

続・オブジェクト指向で 世界を写し取る

フジタ未来経営研究所
リサーチアソシエイト

井庭 崇

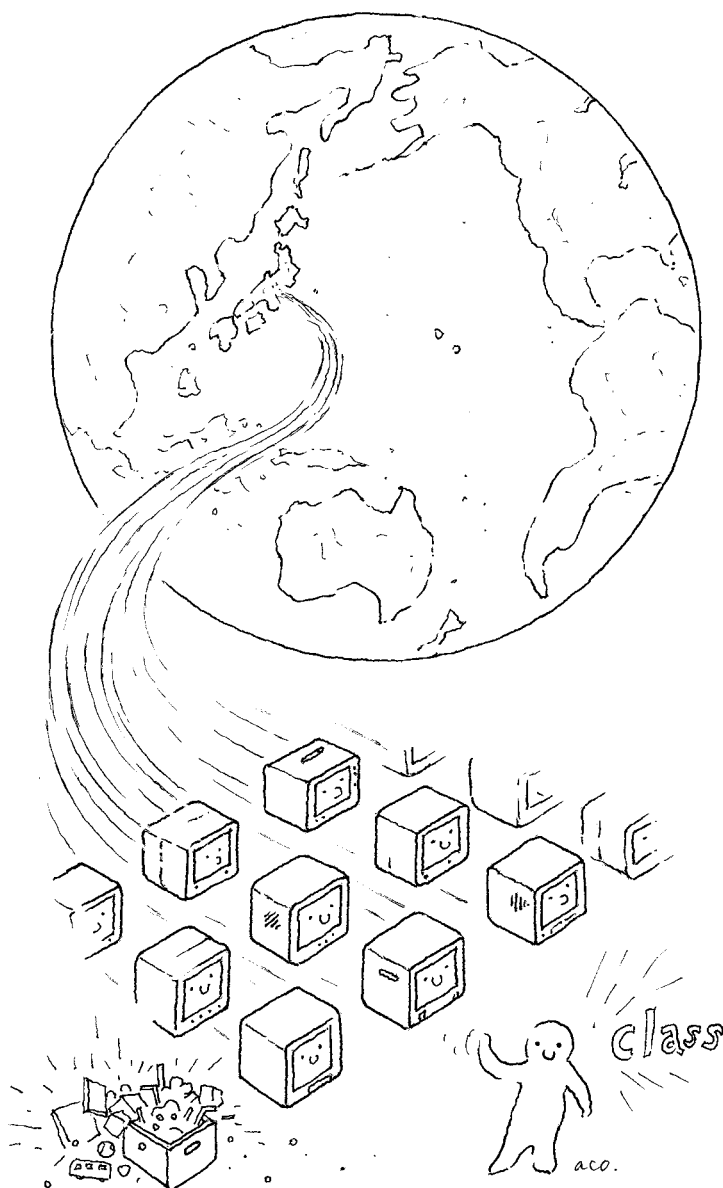


Illustration 巽 亜古

「未来をデザインする」には、その対象となる世界をモデル化することがとても有効な手段となります。世界をモデル化することによって、現状を分析

したり、解決策を考えたり、将来の予測を行ったりすることが可能となるからです。たとえば、昨今の激しい競争のなか、企業はダイナミックにビジネス

スモデルや業務プロセスの構築・見直し・改善をしていくことが求められていますが、その際には対象となるビジネス・業務領域をモデルとして把握している必要があります。頭の中であれ、紙の上であれ、なんらかのあたりで対象となる世界がモデルとして写し取られている必要があるのです。しかし、現実の世界はさまざまな要素が絡み合っているため、世界のモデル化は必ずしも簡単なことではありません。そこで必要となってくるのが世界モデルを作成するための方法論です。今回も未来をデザインするために、どうやって現実世界をモデル化していくかの方法をみていきたいと思います。

前回、現実世界の構造や動きをモデル化するためのひとつの方法として、オブジェクト指向の考え方を取り上げました。オブジェクト指向とは、対象となる世界における物理的・観念的なモノを、「属性」と「振る舞い」の観点から「オブジェクト」に写し取り、それを単位に世界モデルを作成していくという手法です。オブジェクト指向で作成されたモデルはそのままコンピュータ言語に置き換えてコンピュータで動かすことができるため、その世界モデルがどのような動きをするのかというシミュレーションがしやすくなります。最近では、ビジネスの構成要素（たとえば、財務会計、販売管理、管理会計、在庫・購買管理など）をモデル化したオブジェクトを共有して用いているという「ビジネスオブジェクト」の考え方も出てきており、今後ビジネスシーンにおいてもますますオブジェクト指向は重要な考え方となっていくといえるでしょう。

今回は、複雑な世界をモデル化するために、どのように対象世界を体系化して理解・表現するとよいのかという観点からオブジェクト指向に接近してみたいと思います。ここで鍵となるのは、今回中心的

に着目する「クラス」という概念です。このクラスという概念により、私たちは対象世界を把握する際に、人間の認識と同じような方法で上手にオブジェクトを体系化することができるようになるのです。また、「クラス」は世界モデルを作成する効率も大幅に上げることができるという利点も併せもっています。クラスの具体的な話に入る前に、クラスの概念がベースとしている人間の認知の特徴についてみてみることにしましょう。

人間の認知と体系化

私たちは世界のさまざまな物事を認知するとき、複数のものをひとまとめにして捉えたり、体系化して把握したりしています。たとえば、四本足の椅子と一本足の椅子、横長のベンチ、ビーチチェアを見て、それぞれ形が異なっているにもかかわらず、これらをひとまとめに『椅子』として認識することができます。もちろんそれぞれの椅子は、構造だけでなくデザインにおいても違いがあるわけですが、座るためのものであるという特徴に着目するならば、上記のものはすべて『椅子』というものに分類され理解されることとなります。複数の物事を体系化するにあたり、私たちはそれらに共通する性質に着目して、それ以外の部分を省略することで、概念的なひとつのまとまりとして認知するのです。

このように分類して概念としてまとめる能力は、人間が世界を理解したり世界に働きかけたりするうえで本質的なものであり、知覚、記憶、言語などの認知・思考活動の基礎であるといわれています。もし、このような分類・概念能力がなかったとしたら一体どうなるのでしょうか？ まず、見えたものをいちいち「初めてみるもの」として対処しなければ

ならなくなってしまう。また、新しく出会った椅子を『椅子』であると理解できないため、その扱い方がわからず、うまく扱うことができなくなりま

す。『椅子』には座ることができるといふことは、私たちが『椅子』という概念に付随してもっている知識だからです。もし『椅子』という概念がないならば、このような知識を一括して付与することができないので、『この物体には座ることができるといふように個別に記憶しなくてはならなりません。さらに、『この物体』や『あの物体』の特徴を個別に語らなければならなくなるので、何かを伝達するということも極端に難しくなってしまう。このことから、世界を認知するときに、ふだん私たちがしているように個別のものを概念的なまとまりに分類して把握していくことがいかに効率的かということがわかります。

分類して体系的に理解する

さて、世界をモデル化するという話に戻りましょう。世界をモデル化する際にも、人間の世界把握と同じような体系化の方法で整理することができるとなれば、大変便利そうです。私たちの認識している世界のイメージをモデルにそのまま投影できるため、モデルを作成しやすく、理解しやすいくからです。実はオブジェクト指向にも、このような体系化の方法が用意されています。それが「クラス」という概念です。

オブジェクト指向では、概念的なまとまりをクラスによって表現することができます。オブジェクト指向の言葉でいうならば、クラスとは、共通の「属性の種類」や「振る舞い」をもつ「オブジェクト」

をひとまとめにして表現したもののことです。日常的にも私たちは「クラス」という言葉を、複数のものを共通の性質によって分類したまとまりを指す言葉として使っています。それに似た考え方で、たとえば、車の種類のセダンクラスやワゴンクラス、スポーツカークラスというのもそうですし、爬虫類、鳥類、哺乳類というクラスもある共通項で生き物を分類したまとまりです。同じように、オブジェクト指向におけるクラスは、「属性の種類」と「振る舞い」を共通項として、オブジェクトを分類したものの

なです。

オブジェクト指向にクラスの考え方を導入するとどのような利点があるのでしょうか。それは、それぞれの具体的なオブジェクトの特徴をすべて個別に考えなくても、ひとまとまりにしたクラスで一括して表現できるようになるということです。このことがどれだけ重要なことであるかは、先ほどの椅子の話の思い出せば明らかでしょう。前回用いたテレビの例でいうならば、テレビにはチャンネルを変える機能が

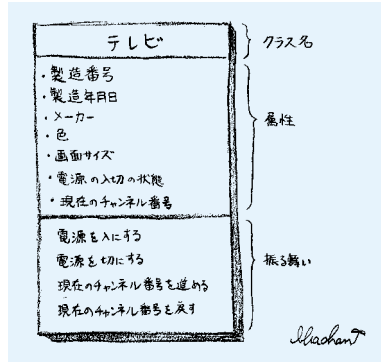
あるということを表現するときに、「この物体にはチャンネルを変える機能がある」、「その物体にもチャンネルを変える機能がある」、「……」というように個別的に列挙していかなくても、『テレビ』にはチャンネルを変える機能がある」というだけで、『テレビ』一般について語ることができるというわけです。世の中にはたくさんテレビがありますが、それらは製造番号やメーカーなどは違うものの、同じような属性や振る舞いをもっているはず。これらのテレビの特徴をもっている個別具体的なものを総称して、私たちは『テレビ』と呼んでいるわけです。オブジェクト指向で写し取ったモデルの世界でも、私たち人間の認識と同じように『テレビ』クラスというものでそれを表わすことが可能になるの

オブジェクト指向の起源

通常、オブジェクト指向という用語はプログラミングをするための概念と思われがちですが、オブジェクト指向は単なるプログラミング・スタイルではなく、より広く用いることができる考え方です。

オブジェクト指向の考え方は、もともと世界を写し取るための手法として考案され発展してきました。オブジェクト指向の起源は1960年代にノルウェーでO・J・ダールとK・ニガードによって開発されたSIMULA (シミュラ) というコンピュータ言語にさかのぼります。SIMULAというのはsimulation language (シミュレーション言語) の略で、その名が示す通り現実世界のさまざまな現象をシミュレートするためのコンピュータ言語でした。SIMULAは、何千もの構成要素からなるような複雑なシステムのモデルを作成してコンピュータ上で動かすことを目的に設計されたので、動的で複雑な現実世界をそのままコンピュータ上に取り込むための工夫がなされました。その工夫が、現在のオブジェクト指向の考え方の本質をなしているのです。結局、この言語そのものは広く普及しなかったのですが、その後多くのコンピュータ言語や分析方法に多大な影響を与えることになったのです。

図1 『テレビ』クラス



です(図1)。

クラスからオブジェクトを生成する

クラスの考え方を別の角度から眺めると、クラスは個別具体的なオブジェクトを生み出すための雛型であると捉えることもできます。イメージとしては、テレビをつくるための設計図のようなもの、もっと単純にいうならタイヤキをつくるための鉄板の型のようなものと考えるとわかりやすいでしょう。オブジェクト指向によるモデル化においても、クラスという型からオブジェクトを生成します。ここでは、『テレビ』クラスから「私のテレビ」や「友人Fのテレビ」、「隣人Yのテレビ」という個別具体的なテレビのオブジェクトが生成されるということを考えてみましょう。

クラスには、そのクラスから生成されるオブジェクトがもつべき属性の種類と振る舞いが定義されています。『テレビ』クラスの場合にも、それから生成されるすべてのテレビ・オブジェクトが共通にもつ属性の種類が定義されています。具体的には、「製造番号」、「製造年月日」、「メーカー」、「色」、「画面サイズ」などがあるでしょう。そして、この『テレビ』クラスから生成された「私のテレビ」、「友人Fのテレビ」、「隣人Yのテレビ」というオブジェクトは、どれも「製造番号」、「製造年月日」、「メーカー」、「色」、「画面サイズ」という属性をもつこととなります。

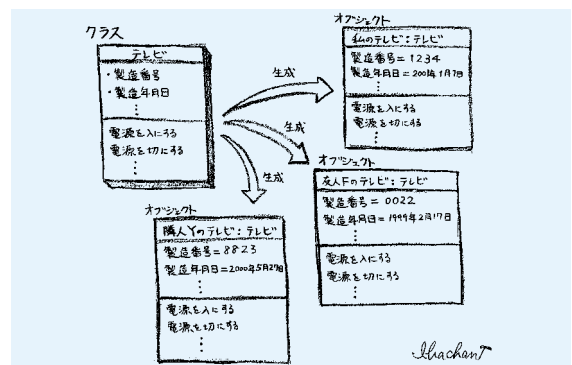
ここで注意が必要なのは、それぞれのオブジェクトのもっている属性の種類は一緒ですが、それぞれの属性の値(状態)はお互いに独立だということです。たとえば、「私のテレビ」オブジェクトの製造番号の値は「1234」でしたが(前号参照)、「友人Fのテレビ」オブジェクトは「0022」、「隣人

Yのテレビ」オブジェクトは「8823」というようにそれぞれ異なっているでしょう。また、「私のテレビ」オブジェクトと「隣人Yのテレビ」オブジェクトはどちらも色が「黒」であるというように同じ値をとる属性もあります。重要なことは、同じクラスから生成された各オブジェクトの属性の種類は一緒であり、その種類についての情報がクラスに定義されているということです。

クラスには、属性の種類のほかにそのクラスから生成されるオブジェクトがもつ振る舞いも定義されています。たとえば、『テレビ』クラスの場合には、個別のテレビ・オブジェクトが必ずもつ振る舞いである「電源を入にする」、「電源を切にする」、「現在のチャンネル番号を進める」、「現在のチャンネル番号を戻す」が定義されています。この振る舞いには、属性の場合と違って具体的な振る舞いの中身が記述されます。なぜなら「電源を入にする」という振る舞いが意味することはどんな場合でも同じだからです。

以上のような「属性の種類」と「振る舞い」が定義されたクラスを用いて、オブジェクトを生成することができます。1つのクラスから複数のオブジェクトを作成することができるので、一度クラスを定義してしまえば、振る舞いが同じであって属性の値だけが異なるというオブジェクトを容易に大量生産できるようになるわけです。現実世界のように非常に多くのモノをオブジェクト指向で表現したいときには、ひとつひとつのオブジェクトを別々に定義するよりも、クラスを用いることでより効率的にモデル化を進めることができるようになります(図2)。

図2 『テレビ』クラスからオブジェクトが生成される



クラスの関係としてまとめて表す

クラスという考え方を導入する利点は、具体的なオブジェクトについて個別に考えなくても、クラスで一括して共通項を表現できるようになることだ、ということを見てきました。さらに、オブジェクト同士の関係性を定義する場合にも「属性の種類」や「振る舞い」をまとめて定義するのと同じようにクラスの考え方が重要となります。オブジェクト指向では世界の動きを複数のオブジェクトの相互作用で表現するのですが、このときもそれぞれのオブジェクトを生成するクラスどうしの関係として一括して定義することができるようです。オブジェクトどうしの関係は、それぞれの型であるクラスどうしの関係に準ずるためです。

たとえば、人間とテレビの所有関係を表すための1つの方法は、「私は私のテレビを所有する」「FさんはテレビFを所有する」「……」というように個別に言うことです。もうひとつの方法は、個別的なものをまとめた概念である『人間』と『テレビ』の関係で表わすことです。この場合は、『人間』クラスと『テレビ』クラスは「所有」という関連をもっている、といえればよいこととなります。『人間』クラスからは「私」「あなた」「Fさん」「Yさん」などのオブジェクトが生成され、『テレビ』クラスからは「私のテレビ」「テレビA」「テレビB」が生成されます。関連についても、それぞれのオブジェクトについて個別に定義するのではなく、ひとまとめにしたクラスを用いて一括して語ることができるので効率的です(図3)。

バリエーションの表現方法と継承

世の中を見まわしてみると、テレビといっても標準的な機能をもったものだけでなく、衛星放送チューナー内蔵型、ビデオ一体型、カーナビ一体型など、他の機能が加わったいろいろなバリエーションがあることに気づきます。現実世界をモデルに写し取りたいわけですから、モデルにおいてもこのようなバリエーションをつまく表現できるとよいでしょう。このようなときに用いられる関係を、オブジェクト指向では「汎化関係」と呼びます。

汎化関係とは、概念における上位・下位の関係を表わしています。汎化関係は図4のように階層的な関係であり、この階層において上位クラスは下位クラスを「汎化」したものになります。逆に、下位クラスは上位クラスを「特殊化」したものになるわけです。たとえば、『テレビ』は『衛星放送チューナー内蔵型テレビ』をより一般化した概念であり、逆に『衛星放送チューナー内蔵型テレビ』は『テレビ』の特殊なものであるといえます(『衛星放送チューナー内蔵型テレビ』や『ビデオ一体型テレビ』は、『テレビ』の一種(a-kind-of)であるといえるので、汎化関係は「a-kind-of」関係と呼ばれることもあります)。

オブジェクト指向におけるクラス階層では、上位クラスの属性や振る舞いを下位クラスが引き継ぎます。このことを「継承」といいます。継承の考え方があることによって、共通の振る舞いや属性は上位のクラスで一度だけ定義すれば良くなり、下位クラスで再び定義する必要はなくなります。たとえば、図4のように『ビデオ一体型テレビ』クラスは、『テレビ』クラスと汎化関係にあるため、『テレビ』

図3 『人間』クラスと『テレビ』クラスの間

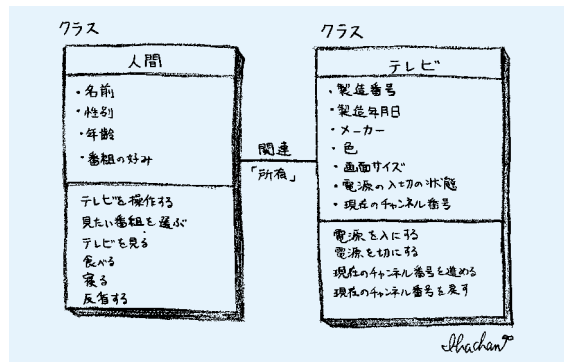
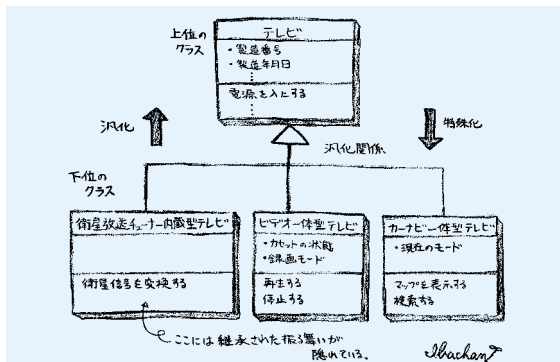


図4 テレビのバリエーションとクラス階層



クラスで定義されている「製造番号」「製造年月日」「メーカー」という属性の種類を継承します。また、「電源を入にする」とか「現在のチャンネル番号を進める」というような振る舞いも継承します。この継承の機能により、『テレビ』クラスがもっている属性や振る舞いを『ビデオ一体型テレビ』クラスで再び定義する必要はなくなります。『ビデオ一体型テレビ』も『テレビ』の一種であるため、わざわざ明記しなくても自明なことで扱われるのです。

そして、クラス階層の下位クラスでは上位クラスにはない新しい属性や振る舞いを追加することができます。『ビデオ一体型テレビ』クラスには、『テレビ』クラスには定義されていない新しい機能であるビデオの操作に関する振る舞いなどが加わります。下位クラスには、上位クラスと下位クラスとの差の部分のみを定義すればよいということになります。以上のように、汎化関係は共通する属性や振る舞いを一括して定義した上位クラスと、それらを特殊化した下位クラスによる階層関係であり、この関係によって、少しだけ違うようなものを体系化して整理することができるようになります(図4)。

認識したものを世界モデルに記述する

今回は、オブジェクト指向で現実世界を写し取るときに、どのようにオブジェクトを体系化していくのかということについて考えました。オブジェクト指向では、クラスという型によってオブジェクトを分類し、効率的に記述できるようになっています。また、複数のクラスを抽象化して上位クラスを作成することによって、より効率的な分類と記述が可能となるのです。

クラスによって体系化したり一括して定義したり

できるということのみてきましたが、写し取られた世界モデルは、その人の独自の視点と視野で得られたものとなるということを、ここでもう一度確認しておきたいと思えます。オブジェクト指向の方法は世界の写し取り方を教えてくれますが、どこをどう切り取るかということは、その人その人に任されているからです。それゆえ、同じものをモデル化しても人それぞれ異なるものになることが多々あります。本連載の目的であるヴィジョンの表現や伝達、あるいは新しいビジネスモデルの構築や業務プロセスの分析などにおいては、この個人差の存在を積極的に捉えていくことが重要だと思えます。自分の中だけに広がっている現状認識やヴィジョンをモデルとして外部化し人に伝える。オブジェクト指向は、コミュニケーションの溝を埋めるためのひとつの表現形式になるといえるでしょう。

次回からは、いよいよ、世界モデルをダイナミックに動かすためのプログラミングやシミュレーションの話に入っていきます。それでは、次回もお楽しみに！

いは・たかし



1974年神奈川県生まれ。慶應義塾大学大学院政策・メディア研究科博士課程在学中。著書に『複雑系入門』(共著、NTT出版)など。

E-mail: iwa@stc.keio.ac.jp

【コラム】100数年、オブジェクト指向で経済社会を表現することを支援するシステムと方法論の構築に取り組んでいますが、徐々にかたちになりつつあります。このプロジェクトの成果も、本誌の研究レポートで紹介していきたいと思っています。

References

- 岩田裕道・手島歩三『ゼロからわかるオブジェクト指向の世界 ビジネス活用のために』日刊工業新聞社、1996年。
- 青木淳『オブジェクト指向システム分析設計入門』ソフト・リサーチ・センター、1993年。
- 青木淳『例題による!! オブジェクト指向分析設計テクニック』ソフト・リサーチ・センター、1994年。
- I.ロス・J.P.フリスピー『知覚と表象』第2版、海文堂、1991年。
- M.W.アイゼンク編『認知心理学事典』新曜社、1999年。
- 森敏昭・井上毅・松井孝雄『グラフィック認知心理学』サイエンス社、1995年。
- Ole-Johan, Dahl, K. Nygaard, "SIMULA: An Algol-Based Simulation Language", *Communication of the ACM*, vol.9, pp.671-678, 1966.