



Title	オブジェクト指向知識設計に関する考察
Author(s)	清木, 泰式
Citation	長崎大学工学部研究報告, 18(30), pp.23-30; 1988
Issue Date	1988-01
URL	<a href="http://hdl.handle.net/10069/24265">http://hdl.handle.net/10069/24265</a>
Right	

This document is downloaded at: 2019-04-24T20:11:29Z

# オブジェクト指向知識設計に関する考察

清 木 泰 弌\*

## Considerations on Object-oriented Knowledge Design

by

Yasukazu SEIKI

Object-oriented programming paradigm is fundamentally different from traditional functional methods. It is characterized by the class of objects, which is the primary elements of programming. In this paper, object-based knowledge design is considered on the metaphorical structures of experiences and the interaction of subject and environment. Knowledge can be decomposed into various categories, and the categories consist of articulations of basic behaviors. Objects are organized by these articulated behaviors to form knowledge organization. Knowledge must be considered together with the learning processes in which the teacher and the learner are treated as actor objects doing their own tasks, i.e., explaining and understanding.

### 1. まえがき

オブジェクト指向という新しいプログラミングパラダイムがソフトウェアシステム設計の重要な方法論として注目を集めている<sup>1)</sup>。従来の手続き型のプログラミング手法は、数値計算やデータ検索など、アルゴリズムとデータが分離したものについては有効であるが、知識や経験といった認知機能に関する対象を扱うには不適當であると思われる。オブジェクト指向は、“物”を中心にした我々人間の思考方法を自然に反映しているので、知識や経験を記述し操作するのに最も適したプログラミングパラダイムであると思われる。

とはいえ、現行のオブジェクト指向言語は必ずしも認知機能のモデル化を意図したものではないので、そのシステム構成や利用形態を改編することが必要である。本稿では、前報告<sup>2)</sup>で論じたメタファーの機能をオブジェクト指向の拡張の基盤として生かしながら、組織体としてのオブジェクトの機能や意味を明らかにし、知識設計のためのプロトタイプを求めてみた。また、オブジェクトの行動を主体と環境との相互作用という観点から捉えながら、知識獲得のための学習過程

を教師と学習者というアクターオブジェクトの機能と共に考察することを試みた。

### 2. 組織体としてのオブジェクト

オブジェクト指向とは、自分自身の処理方法と属性を内臓した〈オブジェクト〉を中心にプログラミングを行っていく新しい手法であり、従来の手続き型プログラミングとはその設計思想が根本的に異なる。なによりも、アルゴリズムとデータの二分化という形式をとらないため、プログラミングのスタイルが、思考形態に沿ったきわめて自然なものになり、しかもモジュール性の高いシステムづくりが可能である。

オブジェクトにメッセージを送ってオブジェクト自身に処理を行わせることの背景には、オブジェクトを、ある作用に対して反応する一種の生体とみなす考え方がうかがえる。このようなオブジェクトの属性をそれ自身に内臓させることはきわめて自然であり、設計者はオブジェクトに固有な操作や環境を記述することだけに専念できるという大きな利点がある。

例えば、地図の上を移動するプログラムをオブジェ

クト指向で考えてみる。地図上にはいくつかの地点があり、その地点の各々は道路もしくはその他のルートに関するデータ、特に、禁止されたルートや危険な地点に関する情報は各地点自身も持っているとする。この様な場合、手続き型のプログラミングで地点もしくはルートをとどるアルゴリズムを書くとしたら、これらの情報をすべてアルゴリズムに組み込まなければならないことは明らかである。これに対して、あるアクターが地図上を各地点の情報に基づいて移動するプロセスを、地図オブジェクト自身に行わせるならば、各地点のメソッドはアクターにしかるべき方向づけのメッセージを送ることだけを考えればよいことになる。すなわち、地点Aに居るアクターは、A自身から例えばルートRに沿って地点Bに行くようにメッセージを送られる。地点BではルートSが禁止されているのでルートTを指定するというように、地図上の地点自身がアクターを動かすことがきわめて容易に行える。また、各地点の状況が変化した場合、その地点オブジェクトのみを変更すればよく、アルゴリズムそのものの本質的な変更を行う必要がないのが大きな特徴である。これに対して、従来の手続き型プログラミングでは、変更に伴う探索や位置確定のためのアルゴリズムが一層複雑になることが予想されるし、また、何よりも地図上の各地点やルートの結合全体を把握することが容易でないという欠点がある。

オブジェクトの集合体を生体の集合体と考えるならば、それは組織体を形成する。オブジェクトは自分自身の中に内部状態と行動形式を所有しており、そして、メッセージ伝達により、すなわちコミュニケーションにより互いに結合しているので、情報とシステムの組織化を具体的に研究するための一つのプロトタイプを与えるものである。実際、オブジェクト指向言語をシステム設計のためのプロトタイプ言語として使用することもある。組織体に関する概念として、全体と部分、結合と分離、生体と環境等が典型的に考えられるが、とりわけ重要な概念は、発生、分化、成長という生体の進化に関連した概念である。

システム設計には、絶えず変更、改良等がつきまとうが、それにとともなるソフトウェアの改編のための労力は大変なものである。特に、手続き型のプログラミングにはそのような不都合が顕著にみられる。このため、設計者自身も含めてオブジェクトの設計と利用に関する組織体の形成過程の立場から、ソフトウェアの進化および淘汰に関する問題を考えていくのが、オブジェクト指向の重要なテーマである。

オブジェクト指向の考え方を生かしたソフトウェア

を押し進めて行くとき、ある仕事をある代理人(agent)に委任するというやり方に行き着く。代理人というオブジェクトを知識領域で働かせることは、それはエキスパートシステムそのものの基本的な機能を与えることになる。代理人オブジェクトは、自分の得意とする専門知識を自由に操作できるので、その管理はすべて代理人オブジェクトに任せられる。このような代理人の集合体が組織化されたとき、真のエキスパートシステムが構築された事になる。オブジェクト指向は、オブジェクトの生体的な主体性を確立することにより、このようなシステム設計を現実のものとする可能性を持っている。

### 3. オブジェクトの形成

オブジェクト指向プログラミングの特徴の一つは、あるオブジェクトクラスの属性がそのインスタンスに継承(inherit)されることである。そのことにより、プログラミングにおいて、新たなオブジェクトクラスを作ることから始めなくとも、既存のクラスを用いて目的とするオブジェクトを形成することが可能となるのである。したがって、オブジェクトクラスの形成をどのような空間で考えるかということは、オブジェクト指向プログラミングを進めるための主要な段階である。属性継承は、必ずしも、オブジェクト指向に必須の条件ではないという意見もあるが、いかなる知識も他の様々な知識との有機的な結合であることを考えるとき、この属性継承により構成されているオブジェクトの世界を意識的に拡張することは、オブジェクト指向を知識設計のなかで生かすための重要な課題の一つである。

これまでの議論の中で、オブジェクトの世界を組織体になぞらえてきたので、オブジェクトの属性形成を主体と環境との相互作用を基盤として考察してみよう。オブジェクト指向のもたらす第一の意義として、どのようなオブジェクトであっても、それがいったん形式を与えられたならば我々の思考がそれに集中し、必然的に関連した条件や付随物および環境が思考空間に連結してくることである。オブジェクト指向言語のオブジェクトクラス段階構造が、それ自体、暗黙の内にある設計思考を示しているように、オブジェクト化のプロセスには、必然的にそれを特徴づける主体と環境の属性が反映されるのである。

#### 3. 1 経験ゲシュタルトとオブジェクト

以下に示す Lakoff & Johnson<sup>3)</sup> による因果関係の原型に基づいて、オブジェクトの形成を考えてみる。

・行為の主体者は、行為を受けるものの状態にある変化が生ずることを目標としている。オブジェクトにメッセージを送るということは、まさに目標のオブジェクトに対して、なにかある反応を求めているからに他ならない。オブジェクト指向はオブジェクトそのものに関心が集中するので、行為の主体者の意図することがオブジェクトの構造に反映される。

・行為の主体者には、この目標に到達するための「計画」がある。

主体者の計画とはプログラマーにとってはプログラミングそのものであることは明らかである。既存のオブジェクトクラスから目的に沿ったインスタンスオブジェクトを生成したり、また、オブジェクトにメッセージを送ってオブジェクトを操作したりすることは、この計画の実行の一部分である。

・行為の主体者は首尾よくその計画を実行する。

計画には特定の順序や実行条件がある。計画の完全遂行にはそれらを常に考慮しなければならない。オブジェクトのメソッドは与えられたメッセージのなかに該当するメソッドがあればそれを実行するが、なければ最上位のクラスまでメソッドの探索を続ける。この探索を、あるレベルでうちきったり、実行を留保することができるように条件づけることにより、計画の順序性を明示的に利用することが可能となる。この点に関しては、後の節で再び考察する。

・行為の主体者は、行為を受けるものに生じた変化を知覚することによって、その変化をモニターする。

オブジェクトにメッセージを送ることによりオブジェクトには内部変化が生じるが、その変化を主体者に知らせるための方法を、オブジェクトはメソッドのなかに用意しておかねばならない。

・一つの特定の行為者に対して、行為を受ける一つの特定のものがある。

オブジェクト指向では、たとえば、数のオブジェクトクラスに対して図形を描けというようなメッセージを送ることは意味がないし、また、それに答えるメソッドも用意されていない。このように、主体者とその対象との関係を相互に規制することにより、意味のあるオブジェクトの結合だけを扱うことが出来る。

以上は、経験ゲシュタルトを形成する直接的因果関係の原型的な特徴のリストであり、直接手で行う操作の特徴を示したもののなのである。手で操作された結果出来たものを、我々は別の種類のものと見なす。その状態の変化を我々は〈物体は物質（内容物）から出て来る〉というメタファーに基づいて概念化し、これから「作る」という特別な概念が作られたのである。

ここにあげたものは、オブジェクトを形成するための直接的な基盤でもあることがわかる。すなわち、主体と主体の目標物との関係そのものが列挙されている。オブジェクトがそれ自身を操作する手順（メソッド）を内蔵することを合わせて考えるとき、〈物体（object）は物質（substance）から出て来る〉というメタファーからなる「作る」という概念は、オブジェクトクラスからオブジェクトが作られるプロセスを最も簡潔に要約したものであることがわかる。

上記の因果関係の原型を基盤として、容器（container）、道具（tool）乗り物（vehicle）という三つのメタファーを考えてみよう。オブジェクトクラスという容器の内容物からオブジェクトが形成される。明らかに、「作る」という行為には道具のメタファーが直接に結び付いている。目標に到達する行為には、乗り物のメタファーが用いられていることがわかる。この三つのメタファーは、後に述べる学習過程における教師と学習者の機能の説明にも用いられるが、ここで述べた「作る」という行為の構造と同様、これらが様々な組み合わせられて複雑な概念が形成される。この概念形成のプロセスは、オブジェクト化を通して知識設計のなかにも取り入れられる。

### 3. 2 アフォーダンスとオブジェクトの属性

Gibbson によって提唱されたアフォーダンス（affordance）という概念は、認知機構の生体的側面を明らかにするために導入された。Gibbson による説明には視覚空間という特異な背景があるので、ここでは Gardner による解説を引用する。「アフォーダンスとは事物や場面に固有な行為の可能性—ある種の生き物がある種の存在物とであったときに起こりうる活動—のことをいう。アフォーダンスの概念により環境内での生体の効果性の分析が可能になる。事物の意味は、事物が生体に提供するアフォーダンスである。事物が我々に取って意味を持つのは、我々が事物を使用したり事物に対して働きかけたり、反応したりするものを与えてくれるからである。」この説明からも容易に類推できるように、アフォーダンスに関する記述はそのままオブジェクトに関する記述として理解できる。ここには、生体的なオブジェクトの属性が最も簡潔に要約されている。また、経験ゲシュタルトの一つの相を形成する相互作用の属性の一面が、アフォーダンスの概念により具体的に表現されている。

アフォーダンスの概念を主体と環境との相互作用のもとで意識的に用いることにより、オブジェクトのもつ本来の性質がさらに明らかになると同時に、その属

性をさらに拡張された空間で把握することも可能となる。

#### 4. 知識の設計と利用のためのメタファー構造

知識は明らかに経験に基づいて形成され利用され、そして組織化されるものである。経験は、前報告でも論じたように、メタファーを基盤として構造化されているものであるから、知識というオブジェクトの形式もメタファーをもちいて記述出来るであろうということは容易に予想される。

オブジェクト指向の最も基本的な性質は、物や現象に対して人為的に境界を設定し、それらを個別化することである。それにより、物や現象の各々に固有な属性を自然に理解し記述することが出来る。知識というものは、普遍的な情報を暗黙的に、一方、特異性に富んだ情報は明示的に与えるものとして意味づけられる。したがって、ある対象をオブジェクト化することは、そのなかにオブジェクト固有の属性が含まれねばならないということから、知識そのもののオブジェクト化につながるものであるとも考えられる。その上、オブジェクトは単なる属性のリストではなく、そのオブジェクトにメッセージを送ることにより、オブジェクトの振舞いも知ることができるという点で、知識も一種の生体であると考えられる。このような意味で、オブジェクト化された知識の集合体をどのように組織化するかということは、単に知識の表現上の問題ではなく、知識を設計し利用する立場から考えても重要な問題である。具体的に知識をどのような行動パターンにおいて組織化するかというテーマに関しては後に述べることにし、ここでは、知識を構造化するための(すなわち、分解と組立のための)経験の諸相を、オブジェクト化というプロセスを通して明らかにしてみる。

##### 4. 1 オブジェクトの枠組みとメタファー

Lakoff & Johnson に準拠しながら、まず、オブジェクト形成の基盤とも考えられる存在のメタファーの特徴の中から、次の3項目を考えてみよう。

・特定の側面を識別する。

我々の関心事をオブジェクト化することは、関心の対象のもつ特定の側面を識別することにほかならない。したがって、オブジェクトは本質的に特異なものである。その属性は、暗黙のうちに普遍的なものから選択されたものである。

・原因を識別する。

ある事象をオブジェクト化することにより、その事象が置かれた時間的および空間的位置が明確になる。

つまり、その前後関係が他のオブジェクトとの結合やクラス関係によって示されることになる。

・目標を定め、行動に動機を与える。

要求そのものは実体のないものであるが、それをオブジェクト化することにより、そのオブジェクトをどのように操作するか、あるいは、どのように記述するかということが具体的に仕様化のなかに組み込まれてくる。また、要求は必然的に他のオブジェクトとの結合を伴うものであるから、それらの属性(それら自身も最初の段階では明確にされていないかも知れない)との相互作用が要求オブジェクトの行動に対して方向づけを与える。以上の諸相は、設計者の主体性を確立するためのものであるが、物体や物質(あるいはオブジェクト)を主体とした知識のカテゴリーを考えるには、次のようなゲシュタルトの諸相に沿ったオブジェクト化が必要である。

・知覚、運動活動、部分/全体、機能、目的

時計を例にとりて考えてみると、その形や大きさ、長針や短針の存在とその動き、時間を示すという機能と目的などが、時計の設計と利用に関する知識の代表的なものとしてあげられる。特定の対象を特徴づけるとき、我々は暗に上記のようなカテゴリーを用いていることは、日常の経験からも理解できよう。

##### 4. 2 基礎水準および類型化と関連性

一般的に経験カテゴリーにはある水準があり、そこではアクセスや想起が最も容易に行われる。これを基礎水準(basic level)という。我々は状況や環境の変化に応じて、カテゴリー化や分類のレベルを様々に変えることが出来るが、基礎水準に基づいた行動や認識の連鎖を形成する傾向がある(Rosch<sup>1)</sup>)。オブジェクトの形成をどのような基礎水準の意識空間で考えるかという問題は、そのまま類型化と関連性の問題となる。例えば、学術文献データの整理をどのようにシステム化するかという問題を考えるとき、様々な専門分野データベース、図書館といった類型化された概念が直ちに関連性の領域に入ってくる。前報告でも述べたように、関連性には主題的、動機的、解釈的の三つのタイプがあるが、どのタイプから出発しても、我々の意識はそれらの間を交互に移動するので、それにとまって派生する類型概念は様々なカテゴリーやクラスに分類されかつ統合されることになる。知識というものが意識のいわばルーティン化されたものによって構成されていることは、我々の日常の行動のなかで、知識が典型的に使用されていることを考えても明らかである。上述した様々なカテゴリーの相は、類型化と関連

性の領域を経験の基本的なレベルで明らかにするための一つの基準であり、オブジェクト化というプロセスを通してその意味が一層理解しやすいものになるのである。

#### 4. 3 行動の分節化とメソッドの組み込み

通常のオブジェクト指向では、オブジェクトはメッセージに対して反応するためのメソッドがその中に組み込まれているが、ここでは、経験ゲシュタルトの各相に対して反応する一次エンタリーを設けることを考える。すなわち、メソッドが起動される前提条件をまず設定するのである。もちろん、これらをメソッドとして形式化することも可能であるが、メッセージとして与えられるものに対する反応形式をすべてメソッドという概念のみで理解することはいささか無理があると思われるので、以下の考察に関してもメソッドという概念をオブジェクトの限定された操作として解釈することにしよう。

オブジェクト指向概念のもたらす重要な意義の一つとして、あまり関心ははられていないが、行動の分節化ということがあげられる。我々の意識は、上述したように物や現象を個別的に扱う傾向をもつので、ものおよびその属性の変化に応じて行動が推移することが自然に行われる。知識が経験と密接に関係を保ちながらその独自の存在を示している事を考えるとき、行動の構造を明らかにしておくことは、それに連結して経験から知識を獲得したり、また逆に行動を導く知識の役割を明らかにするためにも必要である。

経験の構造は、Lakoff & Johnson が経験のゲシュタルトとよんでいる概念の中に示されている。経験のゲシュタルトは、経験の分節化にともなういくつかの相によって特徴づけられ、それがメタファーによる概念の構造化の基盤ともなっている。知識の獲得がある特定の行動のパターンに基づいて行われるということは、知識が物語やスクリプトとよばれる記憶構造と結び付いて存在することを考えても明らかである。一般に、直接的行動、活動、出来事、経験に関するカテゴリーは次のような相をもつゲシュタルトである。

要求仕様化を例にとりて、各相の意味をオブジェクトの形成と役割と共に説明してみることにしよう。

・参加者（主役およびある役割を演じるアクター）要求者とシステム設計者、そしてシステムモデルをもこれに加えることが望ましい。

・部分（背景、重要な事実、エピソード、）要求の発生、要求の記述化、要求のシステム化、システムのシミュレーション、改編、完成、オブジェクト指向では

いわゆる抽象的な対象もすべてオブジェクト化して考えていくので、要求仕様化のプロセスの部分的な情報を把握するにはこれらをどのようにオブジェクト化するかということが、システム設計のかくれた重要項目になる。

・段階（前提条件）要求の発生からそれを記述し、設計するまでの間には行動の分節化以上の大きなレベルでの意識変化がある。それを、接続していくための条件を明示的に与えるには、要求というオブジェクトを中心とした操作プロセスを考えていくことが重要である。

・線の連続性（状態の時間的および因果的結合）時間的な順序関係や状態の変化系列が特定のパターンをもつような場合、これをオブジェクト化することは大いに意味がある。（時刻表や年表はその良い例である。）

・因果関係（エピソードと状態）因果の系列が構造化されるような場合、関係マトリックスや関係テーブルとしてオブジェクト化する。

・目的（目標、計画、一連の重要な出来事）要求仕様化における目標や計画ほど変更の大きいものはない。それらは仕様化が一巡するごとに変化する。このような変化にも容易に対応するのがオブジェクトである。それは動的に生成することも可能であるし、不用になれば消すこともできる。もちろん、不変的な枠組みはオブジェクトクラスとして用意しておけばよい。

知識をこのような相に沿って構造化することは、時間的に継続する行動を分節化するための一つの方法であり、また、個々の出来事を統合してその中から意味のある行為や意図を取り出すための手段でもある。

オブジェクトの中に組み込まれるメソッドは、メッセージに対する一種の反応形態のリストと考えられるが、意味のあるメッセージは、ある順序形態をもって送られて来るから、メソッドの起動には特定の順序関係や関連体系をその背景として設定することが必要であろう。また、前節でも述べたように、ある計画の順序構造をメソッドの起動条件のなかに組み込めば、ある一定の目標レベルに到達した後に次の行動のための条件が整えられるというより現実に近いオブジェクトの行動形態が得られることになるであろう。上記の相に基づいて分節化された行動形態が、メッセージの伝達形式やメソッドの組み込み形式に反映されるならば、知識オブジェクトは、〈もの〉や〈できごと〉を有機的に分類し統合するための一つのプロトタイプとしての役割を果たすことになるであろう。

#### 4. 4 学習過程 知識の理解と説明

知識構造のオブジェクト化を経験のゲシュタルトの相に沿って行うことにともない、知識の設計と利用に関する認知機能を、主体と客体もしくは環境との相互作用として特徴づけることが自然であることが明らかになった。以下の考察では、アクターオブジェクトとして学習者（もしくは質問者）および教師（もしくはエキスパート）を考えてみる。このように、知識を単なる情報の集合体として扱うのではなく、それを組織化するプロセスのなかでアクターが果たす役割と共に理解することの重要性は、近年ますます関心の高まってきているエキスパートシステムにおいても認められている。学習過程は、知識の組織化を人間というアクターオブジェクトの存在を明示的に想定しながら考察していくための最も基本的なテーマといえよう。

この学習過程には、一般に、説明と理解の構造をどのように学習過程の中に組み込むかという基本的な問題が存在する。理解と説明の構造は、必ずしも、教師の役割=説明、学習者の役割=理解、という対応関係のみで明らかにされるものではない。学習者の理解を確認するために説明を求めることや、学習者の質問を教師がどのように理解するかということは、学習一般に伴う日常的な行為である。当然、ここにも理解と説明という二つの概念間の相互作用が存在するが、その相互作用が知識というオブジェクトの形成を目的として成り立っていることを確認しておくことが重要である。理解作用の最も基本的なレベルとしては、これまで繰り返し述べてきたメタファーの役割をここでも強調しなければならない。

説明と理解の相互作用を最も単純なメタファーで表現してみよう。理解とは、ある容器に内容物 (substance) が入ることである。説明とは、用意された容器に内容物を入れることである。理解の説明とは、どの容器にどのような内容物が入っているかを示すことである。これを次のように Smalltalk 言語形式で表現してみよう。

Learner Container ← Knowledge Container new:  
Frame For Accept

Teacher Container ← Knowledge Container new:  
Frame For Access

ここでは、学習者には知識を受け取るための容器が、また、教師には知識集合体へのアクセスのための特別な構造をもつ容器が用意される。

Teacher Container put: Substance from:  
Knowledge Container

教師はエキスパート知識から内容物を取り出す。この

場合、教師だけがこのような操作を許されている。

Learner Container put: Substance from: Teacher  
Container

教師の容器の内容物を学習者の容器に格納する。ここで、なんらかの説明が行われたことになるが、上記の操作に対して、学習者が自分の容器の構造にもとづいて内容物をうまく格納出来たときは、しかるべき評価が返って来ることになる。

このように、容器のメタファーだけを用いて単純に解釈すれば、説明とは容器の内容物を別の容器に移し変えることであるといえる。上記の例では、教師だけが知識集合体にアクセスできるような制限をもうけたが、これは、容器そのものの構造や機能のなかに、学習者と教師との能力の違いを反映させるためである。これをさらに道具や乗り物のメタファーで構造化することにより、内容物を加工して別の物を作ったり、媒体を通して別の場所に運ぶことが可能であるような容器のシステムが形成されることになる。媒体とは様々な表現形態であり、伝達手段であり、要するに情報の移動をもたらすものである。もちろん、ここで述べたことは説明行為の低次のレベルの形式にすぎないが、オブジェクトをメタファーのレベルから捉えることにより、学習過程を通じて教師と学習者の間で繰り返される行動様式が、どのようなオブジェクトを媒介にして成り立っているか、また、進化していくのかということが、よく理解できよう。

#### 5. あとがき

オブジェクト指向という新しいプログラミングパラダイムのもとで、経験に深く根ざした知識というオブジェクトを、メタファーや関連性の概念をその基盤としながら形成することを考察してみた。オブジェクト指向だけが知識設計のための唯一の方法論というわけではないが、システム設計というプロセスを認知レベルから統合することを考えるとき、オブジェクト指向のもたらす意義は大きいと思われる。これまで、知識や経験の概念的枠組みとして考察してきた Schutz の類型化と関連性の概念や Lakoff & Johnson のメタファーの概念が、オブジェクト形成を考えるための重要な基盤となることが明らかにされたという点で、オブジェクト指向というプログラミングパラダイムのもつ認知理論の意味は、さらに検討に値するテーマであると思われる。

今回はオブジェクト指向言語として特定のものを取り上げず、オブジェクト指向概念そのものの拡張を試みながら、それが我々の思考過程を組織体の形成へと

導くことを示唆するだけにとどめた。本稿で述べたオブジェクト指向の機能は、その本来の機能や特徴をすべて言い尽くしたものではない。今後のテーマとして、知識設計および学習過程の基盤となる経験ゲシュタルトを、認知レベルでのオブジェクトクラスの体系として構築しながら、オブジェクト指向の概念を単なるプログラミングパラダイムにとどめることなく、さらにその考えを認知理論のなかで展開していくことを目指している。

#### 参 考 文 献

- 1) Grady Booch, "Object-Oriented Development", IEEE Trans. Software Eng., vol. SE-12, Feb., 1986.
- 2) 清木：要求仕様化のメタファー構造。  
長崎大学工学部研究報告第17巻28号（昭和62年1月）
- 3) Lakoff & Johnson, *Metaphors We Live By*.  
—The Univ. of Chicago Press. (1980)
- 4) H. ガードナー，認知革命，p.295, p.329,  
一産業図書（昭和62年）