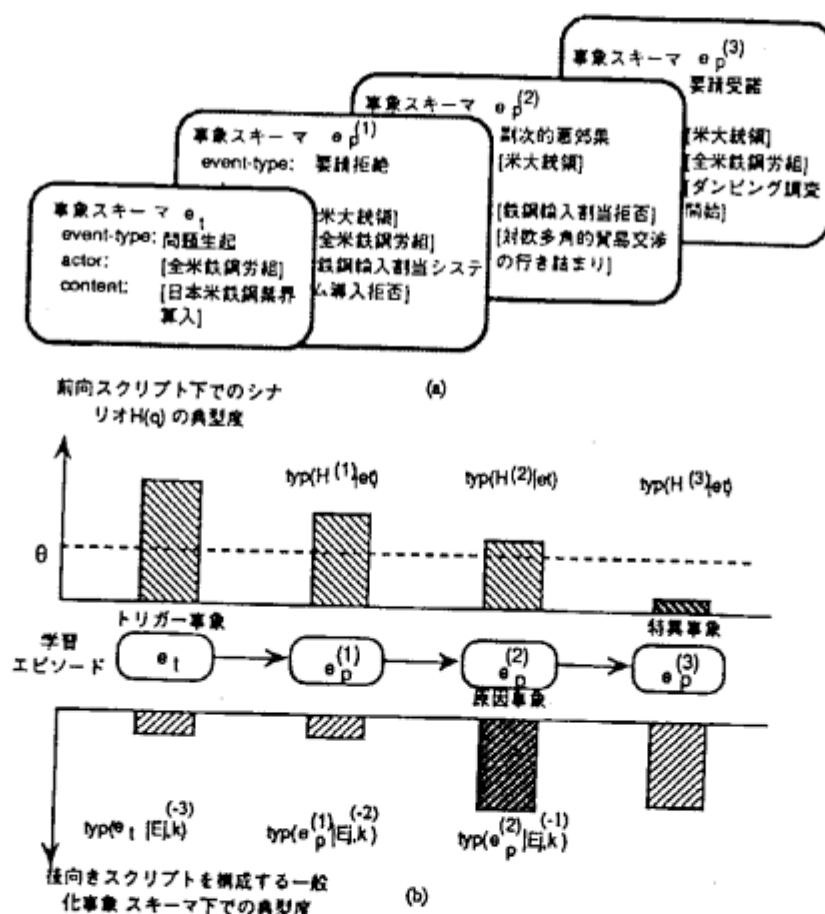


図5.2.6 学習エピソードからの知識獲得

「日本からの鉄鋼輸入の増大に苦慮した全米鉄鋼労組は、大統領に対し鉄鋼輸入の割当制度の設置を要請したが、大統領はこれを却下した。このことが欧州諸国との多角的貿易交渉を推し進める上での労組の反発を食らうことになった。その後全米鉄鋼労組は、日本製鉄鋼に対するダンピング調査の開始を大統領に要請し、大統領はこれを承諾した」

を事象スキーマの系列として表現したものである。システムは事例説明にあたり、まず訓練事例エピソードの先頭事象をトリガー事象として、自身の有するエピソード記憶に基づき、その後の予測される事象展開を前向きスクリプトとして並列に活性化する。そして各々のスクリプトのもとで、後続事象を逐一付加して構成されるシナリオ  $H(q)$  の典型性評価を算出する。すると上述のような特異事象は、同図(b)に示すように、それを付加したが故に、あるしきい値以下の典型性に落ち込むような事象として認識される。本例の場合、トリガー事象により活性化されているスクリプトが自由貿易主義的シナリオであるため  $ep(3)$  の「大統領による保護貿易措置の受諾」が特異事象として認識される。このような特異事象を認識したなら、システムはその理由づけを先行事象系列の中に見いだすべく、図5.2.7.に示すように、この事象を起点として後向きにスクリプトを活性化、そのもとで訓練エピソード内の各先行事象の典型性評価を実行、最も高い典型性を呈するものが特異事象の生起を理由づける妥当な原因事象として決定されることになる。

このように、システムはスクリプトのベースとなるエピソード検索の注視点を随時変更しながら、与えられた訓練事例の事象系列から、状況設定要素（トリガー事象）、特異的な結末（特異事象）に至らしめる原因要素（原因事象）を選択的に特定し、

「米産業界が日本との貿易摩擦問題で憂慮しているとき（問題生起：トリガー事象）、もし米内閣がその自由貿易政策の故に対段貿易交渉に行き詰まりを見せるようなことがあれば（副次的悪効果：原因事象）、米内閣は保護貿易政策を受け入れる（要請受諾：特異事象）」

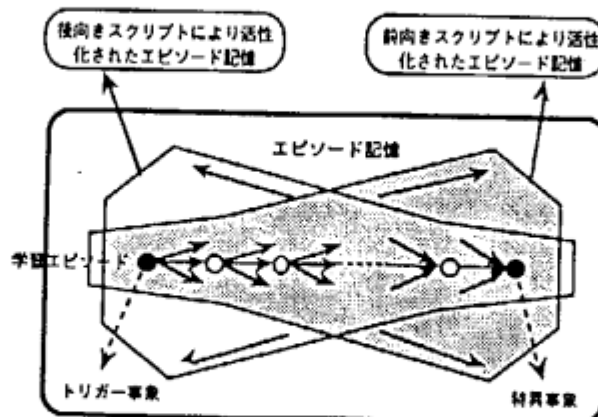


図5.2.7 エピソード記憶の活性化によるスクリプト生成

のようなコンパイル知識として抽出獲得することになる。ここで抽出された各事象の表現には、用いられたスクリプトを構成する一般化事象スキーマの表現が当てられる。従って、その一般化のレベルは、従来のEBLにみられる定数の変数化による画一的な一般化に代わり、最も予測性の高い一般化レベルが選択される。これらの各概念は任意の事例に対する帰属度分布が既知のコンパイルされたファジィ推論ルールが得られることになり、意思決定領域で多用されているファジィ・プロダクションシステム構築に際して十分操作性の高い知識が出力される。

#### 5.2.6. あとがき

本来「学習」は、人間に備わった認知システムのメカニズムと不可分の関係にあり、そこで獲得される知識の内容も、これを用いる問題解決システムと切り放して考えることはできない。このような観点から本論文では、事例記憶に基づく説明学習を提案し、そこで必要となるエピソード記憶からのスクリプトの活性化とそのモデルベース推論のファジィ定式化を中心に報告した。本論文で提唱した手法は、CBRの事例検索時における大局的な文脈情報を提供するばかりでなく、説明生成の「行き詰まり (impasse)」をインデックスとする事例想起のメカニズムを実現する上で有用な示唆を与えるものと考えている。

#### 【参考文献】

- [Schank, 82] R. Schank: Dynamic Memory: A Theory of Reminding and Learning in Computers and People, Cambridge Univ. Press, New York (1982)
- [榎木他、89] 榎木、片井、岩井：事例説明に基づく学習目標概念の自己生成と知識獲得、計測自動制御学会第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予稿、札幌、(1989).
- [Mitchell et al., 86] T.M. Mitchell et al.: Explanation-Based Generalization: a unifying view, Machine Learning, 1, 47/80 (1986)
- [Doyle, 88] J. Doyle: On Operationality and Learning, Carnegie Mellon Univ., CMU-CS-88-122, (1988)
- [榎木、88] 榎木：学習における最近の話題、計測と制御、27-10, pp. 931-934, (1988).
- [榎木他、87] 榎木、岩井、片井：社会現象の深層構造把握のための因果連鎖知識の多重構造化、計測自動制御学会論文集、23, 977/984 (1987)
- [Sawaragi, et al., 87] Sawaragi, T. and S. Iwai : Cognitive simulation for intelligent decision support. In W.R. Ras and Zemankova (Eds.), Methodologies for intelligent systems. Elsevier Science Publ., New York. 362-369.(1987)
- [Sawaragi, et al., 90] T. Sawaragi, S. Iwai and O. Katai : Dynamic Organization of Fuzzy Control Rules Based on Association Networks, Proc. of NAFIPS'90 in Toronto, (1990).
- [Mamdani, 74] J.E.H. Mamdani: Application of Fuzzy Algorithms for Control of Simple Dynamic Plant, Proc. of IEEE121, 1585/1588 (1974)

### 5. 3 事例に基づく組立経験知識の学習モデル

大阪大学産業科学研究所      安部 憲広

#### 要旨

機械の設計や組立問題を解決しようとする際に問題となる点の一つとして、形状のもつ複雑さがある。実際に機能する機械部品を組み立てる際には、形状だけではなく、どのような順序で機械要素を組み立てて行くかも考慮する必要がある為、組立知識をいわゆるルールで記述することは困難である。本研究では、正しいけれども不完全な組立法から正しい組立法を求める過程を利用して、以後同様な組立指示に対して失敗を繰り返さぬよう、経験事例を活用するシステムにつき報告する。

#### 5.3.1. はじめに

我々のグループでは、ロボットへ作業指令を与える場合に、ユーザの負担を軽減するためにユーザが自然言語により概略的な手順を指示するだけでロボットが自動的に詳細な手順を生成し、目標を達成する方法について研究してきた[安部 88]。システムは組立に関する初歩的な知識を持っているとしているが、その知識は常に正しいとは限らない。そこで誤った組立を行なった場合に、その原因を究明しておかないと類似した組立で再び同じ失敗を繰り返してしまう。

そこで本研究では、失敗の原因を解析し、誤りを解消させるため、組立過程における依存関係の管理にTMS [Doyle 79]を用い、組立に失敗した場合には、TMSのDDB機能によりその原因を導出し、仮説を修正することにより組立の再計画を行なう。さらに失敗時の状態を要約したインデックスとともに成功時の組立法を記録しておくことにより、同じような組立に出会った場合に、以前に用いた組立法を想起してそれを適用することで、同じ誤りを繰り返すことなく組立を行なう。これにより非常に効率のよい機械組立が可能となる。

#### 5.3.2. システム概要

本研究のシステム構成を図5.3-1に示す。入力文(組立説明文)は、まず言語解析部において構文解析・意味解析され、組立動作を表わす動作フレームに変換される。この段階で

は動作を行なうために必要な情報が完全には満たされていないので、機械の組立に関する知識を用いて欠けていた情報を補う。それでも動作フレームが一意には決定されない場合には、デフォルトを用いて動作フレームを組立操作仮説として1つ生成する。そのようにして得られた動作フレームにより、組立動作のシミュレーションを行なう。また、各組立ステップにおいて組立に関する因果関係を記録する。因果関係の記録はTMSにより管理されており、組立状態は組立操作ノードと各部品の三次元位置データノードで表わされている。生成された組立品は組立品評価部において検証される。検証の結果、誤った組立と判断された場合は、

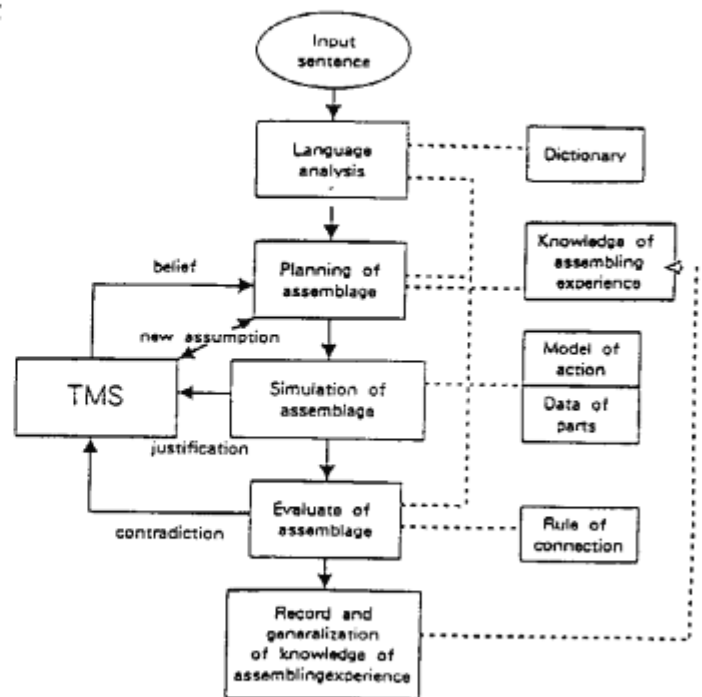


図5.3-1 システム構成

そのことをTMSに通知し（contradictionノードの登録）、TMSはそのDDB機能により誤りの原因と考えられる組立操作仮説を導く。そして導き出された組立操作仮説に対し、その操作の対象部品の接触関係を変化させるように代替仮説を生成する。代替仮説を生成すると、その前の仮説及びそれに依存している組立状態が正しくないものとされ、現在正しい組立状態を表わすデータフレームにより現在状態を表わし直し、その代替仮説のもとに組立の再計画を行なう。

また、誤りの原因とみなされた組立操作仮説が対象としている部品の機能、その部品に対する組立操作及びその時のターゲット状態をインデックスとしてその組立操作仮説に対する修正戦略、すなわち組立法を記録する。将来の組立において、過去に同じような組立に出会ったことが組立経験知識のインデックスにより想起されると、その時の組立法を用いて、同じ誤りを繰り返すことなく組立を行なう。

## 5.3.2. 組立の再計画

### 5.3.2.1 依存関係の記録

誤った組立を行なった場合に、誤りに関係のある部分だけに注目し組立の再計画を行なう場合には、TMSのDDB機能が必要である。そして重要となるのは、各組立ステップにおいて記録していく組立に関する理由付けである。誤った組立を行なった場合に、その誤りの原因と考えられる組立操作仮説を効率良く導くためには、一連の組立過程においてははっきりとした因果関係が記録されていなければならない。そこでその因果関係をどのよ

うに記録していくかについて考える。

本研究においては言語解析の結果、組立動作（insert など）、オブジェクト・ターゲット（部品中の基本立体〔山田 86〕）は正確に同定できると仮定している。しかし、組立動作に対する操作量は明示されていない場合が多いと考え、その場合に組立操作量をデフォルトで定めた値に設定して組立操作仮説とする。組立操作が決定されると、それをもとに組立シミュレーションを行なうが、その時のターゲット状態によって組み立てられた部品間の状態が異なる。そこで、ある部品が現在サブアセンブリ中のある空間位置を占めるようになった理由は、ある組立操作を施されたことと、その時にあるターゲット状態が存在していたことである。

実際に記録したノードは、組立操作（action）ノードと、部品の三次元位置データ（location）ノードである。組立操作ノード、三次元位置データノードは、それぞれ動作フレーム、部品の三次元データフレームに関連付けられている。

### 5.3.2.2 検証と矛盾候補仮説の導出

組立操作が正しく行なわれたかどうかを確認するために検証が必要であるが、先だって行なわれた研究により、検証に用いる2つの概念として、部品の機能クラスとその機能クラス間に必要な接続関係を定義した〔佐古 88〕。



図5.3-2 部品の機能クラス

| Dependency relation                     | Necessary relation         |
|-----------------------------------------|----------------------------|
| ① Substep                               |                            |
| (a) screw ⇔ clearance hole              |                            |
| \$ .....                                | rigid                      |
| (b) setscrew ⇔ solid                    |                            |
| \$ .....                                | rigid                      |
| (c) shaft ⇔ coupling .....              | rigid                      |
| (d) motor ⇔ motor shaft .....           | rotate                     |
| ② All step                              |                            |
| (e) shaft ⇔ bearing .....               | cylindrical, rotate        |
| (f) teeth of gear ⇔ teeth of gear ..... | gear                       |
|                                         | rigid - Freedom 0          |
|                                         | rotate, gear : - Freedom 1 |
|                                         | cylindrical - Freedom 2    |

図5.3-3 必要な接続関係

図5.3-2に示した機能クラス間で満たされなければならない接続関係と、現在生成されている部品間の接続関係とを比較して、一致しない場合は組立が誤りであると判断する。検証により組立が誤りであると判断された場合は、その誤りの原因を突き止め修正するためにTMSにデータ間に矛盾が存在することを示すcontradictionノードを適切な理由付けをつけて登録する。必要な接続関係が満たされていないことをTMSノード

として登録されている部品の三次元位置データノードとの関係で考えてみると、部品間に必要な接触関係が満たされていないということは、それらの部品がサブアセンブリ中で不適切な空間的位置の関係にあるのが原因であると考えることができる。よって *contradiction* ノードの理由付けは必要な接触関係を満たしていない機能クラスの部品群の三次元位置データノードを *In* リストに持つ *SL* 型理由付けとなる。

### 5.3.2.3 代替仮説の生成

組立に失敗した時、その原因と考えられる組立操作仮説を導出すると、次にそれをどのように修正するかが問題となる。本研究では、その組立操作の対象部品についての、他の部品との接触関係に基づいて代替仮説を生成する。組立の操作量は一般に曖昧であると仮定しているため、誤った操作量の組立操作仮説を生成することにより、本来存在すべきでない空間をある部品が占有したり、その逆が起こり、望ましくない部品間の接触関係が生成されるのである。以上の理由により、組立誤りの原因の候補仮説の接触関係の1つを除去することにより、代替仮説を生成する。

接触関係の除去方法については、接触のタイプを分類し、そのタイプごとの接触関係の除去戦略が知識として与えられている。本研究においては組み立てられる部品の軸どおしの関係は平行か垂直と仮定し、また基本立体は円柱のみに限って部品を表現しているので、接触のタイプは4通りとなる。

図5.3-4において部品Aを組立失敗の原因の候補仮説と考えた場合に、部品Bとの接触関係の除去・代替仮説の生成は以下になる。

#### 1) Type A, B の場合

この組立操作仮説の操作量をデフォルト値だけ減じた操作量を持つ仮説を代替仮説とする。

#### 2) Type C, D の場合

この組立操作仮説の操作量を

$$\alpha = (z - Z) + R$$

だけ減じた操作量を持つ仮説を代替仮説とする。

以上より誤った組立状態を解消し、新たな仮説のもとに再計画を行うことになるが、その前にサブアセンブリの状態を修正しなければなら

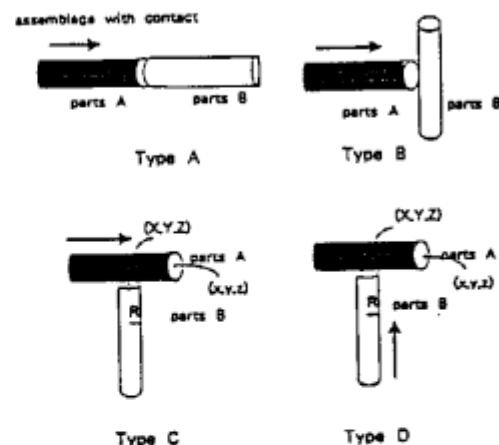


図5.3-4 代替仮説

ない。新たな仮説を生成した結果、*out* になった部品の三次元位置データノードはそのノードが表わす三次元位置が正しくなく、その部品に対する組立はやり直される。

### 5.3.4 組立法の学習

#### 5.3.4.1 組立経験知識の学習

ある組立を行なう時に、それが以前であった組立例に似ている場合は、前に成功した時の組立法を用いることによりうまくいく可能性が高い。そこで部品の大さきなどには依存せずに常に同じ組立状態が得られるように記述された組立法を記録しておかなければならない。本研究の場合、接触関係の除去により組立誤りの解消を図るわけであるから、組立経験の結果、ある部品とある部品のある接触タイプによる接触関係が生じてはならないという知識が得られる。そこでその一方の部品に対して組立操作を施す場合に、もう一方の部品との間に望ましくないタイプの接触関係を生じないように組立操作量を決定する方法というのが記録すべき組立法、すなわち組立経験知識である。

以上より、ある部品に対する組立操作に対して、

i) 生じてはならない接触関係のタイプ

i i) その接触関係におけるもう一方の部品の機能

を記録しておくことにより、実際にその部品に対する組立操作を行なう時に、デフォルトによる操作量  $s$  を求め、生じてはならない接触関係が生じないような修正値  $\alpha$  を接触関係の除去に関する知識より求め、 $s$  から  $\alpha$  を減じた値  $x$  を実際の組立操作量とすることにより、以前成功した組立例の場合と同じ組立状態が得られる。

次に組立経験知識に対して、どのような組立の時にその組立法を適用することができるかを示すインデキシングの問題について考える。組立過程において、その組立に関する理由付けを記録していく際の方法として、ある部品に何らかの組立操作を施す場合に、その操作を施すことが可能となっているターゲット状態が、その時点で存在することが、その操作を施した後の部品の空間的位置に対する理由付けとなっている。そこで、ある組立操作仮説が誤りであることが判明した場合に、その誤りを導く可能性のある特徴的な状態というのは、その操作を施すことが可能となっている状態、すなわち、ターゲット状態であると考えられる。よって、同じ組立部品の機能、同じ組立動作、そして同じターゲット状態に出会った時が、まさに過去に獲得した組立法を想起すべき時である。従って記録すべき組立経験知識は、次のような形となる。

組立経験知識

- ・インデックス：組立部品の機能、組立動作、ターゲット状態
- ・組立法：生じてはならない接触関係のタイプ、接触対象部品の機能

#### 5.3.4.2 組立経験知識の適用

経験の結果得た知識の学習後、同じ組立動作を、同じ機能を持った部品に、同じターゲット状態に対して行なう組立に出会った場合に、過去に成功した時の組立法を想起して同



じ方法を用いて行なえばよい。ところが、望ましくない接触関係を構成する可能性のあるもう一方の部品が現在の組立品中に存在している場合と存在していない場合とで、過去に獲得した組立経験知識が直接的に適用できるかどうかが決定される。もう一方の部品が存在しない場合は、存在していない部品との接触関係を避けるように現在の組立を行なわなければならない。すなわち、実際には存在していない部品の存在を仮定しなければならない。将来、その部品についての組立が行なわれ、その部品が占めるべき空間を推測しなければならない。すなわち”組立予測”が必要となる。

ある部品に対する組立操作が将来行なわれて、それが現在の組立品中のどのような空間を占めるようになるかを推測する場合に、手がかりとなるのは、その組立操作に対するターゲット、すなわち、－基本立体が存在するということが必要であるということである。

逆に、そのようなターゲット（－基本立体）が同定できた時には、通常、そのターゲットとそれに対して組立操作を施される部品（あるいは部品の一部、＋基本立体）の形状は、ほぼ同じと考えられるので、その部品との接触関係を回避するような操作量の計算に、そのターゲット（－基本立体）に関する数値データを用いて行なうことができる。

そこで、将来の組立に対するターゲットをどのように同定するかということが問題になるが、ここでは3つの基準によりそれを行なう。

（i）－基本立体であること。

（i i）現在の組立対象部品の方向に対して、避けるべき接触関係のタイプを構成するような方向を持つこと。

（i i i）現在の組立対象部品に対する組立動作における移動空間と交わりを持つこと。

#### 5.3.4.3 組立経験知識の一般化

組立経験を積み重ねていくにつれて、獲得した知識も増大していくが、新たな組立ステップに進むごとに、常に現在の組立状態と組立経験知識のインデックスを比較しなければならないため、知識の量が多ければ多いほど、その比較にコストがかかることになる。よって獲得した知識を精練・一般化する必要が生じる。また、知識を精練・一般化することにより、より広範囲に組立経験知識を適用できることになる。

本研究においては、獲得した組立経験知識に対して、同じ組立法を有する2つの組立経験知識のインデックス部の一般化を、図5.3-2の部品の機能の階層構造を用いて試みることにより、2つの組立経験知識を1つに統合する。

##### ◎ 2つの組立経験知識の統合

- ・対象：：同じ組立法を有する組立経験知識
- ・処理：：インデックス部の更新
- ・方法：（1）インデックス部の3パート（組立部品の機能、機能部品に対す

- る組立動作、ターゲット状態)のそれぞれの共通部分をとる。
- (2) インデックス部の3パートのそれぞれの非共通部分に対して、その機能の階層構造をさかのぼり、共通なものがあれば、それを(1)で導き出したものに加える。
- (3) インデックス部の3パートのいずれかが、(1)、(2)いずれよりも何も導けなかった場合、統合失敗。

### 5.3.5 組立実行例

実験には、移動ロボットに搭載するアームの組立マニュアル [Heath 82] を用いた。例として用いたのは、アームの関節を回転させるモータ部を4つの部品で組み立てる組立工程である。その説明図を図5.3-5(a)に、説明文を図5.3-5(b)に示す。

この組立工程において、(2)の `set screw` に関する組立動作は `start` であるが、言語解析の結果、意味がわからずに `insert` と解釈していると仮定している。

(実際には、`start` には「<ねじ、釘などを>あらかじめ少し打ち込んでやる」(リーダーズ英和辞典, 研究社)という意味があるが、この研究開始当初はわからなかつ

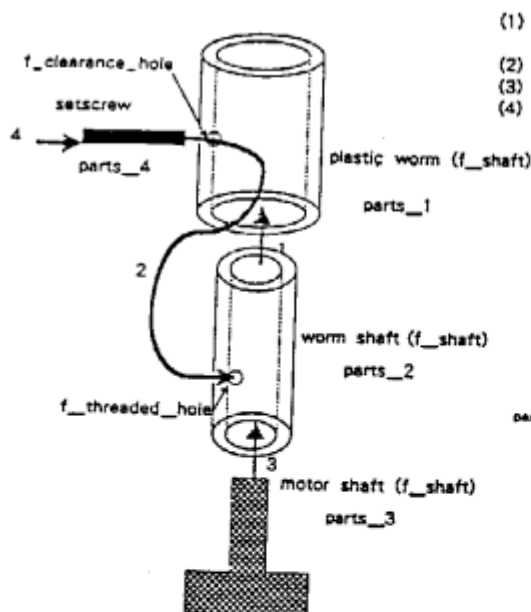


図5.3-5(a)組立説明図

- (1) Press the worm shaft into the worm with the 3/16" nut driver until the mounting holes align.
- (2) Start the 2-56 x 1/8" allen setscrew into the worm shaft.
- (3) Mount the worm assembly onto the wrist rotate motor shaft.
- (4) Tighten the setscrew with the #2 allen wrench.

図5.3-5(b)組立説明文

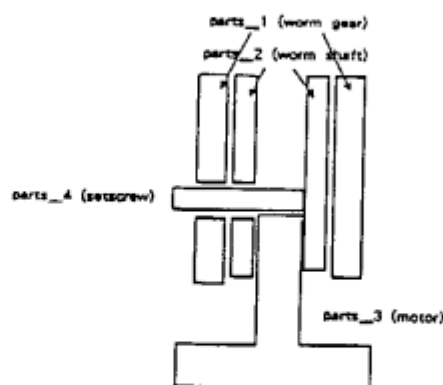


図5.3-6 誤った組立

た) また、(3)の `motor shaft` に関する組立動作 `mount` についても、`insert` と解釈している。(これは、オブジェクトを+基本立体、ターゲットを-基本立体とするためである)

(2)、(3)の `insert` 動作において、部品の挿入の長さが示されていないので、

デフォルトとして”奥まで挿入する”というのをを用いて、組立操作仮説を生成している。

この時に(1)～(4)の組立操作を行なった後の組立状態を図5.3-6に示す。

検証の結果、図5.3-3のルールに違反し、set screwとworm shaftとmotor shaftの位置関係がおかしいということが、TMSのcontradictionノードとして登録される。その結果、DDB機能により、(2)のset screwに関する組立操作仮説AS1(ノード4)と(3)のmotor shaftに関する組立操作仮説AM1(ノード6)が矛盾の原因として導かれ、”set screwに関する組立操作仮説を優先する”というヒューリスティックにより、(2)の組立操作仮説AS1(ノード4)を修正することにする。set screwに関する接触関係を調べてみると、worm shaftとの接触関係とmotor shaftとの接触関係がある。motor shaftとの接触関係を除去することになると、Type Dの接触であるので、除去戦略により、代替仮説AS2(ノード12)を生成し、ノードの状態を変更する。(worm shaftとの接触関係を除去することにすれば、代替仮説を生成すると、再び図5.3-6の失敗状態となり、今度はmotor shaftとの接触関係を除去することになる。)この時のノードの状態を図5.3-7に示す。

図5.3-7より、ノード5, 7, 9がoutになっているが、これは、(2), (3), (4)の組立操作を行なった後の部

品の三次元位置が現在は成り立っていないということを示しており、(2), (3), (4)の組立操作を(2)についての新たな組立操作仮説AS2(ノード12)のもとに行なうことになる。最終的に得られる組立経験知識を図5.3-

| NODE   | FACT                         | STATUS | JUSTIFICATION    |
|--------|------------------------------|--------|------------------|
| 1      | action(insert,parts_2,s(1))  | in     | {s1,{},{}}       |
| 2      | location(parts_1,s(0))       | in     | {s1,{},{}}       |
| 3      | location(parts_3,s(1))       | in     | {s1,{1,2},{}}    |
| AS1 4  | action(insert,parts_4,-)     | out    | {}               |
| 5      | location(parts_4,-)          | out    | {}               |
| AM1 6  | action(insert,parts_3,s(3))  | in     | {s1,{},{oc}}     |
| 7      | location(parts_3,-)          | out    | {}               |
| 8      | action(tighten,parts_4,s(4)) | in     | {s1,{},{}}       |
| 9      | location(parts_4,-)          | out    | {}               |
| 10     | contradiction                | out    | {}               |
| 11     | {nogood,6,4}                 | in     | {s1,{3,2},{}}    |
| AS2 12 | action(insert,parts_4,s(2))  | in     | {s1,{11,6},{oc}} |
| oc     | other_candidate              | out    | {}               |

図5.3-7 TMSの状態

8に示す。

| experience       |                 | experience       |                    |
|------------------|-----------------|------------------|--------------------|
| [INDEX]          |                 | [INDEX]          |                    |
| function         | : f_motor_shaft | function         | : f_setscrew       |
| action           | : insert        | action           | : insert           |
| target           | : f_a_primitive | target           | : f_clearance_hole |
|                  | f_worm          |                  | f_shaft            |
|                  |                 |                  | f_threaded_hole    |
| [METHOD]         |                 | [METHOD]         |                    |
| collision type   | : s             | collision type   | : d                |
| collision object | : f_worm        | collision object | : f_motor_shaft    |

図5.3-8 獲得された組立の知識

### 5.3.6 おわりに

組立説明文からの機械自動組立において、自然言語の曖昧な部分を仮説を用いて補って組立を行ない、失敗した場合効率よく再計画を行ない、その結果を学習するシステムについて報告した。

TMSのDDB機能を用いて、部品間の依存関係により誤りに関係のある部分だけを修正することは有効であり、多数の仮説のもとで、組立部品間に独立な部分が多いほどDDB機能による有効性は顕著になる。

また失敗原因の仮説からその対象部品の接触関係の除去をもとに代替仮説を生成することで、意味のある仮説の修正が行えるが、逆に言えば、部品の接触関係の数だけしか代替仮説が生成できないことになり、接触関係に無関係な組立失敗の場合、解決不可能となる。

### 参考文献

- (1) 安部憲広, 石川智浩, 辻三郎: 組立説明文からの組立手順の生成, 人工知能学会誌, Vol. 3, No. 5, pp. 60-68 (1988)
- (2) Doyle, J.: A Truth Maintenance System, Artif. Intell., Vol. 12, pp. 231-271 (1979)
- (3) 山田誠二, 安部憲広, 辻三郎: 電機ドリル分解・組立コンサルタントシステム, 人工知能学会誌, Vol. 1, No. 1, pp. 116-123 (1986)
- (4) 佐古慎二, 安部憲広, 辻三郎: 自然言語による動作指令の詳細化とその検証, 情報処理学会グラフィクスとCADシンポジウム論文集 (1988)
- (5) "HERO ROBOT Arm Accesary Manual", HEATH COMPANY, MICHIGAN (1982)
- (6) 西村和久: 機械組立法の学習, 大阪大学修士論文 (1990)
- (7) Shank, R. C.: Dynamic memory, Camblidge, UK: Camblidge University Press (1982)
- (8) J. L. Kolodner, R. L. Simpson, K. Sycara: A process model of case-based reasoning in problem solving, In Proceedings of IJCAI-85, pp. 284-290, LA (1985)
- (9) Reid G. Simmons: A Theory of Debugging Plans and Interpretations, In Proceedings of AAAI-88, pp. 94-99 (1988)

#### 5. 4 事例を用いた自然言語における動詞の意味同定

東京工業大学 山村雅幸, 小林重信

##### 要 旨

現在の多くの知識システムに利用されているルールベース推論に代る新しい問題解決パラダイムとして、事例ベース推論( Case-Based Reasoning, 以下CBR)が注目されている。本研究では、CBRを利用して自然言語における動詞の意味同定を行うモデルを提案する。そこでは、事例は問題文、文構成要素の格および動詞の意味からなる。問題文が入力されると、名詞の概念階層に基づいて組織化された事例ベースから事例が検索される。事例に基づいて、問題文に対する可能な意味およびなぜその意味が妥当であるかを示す根拠が出力される。出力に対して正解が教示されると、事例ベースは再組織化される。このようなモデルを、FUJI-XEROX 1108 上の Interlisp-D に実装し、その有用性を示す。

##### 5.4.1 はじめに

機械翻訳の基本的なタスクは、個々の対応関係に基づいた置換プロセスと考えられる。このとき、ある種の基本原理や一般的な知識に基づく定型的な処理が可能ではあるが、多くの場合、個別性、特殊性を考慮したがきめ細かい対応が必要である。

機械翻訳システムをはじめ、現在の多くの知識システムに利用されているルールベース推論は、基本的に演繹的推論を用いることから、知識ベースの正当性および一般性の保証を前提とする。このため、特に機械翻訳のように個別的、特殊な処理が必要とされる領域では知識の改良や保守には多大なコストと時間が必要であり、知識獲得におけるボトルネックとなっている。

一方、ルールベース推論に代る新しい問題解決パラダイムとして、CBRが注目されている。CBRは、過去に経験した事例を直接利用して、新たな問題解決を試みる。このため、事例間にはルールベースのルール間に要請されるような整合性の保証は不必要で、事例ベースの保守や拡張が比較的容易であるという特徴がある。したがって、機械翻訳のように、個別的、特殊な開いた世界を対象とする問題領域に対しては適切な枠組みと考えられる。しかし、CBRには概念的枠組みしか提案されていないのが現状であり、明確な枠組みの確立が要請されている。

本研究では、機械翻訳の副領域として自然言語における動詞の意味同定問題を取り上げ、CBRの課題として効率的な事例検索と事例ベースの再組織化に注目して、CBRによる接近を試みる。

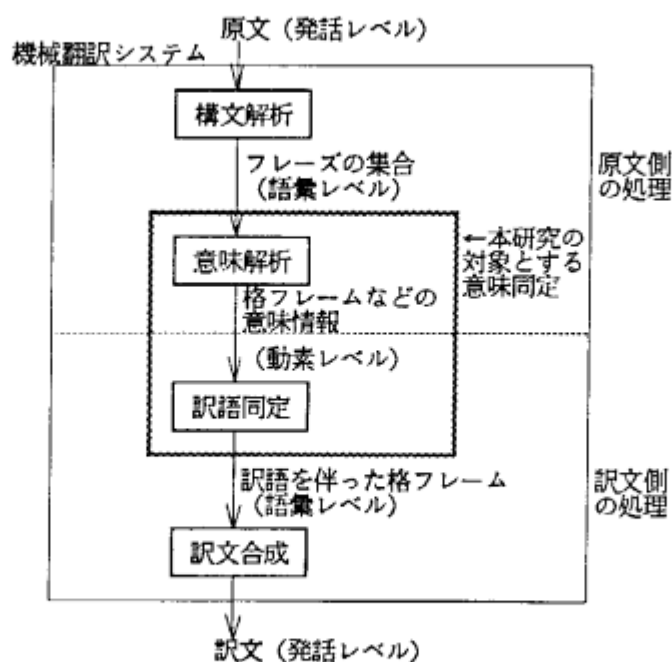


図 5.4-1 機械翻訳処理の枠組み

## 5.4.2 問題設定

### 5.4.2.1 機械翻訳

本研究では、機械翻訳の処理を図5.4-1に示すような入れ子構造としてとらえる。発話レベルとは、文章を記述するレベルである。語彙レベルとは、品詞などの文法的役割を記述するレベルである。動素レベルとは、概念依存関係 [Schank 82] のような、特定の言語に依存しない意味的役割を記述するレベルである。4つのフェーズはこれらのレベルの間で記述の詳細化や変換を行う。本研究では、このうち意味解析および訳語同定を取扱う。これを総称して意味同定と呼ぶ。

本研究では、英語から日本語への翻訳を取扱う。入力文における動詞は唯一とし、文脈情報は利用しないものとする。各フェーズでは解答が一意に定まらないことがあるが、このときは各候補に対してそれぞれ次のフェーズが起動されるものとする。

### 5.4.2.2 意味解析

構文解析の結果が同一のときも、意味が同一であるとは限らない。次の例を考える。

- 例. a) I take a car for \$10000.      for: ~の値段で[金額].      take: 買う  
 b) I take a car for Tokyo.      for: ~へ向けて[到着地].      take: 乗る

2つの例文において、前置詞句の文法的役割は同じ副詞句であるが、前置詞句を構成する名詞の概念が異なるために、その意味的役割が異なり、結果的に動詞の意味が異なっていると考えられる。このように、動素レベルの記述は語彙レベルの記述より詳細な情報を持つ。前置詞句のように語彙レベルの記述から動素レベルの記述が一意に定まらないときは、前置詞句を構成する名詞のような他の要素をもとに、記述を詳細化する必要がある。



|                                      |               |
|--------------------------------------|---------------|
| (LABEL ICASE001)                     | ←事例識別子        |
| (SENTENCE "He took a car to Tokyo.") | ←問題文          |
| (VERB TAKE)                          | ←問題文における動詞    |
| (PREPFRAME (to! OBJECT))             | ←意味解析に必要なフレーム |
| (CASEFRAME (AGENT OBJECT to!))       |               |
| (AGENT He)                           |               |
| (OBJECT Car)                         |               |
| (prep! Tokyo)                        |               |
| (to! Tokyo)                          |               |
| (EFFECT)                             |               |

図 5.4-3 意味解析フェーズにおける問題の表現

|                                      |            |
|--------------------------------------|------------|
| (LABEL ICASE001)                     |            |
| (SENTENCE "He took a car to Tokyo.") |            |
| (VERB TAKE)                          |            |
| (PREPFRAME (to! OBJECT))             |            |
| (CASEFRAME (AGENT OBJECT SPACE-TO))  |            |
| (AGENT He)                           |            |
| (OBJECT Car)                         |            |
| (prep! Tokyo)                        |            |
| (to! Tokyo)                          |            |
| (EFFECT (to! SPACE-TO))              | ←決定した前置詞の格 |
| (SPACE-TO Tokyo)                     | ←決定した格の属性値 |

図 5.4-4 訳語同定フェーズにおける問題 (意味解析フェーズにおける事例) の表現

|                                      |            |
|--------------------------------------|------------|
| (LABEL ICASE001)                     |            |
| (SENTENCE "He took a car to Tokyo.") |            |
| (VERB TAKE)                          |            |
| (PREPFRAME (to! OBJECT))             |            |
| (CASEFRAME (AGENT OBJECT SPACE-TO))  |            |
| (AGENT He)                           |            |
| (OBJECT Car)                         |            |
| (prep! Tokyo)                        |            |
| (to! Tokyo)                          |            |
| (EFFECT (to! SPACE-TO))              |            |
| (SPACE-TO Tokyo)                     |            |
| (MEANING NOTTEIKU)                   | ←解 (動詞の訳語) |

図 5.4-5 訳語同定フェーズにおける事例の表現

図5.4-2に本研究で採用するCBRの流れを示す。各フェーズでは、不完全な格フレームが問題として入力され、事例ベース推論によって補われる。検索される事例はその問題の解として直接利用可能であるため、事例の修正は不必要である。あるフェーズで出力される解は、次のフェーズの問題となる。各フェーズにおいて出力される解に対して、教師による正しい解の教示が行われる。教示された解に基づいて新しい事例が生成され、事例ベースは再組織化されて、将来の問題に利用される。

検索および再組織化のための知識として、木構造をなす名詞の概念階層を利用する。



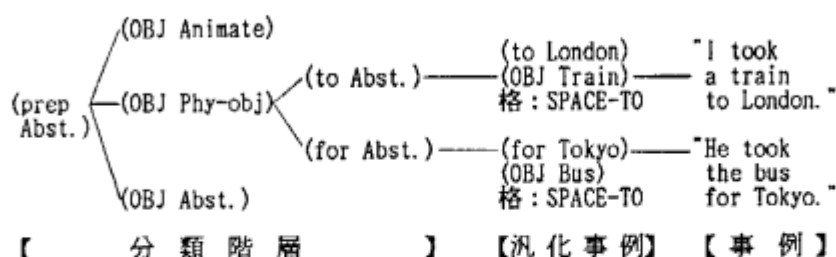


図 5.4-6 意味解析における事例ベース

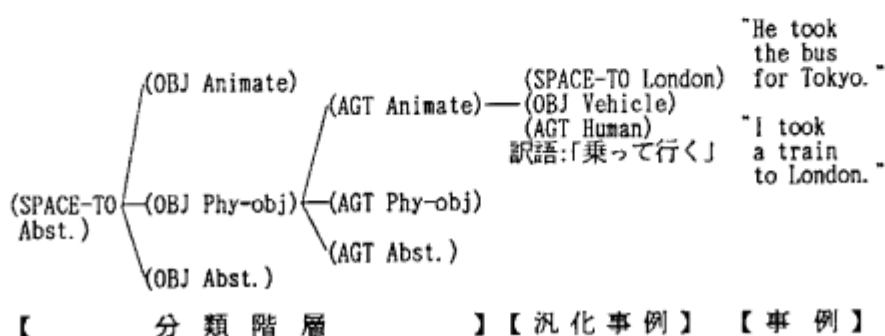


図 5.4-7 訳語同定における事例ベース

#### 5.4.3.2 用語の定義および説明

- (1) 問題文：翻訳の対象となる英文である。
- (2) 問題：意味解析では、問題文および問題文を構文解析した結果、すなわち動詞、主語や目的語を構成する名詞、前置詞句とそれを構成する名詞からなる。訳語同定では、これに意味解析の解が加わる。意味解析、訳語同定の各フェーズにおける問題の例を図5.4-3、図5.4-4に示す。
- (3) 解：意味解析では前置詞の格(意味的役割)、訳語同定では動詞の訳語である。
- (4) 事例：問題と解の対からなる。訳語同定における事例の例を図5.4-5に示す。
- (5) index：格フレームの属性のうち、事例識別子、問題文および解を除いたもの。
- (6) 分類階層：一般概念を属性値としてもつindexを、ある順序にしたがって階層的に組織化したもの。分類階層は所与とし、再組織化の対象とはしない。
- (7) 汎化事例：分類階層によって同じカテゴリーに分類された事例のうち、同一の解をもつものを汎化した事例。各属性ごとに概念階層にしたがって属性値が汎化される。
- (8) 照合：問題が分類階層(汎化事例)に照合するとは、分類階層(汎化事例)のindexが問題にも含まれ、かつ分類階層(汎化事例)の属性値が問題の属性値の上位概念であること。
- (9) 事例ベース：分類階層、汎化事例、事例からなる。初期の事例ベースは、分類階層のみからなる。意味解析、訳語同定における事例ベースの構成例を図5.4-6、図5.4-7に示す。

#### 5.4.3.3 検索アルゴリズム

与えられた問題は、分類階層をトップノードから始めて横型に探索し、照合した分類階層の末端ノードの下にある汎化事例を収集する。以下に概念的な流れを示す。

- (1) 分類階層上の最も上位のノード topnode を target index とする。
- (2) 残っている target index がなくなるまで、以下(4)までを繰り返す。
- (3) target index が汎化事例であればそれを収集し、他の target index について続行。
- (4) 問題が target index に照合するとき、その target index の下位ノードである index を新たに target index として、古い target index を抹消する問題が target index に照合しないときは、単にその target index を抹消する。
- (5) 残った target index がなくなったとき、収集された汎化事例と問題との照合を試みる。
- (6a) 照合する汎化事例が存在する場合、その下位にある事例を解として出力する。
- (6b) 照合する汎化事例が存在しない場合、(3)で収集された汎化事例の有無を調べる。
- (7a) 収集された汎化事例が存在するとき、それらの下位ノードのすべての事例から最も類似度の高いものを選択する。ここで類似度は、問題と事例の属性値の差、すなわち概念階層上のパス長の加重和で定義される。加重は index の属性間の優先順位を表わし、意味同定では問題中の (PREPFRAME)、訳語同定では (CASEFRAME) における属性値の順序とする。
- (7b) 収集された汎化事例が存在しないとき、意味解析では同じ格を持つが異なる前置詞を含む汎化事例が探索が停止した分類階層の末端ノードの下位にあるならば、それを選択する。訳語同定では意味解析フェーズにおける類似した前置詞という概念に対応するものがないため何もしない。

この検索アルゴリズムでは、複数の異なる分類階層の下にある汎化事例の収集、および(7a)のように厳密な照合を緩和した形での類似の事例の検索を許している。これは、このことによってルールベース推論との相違点を明確化する意図に基づく。

#### 5.4.3.4 教示・再組織化

検索された解に対して、教師が正解を教示する。教示に基づいて事例ベースは再組織化される。再組織化は大きく3つの場合に分けられる。

- (1) 照合した事例の中に正解が含まれるとき、その事例を下位ノードとしてもつ汎化事例の下位に新しい事例を記憶する。
- (2) 照合した事例の中には正解が含まれないが、収集した汎化事例の中には正解が存在するとき、その汎化事例と新しい事例を最少汎化したのち、その下に新しい事例を記憶する。
- (3) 収集した汎化事例の中にも正解が存在しないとき、新しい汎化事例を生成したのち、その下に新しい事例を記憶する。

事例の再組織化は汎化事例の最少汎化と事例の付加のみからなり、分類階層自体の再構成は行わない。このため、将来の誤った分類を回避する、いわゆる事例の修復 (repair) は行わない。これは、検索と再組織化の効率化を目的として処理を単純化したためである。修復を伴う再組織化は今後の課題である。

#### 5.4.4 適用例

前章までに実装したモデルの動作を具体的な問題で例示する。モデルの有用性を検証するための例としては、なるべく広い意味に用いられる動詞が望ましい。takeはそのような動詞の代表であると思われる。典型的な意味には「手に取る」などがあるが、文中の他の語句によっては、根本的に意味が異なり、動素レベルをとってみても1通りの意味では到底カバーしきれない。そこで、takeという動詞を例に用いて、問題を入力し、意味解析、および基本的な意味が決定された後の適切な訳語の同定を行なう。

以下の3つの例題は、意味解析フェーズにおける検索と同時に、訳語同定フェーズにおける再組織化を例示する意図から構成されている。意味解析フェーズの検索を例示する側面からは、問題1は過去の事例に照合するもの、問題2は照合する事例はもたないが、照合しないもののうち類似である事例を検索できるもの、問題3は照合する事例をもたず、同じ前置詞をもつもので類似の事例が見つからないが、他の前置詞をもつ事例において参照可能なものが存在するものに対応する。また、訳語同定フェーズの再組織化を例示する側面からは、問題1は既存の汎化事例に単純に事例を付加するもの、問題2は既存の汎化事例を再汎化するもの、問題3は新しく汎化事例を生成するものに対応する。

モデルは XEROX 1108 上の Interlisp-D に実装されている。ここで示す例題の処理時間の目安として応答時間は、各フェーズとも検索に10秒程度、汎化をとまなう再組織化に数秒である。

問題1: "He took Mary to the post office."

〔意味解析〕 前置詞toの目的語が post-office であることと、動詞の目的語がMaryであることから、問題は分類階層の末端ノード [(to Phy-obj)(OBJECT Animate)] に照合し、さらにその下位ノードである汎化事例 [(to Phy-obj)(OBJECT Human)] に照合する。検索の結果として、この汎化事例に付与された解 SPACE-TO(運動の終点)、およびこの汎化事例の下にある2つの事例 "Will this road take me to the station?" ,

および "This bus will takeyou to the library." がその解の根拠として出力される。正解として格 SPACE-TO が教示されると、この解は検索時に照合した汎化事例の解と同一であることから、汎化事例は変化せず、現在の問題と解を新しい事例としてその汎化事例の下に再組織化する。

〔訳語同定〕 格 SPACE-TO をもつ前置詞の目的語が post-office であることと、動詞の目的語、主語がMary, heであることから、問題は分類階層の末端ノード [(SPACE-TO Phy-obj)(OBJECT Animate)(AGENT Animate)] に照合するが、その下に汎化事例をもたない。その結果、システムは検索すべき事例は存在しないと認識する。正解として、「連れていく」が教示されると、分類階層 [(SPACE-TO Phy-obj)(OBJECT Animate)(AGENT Animate)] の下位ノードとして、新しい汎化事例 [(SPACE-TO Post-office)(OBJECT Mary)(AGENT He)] を生成し、さらにその下に、現在の問題と教示された解を事例としてその汎化事例の下に再組織化する。

問題2 : “He takes a fancy to these books.”

〔意味解析〕 前置詞toの目的語がbookであることと、動詞の目的語がfancyであることから、問題は分類階層の末端 [(to Phy-obj)(OBJECT Abstraction)] に照合するが、その下の汎化事例に照合するものがない。そこでシステムは、照合した分類階層の末端の下位に位置するすべての事例について、問題との類似度を測定する。その結果、最も類似する事例として、“I take a dislike to apples.” を示し、その解である RECIPIENT (行為を受ける対象)を出力する。正解として、格 RECIPIENT が教示されると、この解をもつ汎化事例が、現在の分類階層の末端の下に存在するので、その汎化事例のindexを汎化して、

[(to Phy-obj)(OBJECT Feeling)] とする。さらにその下位ノードとして、問題と教示された解を新しい事例として再組織化する。

〔訳語同定〕 格 RECIPIENT をもつ前置詞の目的語がbookであることと、動詞の目的語、主語がそれぞれfancy, heであることから、問題は分類階層の末端 [(RECIPIENT Phy-obj)(OBJECT Abstraction)(AGENT Animate)] に照合するが、その下に汎化事例をもたない。その結果、検索すべき事例は存在しない。正解として、「好き」が教示されると、分類階層 [(RECIPIENT Phy-obj)(OBJECT Abstraction)(AGENT Animate)] の下位ノードとして、新しい汎化事例 [(RECIPIENT Book)(OBJECT Fancy)(AGENT He)] を生成し、さらにその下位ノードとして、問題と教示された解を新しい事例として再組織化する。

問題3 : “I took a train to London.”

〔意味解析〕 前置詞toの目的語がLondonであることと、動詞の目的語がtrainであることから、問題は分類階層の末端 [(to Abstraction)(OBJECT Phy-obj)] に照合するが、その下に汎化事例をもたない。しかし、前置詞toと同じ解(ここでは SPACE-TO)を共有している他の前置詞forにおいて、同じ分類階層 [(for Abstraction)(OBJECT Phy-obj)] にその解を共有する汎化事例が存在する。その汎化事例の下の事例 “He took the bus for Tokyo.” による解 SPACE-TO が出力される。正解として、格 SPACE-TO が教示されると、問題に照合している分類階層の下位ノードとして、新しい汎化事例 [(to London)(OBJECT Train)] を生成し、さらにその下位ノードとして、問題と教示された解を新しい事例として再組織化する。

〔訳語同定〕 格 SPACE-TO をもつ前置詞の目的語がLondonであることと、動詞の目的語、主語が, train, I であることから、問題は分類階層の末端 [(SPACE-TO Abstraction)(OBJECT Phy-obj)(AGENT Animate)] に照合するが、その下に照合する汎化事例をもたない。そこでシステムは、照合しない汎化事例の下位に位置するすべての事例について、問題との類似度を測定する。その結果、該当する事例として、“He took the bus for Tokyo.” を示し、その解である「乗って行く」を出力する。正解として、「乗って行く」が教示されると、この解をもつ汎化事例は、現在照合している末端の分類階層の下位ノードに存在するので、その汎化事例のindexを汎化し、 [(SPACE-TO Location)(OBJECT Vehicle)(AGENT Human)] とする。さらにその下に問題と教示された解を再組織化する。

#### 5.4.5 結 論

本研究では、自然言語における動詞の意味同定にCBRを適用して、その有用性を示した。その効果と残された課題を以下にまとめる。

本モデルでは、与えられた正解と異なる解の出力を抑制することは行なっていない。その理由は、1つの自然言語文の意味、とくに語彙レベルの意味には多くのバリエーションをもたせることができるからである。人間が自然言語を翻訳する際、1つの文章に対して、意味が一意であっても、翻訳者によって、あるいは状況を考慮した尊敬語・謙譲語などの利用によって、語彙レベル、発話レベルの翻訳結果に違いが生じてくる。

本モデルは、このような様々な翻訳結果を制限することなく、可能性のあるものをすべて記憶し、出力するという点において、ルールベース推論では困難なタスクを、CBRによって容易に実現したといえる。

本モデルでは、事例ベースの三層構造を提案した。分類階層の利用によって、検索速度の向上が図られている。また汎化事例の導入によって、獲得される事例の逐次的な再組織化に有効に作用するという効果があげられる。

CBRシステムとしての本モデルは、以下のことが扱えないという点において限界があり、今後の研究課題として指摘される。

- (1) 分類階層の逐次的再組織化：本モデルにおいては、事例ベースの検索のため、分類階層の所与を仮定している。逐次的に更新される分類階層を考慮することが、将来のCBRの発展のための課題となることが考えられる。
- (2) 木構造を仮定しない概念階層の利用：本モデルでは木構造を仮定しているが、木構造を仮定しない概念階層において処理できるモデルの構築が期待される。
- (3) 特徴付けにおける属性値間の相互作用の表現：本モデルにおける問題の特徴付けは、各属性値間を独立に表現する。属性間の関係を記述できるようにindexを表現することが期待される。

#### 参考文献

- [Schnk 82] Schank, R. "Dynamic memory: A theory of learning in computers and people," Cambridge University Press (1982).
- [Hammond 86] Hammond, K. J. "CHEF: A Model of Case-based Planning," Proc. of AAAI-86 (1986).
- [佐藤 89] 佐藤理史, 長尾 真, "事例に基づいた翻訳," 自然言語処理 70-9 (1989).
- [松井 90a] 松井 丈, "事例ベース推論を用いた自然言語における動詞の意味同定," 工学修士論文 東京工業大学総合理工学研究科 (1990).
- [松井 90b] 松井 丈, 山村雅幸, 小林重信, "事例ベース推論を用いた自然言語における動詞の意味同定," 第11回知識・知能システムシンポジウム (1990).

## 5. 5 事例に基づく問題解決過程の制御モデル

徳島大学工学部 小野典彦

### 要 旨

問題解決事例に EBL を適用し、問題解決過程を適切に制御するための知識の獲得を目指した研究が数多く展開されている。本報告では、問題解決過程そのものに再帰性があり、EBL だけではこのような知識の獲得が難しいと考えられる探索問題の一例として、不定積分過程をとりあげ、これを制御するための知識を事例ならびに説明に基づいて獲得するためのモデルについて述べる。このモデルの主たる特色は事例に基づいた探索制御過程にあり、そこでは問題の特徴付けるための領域に依存した付加的情報は導入されず、各問題に適用可能なオペレータの集合だけを用いて問題の特徴付けおよび索引化が施こされ、さらに問題解決事例の蓄積ならびに動的な組織化を通して、探索過程を制御する上で重要な特徴および探索過程の共通性が特定される。

### 5. 5. 1 はじめに

問題解決事例に EBL[Mitchell 86][DeJong 86] を適用し、問題解決を適切に制御するための知識の獲得を目指した研究が数多く展開されている。これらはマクロ [Korf 85] を獲得するもの [Mooney 88][Shavlik 87][山田 89] およびオペレータの適用条件を洗練化するもの [Minton 88][Mitchell 83][Keller 87] とに大別される。前者の接近法の問題点としては、問題解決過程そのものに再帰性がある有限個のマクロでは問題解決空間を被いつくせない場合が少なくないことが挙げられる。後者の接近法についても、単純な探索や導出原理に基づいて問題解決を行なっているかぎり、説明構造を特定する過程がボトルネックとなる。

本研究では、問題解決過程が本来再帰的な構造をもつ探索問題として初等的な不定積分をとりあげ、その問題解決過程を予め再帰的な側面をもつ過程とそうでない過程とに分割し、そもそも演繹的な定理証明技法だけでは説明構造を特定し得ない前者の過程に対しては事例ベース推論 [Kolodner 88][Riesbeck 89] を適用し、EBL については後者のみに適用することを試みると共に、主として事例ベース推論をこのような再帰的な探索過程に適用する上での諸問題について考察する。

### 5. 5. 2 不定積分過程

よく知られているように Mitchell らは不定積分過程を探索問題として定式化し、その探索過程を適切に誘導する制御知識を自動的に獲得することを試みている[Mitchell 83]。ここでは数式そのものを状態、それを書き換える演算をオペレータ、積分記号のとれた数式を目標状態としているわけであるが、ここでは数式そのものではなく、数式の書き換え過程そのものを状態と考える。

これはつぎの例に代表される再帰的な問題解決過程を通してはじめて解決される問題の扱いを容易にするためである。

$$\begin{aligned}\int e^x \sin x dx &= e^x \sin x - \int e^x \cos x dx, \\ \int e^x \sin x dx &= e^x \sin x - (e^x \cos x + \int e^x \sin x dx), \\ \int e^x \sin x dx &= \frac{1}{2}(e^x \sin x - e^x \cos x)\end{aligned}$$

このような問題は積分式そのものを状態と考えるよりは、つぎの等式集合を状態と考え、

$$\left\{ \int e^x \sin x dx = e^x \sin x - \int e^x \cos x dx, \int e^x \cos x dx = e^x \cos x + \int e^x \sin x dx \right\}$$

さらにこれをつぎの各積分式を未知数とする連立一次方程式とみなし、

$$\int e^x \sin x dx (= X), \int e^x \cos x dx (= Y)$$

空集合を初期状態、新たな等式（方程式）をこの集合に付加する演算をオペレータ、可解となった連立一次方程式を目標状態と定義したほうが扱いやすい[白井 89]。そこで不定積分過程をつぎのような連立一次方程式の形成過程とみなすことにする。

$$\begin{aligned}\{\} &\Rightarrow \{X = e^x \sin x - Y\} \Rightarrow \\ &\{X = e^x \sin x - Y, Y = e^x \cos x + X\}\end{aligned}$$

これは図 5.5.1 に示すような階層的な問題解決空間をもつ問題解決器として考えてよい。

(1) 監視過程: 積分式を入力すると（あるいは次に計算すべき積分式を検出すると）、初期状態を空集合に設定し、それを下記の簡約化過程に伝達し、簡約化過程がこの数式を左辺とする新たな方程式を生成するのを待つ。この過程は、こうして更新されていく状態（方程式の集合）を常時監視しており、方程式が可解となった時点で、つまり上記の目標状態が達成された時点で、その旨を簡約化過程に通知すると共に、この連立方程式を解き、その解を出力する。

(2) 簡約化過程: 簡約化過程は、監視過程より積分式を受け取り、それを左辺とする方程式を1つ生成し、それを監視過程に返す。ただしこの過程には、 $xn$  の積分公式のようなプリミティブなオペレータおよび置換積分や部分積分などの積分式の簡約化オペレータだけが与えられており、これらが直接適用できない場合には、被積分関数を下記の数式変換過程に伝達し、それを等価な数式に変換させ、積分式が直接処理できる形式に変換されるのを待つ。

この過程は、受け取った積分式を初期状態として、それを新たな数式に書き換える演算をオペレータとする独立した問題解決器となっており、後で述べるようにここで適用されたオペレータ系列を事例ベースとして構築していくことにより、一種の制御知識を獲得するが、事例がない場合のデフォルトの制御戦略としては前向き深さ優先探索を採用している。

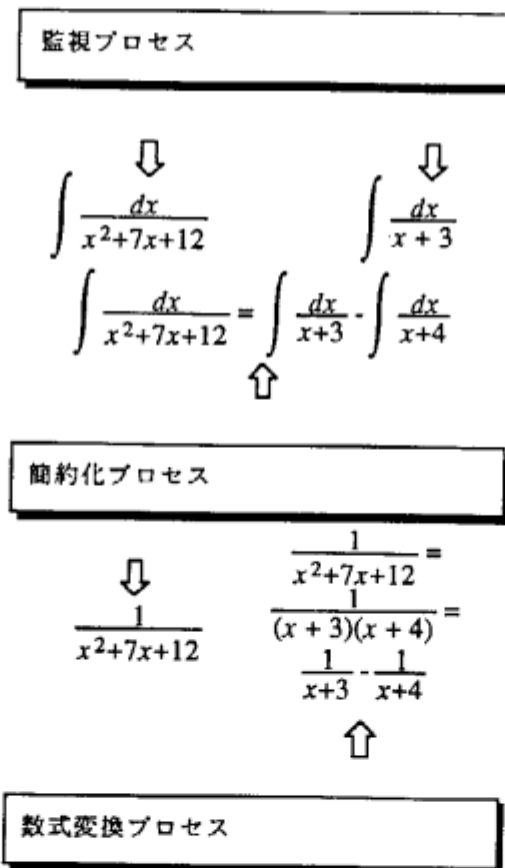


図 5.5.1 不定積分過程



(3) 数式変換過程: 数式変形過程は、上記のように簡約化過程が直接は変換できない積分式に出会った場合に呼び出され、その被積分関数を受け取り、簡約化過程が直接処理可能な形式に書き換えて返す。

この過程は、受け取った数式を初期状態として、それを等価な数式に変換するための演算をオペレータとする問題解決器となっており、後で述べるようなマクロを制御知識として獲得するが、デフォルトの制御戦略としては前向き深さ優先戦略を採用している。

### 5. 5. 3 説明および事例に基づく探索制御

さて、前節で述べたような階層的な問題解決器を採用するということは、既にある程度の制御知識を問題解決器に与えたのと同じことである。このような階層性はまったく与えずに、逆にこれを問題解決事例を通して学習することを目指すことが学習研究の本筋かもしれない。

もちろんこのような階層的な問題解決を施さなくとも、不定積分などはフラットな問題解決空間上で解決できる（本研究で扱っている範囲の初等的な問題であれば、領域知識を与えるのはけっして難しくない）。したがって、問題解決過程の階層性など与えなくとも、訓練事例を通して与えられた問題を直接その解に導くためのマクロの集合を獲得できる可能性は十分あるといわざるをえない。

しかし通常の EBL システムのように、問題解決をあくまでも演繹的かつ単純な探索や導出原理だけを通して行ない、そこで得られた解の筋道を説明として、それを一般化するだけでは、飛躍的な一般化が施せないばかりでなく、獲得されたマクロの適用性もきわめて低いものにとどまる。たとえば、つぎのような例題の解決事例から

$$\int x^2 e^{3x} dx, \int \frac{1}{x^2+4} dx$$

つぎのような問題を解決するための各々独立なマクロは獲得できても

$$\int x^2 e^{12x} dx, \int \frac{1}{x^2+9} dx$$

それらはいずれもつぎの問題には適用できないといったことが EBL では起こり得るのである。

$$\int x^3 e^x dx, \int \frac{1}{x^2+2x+9} dx$$

すなわち、このような与えられた問題をその解に直接導くような粒度の大きなマクロに関しては、たとえ獲得できたとしてもその適用性は必ずしも高くないのである。

このように問題解決事例の説明構造をあくまでも演繹的な定理証明技法に基づいて特定し、さらにそれを単一事例の解析に基礎をおく一般化技法である EBL に基づいて一般化しようとするかぎり、不定積分過程が本来有しているような再帰的な構造を自動的に特定することは難しい。

もちろん不定積分過程にも、たとえば上で述べた数式変換過程のように再帰性をほとんどたない過程があるわけであるから、EBL がまったく適用できないわけではない。むしろそのような過程の制御知識の獲得には EBL を適用するのが得策であろう。

しかしそれよりも上位の再帰的構造をもつ過程の制御知識を自動的に獲得しようとするならば、そもそも問題解決事例の説明構造を陽に生成し、さらにその正当性を保証することが難しい以上、EBL とは別の接近法を採用すべきであろう。近年ルールベース推論を補完し得る学習および推論の新たなパラダイムとして注目されつつある事例ベース推論は、蓄積された問題解決事例の集合を根拠として、あたかも陽に表現された制御知識をそこから生成し、それにしたがって問題解決を展開させることのできる問題解決器の構築を目指した研究分野と考えることもできる。このような推論機構が不定積分過程のような再帰的問題解決過程の制御にどこまで適用できるのかという問題はきわめて興味深い。

以上の観点から、不定積分過程を例にとり、つぎのような説明ならびに事例に基づく探索制御を試みる。

#### 5. 5. 4 事例に基づく探索制御

前節で述べたように簡約化過程では、積分式の簡約化過程を監視しながら、それを簡約化事例として記憶すると共に、新たな積分式が与えられると、それに類似した積分式の簡約化事例を想起し、それに適応化(adaptation)を施しながら、それを解決しようとする。

このような問題解決における探索制御に過去の事例を適用する場合には、(i) 数式の特徴をいかに表現するか、(ii) 事例をいかに表現するか、(iii) 事例ベースが蓄積されてきた場合に、(iv) いかにして事例を一般化するか、また (v) いかにして重要な特徴とそうでない特徴を抽出していくか、さらに (vi) 与えられた問題を事例にいかにして適応化するか、などの諸問題を解決していかなければならない。以下ではこれらのうち本研究に特徴的な側面について述べる。

##### 5. 5. 4. 1 数式の特徴集合

数式を特徴付けるための 1 つの方法は、数式の特徴に相当する記号を予め問題解決器に与えてしまうことである。確かにわれわれは数式を与えられると「この分数は分母が完全平方式である」あるいは「この分数は部分分数に展開できる」などといった特徴抽出を行

なっているようにも見える。しかしそのような情報は与えるよりはむしろ学習することに意義があること、またこのような数式処理に依存した情報を付加することは望ましくないことなどから、本研究ではこのような領域に依存した知識は付加せずに、問題解決器が予め有している基本的な知識だけから、徐々に蓄積されていく問題解決事例を通して、探索過程を左右する数式の重要な特徴とは何なのかを段階的に獲得させることを目指す。とはいえ、特徴集合の上限を決定しないわけにはいかないことから、本研究では各数式の特徴をそれに適用可能なオペレータの集合としてとらえるものとする。この集合が問題の類似性に関係するのか、つまり問題の解法を弁別する上でその問題に照合するオペレータの集合中のどの要素が重要で、どの要素がそうでないのかを、事例を通して問題解決器に学習させることが目標である。

ここでオペレータを通して抽出可能な特徴について簡単に触れておこう。本研究ではオペレータを4項組<識別子, 左辺, 右辺, 制約条件>で表現している。たとえば、数式変換過程はつぎのオペレータを利用して、

<(str,703),A/(B\*Func+C),1/(D\*Func+1)\*E,[div(B,C,D),div(A,C,E),noteq(E,1)]>  
つぎのような数式変換を施す。

$$\frac{12}{x^2+4} \Rightarrow \frac{3}{\frac{1}{4}x^2+1}$$

ここで識別子を構成する str および 703 は各々オペレータのクラスおよび番号である（このオペレータは上のような式変換を行なうだけでなく、数式が有理式であって、分母中の単項式と分子が共通因数をもつという特徴を検出するためにも利用される）。一方、簡約化過程は  $x^n$  や  $\frac{1}{x^2+1}$  の積分公式に相当するプリミティブなオペレータから置換積分や部分積分などの複合的なオペレータまで、積分式を別の数式に書き換えるためのオペレータをもっている。直感的には、プリミティブなオペレータに照合する積分式は、オペレータの識別子だけで特徴付けられるが、部分積分や置換積分のオペレータにしか照合しない積分式の場合は、それらの識別子（それぞれ1種類しかない）だけでは特徴になりそうもない。しかしこれらの複合的なオペレータは、積分式との照合時に間接的に利用されるオペレータの識別子が抽出されるように記述されていることから（これはあくまでも置換積分や部分積分を可能とするための記述であって、特徴抽出のために付加的に導入されたものではないことに注意されたい）、このような積分式の特徴もとらえることができる。たとえば、つぎの各積分式の場合には、

$$\int x^3 e^x dx, \int \frac{\frac{1}{2}}{\left(\frac{1}{2}x\right)^2+1} dx$$

つぎの集合が各々特徴として抽出される。

$$\left\{ \int fg' dx = fg - \int f'g dx, f: \int x^n dx = \frac{1}{n+1} x^{n+1}, g: \int e^x dx = e^x \right\},$$

$$\left\{ \int \varphi(f(x))f'(x) dx, \varphi: \int \frac{1}{x^2+1} dx = \tan^{-1} x, f: \int x^n dx = \frac{1}{n+1} x^{n+1} \right\}$$

#### 5. 5. 4. 2 簡約化事例の表現

簡約化事例は (i) 積分式の特徴および (ii) 簡約化過程で適用されたオペレータ系列の対である。たとえば、つぎの問題の解決事例は

$$\int x^2 e^x dx$$

2つの部分積分およびつぎのプリミティブよりなる系列を用いて表現され、

$$\int e^x dx = e^x$$

つぎの問題の場合には

$$\int \frac{12}{x^2+4} dx$$

数式変換過程によるつぎのような被積分関数の書き換えを介して、

$$\frac{12}{x^2+4} \Rightarrow \frac{6 \times \frac{1}{2}}{\left(\frac{x}{2}\right)^2+1}$$

最終的には置換積分および間接的につぎの2つのプリミティブを適用して解決された事例として記憶される。

$$\int \frac{dx}{x^2+1} = \tan^{-1} x, \int x^n dx = \frac{1}{n+1} x^{n+1}$$

なお、後者のように数式変換過程の助けを介して解決された問題については、その問題に照合するオペレータはなかったわけであるから、数式変換過程で照合したオペレータ集合が特徴として記憶される。これにより、たとえば後者の問題の解決事例をつぎの問題に適用して解決するといったことが可能となる（この特徴付けでは、この2つの積分式の特徴はほとんど一致する）。

$$\int \frac{dx}{x^2+2x+5}$$

さらに後者の事例を利用して他の問題を解決している際に同じ文脈で数式変換過程の助けが必要となった場合には、この事例において、数式変換過程による処理を介して適用可

能となったオペレータの左辺（これは積分式の形式をしている）の被積分関数のパターンを数式変換過程に目標として提示する。

#### 5. 5. 4. 3 問題解決事例の適応化

ここでは適応化として、いわゆる abstraction and respecialization [Riesbeck89] を採用する。ただし、ここでも数式処理に依存した付加的な概念は導入しないという方針から「長さ0以上のオペレータの並び」から「オペレータ系列」への abstraction だけを許すものとする。これによりつぎの問題集合の間で制御知識の移行が可能となる。

$$\int x e^x dx, \int x^2 e^x dx, \int x^3 e^x dx, \dots$$

「オペレータ」から先に触れた「オペレータのクラス」への abstraction も許せばより強力な適応化ができるはずであるが、オペレータのクラスはちょうど SBL におけるパイアスに相当する知識となることから現在はこれは用いていない。

#### 5. 5. 4. 4 事例ベースの構成

簡約化事例の集合は、既に述べた特徴を索引として、いわゆる MOP [Lebowitz 83][Kolodner 84, 88][Hammond 89][Riesbeck89] をノードとする階層的な弁別ネットで表現するものとする。ただし、各 MOP の下に規定数以上の事例が蓄積されるまでは副 MOP の生成は行なわない。これは誤った一般化を行なうことを避けるためである。

したがって、一般に MOP の下に規定数以上の事例が蓄積されるまでは、それらの事例への索引値としては、照合のコストの高いオペレータの集合値が用いられることになる。しかし、事例がその規定数を越え、より特殊な MOP が階層的に生成される際には、各 MOP の下に格納される事例の共通な特徴（オペレータ集合の共通部分）はその MOP のノルム (norm[Kolodner 84][Riesbeck 89]) として吸収してしまいうことができることから、索引値としては問題を弁別するのに重要な特徴だけが最終的には残ることになる。

このような事例集合の動的な組織化により、部分積分を再帰的に適用して解決される問題、最終的には同一のプリミティブによって解決される問題および種々のパターンの2次の有理式の積分などを弁別する問題解決事例のネット（図 5.5.2）が構成される。

#### 5. 5. 5 説明に基づく探索制御

第2節で述べたように数式変換過程では、簡約化過程が直接書き換えられない積分式に遭遇した場合に、被積分関数を書き換えるために起動される。この変換過程は前向きの深さ優先探索過程であって、解に到達すれば（被積分関数が簡約化過程が書き換えることができる形式になれば）成功してマクロ化され、それらは他のオペレータと同様に適用され

る。

ただし前節でも触れたように簡約化過程は過去の問題解決事例に基づき、目標となる被積分関数の形式を予め設定する場合がある。そこで数式変換過程では、このような目標が設定されている場合には目標に照合する右辺をもつオペレータから優先的に適用する。

ここで採用しているマクロ獲得過程は、基本的には Prolog-EBG[Kedar-Cabelli 87] などで採用されている技法を用いて、問題の解決とその一般化を同時進行的に施すなどの処理は行なっているが、獲得技法そのものには特に新しい側面はない。

マクロの問題点はたとえば互いに関係し合うマクロが独立のマクロとして獲得され、マクロ表が大きくなり、照合コストが高くなることであるが、Prolog によるシミュレーションに関する限り、この種の問題はほとんど生じていない。

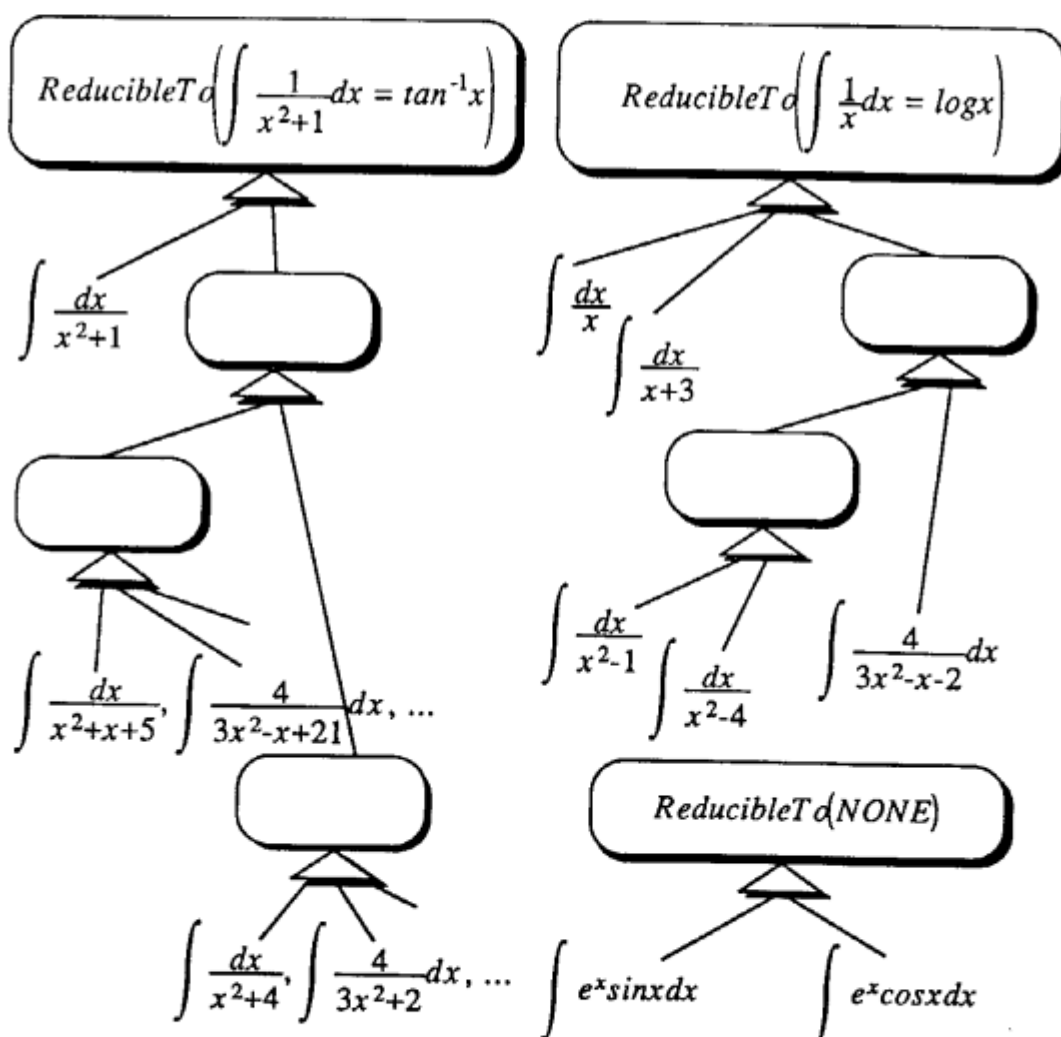


図 5.5.2 問題解決事例の弁別ネット

## 5. 5. 6 おわりに

本研究では、探索問題として初等的な不定積分過程をとりあげ、これを制御するための知識を事例ならびに説明に基づいて獲得することを試みた。ここでは不定積分過程を (i) 再帰性をもち単純にマクロを生成しても問題解決空間を被うことができない過程と、(ii) マクロを生成するだけでも有用な制御知識となり得る過程とに分割し、前者には事例ベース推論を、後者には EBL を適用した。この試みの主たる特色は前者の事例に基づいた探索制御過程にあり、そこでは問題の特徴付けるための領域に依存した付加的情報は導入されず、各問題に適用可能なオペレータの集合だけを用いて問題の特徴付けおよび索引化が施こされ、さらに問題解決事例の蓄積ならびに動的な組織化を通して、探索過程を制御する上で重要な特徴（オペレータ）および探索過程の共通性が特定される。

## 参考文献

- [DeJong 86] DeJong, G. and Mooney, R.: Explanation-Based Generalization: An Alternative View, *Machine Learning*, Vol.1, No.2, pp.145-176 (1986).
- [Hammond 89] Hammond, K.J.: *Case-Based Planning*, Academic Press Inc. (1989).
- [Kedar-Cabelli 87] Kedar-Cabelli, S.T. and L.T.McCarty: Explanation-Based Generalization as Resolution Theorem Proving, *Proc. of the 4th International Workshop on Machine Learning*, pp.383-389 (1987).
- [Keller 87] Keller, R.: Concept Learning in Context, *Proc. of the 4th International Workshop on Machine Learning*, pp.91-102 (1987).
- [Kolodner 84] Kolodner, J.L.: *Retrieval and Organizational Strategies in Conceptual Memory*, Lawrence Erlbaum Associates (1984).
- [Kolodner 88] Kolodner, J.L.: *Proceedings: Case-Based Reasoning Workshop (DARPA)*, Morgan Kaufmann Publishers, Inc. (1988).
- [Korf 85] Korf, R.E.: *Learning to Solve Problem by searching for Macro Operators*, Pitman Advanced Publishing Program (1985).
- [Lebowitz 83] Lebowitz, M.: Generalization from Natural Language Text, *Cognitive Science*, Vol.7, pp.1-40 (1983).
- [Minton 88] Minton, S.: *Learning Search Control Knowledge: An Explanation-Based Approach*, Kluwer Academic Publishers (1988).
- [Mitchell 83] Mitchell, T.M.: *Learning and Problem Solving*, *Proc. of IJCAI* (1983).
- [Mitchell 86] Mitchell, T.M., Keller, R. and Kedar-Cabelli, S.: Explanation-Based Generalization: A Unifying View, *Machine Learning*, Vol.1, No.1, pp.47-80 (1986).
- [Mooney 88] Mooney, R.: Generalizing the Order of Operator in Macro-Operators, *Proc. of the 5th International Conference on Machine Learning*, pp.270-287 (1988).
- [Riesbeck 89] Riesbeck, C.K. and Schank, R.S.: *Inside Case-Based Reasoning*, Lawrence Erlbaum Assoc (1989).
- [Shavlik 87] Shavlik, J.W. and DeJong, G.F.: BAGGER: An EBL System that Extends and Generalizes Explanation, *Proc. of AAAI*, pp.516-520 (1987).
- [白井 89] 白井, 小野, 寺内, 翁長: 問題の簡約化制御知識の学習: 数式の簡約化への適用事例 人工知能学会研究会資料, SIG-FAI-8902-4, pp.31-40 (1989).
- [山田 89] 山田, 辻: 完全因果性によるマクロオペレータの選択的学習, 人工知能学会誌, Vol.4, No.3, pp.321-329 (1989).

## 5.6 事例に基づくプラント診断システム: CAOS<sup>1</sup>

石田好輝 京都大学, 工学部 応用システム科学教室

### 要旨

ここでは, 4.2 で論じたスロット最小化によるダイナミックメモリ上での事例ベース推論を論じる. 領域としてはプロセス診断がえらばれる. 動的メモリ上での事例ベース診断方式を提案し, それをインプリメントしたシステムである CAOS について述べる. CAOS では, 事例ベース診断のために有用な様々な診断事例操作が提案され, それらの操作がどのような類似事例に対してどのように適用されるかが述べられる. 最後に, CAOS の実験を通して, ルールベースと事例ベースの比較が行われる.

### 5.6.1 構造化された事例ベース上での推論

CAOS によりこの記憶構造の枠組がどのように診断事例の索引に用いられ, また事例変形作用素がどのように選ばれるかが示される. 5.6.2 は, インプリメンテーションについて述べ, サンプルセッションを掲載する. 5.6.3 は, 動的な事例ベースを知識ベースシステムに導入したことの意義を論じる.

ここで推論する問題は, 与えられた未知の属性値を持つ新しい事例に対して, その未知の属性値を類似事例の対応する属性値から推論することである. 構造化された事例ベース上での推論は次の2つのステップで行われる.

1. 想起: 事例ベース上で関連する事例を探索する (4.2.2 節)。
2. 事例操作: その関連事例を適当な事例操作で変形する. 作用素は, 与えられた事例が最類似事例にどのように関連するかにより選択される。

想起プロセスは, インクリメンタルアルゴリズムで述べられた再構成の時に用いられるものと同じものである。

次の二つの節で想起と事例操作のそれぞれについてより詳しく述べる。

想起—関連する事例の探索

我々はすでに, 事例が事例ベースにおいてどのように構造化されるかを述べた. これと同じプロセスが新しい事例が入力された時の想起において用いられる。

事例間の様々な関係づけを行うためには, 属性値間に様々な角度から関係づけがなされる必要がある. 次の例はこのプロセス診断に用いられた様々な関係のうちの一部である。

```
(relation name=opposite arg1=high arg2=low)
(relation name=positive arg1=open arg2=high)
(relation name=near arg1=normal arg2=high)
```

<sup>1</sup>この節は, [石田 87] を詳細化した報告であり, [Ishida 89] からの抜粋である。



CAOS では、事例はデータタイプとエフェクトタイプの両方で索引づけされている。想起（類似事例の探索）はこの二つの表現の上で行われる。

#### 事例操作—関連事例の変形

想起された事例およびその事例と与えられた事例との関係に基づいて、いくつかの事例操作の候補が選ばれる。事例操作は、大まかに2つの種類に分類される。すなわち、単一事例操作と複数事例操作である。単一事例操作は、一つの関連事例の情報のみを用いるのに対して、複数事例操作は複数の事例から情報を抽出する。この節では、このような事例操作が、現事例と関連事例の関係を用いてどのように選択されるかを主に述べる。

また、これらの事例操作により現事例の未知の故障が推論できるためには、故障の様子が図1のようなイベント空間で関連づけられていなければならない。これらのボキャブラリやそれらの間の関係は領域に依存する。

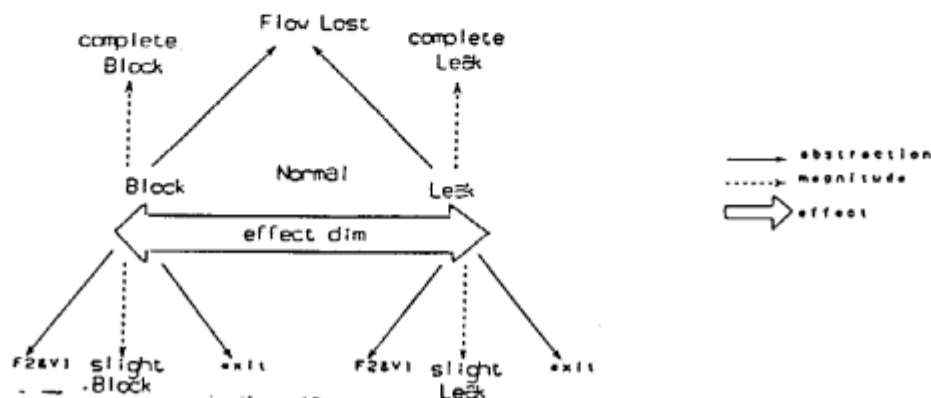


図1 事例操作のためのイベント空間

構造化された事例ベース、事例間の様々な関係を規定するためのボキャブラリ、および上記のイベント空間を用いて事例操作が行われる。次のものは、事例操作のいくつかの例である。これらの事例操作は、より強いヒューリスティックが先に適用されるようルール制御されている。

未知の故障を推論するシナリオを下記に挙げる。

与えられた事例、

```
(event name=target status=current type=data
 F1=high F2=high L1=high V1=open)
```

に対し、全く同じ事例は事例ベースにはないものとし、次の事例が最類似事例として得られたとする。

```
(event name=ThroughputLow_27 type=data
 fault=low location=throughput
 F1=low F2=low L1=low V1=close)
```

すでに、指摘されたようにこの事例は、データタイプは全く異なるが、エフェクトタイプにおいては最も近い。実際、データタイプにおける距離は1であるが、エフェクトタイプにおける距離は0である。

この操作を行う事によって、`fault=high location=throughput`が推論される。なぜなら、それはスループット (throughput) からインフロー F1、アウトフロー F2、バルブ V1、レベル L1 への因果関係に矛盾しないからである。

以下で、典型的な単一事例操作と複数事例操作をそれがどのようにいて選ばれるかの例とともに列挙していく。

・抽象化: もし関連事例とのエフェクトタイプ距離がしきい値以下で、関連事例が現事例の上位事例ならば、故障タイプを抽象化する事例操作を候補として提案する。例えば、与えられた事例:

```
(event name=target status=current type=data
 V1=open F2=low)
```

関連事例:

```
(event name=Leak_41 type=data fault=leak
 V1=open L1=high F2=low)
```

推論された結果:

```
fault=FlowLost
```

・程度変化: もし関連事例が現事例と程度のみが異なるなら、故障タイプを同じにして程度のみを変化させる事例操作を候補として提案する。この関係は故障タイプを決定するのに重要な症状のみが一致し(この場合 DF, V1)、他のあまり重要でない症状の値の程度のみが異なる時に適用される。

与えられた事例:

```
(event name=target status=current type=data
 F2=high V1=close DF=-)
```

関連事例:

```
(event name=Leak_43
 type=data fault=leak
 F2=normal V1=close DF=-)
```

推論された結果:

```
fault=leak and magnitude=big
```

これらの単一事例操作以外に、次の様な複数事例操作がある。もし現事例に最類似の事例同士がある関係を満たすならば、複数事例操作の適用が可能である。次のものは、典型的な複数事例操作の例である。

・外挿 (または内挿):

もし関連事例がその値が故障タイプと相関を持つような症状 (この場合 F2) を持つならば、故障空間で故障タイプを補間 (外挿、内挿) することにより現事例の未知の故障タイプを求めよ。例えば、

与えられた事例:

```
(event name=target status=current type=data
 location=F2&V1 F2=low V1=normal)
```

関連事例:

```
(event name=LeakInF2&V1_2 type=data
 fault=leak location=F2&V1
 F2=high V1=normal)
(event name=NormalInF2&V1_1 type=data
 fault=normal location=F2&V1
 F2=normal V1=normal)
```

推論された結果:

```
fault=block location=F2&V1
```

・結合: この事例操作は二つ以上の事例が現事例にエフェクトタイプで関連し、その関連事例同士は全く独立 (エフェクトタイプが無関係) のとき、事例操作の候補として挙げられる。この事例操作は現事例が複数故障であるとき、それを独立な複数の事例を結合して推論する操作である。

典型的な複数事例操作の一例として、この結合操作を事例操作の候補として提案する OPS83 [Forgy 86] で書かれたルールを図 2 に示す。

```
rule FaultType_propose_combination_mul
{
 &g (goal PhaseStatus=current; step=propose;
 ProblemSpace=FaultType);
 &e1 (event status=current);
 &d (distance from=&e1.id; type=effect;
 value<=&g.Edistance);
 &d1 (distance from=&e1.id; to<>&d.to;
```

```

 type=effect;
 value<=#g.Edistance);
 ~(distance from=#d.to; to=#d1.to;
 type=effect;
 value<0.5);
 &e2 (event id=#d.to; type=data);
 &e3 (event id=#d1.to; type=data);
 - (relation arg1=#e2.fault; arg2=#e3.fault;
 name=opposite);
 -(operator object=#d.to; object2=#d1.to;
 (@.PhaseStatus=done\
 @.PhaseStatus=pending);
 ProblemSpace=combination);
-->
 modify #g (Edistance=0.0; Ddistance=0.0);
 make (operator id=gint ();
 PhaseStatus=pending ;
 status=proposed;
 ProblemSpace=combination;
 priority=5;
 object=#d.to;
 object2=#d1.to);
};

```

図 2 二つの独立な故障を結合する  
事例操作を提案する OPS83 ルール

このルールを訳すると次のようになる。

もし現事例と関連事例のエフェクトタイプ距離がしきい値以下で、かつ関連事例の相互の間にエフェクトタイプの関係が無ければ、現事例の故障タイプをこれらの関連事例の故障タイプを結合した複合故障とする事例操作を提案せよ。

このルールにおいて、二つの類似事例間の独立性は、これらの間のエフェクトタイプの関係の不存在により表現されている。この事例操作がいかに作用するかを以下の例で示す。  
与えられた事例：

```
(event name=target status=current type=data
 F2=low L1=low V1=open)
```

関連事例：

```
(event name=BlockInF2&V1_15 type=data
 fault=block location=F2&V1
 F2=low V1=open)
(event name=AnomalyInControl_31 type=data
 fault=anomaly location=control
 L1=low V1=open)
```

推論された結果：

```
fault=block location=F2&V1 and
fault=anomaly location=control
```

抽象化（総合）は複数事例操作においても可能である。すなわち、類似事例として漏れ(leave), 閉塞(block)の反対の事例があるときにこれらの事例を抽象化してフローの喪失(flow lost)とするなどである。

#### 5.6.2 インプリメンテーションおよびサンプルセッション

試験的事例ベース診断システム CAOS が上記の簡単なバッファータンクの診断についてインプリメントされた。CAOS はプロダクションシステム OPS83 上にインプリメントされている。事例ベースは作業領域に、ネットワーク構造として表現された作業要素として表現されている。いくつかの事例操作の候補の中から、領域固有のヒューリスティックルールや優先度を考慮したデフォルトルールにより、ただ一つの事例操作が選択され実行される。この事例操作選択部のルール制御メカニズムは SEAR [van de Brug 86] のそれを OPS83 用に改訂したものを用いている。

図7に CAOS による事例ベース診断のサンプルセッションを掲載する。推論は次の4つのタスクを次々に行うことにより進められる。すなわち(1)未知の故障を含む現事例の入力、(2)データタイプについての想起、およびエフェクトタイプについての想起、(3)適当な事例操作を一つ選んで適用する、(4)事例操作の結果を出力する。

```

CAse Operation System

1 Input new case
2 Structure cases
```

3 Case-based reasoning  
4 List structured cases  
5 Output case

Enter the number: 3

\*\*\*\*\* Input the new case.

Input the case name: target

Input the attribute value pairs (E to end):

V1 close

F2 high

L1 high

E

\*\*\*\*\* Reminding by data type.

target =super=> LeakInF2&V1\_7

F2 high

V1 close

target =super=> Anomaly\_30

V1 close

L1 high

\*\*\*\*\* Reminding by effect type.

target =super=> Anomaly\_30

L1 negative\_V1

V1 negative\_F2

target =super=> LeakInF2&V1\_7

V1 negative\_F2

\*\*\*\*\* Searching for case operation.

data distance = 0.9

effect distance = 0.9

operator combination is selected for cases:

LeakInF2&V1\_7    Anomaly\_30

\*\*\*\*\* Output case operation results.

anomaly at Control

leak at F2&V1

### 図3 CAOS のサンプルセッション

この例では、故障タイプが未知の現事例

```
(event name=target status=current type=data
 V1=close F2=high L1=high)
```

に対し、次の二つの事例がエフェクトタイプが近いという理由で想起されている。

```
(event name=LeakInF2&V1_7 type=data
 fault=leak location=F2&V1
 V1=close F2=high)
(event name=Anomaly_30 type=data
 fault=anomaly location=control
 V1=close L1=high)
```

これらの事例は相互に独立であるとみなせるので（エフェクトタイプにおいて無関係）、これらの二つの故障を結合して現事例の故障を推論する事例操作が行われる。その結果、現事例の故障はこれらの関連事例の二つの故障を含むと診断される。

この様な成功した診断例の他に、誤診の例として次のようなものがある。すなわち、現事例

```
(event status=current V1=open F2=high
 type=data)
```

で症状 F1=high が落ちていたとき、次の事例がエフェクトタイプにおける最類似事例として選ばれた。

```
(event name=LeakInTank_9 V1=close F2=low
 fault=leak location=tank type=data)
```

症状は現事例と全く反対であるので、故障タイプを反転させる事例操作が行われ「タンクのつまり (tank is block)」という診断が得られた。もし現事例の症状が正しく与えられていると、

```
(event name=ThroughputLow_27 F1=low V1=close
 F2=low fault=low location=throughput
 type=data)
```

が、この事例操作を行うための関連事例として選択されるはずである。従ってその場合、診断は「スループット高 (throughput is high)」となる。

ほとんどの誤診は、このように不適当な事例を想起してまって事例操作を行う事によって起こった。これは、質的に異なった故障が索引付けに置いて近くに位置してしまうことにより起こる。従ってエフェクトタイプ以外の他の索引づけの方法がこれらの誤診の回避に必要であろう。あるいは、これらの表面上似通った事例に異なった事例操作を施すように事例操作選択ルールを洗練することが必要である。

### 5.6.3 ルールベース推論との比較

事例ベース推論では、事例自身が知識源の一つであるから事例からルールを抽出する必要はない。その様な一般化機構は動的記憶の事例構造化メカニズムのなかに組み込まれている。しかし、事例はそれらがその構造により相互参照できるように索引づけされなければならない。さらに事例ベースを構成する事例は様々なパターンのものを選ぶ必要がある。ほとんど同じ類の事例からなる事例ベースからは、柔軟な事例ベース推論は行えない。

事例ベース推論とルールベース推論は、その知識表現に置いて大いに異なり、従って推論方法も大いに異なるが、それらを一つのシステムのなかで共存させることは可能である。この二つを結合させる方法はいろいろある。例えば SEEK [Ginsberg et al 88] は基本的にはルールベースシステムであるが、ルールを洗練するのに事例を用いている。CAOS では、事例操作部にルールベースシステムを用いた。

このルールベースと事例ベースの結合パラダイムは、ルールベースパラダイムを基盤とする多くの知識ベースシステムで起こっている知識獲得の問題に活路を見いだすかもしれない。ルールベースシステムでは、一度その領域のヒューリスティックがみいだされ、ルールとして表現されると固定されてしまう。知識獲得の問題は、事例が得られるごとに変動する知識も、領域が決るとある程度固定される知識も一緒にルールベースのなかに押し込めることに起因すると考えられる。より固定的な知識をルールベースに、事例により



変動する知識を事例ベースに振り分けることにより、ルールベース変更の機会はより押えられる。しかし、本論文でも論じられたように、事例ベースの導入は新たな事例獲得の問題を引き起こす。すなわち、(1) 事例ベースを構成する事例の量でなく、事例の分布にも依存する。(2) 事例をいれる順序も、事例ベースの構造に影響を与える。

#### 5.6.4 むすび

提案された事例ベース診断の枠組で示されたように単なる症状だけでなく、症状間の因果関係により索引することにより柔軟な想起が行えることが示された。このことは、人間の専門家が現事例にたいしてより深い類似性により過去の事例を想起するのに対し、素人は表面的な類似性のみで想起するのに対応している。新たな索引づけを導入することによりより柔軟な想起、従ってより高度な事例ベース診断の行われる可能性が示唆される。

CAOS は特にその事例想起サブシステムだけでも人間の専門家の意志決定支援システムとして有用である。すなわち人間は過去の膨大な事例の中からある構造的関連性を持った事例にアクセスでき、その類似事例の情報は人間の意志決定を大いに支援するものと考えられる。

事例獲得に関して得られた戦略としては、事例はいろいろなものからサンプルされなければならない事、およびそれらの事例をいれる順序に注意すべきことである。

## 参考文献

- [石田 87] 石田好輝: 知識ベース故障診断システムの動向、知的診断を目指して、信学技報、DE-87-12,1-8 (1987).
- [Ishida 89] Y. Ishida: A Framework for Dynamic Representation of Knowledge: A Minimum Principle in Organizing Knowledge, Proc. KR89 ,pp.170-179, (1989).
- [Forgy 86] Chales L. Forgy: OPS 83 User's Manual. Carnegie-Mellon University, (1986).
- [van de Brug 86] A. van de Brug, J. Bachant, and J. McDermott: Taming of R1. *IEEE Expert*, 1(3):33-38, (1986).
- [Ginsberg et al 88] Allan Ginsberg, Solomon M. Weiss and Peter Politakis: Automatic Knowledge Base Refinement for Classification System. *Artificial Intelligence*, 35:197-226, (1988).

## 5. 7 事例を用いた対話型機械調整支援システム

日本たばこ(株)生産技術研究所 中村孝太郎

### 要 旨

エキスパートシステム等で使用される知識ベース推論を補完する機能として事例ベース推論を位置づけ、機械調整支援への適用を行った。領域知識が不完全な場合に必要に応じて推論過程へのユーザの介入を要請する対話型事例ベース推論モデルに基づく方式を検討し、システムを実現した。本システムの機械調整支援への適用を試みた結果、既存のエキスパートシステムで有用な調整方法が見出せない場合に、事例ベース推論を対話的に用いて調整方法案が得られることが示された。すなわち、知識ベース推論を補完する機能として事例ベース推論が有用であることが確認された。

#### 5.7.1 はじめに

問題解決における事例ベース推論の意義としてつぎのことが指摘される。

第1に、知識ベースによる問題解決を補完する機能として事例ベース推論を位置づけることにより、知識獲得負荷の問題を緩和することが期待できる。第2に、探索（生成検査の繰り返し）による問題解決に先行する機能として事例ベース推論を位置づけることにより、探索コストを軽減することが期待できる。第3に、事例ベース推論による問題解決自身が新しい事例の獲得につながり、問題解決の性能を段階的に向上させることが期待できる。

このように、事例ベース推論を知識ベースによる問題解決と探索による問題解決の中間に位置づけることにより、知識獲得の緩和および探索コストの軽減を期待することができる。本論文では、事例ベース推論において領域知識が不完全な場合を想定して、必要に応じてユーザの介入を要請する対話型モデルを提案する。さらに本モデルに基づくシステムを実現し、機械調整支援への適用を試みて、知識ベース推論を補完する機能として事例ベース推論が有用であることを確認する。

#### 5.7.2 対話型事例ベース推論の方式

対話型事例ベース推論モデルについて説明し、各推論ステップの具体的方式について述べる。なお、対話型事例ベース推論モデルおよび領域知識については、本レポートの4.6節において既述しているため、ここでは簡単に概要を述べる。



はユーザに委ねる。理由の解析に従って事例の特徴づけの変更を行う際に、概念階層的知識が使われる。

このように、問題の特徴づけ、事例の修正および事例の修復では、領域知識に基づく推論処理を基本とするが、必要に応じてユーザが介入するとともに、ユーザの判断を領域知識が支援する。この方式の実現にあたってはシステムとユーザ間の柔軟な対話機能が必要とされる。

#### 5.7.2.2 対話型モデルにおける推論過程

本節では対話型事例ベース推論の推論過程の詳細について述べる。

##### ①問題の特徴づけ

問題の特徴づけは、与えられた問題の特徴の分析、およびこれに基づくインデックスの設定からなる。

問題の特徴分析では、問題の領域に想定される問題のタイプおよびタイプごとの属性項目を概念階層的知識から参照し、問題と照合するものを選択する。選択された属性項目の充足は基本的にはユーザ入力によるが、必要なら領域知識に含まれる属性値用語集が参照可能である。

インデックスの設定では、問題の特徴分析において属性値の充足された属性項目の中から、事例の検索のためのインデックスとする属性項目の選択を行う。したがって、選択された属性、すなわち属性項目とその属性値の対が、インデックスとなる。なお、本モデルでは、このインデックスは次の②で述べる一次的な検索を行うために使われる。そして充足された属性項目のうち、インデックスとなったもの以外の属性項目は、類似度の計算に使用される。

なお、属性値は言語表現を基本とする。属性値の入力は、概念階層的知識を反映するテンプレートエディタを用いてユーザの判断処理を追加して行われる。

##### ②事例の検索

事例の検索では、問題の特徴づけで設定されたインデックスに基づいて事例ベースを検索し、現在の問題にその特徴が照合する事例を取り出す。検索の手順は以下の順である：a)設定されたインデックスにより検索する、b)検索された事例集合について現在の問題との類似性を判定する、c)類似性の高いものを任意個取り出す。

ここでインデックスによる検索は、事例の編集時に入力された事例のもつ共通のインデックス（属性項目と属性値）とのマッチングにより行う。類似性の判定は、現在の問題に含まれる属性項目と属性値および検索事例に含まれる属性項目と属性値同士を比較し、類似度を計算することにより行う。本検索方法における類似度は以下を考慮して決定される。

a)事例のもつ属性項目のうち、現在の状況において成立する（設定された）ものがいくつあるか。

b) その成立する属性項目のうち、さらに属性値も現在の問題と共通のものはいくつあるか。

類似度の計算式を次に示す。すなわち、 $k$  個の属性および属性値を有する事例  $C(k)$  の現在の問題  $P$  への類似度  $SD(C(k), P)$  は次式である。

$$SD = \sum_{i=1}^k w_i \cdot \alpha_i \cdot (1 + w_i' \cdot \beta_i)$$

$SD$  : 類似度

$\alpha_i$  : 属性  $i$  が成立するかを表す値

$\beta_i$  : 属性  $i$  が成立する場合、属性値も共通かを表す

$w_i$  : 属性  $i$  の  $\alpha$  値に対する重み (0~1)

$w_i'$  : 属性  $i$  の  $\beta$  値に対する重み (0~1)

$$\alpha_i = \begin{cases} 1 & (\text{成立時}) \\ 0 & (\text{不成立}) \end{cases} \quad \beta_i = \begin{cases} 1 & (\text{共通時}) \\ 0 & (\text{不共通}) \end{cases}$$

なお、上記②の共通の属性値かどうかの判定は、全く同一の値である必要はなく、属性値間にテキストデータ上の包含関係があれば、共通のものとみなす。また、属性項目ごとの重みづけを変更することにより、あらかじめ類似性判定機能の調節をしておくことが可能である。

### ③事例の評価

事例の評価では、問題に適用する上で考慮する必要がある評価基準により、複数の検索事例の評価づけを行い、任意の基準を満たすものを選択する。評価基準の例としては、事例に含まれる解法に関する情報の詳細度等がある。また問題の特徴づけにおいて、インデックスの設定に反映されなかった基準も本ステップで考慮される。例えば、事例の成功／失敗の評価がある。本基準は推論の目的に依存しており、問題の解法を導くことを目的とする場合には、成功事例が選択され、適用時の失敗を回避することを目的とする場合には、失敗事例が選択される。

### ④事例の修正

事例の修正は、選択事例について問題と照合しない部分の変更、および事例に含まれる解法の問題への変換からなる。

選択事例の変更では、問題と事例の条件部間で照合しない属性項目のうち修正が可能な属性項目や、解法に関連する属性項目の修正が行われる。修正箇所とその内容は、修正知識によって明示され、ユーザはこれにしたがい必要な属性に対して修正操作を施す。

解法の問題への変換では、選択事例の条件部の修正に応じて、解法を修正し、これを問題に適用できるように変換する。提示された解法に対して、ユーザは概念階層的知識を参照しながら修正操作を施す。

ユーザの判断による修正操作を容易化するために既述のテンプレートエディタが使用される。ユーザは属性値用語集を参照し、部分複製等によりこれを活用できる。

なお、修正された事例は、実際の問題に適用されるまで一時的に保持される。

### 5.7.3.1 システムの構成

システム構成を図5.7-2 に示す。本システムの基本構成は、事例ベース推論機能、事例ベース管理機能の基本機能、および事例ベースからなる。

事例ベース推論機能は問題の特徴づけ、事例の検索、事例の評価、事例の修正、事例の修復からなる一連の事例ベース推論過程を実行する。推論の実行は領域知識を参照しつつ、必要に応じてユーザによる判断を取り入れながら事例ベース中の事例データの取り出しと成功および失敗事例の登録を基本に行われる。

事例ベース管理機能は、入力ガイド、事例の形式化、および事例の詳細化からなる編集過程を実行する。編集過程は領域知識のガイドに基づきユーザが事例の入力を行った上で、事例ベースに登録する。事例を参照入力する場合は、事例ベースから類似事例を取り出し参照する。

事例ベースは、登録された事例の集合である。1 つの事例の表現はフレーム構造を用いる。フレームがもつ属性値は通常の言語表現の他、概念階層的知識に基づき階層構造上の1 つ下位の概念のインスタンスをもつことも可能とする。概念階層的知識の中で、事例や問題を構造化するための知識は、修正知識も含めてフレーム定義形式で表現され、事例の修正や修復において参照される対象領域固有の概念知識は述語表現形式で表現されている。

ユーザは、事例ベース推論マニピュレータ（以下、CBRマニピュレータと呼ぶ）を介して事例ベース推論機能に対し、補助的操作や制御を行い、事例テンプレートエディタを介して事例入力を行う。

本システムは、ESP 言語上に実装されている。フレーム定義はESP のオブジェクト指向機能のクラス定義を用いた。したがって、各事例に備わる属性はクラススロットに相当する。同様に階層構造中の詳細化可能な属性項目の表現もESP のクラス定義を用いた。

### 5.7.3.2 ユーザインターフェイス

CBR マニピュレータは、事例ベース推論の制御をユーザおよびシステムが効果的にを行うために設けられる。ユーザはこれを用いて、推論過程の起動や監視、および「推論モード」設定等を行う。

事例テンプレートエディタは、事例の修正や編集時の事例の属性項目、属性値の一覧表示およびユーザによる修正・入力の支援を行う。さらに事例の属性項目ごとに詳細化を行うかどうかの設定操作も行う。また事例ベース推論の問題の設定時にも本エディタが用い

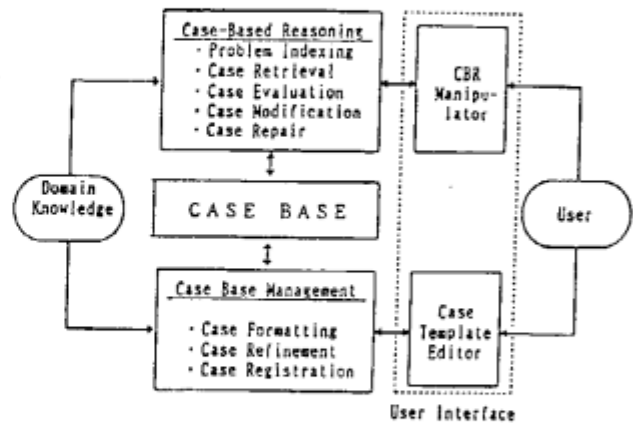


図5.7-2 対話型事例ベース推論システムの構成

図5.7-3 に事例ベース推論のユーザインターフェイス例を示す。CBR マニピュレータですでに設定済の推論パラメータにしたがった事例ベース推論が展開されている。現在の問題を入力するためのテンプレートエディタが表示され、入力操作終了とともに、事例の検索が行われ、ワーキングメモリ表示が更新され、検索事例がウィンドウ上に表示されている。事例ベース推論の現在の所在は、事例ベース推論マニピュレータにより推論過程に同期して表示される。

図5.7-3 ユーザインターフェイス例

本システムをたばこ製造ライン中の自動化機械の一つであるシガレット製造機の機械調整支援に適用する。本節では、機械調整の特徴と事例ベース推論の有効性について述べ、事例ベース推論の役割を説明し、実行例と評価を示す。

シガレット製造機は、たばこ刻み葉から1本のシガレットを作る（最大8000本/毎分）機械である。機械内部に各種の多数の製品検査機能を有し、シガレット1本ごとの検査を可能としている。複合的な加工工程を高速連続運転すること、および多品種生産を行うために機械条件の正確な変更を要すること等の事情から、本機の調整作業は高いノウハウを

必要とする。そこでこの調整作業を支援するため、すでに診断型のエキスパートシステム（診断ルール数、約4200個、フレーム数、約2100個、説明用画像1200枚使用）が開発された。本システムの開発を担当した調整専門家の意見および作成済の調整基準フレームデータの検討から、調整事例データの存在と利用可能性が認められた。

すなわち、機械調整の特徴は、①調整作業は反復的であり、調整結果が満足するまで繰り返し調整を行う。②調整作業—調整結果—評価の対を調整事例として記憶（記録）し次の調整に活かす。③製品銘柄や機械条件等の変動要因による以前と“やや異なった場合”の対応を要する。

調整事例における事例とは、調整目標、調整条件、調整方法、根拠、調整結果、評価、および参考事例の7つの属性によって記述される実行または方法の具体例をいうものとする。さらに調整方法は調整基準と調整操作からなる。調整事例は図5.7-4に示すように4つのタイプに分類される。実行事例には調整結果とその評価が、方法事例には根拠が、更新事例には参照事例が、それぞれ付加されていることに注意されたい。

本システムでは、これらの事例のタイプおよび各タイプの属性構造は、概念階層的知識として定義される。

この概念階層的知識にしたがって、専門家が作成したフレームデータ中の事例表現を形式化し、事例ベースの編集、登録機能により事例ベースが構築された。事例の数は50個程度である。また、これらの事例に関連する機械部分の部品階層知識（PART-OF 知識）も領域知識として格納された。

機械調整では、稼働上や品質上の不具合が解消されることを調整の目標として設定し、調整作業を行うが、標準的な調整基準では目標を満足できない場合には調整事例をもとに調整案のプランニングを行う。これを問題解決と位置づけると、その過程は、過去の類似の事例を想起（検索）し、現在の状況に最も類似のものを選択し、必要な部分を取り出し、詳細な調整方法の明確化をはかったり、適用可能性の確認を行うことからなる。以上の過程を事例ベース推論により実現するものとする。

#### 5.7.4.2 実行例

本システムの挙動を調べるため、「のり量、巻紙位置が適正」となる調整目標をもつ時に「巻紙位置が機械手回し時には

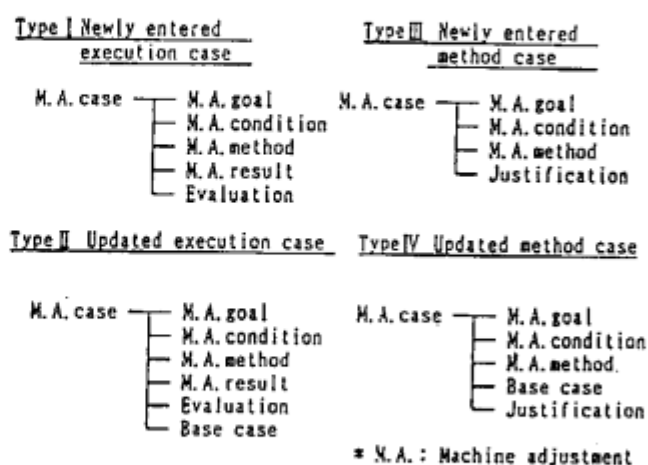


図5.7-4 機械調整事例のタイプ



基準以内だが、実際に高速運転された時、振れが大きくなり巻紙位置も不安定となる」状況での機械調整方法を求める問題について実行した。

まず機械調整支援用エキスパートシステムで初期症状として、「のり量が適正だが、巻紙位置が適正でない」場合を設定し、点検結果等を入力したところ、原因仮説を「ヒータ位置不良」や「のり付け車のあたりが悪い」と判断し、調整方法として「ヒータ取付け位置基準調整」や「のり付け車位置調整」等を提示した。

高速運転された時に不具合となることについての解決は見出せないため、次に同様な場合を想定し事例ベース推論システムに適用した。まず、問題の特徴付けにおいて、調整目標に「のり量が適正、巻紙位置が適正」を設定した上で、それ以外の属性に対しても領域知識に用意されている表現例を参照しながら入力し事例の検索を行った。事例の評価ステップにおいて、検索された3つの事例のうち類似度の大きいものを選択した結果、「のり付け車」に隣接する「ワイドベーストホイールが高速運転時に振れが大きくなり、その場合、シャフト交換を行うことにより振れが小さくなった」調整事例が見出された。図5.7-3の実行例はこの部分に相当する。

事例の修正ステップでは、部品階層知識を参照した結果、概念上の包含関係が認められたため、調整条件の表現中の「ワイドベーストホイール」を「のり付け車」に換え、調整結果の「. . 小さくなった」を空白にし、さらにシャフト交換等に関する調整基準や調整操作について詳細化し、調整案とした。

#### 5.7.4.4 評価

本システムの評価に関して、まず確認された事例ベース推論システムの利点を述べ、さらに本システムの問題点と課題を述べる。

実行例により、エキスパートシステムで満足できる解が得られない状況の時に事例ベース推論を対話的に用いて調整案が得られることが確認できた。すなわち、経験が知識として未だ一般化されていない領域における問題解決をガイドすることが可能であることが示された。

対話型としたことにより事例ベース推論の推論過程において領域知識に基づく推論に対して、必要に応じて柔軟にユーザの判断を取り入れることが可能となった。これにより、事例ベース推論を適切に用いて問題解決に使用することができる。例えば、問題の特徴づけでは、領域知識に含まれる属性の表現例を基にユーザが選択し、事例の修正では、部品階層知識を参照することにより、問題と事例間で属性値の包含関係を見出し、ユーザの修正操作に反映する等である。したがって、ユーザと領域知識との協調を行うことにより、ユーザが専門家の場合においても、そのレベルに応じて、事例ベース推論を使い分けられることが可能である。

本システムでは、初期登録された事例を利用するだけでなく、事例ベース推論により生成された問題の解法に対して、実際の適用結果も登録することにより、事例ベース推論に

再利用することが可能である。また、これらを効率よく行うため、ユーザによる事例の編集や登録を領域知識に基づき支援する事例ベース管理機能が用意されている。したがって、システムの使用回数が増えるほど、事例ベースを高度化でき、有用なシステムとすることができる。

次に本システムの問題点と課題を述べる。

事例ベース推論の推論過程においては領域知識を向上するほど、システムの自動化の程度を高めることが可能である。例えば、概念階層的知識が洗練されるほど、部品間の類似性や共通性を判定する照合機能が高まり、推論性能の向上が期待できる。しかし本システムでは、検索時の類似性判定には、現在のところ階層構造同士の照合機能はなく、判定機能に限界があるといえる。

本論文では、失敗事例の利用や事例の修復に関しては、その有用性を示すまでには至っていない。事例の修正、修復時にもユーザへの依存度の少ない事例ベース推論システムを構築するためには、適用領域の階層構造的知識の確立はもちろんのこと、因果的モデルの利用等が必要である。本システムの適用対象となった製造機械では物理的因果関係が比較的明確であるため、因果的モデルが利用可能と思われる。

本システムでは、知識化される以前の経験を、事例として蓄積し、利用する枠組みを確立することにある程度成功したが、さらに事例が十分蓄積されれば、これを一般化〔中村90〕して知識に変換するための学習機能を付与することも検討することも検討する必要があるだろう。

#### 5.7.5 おわりに

エキスパートシステムで使用されている知識ベース推論を補完する機能として、事例ベース推論を位置づけ、実用的見地から領域知識が不完全な場合を想定して、必要に応じてユーザの介入を要請し、両者の協調をはかる対話型事例ベース推論モデルに基づく方式を検討した。

本モデルを機械調整領域に適用するために、機械調整事例の分析および問題解決過程の同定を行った上で、システムを実現した。機械調整において標準解である調整基準に満足するものがない場合に、事例ベース推論を対話的に用いて調整案が得られることが確認できた。すなわち知識ベース推論を補完する機能として事例ベース推論が有用であることを示した。

本システムはユーザの熟練レベルに対して柔軟に対応できる可能性がある。またシステムによる問題解決自身が新しい事例の獲得につながり、問題解決の性能を段階的に向上できる可能性を示した。

#### 参考文献

[Hammond 86-1] Hammond, K. J. et al. : Extracting Diagnostic Features from Explanations, DARPA Workshop on Case-Based Reasoning, 169/178 (1986)

[Hammond 86-2] Hammond, K. J. : CHEF: A Model of Case-Based Planning AAAI-86, 267/271 (1986)

[Navinchandra 88 ] Navinchandra, D. : Case Based Reasoning in CYCLOPS, a Designing Problem Solver, DARPA Workshop on Case-Based Reasoning, 286/301 (1988)

[Michalski 87] Michalski, R. S. et al. : A Theory and Methodology of Inductive learning, Machine learning, An Artificial Intelligence Approach, Springer-Verlag, 邦訳: 電総研人工知能研究グループ (訳), 知識獲得入門、91/143、共立出版 東京、(1987)

[Sycara 88 ] Sycara, K. : Using Case-Based Reasoning for Plan Adaptation and Repair, DARPA Workshop on Case-Based Reasoning, 425/434 (1988)

[中村 90 ] 中村、小林 : 因果的モデルと操作性規範に基づく機械調整事例の一般化、計測自動制御学会、第11回知識・知能システムシンポジウム予稿集、pp99-104(1990)

## 5.8 事例に基づく知識獲得支援システム

シャープ(株)情報システム研究所 森下太朗、椿和弘

### 要旨

我々の研究の目的は、事例ベース推論を利用して異なるドメインの問題間で共通に使われている知識の抽出を支援することである。本システムは事例ベース推論機構と共通知識獲得支援機構からなる。事例ベース推論機構では、同一ドメイン（プリント基板素子配置問題）の類似事例から初期の解を生成し、初期の解に対する失敗を予測する。次に、類似した失敗を回避する方法が記述された他ドメイン（オフィスレイアウト問題）の事例から、不足していた失敗回避法を抽出・適用し、初期の解を手直ししてより良い解を提示する。また、枠組としてESP及びKL1にて記述したCYCLOPS風のエンジンを作成した。共通知識獲得支援機構では、抽出した失敗回避法をドメインセオリーを用いて、問題間に共通な表現で一般化し、共通知識を生成する。さらに、獲得された共通知識は、他の問題にも適用可能な形で登録される。このような動作をする実験システムを作成中である。以下に2つの機構毎に説明する。

### 5.8.1 事例ベース推論機構

配置問題を題材とした実験システムを例に事例ベース推論機構について説明する。

#### (1)対象とする問題

L S Iやプリント基板などのレイアウト設計を主問題とする。回路の構成表を与え、基板への配置結果を解とする。参照事例としては自分自身のドメインにおける事例以外にオフィスレイアウトにおける事例を用意する。また、設計法は「階層レイアウト設計法」である。

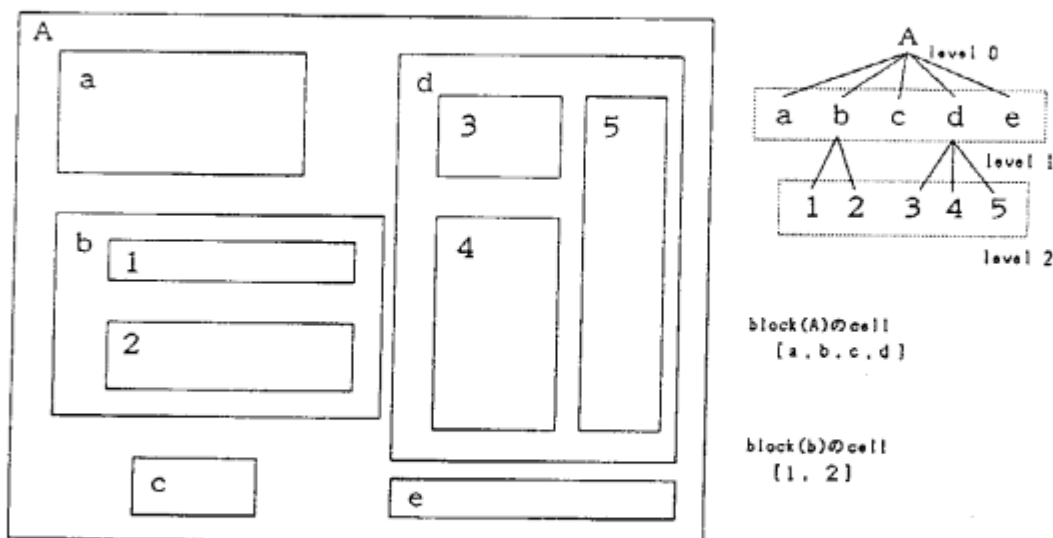


図5.8-1 階層レイアウトの基本概念

現在は素子レベルからのボトムアップ設計のみを扱っている。なお、配置位置に関する理由には以下のような項目がある。

- ・ある規則により、構成物は連続した位置に配置される
- ・主に電源供給の立場から、いくつかの構成物の位置が固定あるいは制限される。
- ・主に配線長を押える立場から、いくつかの構成物の組は近くに配置される。

## (2) システム構成

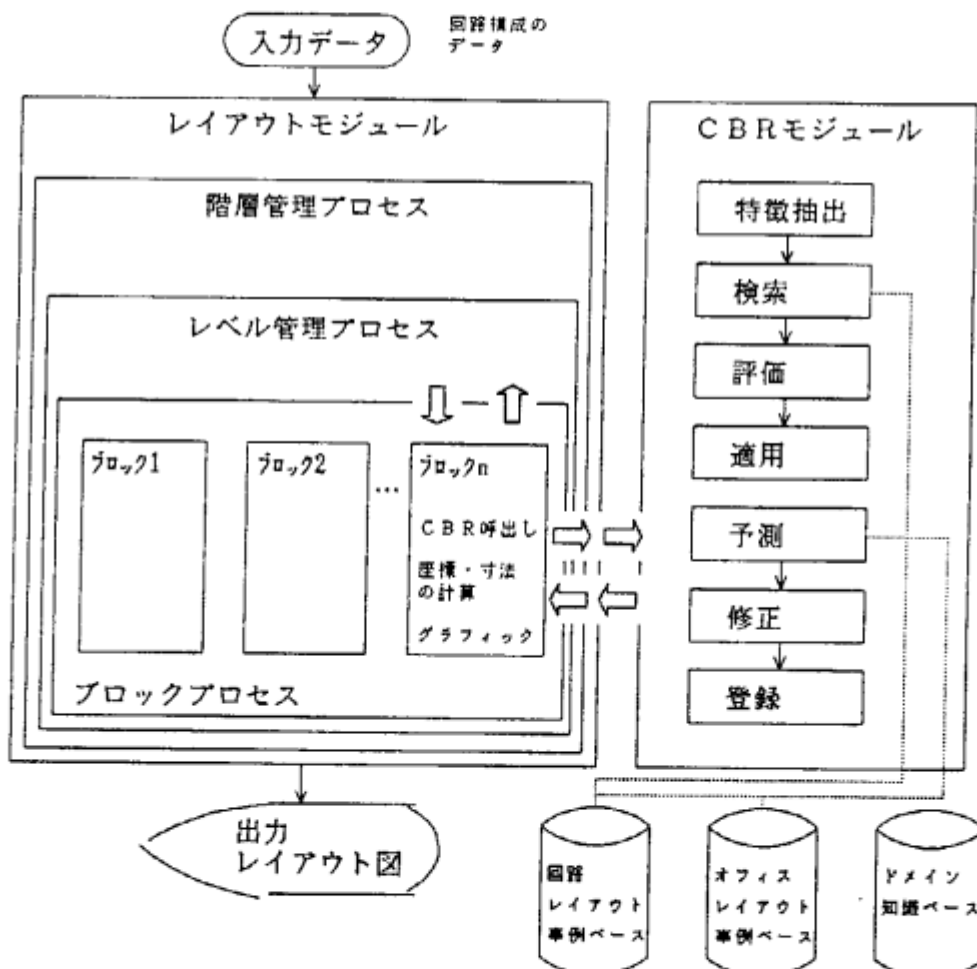
○レイアウトモジュール 各ブロック階層毎に並列にセルの座標計算を行い、計算結果をグラフィック表示する。

○CBRモジュール 過去の類似事例あるいは失敗・失敗回避例を検索・適用する。

レイアウトモジュールの処理の概略は以下のとおりである。

- ・問題(階層別の回路構成物等のデータ)の読み込み、levelとblockの内容のセット。
- ・level毎の各block内のcell配置処理(cellの座標・寸法の計算)と配置結果の表示。
- ・基板に対する最終配置結果の表示。

CBRモジュールは常にレイアウトモジュールのblock内のcell配置処理から呼び出される。



<システム構成図>

部品名・cell名等のデータから、同一ドメインの以前の事例を利用して初期配置を求めた後、失敗を予測してユーザに確認を求めながら、再配置を行いより良い配置結果をレイアウトモジュールに返す。

### (3) 事例表現

事例ベース内の事例は、値がセットされたスロットの集合として表現される。下図は、構成すべきスロットの種類を分類したものである。問題(以降、問題事例)は与えられた既定のスロットの値と、ゴールスロット(ゴールスロットのリスト)により表現する。

|           |                  |                                |
|-----------|------------------|--------------------------------|
| レイアウト事例―― | 属性               | : cell数、cellの名前、機能名、接続関係、回路特性、 |
|           |                  | 素子特性、cellの大小など                 |
|           | ト解               | : 各cellの配置方向(横か縦か)及びcell同士の相対的 |
|           | 位置関係(隣接、並んでいる順序) |                                |
|           | ト失敗              | : 失敗の状況、失敗の説明、考慮されなかった制約       |
|           |                  |                                |
|           | ト失敗回避            | : 初期の解の修正ステップ                  |

(例) ・事例ベース内の事例(タッチスイッチのメモリー回路)

```
case(1,
 [{ドメイン,概略配置},{サブドメイン,回路},
 {名称,メモリー回路},{機能,記憶},
 {構成要素,[ic(TC40H00P),R1(10MΩ),R2(10Ω),R3(10Ω)]},
 {入力信号線数,2},{出力信号線数,1}{要素数,4},
 {配列,[向き([{ic(TC40H00P),縦},
 {R1(10MΩ),横},
 {R2(10MΩ),横},
 {R3(10MΩ),横})],
 配列(水平,[{R1(10MΩ),R2(10MΩ),R3(10MΩ)},ic(TC40H00P)]),
 配列(垂直,[{R1(10MΩ),ic(TC40H00P)},
 {R2(10MΩ),ic(TC40H00P)},
 {R3(10MΩ),ic(TC40H00P)}])]}]
).
```

・問題の与え方(周波数カウンタのメモリー回路)

```
問題 [{ドメイン,概略配置},{サブドメイン,回路},
 {名称,メモリー回路},{機能,記憶},
 {構成要素,[ic(TC40H00P),R1(10MΩ),R2(10Ω),R3(10Ω),R4(20MΩ)]},
 {入力信号線数,2},{出力信号線数,1}{要素数,5}]
```

ゴール[配列]

### (4) CBRの処理の流れ

- ①事例ベースファイルの展開 各スロット毎に事例番号をリスティングする。
- ②入力 問題を与える。
- ③検索 共通スロット名を1つ以上もつものを取り出す。共通スロットが1つも無ければ、ゴー

ルスロットをもつものを取り出す。インデックススロットを指定可能

④評価 ゴールスロットの前提となるスロット(異: 賦与スロット)の無い事例は対象外。事例毎に、

a) 共通スロットの数

b) 共通スロットの値同士の類似度

を計算し、最高得点の事例を選択する。

b)の計算は、スロット値のリスト要素毎に行う。

マッチングボタンには、問題側の要素数の方が多い場合、等しい場合、少ない場合、のそれぞれに対して各要素が、一致する場合、別概念で一致する場合が存在する。

⑤適用 スロット値の置換により行う。問題毎に何が置換対象スロットとなるかを定義しておく必要がある(ゴールスロットが配置であれば構成要素スロットが置換対象スロットである)。置き換えペアは、要素毎のマッチングの度合の「評価」を行い、求める。

⑥予測 ドメインに関する知識を用いて初期配置の状態と同様な状態の失敗状況が記述されている他の事例を検索する。検索事例と同様な失敗を予測してユーザに確認を求める。

⑦修正 検索事例に記述された失敗回避手段を現事例にも適用して再配置結果を求める。

失敗回避手段が適用できない場合はユーザの直接修正によって再配置を行う。

⑧登録 適用が成功すれば事例ベースに登録する。

(5) 領域知識及び領域知識を使ったマッチングのさせ方

①概念階層知識

(a) サブドメインの共通な上位概念

共通な上位概念をもてばマッチングさせる。マッチング対象=概念名である。

[例] 連絡 ——— 接続, 信号交換 . . . (回路)

└— 命令伝達, 情報伝達 . . . (オフィス)

[例] 信号 ——— 弱信号 ——— 低周波信号, 弱電流 . . . (回路)

| └— 思考, プログラミング . . . (オフィス)

└— 強信号 ——— 高周波信号, 強電流 . . . (回路)

└— 騒音 . . . (オフィス)

(b) ドメイン内の上位・下位概念

マッチングさせ方はaと同じ。

[例] 増幅回路 ——— 低周波増幅回路

└— 中間周波増幅回路

└— 高周波増幅回路

(c) インスタンス

マッチング対象はインスタンス。

[例] 概念: ICの仕様

インスタンス: [ICM7216, SN74LS00N, ...]

②対象の属性知識

共通の属性をもてばマッチングさせる。マッチング対象=対象名

[例] 対象: CUP

属性: [dislike(熱), send(データ, register), ...]

(6) 問題点





### ①機能の表現

我々はシステムでの処理を容易にするために機能を下記のように表現する。

ア)機能を主体、動作、オブジェクト(動作対象)、制約で表現する。

VE用語との対応について述べると、主体は機能を行うもの、動作は動詞、オブジェクトは名詞にそれぞれ対応する。制約は省略可能である。CD理論で用いられている方向の概念については現在検討中であり、将来的には組み込む予定である。VEでは基本機能と2次機能があるが、双方とも同様の形式で表現する。

イ)制約の表現は数式とホーン節を含める。

制約の柔軟な表現のために、制約をホーン節で表現する。但し、ホーン節の構成要素(ボディ部)のleaf(末端部)は機能で表現されなければならない。

ウ)定量値の表現を主体、属性(量名)、値の3つ組みで表現する。

ALUの表現例を以下に示す。[ ] は一つの機能を意味し、構成要素は属性名と値の対で表現(‘:’の左右がそれぞれ属性名と値である)する。

[主体:ALU 動作:演算する オブジェクト:数値論理式

制約:接続(ALU、レジスタ):-

[主体:バス 動作:結ぶ オブジェクト:ALU、レジスタ]]

図5.8-3 機能の表現例

### ②概念木

動詞の概念木は知識の修正ができるかどうか、ドメイン固有知識かどうかを調べるのに使用する。名詞の概念木はどのレベルまで抽象化できるかを調べるのに使用する。これらは同一の概念木の島上にあるかどうか判断する。そして概念木上でマッチする最下位の概念での抽象化を行う。

動詞の概念木はVEで用いられる機能用語[日本VE協会 81]に基づく概念木(基本概念木と呼ぶ)と人間の要求している意図の点から機能用語を整理した概念木(目的概念木と呼ぶ)を用意する。人間は使用状況によっても物の機能を考えており、この目的概念木はそのような意図を表現したものである。図5.8-4に目的概念木の例を示す。また、名詞については[田中 87]を基に概念木を作成した。

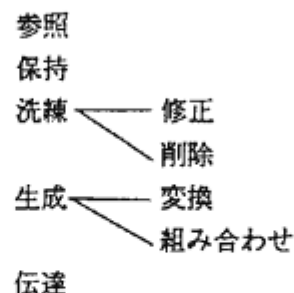


図5.8-4 目的概念木

### ③獲得方法

修正・抽象化は基本的には概念木を用いて対応する知識を置き換えることにより行う。まず問題解決知識内に含まれる主体とその機能の置き換えを動詞の概念木を用いて行う。次に、置き換

えた問題解決知識内の各構成要素が他のドメインにあるかどうかを、事例ベースと機能知識ベース(物の機能)、知識ベース(人間のヒューリスティック)の参照、及び人間への問い合わせにより行う。そしてドメイン固有知識の削除と追加により修正を終わる。抽象化は各ドメインでの一般化、名詞の概念木を用いた共通概念での一般化を行う。各ドメインでの一般化ではEBGを適用してドメイン固有の一般的知識を作成する。そして、それらからドメイン固有知識を削除して、共通知識を作成する。処理フローは下記ようになる。

- [1]動詞概念木と機能知識による修正
- [2]事例、機能知識、ルール、対話による修正
- [3]ドメイン固有知識の削除・追加
- [4]各ドメインでの一般化
- [5]名詞の概念木を用いた一般化
- [6]共通知識の生成
- (3)共通知識獲得支援システム

本システムは異なるドメインに適用可能な知識の獲得支援を対話により行うシステムである。そして第3ドメインでその知識の適用を行う。ドメインは回路とオフィス、問題はレイアウト、共通知識の表現形式としてはホーン節を想定している。本システムの処理する内容は、問題解決知識(ホーン節)を他のドメインに適用可能なように修正すること、二つのドメインのホーン節(各ドメインに特殊化されたもの)を双方のドメインに適用可能な形に一般化すること、一般化したホーン節を想起するためのインデキシングを行うこと、インデキシングしたホーン節が二つのドメインで想起されるかの検証を行うことである。

システム構成は図5.8-5のようになる。

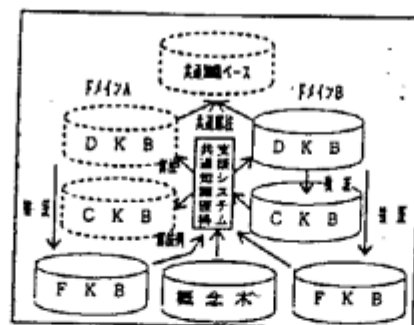


図5.8-5 システム構成

主な処理は修正、一般化、インデキシング、検証、適用である。修正と一般化は修正・抽象化の処理を行い、共通知識を獲得する。インデキシングでは獲得された共通知識、及び共通知識を使用するための知識(ドメイン固有の知識、概念の変換テーブル)を機能の動詞名をキーにインデキシングする。検証ではインデキシングされた共通知識が各ドメインで想起・適用可能かを検証する。そして適用では第三ドメインへの適用を試みる訳である。

#### [参考文献]

- [Hammond 86]Kristian J. Hammond: CHEF:A Model of Case-based Planning,Proc of AAAI'86.
- [ICOT 90]ICOT 平成2年度発電設備診断システムの開発成果報告書(to appear).

- [片井 87]片井修: 概念設計のための機能表現と推論システム, Proc. of the 31th Annual Conference of Systems and Control, JAACE, 1987.
- [Mitchell 86] Mitchell, T. M. et al.: Explanation-Based Generalization; A Unifying View, Machine Learning 1-1, 47-80, 1986.
- [椿 90]椿 和弘、他: 共通知識獲得支援システムの構想, 平成2年度前期全国大会論文集, 情報処理学会(1990).
- [森下 89]森下太朗、他: 事例ベース推論による知識獲得支援、平成元年電気関係学会関西支部連合大会論文集.
- [日本VE協会 81]VE用語研究会報告書: 機能用語の選定とその分類体系－機能定義の効率化研究－、日本VE協会、1981.
- [Schank 86]Schank, R.C.&Riesbeck, C.K.編、石崎俊監訳: 自然言語理解入門-LISPで書いた5つの知的プログラム-、総研出版、1986.
- [田中 87]田中穂積、他: 上位/下位関係シソーラスISAMAP1の作成(I)、(II)、情報処理学会研究会報告 自然言語処理研究会報告No.64、情報処理学会、1987.

## 5. 9 事例に基づく旅行案内システム

(株) 東 芝 末 田 直 道

### 要 旨

我々は C B R の枠組みを明確にし技術ポイントを抽出するために「旅行案内システム」を題材にしてプロトタイプを開発している。

C B R が強力な問題解決器になるためには C B R の修正機構がポイントであると考え、そこからアプローチをした。この修正機構は、語彙と語彙間の二項関係が定義されている概念知識ベースと、解に至った理由を二項関係による意味ネットワークで表現している事例ベースを持つことにより、修正時に、問題を事例の意味ネットワークに当てはめて問題の意味ネットワークをつくり概念知識ベースから、その意味ネットワークの二項関係を満たす値を検索する事を繰り返し、値をより具体的にしていけることを特徴としている。

### 5.9.1 はじめに

事例ベース推論（以下、C B R）は新しい問題解決器の枠組みとして、近年注目されている技術である。C B R の問題解決方法は、人間が発想的に問題解決を行っている方法に近いと言われている。このベースとなる技術は類推である。我々は類推技術を要素技術として柔軟な問題解決器の開発を目指している。そこで、C B R の枠組みを明確にし技術ポイントを抽出するために「旅行案内システム」を題材にしてプロトタイプを開発している。

C B R が強力な問題解決器になるか否かは、事例の修正機構がいかに強力なものになるかに依存しているという観点から、この機構を中心に開発を行ったため、ここでは、事例の修正機構について述べることにする。

### 5.9.2 システム概要

「旅行案内システム」は、あらかじめ決められている項目について、ユーザに条件を入力してもらい、その条件を満たす解を過去の事例とデータベースを使って求めるシステムである。システム全体は、問題の特徴付け、事例の検索、事例の評価、事例の修正、解の適用、知識ベースの修正の各構成要素で構成する。全体の流れを図 1 に示す。各構成要素の説明をする。

- (1) 問題の特徴付け：全項目を事例ベース中の事例の意味ネットワークに使われる頻度の高い順に並べて、条件が入力された項目をそれに従って並べる。
- (2) 事例の検索：各項目について(1)で並べられた順に事例を検索し、適当なところまで

絞り込む。

- (3) 事例の評価：(2)で検索された事例を、最初に知識ベースのisa構造の距離、同じなら適用して成功解の得られた回数の多い順で選択する。
  - (4) 事例の修正：選択された事例の各項目の値、及びそれを説明している意味ネットワークに基づいて、問題の解を求める。
  - (5) 解の適用：(4)で選ばれた解を適用してみて、解を評価する。成功の場合は成功事例として事例ベースに登録する。(検討中)
  - (6) 知識ベースの修正：(5)で適用に失敗した場合は、その理由を解析し、知識ベースを修正する。また、事例にその失敗理由を書き込み事例ベースに登録する。(検討中)
- 入力：各項目の条件、出力はその条件を(ある意味で)満たす解、または、解が存在しない旨のメッセージである。

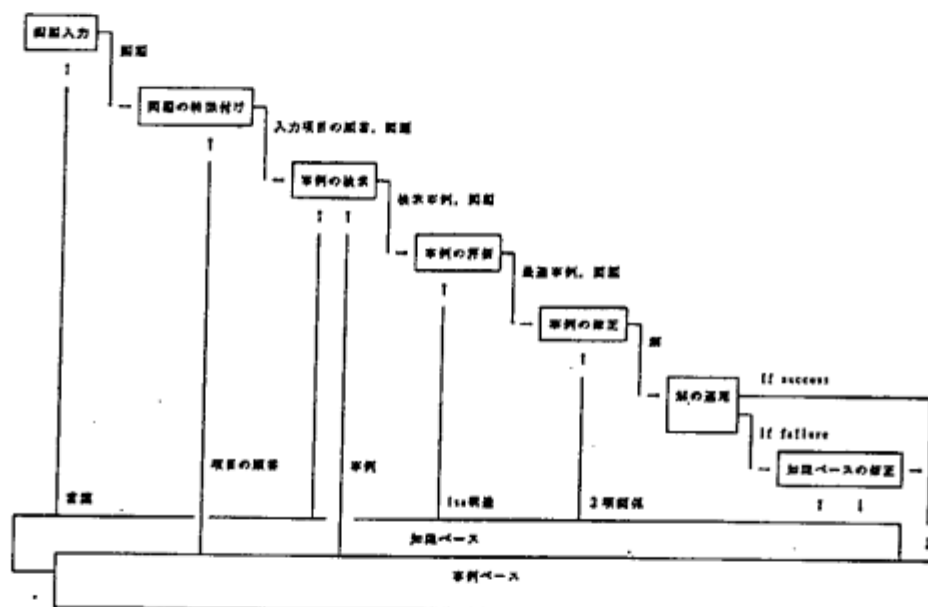


図1 システム構成

### 5.9.3 事例の修正機能

#### 5.9.3.1 基本的な考え方

ここでは、事例修正機能の概要について述べる。C B R の面白味は事例の修正がいか  
柔軟に行われるかにあり、問題の特徴付け、事例の検索、事例の評価等の機能は、事例の  
修正のアルゴリズムに依存した機構であるべきである。

事例は、意味ネットワークで表現する。それにより、人のものの見方や、ものの考え方、及びその構造が表現できる。事例の修正で、問題と選択事例の間で照合しない部分の修正は、意味ネットワークで表現された、ある人のものの見方や考え方に基づいて行う。そのために、その人のものの見方や考え方の知識ベースが必要である。

### 5.9.3.2 データ構造

#### (1) 問題フレーム

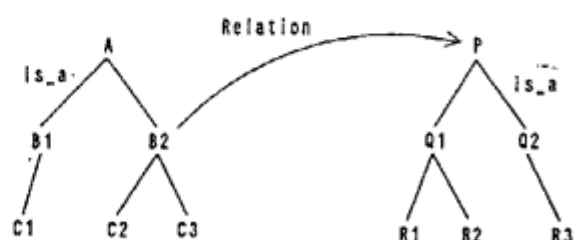
現在、解決をしようとする問題情報を格納する。このフレームの情報はユーザによって概略与えられる。問題解決器は、この情報を基に値がNILまたは抽象度の高いスロットに対して値を埋める。(図2)

#### (2) 概念知識

本システムで使用される、語句が定義されている。語句は問題フレームの各スロットに対応して" is\_a関係"で概念化されている。また、問題フレームの各スロット間の二項関係を表す関係として関係ネットが張られている(図3、図6)。

| slot | value  |
|------|--------|
| 旅行   | 友人との旅行 |
| 目的   | スポーツ   |
| 対象者  | 会社員    |
| 人数   | 3人     |
| 場所   | NIL    |
| 時期   | 夏      |
| 日程   | 短期     |
| 宿泊場所 | NIL    |
| 費用   | NIL    |
| 交通手段 | NIL    |

図2. 問題フレーム



上図のtreeはis\_a構造であり、Relation(B2,P)が定義されているとする。これにより、 $\forall x \in \{B2, C2, C3\}, \forall y \in \{P, Q1, Q2, R1, R2, R3\}$ に対して、Relation(x,y)が導かれる。

図3. 知識ベース内の2項関係の表現

#### (3) 事例ベース

事例ベースには、以前に問題解決された問題フレームが事例フレームとして格納されている。また、各事例フレームには解に至った理由を格納した説明フレームがリンクされている(図4)。

事例1フレーム

| slot | value |
|------|-------|
| 旅行   | 家族旅行  |
| 目的   | スキー   |
| 対象者  | 家族    |
| 人数   | 4人    |
| 場所   | 木島平   |
| 時期   | 1月    |
| 日程   | 2泊3日  |
| 宿泊場所 | 観光ホテル |
| 費用   | 中位の費用 |
| 交通手段 | マイカー  |
| 説明   | 説明1   |

説明1フレーム

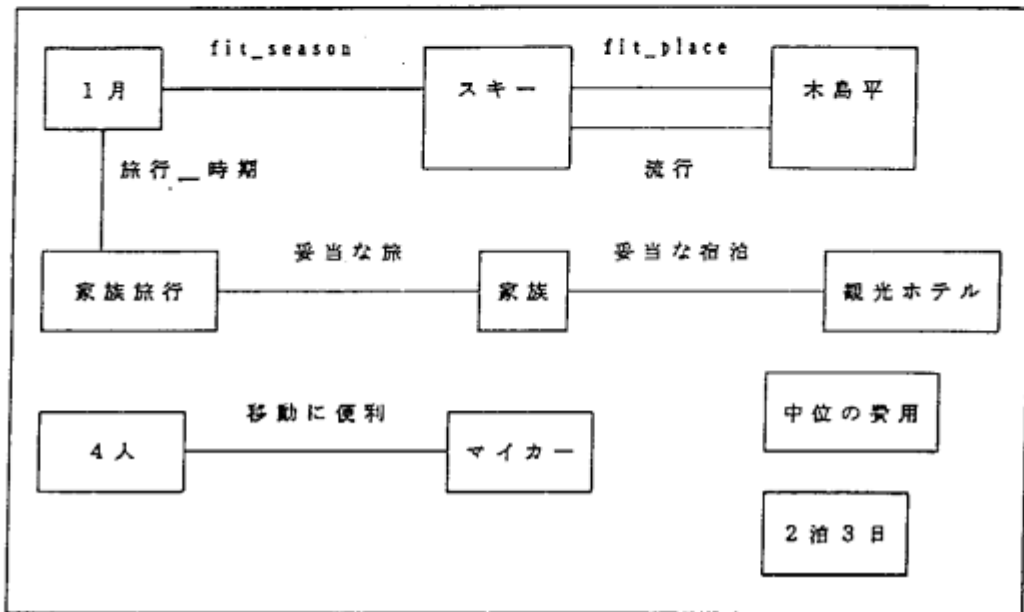
| slot | values                                                            |
|------|-------------------------------------------------------------------|
| 2項関係 | fit_season<br>fit_place<br>妥当な宿泊<br>妥当な旅行<br>移動に便利<br>旅行<br>旅行_時期 |

図4 事例ベース

### 5.9.3.3 アルゴリズム

- (1) 問題を入力する。(図7、図2に示す入力項目)
- (2) 事例を選ぶ。(図4に示す事例が選択されたとする)
- (3) 事例の意味ネットワークの各項目の値を問題の対応する値に置き換える。(図5)

事例の意味ネットワーク



問題の意味ネットワーク

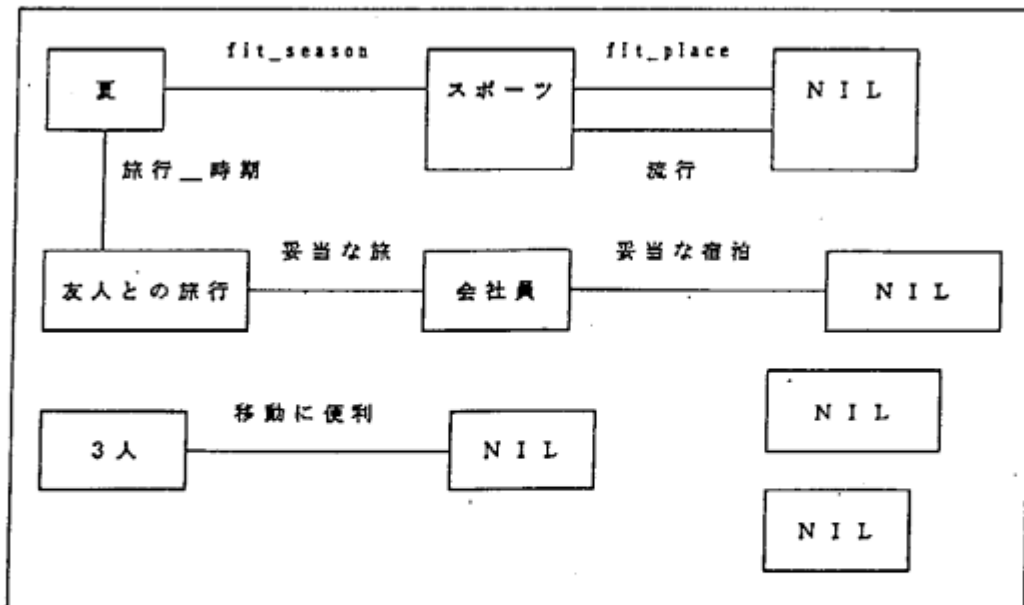


図5 説明の意味ネット

- (4) その意味ネットワークを満たす各項目の値を知識ベースから検索する。
- ・各ノードに対して二項関係で対応する値をセットする。
  - ・同一ノードに対して、異なる二項関係から得られる値のANDをとる。
  - ・そのなかの一つを候補にする。
  - ・二項関係から整合性ある解が得られなかった場合は別の事例を対象とする。
  - ・二項関係が存在しない項目は事例の値をコピーする。

実行例を以下に示す。

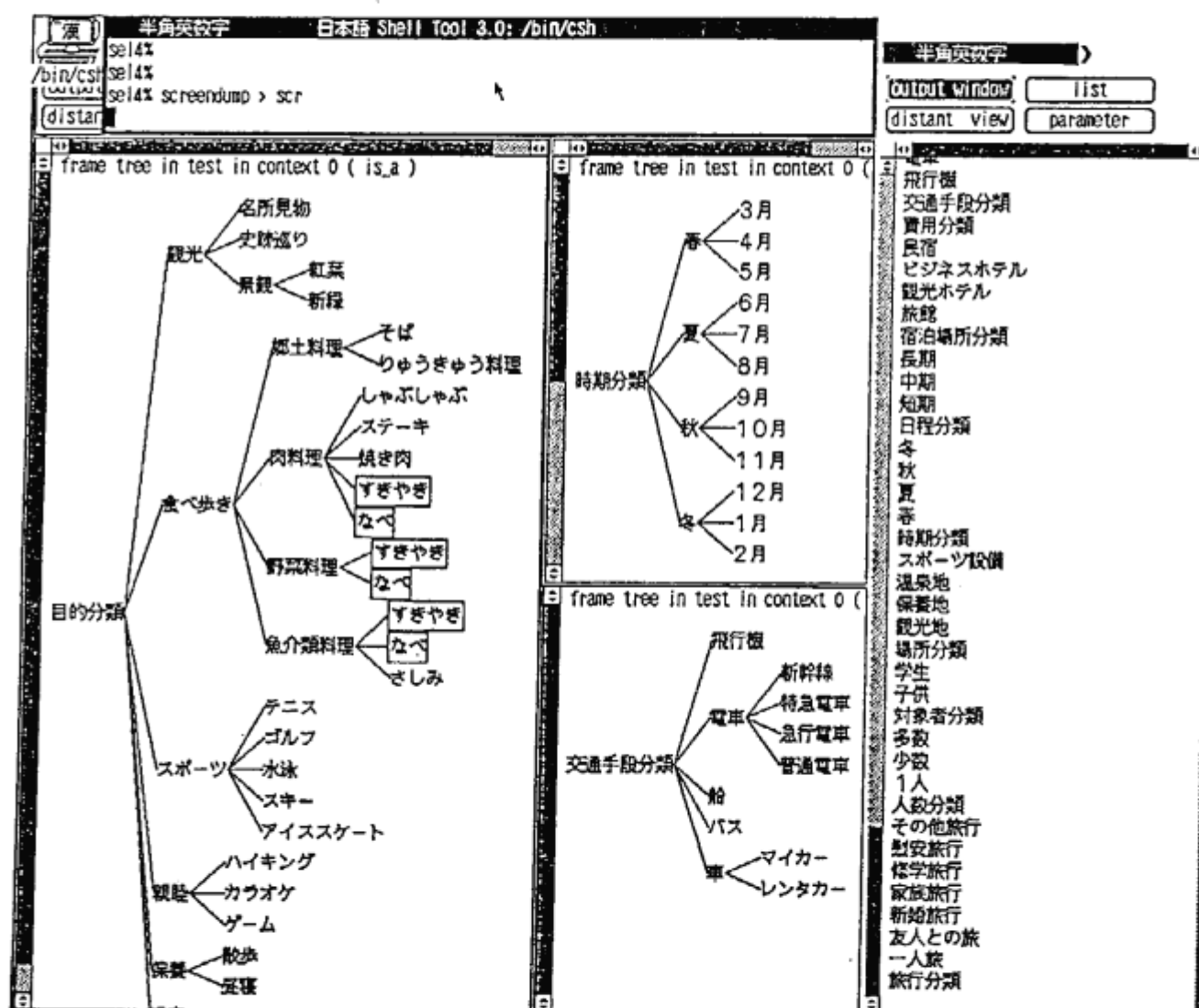


図6 概念知識階層図



distant view

distant view

旅行の条件を、次の項目に従って入力してください。  
表示されている単語で良い場合は、0でつけこず。  
1を入力すると、入力できる単語が出てきます。

一人旅 友人との旅 新婚旅行 家族旅行 修学旅行 慰安旅行 その他旅行  
旅行分類？友人との旅

## 目的分類？スポーツ

人數分類？>3人

### 対象者分類？、会社員

是否分類？>0

OK?

延期? >0

宿泊場所分類？ >0

Selected case number is 13

~~xxx~~ Reasoning starts ~~xxx~~

[illegible]

\*\*\* Solution \*\*\*

旅  
友との旅  
デニス  
3人  
会社員  
軽井沢  
夏  
短期  
観光ホテル  
中位の費用  
マイカー

旅行分類  
目的分類  
人数分類  
対象者分類  
場所分類  
時期分類  
日程分類  
宿泊場所分類  
費用分類  
交通手段分類

#### 5.9.4 おわりに

本稿では修正機構について述べた。現在、問題の特徴付け、事例の検索、事例の評価、事例の修正の機構をインプリメントして評価を行っている。問題点としては解に至った説明を二項関係で表しているが、全ての事柄をこれで表す事には無理がある。そこで何らかの知識表現が必要になってくる。これを、極端に押し進めていくと、事例修正のためのヒュウリスティックスを用意する事にも成りかねなくなり、CBRの利点が失われる。しかし、柔軟な修正を行うためには、このヒュウリスティックスもある程度は必要になってくるものと思われる。

今後、解の適用・評価機構、及び、適用に失敗した場合の学習の仕方について検討を進めて行く予定である。失敗事例からの学習は、その、事例ベースの修正に留まらず、失敗事例の蓄積から、帰納学習により一般化して概念知識の修正にまで機能を拡張したい。

## 5. 10 事例を用いた法的推論システム

ICOT 新田 克己

### 要旨

法律の専門家は判例という過去の事例を用いた推論をしばしば行っている。ここでは事例を用いた法的推論の計算モデルの構築を目的として、解釈の分析と、事例ルールによる事例の表現方法、論理の構築方法などを考察した。また、事例の検索と論理構築を並列処理することによって実行時間の短縮を図る方法を示し、さらに法律システムの構築のための推論エンジンHELIC-IIの構成について説明した。

#### 5. 10. 1 はじめに

判決の予測、裁判における論理展開、法律にしたがった行動計画など、法律の専門家の活動を支援するシステムを開発するには、法律における推論を解析し、その計算モデルを構築する必要がある。

法律のシステムは、条文を論理式で表現し、法的結論をその論理式で証明する定理証明の問題として形式的に取り扱える場合が多い。しかし、現実の裁判などでは、法律で与えられる知識は具体的な問題を解決するには不完全であり、法の解釈がしばしば問題となる。解釈に依存した知識をどのようにとり扱うかが、法律システムを構築する上での長い間の課題であった。

解釈を行うためには、学説や他の法律や判例などが参照される。ここでは判例の表現法と利用法を考察し、法的推論のモデルとその問題点を検討する。また、法的推論の並列化の方法を提案し、法律システム構築のための推論エンジンの構成を述べる [新田 90]。

#### 5. 10. 2 事例を用いた法的推論モデル

##### (1) 事例の表現

事例の中で最も重要なものは判決文である。判決文は過去の事件において、観測事実から結論に至るまでの推論過程を表すものである。その主な内容は、どのようなルールがどのような事実のもとで適用されたかという説明である。

これを図 5.10.1 のような図式で考えることができる [Branting 89] [Branting 88]。この図で  $R_1$ ,  $R_2$  は法律から抽出されたプロダクションルールを表し,  $r_1$ ,  $r_2$ ,  $r_3$  は判断結果から抽出された事例ルールを表し,  $FA$ ,  $FB$ ,  $FC$  は観測された事実を意味ネットで表現したものを表す。

この例では,  $FA$  は  $A$  に,  $FB$  は  $B$  に該当したなどとそれぞれ判断されて, 最終的に  $Goal$  という結論に至ったことを表す。ここで,  $r_3$  は状況  $FB$  のもとでルール

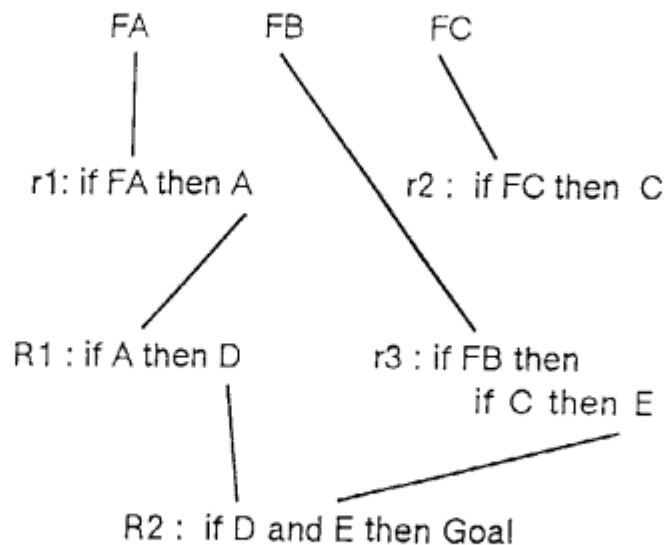


図 5.10.1 判決文の説明

(IF C THEN E) が生成されたことを表す。これは、互いに類似する 2 つの概念 C と E を、ある状況では同一視したいときなどに用いることができる。

このように、ルールには条文から抽出されるルールの他に、判例として与えられるルール、一般常識を表すルール、新たに裁判官によって創造されたルールなどを含む。一般的なルールと判断のルール（事例ルール）との違いは、前者は条件部がすべて満足されないと起動されないのに対し、後者は条件部がかなり満足されれば起動される可能性があること、前者の条件部は固定されているのに対して、後者の条件部の概念が拡張（一般化）されうることである。また、前者の推論結果は常に正しいが、後者の推論結果は正しいとは限らない [Branting 89]。

意味ネットは、

{ 対象, 関係, 値 }

の 3 つ組の集合で表現する。事例ルールの中では

IF { {対象 1, 関係 1, 値 1}, 重要度 1 } ∧  
{ {対象 2, 関係 2, 値 2}, 重要度 2 } ∧

...

{ {対象 n, 関係 n, 値 n}, 重要度 n }

THEN

述語 / リンク / ノード / ルールの生成

のように、意味ネットの表現がそのリンクの重要度とともに記述される。

事例ルールによって生成されたデータにはその根拠（使用した事実とルールの組）が付加される。事例ルールによる結論はあくまでも仮説に過ぎないから、後でその根拠を示す必要があるからである。

## （２）事例の適合

新しい事件が与えられたときの、過去の事例の利用方法について述べる。新しい事件で成り立つ事実は、意味ネットとその事件を特徴付ける述語の集合で表されているとする。

このような入力データに対し、法律のルール（そのほとんどがプロダクションルールとして記述されている）を直接適用しようとしても、通常はルールの条件部が満足されない。前に述べたように事実と法的概念の抽象度が違うからである。そこでまず事例ルールを起動すると、観測事実の中からいくつかが目ざされて抽象的な概念が生成され、その結果、法律のルールが起動されることになる。

事例ルールの場合、条件がすべて満足されなくともある程度の類似性があれば起動される [Branting 89]。例えば、

```
IF { {船 #1, 船長, 山田}, 20 }
THEN ~
```

というルールがあったとき、

```
{ 飛行機 #1, 機長, 田中 }
```

というデータがあったとする。2つの3つ組の対象、関係、値のそれぞれが、与えられた階層構造のもとで同一の上位概念を持つ場合に、その3つ組は対応がとれるものとする。この例では、[船 #1] と [飛行機 #1] はともに [乗物] という概念の下位概念（またはインスタンス）であるという知識と [船長] と [機長] はともに [責任者] という概念の下位概念であるという知識と [山田] と [田中] はともに [自然人] の下位概念（またはインスタンス）であるという知識があれば、{船 #1, 船長, 山田} は {飛行機 #1, 機長, 田中} と対応がとれる。対応をとるにあたって、上位概念をたどる段数が少ないほど、両者の類似度は近いと判断される。

判決予測を行うときには、入力された事件の意味ネットが、事例ルールの適用によって次第に詳細化と抽象化が行われ、あらかじめ用意された特定の事件のパターンに合致するとき最終結論に至る。

## （３）判決の例

判決の例として、労働基準法 79 条

[労働者が業務上死亡した場合においては、使用者は、遺族に対して、平均賃金の1000日分の遺族補償を行わなければならない]

に関する判決を考えてみる。〔業務上の死亡か否か〕の判断基準は、業務遂行中の死亡であること、業務と死亡との間に因果関係があること、の2つであることが判例で確立している。

ここで因果関係の判定は容易ではない。他の原因が重なっていることもあるし、業務が間接的な原因になっているかもしれないからである。従って、多くの事例が類型化され、それぞれに判断基準がルールとして決められている。

以下の事例は、農協職員山田氏が勤務時間後に、料理屋での会議に業務命令で参加し、その後、帰宅の途中で組合員鈴木氏の家に保険加入の勧誘の目的で行こうとして事故にあった事件である〔長野地判39.10.6の改変〕。

裁判では、業務上の死亡かどうかが争われた。その論点の1つとして、被告（労働基準監督署）は、(1)山田氏は庶務係なので保険勧誘は本来の業務（職務）ではない、(2)鈴木氏の家へ勧誘に行けとの具体的命令は出していない、の理由で業務でないとしたのに対し、原告（山田氏の遺族）は、(1)保険勧誘の成績は評価の対象となっていたので実質的業務である、(2)保険勧誘を奨励する局長の訓示やキャンペーンがあったので包括的業務命令があったとみなせる、と主張した。

まず、この事件の観測された事実の意味ネット表現の一部を図5.10.2に示す。

被告が、保険勧誘は山田氏の直接の業務（職務）でないことを理由に、業務でないとしたことは、以下の事例ルールで表現できる（ルールの要部のみ示す）。

```
IF { {雇用関係 #1, 従業者, 山田氏}, 20} ∧
 { {雇用関係 #1, 雇用者, 郵便局}, 20} ∧
 {not {雇用関係 #1, 職務, 保険勧誘}, 100}
THEN [make(職務外)]
```

それに対し、原告が、保険の勧誘が勤務評価の対象であったので、実質的に業務であったとしたことは以下のように表すことができる。

```
IF { {雇用関係 #1, 従業者, 山田氏}, 100} ∧
 { {雇用関係 #1, 雇用主, 郵便局 #1}, 100} ∧
 { {郵便局 #1, 業務, 保険勧誘}, 100} ∧
 { {郵便局 #1, 責任者, 局長 #1}, 100} ∧
 { {局長 #1, 職務, 勤務評価}, 100} ∧
 { {勤務評価 #1, 対象, 山田氏}, 100} ∧
 { {勤務評価 #1, 内容, 保険勧誘}, 100}
THEN [make(職務範囲)]
```

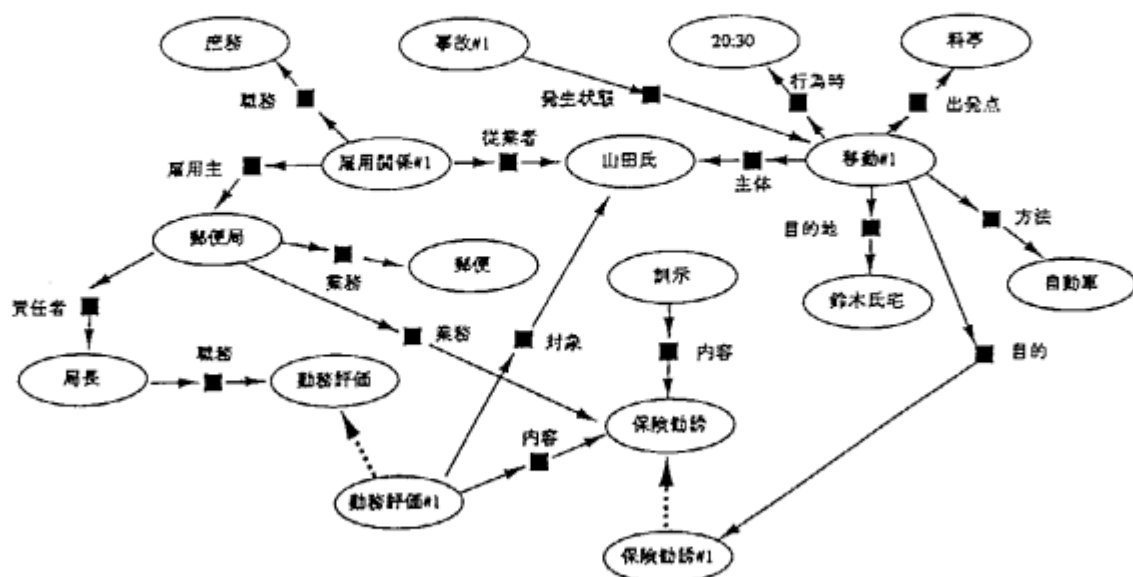


図 5.10.2 山田事件の意味ネット表現

ここで新しい事件について考える。これは、建築会社の材木運搬トラックの運転手である田中氏が材木の積み降ろし中に事故にあったものである。田中氏の材木の積み降ろしは雇用契約に含まれていないとすると、労災認定においてどのような判断がなされるかを予測する。

この事件を意味ネットで記述したものの一部を図 5.10.3 に示す。

このネットを図 5.10.2 と比較し、[雇用関係 #1] と [雇用関係 #2]，[山田氏] と [田中氏]，[郵便局] と [建築会社]，．．のようにマッピングすれば対応がとれる。すなわち、山田事件によってなされた双方の主張（職務でない vs 成績評価の対象）の事例ルールを田中事件に適用すれば、田中事件でも同じ議論が成立することが推定される。

#### (4) 推論結果

同じ観測事実を与えても、どの事例ルールを適用するかによって異なる結果が得られる。事例ルールの適用による新たなデータの生成は、1つの仮説の生成とも考えられる。異なる事例ルールの適用によって異なる仮説が生成されるが、作業記憶の中でそれらは同一に取り扱われるのではなく、異なった世界で成り立つ事実として管理されなくてはならない。仮説の組み合わせによっては矛盾を生じるから、そのような仮説の組み合わせを防止するある種の真理維持機能が必要である。

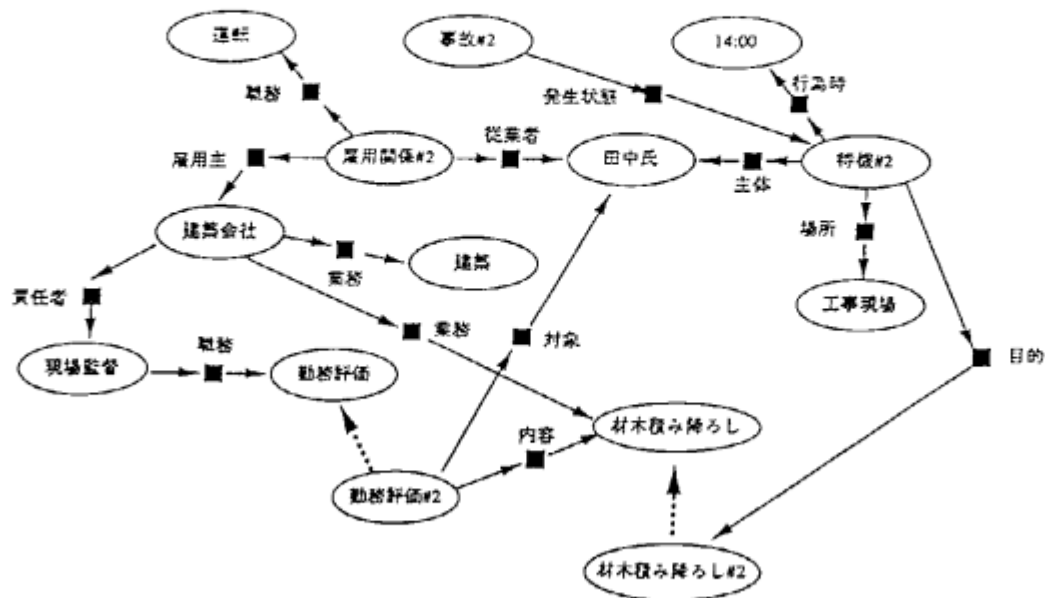


図 5.10.3 田中事件の意味ネット表現

### 5. 10. 3 法的推論システム

#### (1) データフローグラフ

一般に事例ベース推論は、膨大な事例を検索し、照合しなくてはならないから、実行時間が非常にかかることが予想される。そこで、ここでは並列処理による推論時間の短縮の可能性について検討する。

前に述べたように、法律における説明は一般のプロダクションルールと事例ルールで表される。プロダクションルールの条件部は R E T E ネットなどのデータフローグラフに展開することで高速化が図れることが知られている。事例ルールの場合も R E T E と類似のデータフローグラフに展開することができる。このデータフローグラフが通常の R E T E ネットと異なるのは以下の点である。

- 3つ組の各引き数が同じ上位概念ノードを持つかどうかのチェックが必要である。
- R E T E ネットの場合には全ての条件が満足されたトークンの組合せが出力されるのに対し、このグラフでは一部の条件が満たされないトークンの組み合わせでも、ある基準以上の類似度があれば出力される。



このデータフローグラフを並列論理型言語KL1で実現する。これは、事例規則の条件部をコンパイルして、KL1プロセスをノードに対応させ、ストリームをリンクに対応させたデータフローグラフを作成することによる。ルートノードのプロセスから、新しい事件の事実データをトークンとして流すと、条件照合の並列処理が実現できる。

## (2) 法的推論エンジン

これまでに述べてきたように、法的推論を行うためには事例を利用して、意思決定や論理構築を支援する機能が必要である。場合によっては、ルールを生成することもある。生成された論理が苦しまぎれのへ理屈であっても、実用に当たっては無いよりましであるし、相手の反撃を未然に防止するのに役に立つこともある。また、考えの漏れを防ぐのにも使えることがある。

現在設計中の、へ理屈生成を行うための法的推論エンジンHELIC-II (Hypothetical Explanation of Legal Inference Constructor - Intelligent Instructor) はこのような目的のためのエンジンである。HELIC-IIは、解決しようとする事件の状況(と目標とする法的な状態)を入力すると、状況から目標へ至る論理(説明)を生成する(図5.10.4)。

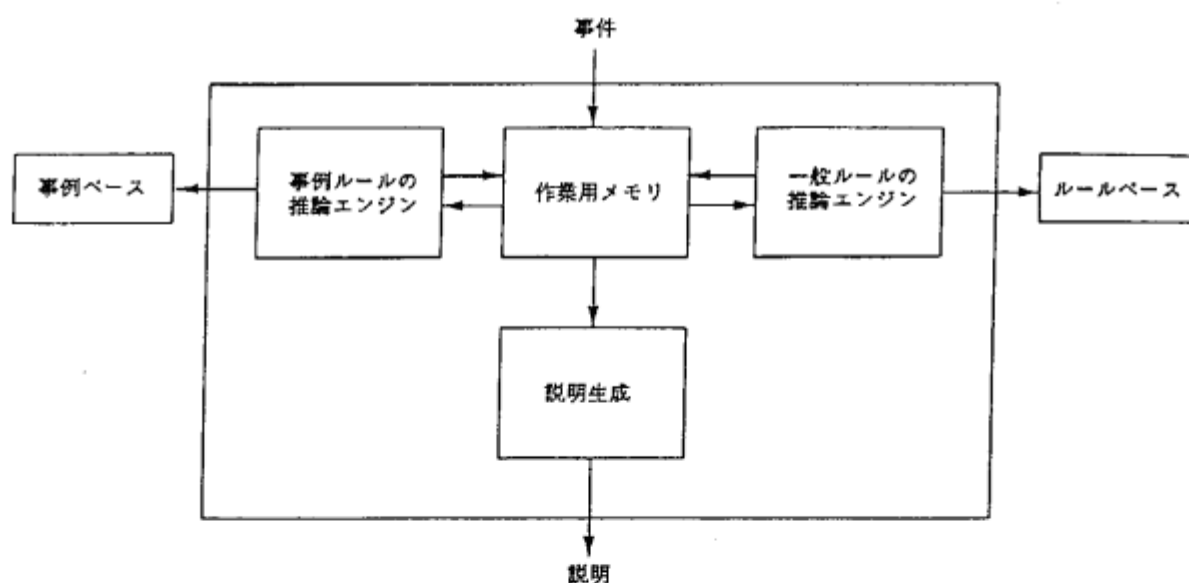


図5.10.4 HELIC-IIの構成

このシステムは、プロダクションルールの推論を行うエンジンと事例規則の推論を行うエンジンとからなる[Rissland 89]。プロダクションルールのエンジンからはルールベー

スがアクセスでき、事例規則のエンジンからは事例ベースがアクセスできる。解決しようとする事件を入力すると、これは意味ネットと述語の集合として作業記憶に格納される。双方のエンジンは独立に作業記憶にアクセスし、推論結果をその根拠と共に作業記憶に格納する。最終的に適用できるルールや事例がなくなった段階で、目標の状態に達したものをすべて選び、用いたルールと事例を出力する。

#### 5. 10. 4 おわりに

法的推論における判決文の役割と、事例検索の並列化の可能性について述べた。技術的課題を以下に示す。

1. 事例の記述が何通りもできるのは問題である。異なった記述をしても抽象化で同じパターンにまとめられる保証はない。事件の記述のための制限と主要概念の辞書が必要である。
2. 事例規則の数が多いときに組み合わせが膨大になる恐れがある。何らかの推論制御が必要である。
3. 立証責任の問題が重みに反映されなくてはならない。原告と被告は対等ではなく、条文によってどちらかがその法律概念に該当すること / 該当しないことを立証する責任がある。当然、立証する方の負担が大きく、反証する方は相手の説明の欠陥だけをつけば良い。

#### 参考文献

- [新田 90] 新田, 星田: 事例を用いた法的推論とその並列化, 情報処理学会研究会資料 (1990).
- [Ashley 88] K.D.Ashley, et.al.: "Compare and Contrast, A Test of Expertise", Workshop on Case-Based Reasoning (1988)
- [Branting 89] L.K.Branting: "Representing and Reusing Explanations of Legal Precedents", International Conference on Artificial Intelligence and Law (1989).
- [Branting 88] L.K.Branting: "The Role of Explanation in Reasoning from Legal precedents", Workshop on Case-Based Reasoning (1988).
- [Rissland 89] E.L.Rissland, et.al.: "Interpreting Statutory Predicates", International Conference on Artificial Intelligence and Law (1989).

## 5. 1 1 事例に基づく相互主導的対話システム

徳島大学工学部 小野典彦

### 要 旨

本報告では、質問に回答するだけでなく、システムが主導的に対話を進行させる対話システムの構築を目的として、対話システムのエピソード的な事例記憶の構成ならびに処理について考察する。ここでは、対話プロセスをエピソード的な事例記憶に基づくものとしてとらえると共に、この記憶上には注視点が設定されており、積極的な応答や対話の発散抑制はこれを介して行なわれるものとする。また、注視点は相手の発話に起動されるボトムアップなプロセスと、推測される相手のゴールおよび自己のゴールに基づくトップダウンなプロセスとの組み合わせにより制御される。

#### 5. 1 1. 1 はじめに

会話を円滑に進めることは、われわれ人間にとっても必ずしも容易なことではない。話をするのが苦手な人ならなおさらのことである。しかし次々とほとぼしるように言葉が口から出ていき、いきいきと会話を進行させることができる人が存在することもまた確かである。ここでは、会話上手な人のように、ユーザに快適な印象を与えつつ対話できるコンピュータシステムの実現に僅かでも近付くことを目指して、エピソード記憶に基づく対話システムの構築について考察を行なう。

さて、会話を上手に進行させることができる人は（社交的あるいは話し好きなどといった性格もさることながら）よく“話題が豊富”な人であるという言い方をされる。この“話題が豊富”ということは単に記憶しているエピソードの蓄積量が多いことのみではなく、蓄積されたエピソードから会話の流れに沿ったものが‘思い出され’、その場にふさわしいものを上手に‘選択できる’ことであると考えられる。また、会話を上手に進める人は一方的に話すだけではなく、他の話者の発話が促されるように、他の話者がエピソードを思い出しやすくするような手がかりを話の中に与えているとみることができる。このように、会話は話者の記憶の想起プロセスに基づいており、エピソード記憶の想起能力が会話の進行を司っていると考えることができる。

本報告では、一般的な雑談のように、対局的な目標が明確でないため発散しがちな対話ではなく、話者の一方が他方に自分の言い分を納得させるというような説得的な対話について特に考察を行なう。たとえば、車のセールスのような状況においてはセールスマンは、顧客に、売ろうと考えている車のよさを説明し、その車を買う意義を納得させ、購入してもらうという明確な目標をもって会話を展開する。

エピソード記憶およびその想起プロセスに焦点を当てた記憶研究としては Kolodner [Kolodner 84] の E-MOP, Schank [Schank 82] および Hammond [Hammond 89] の MOP および TOP などがあるが、ユーザとの対話を柔軟に進行させるためには、これらだけで十分な機能を備えているとはいえない。エピソード記憶に基づくシステムは、知識の完全性が保証されない開かれた世界を対象としているにもかかわらず、これらはシステム固有の知識のみで説明されない入力について、それを新たにエピソード記憶に組織化するための方法を提供していないからである。たとえば、車のセールスの商談中に顧客から、いままで予想もできなかった新しい車の活用法を聞かされるかもしれない。しかし、その話をうまく記憶しておくことができれば、せっかく聞いた話を他の顧客のために役立てることができる。本研究は、このような入力に対処するために、イベント間の近隣性と称する概念に基づく知識構造として VOP (Vicinity-Based Organization Packet) を導入し、エピソード記憶の記銘および想起をより柔軟なものとすることを目指す。

## 5. 11. 2 エピソード記憶構造

### 5. 11. 2. 1 プランニングのためのエピソード記憶構造

周知のように、Schank [Schank 80, 82] は、ゴール・プラン [Schank 77] に基づく物語理解を目指して、MOP および場面 (scene) などの知識構造を提案している。一方、Kolodner [Kolodner 84] は、次々に入力されてくるイベントの集合を一般化・特殊化の階層に沿って組織化するための知識構造として E-MOP を提案している。

対話プロセスの基礎となる知識構造としては、(i) E-MOP と同等のイベント組織化能力をもち、さらに (ii) ゴールと行動系列の関係を表現できなければならないことから、ここではその基本的枠組として MOP を採用するものとして、MOP を構成する場面は E-MOP に一般行動系列の記述を加えたもので表現するものとする (図 5.11.1 にその構成の直感的な枠組を示す)。

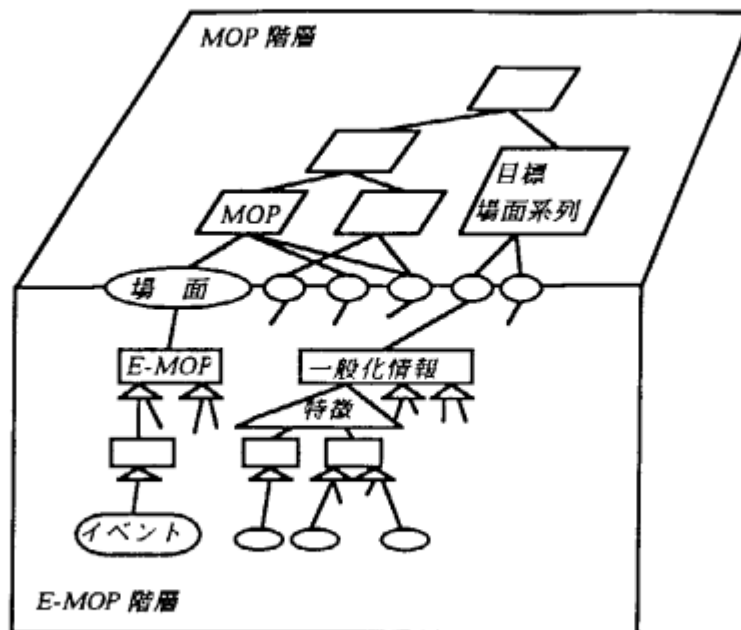


図 5.11.1 対話プランニングシステムのエピソード記憶構造

#### 5. 1 1. 2. 2 近隣性

前節で述べたエピソード記憶は MOP、場面および E-MOP を組み合わせた知識構造の集合体となっており、既に出会ったことのあるエピソード（イベント）であれば、E-MOP 階層を通して、それがいかなる場面对応するものなのかを推測し、さらに MOP 階層を通して、その場面对応する目標を達成するためのものなのかを推測することができる（目標は MOP に記述されている）。逆にある目標を達成するためのプランを選択するためには、この逆のプロセスをたどればよい。

しかし、相手が、自分には思いもよらない（しかも効果的な）プランを採用してきた場合（あるいは自分が偶然行なった行動系列によって目標がたまたま達成されてしまった場合）、対話プランナは、これを新たなプランとして記憶構造の中に取り入れられなくてはならない。このプランをそれ以降のプランニングで採用できないといった不都合が生ずるからである。

このときの問題点は、プランとゴールとの間の因果関係が必ずしも特定できるとは限らないということである。すなわち、相手が採用したプランにせよ、自分自身が偶然採用したプランにせよ、そのプランと達成された目標との間に何らかの因果関係があることは認識できても、その時点でもっている知識だけでは、その関係を説明できないことが多いか

らである。そこで、このようなプランとゴールの間の因果関係に限らず、何らかの関係を  
もつことがわかっている概念どうしを結び付けて記憶しておくために、“近隣性”という  
関係を導入する。

たとえば、つぎのような対話において、最初、新米セールスマン S は‘通勤’と‘燃  
費’の関係を知らなかったとしよう。

S「この新型車は、安定性がよく振動も少なくなっておりまして、長距離ドライブ  
でもいままでのように疲れることはありません」

B「長距離は乗らないからよいのですが、通勤にばかり利用するので燃費が悪くて  
困ります」

顧客 B さんの話からでは、なぜ‘通勤にばかり利用する’と‘燃費が悪く’なるのかにつ  
いての説明はなく、S にはその間の関係が何であるかはわからないが、‘通勤にばかり利  
用する’ことと‘燃費が悪くなる’こととの間に何らかの関係が存在することはわかる。  
このように、その関係の説明は与えられないが、複数の概念間に何らかの関係の存在をし  
ることができるという場合に、これらの間に近隣性が存在するという。この例では‘通勤  
にばかり利用する’ことと‘燃費が悪くなる’ということの間に近隣性が存在することに  
なる。

このように、近隣性を用いて、関連するプランとゴールを結び付けておくことに  
よって、そのプランがそのゴールの達成に貢献し得ることを説明することができなくて  
も、そのプランを採用することが可能となる。もちろん、後にその説明が判明した（主と  
して対話を通して）段階で、この近隣性は得られた説明で置き換えられる。

#### 5. 1 1. 2. 3 VOP

MOP 化できない関係を表現するために前節で近隣性を導入した。この近隣性に基づい  
て、新たな情報をエピソード記憶に記録するための記憶の単位として VOP (vicinity-based  
memory organization packet) を定義する。すなわち、VOP とは、(i) 何らかの近隣性およ  
び (ii) それによって結び付けられた項目（イベント、プラン、ゴール、オブジェクトな  
ど）の対である。VOP はそれが結び付ける項目間のいわゆる predictability [Kolodner 84] を  
表現していることになる。

上記の‘通勤’と‘燃費’の例では‘通勤に車を利用する’という E-MOP と‘車の燃  
費が悪い’という状態の間に‘B さんがそう述べた’という近隣性をもって VOP が生成  
される（図 5.11.2）。

このように VOP は、MOP（あるいはその他の記憶要素）を結合することによって組織化する。MOP を組織化する記憶構造としては TOP [Schank 82][Hammond 89] があるが、以下の点で VOP は TOP と異なる。

- TOP は MOP が領域依存であるのに対して、領域独立な知識であり、格言にたとえられるような一般的な状況を表す。
- VOP は必ずしも TOP で表現されるような一般的な状況を共有しない場合であっても、イベント間の時間的接近性、登場人物の共通性などの近隣性によっても記憶が組織される。

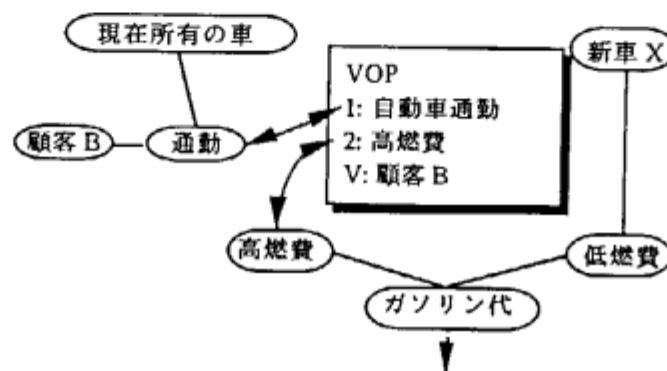


図 5.11.2 もっとも単純な VOP の例

### 5. 1 1. 3 エピソード記憶上での注視点制御

ここでは、記憶に基づく対話プロセスは、エピソード記憶組織そのものの上に表現される注視点を介して制御されるものとする。ここで注視点とは、自己の主ゴール、そのサブゴール、プランおよび関係するその他のゴール群の中で、特に現時点で処理の対象となっているものであって、ここではこれらのゴールやプランの各々に直接注視点を設定するものとする。さらに対話相手のモデルとして、推測された相手の注視点を同時に設定する。さらに話題の焦点を表す注視点を設定する。これらを各々 (i) 自己ゴール注視点、(ii) 自己プラン注視点、(iii) 相手ゴール注視点、(iv) 相手プラン注視点および (v) 話題注視点と呼ぶことにする。

#### 5. 1 1. 3. 1 注視点制御の基本サイクル

単独の対話者の対話プロセスを単純化して考えると、

- (1) 相手の発話中から手がかりを得る。得られない場合は、自己のゴールに基づいて手がかりを得る（被活性化）。

- (2) 手がかりから想起される情報の集合を求め（活性拡大），
- (3) 話題として適切なものを選択し（選択），
- (4) 発話する。

という基本サイクルが考えられる。

相手の発話から手がかりを得る被活性化ならびに活性拡大は，ボトムアップな状況駆動注視点制御であり，自己のゴールに基づく被活性化および選択のプロセスはトップダウンなゴール駆動注視点制御である。

#### 5. 1 1. 3. 2 状況駆動注視点制御

対話相手の発話に含まれる相手のプランの一部から，それを含むいくつかのプランが活性化される。それらのプランから活性の拡大によって相手のゴールの候補群が活性化される。その中で，対話に含まれる情報による裏付けの最も多いゴールを相手のゴールとして認識し，相手ゴール注視点を設定する。相手ゴール注視点が設定されると，そのゴールと自己のゴール群との間の関係 [Wiensky 81, 83] を認識し，自己ゴールを活性化する。自己ゴールが活性化されると，つぎのゴール駆動注視点制御が起動される。

#### 5. 1 1. 3. 3 ゴール駆動注視点制御

システムは主ゴールを達成するために対話を行なうのであるが，システムが同時には唯一のゴールしかもたないという状況は通常ありえない。前節の状況駆動注視点制御プロセスにおいても相手ゴールと自己ゴールの関係から複数の自己ゴールが活性化される。車のセールスマンが顧客に車を得ろうとするときも，売りさえすればよいというわけではない。顧客との対人関係を良好なものとしておくなどのゴール（いわゆる *preservation goal* [Schank 77][Wilensky 83]）ももっており，脅迫して売りつけるというようなプランを排除する制約としても働き，脅迫の方法に注視することを避ける。また，自分は高価な車を売りたいが，顧客が安価な車を欲していることがわかっているような場合，高価な車よりも安価な車を売ること注視点を移行するほうがよいかもしれない。このようにゴール間の関係を調整することによっても注視点が変化する。

ゴール駆動注視点制御は，活性化された自己ゴール群および現在注視点が設定されている自己ゴールの中から，相手ゴール注視点との関係が最も緊密なものに自己ゴール注視点を設定し，注視点が設定されたゴールを達成するプランを想起し，実行する。そして，そのプランの実行中の部分に自己プラン注視点を設定する。

一度注視点を設定されたゴールは，その時点の主ゴールが達成されるまでは注視点から外れても，痕跡を残しており，再度参照された時にはその痕跡を発見する。ただし，注視点の移行の履歴をすべて記録しておいて，その履歴を遡りながら特定のゴールが注視点で



あったか否かを調べるのは効率的でないことから、ゴールに注視点として参照されたか否かだけを記録しておく。ちょうど、迷路の中で一度通った道を堂々巡りすることを避けるために道にコインを落しながら歩き、通過したことのある道に来るとコインを発見してそれとわかるように指定する方法に似ていることから、この方法を Coin Dropping と呼ぶ。これは注視点の移行によって達成されないまま保留されているゴール (suspended goal [Hammond 88]) の認識を可能とする。ゴールの保留は主としてプランのブロック (プランの実行が不可能となる条件の検出) によって生ずるが、保留されていたゴールを発見したときにこの条件が解消されていれば、プランの実行が再開されて保留されていたゴールの達成が可能であるので、このゴールを注視点とすることによって保留されていたゴールが再び活性化される。

#### 5. 11. 4 注視点に基づく記録と再生

##### 5. 11. 4. 1 記録

相手が自分にとってまったく新規なプランを採用してきた場合、それがどのゴールを目指したものであるのかは一般に特定できない。すなわち、どのような近隣性を用いて VOP を生成すべきか特定できない。これを避けるため、ここでは対話は必ず協調的に行なわれるものと仮定する。すなわち、相手はお互いの注視点に関連した目標しか達成しようとしがないものとする。これにより、VOP は現在の注視点との近隣性だけを用いて、記憶組織中に生成される。このような仮定を施さないと、一般に VOP 生成の根拠となる近隣性はその存在について詳細な説明を必要としないので、あらゆる記憶どうしを VOP によって結び付けてしまうといった事態が起こり得る。このように、むやみに VOP を生成してしまうと、VOP を通した活性拡大によって活性化されるゴールおよびプランが必要以上に多くなり、選択時に不都合を生ずる。

##### 5. 11. 4. 2 再生

第3節で述べた注視点制御サイクルによって、現在の注視点からアクセス可能な VOP を通し、関連するエピソードを想起する。エピソード記憶の再生プロセスについては [Kolodner 84] に詳細に論じられている。ここでは、[Kolodner 84] で述べられている再生と VOP による再生の相違点について簡単に触れておく。

- ・ [Kol 84] におけるイベントの想起は精巧化 (elaboration) によって再構成的に記憶を検索することによって行なわれる。再生のための知識 (ルール集合) は予め固定されている。

- ・ VOP による再生は、注視点からみた特徴の近隣性に基づく想起である。VOP 生成時に近隣性として採用される項目は、システムが出会うイベントの系列に依存するため、再生時に使用される情報はシステムへの入力履歴に左右される。

#### 5. 1 1. 4. 3 事例に基づくプランニングとしての対話

ここでは、発話を事例に基づくプランニング (case-based planning [Hammond 89] - CBP) のプロセスと考えている。本報告における対話プランニングと代表的な CBP システムである CHEF [Hammond 89] との相違点は以下の通りである。

- ・ CHEF は客の要求をそのまま受け入れて、条件を満足する料理を選択する。最初の選択が事例には依存しないプランニングによって施された後に、主として予測可能な失敗を回避するために過去の事例が利用されている。
- ・ 対話プランニングシステムの場合には、システム自身が目標をもち、ユーザに積極的に働きかける。また成功事例からも積極的に事例に基づくプランニングを行なう。

#### 5. 1 1. 5 おわりに

本報告では、対話プロセスをエピソード的な事例記憶に基づくプランニングのプロセスとしてとらえ、エピソード記憶上での注視点制御を介して対話を進行させる対話システムの記憶の構成ならびに処理について考察を行なった。また、この記憶組織上での注視点制御および注視点に基づくエピソードの記録および再生の概要について述べた。

#### 参考文献

- [Hammond 88] Hammond, K.J.: *Opportunistic Memory: Storing and recalling suspended goals*, Proc. of DARPA Workshop on Case-Based Reasoning, pp.154-168 (1988).
- [Hammond 89] Hammond, K.J.: *Case-Based Planning*, Academic Press Inc. (1989).
- [Kolodner 84] Kolodner, J.L.: *Retrieval and Organizational Strategies in Conceptual Memory*, Lawrence Erlbaum Associates (1984).
- [Schank 77] Schank, R.C. and Abelson, R.P.: *Scripts, Plans, Goals, and Understanding*, Lawrence Erlbaum Associates (1977).

- [Schank 80] Schank, R.C.: *Language and Memory*, Cognitive Science, Vol.4, pp.243-284 (1980).
- [Schank 82] Schank, R.C.: *Dynamic Memory*, Cambridge University Press (1982).
- [Wilensky 81] Wilensky, R.: *Meta-Planning: Representing and Using Knowledge about Planning in Problem Solving and Natural Language Understanding*, Cognitive Science, Vol.5, pp.197-233 (1981).
- [Wilensky 83] Wilensky, R.: *Planning and Understanding*, Addison-Wesley (1983).

## 6. 結 論

東京工業大学 小林重信

### 6. 1 活動成果の要約

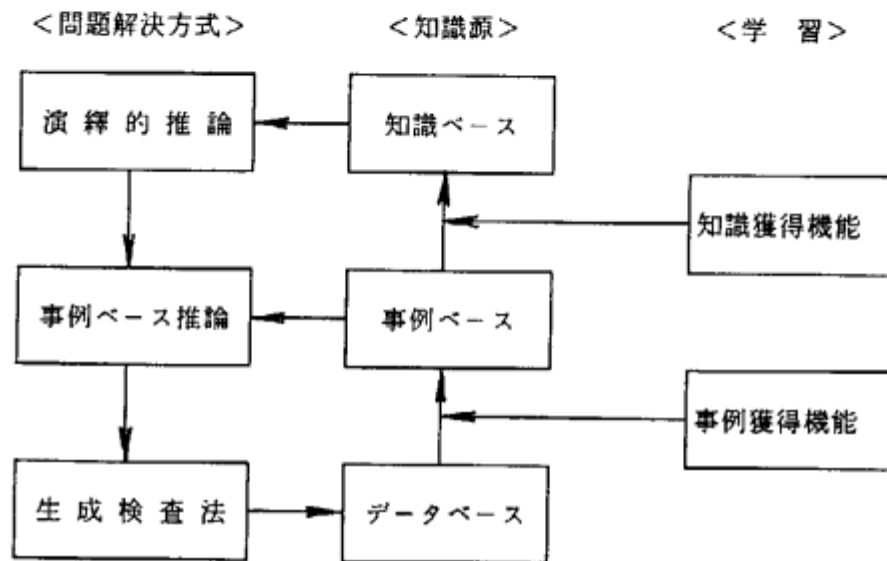
本SWGの活動の主要な成果はつぎのように要約される。

- 1) 計画および設計問題領域におけるCBRの具体的なニーズを抽出し、CBR的接近の必要性を明らかにし、知識システムを補完または代替する強力な問題解決パラダイムとして、きわめて有望であることを明らかにしたこと。
- 2) CBRに関する内外の研究動向を調査し、その結果、CBRの重要性を再認識するとともに、この分野での研究は米国が先行しているとはいえ、研究はまだ緒についたばかりであるとの認識から、わが国におけるCBRの研究開発体制を整え、研究を活性化することの必要性を認識したこと。
- 3) CBRの各構成要素について工学的な検討を行い、特に、事例の定義、事例の特徴づけ、事例ベースの組織化、事例の類似性の定義、最適な事例の検索、事例の修正および修復について、多角的な検討を加え、有用と思われるいくつかのアイデアを提示したこと。
- 4) 本SWGの委員およびオブザーバが実際に開発済み、ないしは開発中のCBRシステムについて、経験から得られた知見を報告し、意見を交換することにより、CBRシステムがもつべき機能を明らかにしたこと。
- 5) CBRは知識システムの側面、探索システムの側面、データベースの側面を併せもつこと、したがって統合的な問題解決システムを構築する上でCBRが中心的な役割を担うことを指摘したこと。
- 6) CBRとEBLの関係について検討し、両者は極めて密接な関係にあることを指摘し、それゆえCBRは問題解決と学習の橋渡しの存在になりうることを指摘したこと。
- 7) 問題解決と学習という観点から、CBRの技術的課題を多数の項目にまとめ、CBRの技術的基盤を確立する上で解決されるべき課題を提示したこと。

以上を要するに、本SWGでは、CBRに関するニーズ分析、研究動向の把握を踏まえて、CBRの要素技術およびシステム化技術について、さまざまな観点からの検討を加え、その成果をまとめた本報告書はわが国におけるCBRの発展に寄与するものと期待される。

## 6. 2 提 言

事例ベース推論は、知識獲得がネックとなっている知識ベース推論の代替的方法に留まらず、問題解決と学習を統合化するという枠組みにおいて中核的な役割を担うものと期待してよい。事例ベース推論は、下に示す枠組みの中に位置付けられ、この枠組みを具体化することが次世代知識システムのアーキテクチャを確立する上で重要である。



現在のエキスパートシステムは、十分な経験をもつ領域専門家の存在を前提として、領域専門家から抽出した知識を知識ベースに移植して、演繹的推論によって問題解決を行うことを基本としている。しかし、この枠組みは、知識獲得ボトルネックの問題を引き起こしている。特に、計画や設計のように、半ば開いた問題領域においては、この問題は深刻である。ここに、C B R 的接近の導入が要請される。

計画および設計の問題領域では、定型的問題を除いて、問題解決過程において利用可能な知識は不完全であることが普通であり、事例の利用がこれを補完しているとみられる。その意味で、C B R は知識ベースに基づく演繹的推論を補完する枠組みとして有用である。C B R の利点は数多くあるが、その中でも特に、学習的側面が重要である。すなわち、C B R による問題解決自身が新しい事例の生成につながり、C B R の問題解決能力を段階的に高めていくことができるところに C B R の価値がある。

事例ベースに事例が十分蓄積されたならば、事例間の類似性の分析または深い知識に基づく事例の理解とその汎化により、事例を知識化することが可能になる。上の枠組みでいう知識獲得機能とは、事例を知識に変換するための学習機能をいう。ここでは、説明に基

づく学習（E B L）およびこれを補完する類似性に基づく学習（S B L）が重要な役割を担うことは言うまでもない。

C B Rによる問題解決が失敗した場合には、もっとも弱い問題解決方式である生成検査法に依存せざるを得ない。生成検査法、すなわち探索ベースでの問題解決は、力まかせの試行錯誤的方法ではあるが、十分な時間とコストをかければ、それなりに良い解を生成する点において安全、確実な方法である。探索の履歴をデータベースに蓄積して、これを再利用することを考えれば、ここでもC B Rの枠組みが利用可能である。探索データはそのままでは事例として利用可能ではなく、事例に変換するためには、深い知識に基づく特徴づけが必要であり、その際E B Lが利用可能である。

以上、議論してきたように、問題解決パラダイムの統合および問題解決と学習の統合は、計画や設計など合成型問題領域へ接近する上での自然かつ必須の枠組みであり、その中でC B Rが中核的な役割を担い、また背後にある学習機能を実現する上でE B Lが重要な役割を担うことに留意されたい。

したがって、要素技術としてのC B RおよびE B Lを精緻化することの必要性ならびに、これらをベースにして、問題解決と学習を統合化したアーキテクチャを確立することの必要性を再度指摘して、本S W Gの提言とする。