

2M-10

複数のクラスに所属するオブジェクト

塚田晴史

tsukada@ntt-elis.ntt.jp

(NTTヒューマンインタフェース研究所)

杉村利明

sugi@ntt-elis.ntt.jp

1. はじめに

オブジェクト指向言語のデータベースや知識ベースへの応用においては、一つのオブジェクトを様々な観点から利用することが有効になる。本発表では、複数のクラスに所属して、場面による使い分けができる multiple object を提唱する。これは所属クラス毎に構造や振舞いの異なる aspect を持ち、aspect の間で共通のデータを持つことも可能なオブジェクトである。

次に CLOS のメタオブジェクトを用いたプロトタイプ作成の方針について述べる。内部構造の異なるオブジェクトの取り扱い、multiple object 自身の所属クラスの問題について考察を行う。

2. 複数のクラスに所属するオブジェクト

複数のクラスに所属するオブジェクトを、人事データベースを例に説明する。会社では、研究/人事/総務などの職種、組合員/管理職といった採用形態などいくつかの分類がある。Mike は、職種は researcher だが、採用形態からは manager である。"Mike" オブジェクトは2つのクラスに同時に所属して、researcher としてアクセスすると researcher の側面が、manager としてはそちらの側面が見える(Fig. 1)。これは Mike の二つの面を表している。二つのクラスのスロットで name と position の名前が重複している。このうち name スロットは同じ値を保持するのが自然だが、researcher クラスの position スロットは研究者としての立場を、manager クラスではラインの中での役

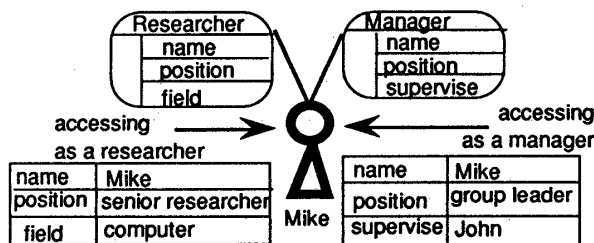


Fig.1 Multiple Object

割を表すので、それぞれ独自の値を持つ (Fig. 1)。

この例から、オブジェクトが複数のクラスに所属する条件を考察する。まずアクセス時に指定したクラスのインスタンスとして扱えることである。一つのオブジェクトにクラス名を指定すると、スロットやメッセージに対する振舞いが、そのクラスに従って定まる。次に一つのオブジェクトの、クラスを指定されたインスタンスの間で、協調が可能でなければならない。情報を共有したり、他の所属クラスを知る必要がある。

以後複数のクラスに所属するオブジェクトを multiple object、multiple object にクラスを指定した一つのクラスに属する側面を aspect object と呼ぶ。multiple object は aspect object の和集合と見なせる(Fig. 2)。

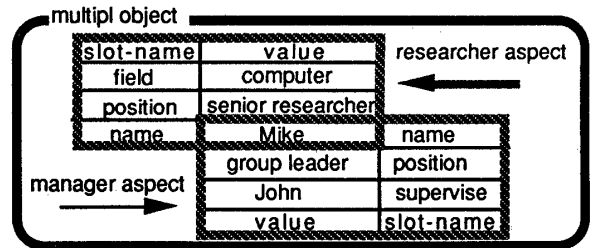


Fig. 2 Multiple Object and Aspect Object

複数のクラスに所属するオブジェクトを、従来の手法で表わそうとすると、まず多重継承が考えられる。だが多重継承では、aspect 毎の切り出しができず、継承の優先順位のため、非優先クラスのメソッドはシャドウされる。これに対して multiple object は aspect を目的のクラスのオブジェクトとして扱うことができる。また composite object では aspect 間のデータ共有ができない。一般に composite object は部品オブジェクトをひとまとめに扱うことに有効なのに対し、multiple object はオブジェクトの多面性を表す。

3. プロトタイプの作成

multiple object の有効性を確認するため、CLOS のメタオブジェクトを用いてプロトタイプを作成中である。作成にあたり CLOS の標準システムは変更せず、その上に新しいシステムを構築する方針を選んだ。このためあるオブジェクトが multiple object の aspect に

The Multiple Object Concept and its Implementation

TSUKADA Seishi SUGIMURA Toshiaki

NTT Human Interface Laboratories

なるか、唯一つのクラスにしか所属できないかは、クラス生成時に指定したメタクラスで決る。

3.1 標準と異なるオブジェクトの導入

今回プロトタイプ作成のために multiple object, aspect object の2種類のオブジェクトを導入した。aspect object は唯一つのクラスに所属して、ユーザからは standard object と同様に取り扱うことができる。しかし他の aspect object とスロットの一部を共有できるため、スロットの内部構造が標準と異なるので、それにあわせてスロットアクセスメソッドを定義した。また multiple object には aspect object を保持する機能が必要である。

オブジェクトの構造はメタクラスが決める。aspect object のために aspect-class、multiple object に multiple-class と2つのメタクラスを定義した。クラス定義で (:metaclass aspect-class) と指定されたクラスは、そこから生成されたインスタンスを aspect とする multiple object を作ることができる。multiple-class は multiple object の構造と振舞いを定め、ユーザからは見えない。

メタクラスはクラスレベルの動作しか定義できない。そこで全ての aspect object に共通なスーパークラス aspect-object を定め、また全ての multiple object は multiple-object クラスに所属するとした (Fig. 3)。multiple-object クラスについては、次節で詳しく述べる。

3.2 multiple object の所属クラス

multiple object は aspect object の和集合と考えられるが、それ自身の所属クラスを何とするかは、利用する立場によって異なる。

ユーザからは、aspect object の属するクラスの集合のORと考えるのが自然である。一方システムは、いくつかのメソッドを multiple object に起動する必要がある (e.g. instance initialization)。そこでメソッドを定義するために、multiple object 自身が一つのクラスに

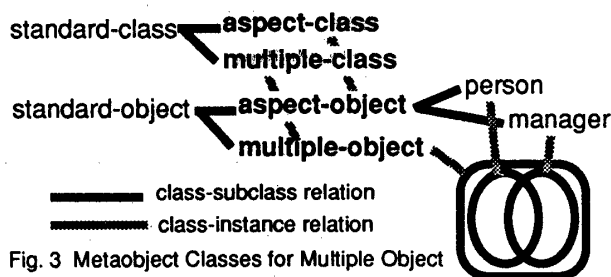


Fig. 3 Metaobject Classes for Multiple Object

所属している必要がある。

システムに必要なメソッドは、aspect object の所属クラスと無関係に動作が決るものなので、全ての multiple object は multiple-object クラスに所属すると規定する。ユーザはここにメソッドを定義できない。

multiple object の所属クラスを次のように整理する。ユーザから見た multiple object の所属クラスとは、aspect object のクラスのリストで、この値を返す関数 aspect-of を導入する。一方システム内部で multiple object は multiple-object クラスに所属して、このクラスのメソッドが動作を規定する。

4. まとめ

複数のクラスに所属し場面による使い分けができる multiple object を提唱した。これは、一つのオブジェクトの様々な側面を表すもので、所属するクラス毎に構造や振舞いの異なる aspect を持ち、aspect 間でデータ共有による協調が可能である。次に CLOS のメタオブジェクトを用いたプロトタイプ作成の方針を述べ、内部構造の異なるオブジェクトの取り扱い、multiple object 自身の所属クラスについて考察を行った。

今後の課題としては、multiple object のリスブオブジェクトとしてのタイプと、オブジェクトシステムにおけるクラスとの整合性に関する考察が必要である。タイプとクラスの融合は CLOS でも試みられているが、まだ不十分な部分も多い。また応用面からの multiple object の有効性も確かめて行きたい。

<謝辞> 本研究を行うにあたり、日頃からご指導いただき様々な助言を下さった日比野グループリーダーに感謝いたします。

6. 参考文献

- [1] D. Bobrow, K. Kahn, etc.: Common Lisp Object System Specification, SIG PLAN NOTICES 23-9 1988
- [2] G. Kiczales and D. Bobrow: Common Lisp Object System Specification 3, Metaobject Protocol, ANSI X3J13, July 30, 1990
- [3] 塚田, 杉村: ダイナミックオブジェクトの提唱と実現, 信学会ソフトウェアサイエンス研究会 SS90-2, 1990
- [4] 塚田, 杉村: Multiple Object の提唱, 情処記号処理研究会, 1990.1