

実体の多面性を表現するためのデータモデル

5R-1

舟橋 栄 *

佐藤 秀樹 †

池田 峰輝 ‡

林 達也 *

*名古屋工業大学 電気情報工学科

†日本電装(株)

‡Heart Field Japan

1 はじめに

本論文では、実体の多面性を表現するためのデータモデルとして、多面的オブジェクト指向データモデル MAORI(Multi- Aspects Object Oriented Data Model)を提案する。このモデル化では、今までのモデル化で不十分だったオブジェクトの多面的な性質の表現、任意の側面の動的な獲得・喪失といった機能を提供する。

MAORI では、実世界の実体は複数の側面(aspect)を持つオブジェクトとしてモデル化される。オブジェクトの各 aspect は、クラスのインスタンスとして表現され、オブジェクトは複数クラスに属することができる。オブジェクトの aspect 集合には、同じクラスが定義する複数 aspect が存在してもよい。これを可能にするために aspect 識別子を導入する。aspect 識別子は、クラス名並びに当該 aspect 集合中で同じクラスが定義するインスタンスを特定する添字から構成される。オブジェクトが持つ aspect 集合の各要素は、この aspect 識別子により特定される。さらに、MAORI は、実世界の実体では有り得ない aspect の組を持つオブジェクトが生ずることがないようオブジェクトの aspect に関する制約を記述するオブジェクトスキーマを備えている。MAORI の枠組み、データ定義、操作言語とその実装についても述べる。

2 実体の多面性

人間を実体とする多面性を扱うデータモデルの例：

1. A 氏は student であり、同時に会社員である。
2. A 氏は卒業した。student なる側面を喪失し、graduate なる側面を獲得。
3. A 氏は複数の課の課長を兼任することになった。同じクラスに複数の側面を獲得。

3 多面的オブジェクトモデル MAORI

3.1 多面的オブジェクト

MAORI の基本要素は、複数クラスに属するオブジェクト、オブジェクトの1つの側面を表わす aspect、aspect

Datamodel for Representing Multi-Aspects of Entities

Sakae FUNAHASHI (NIT),
Hideki SATO (NIPPONDENSO Co., Ltd.),
Miki IKEDA (Heart Field Japan Inc.),
Tatsuya HAYASHI (NIT)

Department of Electrical and Computer Engineering,
Nagoya Institute of Technology (NIT),
Gokiso-cho, Showa-ku, Nagoya, 466, Japan

の表現であるインスタンス、インスタンスの形式（型）の定義とオブジェクトの分類を行うクラスである。

インスタンス i の形式は $\{a_1 = v_1, \dots, a_n = v_n\}$ である。ここに $a_i = v_i (1 \leq i \leq n)$ は属性名と属性値である。

クラスはインスタンス (instance) の形式（型）を定義する。クラス名は "@" で始まる英字列で表す。

オブジェクト O は、 (oid, as) の対である。ここで、oid はオブジェクト識別子、as は aspect 集合である。as の形式は $\{\langle aid_1, i_1 \rangle, \dots, \langle aid_k, i_k \rangle\}$ であり、各 $aid_j, i_j (1 \leq j \leq n)$ はそれぞれ aspect 識別子、インスタンスである。

オブジェクトは、実世界の実体をモデル化する。オブジェクト識別子は、オブジェクトを一意に識別する。オブジェクト識別子は "#" で始まる数字列で表す。オブジェクトの aspect 集合に対する要素の追加、削除により、側面の動的な獲得・喪失が表される。

オブジェクト O は、次のクラス集合 CS の要素に属する (belonging-to)。但し $class(i)$ は、インスタンス i を定義するクラスを表す。 $CS = \{class(i_j) | o = \langle oid, \{\langle aid_1, i_1 \rangle, \dots, \langle aid_k, i_k \rangle\} \rangle, 1 \leq j \leq k\}$ また、クラス C に属するオブジェクトの識別子集合 $\{oid | \langle oid, as \rangle \in C\}$ を $OBJ(C)$ と記す。この定義より、オブジェクト O がクラス C に属する場合、 C が定義するインスタンスを O が aspect として持っていることになる。

図1に、オブジェクトの例を示す。このオブジェクトは、クラス@STUDENT, @WORKER, @MANAGER に属する。@PERSONで定義される属性NAME, AGEは、全ての aspect で値が共有されている。また、@WORKER で定義される属性DEPT, SALARYは、3つの aspect で値が共有されている。

3.2 aspect 識別子と属性参照

aspect 識別子は、 $\langle c_n, i_x \rangle$ からなる。ここで、 c_n は aspect のインスタンスを定義するクラスの名前で、 i_x はオブジェクト内で同一クラスが定義する aspect を識別するための添字である。aspect 識別子の添字は "@" で始まる数字列で表す。

aspect の属性参照は、属性の名前を aspect 識別子で修飾することにより行われる。多重継承を使ってクラスを定義する場合、継承される上位クラスの属性定義は単一の文脈にまとめられる。このため、異なる上位クラスで定義される、同じ名前を持つ属性に関して名前の衝突が生じる。しかし多面的オブジェクトの aspect は各々が独

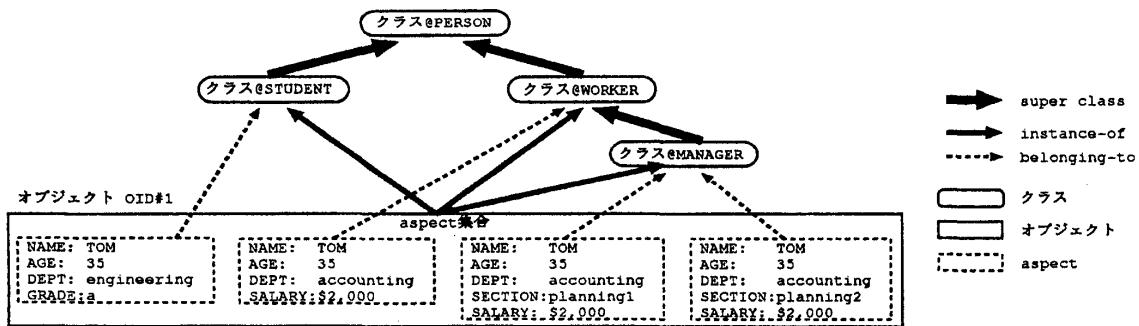


図 1: MAORI オブジェクト

立の文脈を持つため、異なる aspect が同じ名前の属性を持つ場合にも名前の衝突は生じない。図 1 のオブジェクトにおいて、aspect@STUDENT と aspect @WORKER は、名前は同じであるが、異なる属性 DEPT を持つ。すなわち、前者はクラス@STUDENT で定義される所属学部を表す属性であり、後者ではクラス@WORKER で定義される所属部署を表す属性となる。

4 オブジェクトスキーマ

従来の多面的オブジェクト指向データモデルに関する研究では、オブジェクトは複数 aspect を持つことはできるが、それらの間の意味的一貫性の維持は利用者・応用プログラムに任せられていた。このため、実世界の実体ではありえない aspect の組を持つオブジェクトが存在する可能性がある。これに対して、MAORI は意味的一貫性の維持のため、オブジェクトの aspect に関する制約を記述するオブジェクトスキーマを備える。

図 2 は、オブジェクトスキーマの定義構文を示す。オブジェクトスキーマ名は "# " で始まる英文字列で表す。

```
define object schema  ( オブジェクトスキーマ名 )
  ( aspect 式 );
<aspect 式> ::= { ( aspect 指定列 ) }
<aspect 指定列> ::= ( aspect 指定 )
<aspect 指定列> ::= ( aspect 指定 ), ( aspect 指定列 )
<aspect 指定> ::= <クラス指定>
<aspect 指定> ::= ( <競合 aspect 指定列 > )
<競合 aspect 指定列> ::= <クラス指定>
<競合 aspect 指定列> ::= ( <クラス指定> ) | ( 競合 aspect 指定列 )
<クラス指定> ::= <クラス名>
<クラス指定> ::= <クラス名> ( aspect 最大数 )
<aspect 最大数> ::= <自然数>
<aspect 最大数> ::= +
```

図 2: オブジェクトスキーマの定義構文

5 データ定義・操作言語

[2つの課の課長を兼任し社会人コースの学生となっている会社員オブジェクトの作成] を例に MAORIL の mu-Lisp で実装したコマンドを説明する。\$ は mu-lisp でのプロンプト、#mn で、m はオブジェクトタイプの番号、n はカウンターを表す。

オブジェクトの作成

- クラスの定義


```
$ (defclass @STUDENT(@PERSON)((DEPT *char)(GRADE *char)))
```
- オブジェクトスキーマの定義


```
$ defobject @PERSON((@STUDENT)(@GRADUATE))(@WORKER)
(@MANAGER 2))
```
- #10


```
$ (makeobject #PERSON)
```
- #11


```
$ (getobject #11)
```

オブジェクト識別子を第一要素とする梓組のリストができる
- アスペクトの作成


```
$ (makeaspect (#11)@STUDENT)
(@STUDENT @1)
```

同様にして aspect @WORKER, @MANAGER (2つ) を作成する
- ```
$ (makeaspect (#11)@WORKER)
(@WORKER @1)
```
- ```
$ (makeaspect (#11)@MANAGER)
(@MANAGER @1)
```
- ```
$ (makeaspect (#11)@MANAGER)
(@MANAGER @2)
```
- ```
$ (getaspect (#11))
((@STUDENT @1) (@WORKER @1) (@MANAGER @1 @2))
```

6まとめ

本論文では、オブジェクト指向データモデルを拡張した多面的オブジェクト指向データモデル MAORI とそのデータ定義・操作言語 MAORIL を定義した。MAORIL の実装に際し mu-LISP を用いた。クラスの定義のたびに、コマンドで upper-class のリストを指定することによりクラス名の束構造をメモリ内に確保した。また aspect の属性参照に aspect 識別子を導入した。これによりオブジェクトの側面の作成、属性の継承をプログラム化するのが容易になった。今はオブジェクトの属性値の更新、クラス定義の更新など、広く波及する部分の実装のための検討を進めている。

参考文献

- [1] 塚田晴史、杉村利明：“MAC モデル：複数観点からの分類が可能なオブジェクト”，コンピュータソフトウェア, 11, 5, pp.44-57 (1994-09)
- [2] 佐藤秀樹、池田峰輝、舟橋栄、林達也：“多面的オブジェクト指向データモデル”，信学技報, DE95-73(1995-12)