4M-2

Ｐｒｏｌｏｇ言語を用いた知識プログラミングシステムＰｅａｃｅ

若杉 暢庸* 古関 義幸**
* 日本電気技術情報システム開発株式会社
**日本電気株式会社 C&Cシステム研究所

1. はじめに

最近の知識プログラミングシステムは、オブジェクト指向との融合をはかる等で、複数の知識表現方式を持つハイブリッドシステムが多い。しかし、現実の世界に対応した素直な知識表現を行なうには、まだ十分ではない。たとえばオブジェクト指向言語の問題点としては、システムがオブジェクトのタイプをスーパークラス、クラス、インスタンスのように限定していること、またそれに伴いスーパー・サブ、クラス・インスタンスの関係付けが一意に決定されてしまうことが挙げられる。こうした中で、我々は電子交換機の故障診断[1]を行なうため、知識プログラミングシステムPeace(Prolog based Expert AppriCations Environment)を開発した。Peaceはつぎのような特徴を持つ。

・オブジェクトのタイプを限定しない。
・オブジェクト間に自由な関係を張ることができ、またその関係を使用したマルチプルインヘリタンスが可能。
・Prolog言語の持つバックトラッキング、パターンマッチング等を推論に取り入れることが可能。

本稿では特に「関係」を用いたインヘリタンスについて述べる。またProlog言語との親和性について例示する。

2. システム構成

図１にPeaceのシステム構成を示す。

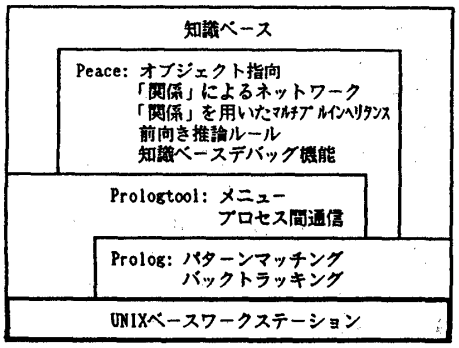

図1. Peace構成

知識ベースは、オブジェクトの記述、ネットワークを構成するためのオブジェクト間の関係付け、前向き推論ルール、およびPrologプログラムとを組み合わせることによって構成される。

オブジェクトは、フレーム表現でのスロット、スロット付加手続き（デモン）、Prologの節を記述したメソッド、他オブジェクトへの関係付け、およびインヘリタンスの仕様定義からなる。

マルチプルインヘリタンスはオブジェクト間に張られる「関係」を通して行なわれる。また、関係付けそのものをパターンマッチングにより参照することが可能である。

前向き推論はオブジェクトとして登録した、ルールブロックにより行なわれる。ルールインタプリタは、do 1, do all, while 1, while allの４種類の制御戦略を持ち、ブロック毎にそれを指定する。

知識ベースのデバッグには、オブジェクトに対するメッセージ（操作）をトレースするメッセージトレーサや、ネットワークを解析し、図示するツール等が使用できる。

本システムは、demo述語[2]を用いてPrologの自然な拡張として実現されており、UNIXベースのワークステーションにおいて、Prologプログラミング環境Prologtool[3]で動作する。なお、記述は上述のツールを含め4K行である。

3. 関係付けとインヘリタンス

Peaceはオブジェクト間の関係付けに制約がなく、ユーザが自由に関係を定義できることから、オブジェクトの意味付け（タイプ設定）、およびインヘリタンスの仕様もユーザにまかされる。

他オブジェクトに対し、関係付けを行なうには、「関係」をオブジェクトとして定義する。これを関係オブジェクトとよぶ。関係オブジェクトには「関係」に対する「逆関係」、およびインヘリタンス仕様を記述することができる。

逆関係とは、たとえばクラス・インスタンス関係におけるinstance_ofとhas_instanceのように、お互いに逆の関係を指すものである。逆関係を定義することにより関係付けの自動更新、冗長性の排除を可能にする。

インヘリタンス仕様は、スロット値、スロット付加手続き、「関係」、Prologの節の４種類についてそれぞれ独立して定義できる。マルチプルインヘリタンスを利用するには、関係付けを複数のオブジェクトに対し行なえばよい。それにより、バックトラ

Prolog Based Knowledge Programming System :Peace
Nobuyasu Wakasugi* Yoshiyuki Koseki**
* NEC Scientific Information System development, Ltd.
** C&C Systems Research Laboratories NEC Corp.

ックのたびに関係付けが参照され、その登録順にインヘリタンスが行なわれる。

4. プログラミング例

図２は、「ｾﾌﾞﾝｽﾀｰはタバコの一種であり、ﾏｲﾙﾄﾞｾﾌﾞﾝとは兄弟関係にある。それらの販売元はＡ株式会社である」、またＡ株式会社の側からみて、「Ａ株式会社の所属は日本であり、またﾏｲﾙﾄﾞｾﾌﾞﾝとｾﾌﾞﾝｽﾀｰの２つの商品を保有している」を表している。これらをPeaceで記述したのが図３である。

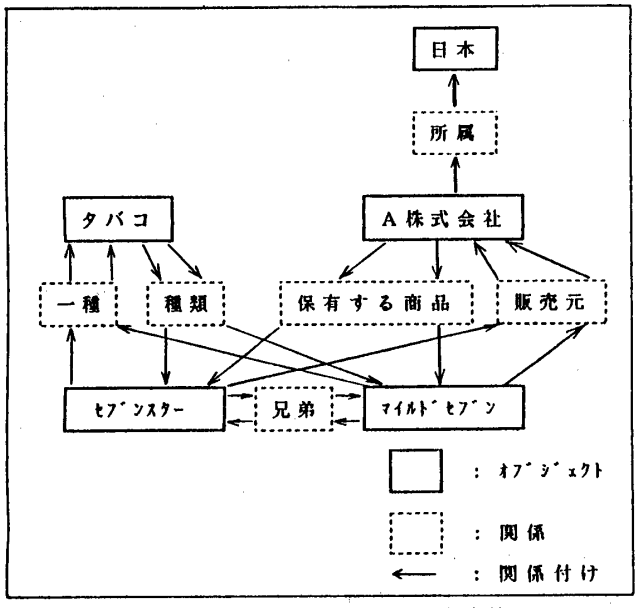

図２．オブジェクトとその関係付け

```
/*  関係の定義  */
%   :- 種類 has
%          $inverse(一種).
    :- 一種 has
          $inherit_slot_value(_);
          $inherit_slot_type(_);
          $inherit_pred(_);
          $inverse(種類).
    :- 所属 has
          $nil.
%   :- 保有する商品 has
%          $inverse(販売元).
    :- 販売元 has
          $inherit_relation(所属);
          $inverse(保有する商品).
    :- 兄弟 has
          $inverse(兄弟).
/*  オブジェクトの定義  */
    :- 日本 has
          $nil.
    :- Ａ株式会社 has
          所属 # 日本;
          保有する商品 # ｾﾌﾞﾝｽﾀｰ;
          保有する商品 # ﾏｲﾙﾄﾞｾﾌﾞﾝ.
    :- タバコ has
          種類 # ｾﾌﾞﾝｽﾀｰ;
          種類 # ﾏｲﾙﾄﾞｾﾌﾞﾝ.
    :- ｾﾌﾞﾝｽﾀｰ has
%         一種    # タバコ;
%         販売元 # Ａ株式会社;
          兄弟    # ﾏｲﾙﾄﾞｾﾌﾞﾝ;
          商品名 : 'Seven Stars'.
    :- ﾏｲﾙﾄﾞｾﾌﾞﾝ has
%         一種    # タバコ;
%         販売元 # Ａ株式会社;
%         兄弟    # ｾﾌﾞﾝｽﾀｰ;
          商品名 : 'MILD SEVEN'.
```

図３．知識ベース記述例

ここで、'#'は関係付け、':'はスロット値の定義に関するオペレータである。また、前述のとおり関係付けの更新を自動的に行なうので、%が先頭に付加されている行は省略してもよい。

上記の知識ベースからＡ株式会社のタバコの名称をすべて求めるには、

```
?- Ａ株式会社 <- 保有する商品 # OBJECT,
   OBJECT <- (一種 # タバコ, 商品名 : NAME).
```

に対しバックトラックを起こすことにより、タバコの名称が順に'Seven Stars'、'MILD SEVEN'のようにNAMEにユニファイされる。また、ﾏｲﾙﾄﾞｾﾌﾞﾝが日本製であるのを知るには、

```
?- ﾏｲﾙﾄﾞｾﾌﾞﾝ <- 所属 # 日本.
```

が「真」になるかどうかを調べればよい。「所属」は、ﾏｲﾙﾄﾞｾﾌﾞﾝ自身には記述されていないが、「販売元」に$inherit_relation(所属)という、所属をインヘリットする定義があるので、ﾏｲﾙﾄﾞｾﾌﾞﾝにおいての参照が可能になる。

以上の例で示したようにPeaceは、Prolog言語と親和性が高いことから、バックトラック、パターンマッチの機能を有効に活用することができる。

5. おわりに

Prolog言語を用いて開発した知識プログラミングシステムの「関係」記述とそれを使用したインヘリタンス、および同言語との親和性について示した。

本プログラミングシステムを用いて、交換機故障診断システムSHOOTX[4]を試作している。同知識ベース構築において、「関係」とそれによるインヘリタンスは非常に有効である。

なお、本研究は第５世代コンピュータプロジェクトの一環として行なっている。

【参考文献】
[1]古関他 知識表現/推論ｼｽﾃﾑPeaceとその故障診断問題への適用, 情処32回全国大会
[2]國藤他 論理型言語Prologによる知識ﾍﾞｰｽの管理, Logic Prog. Conf. '85
[3]近藤他 EWS上のPrologﾌﾟﾛｸﾞﾗﾐﾝｸﾞ環境Prologtool, 情処33回全国大会
[4]古関他 ﾏﾙﾁﾌﾟﾙ知識利用故障診断ｴｷｽﾊﾟｰﾄｼｽﾃﾑSHOOTX, 情処33回全国大会