

5G-8

エキスパートシステム構築ツールCHORUS (3)
— フレームシステム —村田真人 飛鳥井正道 森沢秀一 浅野俊昭
キャノン(株) 情報システム研究所

1. はじめに

我々は、レンズ設計エキスパートシステムなどの開発経験を基に、エキスパートシステム構築ツールCHORUSの開発を行っている。CHORUSでは、ファクト、フレーム、エージェント[1]を組み合わせることに、柔軟な知識表現が可能である。また、フレームシステムと多重世界機構とが統合化されており、フレームに対しても仮説推論が可能となっている。

本稿では、CHORUSの知識表現の一つであるフレームシステムについてその基本機能と、多重世界機構との統合化手法について報告する。

2. フレームシステムの基本機能

フレームは構造的、階層的な知識を表現するのに適した知識表現であり、概念的、抽象的なものを表現するクラスと、クラスの具体的な実例を表現するインスタンスとに分けられる。以下、簡単にCHORUSのフレームシステムの機能について述べる。

クラスは関数defclassにより、以下のように定義される。

```
(defclass クラス名 (上位クラス名 ...)
  (スロット ファセット 値
   ファセット 値
   ...) ...)
```

クラススロットにはその値に制約を加えるためのファセットや、デーモンを記述することが可能である。ファセットとしては、以下のものが用意されている。

```
:value          スロット値
:un-inherit     値がTの場合、スロット値を非継承
:multiple-values 値がTの場合、スロット値はマルチバリュー
```

デーモンとしてはif-needed、if-added、if-removedの標準的な三つが記述可能である。

インスタンスはmake.instanceメソッドにより、以下のように生成される。

```
(make.instance クラス名 :name インスタンス名
  スロット 値 スロット 値 ...)
```

また、関数defmethodにより、クラスに対して以下のようにメソッドを定義することが可能である。

```
(defmethod (メソッド名 クラス名
  [メソッドタイプ])
  メソッド本体)
```

メソッドタイプの指定により、主メソッドの他にbefore、afterメソッドを定義することができ、メソッドコンビネーションを可能としている。

3. フレームシステムと多重世界機構との統合化

上述したように、フレームは階層的構造を表現するのに適した知識表現であるが、多重世界機構と統合化することにより、知識の幅をより拡げることが可能となる。例えば、小林の言う定形的な設計問題[3]を考えてみた場合、与えられたシステムの構造をフレームを用いて記述し、競合する構成要素を仮説として仮説推論を行い、最も適切な構成要素を選択するといったことが可能となる。しかし、推論時に各ワールドで起こる新たなデータの付加、データの削除といった変更をフレームの階層構造を維持したまま、矛盾なくいかに効率的に管理するかという点で統合化は難しく、ART[4]、KEE[5]といったAIツールの他にはほとんど実現例を見ない。

(1) CHORUSにおける統合化

CHORUSでは統合化を、フレームコンパイルと呼ぶ手法により実現している。フレームコンパイルとは、あるフレームに注目し、そのフレームで上位フレームからの継承を含む参照可能なすべてのスロットを、

```
(フレーム名 スロット名 値)
```

のようにファクトとしてワーキングメモリ上に展開する操作をいう。フレームコンパイルの例を図1に示す。展開されたすべてのファクトには、それがどのワールドで付加され、どのワールドで削除されたかという情報、すなわちラベルが付加され、ラベルによって多重世界が表現される。例えば、図2(a)で、ファクト

```
(FRAME1 SLOT1 VALUE1)
```

のラベル

```
(WORLD1 | WORLD2 WORLD6)
```

は、このファクトがWORLD1で付加され、WORLD2とWORLD6で削除されたデータであることを示しており、よってこのファクトが成り立つのは図2(b)のWORLD1、WORLD3、WORLD7ということになる。

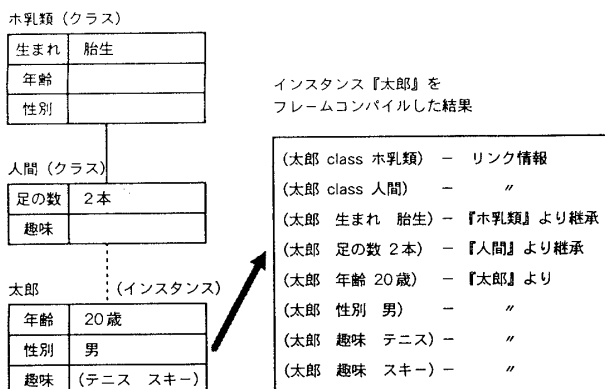
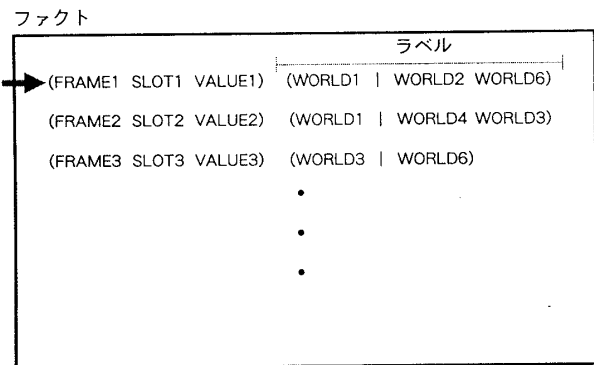
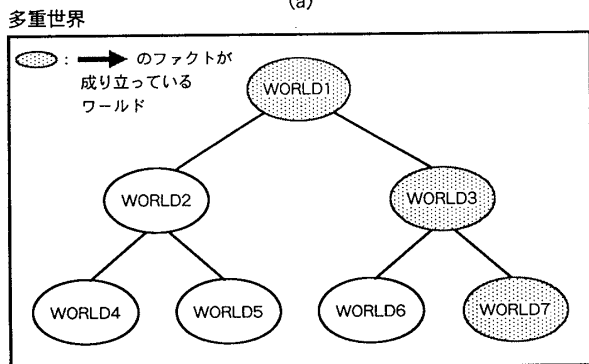


図1 フレームコンパイル例



(a)



(b)

図2 ラベルによる多重世界表現

(2) 展開されるファクトの削減

フレームコンパイルを用いて統合化を行っているAIツールにARTがある。ARTの場合フレームコンパイルは定義したすべてのフレームに対して行なわれる。また、ARTにはフレームを管理するためのシステムフレームが存在し、これらのシステムフレームもすべて同時にコンパイルされるため、展開されるファクトは膨大な数となる。例えば、図1の3つのフレームをコンパイルしただけで展開されるファクトの数は、定義したフレームに関しては約

60個、システムフレームを合せると1000個を越えてしまう。そのためこの方式は、メモリなどの資源が十分に無い環境では有効ではないと思われる。

CHORUSでは、クラスはあくまで概念の集合であって、推論の過程で変更を受けるのは多くの場合、インスタンスが持つ固有の値であることに注目し、フレームコンパイル対象をインスタンスに限定した。また、システムはフレームコンパイル後もクラスを保持しており、ファセット、デーモン、メソッドといった情報は必要に応じてクラスから求めることができるので、フレームを管理するためのファクトを持つ必要がない。これらの理由により、展開されるファクトの数をARTに比べると大幅に減らすことが可能となり（例えば図1で展開されるのは図1の8個のファクトのみ）、2.で述べた機能を維持したままでメモリの節約と共に、推論時のパターンマッチ時間を短縮することができた。

4. おわりに

エキスパートシステム構築ツールCHORUSのフレームシステムについて、その基本機能と多重世界機構との統合化手法について報告した。本フレームシステムの特徴をまとめると以下ようになる。

- (1) デーモン、マルチバリュー、継承の制約といったフレームの基本的な機能をすべて持っている。
- (2) オブジェクト指向の概念を導入して、多重継承や、メソッドの利用が可能である。
- (3) フレームコンパイルによって、多重世界機構と効率的に統合化されている。

問題点としては、コンパイル対象をインスタンスに限定しているために、クラスの値ではパターンマッチができない、すなわちクラスの値はルールのif部に記述できないということがあげられる。また、ワールドをマージして異なるワールドから同一のスロットをそれぞれ継承した場合、それがシングルバリュースロットだった場合には、どちらのワールドから継承した値をそのワールドのスロット値とすべきかといったことも今後の課題である。

【参考文献】

- [1] 飛鳥井ほか：エキスパートシステム構築ツールCHORUS(1)－協調推論機能－、本論文集。
- [2] 飛鳥井ほか：エキスパートシステム構築ツールCHORUS(2)－仮説推論機能－、本論文集。
- [3] 小林：問題解決と典型的タスク、AIセンタ知識情報処理調査委員会知識システム方法論WG資料(1987)。
- [4] Inference Corp., ART Reference Manual(1987)。
- [5] Robert E. Filman: Reasoning with Worlds and Truth Maintenance in a Knowledge-based Programming Environment; Communications of the ACM, Vol. 31-4.