

6G-2 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

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

現在、エキスパート・システム構築用シェルとして、第二世代シェルが多く用いられている。この第二世代シェルは、いくつかの知識表現、及び、手続き記述のための汎用言語とが統合されたハイブリッド型のシェルである。これらのシェルは、グラフィックス表示によるユーザインタフェースを有するのも特徴の一つである。しかし、これらの知識処理、グラフィックス処理と、汎用言語との記述性のギャップは大きい。そのため、習熟したシステム開発者にとっても、知識表現の拡張、あるいは、柔軟なグラフィックス操作を行うことは難しい。

そこで本稿では、これらの問題点を解決すべく、**auk (autonomic knowledge unit:自律的知識単位)** という知識の単位を用いたオブジェクト指向知識表現を提案する。更に、現在、この知識表現を用いた知的システム構築用シェルのプロトタイプを作成中であるので、その概要を示すことにする。

2. auk によるオブジェクト指向知識表現

2.1. 設計目標

- 1). グラフィックス処理と知識処理が、容易に相互に呼び出せるインタフェースを有すること。
- 2). 知識のモジュール性が良く、かつ、統合性の良いこと。
- 3). 基本的に、単一パラダイムであること。
- 4). 表現の拡張が容易なこと。
- 5). メタ知識も表現可能なこと。
- 6). システム開発者のレベルに応じた知識構築環境が用意できること。

2.2. 概要

この知識表現は、知識とは、自分自身を扱う推論方法を知っているオブジェクトであると考え、推論とは、その知識を表現するオブジェクトからの情報抽出であると考えられるものである。このように、静的な知識と動的な知識を兼ね備え、自律的な知識として見ることができるオブジェクトを、**auk** と呼ぶ。また、本知識表現は、グラフィックス機能を有するオブジェクト指向言語上でのインプリメントであることを前提としている。

推論は、いくつかの**auk** がメッセージパスを行うことにより、協調的に実行される。それらの**auk** は、所望の推論系に対応するように、外部インタフェースの基本仕様のもとで拡張され生成される。この拡張は、オブジェクト指向言語上であることから、容易に行える。また、この様に**auk** は、統一した外部インタフェースのもとで実現されるため、モジュール性のよい知識の構築、有効利用が可能となる。更に、推論中、メッセージパスにより、通常のオブジェクト、グラフィックスオブジェクトとの通信が可能である。

2.3. 基本要素

- 1). **aukAtom** auk の基本単位。
- 2). **aukSet** 要素として aukを持つ aukの集合。

2.4. 外部インタフェース基本仕様

- 1). **aukAtom**
メッセージパスにより、情報抽出を行う。
- 2). **aukSet**
以下のメッセージパスで、情報抽出を行う。
add: anAuk 要素にanAuk を追加。
remove: anAuk 要素からanAuk を削除。
get: anAuk レシーバから得られるanAuk とマッチング可能なauk を返す。
redo get:の後に用いられ、他のマッチング可能であるauk を返す。
(××:は1引数を伴うメッセージパス。
anAuk は、あるauk を意味する。)

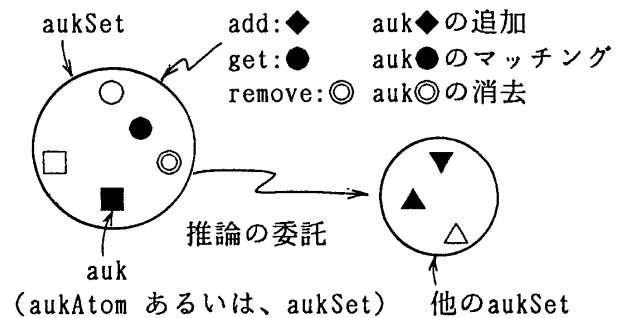


図1 aukSetの外部インタフェース

この基本仕様のもとで種々の推論系に対応する auk を生成する。

TM:Smalltalk-80はXEROX社の商標です。
Shell for building Intelligent System
based on the Object-Oriented Knowledge Representation
Shigeru Fujimura, Shoji Tomita, Noboru Ima, Akira Suzuki
Yokogawa Electric Corporation

3. 知的システム構築用シェル

本知的システム構築用シェルは、人間の知的な作業を、部分的にサポートするシステムを構築するためのツールである。本シェルは、現在、プロトタイプをSmalltalk-80上に作成中であるので、その概要について以下に示す。

3.1. 構成

クラスとしては、auk の基本メカニズムをサポートする基本クラス、および、基本的なルール、フレーム表現をサポートする拡張クラスを用意した。更に、cf値付ルールへの拡張、ルールの探索方法の変更等も適時可能である。

1). 基本クラス

AukAtom

このクラスのインスタンスは、事実、仮説等の表現に用いられる。このオブジェクトは、複数の属性・属性値を有し、各属性には、デーモンの記述も可能である。

AukSet

基本的な外部インタフェースをもつ aukSet のクラス。(全ての aukset のスーパークラス)

add: anAuk

self に、anAuk を要素として単に追加。

remove: anAuk

self から、anAuk を単に削除。

get: anAuk

self の要素で、anAuk にマッチング可能な要素を単に返す。

redu

他のマッチング可能な要素を単に返す。

2). 拡張クラス

ForwardAukSet

ルール表現の前向き推論を行う aukSet のクラスで、AukSet クラスのサブクラスとして実現。aukset の要素が、ワーキングメモリの要素に対応している。Ops5 の動作に類似し、Rete ネットワークを用いて推論を行う。aukSet への aukAtom の追加、削除により推論が起動されるものとし、競合集合が空になるか、その aukSet に halt メッセージが送られると終了する。auk の追加、削除の制約条件は、ルール中にも、もちろん記述可能であるが、その aukSet の属性としても記述可能である。

BackwardAukSet

ルール表現の後向き推論を行う aukSet のクラスで、AukSet クラスのサブクラスとして実現。Prolog の基本動作に類似。すなわち、aukSet 内の auk 要素が Prolog におけるユニット節、ルールが本体をもつ節として考えられる。また、add: が assert、remove: が retract、

get: redo がプログラムの実行と考えられる。そこで、aukSet への aukAtom のマッチングにおいて、マッチング可能な auk を集合の要素として含んでいない場合に、推論が起動され、マッチングされたオブジェクトを返す。redo により逐次推論を続行することができる。

FrameAukSet

フレームの階層構造の機能を取り入れた aukSet のクラス。

3.2. 知識構築環境

図 2 に、知識構築環境の例を示す。各ビューは、aukSet、aukAtom 等を表わしている。これらのビューを用いて、知識ベースのエディット、ファイルへの入出力が行える。更に、デバッグ環境としても用いられ、aukSet 間のメッセージのやりとり(推論)の部分的なデバッグ、トレースが行える。そのため、Smalltalk-80 のプログラミング環境のように、エディット・デバッグ・実行のフェーズを意識せず、開発が効率良く行える。尚、この環境は、より高レベルのシステム開発者用であるが、レベルに応じてカスタマイズも可能である。

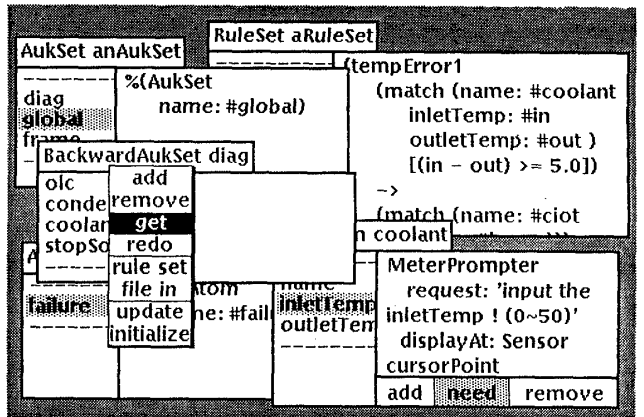


図 2 知識構築環境

4. おわりに

本稿では、オブジェクト指向の概念をとりいれた知識表現を提案し、その知識表現を用いた知的システム構築用シェルの概要を述べた。今後は、この知識表現のもとでの、ユーザのレベルに応じた知識構築環境について検討していく予定である。

参考文献

[1] A. Goldberg, "Smalltalk-80: The Interactive Programming Environment" Addison-Wesley 1984
 [2] C.L. Forgy, "Rete: A Fast Algorithm for the Many Pattern / Many Object Pattern Match Problem" Artificial Intelligence 1982
 [3] 上谷, "統合化プログラミング環境 Smalltalk-80とInterlisp-D" 丸善 1987