

エキスパートシステム評価ガイドラインの 研究開発報告書

平成 2 年 3 月



財団法人 日本情報処理開発協会

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平成元年度に実施した「エキスパートシステム評価ガイドラインの研究開発」の成果の一部をまとめたものです。

序

エキスパートシステムは、知識ベース機能と推論機構により問題解決を図るシステムで、情報処理の高度化を実現するものとして大きな期待が寄せられている。

エキスパートシステムは、処理アルゴリズムが存在しない、あるいは明確でない問題や、条件の組合せの数が膨大で逐一判断処理が行いがたい問題などに適している。即ち従来技術の不得意な機能を補完し、結果として各アプリケーションの質の向上を実現することができる。

しかしながら、知識ベースの正当性や、それを用いての推論処理はブラックボックス性が強く、処理結果に対する信憑性を十分に保証することがきわめて難しい。とくに、原子力や航空管制など非常に厳しい条件のもとに利用される場合、従来のソフトウェアの信頼性問題に加え、エキスパートシステムの推論結果の評価を如何に行うかは、緊急を要する検討課題である。

本プロジェクトは、このような背景のもとに、昭和63年度から3年計画で、システム監査的視点を含め、エキスパートシステムの開発者および利用者の双方の立場から、エキスパートシステム評価のためのガイドラインを提案することを目的として実施している。

2年目にあたる本年度は、内外における技術面での評価方法について、文献および米国への調査員の派遣により調査するとともに、評価ガイドラインの全体の枠組みを中間報告として提案することとした。

ここで提案する枠組みは、基本方針および評価の視点、評価対象システムのプロフィール、システム共通の評価項目、問題領域別評価項目、評価者別の重点評価項目、評価のためのテスト方法などから構成されている。さらに、この枠組みにもとづいて、10事例の調査を行った。本書は、この調査研究成果を取りまとめたものである。

最後に、本プロジェクトの実施に当りご指導ご協力を頂いた、寺野隆雄委員長を始め「エキスパートシステム評価ガイドライン研究委員会」の委員、講師の方々に深甚なる謝意を表するものである。

平成2年3月

エキスパートシステム評価ガイドライン研究委員会名簿

(順不同、敬称略)

委員長	寺野 隆雄	筑波大学大学院 経営システム科学専攻講師
委員	安藤 明	(財)石油産業活性化センター A I 研究室研究員
	石垣 良信	日本アイ・ビー・エム(株) システム・センター 先進テクノロジー担当マネージャ
	宇佐見仁英	富士通(株) システム本部第三システム統括部 科学システム部第二システム課課長
	荻野 徹	三菱電機(株) 情報電子研究所知識処理開発部 エキスパートシステムグループ主幹
	小林 慎一	(株)三菱総合研究所 情報技術開発部情報技術第二室 室長
	末田 直道	(株)東芝 システム・ソフトウェア技術研究所 研究第一部 A I 第2担当研究主務
	桃井 茂晴	日本電信電話(株) N T T 情報通信処理研究所 知識処理研究部主幹研究員
	山崎 知彦	(株)豊田中央研究所 研究7部743 Gグループリーダー
	吉浦 裕	(株)日立製作所 システム開発研究所研究員
	山本 欣子	(財)日本情報処理開発協会常務理事
	鍛冶 勝三	(財)日本情報処理開発協会開発研究室主任研究員
オファサーハ	山口 高平	静岡大学 工学部情報知識工学科助教授
事務局	金剛寺英雄	(財)日本情報処理開発協会開発研究室専任調査役

目 次

1. 総論	1
2. エキスパートシステム評価ガイドライン（中間報告）	7
2.1 エキスパートシステム評価ガイドラインの基本方針及び評価の視点	8
2.2 エキスパートシステムの評価とライフサイクル	15
2.2.1 エキスパートシステム開発のライフサイクルの基本的考え方	15
2.2.2 エキスパートシステム開発のライフサイクル	16
2.2.3 開発フェーズと評価対象	18
2.3 エキスパートシステムのプロフィール	21
2.4 エキスパートシステムの評価項目	24
2.5 エキスパートシステム評価項目の解説	28
2.5.1 知識ベースの妥当性評価	29
2.5.2 問題解決・推論方式の妥当性評価	38
2.5.3 結果の妥当性評価	43
2.5.4 要求仕様に対する達成度評価	48
2.5.5 知識ベースの保守性評価	55
2.5.6 マンマシンインタフェースの妥当性評価	61
2.5.7 システムの総合評価	66
2.6 問題領域別の評価項目	80
2.6.1 設計型システム	84
2.6.2 計画型システム	88
2.6.3 解釈型システム	93
2.6.4 診断型システム	98
2.6.5 制御型システム	103
2.6.6 複合型システム	111
2.7 評価者別の重点評価項目	117
2.8 エキスパートシステムの評価とテスト方法	122
2.8.1 概要：テスト方法の類型化	122
2.8.2 静的テスト／分析	123
2.8.3 動的テスト	128
2.8.4 まとめ	130

3. エキスパートシステム評価ガイドラインに基づく事例調査	133
3.1 エレベータ設計支援エキスパートシステム	134
3.2 電子装置の実装設計支援システム	150
3.3 計算機ハードウェア構成設計支援エキスパートシステム	155
3.4 トッパ―通油計画立案支援システム	164
3.5 潤滑油精製プラントのスケジューリングエキスパートシステム	170
3.6 抄紙生産計画エキスパートシステム	177
3.7 ダムゲート診断評価システム	184
3.8 機械材料選定エキスパートシステム	191
3.9 エンジンエレクトロニクス技術相談システム	199
3.10 知識ベースによるコンピュータ・システム／ ネットワーク自動運用システム (SPARK)	206
4. エキスパートシステム評価の文献調査	215
4.1 エキスパートシステムの失敗の原因	215
4.2 原子力発電所向けエキスパートシステム開発におけるV & V方法論	221
4.3 知識ベースシステムのV & Vアプローチ	228
4.4 エキスパートシステムの検証と妥当性検査	235
4.5 専門家間の概念構造の比較	240
4.6 決定理論を用いた経験的知識の正当性の評価法	244
4.7 エキスパートシステムの妥当性試験手法	250
4.8 エキスパートシステムの検証及び妥当性検査	259
4.9 知識ベースのテスト	265
4.10 エキスパートシステムの性能に対する妥当性検査	271
4.11 KB-Reducer: 知識ベースの非一貫性及び冗長性の検査に対する新アプローチ	276
4.12 Digital社におけるコンフィギュレーションエキスパートシステム	283
4.13 エキスパートシステムのテスト、検証、引渡し －Mentor履歴の事例－	290
4.14 ORにおける視覚対話型モデリング手法：その成功と可能性	295
4.15 エキスパートシステム評価関連文献一覧	300
5. 結論と今後の課題	311
付録1 エキスパートシステム評価のヒアリング調査	315
付録2 「エキスパートシステムの評価技術動向調査」海外出張報告	318
付録2.1 エキスパートシステムの評価技術動向調査（その1）	318
付録2.2 エキスパートシステムの評価技術動向調査（その2）	325

1. 総論

1. 総論

(1) はじめに

知識工学が提唱されてから今日まで、数多くのエキスパートシステムが開発されてきた。これらを通じて、人工知能技術は、従来では、取組む気にもならなかったような問題にも、専門家－知識技術者というペアで挑戦できるような新しいパラダイムを提供してきたといえる[ICOT-JIPDEC 1989]、[AI白書 1989]。また、エキスパートシステムの開発に直接的にかかわってきた人々の間では、人工知能の科学としての成功がようやく実用システムとしての成功に結びついてきたこと、ならびに、人工知能技術を指向したシステムに対してニーズ指向型のシステムが急増していることが認識されるようになった。

しかし、その一方では、十分な問題解決能力を持ちながらも、フィールドテストを経て、実用化の段階に達したエキスパートシステムの数は意外に少ない。この理由としては、さまざまなものが考えられるが、そのひとつは、開発中、あるいは、完成したシステムを客観的な基準で評価するための方法・基準が確立されていないことにある。特に、エキスパートシステム開発方法論と組合せた評価方法の整備が急務であると考えられる。

このような現状認識のもとに、当協会では、1988年度から3年間の予定で、エキスパートシステム評価のフレームワークの確立を目指して、調査研究を行っている[JIPDEC 1989]。本年度は委員数を充実させ、より本格的な検討を開始した。特に、エキスパートシステム評価のガイドラインを提案することを主なねらいとし、月1回程度の頻度で委員会を開催し、その中で、各委員が直接・間接的に開発にたずさわった事例の分析、文献調査、海外の動向調査、ならびに、付録1に示した方々からのヒアリングなど、積極的な調査研究活動を行なった。本報告書は、これら第2年度の活動を取りまとめたものである。

本報告書の構成は次のとおりである。まず、1章の後半の部分で、総論として、エキスパートシステム評価ガイドライン設定に対する本委員会における基本的な考え方を述べる。2章は、本報告書の中心であり、エキスパートシステム評価ガイドラインの中間報告を行う。3章は、このガイドラインをまとめるにあたって、各委員から報告のあった事例分析の結果を述べる。4章は、エキスパートシステム評価に関連する最近の研究動向について文献調査を行った結

果である。5章では、結論をとりまとめ、今後の課題・計画について論ずる。
付録1は、今年度の委員会におけるヒアリング調査の記録であり、付録2には、
同じく今年度実施した海外調査の報告を集録する。

(2) エキスパートシステム評価の視点

エキスパートシステムは、通常、人間の知性を用いなければ解けないような問題を解決するのに、専門家の知識を知識ベース化してコンピュータに組み込んで、人間に代ってあるいは人間を支援して問題解決を行うことを目的とした計算機システムと定義される。ところが、現在のところ実用になっているシステムを見ると、特に専門家が使う問題解決支援システムとしての性格が強い。また、最近の、ニューラルネットワーク、ファジィ理論などを利用したシステムは、従来の知識ベースと推論機構という枠組みからは外れる面も強い。そこで、本報告書で対象とするエキスパートシステムは、次の性質をもつ計算機システムとして定義する。

- ① 広義の人工知能技術を使って実現したシステムであること。
- ② 利用者（専門家／初心者）にとって、問題解決の代替もしくは支援を行う能力をもつこと。
- ③ 利用者の問題解決に対する要求に対応して開発されるシステムであること。

このようなシステムに対しては、客観的な評価の必要性も高く、本調査研究の対象として適切である。また、このような意味で実用に供されているエキスパートシステムの中には、従来の手続き型言語で実現したものや、最初はAIツールで開発していても、運用上の制約などから従来型の言語で記述しなおしたものも多い。本調査研究では、このようなシステムも評価の対象となるエキスパートシステムと考えることとする。

ただし、エキスパートシステムの中には、データベースや数値計算システムなど従来型システムと組合せて実現されている例も多い。このようなタイプのシステムについては、本報告書では、とりあえずエキスパートシステムの部分のみを評価対象とする。これは、従来型システムに対しては、少なくとも原理的には従来の評価手法が適用できること、ならびに、エキスパートシステム部分の評価手法が確立できれば、それらを統合化することは比較的容易と考えら

れることによる。

一般に、システムを評価する視点としては、経済的な視点・社会的な視点・技術的な視点の3つが存在する。エキスパートシステムの場合にも同様である。経済的な視点としては、たとえば、エキスパートシステムがどれだけ利益を産み出すか、開発の費用・効果をどう判断するかなどの問題が生ずる[Schorr 1989]。社会的な視点としては、たとえば、利用者が知識ベースの知識を規準として納得できるかどうかなどという問題がある。これらの視点は非常に重要なものではあるが、本年度は、人工知能技術から直接派生する技術的な諸問題を主な検討対象とした。これは、技術的な視点からの評価手法が定まっていない現状においては、まず、これを評価ガイドラインとしてまとめることが重要と考えたためである。

したがって、本報告書で述べるエキスパートシステムの評価ガイドラインは、特に、システムのVerification/Validationを中心としたものである[O'Keefe 1989]。Verificationとは、システムの内的な正しさを保証する作業、すなわち、システム開発時に生ずるエラーをなくす作業を意味し、Validationとは、システムを外部からみたときの正しさ、すなわち、システムが現場で運用されているときに正しい望みの結果がでるようにするための確認作業を意味する[Deutsch 1982]。

これらの作業を実施するには、システムの正しい仕様があらかじめ与えられている必要がある。しかし、エキスパートシステムではシステム開発開始以前の段階で仕様が明確に定まっていることはほとんどなく、開発方法の中心であるプロトタイピングも手法として確立しているとは言い難い。そこで、本委員会では、プロトタイピングによるエキスパートシステム開発方法を、ごく一般的な形でとりまとめ、それにしたがって、評価ガイドラインを作成することとした。

従来型システムの評価方法と比較すると、エキスパートシステムの評価には、容易な点と困難な点とが存在する[O'Keefe 1989], [Guida 1989]。容易な点としては、ルールやフレームに代表される知識表現手法を採用することによって、情報のモジュラリティや可読性・一覧性が高まり、個々の知識の正しさの確認作業が単純になることなどが上げられる。困難な点としては、知識の使い方がシステムのもつ推論機構にまかされるために、システムの動きがあらかじめ予

想できず、テスト作業で組合せ的な爆発が生ずる点などが上げられる。本報告では、これらの点を考慮した上で、適切なレベルでの、エキスパートシステム評価ガイドライン確立を目指して調査研究を行った。

(3) 主要な成果と今後の予定

本報告の主要な成果は次の5点にまとめられる。

- ① 技術的な側面を中心にエキスパートシステム評価項目を取りまとめ、広い範囲の問題に適用できるような評価ガイドラインを設定したこと。
- ② エキスパートシステムの種類と評価者の視点とから、ガイドラインの諸項目を整理して示し、実際の評価に適用できるようにしたこと。
- ③ エキスパートシステム評価の事例分析を行い、評価項目の分類の妥当性について検討を行ったこと。
- ④ 海外調査、ヒアリング、文献調査などを行って、エキスパートシステムの評価の問題について現状分析と調査を実施したこと。
- ⑤ エキスパートシステム評価の代表的な理論・手法について、広く文献調査を実施し、その結果をとりまとめたこと。

本委員会では、これらの成果に基づいて、今後は以下のような調査研究活動を継続して実施する予定である。

- ① 要求仕様書とガイドラインの評価項目との関連の明確化
- ② ガイドライン項目の選択基準・評価基準の設定
- ③ ガイドラインの評価

【参考文献】

- [Deutsch 1982] Deutsch, M.S.: Software Verification and Validation - Realistic Project Approaches, Prentice-Hall, 1982.
- [Hayes-Roth 1983] Hayes-Roth, F., Waterman, D.A., Lenat, B. (eds.): Building Expert Systems, Addison-Wesley, 1983.
- [JIPDEC 1989] 日本情報処理開発協会: エキスパートシステム評価に関する研究開発, 1989.
- [ICOT-JIPDEC 1989] 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(全5分冊), 1989.
- [Schorr 1989] Schorr, H., Rappaport, A.(eds.): Innovative Applications of Artificial Intelligence, AAAI Press, 1989.
- [O'Keefe 1989] O'Keefe, R.M., O'Leary, D.E.: The Verification and Validation of Expert Systems, 11th IJCAI Tutorial, MP4, 1989.
- [A I 白書 1989] ICOT-JIPDEC AIセンター編: 人工知能の技術と利用 - A I 白書 -, 日本情報処理開発協会, 1989.
- [Guida 1989] Guida, G., Tasso, C. (eds.): Topics in Expert System Design, North-holland, 1989.

2. エキスパートシステム評価ガイドライン（中間報告）

2. エキスパートシステム評価ガイドライン（中間報告）

2章では、本年度検討したエキスパートシステム評価ガイドラインの中間報告について述べる。まず、2.1では、本年度作成した評価ガイドラインの基本方針を述べ、エキスパートシステムを評価するための視点を明らかにする。ついで、2.2では、エキスパートシステム評価に必要不可欠であるシステムのライフサイクルに対する我々の考え方を述べる。2.3では、評価の前提となるエキスパートシステムの特徴・特性をプロフィールとして取りまとめる。2.4では、V & V手法を確立するという立場からエキスパートシステム評価のための基本的な項目を示す。ついで、2.5において、評価項目の詳細な説明を行い、評価を行う上で注意が必要な点をあきらかにする。2.6では、[ICOT-JIPDEC 1989]の分類にしたがってエキスパートシステムの型に特徴的な評価項目について解説する。2.7では評価者別に重点的に評価すべき項目を説明する。2.8では、エキスパートシステムの評価・テスト方法についての網羅的に紹介を行う。

【参考文献】

[ICOT-JIPDEC 1989] 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（全5分冊），1989.

2.1 エキスパートシステム評価ガイドラインの基本方針及び評価の視点

エキスパートシステム評価ガイドラインを設定するにあたっては、エキスパートシステムのライフサイクル、対象とするエキスパートシステムの範囲、エキスパートシステムに対する要求仕様とドキュメントの扱い、Verification & Validation の考え方、ガイドラインの想定利用者・利用時期の5つの観点から、次のような方針をまとめ、評価項目の分類を行った。

(1) エキスパートシステムのライフサイクル

1章でも述べたように、我々は、エキスパートシステムもひとつのソフトウェアである以上、なんらかの利用者要求にしたがって、システム開発が開始され、プロトタイピングを繰返すことによって、仕様が次第に明確化されるという立場をとる。我々の提案するライフサイクルモデルは、プロトタイピングに基づくフェーズドアプローチであり、各フェーズが終了するごとに、直前の段階で定義された要求仕様に対するV & Vが実施され、次のフェーズへむけての仕様が決定する。

もちろんプロトタイピングの考え方は、従来型ソフトウェアの開発においても成立するものであるが、エキスパートシステムの場合には、特に、①問題解決に必要な情報（知識）とそれに必要なアルゴリズムの性質が開発初期には明確になっていない点、②システムが成功をおさめるには、継続的な保守・拡張の作業が不可欠となる可能性が強い点の2つが特徴的である。

プロトタイピングに基づくライフサイクルモデルには、[Martin 1982], [Birrel 1985], [Mayhew 1987], [Miller 1989], [Weitzel 1989]などに見られるようにさまざまな形態が提案されており、我国でもツールベンダーを中心にいろいろな考え方が提供されている。本報告では、特定のモデルに依存しないように、2.2で述べるような形態を定め、実験型プロトタイピング、発展型プロトタイピング、実用システム開発の諸フェーズにしたがってシステム開発が進展するものとした。そして、この枠組みにしたがってV & Vの考え方をとりまとめた。

また、研究的要素が強いエキスパートシステムについては、ライフサイクルという概念をあてはめにくいという意見も存在する。これについては、初期的な要求仕様をまとめるまでの試行的な研究開発を、特に、探索型プロトタイ

ングと呼び、探索型プロトタイプについては、要求仕様が与えられないために、V & V の対象とはしないこととした。

【参考文献】

- [Martin 1982] Martin, J.: Application Development without Programmers, Prentice-Hall (1982).
- [Birrel 1985] Birrel, N.D., Ould, M.A.: A Practical Handbook for Software Development, Cambridge University Press (1985).
- [Mayhew 1987] Mayhew, P.J., and Dearnley, P.A.: An Alternative Prototyping Classification, The Computer Journal, Vol. 30, No.6, pp.481-484 (1987).
- [Miller 1989] Miller, L.A.: A Comprehensive Approach to the Verification and Validation of Knowledge-Based Systems, Proc. IJCAI'89 Workshop on VV&T of KBSs, 1989.
- [Weitzel 1989] Weitzel, J.R., Kerschberg, L.: Developing Knowledge-Based Systems: Reorganizing the System Development Life Cycle, Comm. ACM, Vol.32, No. 4, pp. 482-488 (1989).

(2) 対象とするエキスパートシステムの範囲

本報告で評価の対象となるのは、実運用システムを最終的なターゲットとするエキスパートシステムである。このようなエキスパートシステムの範囲を明確にするための概念としては、開発のライフサイクル、システム構成、システムの利用目的、システム開発で使われる人工知能技術などが上げられよう。これらに対して、本報告では次のような考え方を採用する。

① ライフサイクル

(1) ならびに 2.2 で述べるように、プロトタイピングに基づくフェーズドアプローチで開発されるエキスパートシステムにおいて、実験型プロトタイピング以降のフェーズを評価の対象とする。

② システム構成

一般にエキスパートシステムは、実用レベルに近づくほど、従来型システムとの統合化が重要な課題となる[ICOT-JIPDEC 1989], [Schorr 1989]。しかし、このような統合的なシステムのV & Vについてのガイドラインを取りまとめるのは容易ではない。そこで、本報告書では、ソフトウェアシステムのうち、エキスパートシステムとして作られている部分のみを対象にV & Vの諸問題について考察することとする。

③ システムの利用目的

エキスパートシステムの利用目的としては、「専門家」または「一般利用者」のいずれかの作業を「支援する」のかまたは「代替する」のかを組み合わせた、4通りの考え方がある。また、システムの開発主体としては、「組織の開発担当部門によるもの」、「システムを必要とする個々人によるもの」が考えられる。本報告では、システムの利用目的としては、上の4つの場合をすべて網羅し、また、組織の開発担当部門が主体的に開発を行う、ある程度規模の大きなエキスパートシステムを対象とする。

④ 人工知能技術

本報告で主に対象とするエキスパートシステムは、知識ベースと推論機構とから構成されるいわゆる知識ベースシステムであるが、システムで採用される人工知能技術としては、これのみにとらわれず、最近注目を集めている、ニューラルネットワークやファジィ技術を利用したシステム[AI白書 1989]についても、同じ枠組みで考察する。

【参考文献】

[ICOT-JIPDEC 1989] 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(第5分冊：知識システムの事例), 1989.

[Schorr 1989] Schorr, H., Rappaport, A. (eds.): Innovative Application of Artificial Intelligence, AAAI Press, 1989.

[AI白書 1989] ICOT-JIPDEC AIセンター編：人工知能の技術と利用－AI白書－, 日本情報処理開発協会, 1989.

(3) エキスパートシステムの要求仕様とドキュメントの扱い

我々のプロトタイピングアプローチでは、開発の各フェーズごとにシステムの要求仕様は定義され、次のプロトタイプ開発もしくは、次フェーズへと伝えられていく。各段階で作成されるドキュメントは、次第に詳細なものとなる。そして、各段階におけるドキュメントは、次段階で作られるシステムに対するV & Vの基準として利用される。

このような考え方は、従来型システム開発における要求仕様とドキュメントの関係と等しい。しかし、エキスパートシステムの場合は、次の点で、各フェーズでのドキュメントの役割が異なっている。

- ① エキスパートシステム開発では、適切な対象業務範囲の決定が重要なポイントであるため、実験型プロトタイピング開発にはいる直前の企画提案書・調査報告書の役割が重要である。
- ② いわゆる専門家の知識によって、システムの性能が大きく左右されるため、知識分析書とプロトタイプとの間の整合をとることが重要である。

(4) Verification & Validation の考え方

さまざまな文献で定義されているのと同様に、我々は、Verificationという用語で、システムの内的な正しさを保証する作業、すなわち、システム開発時に生ずるエラーをなくす作業を指し、Validationという用語で、システムを外部からみたときの正しさ、すなわち、システムが現場で運用されているときに正しい望みの結果がでるようにするための確認作業を指す[Deutsch 1982], [EPRI 1988]。

1章でも述べたように、従来型システムのV & Vと比較すると、エキスパートシステムのV & Vには、容易な点と困難な点とが存在する。この原因は、基本的には、エキスパートシステムのもつ、宣言的な問題解決知識のモジュラリティの高さと推論機能の柔軟さにあると考えられる。V & Vのガイドラインを設定するにあたっては、この点について十分な考慮が必要である。

そこで、我々は、まず、できるだけ広い範囲の問題に適用できるように一般的な評価項目を設定した。これについては、2.4と2.5で述べるとおりである。

次に、開発するエキスパートシステムの対象分野の違いによって生ずる評価方法の違いを明確にするために、[ICOT-JIPDEC 1989]で示された5つの分類、

すなわち、合成型の問題として、設計型・計画型システム、解析型問題として、解釈型・診断型・制御型システムにしたがって、各問題で重要となるV & Vの視点について説明を行うこととした。さらに、知的C A Iを念頭に、複合型問題についても考察を行った。これらの分類はややあらく、提案したガイドラインを実際に適用する場合には、不十分な面もあるが、広い範囲のシステムをカバーするための第一近似と考えることができる。この結果については、2.6で述べる。

さらに、プロジェクトに参画する評価者を、マネージャ、専門家、第3者専門家、ユーザ、開発担当者、運用担当者の6種類に分けて、各評価者別に重点的に評価すべき項目について解説を行った。これについては2.7で述べる。

【参考文献】

- [Deutsch 1982] Deutsch, M.S.: Software Verification and Validation - Realistic Project Approaches, Prentice-Hall, 1982.
- [EPRI 1988] EPRI Report: Verification and Validation of Expert Systems for Nuclear Power Plant Applications, EPRI NP-5978, 1988.
- [ICOT-JIPDEC 1989] 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(全5分冊), 1989.

(5) ガイドラインの想定利用者と利用時期について

本報告で扱うエキスパートシステム評価項目の利用者としては、エキスパートシステムの開発にたずさわる開発担当者もしくはプロジェクト管理担当者を想定した。このような担当者レベルの技術者・管理者こそ、エキスパートシステムのV & Vの概念を明確に理解することが重要であるし、ガイドラインを利用することによって、より信頼性の高いシステムを実現する義務をおっていると考えたからである。

さらに、この想定利用者は、(必ずしもエキスパートシステムに限らない)ソフトウェア開発プロジェクトの経験を持ち、かつ、人工知能の基礎概念についてひととおりの知識を持つことを仮定している。したがって、本報告書では、ソフトウェア開発プロジェクトに対する一般的な考察や、人工知能技術に対する解説をできるだけ簡潔な範囲にとどめている。これらについて、より詳し

い情報が必要な場合は、適宜参考文献を参照されたい。

また、評価項目の利用時期としては、対象分野・システムの調査が終了した時点、もしくは、対象システムに対する探索型プロトタイピングの作業が終了した時点以降を想定している。そして、2.2で述べるライフサイクルにしたがって、実運用システムが完成し、運用・保守にはいるまで、各フェーズごとに利用できるものを目指した。

さらに、扱いうる問題の範囲はできるだけ広くとり、かつ各項目は網羅的なものにするが、全貌が見分けられなくなるほど膨大なリストはつくらないようにつとめた。

【参考文献】

[Deutsch 1982] Deutsch, M.S.: Software Verification and Validation - Realistic Project Approaches, Prentice-Hall, 1982.

[Birrel 1985] Birrel, N.D., Ould, M.A.: A Practical Handbook for Software Development, Cambridge University Press (1985).

[ICOT-JIPDEC 1989] 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(全5分冊), 1989.

[AI白書 1989] ICOT-JIPDEC AIセンター編：人工知能の技術と利用－AI白書－, 日本情報処理開発協会, 1989.

(6) 評価項目の分類

本報告書で述べる評価項目は以下のように大きく7つの視点と3つの共通の考え方とに分類されている。これらに関する詳細な記述は、2.3と2.4に述べる。

① 7つの視点

- 知識ベースの妥当性評価
- 問題解決・推論方式の妥当性評価
- 結果の妥当性評価
- 要求仕様に対する達成度評価
- 知識ベースの保守性評価
- マンマシンインタフェースの妥当性評価
- システムの総合評価

② 共通の考え方

- 結果系：プロダクトに関する評価
- 要因系：評価方法に関する評価
- 要因系：開発／保守方法・技法・ツールに関する評価
- その他：経済性、社会性

2.2 エキスパートシステムの評価とライフサイクル

2.2.1 エキスパートシステム開発のライフサイクルの基本的考え方

(1) なぜプロトタイピングアプローチか

エキスパートシステムは従来計算機化が難しいとされていた非定型業務を対象としているものが多い。つまり、専門家の経験や勘に頼っていた種々の診断問題、生産計画等に代表される計画問題、高度な専門知識を必要とする設計問題等がその対象分野になっている。これらの分野における問題解決方法は従来の定型業務のように決定的な処理手順で表現することが難しいケースが多く、非決定的な処理にならざるを得ない。このような問題構造を一般に悪構造 (Ill Structured) とよばれている。従って、これらの問題をウォーターフォール型アプローチでトップダウンに開発することは難しい。そのため、これらの問題は、部分的に少しずつ作成し問題構造を明確にしていきながら開発を進めるプロトタイプアプローチが適している。一般的にプロトタイプを作成する目的として三つの立場が考えられる。

- a) 技術的に未経験な部分、不安な部分を切り出して、その部分の簡単なモデルを作成して技術的可能性を検証する。
- b) システムの技術ポイントを抽出するために、既知の部分から作成し、不明な部分はブラックボックス化しシステムイメージを構築する。
- c) システムのイメージを共有化しシステム運用時に仕様上のトラブルを防止するために、システムの動作イメージが見えるようなプロトタイプを作成する。

Ill Structured な問題に対応するという観点でプロトタイプアプローチを見た場合、a) の立場で開発する場合は主流となるため、以下は a) を前提に議論する。

(2) 要求仕様は作成できるか

エキスパートシステムの対象分野は問題の構造が Ill Structured であると同時に問題自体を明確に規定することが難しい悪定義 (Ill Defined) 問題であると言われている。つまり、対象とする範囲、問題解決器がどのような回答を出せば良いとするのか等が開発当初においては明確に定義付けることが困難である問題を Ill Defined な問題であると言う。従って、開

発当初においては要求仕様を作成することは出来ないとする考え方がある。エキスパートシステムの有効性が叫ばれ始めた頃においてはエキスパートシステムの持っている問題解決能力を有効に利用するためにはどうしたら良いかと言う観点でシステム開発を行っているケースが非常に多かった。つまり、シーズ中心の開発である。また、現在においても、高度な推論機構を試行し、併せて知識構造を深く究明するためにシーズ指向のケースも有る。我々はこれを研究開発エキスパートシステム（EES: Exploring ES）と呼ぶことにする。

一方、エキスパートシステムといえども一つのシステムであるため、どのような機能を果たし、どの程度の性能のシステムが欲しいかが明確でないままにシステム開発を開始すること等ありえないとする立場もある。近年、実用に耐え得るエキスパートシステムが出現しているが、これらの多くは、これらの立場での開発である。つまり前者がシーズ指向で有るのに対し後者はニーズ指向であるといえる。これを我々は、ビジネスエキスパートシステム（BES: Business ES）と呼ぶことにする。

本報告書はエキスパートシステムの評価に関するものである。従来、要求仕様は、そのシステムの評価基準のベースになるべき物である。また、前述のようにエキスパートシステムが実システムに組み込まれるようになっている現状を考慮すると、我々は後者の立場をとることにする。もちろん、エキスパートシステムがIll Definedな問題を取り扱うことには変わりが無いため、開発当初ではシステムとしての本質的な要求を定義し、プロトタイプ各フェーズで要求仕様を段階的に明確にしていくというアプローチをとることにする。

2.2.2 エキスパートシステム開発のライフサイクル

エキスパートシステム開発のライフサイクルを図2.2-1に示す。

調査・計画	プロトタイプ		実用システム 開発	運用テスト	運用 洗練化
	実験型 プロトタイプ	発展型 プロトタイプ			

図 2.2-1 エキスパートシステム開発ライフサイクル

(1) 調査・計画フェーズ

このフェーズは従来のシステム開発における初期フェーズと同じで以下の項目を明確にする。

- ・ 要求定義
- ・ システムの位置づけ（他のシステムとの関連性）
- ・ 開発プロジェクト体制
- ・ 開発スケジュール

(2) 実験型プロトタイプフェーズ

このフェーズでは、実験的にプロトタイプを作成する。作成されるプロトタイプは、知識量、性能、マンマシン機能等は実用レベルに達しなくてもよい。ここでは技術的に核になる知識・推論機能がともかく実現されておればよい。このプロトタイプにより、実用化のめどをたてる事が重要である。

(3) 発展型プロトタイプフェーズ

このフェーズでは、規模的には小規模でもシステムの機能が見えるレベルのプロトタイプを作成する。このプロトタイプを関係者に対しデモを行い、実システム開発を組織的にオーソライズさせるのに利用する。また、エンドユーザ、開発者、専門家のそれぞれが、このフェーズ後半にはシステムイメージが明確になり実用システムの要求仕様を詳細に記述できるレベルになれるようなプロトタイプにすべきである。

(4) 実用システム開発フェーズ

プロトタイプが評価用またはデモ用であるのに対して、実用システムとは実際に業務内に組み込まれて利用されるシステムであると定義する。このフェーズでは従来のシステム開発と基本的にはなんら変わりはない。ただ、このフェーズにいたるまでにプロトタイプを作成しているため、その財産（仕様・知識・技術・プログラム等）を、有効に利用できる。しかし、必ずしもプロトタイプシステムを改良して実用システムにしていくべきであると言うわけではない。実用システムの仕様に合わせて、言語・ツール、推論機構等を見直して、再構築するアプローチをとる必要もある。

(5) 運用テストフェーズ

このフェーズではテストランを行う。テストデータによる検証、従来システムとの並行ランによって、システムを評価し運用に耐えられるレベルまで洗練化する。

(6) 運用・洗練化フェーズ

システムを実運用していきながら、システムの問題解決能力を一層向上させていくフェーズである。

2.2.3 開発フェーズと評価対象

図 2.2-2は各フェーズでの評価対象を表している。

開発フェーズ	調査計画	プロトタイプینگ		実用システム 開発	運用テスト	運用 洗練化
		実験型 プロトタイプ ینگ	発展型 プロトタイプ ینگ			
成果物／評価対象	企画提案書 (第0版仕様書)	第Ⅰ版仕様書	第Ⅱ版仕様書	実用版要求仕様書	——	変更要求仕様書
	—	第Ⅰ版プロトタイプ	第Ⅱ版プロトタイプ	実用システム	実用システム	デリバリシステム
	調査報告書	第Ⅰ版知識分析書	第Ⅱ版知識分析書 第Ⅱ版プロトタイプ設計書 第Ⅱ版プロトタイプテスト文書	実用版知識分析書 実用版設計書 実用版テスト文書	運用テスト文書 運用マニュアル	運用実績書 保守報告書 改訂版ドキュメント一式 改訂版デリバリシステム

図 2.2-2 開発フェーズと成果物・評価対象

(1) 調査・計画フェーズ

ここでは、企画提案書、調査報告書を作成する。企画提案書には概ね前節の(1)で述べた項目が記載される。これは、以下のフェーズでの評価のベースとなるものである。

また、調査報告書は企画提案書の妥当性を裏付けるものであり、以下の様な項目が必要に応じて調査される。

- ・現状の問題点
- ・あるべき姿
- ・問題を解決するための技術ポイント
- ・世の中の動向など

この様な調査により実験型プロトタイプフェーズにおいて何を確認していったら良いかを明確にしていく。

(2) 実験型プロトタイプフェーズ

このフェーズで何を検証したいのかを明確にするために第Ⅰ版要求仕様書を作成する。実験型プロトタイプを作成した後、この仕様書をもとに評価を行う。例えば、以下のような評価項目が考えられる。

- ・開発システムが目標としている効果が得られる可能性があるかの評価
- ・知識が本当に得られるかの評価
- ・開発環境の評価（言語、ツールなど）
- ・開発体制の評価

(3) 発展型プロトタイプフェーズ

このプロトタイプの目的は前述の様に開発を組織的にオーソライズさせ、実用システムの仕様を詳細に定めることである。そのためにどのような機能を実現させ、何を検証するかを第Ⅱ版要求仕様書に記述する。評価項目としては実験型プロトタイプ時のものに加え次のようなものが考えられる。

- ・性能見通しの妥当性
- ・知識規模の見通しの妥当性
- ・知識源の妥当性
- ・問題解決・推論方式の妥当性

(4) 実用システム開発フェーズ

ここで作成される実用版要求仕様書は次の運用テストフェーズでのシステム評価の基準と成るものである。

(5) 運用テストフェーズ

2. 4節以降で述べる評価項目の詳細は主に、このフェーズで行われる。

(6) 運用・洗練化フェーズ

このフェーズではこのシステムを実際に活用しながら知識を修正・追加・削除などを行いシステム能力をより一層向上させていく。このため知識の妥当性の検証が重要になってくる。

2.3 エキスパートシステムのプロフィール

エキスパートシステムを評価するに当たって前提となるプロフィール情報、即ちシステムの特徴・特性を説明する。この項目は以下の通りである。

①システムの目的と期待効果

②システムの概要

a. システムの機能と構造

b. システムの開発者と開発手順

c. システムの動作環境

d. システムの規模

③客観的な評価の可能性について

④今後の課題

(1) システムの目的と期待効果

ここでは、開発動機、システムの目的、システムの責任のレベル、利用対象者、期待効果、既存技術の検討について述べる。

システムの責任のレベルではLight DutyからHeavy Dutyのレベルまでのうちのどのレベルなのかを明らかにする。利用者対象者では、専門家、担当者、初心者、不特定利用者なのかを明らかにする。

(2) システムの概要

①システムの機能と構造

ここでは、システムが実現する機能、そのシステム構造（推論方式、知識ベース）、及びシステム機能実現の困難さの程度を述べる。

システムの構造では、どのような推論方式であるか、不確実な情報処理をしている場合は、例えば統計的にどう処理されているのか、または信念の問題、ファジー推論等をどう採用しているのかを述べる。

知識ベースでは、知識源の特徴とその種類（一般的法則、既存ドキュメントの知識、単独の専門家からの知識、複数の専門家からの知識など）について述べる。他システムとの連携をしている場合は、その連携方法を述べる。

システム機能実現の困難さは下記の要因を考慮して述べる。

- リスクの度合
- 利用頻度
- 支援機能／代行機能の区分
- システムの規模
- 対象範囲と要求される解の正解率

② システムの開発者と開発手順

システムの開発者では、KE、専門家、エンドユーザなのかを述べ、開発のライフサイクルを説明する。

③ システムの動作環境

ここでは、システムを開発したときの環境及びデリバリ環境の各々につき以下の項目について記述する。

- 対象ハードウェア
- 使用メモリ量
- 使用ファイル量
- OS
- 処理方式（インタプリタ／コンパイラ）
- 言語／ツール

④ システムの規模

システムの規模では、定量的にルール数、フレーム数、ライン数を述べる。

（３）問題のタイプとシステムの特徴

① 問題のタイプ

問題のタイプでは、合成型問題（設計、計画）、解析型問題（解釈、診断、制御）、複合型問題なのかを明らかにする。

② システムの特性

ここでは、システムの特性を表形式にて、項目の関連性、システムの実現度、及びコメントを述べる。

各問題のタイプごとに、下記の項目に分類して項目を列挙した。

- 問題・作業の特徴

- 基本タスク（問題に使えるテクニック）
- 問題解決機能（実現手法）

（詳細な表は、「2.6 問題領域別の評価項目」を参照のこと）

（４）客観的な評価の可能性について

ここでは、対象とするシステムは客観的な評価が可能なのかどうか、出来ない場合にはその理由、出来る場合にはどの程度のことが出来るのかを述べる。

（５）今後の課題

ここでは、現段階のシステムが、プロトタイプシステムなのか、まだ実際には運用経験はないが実用化システムなのか、実際に運用している実運用システムなのかを明らかにする。

また現段階でのエキスパートシステムの成長性をどう評価し、今後より一層の役に立つシステムを目指すべきなのか、その課題を述べる。

2.4 エキスパートシステムの評価項目

評価項目は、「知識ベースの妥当性評価」、「問題解決・推論方式の妥当性評価」、「結果の妥当性評価」、「要求仕様に対する達成度評価」、「知識ベースの保守性評価」、「マンマシンインタフェースの妥当性評価」、及び「システムの総合評価」から構成されている。各カテゴリーに属する評価項目は、評価内容に応じて、更に、結果系を評価する「プロダクトに対する評価」、要因系を評価する「評価方法に対する評価」及び「開発／保守方法・技法・ツールに対する評価」、そして、その他の特性要因を評価する「その他の評価（経済性、社会性）」の3つに細分される。

① 知識ベースの妥当性評価 6項目

知識ベースに格納されている知識が、問題の対象領域において妥当であるかを評価するための評価項目である。

② 問題解決・推論方式の妥当性評価 5項目

問題解決・推論方式が、その対象の問題領域において妥当であるかを評価するための評価項目である。

③ 結果の妥当性評価 4項目

エキスパートシステムから出力された結果が、その対象の問題領域において正しか否かを評価するための評価項目である。

④ 要求仕様に対する達成度評価 3項目

エキスパートシステムが要求仕様で規定した要件をどの程度達成しているかを評価するための評価項目である。

⑤ 知識ベースの保守性評価 6項目

知識ベースを更新する場合に、知識ベースの内容を理解し、修正すべき箇所を発見・同定し、そして、修正した結果が正しいか否かを確認するのが、どの程度容易かを評価するための評価項目である。

⑥ マンマシンインタフェースの妥当性評価 2項目

エキスパートシステムのマンマシンインタフェースが、利用時、保守時からみて使いやすくできているかを評価するための評価項目である。

⑦ システムの総合評価 10項目

エキスパートシステムが情報処理システムとして総合的にみて満足できるものかどうかを評価するための評価項目である。

以下に、評価項目を示す。

(1) 知識ベースの妥当性評価

《プロダクトに対する評価》

- (A) 知識ベースの正当性評価
- (B) 知識ベースの完備性評価
- (C) 知識ベースの一貫性評価

《評価方法に対する評価》

- (D) 経験的知識を正当化する方法とその確認

《開発／保守方法・技法・ツールに対する評価》

- (E) 知識表現の選択と適切さの確認
- (F) 知識獲得方法の選択と適切さの確認

(2) 問題解決・推論方式の妥当性評価

《プロダクトに対する評価》

- (A) 推論処理の妥当性評価
- (B) 推論時間の妥当性評価

《開発／保守方法・技法・ツールに対する評価》

- (C) 推論手法の選択と適切さの確認
- (D) 不確実情報に対する処理技法の適切さの確認
- (E) 学習機能の適切さの確認

(3) 結果の妥当性評価

《プロダクトに対する評価》

- (A) 解の正解率評価
- (B) 解の網羅性評価
- (C) 解の最適性評価

《評価方法に対する評価》

- (D) 解の評価方法の適切さの確認

(4) 要求仕様に対する達成度評価

《プロダクトに対する評価》

- (A) 要求仕様内容に対する評価
- (B) 要求仕様に対する達成度評価

《開発／保守方法・技法・ツールに対する評価》

- (C) 要求仕様設定手段の評価

(5) 知識ベースの保守性評価

《プロダクトに対する評価》

- (A) 変更箇所発見の容易性評価
- (B) 知識ベースの理解性評価
- (C) 変更箇所の局所性評価
- (D) 知識ベースの統一性評価

《開発／保守方法・技法・ツールに対する評価》

- (E) 品質安定性の確認方法の評価
- (F) 保守マニュアルの整備状況の確認
- (G) 保守用ツールの整備状況の確認

(6) マンマシンインタフェースの妥当性評価

《プロダクトに対する評価》

- (A) 利用時のインタフェース評価
- (B) 保守時のインタフェース評価

(7) システムの総合評価

《プロダクトに対する評価》

- (A) テーマ選定の妥当性評価
- (B) 利用者の満足度評価
- (C) 統合性の評価
- (D) 成長可能性の評価
- (E) 開発／保守用ドキュメントの評価

《開発／保守方法・技法・ツールに対する評価》

- (F) 開発方法の適切さの確認
- (G) ツール・言語選択の適切さの確認

《その他の評価（経済性、社会性）》

- (H) システム開発の生産性評価
- (I) システムのコストパフォーマンス評価
- (J) セキュリティ対策の評価

2.5 エキスパートシステム評価項目の解説

2.4節で述べたエキスパートシステム評価項目の解説を、下記の項目に従って行う。

- ① 知識ベースの妥当性評価
- ② 問題解決・推論方式の妥当性評価
- ③ 結果の妥当性評価
- ④ 要求仕様に対する達成度評価
- ⑤ 知識ベースの保守性評価
- ⑥ マンマシンインタフェースの妥当性評価
- ⑦ システムの総合評価

なお、本評価項目を利用する際の留意点は次の通り。

- 本評価項目は、開発担当者がレビューするときに用いる。
- 本評価項目は、実用システムに対して適用する。
- プロトタイプ版システムの場合は、本評価項目から適時抽出選択して適用する。
- なお、本評価項目は、以下のように分類してある。

（結果系）

《プロダクトに対する評価》

（要因系）

《評価方法に対する評価》

《開発／保守方法・技法・ツールに対する評価》

（その他）

《その他の評価（経済性、社会性）》

2.5.1 知識ベースの妥当性評価

(1) 定義

知識ベースに格納されている知識が、問題の対象領域において妥当であるのかを評価することである。

(2) 目的と必要性

知識ベースの質が、結果の良否に最も直接影響を与える要因であるので、知識ベースの正当性、完備性、一貫性の評価を充分に行う必要がある。

《プロダクトに対する評価》

(A) 知識ベースの正当性評価

(1) 定義

知識ベースに格納された知識の正しさを評価することである。

(2) 目的と必要性

知識ベースに格納された知識自体が正確であることは、基本的な要件である。特に、経験的知識を取り扱わねばならないときには、その要求は強い。

(3) 説明

専門知識は、教科書的知識と経験的知識に大きく分けられる。教科書的知識とは、一般的に実験や実証が可能、または公理や法則から導くことが出来る知識をいう。従って科学的な根拠、理論付けをすることが可能であり信頼性が高い。評価の観点からいうと、教科書的知識は正当性が裏付けられやすい。

一方、経験的知識は、一般に文書化されておらず、ときには当の専門家自身もその根拠に対し十分な説明が出来ない知識である。経験的知識の中には、旨い絞り方、無駄な思考を避ける知識、曖昧な情報を扱う知識などが含まれている。このため、効率よく複雑な問題を解決するのに役に立つ反面、科学的な根拠に乏しい、理論付けすることが難しいなどの欠点があり、慎重に取り扱わねばならない。

知識の正当性の評価では、その分野の第一人者といわれる専門家に依らざるを得ないことが多い。複数の専門家がいた場合、抽出出来る知識が異なることはもとより、同一対象に対する知識さえ異なることがある。この

ため、複数の場合、誰に決定権があるのかを明らかにしておかねばならない。

評価項目例を以下に示す。

- ①教科書的知識の知識源を原典に遡り、その出所を確認しているか。
- ②教科書的知識を解釈して利用する際、対象の問題にふさわしいのか、別の代替案はないのか等を充分吟味したか（解釈の多義性の問題）。
- ③経験的知識の信憑性はどの程度なのか、実証できるのか等について充分検討したか（評価方法については、「経験的知識を正当化する方法とその確認」を参照）。

（B） 知識ベースの完備性評価

（1）定義

対象とする問題を解くのに必要な知識が全て入っているかどうかを評価することである。

（2）目的と必要性

必要な知識が欠けている場合、正しい結果が得られない可能性が高い。とくにキーとなる知識の欠落によって、目的とはまったく異なる結論となることすらある。まず対象とする問題の範囲を明確にし、それに対する必要十分な知識が格納されていることを確認することは極めて重要である。

（3）説明

知識ベースに対しては、プロトタイプ版では知識の完備性を完全に追求する必要はない。しかし、実用版をユーザに開放するときは、より完全な完備性を追求し、その評価が必要である。

また、エキスパートシステムにおいては、実運用段階に入っても更に知識ベースが更新されるため、その都度、知識ベースの完備性評価をすることが必要となる。

知識ベースの完備性を評価する方法は2つある。動的テストに基づく評価と、静的テストに基づく評価である。前者は、最終的にはユーザがシステムを使用して、満足な結果を得るかどうかを運用テスト時に見つけるものである。満足した結果が得られていないとすると、後者により、その原因は知識の欠如なのか、知識ベースに格納されていても活用されていないのか（知識の未到着、知識の袋小路現象）などを突き止めねばならない。

(動的テストに基づく評価)

- ・対象としている問題領域で正しいと考えられている結論が、どの程度導出されるかの評価である。
- ・多様なテストケースを与えて、テストすることである。

(静的テストに基づく評価)

- ・欠如ルール、未到着ルール、袋小路ルール等を発見することである。

(C) 知識ベースの一貫性評価

(1) 定義

知識ベースの一貫性を阻害する要因が、内在しているか否かを評価することである。

(2) 目的と必要性

知識ベースが大規模になればなるほど、知識の一貫性を保つことが容易でなくなる。しかしながら、知識の一貫性の欠けたシステムは類似の要求に対して全く異なる結論を導く可能性もあり、ついには実用性がなくなる。特に、運用時の知識の更新に対する一貫性評価はより重要である。

(3) 説明

知識ベースの一貫性とは、競合した知識がないこと、冗長な知識がないこと、未参照の知識がないこと、知識間で巡回していないことをいう。また知識ベース内の知識に相互に関連する知識がある場合には、その一貫性を保つことが必要である。特に保守の際に、その関連を無視して、一方のみ更新するというようなことがおこりやすい。

知識ベースのルールをシステマチックに解析し、一貫性の欠陥を見つけ出すことが出来る。例えば、適切でない応答を起こしている原因を見つけるため、トレースしてルールの発火やその順序を追い、知識が如何に不正確で適切でないかを突き止めることが出来る。

なお、知識ベースの一貫性を保つための方法には、

- ・知識ベースをルール群に機能分けする
- ・ルール番号による識別化
- ・黑板モデルやフレーム表現を用いて、知識をモジュール化して、知識ベースの構造自体を見通しをよくすること
- ・TMS / ATMS 機構を用いて知識ベースの一貫性を保つこと

等がある。

《評価方法に対する評価》

(D) 経験的知識を正当化する方法とその確認

(1) 定義

経験的な知識を正当化する手段が良かったかどうかを確認することである。

(2) 目的と必要性

経験的な知識とは専門家の“うまいやり方”を用いることによって、問題解決能力を高めることが可能である知識のことをいう。但し、専門家自身を客観的に評価する基準が明確ではなく、経験的知識の評価は困難であるといえる。

しかしながら原理原則や教科書的な知識の少ない対象では、経験的な知識に頼らざるを得ない。それらの知識を何等かの形で正当化する方法を見だし、その確認をする必要がある。

(3) 説明

経験的な知識の評価の基本は、客観的に正当であることを確認することである。

その方法としては、

- ・統計的手法による検証
- ・複数の専門家による合意形成
- ・深い知識から経験的知識が導き出されるのかの検討

等がある。

また、経験的知識を生かすためにも、経験以外の教科書的な知識を可能な限り整備する努力が必要となり、その効果が大きいことが分かってきた。

《開発／保守方法・技法・ツールに対する評価》

(E) 知識表現の選択と適切さの確認

(1) 定義

知識表現の選択が対象とする問題に適切であったかどうかを確かめることである。

(2) 目的と必要性

知識表現は、記述のしやすさ、表現された知識の読みやすさ、保守の手間、実行効率、メモリ効率などに影響を及ぼすので、その選択が適切であるのか否かの評価は必要不可欠である。

(3) 説明

エキスパートシステムを開発する際、対象とする問題にふさわしい知識表現を採用しているのか、さらにその表現をどう生かすようにしてあるのかを確認することは重要なことである。

ここでは、ルール表現、フレーム表現、黒板モデルに基づく表現の特徴について述べる。

1. ルール表現

ルール表現には、以下のような長所、短所、そして留意点がある。

(長所)

- 個々のルールは作りやすくてわかりやすい。
- 追加や変更が容易である。
- 推論メカニズムが簡単である。

(短所)

- ルールの相互関係が不明確である。
- 知識の全体像がつかみにくい。
- ルール数が大きくなると処理効率が非常に悪い。
- 人間の知識構造とは異なる。
- 推論に柔軟性がない。

(ルール表現を用いる場合の留意点)

- 大規模化するとその全体構造が不明確となりがちであり、そのためルールに識別番号を導入したり、モジュール化を図る等の手だてをしなければならない。
- 各種のルールコンパイルによる推論速度の向上。
- 黒板モデル、協調推論を用いたルール群のモジュール化。
- 推論エンジンのトレーサの提供。
- ルールツリーの自動生成。

2. フレーム表現

フレーム表現は、複雑な概念構造の記述や、局面に応じて異なった推論方法を応用したい問題には特に強力である。その反面、知識ベース全体としての完備性や一貫性を管理することが困難になる。特に、付加手続きの乱用はこの点での危険が大きい。なお、付加手続きの機構（メッセージ交信の機構も）を外してフレーム表現をプロダクションシステムのデータベースとして利用するケースも多い。

また、フレーム表現では、ある典型的な状況や事象を手続きとして表現しておき、これを推論過程で利用することにより、期待駆動型推論(Expectation Driven Reasoning)をモデル化できる。これにより推論方法が固定されず柔軟性のある方式を採ることが出来るが、開発者側の負担が重くなりがちである。このため、知識エディタやトレースなどの開発環境の整備が不可欠である。

フレームの表現は、問題のタイプにより、次の3種類の利用形態が考えられる。

① フレームシステムを単に事実データベースとして用いる方法

これは、フレーム型データ構造をデータベースとするプロダクションシステムである。

② 付加手続きの役割を利用した方法

これは、インヘリタンスとデーモンの機能を備えたフレーム型データベースを持ったプロダクションシステムである。

③ メッセージ交信による方法

これは、フレームがお互いにメッセージの交信を行い、局面に応じて知識の種類や推論方法を動的に変えながら、全体として最終的な目標へ向かって行われる仕事に適した方法である。

3. 黒板モデル表現

黒板モデルによる表現には、以下のような長所、短所がある。

(長所)

- ・分散処理と作業記憶の分割による推論の高速化。
- ・知識源のモジュール化による拡張性、信頼性の向上とメンテナンスの容易性。
- ・曖昧さを持つ情報を多様な知識を協調的に用いて解釈可能とする。
- ・並列処理の可能性。

(短所)

- イベント駆動型推論方式（イベントによる知識源の駆動）は、誤った情報の混入や必要な情報の欠落に対して弱い。
- アジェンダ機能による知識源起動では、優先順位での重要度が低い問題はネグレクトされる可能性がある。

黒板モデル表現は、問題のタイプにより、次の３種類の利用形態が考えられる。

① イベント駆動型推論方式

この方式では、知識源はブラックボード事象と呼ばれるブラックボード上の特定の変化に反応して活性化される。ブラックボードモニタは活性化した知識源の詳細な条件をチェックし、もし実行可能であれば、その知識源のインスタンスを作り、これをスケジューリング行列に入れる。スケジューラは現在の制御の焦点のもとで最適なインスタンスを選択し、これを実行する。

② 目標駆動型ブラックボードモデル

これは、イベント駆動型推論方式の問題点を克服するため、ブラックボード事象から目標を生成するメカニズムを導入することによって、イベント駆動型推論方式と目標駆動型制御を統合化する方法である。

③ アジェンダによる知識源駆動

アジェンダとは、未解決のゴールなどを書き並べておく備忘録のことである。この方式では、アジェンダの中から、重要度順（優先順位）あるいは解けそうな順に問題を拾い出していく。

(F) 知識獲得方法の選択と適切さの確認

(1) 定義

知識獲得の方法の選択が適切であったかどうかを確かめることである。

(2) 目的と必要性

知識獲得方法の選択を誤ると、十分な知識が引き出せなかったり、収集できなかったりする。また、知識獲得の効率を向上させるため、知識獲得方法や支援ツールの選択が適切であるのかを確認する必要がある。

(3) 説明

エキスパートシステムの能力は知識ベースの中に格納されている知識の量と質で決まる。従来から知識獲得方法として専門家へのインタビューが

あるが、その困難性が指摘されており、最近ではいろいろな知識獲得支援ツールが開発されている。現在商品化されているツールは少ないが今後出てくるので、ツールの自動化も含めて、以下に知識獲得支援ツールの大まかな選定基準を示す。

① 知識獲得支援ツールの選定基準

知識獲得支援ツールの選定は、知識源を何に求めているのか、以下の区分によって選定する必要がある。

a. 専門知識の獲得ツール

- ・タスク限定による知識獲得の容易化 (Task Specific Tool:TST)

例) CSRL, DSPL, Copernics

- ・知識獲得支援機構の実現 (Knowledge Acquisition Tool:KAT)

例) TEIRESIAS, MORE, NEXTRA, MOLE, SALT, SEAR

b. 教科書知識の洗練化ツール

- ・知識洗練化機構の実現 (Knowledge Refinement Tool:KRT)

例) SEEK, SEEK2

c. データ群からの知識生成ツール

- ・Rule Induction Tool:RIT

例) EXTRAN, RULE-MASTER

- ・Neural Network Tool:NNT

例) Neural Works

d. ドメインモデルから知識生成ツール

- ・Domain Specific Tool:DST

例) G2, IN-AIE(ICAT), Test Bench

② 専門家へのインタビュー

専門家へのインタビューにより知識を得る場合、専門家の性格により自分の仕事を実際以上に複雑に述べたり、逆に実際より簡単であるかのように言う場合がある。また、曖昧な質問をすると意図に沿わない知識が獲得される。その結果作成された知識ベースは、専門家の持つ知識構造とも異なるのみならず、知識獲得自体失敗に終わることとなる。

専門家へのインタビューの際の基本的な留意点は以下の通りである。

- ・対象とする専門領域の知識を可能な限り事前に修得しておくこと。
- ・専門家のおかれた環境や立場を勘案してインタビューすること。

- 自分の理解したことを再確認する目的の質問も行うなど、専門家と自分の間のギャップを埋める工夫をすること。
- わかりにくい点では、質問の仕方を変えるなど角度を変えて専門家の意見を引き出すこと。

2.5.2 問題解決・推論方式の妥当性評価

(1) 定義

問題解決・推論方式がその対象の問題領域において妥当であるのかを評価することである。

(2) 目的と必要性

エキスパートシステムは知識ベースと推論機構から構成される。従って、たとえ知識ベースが妥当であっても問題解決・推論方式が妥当でなければ、やはり正当な結果を得ることはできない。特に最近では高次推論など複雑な推論処理が要求されるようになり、その影響は増大しつつある。

《プロダクトに対する評価》

(A) 推論処理の妥当性評価

(1) 定義

推論処理がその対象の問題領域において妥当であるのかを評価することである。

(2) 目的と必要性

問題解決推論方式が妥当でないと、たとえ知識ベースが正しくても解ける問題が解けなくなってしまう、または正しい答えが求められない等の可能性が高く、その評価は重要である。

(3) 説明

エキスパートシステムの初期の時代には、推論処理は単純にしておき知識の側で高い機能を実現し得ると考えられていた。しかし、最近では類型的タスクに代表されるように問題解決を重視する必要性が認識されている。このため、推論処理も次第に複雑なものが要求されており、その論理的な正しさを確認することが重要となっている。

推論処理については、一般のプログラムとの共通点も含め、以下の留意点を考慮して評価する。

- ① 知識表現との整合性（分担の明確化）
- ② 動作の透明性
- ③ 停止性
- ④ モジュール性

⑤ 動作モードと状態遷移

⑥ AI 言語上の各問題点

- メモリ管理
- 再帰呼び出し
- 動的なプログラムの書換え
- インタプリタ／コンパイラ

(B) 推論時間の妥当性評価

(1) 定義

問題解決に要した推論時間が妥当であるのかどうかを評価することである。ここでは、トータルなレスポンス時間は評価しない。

(2) 目的と必要性

エキスパートシステムの処理時間には推論処理の効率の影響が大きい。従って、推論時間の妥当性はエキスパートシステムの実用上の重要な要素である。

(3) 説明

エキスパートシステムでは、推論の結果を得るまでに要する推論時間は、知識ベースの内容と推論機構の性能に依存する。

推論に必要な時間を具体的に述べると、プロダクションシステムであれば認識－実行サイクルの時間、フレームシステムであれば発生や書換えに要する時間や継承関係をたぐる時間などであり、これらが評価対象となる。

以下の留意点を考慮して評価する。

- ① 対話性
- ② 知識ベースの規模の考慮
- ③ リアルタイム性の要否
- ④ ガーベジコレクションの有無、頻度

《開発／保守方法・技法・ツールに対する評価》

(C) 推論手法の選択と適切さの確認

(1) 定義

推論手法の選択が適切であったのかどうかを確かめることである。

(2) 目的と必要性

推論手法の選択を誤ると、知識ベースをうまく扱えない、知識ベースを構築する際に手間がかかる等の問題が生ずるのみならず、当然期待する結果が得られぬ可能性が高くなる。

(3) 説明

エキスパートシステムの基本機能は、知識表現と推論手法とで決まる。特に、設計やスケジューリングの合成型問題では、推論の性能において推論手法の占める比重が高いことが指摘され、開発の初期の段階において推論手法の選択が重要である。

開発の初期段階において、対象問題に応じて以下の項目を検討し、推論手法を定める必要がある。

- ①問題解決モデル (階層的生成検査法、トップダウン精密化、拘束最小化、関係依存後戻り)
- ②問題解決パラダイム (生成検査法、抽象化、分解・統合、制約伝播)
- ③演繹的推論手法 (プロダクションシステム、述語論理)
- ④高次推論手法 (不確実性推論、デフォルト推論、仮説推論、分散協調型推論、定性推論、事例ベース推論)

以下の留意点を考慮して評価することになる。

- ①推論手法の複雑さと知識獲得の負荷とのトレードオフ
- ②システムの透明性
- ③デリバリ環境

(D) 不確実情報に対する処理技法の適切さの確認

(1) 定義

不確実な情報に対する処理技法の適切さを確認することである。

(2) 目的と必要性

エキスパートシステムに対する期待の一つに、不確実な情報に基づく推論があげられる。従来のベイズ確率のほか、MYCINの方法やDempster-

Shaferの方法、及びファジィ推論に基づく方法などが使われるようになってきている。しかし、安易に不確実性を導入すると、推論結果の妥当性の保証が困難、知識ベースのチューニングに要する負荷大、保守性の低下、等といった問題を引き起こす場合がある。従って、対象とする問題に応じて不確実情報に対する処理技法が適切であるのかを充分確認する必要がある。

(3) 説明

以下の留意点を考慮して評価する。

- ①対象問題に内在する不確実性の明確化
- ②不確実性の性格付け（基数的、序数的、メンバーシップ的など）
- ③不確実性の局在化
- ④期待される出力と処理技法との整合性
- ⑤不確実性を用いないアプローチの検討

(E) 学習機能の適切さの確認

(1) 定義

学習機能の活用が適切であったのかどうかを確認することである。

(2) 目的と必要性

学習機能とは、ここでは問題解決の過程で問題解決能力の向上を図る機能あるいは知識獲得の自動化を促進する機能であるとする。学習機能の利用を誤ると、思ったほど問題を解く能力が向上しない、知識の抽出能力が十分でないなど、期待した効果が得られぬとともに効率上の影響も出てくる可能性があり、評価する必要がある。

(3) 説明

学習機能の例としては、例えば設計問題であれば、与えられた要求に対し部分的には様々な実現方法の知識があり、それらを組み合わせて探索的に設計作業を行う。その結果を蓄積・利用して、次回からの設計において探索の効率を上げる学習機能などが考えられる。

学習機能は、エキスパートシステムに対する利用の期待が高いが、その実現には技術的課題が多い。このため開発の各フェーズにおいて、学習機能の適切さ（必要性も含め）について評価することが重要である。

具体的な評価としては、対象システムと以下のような現状技術で可能な

学習機能との関連性及び実現の度合をチェックする。

- ①機能的な学習機能
- ②演繹的な学習機能
- ③コネクショニズムによる学習
- ④事例ベース推論
- ⑤知識ベースの洗練支援システム

2.5.3 結果の妥当性評価

(1) 定義

エキスパートシステムから出力された結果が、その対象の問題領域で正しいか否かを評価することである。

(2) 目的と必要性

如何なるシステムにおいても結果の妥当性を確認することは重要である。特に、エキスパートシステムにおいては完全なテスト手法や十分なテストデータが得られぬケースも多く、結果の評価を厳格に行う必要がある。

(3) 説明

エキスパートシステムにおいては、その解が正しいのかを確かめる方法がなかったり、あってもそれを客観的に評価することが出来なかったりする。従って解の妥当性を確かめるには、まず妥当性を確かめる手段があるのか、もしないならば工夫して手段を設定せねばならない。正解がわからない場合は、知識を提供した専門家に正解を提示してもらい、開発担当者、マネージャを含めてその解が妥当であるのかを前もってレビューすることが必要である。なお、この評価項目は、問題解決・推論方式、知識ベース、要求仕様と密接に関わり合う。

《プロダクトに対する評価》

(A) 解の正解率評価

(1) 定義

テストに使用したテスト件数の内、何件正しい結果が出たのかの割合を評価することである。

(2) 目的と必要性

解の正解率はシステム利用上の最大の目安である。ユーザはこれによって当該システムの使用方法、使用形態ないしは実用の可否の判断をすることができる。

(3) 説明

システムの正解率が具体的、かつ客観的に測定出来るならば、システムのカバーする範囲、システムの性能が明確となるため、まず測定可能になることが大切である。

また、エキスパートシステムの場合、過去の事例が少ない、専門家の経験的な知識を重視するなど、きわめてテストする環境が悪いのが一般的である。従って、従来の情報処理システムと同様の正解率の概念では適切でないケースも考えられるため、ユーザ、マネージャ、専門家の間で、正解率の定義につき予めコンセンサスを得ておくことも重要である。

最終的に専門家が判断を下すシステムでは、必ずしも全ての場合に対して正解を出さなくてもよい。非常に複雑で例外的なケースは専門家に任せ、一般的なケースのみ自動的に処理することで、専門家の負担はかなり軽減される。また、サービスのスピードアップや、専門家の工数削減によるメリットが、不正な解によるリスクを上回る場合には、完全な正解を与えなくても充分実用的なシステムであるといえる。

一方、システムが提示した解を修正することが非常に困難な場合や、不正な解によるリスクが非常に高い場合には、正解の割合は非常に高いものでなければならない。

評価をする際、100%の結果の正解率を追い続けると、いつまでたっても実用化システムにはならない。60~70%の正解率でも以上のことを考慮して使用に耐え、役立つものがあればよい。最終的には人間の判断に頼るシステムが実務的であるともいえる。

以下の留意点を考慮して評価する。

①何を持って正解率とするのかを前もって定義しておくこと

②ユーザからの要求

- ・エンドユーザが使用するシステムであれば、解の正解率を高めなければならないし、結論に至った過程を十分に説明する機能がなければならない。
- ・専門家が使用する場合は、誤った場合でもなぜ誤ったのかが判断つくことが多いので、解の正解率がシビアであるにこしたことはないが、むしろ、推論過程を説明する機能を充実すべきである。

③専門家との比較

- ・知識を提供した専門家は、予め正解のパターンを作成しておき、開発担当者とレビューをしておくこと。
- ・複数の専門家が結果に対し検討するときは、必ず共通見解を出して、合意形成を得て行うこと。

④ 限界値テスト

- 通常想定できるデータに対しての結果のみをテストすれば、当然解の正解率は良くなる。従って、希にしか発生しないデータ、例外的なデータも含めて限界値テストをする必要がある。しかし、正解率の計算に際しては、このことを十分に考慮する必要がある。

⑤ 正解のパターンを充分用意すること

- 意義のあるテストデータは充分あることが当然望ましい。そして予め用意されている正解があることが前提である。

⑥ テストケースが異なると予期しない振舞いをする

- テストデータの変更や追加をすると、今までのデータではうまくいっていても、変更後はうまくいくとは限らない。その場合には、知識ベースや推論過程に問題がある可能性もあり充分な追求が必要である。

(B) 解の網羅性評価

(1) 定義

正しい解が複数求められるべき問題の場合、どの程度網羅的に解を得ることが出来るのかを評価することである。

(2) 目的と必要性

解の網羅性は、結果に対して正しい判断を行うためにも必要不可欠な条件である。また大規模なシステムにおいては単に網羅性を上げるのではなく、効率との兼ね合いで最適な網羅性を判断することも重要となる。

(3) 説明

合成型問題（設計問題、計画問題）においては、基本的な問題解決パラダイムが生成検査法であることがあるが、この場合候補となる複数の解を生成し、それを検査によって絞り込むアプローチをとる。従って、生成時には、候補解が漏れなく上がっていることが望ましい。候補の生成問題は、探索問題に帰着する。即ち、探索空間が小さければ網羅的な探索が可能で解の網羅性がよいが、探索空間が大きくなるにつれ、探索の効率向上をさせるため、問題解決戦略を導入したり、可能性の低い解候補の探索を排除するために枝刈り規則を利用することが多い。

従って、計算量、要求仕様、解の網羅性の3者が密接に関わりあっている。

以下の留意点を考慮して評価する。

- ① 計算量や対象とする問題の特性等から、どの程度の網羅性を目標とするのかを予め充分検討すること。
- ② 解候補を生成して、その時点でその解候補が妥当であるのかどうか不明のときは、仮にそれを選択し仮説として扱う。もとの仮説と矛盾が生じなければその範囲内で正当化され、矛盾が生じたときは依存関係に基づいて後戻りする。従って解の網羅性を評価するに当たっては、問題解決パラダイムを、充分検討すること、及びこのパラダイムで取り扱われる知識を充分検討しなければならない。
- ③ 解候補の生成に続いて検査が行われ、もし不適切であればその解候補は撤回され、後戻りして別の解候補を探索する。従って問題解決パラダイムで取り扱われる知識を十分に検討することが必要である。

(C) 解の最適性評価

(1) 定義

最適解を求めることが目的である場合、対象とする問題の領域において、得られた解が最適であるか否かを評価することである。

(2) 目的と必要性

解の最適性が保証されないとユーザ自身がその最適性を調べなければならない。また、理論的に完全な最適性を求めることは無理な場合であっても最適度が高いほど、そのシステムの有用性が確認されたことになる。

(3) 説明

対象とする問題が閉空間を取り扱い、システムからの結果がシビアでなければならない問題の場合は、解が最適性を満足しなければならない。解が最適であることを保証するためには、最適解に対する評価基準が重要である。

設計問題、計画問題は、一般に最適解は未知である。従って、専門家が作成した過去の設計案・計画案との比較、またはOR手法（線形計画法、動的計画法、整数計画法など）での解と比較してどちらが、より制約条件、評価基準値を遵守しているのか等を検討することが必要である。

《評価方法に対する評価》

(D) 解の評価方法の適切さの確認

(1) 定義

結果の妥当性を評価する手段が適切であるか否かを確かめることである。

(2) 目的と必要性

評価方法を誤ると当然正しい評価が行われない可能性が高い。適切な評価方法がとられたか否かの確認は、評価の基本要件である。

(3) 説明

結果の妥当性を評価する方法は大きく分けると専門家による評価と何らかのツール、手法、データを用いて評価する場合がある。

[人間が評価方法を確認]

- ① 知識提供の専門家による評価
- ② 第三者専門家による評価
- ③ ユーザによる評価
- ④ レビューアによる確認

[データにより評価方法を確認]

- ① 知識の出典、結果の根拠の確認
- ② 過去の事例による対比テスト

[手法により評価方法の確認]

- ① 統計的手法を用いて解析
- ② プロトコル解析

[確認用ツールにより評価方法の確認]

- ① 数値シミュレーション
- ② ハードウェアシミュレータ
- ③ 実装置による疑似故障

2.5.4 要求仕様に対する達成度評価

(1) 定義

エキスパートシステムが要求仕様で規定した要件をどの程度達成しているのかを評価することである。

(2) 目的と必要性

要求仕様は、ユーザニーズを規定したものである。要求仕様に規定されている要件を達成していないシステムは当然ユーザの満足度が得られず、実用されない可能性が高い。

(3) 説明

要求仕様に対する達成度評価は、エキスパートシステムの開発フェーズと開発フェーズごとに作成される要求仕様に依存する。図2.5-1にエキスパートシステムの開発フェーズと要求仕様の関係を示す。エキスパートシステムの開発では、各開発フェーズにおいて要求仕様書が作られる。初期段階で不十分であった仕様検討が、プロトタイピングのフェーズが終了した段階で詳細まで決定できるようになる。従って、図2.5-1中の実用版要求仕様書が通常のウォーターフォール型のシステム開発における要求仕様書に相当する。この要求仕様書は実現性が充分保証されたものであり、この段階でシステムの詳細設計が可能となる。

開発フェーズ	調査・企画	プロトタイピング		実用システム開発		運用テスト	運用・洗練化
		実験型プロトタイプ	発展型プロトタイプ	改善			
				再開発			
要求仕様	企画提案第0仕 書(要求仕様書)	第Ⅰ版仕様 要求書	第Ⅱ版仕様 要求書	実用版仕様書		——	変更要求 仕様書

図2.5-1 エキスパートシステムの開発フェーズと要求仕様書

《プロダクトに対する評価》

(A) 要求仕様内容に対する評価

(1) 定義

要求仕様として記述すべき項目が漏れなく記載され、その内容が充分にかつ理解しやすく記述されているのか否かを評価することである。

(2) 目的と必要性

記載内容が不十分であれば、どのようなシステムをユーザは望んでいるのかのイメージが掴みにくいため、システム構築する際に支障を来す。

(3) 説明

プロトタイプ段階では、完全な要求仕様が存在することは少なく、ラフな機能要件がまとめられている程度である。しかし、プロトタイピングの進展と共に、次第にシステムの目的、及び内容が詳細化、明確化されるものとする。その終盤には、機能・性能を含む完全な要求仕様書が作成されるべきである。そしてそれに基づき実運用システムが開発されることとなる。

要求仕様の種類には、開発フェーズに沿って、以下の5種類が存在し、各要求仕様書は、各開発フェーズの最初の作業で作成するものとする。

企画提案書（第0版要求仕様書）（調査・計画フェーズ）

第Ⅰ版要求仕様書（実験型プロトタイピングフェーズ）

第Ⅱ版要求仕様書（発展型プロトタイピングフェーズ）

実用版要求仕様書（実用システム開発フェーズ）

変更要求仕様書（運用・洗練化フェーズ）

① 企画提案書

企画提案書では、プロジェクトを開始するに当たって、目的とするシステムの概括な要求仕様書（第0版要求仕様書）を作成する必要がある。企画提案書では、厳密な要求定義は必要ないが、実現可能な範囲での以下の項目を満たすドキュメントを作成する必要がある。

a. システム化の目的

システムの概要、詳細な内容、狙い、特記すべき事柄の他にシステムの位置づけ等も明確にしておく必要がある。

b. プロジェクト要件（開発費、納期、要員計画）

システム構築の難易度にもよるが、Ill-Definedな問題を扱うシステムで

は、開発工数の見積が非常に難しい。従って、開発フェーズに従った工程スケジュールを策定して、それぞれのフェーズごとに開始時期、開発工数などを見積る必要がある。開発費では人件費の占める割合が多いが、使用するハードウェアにも依存する。特にプロトタイピングフェーズと実用システムで使用するハードウェアを変更する場合には、開発費も増大するので、充分検討しておく必要がある。また、プロトタイピングのフェーズ終了時には、最悪の場合プロジェクトを放棄させる場合もあるので、プロジェクトの終了条件も検討しておく必要がある。

c. 実現したい機能

実現したい機能とは、エキスパートシステムで最も重要な機能要求を記述するものである。利用者からエキスパートシステム構築後に不満が出ないようにするには、利用者や知識提供者（専門家）と充分議論して機能要求をまとめあげる必要がある。この段階で、利用者や専門家の希望する機能を見通しもなく受け入れると失敗の原因となるし、精一杯努力して構築できたとしても利用者から不満がでる原因となる。充分議論をして、エキスパートシステム実現の可能性を念頭において機能要求をもとめる必要がある。

実現の可能性に関しては、次の項目を満たすことを目安に検討する。

- 実際に問題解決に当たっている専門家がいますか。
- 専門家なら数週間以内にとけるか。
- 解法が言葉で論理的に説明できるか。
- 広い範囲に一般常識を必要としないか。
- 人間の感覚器（パターン認識等）に近い判断を必要としないか。
- 学説が幾つもあって定説がないか。

d. 導入効果

システム構築時の導入効果をできるだけ定量的に評価できるようにしておく必要がある。定量的評価項目に設定は非常に難しいが、エキスパートシステム構築前に、以下に視点から検討しておくといよい。特に幹部に報告するときは金額換算するといよい。

• 省力化（自動化）

何人人員を削減できるか。あるいは、何%の自動化ができるかなどを人的面から定量的に捉える。

- 時間の短縮

従来の手法（人手）に比べて、どの程度短時間に製品を開発できるようになるかなどを時間効果として捉える。

- 精度（歩留り）の向上

診断の精度や運転精度の向上など、従来のやり方と比べてどのくらい向上するかを定量的に捉える。

- 宣伝効果

対外的な宣伝効果を広告費で換算し、定量化して評価する。

- 知識の伝承

知識の伝承の効果を教育期間で換算し、定量化して評価する。

② 第Ⅰ版要求仕様書

実験型プロトタイプリングフェーズでは、企画提案書に書かれた第Ⅰ版要求定義のシステム化の目的、プロジェクト要件、機能要件、導入効果の各項目を詳細化すると共に、おもに機能面に重点をおいた機能確認を行う。機能確認は、専門家から知識の獲得を行いながら機能面での抜けや実現の難しい機能について洗いだしを行っていくものである。

③ 第Ⅱ版要求仕様書

発展型プロトタイプリングフェーズでは、おもに性能面や実現不可能な機能に対する代替案、人間との関わり具合など、実用システムを構築するのに必要な項目を検討していく。

従って、第Ⅰ版要求仕様書のシステム化の目的、プロジェクト要件、機能要件、導入効果の各項目を詳細化すると共に、以下の項目が付け加えられる。

a. 性能要件

特に制御問題などで時間的な制約条件の厳しいシステムでは、場合によっては、処理効率のよい言語に書き直す必要がある。書き直した場合、性能の見積りと書き直した機能のトレードオフ（場合によっては機能ダウン）等について仕様書に反映させる。

b. 信頼性

システムの信頼性を保証するためには、事前に検証方法を検討しておく。例えば、運転制御システムでは、専門家との並行運用期間、診断システムにおけるテストセットの決定法などを検討する。これらの実用システム用の要求仕様に基づくシステム設計が出来ないと判断した場合は、この段階

でプロジェクトの継続を中止する。

④ 実用版要求仕様書

実用システムの開発では、プロトタイプリングフェーズの結果作成されたシステム開発の経験を通じて、実際の業務で使用するシステムの構築を行う。実用システムの開発に当たり、プロトタイプシステムを有効利用して改良していく場合と、実用システムの要求仕様に合わせて言語・ツール、推論機構を見直して再構築する場合がある。

実用版要求仕様書は、従来まで作成してきた要求仕様を見直し、ユーザの要求する機能・性能を詳細化、完全化を目指し、その上でユーザの要求を凍結して作成することになる。特に要求仕様書で規定した性能要件はテスト可能であること。記述内容は以下の通り。

システム化の目的

プロジェクト要件（開発費、納期、要員）

機能要件（要求される機能、想定される入力データの範囲・出力結果など）

ユーザ要件（導入効果、ユーザインタフェース）

性能要件（応答時間、正確性、信頼性、保守性、使用容易性など）

環境要件（資源、ハードウェアの制約など）

以上述べた項目が漏れなく記述され、その内容が充分かつ理解しやすく記述されていることである。

なお要求仕様の内容に対する評価項目は以下の通りである。

- ・ 正当性(correctness) : 要求仕様が期待と合致している度合。
- ・ 完全性(completeness) : 要求仕様の記述に誤りや脱落がないこと。
- ・ 一貫性(consistency) : 要求仕様全体を通して矛盾のないこと。
- ・ 伝達性(communicability): 要求仕様を誤りなく読者に理解させる度合。
- ・ 連絡性(traceability) : 関連する文書内の各要素との照合が容易に出来る度合。
- ・ 実現可能性(implementability): 要求仕様に合致する製品を製作し、実現できる度合。

⑤ 変更要求仕様書

変更要求仕様書では、運用テストを積み重ねながら、システムの保守性、変更の容易性、拡張性について具体的な知識の洗練化仕様を設定する。

当然これらの項目は、予め知識ベースの構造を決定する段階でも充分考慮すべきである。更に、運用テストで得られた知見、経験を踏まえて上記の項目の評価結果を実現することになる。

(B) 要求仕様に対する達成度評価

(1) 定義

エキスパートシステムが要求仕様で規定した要件をどの程度達成しているのかを評価することである。

(2) 目的と必要性

要求仕様は、ユーザニーズを規定したものである。要求仕様に規定されている要件を達成していないシステムは当然ユーザの満足度が得られず、実用されない可能性が高い。

(3) 説明

要求仕様に対する達成度評価は、プロジェクト要件に対しては企画提案書（第0版要求仕様書）、機能・性能要件に対しては第Ⅱ版要求仕様書、実用版要求仕様書を中心に達成度評価を行うことになる。

要求仕様に対する達成度評価に際しての留意点は以下の通りである。

- ・システムの目的に沿うシステムが開発されたか。
- ・開発費、納期、要員の割当てにおいて特に問題点はなかったか。
- ・要求した機能を満たしたシステムが出来たか。
- ・応答時間は満足がいくか。
- ・期待した解を求めることが出来たか。
- ・使いやすいシステムか。
- ・資源、ハードウェアの環境条件に問題はないか。

など

(→本評価項目は、「利用者の満足度評価」と関連する。)

《開発／保守方法・技法・ツールに対する評価》

(C) 要求仕様設定手段の評価

(1) 定義

要求仕様を設定する手段が適切であるのか否かを評価することである。

(2) 目的と必要性

要求仕様を取りまとめる前に、記述する方法や手段を充分検討しておかないと、ユーザの要求するニーズを的確に表現できなかったり、要求仕様を固める際の作業効率に影響を及ぼす。

(3) 説明

要求仕様を作成する際、専門家の主観に依存したドメインが対象となるため、システム分析の手法・技法をそのまま適用できるわけではなく、観点の依存性、異なる観点がある場合その関連性を考慮する必要がある。

要求仕様を設定するための方法は以下の通りである。

① ニーズ抽出法

- ・ 5 W 1 H 法

② 対象の構造分析をする方法

- ・ ブロックモデル
- ・ 階層構造モデル

③ 要求分析をするためのツール（例）

- ・ I S M (Interpretive Structural Modeling)
- ・ T U P P S (Tool, User and Project-Planner System)
- ・ C - N A P (Customer-Needs Analysis Procedures)
- ・ P P D S (Planning Procedure to Develop Systems)

④ 要求モデル

- ・ 有限状態機械
- ・ データフローモデル
- ・ 刺激応答パスモデル
- ・ 機能階層モデル
- ・ 並行プロセスモデル
- ・ データ指向モデル

(E-Rモデル、オブジェクトモデル、関数的仕様、述語論理的仕様、代数的仕様、公理的仕様、操作的仕様)

2.5.5 知識ベースの保守性評価

(1) 定義

エキスパートシステムの機能あるいは性能の不足、誤りが明らかになったとき、利用者の要求が変化したときに、それらに対応して、知識ベースを保守する必要がある。知識ベースの保守性評価とは、知識ベースが、この保守作業の遂行にどの程度適しているかを評価することである。

(2) 目的と必要性

エキスパートシステムでは、設計段階で要求機能を厳密に決定することが困難であるため、実稼働後の機能拡張、修正が多い。この機能変更は主として知識ベースの保守により実現される。そこで、知識ベースの保守性はエキスパートシステムの重要な評価項目である。また、この評価は、システムの改良や次期システムの開発の指針を得るためにも必要である。

(3) 説明

知識ベースの保守が従来ソフトウェアと異なる点は、対象分野の専門家が作業に参加することである。そこで、専門家による保守を円滑化することが知識ベースに特有の問題となる。

保守の手順としては、まず、エキスパートシステムの変更をしたい機能あるいは強化したい性能に対して、知識ベース中の変更すべき部分を見いだすことから始まる。つぎに、その部分を理解した後、必要な拡張、修正を行い、テストを行う。さらに、保守に伴って予期せぬ誤りが作り込まれていないこと（品質の安定性）を確保するために、システム全体の連動テストを行う場合もある。その過程で、各種のマニュアル、ツールを利用する。

そこで、評価項目として、変更箇所発見の容易性、理解性、修正の容易性（すなわち、変更箇所の局所性）、品質の安定性の確認方法、保守マニュアル及びツールの整備状況を取り上げた。

以上は、保守担当者からみた場合に必要となる評価項目であるが、一方、知識ベースのモジュラリティ、統一性（一貫した方針で作られていること）など、内部構造の観点からの評価項目も必要である。

以下では、専門家による保守を考慮しながら、上記項目について説明する。

《プロダクトに対する評価》

(A) 変更箇所発見の容易性評価

(1) 定義

変更したい機能に対して、知識ベース中の変更箇所がどの程度容易に見いだせるか、あるいは、知識ベースの一ヶ所を変更したときに、関連する変更箇所をどの程度容易に見い出せるかを評価することである。

(2) 目的と必要性

変更箇所発見の容易性は、保守に伴う誤りの作り込みを防止するため、また、保守の効率確保のために重要である。評価結果は、知識のモジュール分割の改良に反映される。

(3) 説明

変更箇所発見の容易性は、エキスパートシステムの外部機能と内部機能の対応、知識モジュール間の独立性に依存する。関連する評価項目は以下の通りである。

- ①モジュール分割が、システムの外部機能に対応しているか。
- ②モジュール分割が、一まとまりの推論処理に対応しているか。
- ③従来言語で記述された部分については、従来のソフトウェア工学におけるモジュラリティ向上の方法論が守られているか（例えば、情報隠蔽、同じ機能を一ヶ所にまとめることなど）。
- ④ドキュメントは整備されているか。

(B) 知識ベースの理解性評価

(1) 定義

保守担当者（専門家、KE、ユーザなど）が、知識ベースの内容をどの程度容易に理解できるかを評価することである。

(2) 目的と必要性

理想的には、知識ベースの内容を理解しなくても知識が追加でき、修正部分のみ理解すれば修正できるのが望ましい。しかしながら、そのためには、知識の冗長性や矛盾をシステムが自動的に検出、修正する必要がある、現状のエキスパートシステム技術ではこの理想は実現できない。従って、保守に当たっては、保守部分を含めた一定範囲の知識を理解する必要がある。そこで、知識ベースの理解性を評価項目として取り上げる。評価結果

は、システムの改訂や次期システムの開発時に、対象世界のモデル、知識のモジュール分割、記述方法の改良に反映される。

(3) 説明

前述のように、知識ベースの保守には、KEの他、対象分野の専門家が参加する。特に、知識ベースが複数の分野の知識を含む場合には、異なる分野の複数の専門家が参加する。専門家は通常プログラムの知識や技術がないため、以下に述べる知識ベースの保守特有の評価項目があげられる。

- ① 知識ベースの各モジュールが、保守担当者を明確に想定した上で開発されているか。
- ② 専門家による保守を想定したモジュールは、プログラムの知識がなくても理解できるようになっているか。
- ③ 複数分野の知識は、適切なモジュール分割が行われているか。

知識ベースもソフトウェアの一種であるので、プログラムの分かりやすさに関する一般論は成り立つ。すなわち、記述の対象となる実世界が知識ベース中に素直にマッピングされているほど、理解が容易である。そこで、以下の評価項目があげられる。

- ④ 対象世界のモデルが素直で、分かりやすいか。
 - ⑤ 知識ベースが分かりやすいか。
 - ・ 知識ベースのモジュール構造がモデルに忠実で、機能と構造が対応しているか。
 - ・ 知識表現の選択が適切か。
 - ・ 個々のルールやフレームに記述が分かりやすいか。
 - ・ 従来言語で記述した部分については、記述のレベルが高いか。
- 上記の他、保守ドキュメントの整備も重要である。

(C) 変更箇所の局所性評価

(1) 定義

エキスパートシステムの性能、対象範囲の拡張や修正に対して、知識ベース中の変更箇所がどの程度限定されるかを評価することである。

(2) 目的と必要性

変更箇所の局所性は、保守の効率を確保するため、及び保守に伴う誤りの作り込みを防止するために重要である。また、専門家による保守を円滑

化するためには、変更箇所が専門家の担当範囲を越えて波及しないこと、例えば従来言語で記述されたプログラム部分に及ばないことが重要である。そこでこの性質を評価項目として取り上げた。評価結果は、知識のモジュール分割の改良に反映される。

(3) 説明

専門家による保守を円滑化することに関連し、以下の項目があげられる。

- ① 知識ベースの各モジュールが、保守担当者を明確に想定した上で開発されているか。
- ② 分担の異なるモジュール間のインタフェースが明確に規定されているか。
一方、知識ベースには教科書的知識のように変更の少ないものと、経験的知識のように比較的変更の多いものがある。そこで変更範囲を局所化するためには、変更の多い部分とをきり分けておくことが有効である。即ち、
- ③ 変更の少ない知識と多い知識がきり分けられているか。

さらに、ソフトウェア全般の評価項目として、モジュラリティに関するものなどがある。これらは、(A)の変更箇所発見の容易性で述べた①～④と同様である。

(D) 知識ベースの統一性評価

(1) 定義

知識ベースがどの程度一貫した方針で作られているかを評価することである。

(2) 目的と必要性

保守が長期にわたる場合や保守の担当者が複数である場合は、知識の競合や矛盾を避けることは困難である。しかし知識ベースが一貫した方針で作られていれば、その方針に従って保守を行うことにより、誤りの発生を減らすことができる。また担当者のレベルがまちまちである場合にも、ある程度知識ベースの質を保つことができる。そこで、知識ベースの統一性評価項目をとりあげた。

本評価結果は、システムの改訂や次期システムの開発時に、対象世界のモデルや知識のモジュール分割に反映される。

(3) 説明

(知識ベースだけでなく、何事に関してもいえることだが) 統一性は、方針が明確かつ一貫しているかどうか、方針が周知徹底されているかどうかによって依存する。以下に関連する評価項目をあげる。

①対象世界のモデルが明確かつ一貫しているか。

②ソフトウェアがモデルに忠実で一貫しているか。

- ・モジュール分割の観点が一貫しているか。
- ・知識の種類に対する知識表現の選択が一貫しているか。
- ・記述(名前やデータ構造の選択)が一貫しているか。

《開発/保守方法・技法・ツールに対する評価》

(E) 品質安定性の確認方法の評価

(1) 定義

保守に伴って予期せぬ誤りが作り込まれていないことを検証するための方法が確立されているかどうかを評価することである。

(2) 目的と必要性

エキスパートシステムでは、断片的な知識が実行時に連動するため、知識の修正に伴う副作用を発見、修正することが、従来ソフトウェアに比べ困難である。そこで品質の安定性を確認することが重要であり、そのための方法が確立されている必要がある。評価結果は、保守マニュアルに反映される。

(3) 説明

保守前のシステムに対するテストを再度行うなどの方法が確立され、それが保守マニュアルなどに記載されているかどうかを評価する。

(F) 保守マニュアルの整備状況の確認

(1) 定義

保守に必要なマニュアルが整備されているかどうかを確認することである。

(2) 目的と必要性

保守マニュアルの整備は、保守の効率確保、及び知識ベースの質の確保のために重要であるから、評価項目として取り上げた。評価結果は、保守

マニュアルの拡充に反映される。

(3) 説 明

以下の評価項目があげられる。

- ① 必要な保守マニュアルが揃っているか（例えば、各種保守作業の手順マニュアル、ツールの操作マニュアルなど）。
- ② 各マニュアルには必要事項が正確に記載されているか（保守環境、作業、操作、事後の確認、記録の方法など）。
- ③ 標準にそった一貫した記述になっているか。
- ④ 読者（保守担当者（専門家、KE、ユーザなど））を明確に想定し、想定読者が理解できるような説明になっているか。

(G) 保守用ツールの整備状況の確認

(1) 定 義

保守に必要なツール類が整備され、容易に使用できる状況になっているかどうかを確認することである。

(2) 目的と必要性

保守用ツールの整備は、保守の効率確保、及び知識ベースの確保のために重要であるから、評価項目として取り上げた。評価結果は、ツールの拡充に反映される。

(3) 説 明

知識ベースの保守に特有のツールとしては、専門家用ツールがある。どのような専門家用ツールが望ましいかは一概に決められないが、一般的にはプログラムの知識がなくても操作でき、専門家が日頃親しんでいる表現（ワークシートや自然言語）や用語により知識を保守できるツールが望ましい。専門家用ツールの種類としては、以下のようなものがある。

- ① 分野別知識エディタ（ワークシートや自然言語による入力、セマンティックチェックなどの特徴がある）
- ② 説明ツール（推論過程の説明など）
- ③ 解析ツール（ルールやフレームの関連表示）
- ④ テストツール（ルールのステップ実行、テストデータの管理、再実行など）

2.5.6 マンマシンインタフェースの妥当性評価

(1) 定義

エキスパートシステムのマンマシンインタフェースが、利用時、保守時からみて使いやすくできているのかを評価することである。

(2) 目的と必要性

マンマシンインタフェースは、エキスパートシステムやES構築ツールの顔であり、提供されている機能がいかにもすぐれていても、あるいは、性能が良くても、マンマシンインタフェースが良くないと使われない運命をたどる可能性が非常に大きいので、マンマシンインタフェースは極めて重要である。

(3) 説明

利用時インタフェース、保守時インタフェースの違いを以下に述べる。

① 利用時インタフェースの範囲・利用対象

- ・開発されたエキスパートシステムを一般利用者が利用する局面
- ・適用業務の利用者（専門家または非専門家）が利用対象

② 保守時インタフェースの範囲・利用対象

- ・知識ベース保守時は、適用業務の利用者、ナレッジエンジニアが利用対象

《プロダクトに対する評価》

(A) 利用時のインタフェース評価

(1) 定義

開発されたエキスパートシステムの利用者インタフェースの良さを評価することである。

(2) 目的と必要性

利用時インタフェースが悪いとエキスパートシステムの実用性が失われる可能性が高く、その評価は重要である。

(3) 説明

エキスパートシステムのマンマシンインタフェースには、一般の情報処理システムと同様の面とエキスパートシステム特有の面とがある。以下に主要なマンマシンインタフェースの評価項目を示すが、このうち「説明機

能」はエキスパートシステム特有の項目である。

①表示機能と操作性

- a. 結果の表示や判断・応答の入力補助情報の表示において適切な表現方法が用いられているか。
 - ・文脈 (Context) の表示方法・順序
 - ・情報の内容に応じた表現形式 (図、グラフ、リスト、グラフィック、色)
 - ・情報の重要度に応じた表現
 - ・一覧あるいは関連情報への容易なアクセス
 - ・標準への遵守性 (情報の正しい解釈を助ける)
- b. 入力の操作性が良いのか。
 - ・人間工学的操作性
 - ・標準への遵守性 (操作の誤りをなくす)
 - ・入力機器・媒体の選択
 - ・長時間使用しても身体的な苦痛が少ない

②ガイダンス機能

- a. HELP機能の充実と操作性
 - ・文脈による必要最小限のHELP
 - ・利用者主導の能動的なHELP
 - ・操作誤り等による受動的なHELP
- b. 明確な文脈の表示 (明示／暗示)
 - ・質問の意味
 - ・表示情報の意味
 - ・業務実施中としての認識 (トランザクション処理の意味)
 - ・次に何をすべきかが理解できる
- c. 利用者の習熟度に応じたガイダンス
 - ・習熟の速さ・正確さ
 - ・習熟度に応じた表現

③エラー処理

- ・エキスパートシステム自体の不備、不完全性によるエラー処理
- ・利用者の操作誤りによるエラー処理
- ・エラーの認知機能

- フェイルソフト／フェイルセーフ機能
- 回復機能
- 再開始機能
- 代替切り替え機能

④ 自然言語機能

a. 自然言語処理機能の意義

- 曖昧さや不確定性の表現方法として
- より理解しやすい結果の説明機能として
- 標準インタフェースとして（コマンド／パラメータより自由な日本語である）

b. 使用局面

- 質問の生成
- 結果の表示
- 応答入力
- ガイダンス／説明機能の実現

⑤ 処理速度

- 全体の処理時間
- 応答時間

⑥ 説明機能

次のような機能の必要性和充実度

a. WHY

- 何故このような質問をするのかを説明する
- 現在の文脈を理解させる
- 正しい応答入力を促す

b. HOW

- どのような推論過程でそのような結論に至ったのか
- 結果の妥当性を説明する
- 結果の正しい解釈を促す

c. WHAT

- より詳しい質問の内容の表示
- 利用者の習熟度に応じた質問方法

(B) 保守時のインタフェース評価

(1) 定義

エキスパートシステムの知識ベースの保守を行う場合のインタフェースの良さを確認することである。

(2) 目的と必要性

エキスパートシステムにおける知識ベースの保守は運用段階に入っても継続され、また開発者のみならず専門家を始めとする利用者自身によって保守が行われるケースも多い。保守時のマンマシンインタフェースは保守作業の生産性と質を向上させるため極めて重要である。

(3) 説明

知識ベースの保守では、その目的の支援ツールを用いる場合と、エキスパートシステム自体が持つその機能を用いる場合がある。いずれの場合もマンマシンインタフェース上の評価のポイントは以下の通りである。

- ① 知識ベースの可視性を高める機能を持つか。
- ② 変更すべき箇所の同定を容易にする機能を持つか。
- ③ 専門家の知識表現によって入力した知識をシステム固有の表現に変換する機能を持つか。
- ④ 保守時の知識ベースの品質低下が起こらぬよう、矛盾や一貫性等を検査する機能を持つか。
- ⑤ 既存のテストデータによる回帰テストが行える機能を持つか。

なお、表示機能、操作性、ガイダンス機能など一般的なマンマシンインタフェースに関しては「利用時のインタフェース評価」と同様である。また保守を担当する人が、開発者か利用者により、それぞれの専門性やレベルに応じたインタフェースであることが必要不可欠である。

- 解法そのものが厳密に定義しがたい、あるいは解法は分かっているが複雑に入り組んでおり、アルゴリズムやデータの構造の設計が難しい問題であること（組合せの爆発が生じる問題）。
- 従来のプログラム手法で問題解決は出来るが、エキスパートシステムのプロトタイピング的アプローチを採用した開発を行いたい場合であること（開発の方法論として、プロトタイピング手法を採用）。
- 問題の領域が安定しており、開発運用中に大きな変化がないこと。
- 従来のプログラムであると、条件が頻繁に変わるためその都度の作り直しはかなわない問題であること（変更への対応と迅速化）。

③ 解の完全度の要求度

下記の何れかであれば推奨する。

- 解析型問題で100%の正解率を要求しない。
- 合成型問題で最適解を要求しない。
- 最後は人間が判断する。

④ プロトタイピングへの適用性

- システムは逐次的に成長発展が可能であること。
- 開発に時間とコストをかけられる問題であり、当初は実験システムとしての性格を持つ問題であること。

⑤ 知識獲得の成功の可能性

- 真の専門家が存在する問題であること。
- 専門家の絶大な協力が得られること（時間的、情動的にも）。
- 専門家は知識をより明確に話せること。

（B） 利用者の満足度評価

（1） 定義

利用者が対象とするシステムをどの程度満足しているのかを評価することである。

（2） 目的と必要性

実用システムとして使われるのかを見極めるための最重要評価項目は利用者の満足度であり、それを評価する必要がある。

（3） 説明

利用者の満足度を高めるには、例えば下記の項目が実現されている必要

2.5.7 システムの総合評価

(1) 定義

エキスパートシステムが情報処理システムとして総合的にみて満足できるものかどうかを評価することである。

(2) 目的と必要性

情報処理システムの評価は、部分やある視点での評価に加えて、システム全体としてのバランスや機能間の統合性の評価など総合的な観点からの評価が必要である。特に、エキスパートシステムでは知識処理的な部分とネットワークやデータベースを始め従来型技術による部分とが統合されている可能性が高く、総合性評価の必要性は極めて高い。

《プロダクトに対する評価》

(A) テーマ選定の妥当性評価

(1) 定義

エキスパートシステムとしてテーマの選定が妥当であるのか否かを評価することである。

(2) 目的と必要性

従来システムのアプローチでも充分対処できる内容のシステムを、エキスパートシステムとして構築しても十分な効果が得られぬことが多い。

テーマ選定の良否はエキスパートシステム評価の基本要件である。

(3) 説明

実用化を狙うエキスパートシステムでは、「エキスパートシステム化へのニーズの高さ」、「エキスパートシステム向きの問題」、「解の完全度の要求度」、「プロトタイプ化への適用性」、「知識獲得の成功の可能性」等を考慮する必要がある。

① エキスパートシステム化へのニーズの高さ

- ・ユーザがエキスパートシステムへの投資効果が高いと感じていること。
- ・実際に使われることを想定としていること。
- ・システム化の対象範囲や制限条件を現実的に把握が可能であること。

② エキスパートシステム向きの問題

- ・専門家の経験的な知識を記号処理により問題解決できる問題であること。

がある。

- 欲しい機能が実現され、提供されている。
- 操作が簡単である。
- レスポンスが速い。
- 入力データの準備に時間がかからない。
- 必要最小限のデータを入力すれば済む。
- 出力結果が視覚化されている。
- 出力結果の信頼性が高い。
- 説明機能があり、分かりやすい。
- 学習機能があり、操作性が向上する。
- 再開機能があり、利用を中断することが出来る。
- システムを使うことによって作業の生産性が向上する。
- システムを使うことによって作業の質が向上する。
- 営業において、システムが差別化の手段となる。
- ユーザの改善要望を聞き、改善・機能拡張がタイムリーに行われている。
- 教育プログラムが充実し、随時実施されている。
- ユーザマニュアルが充実し、分かりやすい。

以上のような項目を、満足度を決定付ける重要な因子、例えば、操作性、機能性、解の信頼性、戦略性などに対応づけ、レーダチャートやフェース分析 (Face Analysis) の形式で表現して、ユーザの満足度を総合的に評価する方法がある。

(→本評価項目は、「要求仕様に対する達成度評価」、「利用時のインタフェース評価」、「システムのコストパフォーマンス評価」と関連する。)

(C) 統合性の評価

(1) 定義

従来型技術による機能との連携や統合性の良さを評価することである。

(2) 目的と必要性

知識処理的機能と従来型機能との統合は、実用的なエキスパートシステムでの必須条件であるが、両者の最適な分担や効率上の負荷を最小限におさえるなど課題も多く、その統合性に対する評価は必要不可欠である。

(3) 説明

多くの実用段階でのエキスパートシステムは、スタンドアロンのシステムとしてではなく、既存の機能や技術と連携して、双方の良い点を生かして実用性の高いシステムにまとめ上げている。例えば、通信機能、データベース、統計処理、数値計算、画像処理などの従来技術による機能との統合が行われることが多い。

これらの統合化に際しては、既存機能にできるだけ手を加えたくないという配慮が働くことが多く、その結果、効率やメモリスペース上の負荷が過度になる、部分的機能の重複がおこるなど、望ましくない結果となる可能性がある。特にデータベースと知識ベースの統合は両者のデータ構造がかなり異なるため、一般にその間のインタフェースとして変換機能が必要となり、その品質が問題となる。

既存機能との統合化は、次の4つの形態に大きく分けられる。それぞれの形態毎にその連携がよいのかどうかを評価する。

① エキスパートシステムから既存機能と呼び出す形式

例) 各種パッケージのCALL

データベースの利用

画像制御ルーチンの利用

通信機能の利用

② 既存システムのフロントエンドとしてエキスパートシステムを実現する形式

この場合、ユーザに対し知的なユーザインタフェースを期待することが出来る。

例) データベースの知的検索

知的ユーザインタフェース機能

③ 既存システムの一部をエキスパートシステム化して、付加価値を高めた形式

例) プロセス制御の監視機能部分

④ 既存機能との協調型の形式

既存機能と知識処理機能が別タスクで起動

(D) 成長可能性の評価

(1) 定義

エキスパートシステムの対象範囲の拡大や機能の向上など、システムの成長可能性を評価することである。

(2) 目的と必要性

エキスパートシステムはプロトタイピングにより部分的な試作を積み重ねて対象範囲を拡大して行くが、その成長の可能性が明確でないと、開発の中断や、使用の中止などに至る可能性がある。これらの即断を避けるためにも、その成長の可能性を見極めることは重要である。

(3) 説明

エキスパートシステムにおいては、一般に開発が完了すれば実用に耐えられるシステムが完成すると考えることが出来ず、運用中も知識が拡張されシステムが成長することを前提としなければならない。

従って、現時点でのエキスパートシステムの能力は、専門家と比較して幼稚であり劣っているとの評価結果が出された場合でも、エキスパートシステムの成長可能性を忘れてはならない。今後、知識を追加、修正し増大させて行くことにより、エキスパートシステムの判断能力などを成長発展させる見込みがあるのかがキーポイントである。

評価の視点は、以下の通りである。

- ① 知識提供の専門家、開発担当者、意思決定するマネジャ、第3者の専門家等からなるレビュー会議において、エキスパートシステムの成長可能性について充分検討したか。
- ② 特に、プロトタイピングシステムの段階において、エキスパートシステムの成長の可能性について充分レビューしたか。
- ③ 現時点でのエキスパートシステムの能力は専門家と比べて幼稚であり、劣っていると評価された場合でも、エキスパートシステムの成長の可能性を考慮して総合判断をしたか。

(E) 開発／保守用ドキュメントの評価

(1) 定義

開発用ドキュメント、保守用ドキュメントがどの程度完備され、その内容がどの程度充実しているのかを評価することである。

(2) 目的と必要性

知識ベース、推論機構の正しさを調べる際、開発ドキュメントをレビューして確かめる必要があり、その評価は重要である。

(3) 説明

ドキュメントの充実の必要性は、システムの状態により異なるが、少なくとも実用システムにおいてはその完備は必要不可欠であり、またドキュメントによるレビューも重視される。

評価項目は以下の通りである。

- ・ドキュメント形式が標準にあっているか。
- ・ドキュメント内容は、矛盾がなく意味的に正しいか。
- ・ドキュメントは、分かりやすくかつ誤解がないように書かれてあるか。
- ・ドキュメントをレビューすることにより、そのエキスパートシステムの機能・性能の妥当性及び開発上の問題点等が充分指摘できるか。

各開発フェーズで作成されるドキュメントの体系は以下の通りである。

① 調査・計画フェーズ

a. 調査報告書

- ・現状システムの分析
- ・他社システムの現状
- ・AI技術の現状

b. 企画提案書

- ・システム化の目的
- ・プロジェクト要件（開発費、納期）
- ・実現したい機能
- ・導入効果

② 実験型プロトタイプリングフェーズ

a. 第I版要求仕様書

(※1)

- ・システム化の目的
- ・プロジェクト要件（開発費、納期、要員）

- 機能要件（実現したい機能）
- ユーザ要件（導入効果）
- 性能要件（品質・性能）

b. 第Ⅰ版知識分析書 (＊２)

- インタビューメモ
- ケーススタディメモ
- 用語定義
- 運用イメージ
- 定式化
- 専門家モデル

③ 発展型プロトタイプリングフェーズ

a. 第Ⅱ版要求仕様書 (＊１と同じ)

b. 第Ⅱ版知識分析書 (＊２と同じ)

c. 第Ⅱ版プロトタイプ設計書 (＊３)

- 知識ベース設計書（ルール、フレーム、メソッド等）
- 問題解決・推論方式設計書
- マンマシンインタフェース設計書

d. 第Ⅱ版プロトタイプテスト文書 (＊４)

- テスト計画書
- テスト仕様書
- テスト結果書

④ 実用システム開発（改善、再開発）フェーズ

a. 実用版要求仕様書 (＊１と同じ)

b. 実用版知識分析書 (＊２と同じ)

c. 実用版設計書 (＊３と同じ)

d. 実用版テスト文書 (＊４と同じ)

⑤ 運用テストフェーズ

a. 運用テスト文書 (＊４と同じ)

b. 運用マニュアル

- サービス体制
- ユーザマニュアル
- 保守マニュアル

⑥ 運用・洗練化フェーズ

a. 変更要求仕様書 (※1と同じ)

b. 運用実績書

- ・トラブル、クレームとその処置

- ・結果の統計的解析結果

c. 保守報告書

d. 改訂版ドキュメント一式

⑦ その他

今後、マニュアルの整備にともない使いやすくするために、以下の方法を検討する必要がある。

- ・オンラインマニュアル

- ・多インデックス検索機能

- ・マルチメディア／ハイパーカード

- ・HELP機能との一体化

- ・知的チュートリアル

《開発／保守方法・技法・ツールに対する評価》

(F) 開発方法の適切さの確認

(1) 定義

開発方法が適切であったのかどうかを確認することである。

(2) 目的と必要性

システムの開発方法は、開発の生産性や成果物の品質に大きな影響を与えるため、その評価は重要である。

(3) 説明

開発方法を規定する要素は次のようなものである。

- ・開発工程の分割

- ・開発方法論（技法）、及び利用ツール

- ・各工程における作業内容

- ・工程毎のドキュメントの種類と内容

- ・各工程の開発管理（レビュー、完了の確認）方法

有効性が実証された開発方法を採用し、そこで規定された通りに作業を行えば、効率よくかつ高質な作業を実施できる可能性が高くなる。

開発方法の適切さに関しては、下記の点を評価すべきである。

①作業関連

- 工程分割の妥当性
- 開発方法論の適切性
- 各工程で行うべき作業内容の適切性
- 各工程で使用する技法、ツールの適切性と有効性

②管理関連

- 開発管理の基本方針
- 各工程の管理方法

③ドキュメント関連

- ドキュメント体系
- ドキュメントの記述基準
- 各ドキュメントの記述内容

(G) ツール・言語選択の適切さの確認

(1) 定義

ツールまたは言語選択の選択が適切であったのかどうかを確認することである。

(2) 目的と必要性

ツールまたは言語の選択は、知識表現や問題解決・推論方式を決定することとなるので、その選択の良否を評価することは重要である。

(3) 説明

エキスパートシステム開発において、実用レベルでのポイントとなるのがツール・言語の選択である。開発全体に与える影響が大きく、各ステップ毎に見直しが要請される。その結果プロトタイプ版と実用版で異なるものを用いる場合も多い。

ツール・言語の選択肢は次のように分類できる。

①ハイブリッドツール

②領域特化シェル

③AI言語

④従来型言語

ツールの評価の要点は以下の通りである。

- ① ツールの機能／知識表現
- ② 開発者インタフェース
- ③ 利用者インタフェース
- ④ システムインタフェース
- ⑤ ツールの性能
- ⑥ ツールの利用目的
- ⑦ ツールが得意とする分野
- ⑧ 開発担当者のスキルと開発期間
- ⑨ 他システムとのリンク
- ⑩ 移植性、保守性
- ⑪ 開発環境／デリバリ環境

(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、
第2分冊：知識システム構築ツールの評価、平成元年3月を参照)

《その他の評価（経済性、社会性）》

(H) システム開発の生産性評価

(1) 定義

エキスパートシステムの開発作業の生産性について評価することである。

(2) 目的と必要性

エキスパートシステムの開発には、プロトタイピング手法の利用、ツールの活用などの特徴がある。それらの効果の度合を明確にすることは、その後の開発業務の企画や実施において貴重な情報となる。

(3) 説明

エキスパートシステムは、それが用いてる基本的な手法によって、「完全な知識＋演繹推論」(Well-structuredな問題) (以後、タイプⅠと呼ぶ) と、「不完全な知識＋高次推論」(Ill-structuredな問題) (以後、タイプⅡと呼ぶ) に分けられる。

タイプⅠのエキスパートシステムは、通常、ルールベースシステムやフレームベースシステムを採用して構築されるが、従来のソフトウェア開発技術でも十分に構築可能であるので、従来技術に比べ、エキスパートシステム開発技術を採用したことにより、どの位生産性が向上したかの観点で

定量的に評価することが適切であると思われる。

タイプⅡのエキスパートシステムは、そもそも Ill-structured な問題を対象としているので、問題構造や知識の曖昧性を扱う必要があり、従来のソフトウェア開発技術では手に負えなかったり、あるいは、うまく構築することができない。従ってエキスパートシステム開発技術を採用して構築せざるをえず、従来技術と比較しての生産性という観点で評価するのは不適切である。むしろ成長性の観点をも考慮に入れて、長期的な生産性を評価することが必要である。

(Ⅰ) システムのコストパフォーマンス評価

(1) 定義

エキスパートシステムに対して投資したトータルコスト（開発コスト、運用コスト、保守コスト）を評価することである。

(2) 目的と必要性

企業においては、システムのコストパフォーマンスは重大な関心事である。情報処理システムに関しても、手法や手段によってその差が大きいと想定される場合には、適切な比較評価を行い、その後の計画に充分活用することが必要である。

(3) 説明

基本的な評価項目は、以下の通りである。

- ① エキスパートシステム導入後、以前に比べて経費は節減されたか、あるいは増加したか。
- ② 投資に見合った、作業の質が改善されたか。
- ③ 投資に見合った、生産性の向上が得られたか。

以下に示す利点及び弊害という観点で評価する。

(a) 実用システム導入による利点

- | | |
|------------|------------|
| ① 作業効率の向上 | 生産性の向上 |
| | 省力化 |
| | 専門家の減少 |
| | 作業のやり直しの減少 |
| ② 専門家の作業代行 | 専門家の代替が可能 |

- ③ 専門家の作業のサポート 専門家の負荷軽減
必要情報を入力するにかかる時間の短縮化
- ④ 作業量の増加 対処可能な業務量の拡大
- ⑤ 作業の質的改善・向上 業務の高品質化
業務の均質化
誤りの減少・防止
- ⑥ 教育 初心者教育
専門家の育成
熟練技術の伝承
知識の蓄積・標準化（普及、共有、継承が可能）
- ⑦ システム化によるメリット 知識の整理・体系化
ノウハウの知識ベース化
保守や拡張の容易化
エンドユーザによる操作の容易化
- ⑧ その他 既存の製品・サービスに対する付加価値の向上
新しいビジネスチャンス
知識工学／エキスパートシステムの技術修得、評価

（b）エキスパートシステム導入により発生する可能性のある弊害

- ① 職場モラルの低下、拒否反応
- ② 定型の結果しか得られない
- ③ 責任の所在が不明確
- ④ 技術のブラックボックス化
- ⑤ 慣れるのに手間、時間がかかる
- ⑥ エキスパートシステムによる自動的判断の危険性

(J) セキュリティ対策の評価

(1) 定義

エキスパートシステムを運用するに際して想定されるセキュリティの侵害に対し、どのような予防措置、対策が講じられているのかを評価することである。

(2) 目的と必要性

セキュリティ対策の不足により侵害を受け破壊した場合、システムへのダメージが大きいので、その評価は重要である。

(3) 説明

他の情報処理システムと同様、エキスパートシステムにおいても信頼性、安全性の両面からのセキュリティ機能の完備は必要不可欠である。

従来のシステムでは、すでにデータの消失、改ざん、漏洩などによる多くの事故が発生しており、今後ますます重要な問題に発展しつつある。特に、エキスパートシステムには貴重な知識が蓄積されており、改ざんされたり、悪用された場合には従来以上のセキュリティ問題が発生する可能性もある。

セキュリティ対策上留意すべき点は以下の通りである。

① リスク分析の必要性

特に、エキスパートシステムが社会システムやその他のシリアスな対象に用いられた場合は、まず起こり得るリスクを事前に充分検討しなければならない。例えば、解答が誤った場合、利用者の財産、生命に危具を与え、取り返しのつかない事態を招く恐れもある。

② セキュリティ対策には、技術的対策、運用上の対策、設備上の対策などがある。全ての角度から充分な対策を講ずるべきである。

③ システム自体の実行権および知識ベースへのアクセス権を厳格に設定し、充分なアクセスコントロールを実施すること。

④ 不正アクセスや改ざんが行われたか否かを検査する手段を設けること。

⑤ ネットワークのデータ伝送路上での充分なセキュリティ対策を講じること。

⑥ 知識、ノウハウに対する知的所有権、侵害を受けた場合の法的責任などについても事前に明確にしておくことが必要である。

参考文献

- 〔01〕 富士通：エキスパートシステムの構築技法、テキスト、昭和63年7月。
- 〔02〕 フレデリック・ハイズ・ロズ、ドナルド・A.ウォータマン、ダグラス・B.レナート：
エキスパートシステム、A I U E O 訳、産業図書（昭和60年8月）。
- 〔03〕 Gary Kahn, Neil Jacobstein: Managing Expert Systems Projects,
AAAI88, TUTORIAL, MP3, 1988.
- 〔04〕 富士通：E S / S D E M エキスパートシステムの構築技法、昭和63年。
- 〔05〕 阿部憲広、滝寛和：エキスパートシステム入門、共立出版、1987年10月
- 〔06〕 John Walters, Norman R. Nielsen: Crafting Knowledge-Based Systems,
JOHN WILEY & SONS, 1988.
- 〔07〕 特集エキスパートシステム、情報処理、Vol.28, No.2, (1987年2月)。
- 〔08〕 昭和63年度人工知能学会全国大会（第2回）論文集、7.21-7.23、
人工知能学会。
- 〔09〕 小林重信：知識工学、昭晃堂、昭和62年10月。
- 〔10〕 ICOT-JIPDEC AIセンター：昭和62年度人工知能技術動向等に関する調
査研究報告書、付属資料（エキスパートシステム事例）、昭和63年3月。
- 〔11〕 ICOT-JIPDEC AIセンター：昭和61年度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、昭和62年9月。
- 〔12〕 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、平成元年3月
第1分冊 計画・設計型知識システムの構築方法論
第2分冊 知識システム構築ツールの評価
第3分冊 知識獲得支援
第4分冊 知識システム技術者育成ガイドライン
第5分冊 知識システムの事例
- 〔13〕 松本吉弘：情報処理ハンドブック、Ⅱ編ソフトウェア工学、2章仕様化、
pp.967-977、1989年5月、オーム社。
- 〔14〕 秋田興一郎：エキスパートシステム導入実戦ガイド、電気書院、
昭和63年10月。
- 〔15〕 日本情報処理開発協会中央情報教育研究所（CAIT）編：高度情報処
理技術者育成指針－ナレッジエンジニア編－、平成元年3月。

- 〔16〕日立製作所：システム開発標準 エキスパートシステム構築標準手順
E S G U I D E（第1版）、昭和62年10月。
- 〔17〕上野晴樹、石塚満編：知識の表現と利用、昭和63年2月、オーム社。
- 〔18〕山口高平：分化し始めたA I ツール、TURING MACHINE、Vol.1、No.1、
1988年12月。
- 〔19〕末田直道、他：エキスパートシステム構築技法T U P P S - E S、
東芝レビュー、Vol.44、No.10、(1989)。

2.6 問題領域別の評価項目

ここではエキスパートシステムの対象とする問題を合成型問題、解析型問題、複合型問題の3つに分け、問題ごとに評価すべき内容の特徴、必要性について述べる。なお合成型問題は設計型問題と計画型問題に細分し、解析型問題は解釈型問題、診断型問題、制御型問題に細分して検討する。

問題の対象は「ドメイン」と呼ぶ。一方、対象とする問題を分析したとき、その問題を解決するのに必要な特有な作業の型がみいだせる。これを「タスク」と呼ぶ。ここでは、エキスパートシステムをタスクとドメインの2つの軸、即ちタスクドメインの特徴的な問題としてとらえることにする。

従って、本節では、各問題領域に対し、ドメインの特徴及び基本的なタスクの特徴を整理して述べ、それぞれの問題について、本報告書で設定した評価項目の必要性の度合及び特徴的な評価項目について述べる。

なお本節で扱う「問題領域」とは、設計、計画など一般的に用いられる用語を使用しているが、その定義は小林らの定義に基づくもので、業務分野に共通する基本的な問題解決タスクの分類に従っている[AIセンター 89]。

以下に各問題について、システムの特性、即ち定義、問題の特徴、基本タスク、問題解決機能（実現手法）を表2.6-1にして取りまとめた。

【参考文献】

[AIセンター 89]知的情報処理システムに関する調査研究報告書、(財)日本情報処理開発協会、ICOT-JIPDEC AIセンター、1989.

表 2.6-1 システムの特性一覧表

	設計型システム
定義	らを決 え要素を 与要様 が成仕 様構部 仕、内 求にの 要要素 のた要 力る成 出す構 と現各 力実つ 入をか のれ、 ムこせと。 テきわこ スト合る シたみす れ組定
問題の特徴	大性 が的要 間本必 空基の 解が化 の成適 造合最 構るの ムよ素 テに要 析成証 シ解構 1. 2. 3. 4. 5. 6. 7. 検査対 設計設
基本タスク	表現 の現 連表 関的成 の層生 そ階動 の自テ との自 素題の 要問案 設計替 構設代 1. 2. 3. 4. 5. 6. 7. 部上設 並列 計列 的問 シレ 案の 問題
問題解決機能	戦略 化戦 密化 精略 ン戦 ウ化 ダ小 プ最 ツ束 ト拘 1. 2. 3. 4. 5. 6. 7. 最推 檢証 仮説 類推 協調 型推

	計画型システム
定義	可 用 デ ータ 利 用 に 、 列 系 に め 時 を た る す ど 現 な 実 物 と を こ 標 金 、 目 的 、 人 化 た れ （ 組 源 え 資 与 な し 能 と
問題の特徴	1. 問題の再作成 2. 相互利用の促進 3. 相互利用の促進 4. 相互利用の促進 5. 相互利用の促進 6. 相互利用の促進 7. 相互利用の促進
基本タスク	1. 相互利用の促進 2. 相互利用の促進 3. 相互利用の促進 4. 相互利用の促進 5. 相互利用の促進 6. 相互利用の促進 7. 相互利用の促進
問題解決機能	1. 相互利用の促進 2. 相互利用の促進 3. 相互利用の促進 4. 相互利用の促進 5. 相互利用の促進 6. 相互利用の促進 7. 相互利用の促進

	解釈型システム
定義	たし れ定 さ推 測を と。 観態 この て状 るの え 通ム と をテ を サス け ンシ 付 セ、 味 はて 、意 いし な 析的 る あ 分理 器を 物 測タ に 計一 れ デこ
問題の特徴	性 関 相 次入 高混 のの タ音 一難 在 一 デな 存 間則 のな 空規 り要 不誤 必信 対感 時不 誤必 信対 感 1. 2. 3. 4. 5. 6. 7.
基本タスク	合 照 な定 定 全同 推理 完の の処 理 不造 態の 処性 出の 構状 さの 様 抽と ムム い性 多 のル テテ ま全 の 徴デ スス い完 積 特モ シシ あ不 解 1. 2. 3. 4. 5. 6. 7.
問題解決機能	分け ス ラ 能 ク 能 る能 価 査 機 法 よ機 評 査 論 に索 の成 推 論 層檢 造生 性推 論推 階ル 構的 実推 型 類デ 分層 確説 調 分モ 部階 不仮 協 1. 2. 3. 4. 5. 6. 7.

	診断型システム	
定義	常因 異一原にデ、ムたて、テれしスを用シ測利る観を。 す、識るとき知あ対象とで対したとは生にとがムす問題定問故ス同断はシを診たび所ま及箇	
問題の特徴	利用の 実性 コスト 性と要御 知識確用間必論 知利不利時の推 のののの化の ル識識タめ象断 ベ知知一た抽診 レ的的デの的 計驗業測識話 設経経操計知対 1. 2. 3. 4. 5. 6. 7.	
基本タスク	表現 表的包 的内 層層を定 階層決る 類のス決定よ 分造タ択同に の構の選のル 象ム問題の因デ 現テ問点原モ 常ス釈測常い 異シ解計異浅深 1. 2. 3. 4. 5. 6. 7.	
問題解決機能	論調 推推論 型の推論完全 動動態推論と効 駆驅実推型性ノ 象標確説調率用 事目不仮協効費 1. 2. 3. 4. 5. 6. 7.	性の調和 全分析

	制御型システム	
定義	て態 し状。 視のあ 監ムで をテと 況スこ 状シる のにえ ムり加 テ通を スタ作 シれ操 は、さ に定う と予よ 題める 問予す 御移 制て遷 いが	
問題の特徴	性 存 依性 歴所 履局 るる よよ にに れ性の 遅形化 間線定 時非安 制実対 1. 2. 3. 4. 5. 6. 7.	要求 点 接性の 頼援の 支計 と信 実と 精セ 口間的 話設
基本タスク	包測行化 現内予実ネス を表をの的エン の性ス状態先省ダ理 造特たるのでガの 構動のよ作化タさ ムム題に操業一い テテ間ル化操レま スス断デ安定べい シシ診モ安安オあ 1. 2. 3. 4. 5. 6. 7. 8.	
問題解決機能	測 予 ス 選 る測価ン析回 よ予評ダ解の 能に的イのク 機タよ目ガクッ 断一に多ソッコ 診レ論のヨロー のユ推則シドタ 態ミ性御クツン 状シ定制アディ 1. 2. 3. 4. 5. 6. 7.	

2.6.1 設計型システム

(1) ドメインの特徴

① 設計型システムの定義

設計問題とは、設計対象のシステムの入力と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み合わせ、かつ各構成要素の内部仕様を決定することと定義する[AIシンク 89]。

② 設計型システムの実現手法

次のような専門家の問題解決過程の側面に着目していることが多い。

a. 探索的側面

所望の構成・特性を得るため、結果の候補案を何らかの方法で生成して、解析・検査し、制約条件に合致していれば結果として採用する。ES化ではこの候補案の追求を探索問題に帰着させている。

b. 意思決定的側面

候補案を生成しても、その時点ではそれが結果として適切であるかどうかを判定することが不明であれば（さらに詳細化すれば判定できる、従属関係にある他の部分の設計がある程度すすめば判定できるなど）、候補案を仮に選択しておく必要がある。ES化では、この候補案の選択を意思決定の問題に帰着させている。

c. 再利用的側面

専門家の問題解決過程では過去に経験した成功および失敗の事例は有効に利用されており、新しい設計問題が与えられた場合にも、これと類似の過去の事例を検索し、これを参考にしながら問題解決の過程の効率を向上させていると考えられる。ES化ではこれを過去の事例の再利用の問題に帰着させている。

d. 知識獲得的側面

類似の設計事例を利用できない全く新しい型の問題においては、専門家は問題解決に有用な知識を自ら獲得していくことが必要になる。ES化では、この知識の獲得の過程を多数の正例（成功例）や負例（失敗例）の背後に成立する法則・原理・規則を帰納的に導く、あるいは既に獲得済の知識を利用して新しい事例を説明しかつこれの一般化を演繹で導く、といった知識獲得の問題に帰着させている。

(2) タスクの特徴

設計型システムに分類される専門家の問題解決過程を情報処理技術の観点から眺めると、以下に示すような特徴を有している。

a. 広い解空間

設計対象のシステム構造を決定することは、通常は組み合わせ問題となり、広い探索空間をいかに効率的に探索するか、探索空間をどう表現しておくかが課題となる。

b. 多重の最適化

設計対象が複雑な構成をとる場合には、一般的には問題を階層的に分割しておく手法がとられる。このためある階層の構成要素を決定すること（最適化）は、さらに下位の階層の構成を設計する必要があるため、多重の最適化問題となり、これをいかにうまく階層分けし全体として最適化していくかが課題となる。

c. 検証の完全性

設計を階層的に分割を行う場合、上位階層と下位階層の無駄な探索の繰り返しを避けるため、上位階層の構成条件が下位階層で必要とする条件を必ず満足するように構成しておく必要があるため、各階層内での検証の完全さが課題となる。

以下に、設計型システムの定義、問題の特徴、基本タスク、必要な問題解決機能を示す。

- ① 定義
システムの入力と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、構成要素を組み合わせて、各構成要素の内部仕様を決定すること。
- ② 問題の特徴
1. システムの構造が大
2. 解析による組合せの空間が基本的
3. 構成要素の最適化の必要性
4. 検証の完全性
5. 検証の効率性
6. 対話形式での設計支援
7. 設計段階に応じた支援形態
- ③ 基本タスク
1. 構成要素との関係の表現
2. 設計問題の自動生成の評価
3. 代替案の自動生成の繰り返し
4. 部分シミュレーションの利
5. 上位設計事例の活用
6. 並列設計
7. 並列設計

- ④ 問題解決機能
1. トップダウン精密化戦略
 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

(3) 設計型システムのV & V

設計型システムには、以上述べた特徴があるため、次のような項目を重視した評価が不可欠である。

① 知識ベースの妥当性評価

a. 必要性

専門家だけでなく通常の一般の担当者が有しているような常識的な背景知識を、ESに持たせることが不可能なため（領域知識として意識していない、量が多い、など）、専門家が設計する場合に比べて、無意味な結果の候補を生成する場合もある。また、設計対象のモデルをES内部でどのように表現しておくかは、妥当性の高い結果が得られるかどうかに関わってくる。

b. 特徴的な評価項目

知識表現の選択と適切さ、知識獲得方法の選択と適切さ、知識ベースの正当性、知識ベースの完備性、知識ベースの一貫性

② 問題解決・推論方式の妥当性評価

a. 必要性

設計問題では、探索の空間が広いため、専門家の持つ問題解決過程を正確に推論手法に反映させないと要求仕様で規定された時間内では結果が得られないことが多い。このため、問題解決・推論方式の妥当性評価は最重要である。

b. 特徴的な評価項目

推論処理の妥当性、推論時間の妥当性、推論手法の選択と適切さ、学習機能の適切さ

③ 結果の妥当性評価

a. 必要性

業務の中心課題が最適化であるため、結果の最適性が重要となる。ただし、最適性を追求するあまり、専門家でも実施していない推論機構や評価

項目を取り込むと、ある条件のもとでは、よりよい結果が得られるものの、専門家では簡単に許容できるような条件の変化にまったく対処できなくなる恐れがあるため、注意が必要である。

b. 特徴的な評価項目

解の正解率、解の最適性

④ 知識ベースの保守性評価

a. 必要性

設計分野では、専門家は業務の実施に伴い常により効率のよい設計手法を模索している。このため、専門家が自ら知識ベースを洗練化できるようにESを前もって作成する使用・運用形態が理想的である。

b. 特徴的な評価項目

知識ベースの理解性、知識ベースの変更箇所発見の容易性、保守用ツールの整備状況

⑤ マンマシンインタフェースの妥当性評価

a. 必要性

設計業務では、計算機で実現可能な部分は限定され、パターン認識などの人間が得意な機能については当面人間に作業させるのが妥当な解決策である。このため、人間と計算機との機能分担（プロトコル）が複雑となるとともに、操作者に理解しやすい図形・図面などを多用することが多いため、マンマシンインタフェースは重要である。

b. 特徴的な評価項目

利用時のインタフェース、保守時のインタフェース

⑥ システムの総合評価

a. 必要性

設計型システムでは、知識ベースと利用者インタフェースのみでESを構成することは非常にまれで、従来のソフトウェア開発手法に基づいて開発された計算機シミュレーション、データベース、ネットワーク、他のハードウェアなどと組み合わせることが多く、また、設計完了後の人手介在による誤りを避けるために、設計段階から製造段階までを広範囲にシステム化することが多いため、統合性を重視する必要がある。

b. 特徴的な評価項目

システムの統合性、ESの成長可能性、ツール・言語選択の適切さ

2.6.2 計画型システム

(1) 計画型システムの特性

[AIセンタ 89]では計画型システムの特性を次のようにまとめている。

- ① 定義
与えられた目標を達成するために、利用可能な資源(人・金・物など)を
時系列データと組織化すること。
- ② 問題
1. 広多不部既費対
2. 特徴的な実計画と型
3. 探索的な計画の便計
4. 空価環境の相の作成
5. 間属性の相互支援
6. 測相互作用のドオ要求
7. 作用の要求
- ③ 基本
1. タ計画組合約環境位画
2. ス画組合約環境位画
3. ク過せ条のレ事評
4. 程的件予べ例価
5. の探の測ルのの
6. 階素相 へ 検多
7. 層 化 作用 知的 後戻り
利用
- ④ 問題
1. 解決ト拘網仮類計リ
2. 決ッ束羅説推画ス
3. 機ブ最的推にのク
4. 能ダ小探論よ多下
5. ウ化素にる目で
6. シン戦機能の画評意
7. 精密化 戦略 知的 探索 利用
事例 決定

(2) 計画型システムの特質

1) 問題クラスの同定

[AIセンタ 89]では合成型問題について、対象のシステムを3クラスに分類している。クラス1はシステムが与えられた時、サブシステムが与えられた時、それを構成要素としたシステムの構造を求める問題である。クラス2はシステムの特性とサブシステムの特性が与えられた時、それを構成要素とした最適なシステムの構造を求める問題である。クラス3はシステムの特性とシステムの構造が与えられた時、最適なサブシステムの特性を求める問題である。このなかで、計画型問題は一般にクラス2の問題として同定する。

2) 問題解決戦略

計画型問題に対する基本的な問題解決アプローチでは生成検査法を採用している。生成検査法とはシステム構造や構成要素特性を仮に生成し、システム特

性の検査を行なうことを繰り返して、より良いシステムを探索していく方法をいう。

3) 組合せ問題

計画型システムはシステム特性が与えられたとき、これを実現するシステム構造および構成要素特性は一般に無限に存在する。すなわち、組合せ爆発の問題があり、この問題をどう克服するかが本質的な課題となる。また、構成要素特性は多種、多用であり、「鶏と卵」的な順序付問題も内在している。

4) 最適化問題

計画型エキスパートシステムでは、数式モデルとして定式化しにくいものを扱うため、一般に最適解は未知である。したがって、拘束条件を満足する実行可能解をいくつか導き、それらを比較することにより準最適解を求めることになる。ここで、最適解の評価には、トレードオフ、つまりあちらが立てばこちらが立たずといった多目標問題も内在する。

(3) 計画型システムのV & V

① 知識ベースの妥当性評価

a. 必要性

計画者のもつ経験的知識には、計画者自身も説明出来ないものもある。これは、過去において計画者が実際に行なった計画立案を分析することにより、知識を抽出することになる。この場合、計画者の試行錯誤の過程を正しく抽出することが必要である。

b. 特徴的な評価項目

計画型エキスパートシステムで用いられる知識のタイプは、おおよそ、順序付(ordering)、割当(assignment)、分割(division)、結合(combination)に分類することができる。このことを踏まえて、知識ベースの正当性、完備性、一貫性の評価を行なうことになる。

② 問題解決・推論方式の妥当性評価

a. 必要性

計画型問題の問題解決プロセスには、探索的側面、意思決定的側面、再利用的側面、知識獲得側面があり、これらを考慮して問題解決戦略を検討することになる。また、推論方式は生成検査法を基本とするが、この場合、状態の定義や状態遷移をうまく設定しなければならない。

b. 特徴的な評価項目

解空間が非常に広く、計画パラメータの選択に自由度がある場合、探索時に効果的な枝刈りが出来るかが、推論処理や推論時間の妥当性評価に密接に絡む。

③ 結果の妥当性評価

a. 必要性

数式モデルとして定式化しにくいものを扱うため、最適解は未知である。したがって、準最適解を求めることを狙うわけであるが、この時に、組合せの爆発を回避し、準最適解の候補になると予想される可能解を網羅する必要がある。

b. 特徴的な評価項目

いくつかの可能解から準最適解を選択する場合に、それらの可能解が比較可能でなければならない。つまり、評価関数等を正しく設定し、解の評価を適切に行なうことが上げられる。

④ 要求仕様に対する達成度評価

a. 必要性

計画型問題は、通常 ill-structured であり、システム化が容易ではない。すなわち、①問題の構成要素が多種・多様である、②要素内の関連が複雑である、③制約条件が動的に変化し複雑である、等により、モデル化することが困難であるということである。したがって、プロトタイピングによるボトムアップのシステム構築手法を用いることによって、ill-structured な部分を徐々に見通し良くしていき、問題解決を行なうことになる。

a. 特徴的な評価項目

計画型システムの最終的な達成目標は、従来のシステムよりも迅速で、よりよい計画が立案出来るようになることである。システムが自動的に最終計画を立案する必要はなく、推論の途中で専門家やユーザが介入しても構わない。すなわち、専門家やユーザの支援をいかに的確に行なえることが重要である。

⑤ 知識ベースの保守性評価

a. 必要性

知識ベースの保守による品質の安定性を保証するために、保守用のツールを整備することが上げられる。これは例えば、テストケースを自動生成する

機能を用意しておき、保守をおこなった場合に、その知識の正当性を検証すること等である。

b. 特徴的な評価項目

計画型システムは、外的環境の変化や計画方針の変更により、使われる知識が異なったり陳腐化したりするので、必然的に知識ベースの保守も多くなる。したがって、知識ベースの保守性の評価は不可決な項目である。

⑥ マンマシンインタフェースの妥当性評価

a. 必要性

ユーザインタフェースの評価として、操作性、ガイダンス、エラー処理等が考えられる。操作性は、入力法をメニュー/マウス等を用いて誤入力を削減することや表示法をグラフィックスを用いてガントチャートのようにビジュアル化すること等である。ガイダンスは良い計画を立案を支援するために制約を満たさない場合の関連する項目の表示や組合せの爆発を起こす場合の回避法を提示してくれることである。専門家インタフェースの評価は、知識ベースの理解性が上げられる。これは知識ベースの各々の知識が容易に理解出来、知識の変更を行った場合に知識間の関連がわかりやすく、さらにその変更による複雑な属性間への影響の伝播が把握出来ることである。

b. 特徴的な評価項目

システムの開発においてユーザや専門家の協力を得るためには、プロトタイプ of 段階でもインタフェースを充実させておくことが重要である。

⑦ システムの総合評価

a. 必要性

ビジネスエキスパートシステムであることを考えると、従来システムと同様、開発されたシステムの有意義を定量的に示さなければならない。利用者の満足度の評価は、従来システムよりも迅速で、よりよい計画が立案出来るようになることである。これは生産性の向上を示しており、計画者の人数の削減、あるいは計画者の代替が可能になることで把握できる。システムのコストパフォーマンスの評価は、まず従来システムで解決出来なかった領域の問題が解決出来るようになったということと、さらにその解決できた度合いが投資に見合ったものであるかということを示される。

b. 特徴的な評価項目

統計処理やOR（オペレーションズリサーチ）等、従来型システムの枠組

みで解決出来る問題をエキスパートシステム化してもあまり意味がない。したがって、テーマ選定の妥当性について、特に検討しなければならない。

2.6.3 解釈型システム

(1) 解釈型システムの特性

解釈型システムの特性は以下のようにまとめることができる [Alten 89]。

- ① 定義
計測器あるいはセンサを通して観測されたデータを分析して、システムの状態を推定し、これに物理的な意味付けを与えること。
- ② 問題の特徴
1. 時空間データの高次相関性
2. 不規則な雑音の特
3. 誤りのある存在
4. 必要のないデータ
5. 信号処理の欠落
6. 対象に
7. 感覚データに関する技術の連続の利用
- ③ 基本タスク
1. 特徴抽出
2. モデルの構築
3. システムの構造
4. システムの不完全性
5. 不完全性の多様性
6. 不完全性の多様性
7. 不完全性の多様性
1. 特徴抽出
2. モデルの構築
3. システムの構造
4. システムの不完全性
5. 不完全性の多様性
6. 不完全性の多様性
7. 不完全性の多様性
- ④ 問題分類
1. 分類
2. 階層構造
3. 階層構造
4. 階層構造
5. 階層構造
6. 階層構造
7. 階層構造

(2) 解釈型システムと他のシステムの関連

解釈型のシステムは診断型、制御型と並んで広い意味で解析型と呼ばれることが多いが、形態としては単独で存在する場合よりは他のシステムと協調して機能することが多い。実際、ほとんどの場合解釈型のシステムは診断型、制御型のシステムの前処理的なタスクを受け持つと考えられる。診断型システムでは異常の有無とその対策を推論するために診断対象からのデータを基に対象の現在の状態を推定しなければならない。また制御型システムにおいても制御対象の状態を推定することは重要な要素である。換言すると、診断型、制御型のシステムはその中に解釈型システムのタスクを包含することが一般的と言える。

したがって解釈型システムの特質を考えるためには他の解析型システムの特質を併せ考えなければならない。

(3) 解釈型システムの特質

(1)、(2)より、V&Vを考えるに当たって考慮すべき解釈型システムの特質は、以下の様にまとめることができる。

a. 実時間性

前述のように解釈型システムは他のシステムの前処理的なタスクを受け持つことが多く、しかも続く処理は一般的にタイムクリティカルな処理であることが多い。したがって処理の実時間性は最大の特徴と言える。

b. モデルとの親和性

解釈型システムは一般的には有限の解空間の中から、あらかじめ与えられたモデルとの比較によって対象システムの状態を推論することが多い。解空間が非常に広いかまたは無限であるような設計問題とはこの点が異なる。したがってあらかじめ与えられるモデルと現実のシステムの親和性が解釈型システムを正確かつスピーディに実行するための必要条件となる。

c. 組み込み型システム

診断や制御を含めた形での組み込み型システムは技術的にも難易度が高く、信頼性の面からも現状では現実的とは言えないが、解釈型システムとしての側面で見ると組み込み型システムの要請は高く、また将来的には診断や制御までも含んだ形での組み込み型システムが中心的な課題になると思われる。

d. 高信頼性

c.の組み込み型システムとしての特徴とも関連するが、解釈タスクの部分で誤りをおかすと後続のシステムの処理がすべて意味をなさなくなってしまう。致命度という観点で考えると解釈型システムの誤りが他のシステムに対して決定的な重要性を持っている。したがって成熟性、誤許容性、回復性などのファクタも重要である。

e. 大量の時系列データ

センサなどを通して得られるデータは時系列的で大量であることが一般的である。こうしたデータを素早く処理して対象システムの状態を推論することが必要である。

(4) 解釈型システムのV & V

以上述べた解釈型システムの特性・特質を基に、V&Vにおける考慮点を検討する。

① 知識ベースの妥当性評価

a. 必要性

解釈型システムが備えるべき高信頼性を保証するものとして知識ベースの妥当性評価は重要である。

b. 特徴的な評価項目

解釈型システムでは経験則的な知識の扱いもさることながら、信号処理技術を用いたり、場合によっては感覚データなどを扱うことも考えられる。したがって知識ベースの妥当性を評価する際にはシステムが扱っているモデルの正当性や精度に重点を置いた検討がなされるべきであろう。

② 問題解決・推論方式の妥当性評価

a. 必要性

解釈型システムの実時間性や大量の時系列データの扱いに大きな影響を及ぼすのが問題解決・推論方式である。

b. 特徴的な評価項目

雑音やあいまいなデータ、誤りを含む信号などの入力データを解釈し、モデルとの不完全な照合を素早くしかも正確に実行する必要がある。処理の即時性を実現するためには学習機能を備えていることが望ましい。同様のデータに対して常に同じ推論の連鎖をたどるのではなく、例えばチャンキングなどによって処理の高速化がはかれる。また処理の正確性のためには部分構造の評価機能や階層的生成検査法、仮説推論などの手法によって不完全なデータを適切に処理しなければならない。いずれにせよ学習機能や不完全なデータの処理はシステムの高信頼性とはある意味では相入れない概念であり、この辺が解釈型システムのV&Vのポイントと言えよう。

③ 結果の妥当性評価

a. 必要性

高信頼性、特に解釈型システムの致命度（他のシステムに対して前処理である解釈型システムの誤りは致命的である）を考えると結果の妥当性は是非保証されなければならない。

b. 特徴的な評価項目

解の正解率が高いことも当然望ましい点であるが、解釈型システムの致命度を考慮すると、最適性が満たされなくとも解の網羅性は満足していなければならない。

④ 要求仕様に対する達成度評価

a. 必要性

解釈型システム、あるいはそれを含んだ診断型システムや制御型システムの要求仕様が完全な形で与えられることは稀で、プロトタイプシステムの完成によってそれまで潜在化していた診断対象や制御すべきケースが明らかになり、考慮しなければならないケースが増大することが一般的である。

b. 特徴的な評価項目

したがって固定された仕様に対する達成度を考えるよりは、むしろシステムの柔軟さ、変更に対するロバストネスを考慮しておくべきである。ただし解釈型システムの他の特徴として組み込み型であることが挙げられるが、これは柔軟さや変更に対する強さとは必ずしも両立しない概念であることもまた考えなければならない。

⑤ 知識ベースの保守性評価

a. 必要性

④で述べたシステムの柔軟さを実現するためには、知識ベースの保守性が良くなければならない。

b. 特徴的な評価項目

知識ベース中にはいわゆる経験則の他に信号処理的な知識や計算アルゴリズムなどが含まれていることがあるため、こうした知識の保守性がよいことも重要である。

⑥ マンマシンインタフェースの妥当性評価

a. 必要性

解釈型システムは組み込み型であることが特徴であることから想像できるように、概して人間に対するインタフェースは処理の実時間性以外はそれほど必要としない。

b. 特徴的な評価項目

ただし、異常検知システムのように解釈型システムのみで成立しているような場合は処理結果を人間に表示し、その判断材料としなければならないた

め、結果の要約的な表示やグラフィックスの併用などによる理解し易い結果の表現法を考えなければならない。

⑦ システムの総合評価

a.&b. 必要性／特徴的な評価項目

解釈型システムの特質は前述の通り、実時間性、モデルとの親和性、組み込み型システム、高信頼性である。個々のシステムはそれぞれ場合によって目標の重点が異なると考えられる。こうした状況を考慮して、実時間性と高信頼性、組み込み型と柔軟性などの相反する特性をコントロールして、場合に応じて最適な特性を備えたシステム構築を行うことが総合システムとしての完成度につながる。

2.6.4 診断型システム

診断型問題とは対象とするシステムに異常または故障が生じた場合に、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し、正常な状態への復旧をはかることと定義される[AIエンタ 89]。

(1) 診断型問題の特徴

① 支援的な利用形態

診断型システムの目的は、正常状態への復旧をはかることであり、多岐にわたる診断知識の蓄積と、それらの適切な利用手段とが要請される。現状では、実際の行動は専門家あるいはオペレータなどによって行われ、エキスパートシステムはそれを支援する役割を担っている場合が多い。このため、説明機能や動作の透明性など、対話性の確保が重要である。

② 設計レベルの知識の利用

プラントや機械設備などのエンジニアリングシステムでは、対象の構造に関する「深い知識」を利用することができる。深い知識を導入することにより、診断の完全性、未知の状況への対応、適用範囲の判断、などの機能の実現が期待できる。ただし、深い知識を利用するために定性推論などの手法を用いる場合には、推論方法の妥当性に注意が必要となる。

③ 経験的知識の利用

専門家は、故障原因に特有の兆候や故障の頻度など診断についての経験的知識を、診断作業の効率化などに利用している。特に、対象システムのふるまいが解析できない場合には表層的な因果関係の知識に依存せざるを得ない。このように診断型システムにとって経験的知識は重要な位置づけにある。しかし、経験的知識について正当性、完備性、一貫性といった妥当性を評価することは経験的であるがゆえに困難な場合が多い。知識ベースの妥当性評価についての対処を、開発の早い時期に定めておく必要がある。

④ 計測コストへの配慮

診断の基本的な流れは、対象システムのふるまいを計測し、状態を推定し、不具合の原因を同定することである。計測のためのコストと診断精度との間のトレードオフについては、要求仕様の段階で考慮すべきである。

⑤ 例外的な状況の取扱い

診断問題では、対象システムにとって例外的な状況を扱うため、全ての場合を網羅することは不可能である。そこで、診断システムの目的に応じて対象システムを適切な抽象化レベルでとらえる必要がある。また、システムの妥当性を検査する際のテストケースの選択には十分な注意が必要である。

(2) タスクの特徴

① 異常事象の分類階層的表現

対象システムの状態を推定するために異常事象を分類階層木として表現するタスクである。適切に階層化することにより診断の見通しが良くなり、また、診断システムの保守性も向上する。なお、異常事象の表現と質問制御戦略とは関連が深いが、区別して考えることが必要である。

② システム構造の階層的表現

対象システムの構造に関する知識を、物理的あるいは機能的な観点から階層的に表現するタスクである。この表現は、診断推論に直接利用される場合と、診断知識の抽出に利用される場合とがある。

③ 解釈問題のタスクを内包

測定データを診断推論が可能な形式にする解釈のタスクである。データの特徴抽出、モデルとの照合、システム状態の推定、あいまいさの処理、不完全性の処理、最適な解釈の選択、などのタスクが必要となる。

④ 計測点の選択的決定

故障原因を追求していく過程で効率的な計測点を順次選択的に決定するタスクである。対象システムの構造に関する知識を利用し、計測から得られる情報と計測に要するコストとのバランスを考慮して決定する。

⑤ 異常原因の同定

システムの構造と構成要素の特性とをふまえて、測定データから異常原因を同定するタスクである。診断推論の非単調性や、診断知識の不確実性などに留意する必要がある。

⑥ 浅いモデルによる効率性

経験的な知識を利用して異常原因の近傍へすばやくアクセスするタスクである。対象システムが大規模な場合や、応答速度が重視される場合に重要となる。また、診断システムの導入をスムーズに行う上でも重要な要素である。

⑦ 深いモデルによる完全性

対象モデルの設計レベルの知識を用いて、ある抽象化レベルでの完全な診断を行うタスクである。測定データを最も合理的に説明する仮説を生成する考え方などが提案されている。

(3) 診断型システムのV & V

以上に述べた特徴から、診断型システムのV & Vでは次のような点がポイントとなる。

① 知識ベースの妥当性評価

a. 必要性

診断型システムには、異常な状態から正常な状態への復旧を支援するために、多岐にわたる診断の知識を蓄積していることが要請される。このため知識ベースの妥当性評価が特に重要である。知識ベースの正当性はもちろん、経験的な知識の利用や、知識獲得への系統的な取り組みなどにも注意が必要である。

b. 特徴的な評価項目

知識ベースの正当性評価、経験的知識を正当化する方法とその確認、
知識獲得方法の選択と適切さの確認

② 問題解決・推論方式の妥当性評価

a. 必要性

実用的な診断型システムでは知識ベースが大規模になり、それにつれて問題解決や推論方式も複雑になるため、妥当性評価が必要となる。経験的な診断知識には不確実さが伴うため、推論機構が不確実性を適切に扱っているかどうかを評価する必要がある。また、設計レベルの知識を用いる場合には、定性推論や仮説推論などの手法の選択についても評価する必要がある。

b. 特徴的な評価項目

推論処理の妥当性評価、推論手法の選択と適切さの確認、不確実情報に対する処理技法の適切さの確認

③ 結果の妥当性評価

a. 必要性

診断型システムの実用性は、診断結果の妥当性に大きく依存する。診断型システムでは、複数の原因が考えられる場合が多いが、それをいかに絞り込めるかがポイントとなる。また、知識ベースの洗練へフィードバックする意味から

も結果の妥当性評価は重要である。評価に当たっては、実際の利用状況を反映したテストケースを選択しなければならない。

b. 特徴的な評価項目

解の正解率評価、解の評価方法の適切さの確認

④ 要求仕様に対する達成度評価

a. 必要性

診断型システムの多くは、単に原因を追求することではなく、専門家あるいはオペレータなどを支援して正常な状態へ早く復旧できるようにすることを目的としている。この目的に沿ってユーザーニーズを要求仕様としてまとめ、適宜達成度を評価する必要がある。

b. 特徴的な評価項目

要求仕様内容に対する評価、要求仕様に対する達成度評価

⑤ 知識ベースの保守性評価

a. 必要性

エンジニアリングシステムの診断では、対象システムの仕様が多様化しており、また仕様の変更も頻繁に起きるため、知識ベースの保守性が課題となっている。開発の初期から知識ベースの規模と運用方法とを想定し、保守性の評価を行う必要がある。

b. 特徴的な評価項目

変更箇所発見の容易性の評価、知識ベースの理解性評価、保守用ツールの整備状況の確認

⑥ マンマシンインタフェースの妥当性評価

a. 必要性

診断型システムの支援的性格から、利用者インタフェースは特に重要視される。例えば故障診断では、点検修理方法の動画表示などが効果的に用いられている。利用者のレベルが高い場合には推論の制御を利用者主導で行えるようにすることも対話性の要件の1つと考えられる。

b. 特徴的な評価項目

利用時のインタフェース評価、保守時のインタフェース評価

⑦ システムの総合評価

a. 必要性

診断型システムは、測定システムや診断データベースなどとの統合をはかることにより実用性の高いものとなる。開発においては診断型に特化したツールがいくつか市販されており、適切なものを利用できれば開発効率の向上がはかれる。また、導入の効果を客観的に示す指標が定めにくいため開発の初期に充分詰めておかなければならない。これらの観点を含めた総合評価が必要である。

b. 特徴的な評価項目

利用者の満足度評価、統合性の評価、成長可能性の評価、ツール・言語選択の適切さの確認、システムのコストパフォーマンス評価

2.6.5 制御型システム

(1) 制御型システム概要

ここで言う制御型システムとは、制御の対象となるシステムに対して外部から与えられた運用計画および方針を基に、

- ・ 時間・環境・システム状態の変化により直接操作または操作値を出力し（定常時制御）、
- ・ 計画されたシステム状態からのずれが生じた場合にはそれを回復する操作または操作値を出力する（異常時制御）

エキスパートシステム（ES）である。

このタイプのESの適用分野としては、

- ・ 船舶・航空機・人工衛星等の運行・操縦制御
- ・ 溶鉱炉・セメントキルン等のプラント制御
- ・ 製造工程制御
- ・ コンピュータハードウェア／ソフトウェアの実行制御

その他多くのものが考えられる。現在でも何らかの自動／半自動制御がES以外の手段で行なわれているものが多い。例として挙げたESの適用分野を見ても明白のように制御システムおよびその中枢としてのESが正しく動作することが絶対の条件になっている。ESの判断の誤り、誤動作は場合によっては人命にかかわったり社会的に大きな損失をもたらしたりする。従って、ESの検証および妥当性検査（V&V）は必須条件である。制御型システムは入出力データが多く、しかも実時間性の複雑なシステムがほとんどであり、実際にはV&Vの実施には多くの難しい問題を含んでいる。

(2) 制御型システムの特徴

制御型システムはESの基本タスクとしての分類ではなく、適用業務としての一分野としてとらえられるべきである。ここでは他の適用業務分野と比較して、特に制御型システムの問題領域に顕著な特徴をあげる。

① 対象システムの複雑さ

制御の対象となるシステムは個々にはかなりの違いはあるものの、問題のドメインは明確で比較的小さいものが多い。目標値あるいは目的関数の種類は限られており、その評価値も明確である。しかしながら対象システムは数多くの

サブシステム・部分により構成され、お互いに干渉しながら全体として動作しており、それを制御するESの内部モデルは複雑になる。対象システムの状況を表わす入力データの数も多く、場合によっては数千から数万のデータより状況を判断しなければならない。更に状況の判断には現在のデータ値だけでなく傾向値や周期的変動等の履歴データや事象・状況の発生順序による文脈判断が必要な場合も多い。制御値・操作の決定においてもサブシステム・部分の状況を判断しそれから全体としての状況判断に至る総合化機能の実現、判断より個々の操作値・操作および操作順序を決定する分解・組立て機能の実現が必要である。すなわち制御型システムはそのサブシステムとして解釈型／診断型／設計型／計画型のシステムを含む複合型システムである。

②曖昧さおよび非単調性

対象システムは時間、環境の変化、イベントの発生等により急速に変化する。ES内部に保持しているデータやモデルはそれにつれて急速にデータとしての正確さを失ってゆく。新しく入力されたデータはモデル内部のデータと相いれない場合にはその内部データに依存して生成された知識は更新されなければならない。ES内における制御知識はほとんどがある特定の状況でのみ成立するものである。また入力データそのものも必ずしも正確ではなく、時間遅れ、計測誤差等による曖昧さを含む場合が多い。そのような場合でも安定した制御を限られた資源、時間内で行う必要がある。

③実時間性

対象システムの性格によりかなり差異はあるが、制御型システムは通常実時間性が厳しく要求される。システム状況はダイナミックに急激に変化し、素早い操作を行わないと対象システムが動作不能になったり制御不能になったりする場合もありうる。継続して起こる多量の変化する入力データに対して正確な制御値・操作を許容される時間内で決定しなければならない。応答時間は数ミリ秒の厳しいケースから数分の比較的軽いケースとかなり差異はある。また応答を返さなくても対象システムが通常安定稼働するものと、限定された時間内に応答がないと対象システムが停止する場合ではシステム要件はかなり異なる。しかし他の問題領域システムよりは非常に強い実時間性が要求される。

④ 高信頼性と精度の向上

先に述べたように制御型システムの誤動作は大きな損失をもたらす。出力としての制御値・操作の決定には非常に高い精度が要求される。さらに制御型システムでの精度には実時間システムとしての要件がある。精度の高い出力を行っても、それが要求された時間内の応答でない場合には役に立たないばかりでなく、時には障害を引き起こす可能性もある。すなわち制御型システムにおける出力の精度は時間と質で決定される。質があまり高くななくても一定時間に応答できるシステムは、質がより高いが一定時間に応答できないシステムより勝る場合が多い。

制御型システムの場合にはESの精度を向上する仕組みを比較的作成し易い場合が多い。トランザクション処理としてのESの性格により多数の推論ケースを得ることができる。ESの妥当性検査においては推論過程の記録と実制御システム（専門家そのものの場合も有りうる）との比較、または確立した評価方法によりテストは比較的实施しやすい。ファジィ制御、ニューラルネット、事例推論等を使用している場合には過去の推論の統計値の使用等により自動的に精度を向上することも可能である。

（３）制御型システムの基本タスク

前項の特徴および問題点に対処するための制御型システムの基本タスクを次に説明する。

① 複雑さに対するESの基本タスク

・ 入力データ解釈および内部モデル更新

フレーム、セマンテックネット、オブジェクト指向等の知識表現方法を用いて対象システムの状態をES内部に表現する。

・ 履歴・文脈判断

時系列推論、状態遷移ネットワーク等の手法を用いて対象システムの状況を把握する。

・ 複数知識ベースの構成と実行

サブシステム・部分の状況を判断する複数知識ベースと総合判断知識ベースの同時実行を可能にする仕組みと相互コミュニケーション機能を提供する。黑板システム等がその例である。

・ 個々の制御値・操作の生成

制約、システム動作、過去の事例等を用いて制御値・操作を決定する。

② 曖昧さおよび非単調性に対する E S の基本タスク

・ 曖昧さを取り扱う知識表現

ファジィ、ベイズ等の手法を用いて対象システムの広い動作範囲において安定した制御がコンパクトな知識ベースで実現できる。

・ 非単調推論

E S 内知識のダイナミックな更新、A T M S 等の手法を用いて常に対象システムの正確なモデルを保持できる。

③ 実時間性に対する E S の基本タスク

・ タイマ機能

応答時間を保証するためには応答を要求するようなイベントが発生してから経過時間を計測するタイマが必要である。必要とする応答時間はイベントによって必ずしも一定ではない。また幾つかの推論が同時平行して実行される場合には複数のタイマが設定できなければならない。時間の進行により推論のサイクルに割り込める仕組みが必須である。

・ 推論の変更

一定時間内に必ず応答が要求される場合には推論の途中での応答可能性をチェックする場合がある。そのままでは時間内に応答を生成するのが不可能な場合には途中で推論方法を変更し、以前の事例を用いたり、簡便な知識ベースで最低限の精度の応答を生成する。

・ 優先順位処理

E S が推論中に対象システムに重要なイベントが発生しその応答を生成するのが急務となるケースはよく発生する。優先順位に基づく推論の一旦停止／優先推論の実行／元の推論の再開等の仕組が必要である。

・ 高速推論・並列推論

必要とする応答時間内で精度の高い制御値・操作を決定するには限られた資源内で高速推論を行う仕組が必要である。場合によっては複数知識ベースを用いて平行推論を行う場合もある。

④ 高信頼性と精度の向上に対する E S の基本タスク

- 安定した制御値・操作の保証

E S が出力した結果の精度をチェックし対象システムの操作許容範囲内であることを保証する。これには時間的制約も含まれる。場合によっては E S 手法ではなく動作範囲が確定している従来手法による二重チェックも用いられる。

- 推論途中結果の精度計測

何らかの精度評価関数の設定ができる。あるいは推論結果を改良してゆく推論方式を採用する。

- 推論の過程・結果を記録・利用

結果の精度を評価する方法を確立し、それにより E S 内の知識ベースを自動的に更新することができる。

- 高使用可能性

制御型システムは長時間にわたる継続した安定稼働が原則である。

E S 自身の安定稼働は勿論のこと、障害時／誤動作時のバックアップへの切り換え／回復機能が重要である。その場合には E S 内の知識ベースおよび対象システムのモデルのバックアップ E S への転送の仕組が確立していなければならない。あるいは E S を二重化する場合もある。

(4) 制御型システムの V & V

① 知識ベースの妥当性評価

a. 必要性

制御型システムでは知識ベースの妥当性が全体システムの出力品質に直接影響する。知識ベースには制御の方針・論理・手順が収容されるので、単体の知識の妥当性よりも組み合わされたサブシステム全体システムとしての知識ベースの妥当性がチェックされなければならない。

b. 特徴的な評価項目

知識ベースの一貫性、すなわち入力されるデータ・状況に対して予測できる正しい動作を生ずることが重要である。正当性・完備性も勿論重要であるが、それらはシステムの開発においていわゆるデバ

ッギング・テストにより検査される機会がある。

② 問題解決・推論方式の妥当性評価

a. 必要性

適切でない問題解決・推論方式を選択すると、制御の方針・論理・手順を表現しにくかったり、開発・保守の生産性が低下して、制御システムにESを使用する利点はなくなってしまう。更に実時間での応答が不可能になり、システムとして作動しない危険もある。

b. 特徴的な評価項目

システム出力の一貫性を保つために、問題解決・推論方式において因果関係が明確なできるだけシンプルで表現力のあるものを採用するのが望ましい。更にタイマ処理、優先順位処理の有無・レベルが重要評価項目である。

③ 結果の妥当性評価

a. 必要性

制御システムの出力品質は全体の業務プロセスへの損益に直接影響する。何らかの方法で出力品質を測定することがシステム改善の手掛りであり、大きな利益を生み出す機会でもある。

b. 特徴的な評価項目

正解率あるいは最適性等の測定方法が明確であればそれを使用する。しかし場合によってはそのような尺度が不明確であったり、無かったりする場合も多い。その場合は全体としての業務プロセスの生産性向上等を尺度とする。更にシステム改善のためにその尺度が知識ベースあるいは推論方式の妥当性に容易にフィードバックできることが重要である。

④ 要求仕様に対する達成度評価

a. 必要性

制御システムにESを用いるのは、他の方法と比べて何等かの優位性、例えば開発・保守の生産性、結果の品質向上等の期待があるからである。要求仕様にはそれらが含まれるべきであり、それに従って評価されなければならない。

b. 特徴的な評価項目

特に信頼性、性能の評価が重要である。

⑤ 知識ベースの保守性評価

a. 必要性

制御型システムにおけるESの利点は特に知識ベースの保守性の飛躍的な向上である。専門家が持つ制御知識を理論やアルゴリズムに変換することなく直接的に表現できるメリットは大きい。知識ベースの保守性を妨げるような設計はせっかくのESの利点をなくすることに等しい。

b. 特徴的な評価項目

知識ベースには多くの制御知識が表現されて複雑にからみあっている。結果の妥当性評価により知識ベースの保守を行う場合には制御知識と結果の因果関係が明確で、しかも変更個所が容易に発見できることが必須である。

⑥ マンマシンインタフェースの妥当性

a. 必要性

制御型システムでは直接の利用者は人間系でなく、その意味のマンマシンインタフェースの考慮は不要である。システムの開発・保守あるいは妥当性評価のためのマンマシンインタフェースは必要である。

b. 特徴的な評価項目

特になし。

⑦ システムの総合評価

a. 必要性

制御型システムは単独で使用されることはなく、制御プロセスの構成要素として働く。システム全体での総合評価が要素としてのESの評価でもある。

b. 特徴的な評価項目

他の要素との関係や統合性の良さが重要な評価項目である。

(5) 検証および妥当性検査 (V & V) の方法

制御型システムにおいては全体のシステム要件とそれを駆動するESの要件が比較的によくリンクする。たとえば実時間性の強いシステムではタイマ機能や優先順位処理などは必要条件となるだろう。全体のシステム要件からESの要件・機能にマップするチェックリストは作成が可能であろう。項目によってはレベルの設定も可能であろう。

制御型システムにおいては対象となるシステムの性格によりESへの妥当性の要求度合にかなりの差異がある。しかしながら適用分野の重要性の考慮、および出力がそのまま制御値・操作として使用され、専門家によるチェックを経ない場合が多いこと等から他の問題領域よりもより厳しい要求があることが多い。

これまでに述べた様にシステムは複雑になりがちでテスト以外に実際的な妥当性検証方法は今のところないように思える。複雑な制御システムは通常モジュラ構造になっており、対象システムのサブシステム・部分を取り扱う局所的な知識ベースは比較的小さくテスト以外の方法でも検証は可能かもしれない。しかしそれは全体としてのESの動作を検証できるというわけではない。

これはESに限ったことではないが、対象システムの誤動作が生み出す損失をリスク分析し効率的な重点テストを計画したり、テスト回数を設定することは可能であろう。

制御型システムはその出力の評価が統計的に分析が可能である。それは出力回数が多いことと、目標とする対象システムの安定稼働や生産性により比較が容易であるからである。さらにテストにおいて対象システムの実稼働時に現在の制御システム（または専門家）との比較を行うことが可能な場合が多いからである。さらには入力データの生成による限界分析・感度分析も比較的容易であろう。

2.6.6 複合型システム

(1) 問題領域の特徴

① 定義

複合型システムは、解析型問題(analysis problem)と合成型問題(synthesis problem)が混合したシステムとして定義できる。従って、解析型問題と合成型問題の特性を併せ持つ。制御型システムと非常に似たところがあるが、ここでは、知的ナビゲーションシステムや知的C A Iシステムなどに見られるように、ある時点の状態を分析し、その状態に対して最適な解(プラン)を見つけだすシステムとして定義する。つまり、まずシステム自身の内部状態、あるいは外部からの情報を分析し、対象物の状態を推定する。その推定結果を基に、決められた拘束条件のもとで、所与の要件を満たす解を組み上げ、実行するタイプのシステムである。具体的な説明のために、知的C A Iシステムを事例として、問題の特徴、V & Vの問題意識、V & Vについて述べる。

② 問題の特徴

基本的には解釈型問題を扱う典型的な解釈型システムと合成型問題を扱う典型的な計画型システムの特性を受け継いでいるが、それぞれの特性はかなり異なる。知的C A Iシステムでは、生徒の理解度を把握する部分が解釈型問題であり、その結果に基づいて教授戦略を立案するのが合成型問題として捉えることができる。その詳細を以下に述べる。生徒に色々な質問をすることによって、その生徒の理解モデル(一般的には誤答モデルが用いられる)と比較することによって、その生徒の理解度を分析するとともに、何処が理解出来ていないかを推定する。従って、一般的に知的C A Iシステムでは、教師モデル、生徒モデルなどの対象(人間の思考)に関するモデルを利用する。しかし、真の誤りや生徒の理解度を的確に把握するのは非常に難しい。なぜなら人間の思考を外的情報から推測するためで、得られる情報が表層の情報であり、真の誤りはその裏側に隠れている場合が多いからである。このように非常に多くの不必要な情報や重要な情報が欠落した不完全な情報から真に必要な情報を探す(推定)のは非常に難しい。さらに、このような不完全な情報に基づいて状態を推定すれば、不十分な状態推定となる。このように、解釈型問題と言えども不完全な情報に基づく解釈であり、対象モデルの記述と共に曖昧性を含んだ問題という特徴を充分理解する必要がある。次に、推定された状態に対して、最適な

教授戦略を立案する必要がある。教授戦略としては、生徒の能力と教授目標にそった適切な学習プランを立案し、その計画に基づいて最適な教育を実施する。生徒の能力推定は、不十分な状態推定から行われるのでかなりの曖昧性を含んでいる。したがって、策定されるべき学習プランの問題空間は、ベースとなる生徒の能力推定があやふやなので問題空間自体かなりの曖昧性を持つことになる。教授目標は比較的明確なので、現在の理解度と生徒の能力から教授目標を達成するための学習プランを立案するのはさほど難しくはない。多くの場合、学習カリキュラムとしていくつかのパターンかを用意し、その中から最も良さそうな学習カリキュラムを選択することになる。このことは、既存計画の再利用に相当する。選択された学習カリキュラムが適切か否かを評価するのは難しい。人間の思考モデルと物理的な学習カリキュラムとの総合評価となる。その評価結果によっては、学習カリキュラムを動的に変更する必要もある。

(2) V & Vの問題意識

複合型システムのV & Vを考える上での問題意識として以下の項目が考えられる。

①対象のモデル化

知的C A Iシステムを例にとると、モデル化する対象として、学習者側の理解（誤答）モデル、教師側の教授モデルなどがある。学習者の理解度を正確に把握して、何が理解できなくて、どこをどのように間違ったかを推定することが知的C A Iシステムでは非常に重要となる。従って、学習者の理解度をベースとしたモデル化が行われ、そのモデルとシステムが持っているモデルの比較検討が行われ、学習者の理解度を把握する。対象のモデル化には色々な手法が採られるが、解釈型問題で最も一般的なのが分類階層による方法である。分類階層でモデル化できれば、問題解決機構としては、分類階層に沿った探索手法を用意すればよい。しかし、大規模探索問題では、単純な分類階層では組み合わせ的爆発を防ぐことができない。また、複雑な問題では、単純な分類階層ではモデル化がかんなんな問題も多い。本質的には、深い知識による対象のモデル化が必要となろう。

②不完全な情報に基づく対象の状態推定

学習者の状況を把握する場合、全ての情報が集められていない場合が多い。システムは会話しながら、必要な情報を収集しようとするが“わかりません”

という回答に頻繁に出くわしてしまう。このような欠落した情報や不明瞭な情報から状況を把握する場合があるので、曖昧さを処理する機能が必要となる。もっとも似たモデルを探すためには、少々の不完全性を克服できるモデルのあいまい検索機能が必要となる。一般的な曖昧検索を実現する方法として類推がある。類推は、与えられた対象（モデル）の間に成立する類比を見出し、その類比を利用して、1つの対象に成立する知識を別の対象に変換することによって、問題の解決の手掛かりを得る推論方式である。

③ 解釈の多様性

曖昧さを許して状態を推定すると、その推定結果を解釈する上で多様性が生じる。その多様性を処理する機能も必要となる。多様性をも考慮した、学習者の理解度が推定されたならば、理解度を増進させるための教授戦略を立案する。そのための基本タスクとして、まず、学習者の状態の推定結果から、教授戦略立案のための制約条件の作成がある。作成された制約条件は、相互の関連性を見捨てて作成される場合が多いので、作成された制約条件間で矛盾があったり、教授戦略の立案が不可能な制約条件が作成される場合がある。それらの制約条件の相互作用を処理するタスクが必要となる。

④ 組み合わせの爆発を防ぐ推論

最終的に制約条件が決定したら、合成型問題と同様な組み合わせの爆発を防ぐ手法での探索タスク（推論機能）が処理をする。組み合わせ的爆発を回避する方法として、探索空間を階層的に分割した階層的探索手法がとられる。階層的探索手法で一般的に利用される手法は、階層的生成検査法である。階層的生成検査法は、『システムの構造を階層的に表現し、部分構造を段階的に生成し、その妥当性の評価を行って、有効ではないと判断される部分構造を排除しながら、最適なシステム構造を見つけだそう』（知識工学：小林重信著から引用）とする手法である。組み合わせ的爆発を回避する別の方法にトップダウン詳細化戦略がある。この方法は、『まず詳細な部分を見捨て、抽象度の高い水準での粗い計画を立て、次により詳細な部分を考慮に入れて計画を詳細化する過程を繰り返しながら、最適なシステム構造を決定していく』（知識工学：小林重信著から引用）手法である。この方法では、高レベルの探索空間を小さいものに限定し、かつその解が次のレベルでの探索に対する制約条件となる。小さいものに限定された探索空間を副問題として捉え、副問題の目標をサブゴールとして定義すると、サブゴール間の相互作用が制約となり、制約をシステムが

達成する目標の部分記述とみなすことができる。各副問題ごとに局所的に何らかの決定の行える情報が利用可能な場合、その副問題ごとに完全な解を与えないで、最小限の部分的な解の記述に留め、決定の一部を先送りにする方法を制約条件の最小化と言い、問題の分割化をおこなうための有効な手段となる。問題を副問題に分割し、制約を副問題のサブゴール間の相互作用として推論していくには、推論の過程でのサブゴールを評価しながら推論していく必要がある。一つの副問題で複数のサブゴールが生成される場合には、最良優先探索などの手法で解を生成していく。ある制約条件の下で、解が生成されたとして、その解が全体として最適な解とは言えない場合がある。解全体の整合性をとりながら、より良い最適解を捜し出す方法に焦点調整法がある。焦点調整法は、連結されたサブゴールの集合として解を捉えれば、全体の解の中で不具合／不適切なサブゴールを捜し出し、その不具合／不適切な状態を解消して、より良い解とするために複数のサブゴール間でサブゴールの交換を行うものである。この交換は、当然、より良い解を生成する場合に交換が成立する。焦点調整法を適用する場合に、作成した計画案の評価法が重要である。焦点調整法で用いられる一般的な評価法は、得点法である。各々のサブゴールを評価するために、問題の特性をよく捉えられる方法で評価関数を決める。その評価関数に従って、各サブゴールの得点を計算する。焦点調整法では、その得点の低いサブゴールから交換していくことになり、交換によってサブゴールの得点が高くなる場合に交換が成立する。この交換は2つ以上のサブゴールでも成立させる必要がある。

(3) 複合型システムにおける V & V

① 知識ベースの妥当性評価

複合型システムでは、対象の状況を捉えるための解釈部の知識ベースとその結果を制約として作成される計画部の知識ベースを分けて評価する必要がある。解釈部の知識ベースでは、生徒の理解モデルが中心となる。理解モデルは、人間の思考のモデル化、対象領域の問題（教材）に関する知識のモデル化が重要となる。特に、人間の思考のモデル化は、人間の思考自体、構造が非常に複雑であり、かつ、かなりの曖昧性を含んでいるため、非常に難しい対象である。対象が難しいので、特に知識ベースの評価が重要となる。曖昧性を含んだ知識、特に浅い知識を前提としたシステムでは、知識の欠落が頻繁に起きる可能性が

あるので、知識ベースの完備性が重要となる。計画部の知識ベースでは、解釈部で決定された制約が曖昧性を多分に含んでいる可能性がある。間違っただけの計画案を作成する可能性がある。それらを防ぐための知識など、不完全性を前提とした知識ベースであり、妥当性を評価するのは非常に難しい。

② 問題解決・推論方式の妥当性評価

複合型システムでは、解釈部の推論の結果を基に計画部が教授戦略を立案する。その結果に基づいた教材を生徒に呈示し、その反応を基に解釈部が生徒の状態を推論する形となる。このように人間とシステム、システム側も解釈部と計画部が連なった形の推論が行われるので、全体の整合性のとれる問題解決・推論方式である必要がある。

③ 結果の妥当性評価

知的C A Iシステムでは、結果の妥当性の評価が極めて難しい。その理由は、誤る原因となるのが生徒自信にあるため、開発者側だけでは結果（誤り原因の同定）が正しいか否かの判定が行えないからである。特に、生徒の誤り方も千差万別であるためにK Eと専門家では、完全なる結果の妥当性の評価が行えない。そのため、実際に結果の妥当性を評価するには、授業に適用しながら妥当性を評価していく必要がある。

④ 要求仕様に対する達成度評価

知的C A Iシステムにおける要求仕様は、対象生徒の理解能力（出来る子、全然出来ない子など）に応じた最適教材の呈示による最大学習効果にある。

このように目的意識の比較的明確なシステムでは、要求仕様に対する達成度の評価は絞り込みやすいが、最大学習効果など抽象的な評価項目を定量的に捉えるのは非常に難しい。

⑤ 知識ベースの保守性評価

曖昧性、不完全性を前提としたシステムにおいては、知識ベースの変更が頻繁に発生することを前提とする必要がある。従って、容易に知識ベースの変更ができる構造にするとともに、変更に伴って知識ベース全体の整合性が崩れないような保守ツールの整備が重要となる。

⑥ マンマシンインタフェースの妥当性評価

知的C A Iシステムでは、知識処理部はオーサリング（教材呈示）システムに組み込まれる形となるので利用者インタフェースは表に現れてこない。しかし、教授戦略に基づいた最適の教材を呈示する必要がある。知識処理部の

出す結果と呈示する教材の幅と深みとは密接に関連する。従って、知的C A Iシステムでは、教材作成時から推論システムとの連携を考慮した利用者インタフェースの開発を行う必要がある。

構築時、保守時のインタフェースは、特に頻繁に変更作業が発生することを考慮した変更の容易性を実現したものとする必要がある。また、知識ベースの完備性、一貫性を容易に検証できる知識構造の図的表示機能などを具備したインタフェース機能が望ましい。

⑦システムの総合評価

複合型システムでは、解釈部と計画部が連携して問題を解決する必要がある。ので、全体の整合性を考慮した評価法が重要となる。知的C A Iシステムで最も難しい問題が未知事象に対する評価法である。浅い知識の表現では、知識の網羅性の保証が非常に難しく、生徒の誤り原因が推定不可能な状態が発生するであろう。その場合の対処法として、知識ベース構築者に不足の知識を促す（ログ等で）ための機能を考慮する必要がある。つまり、知的C A Iのような非常に難しい対象を知識化するシステムの総合評価としては、運用時における未知事象にどのような対象できるかという点まで考慮したシステムであるか否かが評価の重要なポイントとなる。

2.7 評価者別の重点評価項目

先に述べたエキスパートシステム開発のライフサイクルとエキスパートシステムの評価項目を鑑み、評価者別の重点評価項目について説明する。

(1) 評価者の定義と役割

エキスパートシステムの評価者として、マネージャ、専門家、第3者専門家、ユーザ、開発担当者、運用担当者を取り上げ、その定義と役割を述べる。

① マネージャ

マネージャとは、システム開発プロジェクトのリーダーとして、プロジェクト全体の管理を行ない、プロジェクトを円滑に進める責任を担う人をいう。すなわち、ここではプロジェクトマネージャとして捉え、経営管理者であるかは特に言及しない。

エキスパートシステム開発のライフサイクルの調査・計画フェーズでは、まずシステム導入に関する企画提案書の妥当性を裏付けるための調査報告書を作成する。この報告書には、現状の問題点、問題を解決するための技術ポイント等が示される。これは、以降の開発フェーズにおいて何を確認すべきかの指針となる。これに沿って、プロトタイピング、実用システム開発、運用テストの各フェーズにおいて、要求仕様の内容や要求仕様に対する達成度を評価することにより、次フェーズ移行への見極めを行なう。そして、実用システム完了時には、性能やコストパフォーマンス等といったシステムの総合評価を行なうことになる。

② 専門家

専門家とは、システム化の対象とする問題領域において専門的知識を有し、その知識を提供することによって、積極的にシステム開発に参画する人をいう。

専門家は自分が提供した知識の妥当性評価をはじめ、問題解決方法や推論方式、システムで得られる結果の妥当性の評価、知識ベースの運用を行なう際の保守性評価等、評価を行なう項目は多義にわたる。また、システム開発の各フェーズでは、要求仕様書を作成し、それに従って構築されたシステムを検証することにより、要求仕様の達成度やシステムの成長可能性等の評価を行なう。

なお、ここで定義する専門家はその領域に対しての知識提供者であり、通常一人と考えており、その他は第3者専門家として扱うことにする。

③第3者専門家

第3者専門家は、前述の専門家と同等の専門的知識を有し、専門家から獲得した知識によって構築した知識ベースの妥当性、システムで用いている問題解決方式の妥当性、及びシステムで得られる結果の妥当性等の評価を行なう。但し、第3者専門家は実際にシステムを検証する必要はなく、ドキュメントだけによって評価を行なうこともある。

④ユーザ

ユーザとは、システム化のニーズを提案し、開発されるシステムを最終的に利用する人をいう。システム開発の各フェーズで、利用者の立場として要求仕様書を作成し、主に推論時間の妥当性やマンマシンインタフェースを含めたシステムの操作性の評価を行なうことになる。

⑤開発担当者

開発担当者は、専門家の持つ専門的知識の抽出、整理を行ない、それを実際にコンピュータ上に実現し、システムを構築する。

エキスパートシステム開発のライフサイクルの調査・計画フェーズでは、ユーザが提示するシステム要件を把握し、システム化対象の妥当性や開発手法の妥当性の評価、及び開発ツール・言語等の開発環境の評価等を行なう。プロトタイプフェーズでは、専門家やユーザから提出された要求仕様書をもとに、プロトタイプシステムを構築し、知識規模・知識源の妥当性評価、問題解決・推論方式の妥当性評価を行なう。実用システム開発のフェーズでは、プロトタイプでの成果や実用版要求仕様書から実用版設計書を作成する。ここでは、保守性評価に関する項目やシステムの総合評価に関する項目も折り込まれる。

⑥運用担当者

運用担当者は、運用・洗練化のフェーズにおいてシステムの運用を行なう。したがって、開発担当者と同一者でも構わない。知識ベースの保守は主に専門家が担うが、運用担当者は、知識間の制御に関わる知識(メタ知識)等の保守を行なうことになる。

(2) 開発のライフサイクルにおける評価者別の重点評価項目

各々の評価者がエキスパートシステム開発のライフサイクルにおいて重要な評価項目を表2.7-1に示す。ここで、重要な項目は○で表し、その中でも特に重要なものを◎で表している。また、△は条件付で重要になるものであり、数字は以下の事由を示している。

1. 解の正当性が定量的に測定可能である場合
2. 応答時間、インターフェース等、利用者に限定した部分の要求仕様

表 2. 7 - 1 評価者別の重点評価項目一覧

分類	評価項目	開発フェーズ	評価者	プロトタイプینگ						実用システム開発				運用テスト				運用・洗練化			
				実験型			発展型														
				マネージャ	ユーザ	開発担当者	専門家	ユーザ	開発担当者	専門家	ユーザ	マネージャ	開発担当者	専門家	第3者専門家	ユーザ	マネージャ	ユーザ	専門家	運用担当者	マネージャ
知識ベースの妥当性評価	(A) 知識ベースの正当性評価 [P]												◎	○					○		
	(B) 知識ベースの完備性評価 [P]												◎	○					○		
	(C) 知識ベースの一貫性評価 [P]												◎	○					○		
	(D) 経験的知識を正当化する方法とその確認 [E]						○			◎					○						
	(E) 知識表現の選択と適切さの確認 [D]						○	○		◎	○										
	(F) 知識獲得方法の選択と適切さの確認 [D]						○	○		◎	○										
問題解決・推論方式の妥当性評価	(A) 推論処理の妥当性評価 [P]								○	○			○	◎	○				○	○	
	(B) 推論時間の妥当性評価 [P]								○	○			○	◎	○	○		○	○		○
	(C) 推論手法の選択と適切さの確認 [D]						○	○		◎	○										
	(D) 不確実情報に対する処理技法の適切さの確認 [D]						○	○		◎	○										
	(E) 学習機能の適切さの確認 [D]						○	○		◎	○										
結果の妥当性評価	(A) 解の正解率評価 [P]									○			◎	○	△ ₁	△ ₁		○	○	△ ₁	○
	(B) 解の網羅性評価 [P]									○			◎	○				○	○		○
	(C) 解の最適性評価 [P]									○			◎	○				○	○		○
	(D) 解の評価方法の適切さの確認 [E]						○			○					○						
要求仕様に対する達成度の評価	(A) 要求仕様内容に対する評価 [P]	○				○			○			○	○			◎				○	○
	(B) 要求仕様に対する達成度評価 [P]						○	△ ₂		○	△ ₂		○		△ ₂				△ ₂	○	
	(C) 要求仕様設定手段の評価 [D]						○	○		○	○			○							

P: プロダクトに対する評価 E: 評価方法に対する評価

D: 開発/保守方法・技法・ツールに関する評価 O: その他の評価

(次頁に続く)

開発フェーズ		調査・計画		プロトタイプング						実用システム開発				運用テスト				運用・洗練化							
				実験型			発展型																		
				マネージャ	ユーザ	開発担当者	開発担当者	専門家	ユーザ	開発担当者	専門家	ユーザ	マネージャ	開発担当者	専門家	第3者専門家	ユーザ	マネージャ	開発担当者	専門家	第3者専門家	マネージャ	ユーザ	専門家	運用担当者
分類	評価項目	評価者																							
知識ベースの保守性評価	(A)変更箇所発見の容易性評価 [P]																								
	(B)知識ベースの理解性評価 [P]																								
	(C)変更箇所の局所性評価 [P]																								
	(D)知識ベースの統一性評価 [P]																								
	(E)品質安定性の確認方法の評価 [D]																								
	(F)保守マニュアルの整備状況の確認 [D]																								
	(G)保守用ツールの整備状況の確認 [D]																								
M M 妥 当 性 評 価	(A)利用時のインターフェース評価 [P]																								
	(B)保守時のインターフェース評価 [P]																								
シ ス テ ム の 総 合 評 価	(A)テーマ選定の妥当性評価 [P]	◎		○																					
	(B)利用者の満足度評価 [P]		○																						
	(C)統合性の評価 [P]	○		○																					
	(D)成長可能性の評価 [P]				○	○		○	○		◎	○	○	○		○		○	○	○				○	
	(E)開発／保守用ドキュメントの評価 [P]	◎	○	○	○	○	○	○	○	○	◎	○	○	○	○	◎	○	○	○	○	◎	○	○	○	◎
	(F)開発手法の適切さの確認 [D]	◎		○																					
	(G)ツール・言語選択の適切さの確認 [D]				○			◎			○														
	(H)システム開発の生産性評価 [O]	○														○		○		◎					
	(I)システム開発のコストパフォーマンス評価 [O]	○														◎		○		○					
	(J)セキュリティ対策の評価 [D]	○														○	○	○	○		○	○	○	○	○

P:プロダクトに対する評価 E:評価方法に対する評価
 D:開発/保守方法・技法・ツールに関する評価 O:その他の評価

2.8 エキスパートシステムの評価とテスト方法

本節では、エキスパートシステムに対するテスト方法について、研究課題をも含めて、網羅的に検討する。

2.8.1 概要：テスト方法の類型化

エキスパートシステムの品質を向上させるには、エキスパートシステムはソフトウェアの一種であり、ソフトウェアはエンジニアリングシステムの一種でもあるので、これらの分野で高度・大規模システムに対して有効性が実証された開発・検証・管理技術の根本思想（原理、原則）を積極的に活用することが要諦である。即ち、エキスパートシステムのテストに関しても、ソフトウェアに対するテストや、エンジニアリングシステムに対するテストの考え方が十分適用可能で、有効である。

テストはテスト対象を実行させるか否かによって、次の2つに大別される。

① 静的テスト …… テスト対象を実行させない

② 動的テスト …… テスト対象を実行させる

また、テストをする手段が人手を主体としたものか、コンピュータを活用したツールを主体としたものかによって類型化することもできる。人手を主体とした手法は、レビューとか、ウォークスルーとか呼ばれる。ソフトウェアでは、最近では、重大な欠陥の検出能力の点でレビューの方が優れていることが実証されたこともあり、レビューを重視するようになってきている。

テストに際して、テスト対象に対する分析や解析を主眼にしている場合には、テストという用語よりも、分析や解析という用語（英語ではanalysis）がしばしば用いられる。

動的テストは、テスト対象をブラックボックスとして扱うか、ホワイトボックスとして扱うかによって、更に次の2つに大別される。

① ブラックボックステスト

② ホワイトボックステスト

ブラックボックステストは、主として要求仕様書（特に機能仕様書）に規定された機能が正しく実現されているかを調べるために行われるので、機能テストとも呼ばれる。一方、ホワイトボックステストは、システムの内部構造が設計仕様書に規定された通りに正しく実現されているかを調べるために行われるので、構造テストとも呼ばれる。

妥当性検査 (Validation) と検証 (Verification) という観点で言えば、妥当性検査は「正しいシステムが作成されたか」を調べ、一方、検証は「システムが正しく作成されたか」を調べることを、それぞれ主眼に置いている [O'Keefe et al., 1987]。妥当性検査は、完成したシステムがユーザの真のニーズを満たしているかを問題にしているもので、いわゆる狭義のテストの範ちゅうを越えている。例えば、獲得した知識や規定された要求仕様そのものの正当性は Validation の世界と考えられる。しかしながら、現実には、要求仕様を一応ユーザの真のニーズを表すものと仮定し、そこで要求された機能・性能を実現しているか、というテストを妥当性検査と考える場合もある。一方、検証は、要求仕様そのものは正しいという前提条件で、要求仕様書で規定された条項 (要件) が各工程毎に、それぞれのレベルに応じて正しく実現されたかを問題にしているもので、工程間及び工程内の主として一貫性を検査する狭義のテストと同じ範ちゅうであると考えられる。

真のエキスパートシステムは一般に悪構造の問題で、開いた世界 (open world) (即ち、人間がすべてを完全には制御・想定・予測できない現実の世界) を対象としているので、モデル化の際に仮定・設定した閉じた世界 (closed world) での検証はテストに要求される必要条件ではあるが、十分条件を満たしておらず、モデル化の際に意図的に排除した因子をも含めて、開いた世界での妥当性検査が一層重要となってくる。即ち、そのエキスパートシステムが現実の世界でほんとうに使えるのか、ほんとうに役に立つのかがポイントとなってくる。

2.8.2 静的テスト/分析

静的テスト/分析は、その手段によって、更に次の2つに大別される。

- ① 人手によるレビュー/ウォークスルー/テスト
- ② 自動化ツールによる静的テスト/分析

(1) 人手によるレビュー/ウォークスルー/テスト

a. レビュー

レビューとは、「人手によってドキュメントやプログラムが正しいかを克明に調べる」ことであり、特に、デザインレビューは、「アイテムの設計段階で、性能・機能・信頼性などを考慮しながら設計について審査し改善を図ること。審査には、設計・製品・検査・運用などの各分野の専門家が参加する」(JIS用語)であり、「製品の設計品質及びそれを具現化するために計画された製造・輸送・据付け・使用・保全などのプロセスについて、客観的に知識を集めて評価し、改善

点を提案し、次の段階に進みうる段階にあることを確認する組織的活動の体系」ということができよう。

エキスパートシステムのデザインレビューにおいて、今後重要となる分析／解析は、リスク分析 (Risk analysis) とハザード解析 (Hazard analysis) である [Leveson, 1986]。

リスク分析は、「システムの誤りによって起きると考えられる損失を予測するためのプロセス」と定義される。分析の目的は、リスクにさらされる可能性を確認し、そのリスクの程度を数量的に推測し、そのリスクを優先順位別にランクし、そして、コスト効果の高い解決を可能にすることである。

エキスパートシステムが医療・原子力・航空・宇宙・防衛・化学工業のような安全性が致命的な分野で、リアルタイムで直接的な監視・制御に使用された場合、それ自身および全体システムの安全性が重大な問題となってくる。エキスパートシステムの安全性が実証されないかぎり、これらの分野でのエキスパートシステムの利用は限定されるであろう。(特に、エンジニアリングにおける有名な Murphy の法則*を常に意識することが肝要である。)

[注]*: Murphy の法則を一言でいえば、複雑なエンジニアリングシステムにおいて、極端に確率が小さいが、悪いことが起こる可能性があれば、確率とは関係なく、使用中に、その悪いことが必ず起こるという法則である。

信頼性 (Reliability) は、一般に、システムが特定の諸環境条件下で、意図した機能を遂行する確率と定義されている。一方、安全性 (Safety) は、意図した機能が遂行されているか否かにかかわらず、事故 (mishap) を引き起こす諸条件が生じない確率と定義されている。しばしば、信頼性と安全性は同義であるとみなされるが、この2つの概念を明確に区別することが、その実現・対策を、あるいは、評価を考える上で、要諦である。

信頼性での故障 (Fault) に対して、安全性では、人間または資材に損害を与える実在あるいは潜在の状態を“ハザード” (Hazard) と呼ぶ。ハザード解析は、システムにおけるハザードを識別し、その影響のきびしさと、発生確率を調べ、安全性を評価する方法である。

ソフトウェアのハザード解析では、ソフトウェアのハザードと、安全性に重大な影響を及ぼす単一および複合の障害 (failure) のシーケンスを識別し、タイミング要件を含めたソフトウェア安全性要件を決定し、ソフトウェアの安全性を解析して、計測する。ソフトウェア安全性要件には、システムが何をなすべきだけでなく、何をしてはならないかも含まれていなければならない。

ハザード解析の補助手段である故障の木解析 (FTA: Fault Tree Analysis) で

は、木の頂上（根）にシステムの使命達成上、好ましくない事象（故障）をあげ、次にその発生の原因となるとされるすべての事象をあげ、これらをANDあるいはORなどの論理記号で結び、さらに、それらの事象の原因となる副事象を調べ、順次木の枝のように組み立てていき、故障発生経路とその根本原因をつきとめる解析方法である（図2.8-1参照）【Leveson, 1986】。

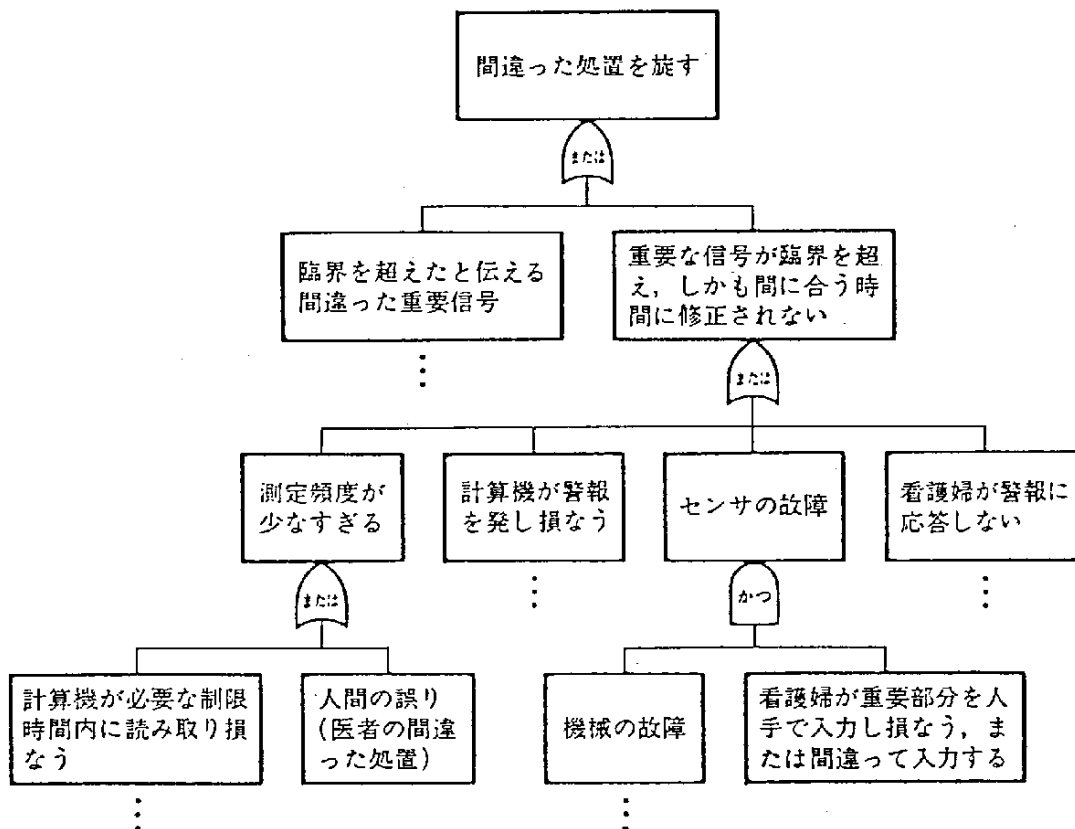


図2.8-1 故障の木解析（F T A）の例：患者監視システムの場合

ソフトウェアのF T Aでは、ソフトウェアの詳細設計あるいはコーディングの段階が終了した時点で、F T A手順を使用して、故障の木の上にあるクリティカルな制御故障から後向きに遡行して、プログラムが故障の木の頂上の故障（事故）を引き起こす原因となるかどうか検証する。

デザインレビューを実施することにより、エキスパートシステム開発の工程短縮、品質及びコスト有効性（Cost effectiveness）の評価及び確保が可能となる。

b. ウォークスルー

ウォークスルー (Walk-through) もレビューの一種で、「1人の設計者またはプログラマーが開発チーム内の他の要員を指導して、彼らが書いた設計またはコードのセグメントを通じて、技法、仕事のやり方、犯しやすい誤り、開発規格に対する違反やその他の問題について他の要員に質問したり、意見を述べたりする再検討の過程」(ANSI/IEEE用語)であり、特に、コードウォークスルーは、「人手でプログラムコードの1行1行が正しいかを克明に調べる」ことであり、知識ベースや問題解決・推論プログラムを1行1行正しいかを克明に調べることであり、信頼性を向上させることが可能となる。

c. テスト

ここで検討対象とするテストは、チューリングテスト（あるいは目かくしテスト）である。

チューリングテストは、コンピュータに“知能”があるかを判定するテストで、特に評価者のエキスパートシステムに対する偏向を除去する有力な手段である。

MYCINでは、菌血症の評価と脳膜炎の評価が行われ、チューリングテストの重要性を如実に示した [Gaschnig et al., 1983]。最初の菌血症の評価では、MYCINの能力を評価した専門医は、評価対象がコンピュータの出力結果であることを事前に知っていた。彼らのコメントや批評の大半は、医療分野におけるコンピュータの正しい役割に対する彼ら自身の偏向を反映したものであった（例えば、「コンピュータでは、患者を診ていないので、何の病気であるかの適切な診断は下せるとは思えない」という指摘が示している。）。この評価結果を踏まえて、脳膜炎の評価では、100個のテストケース（症例）に対し、MYCINが下した診断結果及び治療方法 (recommendation) と、同じ症例を被験者9名（伝染病が専門の医学部局員から医学生まで）に依頼して得た診断結果及び治療方法とを混ぜて、外部からはどちらのそれなのかを分からないようにして、公的機関の専門医8名に診断結果及び治療方法の妥当性について意見を求めた。このような形式のテストを行うことにより、評価者の潜在的な偏向を除去することができ、MYCINの能力のレベルが医学部専門医から医学生のどのあたりに位置するものであるかが分かった。

エキスパートシステムがチューリングテストに合格することは必要条件ではあるが、このチューリングテストに合格したからといって、真の知能が実現されたかは別問題である。

(2) 自動化ツールによる静的テスト／分析

静的テスト／分析を自動化するツールとしては、知識ベース、特にルールベースを対象とするものが研究開発されているので、ここで紹介する [O'Leary & O'Keefe, 1989]。

a. ルールベースの一貫性及び完備性の分析 [Nguyen et al., 1987]

ルールベースの一貫性を阻害するルールは次に示すように分類される。

- 冗長ルール (Redundant rule)
- 競合ルール (Conflicting rule)
- 包摂ルール (Subsumed rule)
- 不必要なIFの条件項を持つルール (Unnecessary IF Conditions)
- 循環ルール (Circular rule)

そして、ルールベースの完備性を阻害する誤りは次に示すように分類される。

- 未参照の属性値 (Unreferenced attribute values)
- 誤った属性値 (Illegal attribute values)
- 欠如ルール (Missing rule)
- 未到着な結論部を持つルール (Unreachable conclusions) (IFの条件部が決して発火しない) (後向き推論の場合)
- 袋小路のIFの条件項を持つルール (当該IFの条件項と同じ結論項を持つルールや事実がない) / 袋小路の目標 (目標と同じ結論項を持つルールがない) (Dead-end IF conditions and dead-end goals) (後向き推論の場合)

これらの一貫性及びの完備性を阻害するルールの検出は、現在自動的に検出することが可能となっており、一部商用ツール化されている。

b. ルール間の相互参照リスト

ルール間の相互参照リスト、即ち、ルールのTHEN部の条件と同じ条件をIF部に持つルールとの相互参照のマトリックスの作成とか、IF部にある変数の条件式が、変数の属性や定義域からみて妥当であるかなどを自動的にチェックすることも可能である。これらのツール機能は、ルール間の相互作用をレビューする際に有用である。

c. 確信値の矛盾検出

ルールに確信値 S (Sufficiency factor) と N (Necessity factor) を付ける Bayesian的な方式では、その S と N の間の論理的矛盾を自動的に検出することも可能になってきている。

2.8.3 動的テスト

(1) ブラックボックステスト／機能テスト

ブラックボックステストあるいは機能テストでのテスト対象は、実現した機能、主としてエキスパートシステムから導出される結果や結論である。

本テストを実施するにおいてもっとも重要な点はテストデータの選定である。一般に、ブラックボックステストあるいは機能テストを完璧に行うには膨大なテストデータを必要とし、その結果、膨大なテスト時間（データ準備、マシン実行などに要する時間）を必要とするので、この点を事前に検討しておく必要がある。

テストデータの選定は従来のソフトウェアのテストの際の選定基準や考え方と同じである。但し、エキスパートシステムは一般にソフトウェアと異なり100%正解の答を出すとは限らないので、解の正解率や網羅率を算出するためのテストでは、統計的に見て、例えば実務的に出現頻度が妥当なテストデータを選定する必要がある。

結果や結論に対する動的テストで検査・評価すべき品質特性の例を下記に示す [Walters & Nielsen, 1988]。

- ①一貫性 (Consistency) …… 知識ベースのどのルールが使われようと、あるいはどんな推論経路をとろうと、同じ入力データに対し、どの程度同じ結論に到達するかを表す度合である。
- ②完備性 (Completeness) …… エキスパート・システムが対象としている問題領域で正しいと考えられている結論すべてが、どの程度導出されるかを表す度合である。
- ③健全性 (Soundness) …… 導出された結論すべてが、どの程度正しいかを表す度合である。
- ④精 度 (Precision) …… 入力データがほんの少し変化した場合に、出力がどの程度変化するかを表す度合である。
- ⑤応答時間性 (Response time) …… 要求された時間内でどの程度結論を導き出せるかを表す度合である。
- ⑥入力順序独立性 (Sequence independence) …… データの入力順序を変えても、導き出される結論がどの程度変化しないかを表す度合である。
- ⑦強固性 (Robustness) …… 誤りのあるデータ、不正確なデータ、または、一部のデータが欠落していても、導出される結論がどの程度正しい結論であるかを表す度合である。

(2) ホワイトボックステスト／構造テスト

ホワイトボックステストあるいは構造テストでのテスト対象は、実現した機能そのものよりも、主として知識ベースに格納されているルール、フレーム、メソッドなどの形式の個々の知識と、結論を導出した推論過程 (Line of reasoning) である [O'Leary & O'Keefe, 1989]。

個々の知識 (特にプロダクションルール) をテストするには、その知識が発火、あるいは使われる条件を明確にし、その条件にあったテストデータを用意する必要がある。従来のソフトウェアのテストの網羅率と同種の網羅率が考えられるが、その網羅率の有効性をも勘案して、網羅率テストを実施する必要がある。

エキスパートシステムでは、確信度を扱うこともあるので、ホワイトボックステストを行って、確信度の伝播過程を検査することも重要である。

結論を導出した推論過程の検査は、通常トレース機能や説明機能を用いて行われるが (、現在の技術水準や開発者の負担を考慮に入れると)、なぜその推論過程を通ったかの正当化説明は後向き推論ではある程度成功しているが、前向き推論では単なるトレース情報にしかすぎず、説明になっていない。なぜならば、前向き推論では、ルールの発火の順序は実質的にはプログラミングの際に決められているので、なぜそのルールが選ばれたかの意味づけができないからである。いずれにしろ、どちらの推論方式でも、説明機能は、基本的にはエキスパートシステムの挙動を忠実に再現するだけであり、ルールベースシステムにおいては、ルールの連鎖を示すだけである (これは、浅い知識による浅い説明のみを実現しているといえる。)。もし、前向き推論で正当化説明を可能にするには、プログラムの検証で1つ1つのステートメントに対してプログラマが前もって記述したアサーション (assertion) が必要のように、例えば、各ルールに人間が前もって記述した正当化説明文を付けておかなければならず、コスト有効性の点で実施されないであろう。別のアプローチとして、ルールベースの他に、深い知識として、領域に関する機能・構造・挙動の3種の知識を用意すれば、何故そのルールがその状況で使われたかのルールの正当性を説明することが可能となる (これは、深い知識による深い説明といえる。)[山口ら, 1987]。しかしながら、深い知識のエキスパートシステム化に関しては、今後の課題でもある。

エキスパートシステム自身による正当化説明の理解性を向上させるには、推論過程の連鎖を説明用に再構成する能力が要求されるが、説明再構成機能はまだ研究課題であり、動的テストでは利用できないであろう。そのため、テスト時には、人間が、その推論連鎖を正当化の説明をしつつ、調べることになる。

2.8.4 ま と め

以下に、本節で説明したエキスパートシステム（ES）のテスト方法を一覧的に示しておく（本節で説明していないテスト方法も、一部含まれている。）。

テスト方法／手段

テストの対象及び目的

静的テスト／分析

デザインレビュー

ウォークスルー

リスク分析

ハザード解析

故障の木解析(FTA)

チューリングテスト

ツールによる静的分析

知識ベース

推論機構

システム全体

信頼性全般

精査

—

—

—

—

一貫性

完備性

ルール間の

相互参照

確信値の

矛盾検出

信頼性全般

精査

—

—

—

—

データフロー

制御フロー

信頼性全般

—

セキュリティ

リスク

安全性

ハザード

信頼性

能力

—

動的テスト

テストデータ法

BBT^{*1}／機能テスト

WBT^{*2}／構造テスト

回帰テスト^{*3}

説明機能

内観分析によるテスト^{*4}

個別目的

—

個々の知識

—

正当性

—

個別目的

—

個々の

ステートメント

—

正当性

妥当性

個別目的

機能

正解率など

—

安定性

—

—

【注】*1: ブラックボックステスト

*2: ホワイトボックステスト

*3: Regression Test; 保守後、過去のテストで使ったデータすべてを使用して、既存の機能が劣化していないことを確認するテストである。

*4: Protocol Analysis; 専門家に自分自身の問題解決過程を克明に話してもらい、それを分析し、ESの推論過程と比較して、ESの推論過程の妥当性を調べる。

【参 考 文 献】

- [1] Gaschnig, J.G., Klahr, P., Pople, H., Shortliffe, E., and Terry, A., "Evaluation of Expert Systems: Issues and Case Studies," in Building Expert Systems, Hayes-Roth, F., Waterman, D.A., and Lenat, D.B., Eds, Addison-Wesley, Reading, pp.241-280, 1983.
- [2] Leveson, N.G., "Software Safety: Why, What, and How," ACM Computing Surveys, Vol.18, No.2, pp.125-163, (June 1986).
- [3] Nguyen, T.A., Perkins, W.A., Laffey, T.J., and Pecora, D., "Knowledge Base Verification," AI Magazine, Vol.8, No.2, pp.69-75, (Summer 1987).
- [4] O'Keefe, R.M., Balci, O., and Smith, E.P., "Validating Expert System Performance," IEEE EXPERT, Vol.2, No.4, pp.81-90, (Winter 1987).
- [5] O'Leary, D.E. and O'Keefe, R., Verifying and Validating Expert Systems, IJCAI-89 Tutorial Program MP4, Aug. 21, 1989, pp.37.
- [6] Walters, J. and Nielsen, N.R., Crafting Knowledge-Based Systems: Expert Systems Made Easy Realistic, John Wiley & Sons, Reading, New York, pp.342, 1988.
- [7] 山口, 溝口, 田岡, 小高, 野村, 角所, 「深い知識に基づく一般ユーザー向けの説明機能」, 電子情報通信学会論文誌D, Vol.J70-D, No.11, pp.2083-2088, (1987年11月) .

3. エキスパートシステム評価ガイドラインに基づく事例調査

3. エキスパートシステム評価ガイドラインに基づく事例調査

本事例調査は、システムのプロフィールを示す部分と評価結果の部分に大きく分け、下記の項目に従ってとりまとめを行った。

① プロフィール

- システムの目的と期待効果
- システムの概要
- 問題のタイプとシステムの特徴
- 客観的な評価の可能性について
- 今後の課題

② 評価結果

- 知識ベースの妥当性評価
- 問題解決・推論方式の妥当性評価
- 結果の妥当性評価
- 要求仕様に対する達成度評価
- 知識ベースの保守性評価
- マンマシンインタフェースの妥当性評価
- システムの総合評価

3.1 エレベータ設計支援エキスパートシステム

3.1.1 プロフィール

3.1.1.1 システムの目的と期待効果

本システムが取り扱う業務は、顧客の要求仕様に応じてエレベータを構成する機器を選定し、それらを建物の限られたスペースに三次元的にレイアウト配置するという業務で、一種の見積もり設計である。

最近のオフィスビルやマンション等の建物は、他にはない付加価値を持たせることが不可欠となってきた。そのため、インテリジェント化などの利便性を追求するとともに、美しさ、イメージ向上を狙って、高級化、差別化した建物が次々と出現してきている。この多様化する建物に対応するため、顧客のエレベータに対する要求も、機能、デザインの両面で複雑になってきている。たとえば、建物の利用可能スペースを最大限に取ろうとすれば、エレベータを取り付けるスペースが狭く限られたものになる。あるいは梁が突出した昇降路になりエレベータ機器の取り付けの妨げになったりする。このような様々な要求仕様に対して、構造面のみならず、法規面、経済面、安全面を考慮しつつ、図3.1-1に示すようなエレベータシステムを構成する機器をいかに選定し、いかに配置するかが問題となる。

一方、建築工法、設備の改良が進み、建設工期の短縮化傾向も著しいものがある。それに伴って、顧客のビル設備計画に対し、商談発生時から顧客の要求に迅速に応える必要があり、人の流れを考慮する交通計算に基づくエレベータの台数、速度、容量などの提示と建物のエレベータ収納スペースの複雑さを考慮したレイアウト図を提示している。しかし、これらの検討には、エレベータを構成する個々の機器に関する設計ノウハウを理解した熟練技術者が多くの手間をかけている。それゆえ、この業務の効率化・自動化を図れば、エレベータビジネスにおける多くの需要に対して短期間で応えられることが期待できる。

3.1.1.2 システムの概要

(1) システムの機能と構造

本システムは図3.1-2に示すように機器選定システムとレイアウトシステムの2つのサブシステムから構成される。

機器選定システムの知識ベースには機器データ（適用範囲・条件など）や

選定ルールが定義されている。また、レイアウトシステムの知識ベースには機器データ（形状・寸法など）やレイアウトルールが定義されている。両サブシステムは、それぞれ、それらを利用して推論を行っている。

システム全体の流れは図3.1-3に示すように、まず、客からの要求仕様（速度・定員・積載・建物の条件など）を入力値として選定を行う。次に、その選定結果（機器の型名など）と要求仕様を入力値としてレイアウトを行う。

① 機器選定システム

顧客からの要求仕様に対して、エレベータシステムを構成する機器（巻上機・おもり・レール・ロープなど）の型名・サイズ・径などの選定を行う。選定の規則はほとんどプロダクションルールで記述できる。しかし、選定を行う際の仕様項目や選定基準値（軸荷重、安全率など）、あるいは選定項目が多様で、かつそれらが有機的に関係づけられるという理由から、ツールとして当社の知的スプレッドシート《A c e k i t》を用いた。これは、図3.1-4に示すようにすべての知識を「階層的に構造化された表」と「その表の中の項目間に成立する関係」として表現する。項目間の関係は数式でも規則でもよい。知識を表という単位で扱い、問題解決機構を表操作の中に組み込む。

機器選定システムでは、仕様入力項目をまとめた仕様入力表と選定項目をまとめた選定結果表がある。また、各機器につき1枚の表と選定に必要なとなる項目（軸荷重、かご自重など）につき1枚の表を持っている。総数50枚となる。図3.1-5に示すように必要となる表だけを表示し、他の表は消去しておく。選定後、その詳細を知りたいときに他の表を参照する。

ある表の1つのますめ（セル）の値を決定するルールは、プロダクションルールで定義されており、他のあらかじめ定義された複数のセル（使用変数）の値がすべて入力、変更あるいは決定されたとき（すなわち、値が未入力の状態のセルがないとき）、起動され推論を行う（データ駆動型前向き推論）。このとき、定義されたすべてのルールが起動され、波及的に処理が進む。図3.1-6は機器の選定ルール例を示している。

②レイアウトシステム

レイアウトシステムは顧客よりの要求仕様（建物の高さ、スペースなど）と機器選定システムの出力結果を入力項目とする。エレベータシステムを配置すべき建物のスペースは狭く限られたもので、その中にいかに機器群を収めるかが問題となる。それゆえ、各機器の機能・安全面・経済面・取り付け条件・保守条件を考慮してレイアウトを行う。

この三次元的なレイアウト問題に対して、図3.1-7に示すようなモデル化を行った。すなわち、実作業において設計者が行うのと同様に機械室平面・昇降路平面・地下室（PIT）平面の三平面に切り分け、各平面間の関係付けを考慮しながら平面ごとに二次元的なレイアウトを行う。

平面間の関係づけ（制約条件）は、ロープやレールあるいは機能的な条件より定義づけられる。

また、各平面における二次元的レイアウトは、

- 機器が相互に干渉しない。
- 各機器の機種に応じた取り付け寸法、メンテナンス寸法を確保する。
- 機器の機能が十分果たせるような適切な配置を行う。

などの制約条件を考慮する。

このとき、たとえば機械室平面において制約条件を満たしたレイアウトをしても昇降路平面において不具合（機器の干渉など）が生じてしまってはいけない。図3.1-8は各機器（群）の関係づけを示している。この図でわかるようにひとつの機器の位置をずらすと他の機器の位置に影響を及ぼす。すなわち、問題部分を修正するとその影響ですでに配置完了済みの機器の位置、あるいは位置関係が不適当なものになってしまう。それゆえ、すべての機器を適切な位置に配置させるよう総合的に判断しなければならない。

出力例を図3.1-9に示す。機械室平面・昇降路平面・PIT平面をウィンドウにして並べ、各平面のレイアウトをそれぞれ行う。ウィンドウの枠は壁面に相当する。

与えられた条件のもとで、個々の機器を1つずつ配置していくが、不具合が生じると機器の位置・寸法・形状など許された条件、範囲で変更を試みる。

具体的には、

- かごは床面積一定ならば幅寸法と奥行き寸法は変更してよい、ただし、ドア装置は固定しかつ操作盤の取り付けスペースは確保する。

- おもりは重量が一定であれば、縦、横、長さを変更してよい。
- 調速機は適切配置位置4ヵ所のうちいづれかに配置する。

などがある。

また、本システムの持つ対話機能は、システムがレイアウト処理を試行錯誤的に進めていく際、機器の配置がうまくいかないときの不具合情報やそれに対して何をどれぐらい変更して処理を進めるかといった情報を表示し、ユーザはそれを確認しながら処理を進める。

レイアウトが完了後、ユーザはそのレイアウト結果に決定するかあるいは次候補を求めるかをメニューにより指示する。

「次候補」を選択すると、制約条件を満たす他のレイアウト配置を求める処理に移る。また、「終了」を選択するとそのレイアウト結果の詳細な寸法を算出する（図3.1-10参照）。

（２）システムの開発者と開発手順

本システムの開発にあたっては、専門分野外の技術者がKEとなり、いちからエレベータの構造、機能、設計方法を学び、そのプロトタイプを開発した。知識獲得は、各種資料を参考にヒアリングを行った。開発期間は、1.5人年である。プロトタイプの評価後、実用システムの開発を専門分野の中堅設計者がKEとなり、現在自主的に行っている。

（３）システムの動作環境

本システムは、現在実用システムを開発中であり、デリバリ環境については省略する。

- | | |
|-----------|--|
| ①対象ハードウェア | 三菱電機 MELCOM PSI |
| ②OS | SIMPOS |
| ③処理方法 | コンパイラ |
| ④言語／ツール | 言語ESP (Extended Self-Contained Prolog)
ツールA c e k i t (機器選定システムに使用) |

（４）システムの規模

- | | |
|--------|--|
| ①ルール数 | 約350 |
| ②フレーム数 | 約40 (ESPのクラスをフレームの単位とし、レイアウトシステムの知識表現部を表記) |

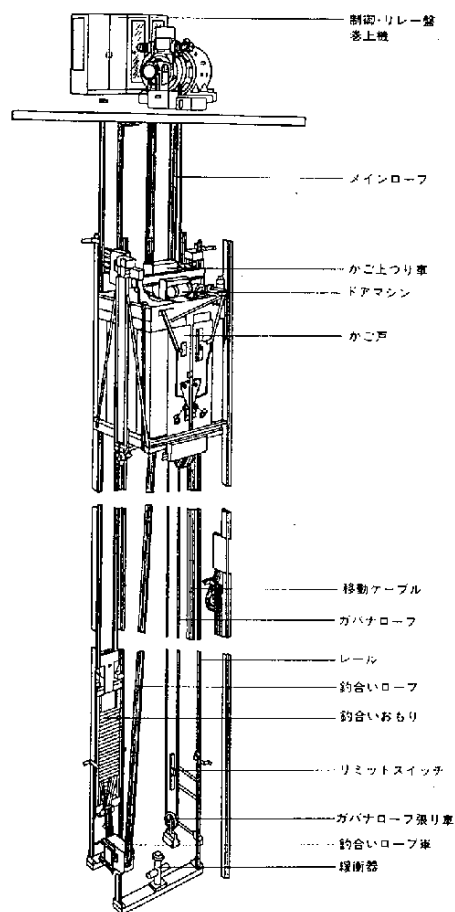


図3.1-1 エレベータシステムの概要図

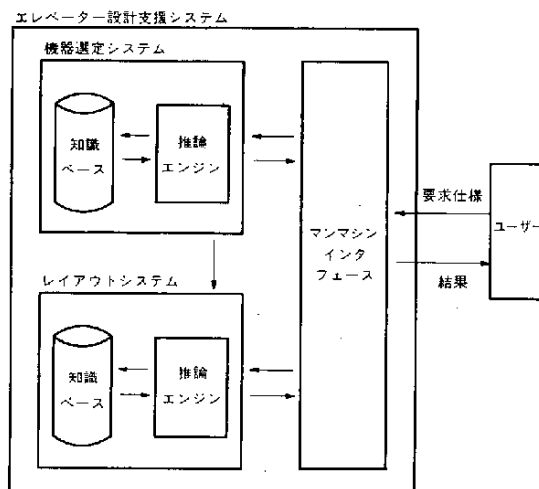


図 3.1-2 エレベーター設計支援システム構成図

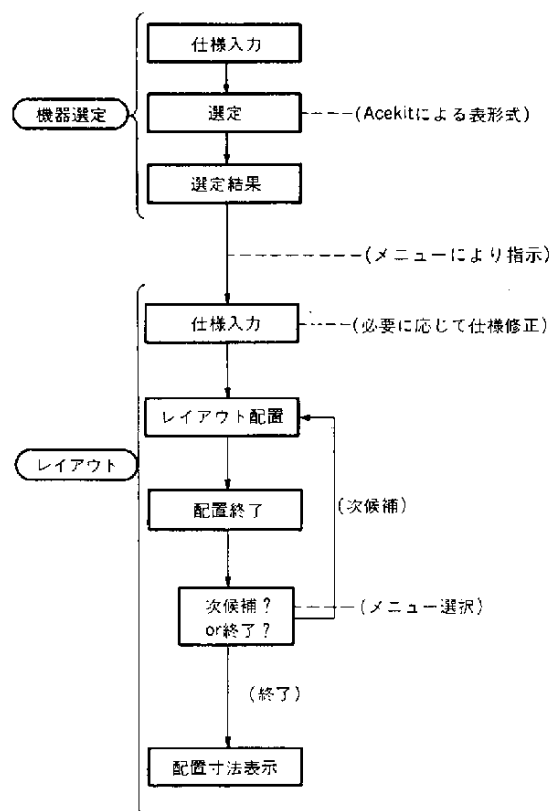


図 3.1-3 システムの流れ

機種がDで、CAPがEで
かご自重がFのとき、負
担荷重はBとする。

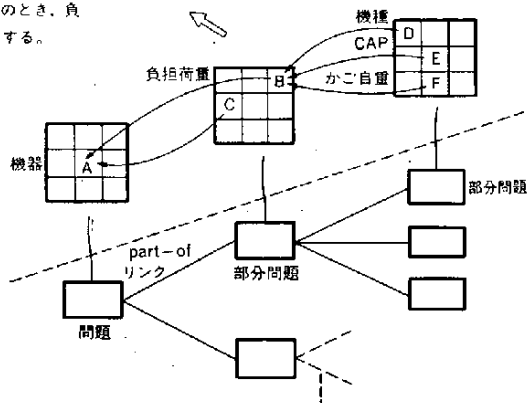
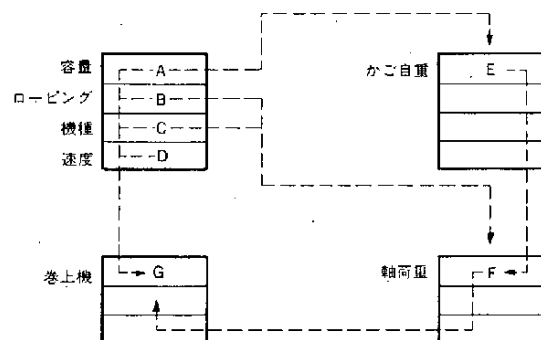


図3.1-4 Acekitにおける知識と問題の階層構造

向光	国内	CWT幅	280.0	mm	輪軸直	4359.8
用光	専用	WG	1200.0	mm	ブ方式	ハーフラップ
*透視	VM	CWTSF			ピンダ	1
*速度	120	m/min	*オモリ用GOV表	あり	白塗	817.7
制御方式			カゴ用非常止め制御形式		リ重	1725.8
*CAP	1600	kg	オモリ用非常止め制御形式		吊重	375.0
ローピング	1		ハーフラップ一本	1	CAP	1600
ラップ方式	ハーフラップ		1階ケーブル最重	4.7	kg/m	0.0
巻上げケーブル最重	494.0	kg/m	*J1のケーブル	1250.0		0.0
昇降速度	9	m/min			巻重	216.2
*昇降行程	9	m				
構造						
カゴ用ブレーキシステム		*巻上機	EM-2400		つり車 (カゴ用)	
カゴ用アイドleshシステム		フラスロ	SG-308		つり車 (オモリ用)	
*ワイヤ方式		*シンベーム	H194*150*		非常止め (カゴ用)	GSB-500
トップクリアランス		ナメバ			非常止め (オモリ用)	GSC-500
第二種検査車		*GOV (カゴ用)	DG-540M		カイドシュー (カゴ用)	LURT53
SIDEEXIT		*GOV (オモリ用)	DGC-540		カイドシュー (オモリ用)	LUR763
*A径	250	*スレール (カゴ用)	18		巻上げロープ	12
*B径	250	*スレール (オモリ用)	13		巻上げロープ本数	6
HB	345	バックワ (カゴ用)	OBD-450		ドク装置	K3
JJ		*バックワ (カゴ用) 備数	1		つり車	CG-526
EE	60	バックワ (オモリ用)	OBD-450		巻上げロープ	12
ピンク深さ		*バックワ (オモリ用) 備数	1		つり車ロープ本数	5
		はり車 (カゴ用)	HG-326H		(吊具用)	
		はり車 (オモリ用)	HGC-324		(カゴ用給油)	RL-B3
		カゴ付	KWM		(カゴ付給油)	
		コフ付	KWM			
		*オモリ	920			

図 3.1-5 機器選定システムの画面例



〈ルール例〉

容量がA以下、ローピングがB、機種がC、速度がD以下のとき、

巻上機はGとする。ただし、軸荷重がF以下のかぎり適用可能とする。

図 3.1-6 選定ルール例

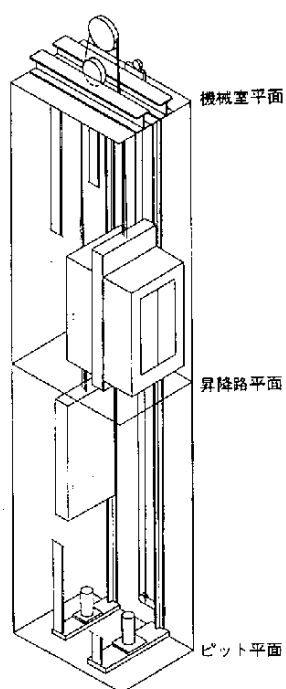
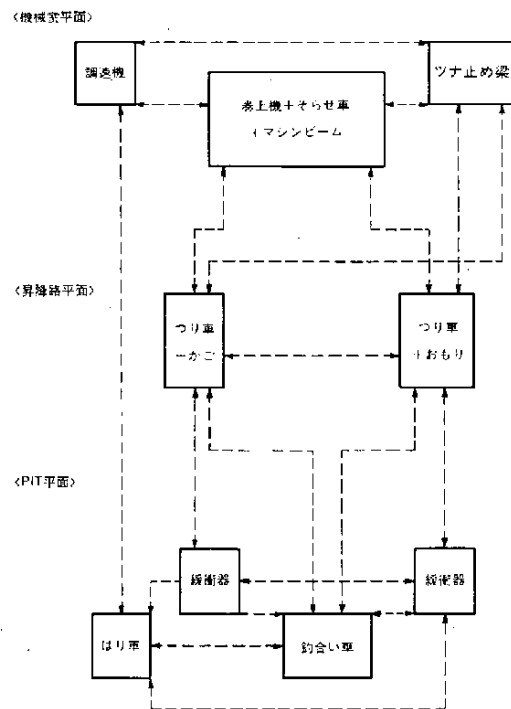


図 3.1-7 モデル図



注 ←→ は配置の際、相互に影響を及ぼす関係があることを示す

図3.1-8 各機器（群）の相互関係

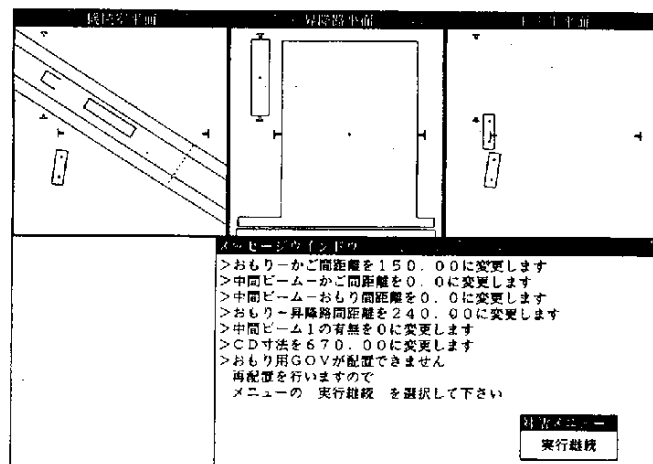


図3.1-9 対話の画面例

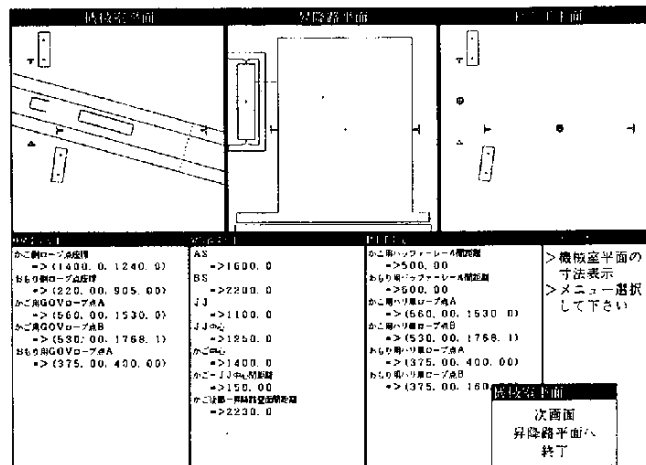


図 3.1-10 寸法算出の画面例

3.1.1.3 問題のタイプとシステムの特性

(1) 問題のタイプ

問題のタイプは、合成型問題（選定・レイアウト）である。

(2) システムの特性

本システムは、設計型システムとして位置づけられる。

その特性を表3.1-1に示す。

3.1.1.4 客観的な評価の可能性について

本システムはエレベータ設計業務を行っている専門家の支援を行う目的であるため、当然その専門家による客観的な評価は可能である。ただし、その専門家のかかなり直感に近いかたちでの評価にとどまる。

本業務は正確な設計方法が確立されているわけではなく、専門家のカンによる部分が多い。これは専門家がいい加減にやっているというわけではなく、設計結果にある程度許容範囲があって、その範囲であればよしとされる。他の専門家が同じ要求仕様で設計するとその専門家の設計結果とは違った内容となる場合もある。このように本業務は設計結果に個人差が出る。そのため、正確な評価をしづらいし、専門家によっても評価が違ってくる。

具体的な評価方法は過去の物件のデータを入力して、正解がどの程度であるかを見極めるしかないと思われる。

3.1.1.5 今後の課題

本システムはプロトタイプの開発を終了している。その評価は専門家によって行われ、現在実用システムを開発中である。プロトタイプの評価は、本業務の行われている職場においてフィールドテス

トという形で行われた。そして、必要機能として

- ① ユーザーの要求を対話的に受け入れ、機器の配置順序やレイアウト操作の優先順位づけ、又は制約条件の固定が自在にできる。
- ② レイアウトから機器選定へのフィードバック（配置がどうしてもうまくいかないときは、機器選定をやりなおす）が可能である。
- ③ 保守・運用が可能な構造を持つ（機種の新規更新や新機種の対応）。

などの課題が明確化した。

現在、これらの機能を実現し、より柔軟で高度な支援を行うシステムの構築を進めている。

表3.1-1 システムの特性一覧表

コメン	実現性	関連性	設計システム
て配 えっ 加従 にに 要る に要 約成 す 左制 構置			らを決 え素を 与要様 が成仕 様構部 仕、内 求にの 要の素 の要素 力る成 出する と現各 力実つ 人をか のれ、 ムこせ テきわ スと合 シたみ 組定
のい 証悪 問題 検が に問 性が 特率 と効	○ ○ ○ △ × ○ ○	○ ○ ○ ○ ○ ○ ○	大性 が的 間本 空基 解が の成 造合 構の ムよ テに ス析 シ解 1. 2. 3. 4. 5. 6. 7.
	○ △ ○ ○ △ △ ×	◎ ○ ◎ ○ ○ ○ ◎	表現 の現 連表 関的 の層 の生 の成 の動 のミ の解 との 素の 要問 成計 構設 1. 2. 3. 4. 5. 6. 7.
	○ ○	○ ○	戦略 化戦 精密 精略 ン戦 ウ化 グ小 ダ機 プ最 ト拘 1. 2. 3. 4. 5. 6. 7.

3.1.2 評価結果

(1) 知識ベースの妥当性評価

① 経験的知識の評価

本業務は多量の資料に加えて多くの経験的知識を使って設計業務を行っている。経験的知識とは、機器選定では要求仕様が厳しい場合の選定ルールやレイアウトの配置ルールなどである。よって、システムが動くための最低の知識は獲得されている。しかし、最適解を求めるに至る十分な知識は得ていない。

② 教科書的知識の評価

教科書という用語と理路整然と大切な事だけが記載されているイメージがあるが、本業務で使われている膨大な資料は設計業務の幹となる知識や取るに足らない知識が混同され、しかも断片的に記載されているため、その中から、必要となる知識を抽出することが困難であった。機器選定のための数値的知識、レイアウトのための機器形状の知識は十分に獲得されているがそれらを使う知識が十分といえない。

③ 知識ベース全体の整合性評価

機器選定システムは複数のプロダクションルールが同時に動くが、その整合性をチェックする機能をもっていない。それゆえ、知識ベースは十分な整合性をもっているとは言えない。

④ 経験的知識を正当化する方法とその確認

専門家による検証しか方法としてはなさそうなので、その評価方法は十分であったと思われる。

⑤ 教科書的知識を正当化する方法とその確認

専門家（及び、KE）による検証しか方法としてはなさそうなので、その評価方法は十分であったと思われる。ただし、方法としては十分であっても、その作業を十分行ったというわけではない。

⑥ 知識ベース全体の整合性を確認する方法

事例の検証しかなさそうなので、その評価方法は十分であったと思われる。ただし、方法としては十分であっても、その作業を十分行ったというわけではない。

⑦ 知識表現の選択と適切さの確認

機器選定システムについては、前述したとおり、選定の規則はほとんどプロダクションルールで記述できる。しかし、選定を行う際の仕様項目、選定基準値（軸荷重、安全率など）または選定項目が多様で、かつそれらが有機的に関係づけられるという理由から、ツールとして三菱電機の知的スプレッドシート“Acckit”を用いた。これはかなり業務の形に適しており適切であったとおもわれる。

レイアウトシステムについては、前述したとおり、PSIのAI記述用言語ESPのオブジェクト指向パラダイムを知識表現形式として採用し、機器の属性（形状、寸法、機能、適切配置位置など）をその中で記述している。レイアウト問題についてのツールは世の中に見当たらなかったもので、この方法で記述したことは適切であったと思われる。

⑧ 知識獲得方法の選択と適切さの確認

KEとして、エレベータの専門知識のない計算機技術者が、専門家よりのヒアリングを行ったり、資料を参照するなどして知識獲得をかなり一方的に行った。これは、まだ実用の見込みのないプロトタイプの製作ということで日常業務が多忙である専門家の十分な協力は期待できなかった。そういう意味では、専門家のより積極的な協力やあるいは専門家がKEとなって自主的に行ったほうがより適切であったであろう。

（２）問題解決・推論方式の妥当性評価

① 推論処理の妥当性評価

推論処理の方針は専門家の経験的な知識に依存しそれに近い形で実現したので、論理的に正しいかというよりどれぐらい専門家のやりかたに近いかということが問題になる。

② 推論時間の妥当性評価

推論時間は実際に専門家が費やす時間の数百倍の速さなので問題はない。妥当である。

③ 推論手法の選択と適切さの確認

機器選定システムはプロダクションルールを用いているが、前向き推論のみなので十分ではない。これに加えて、後向き推論、両方向推論、あるいは曖昧性を扱うファジィ推論などが付加されると十分であるとおもわれる。

レイアウトシステムはユニフィケーション、バックトラックを用いて縦型

探索を行っているが、枝刈りを行う必要がある。

④ 不確実情報に対する処理技法の適切さの確認

不確実情報に対する処理技法は考えていない。

⑤ 学習機能の適切さの確認

学習機能としてあえてあげるならば、ツールのもつ“履歴機能”である。

これは過去のデータを履歴として格納し、それらを一部の属性の属性値の指定により検索し再利用するものである。この機能そのものは十分つかえるが、これ以外にも必要とされる機能は課題としてあるが実現されていない。

(3) 結果の妥当性評価

① 結果の妥当性の評価

専門家による判断、あるいは事例などによる確認を行った。

② 解の評価方法の適切さの確認

確認は行っていない。

(4) 要求仕様に対する達成度評価

① 要求仕様に対する達成度評価

専門家による判断、あるいは事例などによる確認を行った。

② 要求仕様設定手段の適切さの確認

行っていない。

(5) 知識ベースの保守性評価

① 変更箇所発見の容易性評価

ドキュメントや資料との比較を行うので容易ではない。

② 知識ベースの理解性評価

知識ベースのデータ記述部は表に近い形で記述されているのでわかりやすい。

③ 品質安定性の確認方法の評価

専門家（あるいは中堅設計者など）が保守要員にならなければ品質の安定性の確保は困難である。

④ 保守マニュアルの整備状況の確認

保守マニュアルはまだ製作されていない。

⑤ 保守用ツールの整備状況の確認

保守用ツールはまだ製作されていない。

(6) マンマシンインタフェースの妥当性評価

① 利用時インタフェースの評価

対話ウインドウを用いて機器配置位置の変更などをメッセージでユーザに提示し、ユーザはメニューで指示する。また、レイアウト画面が示されるのでレイアウト状況が把握できる。

② 保守時インタフェースの良さの確認

保守時のインタフェースはよくない。

(7) システムの総合評価

① 利用者の満足度評価

利用者はまだいない。

② 統合性の評価

他システムとの連携はない。

③ 開発方法の適切さの確認

プロトタイプング手法を適用したのは妥当であった。ただし、実用システム開発に当たっては知識ベースの整合性や実用性を高める目的で、もう一度知識の再整理を行い知識ベースの再構築を行っている。

④ ツール・言語選択の適切さの確認

機器選定システムで用いたツール A C E K I T は表形式のインタフェースを有しているので混沌とした知識を表というわかりやすい形で表現できた。レイアウトシステムは E S P を用いた。これは制約条件を満たす多数の候補解を考える場合有効であった。

3.2 電子装置の実装設計支援システム

3.2.1 プロフィール

3.2.1.1 システムの目的と期待効果

本システムは、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わっており、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。

本システムは熟練した専門家の設計作業を支援するもので、設計完了から具体的な初期の試作装置が完成するまでの時間が長く、設計結果の大きな変更は許容できないため、最終的な結果については専門家による判断を行う。

3.2.1.2 システムの概要

(1) システムの機能と構造

① システムの構造

システム全体は、シミュレータ、データベース、インタラクティブインタフェースなども含んでいる。

知識処理技術を適用した部分では、未定義、不確実な入力にをもとに拘束条件を満たす準最適解（実装構成）を複数得る機能を実現した。

② システムの機能

問題解決知識は専門家からインタビューで得た。

また、設計対象に関する情報は、設計結果のドキュメント（問題解決過程そのものの記述はない）から得た。例えば、装置の実装構造としてIC／プリント板／モジュール／装置／信頼性等である。

知識表現は、3次元構造物を属性モデルで表現（フレーム）し、構造物間の接続関係をネットワークで表現（フレーム）し、制約条件は連続量として取扱った。

推論方式としては、未定義入力項目を許容し、制約条件を最小にしておき、前向き推

論する方式を中心とした。

(2) システムの開発者と開発手順

システムの開発者：K E

システムの開発の経緯：各バージョンでプロトタイピング開発手法を適用

Ver(-1)：従来の手続き言語を利用したシミュレータ開発〔陣野 77〕

Ver 0 : 未定義情報の取扱いの手法を検討

Ver 1 : 演繹推論適用性の確認（知識ベースと推論機構の分離）

Ver 2 : 知識の投入とシステム評価

Ver 3 : E S 構築ツールを適用して再記述と評価

(主に記述性、処理時間、占有メモリ量等のツール依存部)

Ver 4 : 実用システム化（現在開発中）

(他のシミュレータやデータベースシステムとの結合、専門家インタフェース部・利用者インタフェース部のグラフィクス化など)

(3) システムの動作環境

センタにホスト計算機を配置しデータベースを格納し、推論と利用者のマンマシンインタフェース機能はワークステーションに機能分散して、会話性を重視した。

(4) システムの規模

現在、実用システムを開発中であり、最終規模は未定である。

3.2.1.3 問題のタイプとシステムの特性

本システムは、設計型エキスパートシステムであり、表3.2-1の特性を持つ。

表3.2-1 システムの特性一覧表

	設計型システム	関連性	実現性	コメント
定義	システムの入力と出力の要求を、仕様と構成要素の組み合わせ、決定する。システムの実現は、内部仕様の決定による。	○	△	設計開始時点では入力に未定義な項目がある。設計段階に順次入力（時系列で具体化）
問題の特徴	1. システム構造の解空間が大 2. 解析による最適化の必要性 3. 構成要素の完全性の確保 4. 検証の形式化と支援 5. 対話形式の応じ 6. 設計段階の決定支援	○ × ○ × ○ ○	○ × △ × △ ○	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 最適化は定性的にしかできない 検証はフィックスの活用 グラフィックスの活用 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 2. 設計問題の階層的表現 3. 代替案の自動生成 4. 部分システムの評価 5. 上位レベルへの知的後戻り 6. 設計事例の検索と利用 7. 並列的問題解決	○ ○ ○ ○ ○ ○ ×	△ ○ ○ ○ △ △ ×	属性モデルで3次元構造を表現 接続形態・包含・上下関係等 解空間を連続量による領域として表現 各実装段階とに準最適化（一般的絞込み手法未） 文書レベルで過去の設計事例を検索
問題解決機能	1. トップダウン精密化戦略 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. 類推による設計事例の利用 7. 協調型推論	○ ○ ○ × ○ ○ ○	△ △ △ × × × ×	入力に未定義な項目（設計完時点でも可） 制約条件の数は最小にしておく 最適化 将来必要 将来必要 将来必要

3.2.1.4 客観的な評価の可能性

最終的には装置そのものはハードウェアとして実現できるため、各種の物理的特性（電氣的、熱的）、コスト、開発期間などで評価可能である。

3.2.1.5 今後の課題

設計期間の短縮、専門家でも想定していなかったより良い構成が得られた、など開発の効果はあるものの、最終的には人間の持つ視覚的なバランス感覚など、計算機ではまだモデル化できない機能があり今後の課題である。

3.2.2 評価結果

(1) 知識ベースの妥当性評価

① 知識ベースの正当性評価

専門家によって設計された既設計事例と同一の入出力条件で走行させその結果確認するとともに、数例については内部の詳細も追跡した。

② 経験的知識を正当化する方法とその確認

ヒューリスティック探索の一部を網羅探索に記述変更してシミュレーションし、解のずれの値を確認した。

③ 知識表現の選択と適切さの確認

知識の新規追加性を、新しい実装構造の追加で検証した。また、これに伴う処理時間、占有メモリ量の増加の程度を確認した。

(2) 問題解決・推論方式の妥当性評価

① 推論処理の妥当性評価

専門家が新たに実施する設計業務と並行に同一の入出力条件を設定して走行確認した。

② 推論時間の妥当性評価

未定義項目数と処理時間の増加の関係を「処理時間／未定義項目数」で確認した。

③ 不確実情報に対する処理技法の適切さの確認

未定義項目を順次減少させその影響度をデフォルト値での解の総数、処理時間減少率などで評価した。

(3) 結果の妥当性評価

① 解の正解率、網羅性、最適性評価

知識獲得した以外の複数専門家（評価専門家2名）により机上確認、またES走行で、ともに新規装置設計条件で確認した。

(4) 要求仕様に対する達成度評価

① 要求仕様内容に対する評価

従来のソフトウェア開発手法に準拠した外部仕様、機能仕様、インタフェース仕様、状態遷移で規定した。

② 要求仕様に対する達成度評価

従来のソフトウェア開発手法に準拠した。また、プロトタイプシステム開発段階では特定の項目のみ指定して評価した。

(5) 知識ベースの保守性評価

- ①変更箇所発見の容易性、知識ベースの理解性評価、保守マニュアルの整備状況の確認
新規装置の設計時(2回)にESを利用する運用で確認した。

(6) マンマシンインタフェースの妥当性評価

①利用時のインタフェース評価

限定ユーザでの試行運用とアンケート実施で評価した。評価基準は、5段階評価で1、2がないこととした。

②保守時のインタフェース評価

知識ベースそのものを操作することがES利用であるため知識ベースの保守性と同じ条件で評価した。

(7) システムの総合評価

①テーマ選定の妥当性評価

社内共通の知識処理適用性確認チェックリストで確認した。

②利用者の満足度評価

専門家インタフェースを含め、限定ユーザでの試行運用し、アンケート調査した。5段階評価で1、2がないことを基準とした。

③開発方法の適切さの確認

開発工数・期間・追加変更性を従来言語版と比較した。

④システム開発の生産性評価、システムのコストパフォーマンス評価

開発工数・期間を従来言語版と比較した。

【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 計算機ハードウェア構成設計支援エキスパートシステム

3.3.1 プロフィール

3.3.1.1 システムの目的と期待効果

計算機ハードウェアの構成設計とは、顧客の要求に合わせて中央処理装置、端末、記憶装置などの機器の組み合わせを設計する業務である。計算機メーカーは、この業務に膨大な人員及び経費を投入している。そこでSE（システムエンジニア）を支援し、業務の効率と信頼性を向上させることを目的として、本システムの開発に着手した[1]、[2]。

本システムは、機器の仕様、設計の手順、将来の拡張や営業面を考慮した設計ノウハウなど多種多様な知識を必要とするが、これらの全てを最初から知識の収集をすることは困難である。従って、実稼働後も知識の追加が必要である。また新機器の発売、既存機器の仕様変更や発売停止のため、関連する知識の追加、変更が頻繁である。このように本システムの開発に当たっては、保守を容易にすることが重要課題であった。

そこで、保守性の優れたシステム開発技術として、宣言的に記述された知識を用いて問題を解く知識処理技術を取り上げ、これに基づいて設計支援エキスパートシステムを開発した。

3.3.1.2 システムの概要

(1) システムの機能と構造

本システムの主な機能は以下の通りである。

① 設計作業の誘導

設計手順に関する知識の基づき、メニューなどによりSEの設計作業を誘導する。

② オプション機器の追加、削除

オプション機器の必要性を自動的に判断して付加したり、逆に不要となったオプション機器を削除する。

③ 上級SEのノウハウによるアドバイス

設計状況に応じてマニュアルにはない熟練者のノウハウに属するアドバイスをを行う。

④ 機器間の接続の正当性チェック

⑤ 知識保守の支援

知識の保守を容易化するために知識入力誘導、入力値のチェックを行う。

本システムの構成を図3.3-1に示す。この中で、知識ベースは設計手順、機器仕様、機器間の接続関係、上級SEのノウハウを格納する。推論エンジンは、これらを用いて上記の①～④の機能を遂行する。知識利用アプリケーションは、推論エンジンを制御して、利用者の要求に応じた推論を行わせる。知識保守アプリケーションは、上記⑤の機能を実現する。構成図ファイルは、設計結果あるいは中間結果のハードウェア構成図を記憶する。

知識のうち機器仕様と機器間の接続関係は仕様書から抽出した。上級SEのノウハウと設計手順は上級SEからのヒアリングより獲得した。

本システムは、ソフトウェア構成設計ES、ストレージ見積りES、システム生成パラメータ作成ESと連動し、計算機システム設計業務の全体を支援している。

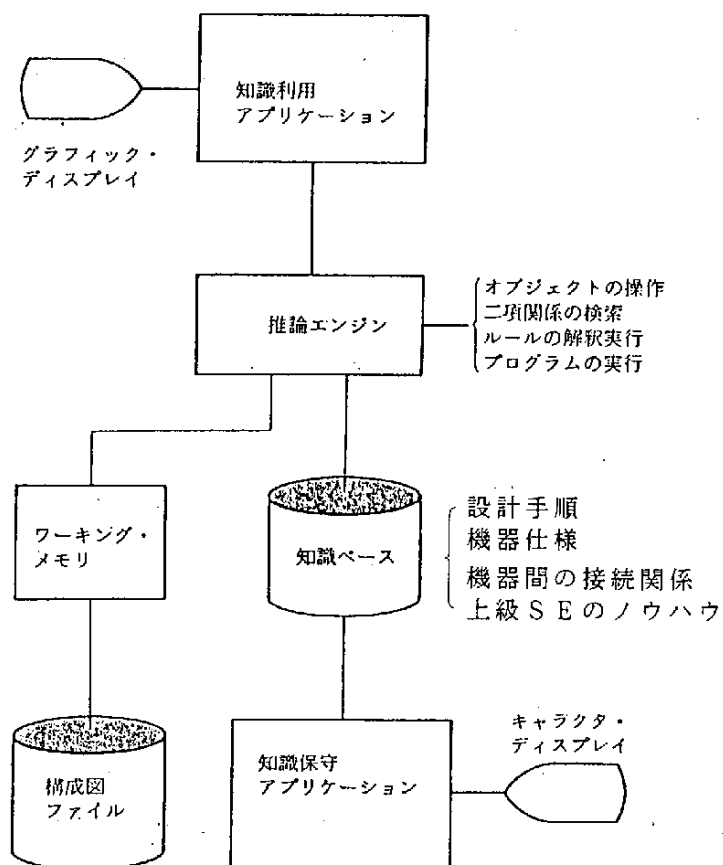


図3.3-1 システムの構成

(2) システムの開発者と開発手順

図3.3-2に示すように、一旦プロトタイプを開発し、それを評価、改良して実用化した。プロトタイプは研究部門が中心となって開発した。実用化は向上が中心となり、複数の専門家の協力を得て行った。

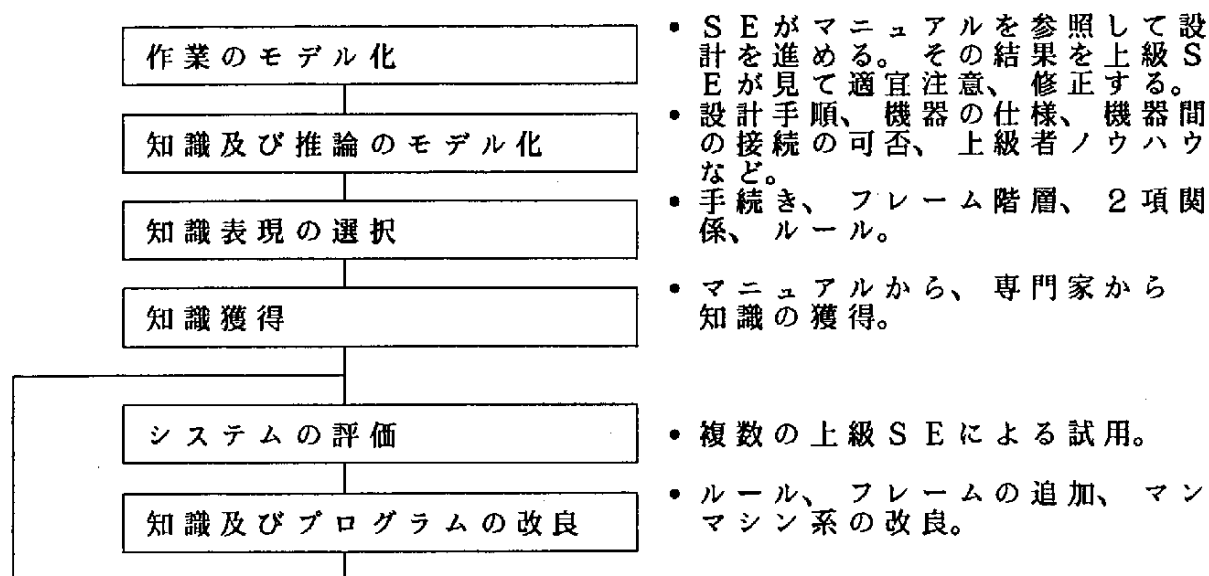


図3.3-2 システムの開発手順

(3) システムの動作環境

プロトタイプシステムは大型機 H I T A C / M シリーズ上の Prolog、実用システムはワークステーション 2050 上の Prolog で稼働する。

(4) システムの規模

表3.3-1にシステム（プロトタイプ）の規模を示す。

表3.3-1 システムの規模（プロトタイプ）

ルール数	約 50
フレーム数	約 350
二項関係	約 200
総ライン数	約 9,000

3.3.1.3 問題のタイプとシステムの特徴

本システムは設計型システムであり、S E の設計業務を対話的に支援する。また設計結果のうち機器間の接続について正当性を検証する（表3.3-2を参照）。

3.3.1.4 客観的な評価の可能性について

複数の専門家に評価してもらうことにより、客観的な評価が可能である。

表3.3-2 システムの特性一覧表

	設計型システム	関連性	実現性	コメント
定義	システムの入力と出力の関係を定義し、その実現に必要な要素を決定する。	○	○	
問題の特徴	1. システムの構造を定義する。 2. システムの動作を定義する。 3. システムの性能を定義する。 4. システムの信頼性を定義する。 5. システムの拡張性を定義する。 6. システムの保守性を定義する。 7. システムのセキュリティを定義する。	○ × ○ ○ ○ ○ ○	○ ○ ○ ○ ○ ○ ○	
基本タスク	1. システムの構造を定義する。 2. システムの動作を定義する。 3. システムの性能を定義する。 4. システムの信頼性を定義する。 5. システムの拡張性を定義する。 6. システムの保守性を定義する。 7. システムのセキュリティを定義する。	○ × ○ ○ × ○ ×	○ ○ ○ ○ ○ ○ ○	
問題解決機能	1. システムの構造を定義する。 2. システムの動作を定義する。 3. システムの性能を定義する。 4. システムの信頼性を定義する。 5. システムの拡張性を定義する。 6. システムの保守性を定義する。 7. システムのセキュリティを定義する。	× × ○ ○ ○ × ×	○ ○ ○ ○ ○ ○ ○	

3.3.2 評価結果

(1) 知識ベースの妥当性評価

① 知識ベースの正当性評価

設計手順、上級 S E のノウハウ知識は、複数の上級 S E (知識提供者及び第 3 者) により確認した。機器の使用、機器間の接続の可否に関する知識は仕様書により確認した。

② 知識ベースの完備性評価

実際のハードウェア構成事例を用いて評価した。

③ 知識ベースの一貫性評価

知識を充分モジュール分割しているので、一貫性は問題にならなかった。

④ 経験的知識を正当化する方法とその確認

本システムにおける経験的知識とは、上級 S E のノウハウを表すルールに相当する。複数の上級 S E により確認した。

⑤ 知識表現の選択と適切さの確認

知識表現の選択は以下の通りである。

全体の制御・・・知識ベースとは独立した手続き (K E が専門家なしで保守できるように)

設計手順・・・フレームのデモン (例: C P U の選択に続いてコンソールディスプレイを選択することを C P U フレームのデモンに記述)

機器の仕様・・・i s - a 階層を持つフレーム

機器間の接続の可否・・・二項関係

上級 S E のノウハウ・・・ルール

また、知識の入力、保守作業を通じて表現の適切さを確認した。

⑥ 知識獲得方法の選択と適切さの確認

知識獲得の手順は以下の通りである。

イ. 作業モデルの作成 (モデル = S E がマニュアルを参照して構成を設計する。その作業を上級 S E が見て、適宜注意、修正を行う)

ロ. 知識モデルの作成 (設計手順、機器の仕様、機器間の接続の可否、上級者のノウハウなど)

ハ. 知識表現の選択 (上記⑤の知識表現を選択)

ニ. 知識の抽出 (各種仕様書から抽出、上級 S E からのヒアリング)

これは、標準的な知識獲得方法であり、適切と考える。

(2) 問題解決・推論方式の妥当性評価

① 推論処理の妥当性評価

推論内容は、ルールの前向き推論、フレームのデモン、is-a階層の継承である。論理的な正しさは実際の例を用い複数の専門家により確認した。

② 推論時間の妥当性評価

対話システムであるため、一度に多くの推論を行うことはない。対話的な使用に耐えている。

③ 推論手法の選択と適切さの確認

作業モデルに基づいて知識表現を決定することにより、推論手法は自然の定まった。

④ 不確実な情報処理に対する処理技法の適切さの確認

本問題では、不確実な情報を扱う必要がない。

⑤ 学習機能の適切さの確認

学習機能は持っていない。

(3) 結果の妥当性評価

① 解の正解率の評価

解の正しさは上級 S E が判断した。

② 解の網羅性評価

複数の解が可能である場合（例えば、1つのメイン機器に対して複数のオプション機器が利用可能である場合）全ての解を表示し利用者に選択してもらう。

③ 解の最適性評価

解が一意に決まる場合が多く、複数可能である場合には利用者に選択してもらう。利用者が S E であるので選択に困ることは殆どない。

④ 解の評価方法の適切さの確認

知識の提供者及び第3者を含む複数上級 S E が評価した。

(4) 要求仕様に対する達成度評価

① 要求仕様内容に対する評価

開発の初期段階ではラフな内容であり、プロトタイプの開発など工程が進むに連れて仕様が明確化された。

② 要求仕様に対する達成度評価

最終的な要求仕様は達成している。

③ 要求仕様設定手段の評価

専門家自身が S E なので、専門家に要求仕様を記述してもらった。

(5) 知識ベースの保守性評価

① 変更箇所発見の容易性評価

対話の誘導は、手続き及びフレームのデモンが、またアドバイスや修正はルールが機能の種類と知識の種類が対応している。さらに同じ種類の知識が中央処理装置の構成設計、ディスク駆動装置の構成設計など設計対象によってモジュール分けされている。この様に外部の機能と内部の知識が対応しているので、変更箇所の発見は容易である。

② 知識ベースの理解性評価

保守の担当はルール、機器の仕様、機器間の接続の可否は専門家に、手続き及びデモンは K E に担当を明確にして、知識のモジュール分けを行った。また実際上も想定した担当者により保守が行われている。

また、(1)の⑥で述べた明確なモデルに基づいて、同⑤のような知識表現を一貫して用い、前述したように外部機能と内部機能の知識の対応を図ったことが知識ベースの理解性を増している。

③ 変更箇所の局所性評価

保守担当者を明確に想定した上で、知識をモジュール分けしている。設計手順など変更の少ない知識と、機器の仕様など変更の多い知識を分けてるため変更箇所の局所性がよい。

④ 知識ベースの統一性評価

明確なモデルに基づいて知識を選択し、その知識表現を一貫して用いている。

⑤ 保守マニュアルの整備状況

保守マニュアルは整備されている。

⑥ 保守用ツールの整備状況

専門家による保守を円滑化するために、入力の誘導、入力値のチェックを行う分野別構造エディタを設けた。

(6) マンマシンインタフェースの妥当性評価

① 利用時のインタフェース

マルチウィンドウ、メニュー、アイコンを多用し原則としてテキスト入力を不要化した。

② 保守時のインタフェース

保守用ツールの整備状況で述べたように、専門家による保守を円滑化するための分野別構造エディタを設けた。

(7) システムの総合評価

① テーマ選定の妥当性評価

計算機メーカーでは、ハードウェア構成設計業務に膨大な人員及び経費が投入されているので、支援システムに対するニーズは大きかった。この分野では新しい機器の発売、既存機器の仕様変更や販売停止などに伴う知識の変更が頻繁であるため、保守に適した宣言的な知識表現が望ましかった。また記号処理により解決できる問題である。以上から本問題は知識処理技術の適用対象として妥当であった。

② 利用者の満足度評価

従来38時間程度要していた作業が、6時間に短縮された。(但し本システム以外にソフトウェア構成設計ESを併用した場合)本システムは数年前から実業務で使われている。

③ 統合性の評価

本システム(ハードウェア構成設計ES)は、ソフトウェア構成設計ES、ストレージ見積りES、システム生成パラメータESと連動し、計算機構成設計全体を支援している。

④ 成長可能性の評価

ハードウェア構成設計の分野では、新機種が頻繁に発売されるが、本システムはこれらの仕様及び接続情報を追加しながら数年間使われている。

⑤ 開発/保守用ドキュメントの評価

ドキュメントは整備されている。

⑥ 開発方法の適切さの確認

一旦プロトタイプシステムを開発しこれを評価した後、実用システムを開発した。プロトタイプに開発に当たっては、対象世界のモデリング、知識の枠組の設定、知識の抽出という手順を遂行した。基本方針としては、保守担

当者を明確に想定した上で、担当者が単独で保守できるようにシステム及びツールを開発した。

⑦ ツール・言語選択の適切さの確認

開発言語はPrologである。

⑧ システム開発の生産性評価

知識処理技術を用いたことにより、保守性の高いシステムとなった。

⑨ システムのコストパフォーマンス評価

従来38時間程度要していた作業が6時間に短縮されたこと、非常に頻繁に利用されていることからコストパフォーマンスは良好である。

⑩ セキュリティ対策の評価

データはハードディスクに記憶するほか、FDにバックアップをとっている。

【参考文献】

- [1] 飯塚、辻、山中：コンピュータのシステム構成設計を支援するエキスパートシステム、日経エレクトロニクス、No.415、pp.163-183、(1987)。
- [2] 浜崎、山中、竹山、佐藤：EDP構成設計支援エキスパートシステム、日立評論、Vol.70、No.11、pp.123-130、(1988)。

3.4 トッパー通油計画立案支援システム

3.4.1 プロフィール

3.4.1.1 システムの目的と期待効果

(1) 開発の動機

石油精製の生産計画におけるコンピュータ利用はLP(線形計画法)の利用技術が確立しており、長中期計画から月次計画までLPモデルを用いて計画を立案している。しかし具体的な日々の計画立案では、原油の受入やチャージタンクの切替等の時变的要素を考慮しなければならず、LPでは対処しきれていない。多段階LPによる日程計画立案の取り組みもあるが、入力データ・計算量の増大や出力結果が实际的ではない面があり実用化には至っていない。そこで、知識工学的手法を取り入れた日程計画立案システムのひとつとしてトッパー通油計画立案支援システムの構築を行なうことにした。

(2) システムの目的

トッパーとは石油プラントの最上流に位置し、原油を大気圧に近い状態で蒸留する装置である。そして、トッパー通油計画は、①原油タンクの初在庫・原油比率、②原油受入計画、③原油出荷計画、④原油処理量等の情報をもとに、種々の制約、すなわち、①原油タンクの容量制限、②計画の連続性、③静置期間、④タンクの切替制限、⑤チャージ条件、⑥予備チャージタンクの確保等を満足するように原油タンクのタンク繰りのスケジューリングを行なうことである。この計画の立案は最適性・健全性を要求され、計画担当者は複雑な意思決定を短時間で行なわなければならない。したがって、利用可能な情報と計画担当者のノウハウを統合した計画立案支援システムによって迅速化・省力化はもちろん最適な計画を追求する上で非常に役立つ。

(3) システムの責任レベル

システムが立案した計画を採用するかは計画者が判断する。つまり、必要があれば計画者が随時、修正を行なう。このことから、システムの責任レベルとしては Light Duty であるといえる。

(4) 利用対象者

システムの利用者は計画担当者、すなわち知識を提供した人(専門家)である。

(5) 期待効果

数式モデルで記述することができないため最適解を求めることは困難である。計画者がもつ経験的知識を整理することにより最適解(最適計画)がどのようなものかわかってくるのではという期待感がある。

(6) 既存技術の検討

多段階線形計画法によるアプローチ(前述)。

3.4.1.2 システムの概要

(1) システムの機能と構造

a. 構成要素と機能

システムのもつ構成要素を次に示す。システムの機能は拘束条件を満たすように、これらの構成要素を組み合わせることである。

- ① トッパー処理量選択……トッパーの処理量はどれだけか。
- ② チャージタンク選択……何日から何日までどの原油タンクをトッパーにチャージするか。
- ③ 受入タンク選択……原油の受入時にどの油種をどのタンクにどれだけ受け入れるか。
- ④ タンクシフト選択……何日にどのタンクからどのタンクへどれだけシフトを行なうか。
- ⑤ 出荷タンク選択……何日にどのタンクからどれだけ出荷するか。

b. 状態の定義

推論方法は計画ノードによる生成検査法を用いており、計画ノードを次のように定義している。

- ① 計画ノードはオペレーション部とチェック部から成り立つ。
- ② オペレーション部は計画立案の意思決定に係わるものであり、前述した構成要素のうちのいずれかの情報である。
- ③ チェック部はチェックリスト生成処理により求まるチェックリストの情報である。なお、チェックリスト生成処理はこの段階で制約を満足していないチェックリストを生成する処理である。

c. 推論方法

- ① 計画ノードの展開は、深さ優先で探索を進める。
- ② オペレーション部はノード間に継承関係がある。
- ③ 計画ノードにチェック部が存在しない場合は、そのノードは待機状態であ

ることを示す。

- ④計画ノードのチェック部が空リストの場合は、そこまでの探索パスが実行可能解であることを示す。

(2) システムの開発者と開発手順

システムの開発はK E、専門家、各1名で行なっている。システムの見通しを良くするため、3つのプロセスに分割し、各プロセス毎に開発することにした。I/Oプロセスは、マルチウインドウ/グラフィックス等の機能を用いて、操作のしやすいシステムを実現することを担う。データ処理プロセスはシステムで必要なデータのセットと在庫計算、性状推定計算等の計算処理を行なうプログラム群である。推論プロセスは知識ベースに格納されている知識を用いて推論を行ない、実行可能な計画を立案する機能を担う。

開発手順は、まず要求仕様が明確であるデータ処理プロセスを構築し、次にマンマシンインタフェースを充実させるため、I/Oプロセスを構築している。この段階で対話形式により計画を立案できることが可能である。推論プロセスは計画者が実際に対話形式で計画立案を行ない、この立案過程の履歴をシステムで保持し、それをK Eが解析しながら、計画者のヒアリングを行ない、知識を抽出し、ルール化することにより構築している。

(3) システムの動作環境

- | | |
|-----------|-----------------------------------|
| ①対象ハードウェア | Symbolics-3620 |
| ②使用メモリ量 | メインメモリ 8MB ディスク容量 190MB |
| ③OS | Lisp(Genera7.2) |
| ④処理方式 | コンパイラ |
| ⑤言語/ツール | Symbolics-Common-Lisp、Flavors、ART |

(4) システムの規模

- | | |
|---------|-------------|
| ①ルール数 | 30 |
| ②フレーム数 | フレームは用いていない |
| ③Lisp関数 | 300 |
| ④ライン数 | 5,000 |

3.4.1.3 問題のタイプとシステムの特

(1) 問題のタイプ

問題のタイプは、拘束条件を満たす最適解の構成要素を決定する計画問題である。そして、チャージタンクの切替回数やタンクシフトの回数を増やすことによって、可能解は無限に広がる。前述したように、最適解は未知であるので、生成検査法によって準最適解を探索するアプローチを用いている。

(2) システムの特性

表3.4-1に、計画型システムの特性に照らし合わせ、各々の項目の関連性と実現性を示す。

表3.4-1 システムの特性一覧表

計画型システム	関連性	実現性	コメント
定義	○	○	みへ 組ト を一 素チャ 要素 要チ 成ム 構イ のたる ムてて テセ当 スわり シ合割
問題の特徴	○ ○ ○ ○ ○ ○ ○	△ ○ × ○ × × ○	る 片 す 素 画 性 断 存等 計 全 判 に性 勢の 安が 限全 情つ と人 無安 油1 トは は、原 がス に 解性、 ド中 コの 能率 候、一 討油 終 可効 天ノ 検原 最
基本タスク	○ ○ ○ ○ ○ × ○	○ ○ × × × × ×	い 難 要 は 困 必 携 が 連 の け 化 確 合 る の 連 あ ム 徴 明 の 素 関 は テ 特 の 法 手 要 に 性 ス の 案 成 難 要 シ 画 価 立 構 複 他 計 評
問題解決機能	○ × ○ × × × ○ ×	○ × △ × × × × ×	い な い な い な い 確 題 羅 て 明 問 網 い の 分 の 用 り 順 部 解 は あ 手 な 適 S 性 定 立 最 M 要 決 準 T ー 必 一

3.4.1.4 客観的な評価の可能性について

トッパーで処理した原油のコスト面だけの評価であれば、月間トータルの最適化は、月間LP解で求められている油種別処理量との距離を測定することによって比較可能である。ただ、原油タンクには複数の油種が混合されており、LP解どおりの計画を立案することは困難である。また、LP解に近づけるためにチャージタンクの切替を頻繁に行なうと切替ロスが増加することになり良い計画とは言えなくなる。さらに、悪天候によるタンカーの遅延や原油・石油製品の需給バランスの変動等に対する適応性なども考慮するとなると、立案された計画に対して、どのように評価したらよいか定かではない。

3.4.1.5 今後の課題について

知識処理システムとして考えると現段階ではプロトタイプシステムである。つまり、拘束条件を満足する解を求めることはできるが、以下の項目についての問題解決機能が備わっていないため、実地的な計画を立案するには至っていない。

(1) 並行チャージの選択

並行チャージとは、複数のタンクから同時にチャージすることである。利点としては、切替回数やタンクシフトの回数を減少させることができること、性状の調整が容易に行なえること等が上げられる。反面、運転管理が複雑になるので、頻繁に行なうことはない。

(2) 原油受入時のロットの分割

原油をタンカーから受け入れる場合に、それを1基のタンクで受けるのが良いのか、2基で受けるのが良いのか、3基以上で受け入れるのが良いのか明確な基準はない。さらに複数のタンクで受け入れる場合に、どのように分割したら良いかの判断根拠の設定は困難である。

(3) 制約条件の緩和

制約条件には、静置期間の短縮や休祝日のタンク切替解除等、緩和の許されるものもある。緩和の許されるものについて、これらをどういう状態の時にどのように緩和したら良いか選択すること。

これらについては、評価項目でいうところの「経験的知識を正当化する方法とその確認」及び「解の評価方法の適切さの確認」が出来ていないことに起因していると考えられる。

3.4.2 評価結果

(1) 知識ベースの妥当性評価

現在の段階での知識ベースの規模は、30ルール程度であり、知識ベースの正当性、一貫性については問題ない。完備性については不十分であるが、欠如している部分は計画者が介入することにより対応している。

(2) 問題解決・推論方式の妥当性評価

推論方式は計画ノードによる生成検査法を用いている。その妥当性については、実際のデータを用いて、計画者が立案した計画と比較することによって確認している。

(3) 結果の妥当性評価

拘束条件を満足する解を求めることは出来るが、「解の評価方法の適切さの確認」が出来ていないため、解の最適性評価は行われていない。

(4) 要求仕様に対する達成度評価

前述してあるように、並行チャージの選択や原油受入時のロットの分割等の問題解決機能が備わっていないので、知識処理システムとしての達成度はまだまだであるといえる。

(5) 知識ベースの保守性評価

このシステムは現在開発途中であり、保守についてはまだ検討していない。ただし、知識ベースの理解性については、言葉による記述とルール表現とそのルールが適用された場合のノードの関連をセットにしてドキュメントとしてまとめ、わかりやすくしてある。

(6) マンマシンインタフェースの妥当性評価

ウィンドウ処理やグラフィックス処理を多用し、推論過程や推論結果をビジュアルにわかりやすくしている。システムの操作は、すべてマウスによるメニュー選択が出来、コマンド入力による誤操作をなくしている。また推論過程におけるユーザの介入を許し、その状態でのガントチャートの表示や制約を満たしていない項目や参照すべきデータの提示を行なっている。

(7) システムの総合評価

従来のシステムと比べて操作性が良く、対話形式により従来よりも迅速に計画立案が出来るので利用者の満足度は高い。

3.5 潤滑油精製プラントのスケジューリングエキスパートシステム

3.5.1 プロフィール

3.5.1.1 システムの目的と期待効果

(1) 開発の動機

石油精製工程は、概ね連続的なプロセスからなり、その生産計画作成には線型計画法(LP)に代表される数理手法が適用され、多大な効果を上げてきた。しかし、潤滑油基材油精製プラントは、プロセスの切り替えが発生する非連続的な処理が必要となる。非連続プロセスに対してはLPの適用に限界があり、過去にsuccessive LPの適用やシミュレータの開発が試みられたが、運転切り替えの際に発生する多様な状況をアルゴリズムとして定式化することが難しく、満足な解は得られていない。

(2) システムの目的

従来手法では開発が困難だったために、スケジュールの作成は専門担当者に委ねられてきた。しかし、近年のプロダクト油種の増加、プラントの新設等に伴い、担当者の作業ロードが増加しており、スケジュール担当者の作業負荷の削減を目的に潤滑油精製プラントのスケジューリングエキスパートシステムを開発する。

(3) 利用対象者

利用対象者はスケジュールリング作成専任担当者であるので、専門家が利用するシステムといえる。

(4) 期待効果

スケジュールリング作成専任担当者の負荷の軽減とともに作成時間の短縮、及びより精製コストの安いスケジュールが作成できる。

(5) システムの責任レベル

潤滑油精製プラントのスケジューリングエキスパートシステムは、スケジュールリング担当者の支援システムであり、最終スケジュールは専任担当者が決定する。従って、責任レベルはLight Dutyと言える。

3.5.1.2 システムの概要

(1) システムの機能と構造

① システムが実現する機能

潤滑油機材油精製プラントは、原油タンクから常圧蒸留装置にフィードされた原油を、減圧蒸留装置にかけて複数種の潤滑油原料油を採取するためのプラントである。原料油は工程1～4を通り、その間に粘度調整、ワックス分除去、敷きそう調整等が行われ、潤滑油基材油が得られる。各工程は、一時点に於いて一油種しか処理することができず、工程間の一次保留タンクも油種によって決まっている。又、工程1、2は2系統の装置からなり処理能力に差があるものの、装置2のプロダクトを装置3にかけるような運転も可能である。これらは、複数工程からなる非連続なバッチプロセスである。

一定期間の常圧蒸留装置の運転計画はLPを利用した生産計画、基材油の需要量は下流のブレンディング装置の運転計画により決められている。本スケジューリングエキスパートシステムは、系内への受入量、系外への排出量が決められた制約の中で、装置1～6で何時、どの油種を処理するかスケジュールを立てるシステムである。

② システム構造

黒板モデルをベースとした、得点評価による仮説生成検査法に基づく前向き推論システムを採用した。静的対象物（装置、タンクなど）についてはフレームによる知識表現を行った。知識ベースに格納した知識の内、特に、切り換え等については、専門家からの知識獲得によって集めた。静的対象物に関しては、KEがモデル化を実施した。

③ 他システム連携

インベントリバランスカリキュレータを中心とした各種計算、在槽量などのDBを用いている。また、スケジュール結果の表示、各種レポート出力機能とのインタフェースをとっている。

これらのデータは、基本的には初期データと結果の処理のためのシステムであり、エキスパートシステムとしての、リスク、解の正解率などの問題は生じていない。

(2) システムの開発者と開発手順

開発者は、開発のライフサイクルによって変わった。各開発サイクルと開発者を示す。

① 研究モデル開発 (85. 10 ~ 86. 4)

2 工程 4 油種のスモールモデルによるスケジューリングエキスパートシステムの Feasibility-study を実施した。これにより計画型問題へのエキスパートシステム適用の有効性を実証した。さらに、ノード展開によるバックトラック実現等の基本的なノウハウの獲得を行った。

開発体制は、専門家 1 名 (作業負荷は 0. 5)、開発者 (KE) 1 名、メーカ SE (KE) 2 名である。

② 実用プロトタイプ開発 (86. 11 ~ 87. 5)

実際のプロセスと同規模のモデルで実用性の検証をした。開発体制は、KE 2 名、専門家 1 名の体制となった。

③ 実用システム開発

実用プロトタイプ版の拡充、修正を実施するとともに、実用化のための既存システムとのインターフェース部の開発を実施した。開発体制は、実用プロトタイプ開発と同じである。

(3) システムの動作環境

① 対象ハードウェア FACOM M-760 (大型汎用コンピュータ)

② 使用メモリ 4 MB (仮想空間)

③ 使用ファイル エキスパートシステム部だけでは数 MB 以下

④ OS OS IVF 4 MSP

⑤ 処理方式 TSS 配下の会話型処理 (コンパイルコードによる)

⑥ 言語 / ツール UTILISP / ESHELL

(4) システムの規模

① KS (知識源) 13

② ルール 140

③ フレーム 77

④ 黒板上のレベル 66

⑤ 利用者定義関数 122

3.5.1.3 問題のタイプとシステムの特性

(1) 問題のタイプ

合成型問題（計画問題）

(2) システムの特性

表3.5-1に、システムの特性を示す。

3.5.1.4 客観的な評価の可能性について

(1) 実行結果の評価

月次モデルを例に、エキスパートシステムの結果と専門家の結果を比較検討する。各プラントによる差異はあるものの、トータルでみると切り替え回数、コスト共にほぼ専門家と同様な解を得ている。

しかし、エキスパートシステムの結果は、在槽量の推移をみると、切り替え回数を減らすために、在槽量の許容範囲内の限界を狙った運転をしており、振幅が大きい結果となっている。

(2) 効果

スケジュール担当者のスケジュール作成ロードは、導入前に1日～1.5日かかっていた作業が1時間以内で作成できるようになっており、導入の効果は上がっている。

3.5.1.5 今後の課題

専門家の支援システムとして実利用中であるが、今後はスケジュールの質の向上（振幅が少ない計画立案等）とともに推論時間（問題によっては、数十分）の短縮化を図る予定である。

表3.5-1 システムの特性一覧表

	計画型システム	関連性	実現性	コメント
定義	利用可能なデータ系列、時刻、場所、対象物、現象、目標、人、組織、資源、資金、能力と			
問題の特徴	1. 広大なデータベース 2. 多様な環境 3. 複雑な相互関係 4. 部分最適と全体の最適 5. 既成事実の打破 6. 費用対効果 7. 測定の精度向上	◎ ○ × ○ × ◎ ◎	◎ ◎ ○ ◎ ○ ◎ ◎	
基本タスク	1. 階層化 2. 探索的相互作用 3. 環境の測定の知能性 4. 条件の予測と検出 5. レベルの多様性 6. 事例の評価 7. 計画の最適化	◎ ◎ ◎ × ◎ × ◎	◎ ◎ ◎ ○ ◎ ○ ○	
問題解決機能	1. 戦略的探索 2. 精密化 3. 戦略的探索 4. 戦略的探索 5. 戦略的探索 6. 戦略的探索 7. 戦略的探索	○ × ○ ○ ○ × ×	○ ○ ○ ○ ○ ○ ○	仮説生成検査

3.5.2 評価結果

(1) 知識ベースの妥当性評価

① 知識獲得方法

制約条件などの洗いだしは、専門家とKEの共同作業、経験的知識は専門家からのヒアリング、対象のモデル化はKEが実施した。

② 知識表現

ルール表現、フレーム表現、手続き(LISP)による表現(ESHELL採用)を採用した。

③ 直観的な知識

計画問題の特性として曖昧性を含んだ直観的な知識は、使用していない。

④ 整合性の確認

整合性については知識源(KS)単位での知識の整合性の確認を実施した。しかし、ルール数が100程度であり、特に問題は生じていない。

(2) 問題解決・推論方式の妥当性評価

① 推論処理の妥当性評価

解の合理性の評価を実施した。

② 推論時間の妥当性評価

通常は5～10分で終了するが、何回な問題では数10分に及び、会話型システムとしては妥当性に欠ける(改善の必要性あり)。

③ 推論手法の選択の適切さの確認

日毎、装置毎のスケジュール案を作成しながら推論を進めて行く。しかし、問題によっては組み合わせの爆発が生じて unnecessary 仮説を生成している可能性もある。枝刈りの処方等に改善の余地はある。

④ 不確実情報に対する処理技法の適切さの確認

Fuzzy、CFなどの不確実な情報処理は実施していない。

⑤ 学習機能の適切さの確認

学習機能なし。

(3) 結果の妥当性評価

① 専門家による検証

解の途中過程を含めた専門家による検証を実施した。

②統計的手法

結果の正当性評価のために統計的手法は使用していない。

(4) 要求仕様に対する達成度評価

①要求仕様設定手段の評価

富士通のエキスパートシステム開発のための作業標準ES/SDEMによる。

②要求仕様内容に対する評価

以下の制約条件を満たすスケジュールを作成すること。

- ・タンクの溢れ、フィールド油の不足を生じさせないこと。
- ・切り替え回数を少なくすること。
- ・切り替えコストを低下させるために、切り替えコストの安い油種を選択すること。
- ・切り替え禁止油種を守ること。
- ・連続的に処理したほうが良い油種があるので考慮すること。

③要求仕様に対する達成度評価

解としては専門家と同等レベルのスケジュールが作成できるようになった。実行時間に関しては不十分である。

(5) システムの総合評価

①テスト技法の評価

推論の途中経過を含めた専門家との比較を実施した。

②利用者の満足度評価

スケジュール担当者の負荷軽減(1~1.5日→1時間)。レポート類の自動作成など利用者は満足している。

③利用時のインタフェース評価

実業務用(グラフィック画面、帳票類の出力など)のユーザインタフェースを備えている。

④開発方法の適切さの確認

実験型(研究)プロトタイピング、発展型(実用)プロトタイピング、実用システム開発とES/SDEMに則ったライフサイクルで開発した。

3.6 抄紙生産計画エキスパートシステム

3.6.1 プロフィール

3.6.1.1 システムの目的と期待効果

(1) システムの目的と期待効果

システムの目的と期待効果は、以下の通りである。

- 計画立案の効率化を図る。
- 合理的計画による設備の最適運用を図る。
- 知識を顕在化させ共有化することにより人に依存しない生産計画を立案する。

(2) システムの位置づけ

システム全体の位置づけを図3.6-1に示す。

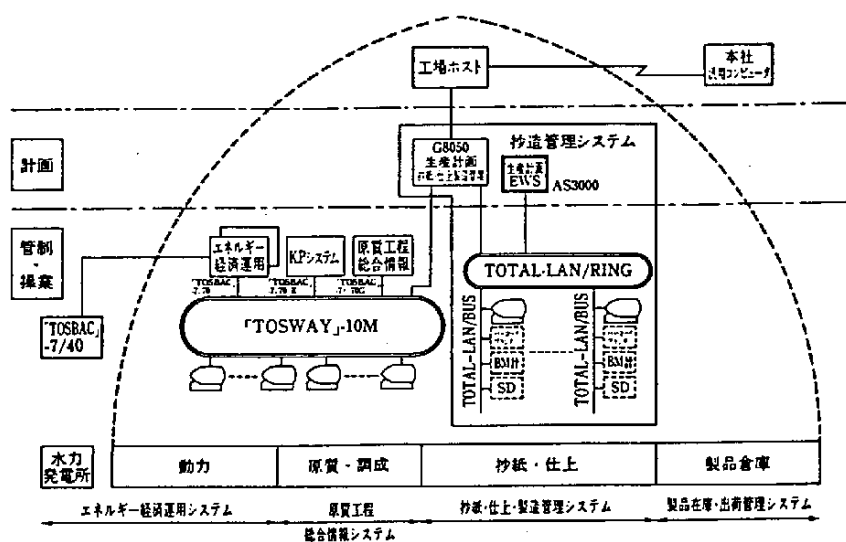


図3.6-1 システムの位置づけ

3.6.1.2 システムの概要

(1) 抄紙とは

抄紙とは、複数のパルプを混合して紙を抄くうことである。パルプはそれぞれのパルプ製造工場で作られ、抄紙機までパイプで送られ、抄紙機によって紙として出力される。工程の流れは以下の通りである。

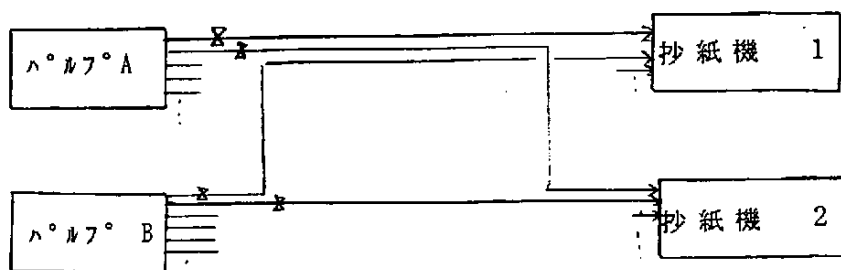


図3.6-2 抄紙の工程

(2) システムの規模

システムの規模は以下の通りである。

- ① 抄紙機 10数台
- ② パルプ種 約20種
- ③ 製品数 約200種類

(3) 抄紙生産計画ESの流れ

図3.6-3に、抄紙生産計画エキスパートシステムの流れを示す。

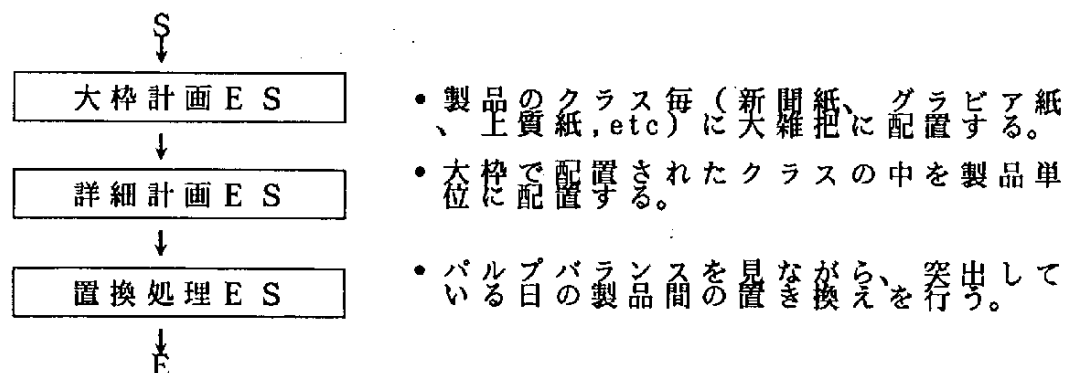


図3.6-3 抄紙生産エキスパートシステムの流れ

(4) 抄紙生産計画の特徴

抄紙生産計画の特徴は、以下の通りである。

- ・通常の制約を守るとほとんどの場合解が出ない。すなわち制約緩和がポイントである。
- ・注文はロット単位でくるが必要に応じてロット分割を行ない対応している。このため、解空間は無限になる。

(5) 抄紙生産計画エキスパートシステム化のポイント

エキスパートシステム化する際のポイントは、以下の通りである。

① 絶対制約

- ・納期は厳守すること。
- ・パルプバランスの制約（パルプ生産能力を越えないこと）。

② 評価

- ・パルプバランスがとれていること（パルプ生産量の分散が小さい）。
- ・輸送バランスがとれていること。
- ・制約緩和が少ないこと。

③ 技術的ポイント

- ・大枠、詳細配置に於ける『布石方式』（制約の厳しい注文を見つけ、先に配置する）。
- ・置き換え処理に於ける実行結果の評価の仕方（枝刈りの基準）。
- ・既存システムとの整合性ある結合。
- ・ユーザ支援のためのマンマシンインタフェース。
- ・A S I R E X で作成。

3.6.1.3 問題のタイプとシステムの特性

本問題は典型的な計画型問題である。表3.6-1にシステムの特性一覧表を示す。

3.6.1.4 客観的な評価の可能性について

エキスパートシステムが解く計画型問題の解は最適なものである保証はない。しかし、ある評価関数（例：ラインバランス、納期遅延率等）を定めることにより定量的評価は行える。また、専門家の評価により、制約違反などの発見も比較的容易に行うことが出来る領域である。

表3.6-1 システムの特性一覧表

	計画型システム	関連性	実現性	コメント
定義	与えられた目標を実現するために、利用可能な資源（人、金、物など）を時系列データとして組織化すること。			
問題の特徴	1. 広い探索空間 2. 多様な評価属性 3. 不確実な環境予測 4. 部分計画間の相互作用 5. 既存計画の再利用 6. 費用と便益のトレードオフ 7. 対話型計画支援の要求	○ ○ ○ × × ○ ○	△ ○ △ △ ○ ○ ○	枝切りに有効な知識の収集 最適解ではなくより良い解 外部環境の変化への対応 支援システムとの結合
基本タスク	1. 計画過程の階層化 2. 組合せ探索 3. 制約条件の相互作用 4. 環境の予測 5. 上位レベルへの知的後戻り 6. 計画事例の探索と利用 7. 計画評価の多様性	○ ○ ○ × ○ × ○	○ ○ ○ － ○ ○ ○	初期配置→置替え 悪影響の認知 C B R の利用
問題解決機能	1. トップダウン精密化戦略 2. 拘束最小化戦略 3. 網羅的探索機能 4. 仮説推論による知的探索 5. 類推による計画事例の利用 6. 計画の多目的評価 7. リスク下での意思決定	○ ○ ○ ○ × ○ ×	○ ○ ○ ○ ○ ○ －	仮説生成 C B R の利用 評価関数の設定

3.6.2 評価結果

(1) 知識ベースの妥当性評価

① 知識獲得方法の選択の適切さの確認

- スケジュール作業の履歴を残し、そのトレースをフォローし、知識を抽出し、フィジビリティフェーズとプロトタイプフェーズの時期に知識収集・整理期間を約1年かけて行なった。
- なお、開発期間は以下の通りである。

フィジビリティフェーズ (半年)

↓

プロトタイプフェーズ (1年)

↓

実用フェーズ (1年)

- 専門家(1名)の積極的参加(非常に前向きの姿勢)。

② 直観的な知識の断片を正当化する方法

- 特になし。制約ルールが、専用エディタにより見通しの良い表示になっている。
- 制約間に矛盾があるか否かのチェックを入れることは可能である。

(例) Aの後にBを行なってはいけない(禁止ルール)

Aの後にBを行なった方がよい(推奨ルール)

ただし、A、Bは抽象概念で表わされていた場合は、概念をたどったマッチングである。

③ 知識表現の選択と適切さの確認

- プロトタイプフェーズの時は、プロダクションルールとフレームを主体で行なったが、読みにくくメンテナンスしにくいとの問題点が出た。このため、制約ルール等はデータファイル化し専用エディタによりメンテナンスをしやすいとした。また、他のルールはルールをパターン化しエディタによって修正・追加を行いやすいとした。
- 実運用の段階で、ルールメンテナンス用ツールを用いてメンテナンスが頻繁に行なわれている。

④ 知識ベース全体の整合性を確認する方法

- 特にサポートしていないが、一貫性としては、以下のような項目が想定される。

- 論理的一貫性
- 意味論的一貫性
- 他の知識ベース(データベース)との競合
 - 語彙(製品名、材料名はデータベースよりコンバート)

- 常識
- 辞書

(2) 問題解決・推論方式の妥当性

① 推論処理の妥当性評価

- プロトタイプ作成、及び実システム作成時に過去の実データを用いてテストし、実際のスケジュール表と比較チェックし、その正当性を検証した。

② 推論時間の妥当性評価

- 従来3日かかって月次計画を決めていたが、本システムでは、平均の大枠配置は20分、詳細配置は45分、置換配置は60分である。従って、合計約2時間であり、実際には充分利用できる範囲である。
- 場合によっては解が出ない可能性もある。その時は、時間を制限して推論させ、途中結果をユーザに見せ、マンマシンインタフェースにてユーザが解を与えることのできる支援機構をもっていることによって充分対応可能である。

③ 推論手法の選択と適切さの確認

- 最終的制約としてパルプバランスをとることが必要であるため、かなり厳密なスケジュールが必要である。このため、大枠、詳細、置換と3つのフェーズに分けて行なう推論方式をとった。
- 推論方式は、大枠・詳細配置はジョブの余裕度を考慮した「分布方式」という生成検査法を採用した。
- 置換処理は仮説推論（生成検査法）を採用した。

④ 不確実情報に対する処理技法の適切さの確認

- 特になし。

⑤ 学習機能の適切さの確認

- 機能なし。

(3) 結果の妥当性の評価

① 専門家による確認が妥当な問題やタスクの範囲

- (2)-①の推論処理の妥当性評価結果の通り、妥当な結果であった。

② 統計的手法

- パルプバランスの評価をするために、各パルプの平均量、分散を用いている。

(4) 要求仕様に対する達成度

① 要求仕様内容に対する評価

- 要求は当初から明確になっていた（ニーズ先行）。

(5) 知識ベースの保守性評価

- ① 専用エディタにより知識を読み易くした。
- ② 知識をタイプに分けその単位でメンテナンスできるようにした。

(6) マンマシンインタフェースの妥当性評価

- ① 出力されたスケジュール表をユーザは容易に手直しできる。
- ② 専用の知識エディタが用意されている。

(7) システムの総合評価

① テスト技法

- ・ 実際のデータの入手（計画型問題は行ないやすい）
- ・ 評価の定量化……最終評価のバルブバランスの定量的評価

② 利用者の満足度評価

利用者の満足度を測るために、処理時間、ユーザの推論結果の手直し度、回答不能率、バルブ供給量の分散の評価項目が重要であり、今回は概ね満足のいく結果であった。

③ 統合性の評価

- ・ 工場内のミルワイドシステムのサブシステムとして位置づけられ、各サブシステム間のインタフェースが確立し有効に機能している。

④ 開発手法の適切さの確認

- ・ 3フェーズに分けて開発を行った。
- ・ フィジビリティフェーズ（実験型プロトタイピング）
技術的可能性の検討を行うフェーズであり、このためにシステムの要求仕様（性能、範囲）の明確化、社内（客先）の根回し、オーソライズが特に重要であった。
- ・ プロトタイプフェーズ（発展型プロトタイピング）
システムイメージを明確化するフェーズであり、このために知識の整理・収集、社内（客先）のシステムへの期待感の高揚が特に重要であった。
- ・ 実用フェーズ
テストランが特に重要であり、このためシステムの単独運用を行うと同時に、知識のメンテナンスを充分に行った。

3.7 ダムゲート診断評価システム

3.7.1 プロフィール

3.7.1.1 システムの目的と期待効果

我国で、発電用ダムおよび関連した構造物が建造されはじめたのは昭和初期である。それから60年近く経過した現在においても当時の構造物の多くが使用されている。(財)電力中央研究所では、古いダムについてその安全性調査を10年以上にわたって実施しており、すでに多くの経験・知識を蓄積している。そこで、昭和59年度より、これらの経験・知識を集大成し、総合的な安全性判断・寿命予測手法の確立をめざして研究を実施している。これまでに、現場での計測法の改善、全国共通の調査票作成と調査情報のデータベースシステム化、有限要素法による簡易解析システムの開発などを行ってきた。本エキスパートシステム(Dam Gate Diagnosing Advisor; DGDA)はこの一環として、ダムゲート(水門)の診断用に開発したものである。

システムの目的は、従来の管理要領で「専門家に依頼する」としている部分の作業を明文化し、数多くのデータに基づいて、対象構造物の客観的・論理的な診断・評価を行いやすくすることである。そのために、DGDAは、ダムゲートのデータベースから、変位・応力などの定量的な測定データ、ならびに外観状態などの定性的なデータを、また、簡易解析システムから、測定データに対応する解析値をとりこんで、個々のダムゲートの総合的な診断・評価を行なう。データベース、簡易解析システムから得られる対象システムの構造・状態に関する情報と、ルールで表現した専門家の経験則を組み合わせる処理するのが特徴である[中村 1987]。

3.7.1.2 システムの概要

DGDAの機能構成は、図3.7-1に示すとおりである。システムはデータベース/簡易解析システムとともに利用される。利用者は、まず、現地での計測結果をもとに調査票を作成し、これをデータベースに投入する。次に、診断対象となる特定のダムゲートを検索・指定し、簡易解析システムを起動する。構造解析が終了すれば、解析結果とデータベース情報とを利用してエキスパートシステムによる診断・寿命評価を行う。エキスパートシステムに入力されるデータは、ダムゲートの構造形態・外観状態・操作状態・環境状態等の定性的情報(約

100項目／ゲート）と測定地値・解析値等の定量的情報（約 100項目／ゲート）からなる。これらのデータはフレームに変換した後、当研究所の専門家の経験的な知識（ルール、手続き）に基づいて推論処理を行なう。これによって安全性判定ならびに余寿命予測を行なう。

知識ベースはフレームとルール、手続きとから構成される。フレームには上に述べたダムゲートの定性的／定量的情報と対象システムの構成情報とを保持する。ルールは2つのグループに分れている。第1のグループは、入力したデータのチェックと各構成部分の評価スコアの計算とを行なうときに使用される。第2のグループは、さらに2種類に分類されており、外観状態と構造力学的観点とからダムゲートの診断・評価を行なうときに使用される。手続き的知識はフレーム中の個々の構成情報スロットに付随する関数であり、それぞれ評価スコアを計算するために使用される。

本システムの扱うデータは測定・解析による客観的な定量データとアンケートによる主観的な定性データの両方を含む。したがって、それらを用いて下す結論は本質的にあいまいなものとなる。本システムではここで生ずるあいまいさを処理するために、評価用スコアの導入と複数の解の提示という、2つの方式を採用している。これは次の理由による[中村 1986]。

- ① システムの下す複数の結論は互いに矛盾する可能性がある。ところが客観的な寿命評価手法が確定していないという現状のもとでは、システムを利用する専門家に、システムの結論の評価をまかせるべきである。そのためには、システムが結論し得るすべての解を表示することが適切である。
- ② ダムゲートのようなエンジニアリングシステムを対象とする場合、個々の推論結果は、データまで遡って明確に説明できるものでなければならない。それには、スコアの評価を陽に計算する方式が明解である。そしてこのスコアの妥当性を保証するために、AHPを使って定めた評価テーブルを採用している。

システムの開発の経緯は表3.7-1のようにまとめられる。システムのうち、第2版までが探索型プロトタイプの性格が強く、問題解決手法の妥当性の検討に当てられた。第3版は、実験型プロトタイプであり、これによって図3.7-1の機能をほぼ全面的に実現し、システムの実験評価を行った。大型コンピュータ上に再開発した第4版からが、発展型プロトタイプであり現場での評価を繰返しながら現在にいたっている[中村 1990]。

表3.7-1 開発経緯のまとめ

1985年 4月:	既存部分(DB,FORTRAN)開発
1985年 4月:	エキスパートシステム化決定 (それ以前の計画では統計的手法を使う)
1985年 5月:	一次案としてのルール作成 (IF-THEN + 基本的な計算式)
1985年 6月:	デモ用プログラムをM200上に作成 (デモ終了後廃棄) (第1版)
1985年 9月:	技術的検討
1985年 9月:	K E E の導入
1985年 10月:	DB情報を元にダムゲートのフレーム化 ルールの追加・修正 (第2版)
1985年 11月:	ルールの追加・修正 (第3版) システムのテスト
1986年 3月:	大型機上で実用システム開発 (第4版)
1989年 :	試行中

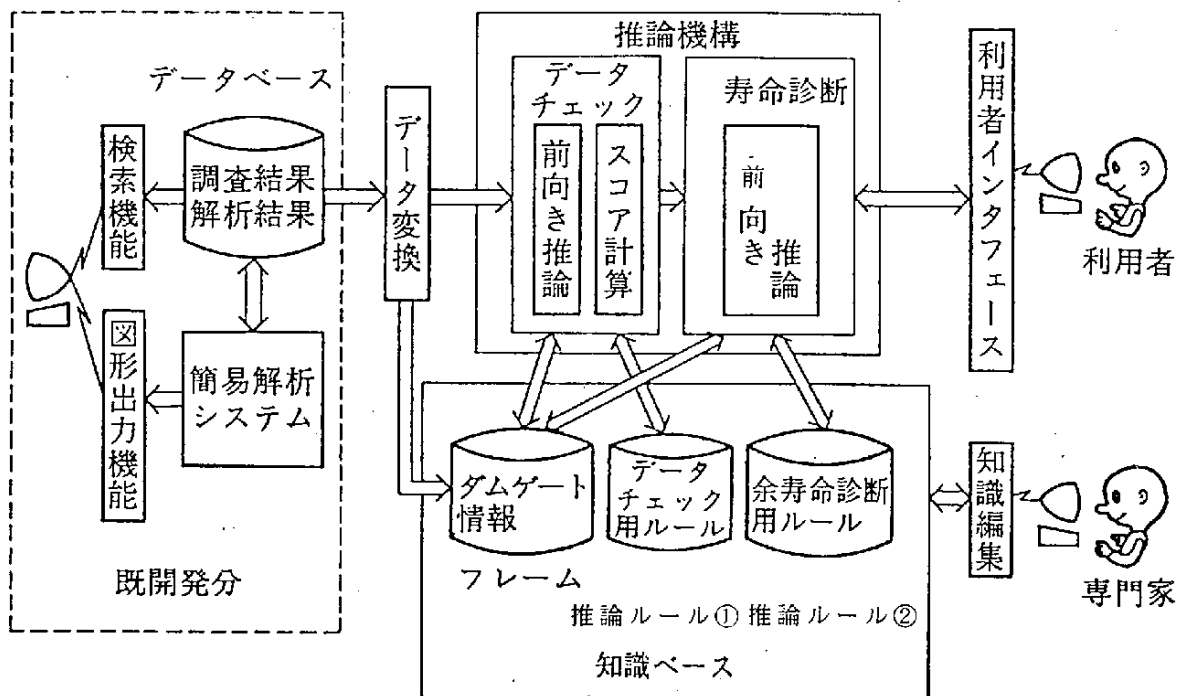


図3.7-1 ダムゲート診断評価システムの構造

3.7.1.3 問題のタイプとシステムの特徴

本システムは、診断型エキスパートシステムであり、表3.7-2の特性をもつ。

表3.7-2 システム特性一覧表

	診断型システム	関連性	実現性	コメント
定義	システムと診断問題の定義、問題の観測、異常さの同定、知識の活用、故障の診断、対象の活用、生じる原因、及び、システムの利用、実用性、コスト、制御	○	○	典型的な診断問題である。ただし、単純な線形予測外挿を用いる。
問題の特徴	1. 設計経験的知識の活用 2. 経路探索的知識の活用 3. 作業測定の知識の活用 4. 知識の活用 5. 知識の活用 6. 知識の活用 7. 知識の活用	○ ○ ○ ○ ○ △ ×	○ ○ ○ △ × × ×	ゲームによる構造知識、AHPによる情報の一部、アンケートは非常にコストがかかるフレーム/ルールで対応可
基本タスク	1. 異常現象の検出 2. システムの診断 3. 問題の選定 4. 原因の推定 5. 異常現象の検出 6. システムの診断 7. 問題の選定	× ○ ○ × ○ ○ ○	× ○ ○ × ○ ○ ○	フレーム、AHPの階層、ルール、数値計算による
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. 仮説検証推論 5. 協調推論 6. 効率性推論 7. 費用/効果分析	○ × ○ × × ○ ×	○ × ○ × × ○ ×	Ver.4は全て前向き推論、AHPとルール/フレームの組合せ、本ES自身、診断の効率向上を目指している

3.7.1.4 客観的な評価の可能性について

DGDAは予防保全を目的とする診断システムである。そのために、システムが下した結論の正しさを実地において確認するという形での評価は原理的に不可能である。また、従来のダム構造物に対する事故例もごくわずかしかないために、システムの診断結果を統計的に処理し、その信頼性を判定することもできない。

したがって、システムの客観的な評価は、知識ベース中の知識が利用者にとって妥当なものであるか、また、利用者インタフェースが適切なものであるか、統合性、発展性の意味で成熟したシステムとなっているかでなされる。これには、ソフトウェア自身の開発とは離れた部分でのシステムの権威付けが必要であり、コンセンサスを得るまでに非常に時間がかかるものとなっている。

3.7.1.5 今後の課題

DGDAは、現在、実用化に向けての最終段階にある。また、DGDAと同じアーキテクチャで、水力発電所の主要な鋼構造物である水圧鉄管の診断システムも開発している。これらのエキスパートシステムは、いずれも、ダム関連のデータベース、構造解析システムと組合せて、一つの統合システムとなっている。今後は、エキスパートシステムとしての評価作業がほぼ終了した後に必要となる、永続的な保守管理のプロセスについて具体的な検討を進めていく予定である[中村 1990]。

3.7.2 評価結果

(1) 知識ベースの妥当性評価

本システムの知識ベースは、部分的に存在する診断評価基準を、専門家の経験則に基づいて整理したものである。知識ベースは比較的小さく、また、推論ステップも短いので、全数チェックに近い形でシステムのテストを実施することができた。知識そのものが妥当であることを利用担当者に確認させるためには、上位レベルの診断知識をドキュメント形式で提示することとした。下位レベルの知識を評価するためには、上で述べた理由からAHPを利用した。

(2) 問題解決・推論方式の妥当性評価

DGDAは、対象知識の表現にフレームを用い、推論には前向きのルールを利用している。さらに、不確実な情報処理を容易化するためにAHPを利用した。本システムで、AHPがうまく適用できたポイントは次の2点にある。まず、知識処理技術の面からは、フレーム概念によって、知識の階層的整理が容易に行えたこと、ルール概念によって経験的な意思決定知識を記述できたことが上げられる。一方、AHPの面からは、評価を手軽に行えたこと、知識の整合性を検証することができたことが上げられる。

(3) 結果の妥当性評価

DGDAの実験評価は、第3版システムを使って最初に実施された。それには、すでに修理・交換ずみのデータを含む、専門家による調査が終了したダムゲート25ヵ所の情報が使用された。この結果、システムの診断結果は、従来の専門家のものよりもやや厳しくなることが確認された。以後は、新しい測定結果の入手、ならびにシステム試用の要請にしたがって、DGDAを稼働させ、知識ベースを少しずつ改良することで、より妥当な結果が出るようにシステム評価を継続している。

(4) 要求仕様に対する妥当性評価

本システムの開発は、専門家が中心となっていたため、利用者からの要求仕様を明確化する作業は、発展型プロトタイピングの最終段階で行われた。そのため、とくに、要求仕様に対する妥当性の評価は行っていない。

(5) 知識ベースの保守性評価

DGDAの知識ベースは、ダムゲートの構造、ならびに、構造解析の視点から自然に定まる部分と、診断意思決定に関連する不明瞭な部分とからなる。そこで、

知識ベースの保守を容易化するために、意思決定にかかわる部分のルールを利用者にわかりやすい形でドキュメント化している。また、システムの開発は第4版システムからは、一貫して同一担当者があたっており、通常のソフトウェア開発で作られるような開発ドキュメントは整備されている。

(6) マンマシンインタフェースの妥当性評価

マンマシンインタフェースの評価は、利用担当者が第4版システムを試用する過程で行われた。第4版システムは、大型コンピュータ上でTSSを使って実現されているため、インタフェースの機能は十分ではない。特に、レスポンス時間、図形出力、レポート出力などの点で不満が見られた。

(7) システムの総合評価

システムの総合評価は、知識ベースの妥当性を専門家・管理者間で検討する過程と、システムを利用担当者が試用する過程とで行われた。その結果、システムのもつ知識に対しては、ほぼコンセンサスが得られており、また、システム機能としては測定データと構造解析結果とを比較対照できる点が評価されている。

【参考文献】

- [中村 1986] 中村秀治, 松浦真一, 寺野隆雄, 篠原靖志: 水力鋼構造物の寿命予測エキスパート・システムとその適用, 土木学会論文集, No. 374/1-6 (1986), pp. 513-521.
- [中村 1987] 中村秀治, 寺野隆雄: 土木構造物エキスパートシステム, オーム社, 1987.
- [中村 1990] 中村秀治, 山本広祐, 寺野隆雄: 水力鋼構造物・評価診断システムの試験的利用について, 電力土木, to appear 1990.

3.8 機械材料選定エキスパートシステム

3.8.1 プロフィール

3.8.1.1 システムの目的と期待効果

機械設計のためのコンピュータシステムとしては既に多くのものが開発されており、現場で実用に供されているものも少なくない。既存のシステムをその用途で分類すると、構造物の剛性や変形などの構造静特性を扱うもの、固有振動数、振幅などの構造動特性を対象にしたもの、あるいはロボットの運動特性などを計算するものが多いが、近年構造物の寿命が長くなるにつれて問題になっている疲労破壊を扱うことができるシステムは数少ない。疲労破壊に関しては基礎的なデータが十分ではなく、また解析手法的に見てもまだ解明されていないことが多いため、従来は強度屋などと呼ばれる専門家の領域となっていた。本システムは基本となるデータベース、解析手法に加えて専門家の持つノウハウを知識ベースとして実装し、これらを組み合わせることによって疲労破壊に関する専門家でなくとも、機械設計に関する基本的な知識を有する技術者であれば疲労強度の問題を扱うことができるようなシステムを目指したものである。

3.8.1.2 システムの概要

(1) システムの機能と構造

① 全体システムの機能

本システムは機械材料の疲労強度を考慮した設計支援システムである。材料の疲労強度の計算をするための、機能は以下の通りである。

・データベース検索

基本になる強度データ（破壊に至る応力の強さと繰返し数の関係を示したもの）や亀裂特性データ（亀裂進展速度と応力の関連を示したもの）が各所で収集されており、これらをデータベース化して解析の基本データとして用いる。

・疲労強度設計

検索したデータは標準試験片を用いた試験によって作成されたものである。実際に設計の対象としている構造物の疲労強度を計算するためには、検索したデータを基に様々な特性（組み合わせ応力、寸法、形状、表面状態など）を考慮しなければならない。このために多くのアルゴリズム、計算理論が提案されており、設計の対象によってこれらを使い分けなければならない。このプロセ

スには経験的な知識が重要である。

② エキスパートシステムとしての機能

上の疲労強度設計の一部をエキスパートシステムとして実現している。具体的には設計の対象によって最適なアルゴリズムや計算式を求める問題を診断問題として定式化し、後ろ向き推論によって解を導き出している。

・推論方式

後ろ向き推論を用いている。不確実な情報は確信因子で処理する機構を持っているが、現在までのところこのようなルールは存在していない。

・知識表現

プロダクションルールのみを用いている。

・知識源

一般的法則、既存ドキュメントからの知識をもとにしている。専門家の知識を反映していないのは技術的な問題ではなく、本システムを開発したプロジェクトの条件・制約による。

・システム機能実現の困難さ

本システムは設計のための情報を提示するものであるため、特にリスクは無い。

・利用頻度

利用頻度はユーザによって異なるが、単純なデータベース検索に始まり、基本的な解析ルーチンがほとんど組み込まれているため、利用頻度は比較的多いと考えられる。

・システムの機能

システムの機能は設計の支援を行うことであり、代行を目的としたものではない。したがって、ユーザインタフェースの良否がシステムにとって重要である。

・システムの対象範囲

システムの対象範囲は主として扱うことのできる材料で定められる。現システムは鉄系の材料を主体としている。セラミクスなどの新しい材料への対応も求められているが、基礎的なデータが無く、まだ困難である。

③ システムの構造

システムの構造は、図3.8-1に示す通りである。

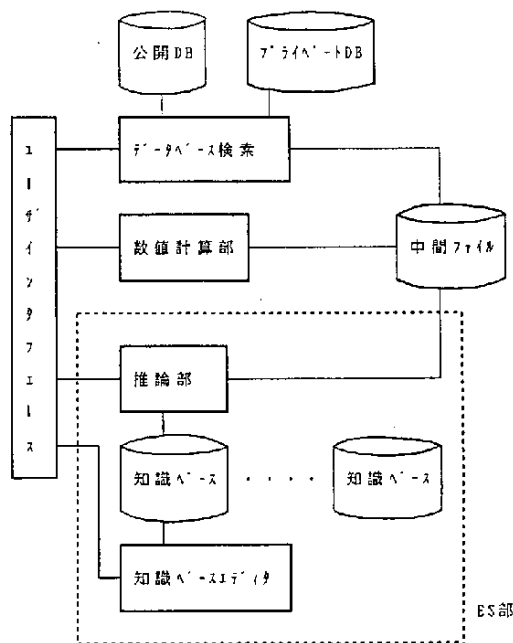


図3.8-1 システムの構造

(2) システムの開発者と開発手順

本システムの開発は複数のユーザ（20程度の異なったバックグラウンドを持ったユーザ）の要求をまとめる形で行ったため、具体的なシステム開発はKE、SEが主導することとなった。すなわち、システムの仕様に関するドキュメントを開発者側（KEおよびドメインの知識、経験を持つSE）が提示し、それに対してユーザ側がコメント、要望を加えるというステップを繰り返すことによって、システムの仕様を明確にし、開発を行った。したがって、システムの機能は多少汎用的になっており、設計の対象を特化した形での実用システム（一口に疲労設計と言っても具体的な対象によって使用材料、環境、応力の強さ、繰り返し数などの基本条件が大きく異なる）にはいま一步というところで、それは今後の課題となっている。

開発手順については特に準拠しているものではなく、既存のシステム開発の方法にプロトタイピングを併用したものとなっている。

(3) システムの動作環境

システムの動作環境は、以下の通りである。

① 対象ハードウェア

- ECLIPSE DSシリーズ
- APOLLO DNシリーズ
- SUNシリーズ

② 使用メモリ量 約8MB

③ 使用ファイル量 約100MB (プログラム領域、データベース、知識ベース含む)

④ OS UNIX 4.2BSD

⑤ 処理方式 コンパイラ方式

⑥ 言語/ツール

ES部	… ZEUSのサブセット
数値計算部	… FORTRAN, C
データベース部	… FORTRAN
ユーザインタフェース部	… C, X-window

(4) システムの規模

システムの規模は、およそ以下の通りである。

① 数値計算部 約45Kライン

② データベース部 約5Kライン

③ ユーザインタフェース部 約10Kライン

④ ES部 約5Kライン、約200ルール (14知識ベース)

3.8.1.3 問題のタイプとシステムの特徴

(1) 問題のタイプ

① システム全体

システム全体としては設計問題を扱っているが、自動的な設計を行うことが目的ではなく、ユーザである技術者が最終的な判断を下すのに必要なデータをシステムが提供するという形式になっている。すなわち、DBから必要なデータを検索し、それを基に疲労強度を算出し、目的とする機械構造物を構築するのに必要なだけの強度を備えているか否か判断するという流れになっている。

② ES部

エキスパートシステムが扱っている問題のタイプは診断型である。すなわち、データベースから求めた疲労強度は標準試験片を基にして算出したものであるため、設計の対象としている部品の強度に換算するためには、部品の大きさ・

形状・表面の状態など様々な要因を考慮することが必要である。このために様々な解析的な手法が提案されているが、これらを適切に使い分けることは容易ではない。この点に関してアドバイスすることがエキスパートシステムの目的である。

(2) システムの特性

表3.8-1に、システムの特性を示す。

表3.8-1 システムの特性一覧表

	診断型システム	関連性	実現性	コメント
定義	診断問題は、対象とするシステムに異変が生じたとき、その原因を特定し、正常な状態に戻すことである。			異常事態の発生、観測、診断、選択、実行、評価、フィードバック。
問題の特徴	1. 設計経験的知識の活用 2. 経験的知識の活用 3. 経験的知識の活用 4. 経験的知識の活用 5. 経験的知識の活用 6. 経験的知識の活用 7. 経験的知識の活用	○ ○ ○ ○ ○ ○ ◎	○ ○ ○ ○ △ ◎ ◎	固有の経験則、少 インタフェース上重要
基本タスク	1. 異常現象の検出 2. システムの診断 3. 診断結果の表示 4. 診断結果の表示 5. 診断結果の表示 6. 診断結果の表示 7. 診断結果の表示	○ ◎ ○	○ ◎ ◎	手法選択に相当
問題解決機能	1. 事象認識 2. 目標設定 3. 仮説生成 4. 仮説生成 5. 仮説生成 6. 仮説生成 7. 仮説生成	○ ○ 	○ ○ 	

3.8.1.4 客観的な評価の可能性について

エキスパートシステムが知識ベース中の知識通りに動作しているか否かは簡単に評価・検証できるが、用いられた知識そのものの正当性を検証することはかなり困難である。厳密に言うと、その指示に基づいて実際に対象物を設計・製造して実験して見なくてはならず、しかも疲労破壊の検証には長時間を要するし、本来統計的な現象であるため、単一のケースでは意味を持たないからである。

3.8.1.5 今後の課題

現段階のシステムは実現化システムではあるものの、実際には運用経験が少ないので、まず現場での運用実績を積み重ね評価・改良していくことが重要である。エキスパートシステムとして見ると具体的には次の2点が大きな課題である。

(1) 設計対象に応じた知識ベースのカスタマイズ

疲労設計は具体的な設計対象によって、基本となる材料データ、解析手法の使用法などが異なる。また、各ユーザはこうした点を考慮して独自の知識を蓄えており、これらの知識をシステムに取り込むことが必要である。

(2) ユーザインタフェースの充実

前述した通り、本システムはユーザの設計業務を支援することを目標としている。したがって、ユーザが正確な設計を実現するために必要なデータ、計算アルゴリズムを自然な形で提示することができなければならない。この意味でユーザインタフェースは非常に重要である。

3.8.2 評価結果

(1) 知識ベースの妥当性評価

知識表現形式としてプロダクションルールのみを用いており、知識ベースの規模も小規模であるため、シンタックス的な妥当性あるいは個々の知識が期待通りの動作のみをすることについては簡単に確認できる。また意味的な妥当性に関しても現段階では教科書的な知識に限定しているため、大きな問題は無い。将来的には各ユーザ毎にカスタマイズされる可能性があり、この際には知識の意味的な一貫性が重要になる。

(2) 問題解決・推論方式の妥当性評価

プロダクションルールに基づく後向き推論を行っているが、現在の問題の範囲ではこの程度で妥当と考えられる。

(3) 結果の妥当性評価

診断型のシステムで知識ベースの規模も小さいため、網羅的な検証が可能であり特に問題は無い。

(4) 要求仕様に対する達成度評価

機能的にはほぼ要求仕様を満たしているが、性能面（実行速度）で多少問題がある。特にES部と数値処理部とのインタフェース、結果の表示時などに時間がかかり、改善の余地がある。

(5) 知識ベースの保守性評価

プロダクションルールのエディタによって知識の追加・削除・修正などが行える他、ルールの重複・冗長性のチェックなどの処理もできる。知識ベースがさらに複雑になるとビジュアルなインタフェースなどが欲しいが、現状の知識ベースの規範ではこの程度で充分であろう。

(6) マンマシンインタフェースの妥当性評価

設計時に必要な情報を提示することによって設計の支援を行うシステムであるため、マンマシンインタフェースは非常に重要である。グラフィカルなインタフェースを多用することによって機能的には一応のレベルに到達しているが、レスポンスについては改良の余地がある。

(7) システムの総合評価

数値計算に重点を置いた解析的なシステムと設計のためのノウハウを提示するSE部とを有機的に組み合わせたシステム構成によって設計のプロセスを総合的に支援することが出来るが、ユーザ毎のカスタマイズ、実行速度の向上などの改良によってさらに改善の可能性がある。

3.9 エンジンエレクトロニクス技術相談システム

診断型システムの事例として、トヨタ自動車(株)で開発された自動車エンジンのエレクトロニクス部品について技術相談(故障診断)を行うシステム[温井]、[大鹿]を紹介する。

3.9.1 プロフィール

3.9.1.1 システムの目的と期待効果

自動車へのエレクトロニクス部品の適用が増えている。エレクトロニクス部品は機械部品に比べ機能・構造が複雑であり、また不具合の発生頻度が少ないことからその診断には高度な専門知識が要求される。さらに、多岐にわたる製品に関する最新の診断情報の保守を効率的に行う必要がある。このような背景からエレクトロニクス部品についての診断を支援するシステムが要望されていた。

本システムは、販売店で対応が困難な不具合についてコンサルテーションを行うことを目的としている。システムは、サービス部門の技術者が整備士からの電話ないしファクシミリによる問い合わせに応える際に用いられる。現在は問い合わせの中で4割程度が本システムによって回答されている(図3.9-1参照)。

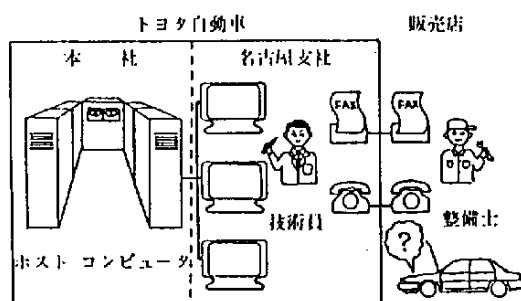


図3.9-1 システムの利用形態

エキスパートシステム技術を利用することにより、診断の専門家のノウハウと製品開発者の知識および市場情報を活用し、診断精度の向上や診断の迅速化などのサービス性向上に効果を発揮することが期待されている。

3.9.1.2 システムの概要

システムの機能は利用者との対話により原因を絞り込むことである。その際、これまでの専門家によるコンサルテーション手順を反映させて、車両型式や基本的な点検結果などを、はじめのステップでまとめて入力し、その後点検手順などを考慮しながら問診を進めるようにした。

ソフトウェアの構成は図3.9-2のようになっている。知識ベースは、①不具合モードの分類木、②問診対話のための質問文と点検要領および基準値、③機能系統と構成部品、④原因に対応する点検修理手順、の4種類からなる。

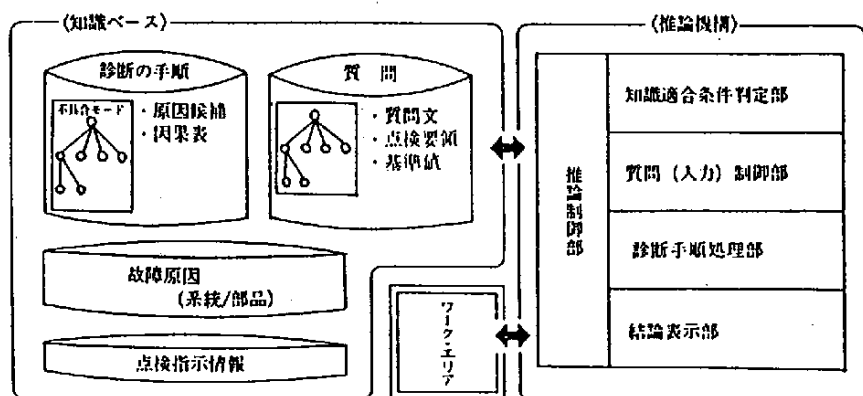


図3.9-2 ソフトウェア構成

推論は、不具合モードの分類木をたどりながら、各ノードに記述された症状—原因を表す因果表に従って問診質問を行い、原因仮説に重みを与えていくことで進められる。

本システムには、修理に要する時間や交換部品のコストを最低限に抑えるため出力結果の正当性が要求される。また、多様な車種・型式に対応する必要があることから、知識ベースの保守性が特に要求される。

システム開発には、サービス部、技術部、情報システム部から成るプロジェクトチームがあたり、図3.9-3に示すような経緯を経て昭和63年3月から稼働している。現在も取扱い範囲の拡大などの改良が続けられている。

61/5		61/12	62/6	63/3
開発フェーズ	プロトシステムの開発	実用システム検討のためのプロトシステムの拡張	実用システムの開発△	運用開始
対象エンジン	1 G系エンジン	←	全電子制御エンジン	
対象不具合	始動不良	エンスト、ドライバリティ不良など	←	
成 果	知識整理	マトリックス表	←	
	知識ベース	フレーム	分類木 フレーム + プロダクションルール	←
	推 論	←	確信度の加算方式	←
	そ の 他	←	←	応答性向上 運用支援機能

図3.9-3 開発経緯

開発には富士通の汎用大型機と端末とを用いた。推論機構はLISPで約5000ステップ、フレーム数は2300個、メモリは5.2MBである。現在はWS上にC言語で書かれた専用のツールを開発し利用している。

3.9.1.3 問題のタイプとシステムの特性

本システムは診断型システムであり、表3.9-1のような特性を持つ。

表3.9-1 システムの特性一覧表

[illegible]

3.9.1.4 客観的な評価の可能性について

診断結果については、該当する部品の交換によって不具合現象が解決されたかどうかを調べることでより評価できる。その他の技術的な項目については評価基準の設定が難しいが、システムの改良を行う際の要求仕様として徐々にまとめられてきている。経済的・社会的な項目についての評価は重要であるが、利用者の満足度のように定量的に捉えにくいことが多く、従来システムの評価とは異なる新たな視点が要請される。

3.9.1.5 今後の課題

本システムは昭和63年3月より運用に入っている。現在では利用者による保守体制も固まり、取扱い範囲の拡大などが行われている。今後は、知識ベースの充実をはかるとともに、マンマシンインタフェース機能の追加を行い実用性をさらに向上させていく予定である。

3.9.2 評価結果

(1) 知識ベースの妥当性評価

開発チームにはサービス部門と設計部門から専門家が参加している。このため、原因の絞り込み方などの経験的知識と設計知識とを突き合わせることでより正当性および完備性を確保している。

知識表現については、実験プロトタイプの評価から特に保守性を重視して定めた。現在は分類木と表形式とを組合せた構造を型式などによって階層化した表現を用いており、本システムの目的をほぼ満たしている。ただし、不確実性の表現などについては、さらに改良が必要と思われる。

(2) 問題解決・推論方式の妥当性評価

実験プロトタイプでは、付加手続きなどを利用して複雑な推論を行っていたが、実用システムでは推論機能を整理し、知識ベースの構造と整合性のとれた処理を実現している。また、1件あたりの処理時間は5分程度におさえられており運用上問題がない。

(3) 結果の妥当性評価

実運用に入った後、システムが提示した原因の中に真の原因があった割合は約8割であった。この結果は、現場からはなれたところでのコンサルテーションであることを考慮すれば要件をほぼ満たしていると言えよう。一般に、診断型システムでは解の網羅性を追求することは困難である。深い知識に基づく推論や、事例ベース推論などの技術の発展が期待される。

(4) 要求仕様に対する達成度評価

プロフィールでも述べたようにエレクトロニクス部品の診断をシステム化することに対しては大きなニーズがあり、また、経験的な知識を必要とすることや知識の保守が重要であることなどからエキスパートシステムのアプローチが適切であった。

要求仕様については開発プロジェクトに利用者も参画していたので、プロトタイピングの各フェーズで確認を行いながら開発が進められた。ただし、仕様の文書化については従来システムの開発と比較すると充分とは言えず、今後の開発においては留意すべきであると考えられる。

(5) 知識ベースの保守性評価

知識ベースは図3.9-2に示したように機能によってモジュール化されている。また、各知識は年式や型式によって少しずつ異なる部分があるため階層化により理解性を高め保守を容易にしている。さらに、各知識を編集するためのエディタと簡単な整合性チェック機能などが整備され、現在では保守作業は利用者側で実施されている。

(6) マンマシンインタフェースの妥当性評価

現在はシステムの利用者が限定されているために、通常の文字による対話で充分である。今後広く展開する場合には出力の視覚化などの機能を充実させていく必要がある。

(7) システムの総合評価

本システムは膨大な製品情報を診断知識として整理し活用するシステムとして特徴づけられる。導入の効果としては、①診断業務の生産性向上、②診断知識・市場情報の共用化・標準化、③きめ細かなサービス対応、の3点があげられる。今後、診断型のシェルとして整備することが計画されており、異なる対象への適用をとおして評価・改良を進めたいと考えている。

【参考文献】

〔温井〕 温井、大鹿、福井：エンジンエレクトロニクス技術相談システムの開発、自動車技術、42-9、(1988)。

〔大鹿〕 大鹿、福井、宮後：エンジンエレクトロニクス診断エキスパートシステムの開発、トヨタ技術、38-2、(1988)。

3.10 知識ベースによるコンピュータシステム／ネットワーク自動運用システム(SPARK)

3.10.1 プロフィール

3.10.1.1 システムの目的と期待効果

現在のメインフレームを中心とするコンピュータシステムおよびそれに付随するネットワークは複雑化の一途をたどっている。大規模なものは中央に十数台のCPUを配置し、それぞれOSやサブシステム、適用業務プログラム等のソフトウェアを稼働させてお互いに通信しており、数千台のDASD装置・テープ装置を接続して大容量のデータを入出力している。さらにネットワークには関連他社／自社のコンピュータシステムや端末／ワークステーションが数千台から時には数十万台接続されて使用されている場合も珍しくない。このようなシステムでは安定した運用が最も重要であり、金融機関／公共機関等の公共性の強いシステムではその故障・誤動作が社会的な影響を及ぼすため高信頼性・安定稼働が絶対的な要件とされている。

しかしながらこれらのシステムを運用するのは相変らずオペレータと称される専門家に大部分を頼っている。幾つかのOSやサブシステムにて自動運用機能がみられるもののその自動化レベルは低く、システムやネットワークの障害においてはその回復をほとんどオペレータの判断・手操作に頼っているのが現状である。オペレータの職務はその重要性にもかかわらずシステムの定常運転時には単純作業の繰返しとして不当に低く見られており、その業務の厳しさと需要の急増から人員の確保が年々難しくなっている。

これに対し今までもコンピュータおよびネットワークの自動運用システムの構築が試みられてきたが、あくまでも定常運転時のスケジュールに沿った操作を中心としており、コンピュータシステムの異常時の対応や故障の回復、さらには資源の有効利用等の分野において実システムで使用に耐えられるものはほとんどないと言ってよい。

今までの自動運用システムは運用知識のシステム化に従来方式の手順言語を使用したものがほとんどである。対象システムからのメッセージ出力等のアトランダムな事象発生によりそれに対する対応をプログラムしたモジュールが実行されて操作コマンドが自動入力される。従ってプログラム化を進めていけばどんな対応も可能であるが、対象システムの構成・動作が複雑になるにつれて

プログラムは飛躍的に難しくなる。障害回復等の高度な自動操作をプログラムしようとする単純なケースは別としてほとんど不可能になる。さらに対象システムは変更が日常のように行なわれており、技術革新も激しくプログラムはすぐに陳腐化する。自動運用システムは24時間365日稼働が要件であり、プログラムの更新・保守は容易ではない。

このような背景をもとに自動運用システムにおける諸問題をエキスパートシステム構築の諸技術により解決しようとした試みが本論でのべる「知識ベースによるコンピュータシステム／ネットワーク自動運用システム（SPARK）」である。SPARKは今までの自動運用システムおよび、物理操作を除いたオペレータの仕事を完全に置き換えることを目的としている。SPARK（Super Powerful Automated System/Network OpeRation System by Knowledge Base s）はその誤動作の影響の大きさからHeavy Dutyな実システムである。SPARKの直接の利用者はDP部門の専門家またはオペレータ自身である。

3.10.1.2 システムの概要

（1）システムの機能と構造

SPARKは次の様な機能を持っている。

- ・コンピュータシステム（複数）／ネットワークの自動開始・終了
- ・スケジュールに沿ったタスク／ジョブの自動実行
- ・コンピュータ資源使用・適用業務実行の安定稼働監視
- ・障害時の問題判別・自動回復
- ・ネットワークの障害発生に対する診断機能
- ・ネットワークの障害発生に対する回復ガイダンス機能

SPARKでは、この様な機能をオペレータの操作に関するルール、およびシステム／ネットワークの構成データベースとして実現している。基本的な推論方式としては単純なRETEアルゴリズムによるプロダクションシステムであるが、文脈（状況）に沿った操作の相違を表現するために状態遷移ネットワークがルール内に表現できる様にしている。ネットワークにおける障害の診断のためにトポロジカルなグラフサーチを使用している。またその回復手順を組立てるために過去の事例を使用する事例推論を多用している。

S P A R Kの要件は次の様になる。

- 対象システムとしてはメインフレームシステムのハードウェア／ソフトウェア／適用業務およびネットワークに接続する全ての機器・装置・施設を含むこと。
- オペレータが判断・実施しているタスクはすべて知識ベースとして実現できること。
- オペレータの操作を完全に代行することを目標にしているため誤動作は許されない。100%の正解率を持つこと。
- 50～100メッセージ／秒の入力にも耐えられる実時間性があること。
- 知識ベースの開発・保守作業生産性に飛躍的な向上があること。
- 24時間365日稼働を原則とし、システム稼働中に知識ベースの更新・保守が可能であること。
- 過去に実施した障害回復の手順が使用できること。

知識ベース構築の際に基となる知識源は個々のD P部門によって利用できるものが異なる。操作マニュアルが整備されている場合はある程度直接そのマニュアルよりルールを作成することが可能である。しかしかなりのケースはマニュアルはあるものの現状を反映していなかったり、オペレータの頭の中にだけ存在する知識のみであったりする。さらに対象システムの構成に関する知識も同様であり、この場合はD Pの運用担当者が相手となる。障害の回復に関する知識については複数のソースがあり、機器メーカーから提供される問題判別の手引類、オペレータの経験、D P部門のマニュアル等がある。

(2) システムの開発者と開発手順

S P A R Kは知識ベースを開発すればどの様な対象システムでも制御可能なコンピュータシステム／ネットワーク自動運用に関するアプリケーションシェルである。アプリケーションシェルは日本アイ・ビー・エムと住友金属工業の共同で1988年に開発された。シェルには知識ベースを開発するためのいわゆる知識獲得ツール群と実際に知識ベースを駆動するドライバが含まれる。

シェルの開発には住友金属工業のD P運用担当者が多数参加して彼等の運用知識が表現し易いように設計された。S P A R Kを使用して自動運用システムを

構築するステップは次の様な概略になる。

①計画段階

S P A R Kでの開発経験者を混じえて、D P運用責任者と担当者の中で運用上問題になっている部分および効果分析を行ない、どの範囲を自動化するかを決定する。さらにその運用知識はどの様に採集するかを決定する。

②開発段階

開発担当者が現在の運用知識を分析し、ルールと構成データベースとしてシステム化を行う。このステップはしばしばオペレータや運用担当者が自ら実施する。S P A R Kテストドライバーを稼働させてシミュレーションテストを行ってゆく。

③運用段階

S P A R Kドライバーを稼働して実際にシステムを自動運用する。ネットワーク障害の診断・回復については常時、その他については指定により推論過程がログとして記録され、妥当性検証とともに積極的に次回の推論において参照される。

④更新・保守段階

S P A R Kはシステムの本番稼働中にダイナミックに知識ベースを更新・保守することが可能である。本番システムとテストシステムを平行稼働することもでき、テスト済のルール／構成データをダイナミックに追加する。更新・保守は開発担当者またはオペレータである。

(3) システムの動作環境

S P A R Kシステム開発における動作環境は次のとおりであった。

- ①対象ハードウェア I B Mメインフレーム3090
- ②使用メモリ量 測定していない。通常のメインフレームアプリケーション開発と同等であると思われる。
- ③使用ファイル量 使用メモリ量に同じ。
- ④O S I B M M V S / X Aを使用。
- ⑤処理方式 S P A R Kドライバーはコンパイラで開発された。
- ⑥言語／ツール P L / Iをベースとしたエキスパートシステム構築ツールのナレッジツールを使用。

デリバリにおける動作環境は次のとおりである。

- ①対象ハードウェア I B M M V S / X A または E S A が稼働するメインフレーム
- ②使用メモリ量 個々のシステムにより変動が大きい。
- ③使用ファイル量 使用メモリ量に同じ。
- ④O S I B M / X A または E S A
- ⑤処理方式 知識ベースがダイナミックにインタプリトされる。
- ⑥言語／ツール S P A R K シェル

(4) システムの規模

S P A R K による自動運用システムは個々のケースにより規模に大きな差異がある。現在までに開発された最大規模のものは2500～3000ルールである。

3.10.1.3 問題のタイプとシステムの特徴

(1) 問題のタイプ

S P A R K のアドレスする問題領域はコンピュータ／ネットワークの自動運用すなわち制御型問題と言える。しかしながらネットワークの障害診断・回復部分は明らかに診断型の問題領域を取り扱っており、その他のタスクも内部的には様々な問題領域にアプローチしており、正確には複合型というべきかも知れない。

(2) システムの特性

本システムは制御型であり、表3.10-1のような特性を持つ。

表3.10-1 システムの特性一覧表

	制御型システム	関連性	実現性	コメント
定義	制御問題と、は、システムの状況を確認して、監視の状態で、移行する。	◎	◎	
問題の特徴	1. 時非安定実対 2. 遅線化精口時的 3. 遅型化精口時的 4. 遅型化精口時的 5. 遅型化精口時的 6. 遅型化精口時的 7. 遅型化精口時的	◎ ◎ ◎ ◎ ◎ ◎ ◎	○ ○ ◎ ◎ ◎ ◎ △	
基本タスク	1. シン診モ安オあ 2. シン診モ安オあ 3. シン診モ安オあ 4. シン診モ安オあ 5. シン診モ安オあ 6. シン診モ安オあ 7. シン診モ安オあ 8. シン診モ安オあ	◎ ◎ ◎ ◎ ◎ ◎ ◎ ◎	○ ○ ◎ △ ◎ ◎ ○ ○	
問題解決機能	1. 状態シ定制アディ 2. 状態シ定制アディ 3. 状態シ定制アディ 4. 状態シ定制アディ 5. 状態シ定制アディ 6. 状態シ定制アディ 7. 状態シ定制アディ	◎ ○ ◎ ◎ ◎ ◎	○ ○ 	

3.10.1.4 客観的な評価の可能性について

SPARKの妥当性あるは有効性についてはエキスパートシステムによらない一般的な制御システムと同じ評価方法が使用できるであろう。対象システムの安定稼働についてその損失の大きさと回数により他のシステムと比較することができる。あるいは生産性の向上について幾つかの評価が定量的に可能である。評価項目には次のようなものが考えられる。

- 一定期間のシステムの障害・誤動作の回数とその損失の削減率
- オペレータ／運用担当者の削減数
- 単位時間のトランザクション／ジョブ処理量の向上率
- 自動運用システムの開発期間・コストの削減率

3.10.1.5 今後の課題

SPARKは現在幾つかのDP部門で実際に使用している実運用システムである。日本アイ・ビー・エムはSPARKシステムをベースにSANOASとよばれるソフトウェア製品を1989年より売り出した。現在は大型規模のシステムユーザを中心に使用されている。今後より多くのユーザーに展開するには次のような課題を解決してゆく必要がある。

- 小規模ユーザのためのコンパクト版の開発
- より一層の高性能シェル化
- オブジェクト指向データベースを利用したより高度なモデリング機能の充実
- 知識獲得時におけるPWS機能をフルに使用する、より生産性の高いツールの開発
- 知識ベースの一貫性・妥当性チェックをサポートするツールの開発

3.10.2 評価結果

(1) 知識ベースの妥当性評価

S P A R Kにおける知識ベースの妥当性評価とはシステム全体の運用実績に他ならない。あるS P A R K使用者においてはコンピュータシステムの全体障害が皆無で一年以上使用しており、システム全体の開始/終了時間を50%以上も短縮している。障害時の自動回復においてもかなりのケースをカバーしており、システムの妥当性はかなり高いと思われる。

(2) 問題解決・推論方式の妥当性評価

問題領域は完全な事象駆動処理であり、しかも多種多数のサブシステム・部分がアトランダムにメッセージを出力する中で正確な状況判断が必要である。R E T Eアルゴリズムによるプロダクションシステムの採用はモデルの更新とアクションの決定に有効な問題解決の方法を与えている。あるS P A R Kユーザでは以前に従来型手順言語で自動運用システムを開発した経験がありそれをS P A R Kにより完全にスクラップ&ビルドしており、以前に2万5千行のコーディング量があったのを1,200ルールに削減している。さらにネットワークの障害回復に使用した事例推論は、問題領域における同一障害の繰返しの傾向に着目した方法であり、未経験者による専門家並の問題解決を実現している。

(3) 結果の妥当性評価

S P A R Kにおいては知識ベースの妥当性と同等である。

(4) 要求仕様に対する達成度評価

S P A R Kシステム開発段階における機能目標を完全に満足しており完成度は高い。

(5) 知識ベースの保守性評価

実時間性の問題領域において高性能と保守性のトレードオフは開発当初からの重要問題であった。すなわち高性能を追及すればコンパイル方式の方が有利ではあるが、保守性を追及すれば絶対にインタプリト方式にすべきである。結論としてはルールを関係D B内に持ち、推論機構を持つドライバーが必要に応じてルールを人力できるようにすることによって解決している。複数のドライバーが平行に推論を進めることにより高性能を実現している。さらにテストドライバーの開発により本番システムとの平行テスト、および知識ベース更新の自動スケジュールによるダイナミックな本番移行も可能である。

(6) マンマシンインタフェースの妥当性評価

S P A R Kは主たる利用者がD P部門の専門家であり、開発時点での知識獲得が容易なように設計されている。もともとの運用知識源はマニュアルであり、オペレータの知識である。従って一般に用いられている手順フロー図から容易にルールが作成できるようにルール形式を設計してある。開発の生産性をより高め、さらには検証が容易になるように直接手順フローを知識獲得セッションにより保管・検索するようなシステムが今後の課題である。実行時におけるマンマシンインタフェースについては自動制御という性格からあまり考慮されていない。障害回復時にはより視覚的なシステム／ネットワークの表示が望まれる。

(7) システムの総合評価

S P A R Kは一部の物理的操作を除きほぼオペレータの仕事の大部分を実現している。その意味では総合システムとしての完成度は高い。しかしあまりにも大規模なため小規模ユーザにとっては導入効果が明確でない場合もある。そのためにより一層の高性能化とともにコンパクト化をはかる必要があるだろう。障害の自動回復についてもモデル表現の高度化による、よりレベルの高い自動制御が今後の課題である。知識ベースの一貫性・完備性をチェックできるサポートツールも整備していかなければならない。一部ユーザはすでに試行しはじめているが、コンピュータネットワークの大規模化の傾向からも分散型の協調制御も今後の研究課題である。

4. エキスパートシステム評価の文献調査

4. エキスパートシステム評価の文献調査

海外におけるエキスパートシステムの評価に関する動向を把握するため、幅広く文献の収集を行った。その中から、本プロジェクトで特に有用と判断した14文献について詳細な調査を行ったので、その内容を以下に紹介する。

なお文献の分類は下記の項目に従う。

- 開発方法論全般 (Methodological Basis)
- 検証と妥当性検査 (Verification and Validation)
- 評価 (Credibility and Evaluation)
- 評価手法及び評価技法 (Methods and Techniques)
- 統計的手法 (Statistical Tools)
- テストの自動化 (Automation)
- 評価の事例 (Experience)
- エキスパートシステムのライフサイクル問題 (Life Cycle Issues)
- その他 (Others)

4.1 エキスパートシステムの失敗の原因

タイトル:	Why Expert Systems Fail
著者:	BELL, M.Z.
出典:	Journal of the Operational Research Society, Vol.36, No.7, pp.613-619, (1985)
分類:	Methodological Basis
キーワード:	開発フェーズ、失敗の原因

(1) 概要

エキスパートシステム開発が失敗する原因のいくつかは従来型のシステム開発の失敗の原因と同じものであるが、次のようなエキスパートシステムに固有の問題もある。

- 対象としている問題が理解できない、あるいはされないものである。
- 常識、知識の限界についての知識、が必要とされる。
- テストができない。
- 信頼されない。

(2) はじめに

エキスパートシステムの開発はフェーズで明確に分けられていると言うよりは、インクリメンタルな開発手順になるが、内容的には通常のソフトウェア開発と同様以下の6つのフェーズを含む。

- ① 分析 (analysis)
- ② 設計 (design)
- ③ 開発 (implementation)
- ④ テスト (testing)
- ⑤ ユーザの受け入れ (user acceptance)
- ⑥ 保守 (maintenance)

以下で各フェーズごとに失敗の原因を検討する。

(3) 問題の分析と知識の抽出

専門家から必要な知識 (expertise) を得ることがエキスパートシステム開発の必要条件である。ここでの問題は以下の通りである。

① 専門家が多忙で時間がさけない。

知識獲得に多大の時間を要するため、多忙な専門家が協力するのは無理である。

② 専門家が彼のアイディアを説明できない。

専門知識の内容を説明するのは難しい。専門家自身も彼が得た結論の道筋を理解していなかったり、実際に用いている問題解決方法とは異なった説明をすることもある。

③ 専門家が知識抽出に協力的でない。

専門家がエキスパートシステム構築に反対だとすると、知識は得られない。このため、専門家は開発者と同じ立場に立ってもらうことが必要である。

④ そもそも専門家が存在しない。

必要な知識が存在しない分野もある。音声認識などがその例で、我々すべてが音声認識の専門家とも言えるが、だからと言ってマイクロフォンからの波形を見て我々に音声認識のプロセスが説明できるわけではない。

(4) 設計と開発

数少ないケースに対してのみ有効であるようなシステムは簡単に構築できるが、多くのケースに有効なシステムの開発は難しく、さらに、想定されるほとんどのケースを扱うことができるようなシステムを構築することはほとんど不可能である。

(5) 常識と閉世界仮説

常識の欠如はエキスパートシステムの大きな問題である。エキスパートシステムは事前に言われたことだけを知っているが、全ての事をあらかじめ言うておくことは不可能である。

(6) 知識表現の選択

知識表現の選択に関する中心的な問題は以下の3点である。

① 知識表現言語の機能が十分でない。

②知識表現言語の機能が十分すぎる。

③知識表現言語の形式が専門家の知識の形式に合わない。

対象としている問題の特性に合致した知識表現手法を選択することが重要である。

(7) テスト

エキスパートシステムのテストが困難なのは以下の理由による。

①システムが異常な事象を扱うように設計されていること。

現実には発生させにくい異常事象はシミュレーションで検証するしかない。しかしシミュレーション時にエキスパートシステムを構築したのと同じ誤った仮定をしていない保証はない。

②システムがカバーしている領域が広すぎる。

③システムのテストに費用がかかりすぎる。

石油市場の取引システムを開発したとする。この場合売り買いの結果が市場そのものに影響するので、実際に売買してみないとシステムのテストができないが、失敗した場合あまりにも高くつく。

④システムのテストに時間がかかりすぎる。

適切なテストケースを生成するだけで多大の時間を要することがある。

⑤正しい解が何かわからない。

医療の診断問題などでは何が正しいかについては専門家の間でも意見が別れることがある。

以上のような場合は何れにせよ網羅的なテストは不可能なので、何らかの妥協が必要である。これについては次のような手法がある。

①システム構築のためのデータとテストのためのデータを分離する。

システムのテストに使えるデータのうちの90パーセントをシステム開発に用い、残りの10パーセントはテストのためにとっておき、それでシステムのエラーの頻度を見積もる。全てのデータを用いないことによってシステムの性能が落ちることがあり得るが、実際にシステムがどの程度使えるかを知るためにはこの方がよい。

②システム開発者とテストのためのデータ作成者を別にする。

システム開発時とテストデータ作成時に同じ誤った前提を持ち込むことを

避けるために、この2つを異なったグループで実行した方がよい。

③ システムの取った行動の正当性を別の専門家が判断する。

システムに知識を提供した専門家と異なった専門家が結果の正当性を判断する。ただし、これはシステムが正しいことを示すのではなく、単にもっともらしいということを示すものである。

(8) ユーザの受け入れ

ユーザの受け入れ時の評価はシステムの性能の評価と異なり、次のような人間的な要素が客観的な評価を妨げる。

① 信頼の欠如

システムはエンドユーザからの信頼を得なければならない。このためには説明機能が重要であるが、リアルタイムシステムなどの場合、処理の時間的な制約によって説明が不要、不可能な場合もある。ユーザと協調して問題解決に当たることができるような場合にはシステムが常に完璧である必要はない。

② 信頼しすぎ

一般に人々はコンピュータから出てくる結果を過度に信頼する傾向がある。ユーザは本来エキスパートシステムにもできる仕事の一部を負担すべきだし、こうすることによってコンピュータを過度に信頼しすぎる危険を軽減することができる。

③ 法律上の責任

エキスパートシステムの結果によって誰かが被害を受けたり、損害を被った場合の責任は誰にあるのだろうか。法律上の責任は特に医療システムの場合に大きな問題を生ずる。

④ 仕事の満足度

誰も自分の仕事がコンピュータで置き換えられたり、また仕事の重要性や面白さが減じられたりすることを好まない。システムは人間に替わるることではなく、彼を助けるものでなければならない。そうすることによって、人間を「単なる機械には難しすぎる」問題に専念できるようにするものでなければならない。

(9) 保 守

- 知識の追加は大幅な書き直しを意味するのだろうか。
- 所得税のためのシステムに対して、所得税法の変更をしたらどうなるだろうか。
- システムはどのくらいの間にユーザに最新の機能を提供することができるだろうか。

(10) 結 論

エキスパートシステムの開発で従来型のシステム開発と異なる困難な点は知識の抽出に関わる問題である。常識のからむ誤りを避け、思った通りの動作をするように知識を抽出しなければならない。

一番の問題は閉世界仮説の問題である。システムは自分が知らないことが何かを知らなければならない。少しの知識は、ある意味では全然知識を持たないことより危険である。

(11) おわりに

エキスパートシステムの分野はまだ始まったばかりである。したがって開発されたすべてのエキスパートシステムが現場で実際に用いられると期待するのは現実的ではない。多くの開発者がこれまでしてきたことに対するより良い方法も見つけられているので、今後はさらに良いエキスパートシステムも開発されるだろう。

4.2 原子力発電向けエキスパートシステム開発におけるV & V方法論

タイトル: Verification and Validation of Expert Systems for
Nuclear Power Plant Applications.
著者: Naser, J. A.
出典: EPRI Report, No. NP-5978, 116 pp., August 1988
分類: Methodological Bases, Life Cycle Issues
キーワード: Software V&V, Expert Systems, Nuclear Power Plants,
Life Cycle, EOP Tracking System, REALM Expert System

(1) 概要

本報告では原子力発電所向けエキスパートシステム開発におけるV & V方法論について述べる。報告の基本的な立場は、エキスパートシステムもソフトウェアシステムであるから、明確なライフサイクルをもち（従来の手法を含む）V & V手続きを適用できるというものである。

まずエキスパートシステムと従来型システムの類似点・相違点について分析を行い、従来のV & V方法論が適用できる範囲を明らかにする。次に、従来のV & V手法で対応できない場合に使う、エキスパートシステムに特化したV & V手法を示す。これと従来のV & V方法論とを統合することで原子力分野向けのV & V方法論が構成される。さらに、エキスパートシステムをV & Vを行う際の難しさに応じて6種類のタイプ分類し、各タイプに合ったV & Vのガイドラインを示す。

実際の適用にあたっては、この方法論を個々の問題向きに特化することも必要である。そこで、EOP (Emergency Operating Procedures) Tracking System と REALM (Reactor Emergency Action Level Monitor) という2つのエキスパートシステムに対して、このV & V手法を適用した結果を報告する。

(2) 序 論

Verificationとは、システム内部の正しさを保証することである。すなわち、開発者のプランを詳細化、プログラム化するときに生ずるエラーをなくす作業である。一方、Validationとは、システムを外部からみたときの正しさをさす。すなわち、システムが現場で運用されているときに正しい望みの結果がでるようになるための確認作業である。

エキスパートシステム技術はまだ新しく、特に原子力分野の応用は少ない。V & V問題もまだ研究課題である。しかし、エキスパートシステムは、従来システムと異なる面もあるが、似た面もあり、したがって、従来システムのV & V技術と組合せることによって成果を上げられる。

(3) ソフトウェアのV & Vに対する従来の接近法

従来型ソフトウェアのV & V手法は確立されており、そのためのツールや方法論も存在する。従来型システムはWaterfall型のトップダウン的方法で開発されることが多く、その手法は、Constantine, Yourdon, DeMarco等によって構造化設計法として発展してきた。V & Vもソフトウェア開発のライフサイクルに組込まれており、基本的に、要求を明確に定義し、それに対してV & Vを行うこととなる。

(4) エキスパートシステムの概要

エキスパートシステムは、知識ベース、データベース、推論機構、制御機構、利用者インタフェースから構成される。システムの開発はシステムを既存のツールを使って開発する場合は普通、推論機構、制御機構はValidateされたものとするが、カスタマイズした場合はそうではない。利用者インタフェースに対するV & Vは従来の方法で行うことができる。知識ベースを評価するには、Accuracy, Logical Consistency, Logical Completeness, Functional Completenessをチェックしなければならない。

多くのエキスパートシステムは図4.2-1に示す6つのタイプにわけることができる。

(5) オブジェクト指向プログラミングとエキスパートシステムのV & Vとの 関連

オブジェクト指向プログラミングによってシステムを開発した場合、

- ①オブジェクトで表現した知識と要求定義との対応がとりやすいためにValidationが容易になる。
 - ②ルール数が少なくなる。
 - ③継承機能を使うことでプログラミング作業が軽減される。
 - ④モジュラリティが向上する。
 - ⑤従来型システムのテスト手法が採用しやすくなる。
- という利点が生ずる。

(6) エキスパートシステム V & V手法の選択

エキスパートシステムは通常段階的な方法で開発されるために、開発初期段階では要求仕様が明確でない。しかし、V & Vを開発のライフサイクルに組み込み、開発の各段階ごとに要求定義が定まった時点で、それをテストに対する基準としていくことが必要である。従来型システムのV & V手法も使える部分には併用すべきである。また、何を何故テストするのか、誰がテストを行うのか管理するのかを明確にしておくことも重要である。

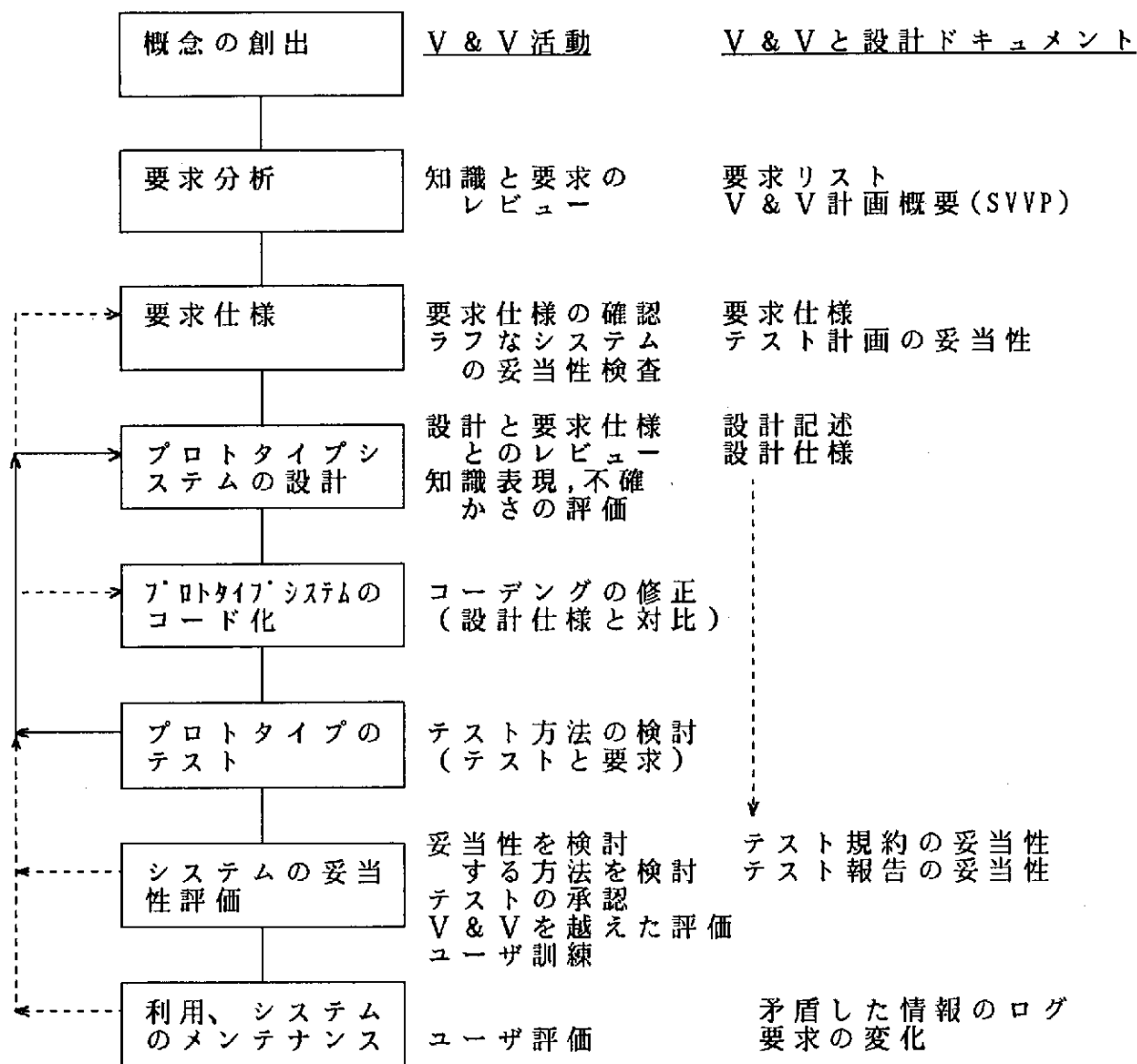
(7) 知識ベースのテスト

エキスパートシステム全体のテストを完全に実施することは困難であるが、知識ベースのテストとはその正確さと適切さを調べることである。このためには、以下のような項目がある。

- ①知識表現の適切さ
- ②知識の正確さ（専門家の知識との比較；根本的な基準との比較）
- ③ルールベースの一貫性と完全性
- ④推論方式の正しさ

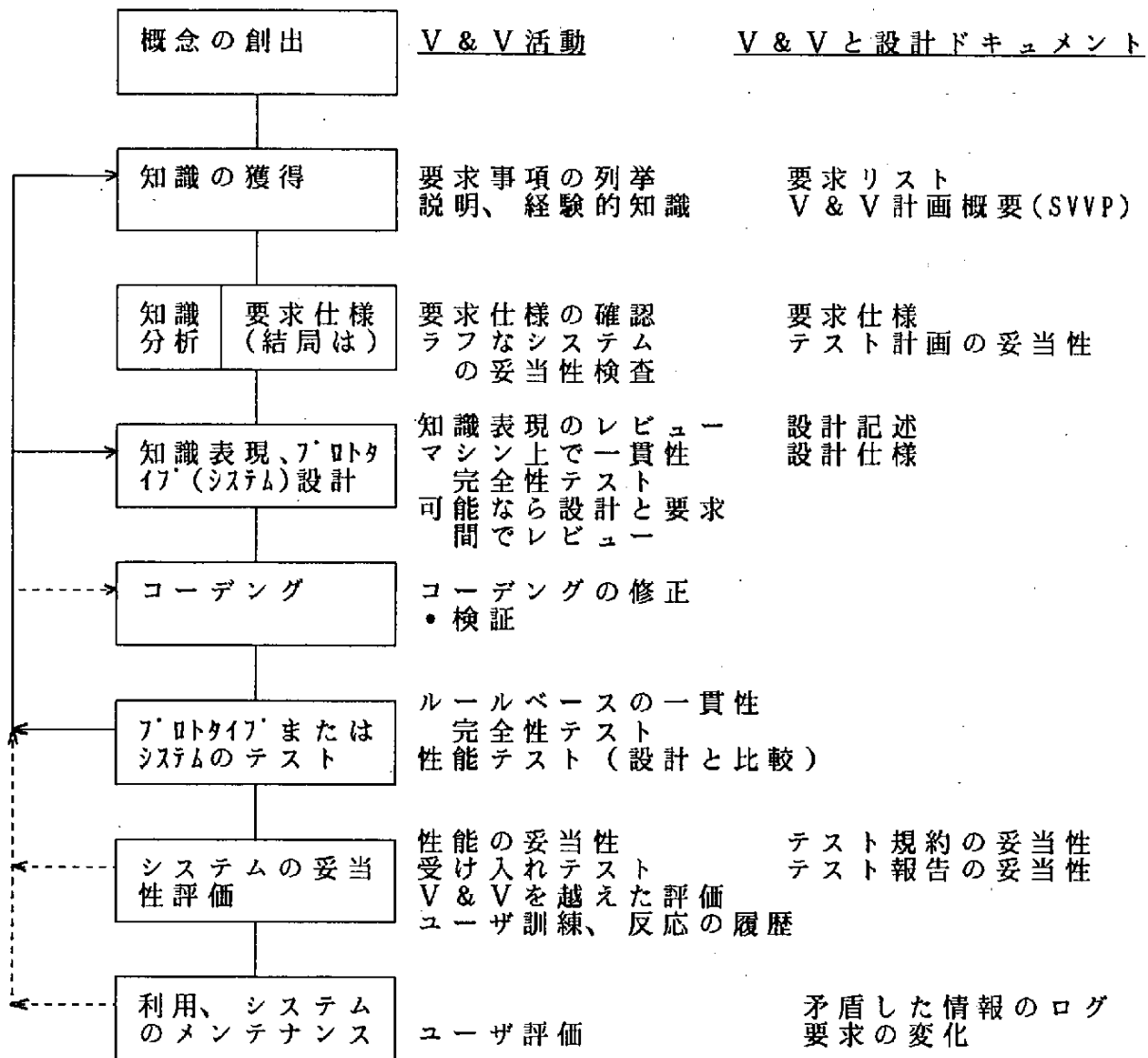
(8) エキスパートシステムのV & V計画

タイプ1、2とタイプ3、4のエキスパートシステムに対しては、図4.2-2のようなV & Vの計画を立てることができる。しかしタイプ5、6のエキスパートシステムに対しては、このような明確な計画はたてることが難しい。



(a) エキスパートシステム (タイプ 1、2) のライフサイクル
(妥当性ある成文化された知識が既にあるとき)

図 4.2-2 エキスパートシステムの V & V (その 1)



(b) エキスパートシステム (タイプ 3 & 4) のライフサイクル
(知識獲得を行うとき)

図 4.2-2 エキスパートシステムの V & V (その 2)

(9) EOP と REALM に対する V & V

EOP (Emergency Operating Procedures) Tracking System は、緊急時の原子力プラントの状況をリアルタイムで分析し、適切な緊急時操作手順を運転員にその根拠とともに提示するエキスパートシステムである。このエキスパートシステムはルールで記述され、タイプ 1 に属している。その V & V には、EOP に記述されている知識が正しく知識ベースに投入されているかどうかを確認すればよい。

REALM (Reactor Emergency Action Level Monitor) は、原子力プラントの異常事態を同定し、その重要度を評価する real-time Expert Assistance である。REALM はタイプ 3 のシステムで、ルールベース推論とオブジェクト指向プログラミングとを用いて実現されている。その知識ベースは、EAL (Emergency Action Level) という記号化された知識と専門家から獲得した知識とからなり、複数のコンポーネントから構成されている。そのためこのシステムの V & V 手法は非常に困難である。ここでは個々のコンポーネントごとに V & V を実施した。

(10) 結 論

本報告では原子力発電所向けエキスパートシステム開発における V & V 方法論について述べた。そのポイントは従来の V & V 方法論にエキスパートシステム独特の手法を組合せることであった。エキスパートシステムの V & V はまだ新しいテーマであり、今後の技術の進展に期待したい。

4.3 知識ベースシステムのV&Vアプローチ

タイトル: A Comprehensive Approach to the Verification and Validation of Knowledge-Based Systems
著 者 : Miller, L.A.
出 典 : Presentation at the Workshop on Verification, Validation and Testing of Knowledge-Based Systems, held at the IJCAI'89, Detroit, Michigan (19 August, 1989)
分 類 : Methodological Basis, Life Cycle Issues
キーワード: V&V, Testing Criteria, Testing Method, Life-Cycle

(1) はじめに

本稿では、知識ベースシステムの実用的なV & V (Verification and Validation) のアプローチを述べる。V & Vの定義はIEEE729-1983によると次の通りである。

Verification (検証) とは、開発フェーズのプロダクトを決定するプロセスが、前の開発フェーズで確定した全ての要求を満たしているかである。

Validation (正当性) とは、ソフトウェアの要求に対して最終的に承諾を得るプロセスのことである。

以下に述べる点は、長年かけて得られたV & Vに対する知見である。

- ① 知識ベースの安全性、信頼性の問題はV & Vに対する要求の強さに比例して対処すべきである。特に、組み込みシステム、リアルタイムシステム、センサ入力システム、他の複数のソフトウェアとインタフェースを持つ場合は、積極的にV & Vを行う必要がある。
- ② ただし、V & Vを緻密に行うとプロジェクト予算の半分を使うケースがあるなど、かなりのコストがかかるので、実際上は、出来なかったV & Vに対してどう保護するのが問題となる。
- ③ AIの開発者は通常ハッカーのような態度であり、従来のソフトウェア工学を生かしたよいプログラミングには耳を傾けたがらないが、知識ベースシ

システムの開発ツールに対しては大変興味を持っている。

- ④ 開発システムとデリバリシステムとのギャップを小さくするよう努めなさい。この間の差があれば最終版のシステムとすべくメンテナンスをしなさい。
- ⑤ テストは、最も問題の起こしがちな他のソフトウェアとのインタフェースに重点を置くべきである。様々なユーザの要求や知識ベースにはそれなりの方法を必要とするから、テスト方法だけがベストのV & Vではない。

以下に、知識ベースシステムの複雑さ、知識ベースシステムの要素、知識ベースシステムのV & Vのタイプ、テスト基準、テスト方法の適用、ライフサイクルについて述べる。

(2) 知識ベースシステムの複雑さ

表4.3-1では、知識ベースシステムの複雑さを、探索空間、不確かさ、決定プロセス、知識源、他システムとの連動、の5つの要因で示している。この要因の並び順序は、開発の困難さが増すに連れ上から下にいく。

表4.3-1 知識ベースの複雑さによる分類

	単純なシステム	複雑なシステム
探索空間	• 単純 / 小さい • 大きい探索が簡単	• かなり大きい • 不確定の要因あり
不確かさ	• なし	• 曖昧な推論 • 雑音、ミステーク、不確なデータによる推論
決定プロセス	• 後向き推論 • 前向き推論	• TMS • 制約指向推論 • 組み合わせ
知識源	• 明確な要求 • 成文化された知識	• 曖昧な要求 • 複数の知識源の引出し
他システムとの連動	• スタンドアロン型 • コンソールインタフェース	• 組み込みシステム • テータインタフェース • リアルタイムシステム • 割り込み型

(3) 知識ベースシステムの要素

知識ベースシステムは、知識ベース、シェル、推論機構、外部インタフェースの4つの要素で構成される。この各要素のサブ要素は、表4.3-2の通りである。

サブ要素をさらに手続き型（P）、事実型（F）、ルール型（R）の3つに分類する。

（4）知識ベースシステムのV & Vのタイプ

V & Vのタイプは以下のように分けられる。

- 手続きベースのV & V
- 宣言ベースのV & V
 - ルールベースのV & V
 - 事実ベースのV & V

（5）テスト基準

テスト基準を、機能基準、性能基準、一般的な基準の3つに分類し、その詳細な基準を、表4.3-3に示す。

表4.3-2 知識ベースシステムの要素

主要素	サブ要素	分類
知識ベース	ルールベース 事実データベース フレーム デモン 手続き、機能	R F F F P P
シェル (外部インタフェースのサブセット)	ユーティリティ ユーザインタフェース 知識表現	P P P
推論機構	パターンマッチャ 決定手続き ルールの順序制御	P P P
外部インタフェース	OS データベース管理システム I/Oデバイス(1-サインタフェース) プログラミング言語 他のアプリケーション ユーティリティ、機能、プロシジャコール	P P P P P P

表 4.3-3 テスト基準

分類	基準	分類	基準
機能基準	一貫性 完全性／網羅性 正確性 解の個数 説明 メタ知識	一般的な基準	理解性 交信性 アクセスコントロール セキュリティ 安全性 フォールトレラント/エラーハンドリング 頑健性 変更可能性/柔軟性/ 拡張性/増大性 自己記述性 ドキュメントの適切さ コスト 簡潔性 装置独立性 モジュラリティ 相互運用性 通信基準 読解性 データ基準 再使用性 単純性 構造的性 テスト可能性 追跡可能性
性能基準	実行速度 メモリ要求 I/Oハンドリング 性能（専門家が問題解決に要する時間,質,成功回数,経験のレベルとの対比） 効率性 データベースのアクセスタイム 解決までに要する時間 精密さ		
一般的な基準	移植性 保守の良さ 信頼性 可用性 利便性 運用可能性 ヘルプ／指導性		

(6) テスト方法の適用

テスト方法は、動的テストと静的テストに分類される。

a. 動的テスト

① ランダムテスト (Random test generation)

ランダムにテストデータまたはテスト項目を選択してテストすることである。

② 機能テスト (Functional Testing)

システムの機能要求仕様書に基づいて、各機能についてテストすることである。俗にブラックボックステストという。

③ 構造テスト (Structural Testing)

ソフトウェアの構造を考慮して、主として制御構造やパスを対象にテストする。俗にホワイトボックステストという。

④ 回帰テスト (Regression Testing)

システムまたはシステム構成要素の変更中に生じた欠陥を検出し、意図しない有害な影響が生じていないことを、または修正後のシステムまたはシステム構成要素が、その規定された要求事項に合っていることを検証するために再試験することである。

⑤ 説明にもとづくテスト (Explanation)

知識ベースシステムを実行して、ルールをトレースし付随して出力される説明に基づいて、ルールの実行順序がよいかどうかを調べることである。

⑥ シミュレーション (Simulation)

入力データのシミュレータはできるだけ現実の入力パターンを再現して生成する。特に組み込み型リアルタイムシステムでは有用である。

b. 静的テスト

① 異常検出 (Anomaly Detection)

ルールの異常検出には、構造的な異常検出と意味的な異常検出がある。

構造的な異常検出には、行き止まりノード、非参照な事実、重複ルール、包摂、クローズ内直接サイクル、クローズ間間接サイクル、直接的にわかる矛盾、間接的にわかる矛盾、関係ない述語の挿入がある。

意味的な異常検出には、不備なルール、事実の範囲の誤り、誤りのある値、非互換の値、対象が誤りのある部分的な関係がある。

② 構造的ウォークスルー (Structured Walk Throughs)

開発チーム内の他の要員が書いた設計書、コードを通して、技法、仕事のやり方、犯しやすい誤り、開発規格に対する違反やその他問題について質問、意見を述べ再検討する過程である。

③ 非公式なパネル形式による検査 (Informed Panel Inspection)

そのシステムが採用した推論機構について専門家を集めてパネルディスカッション形式で検査することである。

④ クリーンルーム方式による検査 (Clean Room Inspection)

今までは、開発者がデバックをしてプログラムを完成したが、本手法は開発者にはプログラムを実行してデバックすることを禁止して、静的な検査手法を用いてプログラムを検査することである。クリーンルームとは、半導体製造工場からきており、最初からゴミを入れないことを意味している。

⑤ 数学的検証 (Mathematical Verification)

プログラムがその仕様を満たすことを数学的に証明する形式的技法である。例えば、各プログラムにASSERTION文を入れてプログラムの定理証明を行うことである。

⑥ 記号を用いた実行 (Symbolic Execution)

実際の値を入力データとするのではなく、記号を入力してプログラムをシミュレートし、出力はこれらの記号を含んだ論理式または数式として表す検証の技法である。

⑦ フレーム表現が良形式かの検査 (Well-formedness)

良形式であれば知識の妥当性が増加し、知識の適用する範囲がより広くなるので良形式かどうかを検査することである。

⑧ 故障の木解析 (Fault-Tree Analysis: FTA)

システムの致命的故障あるいはシステムの使命達成上その発散防止したい、“好ましくない事象”を検出する図式解析法である。

⑨ 自動データフロー解析 (Automated Data-Flow Analysis)

この方法は、ある特定のデータに着目して論理的なシーケンスに基づいて実行する。データフロー解析をすれば、着目する変数がどこで定義され、どこで参照され、どの時点で値が未定義かがわかる。

(7) 知識ベースシステムのライフサイクル

知識ベースシステムのライフサイクルを図4.3-1に示す。

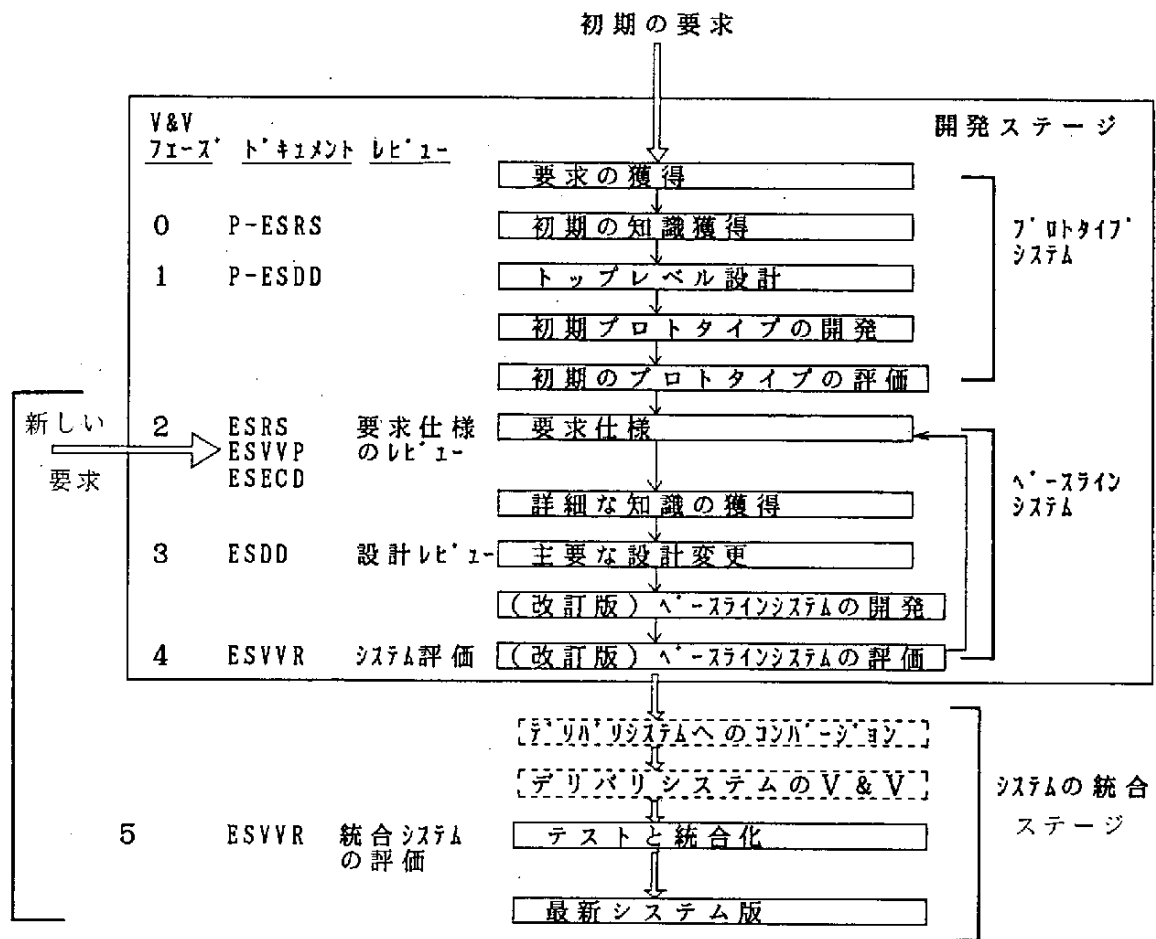


図4.3-1 知識ベースシステムのライフサイクル (提案)

4.4 エキスパートシステムの検証と妥当性検査

タイトル: Verification and Validation of Expert Systems

著 者 : Geissman, J.R. and Schultz, R.D.

出 典 : AI EXPERT, Vol.3, No.2, pp.26-33, (Feb. 1988)

分 類 : Validation and Verification, Life Cycle Issues

キーワード : チェックリスト、ライフサイクル

(1) V & Vとは

エキスパートシステムが使われるためには検証(Verification)と妥当性検査(Validation)が必要である。検証とはエキスパートシステムが正しく開発され、技術的なエラーがないか調べることを意味し、妥当性検査とはエキスパートシステムがユーザのニーズを満足することを保証することである。検証では上位レベルの設計が要求仕様の全ての項目を取り込んでいるか、詳細設計の中で記述されているように動くかを調べ、妥当性検査では完成されたプログラムが要求仕様書の中の機能を満足するかどうか、意図された目的のために使えるかどうかを検査する。

(2) 開発局面

エキスパートシステムの開発には4つの局面がある。

① 第1局面: 問題の決定

大きな目的はプロジェクトが真の要求を満たし、技術的に可能であることを保証することである。

② 第2局面: 初期プロトタイプ

初期プロトタイプの段階での大きな目的は拡張されたエキスパートシステムの技術的、経済的な可能性をすばやく示すことである。

③ 第3局面: 拡張プロトタイプ

初期プロトタイプと計画の通りの拡張プロトタイプの目的は動作の範囲として問題の複雑さ全部を扱えるところまで広げることである。

④ 第4局面: 製品版システム

製品版システムの目的は実行運用環境（P C，メインフレームなど）の範囲をカバーすること、重要なシステムの機能やメモリ使用などに関して最適化することである。プロトタイピング局面が明確にたどれるように記述されていたら、要求をまとめた仕様と知識ベース設計のドキュメントによって、製品版システムを検証することができる。

（３）エキスパートシステムの要素

エキスパートシステムには３つの処理モジュールが入っている。それは、推論エンジン、説明機能、そしてユーザフレンドリなインタフェースである。その他に知識モジュール（データベースと経験的知識が格納されている）、作業領域用のダイナミックメモリモジュールがある。

知識ベースにはある問題をどのようにして解くかということと、考慮しなくてはならない重要な事実の記述が入っている。知識は特有の知識表現で表現される。例えば、知識はIF-THEN ルール、事実、ゴールといった形式で書くことができる。

作業領域には、現在取り組まれてる問題に関する記述が入っている。推論エンジンは作業領域からの情報を解釈し理解するために、知識ベースから手続きを選択する。ユーザインタフェースによってユーザに入力情報を要求し、結論を表示させるという対話ができる。

（４）V & Vを実施するための問題点

V & Vを実施する際に、以下のような問題点がでてくる。

① エキスパートシステムの開発計画が曖昧である。

曖昧な目的のまま開発を開始すると、「何もないよりはましだから」といって、「実際にシステムが何をするのかはどうでもいい」という結論に達することがある。この考え方にもそれなりの理はあるだろうが、システムは何をするのかということが重要である。

② テスト可能な要求仕様を見つけることは難しい。

エキスパートシステムは非手続き的なアーキテクチャであるため、実行の流れに類似した形でのコードはつくらない。それゆえ、コードを試験して実行をトレースする技術は推論エンジン自体にはうまく当てはまるとしても非手続き的なアーキテクチャには当てはめることができない。

(5) 6段階のV & V方法論

上記のような問題を解決しV & Vを行なうため、以下に6段階のV & V方法論を示す。

〔第1段階〕初期プロトタイプを開発して、テスト可能な要求仕様を作成すること。

要求仕様及びプロトタイプは、プロトタイピングを繰り返すことによってスパイラルモデルと似たような発展をする。

〔第2段階〕明確に定義された問題解決戦略(Paradigm)の用語を使って設計すること。

コードレベルでできあがったシステムは、意味のある解析可能な構造をしているのかを確かめる。そのため既に確認されたパラダイムだけを用いて設計する。

エキスパートシステムにおける「構造化設計」とは、以下のことを意味する。

- 1つかそれ以上の知識処理パラダイムを正式に定義すること。
- 妥当で保証されたパラダイムを限定されたセットのみでしか使わないこと。
- 例外的に取り入れた全ての手続き型のコードに対して独立にV & Vを行うこと。

〔第3段階〕推論エンジンの正しさを証明すること。

推論エンジンの証明とは、推論エンジンが第2段階で規定した1つまたはそれ以上の知識処理パラダイムを実行するかどうかをテストする。

推論エンジンの保証とは次の通りである。

- 推論パラダイムを形式的に定義すること(ANSI規格のように)。
- 特定の推論パラダイムに関して、特定のエキスパートシステムツールの用語を用いて、どう表現されているかを仕様化すること。
- 抽象的に表現されたパラダイムに適したテストの仕方を決めること。
- 特定のツールの言語に適するようにテストを翻訳すること。
- 特定のパラダイムを保証するため、テストを実施すること。
- (満足されたルールの一覧、作業用メモリの内容、継承ネットワーク、最新のゴールといった)推論エンジンによって与えられる、全てのデバッグ用または説明の情報が、正しいかを確認して、後の検証の局面においてもあてが出来るようにすること。

〔第4段階〕検証を容易にする設計を行うこと。

エキスパートシステムがどのように知識を使うべきであるか、そして、プロトタイプの抽象概念にどのように基づくかという高位レベルの記述を行なう。

エキスパートシステムの上位レベルの設計の重要な要素は以下の通りである。

- 個々の要求をどのように扱うつもりでいるか。
- フォローすべき知識処理パラダイムを決定する（これらは保証された推論エンジンにサポートされたものの中から選ばれるべきである）。
- 信頼して処理を行える原理的な事実に関する知識を用意する。
- 全ての解法を小問題に分割し、1つの小問題から次の問題に移り、小問題同志のコミュニケーション、そしてこの構造がどのように表現するか（例えば、後ろ向き探索システムにおけるゴールのツリー）の方法を設計する。
- 必要なデータをどのように得るつもりでいるか（ユーザへの問い合わせ、デバイスからの読みとり、対話型データベースシステムを含む）をも含めた外界との全ての相互作用を設計する。
- 解法がより所としている基本的な諸仮定、それが妥当である範囲、境界条件を設計する（この情報は、例えば解析的な仮定が妥当である条件や入力の範囲といった手続き型言語の正しさを証明するのを助ける不変量の主張に相当する）。
- エキスパートシステムの中で、問題の現在の状態、すなわち、世界の状態が作業用メモリの中で一般的な用語でどのように表現するかを設計する。

〔第5段階〕知識ベースを検証すること。

知識ベースを以下のチェックリストを用いて確認する。

- 知識ベースが保証されたパラダイムを遵守しているかどうかを確かめよ。
- 小問題の構造を検証せよ。そして各々の小問題が互いに独立であるかを検証せよ。
- 静的、個別的な意味で知識が正しい、または適切であることを確かめよ。
- LISP、C、データベースシステムを例外的に呼び出すようなパラダイムの要素でない知識ベースの部分を識別せよ。
- 不変量と境界条件が成り立たなくなったときに知らせてくれる組み込まれた付加的なデモンルールを確かめよ。

〔第6段階〕正式な妥当性検査を行うこと。

妥当性検査の尺度を決定し、客観的にそれらの度合を決定する。

この妥当性検証のステップは以下の通りである。

- 妥当性検証の評価基準に対する尺度を決定せよ。
- 選ばれた評価基準に対する尺度の客観的な測定項目あるいは代用特性を決定せよ。
- エキスパートシステムが確実に正しく扱う必要のあるデータの集合の範囲を規定せよ。
- 当該問題に特化したテストケースのライブラリ及びシナリオとを開発し、知識ベースが修正された時に回帰テストを行え。ソフトウェアのテストと同様に試験者は正常な状況に対して適切な反応をするのかを確認し、システムエラーを引き起こさせるようにすべきである（良いテストとはエラーを見つけるテストである）。
- 自動的にシステムのテストが行われるようにせよ。
- システムの性能に関して、開発に関与していない第三者専門家に妥当性を検査してもらえ。
- エキスパートシステムを、既存のシステムや自動化されていないメソッドと、ある一定期間平行して使用し、その結果を比べよ。
- システムがたとえその時点で機能の一部分しか提供していなくても妥当性検査を早く開始し、システムが完成するにつれて、回帰テストを継続せよ。
- システムの能力に関する詳細な情報をもつこと。

（6）まとめ

V & Vの鍵は要求仕様をつくり込むことである。そしてプロトタイプはエキスパートシステムが直面しているぼんやりとした、明確に定義されていない問題について比較的特化された要求仕様書を得る方法である。

V & Vは開発過程で実施される多くのステップと並行して行われ、繰り返しの行われるプロトタイプ開発方法論にたやすくフィットする。

4.5 専門家間の概念構造の比較

タイトル: Comparing the Conceptual Systems of Experts

著 者 : Gaines, B.R. and Shaw, M.L.G.

出 典 : IJCAI-89, pp.633-638, (1989)

分 類 : Credibility and Verification, Life Cycle Issues

キーワード : 専門家概念、実体・属性グリッド、フレンストーミング方式

(1) 概 要

知識ベースシステムの開発は複数の専門家から知識を獲得することが多い。この場合、専門家によって概念や語彙の使い方が一致していないために暗黙の混乱を引き起こすことがある。この論文ではこのような問題に対し、専門家間の不一致が簡潔に理解できる方法を提案し、複数の専門家からの知識獲得に対する実際的なテクニックを述べている。

(2) 専門家概念の比較により生じる異論

専門家が持っている概念を実体-属性グリッドで表し、それを比較することによって知識を抽出することは、専門家の宣言的な知識構造を獲得するための強力な方法である。この実体-属性グリッドでは専門家から2つの型の情報を抽出している。一つは関連した実体間によって形成される特徴であり、もう一つはその特徴に対して使われる用語である。

ここで、専門家の概念から対象とする領域の関連した特徴への写像と、その特徴に対して使われる用語への写像とによって起こるいくつかの現象を考える。

① 一致……専門家概念が同一の特徴に対し同一の用語を割り当てる場合

② 衝突……専門家概念が異なる特徴に対し同一の用語を割り当てる場合

③ 照応……専門家概念が同一の特徴に対し異なる用語を割り当てる場合

④ 対照……専門家概念が異なる特徴に対し異なる用語を割り当てる場合

一致は専門家概念が共通していることを示している。衝突は対象とする領域において概念の適用が異なっているため、異論の可能性を示している。照応はある概念に使われる用語が異なっているため、これも異論の可能性を示してい

る。対照は専門家概念が共通していない、すなわち専門領域が異なっていることを示している。したがって、直接的に異論を導くことはない。

(3) 専門家概念の比較

専門家概念の構成要素となる各々の実体-属性グリッドは2極値で表されており、ファジー集合のメンバーシップの度合いで定義されているものとみなす。例えば、ある実体に対応する“不連続-連続”という属性があるとする、不連続から連続までの程度を5段階で表現している。

また、2つの属性間の同等の測定は次のように定義する。

$$\forall e (x \equiv y) = \bigwedge_e (x_e \equiv y_e) \quad \text{①}$$

本稿の例では同等に関して Lukasiewicz's operator を使用し、結合に関して Zadeh's sigma count を使用している。

単純化のために最初に2人の専門家の1対比較について述べる。(2)項で示した一致と衝突はある特徴に対して共通の用語を要求する。それは、実体-属性グリッドの交換に基づいて定められる。まず一人の専門家に対して別の専門家から抽出したグリッドを与える。そして、与えられた属性に対する実体の程度を求める。式①は2つの概念が照合することによって形成される特徴が良いかどうかの尺度として用いられる。しきい値は2つ設定しており、上の方のしきい値以上に照合する特徴は一致を示している。下の方のしきい値以下に照合する特徴は衝突を示している。2つのしきい値間に照合する特徴はあいまいに関連していることになる。

照応と対照については2つの専門家概念間から独立しているグリッドを抽出することによって測定している。式①を用いることにより、あるグリッドが他と最も良く照合するものを各々の特徴に対して決定する。先程と同様に、しきい値を2つ設定する。上の方のしきい値以上に照合する特徴は照応を示している。下の方のしきい値以下に照合する特徴は対照を示している。2つのしきい値間に照合するときはあいまいに関連していることになる。

一致、照応、衝突、対照への誘導は、グリッドにあるデータに基づいており、完全にアルゴリズムックである。この誘導はナレッジエンジニアではなくプログラムによって行われる。その根拠は視覚化された出力を通して専門家に明示される。ここで発生した異論については、各専門家が熟考し、お互いに議論す

ることになる。そして、必要があれば変更したり保留したりする。

本文ではこの過程について地理のマッピングを例に具体的に述べてある。

(4) 概念構造の比較方法

(3) 項では2人の専門家での比較方法を示した。しかし n 人には n の知識があり、1つの基本となる知識に対して $n-1$ 回の交換が生じるので、グループの規模によって線形的に大きくなる。この項ではグループでの知識獲得に対する実際的なテクニックを3つのフェーズに沿って述べている。

フェーズ1：領域の論議と具体化

- ① グループで対象とする問題を議論し問題領域を統一する。
- ② コンピュータ利用による知識獲得法を導入し、特定のタスクを定める。
- ③ 各々の専門家が実体-属性グリッドの初期設定を行う。
- ④ これらのグリッドを(3)項で説明した方法で比較を行う。
- ⑤ 比較結果についてグループで討議し、抽出する知識の目的等の改訂を行う。

フェーズ2：概念化とフィードバック

- ① 共通の実体集合を用いて、各専門家からグリッドを抽出する。
- ② このグリッドで再び比較を行い、その結果を各専門家に提示する。
- ③ このデータは照応あるいは対照の属性となる。

フェーズ3：交換と比較

- ① ある専門家概念が他の専門家概念の属性を用いて共通の実体集合を分類し、フェーズ2で抽出したグリッドの交換を行う。
- ② このデータは一致あるいは衝突の属性となる。
- ③ 専門家概念と専門家概念間の関係をグループで議論する。一致については問題はない。衝突は分析され、解決できないものとして注を付ける場合もある。照応は用語の違いによるものか、いくつかの現象を通しての相互関係によるものかとして解釈される。対照については、無関係、あるいは何人かの専門家の見地の関連したものとして解析される。
- ④ この分析は、クラス、属性、オブジェクトの定義やルール記述といった知識獲得のさらなるフェーズを通して行われる。

(5) 結 論

ここで提案した方法は複数の専門家概念の類似と差異を容易に表すことが可能である。また議論の焦点を合わすのにも使用することも可能である。グループで発生する異論は異なる用語や抽象レベル、不一致なもの等を分類出来るようにして問題解決を行なわなければならない。

グループによって貴重な知識体系を開発していくためには、各人の異なる見地を尊重し、ブレーストーミング方式で専門家同士が影響を及ぼしあいながら専門領域の概念的体系の十分な探求を進めていくことが必要であろう。個別に抽出した専門家概念を客観的に分析することにより、社会的圧力による無意味な干渉を防ぐことにもなる。

4.6 決定理論を用いた経験的知識の正当性の評価法

タイトル	: Using Decision Theory to Justify Heuristics
著者	: Langlotz, C.P., Shortliffe, E.H., and Fagan, L.M.
出典	: AAAI-86, pp.215-219, (1986)
分類	: Method and Techniques
キーワード	: 決定木, 感度解析, 期待利得

(1) 概要

本稿では、経験的知識(Heuristics)を『状況(situation) → 行動(action)』という形式で表現する場合の功罪(merit)を決定理論を用いて行う手法について述べる。具体的な説明をするために、MYCINシステムで実際に使われているルールを基に、その意味表現(Context)を用いた解析を行う。

また、経験的知識に対する従来の議論とを対比させながら論じる。人工知能と決定理論の融合がエキスパートシステムにおける知識の評価基準を与え、その能力を高め、エキスパートシステムが適応できる問題解決領域(problem solving domain)を広げることになる。

(2) はじめに

知識ベースにルールを登録する場合にルールの推論能力を確かめる方法やシステム性能に及ぼす影響度合いを解析するための標準的な手段がない。完全なブラインド評価の研究ができるまで、エキスパートシステム構築者はシステムが出くわすであろう全て(殆ど)のケースについて、その分野の専門家(domain expert)から集めた経験的知識が適切であるかどうかを検討しなければならない。

最善の行動(recommends an action)をとるための経験的知識の決定理由を作成するためには、その行動の影響に対する見込みと望みの両方を考えることが重要である。それゆえ、経験的知識の解析には人工知能と決定理論の融合が重要であると主張するゆえんである。決定理論は、確率(likelihood)と利得(desirability)の積として、次式のように表現される。

$$\text{期待利得} = \sum_i P(O_i) \times U(O_i)$$

1)

ここで、 $P(O_i)$ はプランを実行した時の i 番目の結果の起きる確率

$U(O_i)$ はプランを実行した時の i 番目の結果の利得

この期待利得を最大とする行動を選択することになる。

(3) 経験的知識の形式的な解析

議論のために、ここではMYCINの経験的知識のルールを引用して使うことにする。

IF: 1) 考慮中の治療法がテトラサイクリン(tetracycline)の投与で

2) 患者の年齢が8才以下

THEN: テトラサイクリンによる生体組織に対する治療は、適切でないとの強い確証(0.8)がある。

この経験的知識は幾つかのサポート知識(部分知識)の集合として表され、そのサポート知識によって評価される。この場合は、4つのサポート知識として表現される。

若年者に対するテトラサイクリンの投与

=> 発達中の骨では薬のキレート環(化合物)ができる。

=> 歯の変色が起きる。

=> 体の変調を起こす。

=> テトラサイクリンの投与は認められない。

最後の推論(結論)は、その前の推論に基づく推論としてなされる。これらのルールによる推論は、推論(ルール)の粒子の精度(知識の断片のサイズ)には係わらず、常に『状況→最善の行動』というチェーニングによって行われる。例えば、テトラサイクリンの投与を避けることが歯の変色の可能性を本質的に除外するのに有効であるが、治療効果の弱い他の薬剤を必要とすることになる。実際、テトラサイクリンは他の薬剤では効果が出ない場合や禁忌(治療法が望ましくない場合)の状態を除き、8才以下の子供には投与しないように医者の本には書かれている。言い換えれば、薬剤の投与は歯の変色というリスクと治療効果とのトレードオフの選択と言える。このトレードオフが薬剤投与をする場合の決定に対し非常に重要なことになる。

(4) 問題の定式化

この項では、MYCINのルール例を基に、前項で述べたトレードオフについて決定理論を用いて説明する方法について述べる。決定理論での問題の定式化を行うためには、次の3つの質問に答える必要がある。

- ①どんな代替案の可能性があるか？
- ②それらの行動が実行された場合、なにが起きるか？
- ③それぞれの結果の利得はなんであるか？

我々は、テトラサイクリンのルールをもとに実際の例に応用して実験してみる。それによって、エキスパートシステムを構築する場合に実際に使用できるように、結果を如何に一般化できるか示してみる。

実際の例として、8才の男の子で尿道排泄（尿道感染の可能性もある）の患者を考える。しかし、培養試験によってバクテリア感染の確証はない。この場合、この尿道炎(Urethritis)は培養試験では簡単には分からない生体組織に起因するもの(NSU: Non-Specific Urethritis)か非感染型の尿道の炎症が原因で起因するものかが考えられる。このような徴候に対し、若し大人の患者なら一般的に効果があるテトラサイクリンを投与すればよいので、悩む必要はない。しかし、この若い患者にはテトラサイクリンを投与するのか、他の薬剤を投与すべきなのかの決定を悩むことになる。ここにエリスロマイシン(Erythromycin)という薬剤がある。エリスロマイシンは、テトラサイクリンと違って、時々吐き気がするという副作用以外に歯が変色するような体にダメージを与える副作用はない。しかし、通常 of 尿道炎(NSU)には殆ど効果が無い。この問題の決定論的表現として定式化すると、薬剤投与の結果として患者は、次の四つの結果パターンが考えられる(stainingを副作用と訳す)。

- ①CURE/NO STAINING (治癒/副作用無し)
- ②NO CURE/STAINING (治癒せず/副作用有り)
- ③CURE/STAINING (治癒/副作用有り)
- ④NO CURE/NO STAINING (治癒せず/副作用無し)

これらを基に、決定木(decision tree)を書くと図4.6-1のようになる。

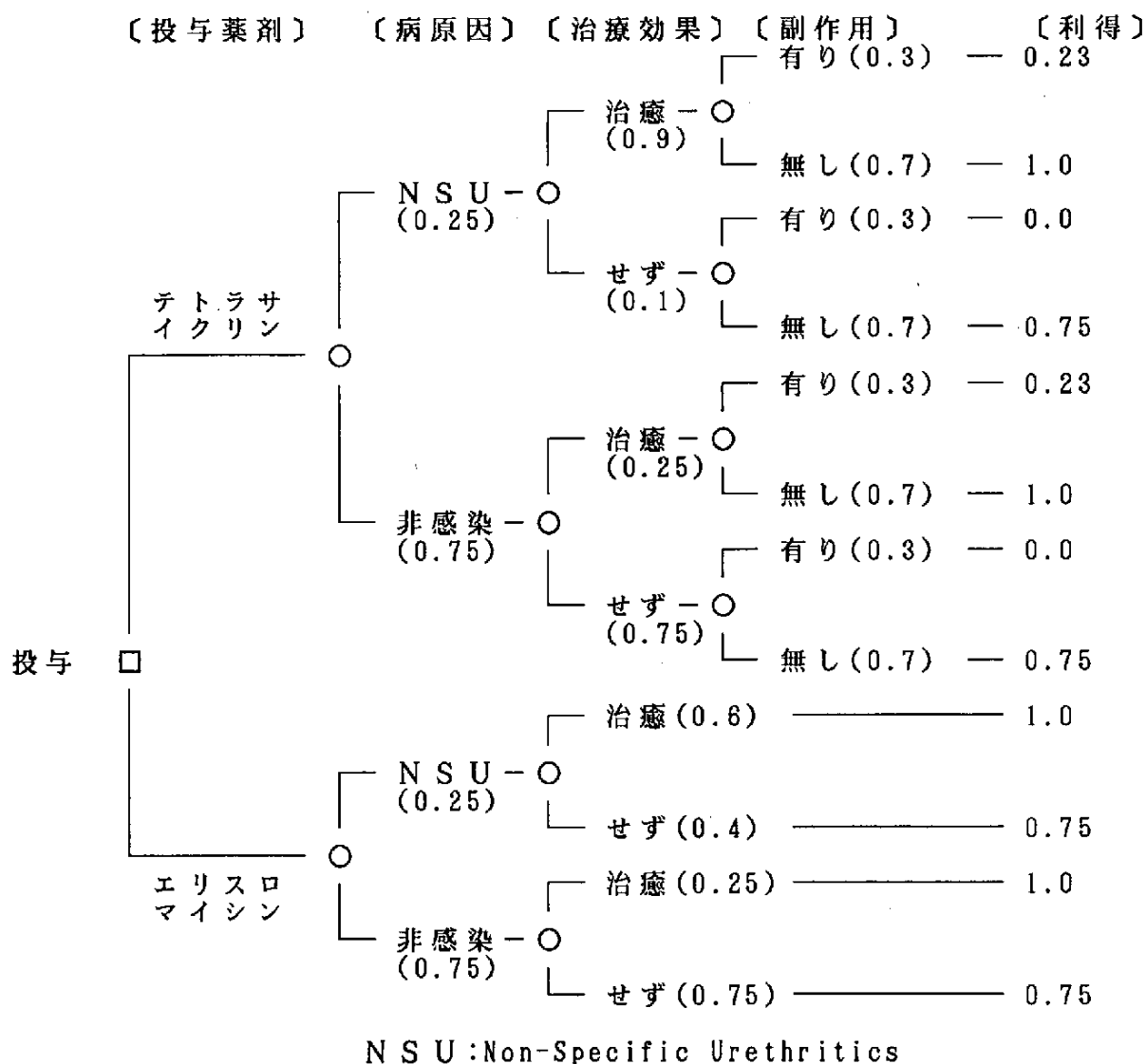


図4.6-1 決定木

(5) 確率と利得の査定

決定木における確率と利得の決定は、その結果が発生する確率とその結果となった場合の得られるであろう利益（利得）として決められる。この薬剤投与の場合は、薬剤投与によって副作用（変色）がなく治癒する結果（CURE/NO STAINING）の利得を1.0、副作用だけが起きて治癒しない結果（NO CURE/STAINING）の利得を0.0とする。副作用（変色）がないかわりに治癒しない結果（NO CURE/NO STAINING）の利得と副作用（変色）が起きるが治癒する結果（CURE/STAINING）の利得は、0.0～1.0の中間値となり、それぞれを選択する確率から、それぞれ0.75、0.23と決められる。

確率は徴候から推定される原因（病名）とその病気に対する薬剤投与の有効

性などから導かれる。この薬剤投与の場合は医者からそれらの情報を収集した。

テトラサイクリンとエリスロマイシンの投与に対する期待値は、式①を適用して以下のようになる。

$$\text{テトラサイクリンの期待値} = 0.63$$

$$\text{エリスロマイシンの期待値} = 0.83 \text{ (} 0.15 + 0.075 + 0.1875 + 0.421875 \text{)}$$

テトラサイクリンとエリスロマイシンの投与に対する期待値の差0.2は何を意味するのであろうか。

(6) 感度解析

ここで、考慮しなければいけないことがある。この事例の場合でも、各々の施薬によって変色の確率が変化し、それに伴って利得も変わるであろう。実際には、閾値(threshold)によって利得が量子化されて使われている。歯の変色の確率と期待利得との関係を図4.6-2に、モンテカルロシミュレーションによるテトラサイクリンとエリスロマイシンの期待利得の差を図4.6-3に示す。

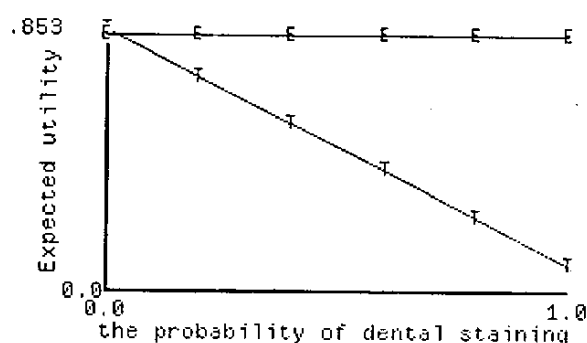


図4.6-2 感度解析の結果

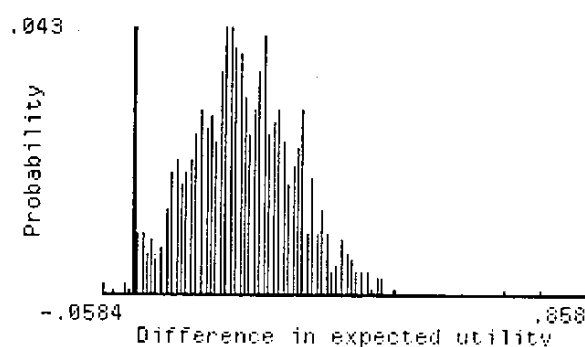


図4.6-3 薬剤による期待利得の差

(7) エキスパートシステムとの関連

細分化したルール(support rules)のシナリオだけでも幾つもあり、それらの評価が成されていないのに、MYCINは何故うまくいったのだろうか？ MYCINのルールが選択のための制約となっており否定的に使われているのに対して、モンテカルロシミュレーションの結果も、テトラサイクリンの投与の方が良い場合が非常に少ないことが裏付けられた結果となっている。また、実際の専門家との比較においても一致しているからである。

(8) まとめ

決定理論がエキスパートシステムの開発者にとって考えなければいけない非常に重要な道具であることを示してきた。決定理論が知識工学と結合することによって、決定理論を用いた手法は知識工学の基礎的役割を成すとともに、本質的なトレードオフ問題の解法や推論の説明昨日などシステムの能力を高める手法としても有効である。

経験的知識の解析に決定理論を応用する方法を提案して、実際の例でその様子を示してきた。この解析手法に必要な様々な情報は、従来のKEが行う経験的知識の通常の評価作業に示唆を与えることができるであろう。

4.7 エキスパートシステムの妥当性試験手法

タイトル: Validation of Expert Systems

著 者 : O'Leary, D.E.

出 典 : Decision Sciences, Vol.18, No.3, pp.468-486, 1987

分 類 : Verification and Validation

キーワード: 知識工学, エキスパートシステム, 知識ベース, 妥当性試験

(1) 概 要

ES 妥当性試験の1つの枠組みを提案している。この枠組みはESの特徴から導きだしたもので、9つの分類からなる。

また、他の従来のES妥当性試験枠組みとの比較を行うとともに、ES妥当性試験に関する今後の課題も述べている。

なお、本論文は以下の構成である。

- ① ES 妥当性試験の枠組みを明らかにするためにESの特徴を利用する考え方を述べる。
- ② ES 妥当性試験法の一般的な特徴を述べる。
- ③ ES 妥当性試験の枠組みを提案する。
- ④ ES 妥当性試験に関するコメントを述べる。

(2) はじめに

ES 妥当性試験に必要な事項のうち、従来のソフトウェア試験と異なるのは以下の項目である。

- 何を知っているか、何を知らないか、何を間違って知っているかを確かめること。
- 専門知識のレベルを確かめること。
- 特定の領域での理論的な基盤を持っているかを確かめること。
- 信頼性を分析すること。

そして、その他の従来のソフトウェアと同様な項目は以下の通りである。

- ユーザインタフェースの品質を確かめること。

- ・ドキュメントを妥当性試験すること。
- ・どんな記述言語、構築ツールが開発に適しているかを決めること。
- ・プログラミング（製造）の品質を分析すること。

（３）ＥＳ検査・試験の特徴

もし、ＥＳが従来の計算機ソフトウェアと同じ特徴を持っているなら、従来のソフトウェアと同一の妥当性試験の方法を適用すべきだが、以下の３つの側面で異なった特徴があるため、これら特徴に着目して妥当性試験の枠組みを決める。

①技術的な側面

- ・数値計算と異なった、記号処理にもとづいている。
- ・全知識の１割程度の知識が標準的な記述でなく、特殊な表現を必要とする（特別な機能を盛り込む）。知識の特殊な表現を適用しない場合には、このような問題は生じない。
- ・ＡＩ言語やＥＳ構築ツールなどを用いてＥＳの開発を行う。

②環境的な側面

- ・ＥＳは、従来のように結論に間接的に寄与するわけではなく、結論に直接影響を与える。
- ・経験的な知識は直接的で、断片的である。

③適用領域に関する側面

- ・科学技術計算と異なり単一の解のみを出力するわけではない。
- ・得られた結論の内容を評価する方法としては、他の問題解決手段とどのくらい結論が似かよっているかで行われ、結論の優先度を明確につける手段がない。
- ・取り扱う分野で定説がないため、複数の専門家全員が合意する結論になりにくい。
- ・よい推論過程から、常に良い結論が得られるわけではなく、その時点で得られる最良の結論が得られるだけで、他の情報が得られた後でも有効な結論であることを保証するわけではない。

（４）妥当性試験の枠組み

ＥＳ妥当性試験の過程は以下のような点で、従来のソフトウェア試験との違

いがある。

- ①従来のソフトウェア試験では、プログラミングのエラーを見つけることが主体であったが、ESの妥当性試験では何ができ、何ができないかを確かめることになる。
- ②ESではその記述言語・ツールが各種あり、それぞれについて少しずつ異なった試験方法が必要となる。また、試験方法はその実行環境にも依存する。
- ③ESの開発手法が従来のトップダウンやボトムアップとは異なっている。

ここでは上記から、以下に示す9つの分類でES妥当性試験の枠組みを提案した。

1. 知識ベースの内容の妥当性(content validity)〔手段で分類した〕
 - a. 専門家による知識ベースの内容の直接確認
 - b. 人間の専門家との比較(チューリングテスト)
 - (1)知識を提供した専門家との比較
 - (2)知識を提供した以外の第三者専門家との比較
 - c. 他のモデルとの比較

〔方法〕

1. a.の具体的な方法としては、周期的なレビューや推論過程のまとめ、知識ベース内容の直接確認などがある。

〔課題〕

1. a.の課題としては、専門家は計算機に慣れていない、人間の処理能力は大きくない、十分なリソースを必要とする、単純作業である、知識ベースが多いと多大な時間を要する、現状の技術ではできない部分も多い、などがある。

1. b.の課題としては、必ず個人のバイアスがかかる、定説の無い部分については第三者は確認が難しい、などがある。

1. c.の課題としては、もともとESは同じ問題に対して他のモデルよりも同等か、より良い結果を得るため、あるいは、従来のソフトウェアでは記述しがたい部分に適用しているため、数値計算処理などでは実現が難しい、

などがある。

2. 基準に対する妥当性(criterion validity)

a. 専門知識のレベルの規定

- (1)人間の専門家を基準とする
- (2)レベルを判定するテストプログラムを利用する
- (3)前もって定義しておいた応答の質を利用する

b. 知識ベースの基準

- (1)一貫性
- (2)正確性
- (3)完全性

c. 妥当性試験の基準の分類

〔課題〕

専門家とESとの専門度のレベルの規定についての違い、知識ベースへの基準の違い、どんな項目について妥当性試験すべきかが不明確、など各種の問題がある部分である。

2. a.の課題としては、専門家のだす結論が多くの条件の組み合わせからなるため、妥当性試験の基準と結論とを1対1に対応できない、応答の質を計ることが難しい、などがある。

2. c.の課題としては、一般的な妥当性試験基準を設定できないことがある。

3. 構成の妥当性(construct validity)〔どんな理論に基づいているか〕

〔課題〕

経験のみに依存する知識には適用できないことが課題である。すなわち、専門家が同じ理論にもとづいているか、という問題があるからである。

なお、一般的には、理論的な裏付けのないESは失敗するといわれている。

4. 客観性(objectivity)

- a. プログラマによる確認
- b. 独立な管理組織による確認
- c. 開発要求元、エンドユーザによる確認
- d. バイアスとブラインド
- e. 開発時に使用したテストデータとは異なるテストデータ

〔課題〕

4. a. の課題としては、プログラマは問題領域知識を持ってないことがある。このため、他の手法との組み合わせることが必要である。

4. b. の課題としては、モデルの作成者による確認は不適當であることがある。

4. c. の課題としては、開発要求元、エンドユーザが知識を有する専門家でない場合は不適當であることがある。

5. 経済性(economics, cost-benefit)

ES の導入効果や開発工数で定量化できるので有用である。ES の妥当性試験のコストは、どれだけ厳密に試験するか、どの範囲まで試験するかで決まる。

〔課題〕

システムの導入効果は、そのシステムが使われるかに依存するので、特に実験的なプロトタイピングシステムではこの観点での妥当性試験は難しい。

6. 信頼性(reliability, stability, ability)

- a. 感度解析
- b. 標準的なテスト問題による確認

〔課題〕

6. a.の課題としては、いろんな条件で多種の出力を得られるという現象が、E Sが良いためののか、または、E Sが不安定なのか、との切り分けが難しいことである。

7. 試験の誤差(systematic variance, experimental variance)

- a. 取り扱う問題の範囲を反映したテスト問題
- b. テスト問題の変化範囲
- c. テスト問題の数
- d. 第Ⅰ種の誤りと第Ⅱ種の誤り

テスト問題の範囲は、E Sが実際に取り扱う範囲にする必要がある。
統計的な妥当性試験には、テスト問題は多数必要である。

8. その他の誤差(extraneous variance)

- a. E Sの複雑さ
- b. 開発のライフサイクルでの位置(段階)
- c. 知識の特殊な表現に依存する部分(fixes)
- d. 試験判定者のいる場所
- e. 試験判定者が試験中に学習する効果

8. d.とは、実験室で試験を実施をする場合、E Sを比較すべき試験の判定者の環境は簡単に調整できるが、現場で実施する場合はE Sにとってより厳密な試験となるという違いがある。

8. e.とは、比較の対象となる試験の判定者が、試験に使用したテスト問題の操作やテスト問題のタイプから学習することである。学習により、試験の判定者がテスト問題の操作を変えて評価したりする。この問題は、テスト問題を経験した専門家と、E S操作の経験の少ない他の専門家との結論の差で決めることができる。

〔課題〕

8. a.の課題としては、ESの複雑度は規模で規定しにくいことである。

8. b.の課題としては、ES開発の初期段階では妥当性試験は重要ではないが、完成が近づくと妥当性試験は重要になってくる。すなわちどの段階でどの程度の妥当性試験をすべきか、などである。

8. c.の課題としては、経験的には1割程度が標準的な知識の表現でないため、完全な知識ベースを実現するために特殊な表現が使用される（特殊な機能がシステムへ組み込まれる）。これらをまとめて試験するために、特別な試験方法を開発することになるが、これら自体が、もともと特殊なため、一般化できにくいことである。

9. エラーによる誤差(error variance)

「エラーによる誤差」については、各種の定義がありまだ定着していない。

専門家のちょっとした思い違い、記憶違いなどで起こるもの全てを一緒にしたもので、ESの妥当性試験の枠組みとして「その他の誤差」に分類できるよう「エラーによる誤差」を少なくしようと試みてはいるが、まだできていない段階である。

（5）ES妥当性試験に関するいくつかのコメント

①蓄積済みの知識の内容の妥当性(content validity)

今後は知識ベース内の個別の知識に着目した場合にもコスト増加が少なく、また、他の専門家でもES全体としての性能を簡単に妥当性試験できるようなツールの出現が望まれる。

ESの妥当性試験は専門家に頼っており、また、ESが拡張性を持つことから、少人数の専門家が担当している。さらに、知識を提供した以外の専門家による試験では、その専門家が該当対象に対して異なった経験知識をもっている可能性があるため、結論に意見の相違をもたらすおそれがある。

いくつかのケースでは、少人数による妥当性試験を、他のモデルや標準的なES妥当性試験法で補っている。しかし、他のモデルを開発することは人間の専門家による試験よりもコストがかさむ。

② 基準に対する妥当性(criterion validity)

一般的には、ESと人間の専門家からだされた結論には同一の妥当性試験基準が用いられている。この方法ではESが人間の専門家よりもより良い結論を導くかもしれないという可能性は無視されている。通常、人間の専門家は全く別なモデルで業務を実施することは少ない。そこで、もしESが人間の専門家と全く同じように業務実施可能であるなら、妥当性試験基準を得る目的で別なESを開発することもありうる。

プロトタイプシステムの開発者は妥当性試験そのもので忙しいので、妥当性試験基準を分類することは難しい。なお、現在までに第三者の立場の妥当性試験者により効果的な妥当性試験基準を開発したとの報告や、テスト問題の難しさのレベルを規定することが、ESの問題解決能力を確認することであるとの報告もある。

しかし、ESの問題解決能力を決定する一般的な方法はまだない。

現在までの報告では、妥当性試験基準としてESのだす結果の正確さのみに着目していたが、完全性や一貫性に着目した報告もある。今後ツールはこれらの側面に焦点をあてて研究を進めるべきである。

③ 構成の妥当性(construct validity)

いくつかのESで理論的な構成を試みてはいるものの、意思決定モデルの効用を無視して、どのように結論が導かれたを確認するだけである。このアプローチの危険性はモデルが専門家固有のものであるため、一般化したり、他のESや人間の専門家との比較を行っていないことにある。

ES開発に効果のある方法は、モデル化は済ませた結論を導く過程の性質をすることである。

④ 客観性(objectivity)

現在一般的には、妥当性試験は第三者的な独立の妥当性試験者を設置していない。このため、妥当性試験作業に当該ESの開発担当者が支援をしており、バイアスがかかり、結論へ影響をおよぼしている。

⑤ 経済性(cost-benefit)

妥当性試験のための工数は定義できるが、妥当性試験の効果をはかるのは難しい。

⑥ 信頼性(reliability)

ESが開発段階であれば再妥当性試験のためのテスト問題を作成する必要

はないが、実用システムでは知識ベースを変更後はテスト問題で再妥当性試験をする必要がある。しかし、知識が追加されているので同じテスト問題でも別な結論を得ることがある。

⑦ 試験の誤差(systematic variance)

ESの試験に用いられているテスト問題の数が少なく、また、典型的な例についてだけの試験である。これは、テスト問題のパラメータの変化できる範囲が限られているためで、現実遭遇する極端な状況下や、あらゆる状況の変化の下でのESの動作を試験しているわけではない。また、プロトタイプシステム段階での妥当性試験では、内在している第I種の誤り、第II種の誤りや、テスト問題の数を考慮していないことに問題がある。

(6) 今後の課題

人間の専門家と同じ結論を得るための妥当性試験基準は何か、ESの専門度のレベルをどうやって規定する、が今後の課題である。

4.8 エキスパートシステムの検証及び妥当性検査

タイトル:	Verification and Validation of Expert Systems
著者:	Green, C. J. R. and Keyes, M. M.
出典:	WESTEX-87, pp. 38-43, (June 1987)
分類:	Method and Techniques, Verification and Validation
キーワード:	要求仕様書、エンジニアリング分析、相互作用分析、 一貫性分析、不確実性分析

(1) 概要

検証及び妥当性検査 (V & V: Verification and Validation) は、コンピュータプログラムがユーザ要件を満たしているかどうかを決定するのに使用される形式的な手法である。エキスパートシステム (ES: Expert System) の受け入れでの障害は、ES用のV & V方法論の欠如である。ESに対するV & Vの実施は、「確固たる文書類の欠如」、「テスト結果を評価するのに使用する手法の不十分さ」、そして、「V & V手法の開発を妨げる悪循環」のために、妨げられている。ESに対するV & Vを実施するのに受け入れられるV & V手法があれば、ES技術の受け入れを助長するであろう。

ESのV & Vに関する提案された最初の手法は、「要件の定義」、「検証」、「テストケースの準備」、「テストの実施」、そして、「評価」からなる。

V & Vの提案された方法論を洗練化し、顧客のESに対する信頼を向上させるためには、ESのサンプルに本方法論を適用する必要がある。ESのV & V実験の対象候補としては、顧客側でのESに対する信頼度が増せば利益があり、そして、妥当性検査及び評価で採用されている戦術の問題点を追跡できるようなESでなければならない。

(2) 検証及び妥当性検査 (V & V) とは

検証及び妥当性検査 (V & V: Verification and Validation) は、下記の条項について決定を下すために行われるソフトウェアの公式のテストからなる。

- ① 各レベルの仕様書及び出荷可能なプログラムコードが、より上位の仕様書へと追跡可能 (traceable) な状態にあること。即ち、仕様書あるいはプログラムコードが、より上位の仕様書の要件を完全に、かつ排他的に実現していることである。追跡可能性を決定する過程が検証と呼ばれる。

- ②出荷可能なプログラムコードが、本来のユーザ要件を正しく実現していること。出荷可能なプログラムコードがユーザ要件を満足している（あるいは満足していない）ことを直接に示す過程が妥当性検査と呼ばれる。

（３）エキスパートシステムのV & Vを実施する際の問題点

V & VをESに適用する際の困難さのいくつかを、以下で吟味しよう。

- ①ESに対するソフトウェア要求仕様書は、しばしば存在しないし、不明確で、素早く変更される。
- ②検証を成功させるには、より上位の仕様書の要件が、下位の仕様書の中に、少なくとも見つけられなければならない（そうでないならば、要件の追跡はムダである。）。
- ③たとえ要件追跡が可能なほど十分に記述された仕様書が利用できたとしても、従来の検証では、構築されたシステムが要件をほんとうに満たしているかどうかを決めるに必要な多くの答を産出するかは疑問である。
- ④従来の妥当性検査では、厳密なテスト手続きを要求している。

- ⑤ESのテストの結果を評価する広く受け入れられた信頼できる手法がない。

ESのV & Vをどのように実施するかを理解が不足しているため、悪循環に陥っている。この悪循環を断ち、ES用のV & V方法論を進歩させるには、V & V技術を実験的にESに適用することが必要である。

（４）エキスパートシステム用のV & V手法

ES用のV & Vに関する実務的かつ理論的な知識が不足しているので、適切な手法の選択として、実験的（cut-and-try）アプローチが最も有効である。本稿で提案する、ES用のV & Vの第一次実験（first-cut）での手法は、下記の5つのタスクからなる。

- ①要件の定義
- ②知識ベース及び支援ソフトウェアの検証
- ③テストケースの準備
- ④テストの実施
- ⑤テスト結果の評価

1) 要件の定義

ESの開発は、しばしば、ソフトウェア要件に関する公式の定義なしで進行している。これは、下記の理由からである。

- ① ユーザ及び開発者は、目的を記述した書類からプロトタイピングを始め、プロトタイピング過程で要件を洗練化することを望んでいる。
- ② ユーザあるいは開発者は、要求仕様書を準備したり、維持するのに必要な経営資源を積極的に提供するつもりがない。
- ③ 開発者は、強制力のある詳細に文書化された要件は、過度に、開発者の創造性を制約させると信じている。
- ④ ユーザは、公式の受け入れテストを実施することを計画していない。

ES開発が有用な要求仕様書なしで進行している場合には、V & V担当者は、作成されたソフトウェアから要件を規定することを試みる望ましくない立場にある。要件をソフトウェアに一致させるのは、政治的な圧迫があるかもしれないし、あるいは、ユーザまたは開発者は、要件を詳細に述べることに興味がないかもしれない。V & V担当者は、開発者側の反対をおして、要件を規定するには、組織上の権限を必要とし、そして、ユーザ、開発者、及びV & V担当者の3者の意見の一致を得るためには、この権限を行使しなければならない。

要求仕様書を関係者全員に受け入れてもらうためには、要件の作成、開発者側のレビュー、及び顧客側のレビューを何回もくり返す必要がある。有用な要求仕様書が手元にあるようになれば、V & Vの残りのタスクを始めることが可能となる。

2) 検証

従来のソフトウェアの検証は、要求仕様書、設計書、及びコードの階層を通過する要求条項の追跡と、設計の適切性を実証するエンジニアリング分析（設計解析）からなる。要求条項（要件）の追跡では、システム／ソフトウェア要求仕様書、設計書、そして、ソースコードが対象となる。正規に実施されるエンジニアリング分析では、致命度（criticality）、感度／安定度（sensitivity and stability）、効率性（efficiency）、そして、保守性（maintainability）が対象となる。ここでは、これらの活動が、ESの検証の際にどのようにして実施されるかについて検討する。

① 要求条項の追跡

要求条項の追跡は、上位の仕様書の要求条項のすべてが、下位の仕様書で満たしているかどうかと、下位の仕様書に余分な要求条項が入っていないかどうかを決定するタスクである。要求条項の追跡は、誤り、脱落、及び余分な機能が、ある開発工程から次の開発工程へと流れ出ていかないことを保証するために採用されている。

② エンジニアリング分析

【共通のエンジニアリング分析】

エンジニアリング分析は、しばしば、検証の一部として実施される。これらの分析は、コーディング作業やテスト作業を実施する前に、システムが意図した通りに動くかを予測するために採用されている。数種のエンジニアリング分析が、下記に示すように、特定のプロジェクトのニーズに応じて、実施される。

【致命度】このソフトウェアの障害が、他の構成要素あるいは使命達成に対し、どの位の影響を与えるか。

【感度／安定度】入力の中の少しの変化、特に特異点や運用限界点付近での変化に対し、採用しているアルゴリズムや試行錯誤手法がどの位敏感であるか。入力の少しの変化に対し、不正確、不安定、あるいは、失敗してしまうのか。数値解析では、しばしば、アルゴリズムの感度の判定を行っている。

【効率性】ソフトウェアは、与えられた制限時間内で、かつ制限資源内で、意図された仕事を遂行することができるか。

【保守性】ソフトウェアは、そのライフサイクル全体を通して、保守を容易にするように作成されているか。

【固有のエンジニアリング分析】

従来ソフトウェアでは採用されていない3種類のエンジニアリング分析は、ESの検証では重要となる。これらは、相互作用分析（IA: Interaction Analysis）、一貫性分析（TA: Truth Analysis）、そして、不確実性分析（UA: Uncertainty Analysis）である。IAは、望ましい結果を生成するためのESの構成要素間での相互作用に関する能力を取り扱う。TAは、知識ベースの中味の一貫性（truth）を評価する。UAは、ESが不確実性の尺度を使用している際に適用可能である。この分析は、割り当てられた不確実値と、その不確実値を組み合わせる手法とが正しい結果を与えるかどうかを決定する。

3) テストケースの準備

ES用のテストケースの選定は、原理的には、従来のコンピュータプログラムと同じであり、かつ、困難さも同程度である。Myersは、従来ソフトウェアのテストケース設計に関して、詳細に述べている。ES用のテストケースの選定基準は、下記のように要約できるであろう。

- ①すべての要件をテストすること。
- ②システムの限界点の1つにシステムの状態を陥らせたり、あるいは、誤り回復や後始末のルーチンを呼び出したりするような一連のテストケースは、独立のテストケースの集りよりも、誤りを指摘するのには、しばしば、より有効である。
- ③コードのすべての項目及びすべての可能な分岐をテストすること。
- ④ESの場合には、システムの出力結果と、システムがその出力結果に到達した道筋の両方をテストする必要がある。
- ⑤すべての特異点及び境界条件をテストすること。
- ⑥条件の組合せに関して分別のある選定をしてテストすること。
- ⑦使命達成が第一であるようなシステム(MCCP)の場合とか、テスト予算が許容される場合には、システムが機能低下したり、あるいは、失敗したりする個所を同定するために、要求限界や設計限界を越えてテストすることも重要である。

4) テストの実施

ESのテストの実施は、従来ソフトウェアのそれと同じである。オペレータは、テストケース準備工程で用意された台本(script)に則ってテストを実施し、そして、台本から逸脱したものすべてを説明を付けて記録する。テスト入力を用意し、テスト結果を自動的に記録する“台本”及び“ログ”ファイルを使用して、従来ソフトウェアのテストを実施するのが有利であることが証明されている。ESを台本に基づいてテストした有望な結果が既に得られている。

5) 評価

ESのV&Vに関する論争の大半は、テスト工程での評価段階に起因している。ESの評価に関する問題点のいくつかを、下記に列挙する。

- ①ESのテスト工程では、従来のソフトウェアのテスト工程で採用されている客観的で正確な期待される解及び評価基準が適用できないかもしれない。
- ②受け入れ可能な解が何であるかの一致がない場合もある。
- ③特に、その問題領域で不確実性を扱う推論を含んでいる場合には、ESは100%の正解率で解を与えることはできないかもしれない。

独立の評価の必要性がはっきりした。ESの開発に参画した同じ専門家が、そのESの評価に採用された場合には、第三者には、その評価結果では満足できないし、心理的なバイアスがないとはいえない。

ESの評価で考慮すべき品質特性と、従来ソフトウェアで考慮すべきそれとは、かなり異なっている。ESの評価で考慮すべき点を下記に示す。

- ①期待される解、等価な解、あるいは、受け入れ可能な等しい解が得られるか。
- ②ESはその解に対して正当性を説明することを期待されている場合には、提出された正当化理由が、顧客が要求している質や様式に合致しているか。
- ③ESに対する顧客の信頼を保証する観点から見て、ESが辿る推論の過程が十分に完備的 (complete) で、かつ頑健 (robust) であるか、そして、採用されている知識が十分に深い。
- ④ESが100%の正解率を達成することが期待できないのならば、不正確な解に対するインパクトとは何なのか：不正確な解は単に準最適解にすぎないのか、ESを無能にしてしまうほど悪いのか、あるいは、ESが支援している使命達成を実際に損なってしまうのか。もし、不正確な解が許容される場合には、不正確な解集合のどの位の部分が受け入れ可能なのか、そして、不正確な解を受け入れた時の最悪の受け入れ可能な結末 (consequence) とは何なのか。

(5) 実験的V & V

ES用のV & Vの方法論の開発には、実験を必要とする。実験は、ESを検証及びテストする際に採用する技法の洗練化を可能にし、その有効性面での信頼を向上させる。

実験的なV & V方法論が下記の能力を実証した暁には、ESに対して実験的なV & Vから本番用のV & Vへと移行するものである。

- ①ESのすべての開発工程でうめ込まれた誤りを検出すること。
- ②V & Vをパスしたことにより、ESの正当性を保証すること。
- ③ESソフトウェアのライフサイクルコストの低減に貢献すること。

ESを検証し、妥当性検査を行う能力は、ES技術の受け入れに十分に貢献する。

4.9 知識ベースのテスト

タイトル:	Testing Your Knowledge Base
著者:	Marcot, B.
出典:	AI Expert, Vol.2, No.8, pp.42-47, (July 1987)
分類:	Method and Techniques, Life Cycle Issues
キーワード:	ライフサイクル、評価基準（頑健性、予測的妥当性など）

（１）はじめに

- 本稿では、知識ベースのテスト方法と、その際に使用する妥当性評価基準について述べる。
- 知識表現としては、プロダクションルールのみを考える。

（２）テストの必要性

- テストは、プロダクションルール開発での必須の作業で、開発サイクルに組み入れるべきである。
- 検証（Verification）には、コンピュータコード（Lispコード、ルールベース）がバグなしで作成されていることを保証することが含まれる。
- 妥当性検査（Validation）には、ルールの意味や内容が充足性（Adequacy）に関するいくつかの慎重に定義された評価基準を満たしていることを保証することが含まれる。
- 知識ベースの開発サイクルは、下記の工程に分かれる。

- ① プロトタイプの創成
- ② 第1世代ルールセットの開発
- ③ テストと、ルールセットの第2世代への拡張
- ④ 第2世代ルールセットのテスト

（３）プロトタイプの創成

- プロトタイプは、典型的には、1ダース程度のルールから成り、その問題に対する知識工学的アプローチの実現可能性を実証する。
- テストでは、そのプロトタイプを評価して、この開発プロジェクトを続行させるべきか否かを決定する。

(4) 第1世代ルールセット

- 第1世代ルールセットは、典型的には、数ダースから数100個程度のルールから成る。
- アルファテスト (Alpha-testing) の段階では、システムの正確性 (Accuracy) と充足性 (Adequacy) がどのレベルに達しているかを決定する。

(5) ルールセットの拡張

- 先ず、この段階では、フルスケールシステムに対する能力基準を決めること。
- 拡張作業には、典型的には、推論構造、高位の制御ルール、そして、低位の事実及びそれらの関係の再構築も含まれる。

(6) 第2世代のルールセット

- 最初のフルスケール知識ベースに対するテストは、利用現場で行われる。
- ベータテスト (Beta-testing) の段階では、システムの正確性、精度 (Precision)、及び信頼性 (Reliability) がどのレベルに達しているかを決定する。

(7) 市場性

- ここで言っている市場性 (Marketability) は、特に、想定利用者へのシステムのアイデアや使用に関する売り込みについてのことである。
- 妥当性テストの主要ステップには、下記の作業項目が含まれる。
 - ① エキスパートシステムが使用される領域及び文脈の定義
 - ② 妥当性に関する個別の能力評価基準の同定
 - ③ 妥当性検査の実施と、テスト結果の分析・評価

(8) 領域・文脈

- 知識ベースがうまく解くことが期待される領域及び文脈を決定すること。
- モデル化においては、モデルが使用される諸条件を厳密に定義すること。

(9) 個別の能力評価基準

- 能力評価基準は、知識ベース開発の各段階ごとに規定されること。
- テストを実施する前に、各能力基準の合格レベルを決定しておくこと。

(10) テスト結果の分析

- 定量的テストを行う場合の分析段階には、問題領域、既知のテストケース、システムの実行方法、出力結果の記録方法などについての決定が含まれている。
- 妥当性テストの結果は、システム開発の評価にも当然使用される。
- システムが本来の目的や能力基準を満たしているかである。

(11) 妥当性検査の評価基準

- 下記に示す評価基準が、知識ベースの妥当性をテストし、評価する目的で使用されるかもしれない。

* 正確性 (Accuracy)

- シミュレーションがどの位良く現実を反映しているかを表す度合である。
- ルールを使用して推論された結果と、史実（既知）データとを比較し、結果の正当性を評価すること。

* 適応性 (Adaptability)

- 将来の開発や適用がどの位容易であるかの可能性を表す度合である。
- 新しい情報が利用可能になった時には、I/Oや制御ルールの汎用性を保持して、事実やルールを改訂すること。
- 既存の、あるいは提案されたハードウェアやソフトウェアシステムとの統合の望ましさを、定期的にレビューすること。
- システムが自己修正的であるのか、あるいは、文脈に敏感であるのかの、いずれなのか。
- システムが、特定のユーザニーズを満たすためにカスタマイズできるのか。

* 充足性 (Adequacy)

- 関係する経験的知識の中で、どの位シミュレーションできるかを表す度合である。
- 推論結果に影響を与えるパラメータ群（変数、条件、関係（変数間の論理的関係及び統計的関係））のリストを設定し、ルールセット中にどのパラメータをいれるかを決定すること。

* アピール性 (Appeal)

- 使用性 (Usability) のこと。
- 知識ベースが我々の直感とどの位一致し、我々の思考をどの位刺激するかを表す度合である。
- アピール性は、市場性 (Marketability) に対してきわめて重大な影響を与える評価基準である。
- 開発工程の比較的早い時期に、I/Oの友好性 (Friendliness) を評価し、

使用性をテストすること。

- ベータテスト (Beta-testing) を行うべき開発段階において、現場で、シミュレーション及び実用性をテストすること。

*** 入手性 (Availability)**

- 現実的な市場性を決定付ける際に重大な影響を及ぼす、同じ問題を解決する、他のより簡単で、妥当な知識ベースがどの位存在するかを表す度合である。
- 他のツールが既に利用可能で、かつ、そのツールがユーザニーズを満たしている場合、ユーザが新しいルールベースに対するニーズを感じているか。

*** 規模性 (Breadth)**

- 知識ベースで使用されるルールの個数に比例する度合である。
- システムが解くべきであると期待される文脈の個数と、ルールセット中で考慮すべき関連のあるパラメータの個数を決定すること。

*** 深 度 (Depth)**

- モデルの各構成要素を表現するために選択された変数の個数及び種類に比例する度合である。
- システムが注目する条件群の各々の範囲と、各条件下で、診断したり、分類したり、助言したりする際にどのパラメータが必要かを決定すること。
- 深度を決めたならば、そのたびに、必要な入力データ及びユーザインタフェースを決定すること。

*** 表面的妥当性 (Face validity)**

- モデルの信用性 (Model credibility) のこと。
- プロトタイプ開発の初期の段階及びフルスケールシステムの拡張の最終の段階で、信用のある専門家が、知識ベース、推論構造、そして出力結果をレビューすること。
- そのレビュー結果は編集され、報告されること。

*** 汎用性 (Generality)**

- 知識ベースが広い範囲の同種の問題を解決するのにどの程度利用できるかの能力を表す度合である。
- システムが専門家レベルの能力で解くことができると期待される汎用的な文脈群を定義し、これらの文脈を越えて利用した場合には、正確な結果が与えられないかもしれない点について強力な警告が用意され、表示されること。

*** 精 度 (Precision)**

- モデルが特定のシステムパラメータ群をどの程度忠実にまねているかの能力を表す度合である。
- 数値変数や計算の場合には、有効桁数のことである。

- すべてのパラメータ群の関係する変動範囲すべてがルールベースや事実群でカバーされていることを保証すること。
- 数値は、必要があれば、浮動小数点や実数形式で表現されること。計算、特に、行列計算や線形代数計算の場合には、倍精度が用いられていること。

*現実性 (Realism)

- 関係する変数群や関係（変数間の論理的関係及び統計的關係）群をどの位考慮しているかを表す度合である。
- ルールベース中に入れるべきパラメータ群（変数、条件、関係）や機能群を、専門家やエンドユーザが使っている用語や概念モデルを使用して設定すること。
- 現実性は、フルスケールの知識ベースを開発する際には、特に重要である。
- 現実性には、質問を出す論理的な順序も含まれる。

*信頼性 (Reliability)

- モデルによる予測結果が経験的に正しい割合を表す度合である（実際は、ルールベース全体の正確性及び正当性は、高度な統計的分析で調べられ、信頼性はその一部である。）。
- 信頼性には、正しい診断結果及び分類結果に対する条件つき確率及び事後確率が含まれる。
- 最終的には、ルール及び出力結果に対する尤度の統計的効用を述べること。

*頑健性 (Robustness)

- モデルの構造（あるいは外れ値の混入）に対し、特に敏感でない結論のこと（モデル化の際に決めた仮定がくずれても、能力が損なわれない割合を表す度合である）。
- どの入力パラメータが中間結果（診断、分類の場合）、最終結果（助言、警告の場合）、そして出力に対して、最小の、あるいは最大の影響を与えるかを決定すること。
- 最大の影響を与える入力パラメータは、ルールベースや機能中で十分に定義されていることを保証すること。

*感 度 (Sensitivity)

- 知識ベースのパラメータ群の変動範囲内で、どの程度史実データと一致する出力を導出できるかを表す度合である。
- 特に、ある特定のパラメータを段階的に変化させ、その他のパラメータを一定に保持し、そして、モデルの出力を史実（既知）データに一致させるようにして、各入力パラメータに対する結果の感度を決定すること。

*技術的及び操作的妥当性 (Technical and operational validity)

- モデル化の際に決めた仮定に対して、認識された現実との相違を同定し、その相違がどの程度重大であるかを決めること。
- ルールや関係が前提とする文脈、条件、仮定を注意深く解明すること。
- 各々の仮定がモデルの結果をどの位制約するかを検討すること。
- 仮定がモデルの正確性、信頼性、頑健性、及び汎用性にどの位影響を与えるか。

*チューリングテスト (Turing test)

- 特定の同じ問題に対して、モデルによる結論と、(人間の)専門家が解いた結論とを、(人間の)評価者に区別してもらって、知識ベースの妥当性を評価すること。

*有用性 (Usefulness)

- システムが、問題解決の文脈下で使用するのに、必要十分なパラメータ群及び関係群を含んでいるかを調べること (少なくとも、モデルによる予測結果が経験的に正しいことがある場合)。
- 有用性に関しては、フルスケールシステムの場合には取るにたらないことではあるが、しかし、プロトタイピングや既存のルールセットへのルールの追加の場合には、重要である。

*予測的妥当性 (Validity i.e. Predictive validity)

- 知識ベースが、過去の経験に照らして見て、どの程度正しい予測結果を出すかを表す度合である。
- システムがうまく解くことが期待されている文脈が与えられた場合、どの位多くの実際の条件で、システムが正確に診断したり、分類したり、助言したりできるかを決定すること。
- 同じ対象領域で、(人間の)専門家が正当性のどのレベルを達成しているかを決定し、知識ベースの正当性に関して現実的な目標を設定すること。

*総体性 (Wholeness)

- モデルに組み込まれている (思考・推論) プロセス及び対話の個数に比例する度合である。
- ルールベースがどの位複雑か。
- それは、いくつの因子を使用しているか。
- 適応性をかんがみて、総体性を考慮すること。

4.10 エキスパートシステムの性能に対する妥当性検査

タイトル:	Validating Expert System Performance
著者:	O'Keefe, R.M. Balci, O., and Smith, E.P.
出典:	IEEE EXPERT, Vol.2, No.4, pp.81-90, (Winter 1987)
分類:	Statistical Tools, Method and Techniques
キーワード:	定性的方法、定量的方法、多重応答問題 (Multiple-response problem)、許容性能範囲 (Acceptable performance range)、第I種/第II種の誤り

(1) 概要

エキスパートシステムの研究開発における要件として妥当性検査の確立をはかる。妥当性検査における問題点を検討し、定性的、定量的な方法を提示する。

(2) 検証、妥当性検査、および評価

本論文では評価の一部である妥当性検査を取り上げる。ただし、全体的な評価と明確に分離することは困難な場合がある。また、検証ではシステムが仕様どおりに実現されているかどうかを実証するのに対し、妥当性検査ではシステムが許容できる程度の精度を発揮するかどうかを実証する。

(3) エキスパートシステムの妥当性検査

妥当性検査においては次の7項目の主要な問題点がある。

① 何を検査するか (What to validate)

中間結果、結論、推論、およびこれらの組合せについて検査できる。開発の初期の段階では推論の検査に重点がおかれ、知識ベースがより完全になったところで結論の検査を行うのが普通であろう。また、結論を正しいか誤りかのどちらかに分類するのは困難なことがある。このような場合でも Hickamらのように数個のカテゴリを用意すれば分類できることが多い。

② 何との比較で検査するか (What to validate against)

既知の事実との比較で検査するべきではなく、専門家の能力との比較で検査

するべきである。

③ 何で検査するか (What to validate with)

テストケースの選択は検査に影響を及ぼす。生起頻度や困難さなどを考慮して包括的なテストケースを用意しなければならない。また、システム開発時に使われたものは検査に用いてはならない。

④ いつ検査するか (When to validate)

エキスパートシステムは、開発の早い段階で許容性能を発揮すべきである。ただし、許容レベルは実験型プロトタイプと発展型プロトタイプとは異なる。実験型では基本的なアプローチの正しさが問題となる。発展型では改良を必要とするところの洗い出しが重要であり、応用がヘビーデューティーでない場合には、フィールドテストによる検査も行われる。

⑤ 検査費用のコントロール (How to control the cost of validation)

検査には時間と費用とがかかりそれを予測することは困難である。しかし、フォーマルな妥当性検査方法を設計し開発プロセスと統合化することにより費用をコントロールできると思われる。

⑥ バイアスのコントロール (How to control bias)

専門家の知識のバイアスを制御するには、ブラインド検査やクロスチェックを利用する。これらはOncocinの検査で用いられた方法である。システム開発者のバイアスを最小にとどめるためには、開発者とは別のチームによる検査を行う。

⑦ 複数の結果の取扱い (How to cope with multiple results)

複数の結果を出力するシステムを検査する場合には「多重応答問題」(multiple-response problem)とよばれる問題がある。例えば医療診断において2種類の薬品を処方した場合に、それぞれは妥当であっても併用してはならない場合などが相当する。

(4) 基本概念

妥当性検査には次の5種類の基本概念がある。

① 許容性能範囲 (Acceptable performance range)

利用者が受け入れられる性能レベルは「許容性能」と呼ばれ、開発中に定めておく。システムがモデル化している専門家の能力に対応づけて決められることが多い。

② 入力領域 (The input domain)

開発者は特定の応用のためにシステムを設計すべきである。検査においてはその応用の見地から、定められた入力領域についての性能だけを検査すべきである。

③ 妥当性検査 (Validity testing)

システムの妥当性は、入力領域内で、設計目的に関連して、許容性能範囲についてのみテストすることができる。

④ 検査のフォーマル性 (Validation formality)

フォーマルな検査では、開発サイクルの中のどこで実施されるかが確立されており、また、検査方法、入力領域、許容レベル、および（適切な場合には）統計的手法の適用が定められている。

⑤ 開発者／利用者のリスク (Builder's/user's risk)

検査結果には第Ⅰ種と第Ⅱ種の誤りが含まれる。第Ⅰ種はシステムが妥当であっても棄却してしまう、いわば「開発者のリスク」である。第Ⅱ種はシステムが妥当でないのに受け入れてしまう「利用者のリスク」である。妥当性検査の設計にあたってはこれらについて客観的に考慮すべきである。

(5) 定性的検査

定性的検査は主観的な性能比較を扱う。ここでは7種類のアプローチを示す。

① 表面的妥当性検査 (Face validation)

予備段階の検査法として有効な方法である。プロジェクトチームのメンバーがシステムの性能と専門家とを主観的に比較する。そのためにテストケースの結果と許容性能レベルとを表面的(at face value)に評価する。

② 予測的妥当性検査 (Predictive validation)

過去の事例を入力し、既知の結果あるいは専門家の判断と比較する。

③ チューリングテスト (Turing test)

検査対象が専門家かシステムかをふせておいて、両者の比較で検査する。コンピュータを使っていることによるバイアスを除けるという利点もある。

④ フィールドテスト (Field test)

プロトタイプを現場において動作させ、エラーを収集する。開発者側の利点は、利用者にテストの責任を負わせられること、および、許容性能範囲が暗黙のうちに得られることである。

⑤ サブシステム単位での妥当性検査 (Subsystem validation)

サブシステムに分割して検査できる場合には、開発と並行して検査が可能、検査の複雑さの減少、エラーの同定が容易、といった利点がある。

⑥ 感度解析 (Sensitivity analysis)

入力やパラメタを系統的に変えることによって結果に及ぼす影響を調べる。感度解析は妥当性検査に用いられた実績がないけれども、強力な定性的評価法となり得る。特にテストケースが得られない場合や不確実性を導入している場合などに有効と思われる。

⑦ 視覚的対話 (Visual interaction)

システムの動作を動画として表示し、対話的なパラメタの変更などを可能にすることによって、これまでに述べた表面的妥当性検査などの環境を提供する。専門家が推論プロセスを検査する際などに有効である。

(6) 定量的検査

定量的検査では、統計手法を用いてシステムの性能とテストケースあるいは専門家との比較を行う。

① t 検定 (2 標本問題) (Paired t-test)

n 個のテストケースについて、平均値の誤差、標準偏差、および自由度 n の t 分布の値から信頼区間 (confidence interval) を計算し、その区間にゼロが含まれていれば合格とする。

② ホテリングの T^2 検定 (Hotelling's one-sample T^2 test)

システムが出力する複数の結果の間に関連がある場合には、t 検定 (2 標本問題) では不適當でありホテリングの検定を用いるとよい。

③ 同時信頼区間 (Simultaneous confidence intervals)

多変量応答システムの検査には、一対応答の違いに対する同時信頼区間を構成し、許容性能範囲との比較を行う。

④ 一致係数 (Consistency measure)

複数の専門家との比較を行う場合には、専門家との間の一致性が問題になる。そのため、級内相関係数 (interclass correlation coefficient) を利用して信頼性を評価する。

(7) まとめ

エキスパートシステムの開発者にとって、妥当性検査と評価とは現在よりもいっそう留意する必要がある。以下に有用なガイドラインの要約を示す。

- 許容性能範囲と規定された入力範囲とについて検査すること。
- 開発サイクルへ妥当性検査を組み込むこと。
- 利用者のリスクと開発者のリスクとを考慮すること。
- 適切な定性的検査を選択すること。
- 可能であれば定量的検査を用いること。

4.11 KB-Reducer: 知識ベースの非一貫性及び冗長性の検査に対する 新アプローチ

タイトル: Knowledge-Base Reduction: A New Approach to Checking
Knowledge Bases for Inconsistency & Redundancy

著 者 : Ginsberg, A

出 典 : AAAI-88, pp.585-589, (1988)

分 類 : Automation

キーワード : 冗長性の検査、非一貫性の検査

(1) はじめに

- 一人の専門家によって作られたエキスパートシステムが色々なサイトに配られ利用されることを考えた時、各サイトでそのサイトに合ったような知識の修正が行われることか考えられ、修正した知識による不具合が発生する可能性がある。
- KB-ReducerはE S開発者に対し知識の冗長性、非一貫性を示してくれるツールである。

(2) KB-Reducerの推論モデル

- 知識ベースの一貫性、非冗長性を議論する場合は演繹推論機構が前提である。
- ここで述べる"KB-Reducer"の特徴は次の3つである。

①単調性: ある入力データ集合 α がCを導くならば、 α のスーパーセットもCを導く。ワーキングメモリ(wm)はassertのみでretractは無い。

②無選択性: 競合解消は無い。マッチしたものは全て発火される。

③データ駆動型エンジンである。

(3) 非一貫性及び冗長性の定義

- 論理で使われる一貫性の概念と知識ベースで使われる一貫性の概念の違いを抽出しておくことは重要である。

論理に於いては、

$$p \rightarrow Q$$

$$p \ \& \ r \rightarrow \neg Q$$

は一貫性がある。

しかし、知識ベースでは一貫性がないとされる。pとrがインプットとして与えられるとQと $\neg Q$ がwmにassertされてしまう。

- 一貫性に関しては知識ベースの方が厳格である。
- 冗長性に関しては論理の世界と知識ベースの世界とで同じ様な概念でつかわれている。

① 妥当な入力セットと意味的制約

- Eが知識ベースに入力されると意味的制約を犯していないかを検証する。

【例】John is a male human being.

John is pregnant.……男は妊娠しないと言う意味的制約を犯している
オブジェクトの一属性は一つの属性値しか持たない(single-valuedness)
制約には論理的制約と数学的制約がある。

- ドメインに固有の意味的制約知識は時間の関数として増えていく。従って今まで一貫性が検証されていても、意味的制約が追加されることにより一貫性が保てなくなることがある。
- 知識ベースが決して矛盾ある結論をassertしないためには意味的制約をベースに全ての潜在的に一貫性が保てないものを削除することが出来なければならない。

(4) KB-Reducer

- KB-Reducerはruleとfindingとhypothesesとdefault-hypothesesとから成っている。
 - rule
ruleは右辺と左辺からなり、左辺ではand,orの結合子が使える。
 - finding
findingは左辺にでてくる文字及び右辺にでてくる文字の論理否定でないものである。

- hypotheses

hypothesesは右辺にでてくる文字、または、いくつかのルールの方右辺にあり且つ他のルールの左辺にある文字である。

- default-hypotheses

default-hypothesesは左辺にだけでてくる文字であり、且つあるhypothesisの論理否定であるものである。

(Knowledge Base)

$$p \vee q \rightarrow R$$

$$a \rightarrow B$$

$$B \& R \rightarrow S$$

$$\neg S \& c \rightarrow \neg R$$

Finding: p, q, a, c

Hypotheses: $R, \neg R, B, S, T$

Default-Hypotheses: $\neg S$

Label for R: $p \vee q$

Label for B: a

Label for S: $a \& p \vee a \& q$

Label for $\neg S$: $\neg a \vee (\neg p \& \neg q)$

Label for $\neg R$: $\neg a \& c \vee (\neg p \& \neg q \& c)$

図4.11-1 Findings, Hypotheses, Default-Hypotheses,そして Label

- def-hypoは知識ベースの中にはassertされないもので、def-hypo Dの環境EはDの逆の仮説を結論付けることを妨げるfindingの集合になるように解釈される。

- def-hypoの計算の方法（図4.11-1参照）

- cが知識ベースにインプットされる。
- $\neg S$ が真であると仮定する。
- それ故、 $\neg R$ が結論付けられる。
- $\{p, c\}$ が入力されるとRと $\neg R$ の両方が成り立ってしまう。
- 暗黙の仮定として、もしインプットとして与えられなかったら、その否定

が成立すると仮定する（閉世界仮説）。例でいくと a はインプットとして与えられていないから $\neg a$ が成り立つと仮定する。

① 知識ベースの順序付け

- KB-Reducerでは、ルールは非サイクルネットの構造である必要がある。
- 知識ベースが非サイクルであると仮定すると、次のような再帰定義に従って決定されるレベルにルールを分割する事が出来る。

- もし r の左辺の全ての条件が満たされていれば r のレベル = 0

- r のレベル = $1 + (r$ が依存するルールのレベルの最大値)

② パーシャルラベルとルールラベル

- 知識ベースが非サイクルであるとする、level-0, level-1, ... というように順にルールを処理することにより、知識ベースをreduceすることができる。
- 全てのルールが処理された後、全てのhypothesesとdefault-hypothesesのラベルが分かる。一つ一つ処理されたルールとして生成された不完全なラベルをパーシャルラベル (Partial label) と呼ぶ。
- パーシャルラベルの更新は次の様に行われる。
 - rule r が現在処理されており、 r が仮説 H を assert したとする。
 - 最初に r の左辺にある文字のラベルの論理結合を持ち、そして結果表示を最小化することによって r の左辺を満足させる最小環境の完全な集合を計算する。生成された最小環境の集合を“rule r のルールラベル”と呼ぶ。
 - 仮説 H の新しいパーシャルラベルは r のルールラベルの論理命題と現在のパーシャルラベルにより決定される。
 - もし上記のように計算が行われたら、ルールのルールラベルはそのルールが知識ベースに追加された時に計算可能になり、計算が他のルールに移るとき、ルールのルールラベルが決して更新されないことを証明できる。

ルール	レベル	ルールラベル	パーシャルラベル (After Processing Rule)
r1: $p \vee q \rightarrow A$	0	$p \vee q$	$A: p \vee q$
r2: $q \vee r \rightarrow B$	0	$q \vee r$	$A: p \vee q, B: q \vee r$
r3: $s \& \neg q \rightarrow \neg B$	0	$s \& \neg q$	$A: p \vee q, B: q \vee r, \neg B: s \& \neg q *$
r4: $A \rightarrow D$	1	$p \vee q$	$A: p \vee q, B: q \vee r, \neg B: s \& \neg q,$ $D: p \vee q$
r5: $B \rightarrow \neg D$	1	$q \vee r$	$A: p \vee q, B: q \vee r, \neg B: s \& \neg q,$ $D: p \vee q, \neg D: q \vee r **$
r6: $D \& \neg A \rightarrow T$	2	$(p \& \neg p \& \neg q) \vee (q \& \neg p \& \neg q) ***$	$A: p \vee q, \neg A: \neg p \& \neg q, B: q \vee r,$ $\neg B: s \& \neg q, D: p \vee q, \neg D: q \vee r$

*印 $s \& \neg q \& r$ as leading to potential contradiction: B and $\neg B$
 **印 q as leading to potential contradiction: D and $\neg D$
 ***印 $D \& \neg A \rightarrow T$ as unsatisfiable(redundant) rule

図4.11-2 非一貫性ルール及び非充足ルールの検出

③冗長性の検査

- 仮説Hをassertするルールのルールラベルを計算するとする。この時次のようなことを決定することによって冗長性のチェックをする。

i) rのルールラベルは非一貫的または無効な環境から成っていないか?

rは決して満足されないので知識ベースは冗長である。

[例] 図4.11-2の r6 はルールラベルが $(p \& \neg p \& \neg q) \vee (q \& \neg p \& \neg q)$ なので絶対に発火しないため冗長なルールであるといえる。

ii) rのルールラベルは仮説Hの現在のパーシャルラベルに包含されていないか?

満足させる全てのケースは前に処理されたルールによってすでにカバーされているので、ルールrはKBから削除される。

[例] r11 $a \vee b \rightarrow H$ Hのパーシャルラベル: $(a \vee b)$

r22 $a \rightarrow H$ r22のルールラベル: (a)

r22のルールラベルは現在のHのパーシャルラベルに包含されているため、r22は冗長ルールである。

iii) rのルールラベルは仮説Hの現在のパーシャルラベルを包含していないか?

一つまたはそれ以上、前に処理されたルールによって結論付けた全てのケースに於けるHをルールrは結論付けるので、前に行われたルールのいくつかは知識ベースから削除される。

これは、上記例の反対のケースである。

④ 矛盾性の検出

- 仮説 H を assert したルール r のルールラベルを計算するとする。 r のルールラベルを使って H のパーシャルラベルを更新する。
- H と競合する全ての仮説 X に対して次の様な subset/superset テストと union テストが行われる。
- subset/superset-test は H のパーシャルラベルの中にどんな環境もあるかを決定する。この H のパーシャルラベルは X のパーシャルラベルの中にあるどんな環境の subset または superset である。
- この様な環境は X と H の両方を知識ベースに assert する入力集合を表現する。環境はフラグが付けられる。

〔例〕 図4.11-2の r_5 において D と $\neg D$ へのパーシャルラベルを見る。

$D: p \vee q$

$\neg D: q \vee r$

$p \vee q$ の環境と $q \vee r$ の環境は相反する仮説を導く。

- E_1 と E_2 が、どちらも他の subset でないような X , H のパーシャルラベルから作られる環境としよう。その様な各ペアにたいして union-test は結合された $E_1 \cup E_2$ の環境が意味的制約を侵していないかをテストする。もし、 $E_1 \cup E_2$ がそれらの制約を侵していないならば、それは、矛盾の仮説を知識ベースに assert させた入力組み合わせを表している。

〔例〕 図4.11-2において r_3 の B と $\neg B$ のパーシャルラベルの union をとった環境を見てみる。

$s \ \& \ \neg q \ \& \ r$ の環境を作りだしたとすると、この環境は意味的制約を侵していないが B と $\neg B$ の両方を導く。

(5) おわりに

①実験結果

- このシステムはExplorer-IIで走行している。
- 実験結果を表4.11-1に示す。

表4.11-1 KB-Reducerの実験結果

ルール数	50	150	370
処理時間	40cpu-s	5cpu-m	10cpu-h
生成環境数	700	4000	35000

- いくつかのケースにおいて冗長なルールを見つけることが出来た。
- 注目すべきことは、1回このシステムを走らすと、知識ベースを修正する影響を決定するので知識ベース全体に対して再処理を施す必要が無くなることである。

②複雑性の考慮事項

- 知識ベースの順序付けの複雑さはルールの数に比例する。
- 縮小化のステップの複雑さは生成される環境の数に比例する。
最悪 3^n の環境を生成する (n : findの数)。

③今後の方向

- 知識ベースのサイクリックを許すこと、推論モデルが選べること、データ指向を弱めること(ある文字が入力にも仮説にも両方使える)等を現在研究している。
- 変数を持ったルールが扱え、述語論理の特徴をもっと生かしたKB-Reducerを開発したい。

4.12 Digital社におけるコンフィギュレーションエキスパートシステム

タイトル:	Expert Systems for Configuration at Digital: XCON and Beyond
著 者 :	Baker, V.E. and O'Connor, D.E.
出 典 :	Communications of the ACM, Vol.32, No.3, pp.298-318, (Mar. 1989)
分 類 :	Experience
キーワード:	CES, XCON, XSEL, XFL, XCLUSTER, XNET, SIZER

(1) まえがき

Digital Equipment Corporation(DEC)では、XCON等のCES (Configuration Expert System)を約十年にわたって開発してきた。ここでは、その経験を以下の観点からまとめる。

- ESの開発、運用を成功させるには、技術的問題以外にビジネスニーズと組織の問題を考慮する必要がある。
- ESの開発には、従来のソフトウェアエンジニアリング以外に独自の技術が必要である。

(2) Digital社におけるCESの現状

1) 稼働開発中のCES

① XCON

主に以下の機能を提供する(原文で述べられている機能の一部を挙げる)。

- 顧客の注文を技術的、営業的視点から解析し、その妥当性を検証する。
- 必要なハードウェアの組み合わせ、及びケーブルを求める。

② XSEL

- 顧客の注文に応じて、システムの部品を対話的に選択する。
- ソフトウェアも含めて、部品の組み合わせの妥当性を検証する。
- 計算機室に関し、システムの設置に必要な事項を求める。

③ X F L

- ・ 計算機室におけるシステムのレイアウトを求める。

④ X C L U S T E R

- ・ 通信網のクラスタの設計を支援する。

⑤ X N E T

- ・ ローカルエリアネットワークの設計と、その妥当性の検証に用いる。

⑥ S I Z E R

- ・ 仕事及び環境に応じて、必要な計算機資源の量を推定するために用いる。

以上のうち、①～④が稼働中、そして⑤、⑥が開発中である。

2) 利用状況

C E S のユーザ部門及びそこでの利用目的は以下の通り。

① 販売部門

顧客の注文の妥当性を検証し、見積りを出すための利用。

② 製作部門

注文の実現可能性の検証、注文の分割（分担して製作するため）、組立て、検査プログラムの選択。

③ フィールドサービス部門

現場での組立て、設置、既存機器との統合。

④ 製作部門及びエンジニアリング部門

システムレベルの設計の評価。

3) 事業への貢献

顧客の満足、生産性の向上、経費削減の点で貢献した。特に、顧客のニーズに合わせてオーダーメイドのシステム構成を提供するというDigital社の戦略を支えている。金額換算で、年に4000万ドル以上の貢献がある。具体的な効果は以下の通りである。

- ① システムの不備がなくなる。また、システムの構成方法が一貫する（同じ機能のシステムが常に同じ構成になる）ので、顧客を混乱させることがなくなる。

- ② 情報の一元管理により、製作、フィールドサービス部門の混乱を防止する。その結果、製品の立ち上がり、組織の配置変更を迅速化できる。

- ③ 問題のある注文（実現不可能な注文など）を早期に発見し、手戻りを防止できる。

- ④ 分担して製作するサブシステム間の不整合が防止できるので、全体の整合性を確認する工程 (Final Assembly and Test) が省略できる。

(3) CESの開発経験

以下では、事業的問題、技術的問題、組織的問題に関する経験及びそこから得られた教訓について述べる。

1) 事業的問題

① 技術の適用分野

ESのような新技術に対して戦略的投資を行う場合、事業の根幹に係わるような適用分野を選ぶべきである。そうすれば、各部署から十分な支援が得られる。CESの場合は、オーダメイドのシステムを提供するという会社の戦略を支えるシステムであった。

② ビジネス部門との協力

CESの各ユーザ部門のマネージャからなるConfiguration Systems Steering Committeeを設立し、部門間の調整を考慮した開発の方向付けを得ている。さらに、もっと上位レベルの協力を組織化しつつある。

③ 開発環境

Digital社では、ES技術に対する期待が大きく、支援が充分であった。

2) 技術的問題

① 技術的挑戦

a. 既存言語及びツールのうまい利用

パターンマッチング機能の強さと処理の速さを考慮して、VAX OPS5を利用した。

b. ES構築のためのソフトウェアエンジニアリング方法論の確立

- ソフトウェアエンジニアリング方法論RIMEを開発し、XCONの再開発に利用して有効性を明らかにした。
- ESに適したテスト技術の開発が今後の重要課題である。

② アプリケーション特有の問題に対する挑戦

a. 分野の変化

コンフィギュレーション分野は、関連する事物の変化が多い。そのためCESのルールの40%を毎年書き換えている。具体的には、以下の変化がある。

- 毎年膨大な新製品が出る。またそれに伴い既存製品の組み合わせ方が変わる。

- 分野知識及びE S技術の蓄積に伴う知識ベースの改良。

- コンフィギュレーション業務自体の変化。（例えば、ネットワークの普及に伴う業務の変化）

b. システムの拡張の必要性

以下の理由から絶えずCESを拡張する必要がある。

- ユーザ層の拡大に伴うニーズの変化
- 既存ユーザのニーズの変化
- 他のシステムとの結合に伴うニーズの発生
- ユーザが一旦システムを使い始めると要求がどんどん高くなっていく。
（例えば、使い勝手の良さ、利用部署毎に最適化した出力など）

c. システムの巨大さと複雑さ

- 各CESのルール数を表4.12-1に、ルール数の複雑さを表4.12-2に示す。
- データベースの規模を表4.12-3に示す。

表4.12-1 各CESのルール数(1988/9時点)

C E S	ルール数
X C O N	1 0 1 2 9
X S E L	3 6 2 9
X F L	1 8 0 8
X C L U S T E R	2 4 3
X N E T (フ・ネット)	1 7 0 7

表4.12-2 ルールの複雑さ(平均)

パート	要素数
I F 部の条件数	6
条件中の要素数	5
t h e n 部の a c t i o n 数	4

表4.12-3 データベースとその規模

データベース	規 模
部品 D B	30000以上の部品 部品データの要素数約40
容器部品 D B (部品を差し込む容器の D B)	725以上の容器部品
部品形状 D B (出力のための図形表現の D B)	1200以上の図形
質問文テンプレート D B	1200以上の文型
システム構成 D B (C E S の解答)	稼働箇所により異なる

- ・従来型ソフトウェアは350個以上のプログラムで、50,000行以上である。

③ 開発管理

a. 巨大さ・複雑さへの対応

従来のソフトウェア開発と同様に、全体のアーキテクチャを定めた後、各部分を開発するアプローチが必要である。

b. 保守性の確保

- ・Digital社のAIトレーニングプログラム及びCES開発専用のトレーニングプログラムを用いて、人員養成期間を短縮した。
- ・ES開発方法論RIMEにより、人員養成期間を短縮すると共に、保守に伴う品質の安定を図った。
- ・自前のツール群により、知識ベースの頻繁な保守に対応した。

c. 短期的開発と長期的開発のバランスに注意した。

3) 組織的問題

開発体制を確立するために、開発に必要な役割(key role)と開発フェーズを明確化した。

① ES開発に必要な役割を表4.12-4に示す。

表4.12-4 ES開発に必要な役割

役 割		仕 事
CHAMPION		戦略的ビジョン、信念、政治力を持ってSPONSORを動かし開発環境を整備する。
SPONSOR		ESの目的達成に関して責任を持ち、リソース、資金を提供して必要な環境を与える。
PROGRAM MANAGER		プログラム開発を管理する。
PROGRAM OFFICE		PROGRAM MANAGERを補佐。業務分析や開発計画の立案を行う。
TECHNICAL TEAM	KNOWLEDGE ENGINEER	AI部分の設計、知識獲得。
	SOFTWARE SYSTEMS INTEGRATION ENGINEER	ESの全体設計、非AI部分の設計、開発、開発環境の整備、リリース管理
	RESEARCH / ADVANCED DEVELOPER	開発が成功した段階で次の新技術を導入する。
	EXPERTS	対象分野の知識を提供する。
	USERS	対象分野のビジネス環境、そこでのニーズなどに関する知識の提供。

②役割に関する教訓を以下に述べる。

- a. ソフトウェアシステムインテグレーションエンジニアの役割を過小評価し、人員を充分割当てしなかったため、大きな手戻りが生じた。
- b. 専門家は複数必要である。一人の専門家に頼ると偏ったESになる。
- c. 技術的問題に注意が偏り、ユーザの意見を軽視した結果、大きな手戻りが生じた。そこで偏りを防止するために役割間に次のような関係を持たせた。
 - ・ プログラムマネージャは、スポンサとユーザに対して責任を持つ。
 - ・ テクニカルチームは、プログラムマネージャに対して責任を持つ。
 - ・ ユーザは、ES導入によるビジネス環境の向上に関し、スポンサに対して責任を持つ。

③ES開発モデルに関しては、以下のタスクからなるモデルを確立した。

- ・ システムの定義及び再定義

ES技術の利用方法、従来技術との役割分担、ユーザニーズの充足などの問題

- ・ システムの拡張、洗練化

知識の獲得と管理、複数システムの統合、各種インタフェースの管理

- ・ システムの配布

リリース管理、ユーザサポート

- ・ システムの評価

上記のモデルは、基本的には従来のソフトウェア開発モデルと同様である。

各タスクの遂行は以下の理由から従来より試行錯誤的になる。

- ・ 技術の発展が速い。
- ・ 分析しにくい問題を対象としている。
- ・ 技術と問題の相性がよく分かっていない。
- ・ ES導入によりビジネス環境が大きく変わる。

(4) 結 論

C E S の Digital 社に対する貢献をまとめると以下のようなになる。

- ① 複数の部署にまたがるコンフィギュレーション業務及び関連情報を統一的に整理し明確化している。
- ② 計算機製品及びその組み合わせが次第に複雑化していく中で、オーダーメイドのシステムを作るという会社の戦略を支えている。
- ③ 業務を効率化し業務変化への対応を容易化し、業務形態の発展を促進している。
- ④ E S 技術の適用実験として、様々な知見をもたらしている。

4.13 エキスパートシステムのテスト、検証、引渡し

— Mentor履歴の事例 —

タイトル: Testing, Verifying, and Releasing an Expert System :
The Case of History of Mentor)

著 者 : Cochran, E.L. and Hutchins, B.

出 典 : Proc. 3rd Conference on Artificial Intelligence
Application, pp.163-167, (Feb. 1987)

キーワード : Mentor, フィールドテスト, 導入時ユーザ訓練プログラム, 技術移転

(1) 概 要

MentorはHoneywell社（H社）のビルサービス代理店が大規模冷却システムを維持管理する際に、H社のビルサービス部門（BSD: Building Services Division）が作業支援するのに使用する携帯用エキスパートシステム（ES: Expert System）である。MentorはPML（（H社に）知的所有権のある「知識工学用手続き言語」）で作成され、ラップトップコンピュータ上で実行される。Mentorの知識ベースは、600個以上のルール形式風のアクションから成っている。これらのアクションは、サービス対象の冷却システムの型式、サービスを行う原因、サービス対象の冷却システムの過去のサービス履歴、及び特定のシステム構成に応じたシーケンスである。本論文では、徹底的なフィールドテスト、知識ベースの妥当性検査、そして諸支援サービス（ユーザ訓練プログラムなど）の開発などのESの最終引渡しを行うために実施した導入準備作業の過程について述べている。ESに関する技術移転は、いくつかの点でユニークであり、Mentorの引渡し準備作業の間に学んだ教訓は、ESがより日常的なツールとなってきたので、広範囲に適用できると予期している。

(2) フィールドテストの諸計画

1) Mentorの特徴

Mentorは、ESとして見た場合、通常のESと比べ次の諸点で異なっている。

- ① 典型的なユーザは、最初はES及びAIというまでもなく、キーボード、ディスク装置、パーソナルコンピュータに馴れていない。
- ② 電圧測定器が道具であるように、それと同じ感覚で使えるように、Mentorを道具として設計した。Mentorは数百人のエンドユーザに使用されることを想定している。エンドユーザすべては、その道具を使用して、仕事をよ

り完璧に、より効率よく行うことに関心を持っている。

③ Mentorには、米国中どこでも使用でき、異なる型式の装置、異なる気候風土、及び異なる使い方のパターンに課せられた多様な冷却システム維持管理要件を満たすことが期待されている。

④ Mentorを実行するハードウェアは、携帯用で、堅牢で（サービス用のトラックのバックに何週間も入れっぱなしにしておいても故障しない）さらに、機械室の天井裏に簡単に持ち運びができ、汚く、雑音が多く、振動の激しい環境でも使用できなければならない。

2) フィールドテストの要件

Mentorのフィールドテストの計画が作成されるにつれて、空調をサービスするのに使用する方法(method)は、次に示す1次要因及び2次要因により、非常にバラエティに富んでいることがはっきりしてきた。

① 1次要因

- ・ サービス対象の機械の型式（遠心式冷却装置対往復運動式冷却装置）
- ・ 付属装置の型式（バックアップポンプ、浄化装置など）

② 2次要因

- ・ 気候風土
- ・ 使用地域の地理上の位置
- ・ ビルサービス代理店の経営資源
- ・ ビルサービス代理店の経験度及び従業員の選り好み

フィールドテストの目的が、Mentorの能力及び弱点を明確にすることならば、次に示す内容を満たす広範なテストが行われなければならない。

① 異なる地域に属する地方で

② 異なる型式の冷却装置

③ 異なるエンドユーザ

④ 異なる季節

フィールドテスト計画は、次に示す諸点により複雑になってきた。

① テスト可能期間が短い（4カ月）

② Mentorに対する洞察的な分析能力を備えた高度な経験を持つビルサービス代理店の従業員をたくさん必要とするが、不足している。

③ サービスを必要とする多様な冷却システムを扱っているBSDの支局数が比較的少ない。

3) フィールドテスト計画の策定及び実施

Mentor用の十分なフィールドテスト計画を作成することができた。フィールドテスト計画では、Mentorのすべての機能に対し、複数のテストサイト及び冷却装置設備を使用して、徹底的なテストを実施することにした。結果として、次に示す条件で、Mentorに対し、冷却システムの正規の使用（年間で典型的な冷却装置に対して15回サービス訪問すると仮定・定義し、換算して）で約6年間の使用累積結果が収集できた。

① 4 カレンダー月

② 東部、北東部、中西部、そして西部にある6カ所のBSDフィールドオフィス

4) フィールドテストの結果

Mentorの診断能力は、フィールドテスト期間中、期待以上であった。Mentorはサービス対象の装置に発生している多種多様な問題を正しく検出し、かつ、すべてのケースにおいて、ビルサービス代理店の従業員が気付いた問題すべてを検出した。Mentorは、通常、検出した問題に対する正しい手順（解決策）を勧告する（勧告文が不完全あるいは誤っている場合には、KBが修正される）。1つ以上の原因による冷却設備の長期間解決できなかった問題に対しても、Mentorは正しい処置方法（アクション）を勧告した。冷却装置に問題が発生し、実際にサービス訪問し、Mentorを使用した場合、Mentorの能力は良好であったが、ビルサービス代理店の従業員が模擬的に訪問して課した特殊な問題の診断に関しては、KBにいくつかの弱点が現われた。加えて、「ユーザインタフェース」、「導入時ユーザ訓練プログラム」、そして「Mentorシステム起動手順」がフィールドテストの初期の頃、混乱の原因となった。結果として、Mentorの様々な側面が改善された。「システムコード」、「KB」、「ユーザインタフェース」、「導入時ユーザ訓練プログラム」、そして「Mentorシステム起動手順」が、フィールドテストの結果、すべて変更され、変更のいくつかは大規模であった。例えば、KBはフィールドテスト期間中に30%も増大し、ユーザインタフェースは完全に改善された。フィールドテストの結果として、ESのテスト及び妥当性検査がどういう意味をもつかに関して、我々の認識が変化した。発生した論争点は、最後に教訓という形で提示する。

(2) フィールドテスト結果に基づく変更

フィールドテストデータの分析結果に基づく変更は、次に示す3つに類型化された。

- ①ソフトウェア誤りによる変更
- ②KBに内在する問題
- ③システムとユーザとのインタフェース問題

(3) 技術移転の過程で学んだ教訓

Mentorは商用として広範囲に使用する目的で引渡されている数少ないESの1つであり、技術移転の過程で我々は貴重な経験をし、ESを成功に導く手順について学んだ。以下に、重要な教訓を示す。

- ①十分な計画が要諦である。

技術移転では、関心やニーズの異なるたくさんのグループの人々を巻き込むので、これらの人々の関心やニーズを満たす詳細な計画が必要である。

- ②ターゲットグループの人達を開発の初期の段階から参画させ、そして開発の最初から参画しているグループの人達を継続的に参画してもらうような計画を立てよ。技術的に高度な製品を、ターゲットグループに対し、あたかも既成の事実のようにして提供した場合には、技術導入は成功しそうにもない。仮に、最初はユーザに受け入れられたとしても、その開発に参画していないグループによる支援ではうまくゆくはずがない。支援に関する技術的要件、システムの複雑さに馴れるには長期間必要、そして、NIH (not invented here, 他人の発明したものは使わない) 症候群は、成功をさまたげる強固な障壁である。

- ③KBの検証は複数の専門家にやってもらうこと。

この教訓は知識工学界で論争の種になっているが、我々の経験では、ESを知識のある広範なエンドユーザに受け入れてもらうには、誰もが受け入れることのできるアプローチ（取り組み方）を提供するか、あるいは、異なる環境ごとに適する様々なアプローチを提供するかのいずれかでなければならない。いずれの方法を採用するにしても、1人の専門家のみに頼るKEは、代替アプローチに対し、システム開発期間中及びユーザ訓練期間中に、準最適として、システムに組み入れるのか、あるいは明示的に棄却するのかの決定をやらないという危険を冒すことになる。

- ④フィールドテストは早期に、頻繁に行うこと。

ESのテストを研究所内に限定することもできないし、プロジェクトの開発者のみで実施することもできない。代わりに、テストは、実際の使用環

境下で、実際に使用するユーザをできるかぎり十分に反映したサンプルで行われなければならない。テストの目的は、KBの問題点を同定するだけでなく、不十分なユーザインタフェース、不十分なユーザ訓練など、ユーザ側の受け入れに際しての障壁をも同定することであり、この点が肝心である。

今後とも、Mentorの進化過程の監視を続ける予定である。ESが現場でより長期間使用されることにより、今後の技術移転やES支援活動を計画するに際して貴重な更なる経験を積むことができるであろう。

4.14 ORにおける視覚対話型モデリング手法：その成功と可能性

タイトル： Visual Interactive Modelling in Operational Research:
Successes and Opportunities

著 者： Bell, P.C.

出 典： Journal of the Operational Research Society,
Vol.36, No.11, pp.975-982, (1985)

分 類： Others

キーワード： コンピュータグラフィックアプリケーション、視覚対話型モデリング

(1) 概 要

視覚対話型モデリング手法は今やOR専門家にとって一般的なツールの一部であり、それに関する新しいアプリケーションが定期的に報告されるようになっている。本論文では視覚対話型モデリング実施の状況を考察し、成功アプリケーション領域および将来の可能性を特定する。

(2) はじめに

視覚対話型モデリング手法 (Visual Interactive Modelling: VIM) は単なる一手法ではなく幅広い対話型のグラフィックなモデル構築方法に対して用いられる一般的な用語である。VIMによるモデルには二つの基本的なタイプがあり、グラフによる表現モデル (Representational Graphic Models: RGM) とアイコンによるグラフィックモデル (Iconic Graphic Models: IGM) である。

RGMは表現のためにチャートや図形 (ヒストグラム、線グラフ、パイチャート等) を用いて数学的なモデルからのデータをサマリーする。RGMは主要なコンピュータグラフィックスの商業的アプリケーションであるし、今後もそうであろう。

しかし、このタイプのモデルは概念的には数学的モデルそのものからほとんど進んでいない。したがって問題は表形式の表現に対するグラフィックな表現の価値、および色による表現の価値があるかどうかである。これらの問題の結

果は明確である。例えば I v e s はグラフィックスの使用に関して管理の生産性を向上したとか、より高度の決定をもたらしたとかいう決定的な証拠はないとまで言っている。

I G M は問題あるいはプロセスをグラフィック表示としてモデリングする。特に興味をそそられる分野は対話環境における I G M である。最も成功した I G M のアプリケーション分野は P E R T / C P M などのプロジェクト管理の分野である。さらに V I S (Simulation) アプリケーションには、待行列解析、ジョブショップスケジューリング、交通量解析、組立てライン設計、コンベヤシステム設計、生産計画、多目的計画等がある。地図に基づく V I M は、経路問題、輸送計画、施設配置問題等に良く用いられている。トラベリングセールスマン問題に対する視覚対話型アプローチは Hurriion により試みられており、100都市から200都市に適用された時には、最適値の4%以内の平均精度の答えを与えるとしている。

V I M は問題解決のためのツールであるばかりではなく、注意を換気し O R 専門家の仕事を表現する理解容易な手段を提供する。

(3) 現在の V I M の問題点

現在の V I M の中心的な問題は今後のユーザにとってあまりにも難しく、使い勝手が悪いことである。V I M システムを構築するには複雑で込み入ったコンピュータプログラミングを必要とする。V I M の使い勝手はモデルビルダとエンドユーザという二つのユーザがいることによりより複雑になっている。モデルビルダは通常専門家であり、エンドユーザは通常意志決定者(管理者またはプラントオペレータ)、または教育または訓練を受けている学生である。V I M がエンドユーザの要求により良く答えられるようにするためにエンドユーザとモデルビルダを区別することは有効である。

現在の技術はモデルビルダが一人の人間ではなく O R 専門家とコンピュータプログラマから成り立つチームであることを要求する。O R 専門家は視覚モデルを設計し、そのモデルがスマートに見えるように数学的モデルを構築するための適当な問題解決技法を選択しなければならない。プログラマはコーディングによりグラフィックスおよび対話インターフェイス、さらにはモデルやアルゴリズムを開発しなければならない。

エンドユーザにとって現在の技術に欠けているものは、モデルの対話要素を

理解しモデルを管理者に説明してくれる“お抱えの運転手”がないことである。仮にいたとしても、管理者の意思決定は部下の監視にさらされることになり、モデルの実験手段としての有効性は減じられるかもしれない。この“お抱え運転手”の代替案としてはエンドユーザにモデルの使い方を教えることである。しかしそれは時間的な制約等により上級の管理者による使用を制限することになる。もう一つの代替案はより使い勝手の良いモデルを構築することである。

V Iのエンドユーザを助ける上で最も困難なことは、ユーザはどのようにモデルを使用すべきかには注意を払っているにしても、ユーザがどのようにモデルを使用したいかについてほとんど注意が払われていないことである。モデルはO Rの専門家の問題解決アプローチに従って問題を考察するように構築され、管理者が彼等のアプローチを使って問題を考察できるようにはなっていない。Parkerは彼のV Iモデルを構築するのにまず最初幾つかの表示方法を設計し、問題を持っている人とともに検討している。デスカッションと修正の後でこれらのスクリーンはプログラムされる。

管理者が興味を持つ機能はチェックポイント（すなわち、ある決定を試みて、それがうまくゆかないことが判明したら、同じ決定ポイントに戻って別の方法を試みることができる機能）、前後への素早い移動画面、タッチスクリーン（部品を自分の指で移動したい）、その他の使い勝手の良い入出力装置（ライトペン、マウス、その他）である。V I Sにとって有効な機能は同時実行可能な複数のシミュレーションである。同じようなシステムの同時シミュレーション実行は有効であり、あたえられたシステムの複数の意思決定ルールにおけるシミュレーション結果の同時表示は直接の比較を可能にする。

V Iマーケティングモデル、V I金融計画モデル、V I戦略計画モデル、V I人員計画モデル、および複数のソフトな意思決定規範が主要な公共部門では、大きなV Iマーケティングのポテンシャルがある。これらの分野では最初はグラフィック表現的なV Iモデルが使われるだろうが、我々がこれら非空間システムの視覚表現について有効なものを開発するにつれて、I G Mがすぐさまそれに続くであろう。

(4) 新しい技術

VIMの発展がハードウェアの発達に依存していることは疑いがない。VISはもっと大きなモデルが可能になるより早いプロセススピード、より実際の時間スケール、より多くの色、よりきめの細かい易しい使い勝手の良い対話が可能になれば大きく進歩するであろう。より簡便で持ち運べるモデルも望ましい。

現在はVIMはソフトウェアに依存しておりソフトウェアに制約されている。ユーザはすべてのレベルにおける操作において問題と制限を感じている。VIMをより簡単に使用し理解できるためにもっと大きなソフトウェアの発展が必要である。いつかは意志決定者は自分自身でVIモデルを開発し実行できるようになるだろうが、現在は夢物語である。

(5) 今後の方法論

OR社会と管理者達によりVIMを受入れてもらうためには幾つかの方法論的な問題が解決されなければならない。その中でも最も重要な鍵は価値を立証することである。我々はバッチモードのモデルよりもVIMが必要なことを実証しなければならない。VIモデルのコストは明確で算定可能だが、その利点は算定不可能で計測もできない。

VIモデルの数学的モデルと比較した妥当性の容易さについて、特にエンドユーザによる妥当性検査によって、多くのことが言われたり書かれたりしているが証拠は逸話的である。データの表形式の表示よりグラフィックス表示より価値があるという証拠はほとんどない。OR専門家は問題解決に対する視覚対話型アプローチの価値について独自の方法で証明しなければならないだろう。

その他の発展が必要な領域は表示そのものの設計である。色や動きは表示の意味を偏向するかも知れない疑いがある。しかし我々がどのように中立的な表示が設計できて意志決定者が彼等の決定が真実彼等自身によるものであると確信できるという訳ではない。

ParkerとBellはソフトシステム(Soft System)に基づいた視覚対話型問題解決(Visual Interactive Problem Solving:VIP S)の方法を提唱している。創造性研究(ブレインライティング(brain writing)、ガイドッドファンタジィ(guided fantasy))からのアイデアを利用することにより、この創造的な分野は発展の大きな機会がある。

(6) 結 論

VIMは実に短かい期間に多くの成功をおさめ、ORの重要な部分になってきた。管理者に考えられているようにVIMの出現はORにとって良い影響をあたえている。次の4ステップがこの良い立場を強固なものにするであろう。

最初に視覚対話型手法はモデル構築手法というよりも問題解決アプローチとして強調されるべきである。すなわち、VIMよりもVIPSが強調されるべきである。次に、VIMは単なる視覚対話型シミュレーションより以上のもので多くのタイプの問題が利益を受けるものであることが強調されるべきである。第3番目に、グラフィックス表示の設計により注意が払われるべきである。最後に、VIPSは上級の管理者が一晩中起きていられるような重要な意思決定問題に成功裡に適用されるべきである。

OR社会はこの機会を捕え、視覚対話型手法が育つにつれて多くの利益を得られるようにすべきである。

4.15 エキスパートシステム評価関連文献一覧

[会議名の略称一覧]

- AAAI: 1-th National Conference on Artificial Intelligence
(American Association for Artificial Intelligence)
- AISIG: Artificial Intelligence Systems in Government Conference
- CAIA: The i-th Conference on Artificial Intelligence Application
- ESIG: 1-th Annual Expert Systems in Government Conference
(Expert Systems in Government Symposium)
- IJCAI: The i-th International Joint Conference on Artificial Intelligence
- WESTEX: Western Conference on Expert Systems

[1983]

- Gasching, J., Klahr, P., Pople, H., Shortliffe, E., and Terry, A.,
"Evaluation of Expert Systems: Issues and Case Studies," in
Building Expert Systems, ed. by Hayes-Roth, F., Waterman, D.A.,
and Lenat, D.B., Addison-Wesley, Reading, 1983, pp.241-280.
(なぜESの評価をするのか; ESの評価を設計する際の問題点; 評価すべきESの特徴; いつESを評価するのか; どのようにESを評価するのか; 事例: R1)
- Vesonder, G.T., Stolfo, S.J., and Zielinski, J.E.,
"ACE: An Expert System for Telephone Cable Maintenance,"
IJCAI-83, pp.116-121, (Aug.1983).
(ACE (Automatic Cable Expertise); 電話網保守の際に必要とする
トラブルシューティング報告書及び管理資料を作成するES; 構築の詳細
; ACEのテスト; 1982年春以降フィールド・テスト中)
- Miller, F.D., Copp, D.H., Vesonder, G.T., and Zielinski, J.E.,
"The ACE Experiment: Initial Evaluation of an Expert System for
Preventive Maintenance," Joint Services Workshop on Artificial
Intelligence in Maintenance, Oct.1983, pp.421-428.
(ACE (Automatic Cable Expert); 電話網の保守過程でのACEの役割;
フィールドでのACEの能力を評価する際に使用した手法; ES評価
に関する一般的な論争点)

[1984]

Kohoutek, H.J., "Quality Issues in New Generation Computing,"
FGCS'84, pp.695-702, (Oct.1984).

(次世代コンピューティングの品質問題; 知識, 知識表現, KB, 探索過程,
演繹推論, 帰納推論, 学習, プログラミング言語の品質; ハードウェアの
品質)

Suwa, M., Scott, A.C., and Shortliffe, E.H., "Completeness and
Consistency in a Rule-Based System," in Rule-Based Expert Systems,
ed. by Buchanan, B.G. and Shortliffe, E.H., Addison-Wesley, Reading,
1984, pp.159-170.

(ルールベースの完備性及び一貫性に関する原著論文)

Buchanan, B.G. and Shortliffe, E.H., "The Problem of Evaluation,"
in Rule-Based Expert Systems, ed. by Buchanan, B.G. and
Shortliffe, E.H., Addison-Wesley, Reading, 1984, pp.571-588.

Yu, V.L., et al, "An Evaluation of MYCIN's Advice," in Rule-Based
Expert Systems, ed. by Buchanan, B.G. and Shortliffe, E.H.,
Addison-Wesley, Reading, 1984, pp.589-596.

Stevens, A., "How Shall We Judge on Expert Systems," in Expert Systems:
Principles and Case Studies, ed. by Forsyth, R., Chapman and
Hall Computing, 1984, pp.36-48.

(ESの社会的影響; ESの評価; 薬の臨床試験との対比; 医師の診断との
対比)

[1985]

Nguyen, T.A., Perkins, W.A., Laffey, T.J., and Pecora, D.,
"Checking an Expert Systems Knowledge Base for Consistency and
Completeness," IJCAI-85, pp.375-378, (Aug. 1985).

(ルールベースの完備性及び一貫性に関する論文)

Neches, R., Swartout, W.R., and Moore, J., "Explainable (and
Maintainable) Expert Systems," IJCAI-85, pp.382-389, (Aug. 1985).

Kline, P.J. and Dolins, S.B., Choosing Architectures for Expert
Systems, Final Technical Report RADC-TR-85-192, Oct.1985, pp.154,
Rome Air Development Center, Griffiss AFB, NY, 13441.

(ESの設計ガイドライン; 問題の特性に適したAI技術の選定; 53個の
AI技術の解説有)

Liebowitz, J., "Evaluation of Expert Systems: An Approach and Case Study," CAIA'85, pp.564-571, (Dec.1985).

(1986年版 "Useful Approach for Evaluating Expert Systems" の元論文)

Bell, M.Z., "Why Expert Systems Fail," Journal of the Operational Research Society, Vol.36, No.7, pp.613-619, (1985).

Bell, P.C., "Visual Interactive Modelling in Operational Research: Successes and Opportunities," Journal of the Operational Research Society, Vol.36, No.11, pp.975-982, (1985).

[1986]

Liebowitz, J., "Useful Approach for Evaluating Expert Systems," Expert Systems, Vol.3, No.2, pp.86-96, (Apr.1986).

(専門業務をESでやるべきか, 専門家がやるべきかの選定手法; AHP (Analytic Hierarchy Process); Read (Requirements Engineering Automated Development))

Fieschi, M. and Joubert, M., "Some Reflections on the Evaluation of Expert Systems in Medicine," Methods of Information in Medicine, Vol.25, No.1, pp.15-21, (Jan.1986).

(医療分野でのESの能力の問題点; 入力データの信頼性, ルールの信頼性, 確信度計算, 説明機能, 診断の一致性についての検討)

Vera, J.W. and Blóczak, R., "Artificial Intelligence Techniques Applied to Claims Processing: Expert System Research Tool," ESIG'86, pp.116-127, (Oct. 1986).

Frail, R.P. and Freedman, R.S., "OPGEN Revisited: Some Methodological Observations on the Delivery of Expert Systems," ESIG'86, pp.310-317, (Oct. 1986).

(OPGEN (Operations Sheets Generator); 開発方法論; HAIL-1, OPGEN-R, OPGEN-Lの比較評価; 教訓的ガイドライン)

Freedman, R.S. and Frail, R.P., "OPGEN: The Evolution of an Expert System for Process Planning," AI Magazine, Vol.7, No.5, pp.58-70, (Winter 1986).

(上記の論文の改訂版)

Adelman, L., "On the Basis of What Criteria should Decision Support Systems be Evaluated?," ESIG'86, pp.348-350, (Oct. 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, (Fall 1986).
- Kline, P.J. and Dolins, S.B., "Problem Features that Influence the Design of Expert Systems," AAAI-86, pp.956-962, (Aug.1986).
(1985年版 "Choosing Architectures for Expert Systems" の要約論文)
- Richer, M.H., "An Evaluation of Expert System Development Tools," Expert Systems, Vol.3, No.3, pp.166-183, (July 1986).
- Hansen, J.V. and Messier, W.F.Jr., "A Knowledge-Based Expert System for Auditing Advanced Computer Systems," European Journal of Operational Research, Vol.26, No.3, pp.371-379, (1986).
(コンピュータ監査のES; コンピュータ・システムの内部統制の信頼性; コンピュータ監査へのES適用の可能性; AL/X)
- Hansen, J.V. and Messier, W.F.Jr., "A Preliminary Investigation of EDP-XPert," Auditing: A Journal of Practice & Theory, Vol.6, No.1, pp.109-123, (Fall 1986).
(EDP-XPert (コンピュータ監査用ES) の評価; ES の評価手法; 評価基準; 評価結果; AL/X の評価)
- Oliver, A.E.M., "Techniques for Expert System Testing and Validation," 3rd International Expert Systems Conference, Learned Information, pp.271-276, (1986).
(従来のソフトウェア開発で使用されているテスト技法及び妥当性検査技法をどのようにすればESに適用できるか; ESの妥当性検査の経験)
- Langlotz, C.P., Shortliffe, E.H., and Fagan, L.M., "Using Decision Theory to Justify Heuristics," AAAI-86, pp.215-219, (1986).

[1987]

- O'Keefe, R.M., Balci, O., and Smith, E.P., "Validating Expert System Performance," IEEE EXPERT, Vol.2, No.4, pp.81-90, (Winter 1987).
(検証・妥当性検査・評価とは; ESの妥当性検査の考慮事項; 基本的な諸概念; 定性的妥当性検査; 定量的妥当性検査; 統計的考え方の適用)
- Lundsgaarde, H.P., "Evaluating Medical Expert Systems," Social Science and Medicine, Vol.24, No.10, pp.805-819, (1987).
(医療ES評価の文献調査; 医療ESの内の約90%は臨床環境下で評価されていない; 医療ESの評価の基本的な手法; 研究戦略)

Kjeldsen, R. and Cohen, P., "The Evolution and Performance of the GRANT System," CAIA'87, pp.272-276, (Feb.1987).

Cochran, E.L. and Hutchins, B.L., "Testing, Verifying, and Releasing an Expert System: The Case History of Mentor," CAIA'87, pp.163-167, (Feb.1987).

(Mentor (ビルの空調システムの予防保守 E S) ; フィールド・テスト ; K B の妥当性検査 ; E S 引渡しの準備作業 ; 教訓)

Soloway, E., Bachant, J., and Jensen, K., "Assessing Maintainability of XCON-in-RIME: Coping with the Problems of a VERY Large Rule-Base," AAAI-87, pp.824-829, (July 1987).

(X C O N (eXpert C O Nfigurer) ; X C O N の保守面の問題 ; ルール用高位言語 R I M E ; コードの同質性・予測性 ; プログラミング環境 ; 所見)

Cragun, B.J. and Steudel, H.J., "A Decision-Table-Based Processor for Checking Completeness and Consistency in Rule-Based Expert Systems," Int. J. Man-Machine Studies, Vol.26, No.5, pp.633-648, (1987).

(ルールベースの完備性及び一貫性に関する論文)

Johnson, R.G., Joly, G.C., and King, P.J.H., Survey of Techniques for Checking of Rule-Based Expert Systems, Technical Report, Department of Computer Science, Birkbeck College, London University, 1987, pp.8.

(ルールベースの完備性及び一貫性に関するサーベイ論文)

Johnson, R.G., Joly, G.C., and King, P.J.H., BIRCH - The Birkbeck Rule Checker - An Example, Technical Report, Department of Computer Science, Birkbeck College, London University, 1987, pp.8.

(ルールベースの完備性及び一貫性に関する論文)

Mettrey, W., "An Assessment of Tools for Building Large Knowledge-Based Systems," AI Magazine, Vol.8, No.4, pp.81-89, (Winter 1987).

Marcus, S., Stout, J., and McDermott, J., "VT: An Expert Elevator Designer that Uses Knowledge-Based Backtracking," AI Magazine, Vol.8, No.4, pp.41-58, (Winter 1987).

(V T (Vertical Transportation) ; エレベータ設計支援 E S ; 知識獲得ツール S A L T を使用して生成された)

Herbert, M.R. and Williams, G.H., "An Initial Evaluation of the Detection and Diagnosis of Power Plant Faults Using a Deep Knowledge Representation of Physical Behaviour," Expert Systems, Vol.4, No.2,

pp.90-99,(May 1987).

Messier, W.F.Jr. and Hansen, J.V., "Expert Systems in Auditing: The State of the Art," Auditing: A Journal of Practice & Theory, Vol.7, No.1, pp.94-105,(Fall 1987).

Green, C.J.R. and Keyes, M.M., "Verification and Validation of Expert Systems," WESTEX-87, pp.38-43,(June 1987).

(本格的なESのV & Vの論文; ESのV & Vの悪循環; ESのV & V方法論(要件の定義, 検証, テスト・ケースの準備, テストの実施, 評価)の提案)

Pau, L.F., "Prototyping, Validation and Maintenance of Knowledge Based Systems Software," ESIG'87, pp.248-253,(Oct. 1987).

Davis, G.A. and Yeh, D., "An Expert System for Selecting Appropriate Domains for Expert Systems," ESIG'87, pp.260-265,(Oct. 1987).

Marcot, B., "Testing Your Knowledge Base," AI Expert, Vol.2, No.8, pp.42-47, (July 1987).

Citrenbaum, R., Geissman, J.R., and Schultz, R., "Selecting A Shell," AI Expert, Vol.2, No.9, pp.30-39, (Sept. 1987).

Nguyen, T.A., Perkins, W.A., Laffey, T.J., and Pecora, D., "Knowledge Base Verification," AI Magazine, Vol.8, No.2, pp.69-75, (Summer 1987).

O'Leary, D.E., "Validation of Expert Systems - with Applications to Auditing and Accounting Expert Systems," Decision Sciences, Vol.18, No.3, pp.468-486, (Summer 1987).

[1988]

荻野, 市川, "エキスパート・システムの運用/評価に関する一考察,"

昭和63年度人工知能学会全国大会(第2回), pp.393-396,(1988.7.21-23).

Parsaye, K. and Chignell, M., "Measuring Expert System Performance," in Expert Systems for Experts, John Wiley & Sons, Reading, 1988, pp.365-374.

(分類型ESの妥当性検査; 知識用物差し; 混同マトリックス; 混同コスト・マトリックス; 可能混同マトリックス; リスクのコスト換算)

Walters, J. and Nielsen, N.R., "Evaluating the Prototypes," in Crafting Knowledge-Based Systems, John Wiley & Sons, Reading, 1988, pp.121-143.

(ESプロトタイプの評価; テストの目標(完備性, 一貫性, 強固性
(robustness)); テスト技法; デバッキング; 妥当性検査)

Cupello, J.M. and Mishelevich, D.J., "Managing Prototype Knowledge/Expert System Projects," C.ACM, Vol.31, No.5, pp.534-541, (May 1988).

Slagle, J.R. and Wick, M.R., "A Method of Evaluating Candidate Expert System Applications," AI Magazine, Vol.9, No.4, pp.44-53, (Winter 1988).

(ESを適用すべきアプリケーションの選定手法; ユーザ及びマネジメント, タスク, 専門家の各側面から点数付けをし, 荷重平均を計算して選定; チェックリストとして有用)

Cohen, P.R. and Howe, A.E., "How Evaluation Guides AI Research," AI Magazine, Vol.9, No.4, pp.35-43, (Winter 1988).

Lindsay, S., "Identifying Problems that Expert Systems Can Solve," in Practical Applications of Expert Systems, QED Information Sciences, Reading, 1988, pp.39-48.

(ESを適用すべきアプリケーションを選定するためのチェックリスト)

Jacob, R.J.K. and Froscher, J.N., "Facilitating Change in Rule-Based Systems," in Expert Systems: The User Interface, ed. by Hendler, J.A., Ablex, Reading, 1988, pp.249-284.

Bielawski, L. and Lewand, R., "Testing, Validation, and Maintenance," in Expert Systems Development: Building PC-Based Applications, QED Information Sciences, Reading, 1988, pp.221-239.

Geissman, J. R. and Schultz, R. D., "Verification & Validation of Expert Systems," AI Expert, Vol.3, No.2, pp.26-33, (Feb. 1988).

Shaw, M. and Woodward, J., "Validation in a Knowledge Support System : Construing and Consistency with Multiple Experts," International Journal of Man-Machine Studies, Vol.29, No.?, pp.329-350, (1988).

Ginsberg, A., "Knowledge-Base Reduction: A New Approach to Checking Knowledge Bases for Inconsistency & Redundancy," AAAI-88, pp.585-589, (1988).

Kirk and Murray (SAIC), EPRI NP-5978s: Verification and Validation of Expert Systems for Nuclear Power Plant Applications; Methodology and Life-Cycle, 1988.

Parsaye, K., "Acquiring & Verifying Knowledge Automatically," AI Expert,

Vol.3, No.5, pp.48-63, (May 1988).

Bahill, A.T., Harris, P.N., and Senn, E., "Lessons Learned Building Expert Systems," AI Expert, Vol.3, No.9, pp.36-45, (Sept. 1988).

Gordon, S., "The Human Factor in Expert Systems," AI Expert, Vol.3, No.1, pp.55-59, (Jan. 1988).

O'Leary, D.E., "Methods of Validating Expert Systems," Interfaces, Vol.18, No.6, pp.72-79, (Nov.-Dec. 1988).

[1989]

Barker, V.E. and O'Connor, D.E., "Expert Systems for Configuration at Digital: XCON and Beyond," C.ACM, Vol.32, No.3, pp.298-318, (Mar.1989).

Bachant, J. and Soloway, E., "The Engineering of XCON," C.ACM, Vol.32, No.3, pp.311-317, (Mar.1989).

(XCONの問題点(複雑性, 修正容易性); 問題解決手法; ルールベース作成のRIME版ガイドライン; RIMEの利点; 構成用大規模ES(2~3万ルール))

Weitzel, J.R. and Kerschberg, L., "Developing Knowledge-Based Systems: Reorganizing the System Development Life Cycle," C.ACM, Vol.32, No.4, pp.482-488, (Apr.1989).

Press L., "Expert System Benchmarks," IEEE EXPERT, Vol.4, No.1, pp.37-44, (Spring 1989).

Marcus, S. and McDermott J., "SALT: A Knowledge Acquisition Language for Propose-and-Revise Systems," Artificial Intelligence, Vol.39, No.1, pp.1-37, (May 1989).

(ルール方式の計画・設計型ESを生成する知識獲得ツール;

Propose&Revise法; VT (Vertical Transportation); 実用品)

Popolizio, J.J. and Cappelli, W.S., "New Shells for Old Iron," Datamation, Vol.35, No.8, pp.41-48, (Apr. 15, 1989).

(メインフレームで使用可能なシェル; ESとトランザクション処理/データベースとの統合; ADS (Aion Development System))

Ramsey, C.L. and Basili V.R., "An Evaluation of Expert Systems for Software Engineering Management," IEEE Transactions on Software Engineering, Vol.15, No.6, pp.747-759, (June 1989).

(ARROWSMITH-P (ソフトウェア開発プロジェクトの進行状態を

診断するES) ; 2つの知識獲得アプローチ (ボトム・アップ, トップ・ダウン) と2つの知識表現 (ルール, フレーム) との組み合わせた4つのESの比較評価)

McGraw, K.L. and Harbison-Briggs, K., "Evaluating Knowledge-Base Development Efforts," in Knowledge Acquisition: Principles and Guidelines, Prentice Hall, Reading, 1989, pp.299-325.

Harrison, P.R., "Testing and Evaluation of Knowledge-Based Systems," in Structuring Expert Systems: Domain, Design, and Development, ed. by Liebowitz, J. and De Salvo, D. A., Prentice Hall, Reading, 1989, pp.303-329.

O'Leary, D.E. and O'Keefe, R.M., Verifying and Validating Expert Systems, IJCAI-89 Tutorial Program MP4, Aug. 21, 1989, pp.37.

O'Keefe, R.M. and O'Leary, D.E., The Verification and Validation of Expert Systems, pp.31, 1989.

(上記のチュートリアル資料の改訂版)

Preliminary Proceedings of the Workshop on Verification, Validation and Testing of Knowledge-Based Systems at IJCAI-89, Detroit, Michigan, (Aug. 19, 1989).

Miller, L.A., "Tutorial on Validation and Verification of Knowledge-Based Systems," Conference on Expert Systems Application for Electric Power Industry, pp.30, (June 1989).

Miller, L.A., "A Comprehensive Approach to the Verification and Validation of Knowledge-Based Systems," Preliminary Proceedings of the Workshop on Verification, Validation and Testing of Knowledge-Based Systems at IJCAI-89, pp.31, (Aug. 1989).

Landauner, C., "Principles of RuleBase Correctness," Preliminary Proceedings of the Workshop on Verification, Validation and Testing of Knowledge-Based Systems at IJCAI-89, pp.21, (Aug. 1989).

Hayes-Roth, F., "Towards Benchmarks for Knowledge Systems and Their Implications for Data Engineering," IEEE Transactions on Knowledge and Data Engineering, Vol.1, No.1, pp.101-110, (Mar. 1989).

Benbasat, I. and Dhaliwal, J.S., "A Framework for the Validation of Knowledge Acquisition," Knowledge Acquisition, Vol.1, pp.215-233, (1989).

Gaines, B.R. and Shaw, M.L.G., "Comparing the Conceptual Systems of Experts," IJCAI-89, pp.633-638, (1989).

Keyes, J., "Why Expert Systems Fail," AI Expert, Vol.4, No.11, pp.50-53, (Nov. 1989).

竹内啓他（編），統計学辞典，東洋経済新報社，1989年12月，pp.1185.

（テストの妥当性，一致係数，信頼区間，同時信頼区間，分割表などの統計関連用語が参考になる）

5. 結論と今後の課題

5. 結論と今後の課題

本報告では、エキスパートシステム評価ガイドライン研究委員会の第2年度の間報告をとりまとめた。エキスパートシステムの評価に対する共通の認識がない現状においては、本委員会では、エキスパートシステムに対するV & Vの重要性を明確に主張するとともに、エキスパートシステム評価のガイドラインの第1版を、とりあえず、包括的・網羅的な項目を一覧できるような形で取りまとめた。

本報告書の主要な成果は、次の5点にまとめられよう。

- ① 個々のメーカーやユーザの独自の方法論にかたよらないようにエキスパートシステム評価項目を取りまとめ、できるだけ広い範囲の問題に適用できるようなガイドラインを設定したこと。
- ② さらに、エキスパートシステムの種類と評価者の視点とから、ガイドラインの諸項目を整理して示し、実際の評価に適用できるようにしたこと。
- ③ 本委員会の各委員が、直接的に開発にたずさわったエキスパートシステムを対象に、エキスパートシステム評価の事例分析を行い、評価項目の分類の妥当性について検討を行ったこと。
- ④ 本委員会の委員による海外調査、学識経験者からのヒアリングを含めて、広くエキスパートシステムの評価の問題について調査を行ったこと。
- ⑤ エキスパートシステム評価の代表的な理論・手法について、広く文献調査を実施し、その結果をとりまとめたこと。

エキスパートシステムの評価について、本報告で論じたような本格的な調査研究は、我国でははじめてのものと信ずる。また、このような研究は、世界的にみてもきわめて少なく、本報告書ならびに本報告書で引用した諸文献を参照することによって、エキスパートシステムの評価手法に関する最新の情報を得ることが可能であろう。

しかし、エキスパートシステムの評価手法、あるいはV & V手法の研究そのものも、まだ始まったばかりであり、技術的にも未熟な点が多い [O'Keefe 1989]。本報告で示した評価のガイドラインも、一現在のところ一応のレベルに達しているものと思われるが一、改善の余地は多い。本報告書の内容は、あ

くまで一般的な意味でのガイドラインであり、本ガイドラインの利用にあたっては、各自が、自らの問題に応じて各項目の内容を吟味し、自由に項目を取捨選択して自分の問題に合わせていく必要がある。

今後、このガイドラインを実際に役立つものとしていくためには、次の課題に取り組む必要がある。これらの課題については、本委員会の第3年度の活動の中でも取り扱う予定であるが、他の機会でも、エキスパートシステムの評価について考察を続けて行くことは重要である。

(1) 要求仕様書とガイドラインの評価項目との関連の明確化

エキスパートシステム開発に伴って作成される要求仕様書とガイドラインで示した評価項目との関連性を明確にし、エキスパートシステムのライフサイクルに応じたレビュー項目・評価項目をまとめる必要がある。

(2) ガイドライン項目の選択基準・評価基準の設定

今日、エキスパートシステムの利用範囲はきわめて広く、採用される人工知能手法にもさまざまなものが存在する。そこで、エキスパートシステムを総合的に評価した結果を、（できれば定量的なレベルで）まとめる手法の確立が望まれる。そのためには、エキスパートシステム評価の具体的な手段、手法の明確化することによって、①エキスパートシステムの設計審査の指針を作成し、②エキスパートシステム評価のための手順書を取りまとめることが重要な課題となる。

(3) ガイドラインの評価

本報告書で述べたエキスパートシステム評価のガイドラインは、事例分析と膨大な調査とに基づいているとはいえ、現在のところ、机上のものにすぎない。本ガイドラインを有用なものとするためには、ガイドライン自身の評価・検証を行うことが重要である。

エキスパートシステムに対する評価を有用性という立場から厳しく行い、短期的な投資効果・利益を定量的にまとめていることが、最近、米国では、しばしば行われている[Schorr 1989]。確かに、エキスパートシステムを評価する

にはどれだけ利益が上がるかというのは重要な視点ではある。しかし、良く知られているとおり、実用的かつ有用なエキスパートシステムは、短期間で開発することはできないことが多い。したがって、その評価を行うためのガイドラインも短期的なものであってはならない。我々は、人工知能技術の応用システムは功利的な面のみで評価せず、長い目で夢をもった評価を行うことも重要ではないかと考える。そのためには、エキスパートシステムの評価手法は、将来的にはエキスパートシステムの社会的影響・経済的な影響までを考察の対象としなければならないであろう[Winograd 1986]。

【参考文献】

- [O'Keefe 1989] O'Keefe, R.M., O'Leary, D.E.: The Verification and Validation of Expert Systems, 11th IJCAI Tutorial, MP4, 1989.
- [Schorr 1989] Schorr, H., Rappaport, A. (eds.): Innovative Applications of Artificial Intelligence, AAAI Press, 1989.
- [Winograd 1986] Winograd, T., Flores, F.: Understanding Computers and Cognition, Addison-Wesley, 1986.

付 録

付録1 エキスパートシステム評価のヒアリング調査

エキスパートシステムの評価ガイドラインを検討するに際して、以下に示す内容のヒアリング調査を実施した。

(1) 第1回ヒアリング調査

- ・日時：1989年9月22日（木）
- ・講師：花王株式会社 知識・情報科学研究所
所長 美濃 順亮
- ・講義内容：
「開発事例からみたエキスパートシステムの評価についての私見」
 - ① E S の開発の社会的背景
 - ② E S の開発の技術的背景
 - ③ E S 実用化の現状
 - ④ E S の実用化要件
 - ⑤ E S の評価

(2) 第2回ヒアリング調査

- ・日時：1989年10月27日（金）
- ・講師：早稲田大学 理工学部 工業経営学科
教授 東 基 衛
- ・講義内容：
「プロトタイピングの品質保証／評価技術」
 - ① はじめに
 - ② 品質向上の技術
 - ③ 品質定量評価技術（Software Metrics）
 - ④ 品質保証／評価技術への影響要因
 - ⑤ プロトタイピングの評価
 - ⑥ プロトタイピングの評価事例

(3) 第3回ヒアリング調査

・日時：1989年10月27日（金）

・講師：石川島播磨重工業株式会社 メカトロ総合開発部

制御技術グループ 課長 小 出 誠 二

・講義内容：

「エキスパートシステムの評価方法（ドメインシェルと知識表現問題）」

①FTX (Fault Tree based eXpert system) の特徴

②FTXと一般エキスパートシステムの関係

③FTXの問題点と課題

④エンドユーザ向きのエキスパートシステム

⑤プラント運転支援システムの内容

⑥TNX (Transition Network based eXpert system) の概要

⑦ドメインシェルの開発と知識表現

(4) 第4回ヒアリング調査

・日時：1989年11月28日（火）

・講師：新日本製鉄株式会社

第一技術研究所 電子・制御研究開発センター

主任研究員 大 力 修

エレクトロニクス研究所 電子システム研究センター

高 田 寛

大分製鉄所 設備部 電気計装技術室

掛長 平 田 達 朗

・講義内容：

「高炉データ認識へのニューラルネットの適用とその評価」

①並列処理型神経回路シミュレータ

②Layered Structure of the Neural Network Simulator Ami

③高炉操業の特徴

④高炉へのAI適用

⑤高炉データ認識へのニューラルネットの適用とその評価

⑥まとめ

(5) 第5回ヒアリング調査

- 日 時：1989年11月28日（火）
- 講 師：NKK 福山製鉄所 プロセス制御部 技術室
青 木 太 一

• 講義内容：

「高炉エキスパートシステムとその評価」

- ① 高炉の設備概要
- ② システム開発の狙い
- ③ 高炉エキスパートシステム
 - 異常炉予知システム
 - 炉熱制御システム
- ④ 適用状況
- ⑤ 今後の課題

付録2 「エキスパートシステムの評価技術動向調査」海外出張報告

付録2では、1989年8月に米国で実施した、エキスパートシステムの評価技術動向調査ならびにAI技術の先端的研究の調査の結果をとりまとめる。この調査は、寺野委員長と鍛冶委員とが独立に実施したものである。以下、付録2.1には寺野委員長の、付録2.2には鍛冶委員の報告を述べる。

付録2.1 エキスパートシステムの評価技術動向調査(その1)

1. 調査概要

- (1) 調査期間: 1989年8月14日(月)～8月27日(日)(14日間)
- (2) 調査員: 寺野隆雄(筑波大学大学院経営システム科学専攻 講師)
- (3) 調査対象期間:

8月15日(火) カリフォルニア大学バークレイ(Berkeley)校

Professor. Ramamoorthy訪問

- Prof. Ramamoorthy, Prof. Tsai, Prof. Neuhold等とエキスパートシステム評価技術について討論。

8月16日(水)～17日 E P R I (米国電力研究所), Dr. Bill Sun 訪問

Intelli-Corp社, Dr. Hashemi 訪問

- Bill Sun と E P R I の AI 応用状況について、原子力部門の応用を中心に意見交換。
- Dr. Naserとエキスパートシステムの評価技術について意見交換。
- Dr. MayとAIツールPlexsysについて意見交換。
- Dr. HashemiによるPlexsysのデモならびに、意見交換。

8月19日(土)～25日(金) I J C A I - 8 9 (第11回人工知能国際会議)

ならびに知識ベースシステムの評価ワークショップ出席

2. 主要な調査結果

2.1 E P R I

E P R I の A I 研究は、短期的かつ電力会社に直接的に技術移転が可能なものを指向している。そのため、我国の長期的な視点に基づく研究開発計画とは、考え方に違いが存在する。特に、E P R I ではパソコンもしくは小型コンピュータを使った A I システムの実用化に熱心で、大型コンピュータを使うプロジェクトは現在に行っていない。

[エキスパートシステムの評価手法とPlexsysについて]

- エキスパートシステムの評価手法については、E P R I で開発したいくつかのエキスパートシステムに対する評価研究結果をもとに、一般的な基準を作成しようとしている。その中では、特に、軍事用ソフトウェアに対する基準、ならびに、N A S A における基準を参考に、各電力会社で利用できる基準を作成しようと試みている。この結果は、一応2年後にはまとまる予定である。
- 評価の難しさはあるものの、エキスパートシステム技術の優位性については、自信をもっている。
- 評価手法に対する考え方の確立には、まだ、しばらくの期間が必要と思われる。
- Plexsysは、K E E というエキスパートシステム開発ツールを改良し、原子力分野のエキスパートシステム開発の容易化をねらったもので、すでに数年間研究開発を続けている。日本での適用可能性について意見を求められたので、次のようにコメントした。

①分野別のツールとして、Plexsys の完成度は非常に高い。

②日本では、原子力分野でPlexsys を実用システムに利用する可能性はほとんどない。むしろ電力分野での適用可能性のほうが高い。また、化学プラント分野では、Plexsys の市場性は高そうである。

③E P R I のPlexsys に対する研究開発投資は膨大であるが、それが思ったほど電力会社に評価されていない。その原因は、Plexsys の稼働するハードウェア環境が高価なものに限られていること、ならびに、各電力会社の専門家が直接使うには、提供する機能がプリミティブすぎることに

の2点である。

2.2 IJCAI-88

第11回IJCAI (Eleventh International Joint Conference on Artificial Intelligence)が、8月20日から25日まで、米国ミシガン州デトロイトで開かれた。この学会は2年に一度開催される人工知能に関する国際会議であり、AAAI (National Conference on Artificial Intelligence)と並んで、この分野の主要な学会である。

本年は1200件以上の論文が投稿され、そのうち266件が採択された。参加者は、新聞によると4750名程度とのことであった。テクニカルセッションでの日本からの発表は、日本の企業から6件、大学から5件、米国の大学在籍の日本人から2件であった。テクニカルセッションにおける論文の発表の他に、6件の招待講演、5件のパネル討論、2日間にわたる24件のチュートリアル、AIベンダによる製品展示、16件のワークショップなどが開かれた。今年からの新しい試みとして、ビデオによる発表のセッションが設けられた。

以下では、チュートリアル、ワークショップ、テクニカルセッションの中から、エキスパートシステムの評価に関連する話題を取上げる。

(1) チュートリアル

2日間にわたって以下のような24件のチュートリアルが行われた。チュートリアルはその研究分野の2人の研究者によって行われ、講演、資料共に質が高く、得る所が大きい。そのうち、知識獲得、エキスパートシステムプロジェクトの管理、エキスパートシステムのV & Vの3つの内容を示す。

① SA6: Knowledge Acquisition for Knowledge-Based systems

John Boose(Boeing Computer Services)とMildred Shaw(University of Calgary)が講演を行った。知識獲得の方法について、その難しさ、課題などをまとめた後で、人間が行う場合の方式と、(半)自動化するための各種の方式/ツールを紹介した。ツールについては、65個のツールについて、その対象とする問題の性質、特定問題向けか一般的か、自動化の程度、使用のために必要な訓練の度合、守備範囲について分類し、全体的傾向を分析した。解析型問題(診断、分類、選択など)を扱うツールがほとんどで、合成型問

題（設計、計画、構成など）を扱うツールはほとんど無い。特定問題向けのツールの方が一般的なツールよりも多い。特定問題向けツールの方が自動化度が高いものが多い。自動化度が高いツールの方が使用のための訓練の必要度も少ない。その後で、主要な11個のツールについてケーススタディとして紹介をした。課題として、自動化・エキスパートシステム開発ツールとの統合化・複数の専門家からの獲得・知識の検査手法・学習機能の利用・文書解析技術の利用・ハイパーメディアの利用・ソフトウェア開発ツールとの統合化などがあげられた。

② MA5: Managing Expert Systems Projects

Richard Fikes と Neil Jacobsteinが講演を行った。彼らは、ともに、ビッグ4と呼ばれたAIビジネスのベンチャ企業でエキスパートシステム開発に携わってきた。現在ではこれらのベンチャ企業の景気は必ずしもよくない。したがって、この講演者は、AIビジネスの成功と失敗とに直接かかわってきたことになる。そのため、2人の話の内容は比較的つきなみなものであったにもかかわらず、その中には、経験に裏打ちされた迫力があり、講演が終了した時に大きな拍手が起こったのが印象的であった。ちなみに彼らが主張していたことは、研究とビジネスとは別物であり、“役にたつ”システム開発では技術的な冒険を避けなければならないということである。

③ MP4: Verifying and Validating Expert Systems

Daniel O'LearyとRobert O'Keefeが講演を行った。エキスパートシステムのテスト技術や検査技術が確立していない現状のもとでは、この内容は非常に講演しにくいものである。彼らの主張は、エキスパートシステムは、モジュラリティが高いという意味ではV&Vが容易だが、専門知識に根拠を求めているという意味では、V&Vはきわめて難しいということに集約される。彼らの紹介した技法の中では、統計的な手法を用いてシステムの評価を行うやり方が興味深かった。なお、本報告書4章のエキスパートシステム評価文献調査には、この講演で参照された文献が数多く紹介されている。

(2) 知識ベースシステムのV & V、テストに関するワークショップ

テクニカルセッションと並行してAAAI主催の数十のワークショップが開催された。そのうち、Workshop on Verification, Validation and Testing of Knowledge Based Systems の概要を示す。本ワークショップは、Kirstie Bellmannがチェアパーソンとなってこのテーマについて1日のワークショップが開催された。参加者は40人で、そのうち、あらかじめ定められた10人がポジションペーパーの発表を行い、それに、1または2人デイスカッサントとしてコメントしたのちに、自由討論に入るというスタイルで議論が進められた。

エキスパートシステムの評価 (Verification & Validation) は、従来システムに比べて簡単な面と難しい面との両方がある。しかし、具体的な方法については、さまざまな意見があり、はっきりとした方向性が見られなかった。ただし、技術的に確立しているルールシステムについては、さまざまなテスト手法が提案された。

そのうち興味が深いのは、ボーイングコンピュータサービス社、Baum, L.S., Shema, D.B., Boose, J.H., Bradshaw, J.M. の Acquiring and Verifying Control Knowledge for a Blackboard System という発表であった。これは、分類型問題むきの知識獲得支援ツール Aquinas を使って、ブラックボード型システム Erasmus の制御知識の獲得と効率評価を行うものである。すなわち Aquinas は専門家にインタビューしてブラックボード制御のための経験的知識を含む Erasmus sources を生成する。この制御知識は Erasmus のスケジューラによって適切な知識源を選択するのに使われる。専門家は Aquinas でテストケースを作る。これは Erasmus のレポートとマッチングをとって、制御知識の効率を測定する。Aquinas は適当な情報を変化させ、それが知識ベースの効率を改善するかどうかを確かめることによって、オリジナルな知識ベースを自動的に改善する。

また、宇宙用・軍事用のエキスパートシステムに対する基準を設けたいという参加者が多くみられた。そのため、Heavy Duty システムの評価を中心とした議論となり、ニューロ、ファジィなどの新しい技術を用いたシステムに対しては、評価手法の議論はまったくなかった。

(3) テクニカルセッション

近年のIJCAIおよびAAAIは、より基礎的で学際的な論文を受理する方針を打ちだしている。実用的な応用に近い論文はCAIAやIAAIなどのAI応用の学会に投稿することを推奨している。このため、発表される論文は大変専門的な場合が多く、その分野に精通した者でないと理解がなかなか困難である。

論文の内訳は、Foundations (3)、Tools (20)、Parallel and Distributed Processing (11)、Real-time and High Performance (7)、Search (13)、Automated Deduction (22)、Intelligent Tutoring Systems (3)、Cognitive Models (13)、Machine Learning (44)、Planning, Scheduling, Reasoning about actions (30)、Commonsense Reasoning (19)、Knowledge Representation (47)、Speech and Natural Language (16)、Vision and Robotics (16)である。

特に注目されるのは学習に関する論文の数が多いことである。これは、エキスパートシステムのブームが鎮静化しつつある一方、学習するシステムへの興味と期待が増大していることを裏付けている。また、学習の中でニューラルネットに関するセッションが設けられてコネクショニストのアプローチと従来のAIにおけるアプローチの差異が真剣に議論され出したのは、注目に値する。

知識獲得、機械学習の研究発表としては、1986年に発表されたEBLに関連するものが圧倒的に多い。EBLのような演繹的な学習では、学習によって知識が増えると、学習した知識も探索の対象になり、最終的に役に立たない場合には不要なバックトラックが増加することになりかえって効率が悪くなるという問題がある。これを避けるため、学習する知識の表現を制限したり、新しく学習した知識の使用方法を制限する方法が模索されているが、まだ決定打が出ていない。IJCAIではニューラルネット関係の発表は少ないが、今回ID3による帰納学習とBack Propagation法によるFeedforward Networkの学習との性能比較に関する発表が人目を惹いた。結論は月並みで、学習結果に大きな差はない、Neural Networkはノイズに強いが学習に時間がかかるということであるが、3件の発表が同じデータを使用していた点を注目したい。米国ではベンチマークに使えるようなデータを揃えようという動きがあり、その効果が出始めていると感じられた。この分野はエキスパートシステム構築方法論とも不可分の関係にあり、知識獲得と問題解決方法論を組み合わせた総合的な構築ツールを開発しようとする研究も盛んである。

また、今回ははじめての試みとして、主要な分野について、発表論文のまと

めのセッションが設けられた。これは、探索、帰納学習、計画、知識表現の4つの主要な分野について、IJCAI-89の発表を中心に研究動向をとりまとめたものであった。Quinlanによる帰納学習、Deanによる計画、Levesque等による知識表現の3つに参加したが、これらは、直前のセッションの発表までを含め、非常に優れたサーベイとなっており、直接発表を聞いていない場合でも、その内容と研究上の位置づけがよく理解できた。

付録2.2 エキスパートシステムの評価技術動向調査(その2)

1. 調査概要

- (1) 調査期間: 1989年8月22日(火)～9月3日(日)(13日間)
- (2) 調査員: 鍛冶勝三((財)日本情報処理開発協会 開発研究室主任研究員)
- (3) 調査対象機関:
 - ①IJCAI-89(8月23～25日, デトロイト)
 - ②MIT(8月29日, ボストン)
 - ③オハイオ州立大学(8月30日, コロンバス)
 - ④UCLA(8月31日, ロスアンゼルス)

2. 主な調査結果

2.1 IJCAI-89

IJCAI-89の調査結果に関しては、付録2.1を参照されたい。

2.2 マサチューセッツ工科大学(MIT)

2.2.1 調査概略

- (1) 訪問日時: 1989年8月29日(火) 10:00～12:00
- (2) 面会者: Prof. Ramesh S. Patil(AIラボ(Artificial Intelligence Laboratory))

2.2.2 AIラボについて

(1) 組織

Technology Squareビルの1階から3階までがコンピュータサイエンスラボで、そして、4階以上にはAIラボが入っている。Technology Square全体でAIの研究を行っている。Technology Squareに所属する学部・学科は、下記の通りである。

- ①経営学部
- ②化学工学科
- ③航空宇宙学科
- ④都市工学科

これらの4つが教育軸を形成し、そして、コンピュータサイエンスラボとAIラボが研究軸の2本柱を形成しており、この2次元により、組織化がなされている。各教授は、学部の仕事を50%、そして、研究の仕事を50%の配分で行っている。

る。A Iの研究に関しては、学際的に行われている。

(2) 研究対象

A Iラボの現在の研究対象は、下記の通りである。なお、()内は研究担当者名である。

- ①医療診断 (Ramesh S. Patil)
- ②ハードウェア診断
- ③自動プログラミング (Charles Rich, Richard C. Waters)
- ④学習 (Patrick H. Winston)
- ⑤分散処理, オブジェクトオリエンテッドエンバイロメント
(Carl Hewitt)
- ⑥シンタックスオリエンテッドコンピューテーション
- ⑦ビジョン, ロボティックス (Michael Brady, W. Eric L. Grimson,
Tomas Lozano-Perez, John M. Hollerbach, Rodney A. Brooks)
- ⑧自然言語処理 (Robert C. Berwick)
- ⑨TMS, JTMS (Justification-based Truth Maintenance
Systems) (J. Doyle, D. McAllester, Johan de Kleer)

(3) 施設見学

Patil教授の案内で、下記に示すA Iラボのコンピュータ施設を見学した。

- ①データフローマシン
64台のT I製Explorerマシン (Lispマシン) を接続し、Multi-Lisp環境を提供している。
- ②コンサートマシン
Sussman教授のグループが開発した並列処理研究用のマシンである。
- ③コネクションマシン: CM1
非常に多数の小規模プロセッサ (64,000台) を結合したSIMD型マルチプロセッサである。言語としては、*C (拡張C) と*Lisp (拡張Lisp) を提供している。CM2も生産されている。
- ④ロボット
完全自走で、処理系内蔵のロボットである。
- ⑤4指ロボット
- ⑥3D認識システム (両眼)

2.3 オハイオ州立大学

2.3.1 調査概略

(1) 訪問日時: 1989年8月30日(水) 10:00~12:00

(2) 面会者:

• Prof. Balakrishnan Chandrasekaran

(コンピュータ & 情報科学科 LAIR)

• Dr. John R. Josepson

(コンピュータ & 情報科学科 LAIR)

LAIR: Laboratory for Artificial Intelligence Research

2.3.2 Balakrishnan Chandrasekaran教授について

Chandrasekaran教授は、1963年に、インドのマドラス大学電気工学科を卒業し、1967年に、米国のペンシルバニア大学電気工学科で博士課程を卒業した。

彼は、過去10年間、オハイオ州立大学コンピュータ & 情報科学科 LAIR で研究を行っている。

彼の研究の主題は、計算的な観点から、知能がどう働くかを理解することである。特に、問題解決、理解、そして説明のような高位の認知活動によって明示される知能に興味を持っている。

2.3.3 LAIRでの研究内容について

LAIRの研究チームは、知能に関する理論を構築し、この理論を採用した社会的に、そして技術的に有用なツールの開発を目指している。

知識に基づく問題解決に関する研究では、ジェネリックな情報処理タスクを扱っている。ジェネリックタスクは、知識に基づく各種の推論での基本的推論の1つである。各ジェネリックタスクは、入力/出力により機能的に定義され、そして、タスクに固有の問題解決に必要とする知識及び制御によって特徴付けられる。ジェネリックタスクを同定することは、知識に基づく推論のフレームワークを提供することになる。

ジェネリックタスクアプローチは、プロダクションルール、フレーム、意味ネット、そして述語論理よりも、より高い抽象レベルで記述された知識を使用してESを設計可能にするツールを自然に導き出す。LAIRでは、現在、この方法論に基づいた新世代のAIツールを開発する挑戦的なプログラムを実施している。また、ジェネリックタスク及びそのツールを越えるのに、装置の機能を理解するための“深い”モデルと、そのモデルを記述するための機能表現言語についても

研究中である。

医療診断、工学システム診断、そして設計／プランニングのような複雑な実世界での推論について研究することにより、知能の基本的なビルディングブロックを構成するジェネリックな情報戦略を同定できると考えている。また医療及び工学のたぐいさんの実用分野で使われる複雑な推論システムの構築も行っている。

2.3.4 ジェネリックタスクについて

Chandrasekaran教授の研究チームでは、問題解決のジェネリックタスクとして、下記のタスクを抽出し、提案している。

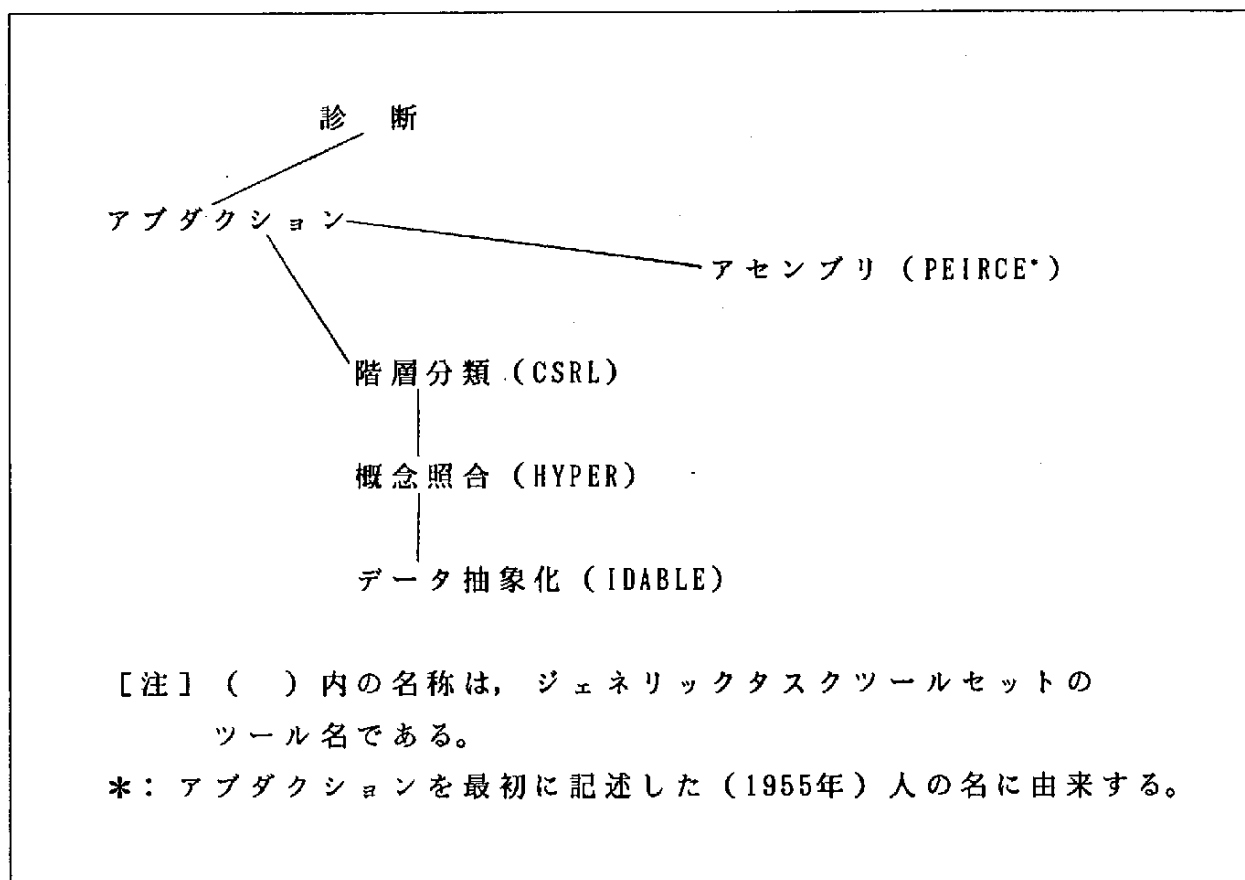
- ①階層的分類 (Hierarchical classification)
- ②プラン選択／洗練化 (Plan selection and refinement)
(ルーチン設計／ルーチンプランニング用)
- ③概念照合 (Concept matching)／仮説照合 (Hypothesis matching)
- ④知識指向の間接推論 (Knowledge-directed indirect inference)
(知的データ抽象化／検索用) (単に“データ抽象化”とも呼ばれる)
- ⑤状態変化の抽象化による予測 (Prediction by abstracting state changes) (単に“状態抽象化”とも呼ばれる)
- ⑥データ説明用仮説のアセンブリ及び合成仮説の批評 (Assembly and criticism of composite explanatory hypotheses)

2.3.5 診断推論について

診断問題は、本質的にはアブダクション (abduction) の問題である。

アブダクションとは、何かを述べているデータから、そのデータを最もよく説明する説明用仮説 (explanatory hypothesis) を推理することである。

診断推論とジェネリックタスクの関係を図付2.2-1に示す。なお、図中の () 内の名称は、Chandrasekaran教授の研究チームが開発したジェネリックタスク専用のツール名である。



図付2.2-1 診断推論のジェネリックタスクと使用するツール

2.3.6 装置の表現法について

装置をどう表現するかの問題が重要である。そして、装置がどのように動いているかのモデル、即ち、人間がそれをどう見ているかのメンタルモデルが重要である。

装置の表現法としては、下記の3通りがある。

- ①機能 (Function)
- ②構造 (Structure)
- ③挙動 (Behavior)

そして、装置の構造を知って、挙動を推論する。そのために、Chandrasekaran 教授の研究チームでは、機能表現言語 (Functional Representation Language) の研究を行っている。

2.3.7 Consolidation理論について

Chandrasekaran教授の研究チームでは、Consolidation理論の研究を行っている。

ある確実な方法で結合された構成要素の集合体を与えた場合、各構成要素に関する個別の挙動の記述から、集合体（装置）に関する全体の挙動の記述へ、どうやれば変換できるのか。Consolidationと呼ばれるこの情報処理タスクでは、従来は、代替手段として、装置に関する定性推論としてもっとも普通の手法であるシミュレーションを必要とした。Consolidation理論は、装置の挙動の概念的表現に基づいている。構成要素の挙動は、少数個の挙動の型のプリミティブ（primitive）を使用して表現される。装置の挙動は、ある型の挙動が別の型の挙動を構造的にどのように組み合わせれば実現できるかを規定した結合則（rule of composition）を使用して、推論される。

2.4 カルフォルニア大学ロサンゼルス校（UCLA）

2.4.1 調査概略

（1）訪問日時：1989年8月31日（木） 10:00～13:30

（2）面会者：

- Dr. Richard E. Korf
（コンピュータサイエンス科）
- Dr. Josef Skrzypek
（コンピュータサイエンス科MPL）
- Prof. Judea Pearlの代理の人
（コンピュータサイエンス科CSL）
- Dr. Rina Dechter
（イスラエル工科大学コンピュータサイエンス科）
- Dr. Michael G. Dyer
（コンピュータサイエンス科AIL）

CSL: Cognitive Science Laboratory

AIL: Artificial Intelligence Laboratory

MPL: Machine Perception Laboratory

2.4.2 UCL AでのAI研究内容について

Richard E. Korf博士によると、UCL Aのコンピュータサイエンス科、C S L, A I R, そして、M P Lで行われているAI研究の担当者及び研究内容は、下記の通りである。

研究者名	研究内容
• Judea Pearl	不確実性推論
• Michael G. Dyer	ニューラルネットによる自然言語処理
• Josef Skrzypek	マシンによる知覚
• Jacques Vidal	ニューラルネット
• Allen Klinger	コンピュータビジョン, 画像処理
• Michael Melkanoff	製造用ロボット
• Stoff Parker	ロジックプログラミング
• Richard E. Korf	発見的探索

2.4.3 発見的探索について

Korf博士より、以下のお話を聞いた。

発見的探索を必要とする問題は、次の3つに大別される。

① 2人ゲーム問題

〔例〕チェス, 囲碁, チェッカー, オセロ等

② 単独行為者の経路発見問題

〔例〕ルービックキューブ, エイトパズル, 巡回セールスマン問題,
道路のナビゲーション

③ 制約充足問題

〔例〕エイトクイーン問題, 地図の色塗り分け問題

また、これらの問題は、次の目標を達成する必要がある。

① 2人ゲーム問題 勝つこと

② 単独行為者の経路発見問題 最小コスト

③ 制約充足問題 制約を満足する解/状態を1個見つけること

そのためのキーポイントは、次の通りである。

① 2人ゲーム問題 発見的評価関数 (H E F: Heuristic Evaluation Function)

② 単独行為者の経路発見問題 2点間のユークリッド距離 (Air Line Distance)

③ 制約充足問題 バックトラック

2.4.4 Neural Networks for Subcognitive Computing について

Josef Skrzypek博士より、以下のお話を聞いた。

コンピュータサイエンス科のMPLでは、ニューラルネットワークを応用した機械による知覚及び視覚の研究を行っている。

下記に、お話の内容を簡条書で示す。

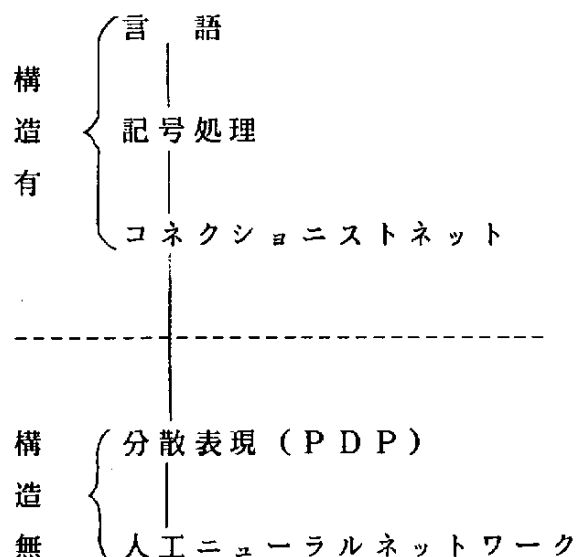
- ① Cognitive Computingは自然言語や推論を、一方、Subcognitive Computingはビジョンやマニュブレーションのように、あまり高級でない機能が要求されるリアルタイムロボットを対象にしている。
- ② UCLAのMPLでのLANで結合されたコンピュータ施設のマップとコネクションマシンの紹介があった（図付2.2-2参照）。
- ③ 視覚の神経回路モデルとして、下記の3層構造（階層）のモデルを採用している（図2.2-3参照）。
 - a. 光受容器層（Layer of photoreceptors）
 - b. コントラスト検出層（Layer of center-surround receptive fields:
（強いて訳せば、例えば、）オン中心型でオフ周辺型の受容野の層）
 - c. 直線検出層（Layer of 0 deg. oriented edge detectors: 0度の傾き
（方位）のエッジの検出器の層）
- ④ コネクショニストコンピューティングを応用してSFINX（Structure and Function In Neural Connection）と呼ばれるニューラルネットワークシミュレータを実現した。今後は、ハードウェア化を目指しており、最初のニューラルコンピュータとなるであろう。
- ⑤ 人工知能 + ニューロ科学 → PDP（Parallel Distributed Processing）

2.4.5 ニューラルネットワークによる自然言語理解について

Michael G. Dyer博士より、以下のお話を聞いた。

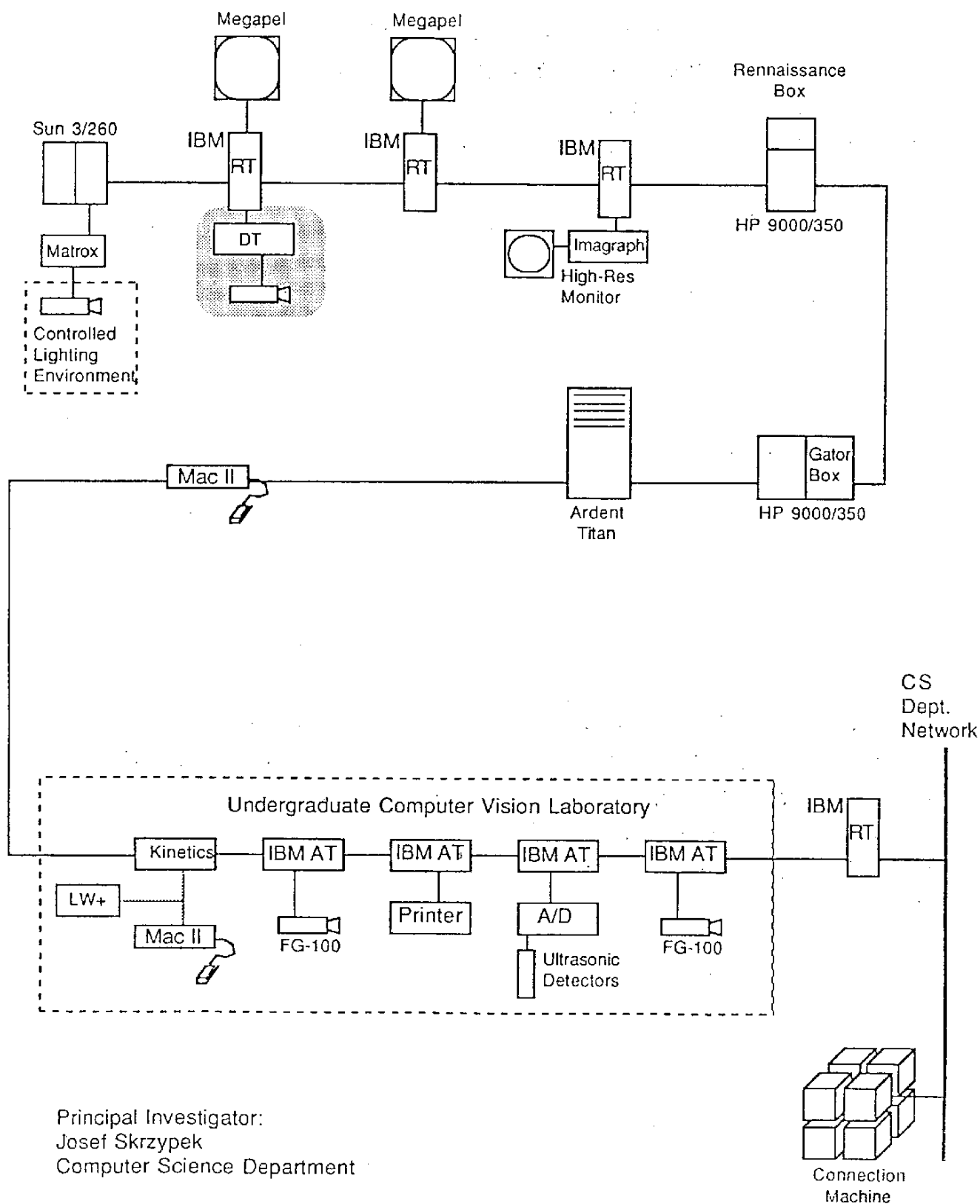
Dyer博士は、自然言語理解にニューラルネットワークを応用しようとしている。

自然言語の理解は、下記の階層構造で行われる。



先ず、必要とする単語や文章を前もって学習させ、ニューラルパターン、即ち、ホログラフィック表現 (Holographic representation) に前もって変換蓄積しておいて、短い小説を読ませ、推論させて、文章を理解させる。

Dyer博士のグループは、ニューラルネットワークだけでなく、上記に示した階層構造のすべてに対して研究を行っている。



図付2.2-2 UCLA MPLのコンピュータ構成のマップ

Sample Simulation

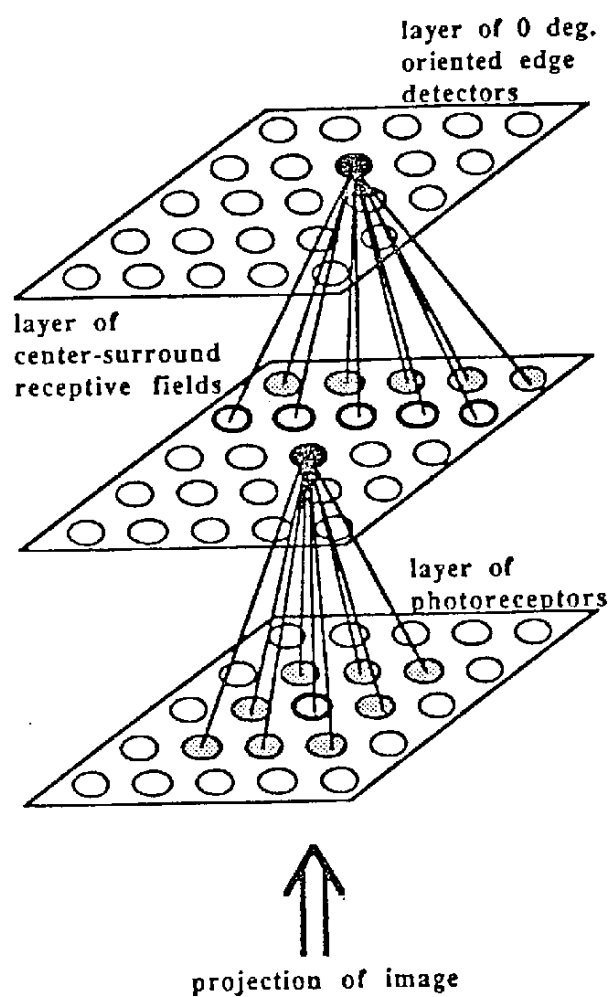


figure 3.

〔左図〕 3 階層モデル

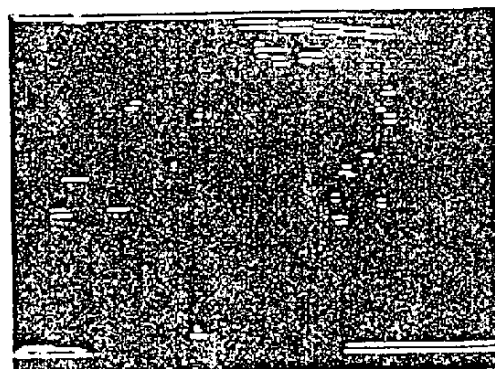


figure 3.c

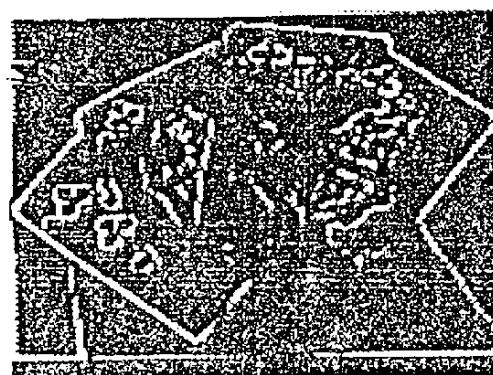


figure 3.b

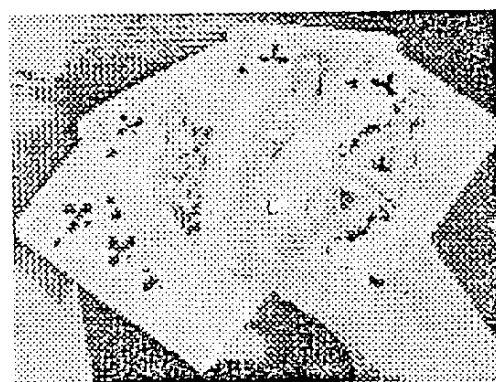


figure 3.a

〔右図〕 反 応 図

図付2.2-3 視覚神経回路の3階層モデルと各層での反応

—— 禁 無 断 転 載 ——

平成 2 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
機会振興会館内
TEL 03(432)9372

印刷所 山陽株式会社
東京都港区虎の門 1 丁目 9 番 5 号
TEL (591) 0240 (代表)

