

4 C - 8

対話による計算機利用支援

(2) CLOSによる知識の表現

樋口 英幸 伊藤 昭
郵政省通信総合研究所

1 はじめに

我々は、計算機ユーザの支援を行なう対話型のシステムの設計/構築を行なっている。この中で、知識表現は推論システムが扱うデータの表現形式として重要な位置を占め、理論/実装ともに色々な研究が行なわれてきている。ここでは、知識の階層化が容易と思われるオブジェクト指向言語の一つ CLOS[1] による知識表現を報告する。これは、比較的歴史の浅い CLOS に対する評価などの目的を同時に含んでいる。

2 CLOSによる知識表現の得失

概念の階層構造には一般に ISA 関係 (抽象/具体関係) と PART-OF 関係 (全体/部分関係) があり、前者の階層関係を中心に階層化することが多く、本システムでもこのアプローチをとっている。PART-OF 関係は、属性として表され、やはり継承関係を持ち、スロットに対するアクセス時にはデーモンを起動できる。これらの機能は他の表現、例えばフレーム表現と CLOS であまり差がない。フレームはクラス/インスタンスの区別を持たないので、そのオブジェクトの意味するものの多義性の度合いが高いこと、CLOS ではスロットの継承がフレームほど強力でないこと、処理レベルとして、型チェックなどの機能が甘いことなどが、それぞれの特性を示している。

さて、ここで CLOS について、もう少し細かくその特性を考えてみる。CLOS は、その母体である Common-Lisp の思想とは大きく異なり、仕様として提示されたメソッド/関数/マクロの数は非常に少ない。しかし、メタクラスと前述したデーモン系のメソッドの機能があるため、仕様から受けるイメージよりは、はるかに強力である。しかし、知識表現に必要な操作を実現するためには、スロット関係の情報を得るメソッド、例えば、その概念クラスのもつスロット名などを得る必要があるが、CLOS ではそのインターフェースが提供されていない。

これに関しては、PCL では特定の内部メソッド、関数で機能の実現は可能であり、それより初期の段階では [3] による、非常に強力なスロット操作のプログラムも検討したが、ここではポータビリティを重視し、スロット名のみをクラスオブジェクト生成時にクラスオブジェクトのスロットに格納することにした。それ以外の問題は、メタクラスのレベルで解決することができた。

3 知識表現の CLOS へのマッピング

ここでの対象となる世界は、ユーザと計算機が相互作用を与えるような対話であり、その状態を外部から観察可能なオブジェクトに写像した世界をオブジェクト世界と呼んでいる。オブジェクト世界内のオブジェクトは実在し、その状態を属性名と属性値で表すことができるようなものである。実在のオブジェクトに対し、そのオブジェクト生成のための雛型に当たるものを「概念」と呼ぶ。概念は継承関係を持ち、その上下関係が具体/抽象関係を表す。これは、それぞれ自然に CLOS のインスタンスとクラスオブジェクトに対応づけることができる。また、オブジェクトワールドは時間の概念を持つ。時間の概念をワールドを観る視点側が持つか、個々のオブジェクトが持つかという2つのアプローチが考えられるが、ここでは後者の立場をとっている。

ここで、以上の考え方に基づくオブジェクトワールドの実現の一例として、UNIX のファイルシステムを対象ドメインとした概念の構造を図1に示す。

4 CLOSによる実装

まず、図1に示した概念オブジェクトは、そのメタクラスが m-class という一つの同じクラスになっている。各概念オブジェクトに独自のメソッドを与えたい時は、概念の階層構造と双対な階層構造でメタクラスを構成すれば良いが、ここでは概念オブジェクトに対する操作は統一することとし、メタクラスは共通というアプローチをとった。

ここで、インターフェースとして提供したメソッドの説明と共に付加した機能を説明する。

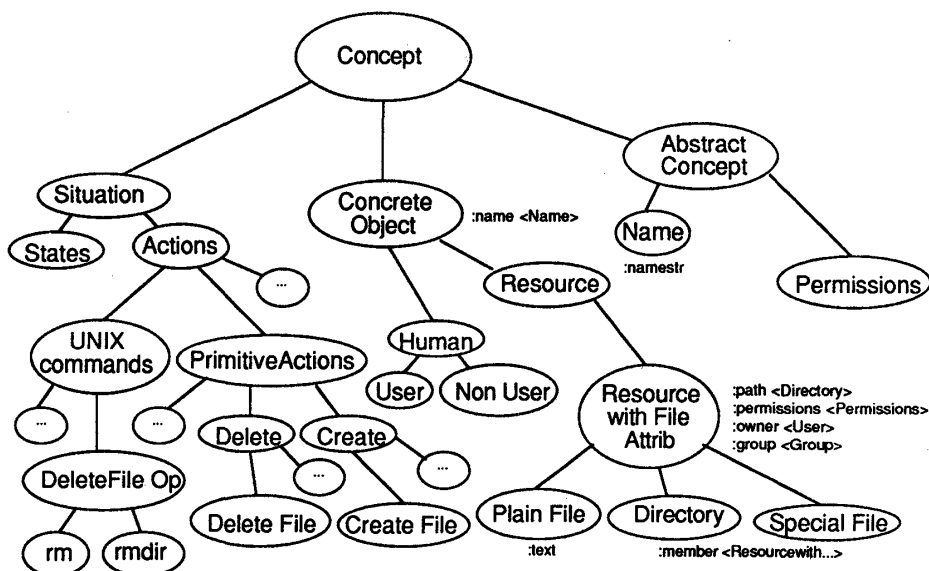


図1 UNIXファイルシステムにおける概念の階層

- (DefConcept conceptname ...) {Macro}

defclass へのマクロであるが、この定義によってできるクラスオブジェクトのスロットに、そのクラスにおけるスロット定義を格納する。スロット定義は、CLOSのセマンティックスを基本としている。主な変更は、次の2点である。1) 型指定では、PCLと異なり、実際に代入時にタイプチェックを行なうこと。2) CLOSのオブジェクトとしては各スロットは、時間と属性値のドットペアのリストで表現されるが、タイプ指定はその属性値のタイプを書けばよい。また、ここでいうタイプ指定はCLOS, CommonLispでいうそれより広範囲を扱うことができる。例えば、member スロットの値が Resource のインスタンスのリストであるという型指定をする記述が可能で、その場合の記述は以下のように行なう。

```
(member :type (* Resource))
```

- (d-spr concept)
- (d-scs concept)
- (allslots concept)
- (inst concept) (いずれも {Method})

d-spr(direct-superclasses) は概念オブジェクトの直接の上位概念オブジェクトを与えるメソッドであり、逆に d-scs(direct-subclasses) は直接の下位概念オブジェクトを与えるメソッドである。allslots はその概念オブジェクトが持つスロット名のリストを与える。inst(instances) は、その概念オブジェクトのインスタンスを与える。これらの情報は、概念オブジェクト上のスロットに格納されている。

- (writer concept slotname value time)

- (reader concept slotname time)

(いずれも {Method})

オブジェクトワールドのインスタンスの持つ属性値の読み出しと書き換えは、すべてこの2つのメソッドを用いて行なわれる。例として、1つのCLOSとしてのスロットは、次のようになっている。

```
((t2 . t) (t1 . nil) (0 . t))
```

時間は正整数である。上の表現は、時刻 [0,t1] でスロット値が t, [t1,t2] で nil, [t2, 現時刻] で t であることを示す。writer においては、代入時に上の意味での型チェックが行なわれる。

5 まとめ

対話型支援システムにおける知識表現に必要な機能をCLOS上で実現した。現在、これを利用してUNIXファイルシステムとXmhドメインの支援システムを設計/構築中である。

参考文献

- [1] D. G. Bobrow, L. G. DeMichiel, R. P. Gabriel, S. E. Keene, G. Kiczales and D. A. Moon(1988): *X3J13 standards committee documents 88-002R*.
- [2] 上野, 石塚 (1987): "知識の表現と利用", オーム社.
- [3] 塚田, 杉村 (1990): *ダイナミックオブジェクトの提唱*, p.6-72, '90 信学会春全大.
- [4] 樋口, 伊藤 (1990): *対話型UNIX支援システムの機能と構成*, 5S-2, '90 情処秋全大.