知識工学と人工知能 56－10
（１９８８．１．１３）

オブジェクト指向知識データベース

中野良平　　小濱千恵

（ＮＴＴ情報通信処理研究所）

知識ベースとデータベースの統合という知識情報処理における基本課題の解決の緒として、フレームモデルと関係モデルとの統合を図る知識データベースモデル（フレーム論理２）について述べる。フレーム論理２は、以前提案したフレーム論理をオブジェクト指向による実現の観点から見直したものである。フレーム論理１と比べると、オブジェクト指向の導入、２モデルの共存から単一モデルへの統合、手続きによる知識記述と制御の導入、等が進化した。また、インスタンス遡及、フレーム解釈、グラフと非正規表現、無矛盾管理、等の概念を提案・整理した。

Object-Oriented Knowledge Data Bases

Ryohei NAKANO　　Chie KOHAMA

NTT Communications and Information Processing Laboratories

How to Integrate knowledge bases and databases is one of the most basic problems in the knowledge and information processing field. This paper proposes the knowledge data base model called Frame Calculus 2, which integrates the frame model and the relational model. Frame Calculus 2, a revised version of the formerly presented one, has been refined from the standpoint of the object-oriented paradigm. This model is characterized by several new concepts:the complete integration of the underlying two models,the retroaction of instances through a frame hierarchy,the interpretation of frames, semantic integritychecking and a graph.

1．はじめに

知識処理の基本要素である知識ベースと情報処理の基本技術であるデータベースを如何に統合するかという問題は、人工知能とデータベースの学際領域におけるホットな基本課題である。この課題に対し、現状では、例えばプロダクションシステムにデータベースアクセスインタフェースが用意されたというレベルである。苦労して溜め込む同一の知識データを推論処理にも情報処理にも使うためには、モデルないし言語レベルの統合こそが必要である。

知識データベースモデルの研究は現在世界各国で活発に行われている。関連する研究も含めて動向を探ると、少し古くは、データベース分野で概念レベルにモデルを高める研究[Co79, HM78]がある。その後、長く基本であった正規形関係モデルを離れ、ＣＡＤやＯＡ用に複合オブジェクトを扱うマルチメディアデータモデルの研究が進められている[Ma87, MS86, WK86]。理論面では非正規形理論が盛んに研究されている[Mi87]。本稿のように知識ベースへの拡張を意図した研究も進められている[MI87, Ve87]。一方、プログラミング言語分野では、オブジェクト指向の考え方が台頭し、ST80[GR83]、Commonlisp[SB86]、C++[St86]、Objective C[Co87]等が登場した。データベース分野もオブジェクト指向の影響を強く受け、拡張モデルは殆どオブジェクト指向を取り入れている。また、最近、非正規関係モデルの基本要素である集合概念と人工知能の基礎である論理を組み合わせて、データ操作を包含した新プログラミング言語の提案も活発である[Ba87, BB87, BN87, Ku87, Yo87, YO85]。また、フレームとデータベースの間の情報変換の研究も見られる[Ch87]。以上のように幅広い研究が進められているが、知識データベースが実用に供し得るには更に更に実証的研究が必要である。

我々は、知識表現としてフレームモデルを選択し、フレームモデルと関係モデルを統合するモデルとしてフレーム論理を提案した[NK87a]。他方、関係代数を高速に実行する主記憶データベースマシンMACHの研究[NK87b]、関係論理を関係代数に変換する研究[Na87]を進めて来た。現在、MACHを拡張して、フレーム論理を実現する研究を進めている。オブジェクト指向は考え方の基本性、根源性故に、プログラミング言語、知識処理、データベースを貫き得るパラダイムと考える。即ち、オブジェクト指向を知識・データベース統合におけるキー概念と考え、それに基づいた実現を想定し、フレーム論理のリファインを進めて来た。便宜上、以前提案分をフレーム論理１、本稿のリファイン版をフレーム論理２と呼ぶ。

本稿は、知識データベースモデルであるフレーム論理２の概要を述べたものである。まず、２章で知識データベースへの要求条件を述べ、３章でフレーム論理１と２の違いをまとめる。次いで、４章でフレーム論理２の詳細を説明する。

2．知識データベースへの要求条件

知識データベースはどのような特質を備えるべきか？その解は多様であり得るし、また応用にも依存する。現在我々は当面の目標とする知識データベース（実際にはその処理系）に対する要求条件を以下のように考えた。

（a）知識データベースの構造表現能力は十分に柔軟であること（フレームモデル並みに）
（b）知識データベースの検索記述能力は十分に強力であること（関係モデル並みに）
（c）知識データベースの検索処理能力は十分に高速であること（データベースマシン並みに）
（d）知識データベースに関し基本的な無矛盾管理ができること
（e）オブジェクト指向プログラミング言語（ＯＯＰＬ）と無理なく結合されていること

3．フレーム論理１とフレーム論理２

フレーム論理２はフレーム論理１と比べて主に以下の点で進化したと考える。

（１）オブジェクト指向を明確に導入

フレーム論理１ではオブジェクト指向を明確に意識していないが、フレーム論理２ではそれを明確に意識した。例えば、ユーザが定義するフレーム群をインスタンスとするメタクラスと

してメタフレームを設け、検索系や更新系の標準的な手続きはそのクラスのメソッドとして位置づけた。

（2）2モデルの共存から単一モデルに統合

フレーム論理1ではモデル写像の考え方を導入し、フレームモデルと非正規関係モデルの融合共存を図った。フレーム論理2では単一モデルにまで昇華し、統合した。同時に、概念の整理とモデルのフォーマライズを強化した。例えば、インスタンスの遡及、フレーム解釈、グラフ表現と非正規表現、等である。なお、スロット構造はフレーム論理1では副スロットまで許していたが、単一値集合までで十分と判明したので改めた。

（3）手続きによる知識記述と制御の導入

フレーム論理1では暗黙推論はデフォルト値による値の単なる代入であった。フレーム論理2ではデフォルトスクリプトによる手続き制御とし、より多様なケースに対応可能とした。また、フレームモデルではよく知られる付加手続きも、フレーム論理1では明確でなかったが、フレーム論理2ではトリガスクリプトとして組み込みを明確にした。

（4）無矛盾管理と空値の導入

フレーム論理1では明確でなかった無矛盾管理を導入した。更に、空値（空集合）もサポートすることとした。

4．フレーム論理2

フレーム論理2はオブジェクト指向の考え方の下に、知識データベースを表現し、操作し、かつ管理するためのモデルである。その概要を以下6つのカテゴリに分け、考え方、簡単な例を交えて説明する。

4．1　知識データベース

知識データベースKdbは、オブジェクトの集合、及びクラス階層から成る。

Kdb＝（{Object}，Hierarchy）

4．2　オブジェクト

オブジェクトは知識データベース表現の基本要素であり、クラスまたはインスタンスである。オブジェクトはシステム一意のオブジェクトid（Oid）を有す。

Object＝Class　or　Instance

クラスはOid、クラス名、スーパークラス規定、スロットの集合、メソッドの集合から成る。

Class＝（Oid，Class_name，Superclasses，{Slot}，{Method}）

クラス名はシステム一意とする。クラスには、システム定義のクラスとユーザ定義のクラスがある。前者は唯一つだけ存在するが、後者はユーザによって幾つでも定義できる。ユーザ定義のクラスはそれぞれ名前が付けられるが、一般的にはフレームと呼ぶ。システム定義のクラスはメタフレームと呼び、以下の形式とする。

（Oid，metaframe，class，{}，{標準メソッド}）

スーパークラス規定では複数のスーパークラスが指定できる。

Superclasses＝{Class}

スロットはスロット名、型、デフォルトスクリプト名から成る。

Slot＝（Slot_name，Type，Dfs_name）

型は以下のいずれかとする。

Type∈{integer，string，frame，integers，strings，frames}

単数形は単一値、複数形は単一値の集合を意味する。また、デフォルトスクリプト名は値が空値の時に起動されるスクリプト（プログラム）を指定する。

メソッドはインスタンスメソッドである。ユーザ定義のクラスでは、メソッドはデフォルトスクリプトに他ならない。

インスタンスはスロット名と値の対の集合に、自分のOid、属すクラス名を付与したもので

ある。インスタンスはいずれか一つのクラスに属す。

Ｉnstance＝（Ｏid, Ｃlass_name, {（Ｓlot_name, Ｖalue）}）

インスタンスのスロット名は実は自由に設定できる訳ではなく、スロット継承と呼ぶ仕組みで決定される（４．３節参照）。

値は６種の型に対応した具体値の他に、空値（null value）を加える。単一値の集合を型とするスロットにあっては、空集合も値となり得る。型が'frame'の場合、その値はＯidであり、'frames'の場合はＯidの集合である。

［補足4.2.1］ここでスロットはインスタンス変数と同義で使った。インスタンス変数を避けたのは、関係論理のタプル変数との無用の混乱を防ぐためである。

［補足4.2.2］メタクラスには２通りの意味がある[SB86]。インスタンスのクラスのクラス（あるインスタンスのメタクラス）と、クラス群をインスタンスとするクラスである。前者はインスタンスの生成、初期化等のメソッドを持つ。メタフレームは両方の意味を持つとする。

［補足4.2.3］通常のオブジェクト指向では、クラスはクラス変数を持つが、ここではインスタンス変数に値を設定できるようにしたことでクラス変数の代用とする。

［補足4.2.4］値としてＯidが指定できるので、オブジェクト参照（referencing）が可能である。参照されるオブジェクトは必ず存在しなければならない。Ｏidによるオブジェクト参照は、データベース分野で知られるsurrogateによるタプル参照[Da83]と同じ概念である。また、Ｏidによるオブジェクト参照は、グラフによる構造表現[BB87]を可能にする。表現できるグラフはＤＡＧ（directed acyclic graph）だけでなく、サイクルも含んでよい。最近のデータベース分野のトピックである非正規形[Mi87]は情報構造としては木構造であるので、クラス階層を全く考えないときでも、フレーム論理２は非正規関係モデルを真部分集合として包含する。

フレーム論理２＝クラス階層＋グラフ構造
∪
グラフ構造
∪
非正規関係モデル（木構造）

［補足4.2.5］メタフレームはユーザ定義のクラスのメタクラスであり、インスタンスの生成、検索、更新等の標準的メソッドを持つ。そのスーパークラスは標準的に'class'とした。

［補足4.2.6］単一値、及びその集合はオブジェクトとして扱っていないように見えるが、それはインプリメント定義としているに過ぎない。そのとき、集合も一つのオブジェクトとして捉えれば、単数形、複数形の区別も不要になる。

［補足4.2.7］リストをどう扱うか？リストは記号処理の中心概念であり無視できない。リストは集合と比べたとき、要素の順序を問題とする。タイピングされたものという制約はあるが、集合を扱うことはできる。そこで、リストもキーをつけて集合にすれば扱える。例えば、四季の平均気温のリスト(20,30,22,12)は、((春,20),(夏,30),(秋,22),(冬,12))のようにすれば集合になる。勿論これだけでは十分ではないが、当面はリストに対しlispのような操作機能を持たないと考えているため、リストをリスト型としてサポートすることはしない。

［補足4.2.8］ユーザ定義のクラスはいわゆるフレームとしての特質を備えているので、フレームと呼ぶことにした。

［例１］フレーム論理１との差を明示するためにも、文献[NK87a]の例を再利用する。フレームの定義を以下に示す。フレーム論理１と比べて大きく異なるのは、子供をクラスとした点、副スロットを廃止した点、不明確な情報を空値で表現できる点、デフォルトスクリプトが登録できる点、等である。なお、空値は？で表わす。

```
(0001,従業員,{superframe},
    {(名前,string,？),(趣味,strings,？),(子供,frames,？)},
    {})；
```

```
(0002,管理者,{従業員},
    {(役職,string,dfs1),(部下,frames,？)},
    {(dfs1,{set manager;})});
(0003,子供,{superframe},
    {(名前,string,？),(年,integer,dfs2)},
    {(dfs2,{set avg[年](子供);})});
```

また、インスタンスは以下のように記述できる。

```
(0010,従業員,{(名前,X),(趣味,{T,M}),(子供,{0100})});
(0011,従業員,{(名前,Y),(趣味,{F}),(子供,{0101,0102})});
(0012,従業員,{(名前,Z),(趣味,{M,G}),(子供,{})});
(0001,管理者,{(名前,A),(趣味,{G,M}),(子供,{0103,0104}),
              (役職,director),(部下,{0011})});
(0002,管理者,{(名前,B),(趣味,{G}),(子供,{0105}),
              (役職,？),(部下,{0010,0012})});
(0100,子供,   {(名前,P),(年,5)});
(0101,子供,   {(名前,H),(年,9)});
(0102,子供,   {(名前,I),(年,6)});
(0103,子供,   {(名前,J),(年,14)});
(0104,子供,   {(名前,K),(年,？)});
(0105,子供,   {(名前,L),(年,16)});
```

4．3　フレーム階層とスロット継承

フレーム間に定義できるスーパー／サブの間柄はis_aのフレーム階層である。あるフレームのスーパー側の閉包を上位フレーム集合、サブ側を下位フレーム集合と呼ぶ。これらは元のフレームを含まない。

任意のフレームの任意のスロットはそのフレームの下位フレーム集合に属すフレームのスロットでもある。これをスロット継承と呼ぶ。

if　has（Frame, Slot）for ∀Frame, ∀Slot
then　has（F, Slot）for ∀F ∈sub_closure（Frame）

一つのフレームが複数のスーパークラスを持つことは可能である。即ち、スロットの多重継承を許す。スロットの多重継承はスロット名の和集合を用いて行う。

インスタンスのスロットはスロットの（多重）継承によってのみ決定されるのであり、独自のスロットは持てない。

［例2］例1において、管理者フレームは従業員フレームから、名前、趣味、子供の3つのスロットを継承した。

4．4　フレーム解釈とインスタンス遡及

知識データベースの中のあるフレームに着目し、そのフレームに属す各インスタンスの値を確定することをフレーム解釈と呼ぶ。確定されたインスタンスの集合は、値の集合をスロット値として取り得るので、形の上からは集合を値とする非正規関係（unnormalized relation）を構成する。しかし、個々の値はOidを取り得るので、オブジェクト参照によるグラフ構造が内包されている。即ち、内容的には通常の非正規関係ではない。

任意のフレームの任意のインスタンスはそのフレームの上位フレーム集合に属すフレームのインスタンスでもある。これをインスタンス遡及と呼ぶ。

if　has（Frame, Instance）for ∀Frame, ∀Instance
then　has（F, Instance）for ∀F ∈super_closure（Frame）

インスタンスのスロット値は設定されていればその値とするが、空値のときは、継承の源のスロットに登録されたデフォルトスクリプトを起動して値を得る（暗黙推論）。デフォルトスクリプトが登録されてないときは、空値のままである。

［補足4.4.1］スロットの多重継承時、スロット名の重複により、結局全くスロットを継承しなかったフレームには（その上位のフレームも含めて）、インスタンスの遡及はない。
［補足4.4.2］正規形関係は属性値が単一値であるが、非正規関係では属性値を、(a)単一値の集合、(b)タプル（tuple valued）、(c)関係（relation valued）、のように拡張してきた。しかし、属性値を関係にしても、所詮、木構造に過ぎないことは［補足4.2.4］で述べた。フレーム論理２では属性値（スロット値）として(a)単一値の集合までしか許さないが、オブジェクト参照を加味しているので、(c)の非正規関係より柔軟な表現が可能となっている。
［補足4.4.3］インスタンス遡及とスロット継承は対をなす概念である。即ち、スロットが継承された道筋に沿ってインスタンスは遡及する。

［例３］例１において、従業員、管理者、子供の各フレームを解釈すると以下となる。このとき、Oid＝0001とOid＝0002のインスタンスが従業員クラスに遡及した。また、Oid＝0002のインスタンスの役職スロット値と、Oid＝0104のインスタンスの年スロット値が未知であるが、それらには共にデフォルトスクリプトが登録されているので、それらを起動することにより、各々、役職＝manager、年＝10を得る。

＜従業員＞

Oid	名前	趣味	子供
0010	X	T	0100
		M	
0011	Y	F	0101
			0102
0012	Z	M	
		G	
0001	A	G	0103
		M	0104
0002	B	G	0105

＜管理者＞

Oid	名前	趣味	子供	役職	部下
0001	A	G	0103	director	0011
		M	0104		
0002	B	G	0105	manager	0010
					0012

＜子供＞

Oid	名前	年
0100	P	5
0101	H	9
0102	I	6
0103	J	14
0104	K	10
0105	L	16

４．５　メソッド

既に述べたように、メタフレームは標準メソッドを、フレームはデフォルトスクリプトをメソッドとして持つ。
標準メソッドには以下のものがある。

- インスタンス生成メソッド
- インスタンス更新メソッド
- グラフ型非正規関係論理
- グラフ型非正規関係代数

最初の２つは文字通りのメソッドである。ユーザはこれらにトリガスクリプトが登録できる。メソッド実行後、トリガスクリプトが登録されていれば、それを起動する。また、グラフ型非正規関係論理、グラフ型非正規関係代数はデータベースの強力な検索言語である。前節で述べたように、フレームは解釈するとグラフ構造を内包した非正規関係になるので、通常の非正規関係モデルの検索系に、グラフ構造を辿る手段を追加する必要がある。

[補足4.5.1] トリガスクリプトは従来のフレームの付加手続き（$if-added,$if-removed）に相当する。

[例４] 実は[NK87a]で提案したものはその拡張がなされている。そこでの例を一部再掲する。
＜１＞ゴルフが趣味の従業員は誰か。
　　(u[名前]):従業員(u):∃(u[趣味])(w)(w=G)
＜２＞子供のいない従業員のＯidを求めよ。
　　(u[Ｏid]):従業員(u):count(u[子供])=0
＜３＞管理者の名前と部下の数を求めよ。
　　(u[名前],count(u[部下])):管理者(u):()
＜４＞趣味が一致する管理者と部下及び一致する趣味リストを求めよ。
　　(u[名前],v[名前],(w:(u[趣味])(w),(v[趣味])(x):w=x)):管理者(u),(u[部下])(v):()

フレームにユーザが登録できるデフォルトスクリプトはスロット値が空値のときに起動されるプログラムである。

[例５] 例１では以下のようなデフォルトスクリプトを登録した。dfs1は空値にmanagerを代入し、dfs2は子供の平均年令avg[年](子供)を代入する。
　　dfs1,{set manager;}
　　dfs2,{set avg[年](子供);}

４．６　無矛盾管理

システムで行う無矛盾管理は、参照制約、スロット多重継承、インスタンス遡及等の基本的なものである。その他、ユーザも無矛盾性条件を定義できるが、それらは検索表現をベースとしたものとする（例６参照）。

[補足4.6.1] 無矛盾管理を余りにも強力にすると、性能に大きな皺寄せが来ることが知られている。商用のデータベース管理システムでも最低限のことしか行っていない。また、述語論理の一般的な反駁能力を頼って知識ベースの無矛盾管理を行うと、格納量が多くなるにつれてどうにも機能しなくなると予想される。

[例６] 意味はさておき以下の例を考える。
　　“管理者はゴルフの趣味をもたねばならない。”
このようなユーザの意味を管理する無矛盾制約は次のように検索に置き換えて考える。
　　“ゴルフが趣味でない管理者はいない。”
　　assert　Q≒φ　where　Q＝((u):管理者(u):～∃(u[趣味])(w)(w=G))
従って、この意味制約はＱの検索により検証できる。

５．おわりに

知識ベースとデータベースの統合という難問題の解決の緒として、フレームモデルと関係モデルとの統合を図る知識データベースモデル（フレーム論理２）について述べた。フレーム論理２は、以前提案したフレーム論理１をオブジェクト指向による実現の観点から見直したものである。フレーム論理１と比べ、オブジェクト指向の導入、２モデルの共存から単一モデルに統合、手続きによる知識記述と制御の導入、等が進化した。また、本稿では触れなかったが、ＯＯＰＬとＯＯＫＤの結合に関しても各種方式があり[Ba87]、議論が必要であろう。今後は、プロトタイプを試作し、本アプローチによる知識データベース方式の有効性を検証する予定である。

参考文献

[Ba87] Bancilhon, F., "Object Oriented Multilanguage Systems: the Answer to Old and New Database Problems?," France-Japan AI and CS sympo87, Cannes, Nov., 1987.
[BB87] Bancilhon, F., Briggs, T., et al., "FAD, a Powerful and SImple Database Language," Proc. 13th VLDB Brighton, 1987.
[BN87] Beeri, C., Naqvi, S., et al., "Sets and Negation in a Logic Database Language(LDL 1)," Proc. PODS, March 23-25, 1987.
[Co79] Codd, E. F., "Extending the Database Relational Model to Capture More Meaning," ACM TODS, 4, 4, pp. 397-434(1979).
[Co87] Cox, B. J," Object-Oriented Programming," Addison-Wesley, 1987.
[Ch87] Chow, E., "Representing Databases in Frames, "Proc. AAAI, Seattle, July 13-17, 1987.
[Da83] Data, C. J., "An Introduction to Database Systems Vol. 2," Addison-Wesley, 1983.
[GR83] Goldberg, A. and Robson, D., "Smalltalk-80 The Language and Its Implementation," Addison-Wesley, 1983.
[HM78] Hammer, M , McLeod, D.," The Semantic Data Model: A Modellig Mechanism for Data Base applications," Proc. SIGMOD, Austin, May 31-June 2, 1978.
[Ku87] Kuper, G. M.," Logic Programming With Sets," Proc. PODS, March 23-25, 1987.
[Ma87] 増永, "マルチメディアデータベース総論," 情報処理, 28, 6, June, 1987.
[Mi87] 三浦, "非正規形関係データベース理論の動向," アドバンストＤＢシステム, 12月, 1987.
[MI87] 牧之内, 石川, "マルチメディア知識ベースシステム操作言語の基本概念," 情処研究会, 87-DB-62, 1987.
[MS86] Maier, D. and Stein, J.," Development of an Object-Orinted DBMS," Proc. OOPSLA, Portland, Sep. 30-Oct. 2, 1986.
[Na87] 中野, "非正規関係論理／関係代数変換法," 情処研究会, 87-DB-61, 1987.
[NK87a] 中野, 木山, "フレーム論理," 情処研究会, 87-AI-50, 1987.
[NK87b] Nakano, R. and Kiyama, M., "MACH: Much Faster Associative Machine, "Proc. IWDM' 87, Karuizawa, Oct. 5-8, 1987.
[SB86] Stefik, M. and Bobrow, D. G., "Object-Oriented Programming: Themes and Variations" AI magazine, 6, 4, Winter, 1986.
[St86] Stroustrup, B., "The C++ Programming Language," Addison-Wesley, 1986.
[Ve86] Vermeir, D., "OOPS: A Knowledge Representation Language, "Proc. Int. Conf. on Compu ter Language, Miami, Fla, Oct., 1986.
[WK86] Woelk, D., Kim, W. et al., "An Object-Orinted Approach to Multimedia Databases," Proc. SIGMOD, Washington, D. C., May 28-30, 1986.
[Yo87] 横田, "非正規形モデルへの演繹的アプローチ," 情処研究会, 86-DB-58, 1987.
[YO85] 山内, 大須賀, "多層論理のモデル理論と証明論," 情処研究会, 85-AI-42, 1985.