6Q-8 オブジェクト指向型マルチメディア知識ベース Jasmine のモデルと操作言語について

石川 博, 鈴木文雄, 牧之内顕文

富士通株式会社

1.はじめに

近年データベースの新しい応用分野としてCAD, OIS. AI等が注目されるようになってきた。それら応用の特徴を要約すると以下のようになる。

・応用は知識集約型である。したがって対象分野のモデルを知識ベースとしてユーザ（アプリケーション・プログラマ）が柔軟に定義できる必要がある。
・モデルは対象にできるだけ忠実でなくてはならない。したがって対象分野の構造，関係，動的振舞い（プログラム）を，分野に自然なマルチメディアデータ（文字，数値，画像，図形等）を組み合わせることにより直接的に表現できる必要がある。
・応用の必要とする知識の量が多い。したがって大量知識でも妥当な処理速度で応用プログラムを実行できなくてはならない。

こうした応用をサポートするシステムを知識ベース管理システムという。知識ベース管理システムに対しては，プログラミング言語の分野で最近注目されているオブジェクト指向のアプローチ[1]が有望である。なぜなら，先ず第一にオブジェクトの属性，関係，メソッドが対象の構造，関係，動的振舞いを直接表現できるからである。次にオブジェクトはメディアデータとその操作を1つにカプセル化することができる。またオブジェクトのポリモルフィズム[1]の考えは，異種メディアに対する統一的インタフェースの実現に役立つ。

しかしながら，プログラミング言語におけるオブジェクト指向のアプローチは以下のような点で改善の余地がある。先ず既存のアプローチは，集合論や関数論のような明確なセマンティクスをもっていない。その結果，ユーザが知識ベースを系統的に作成したりすることを難しくしている。また明確なセマンティクスの欠如は言語機能の理解を困難にする。さらにオブジェクトの二次記憶による管理がないし，大量オブジェクトの中から条件つきでオブジェクトを効率良く検索することができない。

本論文は上記の問題点を解決する知識ベース管理システム（Jasmine）のモデルと操作言語について説明する。

2.知識モデルのセマンティクス

2.1オブジェクト内セマンティクス

モデルのオブジェクト内セマンティクスを関数論をもちいて，定義する。文字・数値（即値型）と図形・画像（参照型）を原子オブジェクト，他のオブジェクトからなるものを複合オブジェクトという（図1参照）。オブジェクトはクラスとインスタンスに分類され，インスタンスは個々の知識に相当し，クラスはインスタンスの構造，関係とメソッド（まとめて属性という）を記述する（図2参照）。オブジェクトの属性はオブジェクトからその属性値のオブジェクトへの関数と見做す。インスタンスの属性値はそのクラスのインスタンスでなくてはならない（図3参照）。関数の諸性質（多値か単一値か，完全か部分か等）は属性のファセットで記述される。ファセットにはデーモンも記述できる。関係にはSUPER とCLASS がある。オブジェクトの等号はオブジェクト識別子に基づく。即値型原子オブジェクトの識別子は値そのものである。参照型原子オブジェクト及び複合オブジェクトの識別子はシステムかユーザのどちらかが生成する。属性値は参照完全性を満足しなくてはならない。

2.2オブジェクト間セマンティクス

モデルのオブジェクト間セマンティクスを集合論を用いて，以下のように定義する。

・クラスはインスタンスの集合である。
・インスタンスは本質的にただ1つのクラス（それを本質的クラスという，逆にインスタンスは本質的インスタンスという）に属する。
・スーパークラス（例PERSON）はサブクラス(PATIENTとDOCTOR) を集合として包含する。さらにサブクラスはスーパークラスの属性（例Name）を遺伝する。
・スーパー／サブ階層のどのクラスも本質的インスタンスをもちうる。ノンリーフのクラス（例PERSON）の本質的インスタンスは不完全な情報（例えば未だ医者でも患者でもない状態）を表現する。
・スーパークラスは，その本質的インスタンスと，互いに集合的に交わらないサブクラス（それらをファミリと総称する）のインスタンスとの集合和になる。一般にクラス（例PERSON）は複数のファミリ（例MALE/FEMALE とDOCTOR/PATIENTファミリ）に分割できる。
・マルチスーパークラスは，同一スーパークラスの異なるファミリのメンバの中で互いに交わりを持つメンバ（例MALEとDOCTOR）でなくてはならない。
・クラスは特別なクラスのインスタンスである。

3.知識ベース操作言語

知識モデルセマンティクスは言語設計に影響を与える。オブジェクト間セマンティクスに対して，集合論的インタフェースがあり，オブジェクト内セマンティクスに対して単一インタフェースがある。

3.1集合論的インタフェース

以下のように条件を満たすオブジェクトを知識ベースから検索することができる。

```
PATIENT where PATIENT.Age > 45
```

操作言語としてはオブジェクト式（クラス名の後に任意個の属性名をつなげたもの，例えばPATIENT.Age)を基本として条件検索ができる。この場合クラス名はインスタンスの上に束縛される繰り返し変数とみなせる。オブジェクト式の結果は最後の属性の値である。遺伝属性（例Age)も指定できる。さらに以下のようにスーパー／サブ階層におけるノンリーフのクラス（例PERSON）も指定でき，その場合はその本質的インスタンスとサブクラス(DOCTOR とPATIENT)にふ

The Model and Language of the Object-Oriented Multimedia Knowledge Base System Jasmine

Hiroshi ISHIKAWA, Fumio SUZUKI, Akifumi MAKINOUCHI

FUJITSU LIMITED

くまれる全てのインスタンスの集合和が検索対象となる。

```
PERSON where PERSON.Address="Tokyo"
```

このように集合論的インタフェースでは完全・不完全，遺伝・非遺伝を問わず知識を統一的に扱える。条件はオブジェクト識別子に基づきオブジェクト式を比較する。以下のように複雑な条件も指定できるし，メソッドも属性なので検索以外の処理も行える。

```
PATIENT.make-medical-certificate(4)
where (PATIENT.Age<20 or PATIENT.Age>30)
 and   PATIENT.Name==PATIENT.Doctor.Name
```

3.2.単一インタフェース

単一インタフェースはメソッドを記述したり，オブジェクトをC言語で操作するためにある。オブジェクト型の変数の宣言，参照，操作は以下のようにする。

```
classname insvar                インスタンス型変数の宣言
insvar.attri. ... .attri        インスタンス型変数の参照
insvar<OID>.attri. ... .attri オブジェクト識別子による参照
insvar.attri. ... .selector(parameters) メッセージ送信
```

スーパー／サブ階層のどのクラスも指定できる。例えば

```
LARGE media;
media.expand(2);
```

Media にはそのサブクラス（例えばIMAGE や GRAPHICS ）のどのインスタンスも束縛される。さらにそれらに対して同一のメッセージを送ることにより，そのメソッドの実現に関わらず統一的に操作することができる（ポリモルフィズム）。一般に，オブジェクト指向のアプローチでは，操作対象をクラスによってモデル化しさえすれば，オブジェクトとして管理・利用できる（オブジェクト指向の原則）。この原則をマルチメディアデータに対して適用すれば，図形画像やプログラム等が知識ベースの中に管理できる（図1参照）。

3.3クラス操作

クラスは特別なクラス（CLASS ）のインスタンスなので，通常のインスタンスと同様に宣言し，操作することができる。

```
CLASS classvar                  クラス型変数の宣言
classvar.selector(parameters)   クラス型変数の参照
<class>.selector(parameters)    クラス名による直接的参照
```

ユーザは集合論的インタフェースの結果クラスをスキャンして，個々のインスタンスをアクセスする。

```
<TPATIENT>=PATIENT where PATIENT.Age<12;
scan=<TPATIENT>.openscan();
while  (p=<TPATIENT>.nextscan(scan))
     { ...  p.print(); ... }
<TPATIENT>.closescan(scan);
```

以下のようにクラスの条件検索もできる。

```
CLASS where CLASS.Property.Default=="outpatient"
```

このように知識ベース操作言語はインスタンスもクラスも統一的に操作できる。これはメタデータに対しても，前述のオブジェクト指向の原則を適用しているからである（図1参照）。

4.おわりに

オブジェクト指向の知識ベース管理システムのモデル及び言語について説明した。本システムはオブジェクト指向型データベース[2]と機能的に似ているが，それらとは，不完全知識の操作及びクラスの条件検索ができる点で大きく異なる。

謝辞

この研究は通産省工業技術院大型プロジェクト「電子計算機相互運用データベースシステムの研究開発」の一環として行った。

参考文献

1.Banerjee, J. et al., "Data Model Issues for Object-Oriented Applications," ACM TOOIS, vol.5, no.1, January 1987, pp.3-26.

2.Stefik, M. and D.G. Bobrow, "Object-Oriented Programming: Themes and Variations," The AI MAGAZINE, vol.6, no.4, winter 1986, pp.40-62.

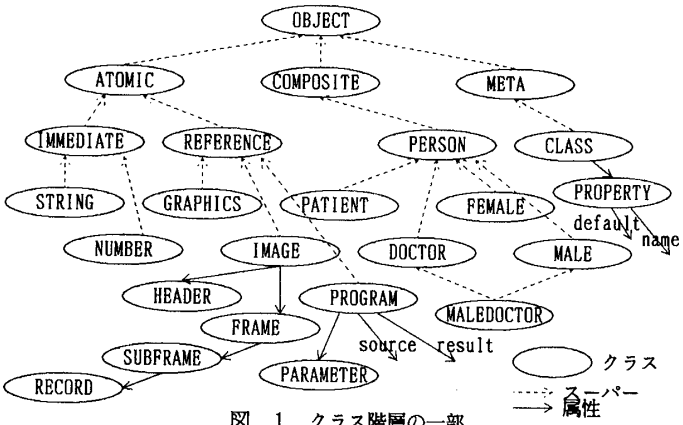

図 1 クラス階層の一部

```
PATIENT
  Kb        MEDICAL
  Super     PERSON
  Property  DOCTOR Doctor   multi   "outpatient"
            STRING Category default
            FLOAT  Weight   mandatory
            FLOAT  Height   if-needed
                            patient.Height<
                            int i;
                            i=self.Weight;
                            return i+100.0;>
  Method  VOID    make-medical-certificate(no)
                  int no;
                  <MEDICAL-CERTIFICATE mc;
                  int i;
                  mc=<MEDICAL-CERTIFICATE>.instantiate();
                  for(i=0; i<no; i++) mc.print();>
```

図 2 クラス定義の例

```
patient007
      Sex          "male"
      Age          "36
      Name         "James„Bond"
      Address      "Tokyo
      Doctor       „doctor010„ doctor021
      Category     "inpatient"
      Weight        76.00
      Height       181.00
```

図 3 インスタンスの例