

知識表現言語XPT/KR

6M-3

藤本 久志 下地 俊一 田口 研治 佐久間 良 塩沢 恵子
 (株)CSK総合研究所

1 はじめに

我々は統合的な知識工学システム開発環境、主にエキスパートシステム構築ツールとしてのXPT (eXpert system Programming Tool)の開発を行っている。今回その改良版としてXPT2を開発した。その知識表現用言語としてPrologにオブジェクト指向を導入したXPT/KRを開発したので報告する。またその他のインターフェイス、黑板モデルの改良点についても言及する。

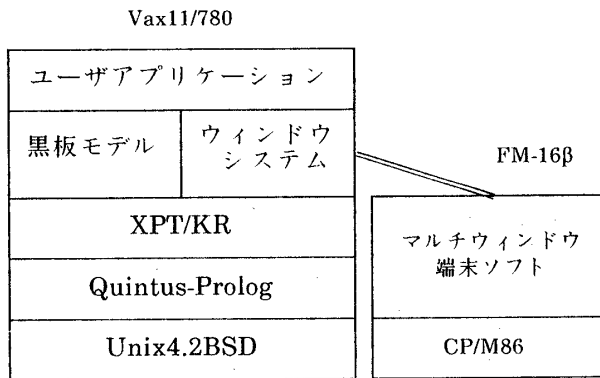


図1 XPT2のシステム構成図

2 知識表現言語XPT/KR

2.1 概要

XPTでは知識表現部分はXPTOと呼ばれていたが、クラス定義のファイルを読み込むという方式を採っていた。しかしユーザの開発効率を向上させるため、対話的に作成する方式に変更した。XPT/KRでは論理型言語Prologの上にオブジェクト指向を取り入れ、そのオブジェクトの中にメソッドとしてPrologのプログラムを記述できるなどいくつかのパラダイムを統合している。

2.2 知識ベース

XPT/KRでは各問題解決分野ごとの知識の集まりを知識ベースとして扱う。その知識ベースの基本構成要素をオブジェクトと呼ぶ。またXPT/KRのシステム自身もいくつかの知識ベースで構成されている。変数に入れることのできるデータのタイプ、変数のインヘリットの仕方を規定したロールそしてオブジェクト間の関係に関する知識ベースなどがあり、ユーザの使用目的に合わせて変更が可能である。また現在システムに登録されている知識ベース名も知識ベースで管理されている。

2.3 オブジェクト

オブジェクトには他のオブジェクトの型紙になることのできるノード(node)とその実体であるリーフ(leaf)の2種類がある。例えば動物に関する知識ベースでは犬、猫などがノードであり、ポチ、ミケなどがリーフである。

オブジェクトはその属性を表すために変数とメソッドを持っている。変数はオブジェクトの静的知識を表すために用いられ、メソッドは動的知識を表すのに用いられる。例えば犬には色、足の数などの変数があり、メソッドとしては動き方などを記述したものがあ

2.4 オブジェクト間の関係

オブジェクト間にはいくつかの関係が存在する。関係には、集合で表現するとある集合とその部分集合であるスーパー/サブ関係と、ある集合とその集合の要素であるクラス/インスタンス関係がある。またその他に全体と部分のホウル/パート関係がある。図2に例を示す。

スーパー/サブ関係はノード間だけに存在し、ホウル/パート関係はノード間、リーフ間に成り立つ。クラス/インスタンス関係はノードとリーフの間だけでなく、ノード間でも可能である。

ノード間でホウル/パート関係を設定している場合、ホウル部にあたるノードのインスタンスを作成するとパート部のインスタンスも作成し、リーフ同士にもホウル/パート関係を設定する。

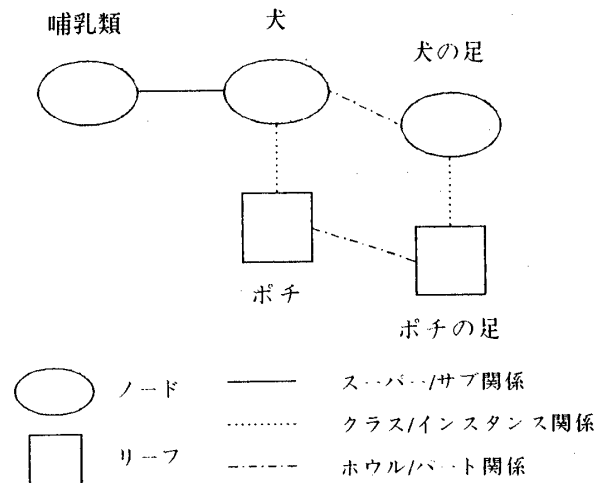


図2 動物の知識ベースの例

Knowledge Representation Language XPT/KR

Hisashi FUJIMOTO Shun'ichi SHIMOJI Kenji TAGUCHI Ryou SAKUMA Keiko SHIOZAWA
 CSK Research Institute

2.5 変数

オブジェクトに関する静的知識を表現するには変数を用いる。変数にはそのオブジェクトのインスタンスの属性を表現するインスタンス変数とそのオブジェクト固有の属性を記述するローカル変数の2種類がある。インスタンス変数名はスーパー/サブ関係間でインヘリットし、クラス/インスタンス間でローカル変数に変化するが、ローカル変数名はどの関係においてもインヘリットしない。オブジェクトの内ノードはインスタンス変数、ローカル変数を持つことができるが、リーフはローカル変数しか保持できない。

変数にはファセットとしていくつかのものがある。実際の変数の値が入れられる value ファセット、変数値の指定が無かった場合に用いられるデフォルト値を入れる default ファセット、入れることのできる値のタイプを規定する datatype ファセットそして値のインヘリットの仕方を決定する role ファセットなどがある。

変数の値が入れられたり、参照されたときに起動されるデモン機能を用いると、温度計の値をモニタしてある一定の値以上になると警告を出すというような柔軟なプログラミングが可能である。

2.6 メソッド

オブジェクトに関する動的知識を表現するにはメソッドを用いる。メソッドにはそのオブジェクトのインスタンスの動きを表現するインスタンスメソッドとそのオブジェクト自身の動きを記述するローカルメソッドの2種類がある。メソッドのインヘリタンスは変数のそれと同じである。

疑似変数としてはXPTでは self, here, supers を用いていたがXPT2では Selfのみとした。メソッドは実際に Prolog の述語として定義され、コンパイルも可能である。

3 黒板モデル

XPTではAGE風の黒板モデルを実現したが、XPT2の黒板モデルは推論の過程を複数の専門家が共通の黒板を見ているという会議をモデルとしている。議長が議題を参照しながら次の発言者と呼ばれるルール群を起動していく。

推論は前向きにも後ろ向きにも可能であるが、前向き推論のルールは Rete ネットワークにコンパイルし高速化を図っている。

黒板上はセクションと呼ばれるいくつかの区画に分けられている。セクション上にはオブジェクトおよび Prolog の節を書くことができる。

4 インターフェイス

インターフェイスについてはマルチウィンドウ、ポップアップメニューそしてマウスを用いてエキスパートシステムを開発することには変わりはない¹⁴⁾。しかしXPT2では対話的に作成することに重点を置いて開発した。その他日本語とグラフィックなどが使用可能である。

XPTに比べて拡張された機能としてはマウスセンシティブティのあるリージョン、スクロールバーそしてXPT2の実行中 Prolog のトップレベルで述語を実行できることなどである。

このインターフェイスはプログラマブルマルチウィンドウターミナル¹⁵⁾を用いて開発された。

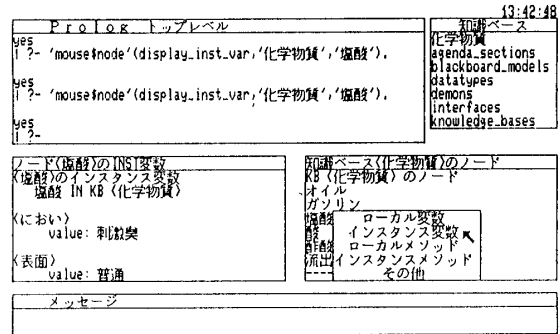


図3 XPT2の画面例

5 おわりに

今回は主にスピードアップ、ユーザの使い易さの面から改良を図ったXPT2の知識表現を中心に黒板モデル、インターフェイスについて述べた。

今後ユーザからのフィードバックを反映しながら、関係データベースとのリンク、ファジィ論理の導入など新しい機能、概念を取り入れさらに改良を行う予定である。

```
?- def_kb('化学物質').
?- def_obj('化学物質' $ '流出物').
?- def_inst_vars('化学物質' $ '流出物',
                ['溶解度','表面','におい']).
?- make_subs('化学物質' $ '流出物','酸','オイル').
?- putz('酸','溶解度','高い').
?- putz('酸','表面','普通').
?- putz('オイル','溶解度','低い').
?- putz('オイル','表面','銀色の膜').
?- make_subs('化学物質' $ '酸',['塩酸','酢酸']).
?- putz('塩酸','におい','刺激臭').
?- putz('酢酸','におい','すっぱい').
?- make_subs('化学物質' $ 'オイル',['ガソリン','軽油']).

?- get('塩酸','溶解度',X).
X = 高い
?- get('ガソリン','表面',Y).
Y = 銀色の膜
```

図4 実行例

参考文献

- [1]佐久間 他:“エキスパート・システム開発ツールXPT(1)-基本設計-”,情報処学会第32回全国大会,6L-1,1986
- [2]下地 他:“エキスパート・システム開発ツールXPT(2)-知識表現-”,情報処学会第32回全国大会,6L-2,1986
- [3]藤本 他:“エキスパート・システム開発ツールXPT(3)-推論方式-”,情報処学会第32回全国大会,6L-3,1986
- [4]藤本 他:“エキスパート・システム開発ツールXPTにおけるユーザインターフェイス”,電通学会全国大会,p6-223,1986
- [5]岡部:“プログラマブルマルチウィンドウターミナル”,情報処学会第33回全国大会,1M-5,1986