

知的情報処理システムに関する調査研究報告書

第3分冊 知識獲得支援

平成元年3月



財団法人 日本情報処理開発協会

ICOT-JIPDEC AI センター



この報告書は、日本自転車振興会から競輪収益の一部である機械工業振興資金の補助を受けて昭和63年度に実施した「知的情報処理システムの導入・活用に関する調査研究」の成果の一部をまとめたものであります。

まえがき

通商産業省情報処理振興審議会は、1986年3月の答申において、1990年代を目標とする第4期「電子計算機利用高度化計画」を取りまとめましたが、この中で、従来のコンピュータ利用技術の延長線上では捉えきれない新しい概念として「知識情報処理」を取り上げています。また、情報処理振興審議会の答申から遡ること4年、1982年4月に通商産業省では、世界に先駆けて、知識情報処理機能を組み込んだ全く新しいコンセプトの「第五世代コンピュータ」の開発をスタートさせました。

欧米諸国もこの日本の第五世代コンピュータプロジェクトに刺激を受け、相次いで知識情報処理技術の研究開発をナショナル・プロジェクトとして取り上げています。

一方、産業界でも、各種の知識システムの開発、導入に積極的に取り組み始めていますが、構築方法論が確立されていないことや知識獲得の問題などがあり、知識システムの多くは、未だプロトタイプの段階に留まっています。

このような現状認識に基づき、ICOT-JIPDEC AIセンターでは、昭和61年度から「知的情報処理システム調査研究委員会」を設置し、学界、産業界で活躍している方々により、3年間にわたって活発な調査研究活動を行い、ここに、全5分冊に及ぶ研究成果を取りまとめました。

本調査の実施にあたっては、委員会の委員の方々はもとより関係省庁ほか関係各方面から御指導・御協力をいただきました。ここに謝意を表する次第であります。

平成元年3月

財団法人 日本情報処理開発協会

会 長 影 山 衛 司

「知的情報処理システムに関する調査研究」概要

わが国における知識システムの研究開発の数は、2,000を越えたとみられ、産業界のあらゆる分野への応用の裾野が広がりつつある。しかし、そのうちフィールドテストを経て、実用化の段階に達したシステムの数は、多めに見積もっても、500以下とみられ、多くのシステムはプロトタイプの段階またはデモンストレーションの段階に留まっているとみられる。その理由として、いくつかの要因が考えられるが、その中で、特に、

- 1) 知識システムを構築していくことを支援する方法論が確立されていないこと、
- 2) 知識システムを試作・実装するためのツールが十分には整備されていないこと、
- 3) 知識獲得を支援する方法論やツールの研究開発が立ち遅れていること、
- 4) 知識システム技術者が不足し、育成の指針が確立されていないこと、
- 5) 開発事例を蓄積・分析し、結果を有効に利用する体制が確立されていないこと、

などが指摘されている。このような現状認識に基づき、ICOT-JIPDEC AIセンターは昭和61年9月に「知的情報処理システムに関する調査研究委員会」を設置し、つぎのようなテーマに関する調査研究活動を行ってきた。

- 1) 知識システム構築方法論を確立すること、
- 2) 知識システム構築ツールの評価体系を確立すること、
- 3) 知識獲得支援の動向調査を行い、今後あるべき支援のイメージを提案すること、
- 4) 知識システム技術者を育成するためのガイドラインを作成すること、
- 5) 知識システム開発事例の収集・分析を行い、今後の開発の資とすること、

調査研究活動の前半（昭和61年9月～昭和62年9月）では、知識システム構築方法論および知識システム構築ツールの評価に重点をおき、その研究成果は昭和62年9月の成果発表会で報告され、その詳細は「知識システム開発方法論」および「知識システム開発ツールの評価」と題する報告書にまとめられている。

調査研究活動の後半（昭和62年9月～平成元年3月）では、計画・設計型問題の構築方法論、構築ツールの性能評価、知識獲得支援の動向分析、知識システム技術者の育成方法および知識システム開発事例の分析に重点をおいて活動を展開してきた。

調査研究の成果は、以下の5分冊からなる本報告書にまとめられている。

第1分冊「計画・設計型知識システムの構築方法論」では、計画・設計型知識システムの特徴分析を行い、計画・設計型問題には探索的側面、意思決定的側面、事例利用的側面および知識獲得的側面の4つがあることを指摘、各側面に対して、問題解決的接近、仮説推論的接近、

事例ベース推論的接近および学習的接近の必要性和枠組みをまとめている。

第2分冊「知識システム構築ツールの評価」では、システム構築ツールを評価するための評価体系を整備、ツールの性能評価を行うための標準問題を作成、AIセンターオープンハウスに導入されているツールに対し性能評価のテストを行い、評価結果を取りまとめるとともに、ツールの評価に関するいくつかの提言を行っている。

第3分冊「知識獲得支援」では、知識獲得支援ツールの位置づけ・分類を行い、文献調査とヒアリングに基づき、各ツールの機能と特徴を客観的にとりまとめ、知識システム開発事例における知識獲得の実際を分析、知識獲得支援のツールのあり方および知識獲得支援ツールに対する要求事項をまとめている。

第4分冊「知識システム技術者育成ガイドライン」では、知識システム技術者の役割を、知識システムの開発手順の段階に対応して、システムアナリスト、プロトタイピングアナリスト、知識プログラマ、保守・管理技術者の4つに類別した上で、修得すべき技術を人工知能技術、知識工学技術、システム技術の3つに分け、その概要をまとめている。

第5分冊「知識システムの事例」では、生体や社会などの解釈／診断問題事例、プラントや人工衛星などの制御問題事例、スケジューリング、レイアウト、LSI設計などの計画／設計問題事例など全部で23の代表的な事例を取り上げ、比較可能な同一の枠組みで紹介し、さらに各システムにおける基本タスクと問題解決機能を明らかにしている。

以上を要するに、本報告書は知識システムの研究開発においてその解決が緊急とされる課題を重点的に取り上げ、産業界と学会で活躍している技術者・研究者からなる横断的な組織によって調査研究を行った成果である。本報告書がわが国における知識システム研究開発の一層の発展に寄与する資として役立つことを願っている。

最後に、本調査研究を取りまとめる上で、数十回におよぶ会合を通じて、終始、熱心にご協力頂いた委員の各氏ならびにこれを支えてくれた事務局の皆様に感謝の意を表する。

平成元年3月

知的情報処理システム調査研究委員会 委員長 小林 重信

昭和63年度 知的情報処理システム調査研究委員会

委員名簿

委員長	小林 重 信	東京工業大学
幹 事	寺 野 隆 雄	(財)電力中央研究所
委 員	関 根 史 麿	花王(株) 知識情報科学研究所
"	菊 地 一 成	キヤノン(株) 情報システム研究所
"	山 崎 知 彦	(株)豊田中央研究所
"	森 啓	日本電気(株) C & C システム研究所
"	千吉良 英 毅	(株)日立製作所 システム開発研究所
"	中 島 俊 哉	富士通(株) 科学システム部
"	小 林 慎 一	(株)三菱総合研究所
"	福 島 正 俊	三菱電機(株)中央研究所
"	森 谷 正 晴	新日本製鐵(株) 君津製鐵所 設備部
"	桃 井 茂 晴	日本電信電話(株) NTT情報通信処理研究所
"	畠 見 達 夫	長岡技術科学大学
オブザーバ	生 駒 憲 治	(財)新世代コンピュータ技術開発機構
"	渡 辺 敏	新日本製鐵(株) 君津製鐵所 設備部
事務局	平 井 吉 光	(財)日本情報処理開発協会 AI振興センター
"	茂 呂 知 明	" "

知識獲得支援ワーキンググループ

委員名簿

主査	寺野隆雄	(財)電力中央研究所
委員	山崎知彦	(株)豊田中央研究所
"	小林慎一	(株)三菱総合研究所
"	福島正俊	三菱電機(株) 中央研究所
"	森谷正晴	新日本製鐵(株) 君津製鐵所 設備部
"	桃井茂晴	日本電信電話(株) NTT情報通信処理研究所
"	畝見達夫	長岡技術科学大学
オブザーバ	渡辺敏	新日本製鐵(株) 君津製鐵所 設備部
執筆協力者	松本俊二	富士通(株) システム技術部
"	滝寛和	(財)新世代コンピュータ技術開発機構
事務局	平井吉光	(財)日本情報処理開発協会 AI振興センター
"	茂呂知明	" "

知識獲得支援・目次

1. 序論	1
1.1 調査研究の背景	1
1.2 調査研究の方法と報告書の構成	4
2. 知識獲得支援ツールの動向	7
2.1 概 要	7
2.1.1 知識獲得支援ツールの歴史	8
2.1.2 知識管理のツール	9
2.1.3 知識変換のツール	10
2.1.4 知識収集のツール	12
2.2 文献調査とヒアリング	17
2.2.1 概 要	17
2.2.2 文献調査とヒアリング	18
2.2.2.1 TEIRESIAS	18
2.2.2.2 ONCOCIN	21
2.2.2.3 MORE	23
2.2.2.4 MOLE	25
2.2.2.5 AQUA	27
2.2.2.6 LEAP	29
2.2.2.7 Argo	32
2.2.2.8 SEEK/SEEK2	35
2.2.2.9 EPSILON/EM	40
2.2.2.10 ETSとAQUINAS	43
2.2.2.11 COLAB	47
2.2.2.12 CONSIST	53
2.2.2.13 CYC Editor	57
2.2.2.14 ES/SDM	61

3. 知識システム開発事例にみる知識獲得の実際	63
3.1 概 要	63
3.2 分析型問題	66
3.2.1 クレジット入会審査エキスパートシステム	66
3.2.2 自動車の故障診断システム	72
3.2.3 高炉操業支援エキスパートシステム	78
3.3 計画・設計型問題	84
3.3.1 シミュレータのユーザインタフェース設計	84
3.3.2 連接水系の翌日発電計画システム	88
3.3.3 ルート計画エキスパートシステム	95
3.3.4 電子装置の実装設計支援システム	100
4. 結 論	105

1. 序 論

20
21

1. 序 論

1. 1 調 査 研 究 の 背 景

知識システムは、これまで、ごく狭い分野の問題を経験則レベルで処理することで効果を上げてきた。そして、計算機システムの問題解決機能の向上には、うまく整理された専門知識が有用なことが実証されるようになってきている[寺野 1988]。その一方で、知識システムはプロトタイプレベルまでは開発が容易であるが、それを実用化するのは非常に困難なことが指摘されている。この困難の多くは、問題解決に必要な知識を獲得する作業の難しさに起因している。知識獲得は、狭義には、対象分野の専門家から経験的知識を抽出する作業と定義されるが、広義には、知識システム構築の初期段階から最終局面に至るまでの各フェーズにおいて適切な支援を行うことと定義される[小林 1988]。

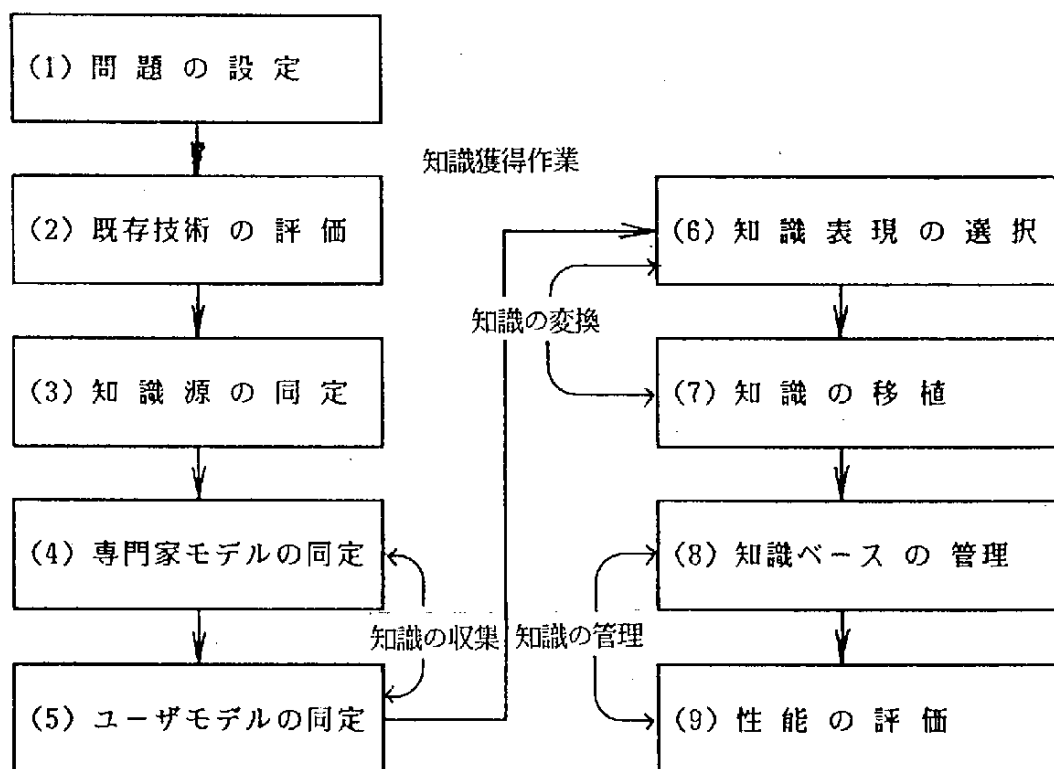


図 1. 1. 知識システムの構築手順 [ICOT-JIPDEC87]

知識システムの開発作業は図1. 1のようにまとめられる。この中には、解決したい問題の設定や既存技術の評価の段階、もしくは、完成したシステムの配布・流通といった段階までが含まれる[JIPDEC AIセンター 1987-1]。このプロセスのうち、(4) 専門家モデルの同定以降のフェーズには、知識、すなわち、問題解決に必要な情報を、専門家などの知識源から収集・定式化し、それを人工知能技術が提供する手法で取扱える形に変換してコンピュータ内に埋め込む作業が必要である。また、知識システムは一度動きだしてから、知識の誤りや矛盾、あるいは、知識の増大に対応して、常に、変更され続けるのが普通である。本報告では、知識獲得作業を知識の収集・知識の変換・知識の管理の3つの意味で用いる。これを図1. 1と対応させれば、それぞれ、"知識の収集"には、(4) 専門家モデルの同定、(5) ユーザモデルの同定のフェーズに、"知識の変換"には、(6) 知識表現の選択、(7) 知識の移植のフェーズに、"知識の管理"には、(8) 知識ベースの管理、(9) 性能の評価のフェーズに対応する作業が含まれる。

知識の収集・知識の変換・知識の管理の3つの作業を、知識システム開発の段階に即して、それぞれ上流・中流・下流ということばで表現する場合もある。上流段階は、対象問題を基本概念レベルで整理し、対象モデルを構築する作業が中心である。中流段階は、部分問題の特性を詳細に把握し、探索空間を明確に設定した上で、ルールやフレームなどの知識処理の典型的な手法で知識を記述していく作業が中心である。下流段階では、知識システムを動かしながら、知識ベースの検証・修正・拡張を行っていくこととなる。

また、別のテキスト[Hayes-Roth 1983]では知識システム開発のプロセスを図1. 2のようにまとめている。この知識システム開発のプロセスと知識獲得の3つの作業との関連は、図1. 3にまとめられる。知識獲得支援ツールは、この知識獲得の作業をできるだけ機械で支援することをねらったソフトウェアシステムであり、現在、実用的な面からも理論的な面からも、積極的な研究開発がなされている。

知的情報処理システム調査研究委員会は、昭和61年度より三年間にわたって知識システム開発方法論の調査研究、市販の知識システム構築ツールの評価のための調査研究を継続して行ってきた。本年度は、これらの活動に加えてあらたに、知識獲得問題を取りあげこれを支援ツールの調査・研究という視点からとりまとめることとした。

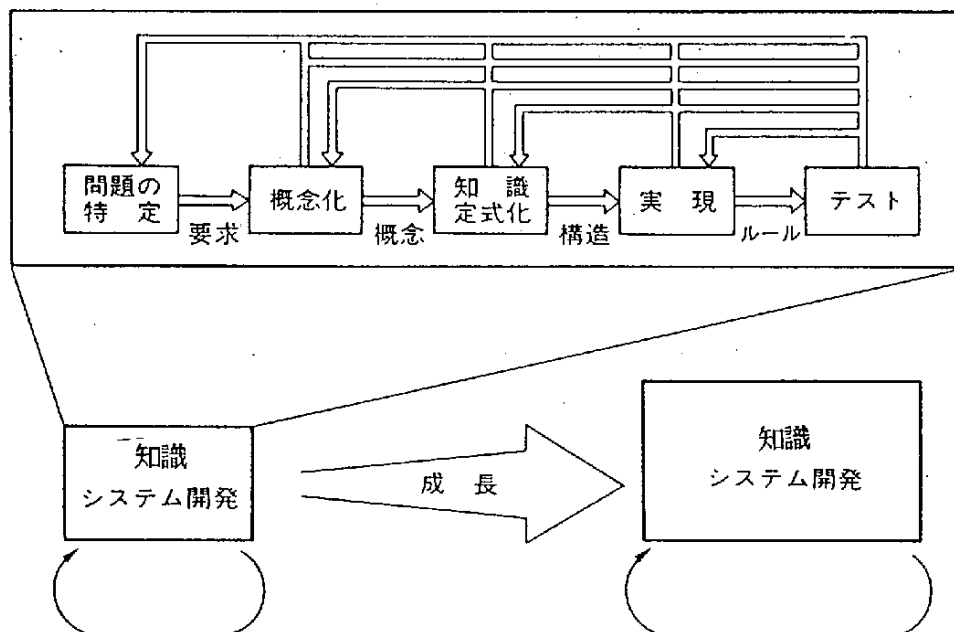


図 1. 2. 知識システム向けのプログラム開発過程 [Hayes-Roth 1983]

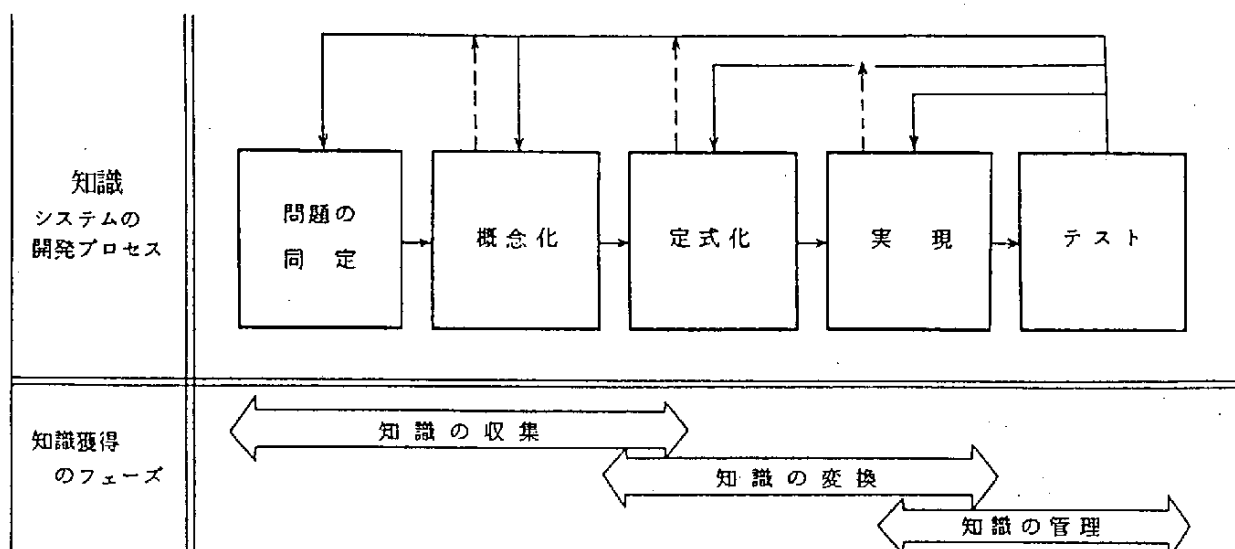


図 1. 3. 知識システムの開発過程と知識獲得のフェーズとの関係

1. 2 調査研究の方法と報告書の構成

本調査研究の実施にあたっては、本委員会の中に、寺野が主査となって、知識獲得ワーキンググループを設置した。そして、知識獲得支援ツールの目的・特徴とその研究開発動向とを代表的な事例に即してとりまとめる方針をとった。そこで、まず、代表的な知識獲得支援ツールの特性について、広範囲の文献調査とヒアリング調査とを実施し、ついで、その知見に基づいて、ワーキンググループ各委員がみずから開発に携わった知識システム事例を分析し、それぞれの対象問題に即した理想的な知識獲得支援ツールのイメージを提案した。ただし、知識獲得支援ツールの大部分は今なお研究論文レベルにあり、その方向性も明確になっているとは言い難い。そこで、本報告では、各委員の提案したそれぞれのイメージを尊重し、あえてワーキンググループとして意見を統一しないこととした。

本報告書は上の2つの作業結果をとりまとめたものである。まず2章は、論文レベルで発表された知識獲得支援ツールの動向とヒアリング調査の結果とをとりまとめている。3章は、事例分析と知識獲得支援ツールのイメージのまとめである。4章では、本報告書の結論を述べる。

なお本報告の執筆担当者と分担は以下のとおりである。外部から快くヒアリングと執筆に応じていただいた、滝 寛和氏(ICOT)、松本俊二氏(富士通)、ヒアリングに応じていただいた伊藤公俊助教授(東京工業大学)に感謝の意を表する。

- 1. 序論, 2. 知識獲得支援ツール, 2.1. 概要: 寺野;
- 2.2.2.1. TEIRESIUS, 2.2.2.2. ONCOCIN: 畝見委員;
- 2.2.2.3. MORE, 2.2.2.4. MOLE, 2.2.2.5. AQUA: 山崎委員;
- 2.2.2.6. LEAP, 2.2.2.7. ARGO: 桃井委員;
- 2.2.2.8. SEEK, SEEK-II: 福島委員;
- 2.2.2.9. EPSILON/EM: 滝 寛和氏(ICOT);
- 2.2.2.10. ETS, AQUINAS: 小林慎一委員;
- 2.2.2.11. COLAB: 森谷委員;
- 2.2.2.12. CONSIST, 2.2.2.13. CYC Editor: 寺野
- 2.2.2.14. ES/SDEM: 松本俊二氏(富士通)
- 3. 知識システム開発事例にみる知識獲得の実際, 3.1. 概要: 寺野

- 3.2. 分析型問題, 3.2.1. クレジット入会審査エキスパートシステム: 小林慎一委員;
3.2.2. 自動車の故障診断システム: 山崎委員;
3.2.3. 高炉操業支援エキスパートシステム: 森谷委員;
3.3. 計画・設計型問題, 3.3.1. シミュレータのユーザインタフェース設計: 畝見委員;
3.3.2. 連接水系の翌日発電計画システム: 寺野;
3.3.3. ルート計画エキスパートシステム: 福島委員;
3.3.4. 電子装置の実装設計支援エキスパートシステム: 桃井委員;
4. 結論: 寺野.

参 考 文 献

- [Hayes-Roth 1983] Hayes-Roth, F., Waterman, D.A., and Lenat, D.B. (eds.): "Building Expert Systems." Addison-Wesley, 1983.
[ICOT-JIPDEC87] ICOT-JIPDEC AIセンター: "知的情報処理システムに関する調査研究報告書-知識システム開発方法論-", 1987年9月.
[小林 1988] 小林重信: 知識システム技術の現状と将来. 計測と制御, Vol. 27, No. 10, pp. 859-868 (1988).
[寺野 1988] 寺野隆雄: 知識システムの構築技術. 計測と制御, Vol. 27, No. 10, pp. 869-874 (1988).

2. 知識獲得支援ツールの動向

1. 2. 3. 4. 5. 6. 7. 8. 9. 10.

2. 知識獲得支援ツールの動向

2. 1 概 要

1 章で述べたように、知識獲得の作業は、知識の収集・知識の変換・知識の管理の3つに分類できる。代表的な知識獲得支援ツールとそれが支援する作業の範囲は図2. 1にまとめられる。ただし、現在発表されている知識獲得支援ツールのほとんどは、知識システム開発の対象分野としては、分析型問題、それも、特に、単純な分類型の診断問題を対象としている。計画・設計型問題に適用可能な知識獲得支援ツールについては、必要な機能や方法論が明確になっておらず、今後の研究が期待されている。以下では、知識獲得支援ツールの歴史を概観し、各作業で利用されるツールの特徴を述べる。

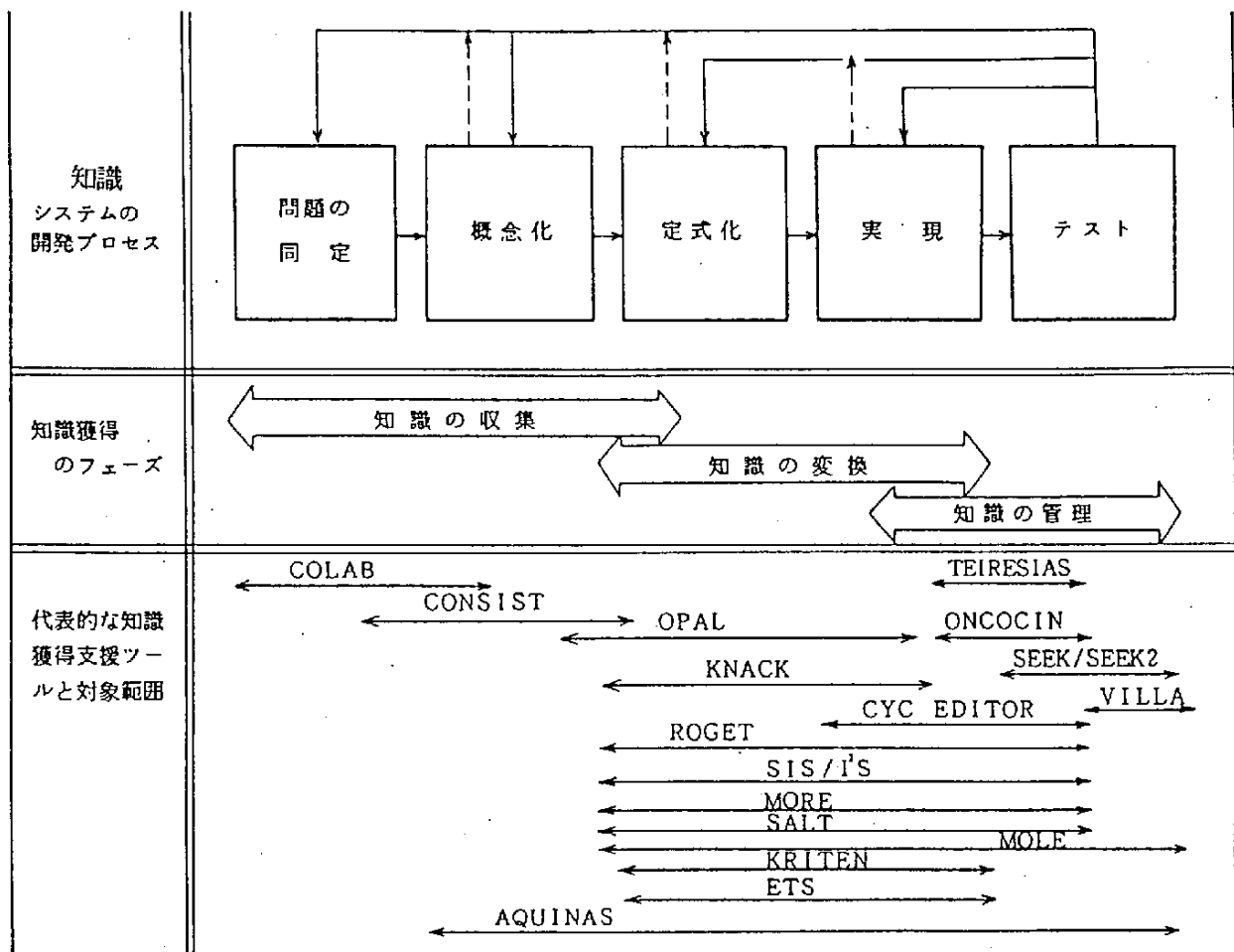


図2. 1. 知識システムの開発過程と知識獲得支援ツールの関係

2. 1. 1 知識獲得支援ツールの歴史

知識獲得は、従来から学習に関連する話題のひとつとして研究が進められてきた。その中でもっとも高度なもののひとつに初等数学と集合論における概念の発見を扱うプログラムAMがある[Davis and Lenat 1982]。しかし、これらの帰納的・演えきの学習に関する成果は、ごく最近までは理論的興味にとどまり、知識工学的な立場から、知識システム開発への応用可能性が検討されたことはなかった[計測自動制御学会 1988]、[国藤 1988]。

実用的な知識獲得支援ツールは、実際に開発された知識システムの保守・拡張作業（知識ベース管理）を効率化するというきわめて現実的な問題に即して研究開発が進められてきた。これは、特定の知識システムをもとに汎用のAIツールが開発された状況ときわめて類似していると考えられる。たとえば、化学実験データの解釈システムDENDRALから発展した知識獲得支援ツールMeta-DENDRAL[Lindsay 1980]、医療診断システムMYCIN用の知識獲得支援ツールTEIRESIAS [Davis and Lenat 1982]、[Davis 1984]などは、この系統に属するツールである。これら初期の知識獲得支援ツールは、知識ベースの保守過程を支援する。TEIRESIASについては2.2.2.1で詳しく解説する。

また、いわゆるハイブリッド型知識システム構築ツール[寺野 1987]、[ICOT-JIPDEC AI センター1987-2]の知識ベースエディタは、知識の変換・管理作業の面から知識システム開発を効率化するための重要な機能であるが、これも一種の知識獲得支援ツールとみることができる。ツールの中には、デシジョンテーブルをルール形式に変換することで知識獲得を支援するものもある。これは非常にプリミティブな機能ではあるが、小規模な分析型知識システムの開発にあたっては有用なケースも多い。また、各種の知識にさまざまな関連性をつけて保存し、操作するハイパーテキスト[Conklin 1987]の概念も、知識獲得を支援するひとつの重要な機能である。知識エディタに属する現在のレベルでもっとも高度なツールは、百科事典的な大規模知識ベースCYCの開発[Lenat 1987]に利用されている専用フレームエディタ[Shepherd 1988]である。これについては2.2.2.13で紹介する。

これらの高機能な知識エディタに対し、対象とする問題の型を限って、汎用のツールとして設計開発されたものも多い。代表的なツールに、専門家のモデル論的なアプローチをとったMORE[Kahn 1985]（2.2.2.3参照）、MOLE（2.2.2.4参照）[Eshellman 1986]、[Eshellman 1987]、ROGET[Bennet 1985]、Personal Construct Theory(PCT)という心理学的手法を適用したETS[Boose 1984]、[Boose 1986]がある（2.2.2.10参照）。また、演え

きの学習手法のひとつである説明に基づく学習 (Explanation Based Learning) を利用した LEAP [Mitchell 1985], ARG0 [Huhus 1988] (2.2.2.6, 2.2.2.7 参照) や、定性推論の概念を利用する AQUA [Ishida 1988] (2.2.2.5 参照) など人工知能研究の新しい成果を利用したツールも発表されている。

最近では、知識獲得手法確立への期待や関心の高まりもあって、3つの作業全般にわたってさまざまなツールが発表されている。その中では、知識獲得作業をできるだけ広い範囲にわたって支援することを目的としている点で、ETS の後継システムである AQUINAS [Boose 1987] (2.2.2.10 参照) が注目されている。問題解決のプリミティブを対象問題の型の応じた類型的タスク (Generic Tasks) の形で与えておき [Chandrasekaran 1986] それにしたがって知識システム開発を行うツールも広い意味で、知識獲得支援作業を全般にわたって効率化するものと考えられる。これに属するツールに CSRL [Bylander 1986] がある。

知識獲得の初期段階、すなわち、知識の収集作業を支援するものもいくつか見られる。たとえば、KJ 法 [川喜多 1967, 1970, 1987] に基づいて知識整理を支援する CONSIST (2.2.2.12 参照) [篠原 1986] や、特に知識システム開発を目的としたシステムではないが、小人数での会議を効率的にガイドする COLAB [Stefik 1987] などはこれにあたる (2.2.2.11 参照)。

以下では、知識獲得の3つのフェーズに即し、下流から上流へむかう方向で代表的な知識獲得支援ツールの概要を紹介する。

2.1.2 知識管理のツール

知識の管理作業や知識ベースのデバッグとは知識ベースの一貫性を保持すると同時に、知識ベースに誤りや矛盾が生じた場合にそれを修正する作業である。この範疇にはいる代表的な知識獲得支援ツールとして、TEIRESIAS と CYC の知識ベース開発用エディタを取上げて説明し、関連する研究に触れることとする。

TEIRESIAS は初期の知識獲得支援ツールである。このシステムは、MYCIN において初期の知識ベースの存在を前提とした上で、後ろ向き推論メカニズムと HOW, WHY の説明機能を用いて、ルール間の矛盾の発見・修正を効率化するようにコンサルテーションを行う。システムは、完全な知識をもつ専門家との (自然言語による) 対話の過程で、ルール中の誤りの発見を修正作業の支援、新しいルール入力 of 支援、ならびに新し

い概念の獲得の機能をもつ。

TEIRESIASは、推論結果の誤りを専門家が指摘することによって、知識獲得を支援するが、その後に作られたONCOCINでは、実行結果によらずに知識ベース内の無矛盾性をチェックする機能を実現している。さらに、専門家と知識技術者とのコミュニケーションを助け、知識ベースをわかりやすく提示する機能をもつ [Suwa 1984]。このような機能が実現されているのは、ONCOCINはMYCINと異なり、不確実な情報（CF値）を扱わないので、ルール間の競合・冗長性・含意性・不完全性などのチェックが容易に行えることによる。また、よく似た機能をもつSEEK/SEEK-2 [Ginsberg 1988]では、知識ベース全体の診断性能を向上させるという観点から、ルールの一般化と特殊化という2つの操作によって知識ベースの強化・洗練が半自動的に行われる。

一方、CYCの知識ベース開発用エディタは、百科事典知識ベースの内容に詳しい知識技術者が、新しい知識をフレーム形式で効率よく投入することをめざしたツールである。CYCの知識ベースは、多くのスロットをもつ複雑なフレームシステムである。それにあわせて、このエディタは、知識ベースの局所的な整合性をチェックする機能が強化されているのが特徴である。最近、ワークステーション上で稼働する知識システム開発用のハイブリッドツールに付随する知識エディタも原理的にはCYCエディタに近い機能をもつものも存在する。しかし、これらは扱える知識ベースの規模という点でCYCエディタには及ばない。

2. 1. 3 知識変換のツール

知識の変換を支援する知識獲得支援ツールは、各分野の専門家が知識技術者の援助なしに、ツールと対話処理を行っていく過程で、短期間に質の高い知識ベースを構築することに重点がおかれている。この種類のツールは、開発事例の多い分析型の知識システムを対象とするものがほとんどである。この中から代表的なものとしてインタビュー戦略を重視したツールMOREと、適用分野の知識記述形式を重視したツールOPALについて詳しく述べ、関連する研究に触れることとする。

MORE [Kahn 1985]は、掘削泥水診断システムMUDの開発経験に基づいて作られた知識獲得支援ツールである。MOREの特徴は、①診断知識を仮説と兆候・条件間の関連をネットワーク表わしたドメインモデルとして記述しておくこと、②インタビューによる不完全な

初期知識ベースの洗練過程で、より強力な診断ルールを知識ベースから抽出すること、

③ルールに確信度を割り振る専門家の作業を支援することの3点にある。

このツールのインタビュー戦略は、仮説と兆候、ならびにそれらの間の形式的な因果関係のみに基づいており、仮説や兆候が実際に何であるかには依存しない。そのため、MOREは診断問題に限れば、どのような対象のエキスパートシステム開発にも利用できるという大きな特徴をもつ。実際、MOREを使って開発されたエキスパートシステムは、コンピュータのディスク故障診断、コンピュータネットワーク診断、回路診断など多岐にわたっている[Eshellman 1986]。

MOLE[Eshellman 1987]は、MOREの後継システムである。MOLEではシステムのもつワールドに関する推論機能を強化することによって、MOREでは不可能であったCF値の割当てを自動化することができる。これは初期知識ベースの改良の段階で、仮説と証拠の間の静的な解析、ならびにテストケースを流して専門家の評価を受ける動的な解析の両者を繰返すことでなされる。

MORE, MOLEと同様に、インタビューによる知識獲得を重視したツールにROGET[Bennet 1985]がある。このシステムは、専門家が問題解決の構造的知識の表現に使う様々な知識を、知識獲得の初期の段階で効率的に専門家から抽出して整理する機能をもつ。またこれらに関連した研究として、我国では、知的なインタビュー機能を重視した知識獲得支援ツールSIS/1's [川口1987]がある。

上に述べたインタビュー戦略を重視するツールに対して、各分野の専門家が慣れ親しんでいる対象の構造モデルを使うことで、専門家が対話的に知識の投入を行う作業を効率化するツールがある。OPAL[Musen 1987]は、この考え方に基づいて開発された知識獲得支援ツールで、医療診断エキスパートシステムONCOSINの知識ベース開発に使用されている。

OPALでは、治療法や薬剤を対象分野の実体ととらえている。そして、これらの実体についての知識を獲得するために、専用のテーブル形式の書式をワークステーション上に用意しておき、専門家はそのテーブルの穴埋めの作業を行なうことで、ここの薬剤の情報やその属性値を容易に知識ベースに投入することができる。また、試験結果に依存して次の診断方法を決定する判断ルールなども、試験結果を表示したテーブルに値の条件と対応する行動を記述することで行なわれ、さらに、手続き的知識は図形(アイコン)を画面上で組合せることで整理することができる。OPALで入力した知識は、オブジェクト指向言語で記述したフレーム型の中間的な知識表現で記録される。そして、特定の治療計画の知識がす

べて投入された後に、ONCOCIN の知識ベースへ変換される。

OPALによく似たアプローチを採用した知識獲得支援ツールに、KNACK [Klinder 1987]がある。KNACKの知識獲得では、専門家は対象領域の知識をレポートもしくはその部分としての確に表現できるとの仮定に基づいている。OPAL, KNACKで採用された方法は、大規模なエキスパートシステムを長期間にわたって開発するケース、または、対象分野の知識の形式が明確なケースにおいて、類似の知識を大量に知識ベースに投入しなければならない場合には有用である。

この節で述べたツールは対象領域に関するモデルを持っているので、短期間に自動的に知識の整理ができ、効率よい信頼性の高い知識ベースができる。ただし、対象領域の変化には追従できないという共通の欠点をもつ。

2. 1. 4 知 識 収 集 の ツ ー ル

知識がない初期状態からの知識の収集・獲得を支援するツールは現在のところ非常に数が少ない。知識獲得支援の範囲が広いという意味で、ETSならびにその後継システムAQUINASがある。さらに、知識システム開発支援を直接ねらったものではないが、知識の整理・発想の支援といった面で有用な、COLAB, CONSIST について述べる。

ETSは分類型問題向きの知識獲得支援ツールである。ETSでは観測される属性と選択項目(分類・仮説など)との合いだの対応関係を明確にし、それに基づいた判断ルールを作成する。このために、Personal Construction Theory (PCT)に基づいて、選択項目とそれに共通な属性・項目を分類する属性を専門家に入力させる。ついで、Rating Grid法を使って各項目の重要度を評価し、確信度つきの判断ルールを生成する。

AQUINASは、このETSをひとつの構成要素とする統合的な知識獲得ツールであり、知識獲得プロセス全体の効率化；知識ベース生成の自動化；知識ベースの改良；知識ベース管理；既存AIツールへの知識ベースの自動変換などを特徴としている。

このシステムに関連したツールとして、ハイブリッドな知識表現のもとでAQUINASと同様の機能の実現をはかったKRITON[Diederich 1987]がある。

CONSISTは、AQUINASとは異なり、知識獲得の初期段階に必要な漠然とした知識の整理作業を支援することを目的としている。このシステムは、ボトムアップな知識整理法であるKJ法に関係の階層という考え方を取入れて適切な抽象度や視点の模索を

支援する。

CONSISTがひとりの専門家の知識整理過程を支援するのに対し、COLABは、知的な電子黒板として開発されたシステムであり、小人数で行われる会議を生産的・協調的にガイドする機能をもつ。システムの思想は、WYSWIS(What You See Is What I See)という考え方であり、中央の大型タッチスクリーンを会議参加者は、手元のワークステーションから自由にアクセスすることができる。

以上、知識獲得過程の広い範囲を支援する統合的なツールと、知識獲得の初期段階の作業を支援するツールの例を述べた。

参 考 文 献

- [Bennet 1985] Bennett, J.S.: ROGET: A Knowledge-based System for Acquiring the Conceptual Structures of a Diagnostic Expert Systems. J. of Automated Reasoning, Vol. 1, No. 1, pp. 49-74, 1985.
- [Boose 1986] Boose, J.H.: Expertise Transfer for Expert System Design. 312 pp. Elsevier, 1986.
- [Boose 1987] Boose, J.H., and Bradshaw, J.M.: Expertise Transfer and Complex Problems: Using AQUINAS as a Knowledge-Acquisition Workbench for Knowledge-Based Systems. Int. J. Man-Machine Studies, Vol.26, No. 1, pp.3-28, 1987.
- [Boose 1987-2] Boose, J., and Gaines, B. (eds.): Special Issue: Knowledge Acquisition for Knowledge-Based Systems Part I, II, III. Int. J. Man-Machine Studies, Vol.26, No. 1, 2, 4, 1987.
- [Bylander 1986] Bylander, T.C., and Mittal, S.: CSRL: A Language for Classifactory Problem Solving and Uncertainty Handling. AI Magazine, Vol.7, No. 3, pp. 66-77, 1986.
- [Chandrasekaran 1986] Chandrasekaran, B.: Generic Tasks in Knowledge-Based Reasoning: High-Level Building Blocks for Expert System Design. IEEE Expert, Vol.1, No.3, pp.23-30, 1986.
- [Conklin 1987] Conklin, J.: Hypertext: An Introduction and Survey. IEEE COMPUTER,

Vol. 20, No. 9, pp. 17-41, 1987.

[Davis 1982] Davis, R., and Lenat, D.B.: Knowledge-Based Expert Systems in Artificial Intelligence. McGraw-Hill, 1982.

[Davis 1984] Davis, R.: Interactive Transfer of Expertise. in Buchanan, B.G., and Shortliffe, E.H. (eds.): Rule-based Expert Systems - The MYCIN Experiments of the Stanford Heuristic Programming Project. pp. 171-205, Addison-Wesley, 1984.

[Diederich 1987] Diederich, J., Ruhmann, I., and May, M.: KRITON: a Knowledge-Acquisition Tool for Expert Systems. Int. J. Man-Machine Studies, Vol. 26, No. 1, pp. 29-40, 1987.

[Eshelman 1986] Eshelman, L., and McDermott, J.: MOLE: A Knowledge Acquisition Tool That Uses its Head. Proc. AAAI 86, pp. 950-955, 1986.

[Eshelman 1987] Eshelman, L., Ehret, D., McDermott, J., and Tan, M.: MOLE: A Tenacious Knowledge-Acquisition Tool. Int. J. Man-Machine Studies, Vol. 26, No. 1, pp. 41-54, 1987.

[Ginsberg 1988] Ginsberg, A., Weiss, S.M., and Politakis, P.: Automatic Knowledge Base Refinement for Classification Systems. Artificial Intelligence, Vol. 35, No. 2, pp. 197-226, 1988.

[Huhus 1988] Huhus, M.N., Acosta, R.D.: Argo: A System for Design by Analogy. Proc. IEEE Conf. AI Application, 1988.

[ICOT-JIPDEC 1987-1] "知的情報処理 システムに関する調査研究報告書-知識システム開発方法論-". 1987.

[ICOT-JIPDEC 1987-2] "知的情報処理 システムに関する調査研究報告書-エキスパートシステム開発ツールの比較評価-". 1987.

[Ishida 1988] Ishida, Y.: An Application of Qualitative Reasoning to Process Diagnosis: Automatic Rule Generation by Qualitative Simulation. Proc. IEEE Conf. AI Application, 1988.

[Kahn 1985] Kahn, G., Nowlan, S., and McDermott, J.: MORE: An Intelligent Knowledge Acquisition Tool. Proc. IJCAI 85, pp. 581-584, 1985.

[Kidd 1987] Kidd, A. L. (ed.): Knowledge Acquisition for Expert Systems - A

Practical Handbook. 194 pp. Plenum Press, 1987.

[Klinder 1987] Klinker, G., Bentolila, J., Genetet, S., Grimes, M., and McDermott, J.: KNACK-Report-Driven Knowledge Acquisition. Int. J. Man-Machine Studies, Vol. 26, No. 1, pp. 65-79, 1987.

[川口 1987] 川口 (他) : インタビュシステムのためのシェル S I S の開発. 第 1 回人工知能全国大会 論文集 6-21, pp. 343-346, 1987.

[川喜多 1967] 川喜多二郎: "発想法." 中公新書, 中央公論社, 1967.

[川喜多 1970] 川喜多二郎: "続発想法." 中公新書, 中央公論社, 1970.

[川喜多 1987] 川喜多二郎: "K J 法." 中央公論社, 1987.

[計測自動制御学会 1988] 計測自動制御学会: "知識獲得と学習に関する特別講演会資料" 1988.

[国藤 1988] 国藤 進, 小林重信: 知識獲得支援ツールの研究開発動向. 富士通 (株) 編, SS研ニューズレター, 1988年1月.

[Lenat 1987] Lenat, D., and Feigenbaum, E. A.: On the Thresholds of Knowledge. Proc. IJCAI-87, pp. 1173-1182, 1987.

[Lindsay 1980] Lindsay, R. K., Buchanan, B. G., Feigenbaum, E. A., and Lederberg, J.: Application of Artificial Intelligence for Organic Chemistry-The DENDRAL Project. 194 pp. McGraw-Hill, 1980.

[Mitchell 1985] Mitchell, T. M., Mahadevan, S., and Steinberg, L. I.: LEAP: A Learning Apprentice for VLSI Design. Proc. 9th IJCAI, pp. 573-580, 1985.

[Musen 1987] Musen, M. A., Gagan, L. M., Combs, D. M., and Shortliffe, E. H.: Use of a Domain Model to Drive an Interactive Knowledge-Editing Tool. Int. J. Man-Machine Studies, Vol. 26, No. 1, pp. 105-121, 1987.

[篠原靖志 1986] 篠原靖志, 寺野隆雄: "関係の階層化を利用した知識ベース構築支援システム." 電子通信学会技術研究報告 (人工知能と知識処理研究会資料) AI86-32, 1986.

[Shepherd 1988] Shepherd, M.: Tools for Adding Knowledge to the CYC LSKB. Proc. IEEE/SICE Int. Workshop on A. I. for Industrial Applications, pp. 365-369, 1988.

- [Stefik 1987] Stefik, M., Foster, G., Bobrow, D.G., Kahn, K., Lanning, S., and Suchman, L.: Beyond the Chalkboard: Computer Support for Collaboration and Problem Solving in Meetings. Comm. the ACM, Vol. 30, No.1, pp.32-47, 1987.
- [Suwa 1984] Suwa, M., Scott, A.C., Shortliffe, E.H.: Completeness and Consistency in a Rule-Based System. in Buchanan, B.G., Shortliffe, E.H. (eds.): Rule-based Expert Systems-The MYCIN Experiments of the Stanford Heuristic Programming Project. pp. 159-170, Addison-Wesley, 1984.
- [諏訪 1986] 諏訪基, 小林重信, 岩下安男, 國藤進: エキスパートシステム開発事例にみる知識獲得の諸相. 計測と制御, Vol. 25, No. 9, pp. 801-809, 1986.
- [寺野 1987] 寺野隆雄: 知識システムの構築環境. 計測自動制御学会講習会資料-エキスパートシステム: 方法論と応用一, pp. 83-96, 1987.

2. 2 文献調査とヒアリング

2. 2. 1 概 要

本節では、知識獲得支援ツールの動向について文献調査とヒアリングを実施した結果を取りまとめる。ここで取上げたツールの情報は網羅的なものではないが、ワーキンググループ各委員が重要と考えたツールはすべて含まれている。これによって、知識獲得支援ツールの最新の動向を知ることができる。以下では、各システムまたはツールごとに、まず対象システム名称、解説の直接の出典文献の情報、知識獲得の支援レベル（知識の収集・変換・管理のいずれか）、対象分野の情報を記述する。その後、システムの概要・特徴・課題などについて順次説明を行う。

2.2.2 文献調査とヒアリング

2.2.2.1 TEIRESIAS

文 献

表題	Interactive Transfer of Expertise
著者	Randall Davis
出典	Artificial Intelligence 12, 121-157, 1979. (also in Bruce G. Buchanan and Edward H. Shortliffe (Eds.): Rule- Based Expert System - The MYCIN Experiments of the Stanford Heuristic Programming Project, Addison-Wesley Pub. Co., 171-205, 1984.)

支援レベル 追加知識の修正

対象分野 診断問題

(1) 概要

TEIRESIASは医療診断エキスパートシステムMYCINにおける知識獲得を対話的に支援するプログラムである。専門家の自然言語表現による入力を内部表現に変換（知識転換：knowledge transfer）するとともに、内部表現を自然言語表現に変換したものの（説明：explanation）を出力する。更に、専門家による新たなルールの追加に伴って起こる知識の矛盾、不足に対する修正を支援する。

(2) 操作例

1つの診断が終了した後、システム側が推論結果が正しいかどうかを専門家に問い質す。正しければ終了する。正しくなければ次のようなステップで問い合わせを行なう。（実際の操作例については文献を参照のこと）

(3) バグの発見

推論過程を遡り、誤ったルールあるいは不足しているルールを同定する。具体的には、まず過った結論部分を専門家に指示させる。誤りのタイプには2種類ある。1)導かれるは

ずの結論が導かれなかった。2)導かれてはならない結論が導かれた。1)の場合、その結論を導くルールを表示し、実行されるべきルールがあるかを問う。あればそのルールの条件部について同様の処理を行なう。なければ新たなルールの入力を促す。2)の場合、それを直接導いたルールを表示する。そのルールについて、ルール自身が正しいものであれば、ルールの前提部の中で満足されるべきでない条件を指示させる。その条件を導いたルールについて同様の処理を行なう。条件が否定を含むものであれば、1)の場合と同様に処理する。

(4) 新ルールの解釈

専門家が自然言語で入力した新ルールを、ルールモデルに従って解釈し、内部表現に変換したのち、自然言語に再度表現し直して専門家に提示する。自然言語の解析は単純なテンプレートによるものであり、複雑な文法等は使っていない。

ルールモデルは専門家によって入力されたルールを基に、よく使われるルールの特徴を結論の種類によって分類、整理したルールに関する知識、つまりメタ知識である。個々のルールモデルは4つの要素からなる。

EXAMPLE: このルールモデルに当てはまる具体的なルールの番号のリスト。

DESCRIPTION: ルールの特徴(条件部の特徴と、結論部の特徴)、属性、述語、強さの組のリストから成る。たとえば「この属性をあつかう場合、この述語関数がよく用いられる。」といった特徴が記録される。

MORE GENERAL: より一般的なルールモデルの集合。

MORE SPECIFIC: より特殊なルールモデルの集合。(IS-A階層を構成する。)

解釈時におけるルールモデルの選択は、IS-A階層を上からたどることによってなされるため、システムが知っているルールモデルの中で最も適切なものを利用できる。自然言語の解釈は次のようなステップで行なわれる。まず、複数形および同義語を統一的な単語に直し、それらの意味(対象、述語関数、属性名、属性値など)を見つける。述語関数についてのテンプレートをもとに内部表現を構成する。候補が複数ある場合には全ての解釈を生成する。最後にルールモデルのDESCRIPTION部を参照し、もっともらしい解釈の候補を選択する。手法は単純だが、TEIRESIASの利用目的には十分な能力を発揮できる。

(5) 解釈の修正

システムによる自然言語の解釈に誤りがあれば、専門家に誤り箇所を指摘してもらい修正する。ルール of 修正に用いられるルールエディタには、修正過程の効率を良くする為の経験則が埋め込まれている。ある節の解釈を修正する場合、別の解釈を再評価し、次の候補として提示する。また、同じ単語には同じ解釈を与えるという制約も有効である。

(6) 考え直し

最終的なルールが、ルールモデルと整合性があるか確かめる。記述の不足が予想される場合には、その旨を専門家に提示し、適当な節を追加するかどうかたずねる。専門家が追加を要請した場合はルールモデルとテンプレートに従って追加すべき節の要素を決定し、未知の部分についてだけ専門家に質問する。

(7) 完了作業および再試行

新たなルールについて、バグを回避するものかどうかをチェックし、ルールモデルの再計算を行なった後、前回の対話の応答を参考にして、診断プロセスを再試行する。

(8) ルールモデルの別の使い道

MYCINに元々備えられている説明機能の出力文を、ルールモデルを利用して改良することができる。

[参考] その他の文献

Randall Davis: TEIRESIAS: Application of Meta-Level Knowledge, R. Davis and D. Lenat (Eds.), Knowledge-Based Systems in Artificial Intelligence, McGraw-Hill, 1982.

TEIRESIAS, Avron I. Barr and Edward A. Feigenbaum (Eds.) The Hand-book of Artificial Intelligence Vol. II, William Kaufmann Inc., VII-B, 1982.

(日本語訳: 田中幸吉, 淵一博監訳, 人工知能ハンドブック, 第II巻, 共立出版, VII-B, 1983)

2.2.2.2 ONCOCIN

文 献

- | | |
|----|---|
| 表題 | Completeness and Consistency in a Rule-Based System |
| 著者 | Motoi Suwa, A. Carlisle Scott, and Edwards H. Shortliffe |
| 出典 | The AI Magazine 3: 16-21, Autumn 1982. (also in Bruce G. Buchanan and Edward H. Shortliffe (Eds.): Rule-Based Expert System - The MYCIN Experiments of the Stanford Heuristic Programming Project, Addison-Wesley Pub. Co., 159-170, 1984.) |

支援レベル 知識の修正

対象分野 診断問題

(1) 概要

ONCOCINはMYCINに似た後向き推論方式の医療診断支援システムである。ただし、あいまい性の処理は行なわない。知識ベースのデバッグを支援する次のような機能を備えている。知識ベース内の矛盾性をチェックする。専門家とKEとのコミュニケーションを助ける。知識を分かりやすく表示する。

(2) 知識ベースのチェック

知識ベースの全体について矛盾 (inconsistency) と不完全性 (Incompleteness) のチェックを行なう。矛盾の原因としては競合 (Conflict) すなわち、同じ状況で成功する2つのルールが相互に競合する結論を導く場合、冗長性 (Redundancy) すなわち、同じ状況で成功し同じ結論を導くルールが2つ以上ある場合、含意性 (Subsumption) すなわち、同じ結論を導く2つのルールの条件が一方が他方を含んでいる場合の3つがある。不完全性とは、ルールの欠如、すなわち、推論が必要な状況に対して、適用できるルールがない場合である。

(3) ルール・チェック・プログラム

知識ベースのチェックは次のような手順で行なわれる。まずチェックの対象となるルー

ル集合として、同じ文脈において同じパラメータの値を結論するルールを集める。それらのルールの条件部に現れる全てのパラメータを集め、可能な値の全ての組合せを作る。パラメータ値が連続値を取る場合には条件として意味のある区間に区切る。各組合せについて、それを満足するルールを見つけ、ルールと条件パラメータ値の組合せ、結論パラメータ値の対応表（表2.2.2.2-1参照）を作り、専門家に表示する。

この表から、競合、冗長、含意、ルールの欠如を発見する。ルールの欠如とは、可能な条件パラメータの組合せについて、それを満足するルールが存在しない場合である。欠如していると思われるルールについては、専門家が必要とすれば、システムが自動的にその条件部を生成する。

(4) ルール・チェックの効果

専門家が見落していたケース、文脈の違い、パラメータと値の設定の枠組みに関する矛盾などを指摘できた。

対応表による表示は、1つ1つのルールを羅列したものより理解しやすい。

表2.2.2.2 - 1 ルールとパラメータの対応表の例

ルール集合：667 600 82 80 69 76

実行部パラメータ：

条件部パラメータ：NORMALCOUNTS, CYCLE, SIGXRT

Evaluation	Rule	Value	NORMALCOUNTS	CYCLE	SIGXRT	Combination
R	80	250mg/m ²	YES	1	1	C ₁
	76 (D)	V ₁	YES	(1)	(1)	C ₁
	667	V ₂	NO	1	1	C ₂
	67	V ₂	NO	1	1	C ₂
M	76 (D)	V ₁	YES	(1)	(OTHER)	C ₃
	—	—	NO	1	OTHER	C ₄
	82	V ₃	YES	OTHER	1	C ₅
	76 (D)	V ₁	YES	(OTHER)	(1)	C ₅
C	600	V ₃	NO	OTHER	1	C ₆
C	69	V ₄	NO	OTHER	1	C ₆
M	76 (D)	V ₁	YES	(OTHER)	(OTHER)	C ₇
	—	—	NO	OTHER	OTHER	C ₈

C ... 競合のある組合せ：C 6 (RULE600, RULE069)

R ... 冗長な組合せ：C 2 (RULE667, RULE067)

M ... ルールが欠如している組合せ：C 4, C 8

2.2.2.3 MORE

文	献
表題	Strategies for Knowledge Acquisition
著者	Gary Kahn, Steven Nowlan and John McDermott
出典	IEEE Trans on Pattern Analysis and Machine Intelligence, PAMI-7, No.5, September 1985
キーワード	人工知能, 自動インタビュー, 因果モデル, 診断, エキスパート システム, 知識獲得, 知識工学
支援レベル	知識の変換, 知識の管理
対象分野	診断問題 (兆候-仮説の関係が陽に表現される)

MOREは、掘削泥水診断システムMUDの開発経験に基づいて作られた知識獲得支援ツールである。その目的は、専門家自身による 1) 初期知識ベースの作成, 2) 知識ベースの洗練, 3) 確信度付与, を支援することである。

MOREの特徴は、診断知識を仮説・兆候・条件間のネットワークによって表すドメインモデルを持つ事である(右図)。このモデルは 1) 仮説(H), 2) 兆候(S), 3) 条件, 4) 接続を表すリンク(実線), 5) 仮説と兆候を結ぶパス(破線), からなる。また、条件には 1) 仮説の先験確率, 2) 検査, 3) 検査条件, 4) 兆候属性, 5) 兆候条件の5種類がある。診断の実行にはこのモデルから生成される診断ルールを用いる。

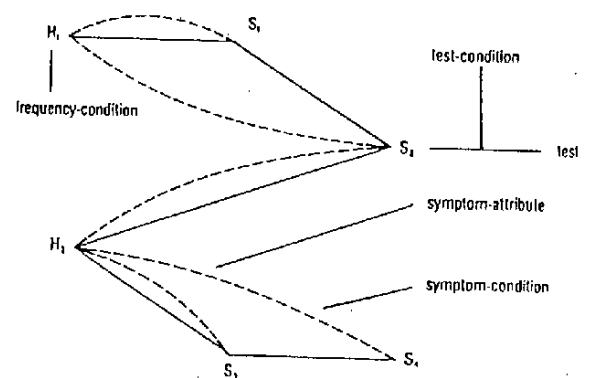


図 モデルの要素

知識ベースの洗練フェーズでは、より強力で正確な診断を行うためにMOREがドメインモデルを分析して、不足している知識の入力を助言する。そこでは、診断型の問題に対して共通に使える次の8つのインタビュー戦略を用いる。

1) 仮説を区別する兆候の決定

共通の兆候しか持たない仮説を見出し、区別するための兆候を質問する

2) 仮説の発生条件の決定

仮説の先験的な生起確率に影響を与える条件を質問する

3) 兆候の差別化

複数の仮説に結び付いている兆候について、より詳細な特徴を質問し差別化する

4) 兆候の発生条件の決定

仮説と兆候の結び付きを強める兆候の発生条件を質問し、負ルールの確信度を高める

5) 仮説→兆候パス間の新兆候の決定

結び付きの弱いパスの中間に入る兆候を質問し、負ルールの確信度を高める

6) 共通兆候をもつ仮説群を区別する新兆候の決定

ひとつの兆候に複数のパスがある場合、パスを区別する兆候を質問する

7) より正確な兆候発見検査法の決定

兆候の存在をより正確に決められる検査法や検査器具について質問する

8) 検査の精度に関する条件の決定

検査の精度に影響する条件を質問し、検査条件として表現する

これらの戦略を用いて、下図のモデルの例では以下のような質問が出される。

- 1) SHALE-CONTAMINATIONが起きやすいあるいは起きにくい条件はあるか?
- 2) INCREASE-IN-VISCOSITYの観測精度に影響する条件はあるか?
- 3) DECREASE-IN-DENSITYがWATER-INFLUXに起因することを明確にする特徴は?
- 4) WATER-INFLUXから生ずる兆候でSHALE-CONTAMINATIONでは説明できないものは?

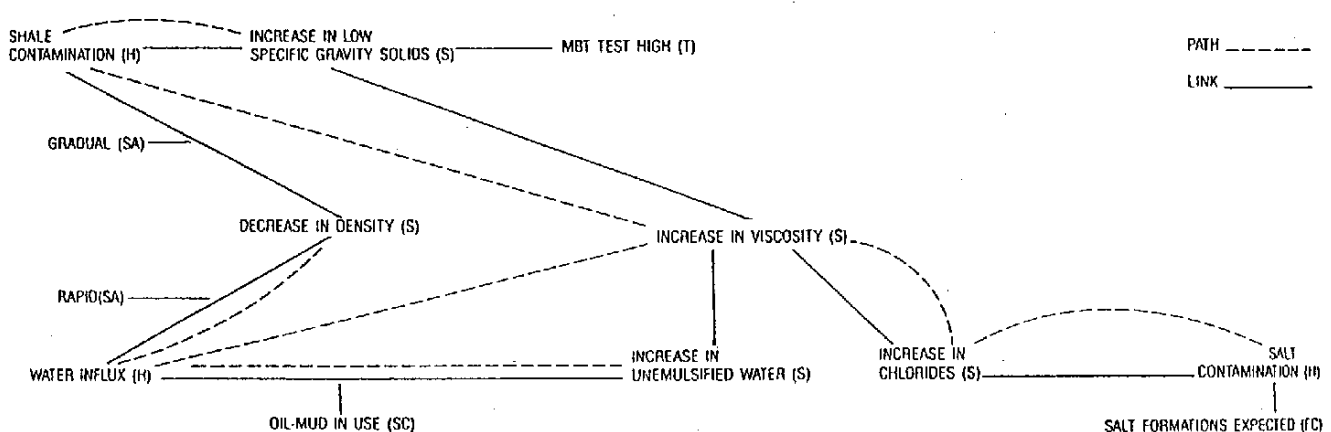


図 モデルの例

MOREのインタビュー戦略は、仮説・兆候・条件間の形式的な因果関係のみに基づいており、仮説や兆候が実際に何であるかには依存しない。そのため、MOREは診断問題に限れば、どのような対象のエキスパートシステム開発にも利用できるという大きな特徴をもつ。実際、MOREを使って開発されたエキスパートシステムは、コンピュータのディスク故障診断、コンピュータネットワーク診断、回路診断など多岐にわたっている。

2.2.2.4 MOLE

文 献	
表 題	MOLE: A Knowledge Acquisition Tool That Uses Its Head
著 者	Larry Eshelman and John McDermott
出 典	AAAI-86
キ ー ワ ー ド	自動インタビュー, 因果モデル, 診断, 診断型問題解決モデル
支 援 レ ベ ル	知識の変換, 知識の管理
対 象 分 野	診断型問題

MOLEは診断型問題を対象とするESの知識獲得を支援する, MOREの後継システムである. MOLEではシステムのもつワールドに関する推論機能を強化することによって, MOREでは不可能であったCF値の割当てを自動化することができる. これは初期知識ベースの改良の段階で, 仮説と兆候との間の静的な解析, ならびにテストケースを流して専門家の評価を受ける動的な解析の両者を繰り返す事でなされる.

MOLEのモデルは, 診断問題のための問題解決に適するように, 仮説の他に次の3種類の要素をもつ.

- 1) 兆候 : 仮説に起因するイベントや状態
- 2) 前条件 : 仮説より前あるいは同時に起き, 仮説の確からしさに影響する条件
- 3) 修飾条件 : 兆候や前条件を限定する背景条件

また, ワールドについて次の仮定をする.

- 1) 仮説の完全性 : すべての兆候は, いずれかの仮説で説明される
- 2) 仮説の排他性 : 説明は必要以上に重複させない (単一故障の仮定)

この仮定にもとづいて, 診断実行時の評価プロセスは双方向推論可能な推論機構により次の7つのステップで行なわれる.

- 1) 重要な兆候について質問
- 2) その兆候を説明するのに必要な仮説を活性化
- 3) 前条件や兆候により, 活性化された仮説を区別
- 4) 証拠から各仮説の支持値を計算
- 5) 兆候をよく説明する仮説, および競合する仮説に比べ大きな支持値をもつ仮説を受理
- 6) 説明されない兆候が残っていて, 関連する質問があれば3へ

7) 受理された仮説であっても兆候の説明に不要なものを棄却

MOL Eの評価プロセスで重要な点は、仮説で説明されるべきもの(covering)と、単に仮説に関連しているだけの物(circumstantial)とを区別することである。推論は単なるcf値の結合ではなく、covering evidenceが説明できるように進められる。

知識獲得は2つのフェーズで行なわれる。はじめのフェーズでは、初期知識ベース構築のため仮説と証拠とを収集する。次に反復洗練フェーズで、1) ノード型、2) リンク型、3) リンクの支持値を定めるために、静的解析と動的解析とを行う。

静的解析ではMOREと同様に知識ベースの構造から、1) 情報が不足しているネットの明確化、2) CF値の割りつけ、3) ネットの構造的欠陥の発見をおこなう。例えば兆候であるか前条件であるかについての情報が不足している場合には次のルールにより定められる。

IF E2がE1を引き起こし

 E1がE2をルールアウトする

THEN E1はE2に対する兆候であり、前条件ではない

動的解析ではテストケースに対する専門家の評価から、1) 不足知識の発見、2) cf値の変更、3) ネットの明確化を支援する。例えば、システムの診断結果と専門家の結果とが異なる場合には、現状のネットでcf値を変えるだけで専門家の結果に到達できるかどうかを調べる。可能であれば、新たな兆候や前条件の導入やcf値の変更などを行う。また、到達不可能な場合は、仮説-兆候の新たなリンクの導入など構造の変更によって正しい結論が得られるように助言する。

MOREの場合には、専門家が使いこなすためには経験を必要とした。MOL Eでは、専門家が不必要な質問に悩まされないようにするため、より柔軟な対話を可能にしている。MOL Eの現在のバージョンは圧延機械の診断とμVAXのチューニングに使われた。現在発電所診断システムの開発に適用している。また、診断以外の分野への適用(計算機コンポーネント選択)を検討している。

2.2.2.5 AQUA

文 献	
表 題	An Application of Qualitative Reasoning to Process Diagnosis: Automatic rule generation by qualitative simulation
著 者	Y. Ishida(京大)
出 典	IEEE 1988
キーワード	プロセス診断, 定性モデル, 定性推論, ルール生成, エキスパー トシステム, 知識獲得
支援レベル	知識の変換, 知識の管理
対象分野	プロセス診断

AQUAはプロセス診断分野の知識獲得ツールであり, MOLEなどのような対話的知識獲得に加えて, 高レベル知識(モデル)から低レベル知識(ルール)への知識変換手法により, 知識ベース構築の負荷を減らすことを目的としている。

システムはACQ(モデル構築)とQUA(定性アナライザ, ルール生成)からなる。ACQはモデルライブラリのテンプレートモデルを組み合わせることにより対象プロセスプラントの定性モデルを構成する。モデルの知識はプラント共通(コンポーネント)とプラント固有(構造)とに分けられている。QUAは定性的fault simulationにより診断ルールを生成し, また定性的安定解析などのグローバルなモデル解析も行なう。

プロセス診断の方法として, モデルベースの診断では, 定性モデル構築が困難, 計算時間がかかるといった問題点がある。一方, 症状ベースの診断では大規模化に伴い, デシジョンテーブルが膨大になる。このため, AQUAでは機能を限定した構築しやすい定性モデルから半自動的にルールを生成する方法をとった。

AQUAではプロセスの定性モデルを微分方程式や代数ではなく, 変化の方向, 位相遅れ, 効果がactivate/disableされる条件(クーラントが沸点を越えたかどうか等)などの定性的効果で考える。微分オペレータ等のプリミティブは持っていないが診断に必要な情報は十分表現でき, また構築が容易である。

AQUAのプリミティブは変数(温度, 圧力, 流速など)と, インタラクション(ある変数から他の変数へ及ぼす効果)の2種類である。モデルは, 変数およびプロセス故障に

対応するdisurbanceをノードとし，インタラクションをアークとする符号有向グラフで表される．

(例) $F_2 \text{ --- } (-, 0) \text{ --- } \Delta F$

関係 遅れ

このモデルを用いた定性シミュレーションを，次のアルゴリズムで行う．

- ・静的伝播：変数がdisturbされると，連結している他の変数に伝播する． n 次遅れで連結している場合は n 次微係数が変化する．
- ・動的伝播：過渡状態の変数があれば，変化させ時間を進める．その結果静的伝播が可能になれば静的伝播を行う．複数可能なら同時刻に変化させる．

この定性シミュレーションを利用して，モデルにプロセス故障を導入しモデル上で影響を伝播させることにより，症状と故障とを結びつける診断ルールを生成する．

この定性推論には，1)振動が増加，減少，周期的の区別がつかない，2)変数に対し矛盾するeffectが加えられた時に状態を定められない，3)時刻に関しいくつかのあい昧が残る．といった課題があるが，それぞれ回避する手段が提案されている．

定性モデルの構築は，バルブや熱交換器などのテンプレートモデルを予め用意し組み合わせることによって行なわれる．プラント共通のコンポーネントの知識と，プラント固有の構造（フローとコンポーネントのトポロジ，接続条件）の知識とを分離したため知識獲得が容易である．また，定性モデルの洗練過程では定性モデルのレベルでシミュレーションを行ない，専門家が位相遅れや符号モデル構造を対話的に修正する．

バッファタンクの例（右図）では，症状ベースの診断ルールの大半が生成できることが示された．専門家との討議で，この手法がプロセスダイナミクスの知識獲得に十分であることがわかった．しかし，専門家は定量的情報も扱うため，定性的なセマンティクスに定量性を取り込む必要がある．現状では定性方程式によって値をチェックすることしかしておらず今後の課題である．

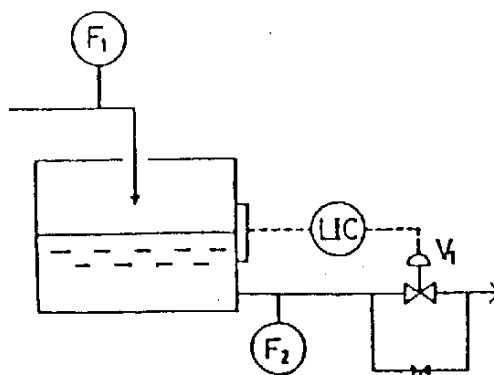


図 バッファタンクのダイアグラム

2.2.2.6 LEAP

システム名 LEAP

文 献

表題 LEAP: A Learning Apprentice for VLSI Design

著者 T.M.Mitchell, S.Mahadevan, and L.I.Steinberg

出典 Proc. of the 9th IJCAI, Los Angeles, CA, pp.573-580, Aug.
1985.

キーワード 知識工学, 自動インタビュー, エキスパートシステム, 知識獲得
LSI設計

支援レベル 知識の投入段階での知識変換

対象分野 LSI設計問題

(1) 対象とする分野とタスクの特徴

適用分野はドメインの知識が構造化できている場合(例: 論理回路設計)を想定している。また、知識変換方法としては専門家の問題解決の実例をもとに、より良い結果を得るルールに関して条件等を一般化していく手法を適用している。

このシステムの特長は(a)会話処理で、システムで未だ内蔵していない問題解決機能を獲得できること、(b)一般化の手法として解析的な手法を適用したこと、(c)制御知識とドメイン知識とを明確に分離したことである。

なお、ここで適用した知識の変換の手法はEBG(Explanation-Based Generalization)と命名され、後日別文献[Mitchell 86]で一般化が論じられている。

(2) システム概要

(i) 着目点すべき点

アプローチ法として、(a)新知識の獲得タイミングを専門家により、より良い構成/代替構成が指定される時としたこと(通常はシステムからのガイダンスにより設計作業を行っている)、(b)知識の変換対象をより良い構成をガイダンスするためのルールを解析的に一般化していく手法を適用していることである。

以下にルールの一例を示す。

IF the required function is to convert a parallel signal
to a serial signal,

THEN one possible implementation is to use shift register.

<もし> 要求機能が直列信号を並列信号に変換する機能で <あれば>
可能性のある改良法はシフトレジスタを使用すること <である>

(ii) 具体的な実現手法

大きく以下の2ステップで実現する.

ステップ1: 専門家による正しい構成を入力

ステップ2: 新しい構成をガイダンスするためのルールを一般化

なお, 一般化で着目する情報としては, 「構成」の記述, 入力信号の特性,
専門家により入力された「構成」等である.

(iii) ステップ2での一般化の手順

より良い構成をガイダンスするためのルールを対象とした一般化の手順をルールのLHSとRHSについて示す.

A) ルールのLHSの一般化法

まず, 回路の各基本要素の動作条件から導かれる制約条件を, 回路網全体の出力と入力へ伝播する.

次に, これらを新しく生成するルールのLHSとする.

B) ルールのRHSの一般化法

ステップ1: ルールのRHSから該構成の機能仕様を得る

ステップ2: 回路機能の正さを検証する

ステップ3: 一般化した構成の機能仕様を導く (back-propagation)

ステップ4: 一般化した構成の基本(original)仕様を導く

ステップ5: 新しいルールを構成する

(3) その他

本手法は一例のみに着目して一般化するため, 誤った知識の投入に弱い.

[参考文献]

[Mitchell 86] T.M.Mitchell, R.M.Keller, and S.T.Kedar-Cabelli, "Explanation-Based Generalization: A Unifying View", Machine Learning, vol. 1, no. 1, pp.47-80, 1986.

[Mitchell 86'] T.M.Mitchell, L.I.Steinberg and J.S.Shuiman, "A Knowledge-Based Approach to Design", IEEE Trans. on Pattern Analysis and Machine

VERIFICATION

```

(NOT (OR (NOT (OR (VALUE Input1(i))
                  (VALUE Input2(i))))
  (NOT (OR (VALUE Input3(i))
            (VALUE Input4(i))))))
  ↓
De-Morgan
  ↓
(AND (NOT (NOT (OR (VALUE Input1(i))
                  (VALUE Input2(i))))
  (NOT (NOT (OR (VALUE Input3(i))
                (VALUE Input4(i))))))
  ↓
Remove-Double-Negation
  ↓
(AND (OR (VALUE Input1(i))
        (VALUE Input2(i)))
  (NOT (NOT (OR (VALUE Input3(i))
                (VALUE Input4(i))))))
  ↓
Remove-Double-Negation
  ↓
(AND (OR (VALUE Input1(i))
        (VALUE Input2(i)))
  (OR (VALUE Input3(i))
      (VALUE Input4(i))))

```

COMPUTING THE
GENERALIZED COMPOSED SPECIFICATION

```

(AND (OR (VALUE Input1(i))
        (VALUE Input2(i)))
  (OR (VALUE Input3(i))
      (VALUE Input4(i))))
  ↓
Remove-Double-Neg
  ↓
(AND (OR (VALUE Input1(i))
        (VALUE Input2(i)))
  (NOT (NOT <bool-fn1>)))
  ↓
Remove-Double-Neg
  ↓
(AND (NOT (NOT <bool-fn2>))
  (NOT (NOT <bool-fn1>)))
  ↓
De-Morgan
  ↓
(NOT (OR (NOT <bool-fn2>
            (NOT <bool-fn1>)))

```

GENERALIZED SPECIFICATIONS OF SUBMODULES

```

P1': (EQUALS (VALUE X(i)) (NOT <bool-fn2>))
P2': (EQUALS (VALUE Y(i)) (NOT <bool-fn1>))
P3': (EQUALS (VALUE <out>(i))
            (NOT (OR X(i) Y(i))))

```

2.2.2.7 A r g o

システム名 A r g o

文 献

表題 Argo : A System for Design by Analogy

著者 Michael N. Huhus and Ramon D. Acosta

Microelectronics and Computer Technology Corp., Austin, TX

出典 CH2552-8 pp.146-151, 1988.

キーワード 知識工学, エキスパートシステム, 知識獲得, L S I 設計

支援レベル 知識の変換

対象分野 L S I 設計問題

(1) 対象とする分野とタスクの特徴

適用分野は論理回路設計を例としている。また、知識変換方法としては専門家がシステムの利用の際に使用した問題解決の経験的知識をルール依存関係のグラフとして表現し、これから、新たなマクロルールを生成し、次回のシステム利用からこの新しいマクロルールを適用し、処理時間の短縮をねらったものである。

なお、ここで適用した知識の変換の手法はE B L (Explanation-Based Learning)である。

(2) システム概要

(a)着目点すべき点

従来の知識変換手法では、設計問題を例とすると、何度実行しても同じ時間がかかるといった問題があるが、この手法では処理時間を短くできる可能性がある。

アプローチ法としては、まず、(i) どの知識がどれがよく利用されているかを見つけるために、J T M S (justification-based truth-maintenance)を、また、(ii) どうやって今回の経験を次回の問題に適用するか、システムを利用した際に使用した知識の抽象化(abstraction)を適用している。

(b)具体的な実現手法

ステップ1：専門家により本システムを利用して設計を実施する。

ステップ2：各ルールの動作履歴を蓄積しておく。

ステップ3：ルールの依存関係のグラフ作成（但し、グラフのリーフは取り除く）

ステップ4 : グラフからマクロルールを生成する.

ステップ5 : マクロルールを知識ベースに登録し, 次回のシステム利用時にこのマクロルールを適用する.

(c)結果例

実際に本システムを運用した場合の結果を表2.2.2.7-1に示す. また, 図2.2.2.7-1に動作フローを, 図2.2.2.7-2にルール依存グラフの例を示す.

(3) その他

この手法は, 知識ベースが検索型で構成される場合有効であるが, 処理時間がかかり大規模知識ベースへの適用には問題がある.

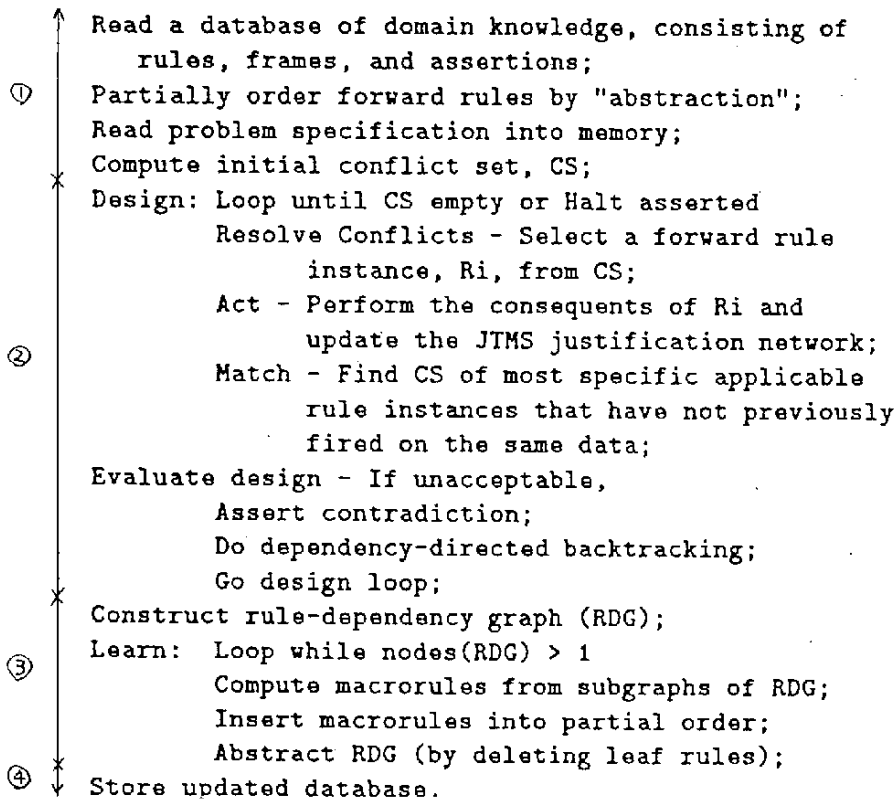


Figure 1: Argo Control Strategy

図2.2.2.7-1 Argoの動作フロー

表2.2.2.7-1 システムの運用結果

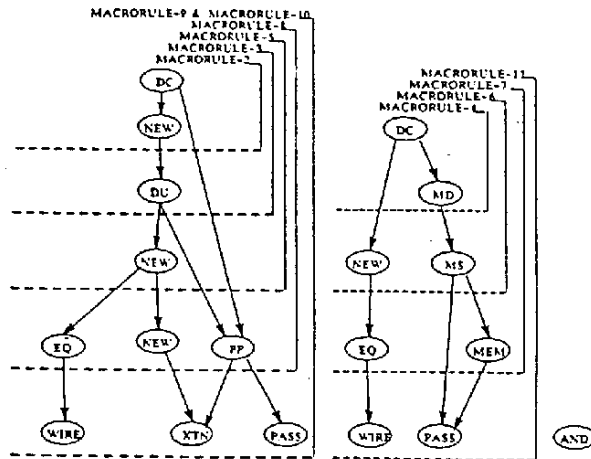
適用対象例はCAM (内容検索メモリ) で, 各回路の構成上の特長を以下に示す.

- 回路2はマクロルール#2, 10で構成される回路
- 回路3, 6は回路構成同一で, 定数のみ異なる回路
- 回路4はマクロルール#1, 10で構成される回路
- 回路5はマクロルール#3, 10で構成される回路
- 回路7はマクロルール#10のみで構成される回路

Table 1: Effects of Learning on VLSI Design

Design	Before Learning/After Learning		
	Time (seconds)	Rules Fired	Design Quality (transistors)
CAM-cell	66.5/59.6	17/3	30/20
Circuit 2	68.8/51.6	17/7	30/26
Circuit 3	66.1/34.9	17/3	30/20
Circuit 4	48.7/33.3	11/5	25/22
Circuit 5	61.7/45.6	15/5	32/17
Circuit 6	64.3/43.2	17/3	30/20
Circuit 7	23.3/19.6	10/1	16/9

Note: Timings were made on a Symbolics 3600.



- DC: Decompose conditional-signal-assignment statements
 NEW: Construct new signal-assignment statements from decomposed statements
 DU: Decompose unconditional-signal-assignment statements
 EQ: Transform a statement containing an equality into a simpler statement
 WIRE: Instantiate a connection between two components
 PP: Transform a set of signal assignments to represent a cascade of pass-transistor nets
 XTN: Instantiate an exclusive-OR pass-transistor network
 PASS: Instantiate a pass transistor
 MD: Decompose an entity into ones containing memory and combinational logic
 MS: Complete the specification of an entity containing memory statements
 MEM: Instantiate an inverter loop for a one-bit memory
 AND: Instantiate an AND gate

凡 ルール : ○ (グラフのノード) 規 利用されたルール: 19
 例 ルールの依存関係: → (" リーフ) 模 生成したマクロルール: 10

図2.2.2.7-2 ルール依存関係のグラフとマクロルールの例 (CAMの場合)

[参考文献]

[Huhus 87] M. N. Huhus and R. D. Acosta, "Argo : An Analogical Reasoning System for Solving Design Problems, MCC Technical Report No. AI/CAD-092-87, Micro-electronics and Computer Technology Corp. 1987.

[Acosta 86] R. D. Acosta, M. N. Huhus, and S. liuh "Analogical Reasoning for Digital System Synthesis", Proc. of the IEEE Int. Conf. on Computer-Aided Design, Santa Clara, CA, pp.173-176, Nov.1986.

2. 2. 2. 8 SEEK/SEEK2

システム名 SEEK/SEEK2

文 献

SEEK:

表題 Using Empirical Analysis to Refine Expert System Knowledge Bases

著者 Peter Politakis, Sholom M. Weiss

出典 Artificial Intelligence 22, 23-48 (1984)

SEEK2:

表題 SEEK2: A GENERALIZED APPROACH TO AUTOMATIC KNOWLEDGE BASE REFINEMENT

著者 Allen Ginsberg, Sholom M. Weiss and Peter Politakis

出典 in: Proceedings IJCAI-85, 367-374 (1985)

表題 Automatic knowledge Base Refinement for Classification Systems

著者 Allen Ginsberg, Sholom M. Weiss and Peter Politakis

出典 Artificial Intelligence 35, 197-226 (1988)

支援レベル 知識ベースの洗練化 (refinement)

対象分野 リウマチの医療診断問題

タスクの特徴

リウマチは7種類の疾病に分類されるが、それらの進行が非典型的であるため、ある症状に対して原因となる疾病の決定が困難である。実際、専門家でさえいくつかの診断基準に対して疑問を持っているほどである。そのため、リウマチ診断の知識ベースを構築する際には、臨床事例を用いたモデルのテストが重要な設計戦略になる。

システム概要

SEEK:

SEEKは、知識ベース（ルールベースと仮定する）全体の性能向上（洗練化）を図るシステムである。知識ベースの性能とは、その知識ベースを用いた推論結果と実際の専門家による推論結果との一致の度合のことであり、両者の推論結果がよく一致しているほどその知識ベースの性能は高いと言う。SEEKによる知識ベースの洗練化の手順を以下に述べる。

SEEKは、まず現在の知識ベースの性能をケースデータ（実際の診断例）を用いて解析する。図2.2.2.8-1はその解析結果である。例えばMixed connective tissue disease（以下MCTDと略す）では、正しく診断されたのは33ケース中9ケース（27%）であることがわかる。False positives が0というのは、MCTDでないのにMCTDと診断されたケース数が0であることを表す。

この解析結果において、知識ベースによる推論結果とケースデータとが一致しない例（誤診例）に着目する。ここでは、図2.2.2.8-1で一番正答率の低いMCTDについて考える。SEEKは、まず、その誤診を導いたルールの連鎖を解析し、各ルールを一般化あるいは特殊化することによって誤診を防ぐことができるかどうか検討する。ここで、ルールの一般化とはルールの条件を緩めることであり、特殊化とはルールの条件を強めることである。従って、一般化されたルールは発火されやすくなり、特殊化されたルールは発火されにくくなる。次に、SEEKは、どのルールを修正するのが効果的かを解析してユーザ（ここではリウマチの専門家）に修正案の候補を示す（図2.2.2.8-2）。図2.2.2.8-2において(1)～(5)で示されているのがルールの修正案であり、番号の若いものほどより多くの誤診を防ぐことができる。ユーザは、この解析結果を参考にしてルールの修正を行う。

次に、SEEKは、このルールの修正によって知識ベースの性能がどの程度向上したかをユーザに示す（図2.2.2.8-3）。図2.2.2.8-3を見ると、MCTDの正答率が27%から52%に向上しているばかりでなく、他の疾病のFalse positives の数も減少しており、知識ベースの性能の向上が認められる。以下、誤診を含む他の疾病に対し同様の処理を繰り返す。

なお、SEEKは、ルールの記述形式がcriteria tableという表形式に限定されて

いる、あるいは、知識の洗練化のためのヒューリスティクスをユーザが容易に変更できない等の限界を持つ。

SEEK 2 :

SEEK 2は、SEEKの拡張版として開発された。SEEKからの主な拡張点を以下に述べる。

(1) automatic pilot の導入

automatic pilot は、SEEKにおける専門家の意思決定（例えば、どの疾病から洗練化を行うか、あるいは、ルールの修正案のうちどれを選択するか等）を代行する機能である。この導入によって、システムが自動的に知識ベースの洗練化を行うことができるようになった。

(2) metalanguage の導入

洗練化のヒューリスティクスを記述する言語として metalanguage を導入し、それをユーザに解放した。これによって、ユーザがヒューリスティクスを自由に変更できるようになった。また、ヒューリスティクスの記述言語は、知識ベースのルールの記述形式と関連があるため、metalanguage の汎用性によって、ルールの記述形式が表形式に限定されなくなった。

			False positives
Mixed connective tissue disease	9/33	(27%)	0
Rheumatoid arthritis	42/42	(100%)	9
Systemic lupus erythematosus	12/18	(67%)	4
Progressive systemic sclerosis	22/23	(96%)	5
Polymyositis	4/ 5	(80%)	1
Total	89/121	(74%)	

図2.2.2. 8-1 知識ベースの性能解析結果

24 cases in which the expert's conclusion MCTD does not match the model's conclusion:

1,4,11,12,14,15,42,47,49,57,60,67,
71,75,78,80,84,93,99,100,104,105,107,130.

Proposed experiments for mixed connective tissue disease

- (1) Decrease the number of majors in Rule 56.
- (2) Delete the requirement component in Rule 55.
- (3) Delete the requirement component in Rule 54.
- (4) Decrease the number of minors in Rule 57.
- (5) Delete the requirement component in Rule 58.

図2.2.2. 8-2 ルールの修正案の提示

	Before		False positives	After		False positives
MCTD	9/ 33	(27%)	0	17/ 33	(52%)	0
Others	80/ 88	(91%)		80/ 88	(91%)	
Total	89/121	(74%)		97/121	(80%)	
Others						
RA	42/ 42	(100%)	9	42/ 42	(100%)	8
SLE	12/ 18	(67%)	4	12/ 18	(67%)	3
PSS	22/ 23	(96%)	5	22/ 23	(96%)	3
PM	4/ 5	(80%)	1	4/ 5	(80%)	1

図2.2.2. 8-3 ルールの修正前後の推論結果の比較

今後の課題

- ・洗練化する問題の一般化

現在は、プロダクションルールの分類型システムに限られている。

- ・metalanguageの拡張
- ・ドメインに固有のヒューリスティクスの導入

決して修正されないルールセット，最初に修正されるべきルールセットといった知識。

- ・大きな知識ベースでの実験

最新版：26 diagnoses, 440 中間仮説, 880 観測, 1000以上のルール

- ・並列処理の導入

最新版では，ケースデータ数が300もあるため，CPUタイムが2～3日かかる。

そのため，高速化が必要。

- ・今以上により洗練化戦略の究明
- ・自動学習ヒューリスティクスの導入

ドメイン知識や経験的解析を用いて修正方法の学習を行う。

2.2.2.9 EPSILON/EM

システム名	EPSILON/EM
文献	
表題	EXPERT MODEL for Knowledge Acquisition
著者	Hirokazu Taki, Kazuhiro Tsubaki and Yasuo Iwashita
出典	The Proceeding of IEEE Expert Systems in Government Conference, 1987
キーワード	知識獲得、専門家モデル、エキスパートシステム、人工知能、インタビュー方法
支援レベル	問題解決モデルの作成支援、問題解決の詳細知識の抽出
対象分野	診断問題、簡単な設計問題(生成検査型)

(1) 概要

EPSILON は、診断型エキスパートシステムの知識ベースの分析から得られた知識表現である専門家モデルに基づく知識獲得支援システムである。知識獲得過程は、次の手順で行われる。

- [1] オペレーションから構成される問題解決のモデル(専門家モデル)を獲得する。
- [2] オペレーションの内容を具体化するための知識獲得を行う。
この際にオペレーションのタイプに対応した効率的なインタビューが行われる。
- [3] 各オペレーションで処理される要素(データ)が獲得される。
- [4] 専門家モデル推論エンジンで専門家モデルレベルでの実行評価が行われる。

(2) 専門家モデル

専門家モデルは、オペレーションで構成される知識表現である。オペレーションは、専門家の作業単位を表す。オペレーションを分解していくと、次の7つの機能タイプのいずれかになる。この7つのタイプを類型オペレーションと呼ぶ。これらは、選択、分類、組み合わせ、交換、順位付け、入力と出力である。各オペレーションは、エバリュエータ(オペレーションの機能を表す知識)、入力要素グループ(オペレーションで処理される要素の集合)と出力要素グループ(オペレーションで処理された要素の集合)で構成される。

メタレベル:	メタスクリプト(オペレーションの推論制御)
オペレーションレベル:	オペレーションの集合、階層情報 オペレーションのタイプ(類型オペレーションタイプ) オペレーションの実行優先順位(プリポスト関係) オペレーションの内容(エバリュエータ) 要素の集合とオペレーションの関係(データフロー)
要素レベル:	要素、要素の属性と属性値 要素の集合(要素グループ)

(3) インタビュー戦略

EPSILON の知識獲得戦略は、プリポスト法と呼ばれている。その過程を次に示す。

- [1] 知識獲得の出発点となるオペレーションの抽出
- [2] プリポストオペレーションの抽出
オペレーションの前/後に行うオペレーションを質問する。
- [3] オペレーションタイプの決定
- [4] オペレーションの内容(エバリュエータ)の抽出
オペレーションタイプごとに、エバリュエータを実現するのに必要な最低限度の知識を獲得する。
- [5] 要素グループの決定
- [6] 要素グループとオペレーションの関係(データフロー)の抽出
- [7] 要素の属性と属性値の決定
- [8] オペレーションの推論制御方法の決定

[1]から[3]までは、専門家モデルのモデル構築を行っている。[4]から[8]までは、専門家モデルの詳細化を行っている。

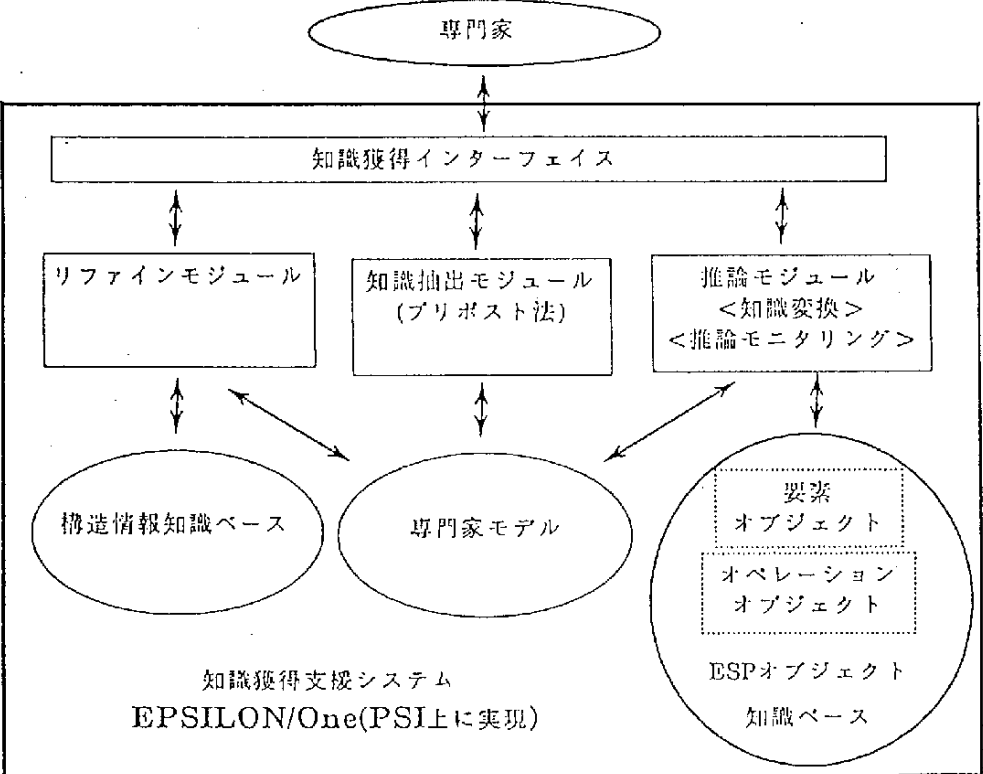
(4) EPSILON システム

EPSILON/EM のサブセットシステムとして、EPSILON/One が PSI-2上に実現されている。(ただし、非階層専門家モデルであり、推論制御が1種類に限られており、エバリュエータ内容の動的変更の不可である。)

EPSILON/One は、FGCS'88(Fifth Generation Computer System 1988)において、デモンストレーションが行われた。

(5) EPSILONの今後の課題

オペレーションの論理的矛盾の検出によるリファイン、帰納推論を用いたオペレーション内容(エバリュエータ)の推定、並列推論向き知識の獲得が現在検討されている。また、CTAS(Classification Task Acquisition System: ICOTで開発された分類問題向きの知識獲得支援システムである、ETSと同様に Rating Grid型の知識を獲得する)の機能を要素獲得に利用することも検討中である。

題 名	知識獲得支援システム EPSILON/One
目 的	専門家から対話的に知識を獲得する知識ベース構築支援システムを実現する。また、上記システムのための知識獲得向きオブジェクト指向知識表現及び、知識抽出・リファイン方法を研究する。
概要 及び 特徴	知識ベースの構築は、非常に困難な作業であり、知識獲得のボトルネックと呼ばれている。これは、知識獲得のための知識表現や知識抽出・リファイン方法が整備されていないためである。 知識獲得支援システム EPSILON/One は、このボトルネックを解消するために、以下の機能を実現している。 (1) 専門家の問題解決モデル(専門家モデル)の構築と具体化機能 ジェネリックオペレーションと呼ぶ機能分類されたルールにオブジェクト指向を導入した知識表現および、オペレーション単位で知識抽出を行うプリポスト法 (2) 知識のリファイン機能 異なる知識表現を用いたリファイン法 (3) 知識の変換機能 獲得した知識の ESP のオブジェクト表現への変換 (4) 知識ベースの評価機能 知識ベースの動的評価
構 成	 <pre> graph TD Expert([専門家]) <--> Interface[知識獲得インターフェイス] subgraph System [知識獲得支援システム EPSILON/One(PSI上に実現)] direction TB Interface <--> Refine[リファインモジュール] Interface <--> Extract[知識抽出モジュール
(プリポスト法)] Interface <--> Infer[推論モジュール
<知識変換>
<推論モニタリング>] Refine <--> KB1([構造情報知識ベース]) Extract <--> ExpertModel([専門家モデル]) Infer <--> KB2((ESPオブジェクト知識ベース)) KB1 <--> ExpertModel ExpertModel <--> KB2 end KB2 --- Elements[要素
オブジェクト
オペレーション
オブジェクト] </pre> 知識獲得支援システム EPSILON/One(PSI上に実現)

2. 2. 2. 10 ETSとAQUINAS

システム名 ETSとAQUINAS

文 献

表題 Expertise transfer and complex problems: using
AQUINAS as a knowledge-acquisition workbench for
knowledge-based systems

著者 JOHN H. BOOSE AND JEFFREY M. BRADSHAW

出典 International Journal of Man-Machine Studies

表題 Personal construct theory and the transfer of human
expertise.

著者 JOHN H. BOOSE

出典 Proceedings of the National Conference on Artificial
Intelligence

キーワード 人工知能, 知識工学, 知識獲得, パーソナルコンス
トラクトセオリ

支援レベル 知識の収集, 変換, 管理

対象分野 診断問題

(1) 概要

ETS, AQUINAS は共に, BOEING のAIセンターで開発されたシステムで, 心理学の研究
成果を応用した知識獲得システムである点に特徴がある。はじめにETS が開発され,
その機能を強化・拡張したものがAQUINAS である。したがって, ETSの機能はAQUINAS
のそれにすべて含まれている。以下ではAQUINAS について紹介する。

(2) 知識獲得の支援レベル

AQUINAS がサポートする知識獲得に関するタスクは以下のとおりである。

・差異の抽出 (Elicit distinction)

- ・問題の分解 (Decompose problem)
- ・知識ベースのテスト (Test the knowledge)
- ・異なったデータタイプの統合 (Integrate diverse data types)
- ・知識ベースの自動的な拡張 (Automatic expansion and refinement of the knowledge base)
- ・複数の知識ソースの利用 (Use of multiple sources of knowledge)
- ・知識獲得プロセスのガイダンス (Provide knowledge acquisition process guidance)

したがって、知識獲得のフェーズを知識の収集、変換、管理の三つに区分したとすると、AQUINASは全てのフェーズに対して有効なツールと言うことができるが、中でもAQUINASの最大の特徴は、Personal Construct Theoryと呼ばれる心理学の手法を応用した知識の整理、収集法にある。

(3) 対象とする分野とタスクの特徴

AQUINAS が用いているPersonal Construct Theory は知識獲得の中心的概念を「差異の抽出」としている。すなわち、あらかじめ設定されている解の候補の中から特定の問題に関して最適な条件を満足する解を見出すためには、その解が他の解の候補と比較してどんな観点でどう異なる性質を持つかを認識すれば良く、そのために必要な作業が「差異の抽出」になるという訳である。したがって、AQUINASが対象としている問題は分析型・診断型の問題ということになる。

(4) システム概要

AQUINAS の構成は以下のとおりである。

Dialog manager						
ETS Repertory grid tools	Hierarchical structure tools	Uncertainty tools	Internal reasoning engine	Multiple scale type tools	Induction tools	Multiple expert tools
Object-oriented DBMS						
CommonLoops/CommonLisp						

図 2.2.2.10-1 AQUINAS の構成

ここでは、AQUINASの中心的概念である「差異の抽出」のメカニズムについて詳述する。以下では、特定の問題分野に対して最適なプログラミング言語を選択する問題

を例にとり,Personal Construct Theoryの原理とAQUINAS の動作を示す。

まず, 解の候補, すなわち選択されるプログラミング言語の種類として,ADA,LISP, PROLOG,COBOL,FORTRAN(実際にはこの他にもう一つ,Ideal language が加わる) を考える。これらの言語の間の差異を記述するものを特徴(traits)と呼び, ここでは

- ・Attitude (Symbolicな処理かNumerical な処理か)
- ・Availability (Widely availableかNot widely availableか)
- ・Application area (Scientific useかBusiness useか)

の三つがユーザとの対話に基づいて抽出される。以下はそのセッションの様子である。

```
Think of an important attribute that two of ADA,LISP and
PROLOG share, but that the other one does not.What is that
attribute?
AQU** SYMBOLIC
What is that attribute's opposite as it applies in this case?
AQU** NUMERIC
What is the name of a scale or concept that describe
SYMBOLIC/NUMERIC?
AQU** ATTITUDE
AQU** 以降がユーザ入力を示す
```

図2.2.2.10-2 セッション例1

こうして解の候補とそれらの特徴づける属性とを定義した後, 個々の解の候補の得点づけをし, Rating grid と呼ばれる領域に書き込む。これによって解の候補の差異が記述できたことになる。得点づけは以下のセッションで行われる。

```
Please rate these things on a scale of 5 to 1, where 1 means
more like SYMBOLIC and 5 means more like NUMERIC.If both seem
to apply, enter a B(oth). . . . .
SYMBOLIC(1) NUMERIC(5)
ADA** 5
LISP** 1
PROLOG** 1
COBOL** 5
FORTRAN** 5
IDEAL-K-AQUISITION-LANGUAGE** 1
```

図2.2.2.10-3 セッション例2

このようにして解の候補と特徴の間の関連が次図のようにまとめられる。

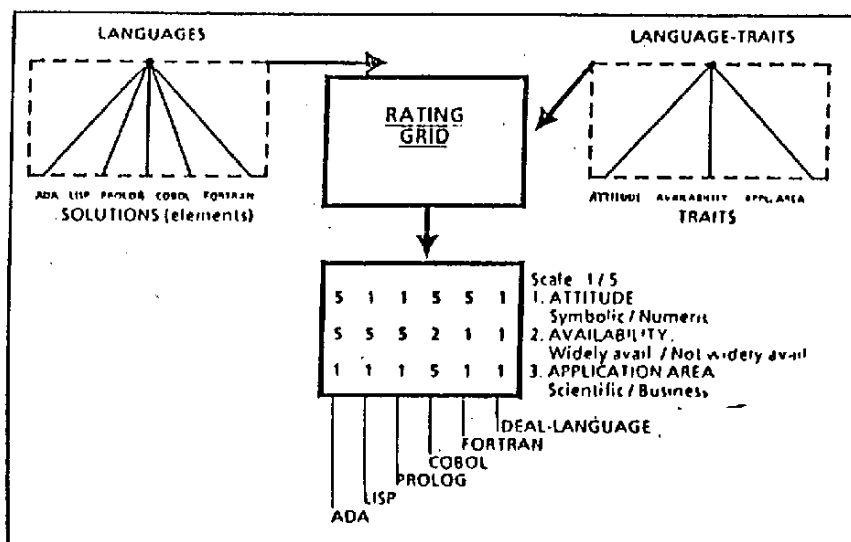


図 2.2.2.10-4 Rating Grid

AQUINAS の前身であるETS ではRating grid は単一であったが、AQUINASでは複数のRating grid が扱えるようになっており、先の解の候補と特徴に加えて、ケースと専門家の4つの階層を用い、これらの階層の対応する箇所に対してそれぞれRating grid を定義できるようにしている。

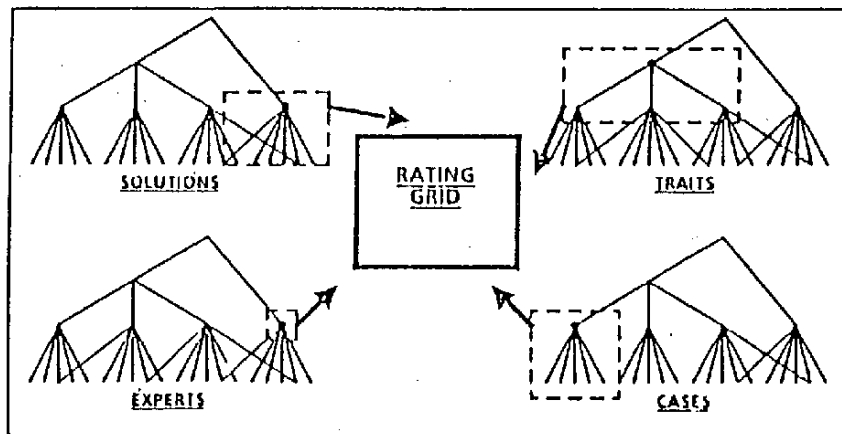


図 2.2.2.10-5 問題の分解

以上のようなプロセスに基づいて、専門家の知識を収集・整理し、最終的にはKEE, KS-300/EMYCIN, LOOPS, M.1, OPS5, S.1など多くのツールに適用できるルールが作成される。

これまでAQUINAS によって30を超える知識ベースが構築され、特に分析型問題に有効であることが確認できた。しかし、すべての問題にこうした階層的なモデルが適用できるとは限らず、今後は合成型問題をも扱えるような方向で機能拡張が計画されている。

2. 2. 2. 11 COLAB

システム名 COLAB (argnoter, cognoter)

文 献

表題 BEYOND THE CHALKBOARD:
COMPUTER SUPPORT FOR
COLLABORATION AND PROBLEM
SOLVING IN MEETINGS

著者 MARK SIEFIK, GREGG FOSTER, DANIEL G. BOBROW, KENNETH KAHN,
STAN LANNING, and LUCY SUCHMAN

出典 Communications of the ACM, January 1987 Volume 30 Number 1

キーワード Collaboration, database

支援レベルブレインストーミング (白紙状態から知識を収集する)

対象分野特に限らない

(1) 要旨: Colab とは、2～6 人用の会議室で
あり、各人に一台のワークステーションが装備され、
ワークステーションは、イーサネットに結ばれている。

本論文は、『Cognoter』、『Argnoter』と称す
る2つのColab toolについて紹介している。

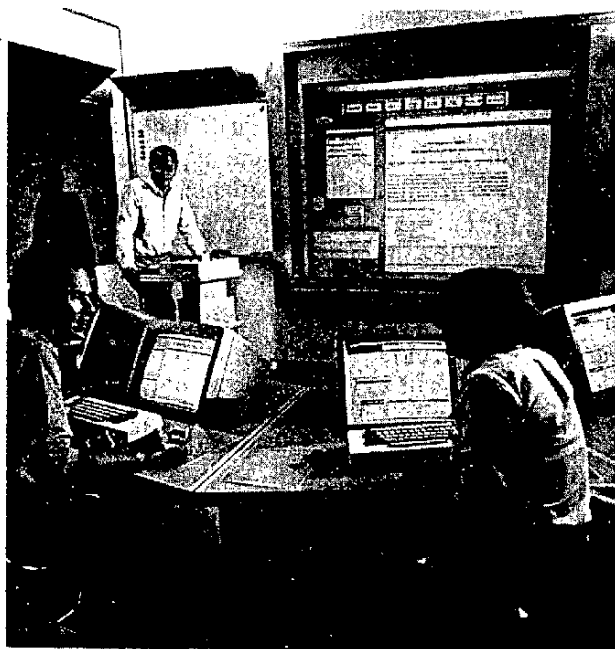
Cognoter: 全く白紙の状態からpresentation
の草案を作るまでをサポートする

Argnoter: 会議参加者の提案の中から適切な
提案を選び出す作業をサポートする

(2) WYSIWIS(What You See Is What I See) 思想

会議参加者全員に同じ情報を提供することにより全員の認識を同じにしようという考え
方である。これは、文書をプリンターに打ち出す最中その文書をディスプレイ上に表示しておこ
うというWYSIWYG(What You See Is What You Get) なる考え方にならって作られた思想で
ある。

従来のWYSIWIS(What You See Is What I See) 思想に基づいて設計された会議用の



windowは、会議参加者ならだれでも参照・書き込みが可能であった。このため、複数の人が同時に書き込もうとしてカーソルを奪い合うことが生ずる。

そこで、今回のColab toolはwindowを

public window

: 会議参加者ならだれでも見て、書き込むことができるwindow

private window

: 会議参加者ならだれでも見ることはできるが、

特定の人でないと書き込めないwindow

に分けてみた。

(3) Colab tool

(a) Cognoter

全く白紙の状態から、

1. brainstorming

2. organizing

3. evaluation

を通して、論文等のpresentationの草案作製のサポートをする。

(i) brainstorming

(private window)

会議参加者に発表項目をsupporting text上に記入してもらう。ただし、他人が記入した項目は、削除できない。(削除は、他人の意見を批判することと同じであり、brainstormingの趣旨に反するため。)

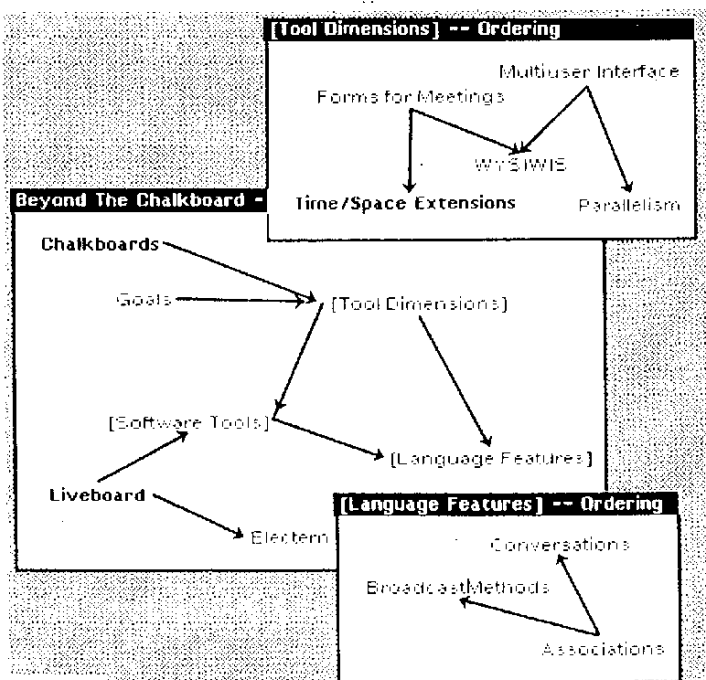
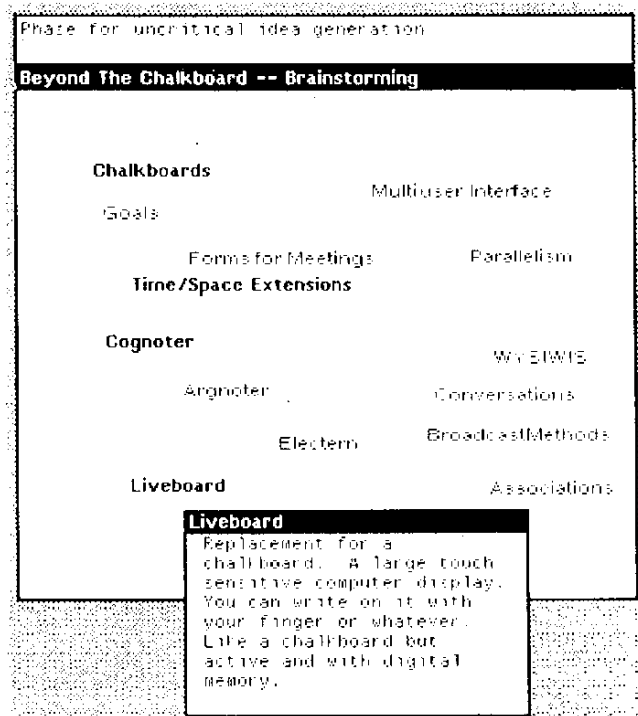
(ii) organizing

(public window)

brainstormingの段階で集められた項目の

- ・適切な位置に移動する。
- ・presentation順序を決める。

(任意の2項目の前後関係を矢印で結ぶ



順序はtransitive。)

・いくつかのグループに分ける。

(organizingの段階でも削除は許されない)

(iii) evaluation

(public window)

organizingの段階で構成されたアイデアを

- ・再構成する（推敲する）。
- ・補填する。
- ・余計な項目等を削除する。

削除は、最終段階(evaluation)においてのみ実行可能。これは、brainstormingやorganizingの段階で削除を許すと、議論に参加する機会を奪ったり、議論を脇道にそらすことになりかねないからである。

(iv) Cognoterの特徴

- ・思考過程を細分化して、性質の異なる3つ段階に整理した。
- ・incremental に項目の順序が決められる。

(b) Argnoter

会議参加者の提案の発表及び評価のために用いられるtoolであり、現在開発中である。

(本論文では、1セットのみの説明)

(i) proposing

提案を出す段階。いくつかのwindowを用いて提案を表示する。

windowはCognoter同様2種類用意する。

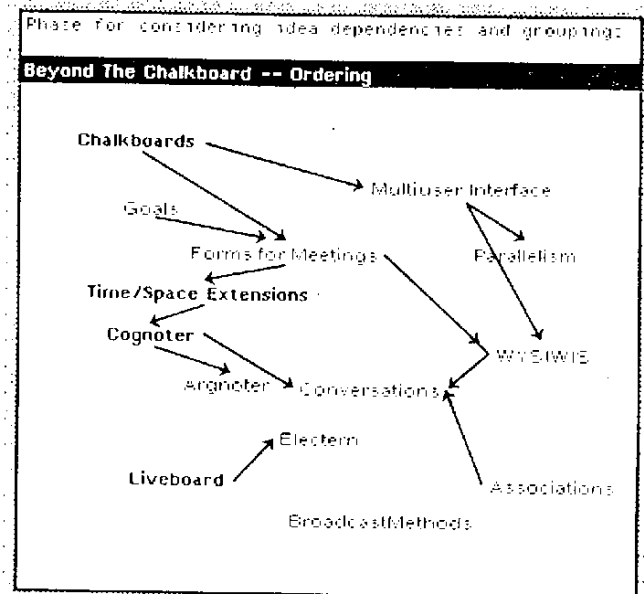
public window: 『WYSIWIS』に基づいて設計され、会議参加者全員が操作できる。

private window: 個人のみ参照・記入が可能。

(ii) arguing

各議員から出された提案を選択・棄却するための理由づけをする段階。

- ・『賛成』ないしは『反対』の評価をしする。
- ・判断理由を付記する(例: 『費用がかかり過ぎる』、『6ヶ月では無理』)。



- ・判断理由を追加したり、提案を変更してもよい。

(iii) evaluation

- ・belief setを設定して、判断理由の『議論の観点』を特徴付ける。
(例: liberal - conservative, marketing - development)
- ・belief setを基に判断理由の理論的關係付けを行う。
(Argnoterは、spreadsheet と同様な働きをするようになる。)
- ・複数の評価基準を設定し、重要度からランク付けする。
- ・『議論の観点』と『評価基準』を明確にした後で、お互いの意見の食違いを調整する。

(4) インプリメント上の問題

(a) Colab toolを構築する上での基本的前提条件

1. 会議参加者には各々一台のワークステーション が与えられ、各ワークステーション は、LANで結ばれていること。
 2. 各々のワークステーション では、同一のソフトウェアが稼動できること。
- 複数の計算機に分散するプログラム は、作成・デバッグが困難であるため、Colab tool構築用の環境を独自に作った。

(b) Colab toolの目標

- ・情報獲得までの所要時間は、できるだけ短く。
(複雑な情報を処理できるためにも、平均検索時間は数ミリ秒が要求される)
- ・情報修正に必要な時間もできるだけ短く。
(使用者に“遅い”と思わせないためにも、所要時間は数秒程度)
- ・データベースの矛盾は、速く解消されること。
- ・データベースは、不慮の故障および操作手違いの影響を受けないこと。

(5) データベースのメンテナンス方法について

・Centralized Model

1. 中央に1つのデータベースを持つ。
2. 会議参加者が並列に別個のデータの修正をすることは許されるが、同一のデータを修正する場合は、database transaction mechanismが調整を行う。

問題点:

database transaction mechanismを用いるとデータの検索速度が落ちる。

・ Centralized-Lock Model

1. 各ワークステーションは、データベースのPC-を持っている。
2. ディスプレイ上に表示するデータは、キャッシュに蓄えておく。
3. centralized lock server からownership をもらわないとデータの修正はできない。

問題点:

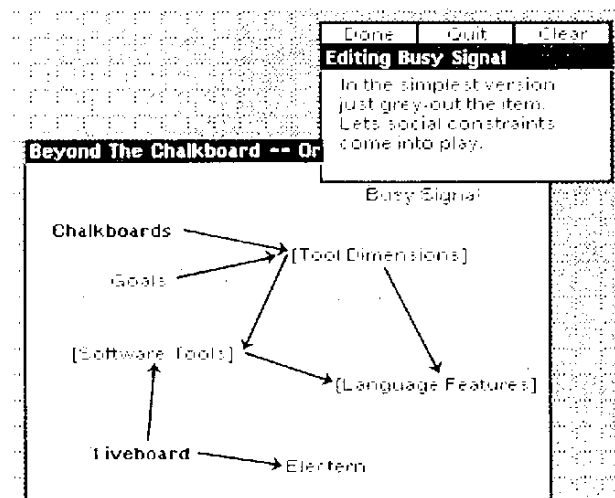
process scheduler の動きが遅いため、lockをかけるのに時間がかかる。

・ Cooperative Model (実際に用いている)

1. 各ワークステーションは、データベースのPC-を持っている。

2. 複数の人が同時に同じデータベースを修正しようとしたときは

- ・ 'voice lock'がかかる
 - ・ busy signal がでる
(修正中の項目が灰色にひかる)
- ことにより防止する。



問題点:

誤って、各ワークステーションごとに

同じデータベースを修正してしまう事がある。

・ Dependency-Detection Model (Cooperative Model の問題点解消案)

1. データに「修正者(修正を実行したワークステーション)」と「修正時刻」のスタンプを押す。
2. 修正時には、『修正後のデータ』、『修正後のデータのスタンプ』、『修正前のデータのスタンプ』を要求する。
3. 修正したワークステーションが異なっていたら人間が介入する。

・ Roving-Lock Model

1. lockをする機能を各ワークステーション分散させて、lockの所要時間の短縮を図る。

(6) 研究課題:

1. collaboration (協同研究、協同著作) の過程のどの部分に計算機による支援を行ったらよいのか、またそのための tool には何が良いのかを検討すること。
(→CognoterとArgnoterはそのtoolの一例である。)
2. collaboration (協同研究、協同著作) の過程が明確になったとして、数々の状況や使用者を想定した場合どの程度まで使い物になるのか。
3. 持ち運びが可能なものになるのか。
4. 数々の状況や使用者に合わせた tool を作る必要があるか。
5. 明確な構造が望まれるのはどのような状況か。簡略な構造が望まれるのはどんな状況か。

(7) 参考

Colab toolは、Xerox PARC上で、インプリメントされている。

(2.2.2.11 担当 森谷)

2. 2. 2. 12 CONSIST

システム名 CONSIST

文 献

表題 知識整理支援システムCONSISTの開発

著者 篠原靖志

出典 電力中央研究所報告・研究報告：Y86003, 1987.

キーワード 人工知能, 知識獲得, KJ法, フレーム, 知識表現

支援レベル 知識の収集

対象分野 限定しない

(1) タスクの特徴

知識システムを開発する場合、製作者（専門家、または、知識工学者）は専門家から集めた元となる知識の断片（項目）を知識ベースとして使えるように整理しなおす必要がある。ところが、一般に、知識ベースを活用したい分野は専門家の明文化されていないノウハウによる部分が多い。そのため、現在の知識システムでは、その実現に著しい困難が伴う。

知識ベースを構築するプロセスには、①対象問題についての基礎知識と問題解決に使われる方法とを整理する上流工程、②使用する問題解決方法に応じて必要な知識を獲得する中流工程、③実際にシステムを実現する下流工程がある。CONSISTが対象とするのはこのうちの上流工程の作業の支援である。すなわち、システムは、対象とする問題に関する知識が十分整理されていない段階から、対象および対象間の関連性に注目して、問題解決に必要な情報を知識ベースに蓄えられるような形式にまで整理する工程を支援する。

(2) システム概要

CONSISTは、知識整理の作業をKJ法を拡張した整理方法をもとに支援する。この際、製作者は、集めた項目を様々な視点から見直して、見落した項目間の関係を見つけたり、注目する関係の種類をはっきりさせたりする作業を行う。このように知識の整理の視点をはっきりさせることによって、はじめて、よくまとまった一貫性のある知識ベースができる。

知識システム開発の上流工程では、インタビュー、文献調査、ブレインストーミング、プロトコル分析などで得られたデータを元に、当該問題の解決に必要な基礎知識や問題解決方法を分析・整理することが重要である。CONSISTではこの分析・整理作業を行うのに「どの対象に着目すべきか」、「対象間に存在するどの種類の関係（目的、前後関係、属性、年齢など）に着目すべきか」を明確にしていく。CONSISTでは（着目すべき対象の集合、着目する関係の種類）の組を“視点”と呼ぶ。

視点は一般に最初から明確になっているわけではなく、また、新たに見出された対象や関係が必ずしも知識を整理するために適切な視点をあたえるわけでもない。知識整理の初期段階では隠れた関係を見出すための発散的な発想のフェーズと発見した関係を利用してよりよい知識整理の体系を見出す収束的な体系化のフェーズとが存在する。

CONSISTでは、発想のフェーズを支援するためにKJ法をベースとしている。整理方法は、(1)項目のグループ化、(2)項目間の関係付け、(3)項目間の関係の吟味、の3つからなる。関係間に階層を導入した知識表現を用いることで、(1)～(3)における整理情報を統一的に扱っている。

体系化のフェーズでは、視点を変えた再整理の容易化をはかった。たとえば、利用者が指示した視点にあわせて、現在の整理結果から自動的にKJ法のネットワーク図を作成して利用者に提示する。さらに、利用者がこのネットワーク図上でKJ法の整理を行うとそれがただちに全体の整理結果に反映される。これにより、複数の視点からの整理情報の見直しなど立体的な情報の整理が可能となっている。

このような整理途中の知識を表現するためには、利用者の着目する視点の変化を柔軟に捉える等、従来の手法に比較してより柔軟な知識表現方法が必要となる。発想のフェーズでは利用者は新たな対象や関係を自由に作り出し新たな視点を定義する。体系化のフェーズでは、利用者は大雑把な見方や詳細な見方、異なる視点からの分析など視点を絶えず変

更しながらよりよい整理結果を求める。さらに、たとえば、「年齢」関係は「属性」関係の一種であるなどこれらの視点（対象と関係の種類）は相互に関連している。

CONSISTではこのような整理途中の知識を表現するために、フレーム表現を拡張した新しい知識表現法を用いた。この方法は対象の間に定義される関係の種類にクラス階層を導入したものである。これによって、整理開始当初は、漠然としている対象間の関係を上位の“一般的な”関係としておき、分析作業が進むにつれて“因果関係”などより具体的な関係に定義しなおすことが可能となる。この知識表現は、利用者が、カード、カード間のリンク、リンクの持ち得る関係の種類などを自由に定義できるカード型データベースとして実現されている（図. 2. 2. 2. 12（1））さらに、CONSISTでは、整理された知識を知識ベースで一般に利用されるフレームやルールなどの知識表現に変換することも容易となっている。システムはSYMBOLICS 3670上で実現されており、現在、地域振興策データの分析などの複雑な作業に使用している。図2. 2. 2. 12（2）にシステムの利用者インタフェース例を示す。

（3）今後の課題

本システムの支援する複数の視点に基づく整理過程は、未整理の情報、知識の整理に有効であることが判明している。ただし、現在のバージョンは、データベースの検索機能のみをもち、推論機構は内蔵していない。今後は、整理途中の知識の抜けや見落しの検出自動化、仮説の検証の自動化などの機能の充実をはかる計画である。

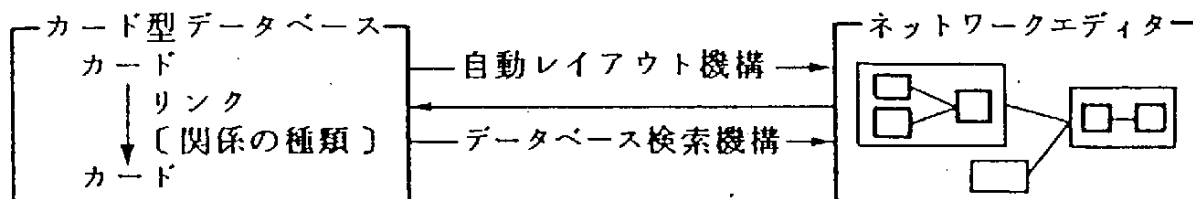


図. 2. 2. 2. 12（1） CONSISTのシステム構成

2. 2. 2. 13 C Y C E d i t o r

システム名 C Y C E d i t o r

文 献

表題 Tools for Adding Knowledge to the CYC LSKB

著者 Shepherd, M.

出典 Proc. the International Workshop on Artificial Intelligence
for Industrial Applications, pp. 365-369 (1988).

キーワード 人工知能, 知識獲得, 知識エディタ, 大規模知識ベース,
C Y C

支援レベル 知識の収集

対象分野 大規模知識ベース開発

(1) タスクの特徴

知識システムの問題解決能力を向上させるには、究極的にはシステムに類推・学習の能力を持たせなければならない。そのためには、まずコンピュータ上に常識的な幅広い知識を手で投入する必要がある。MCCで行われているCYCプロジェクトは、こうした目的で、10年間の歳月をかけて、百科事典の内容すべてを大規模知識ベースへ構築しようという試みである[Lenat 1987]。

CYCプロジェクトでは、当初、通常の百科事典から選択した約400種類の基本的な項目を、最初に豊富な機能をもつフレーム型言語を使って表現した。次の段階で、その情報をもとに複写・編集作業を繰返すことで知識ベースの段階的な開発を行っている。

1987年時点におけるシステムは中央のギガバイトクラス・ファイルサーバを中心に

LAN で接続した10台以上のSymbolicsリスブマシンから構成されている。

CYC Editorは、この百科事典知識ベースの内容に詳しい専門の知識技術者 (Cyclistという) が、新しい知識をフレーム形式で効率よく投入することを目的として開発された。CYCの知識ベースは、図2. 2. 2. 13 (1) に例示したような多くのスロットをもつ複雑なフレームシステムである。それにあわせてCYC Editorは知識ベースの局所的な整合性をチェックする機能が特に強化されている。

Communicating

basicKindOf: Informing

actors: (communicator a) (recipient b)(message c)

action: (InSequence (OpeningCommunications a b)

(Telling a b c)

(RecallMatching b c)

(ClosingCommunications a b))

constraints: (CanUnderstand b a c)

(Desires a (Communicating a b c))

2WayCommunicating

basicKindOf: Communicating

action: (Repeatedly (Communicating a b c_i)

(Communicating b a d_i))

Querying

basicKindOf: 2WayCommunicating

actors: (question c_i) (declaration d_i)

constraints: (ResultOf d_i (Remembering b c_i))

AskedInInterview

basicKindOf: Querying

Remembering

basicKindOf: Informing

actors: (rememberer a)(memory b)

constraints: (Accurate (Remembering a b))

accuracy: (# - - Moderate variance High)

allWaysItFails

defn: (Lambda (u) (MapUnion

(fullAction u) 'waysItFails))

elementOf: InheritedSlots UnlessFillingSlots

builtFrom: fullAction waysItFails

combiner: MapUnion

makesSenseFor: Processes

waysItFails

defn: (Lambda (u &rest) (negate (constraints u)))

elementOf: DerivedSlots UnlessFillingSlots

builtFrom: constraints

combiner: negate

usedInBuilding: allWaysItFails

makesSenseFor: Processes

constraints

elementOf: ScriptSlots PrimitiveSlots

usedInBuilding: waysItFails

makesSenseFor: Processes

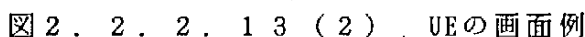
A few pieces of general CYC frames,
plus the one fact known
about AskedInInterview.

図2. 2. 2. 13 (1) CYCで使われるフレームの記述例の一部 [Lenat 1986]

(2) システムの概要

CYC Editorは、テキストベースのUE(Unit Editor)、カラーグラフィック指向のMUE(Museum room Editor)、ならびに、編集集中で生ずる副作用を制約伝播させるバックグラウンドプロセスの3つの構成要素をもつ。編集したフレーム情報は、ローカル(使用中のマシン)にも、グローバル(ファイルサーバ)にも保存することができる。この垂

図2. 2. 2. 13 (2) にUEの画面例を示す。図には4つの編集画面(Editing Pane)が表示されており、それぞれに別のフレーム(unit)情報 (slots, slot values)が表示されている。ここで知識ベース中のフレームの状態を見ることができる。実際の編集対象はこのうちのActive Editing Pane である。スロット値の一覧はSlot Value Pane に表示される。Computable Slotsは対象フレームの情報のうち他のフレーム情報をもとに計算されるスロットを示す。Legal Slotsは対象フレームが保持することが可能なスロットを表わす。Viewed Unitsは編集対象フレームの一覧である。Command Menu, Type-in windowはリスマシン操作のウィンドウである。



バックグラウンドプロセスは、知識ベース編集集中に間違いが生じた場合や、知識投入者の質問に答える場合になどに起動される。

Chosen Unit



Restrictions

図 2. 2. 2. 13 (3) MUE の画面例

(3) 今後の課題

このツールは知識獲得作業の自動化をめざすものではないが、大規模知識ベースを複数人で開発する場合に生ずる困難を避けるための、チェック機能に優れており今後の研究に大きな影響を与えるものと考えられる。

〔参考文献〕

- [Lenat 1987] Lenat, D., Feigenbaum, E.A.: On the Thresholds of Knowledge.
Proc. 10th IJCAI, pp. 1173-1182, 1987.
- [Lenat 1986] Lenat, D., Prakash, M., Shepherd, M.: CYC: Using Common Sense
Knowledge to Overcome Brittleness and Knowledge Acquisition
Bottlenecks. The AI Magazine, Vol. 6, NO. 4, pp. 65-85 (1986).

2.2.2.14 ES/SDM

システム名	ES/SDM
文献	
表題	ES/SDM
著者	松本俊二
出典	日経AI別冊, 1988年7月
キーワード	人工頭脳, 構築技法, 専門家モデル, プロトタイピング, 知識獲得,
支援レベル	懸念化, 詳細化, (特に構造化)
対象分野	特に限らない。

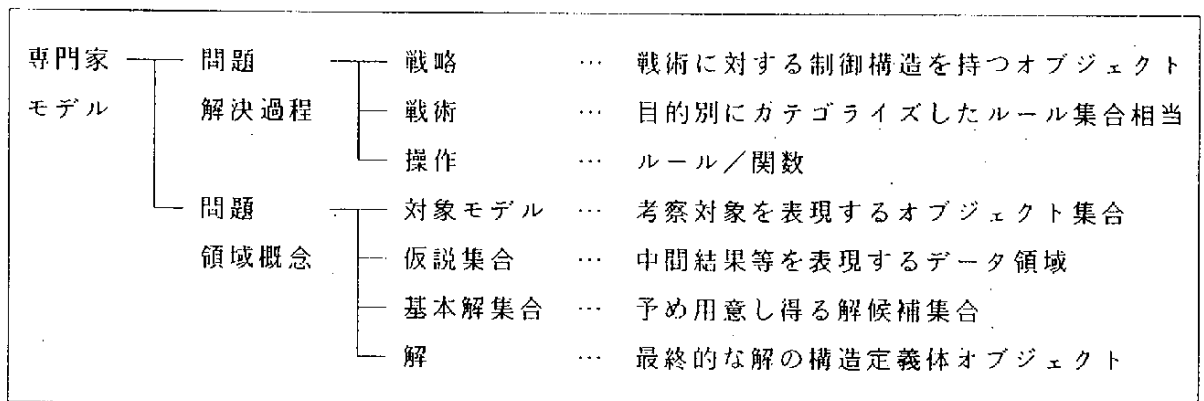
(1)概要

ES/SDMは, ESHELL (富士通エキスパートシェル) の応用経験に基づいて体系化した, エキスパートシステム構築のためのガイドラインである。ルール表現・フレーム表現・イベントドリブンエンジンを持つツールを前提とし, 次のような特徴をもつ。

- ①知識システムの枠組みとして専門家モデルを設定し, 効率的な知識獲得を実現するための中核とする。
- ②プロトタイピングを基調とし, 初期段階で解決過程が不明瞭な問題のシステム化を支援する。
- ③問題解決過程のパラダイム (雛形) を用意し, 部品化・再利用の基礎を提供する。

(2)専門家モデル

利用目的によってプログラムを分類するために, 次のようなモデル設定を行ない, 専門家モデルと呼ぶ。



(3)プロトタイピング

専門家モデルの各要素が初期段階で明確な場合は, 仕様を固め, 知識を収集・整理し, 構造設計を行なう。そうでない場合, プロトタイピングによって次第に仕様を明確にしていくことにより, システム化しにくかった問題領域にもアプローチしていく。

プロジェクト管理の側面から, マネジメントレビューを挟んでステージを設定し, 各ステージの中で, 概念化→構造化→詳細化→検証 (初期フェーズはステージが進に従って省略される) の各フェーズを繰り返し進める。

3. 知識システム開発事例にみる知識獲得の実際

3. 知識システム開発事例にみる知識獲得の実際

3章では、ワーキンググループ各委員がみずから開発に携わった知識システム事例を分析し、それぞれの対象に即した理想的な知識獲得支援ツールのイメージを提案する。

3.1 概 要

3章では、対象とする事例を分析型問題と計画・設計型問題都にわけ、それぞれについて、ワーキンググループ各委員が適切な事例を持寄って、知識獲得支援という視点から分析を行なった。

以下の節において、各事例に関する議論は次の順序でなされる。まず、最初に対象システムの名称を述べ、その概要を記述する。次に、[ICOT-JIPDEC 1987-1]において示した”類型的タスクと問題解決機能との関連”の記述形式にしたがって、当該知識システムの特徴を取りまとめる。ついで、当該システムにおける知識獲得フェーズの特徴について考察を行ない、さらに、当該システムと同様のタスクを扱うことを仮定した場合に望まれる知識獲得支援機能について論ずる。最後に、そのために必要とされる理想的な知識獲得支援ツールのイメージを提案する。

本章で取扱う問題は次のとおりである。

分析型問題としては、クレジット入会審査エキスパートシステム、自動車の故障診断システム、高炉操業支援エキスパートシステムの3つを取上げた。クレジット入会審査エキスパートシステムは、原理的には比較的容易とされている分類問題を扱っている。大量の元データから入会審査に必要な判断知識を、決定木の形式で生成するために帰納的な学習機能を利用している点が知識獲得の面から興味深い。自動車の故障診断システムは、故障原因の集合が非常に大きいタイプの診断問題の典型的な事例である。熟練者の技能を集約し標準化するとともに、多種多様な車種や年式の故障診断に対応することが要請される。このような問題の知識獲得においては、感覚的な情報と設計情報の扱いが重要となる。高炉操業支援エキスパートシステムは、既存システムとの協調・接続が必要な大規模制御問題の典型例である。このような問題の知識獲得では、専門家の漠然とした制御知識とセンサーからの情報とを関連づけることが大きなポイントとなる。

計画・設計型問題としては、シミュレータのユーザインタフェース設計、連接水系の翌

日発電計画システム、ルート計画エキスパートシステム、電子装置の実装設計支援エキスパートシステムを事例として取り上げ、考察を行った。シミュレータのユーザインタフェース設計では、ウィンドウやメニューなどの豊富なシステム構成部品を要求仕様にしたがって組合せながら設計を行なう。これに必要な知識の獲得には、特に解空間を制約するための情報を収集することが重要となる。連接水系の翌日発電計画システムは、実行可能な計画を1つ妥当な時間内に立案することを目的としたシステムである。このような問題の知識獲得においては、数理計画手法や従来型システムと経験則の利用とが重要となる。ルート計画エキスパートシステムは、輸送手順と資源の割当てを扱う。このような問題では、専門家なみの解を得るばかりではなく、それよりも優れた（準）最適解を導くことが要請されるので、知識獲得においては新しい解法を導くための知識の役割りが重要となる。電子装置の実装設計支援エキスパートシステムでは、設計の詳細化にしたがって当初は未定義であった各種のパラメタを決定する必要がある。このような問題の知識獲得には、過去の設計ドキュメントなどの知識源の整理と対象システムの階層構造の整理が特に重要となる。

本報告では、各事例に関する知識獲得支援ツールのイメージを統合化することは試みていないが、それぞれに共通する要請としては次のようなものが上げられよう。

（１）知識収集段階

最高レベルにある専門家の（経験的）知識の獲得や、問題解決の発想を支援する機能が必要とされる。また、それとは別に、対象システムに関するいわゆる深い知識を機械的に収集する機能が望まれる。

（２）知識変換段階

専門家モデル、ユーザモデルを充実させることで、一般レベルの専門家の知識の投入を容易化する機能、ならびに、知識の自動コーディング機能の必要性が高い。さらに計画・設計問題においては、専門家レベル以上の問題解決機能がこの段階では要求されることが多い。

(3) 知識管理段階

いわゆる知識ベースの整合性チェック機能などにくわえて、既存知識をライブラリ化しておきそれを利用してシステムのカスタマイズを支援する機能、診断問題におけるCF値の調整や計画問題における評価関数のチューニングなどを自動的に行なう機能、システム利用例からの帰納的学習機能などが要求されている。

参 考 文 献

[ICOT-JIPDEC 1987-1] “知的情報処理システムに関する調査研究報告書-知識システム開発方法論-”. 1987.

3. 2 分析型問題

3. 2. 1 クレジット入会審査エキスパートシステム

3. 2. 1. 1 システムの概要

(1) 背景

クレジットカードの申請者から、支払い不能の事故を起こす可能性のあるものをあらかじめ取り除くことは、カード会社にとって非常に重要な課題である。入会申し込み者の許可・不許可の審査判定を行う業務は入会審査と呼ばれ、専門の担当者がこれに従事している。通常は、入会申込書に記載された性別、年齢、住所、年収などの項目の記載漏れ、虚偽の記入のチェックがなされたあと、入会審査担当者が総合的な判断を行い、入会の諾否が決定される。こうした入会審査業務の問題点として、以下の点が挙げられる。

- ・リスクから収益への審査基準の移行
- ・カード発行の迅速化の要請
- ・入会申込者の急増
- ・判断の不均一性
- ・審査基準の検証方法の欠如

本システムは上記の問題点を解決すべく開発されたものであり、入会審査担当者が審査判断を下す際、過去の経験から形成された申請者のタイプ分けを用いることに着目し、入会申込書の記載内容から特定のパターンを読み取り、それによって入会を許可するか否かを決定している。この方法をプロフィール法と読んでいるが、従来こうした問題で採用されてきたスコアリング法（申込書の各属性値に与えられた点を集計して判別関数の値とする統計的な手法）に対するプロフィール法の利点は次の様である。

- ・属性値が不連続量でも扱うことができる
- ・複数の属性値の組み合わせで評価する際にも有効
- ・人間の審査判断の方法により近い

(2) 機能

本システムでは申込者を、その属性値によって特定のプロフィールに分類することによって、入会の可否を決定している。プロフィールに分類する手段としては決定木展開を用いている。次図は決定木の例である（データは架空のもの）。

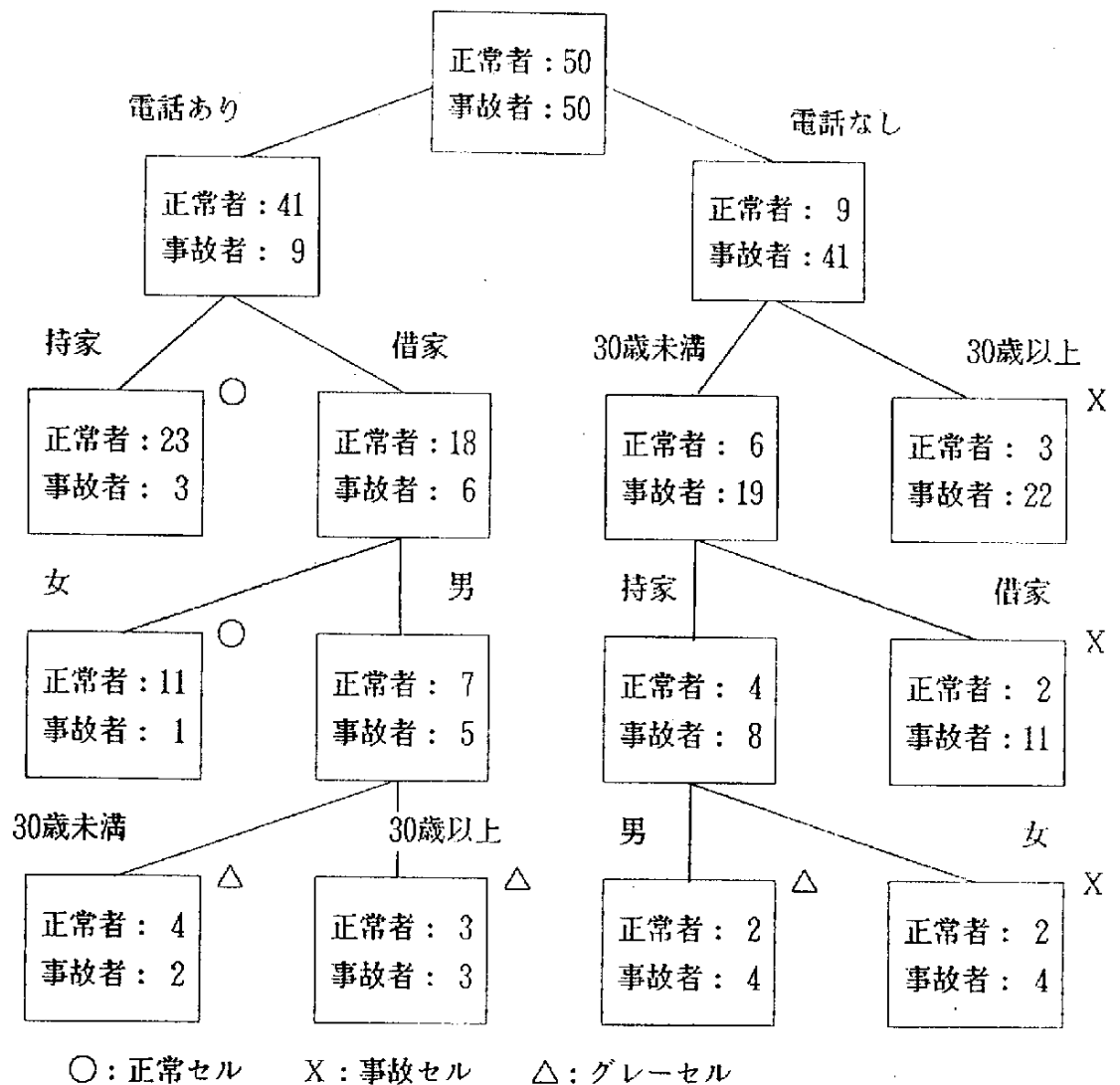


図3.2.1-1 決定木の例

各プロフィールには次のような要求がある。

- ・各プロフィールの良否が明確に分かれること
- ・任意の申込者は必ず1つのプロフィールに分類されること
- ・個々のプロフィールには十分な実績データが得られること

こうした要求を満たすようなプロフィールを作成するのが、上に示した決定木であるが、正常者と事故者を最も良く区別する属性値を決定するために、情報理論に基づくエントロピーを用いている。(Quinlan のID3 アルゴリズムによる)

3. 2. 1. 2 類型的タスクと問題解決機能との関連

主としてエンジニアリング分野を対象に考えられている現在の類型的タスクでは必ずしも適切にこの問題を記述できないが、強いて言えば診断問題ということになる。

	診 断 問 題	関 連	実 現	事 例 と の 関 連
定 義	システムに異常または故障が生じた時、観測されたデータおよびシステムに関する知識を利用して原因の箇所を同定すること			直接的に異常や故障を扱う訳では無いが、そうした問題の際に用いられる分類・選択的な問題解決のパラダイムが使われている
特 徴	1. 設計レベルの知識の利用 2. 経験的知識の利用 3. 経験的知識の不確実性 4. 操業データの利用 5. 計測のための時間とコスト 6. 知識の抽象化の必要性 7. 対話的診断の推論制御	X ○ ○ ○ ○ △ X	X △ △ ○ ○ X X	「設計」は無い 経験的な知識が反映されている 統計的な扱いをしている 実績データを用いている 申込書記入が計測に相当 計測データのみを用いる —
基 本 タ ス ク	1. 異常事象の分類階層的表現 2. システム構造の階層的表現 3. 解釈問題のタスクを内包 4. 計測点の選択的決定 5. 異常原因の同定 6. 浅いモデルによる効率性 7. 深いモデルによる完全性	○ X ○ X △ ○ X	○ X ○ X △ ○ X	決定木として展開 対象システムは人間である データ解釈の結果が決定木である 申込書であらかじめ設定されている プロフィールが結果的に異常原因になる 決定木は効率的な浅いモデル —
問 題 解 決 機 能	1. 事象駆動型推論 2. 目標駆動型推論 3. 不確実性推論 4. 仮説推論 5. 協調型推論 6. 効率性と完全性の調和 7. 費用／効果分析	X ○ △ X X △ △	X ○ △ X X X X	— 正常者／事故者の区別が目標 正常者と事故者の比率を統計的に表現 — — 将来的には必要 将来的には必要

3. 2. 1. 3 知識獲得フェーズの特徴

(1) データからの知識獲得

正会員2504名、事故会員1818名からなるテストデータ(属性数15個)を元に前述の決定木展開アルゴリズムによってプロフィールの設定を行った。しかし、設定された340あまりのプロフィールの70%以上が30未満の少人数セルであった。少人数セルは、含まれるデータが少ないため判断の根拠となる統計データが不正確で審査判断の根拠とするには問題がある。したがって、決定木展開アルゴリズムに以下の改良を加えた。

- ・属性の分類の階層化
- ・枝刈り機能の付加

(2) 専門家からの知識獲得

過去のデータを利用したプロフィールの生成とともに、審査担当者から審査判断上重要なプロフィールの収集を行った。これらのプロフィールは重複があることや属性値の特定の度合いがまちまちであることのため、そのままシステム上で利用するのは困難である。したがって、運用にあたっては、データによるプロフィールはすべての入会申込書をカバーして審査判断の基礎データとして利用し、専門家から抽出した特定のプロフィールは特に注意すべき点があることを審査担当者に警告するために利用する、というように使い分けている。

3. 2. 1. 4 望まれる知識獲得支援機能

上に示したのは帰納的学習に基づいた知識獲得支援機能と言える。こうした手法は、問題の構造や範囲が明確な分類型の問題に対して有効であるが、一般的な知識獲得のタスクを考えると、分類型以外の問題や、漠然とした専門家の知識の収集、明文化といった比較的初期のフェーズを支援することも非常に重要な課題である。

3. 2. 1. 5 理想的な知識獲得支援ツールのイメージ

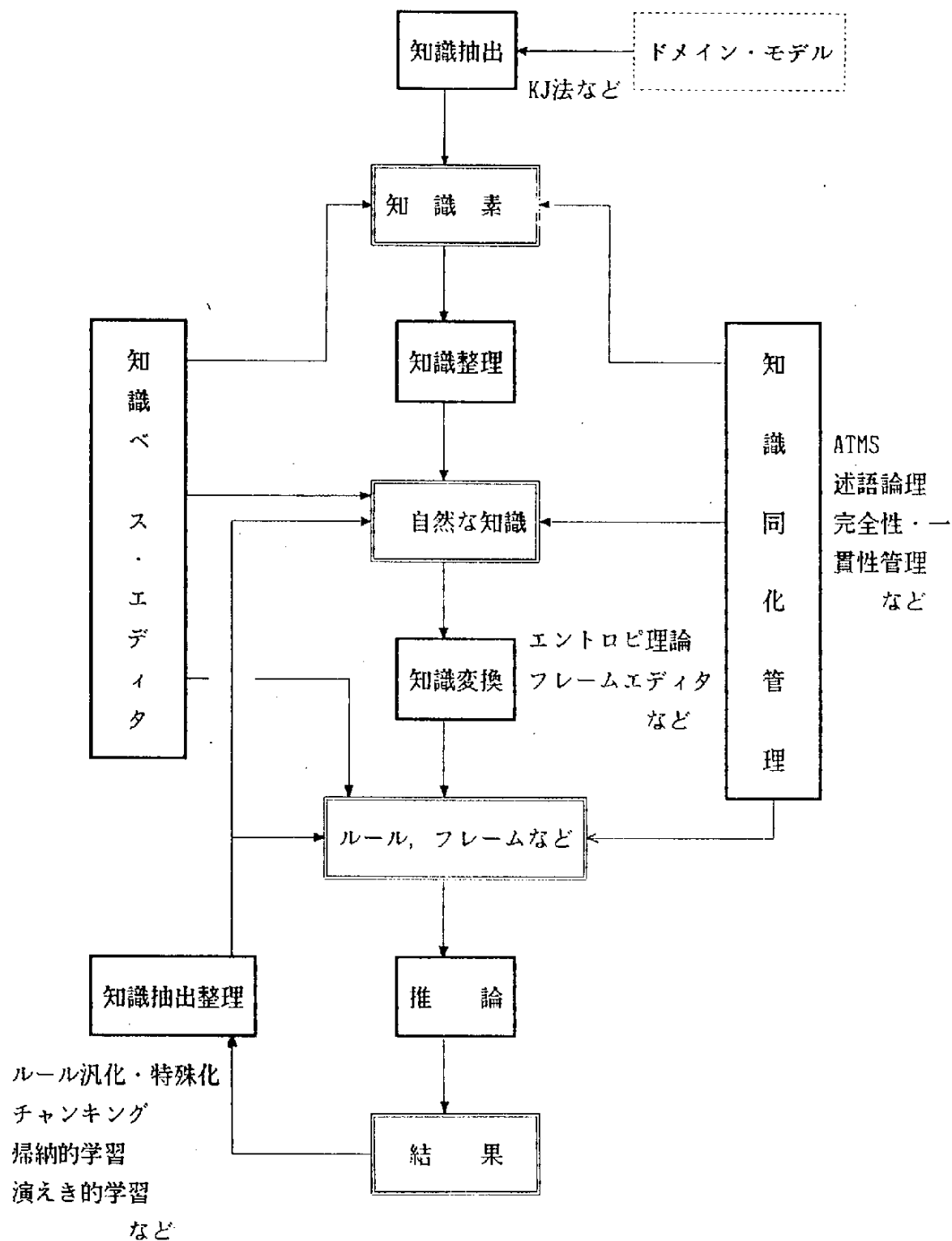


図3.2.1-2 理想的な知識獲得支援ツールのイメージ

参照文献

- 〔1〕 藤田, 玉井 “プロフィール法に基づくクレジット入会審査エキスパートシステム” 品質 Vol.18, No.2 56—62頁

3.2.2 自動車の故障診断システム

3.2.2.1 システムの概要:

自動車は数万点の部品からなる複雑なシステムであり、その点検整備にはメカニックの持つ知識や経験が重要な役割を果たしている。熟練したメカニックは経験から得られた知識を利用して効率よく、しかも適切な診断を行うことができる。このような熟練者の技能を集約し標準化することは重要な課題である。

一方、車種の多様化やエレクトロニクス部品の導入に伴い、故障診断技術にも高度化・自動化をはかる必要性が高まりつつある。これに対応するものとしては、整備工場に設置したコンピュータを利用して計測の自動化を行うオフボード点検診断装置や、車両に自己診断機能を持たせるオンボード診断装置などが研究・開発されている。

エキスパートシステム技術は、診断分野における有力な手法のひとつとなりつつあり、自動車関連でもFORD、日産などのシステムが公表されている。ここでは当所で試作したATREX(Automobile Troubleshooting EXpert system)について紹介する。

本システムは、サービス部門において、診断についての問い合わせに答えることを目的として開発されている。対象が自動車であるため、多様な車種や年式に対応できることが重要である。また、運用を含めて利用部門への技術移転を可能とするため、知識ベースの保守が利用者主体で行える保守性の高いシステムを目指している。

本プロトタイプは次のような診断機能を備えている。1)初期入力情報(車種、年式、型式、故障状況、点検した部品等)から原因を絞りこむための点検手順を指示する。2)点検結果から、その原因と考えられる部品あるいは部品系を推定する。3)ある故障が生じた場合に、それがどのような現象を引き起こすかを示す。この機能により推定された故障原因の妥当性を確かめることができる。特に、2種類の故障原因から引き起こされる現象の中で共通でないものを表示する機能は原因の切り分けに有効である。4)修理手順を指示する。

また、次のようなマンマシンインタフェース機能を備えている。1)タッチパネルによる入力、2)レーザーディスクを利用した、ヘルプ情報や点検修理手順などの動画による表示。

システム構成を図3.2.2-1に示す。開発ツールは大型機ACOS850上のEXCOREを用いている。このホストに回線接続された端末機器でインタフェース機能を実現している。モニタTVはコンピュータ出力とレーザーディスクからの映像の両方を表示する。また、この装置には赤外線タッチパネルが付けられており、利用者はすべての操作を1台の装置で行うことができる。

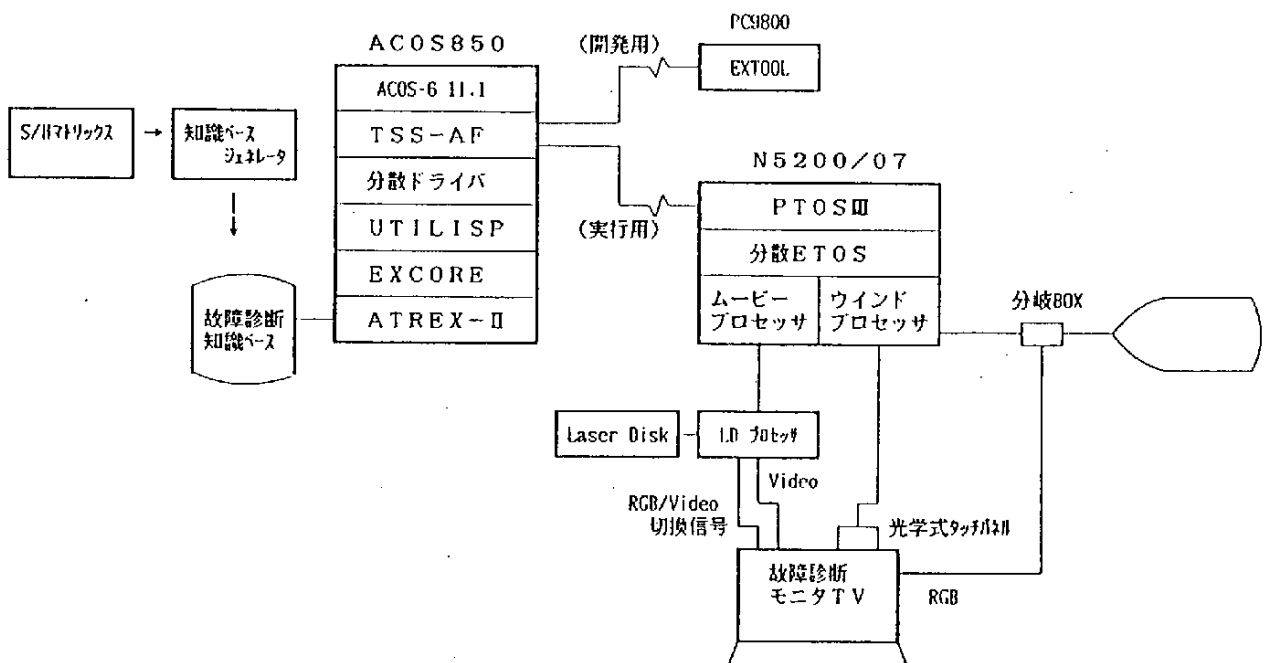


図3.2.2-1 システム構成

このシステムの特長としては次のようなことがあげられる。1)任意の時点で入力を修正追加して再推論する機能と、故障が引き起こす現象を表示する機能とにより、利用者主導で効率よく絞りこみが可能。2)レーザーディスクやタッチパネルの採用によりコンピュータ操作に不慣れでも容易に利用できる。

今後実用化へ向けて次のような課題が残されている。1)知識ベースの保守性向上：部位、車種などの取扱い範囲の拡大に対応するためには知識ベースに構造を導入するなどの対策が必要である。2)システム構成の簡素化：このプロトタイプではマンマシンインタフェースの機能実現のためのハード構成が複雑である。これらを統合化した装置の開発が望まれる。

3.2.2.2 類型的タスクと問題解決機能との関連

	診 断 問 題	関 連	実 現	事例との関連
定 義	システムに異常または故障が生じたとき、観測されたデータおよびシステムに関する知識を利用して、原因の箇所を同定する。	○	△	対象システムについての深い知識の利用が課題
特 徴	1. 設計レベルの知識の利用 2. 経験的知識の利用 3. 経験的知識の不確実性 4. 操業データの利用 5. 計測のための時間とコスト 6. 知識の抽象化の必要性 7. 対話的診断の推論制御	○ ○ ○ △ ○ ○ △	△ ○ △ △ △ × △	浅い知識にコンパイルされている CF値を利用(入力にも指定可) オンボードダイアグのデータを利用 点検手順で考慮されている 因果関係を陽に扱っていない 冗長な質問の抑止など
基 本 タ ス ク	1. 異常現象の分類階層的表現 2. システム構造の階層的表現 3. 解釈問題のタスクを内包 システム状態の推定 あいまいさの処理 4. 計測点の選択的決定 5. 異常原因の同定 6. 浅いモデルによる効率性 7. 深いモデルによる完全性	○ ○ ○ ○ ○ ○ ○	○ △ ○ × △ ○ ×	診断システムのモデルとして自然 車種による構造の違いを表現 症状の表現にあいまい性あり 点検指示機能 将来的には必要

問題 解 決 機 能	1. 事象駆動型推論	○	○	
	2. 目標駆動型推論	○	△	故障から引き起こされる現象を表示
	3. 不確実性推論	○	△	現状はCF値の足しこみであるが、今後 D-S理論などの適用をはかりたい
	4. 仮説推論	○	×	深いモデルに関連
	5. 協調型推論	○	×	
	6. 効率性と完全性の調和	○	×	
	7. 費用/効果分析	○	△	点検コストと絞りこみのかねあい

3.2.2.3 知識獲得フェーズの特徴

自動車故障診断の知識源としては、メカニック、サービス部門スタッフ、設計者、サービスマニュアルなどがある。現在は、これらの知識源が持つ記号化可能な知識および数値データなどを対象としている。メカニックの経験的な知識の中には、音、振動、排気の色、レスポンスなど感覚的な情報に基づくものも多いが、これらの取扱いは計測の方法を含めて今後の課題となっている。また、設計情報を深い知識としてそのまま利用できれば理想的であるが、複雑なメカトロニクス機器であるために現状技術では取扱いが困難である。定性推論などの手法の発展が期待される。

診断問題では、対象機器の様々な仕様に対応できなければならない。自動車の場合には特に多様化が進んでおり、車種、年式、型式によって部分的に異なる数多くの仕様を対象としている。また、技術開発により新たな仕様が加わる頻度も高い。このため、知識ベースを構造化し、診断情報の階層化をはかることが重要な課題となっている。

知識の種類としては症状と原因との対応をとる知識に加えて、利用者との対話制御のための知識も重要である。システムはこの知識を利用して、次のような点を考慮した順序で質問を行う。1) 点検に要するコスト・時間と、絞りこみのかねあい。2) 冗長な質問の抑止。この質問制御の知識についても、診断知識と同様に車種などの仕様に応じた階層化が必要である。

3.2.2.4 望まれる知識獲得支援機能

(1) 知識の収集段階

ここでは専門家モデルの初期生成が目的となる。本システムの場合には、点検手順についての考慮をどのように問題解決手法に反映させるか、といった様々な問題を試行錯誤的に考えた。この段階では、事例や専門家モデルのテンプレートを用意してプロトタイプ開発を支援する機能が望まれる。

(2) 知識の変換段階

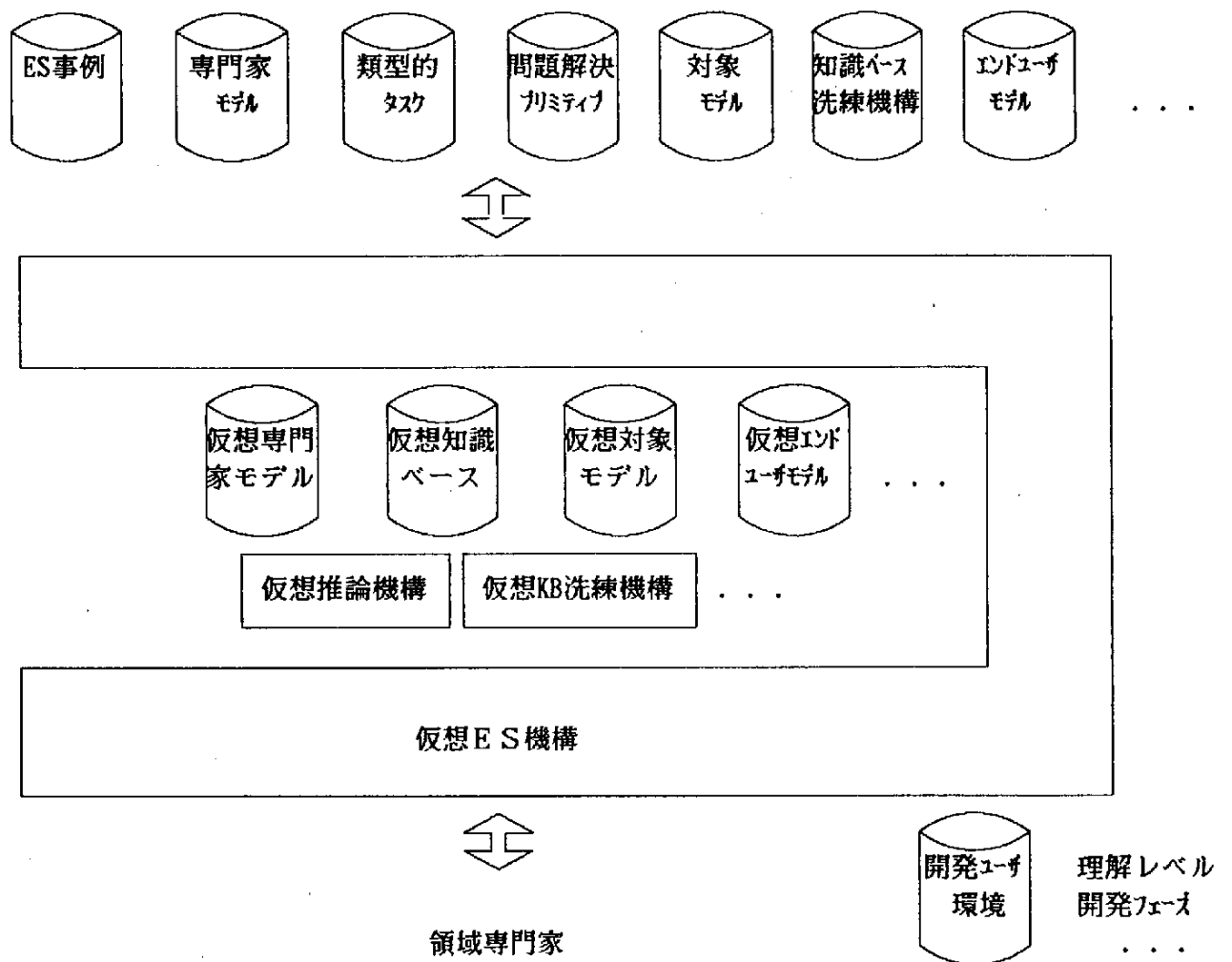
専門家モデルから知識ベースシステムの枠組みを生成する機能、実行効率の向上や知識ベースの構造化などの調整を支援する機能、および、実際の知識の抽出を支援する機能などが求められる。

(3) 知識の管理段階

診断結果のフィードバックによって自動的に知識のチューニングを行う機能や、新たな仕様情報をエンドユーザが容易に追加できる支援機能が必要。

以上をまとめると、プロトタイピングの考え方を理想化したものとして、エキスパートシステム事例、専門家モデルのテンプレート、典型的タスク、問題解決プリミティブ、対象モデルのテンプレート、知識ベース洗練機構のテンプレートなどを持ち、必要に応じて組合せてカスタマイズする機能を持つ仮想ES機構を備えたツールが望まれる。また補助的な機能として、領域専門家自身による開発を支援するために、ES技術についての理解レベルや開発のフェーズなどを環境として定義し適切なインタフェースを提供する機能も有用である。

3.2.2.5 理想的な知識獲得支援ツールのイメージ



3.2.3 高炉操業支援エキスパートシステム

3.2.3.1 概要

(a)高炉プロセスの特徴

一貫製鉄プロセスにおいて「高炉」は最も上工程に位置し、鉬石及びコークスを原料として溶銑（溶けた銑鉄）を得る大規模反応プロセスである。高炉本体は、内容積 5000m^3 にも達し、約40mの炉体上部から鉬石及びコークスを層状に装入し、炉体下部からは約1000℃の熱風が高圧で送り込まれる。炉内では、これらの固体、気体、溶融した液体が交錯し、酸化・還元反応が高温環境下で、大規模かつ連続的に行われる。 5000m^3 クラスの高炉で生産される溶銑は日産10000tonにも達する。

日常の操業では、炉の周辺に取り付けられた数百の熱電対をはじめとする炉内センサー情報を連続的に収集しており、オペレータはそれらのセンサー情報及び生産される溶銑の状態や熱風の吹き込み状況等の目視情報、部分的なプロセスモデルによる推定結果等を総合的に考慮して炉内状況を判断し、何らかの操業上のアクションが必要とあれば実行する。アクションの種類には、①炉体上部での原料分布（MAノッチまたはPWノッチによって制御可能）、②鉬石とコークスの割合（FR：Fuel Rate）、③吹き込む熱風の量（風量）等がある。

高炉は、原料装入から出銑まで、5～6時間を要することからわかるようにプロセスの時定数が数時間と非常に長い。従ってアクションの決定は炉内状況を把握してから10分後位に行われており、一般のプロセス制御とは大きく異なる。

(b)エキスパートシステム導入の狙い

高炉操業にエキスパートシステムを導入する目的は、溶銑の生産量の安定的確保と、品質（溶銑中のSi成分低下等）維持にある。

(c)エキスパートシステムに装備された知識概要

エキスパートシステムがサポートする知識は、次の4つに大別される。

(i)日常操業の知識

通常の連続稼働状況下でコントロールアクション（燃料比、風量、装入物分布等の調整）を行うための知識であり次の二つに分類される。

1. 炉の不安定を回避するための知識である。

2. 安定状態を保ちつつ溶銑品質の向上、生産コスト低減をめざすための知識である。

(ii) 非定常操業の知識

日常の操業で不安定な状態の回避に失敗すると、究極的には”棚つり状態”（装入原料の降下しない状態）に至る。この棚つり状態を通常状態に戻していくための知識である。

(iii) 休風ガイドの知識

高炉操業は、その設備補修や生産調整のため、1・2回/月～1回/数カ月の頻度で計画的に生産を休止する（これを休風という）。この休風状態に円滑に移行し、また休風状態から通常の生産状態に移行するための知識である。

(iv) 設備トラブル対処の知識

通常の生産中に主要な設備のトラブルが起きたときにオペレータが対処するための知識である。

(d) システム構築のステップ

(i) STEP 1（プロトタイプ）

優れた操業オペレータから知識を獲得し、分かり易い知識表現を行い、さらにオフラインシミュレーションにより妥当性の検証を行う。対象知識は、”炉の不安定回避”に的を絞った。

(ii) STEP 2（実機化）

対象知識の範囲の拡張と必要機能の具現化を行う。既存システムとのリンケージを考慮したシステム設計及び実機稼働中の効果検証によるシステム見直しを行う。

(iii) STEP 3（レベルアップ）

知識システムとしての全体機能の向上と高炉の新しい検出端に対応した知識の精密化を行う。

現在はSTEP 2を終了し、第3・4高炉において稼働中である。

3.2.3.2 類型的タスクと問題解決法との関連

* 高炉操業エキスパートシステムは、分析型問題の中の「制御問題」に位置する。

	制御問題	関連度	実現度	事例との関連
定義	制御問題とは、システムの状態を監視していて、あらかじめ予定されたとおりにシステムの状態が遷移するように操作を加えることである。	○	○	左記の問題をオペレータガイダンスとして行う。
特徴	1. 時間遅れによる履歴依存性 2. 非線形性による局所性 3. 安定化の応答性 4. 制御精度の実現 5. 実プロセスとの接続 6. 実時間性と信頼性の確保 7. 離散系における解析問題	○ ○ ◎ △ ◎ △ ×	○ ○ ○ △ ◎ △ ×	・ 時定数、時間遅れ大 ・ 非線形性大 ・ グローバルな安定化をめざす ・ 厳密な精度は要求されない ・ 既存システムとの接続・協調有 ・ 分オーダーのリアルタイム性でよい ・ 特に関連なし
基本タスク	1. システム構造の表現 2. システム動特性の表現 3. <u>診断問題のタスク</u> を内包* 4. モデルによる状態の予測 5. 安定化動作の優先的実行 6. 安定操業下での省エネ化 7. オペレータガイダンス	○ △ △* ○ ◎ ◎ ◎	○ ○ △ ○ ◎ ◎ ◎	・ 領域専門家の思考を表現 ・ 陽には記述していない ・ 部分的にプロセスモデルを利用 ・ 炉の不安定回避を優先実行 ・ 攻撃的操業知識として表現 ・ コミュニケーション手段
問題解決機能	1. 状態の診断機能 2. シミュレータによる予測 3. 定性推論による予測 4. 制御則の多目的評価 5. アクションガイダンス 6. デッドロックの解析 7. インターロックの回避	◎ ○ △ × ◎ × ×	◎ × × × ◎ × ×	・ = 操業状態の判断 ・ 問題に適したシミュレータはない ・ 実施していない ・ 実施していない ・ コミュニケーション手段 ・ 実施していない ・ 実施していない

凡例

記号	関連度	実現度
◎	非常に大きい	かなり実現
○	大きい	ほぼ実現
△	やや有り	やや実現
×	全く無し	全く実施せず

3.2.3.3 知識獲得フェイズの特徴

- ・優れたドメインエキスパートの存在
 - ー ドメインエキスパートは、浅いヒューリスティックスをもっているだけでなく、高炉プロセスの挙動に関する解析的アプローチにも長けていた。
- ・ドメインに精通したKEの存在
 - ー 知識をまとめる知識技術者の立場の人でも高炉操業に精通しており、エキスパートの知識を引き出すに十分な前提知識をもっていた。
- ・操業オペレータにわかりやすい知識表現方法の考案
 - ー 操業オペレータにも理解し易いよう日本語表現を導入した。

3.2.3.4 望まれる知識獲得支援機能

(1)知識収集段階

：どのような知識があるかもわかっていない白紙の段階であり、暗中模索のフェイズである。

<望まれる機能>

- ・ 思い付く操業ノウハウを思うままに記述できる機能
(「文章」や「記号」を併用できるとよい)

(2)知識整理段階

：収集した知識から、知識表現方法（ドメインエキスパートの考え方に近い表現）を整理しながら考案するフェイズである。

（必要とあらば、「知識入力・追加・修正エディタ」の設計を検討する段階である。）

<望まれる機能>

- ・ 知識を切り貼りできる機能
- ・ 表やグラフなどをつくれる機能
- ・ 在来の知識入力エディタがもつ知識表現を利用し、使いやすさ等を比較できる機能

(3)知識のインプリメント段階

：「知識入力・追加・修正エディタ」等を用いて、知識を入力するフェイズである。

<望まれる機能>

- ・ 入力を促すためのアドバイスや質問を効率よく発する機能
- ・ ドメインエキスパートの知識表現に適した知識記述用エディタ

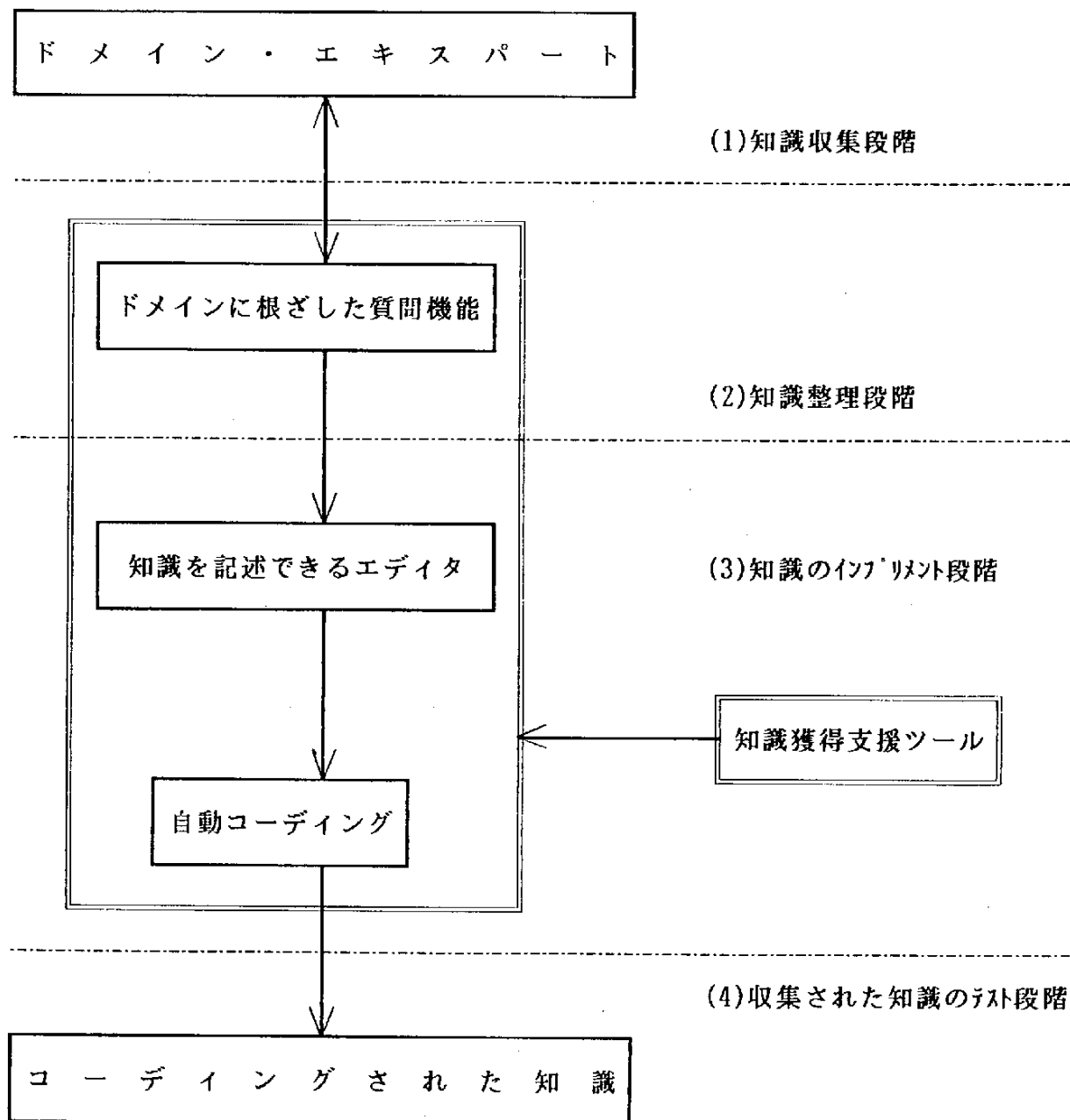
(4)収集された知識のテスト段階

：インプリメントされた知識をテストするフェイズである。

<望まれる機能>

- ・ テスト結果を統計的に分析して、矛盾箇所等を指摘する機能
- ・ 追加、修正時に該当箇所がすばやく提示される機能
- ・ 知識の入力・追加・修正モードに簡単に設定可能な機能

3.2.3.5 理想的な知識獲得支援ツールのイメージ



(参考文献)

- [湯井 87] 湯井ほか, "高炉プロセスの操業監視支援における知識システムの適用" 計測と制御, 26巻8号, 712~719 (1987)
- [小林 86] 小林重信, 「知識工学」, 昭晃堂 (1986)

(3.2.3 担当 森谷)

3.3 計画・設計型問題

3.3.1 シミュレータのユーザインタフェース設計

3.3.1.1 事例ES

シミュレータのためのユーザインタフェースの設計を支援するシステム。

3.3.1.2 システム概要

コンピュータ上に作らるシミュレータの動作をアニメーション等で視覚化し、パラメータの初期値設定、実行中におけるユーザインタラクション、パラメータ値のモニタリングなどをサポートするためのユーザインタフェースの設計を支援する。

ハードウェアとして表示のためのビットマップディスプレイ、入力のためのキーボードとポインティングデバイスとしてのマウスを扱う。

設計の対象となるシステムの部品として、ウィンドウ、メニュー、マウス、キーボード、グラフ、アイコン、文字、表がある。これらを表示およびマンマシンインタラクションのシナリオによって決定される構造に従って組み立てる。

3.3.1.3 典型的タスクと問題可決機能との関係

	設 計 問 題	関 連	実 現	事 例 と の 関 連
定 義	システムの入出力の要求仕様が与えられたとき、これを実現するために、構成要素を組合せ、かつ、各構成要素の内部仕様を決定	○	○	入出力はエンドユーザとのインタラクションである。入力方式と出力方式を可能な部品の組合せの中から選択して行く。要求仕様は固定的なものではなく、実現不能な場合は変更を促すことも必要。
特	1. システム構造の解空間 2. 解析による合成	△ △	△ △	広いが、要求仕様によって大幅に縮小する。 制約条件から必然的に決る要素も少しある。

徴	3. 構成要素の最適化	△	△	最適性の評価基準があいまい。
	4. 検証の完全性と効率性	△	×	満足すべき制約条件が緩い。
	5. 対話形式での支援	○	○	設計がそのまま製品となるので有効。
	6. 段階に応じた支援形態	○	○	設計がそのまま製品となるので有効。
基本 タ ス ク	1. 構成要素とその関連の表現	○	○	固定的。要素の属性から決定されている。
	2. 階層的表現	○	○	表示要素:part-of階層 パラメータ:is-a階層
	3. 代替案の自動生成	○	○	各項目についての選択肢は少数かつ既知。
	4. 部分システムの評価	○	△	シーン毎、パラメータ毎に。
	5. 上位レベルへの知的後戻り	△	△	制約条件が緩いので少ない。
	6. 事例の検索と利用	○	×	部品に関する事例。実現は難しい。
	7. 並列的問題解決	△	×	質問応答に依存しているため少ない。
問 題 解 決 機 能	1. トップダウン精密化戦略	○	○	要求仕様の獲得の流れにそって精密化する。
	2. 拘束最小化戦略	×	×	
	3. 最適化機能	×	×	
	4. 検証機能	△	×	
	5. 仮説推論による知的検索	△	×	
	6. 類推による事例の利用	△	×	
	7. 協調型推論	○	△	知識は分れているがK Sに分けていない。

3.3.1.4 知識獲得支援フェーズでの手法・特徴

(a) システムの要求仕様の整理

設計対象の完成例を思い浮べる。対象物の「構造」,「部品」,「パラメータ」を抽出す。ここでは、構造はインタラクションのシナリオ、部品はウィンドウ、メニュー、マウス、キーボード、グラフ、アイコン、文字、表など、パラメータはウィンドウの大きさ、位置や待ち時間などである。

例を念頭に置きながら人間の設計プロセスを考える。馴染みやすい例題を考えると、対象システムのイメージが湧きやすい。例題をもとに、一般的な知識も引出せ

る。おおまかな骨組を決めるために、登場するオブジェクトとそのパラメータ、およびそれらの間の関係を列挙する。

パラメータについて考察する。それぞれのタイプおよび依存関係について整理する。設計対象システムの実際の運用場面を考える。ここではマンマシン・インタラクションのシナリオである。

(b) 経験則の収集

部品選択のヒューリスティクスを集める。部品の種類の選択について強い制約はない。あるのは推薦。部品のパラメータについては大きさなどについて全体に対する制約がある。

(c) システムの構造設計

全体の制御を考える。部品間の相互依存性が弱いので、原則的にトップダウン精密化でよい。外側の制御は単純なので繰返しを含む手続き記述を用いるのがよい。

(d) 表現方式の選択

部品やパラメータはフレーム、制約条件は論理式、ヒューリスティクスはルールで表現する。

3.3.1.5 望まれる知識獲得支援機能

(a) 知識獲得段階

次のタスクについての支援が必要である。1. 要求仕様の取得。2. 部品についての知識の収集。3. 対象システムについての機能仕様の詳細化。4. 構造仕様へ変換。5. 構造仕様の詳細化。6. 部品間の依存性チェック。

(b) 知識変換段階

マルチパラダイムのプログラミング環境が必要。

記述の一貫性の管理機能が必要。

⑥ 理想的な知識獲得支援ツールのイメージ

(a)全体として、「高度な Programmer's Apprentice」のようなもの。

システムは賢い生徒であって欲しい。気の利いた質問をする。先生のやるきを促す。自分で調べられることは自分で調べる。先生が知らないようなら報告する。

(b)人間KEの能力 + 知識集積制約の自動チェック

(柔軟性) (正確さ, 厳密さ)

(c)ユーザに知識表現, 制御戦略のパラダイムを教えるCAIを含むもの。

ユーザの要求を明確化させる。

(d)ユーザが思いついた時点で直ちにその知識を入力できること。

相互主動型のインタラクション

(e)対象領域についての基本的な百科事典的知識をもつ。

目標ESの中でコンパイルして使う。

(f)人間エキスパートの知識を写すような分野では、

典型的な例題の解法プロセスの例を適当な抽象レベルで入力する。個々の項目について、この例に特異なものか、一般的なものかという情報も。エキスパートが「なんとなく分っている」知識をうまく引出す。

(g)知識の不完全さへの対処法について

矛盾, 冗長性, 知識の不足を指摘し, 修正などの従来技術に加え, 修正方法についてのガイド, あるいは(半)自動修正を行なうとよい。このとき, 修正せずに, うまく誤魔化すという選択肢も選べるとよい。たとえば, システムが適当な修正を行ない, それがシステムの責任であることを憶えておく。

3. 3. 2. 連接水系の翌日発電計画システム

3. 3. 2. 1. 対象システム： 連接水系の翌日発電計画システム

3. 3. 2. 2. システムの概要：

ひとつの河川にそって複数の発電所が連なる連接水系では、限られた総使用水量を発電所間でたくみに融通しながら、需要にあわせた発電を行わなければならない。

実際の連接水系の運用にあたっては、長期的・季節的にはさまざまな数理計画的手法を利用して運用計画を立案する。そして、短期的には、前日に発電計画（翌日発電計画）を作成し、当日この計画に基づいて運転を行い、さらに必要に応じて当日計画を修正するという方式がとられている。

このような短期的な計画立案は運転担当者の経験に基づいてなされることが多い。そのためには次のような広範な専門知識が必要である。(1)多数の小規模な調整池への出水予想、(2)季節調整用貯水池の運用、(3)責任放流などの河川特有の運用制約、(4)需給運用。特に、連接水系の合理的な運用を左右する要素のひとつに発電可能水量の的確な把握がある。

ところが、この作業は河川の自然流量を予測するという複雑な問題を含む。すなわち、自然流量は流域の降水量・積雪量によって変わり、しかも同じ流域であっても土壌の湿潤度によっても変化する。このため、発電計画を策定できる技術者を養成するには長期間の教育と経験とが必要とされる。

そこで、我々は、知識処理技術を適用し、翌日発電計画を知識システム化することを試みた。これは、当所における連接水系運用に関するノウハウを利用し、数値計算プログラムと組合せた複合的なシステムとして試作したものである。

図3. 3. 2-1に本エキスパートシステムの全体構成を示す。発電計画立案に必要な知識を格納する知識ベースはルールとフレームとに分れる。ルールには、翌日の溪流予想のための知識と季節調整用貯水池の発電放流量予測知識とがあり、いずれも定性的な情報が記述されている。ルールは適当な単位ごとに知識源の形でグループ化されている。

一方、フレームは定量的な情報を持ち、各ダムの諸元データ、溪流変化量の分類テーブル（季節的な気温・雨量など）、積雪時の溪流予想テーブル、出水統計データ（月ごとの

放流限界など) から構成される。

システムは次の順序で発電計画をたてる。(1)当日の天候、気温、出水量から各ダムの翌日の渓流量を予測する。(2)当日の需要と出水状況から貯水池発電所の翌日発電放流量を決定する。(3)各発電所の翌日発電放流量を10分きざみの発電出力値に分解し、24時間分のスケジュールを作成する。

本エキスパートシステムの特徴は(1)発電計画に必要な知識を定性的なものと定量的なものとの明確に分離して保持しておき、(2)両者を動的に組合せて計画立案を行っていること、ならびに、(3)24時間のスケジュールを作成するのに既存のプログラムを利用し、(4)高速の数値計算機能と知識処理技術の両方の長所を取入れていることにある。

システムには、大型計算機上で富士通のESHELLを利用して知識処理部分を開発し、既存のFORTRANプログラムと組合せている。知識処理部分ではブラックボード概念に基づく前向き推論方式を採用したが、ブラックボードは単純な作業記憶として使用している。なお、知識処理部分についてはSymbolics 3670上でKEE-2を利用して作成した簡約バージョンも存在する。(図3.3.2-2.)

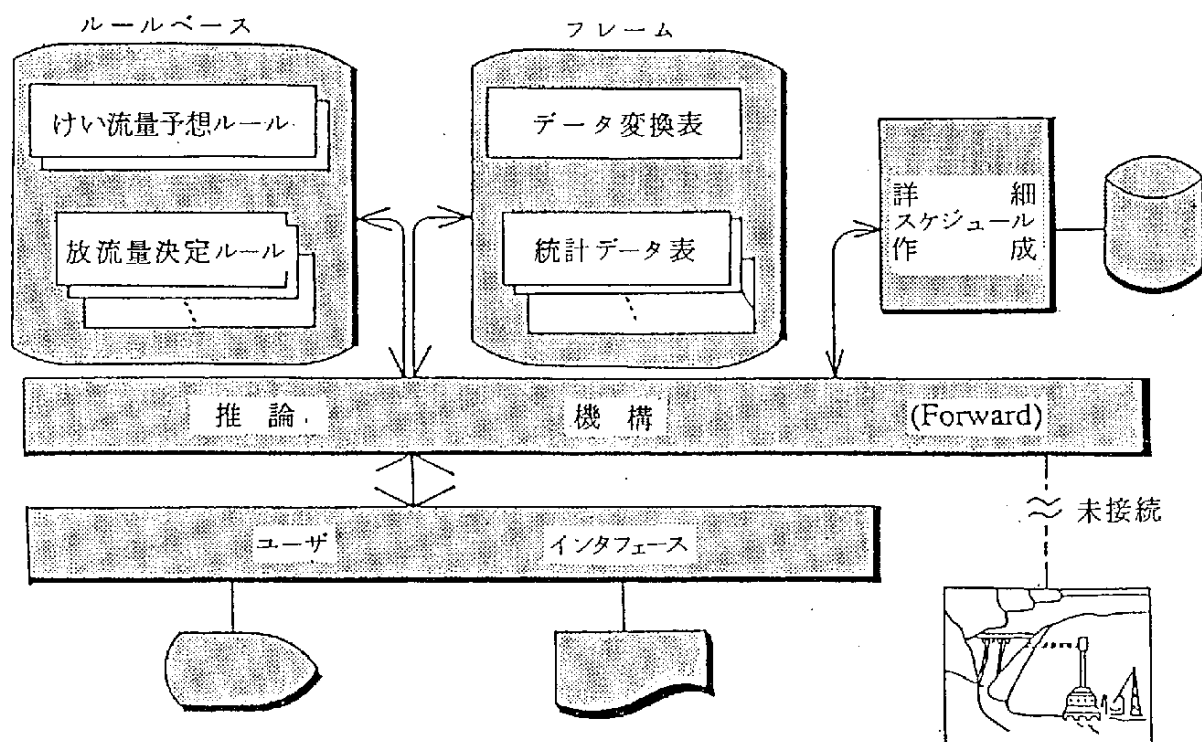


図3.3.2-1 接続水系の翌日発電計画システムの全体構成

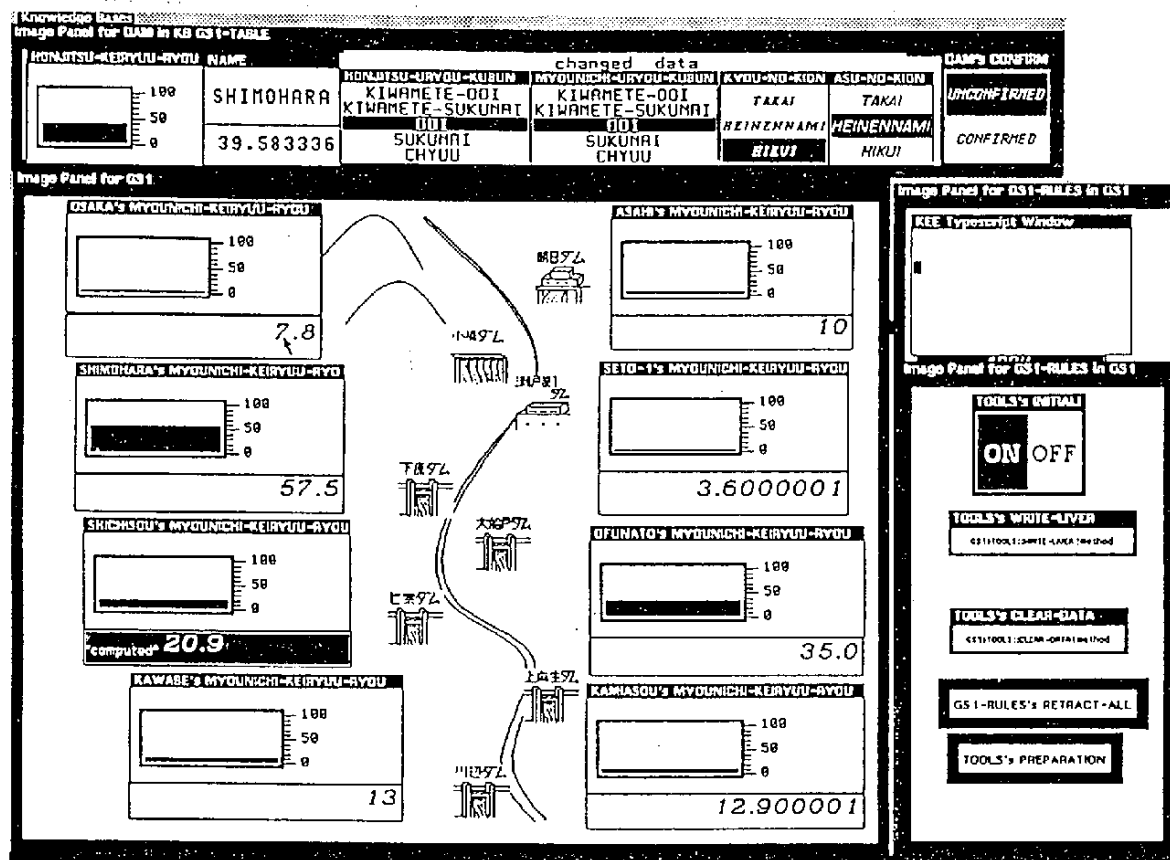


図 3. 3. 2 - 2. KEEバージョンの利用者インタフェース

3. 3. 2. 3. 類型的タスクと問題解決機能との関連

	計 画 問 題	関 連	実 現	事例との関連
定 義	与えられた目標を達成するために利用可能な資源の割当てを時系列として決定する。	○	△	最終目的とする機能はこのとおりだが本システムでは無視している項目が多い。

特 徴	1. 探索空間の広さ	○	△	最適化をめざさず許容解を1つ導く、
	2. 評価属性の多様性	○	○	経済性・安全性・信頼性・余裕度など
	3. 環境予測が不確実	○	○	経験則・統計データの利用
	4. 部分計画間の相互作用	○	○	個別の計画を足合わせると全体になる 互に関連性が強い
	5. 既存計画の再利用	○	X	前提条件の変化に対応するには必要
	6. 費用／便益	X	X	既存制御設備で対応は可能
	7. 対話型支援	X	X	十分な説明機能は必要
基 本 タ ス ク	1. 計画過程の階層化	○	X	水の流れる方向にしたがった計画順序
	2. 組合せ的探索	○	△	全体でも個々のダムでも必要
	3. 制約条件の相互作用	○	○	水が相互作用を伝播する
	4. 環境の予測	○	△	必要性は強い
	5. 上位レベルへの後戻り	△	△	—
	6. 計画事例の検索	X	X	考慮していない
	7. 計画評価の多様性	○	○	問題の定式化／評価方法の明確化
問 題 解 決 機 能	1. トップダウン精密化	○	○	自然に組込まれている
	2. 拘束最小化	X	X	—
	3. 網羅的探索	X	X	最適解を求めない
	4. 仮説推論	X	X	—
	5. 類推	△	X	将来的には必要
	6. 計画の多目的評価	△	X	将来的には必要
	7. リスク下での意思決定	X	X	—

3. 3. 2. 4. 知識獲得フェーズの特徴

システムの開発にあたっては、まず、(財)電力中央研究所の専門家(1人)が、対象とする連接水系に関する経験に基づいて知識を非定型的な形で記述した。それを、ツールの書式にあわせて書き直す作業はプログラマが行った。知識の正当性はシステムを動かしながら確認した。

システムの性能を向上させるためには、既存の計画プログラムの存在が大きい。このプログラムはFORTRANで記述されたものであるが、その開発にあたっては2～3年にわたって、現場での調査、ヒアリングの繰り返しを行っている。その意味では本プログラムの開発には知識システム風のアプローチがなされていると言えよう。

収集した知識は、水系共通の知識と個々のダムごとの知識とに分離して格納した。前者には、各構成要素の接続関係、天候に関する知識、河川制約などが含まれ、後者には、各ダムの容量、放流戦略、環境条件などが含まれている。

水力発電の分野は、一般のエンジニアリングシステムの分野と異なり、自然現象を考慮しなければならない部分が多いので、経験則を活用できる知識処理技術の有望な分野である。そこで使われる問題解決知識は、First Principlesの獲得の難しさから経験則を重視したものとなり、計画立案手法としてはアドホックで未成熟である。

しかし、十分な分析ができれば数理計画法による定式化も可能ではないかと思われる。本システムのような知識システムの開発によって、数理計画問題と知識処理技術との関連を明確にし、さらには、たとえば定性的推論の手法をシステムに組み込むことができればより高度かつ柔軟なシステムの実現も可能と思われる。

3. 3. 2. 5. 望まれる知識獲得支援機能

(1) 知識の収集段階

このような分野では、問題を全体レベルで捉えて知識を収集する段階と、より詳細な部分問題レベルで知識を収集する段階とを分離することが望ましい。

全体レベルでは、意思決定者の知識を数理計画法的なセンスで整理できる機能が必要である。そのためには、対象システムの構成記述を容易にする知的CADシステム、また、

計画過程で生ずる問題を分解し再構成する階層的な整理機能、(複数の)目的関数を整理・明確化する機能などが重要である。さらには、専門家モデルの初期作成を容易にするために、知識ベースの原型(プロトタイプ)を備えていることが要求される。

部分問題レベルでの担当者の知識は、経験則をそのまま利用することが適切で、たとえば、AQUINAS/ETS風のアプローチが有効と考えられる。その中でも特に複数の担当者の知識を順次提示し、その間の調整を可能とする機能が役立つと考えられる。

(2) 知識の変換段階

ETSに見られるような心理学に基づくアプローチや、MOREに見られるインタビュー戦略などを用いて、経験則の抽出しやすくする機能が重要である。さらには、ルールと数値情報(統計データなど)とを関連づける機能や、計画問題に対する宣言的な知識記述を数理計画などの形式に変換する機能も必要となる。

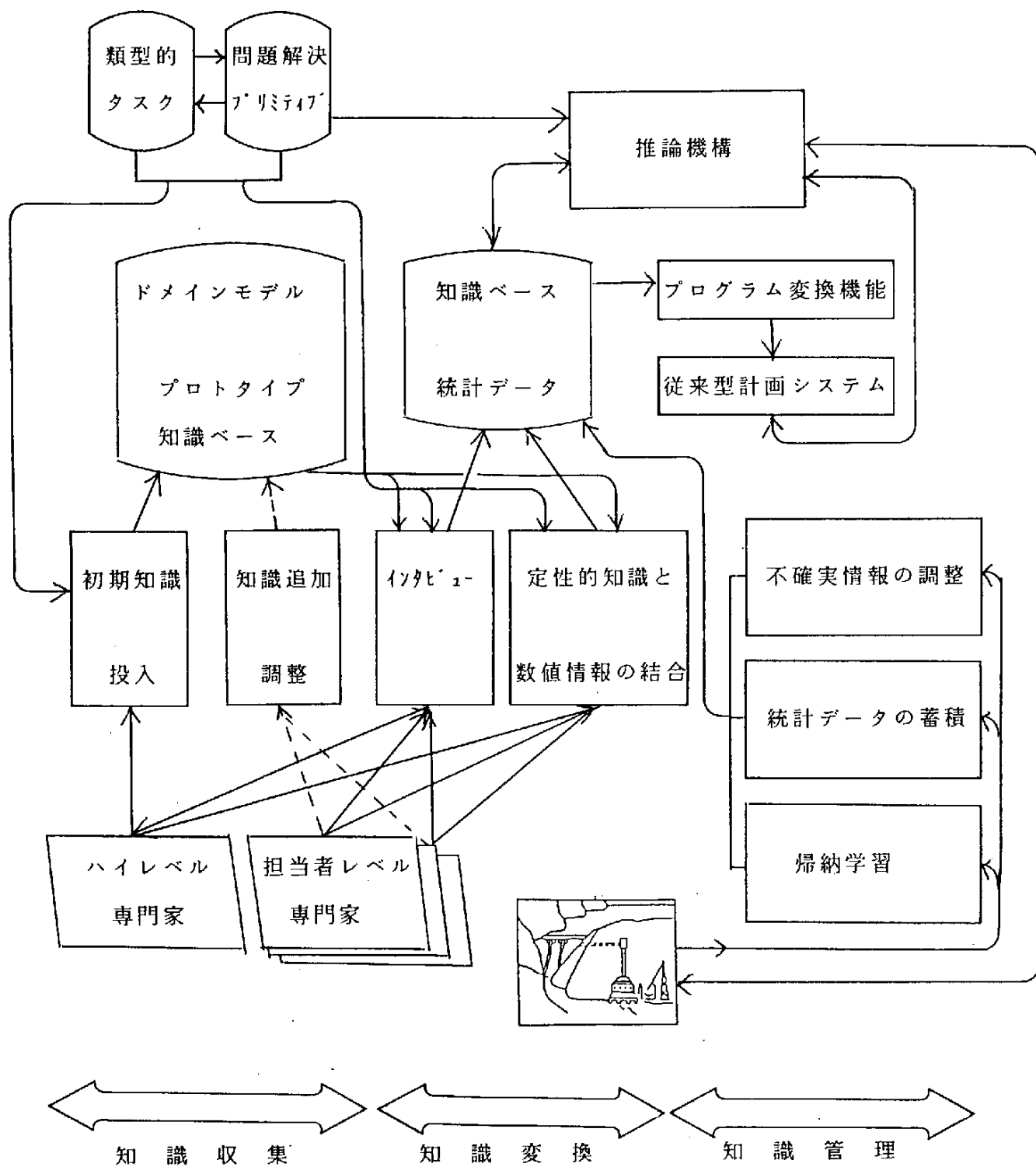
(3) 知識の管理段階

一般に、CFやFuzzyの概念で表される、経験則に含まれる不確実な情報(値)を自動的にチューニングする機能、統計データなどの蓄積機能、実施例・適用例を利用した帰納学習機能などが必要となる。

以上をまとめると本問題に適した理想的な知識獲得支援ツールのイメージは次ページのようになる。

(3.3.2 担当 寺野)

3. 3. 2. 6. 理想的な知識獲得支援ツールのイメージ



3.3.3 ルート計画エキスパートシステム

3.3.3.1 対象システム

ルート計画エキスパートシステム

3.3.3.2 システム概要

輸送手順と資源の割当て問題

3.3.3.3 典型的タスクと問題解決機能との関連

	計 画 問 題	関 連	実 現	事例との関連
定 義	与えられた目標を達成するために利用可能な資源の割当てを時系列として決定する	○	△	資源の割当ての決定に加えて，ルートの設定も行う
特 徴	1. 探索空間の広さ 2. 評価属性の多様性 3. 環境予測が不確実 4. 部分計画間の相互作用 5. 既存計画の再利用 6. 費用／便益 7. 対話型支援	○ ○ ○ ○ ○ ○ ×	○ ○ ○ ○ × △ ×	準最適解を導く 経済性，余裕度，資源使用効率など 交通事情，天候など 部分計画間での同一資源の利用により相互作用が発生 探索効率の向上につながる 専門家以上に評価属性を考慮した解を生成 説明機能は必要
基 本 タ ス ク	1. 計画過程の階層化 2. 組合せ的探索 3. 制約条件の相互作用 4. 環境の予測 5. 上位レベルへの後戻り	× ○ ○ ○ ×	× △ ○ △ ×	抽象－具体という概念を持たない 組合せ的爆発を防ぐ探索方法を用いている 同一資源に関する制約間に相互作用が存在 統計データの利用（固定的） －

問 題 解 決 機 能	6. 計画事例の検索	△	×	「既存計画の再利用」の実現には必要
	7. 計画評価の多様性	○	△	評価方法の明確化が必要
	1. トップダウン精密化	×	×	—
	2. 拘束最小化	×	×	—
	3. 網羅的探索	○	△	探索空間を絞っている
	4. 仮説推論	△	×	将来的には必要
	5. 類推	△	×	将来的には必要
	6. 計画の多目的評価	×	×	—
	7. リスク下での意思決定	△	×	—

3. 3. 3. 4 知識獲得フェーズの特徴

専門家が解の最適性まで考慮していない場合には、エキスパートシステムにはより最適な解の導出が要求される。このとき、知識の抽出について3つの知識獲得フェーズが存在すると考えられる。

(1) 専門家と同等の解を求めるための知識獲得フェーズ

専門家が通常用いている知識を獲得するフェーズ。この知識を用いることにより、専門家が導出するのと同じ解を求めることができる。

(2) より最適な解を求めるための知識獲得フェーズ

より最適な解を求めるためには、一般に評価関数の導入が行われ、いくつかの解候補の中から最も評価値の高いものが解として選ばれる。この場合の知識獲得フェーズでは、まず評価関数の設定のために必要な知識の獲得が行われる。次に、評価値の高い解候補を導き出すための知識を専門家から抽出する。ただし、専門家がこの知識を持っているとは限らない。

(3) 新しい解法を求めるための知識獲得フェーズ

専門家自身は、評価関数による最適解の導出を行っていないため、本当の最適解を求め

る手法は、専門家の手法とは異なる新しい手法かも知れない。KEがこの新しい手法を考案する際に、専門家から参考となる知識を獲得するフェーズが考えられる。ただし、この場合も専門家が必ずしもこの知識を持っているとは限らない。

3.3.3.5 望まれる知識獲得支援機能

3.3.3.5.1 知識の抽出段階

(1) 専門家と同等の解を求めるための知識獲得フェーズ

①経験則の抽出機能

(a) インタビュー機能

専門家が持っている経験則をインタビューによって抽出する。

(b) 推論過程からの抽出機能

専門家の実際の推論過程から、そこに用いられている経験則を抽出する。

(2) より最適な解を求めるための知識獲得フェーズ

①情報収集機能（評価関数設定のため）

(a) インタビュー機能

評価すべき項目等を問い合わせる。

(b) 情報整理機能

インタビューによって得られた情報をKEが整理するための環境。

②専門家の知識の明確化機能

(a) 専門家が導出した解の評価機能

専門家が、評価値の高い解候補を導出するための知識を持っているかどうかを判定する。

(b) 専門家の解法及び解の分析機能

専門家の推論過程や解の中から、評価値の高い解候補を導出するための知識を抽出する。

(3) 新しい解法を求めるための知識獲得フェーズ

①知識整理機能

専門家から得た情報・アイデア等を整理する機能。

3.3.3.5.2 知識の変換段階

(1) ルールの変換機能

抽出された知識をルール形式に正確に変換する機能。

(2) ルールの一般化・特殊化機能

3.3.3.5.3 知識の管理段階

(1) 評価関数のチューニング機能

推論過程等に基づいて、評価関数の各評価因子への重み付けを自動的に調整する。

(2) 知識ベースの洗練化機能

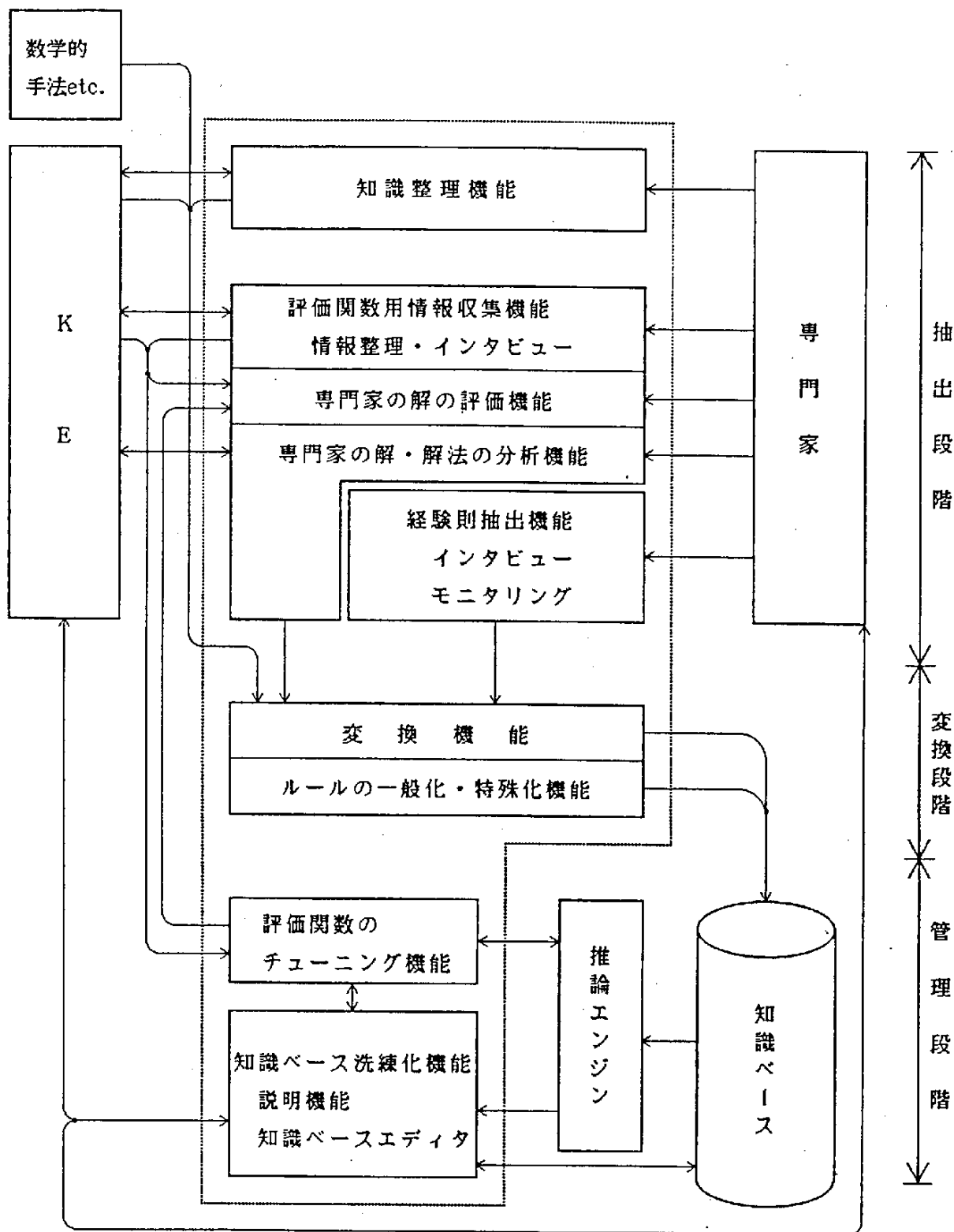
①説明機能

推論失敗要因の説明を行う機能。

②知識ベースエディタ

ルールの修正を行う環境。

3.3.3.6 知識獲得支援ツールのイメージ



(3.3.3 担当 福島)

3.3.4 電子装置の実装設計支援システム

(1) システム概要

実装構造を表現するパラメータを明確にし、このパラメータの相互関係、実現可能な制約条件、評価関数をもとに最適な実装構造を得るシステムである。〔陣野77〕

最近の電子装置の実装構造には、LSI技術の適用により従来にない新しい実装技術や実装構造が採用されつつある。LSI技術を生かしシステム・アーキテクチャの変革に対処するためには、実装に関する個々の技術の開発とともに、これらの技術を最適に組み合わせて装置を実現する、装置構造の最適化構成の技術が不可欠である。また、新しい装置の開発には実装技術が深くかかっており、その開発初期の段階から、実装構造についても平行に検討することが必要で、開発初期段階での適用実装技術・実装構成の方向付けが重要である。これらを、計算機との対話で支援するのが本システムである。

(2) 典型的タスクと問題解決機能との関係

	設 計 問 題	関 連	実 現	事 例 と の 関 連
定 義	システムの入出力の要求仕様が与えられたとき、これを実現するために、構成要素を組合せ、かつ、各構成要素の内部仕様を決定	○	△	設計開始時点では入力に未定義な項目がある。設計段階に応じ決定した項目を順次入力（時系列で具体化）
特 徴	1. システム構造の解空間 2. 解析による合成 3. 構成要素の最適化 4. 検証の完全性と効率性 5. 対話形式での支援 6. 段階に応じた支援形態	○ × ○ × ○ ○	○ × △ × △ ○	解空間を連続量による領域として表現 従来は存在しない新しい構成を期待 準最適化 検証は定性的にしかできない 属性モデルで3次元構造を表現（低速） 設計段階ごとに決定した具体事項を入力

基本タスク	1.構成要素とその関連の表現	○	△	属性モデルで3次元構造を表現
	2.階層的表現	○	○	接続形態・包含・上下関係等
	3.代替案の自動生成	○	○	解空間を連続量による領域として表現
	4.部分システムの評価	○	○	各実装階層ごとに準最適化
	5.上位レベルへの知的後戻り	○	△	ほぼ全空間探索（一般的絞り込み法未）
	6.事例の検索と利用	○	×	文書レベルで検索
	7.並列的問題解決	×	×	—
問題解決機能	1.トップダウン精密化戦略	○	△	入力に未定義な項目（設計完時点で可）
	2.拘束最小化戦略	×	×	—
	3.最適化機能	○	△	準最適化
	4.検証機能	×	×	—
	5.仮説推論による知的検索	○	×	—
	6.類推による事例の利用	○	×	—
	7.協調型推論	○	×	—

(3) 知識獲得フェーズでの手法

以下の4段階で知識獲得を実施した。

(a)知識源の整理段階

(i) 専門家の頭の中から問題解決知識を獲得する。

(ii) 設計結果のドキュメント（why の記述がない）から設計対象を明確にする。

(b)実装階層構造の整理段階

設計対象をモデル化するために設計に関連するパラメータ群を明確にした。

(i) 実装構造： IC／プリント板／モジュール／装置／ユニット 等

(ii) 制約条件： 製造可能性／コスト／他装置への流用性／信頼性 等

(c)知識表現・推論方式の選択段階

(i) 3次元構造物は属性モデルで表現した。（フレーム）

(ii) 構造物間の接続関係はネットワークで表現した。（フレーム）

(iii) 制約条件は連続量として取扱った。

(iv) 未定義入力項目を許容した推論方式を採用した。(制約条件は最小にする)

(d)知識の検証・経験知識の追加・拡充段階

(i) 設計の詳細化段階ごとに決定事項を入力して制約条件として明確化した。

(ii) 具体的な設計事例を入力して結果を検証した。

(iii) 結果を得る推論過程のトレースや具体的な新規設計事例をもとに経験知識を追加し、設計結果の精度向上、推論時間の短縮化を図った。

(4) 望まれる知識獲得支援機能

(a)知識獲得段階

(i) 知識表現が豊富であること

事例では、専門家が直観できる3次元物体構造、2次元図形、等を簡単に表現する必要があった。具体的には、装置の機能・構造を表現する「深い知識」の表現、また、解空間の絞り込みを行うための「浅い知識」の表現である。

(ii) 新しいモデルの追加容易であること

事例では、将来、仮説推論や類推等を追加しないと問題解決能力が低い。

(b)知識変換段階

(i) 事例に基づく学習・説明機能

具体的な設計業務では、設計結果のドキュメントのみで、why (検討経過) が殆ど残っていない。そこで、具体的な設計事例をもとにした説明機能や、設計事例をもとにした学習機能が必要となる。

(ii) 知識の生成機能

設計分野では可能性のある解の空間が非常に広いため、制約条件が少ないと(未定義事項が多いと)推論に膨大な時間がかかる。これに対処するため、具体的には、構造知識から浅い知識を生成して、同一条件での再実行の高速化も必要となる。

(c)知識管理段階

(i) 既存知識ライブラリの利用が簡単であること

設計分野では、既存の設計手法を一部手直しして新しい設計手法を次々に作り出している。これに対処するため、分野毎の専門家の思考のモデル、該分野での分類済の教科書的な知識、分野間での分類済の共通知識等を知識のライブラリとして再利用可能とする必要がある。

(ii) 知識の整合性管理が柔軟であること

本事例では、設計開始時点では未定義な項目が多く、具体的には何が不足しているのか、また、どの知識が矛盾しているかの表示が不可欠である。なお、不足や矛盾はこれらが充足されれば直ちに解消されずに、不足や矛盾の状態をそのまま保持しておき、最後に人手で解消することも必要である。

(d)全体

(i) 知識獲得フェーズ管理が簡単であること

具体的には、フェーズ毎に必要な項目、フェーズ終了段階の判定項目等の管理をいう。

(ii) マンマシンインタフェースが良いこと

具体的な項目として、知識の内部表現をユーザへ見せない、図形表示やオンラインヘルプが拡充していること、設計は会話処理が主体となることから、応答が早いこと、極力自然言語で入出力可能であること、マンマシンインタフェースのユーザ拡張容易であること、等である。

(e)その他のコメント

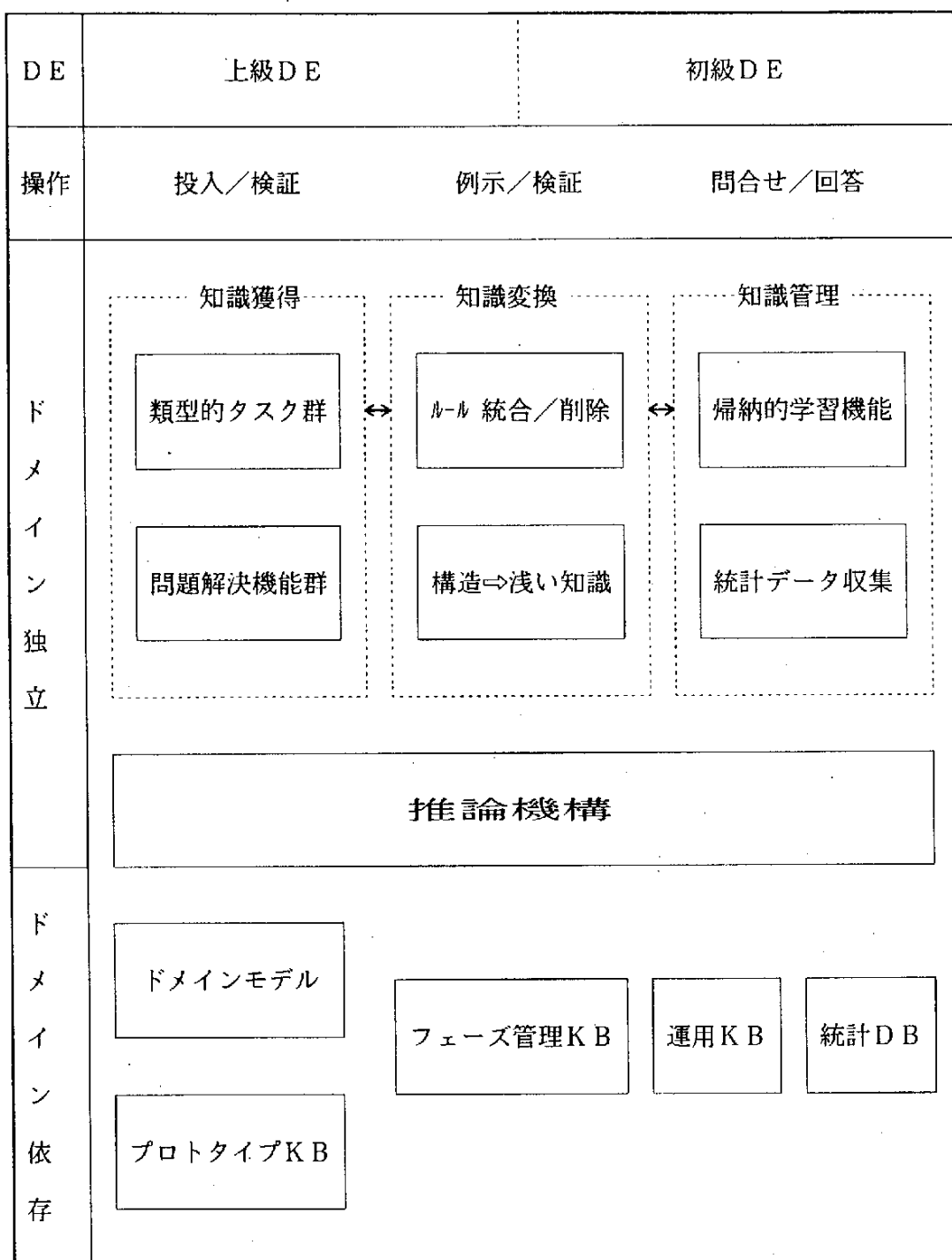
(i) 図形パターン認識機能について

設計分野では、専門家は人間の持つパターン認識の能力をフルに活用しており、大部分の業務を直観でいとも簡単に解決しており、この機能を計算機に持たせることは、この分野での今後の大きな課題である。

(ii) 学習機能について

専門家は、知識を獲得しつつ業務を遂行しており、同一条件でも、設計時点があとになれば、より良い結果を短期間で得ることが可能となる。これはいわば、「知識の学習の戦略」に関する知識を活用して業務を遂行しているとも考えられ、領域知識は単に手段となっている場合もある。こちらもこの分野での今後の大きな課題である。

(5) 理想的な知識獲得支援ツールのイメージ：(E S利用の観点から)



〔参考文献〕

〔陣野 77〕 陣野，桃井，原田「電子装置の実装構造シミュレーション」昭和52年
信学会総全大No. 125(1977).

〔3.3.4 節担当 桃井〕

4 . 結 論

4. 結 論

本報告では、知識獲得作業を知識の収集・知識の変換・知識の管理の3つの段階に分類し、知識獲得支援ツールの目的・特徴とその研究開発動向とを代表的な事例に即してとりまとめた。まず、2章において、代表的な知識獲得支援ツールの特性について文献調査とヒアリング調査との結果を述べ、ついで、3章において、ワーキンググループ各委員がみずから開発に携わった知識システム事例を分析し、それぞれの対象問題に即した理想的な知識獲得支援ツールのイメージを提案した。

1988年11月に開催された第5世代コンピュータに関する国際会議録において、Boose, J.H. は、知識獲得支援ツールの研究動向について報告を行ない、関連する文献の膨大なリストを掲載している[Boose 1988]。その報告では、知識獲得支援ツールの研究戦略として次の4つを示した。①対象問題のタスクと問題解決手法との関連において知識獲得手法を開発する；②ひとつのタスクに注目してその知識獲得手法の研究を行ないその適用範囲を広げていく；③タスクと問題解決手法とを定義し記述しやすい言語を開発する；④知識技術者が使いやすい知識エディタを開発する。

我々の調査研究のアプローチはこのうち②にもっとも近い。これは、与えられた時間的・人的制約のもとでは、このようなアプローチがもっとも適切と考えたためである。その結果、なお、不十分な面は多いものの、知識獲得支援ツールの研究開発の現状と今後の動向については大きな知見が得られたものと考えている。本報告書の結論は、つぎのようにまとめられる。

”人工知能研究の成果としての知識獲得支援ツールは、実験室レベルのごく小さな範囲の分析型問題を対象とする限りにおいて、知識技術者ぬきの知識システム開発を可能とするレベルに到達している。しかし、現実の知識システム開発に現在の知識獲得支援ツールを適用するのは非常に困難であり、今後のいっそうの研究開発を必要とする。”

知識システムにおいて使用される知識表現は、推論機構が推論を実行するための表現であり、人間がそのような形に知識を整理する過程に対しての配慮というのはほとんどされていない。このため、実際の問題にあわせて知識ベースを作成する場合には、与えられた

問題をしばしば強引に人工知能の知識表現に整理・変換することが必要となる。

このような作業は、対象が小規模のものであれば良いが、大規模で多方面にわたる場合には、その全体像を捉えることが難しい。知識獲得支援ツールの研究は、この両者のギャップを埋めるための手段と方法論の開発をめざして進められてきた。今後は、人間の知識の整理法・獲得方法に近い形で作業を進め、最終的に、推論機構向けの知識表現を得るような高度な知識獲得支援ツールの開発が望まれる。そのための課題としては次の4点が考えられる。

1) 大規模知識ベースの開発

知識獲得作業を効率化するためには、現在のところ、かなりの規模の初期知識ベースが必要である。このような目的の知識ベースを、対象分野に依存しないように開発するには非常な困難が伴う。その中で、百科事典の内容をすべてフレーム形式に基づく知識表現で知識ベース化し、“常識”をそなえた知識ベースを実現しようとする、CYCプロジェクトは注目に値する。

2) 問題解決に必要な基本機能の同定と支援ツールの開発

現在のルールやフレームといった低いレベルの概念から、知識システム開発に必要な基本機能のレベルを高度化し、さまざまな分野の問題に共通に利用できる問題解決機能を定める方法が考えられる。これにそった提案が、類型的タスクの概念である。現在、Chandrasekaranの提案している類型的タスクは、分析型問題と単純な設計問題しか適用できないものであるが、これをより広い範囲の問題に適用できるように拡張するとともに、それを支援するツール群を開発することが重要である。

3) 知識システム開発方法論の整備

ツールの適用範囲は、開発方法論の整備によって、明確になり容易に利用できるものとなる。その意味では、現在のいわゆる知識システム開発ツールや知識獲得支援ツールには、開発方法論の裏付けのないものが多い。開発方法論を整備するためには、知識システムの開発プロセスをソフトウェア工学の視点から吟味し、その結果を踏まえた支援ツールを開発することが重要である。

4) システム工学的手法と知識システム手法の統合

システム工学は従来から大規模かつ複雑な対象を分析する上で有用な手法を提供してきた。ところが、そこで使われる手法は、基本的に整理された結果を人間が見て利用するためのものであるため、そのままでは、知識ベース開発に利用できない。今後は、これらの手法を発展・改良して、現在の知識システム開発手法と統合することが重要となろう。

参 考 文 献

- [Boose 1988] Boose, J.H.: Knowledge Acquisition Techniques and Tools: Current Research Strategies and Approaches. Proc. Int. Conf. on Fifth Generation Computer Systems 1988, pp. 1221-1235 (1988).

— 禁無断転載 —

平成元年 3 月発行

発行所 財団法人 日本情報処理開発協会

東京都港区芝公園 3 丁目 5 番 8 号

機械振興会館内

電話 (03) 432-9390

63-A003

