

オブジェクト指向表現のための同一化による構成支援

A method for identifying Object-Oriented representations

片山 佳則

富士通(株)国際情報社会科学研究所

オブジェクト表現の支援環境として必要な、オブジェクトが持つ機能の同一性の判定方法について述べる。これは、オブジェクトが持つ機能的まとまりを有向グラフに対応させ、その隣接行列等の同形によって、同一化を支援するものであり、オブジェクト指向概念の特徴であるモジュール化を引き出すための重要な支援機能である。

1. はじめに

オブジェクト指向に関わる機能や性質については、様々な視点からの検討が行われ、システムや表現言語として開発されている[1,2,3,4]。それらは、互いに異なる視点に立っているが、いくつかの共通の特徴を持っている。言語としては、複数の特徴を持つだけでなく、それを利用するためのプログラミングスタイルを充分支援することが重要である[5]。したがって、これらの共通な特徴を捕らえ、現実のモデル生成の支援方法を検討することが重要である[6]。特に、オブジェクトの概念に基づく設計方法を確立することは、ユーザが表現言語/システムを適切に利用するためにも必要である。

この設計方法の実現に向けて、オブジェクト表現のプログラミング支援環境として、オブジェクトの構成支援機能を論じてきた[7]。これは、オブジェクトの構成を、各機能に関する分析により支援するものである。具体的には、メッセージ処理やデータの制御などに関する構造的情報を利用して、オブジェクトの分割方法を分析し、対象に適したオブジェクト構成や記述の指摘を行う。これは、オブジェクト表現のスタイルを規定する一方法でもある。この支援機能によって、表現システムが持つ関係表現に適したオブジェクト構成が得られる。しかし、オブジェクト指向概念の特徴であるモジュール性を、より強調して引き出すためには、機能的視点から、各オブジェクトが持つ振る舞いや処理方法の同一化を支援し、共通利用や融合等の検討を進めることが重要になる。

本論文では、この同一化に注目した構成支援について、その方法を事例とともに論じる。同一化支援は、階層記述とpart記述の両方に関係する。オブジェクト内部での情報の流れ(方向)や処理とその型に注目し、システムがこれらの情報をもとに、同一性を判定することを同一化支援とする。これは、それらの情報が機能の同等性に結び付くことに対応する。この機能を実現することで、構成支援としての機能が充実し、オブジェクト指向表現を進めるための適切な支援環境が確立できる。

2. オブジェクトが持つ機能の同一化

オブジェクトが持つメッセージ処理機能やデータの制御機能は、それが実現している機能の観点からまとめることができる[7]。そこで、その機能的

まとまり(機能セットと呼ぶ:文献[7]の最小setとメソッドset)を同一化支援の対象として取り上げる。

この同一化支援は、その対象の限定方法からいくつかのレベルを設定できる。本論文では、各機能セットの要素レベルから機能内部の操作情報を利用したレベル、外部の機能との関係を考慮したレベルを設定し、論じる。

2.1 要素レベルの同一化

同一化の対象となる機能セットは、スロットやメソッドをその要素として複数含んでいる。したがって、各機能を実現するための要素であるスロットやメソッドの情報を集め、それらによって同一性の初期の判定ができる。これが要素レベルの同一化支援である。

判定情報は、以下の2つである。

- (1)必要とするスロット数
- (2)機能の処理に関係するメソッド数

スロットは、属性を表現するための枠であり、必要とする属性数の違いは、異なる機能の実現に直接結びつく。したがって、(1)のスロット数の情報は、要素レベルでの同一性判定として重要な視点である。

(2)のメソッド数の違いは、同じ機能でもメッセージに対する処理をどのように記述するかによって起こる。したがって、これらの記述方法を統一できるようなオブティマイズ機構を備え、それに基づいた比較が必要になる。このことから、メソッド数自体の情報は間接的なものである。このレベルではメソッドの引数(arity)も情報として得られる。

本論文では、プログラミング支援環境として、表現システム側で行わせる同一性の初期判定基準のために、要素レベルでは、スロット数の比較を直接利用し、メソッド数については3節で述べるいくつかの規則を与えて利用する。また、各メソッドが持つarityも判定に用いる。

2.2 内部の操作情報を用いた同一化

各機能セット内部の操作情報として得られるものは、文献[7]と同様、メソッド間のメッセージ交換関係とメソッドとスロットとの操作関係である。これらの情報に対して、データの流れに対応させた方向を設定し、有向グラフとして表現する(Fig.1)。
[メッセージ交換]

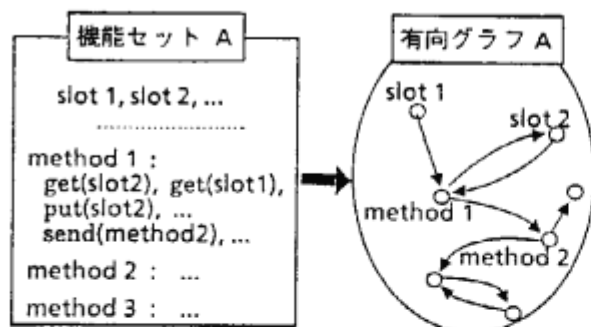


Fig.1 構造的な操作情報の表現

◇ メッセージの送信側から受信側への方向を設定 [スロット操作]

◇ スロットを更新する場合は、メソッドからスロットへの方向を設定

◇ スロットを参照する場合は、スロットからメソッドへの方向を設定

このようにして得られるネットワークの同形を判定することで、構造的観点から同一性を判定し、オブジェクトの構成支援に結びつける。これが操作情報を用いたレベルの同一化支援である。

同形の判定は、各機能セットに対する有向グラフの隣接行列表現を用いて行う。この場合、行列の各行(列)に対応するグラフのノード間の置換群は、メソッドとスロットそれぞれの集合に分割して求める。ただし、この限定だけでは、判定するために必要な置換操作の計算量が膨大になる。このため、この計算量を縮小させるように置換グループの規定条件を設定する。

また、対象とする機能セットが完全に独立でない場合、その情報を用いることで、要素レベルで限定した探索対象を再度広げられる。機能セットが完全に独立でないとは、文献[7]のKEY slots (その機能セットが処理のために参照する機能セット以外のスロット) または KEY methods (その機能セットが処理のために、呼び出す機能セット以外のメソッド) を持つ場合である。

2.3 外部の機能との関係による同一化

オブジェクト概念に基づく場合、モジュール的なプログラムの記述が一つの特徴でもある[8]。したがって、様々なプログラムのユーティリティとして利用される機能は、ある種のモジュールとして記述される。そこでこれらのユーティリティを利用して機能セット間の同一性の判定を行う。各機能セットが、どのユーティリティをどれだけ利用しているか等の情報を取り出すことは、機能セットを分析する一方法にもなる。さらに、機能セット自身が、他の機能セットから利用されていることも情報として取り出す。これらの情報は、機能の同一性を判定するために利用できる。これが、外部との関係情報を用いたレベルの同一化支援である。

本論文で、後に示す同一化支援の事例は、内部の操作情報を用いた同一化までの検討を行っている。これは、その対象例が、コンパクトでアルゴリズムとして明確に定義されていることから、構造情報レベルで十分な情報が得られるためである。

3. 同一化の支援方法

要素レベルの同一化支援に必要な情報は、オブジェクトの機能セットを探索する際に同時に得られる。このレベルでの判定は、基本的なものであるが、対象を限定するためには有効である。互いの機能セットについて、スロット数とメソッド数及びアリティの数を比較することで要素レベルでの同一化支援を行う。ただし、メソッド数に対しては、次の2つの規則が設定される。

規則1: 各機能セットを実現しているメソッドの情報について、引数が0のメソッドは、補助情報と考える。

引数が0のメソッドは、その機能を利用する場合の単なる呼び出しメッセージの受信用として用意されることが多い。この機能が他の機能と直接結びついた形で利用される場合には、この0引数のメソッドを必要とせず、直接その機能を実行する呼び出しが行われる。これらの記述の区別を行わないために、引数が0のメソッドの取扱いには注意が必要である。これは、すべてのレベルに影響する規則である。後の事例ではこの補助情報を用いていない。

規則2: 各機能セットがメソッドに関して独立でない (KEY methodsがある) 場合、KEY methodsにある引数と等しい数の引数のメソッドを対象に加えて探索する。

KEY methodsは、基本的に機能セットに取り込まれるべきメソッドであるが、他の機能との関係から分離されたものを表している。そこで、このメソッドが、取り込まれている機能セットとの区別を行わないために、この規則が必要になる。

スロットに対しても、この規則2に対応した規則が設定できる。

規則3: 各機能セットがスロットに関して独立でない (KEY slotsがある) 場合、KEY slotsに含まれるスロット数も、補足的にスロット数に含めて探索する。

KEY slotsは、KEY methodsと同様、その機能を実現するために必要とするスロットであるため、同一性の判定情報として利用できる。

操作情報を用いたレベルの同一化支援は、有向グラフ(digraph) Dの同形を判定することで実現される。したがって、各機能セットごとにその隣接行列を作成し、その行列の演算と行列間の比較を行うことで同一性を判定する。

mをメソッド数とし、sをスロット数とすると、ノード集合V(D)は、

$$V(D) = \{n_1, n_2, \dots, n_m, n_{m+1}, n_{m+2}, \dots, n_{m+s}\}$$

となる。有向グラフDの隣接行列A=A(D)は、(m+s) × (m+s)の正方行列 $[a_{ij}]$ で、

n_i を始点、 n_j を終点とする弧がN本あるとき、

$$a_{ij} = N$$

n_i を始点、 n_j を終点とする弧がないなら、

$$a_{ij} = 0$$

この隣接行列Aは、次の形になる。

$$A = \begin{bmatrix} A_{mm} & A_{ms} \\ A_{sm} & 0 \end{bmatrix}$$

同形を判定するためには、行列内での置換を行わ

なければならない。この場合の計算量を減らすために置換できる行(列)のグループを設定する。これが2.2で述べた限定であり、基本的置換群はメソッドとスロットの集合上で求める。メソッドに関しては、様々な限定情報が得られ、置換グループが容易に見つかる。そこで、このグループを作成するための限定情報についての同形から探索し、部分行列 A_{mm} の同形を経て、隣接行列Aの同形の探索を行うものとする。

メソッド集合の中で、置換グループを作成するために以下の情報(5項目)を用いる。

- (1) メッセージの引数の数
- (2) 他とのメッセージ交換回数
- (3) 更新するスロットの数
- (4) メソッドを実現している項の数
- (5) 参照するスロットの数

これらの頻度がすべて同じものが置換可能なグループになる。

この5項目の情報を行列表現(頻度行列:F)し、この頻度行列 $F[5 \times m]$ での同形の探索によって、要素レベルの判定と一部の操作情報を用いた判定を同時に行う。これは、機能の同一性を判断するための前提条件でもある。頻度行列Fは、その要素を(1)から(5)の順にソートして作成することで、一意に決定される。この頻度行列レベルでの同形の判定は、各機能の形式上の同一性を調べていることに対応する。同形判定の際には規則1,2を適用する。

次に、部分行列 A_{mm} の同形を判定する。これに必要な置換は、頻度行列Fの要素の比較に従って行われる。置換グループが規定されているため、置換のための計算量は大幅に削減される。

この後に隣接行列Aの同形を判定することになるが、部分行列 A_{ms} や A_{sm} における基本的情報は、前述の頻度行列Fの項目(3)と(5)に含まれている。さらに、機能を実現するための構造的なメッセージ交換情報についての同一性は、すでに頻度行列Fと部分行列 A_{mm} の同形の判定で処理されている。したがって、頻度行列Fの同形と、隣接行列Aの部分行列 A_{mm} についての同形を検索することで、機能セット間の同一性を判定できる。

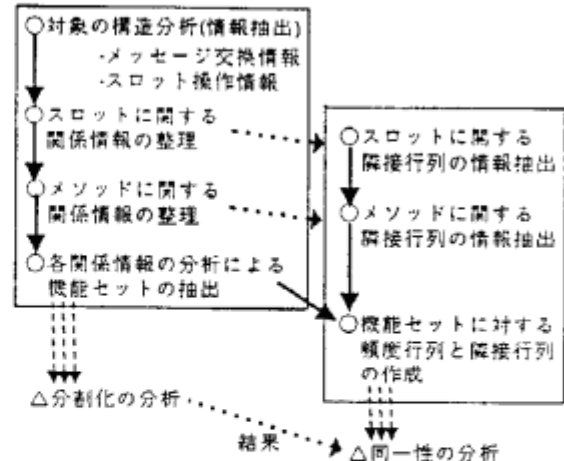
4. 同一化機能の実現とその事例

本論文で提案する同一化による構成支援は、既に提案したオブジェクト構成支援機能と統合して、オブジェクト表現のためのプログラミング環境を実現するものである。これらの支援環境は、ある表現システムに対して限定されるものではなく、オブジェクト指向概念に基づく言語/システム全般に適用できる。

オブジェクト構成支援機能[7]が、すでにQ-Prolog上のオブジェクト指向システムKORE/KR[9]に対して実装されていることから、本論文では、同一化構成支援機能をそれに組み込む形で実現し、その事例を示す。

オブジェクト構成支援機能が、Fig.2の(a)の流れで実現されている。この機能で得られる情報を、直接同一化構成支援でも用い、同時に処理可能な分析

を進める。したがって、(a)の流れに付加する形で(b)のように実現される。



(a)これまでの構成支援 (b)同一化構成支援

Fig.2 構成支援機能の流れ

同一化支援の事例として、すでにKORE/KR上に表現されている多目的最適化アルゴリズムを取り上げる。この事例では、シンプレックス法を実現しているオブジェクトの一部の機能と、別のアルゴリズムを実現しているオブジェクトの機能との同一性が判定される。ここで同一性を判定された機能は、併合挿入法を実現している部分である。

シンプレックス法に対するオブジェクトは、構成支援機能によって3つのまとまった機能セットから構成されていることがわかるが、その中から、同一化支援の対象となる一つの機能セットを取り出し、その情報と頻度行列F1をFig.3に示す。さら

```

THE STRUCTURE-DATA OF decision CLASS ***
*****
***** CLASS STRUCTURE INFORMATION *****
*****
-----
***** main
    KEY slots
        table value
        *****
    **SLOTS**
    insertion
    **METHODS**
    merge/0 insertion/2 div/3
    io/4 insertion1/4 mapsend/5

    [0,2,3,4,4,5]
    [1,5,1,0,2,1]
    [1,0,0,0,0,0] : 頻度行列 F1
    [1,3,2,2,3,2]
    [1,1,0,0,0,0]
-----
*****
    
```

ある機能を表すまとまり

頻度行列の各行が、3節の(1)-(5)に対応

Fig.3 シンプレックス法の一機能の頻度行列

に、これと同一性を判定する機能セットの情報と頻度行列F2をFig.4に示す。

頻度行列F1とF2の行列の各行は、3節で述べた頻度(1)~(5)をそれぞれ表し、各列が個々のメソッドに対応している。

```

THE STRUCTURE-DATA OF merge CLASS ***
*****
CLASS STRUCTURE INFORMATION *****
*****
main
KEY slots
table value
*****
KEY methods
io/4
*****
**SLOTS**
insertion
**METHODS**
insertion/0 insertion/2
divide/3 insert/4 mapsend/5

[0,2,3,4,5]
[1,5,1,2,1]; 頻度行列 F 2
[1,0,0,0,0]
[1,3,2,3,2]
[1,1,0,0,0]

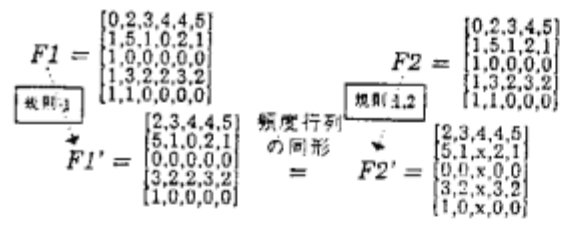
```

ある機能を
表すまとめり

Fig.4 併合挿入法の機能の頻度行列

頻度行列F1は、3節の規則1から、メッセージの引数が0である列は対象外(第一行の要素が0である列を除く)として同形の頻度行列を探索する。

Fig.4の機能セットは、KEY methodsを持つため、3節の規則2が適用され、行列F2の第一行が4である列を加えた行列が探索対象の頻度行列F2'になる。規則2で、探索のために頻度行列に加えらるる列は、ワイルドカードとして処理される。



探索の結果、F1の部分行列F1'と同形である頻度行列F2'が得られた。

このように3節の規則に基づき、同形の頻度行列を持つような機能セットが探索される。そこで、それらの中での隣接行列A_{mm}の同形を判定する。

$$\begin{matrix} F1' \\ \downarrow \\ A_{mm} 1 \end{matrix} = \begin{bmatrix} 1,0,0,0,0 \\ 1,1,0,0,0 \\ 1,0,0,0,0 \\ 1,0,0,2,0 \\ 1,0,0,0,1 \end{bmatrix} = \begin{matrix} F2' \\ \downarrow \\ A_{mm} 2 \end{matrix} = \begin{bmatrix} 1,0,x,0,0 \\ 1,1,x,0,0 \\ x,x,x,x,x \\ 1,0,x,2,0 \\ 1,0,x,0,1 \end{bmatrix}$$

隣接行列の同形

頻度行列の探索の際に列を加えたため、隣接行列A_{mm}2には、その追加列の処理に対応してワイルドカードが追加されている。この事例では、アルゴリズムがあらかじめ明確に設定されている多目的最適化アルゴリズムを取り上げた。このため隣接行列の細部にわたって同形である情報が得られている。

本事例では、Fig.3とFig.4の機能セットの同一性が判定され、その情報からユーザは、この部分を互いに共通利用できることがわかる。この場合、その機能セットが、スロットを持つことから、この機能セットだけを抜き出して新たなオブジェクトを

作成し、PART-OF関係によって共通利用するようなオブジェクト構成を導ける。

5. まとめ

本方法では、構造情報により複数の機能セット間の同一性が判定でき、まとめた記述や構成にすべきことを示せる。4節の事例では、同一と判定された機能セットについてPART-OF関係を用いたオブジェクト構成が実現できることを示した。この構成支援は、階層記述とpart記述の基本的な識別[7]に従い、同一性を判定された機能セットがスロットを持つ場合は、PART-OF関係とし、その他は、階層構造を用いた構成としている。

さらに、同一性を判定するために行列の同形を用いているが、一般には、隣接行列まで完全に同形になる機能を見つけることは困難である。このために本論文では、スロットやメソッドに関する規則を設定し、行列要素の追加等により同形の探索範囲を広げている。これらは、類似機能を見つける一方法である。

特に、頻度行列での同形によって、構造的に似たような機能の検索を行い、このレベルで情報が提示されることは、開発者であるユーザにとって有益である。この他にも、規則を改良・付加し、ユーザとのインタラクションを増やすことで、部分機能の同一性の判定が可能であるが、これらには、判定レベルの問題やユーザへの負担、行列処理など、様々な問題が残されている。

規則2や3の処理方法により、同一性の判定対象が変化するが、本論文で示した同一化支援の各レベルで得られる情報は、オブジェクト表現のモジュール性を高める上で重要な手がかりとなる。全体としての支援機能は、さまざまな表現を開発していくための環境として有用なものである。

今後は、オブジェクト表現のための支援環境として、これまでの機能を踏まえて、整理し、情報の指摘レベルにとどまらず、オブジェクト構成をシステムで処理できるレベルを検討する。

尚、本研究は、第五世代コンピュータプロジェクトの一環として行われたものである。

【参考文献】

- [1] T. Chikayama, "Unique Features of ESP", Conf. FGCS 1984
- [2] F. Mizoguchi, H. Ohwada and Y. Katayama, "LOOKS: Knowledge Representation System for Designing Expert Systems in a Logic Programming Framework", Conf. FGCS 1984
- [3] M. Tokoro and Y. Ishikawa, "A Concurrent Object-Oriented Knowledge Representation Language Orient84/K: It's Features and Implementation, Proc. ACM OOPSLA Conference, 1986
- [4] A. Yonezawa, J. Briot and E. Shibayama, "Object-Oriented Concurrent Programming in ABCL/1, Proc. ACM OOPSLA Conference, 1986
- [5] B. Stroustrup, "What is Object-Oriented Programming?", IEEE Software, May 1988
- [6] B. Grady, "Object-Oriented Development", IEEE Trans. Softw. Eng., vol. SE-12, Feb. 1986
- [7] 片山, "オブジェクト表現開発のためのクラス構成支援について", 情報処理学会, 知識工学と人工知能研究会, 50-8, 1987
- [8] T. Rentsch, "Object Oriented Programming", SIGPLAN Notices, vol. 17, no. 9, Sept. 1982
- [9] 片山, "問題解決支援環境KORE(その3)-知識表現サブシステムKORE/KRとその概要", 第32回情報処理全国大会, 1986