

知的システム構築用シェルAUK(1)
— 知識構築方法 —

3P-7

藤村 茂 富田 昭司 飯間 昇 鈴木 明
横河電機株式会社

1. はじめに

知的システム構築用シェルAUKは、知的なシステムの効率のよい開発を目的とし、知識の構築、および、開発されたシステムの利用において、優れた操作性を実現するという観点から設計したツールである。AUKは、オブジェクト指向知識表現auk (autonomic knowledge unit: 自律的知識単位) [1] を用いて作成されている。そのため、モジュール性/再利用性/統合性のよい知識の構築が可能となっている。また、この知識表現は、種々の表現に対応すべく、容易に拡張できるように設計したため、知識を柔軟に表現することができる。さらに、AUK上での知識の構築は、知識表現aukの特徴を生かし、グラフィカルな知識構築環境[2]内で、インタラクティブに進めることができ、効率的な知識の構築が可能である。

本稿では、知的システム構築用シェルAUKについて概説し、その知識構築方法として、top-down-planning/bottom-up-constructionという手法の構想を示すことにする。

2. 知的システム構築用シェルAUK

2.1 オブジェクト指向知識表現auk

知識表現aukでは、知識の単位を、その知識が有する静的な情報と、その知識を扱う動的な情報のかたまりとしてのオブジェクトと考えることにする。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を定め、この仕様のもとでのメッセージ・パスによって、知識間の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り、内部で、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でき、情報隠蔽がなされている。そのため、モジュール性/再利用性のよい知識の構築を行うことができる。また、統一した外部インタフェースによって、推論方法の異なる知識を統合性よく構築することが可能となる。そのため、全体の知識を、種々の知識に分割し、それらが協調的に問題解決を行うように、自然に表現できる。

この知識を表すオブジェクトのことをaukと呼ぶ。次に、aukの基本要素とその外部インタフェースについて示すことにする。

1). aukAtom

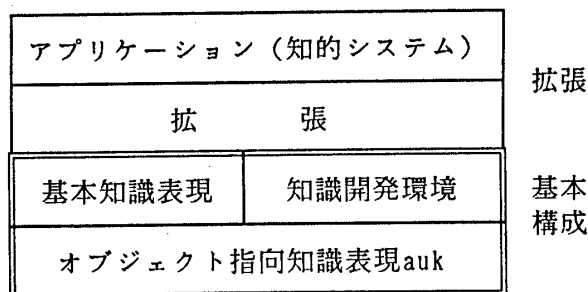
aukの基本単位。スロット名に対してスロット値、デーモンを持つ。外部インタフェースとしては、スロット参照/代入用メソッドが定義される。

2). aukSet

要素としてaukを持つaukの集合。aukの追加/削除/抽出用のメソッドが定義される。単純には、集合演算であるが、その作用の拡張として種々の推論機能が付加される。たとえば、ルールの前向き推論は、aukの追加の副作用としてルールが発火され、推論が行われるように実現される。推論が行われると、そのaukSetの要素は、変化するかもしれない。後向き推論は、aukの抽出を行おうとし、そのaukにマッチングする要素を陽に持っていない場合、ルールが発火されるように実現される。

2.2 構成

知的システム構築用シェルAUKの構成を下図に示す。シェルの基本構成は、知識表現auk上の基本的な知識表現と、その知識構築環境からなっている。基本知識表現としては、現在、ルールの前向き推論/後向き推論aukSet、フレームaukSetのみ用意している。この基本構成において、ユーザは、問題に適した知識表現への拡張、自分自身、あるいは、作成システムの利用者に適した知識構築環境への拡張を行う。そして、そのようなシェルを用いて、知的システムの開発を行うものとする。



3. 知識構築方法

シェルを用いて知的なシステムを作成するためには、まず、表現したい知識を整理し、その整理された知識を、シェル上に表現する方法がとられる。一般的なシェルを用いる場合には、そのシェルの表現の制約により、かなり窮屈な枠組みのなかで、知識の整理を行わなければならない。知的システム構築用シェルAUKを用いる場合には、

”全体の知識を部分的な知識に分割し、それらの知識を、それに適した方法で表現する”というモデルに従って知識を構築していく。その知識を構築する手法を以下に述べる。

3.1 top-down-planning

ある知識を記述する場合、まず、その知識を部分的な知識に分割する。そして、それらの知識に対して、追加／削除／抽出の外部インタフェース仕様を定め、機能を規定する。このような部分知識は、その機能を実現するために、さらに、部分知識への展開、あるいは、bottom-up-constructionによって、詳細化がなされていくものとする。しかし、この段階では、その方法については考慮しない。これらの知識には、知識利用の制約や、推論状況に応じた推論の切り替えなどの、メタな知識も含み、全ての知識が、同じように、部分知識として展開され、外部インタフェースを通じて結びつけられる。

3.2 bottom-up-construction

機能の規定された知識は、aukSetによって実現される。これらの知識は、プロダクション・ルール、あるいは、フレームで表現したかったりする。また、ある時は、あいまい性をもたせたかったり、単に、表計算のみを行わせたかったりする。そこで、目的に一致した推論機能を持ったaukSetが存在する場合は、そのaukSetを用いる。存在しない場合は、既存のaukSetを差分プログラミングによって、機能拡張を行い、目的に応じたaukSetを作成する。

このaukSetが、例えばルールを用いた推論を行う知識の場合は、次に、ルールの記述を行う。この際、基本的なルールをまず記述し、部分的な挙動を確認する。さらに、ルールを詳細化し、規定された機能を実現するように、完成度を増していく。このような作業を、スムーズに進めるためには、部分的に実行ができ、エディット／実行／デバッグのフェーズが、互いに密であるフェーズレスな知識構築環境が必須である。このように、個々の知識内の挙動を実現していくためには、ボトムアップに作業を進めていく。

4. 適用例

Smalltalk-80上に作成した知的システム構築用シェルAUK上で作成した、蒸留塔運転訓練用CAIを例に、その知識構築の過程を示す。

まず、システムの開発に先駆けて、提供されるべき機能の整理を行った。

- 1). 異常時の診断
- 2). オペレータの誤動作に対する指示
- 3). 操作手順の問い合わせに対する指示
- 4). 通常的な監視

このような機能を提供するために、知識の分割を検討した。まず、このような別個の機能を実現するためには、緊急度に応じた推論の切り替えが必要であると考えた。そして、プロセスデータを保持するauk、プロセスの状態を認識するauk、個々の機能を実現するaukに分割した。とりあえず、これらのaukのインタフェースを規定し、いくつかのaukの詳細化を行っていった。プロセスデータを保持するaukは、フレームaukSetを時系列データ用に拡張して使い、プロセス状態を認識するaukは、オペレータの動作をモデルにし、ルールの前向き推論aukSetで実現した。これらのaukは、いくつかのaukから呼び出されるものである。そして、推論切り替えaukにおいては、まず、異常診断用aukのみに作用するようなルールを用意し、部分的にデバッグを行った。そして、異常診断用aukについて、さらに、部分知識への分割を行い、それらの詳細化を行っていった。

このように、top-down-planningとbottom-up-constructionを交互に組合せ、知識の構築を行うことにより、効率的に作業をすすめることができた。

5. おわりに

知的システム構築用シェルにおける知識構築方法としてのtop-down-planning / bottom-up-constructionの手法の構想を説明した。top-down-planningは、システムのプロトタイプリングによって、システムの概観を知る方法として有効である。bottom-up-constructionは、知識の詳細化、デバッグにおいて有効である。

参考文献

- [1] 藤村, "オブジェクト指向知識表現を用いた知的システム構築用シェル"
情報処理学会第37回全国大会 6G-2 1988
- [2] 飯間, "知的システム構築用シェルAUK(2)
—知識構築環境—"
情報処理学会第38回全国大会 1989