

## 4 G-1

## オブジェクト指向とルール型言語の融合について

(株)東芝 システム・ソフトウェア技術研究所  
稲葉 浩人

## 0.はじめに

これまでルール型言語として、プロダクションシステム(PS)がエキスパートシステム構築の分野を中心として使われてきた。しかし、ルール型言語によって大規模なシステムを開発するには以下のような難点がある。

- ・システムが平面的である
- ・単純なPSの認知行動サイクルは様々な推論での複雑な処理をするためには記述力が足りない。

これらの問題を解決する為、われわれはルール型とオブジェクト指向との融合によりモジュール性、表現力の強化した言語を設計開発することとした。特に条件部をオブジェクト化することにより、認知行動サイクルをより柔軟なものにすることを目標として掲げた。ここではその仕様の概要を示す。

## 1. オブジェクト指向化の現状

従来のルール型言語において、条件部のオブジェクト化と呼べるものは、以下の2つがあると考えられる。

- ・RETEアルゴリズム[1]で各ノードをオブジェクト、トークンをメッセージとすることにより条件照合を行う
- ・オブジェクトがルールセットをもち、そのオブジェクトでの推論を行うモードをつける。

しかし、前者はユーザによる利用は困難なものであるし、後者はモードを意識する必要があり、また多オブジェクト間のルール記述ができないという難点がある。

## 2. 条件パターンクラス

上記のような難点を避ける為、ルール条件部におけるオブジェクト化の方法として言語に条件パターンを導入し、そのクラス化を行なうこととする。

条件パターンとは、ルールにおけるいくつかの条件要素をセットにしてまとめたものであり、ユーザが自由に定義できるものである[2]。定義された条件パターンはルールの条件部や条件パターンの一部として使用することができる。

以下に条件パターンの例を示す。

```
(defpattern foo () :slots (c1 c2 x y z)
  c1 (class1 x <= 10 z)
  c2 (class2 y == x))
(defpattern bar () :slots (f1 c3 z)
  f1 (foo x <= 10 z)
  c3 (class3 y == x))
```

この条件パターンの定義によって一種のクラスオブジェクトである条件パターンクラスが生成され、このクラスとそのインスタンスによってルールの条件照合を行う。全てのルールは条件パターンとそのクラスのfireメソッドとの組として実現される。すなわちルールの条件部を一つのパターンとし、実行部はそのパターンに対するメソッド定義となる。

ルールの条件照合は条件パターンクラス間のメッセージ通信によって行われる。すなわち条件パターンクラスに対するメッセージ

**Add-token** : 条件パターンにトークンを追加する。

**Del-token** : 条件パターンからトークンを削除する。  
に対応するメソッドが定義されており、これらを用いて条件照合が行われる。

以下にこのシステム定義のメッセージを用いてclass1がワーキングメモリに加えられた時の照合の際の動作を述べる。まずワーキングメモリにclass1のインスタンスが加わる。するとclass1を含む条件パターンクラスにadd-tokenメッセージがわたり、条件パターンクラスは他の条件要素とのテストを行う。もし成功すれば自分のインスタンスを生成する。その後、その条件パターンをさらに含む条件パターンクラスにadd-tokenメッセージを送る。

## 3. 条件パターンインスタンス

条件パターンクラスのインスタンスを条件パターンインスタンスと呼ぶ。これはRETEアルゴリズムでの $\beta$ メモリおよびルールインスタンスの働きをする。

この条件パターンインスタンスには以下のようなシステム定義のメッセージが定義されており、条件照合時に自動的に各インスタンスに送られる。

**Initialize**: インスタンスが生成された時に送られる。

**Die**: インスタンスが消去される時に送られる。

**Fire**: インスタンスが発火する時に送られる。

これらをoverrideするメソッドを与えることにより、ユーザは条件照合時に自動的に実行される操作を指定しカスタマイズを行うことができる。もちろん、ユーザはこれ以外のメソッドを定義して、それを使用することもできる。

条件パターンインスタンスのインヘリタンスの方法を以下に述べる。まず、スロットの継承はパターン中に現れる条件要素から継承する。ただしこの継承は明示的に指定したものだけについて行なう。次にメソッドの継承はパターンの定義中で指定したクラスを継承する。ただし、与えられたメッセージに対応するメソッドがない時、(1)継承したクラスに対応するメソッドがあれば、それを実行し、(2)なければ、各条件要素にメッセージを送る。という操作を行う。

## 4. おわりに

ルール条件部を条件パターンを用いてオブジェクト化した。

これにより、認知行動サイクルの条件部で、ある条件が満たされた時行われる行動をユーザが指定できるようになる。また、ワーキングメモリ要素をまとめてオブジェクトとして操作することが可能になる。

## 参考文献

[1]Forgy, C. : RETE : A First Algorithm for Many Pattern/Many Object Pattern Match Problem, Artificial Intelligence (19) pp17-37 (1982)

[2] 松木 他 : エキスパートシステム構築支援ツールKDL, 情報処理学会第37回全国大会 (1988)

A Proposal of Object-Oriented Rule-Based Language

Inaba Hiroto

TOSHIBA Corporation