

3 K-8 知的システム構築用シェル AUK の構築支援環境 (1) その設計概念

○藤村 茂 富田昭司 桑原修二

横河電機株式会社

1 はじめに

現在、エキスパートシステム構築用ツールは、専門家の知識をいかにして獲得していくかについての様々なアプローチのもとで開発が進められている。そのために、対象問題のある程度特化することによって、専門家が扱いやすく、さらに高度な機能を提供する種々のドメインシェルが開発されている。しかし、このようなツールは、一般性が低く、ある限られた領域の問題以外には適用することが難しい。そのため、アプリケーションを単なる推論システムとしてではなく、知的なシステムとして考えた場合、それを利用してシステムを作りあげていくことは容易ではない。

また、専門家の漠然とした知識を整理していく手法として、最近、いくつかの問題分析のための方法論が提案されている。しかし、このような方法論は、ツールの機能として備わっているものではない。これは、その方法論自身が知識表現/推論機構と密接な関係になく、より上流過程の開発手法を定式化することを意図しているためである。

そこで我々は、ツールの開発にあたって、方法論を踏まえた知識表現/推論機構が必要であり、さらによりよい構築環境を考慮した方法論が必要であると考え、それらを一体化して考察を行った。そして、オブジェクト指向知識表現 auk (autonomic knowledge unit: 自律的知識単位) を提案している [1]。この概念を用いることにより、方法論の側面からは、トップダウン設計の一般化により単一パラダイムを実現し、ドメインシェルへの拡張を考えた場合にも同一パラダイムとして扱える利点が得られた。

本稿では、この方法論に従った構築支援環境の設計概念とその構成方法について示す。さらに、[2] で問題の整理からデバッグまでを支援する知的システム構築環境と、[3] でこの枠組み内で実現したグラフィカルユーザインタフェース構築環境についても言及する。

2 設計概念

従来から知識ベースの構築手法として、ボトムアップ的に宣言的な知識をまとめあげていくといった手法が提唱されている。また、現在ではインタビュー方式や KJ 法により断片的知識の整理を行っていく方法もいくつかのシステム上に実現されている。しかし、実際のシステムの構築においては、トップダウン的な問題の解析/設計といったものも重要であると考えられ、トップダウン的な構築手法もいくつか提案されている。

一方、現在のドメインシェルは、ある定められた枠組みの中で宣言的に知識を定義していくシステムとして見る事ができる。この定められた枠組みというものは、ある限られたドメイン固有のシステム構築の方法論に準じたものである。さらにそのような枠組み内で入力される宣言的な知識は、入力形態の容易さを実現するために、知識獲得の研究までに領域を広げている。しかし、このようなドメインシェルは、領域の少しでも異なる対象に対しては拡張性が低いために対応が難しい。

そこで、知的システムの構築においては、トップダウン的な設計/ボトムアップ的な構築の2つの手法を相入れ、適時組合せ用いることができる環境を用意する必要があると考えられる。さらに、このような構築過程は、問題の整理/モジュール設計/内部実現/テスト/デバッグに分割されるが、それらの過程および過程間での試行錯誤性も重要な要素となり、それをサポートするような構築環境を実現することも必要である。

3 知識表現/推論機構

オブジェクト指向知識表現 auk では、全体の知識をいくつかの部分的な知識に分解し、それらの知識が協調して問題を解決していくというモデルに従って知識が表現される。この部分的な知識は、状態を表す情報とそのオブジェクトの扱い方を表す情報のかたまりとしてのオブジェクトとして考えられ、これらを aukAccessor と呼ぶことにする。そして、これらのオブジェクトに対して統一した外部インタフェースプロトコル (情報の追加/削除/取り出し) を定め、このプロトコルに基づいてメッセージ

送信を行うことによって知識間で情報交換が行われるものとする。このように考えることによって、内部ではどのような推論が行われようと、外部的には全ての知識を同じように扱うことが可能となり、情報隠蔽がなされる。そのためにモジュール性/統合性のよい知識ベースが構築される。

4 方法論-構築環境

以下では、知識表現 auk を用いた場合、どのようにこの方法論に基づいた構築環境を実現していくかを示す。

まず、トップダウン設計について考えてみる。 aukAccessor が外部インタフェースプロトコルを統一していることから各 aukAccessor を同次元上で考えることが可能となる。そのため、ユーザにとっては各種アプリケーションを考える場合、単一パラダイムのツールとして環境を操作することができる。また、ドメインシエルを考えた場合も、そのドメインに応じた aukAccessor を用意することで、同一の環境を用いることができる。そこで、構築環境としては、 aukAccessor の外部インタフェースを設定し、それらの間を結び付けるための構築環境のみが必要である。しかし、支援機能として、問題に応じて部分問題への分割を知的にサポートしてくれるような機能を組み入れることも必要であろう。

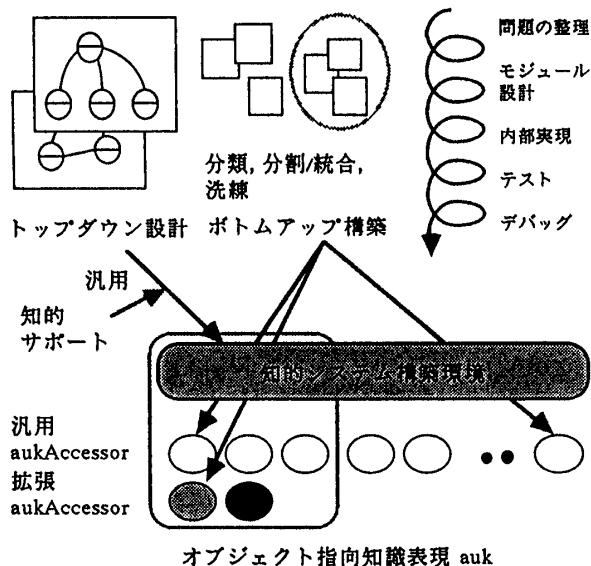


図1: 知的システム構築環境

ボトムアップ構築については、宣言的に記述された知識をいくつかの aukAccessor にまとめあげることが必要で、そのためにはそれらの知識の粒度に応じた分類、分割/統合、洗練などを支援する機能が要求される。さらに宣言的知識を自由に操作するグラフィカルな環境も必須である。

このようなトップダウン設計/ボトムアップ構築の枠組み内で、知的システムを構築していくわけであるが、そ

のような操作を試行錯誤的に行えるような環境も必要である。例えば、 aukAccessor 内であるいは aukAccessor 間で部分的な実行によるデバッグが適時できることが望ましい。また、内部実現されていない aukAccessor 間では、モジュールのスタップ化によりモジュール間のメッセージ送信が確認できることが望ましいと考えられる。

5 システム構成

このような観点のもとで、図1のようにシステムを構成した。現在のシステムは、以下のような汎用的な aukAccessor を用意している。これらを組み合わせて知的システムを構築していく。

aukAccessor	以下の要素の抽象的な概念。設計の段階でスタップとして用いる。
aukBag	要素として auk をもつ集まり (要素の重複を許す)。
forwardAukBag	ルールの前向き推論用の aukBag。
backwardAukBag	ルールの後向き推論用の aukBag。
aukSet	要素として auk をもつ集合 (重複不可)。
frameAukSet	フレーム表現用の aukSet。
conceptualAukSet	aukAccessor を要素とし、概念の階層表現に用いる。
proceduralAuk	外部インタフェースプロトコルの呼び出しによって手続きを起動する aukAccessor。
userAuk	入出力操作を実現する。

6 まとめ

このように知的システム構築用シエル AUK は、問題の整理/モジュール設計/内部実現/テスト/デバッグの過程における知的システム構築方法論、知識表現/推論機構、グラフィカルな構築環境を一体化して考察し開発した統合化汎用シエルおよびドメインシエルの枠組みとして考えられる。

参考文献

- [1] 藤村: オブジェクト指向知識表現 auk を用いた知的システム構築用シエル AUK, 情報処理学会論文誌 Vol.31 No.1(1990)
- [2] 富田: 知的システム構築用シエル AUK の構築支援環境 (2) 知的システム構築環境, 情報処理学会第41回全国大会 (1990)
- [3] 桑原: 知的システム構築用シエル AUK の構築支援環境 (3) グラフィカルユーザインタフェース構築環境, 情報処理学会第41回全国大会 (1990)