

設計と診断を融合したシェルの構成

7K-1

— 基本構想 —

山口 高平[†] 溝口 理一郎[†] 元田 浩[‡] 豊田 順一[†] 角所 収[†][†] 大阪大学 産業科学研究所 [‡] 日立製作所 基礎研究所

1. はじめに

現在、従来の診断型ES(エキスパートシステム)の欠点を克服するために、「知的な人間の専門家」の挙動をシミュレートすることを目的とした「深い知識」に基づくESの開発に関心が集まってきた[1-3]。深い知識をESに導入する利点は、以下の通りである。

- ・ システム構築時には予期していなかった状況にも対処可能となり性能が向上する。
- ・ 新しい状況に対処した時に用いた知識をルール形式にまとめ上げ、浅い知識ベースに格納する行為は、経験を通して知識ベースを拡張していくという学習機能を実現する上で要素技術となる。
- ・ ルールの正当性の提示が可能となり、初心者にも理解できる優れた説明機能が実現できる。
- ・ 対象領域に関するシステムの理解が人間の理解に近く、認知科学的に見て優れたシステムの構築が可能になる。

しかしながら、深い知識に基づくESには利点ばかりではなく、以下の幾つかの疑問が投げかけられている。

- (1) 技術が確立されるのか?
- (2) 診断以外に有効なタスクが存在するのか?
- (3) 浅い知識ベース(経験則)を完全にする負担より、深い知識ベースを構築する負担の方が本当に小さいのか?

本稿では、深い知識および深い推論を浅い知識と浅い推論に対比させて分析することにより、これらの疑問に答えるとともに、設計と診断を融合したメカ専用のシェルを提案する。

2. 浅い知識と深い知識

本稿では、タスクに直結した知識を浅い知識と捉え、浅い知識を生成することに関連した知識を深い知識と捉える。「浅い・深い」という概念は、上記の定義からも分かるように、タスクによって決定されるものであり相対的に変わるもので絶対的に決定できるものではない。

例えば図1では、タスク1においてはAが浅い知識でBが深い知識となるが、タスク2においてはBが浅い知識でCが深い知識となる。

Van de Veldeは文献[4]に於て、深い知識の持つ特徴として、①ドメイン依存 ②タスク独立 ③構造化を挙げて一般的に論じているが、深い知識は後述するように分化するため、雑駁な議論になっていると言わざるを得ない。

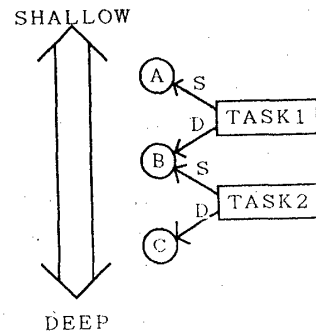


図1

深い知識と浅い知識

表1. 深い知識の特徴

項目	ワールド	P	D	C	I
ドメイン独立		○	×	×	○
タスク独立		○	△	△	△
構造化		○	○	○	×

表2. 推論法の比較

項目	推論法	浅い知識	深い知識
ドメイン独立		○	○
タスク独立		○	×
構造化		○	×

筆者らは既に文献[5, 6]において、深い知識を「障害原理」という Physical Worldと Device Worldの2種類のワールドに分け、これらの深い知識から診断ルールを自動生成する方法を検討しており、さらに文献[7]では深い知識を整理して、深い知識は以下の4種類のワールドに分けられている。

1. Device World
2. Control World
3. Physical World
4. Interpretation World

以上の事から、深い知識の特徴を一様に述べることは不可能であり、深い知識の特徴を分析すれば、表1の通りになる。

すなわち、Physical Worldだけがドメイン・タスクを変えても共通に使える知識であり、Device WorldとControl Worldは対象に関連した知識であるので、ドメインが変われば取り替える必要があり、タスクに対しても完全に独立でない。例えば、車の故障診断において利用するドメインの知識としては部品の耐久性および重要性等があるが[7]、この知識が車の運転操作支援にも利用されるとは考えにくい。また Interpretation Worldは、タスクが変わっても共通に使える場合があると考えられるが、基本的には診断とか運転支援等のタスクに依存した知識であるため、タスクが変われば取り替える必要があり、またローカルな知識であり構造化も成されていない。

Expert System Shell for Designing and Troubleshooting of Machine -Conceptual Design-

Takahira Yamaguchi¹, Riichiro Mizoguchi¹, Hiroshi Motoda², Junichi Toyoda¹, Osamu Kakusho¹

1. ISIR Osaka Univ. 2. Advanced Research Lab. Hitachi Ltd.

3. 浅い推論と深い推論

浅い知識を適用してタスクを効率よく実行するプロセスが浅い推論であるのに対して、深い知識から浅い知識を生成するプロセスが深い推論である。両者の推論比較は、やはり Van de Veldeが行っており、表2に示す通りである[4]。

ここで、深い推論のタスクへの依存性を考える。例えば、車の故障診断ではその浅い知識（ルール）の特徴上、徴候から故障仮説またその逆が生成される必要がある。この場合、徴候は温度や圧力であるが、ある基準値を越えているか下回っているかが主問題となるため、constrain propagation（制約伝播）で処理できる範囲が相当あると考えられる。一方、運転操作支援では、“右に曲がりたい”等のユーザの欲求から操作法を生成する必要があるため、操作は時間的に変化し、制約伝播の他にprediction（予測）の技法が必須となる。このように生成される浅い知識の内容が違えば、その生成法も変わるため、深い推論はタスク依存であることが分かる。

以下、診断における深い推論エンジンの特徴について述べる。本推論エンジンは文献[9]に詳しいが、構造等の情報を使って、すなわち Device World だけの知識により状態を遷移させるプロセスの他に、推論エンジンが Physical World から、適切な constraint を状況に応じて逐次選択するプロセスから成る。通常のconstraint systemと比較すれば、対象物に関連する constraint を予め与えるのではなく、適切なconstraintの集合を推論エンジンが自動的に決定していると言える。またPhysical Worldの物理式は、値が与えられる（原因となる）変数が右辺に、値が計算される（結果として求まる）変数が左辺に置かれ、変数の因果関係[8]が推論に利用され易い形式で記述されている。

4. 設計と診断を融合したシェル

2. で述べた各ワールドの源は図2に示す通りで、Device WorldとControl Worldは、従来のCADとは大いに異なる設計対象モデルから得られる深い知識であり、Physical Worldと Interpretation World は、テキストレベルあるいは常識人から得られる深い知識である。この図は、あるメカを設計すれば、直ちにそのメカの診断が可能となるシェルを表したものである。ここで、1. の深い知識に基づくESに投げかけられた疑問について考察する。タスクと技術については、（診断、propagation）および（運転操作支援、propagation

+ prediction）という関係が明らかになり、前者についてはかなり見通しの立つ技術である。また、知識ベースの構築負担については、DeviceWorldとControl Worldが設計プロセスから獲得され、他のワールドも基本的にテキストから獲得されるため、深い知識ベースの構築負担は小さいと考えられる。

5. おわりに

本稿では、ドメインおよびタスクを変えれば、シェルの知識と推論法のうち、何がそのまま利用できて何が利用できないかを分析するとともに、深い知識に基づく新しいシェルのあり方について論じた。今後、幾つかのドメインおよびタスクを通して、本シェルの有効性を検証していくつもりである。

参考文献

- [1] P.E.Hart : " Direction for AI in the eighties " SIG ART, 79, p.79 (1982)
- [2] D.Michie: " High-road and low-road programs " AI Magazine, 3(1), pp.21-22 (1982)
- [3] B.Chandrasekaran and S.Mittal : " Deep versus compiled knowledge approaches to diagnostic problem-solving " Developments in expert systems edited by M.J.COOMBS, pp.23-34, Academic Press (1984)
- [4] Van de Velde, W. : " Explainable Knowledge Production " 7th European Conference on Artificial Intelligence, pp.8-22 (1986-7)
- [5] 山口、溝口、杉原、小高、野村、角所 : " エキスパートシステムにおける高級な説明機能 " 情報処理学会第30回全国大会, pp.1151-1152(1985-3)
- [6] 山口、溝口、小高、野村、豊田、角所 : " 深い知識を利用した知識コンパイラの構成 " 電子通信学会、人工知能と知識処理研究会資料、AI86-7 (1986-4)
- [7] 田岡、山口、溝口、豊田、角所 : " 設計と診断を融合したシェルの構成 -設計対象モデルの生成- " 情報処理学会第33回全国大会, 7k-2 (1986-10)
- [8] Y.Iwasaki and H.A.Simon : " Causality in Device Behavior " Artificial Intelligence, Vol.29, No.1, pp.3-32 (1986-6)
- [9] 小高、山口、溝口、野村、豊田、角所 : " 設計と診断を融合したシェルの構成 -診断ルールの生成- " 情報処理学会第33回全国大会, 7k-3 (1986-10)

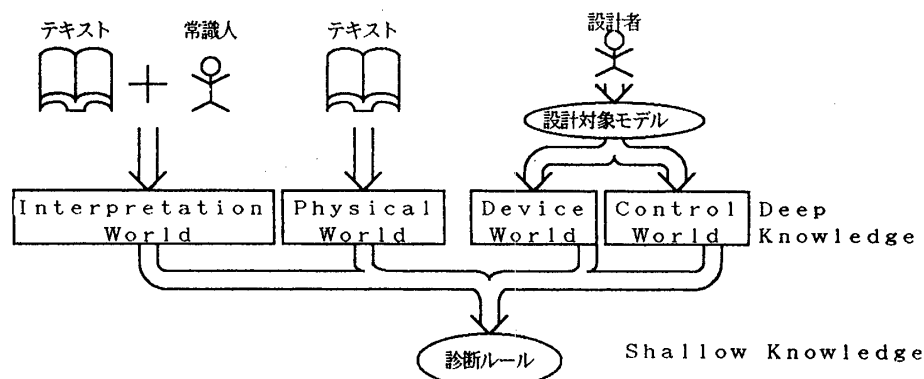


図2

設計と診断を融合したシェル