ICOT Technical Memorandum: TM-0288

TM-0288

オブジェクト表現構築のための クラス構成支援

片山佳則
(富士通)

March, 1987

ICOT
Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan
(03) 456-3191~5
Telex ICOT J32964

Institute for New Generation Computer Technology

オブジェクト表現構築のための クラス構成支援

富士通㈱　国際情報社会科学研究所　片山　佳則

1．はじめに

オブジェクト指向概念を導入した表現システム／言語は，Smalltalk[1] やFlavors[2) 3)]，Loops[4] などに始まり，様々なものが作成され検討されている[5)-10)]．これらの表現システム／言語は，プログラムのモジュラリティ，プログラム開発・拡張の容易性など様々な共通的特徴を持っている．しかし，実際にこれらの特徴を十分に引き出せるようなオブジェクト構成や内部の記述を行うことは難しい．一般には，その技法に慣れた『エキスパート』が行う（又は間に入る）ことで，簡単に処理されていると考えられている．しかし，エキスパートであってもオブジェクトを適切に構成し，開発することは容易ではない．特に新しい分野の知識を表現する場合，その知識に対して最適なオブジェクト構成を決めることは困難である．これらの問題を解決するために，オブジェクト表現構築のための支援機能が必要である．

表現したいプログラムのアルゴリズムやオブジェクト構成が明確である場合，オブジェクト概念を導入した表現システム上に，それらをどのような手順で開発していくべきかについては，設計に基づく段階的開発支援[11)] として既に検討した．この段階的支援の検討においても，やはりオブジェクト自身の表現や関係構造を適切に支援する機能が重要であることが確認された．

表現システムがオブジェクトの適切な構成を導く支援機能を持つことによって，エキスパートの必要性も無くなり，開発者は表現する内容だけを深く検討できるようになる．さらに，各オブジェクト表現のための属性の抽出支援（スロット化支援）も行うことでオブジェクトの機能をより明確にすることができる．このような支援機能を実現することが，その表現システムを広範囲に利用させることにつながる．これらの様々な支援機能を実現することによって，開発が容易になり，オブジェクト構成のための関係表現は，システムが持つ概念ですべて統一して実現され，その意味からもオブジェクトの関係が明確になる．

本稿では，これらの構成支援・開発支援の機能の実現方法を提案し，その支援の事例を取り上げ，具体的な機能・有効性をまとめる．オブジェクト間の構成支援として，次節では，表現システムが持つオブジェクト間の関係表現をまとめ，実際に構築支援の対象としている関係や各関係表現の意味を明確にする．第３節では，具体的なオブジェクト分割の考え方や主な分割方法をまとめる．第４節では，オブジェクト内部の構成（記述）支援として，オブジェクトの内部状態を表現するスロットの導出支援について述べる．第５節において，第３節で示した分割の事例や第４節での支援の事例を示す．

2．関係表現

本稿で述べる支援機能は，オブジェクト表現システム全般に適用できるものであるが，ここでは，表現システムとしてKORE/KR[1,2]を取り上げる．KORE/KR はPrologをベースに作成されたオブジェクト表現システムであり，拡張を繰り返して現在の表現システムに至っている．

KORE/KR が持っているオブジェクト間の基本的関係は，次の３つである．

(1)superset関係：クラスの包含関係を表す．

(2)subparts関係：オブジェクトの機能・構造的関係を表す．

(3)instance-of関係：表現されたクラスを利用するための実体（インスタンス）との関係を表す．

この他に，(a)meta-object関係，(b)manager機能を持っている．(a)は，(1)～(3)のようにユーザに対して明示的な関係ではなく，表現システム内部の組み込み機能(Meta-object)との関係である．したがって，ユーザがその機能や関係を自由に定義できない．ただし，このMeta-object が持つ機能を補うための機能付加[3] (Flavorsなどにあるafter daemons的なもの) は可能である．(b)は，(1)～(3)に対応した関係として表す形を取っていないが，必要なオブジェクトの実行を記録・管理するための機能を実現するものである．KORE/KR は組み込みオブジェクトとしてManager オブジェクトを持っており，これを利用することにより各オブジェクトの記録・管理を行う．

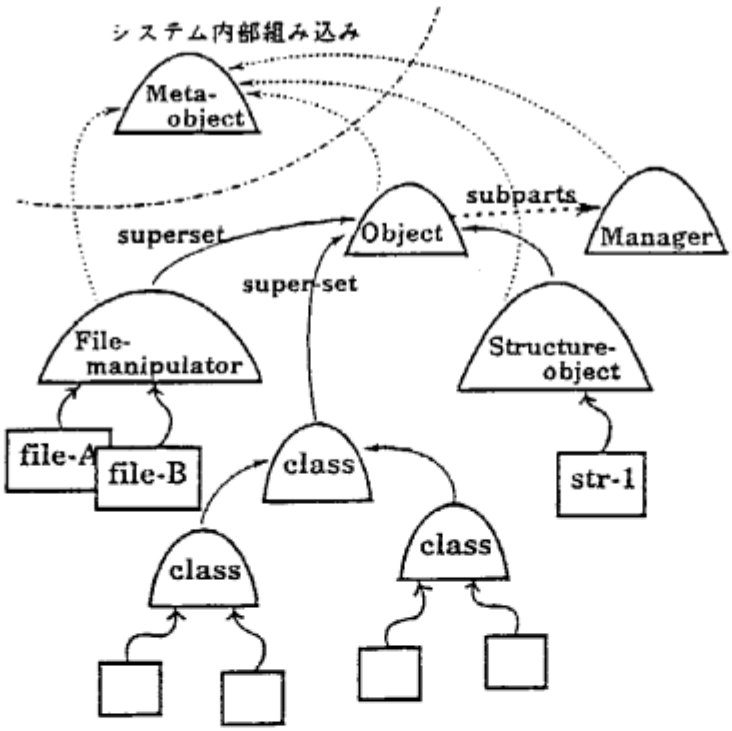


Fig. 1　組み込みオブジェクトの構成

KORE/KR はこの他にインスタンスに関する基本的な機能のためのObjectオブジェクトや，本稿の支援機能を実現するためのStructure-object，ファイル入力用のFile-manipulator

などを組み込みオブジェクトとして持っている．これら複数の組み込みオブジェクトを利用し，開発・実行する(Fig.1)．

ObjectとManager はFig.1にある通りsubparts関係で表現されている．したがって，ユーザ定義のオブジェクトがObjectを継承することで，Manager の機能を利用したいときにManagerのインスタンスをそのオブジェクトのpartとして実現してその機能を利用する．

Manager機能をsubpartsで表していることから，クラスの継承表現を用いてユーザが他のManager オブジェクトを作成することも可能である．組み込みのManager が持つ記録・管理の機能は，各オブジェクトについてのメッセージの送信・受信状況の情報や，スロットへのアクセス情報，スロットの状態変化の履歴などである．

いくつかの基本的な関係を挙げたが，この中で表現のために必要な関係は，(1)と(2)の関係である．その他は，プログラムの実行や管理のときに必要となるものである．したがって，本稿でのオブジェクト関係の支援機能では，この二つの関係を重視し，検討する．

＜superset関係＞　IS-A関係に対応し，概念の包含関係を表し，集合間の関係として部分集合関係(Subset/Superset) を表す．述語間の関係としては，一般化と条件づきによる特殊化(Generalization/Specialization) を表現する．種類についての関係として，AKO(A Kind Of)関係とも呼ばれる．さらに細かい分析[14] も可能である．この関係では，属性・機能の継承が行われる．

＜subparts関係＞　PART-OF 関係に対応し，部分集合としての関係ではなく，機能・内部構造に関して部分関係を表すものである．基本的には，構成要素としての関係が強い[15] この関係では，属性の継承は行われない．この関係は，概念（クラス）間に結ばれたものが各インスタンスに対する関係としてもそのまま受け継がれ，インスタンス間の結びつきをも表す(Fig.2)．

これらの関係とメッセージ交換との係わりは，次のように考えられる．

あるオブジェクトが処理を分担させるために，他のオブジェクトにメッセージを送る場合，メッセージを送る側から見れば，情報を受け渡して計算を進めることである．クラスレベルでこれを表現する場合には，あらかじめ対象オブジェクトが決まっている場合を除いて，それらのメッセージ交換はsuperset関係による継承をメッセージとして動的に探索するか，subparts関係により内部構造の一部として表されているオブジェクトに対して行われるかのどちらかになる．この後者の考え方は，Actor 理論におけるacquaintances[16] という通信情報の局所性に対応する．メッセージを機能分担の処理と考えた場合，オブジェクトが他のオブジェクトと直接メッセージ交換をする場合は，そのオブジェクトとsuperset 関係かsubparts関係のどちらかを必ず持たなければならない．

これらの基本的な二つの関係を表現システム上で適切に構築するための支援が，オブジェクト構成支援の一つの目的である．この構築には，関係表現に合ったクラスの分割を規定する方法が重要である．

superset関係におけるクラスの分割方法は，部分集合関係では，属性の継承を有効に利用できることに視点を置き，記述量と適切な記述位置を定め，他方，一般化のためのクラス分割では，記述対象の意味などの複雑な要因を考慮する必要がある．

subparts関係のクラス分割方法は，継承関係を持たないため，機能・内部構造としての部分関係を的確に表すための部分規則を設定する必要がある．この部分関係では，各部分はその部分特有の属性（性質・特徴）と機能を持たなければならない．オブジェクト表現においてこのような特有の属性と機能を表現するためのオブジェクトは，属性をスロットで表し，そのスロットの入出力だけでなくオブジェクトの機能としての操作も含めて，メソッドを持つ必要がある．さらに部分関係として表現することから，部分に対応するオブジェクトは全体としてのオブジェクトから利用される形（部品としての関係）になっている必要がある．subparts関係を中心に見たオブジェクト構成は，Fig.2のようになる．

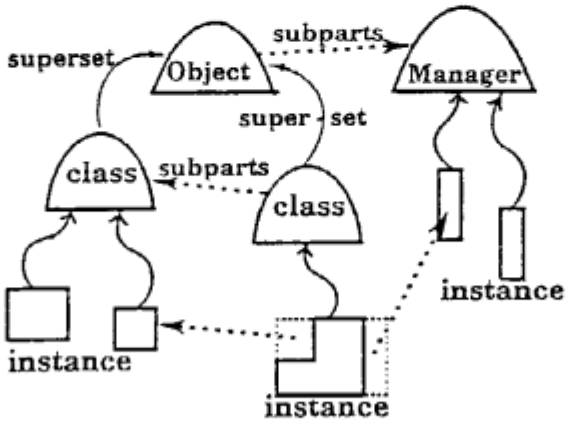


Fig.2　オブジェクトのsubparts関係

オブジェクトの分割方法に加えて，オブジェクトの構成支援には，全体構成の中での各オブジェクトの同一性の検討も必要となる．この同一性に関しては，個々のオブジェクトの機能が明確にされ，他のオブジェクトとの関係が明らかであれば，それらの関係を用いることで同一性検討処理も容易になると考えられる．

3．オブジェクト構成のための分割

本節では，関係表現に合ったオブジェクトの分割と，その構成を与えるための基本的考え方を提示し，それに従った分割方法を述べる．また，オブジェクトから得られる情報の分類を行い，それらがこの分割においてどのように利用されるかをまとめる．

3.1　分割の基本的考え方

はじめに，シンタックス情報に基づく機能・構造としての部分関係による分割を目的とし，subparts関係に注目する．subparts関係以外で残ったものに対して，それまでの各情報をもとにしてsuperset関係の使用を検討する．

オブジェクトを分割するには，オブジェクト内のスロットやメソッドの関係状態から独立性の高いいくつかのセットを見つけ出す必要がある．複数のセットが見つけられない場合は，現時点でそのオブジェクトを分割することは不可能である．さらに，subparts関係としてpartとなることを前提にしてセットを見つけることから，2節で述べたように，各セットはスロット（属性）を含んでいなければならない．したがってこれらのセットは，オブジェクト内のスロットを中心にし，それと結合すべきスロットやメソッドを取り込む形式で作成できる．スロット，メソッドの関係から，取り込むための結合規則は次の二つにまとめることができる．

(a)　スロットの更新(put)操作を持つメソッドは，そのスロットと結合する．

(b)　メッセージ交換関係を有向グラフに対応させた場合，グラフとして強連結(strongly connected)[7]となるメソッドはすべて結合する．

この結合規則だけで作成されるセットは，オブジェクト機能の最小単位表現であるので，最小セットと呼び，これらの最小セットがpartとなるかどうかによって，subparts関係のためのクラス分割を支援する．

あるセットがpartとなるかどうかを分析するために，全体とpartが，スロット・メソッドを通じて，どのような関係にあるべきかを明確にさせなければならない．これに従ってpartとなるセットを見つけ出す．

＜partと全体の関係＞

(1)　全体側からpartが持つスロットやメソッドに対する操作の関係は次の通り．

○　メソッドは自由に呼びだせる．

○　スロットを更新する際には，表現システムが用意している特別なPREDICATE "putvalue"で直接更新することはできない(メッセージ交換による更新のみ)．

○　スロットの参照は自由にできる．

基本的に，全体側からpart側の操作に対する制約は，スロットの更新についてのみで，その他に関しての制約はない．これらは，partが部分機能であること，及び更新操作は各オブジェクト内で閉じた内部の働きであることから明らかである．

(2)　part側から全体が持つスロットやメソッドに対する操作の関係は次の通り．

○　メソッドは呼び出せない．

partは全体側から利用される形であり，その逆にはなれない．［メッセージ交換につい

ては，返答のためだけの場合を考えられるが，ここでは，情報伝達のためだけにメッセージ交換を行うことを考慮していない]

○　スロットの更新・参照はできない．

部分機能がその処理において全体の持つスロット（属性）に関する操作を必要とする場合，それは部分機能であるpartとして閉じていないことになる．

part側から全体に対しては，すべての操作が制約を受ける．これは部分側が主になって全体に対して影響を与えるような操作を行わないことが基本であるからである．

最小セットがpartとなるかどうかは，各最小セットについてここで示したpartと全体の関係を分析することで判断できる．さらに，複雑なオブジェクトを分割したい場合には，最小セット(結合規則(a)(b)だけでの取り込みによるセットの作成)だけでは不十分である．そこで複雑なオブジェクトに対しても適用できるように，分割の考え方をさらに発展させる．

最小セットは，スロット中心に考えていることから，最小セットに含まれないメソッドが多数存在する．これらのメソッドだけでもさらに新たなセット（メソッドセット）を構成できる．このメソッドセットは，スロットを持たないためこれだけではpartの対象にならないが，最小セットの拡張や，superset関係構成の支援，オブジェクト自身の働きを理解する上でも役立つ．

メソッドセット：最小セットに含まれないメソッド群のうち，呼び出し関係のあるメソッドのセット．

このメソッドセットと最小セットにより，再度オブジェクト内のセットの構成を検討する．このレベルで分割単位となるセット（基本セット）の作成は，最小セット作成の際の対象であったスロットやメソッドを，それぞれ最小セットとメソッドセットとして行うことに対応する．結合のための規則を次に示す．

(c)　強連結な関係を意識せず，最小セットと関係のあるメソッドセットを結合する．

規則(a)に対応して，スロットの参照関係から，複数の最小セットを結合することも可能であるが，最小セット間の結合は，基本単位の機能を結び付けることになる．この参照関係は，オブジェクト間のメッセージ交換の際の引数を変更することでpartとして分割することができる．したがって，ここではユーザに対する支援機能として考え，この参照関係による結合を規則としない．

規則(c)で作成された新しい基本セットについて，partとなるかどうかを分析することで，より複雑なオブジェクトのsubparts関係構成支援が行える．part以外の情報は，superset関係のための支援情報となる．

3.2　セットの分割方法に関連する関係情報

結合規則(a)(b)によって分離される時点で，3.1節で示したpartと全体の関係に基づき，各セットがオブジェクトに対してpartとなれるかどうかの判定を行うことができる．

＜スロットとメソッド間の分離＞

①　メソッドが参照「get」しているだけのスロットとの関係を分離した場合：

［スロット側はpartとなり得る］

「partとなり得る」という表現は，そのセットが他との関係でpartになり得ない条件がなければ「partにする」ことを意味する(以下同様)．

＜メソッド間の分離＞

② メソッド間の単に呼び出される関係だけを分離した場合：

a)［呼び出されるメソッド側はpartとなり得る］

b)［呼び出すメソッド側はpartとなり得ない］

次に，オブジェクトから取り出す情報を分類し，それらがどのように利用されるかをまとめる．

［A スロットとメソッドの関係情報］

＜スロット側＞

スロットが受ける操作の内容が，更新『put』を含んでいる場合のメソッドは分離できない(結合規則(a))．

操作内容が参照『get』のみであるメソッドはすべて分離する(分離条件①)．

<関係表現との対応> ：分離にかかわらず，すべてpartとなり得る．

＜メソッド側＞

メソッドが行う操作の内容が，参照『get』のみの場合は分離し，更新『put』を含んでいる場合は分離できない．

<関係表現との対応>:分離した場合はpartになれない，分離しない場合はpartとなり得る．

［B メソッド間のメッセージ交換情報(オブジェクト内部)］

メッセージ交換関係が他のメソッドからメッセージを受け取る『receive』だけか，他のメソッドへメッセージを送る『send』だけの場合は分離できる (分離条件②)

メッセージ交換関係として，sendとreceive の両方ある場合は分離できない (結合規則(b))．

<関係表現との対応> ：メッセージ交換関係が，sendの場合の分離を行った場合は，part にはなり得ない(分離条件②b)，それ以外はpartになり得る．

A，Bの情報を用い，前節の規則・条件を当てはめることで，クラス構成のための適切な分割を行うことができる．

4．スロット化支援

3.1 の分割の基本的考えに基づいた場合，subparts関係としては各オブジェクトが属性としてのスロットをどれだけ持っているかがキーとなる．そこで本節では，このキーとなるスロットの導出についての支援を考える．各オブジェクトの属性に直接関係するのはメソッドが持っている引数である．これらの中で，スロット化可能なものの指摘を行うことでスロット化の検討を促す．

メソッドの引数には，メッセージの受信のためのパターンとして用いるものや，ある操作だけのために常に引数として持っていなければならないもの，ひんぱんにアクセスするものなど様々な状況が考えられる．各状況を分析し，これらの中でスロット化可能なものを見つけ出す．スロットは，そのオブジェクトの持つ性質を表現し，その値によってオブジェクトを区別するものであったり，オブジェクトの機能実現に欠かせないものである．メソッドとその引数との関係をオブジェクトとスロットの関係に完全に対応させることはできないが，同様な関係で使用している場合がある．これらの引数には，メソッドの引数としておくのではなく，スロットとして表すべきものが含まれている．このような引数を見つけ出すことがスロット化支援である．

取り出せる情報で効果的なものは，各メソッドが持つ引数のタイプである．同じメソッドでも様々な記述があり，それらすべての引数情報を取り出してタイプを比較することによりスロット化支援を行う．

＜スロット化のための情報＞

各メソッド，及びそのメソッド呼び出しのすべての引数について，引数タイプの情報を取り出す．同じSelectorのメソッドを仮にMethod Groupと呼ぶと，Fig. 3 に示す引数タイプ検索を行って，各メソッドについて次の情報を取り出すことである．

(1) メソッド記述における引数タイプ (Fig. 3 の(1))

(2) メソッド呼び出しの時の引数タイプ (Fig. 3 の(2))

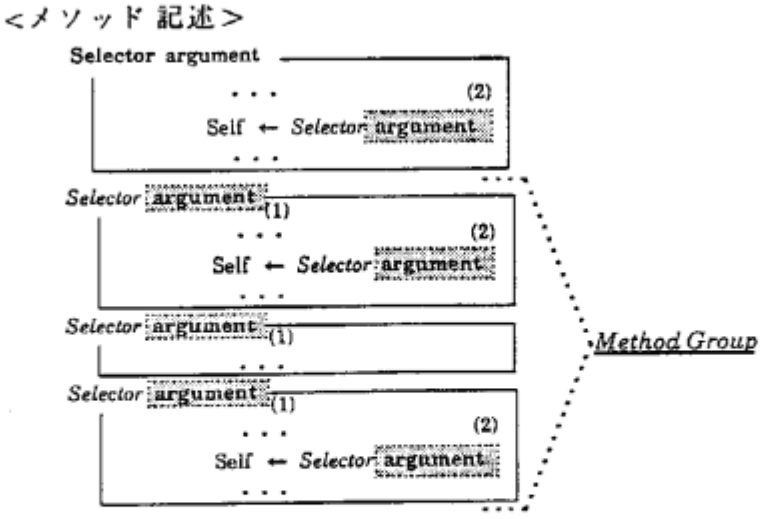


Fig. 3　引数タイプ検索方法

この二つの情報を合わせて，差分リストや単なるlistかどうか，atomやinteger であるかを判断しながらそれぞれの引数タイプの分類を行う．atomやinteger,差分リストを除き，変数やlistのいずれの場合でも引数タイプが変化しないものをスロット化可能な引数とする．各メソッドにおいて，スロット化可能な引数を含んでいるメソッドだけを取り上げ，どの引数がスロット化可能であるかを示すことでスロット導出の支援を行う．

5．支援機能の事例

本節では，これまでに述べたクラス構築支援やスロットに関する支援，オブジェクト開発の支援が実際にどのようにおこなわれるかについて事例を上げて説明する．ここで示すオブジェクトは，すべて表現システムKORE/KR 上で実現したものである．

＜クラス構成支援＞

対象オブジェクトを構成支援システム自身のクラス'structure' とする．支援システムが対象オブジェクトのシンタックスを分析し，3.2 節で示した情報を取り出す(Fig.4)．これらの情報から，3.1 節の規則・条件によってクラス構成の情報A(Fig.5) が導かれる．Fig.5-a が最小セット(SLOTSとMETHODSの対)であり，Fig.5-b がメソッドセットである．これらの情報Aでは，'structure' クラスが三つの最小セットといくつかのメソッドセットから成立っていることを示している(Fig.6)．

オブジェクトによっては，このレベルで新しい構成についての支援を受けられる場合がある．しかし，この場合，各最小セットにはKEY slotsとKEY methods があるため，別のオブジェクトとして分離できない．このKEY の意味は，KEY slots:そのセットが内部処理を行うために，セット以外のスロットの参照を必要としている．KEY methods:そのセットが内部処理を行うために，セット以外のメソッドを呼びだす必要がある．これらのKEY について基本セットによる支援やメッセージ交換の際の引数を再検討することで，より適切なクラス構成が得られる．Fig．5の情報Aから基本セットを求めることで，Fig．7の情報Bの結果を得る．情報Bでは，3.1 節の規則(c)によりメソッドセットを最小セットに結合したためKEY methods がほとんど無くなっている(Fig.8)．KEY slotsについては，その関係が少ない場合はメッセージ交換時のargumentとして処理する形に変更することで，新たなオブジェクトとしての構成を導くことができる．ただしこの場合の変更は，オブジェクト自身の意味などに関わるため，ユーザの意思決定が必要となる．この変更を行った場合，Fig．9の関係図にあるような形で，基本セットSET Bをpartとして分離することができる．最終的には，構成支援システムによって，クラス'structure'がFig.10 a) からFig.10 b) の構成になる．

このクラス構成支援によって，関係構成を支援するだけでなく，この支援過程におけるメソッドセットの情報を用いて，そのメソッドが必要であるかどうかを検討することもできる．このメソッドセット情報によって，あるメソッドがオブジェクト内部では全く利用されていない(SENDもRECEIVEもない)ことが分かり，そのオブジェクトの機能を再検討することにより，不必要かどうかの判定ができる．例えば，Fig.5-b のINFORMATION ABOUT OTHER METHODS にあるrename1/6 など．一度作成した後では見つけにくい孤立メソッドも

このように的確に指摘してくれる．

また，開発と同時にこの支援システムを用いることで，対象オブジェクト内での変更（特にメソッドの引数の変更）に対する影響も把握できる．

```
  THE STRUCTURE OF  structure  CLASS ***
### INFORMATION CONCERNING SLOT ###
  |-----ALL OF SLOT_NAME-----|
  object m_list slot methods
  class next_part_slot part_slot
          ...
*** INFORMATION[1] ************
    |  slot_name :            |
    |     method - type  ... |
    -------------------------
 next_part_slot :
   part_analysis2/0-get part_analysis1/0-get
   part_analysis1/0-put part_analysis/0-put
 structure :
   set_slot_pattern/0-get rename/0-get
   rename/0-put collect_structure/50-put
  ...
### INFORMATION CONCERNING METHOD ###
     |-----ALL OF METHOD-----|
  print_partdata1/1 print_partdata1/50
  part_ana5/4 part_ana21/8 part_ana2/6
          ...
*** INFORMATION[1] ************
    | * method ********* |
    |    ---- send-      |
    |    - receive-      |
    ----------------------
  **  part_ana4/5  *********
   ---- send-
      part_ana411/3 part_ana412/5
   - receive-
      part_ana3/13
  **  collect_structure/50  *********
   ---- send-
      arrange_all/2
   - receive-
      doit/50
  **  setting_name/50  *********
   ---- send-
      getslot/50 getmethod/50 set_slotname/50
      set_methodname/50 get/2 get_list/2
   - receive-
      null
          ...
*** INFORMATION[2] ************
    |  method :               |
    |    slot_name - type ... |
    --------------------------
  setting_name/50 :
     m_list-put s_list-put
     m_name-put s_name-put
     methods-put slot-put
  collect_structure/50 :
     structure-put class-put
  print_structure/0 :
     method_pattern3-get methods-get
     method_pattern2-get class-get
     slot_pattern-get slot-get
   ...
```

Fig. 4　クラス構成支援のための関係情報

```
      THE STRUCTURE-DATA OF  structure  CLASS ***
      *****   CLASS STRUCTURE INFORMATION  *****
       --------------------------
       *****  main
           KEY slots
             structure class
           **********
           KEY methods
             search_method_p2/4 getslot/50 ...
           **********
SET A      RECEIVE
             doit/50
           **********
          **SLOTS**
          method_pattern1 method_pattern3 ...
          **METHODS**
        search_slot_pattern/5 search_slot_p1/6 ...
       ---------------------------
       *****  main
           KEY slots
             m_list s_list
           **********
           KEY methods
             pack_all/4 arrange_sublist/5 rename1/8
           **********
SET B      RECEIVE
             doit/50
           **********
          **SLOTS**
          structure class
          **METHODS**
        arrange_all/2 rename/0 collect_structure/50
       ---------------------------
       *****  main
           KEY slots
             method_pattern3 methods ...
           **********
           KEY methods
             delete_type/2 part_anal1/2 ...
           **********
SET C      RECEIVE
             doit/50
           **********
          **SLOTS**
          main_set part_method part_rec_method ...
          **METHODS**
        part_ana2/6 part_anal/4 part_ana21/8 ...
       ---------------------------
      **********  end of data   **********
```

Fig. 5-a クラス構成のための支援情報A

```
**** INFORMATION ABOUT OTHER METHODS ****
 ----------------------------
    **METHOD SET**
  arrange_sublist/5
     RECEIVE
       arrange_all/2
 ----------------------------
    **METHOD SET**
  rename1/6
 ----------------------------
    **METHOD SET**
  print_structure/0 tab_write/2
     GETVALUE
       method_pattern3 class methods
       slot_pattern slot method_pattern2
 ----------------------------
    **METHOD SET**
  print_sublist/50 print_sublist/1
 ----------------------------
    **METHOD SET**
  part_analysis3/4 part_analysis31/11 part_ana311/10
     RECEIVE
       part_analysis3/0
 ----------------------------
       ...
 ----------------------------
    **METHOD SET**
  part_ana5/4
     RECEIVE
       part_analysis/0 part_analysis1/0
 ----------------------------
    **METHOD SET**
  print_partdata/0 print_partdata1/50
  print_partdata1/1 delete_system/2
     GETVALUE
       main_set class part_set
 ----------------------------
**********  end of data   **********
```

Fig. 5-b クラス構成のための支援情報A

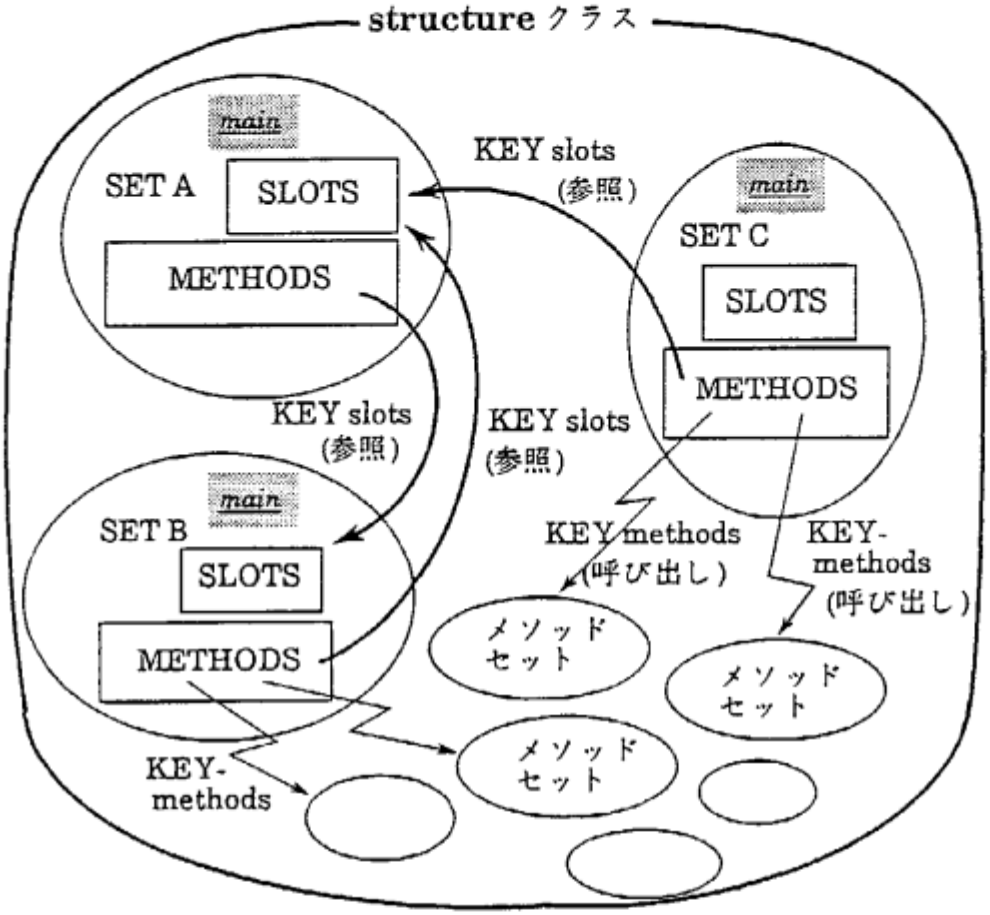


Fig. 6　クラス構成のための支援情報Aの関係図

```
THE STRUCTURE-DATA OF  structure  CLASS ***
*****   CLASS STRUCTURE INFORMATION  *****
 ---------------------------
 *****  main
     KEY slots
       structure class
     **********
     RECEIVE
       doit/50
     **********
    **SLOTS**
    method_patternl method_pattern3 ...
    **METHODS**
 search_slot_p2/4 search_method_p2/4 ...
 ---------------------------
 *****  main
     KEY slots
       m_list s_list
     **********
     RECEIVE
       doit/50
     **********
    **SLOTS**
    structure class
    **METHODS**
 arrange_sublist/5 pack_all/4 rename/0
 collect_structure/50 arrange_all/2
 ---------------------------
 *****  main
     KEY slots
       method_pattern3 methods ...
     **********
     RECEIVE
       doit/50
     **********
    **SLOTS**
    main_set part_method part_rec_method ...
    **METHODS**
 part_analysis3/4 part_ana5/4 part_ana2/6 ...
 ---------------------------
**********  end of data   **********

**** INFORMATION ABOUT OTHER METHODS ****
 ---------------------------
    **METHOD SET**
 renamel/6
 ---------------------------
    **METHOD SET**
 print_structure/0 tab_write/2
    GETVALUE
      method_pattern3 class methods
      slot_pattern slot method_pattern2
 ---------------------------
    **METHOD SET**
 print_sublist/50 print_sublist/1
 ---------------------------
    **METHOD SET**
 print_partdata/0 print_partdata1/50
 print_partdata1/1 delete_system/2
    GETVALUE
      main_set class part_set
 ---------------------------
*********  end of data   *********
```

SET A

SET B

SET C

Fig. 7　クラス構成のための支援情報B

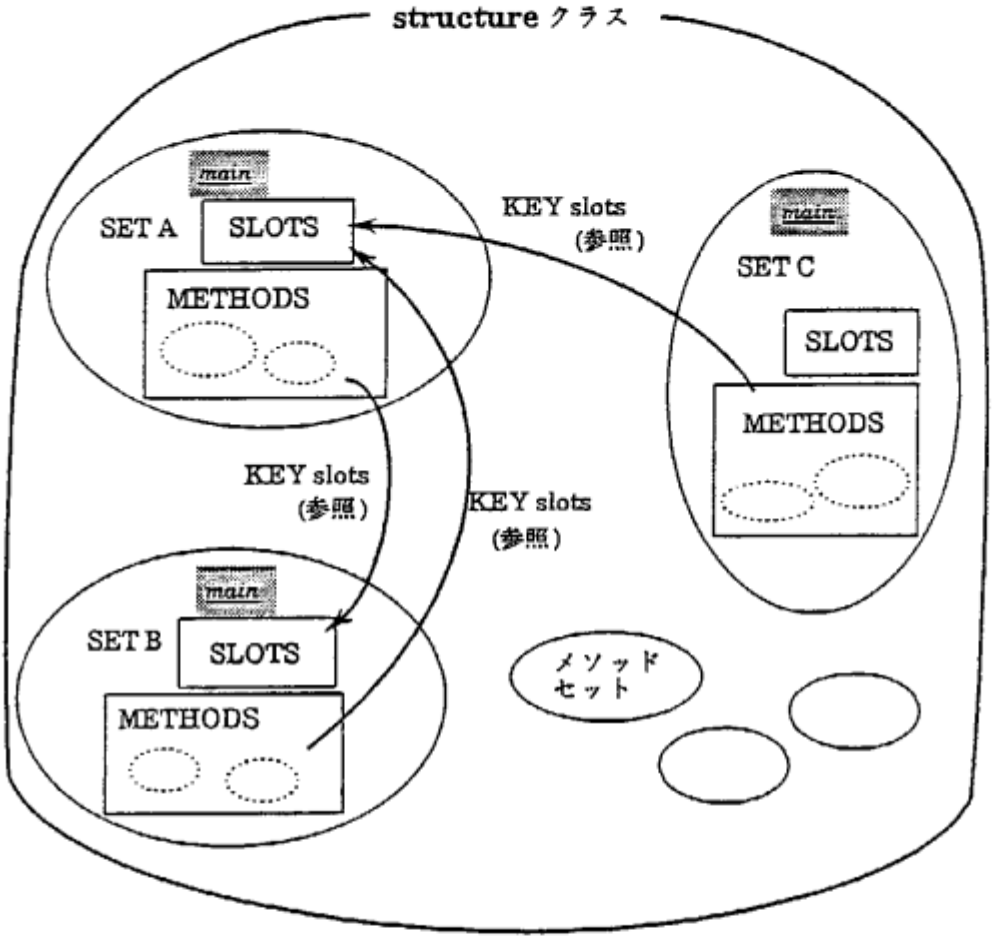


Fig. 8　クラス構成のための支援情報Bの関係図

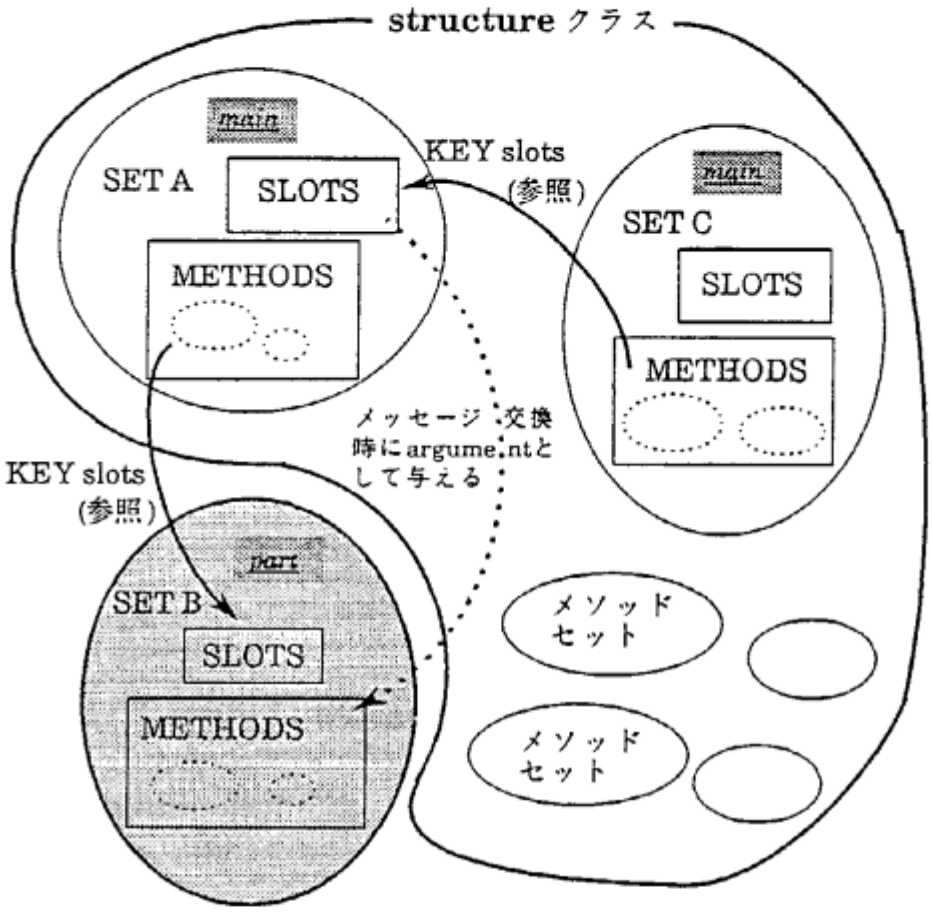
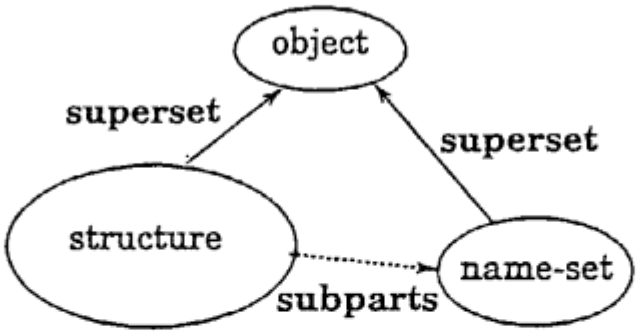
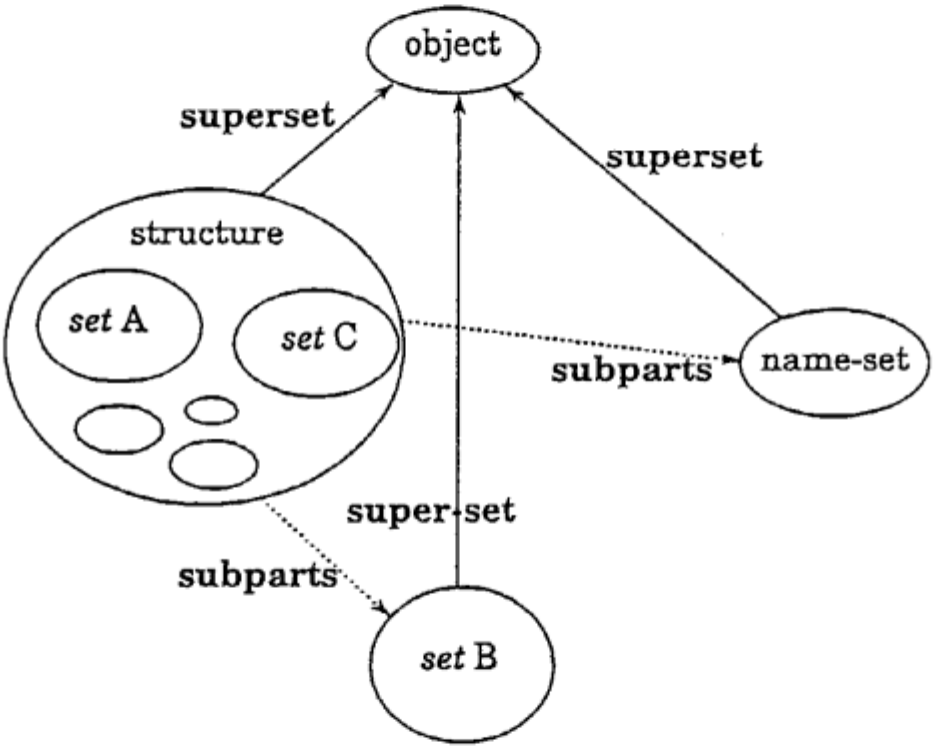


Fig. 9　最終的な支援情報の関係図



(a) 開発時のクラス構成



(b) 支援後のクラス構成

Fig.10 クラス'structure' の構成

＜スロット化支援＞

ここでは，第４節で述べたスロット化支援の情報の事例を示す．各メソッドについて，すべての引数タイプを集める．その際には，スロットと成り得ない差分リストやatom,integer を省きながら情報を取り出す．最終的に，スロット化が可能な引数を含んでいるメソッドだけを取り上げて表示し，どの引数が可能であるかを示す．ここでの対象例も，クラス構成支援の事例であるstructure オブジェクトを用いる．支援情報をFig.11に示す．

```
***Argument Information***
 | METHOD,ARGUMENT-LIST |
 ------------------------
rename1,[0,0,1,1,0,0|non]     arrange_sublist,[0,0,0,0,1]
search_slot_pattern,[0,1,1,0,0]     search_method_pattern,[0,1,1]
search_slot_p1,[0,1,1,1,0,0]     search_method_p1,[0,1,1,1]
search_method_p2,[1,1,0,0]     search_slot_p2,[1,1,0,0]
part_ana311,[0,0,0,1,1,1,0,1,1,1]     pattern_check,[0,1,0,0]
part_ana1,[0,1,0,0]     part_ana3,[0,0,1,0,0,1,1,1,1,1,1,1,1]
part_ana4,[1,1,1,1,1]     part_ana412,[0,1,1,0,0]
parge_list,[1,1]     part_ana5,[0,1,1,1]
part_ana2,[0,1,0,1,1,0]     part_ana21,[0,1,1,0,0,1,1,0]
```

Fig.11　スロットの導出支援情報

Fig.11において，メソッドとペアで表示されている数字のリストが，どの引数がスロット化可能であるかを表している．

1：スロット化可能な引数　　0：スロット化不可能な引数

non：引数が差分リストの形を取っているのでスロット化不可能

この例のStructureオブジェクトでは，Fig.11の引数のうち,search-method-p2やsearch-slot-p2の第2,3引数などがスロット化できることがわかった．

スロット化可能な形式で示されたargumentについて，実際にスロットとして表現するかどうかは，それがオブジェクトの性質を表すものに成り得るかどうかを判定することが必要である．スロット化支援の立場から，現段階での最終的な判断はユーザに任されている．

6. 考察

これまで実例とともに述べた，クラス構成支援システムの動作及び支援機能について考察する．

スロットに関する支援を受け，各オブジェクトについて考えられるすべての属性を検討し，その後に，クラス構成支援を受ける．これによる効果は，以下のようにまとめられる．

○ 作成したオブジェクトの機能や属性を，十分にまとめることができる．

○ オブジェクト内のセットを用いて，subpartsなどの関係表現を適切に導くことができる．

○ オブジェクトの関係表現の意味がシステムで統一して実現される．

これらは，作成されたオブジェクトを利用するためにも重要な働きを担う．

オブジェクト開発と並行して，このクラス構成支援を用いることで，対象の機能や内部構造（スロットを中心にしたメソッドのまとまり方）を正確に把握できることから，実際の開発及びデバッグにも貢献できる．

さらに，設計型のエキスパートシステムなどでも問題となるように，結果の評価は重要な問題である．この支援システムについても最終的なプログラムについての評価が必要となるが，現段階では，支援情報を提供することに止めていることから，それらの評価はすべて開発者に任される．支援システムの目的は，関係表現を適切に用いたオブジェクト開発であるから，実際の評価は，開発されたオブジェクトについて関係表現がどのように用いられているか，それが適切であるかを分析することになる．

現段階のクラス構成支援システムでは，開発段階で受け取り側が明確でない場合のメッセージについての処理が十分ではない．特に，メッセージの引数としてオブジェクト名を受け取り，それに対してメッセージを送る場合を対象外としている．これに加えて今後は，スロットの利用に対して同種の形で用いているものの判断と整理なども検討する必要がある．また，オブジェクトが大規模になると，関係情報に対する検索の計算量についての考慮が必要となる．

7. 結び

オブジェクト指向概念を導入した表現システムのオブジェクト構成問題に対処するための構築支援システムについて述べた．この基本は，オブジェクトが持つスロットやメソッド，及びそれらの利用状況を用いて，関係表現として最適なオブジェクト構成への支援を行うものである．今後は，プログラミングスタイルを確立するための一つの方向を示せれば良いと考えている．

最後に，本研究に関して貴重な議論，助言をして頂いた戸田部長，情報社会学室の新谷，平石研究員に感謝します．

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ったものである．

[参考文献]

1) Goldberg,A.and Robson,D. : Smalltalk-80 The Language and its Implementation, Reading,Massachusetts, Addison-Wesley (1983)

2) Weinreb,D.and Moon,D. : Lisp Machine Manual (1981)

3) Moon,D.: Object-Oriented Programming with Flavors, Proc. ACM OOPSLA Conference, pp.1-8 (1986)

4) Bobrow,G.and Stefik,M. : The LOOPS Manual,Xerox PARC Knowledge-based VLSI Design Group memo KB-VLSI-81-13 (1983)

5) Chikayama,T. : Unique Features of ESP, Proc. International Conference on Fifth Genaration Computer Systems, pp.292-298 (1984)

6) Mizoguchi,F.,Ohwada,H.and Katayama,Y. : LOOKS:Knowledge Representation System for Designing Expert Systems in a Logic Programming Framework, Proc. International Conference on Fifth Genaration Computer Systems, pp.606-612 (1984)

7) Furukawa,K.,Takeuchi,A.,Kunifuji,S.,Yasukawa,H.,Ohki,M.and Ueda,K. : Mandala:A Logic Based Knowledge Programming System, Proc. International Conference on Fifth Genaration Computer Systems,pp.613-622 (1984)

8) Tokoro,M. and Ishikawa,Y.:A Concurrent Object-Oriented Knowledge Representation Language Orient84/K: Its Features and Implementation, Proc. ACM OOPSLA Conference, pp.232-241 (1986)

9) Bobrow,G.,Kahn,K.,Kiczales,G.,Masinter,L.,Stefik,M.and Zdybel,F. : COMMONLOOPS Merging Common Lisp and Object-Oriented Programming,ISL-85-8, Xerox Palo Alto Research Center, August (1985)

10) Yonezawa,A.,Briot,J. and Shibayama,E.:Object-Oriented Concurrent Programming in ABCL/1,Proc. ACM OOPSLA Conference, pp.232-241 (1986)

11) 片山：オブジェクト表現を用いたプログラム開発の支援環境－プログラム開発における計画・管理システムの実現－,ICOT Technical Report TR-169 (1986)

12) 片山：表現システムKORE/KR の概要, 富士通㈱国際情報社会科学研究所, Brainware (1985)

13) 片山, 新谷：知識テーブル（その利用)-知識ベースにおける対象指向表現とその処理機構の試作- , 第31回情報処理全国大会論文集, pp.1233-1234 (1985)

14) Brachman,R.J. : What IS-A Is and Isn't:An Analysis of Taxonomic Links in Semantic Networks, IEEE Computer 16(10), pp.30-36 October (1983)

15) 片山, 新谷, 平石：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論文集, pp.1147-1148 (1986)

16) Hewitt,C. : Viewing Control Structures as Patterns of Passing Messages, Journal of Artificial Intelligence, Vol.8,pp.323-364 (1977)

17) Harary,F. : GRAPH THEORY, ADDISON-WESLEY,October (1972)