

認知文法に基づくオブジェクト指向の理解

金田 重郎

あらまし

Object 指向は、Object 同士の相互作用としてシステムの振る舞いをとらえる考え方であり、当初はプログラミング手法（Object 指向プログラミング）として提案された。その後、その適用範囲は拡大し、Object 指向分析、Object 指向設計、Object 指向プログラミング、とソフトウェア開発の最上流から下流までのすべての分析・設計プロセスにおいて、最も重要な「対象世界をモデリングするためのツール」となっている。ソフトウェア工学分野では、Rational Unified Process を始めとして、Object 指向に立脚する多くの方法論が提案され、実際に利用されている。そして、ある面、残念なことであるが、これらの方法論は、すべてその起源を欧米諸国に持っている。Object 指向を始めとする英米の方法論の背後には、英語の認知構造が反映されていると見るのが自然であろう。とりわけ、対象世界を自然言語を用いて表現する最上流の要求分析や要件定義のプロセスでは、背後にある母語の影響が少なくないことが予想される。しかし、何故か、我が国のソフトウェア工学の歴史では、海外の方法論を導入する際、英語・日本語と言う「母語」の差異から生じるであろう問題については、全く無視されている。Object 指向の著名なテキストは、多くの場合、英語を原著言語として書かれている。結果として、我が国の学生・SE は、日本語に「翻訳」されたテキストを用いて、Object 指向を学んでいる。このことから、思わぬ誤解が生じることがある。例えば、「Object 指向は、対象世界の『もの』に着目して、ビジネスを分析する手法」とされることがある。しかし、認知文法では、英語の

Object は、日本語の「もの」ではない。英語圏では、Object は、名詞の可算用法、いわゆる可算名詞によって表現される。「Person」「Event」等は、Object となり得るが、「Water」は Object ではあり得ない。しかし、日本語訳の中だけで Object 指向を理解しようとして、その結果、Object を日本語「もの」に等しいと誤認識すると、「発注」等の「イベント」を Object として扱うべきか否かについて、経験の浅い SE（ソフトウェアエンジニア）あるいは学生は困惑する結果となる。そもそも、ソフトウェア開発の最上位の要求分析フェーズでは、ヒアリング結果をクラス図に直して、モデルの精緻化が図られる。上記視点に立てば、日本語によるヒアリング結果・仕様記述をクラス図へ変換する作業は、日英翻訳に等しい。そこに、日本語による Object 指向分析の困難性がある。この問題を解決するために、本稿では2つの主張を行う。第一に、日本語と英語の特性を考慮すると、クラス図を作成する際には、日本語からそのまま名詞や動詞を抜き出してはならず、日本語テキストから主語を探したり、因果関係を確認する「クリティカル・リーディング」をクラス図作成時に行う必要がある。もう一点の主張は、認知文法に基づくクラス図の理解である。これによって、Object は「もの」ではなく、可算用法を有する名詞となる。「Is-a 階層」「Has-a 階層」への理解は、日本語のみで行なっている場合に比べて、深いものとなり、デザインパタン等に現れる「Is-a 階層」「Has-a 階層」を理解したり、利用することが容易となる。

1. はじめに

「Object 指向」は、今日のソフトウェア工学において、極めて重要なモデリング手法のひとつである。Object 指向が、シミュレーション言語 Simula に起源を持つことは広く知られている。Simula は、ノルウェーのクリステン・ニガードとオルヨハン・ダールが 1962 年から 1967 年にかけて実装した言語である。現在の Object 指向プログラミング言語の主要な概念は、すべて、この Simula において実装されていたとされる。

その後、Object 指向の考え方は、Object 指向プログラミングだけではなく、要求分析・要件定義¹ プロセスに相当する「Object 指向分析」、及び、詳細設計段階の「Object 指向設計」へと、その適用範囲を拡大している。これら Object 指向アプローチの中で、本稿では、対象ビジネスをクラス図 (ER 図) として写し取る「Object 指向分析」を議論の対象とする。Object 指向分析では、実装設計までは考慮しない、比較的粗いクラス図 (ER 図) が利用される。これを、概念クラス図 (概念 ER 図) と呼ぶことがある。Object 指向分析では、ユーザインタフェースや帳票はモデリングの対象ではなく、対象ビジネスに内在する本質的な「もの」や「こと」が対象となる。

Object 指向分析では、分析対象世界のビジネスをモデル化するが、モデル化にはドメイン知識が必要である。従って、通常は、モデル化は、SE (ソフトウェア技術者) のみでは遂行できない [1]。本来は、専門家以外の業務専門家が、Object 指向分析を理解しているべきである。政策科学の専門であっても、Object 指向分析を身

につければ、政策研究のひとつの手段たり得る。しかし、Object 指向分析の習得は、情報工学の専門家でも容易ではない。特に、日本人には習得が難しいのではないかというのが、後述する本稿の主旨である。

Object 指向のテキストの中には、「Object 指向は、対象世界に存在する『もの』に着目して、対象世界をモデル化する手法である」と言った説明がなされていることがある。Object 指向が日本に紹介された当初、このような説明が為されたと思われる。悪い意味で、その伝統は生きている。しかし、これは誤った理解である。せめて、「実在、あるいは概念的な『もの』」とすべきである。日本語版 Wikipedia の説明はそうになっている。しかし、この定義も決して十分ではない。Object を日本語の「もの」として捉えてしまうと、「日本語『もの』が対象とするあらゆるもの」が Object 候補となる。そして、初学者は、要求ヒアリング結果 (テキスト) を目の前にして、モデル作成に困難性を感じるはずである。

本稿では、上記の問題が生じる大きな原因が、「英語を母語とする国で開発されたソフトウェア工学の方法論を、そのまま、母語の違いに留意せず、日本に持ち込んでいるところに、困難性の根本的な原因がある」との立場から分析する。すなわち、「我が国の学生・SE (ソフトウェア技術者) が、『海外の Object 指向の方法論をそのまま日本語で表現したテキスト』を用いて Object 指向を学び、そして、日本語で実施している」ことに疑問を呈する。たとえば、日本語の Object 指向の説明では、「Object」=「もの」とする。しかし、認知言語学によれば、「Object」とは、可算の概念 (名詞) であって、「会社」



図1 クラス図の例

¹ 「要求」「要件」という言葉の定義は学術的に確定していないが、おおよそ、「要求」とは対象ビジネスあるいはユーザが持っている実現したいものであるのに対して、「要件」は、それを情報システムの基本的な機能として落とし込んだものである。

や「人」は Object であるが、「水」や「砂」は、最初から、Object の候補にはならない [4]。当該名詞で表現されている対象を数えられる名詞概念 (可算名詞) のみを Object 候補とするなら、Object は、クラス図におけるクラス (ER 図のエンティティ) そのものである。

そして、上記の視点から、本稿では、ER 図や UML [2] のクラス図等の Object 指向のためのダイアグラムは、英語の認知構造をそのまま素直に図形化したものに過ぎないことを明らかにする。英語のネイティブにとっては、要求分析結果のテキスト記述をクラス図に置き換えることはストレートフォワードな作業である可能性が高い。一方、ターゲットが英語の認知構造とすれば、日本人にとっては、日本語テキスト記述をクラス図に変換することは、日英翻訳に等しくなってしまう。このような大きな言語的ギャップの存在を無視し、英語圏の方法論を直訳したテキストを初学者に与え、「習うより慣れる」的な徒弟産業的教育方法を採用することは、果たして、ソフトウェア工学教育手法として、適切なのであろうか。

なお、「言語の差異が、対象世界の認識に影響を与えているのではないか」との見方 [5] は、「サピア=ウォーフの仮説」として広く知られている [6]²。本稿は、ソフトウェア工学、とりわけ、Object 指向において、このサピア=ウォーフの仮説的な問題意識を導入しようとするものであるとも理解できる。

以下、第2章では、概念クラス図 (概念 ER 図) の役割を確認する。第3章では、認知文法に基づいて、クラス図が、英語の認知構造そのものであることを確認する。第4章では、英語における is-a 階層が日本のそれとは違うことを明らかにする。第5章では、日本語の持つ特性と、クラス図への変換における課題について述べる。第6章では、日本語がそのままクラス図には適用できない例として、関連名の有るべき姿を探る。第7章では、上記の日本型の関連名が採用されている例として、概念データモデリングについて言及する。第8章では、本稿で得

られた知見から、ソフトウェア工学の課題にひとつの仮説を提示する。第9章は、まとめである。

2. クラス図 (ER 図) について

本論に入る前に、本稿におけるクラス図 (ER 図) について確認しておく。図1は、クラス図の記述法を説明するための図である。クラスにおいては、最上位にクラス名を記し、その下に属性名を、更にその下にはメソッド名を列挙する。尚、図1では、プライマリーキー (PK)³ について表示している。クラスには、当該クラスを識別するための、何らかの識別子は必要である。しかし、要求分析のための概念クラス図では、データベース設計に必要なレベルの PK 表現にする必要はない。本稿の以下の記述では、対象世界で当該クラスを識別するための識別子を用いても、PK を用いた実装設計レベルには落とさないで説明する。

図1の各クラスの下には、実際には、それぞれのクラスのインスタンスに相当する「オブジェクト (Object)」が貼りつくことになる。この「オブジェクト」のレベルで描いたのが UML の「オブジェクト図」である。また、図1のクラス・クラス間には、「関連」が描かれることがある。クラス図上のクラスや関連は、それが存在し得ることを表記するが、実際のインスタンスである「オブジェクト」の段階では、常に存在することを保証するものではない。この個数に関する制約関係を表現するのが、関連に描かれている、「0..1」と言った多重度 (マルチプリシティ) である。「0..1」とは、0 又は 1 との意味であり、相手側のクラス (正確には、クラスのインスタンスである具体的な「オブジェクト」) から見た際には、そのようなリンクがないか、あっても、高々1本であることを意味している。ここで、「リンク」とはオブジェクト間に存在する関係を表している。「0..*」の場合、当該リンクは存在しないこともあるが、

² この仮説は、証明不可能とされている。詳細は、文献 [6] の訳者解説追補を参照されたい。

³ データベース設計において、当該レコードを一意に識別できる属性、例えば、社員番号、学生番号、商品コードなどを Primary Key と呼ぶ。また、その Primary キーが、他レコードによって参照されて、当該他レコードとプライマリーキーを持つレコードとの関連を明示する場合には、フォレンジキー (外部キー) (FK) と呼ぶことがある。例えば、納税通知を送る場合、当該納税通知を識別する納税通知書 ID は PK であり、納税通知の中に記録されている受取人、つまり、納税義務者の ID は、FK になる。

逆にリンクが存在する場合には、1本にとどまらず、多数のクラス配下のインスタンス（「オブジェクト」）にリンクが張られることになる。例えば、利用者からみると、利用者登録してあるだけで、図書貸し出しを全く受けていない場合もあれば、多数の本を借りているケースもある。

初歩的なテキストでは、関連は、その関連がつかないでいる2つのクラスを含めて、S+V+Oの形式を持っていると説明されることが多い。英語の場合には、確かに、「クラス+関連名+クラス」と並ぶが、語順の異なる日本語ではそのように並ばない。また、日本語では、「～が～を××する（S+V+O）」では無く、「～を～に××する」と言った表現も多い。そのため、日本では、関連名を動詞で書くことはまれであり、名詞やサ行変格活用動詞の語幹で命名することの方が多。その理由については、後ほど、第6章において分析する。

クラス図とは、対象世界を正確に認識する（モデル化する）ためのツールである。自然言語で描くより、はるかに正確なモデルが描かれる。特に、多重度の存在は、自然言語に比べて、モデルを精緻なものとする。注意がいるのは、クラス図では、関連（動詞）は、すべて「現在形」により描かれることである。例えば、自動車を住民が所有することを、「所有クラス」で表現したとする。この場合、当然、車を売ってしまえば、「所有クラス」は消えるべきであるが、

そのようなことは、考慮に入れない。具体的なインスタンスのレベルで、存在したり、存在しなかったりしても、それを、ひとつの「所有クラス」として、クラス図では表現するのである。クラス図では、対象世界に存在する「クラス（もの）」や「動作（こと）」が写し取られるが、時間の前後関係は無視されている。時間の前後関係を表示するのは、UML [2] で言えば、シーケンス図の役割である。即ち、概念レベルのクラス図は、対象世界の中にある「もの」「こと」を取り出すためのツールである。実装では、概念クラス図のレベルの設計では不十分となる。

図2は別のクラス図の例である。ある会社において、商品を仕入れ先に発注したとする。「仕入れ先に商品を発注する」となるので、関連名は「発注する」となる。しかし、発注は一瞬の出来事（アクション）であるが、実際の業務では、その情報が時間の経過とともに消えては困る。そこで、「発注する」と言う「こと」に対応する「発注」というクラスを作り、そこに発注年月日を書く。所謂「もの-こと-もの」パターンである [3]。この「こと」に相当するクラスに所属するオブジェクト（インスタンス）は、発注が重ねられる毎に次々と数が増えてゆく。言い換えると、要求分析段階のクラス図における「関連」は、現在形を用いて、分析対象世界に生じている「こと」を表現しているにすぎず、実装に際しては、当該クラスに、時間的な記録などを追加してゆく必要がある。

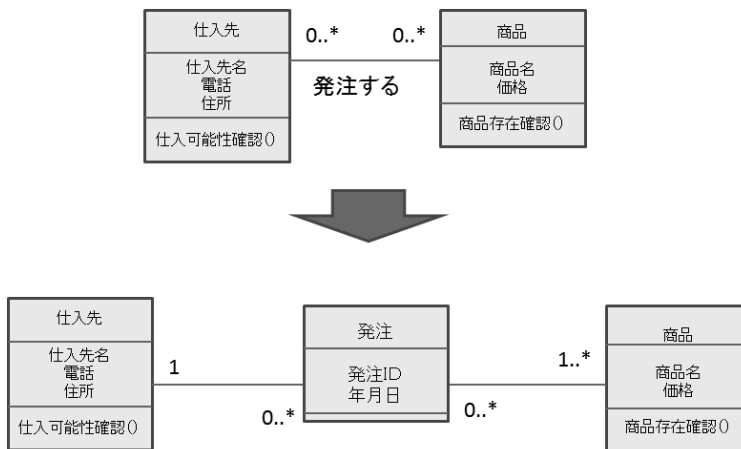


図2 別のクラス図の例

表1 認知文法における名詞・動詞と Object 指向の対応

品詞	認知文法における用法	Object 指向における要素
名詞	可算用法 (可算名詞) 非可算用法 (非可算名詞)	クラス名 (エンティティ名) 属性名
動詞	完了用法 (動作動詞) 未完了用法 (状態動詞)	メソッド (及び関連) (注1) 関連名

(注1) 動作を表す完了用法の動詞でも、その動作が現実社会に残るのであれば、現実社会をコピーであるクラス図 (ER 図) の側にも、「関連」として、その影響は残すべきである。

次章では、このクラス図を、認知言語学の観点から分析してゆく。

3. 認知文法から見た Object 指向

3.1 英語とクラス図 (ER 図)

認知言語学は、近年になって注目されてきた学問分野であり、G. レイコフの認知意味論と R.W. ラネカーの認知文法を中心に発展を遂げてきた学問分野である [7]。認知文法では、英単語のコアイメージによって、感覚的に英文を捉えてゆく [8]。前置詞のコアイメージについては、テレビの英語教育でも採用されている。以下に名詞と動詞に対する基本的な認知モデルを示す。内容は、今井隆夫の著作 [8] から引用している。

ラネカーは名詞の「可算用法」と「不可算用法」には、以下の2つのコアイメージを提示した。ただし、名詞自体 (単語) 毎に可算・不可算の区別があるわけではなく、文脈に依存して変化する。このため、「可算名詞」と呼ばずに、「可算用法」と名付けている。

- (1) 可算用法
 - ①区切りがあり、それ以上分割できない。
 - ②内部は不均質と捉えられる。
 - ③数を増やすには、複製する必要がある。
- (2) 不可算用法
 - ①区切りはなく、分割可能。
 - ②内部は均質と捉えられる。
 - ③伸縮自在

今井つつみ [4] に従えば、上記可算用法の名詞 (概念) こそが、「Object」だと言う事になる。例えば、「人」、「会社」は、Object として認識されるが、「水」や「砂」は、最初から

欧米人の Object 候補としては上がってこない。上記をみれば、クラスとは可算用法の名詞であり、クラスの属性とは、不可算用法の名詞が候補となる。「クラス」は集合と等価であるから、数えることができる名詞と、クラスとは同じである。要するに、クラスに等価なものを Object と言っている。

ER 図の提案者である Chen は、1976 年の ACM Transaction の有名な論文 [14] で、エンティティの例として、「人」、「会社」の他に「イベント」を挙げている。たとえば、「負債」、「発注」、「受注」等がクラスとなるか否かについて日本人が悩むのは、「Object とは『もの』だ」と言う翻訳文化的な固定概念に固執しているからであり、ER 図の発明者の頭の中では、例えば「Order」は、最初から Object (エンティティ) に見えていたことになる。

次に動詞に目を転じてみよう。ラネカーは動詞を以下のように分類している [8]。

- (3) 完了用法 (従来の動作動詞に対応)
 - ①時間における区切り (始まりと終わり) がある。
 - ②さまざまな動きのまとまりがひとつの動作になっている。
 - ③動作を繰り返すことができる。
- (4) 未完了用法 (従来の状態動詞に対応)
 - ①時間における区切り (始まりと終わり) がない。
 - ②時間上のどこを取っても同じという安心感がある。
 - ③区切りがない事態なので、動作の繰り返しはできない。

状態を表現する未完了用法 (状態動詞) については、「関連」として表現される可能性が高い。一方、動作を表現する完了用法 (動作動詞) はメソッドになる可能性が高い。ただし、メソッ

ドが対象世界の現実の中で永続的に作用している（当該動詞の作用の「結果」が残っていることを意味しているものであり、当該動詞の動作が永続することを意味しているものではない）のであれば、現実社会での永続している状態は、モデル側（クラス図、ER 図）にも反映すべきである。その場合、完了用法の動詞（動作動詞）であっても、「関連」としても表現するべきであろう。なお、完了用法の動詞は、始まりと終わりのある動詞である。シーケンス図では、ライフラインに、活性化している動作区間として描かれる。

上記の様に見れば、クラス図（ER 図）とは、英語の認知構造を、そのまま素直に写し取ったものである。名詞・動詞の種別と Object 指向との対応を表 1 に示しておく。永続的な関係は、クラス（エンティティ）間のポイントであり、そのリンクを辿って、クラス（エンティティ）を迅速に探すことを可能とする。その意味では、ヒアリング結果として得られた静的な関係を、関連としてインプリメントしておくことは、対象ビジネスにおける関連性を手繰る作業を効率化することが期待される。

図 3 は、英語の原著にある ER 図の例である。関連名は、受動態で表現されている。認知文法では、表現形式が異なれば意味が異なると思なしている。能動態と受動態では、意味が異なる。認知文法的には、動詞の過去完了は、「アクションが終わった状態」を意味する。そこで、同じアクションを表す動詞であっても、過去完了形を用いて、当該動作が終了した状態を表現している。

図 3 の中で、occupy は、占有するという意味である。この意味で用いられる時には、受動態が多いとされている。受動態の方が、状態をより強く意味している。これは、受動態では中心を占める be 動詞が後述する「状態動詞」であることが大きく寄与していると思われる。manage も受動態で利用されているため、状態の意味が強く出ている。図 3 における関連は、クラス間（エンティティ間）の静的な関係でなければならぬので、意図的に受け身で表現されていると見て良いであろう。

関連は、英語圏では、S+V+O 形式に対応するとされるが、周知のように、「V」は他動詞である必要はない。自動詞であっても、前置

詞と一緒にすれば、見掛け上、S+V+O 形式となる。受動態も「by」を含めて、フレーズとすれば V となる。英語の動詞には、日本語に比べて、いわゆる状態動詞（未完了用法）の種類は多い。S+V+O 形式は、多くの英語の動詞をそのまま写し取れるシンタックスである。「関連」とは、「たまたま、そのような動詞に相当する関係性が 2 つのクラスの間」に存在することを気付いたから描いたものではなく、対象世界に内在する静的な状態をダイレクトに写し取ったものである。

3.2 Is-a 階層と Has-a 階層

以上の議論では、英語の第 1 から第 3 文型を分析対象とした。しかし、英語では、「Give me chocolate!」の様に、動詞が 2 つの単語を取る場合がある。これが、第 4 文型、及び、第 5 文型である。この文型の認知構造についても、確認が必要である。認知文法では、1 つの動詞が 2 つの単語を取る場合、その 2 つの単語には、以下の 2 通りの意味しかないとする。

動詞 + 単語 1 + 単語 2 の場合について、

- (1) 単語 1 + have 動詞 + 単語 2
- (2) 単語 1 + be 動詞 + 単語 2

上記 (1) はいわゆる第 4 文型である。第 4 文型は、S+V+O+O である。例えば、「He gave me a present.」のような形である。この種の動詞には、give、pass、send、teach、tell、bring、lend、read、sell、show などがある。このコアイメージは明白である。第一引数に向かって、第二引数から、物かあるいは情報が移動している。結果として、物なり情報なりを、上位階層が保持するので、have 動詞と等しい。第 4 文型が実行された結果が、「Has-a 階層」であるとも言えよう。

次に、(2) は第 5 文型である。第 5 文型は S+V+O+OC の形式を持つ。第 5 文型には、例えば、「We call him Hide.」、 「She always makes me happy」などがある。コアイメージは明らかである。第一引数 = 第二引数である。つまり、be 動詞による代入である。何かを何かに定義づけたり、代入している。代入された側の第一引数がカバーする意味の空間の中に、第

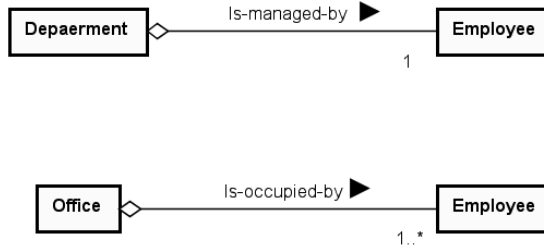


図3 英文原著における関連の例 (文献 [15] p.25, Fig.2.4 から描き直した.)

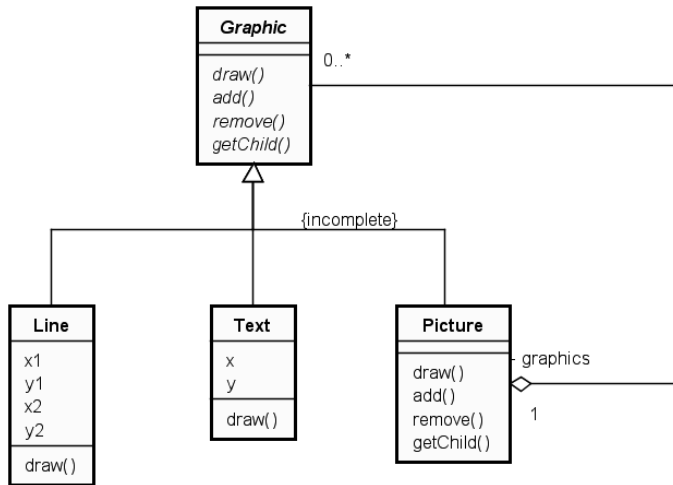


図4 Composite パターン [17]

二引数が入っているイメージである。即ち、第5文型は、「Is-a 階層」そのものである。

このように見てくると、「Has-a 階層」「Is-a 階層」は、「たまたまそのような階層がある」と言うようなものではない。第4文型/第5文型の動詞が実行された結果を表現するためのツールが「Has-a 階層」「Is-a 階層」であり、これによって、第1文型から第5文型までのすべての英語の文型を Object 指向がカバーしている。Object 指向は、英語の認知構造そのものである。

上記の様に「Is-a 階層」「Has-a 階層」をコアイメージで理解すると、欧米のテキストに描かれたクラス図の理解が容易になる。例えば、図4は、GOFによるデザインパターン [17] の中の、Composite パターンである。ここでは、Graphics という上位構造が、Line, Text, Picture と言ったクラスの「Is-a 階層」での上位となっ

ている。この場合は、インタフェースを統一するべく Graphics というクラスを定めて、Line, Text, Picture の上位に新たに定義したというニュアンスと思われる。また、Graphics が Picture の「Has-a 階層」なのは、Picture の部品として、Graphics を位置付けていることを意味している。

以上見て来た様に、クラス図 (クラス図) は、英語の動詞・名詞について、その認知構造を素直にそのまま写し取っている。英語ネイティブに取って見れば、自分が英語によって考えている事を、クラス図 (ER 図) に置き換えて表現する際に、難しさは少ないはずである。

最後に、英語テキスト (原文) とクラス図 (ER 図) との対応を見てみたい。以下の英文は海外の Web サイトにあった日本での生活について説明した文章である。英語はそのまま素直に、クラス図に展開できる。Pay に対して受け身が

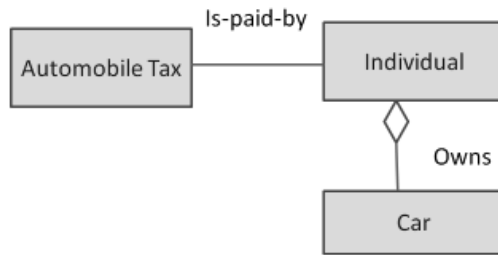


図5 英語による自動車税のモデル

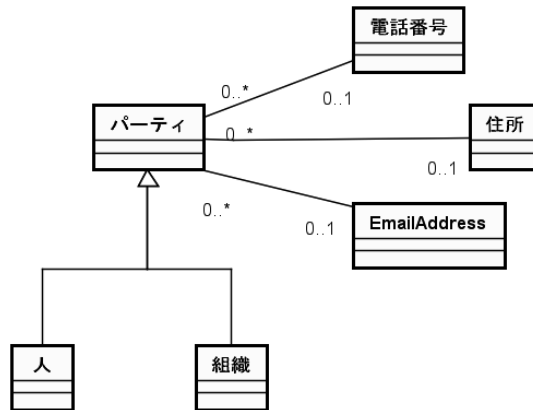


図6 パーティパターン [16]

用いられているのは、おそらく、状態を表現したいからであろう。クラス図(ER図)の形で、図5に示す。

(英語) A prefectural automobile tax is paid annually by individuals who own a car.

認知言語学では、S+V+Oの形は、SからOへ何らかの力が及んでいるイメージである。つまり、力の主体は、Sである。逆に言えば、S+V+Oの形で「関連」を張るのは力が及んでいる場合に限定するべきであろう。このサンプルでは、税に影響を与えるのは、所有関係そのものである。

4. 英語における階層構造

前章の議論で、クラス図が、英語の言語の構造そのものであることを示した。しかし、クラス図を考えると、欧米人の持っている特性について触れておかねばならない。一つは、欧米人の発想がトップダウンである点。そして、も

う一つは、「類似性」自体の認識の差の問題である。まず、最初に、「類似性」の認識について確認しておく。

4.1 Is-a 階層に関する別の見方

ニスベット [18] の西洋人と東洋人の認識の違いは良く知られている。ニスベットは、その著書 [18] の中で、2グループの花の絵を示した。そして、それとは別のあるサンプルがどちらの集団に近いかを問うた。その結果、西洋人はサンプルと同一の属性、即ち、「真っ直ぐな茎」をもった集団を近いとして選択する傾向があり、一方、東洋人は、そのような特定属性ではなく、なんとなく、家族的に同一の雰囲気をもつ集団を近いとして選ぶ傾向があることを示した。要約すれば、以下ようになる。

- 東洋人の類似性とは、家族性であって、全体として(雰囲氣的に)似ているものを「類似」として選択する傾向がある。

- これに対して、西洋人の類似性とは、ある属性が同一の値を取ることを重要視する。

上記傾向は、クラス図の世界では、より明確に現れる。図6は、アナリシスパターン [16] の Party パターンである。ここでは、「人」も「組織」も、属性として、「名前」「住所」「Email アドレス」を持っているという理由で、同一の「クラス」に入れられているとも理解できる。ただし、「法人」のように、組織体があたかも人間のよう振る舞うことは法律学ではあるわけであり、Party パターンの「組織」はそのようなものと理解することもできよう。

図7のデザインパターン [17] の Chain of Responsibility パターンの例である。ここでは、最上位には、「HelpHandler」クラスがある。HelpHandler クラスは、メソッドに HandleHelp () を持っている。たとえば、ヘルプを表示するメソッドであろう。図7を見ると、驚いたことに、HelpHandler の下に、「Application」が存在する。日本人の感覚では、「Is-a 階層」の下に来るものは、上部のクラスの特異なもの、部分的なものというイメージである。しかし、ここでは、HelpHandler の下に「Is-a 階層」で「応用プログラム」が存在する。ここで、「エッ？極めて意味の広い『応用プログラム』が『ヘルプハンドラー』と言う特殊なもの『子孫』だなんて！」と思っはいけないと思われる。む

しろ、「HandleHelp ()」という共通のメソッドを持っているかどうか、このパターンの興味の対象であり、その意味では、ヘルプ機能を持つ Application は、HelpHandler の特殊例だと言えるのであろう。但し、前節で述べたように、「Application を HelpHandler の一種と定義したから Is-a 階層」との側面も無視できない。

4.2 トップダウン特性の強さ

もう一点は、キリスト教圏における「トップダウン発想」である。これは、一神教であるキリスト教の影響を受けたものと想像されるが、本稿では、この議論には深入りしない。詳細は、著者の文献 [5] を参照されたい。

ここで、注意しなければならないのは、西洋人がごく事前に対象世界の中心にあるものをまず認識し、それから、枝葉に降りてゆく、「トップダウン型」の認識構造を持っていることである。例えば、増田 [19] は、西洋の肖像画では、中心に存在する人物像のみが鮮明であり、背景がぼけていることが多いのに対して、東洋の肖像画では、ほとんどの場合、人物のみではなくて、周囲の背景の細かい説明が行われていることを紹介している。おそらくは、東洋人の発想として、中心にある対象を認識する際に、かならず、周囲を眺めてから、全体の中で、中心の人物を認識する性向があるものと思われる。

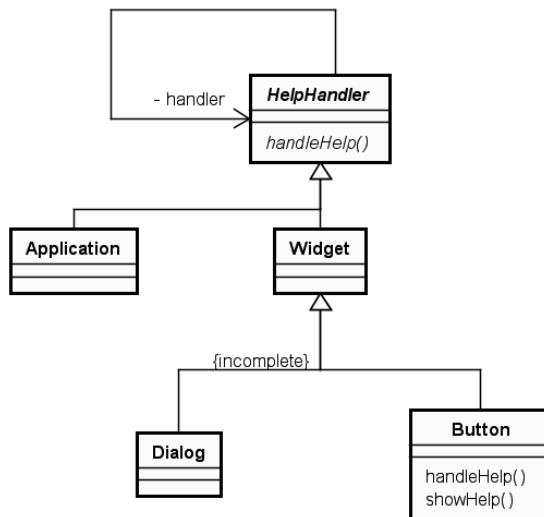


図7 Chain of Responsibility パターン [17]

このトップダウン発想が先にくる性質はクラス図を理解する上でも無視できない。前述した図4は、デザインパターン [17] の中でも最もよく議論される composite パターンである。このパターンは、日本的な考え方で言えば不自然である。text や picture の上位概念に、わざわざ、Graphic を持ってきている。そして、Picture の下位構造に、Graphic があるとしている。興味深いのは、Picture と Graphic を分けている点である。西洋人の認識では、トップダウンに、まず、全体を仕切る「Graphic」という概念があり、これによって統一的なインタフェースが規定される。そして、Graphic の下位クラスとして、text、Widget、そして、Picture が存在するとして、上位から下位へ、トップダウンに対象がモデル化される。

以上見て来たように、英語における「Is-a 階層」は、我々日本人が思っているような、「家族性」や「仲間」関係の上下関係ではない。「Has-a 階層」も、部品として下位クラスを持っているというより、物や情報の移動を意味していた。あくまで、英語における「Is-a 階層」「Has-a 階層」とはそう言ったものであることを認識して、クラス図を設計する必要がある。Has-a 階層も Is-a 階層も、トップダウンなツリー状構造を作り上げる。欧米人の発想が基本的にトップダウンであり [5]、また、Object を処理する言語プロセッサのパーザは、トップダウンなコンテキスト・フリー・グラマー (CFG) で成立している。プログラミング言語自体が階層的な divide and conquer のデータ構造と親和性が高い。Is-a

や Has-a の階層は、プログラミング言語自体の処理から行っても、自然なものではないだろうか。

4.3 小括

以上述べたように、英語の場合には、テキストをそのまま素直にクラス図 (ER 図) に変換できる可能性が高い。英語におけるクラス図は以下の様に小括できる。

- クラス (エンティティ) 間の「関連」は、本来は状態を表す動詞が対応している。一方、動作を表す動詞はメソッドとなる。関連には、S+V+O 型、及び Is-a 階層、Has-a 階層の3種類がある。Is-a 階層と Has-a 階層は、「たまたまそのようなリンクもある」という性質の関連ではなく、クラス図において、主要なリンク構造である。それは、第4文型、第5文型を表現する。言い換えれば、クラス図によって、英語の第1文型から第5文型までが表現する対象世界は、網羅的に表現される。
- プログラミング言語は、コンテキスト・フリー・グラマーで処理される構造をもち、本質的に、トップダウンな処理に適している。その意味では、Is-a 階層や Has-a 階層、とりわけ、Is-a 階層は、ソフトウェア設計上、望ましいリンクである。ポリモーフィズムが有効なのは、Is-a 階層である。

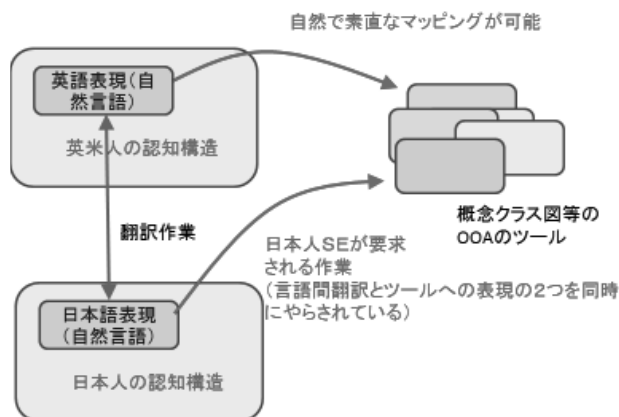


図8 テキスト記述からのクラス図生成

- 関連がS+V+Oで表現されるとき、OへはSから「力」が流れている。ある「O」に対して、ある動詞で表現される行為が続いている時、その動詞が直接に関連する最も強い「力」を持っているObjectを主語として選ぶべきである。なぜなら、その主語から「O」に対して関連を貼っておけば、ビジネス実現時において、強いパスの上流に位置するObjectから参照が来る可能性が高いことが期待される。

以上の議論では、英語とObject指向の関係論を論じた。問題は、日本語記述のクラス図化である。次章からは、日本語の持っている課題について考察する。

5. 日本語文のObjectへの変換

5.1 日本語の特性

日本語は、英語とはかなり異なった認知構造をもつ言語である。従って、日本語のテキストを如何にして上記の英語の認知構造であるクラス図に当てはめるかが問題となる。ヒアリング結果等である（日本語）テキストをクラス図に変換する作業は、日英翻訳に近い高度な作業とすることになる（図8）⁴。日本語と英語では、同じ内容を表現していても、着目している着眼点異なる。例えば、今井隆夫 [8] は以下のように説明している。

- 日本語では、「車間を維持しましょう」というように、車と車の間の距離を保つ行為ではなくて、その結果生じる状態が記述される。これに対して、英語では、「Don't tailgate!」と行為が言語化される。
- 日本語は「お先にどうぞ」というように他者の行為が言語化されているのに対して、英語では、「After you.」と相手の後から通過する自分の行為が言語化される。

- 「ここはどこ」では日本語では自分のいる場所が言語化されているのに対して、英語では、「Where am I?」と自分の存在が言語化される。

日本語では、環境や結果の状態、そして他人が言語化されるのに対して、英語では、人間、自分を言語化する。また、英語では、人以外の「Object」が主語となって、主体的に対象となる目的語に影響を与えることも多い。これに対して、日本語では非生物が主語となることを避ける。別の例文を見てみたい。日本語表現は、自治体の自動車税の説明にしばしば出現する表現である。基本的には、国の地方税法・第145条の表現に準じている。英語は、前述の例文である。

（日本語） 自動車を所有している住民には、自動車税が課税される。

（英語） A prefectural automobile tax is paid annually by individuals who own a car.

【分析結果1：】日本語記述では、状況や環境は書かれているが、業務を担当する「アクター」や「アクターの行為」が陽には書かれないことが多い。テキストから名詞と動詞を取り出すプロセスでは、頭の中でアクターを探してリライトする必要がある。したがって、モデル化に際して、クラス図（ER図）だけで作業することが妥当かどうかは疑問が残る。日本人には分かりやすい、動詞を中心とするシーケンス図を併用するべきであろう。

5.2 日本語動詞の種別

次に日本語の動詞の種別を確認したい。日本語の動詞は以下の4種類あるとされる [20]。但し、文脈依存であり、その区別は厳密ではない。

- 【状態動詞】 動作・作用ではなく、状態を表す動詞であり、時間を超越した概念を表す。「～ている」を付加すると不自然であ

⁴ 機械翻訳の研究者の間では、日英自動翻訳は英日自動翻訳に比して、難しいとされている。プリエディット無しの日英機械翻訳は、未だ道遠という状況であり、とりわけ、重文・複文などのひとつの文の中に動詞が2個以上ある表現、並列文、などの処理は困難とされる。

る。「ある」「いる」「知らない」などがある。日本語では状態動詞の数は少ない⁵。

- 【継続動詞】人間の動作、あるいは、自然現象を表す動詞である。「歌う」「見る」「散る」等があり、その数は多い。「～ている」をつけることにより、現在進行形が表現される。「見ている」は、現に今、(桜やTVを)見ていることになる。
- 【瞬間動詞】「死ぬ」の様に、ある瞬間の行為について述べた動詞である。「知る」「消える」「触る」などがある。継続動詞に似ているが、瞬間動詞に「～ている」をつけると、現在進行形の状態ではなくて、瞬間動詞の動作が終わってしまった状態を意味する。
- 【第4種の動詞】「似ている」「そびえている」など、動詞ではあるが動作そのものが考えられず、単なるもとの状態を表す。

まず最初に、「状態動詞」は種類が少なく、「ある」「いる」位しかない。関連で表現されるべきである。但し、状態動詞は、継続動詞や瞬間動詞に「～ている」「～である」を付して表現されるケースが多いと思われる。「雇用する」は動作に見えるが、「雇用している」とすれば、英語の受動態のように、状態を表すニュアンスが強くなる。

継続動詞に目を転じる。継続動詞に「～ている」が付いているケースは、現在進行形であり、要求分析のヒアリング結果には想定し難い。継続動詞が単独で用いられている場合は、本来はメソッドとして実現されるべきであろう。ただし、記録を残す必要がある場合や、そのメソッドが他のクラスを関連として持っている場合には、関連として表現することも考えられる。

瞬間動詞も、基本的には、メソッドとしたところである。しかし、「～している」型で表現すると、瞬間動詞は、後に影響が残る。このため、記録を残す意味で、関連として作成する必要が生じることが多いと思われる。例文を見

てみたい

(日本語) 商品を納入元に発注する。

上記の例において、「発注している」というのは、現在進行形ではなくて、動作の継続(「発注する」は瞬間動詞)と見るべきだろう。「あの商品は発注しているヨ」などの表現がしばしば用いられる。「発注している」あるいは、短縮形「発注」を関連名として利用すべきである。

【分析結果2】動詞が瞬間動詞であり、その動詞の効果自体が継続し続ける場合には、英語の状態動詞と同じ「関連名」として表現すべきである。即ち、動詞に「～ている」が着いていない場合であっても、「～ている」を付して、動詞の種類を調べるとともに、当該動詞の作用が継続する場合には、瞬間動詞であっても関連として表現すべきである。

日本語の場合には、動詞の種別が決まったからと言って、ストレートにメソッドか関連かを切り分けることは数少ない状態動詞を除けば、難しそうである。本質的にどのような行為が行われ、情報が残ってゆくかどうかを考えないと、日本語テキストをクラス図にすることはできない。

6. 関連名に関する考察

本章では、以上の議論を踏まえて、クラス図(ER図)における日本語の関連名について考察する。

6.1 日本語における関連名

本章では、関連名を議論の対象とするので、メソッドについては除外したクラス図(実効的にはER図)を用いて議論する。ER図・クラス図の関連は、英語圏では「S+V+O」の関係で表現され、「V」が関連名となる[22]。たとえば、以下のようになる。

⁵ 英語では、状態動詞の数は多い。例えば、「have」「live」「know」などの日常的な動詞は、いずれも状態動詞である。これら英語の状態動詞は、日本語では、後述の瞬間動詞である。「～ている」をつけないと、状態動詞にはならない。

Student! → 〈Enrolls〉 → Course

左右がクラス名（エンティティ名）、「〈〉」で囲ったものが関連名である。ただし、1対多の関係の場合は、動詞は三人称単数となり、多対多の関係の場合には、複数形に対する動詞となる。前置詞なしに目的語を持つため動詞は他動詞である。英語の場合、仮に、自動詞であっても、前置詞を追加すれば、S+V（フレーズ）+Oの関係は維持される。例えば、

Student → 〈belongsto〉 → Club

である。

英語においてS+V+OがObjectとObjectの関連に対応づけられるのは理由がある。認知言語学（英語の）では、SからOへは、「エネルギーが流れてゆく」関係にあるとする。そこには、SとOが結果として関連ができてしまったというような消極的な意味あいではなく、積極的に何等かの関与が行われているイメージである。例えば会社が人を雇用する（Employ）とか、学生が科目を履修する（Take）には積極的な主体からの働きかけがある。それに比べると、日本語の「(会社が商品)を」発注する」などは、そのような積極的な関与ではない。商品が自分で、相手の会社に発注するわけではない。「発注する」は、「もの-こと-もの」パタンの典型であるが、2つの「もの（クラスまたはエンティティ）」の間には、エネルギーの伝搬はないことに注意されたい。

一方、英語とは異なり、日本語では、関連名として、素直に動詞を付けることが難しい。以

下は、関連の例をWebで検索して、多数ヒットした例である。

- ・クラス（エンティティ）「学生」「科目」間受講、履修、履修する、
- ・クラス（エンティティ）「学生（生徒）」「クラス（学級）」間所属、所属する

以下のことが指摘される。

- ・関連名を書いても、「学生・受講・科目」あるいは「学生・受講する・科目」と単語が並ぶだけであり、文章にはならない。読み下すには、助詞を自分で補い、かつ、語順を入れ替える必要がある。結果的に、クラス図・ER図の理解を助けるドキュメントにはなりがたい（だから、関連名は重要視されてない）。
- ・英語の他動詞の定義を準用すれば、関連名を読みくだすときには、一方のクラスに目的格の格助詞「を」がつくはずである。しかし、上記の「所属」「所属する」は「を」ではなく「に」がつく。格助詞が一定しない。これでは読み手は、助詞を補おうにも、どんな助詞を補って良いか分からない。
- ・日本語の「こと」が変化してクラス（エンティティ）となる「もの-こと-もの」パターンが多く存在するため、関連名=クラス（エンティティ名）となる。結果的に関連名は名詞である。名詞と言っても、サ行変格活用動詞の語幹が多い。ただし、語幹として「履修」のような特殊な名詞を利用した場合にはまだ良いが、「選択」のように一般的な名詞を使うと、名詞だけ読んでも、

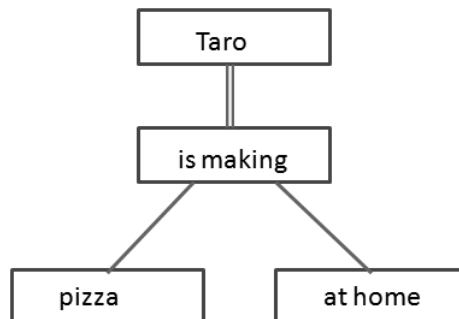


図9 英語におけるクリスマスツリー型文構造（金谷の文献 [21] から金田が描き直したもの）

意味がわからないので注意が必要である。

- 自動詞と他動詞の区別を持たない日本語動詞でも、使役形は存在する。「食べる」に対する「食べさせる」である。使役も他者に働きかける動詞であり、意味的に他動詞であり、関連名に使ってよい。しかし、S+V+Oのみが正しいと単純に考えていると、使役形を使うことに心理的抵抗感がある。

では、日本語では関連名はどのように記述すべきなのだろうか。以下、対策を探るため、日本語文法理論から分析してゆく。

6.2 三上章の日本語文法

現在、学校で教えられている学校文法は、橋本進吉による橋本文法である。橋本文法は、主部+述部から構成される英語文法をそのまま日本語にあてはめた文法である [24] [25]。「舶来崇拜主義」の一つの具現化であるが、反面、日本語文法としては不十分であるとされ、特に、日本語を母語としない外国人への日本語教育における橋本文法の利用は困難とされる [21]。

そこで、本稿では、現在の日本語学の中核とされる「三上章による日本語文法 [24] [25]」に注目する。特に、三上文法を継承し、他動詞・自動詞の解釈に独自の理論を構築した金谷武洋 [21] の理論に着目する。

6.3 三上章の「盆栽構造」

三上は、「象は鼻が長い [24]」などの著作の中で、欧米の主述関係を日本語文法としてそのまま持ち込んだ学校文法の問題点を指摘している。詳細は、三上の文献 [24] [25] を参照していただかなければならない。金谷武洋の説明 [21] に従って、三上文法の簡単な紹介を行う。

金谷は英語と日本語の違いを、図9、図10のような「クリスマスツリー型」「盆栽型」の図で説明できるとした [21]。英語の場合は、文章を、まず、主部（主語）と述部という2つのパートに分割する。主部の中心には名詞が存在する。一方、述部の中心には動詞が存在し、動詞は「格フレーム」と呼ばれる動詞を中心とする幾つかの格情報（補語）をもつ。図9に示すように、補語の順番は厳密であり、「pizza」と「at home」の順序を勝手に入れ替えることはできない。

これに対して、日本語は、用言を中心とした、逆ピラミッド構成である（図10）。まず、係助詞「は」は特別な役割をもっている。係助詞「は」は、主格の格助詞「が」を兼務することもある。しかし、他の格助詞とは異なった、大きな力を持っている。金谷は、これを「スーパー係助詞」と呼ぶ。「は」は、コンマやピリオドを越えて、後の文にまで影響を与える⁶。図10の例では、「は」は文章全体の「主題」を宣言している。即ち

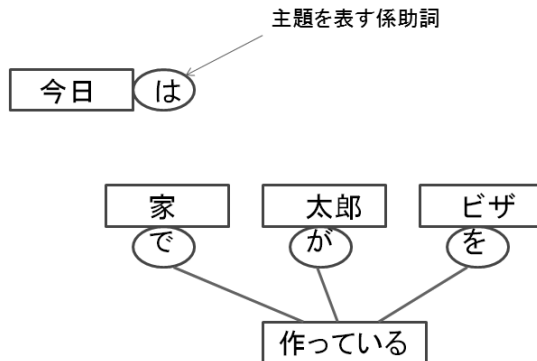


図10 日本語の盆栽構造（金谷の文献 [21] から金田が一部加筆して描き直した）

⁶ 例えば、「吾輩は猫である」の出だしは、有名な以下の文である。「吾輩は」という主題が延々と後の文章にかかっている。「吾輩(わがはい)は猫である。名前はまだ無い。どこで生れたかんと見当(けんとう)がつかぬ。何でも薄暗いじめじめした所でニャーニャー泣いていた事だけは記憶している。」

～は、XXXXXX。

という文章では、「は」はこの文章の主題を示し、残った「XXXXXX」は「解説」である。

図 10 は、「今日は、家で太郎がピザを作っている」という文である。最初に、この文章の主題が「今日」であることが宣言される。「太郎が家でピザを作っている」は、この「今日」の「解説」である。解説部分の主語は、「太郎」である。しかし、日本語の主語の「太郎」(が)は、英語の主語(図9)のような全体支配権は持たない。他の「で」格(場所格)、「を」格(目的格)と同等である。それが証拠に、図 10 の3つの補語は、順番をどう入れ替えても、文意には影響がない。日本語としての自然さも壊れない。一方、「は」係助詞を最後に持ってきた「家で太郎がピザを今日は作っている」は不自然である。なお、三上文法によれば、日本語の基本文型は、以下の3種類しかない。基本的に、用言中心の文型であり、主語は文脈で決まるものであり、省略しているわけではない。

- 名詞文：名詞+「da」 (例：赤ちゃんだ)
- 形容詞文：形容詞 (例：美しい)
- 動詞文：動詞 (例：泣いた)

主語は「省略できる」とするのは、欧米の文法に曇った目で眺めた日本語である。日本語には、欧米の文法理論で言うところの、即ち、述部と対抗する重さをもった主部など存在しない。例えば、「商品を仕入れ先に発注する」という文は、「顧客は商品を発注する」に比して、日常語としてより自然に感じる。「商品を仕入れ先に発注する」においては、英語では、発注主体(主語)を明確に書かねばならない。欧米流の主語

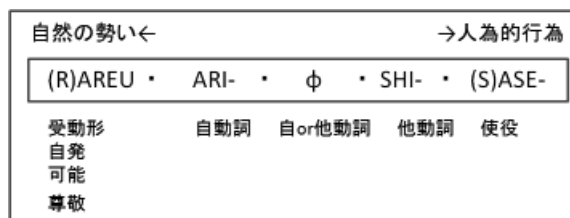
は、この日本語文では存在しないが、省略されているわけではない。この例からも分かるように、日本語として自然な文章を S+V+O 形式の関連名として表現することは(文法的には間違いではないにせよ)不自然なものになり得る。

次に問題となるのは、S+V+O の V が他動詞であることである。他動詞(あるいは前置詞付の自動詞)だから、前置詞がクラス(エンティティ)の前に要らないので、簡単に関連名を読みみくさせる。しかし、日本語ではそうはゆかない。「目的語をとれば(即ち『を』格を持てば)他動詞で、それ以外は自動詞」というわけにはゆかない。

図 11 は、金谷による、日本語における、他動詞・自動詞を区別するモデルである。日本語動詞を、連用形、即ち用言を中止した時の語尾の形で分類している。金谷は、1) そもそも、日本語動詞は自動詞と他動詞の区別をもつとは限らず、自動詞と他動詞の区別がないものが多い(例えば、「飲む」「食べる」は「を」格を取るが自動詞)。2) 日本語では、自動詞は「自然の勢い」を示すものであり、他動詞は「人為的行為」を示す、とした[21]。そして、図 11 に示すように、受動形から使役までは、一連のスパンの中にあり、他動詞の連用形には「SHI」がつくことを論じた。連用形が「SHI」即ち、原型「SU」は、「SURU」の古語である。従って、サ行変格動詞は基本的に他動詞となる。ただし、だからといって、「××する」がすべて他動詞とも言えない。日本語の他動詞・自動詞は、構文からは定め得ず、意味的に判定するしかない。

6.4 三上文法から見た関連の分析

以上の準備のもと、関連について再検討して



(動詞の活用は「連用形」であることに注意)

図 11 金谷による受動態から使役までの動詞の分類

みたい。例えば、以下の2つの文章を考える。

- ・「顧客は商品を注文する」
- ・「仕入れ先に商品を発注する」

上記の2つの文章は、欧米流の「主語」を持っているか否かとの観点から見れば、全く異なっている。前者の（欧米流の）主語は「顧客」である。しかし、後者の（欧米流の）主語は分からない。発注した主体が、担当者なのか、コンピュータなのか分からない。分からないのは主語が（存在するが）省略されているわけではない。日本語では、主語などどうでも良いから書いていない。上記の2つの文章で、どちらが日常日本語文として自然かは自明であろう。最初の文は、むしろ、「顧客から商品を受注する」とすべきであろう。この文章で、「受注する」の主語を探すと、会社や担当者になる。

なお、上記の「顧客」と「商品」の関連は、多対多である。リレーショナルモデルではテーブルにするためには、間に別のテーブル（クラス）を挿入して、1対多、多対1に変換する。「顧客が商品を注文する」については、欧米流であれば、関連名をそのままテーブル（クラス）名として「Order」が妥当であろう。「Customer → Orders → Goods」となる。ここには何らの不自然さもない。Orderは（目的語を取れば）他動詞であり、本来、名詞でもある。

日本語の方はそうはゆかない。欧米流の主語主義で行けば、

顧客 → 〈注文〉 → 商品

として、「注文テーブル（クラス）」とでもしたいところである。しかし、第一に語順を入れ替

えて、品詞を補わないと文章としては成立しない。更に、どうも、モデルとしては、「注文」が不自然である。フィーリングとしては、「受注」としたい。ビジネスを担当する会社の立場でモデリングしているからである。しかし、それでは、主語+動詞+目的語の関係が崩れる。

「自然な」関連の意味を表現する日本語文は、しばしば、S+V+Oパターンには当てはまらない。この「顧客から商品を受注する」という自然な文の場合には、実際には、「会社は」という（英語流でいうところの）「主語」が隠れている。したがって、動詞「受注する」を使う限り、クラス（エンティティ）「顧客」「商品」の間に置くのは、心苦しい。本来は、「注文する」が正しい。だから、「受注」とかいう名詞にして、「顧客」と「商品」という2つのクラス（エンティティ）には「受注」関係がある、と納得させるしかないのではないか。これは、主語のはずだった「顧客」クラス（エンティティ）の意思ではなく、モデリング対象である企業から見た用語である。「顧客が商品を注文する」の背後に「会社は商品を受注する」という別の説明が存在している。

「会社は」の「は」金谷の言う「スーパー係助詞」である。日本語の場合には、むしろ、このような主題を決定づけるフレーズが隠れていることを、陽に認識して、関連名について考えるべきである。例えば、英語圏の「顧客が商品を注文する」という視点よりは、「顧客から商品を受注する」のほうが自然である。そうなると、2つのクラス（エンティティ）間の関係を関連名として与える場合に、どちらかのクラス（エンティティ）が主語となることはできない。無理に主語にすると、企業という主題（視点）で分析するケースとしては不自然になる。S+V+O

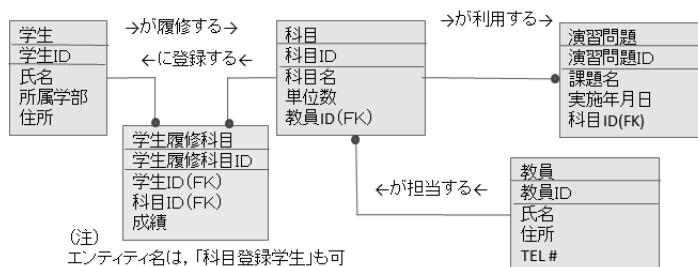


図 12 日本語に適した「関連名」の付与

から離れて、モデリング対象である企業の立場から、その2つのクラス（エンティティ）間に存在する「機能的な関係」を書いた方が素直であろう。少なくとも、関連名を全体的な視点を統一して理解できる。主語（主題）を「会社」とすると、「関連」には一定以上の粒度が要求される。ビジネスレベルの視点で用語が見つからなければ、モデリングから自動的に排除される。一方、「担当者は」や「詳細業務規則では」と主題を宣言すれば、分析粒度はおのずから規制される。

6.5 日本人に自然な関連名

本節では、英語における S+V+O のように、日本語でも自然な関連名を作れないのか確認する。例えば、「AがBをCする」という文章は、日本語では、そのまま「AがCするB」と名詞止め文（連体修飾節+名詞）に書き換えることができる。図12はそのような例である。←の意味等は説明を要しないと思われる。多重度が1対多の場合、FK（「フォレイン・キー」、「外部キー」とも呼ぶ。あるクラスから、他のクラスをポインティングしている属性である）は、「多」側のクラスの属性となることは自明である。図12は、読みやすいと思われまいだろうか。なお、図12は、主格「が」と目的格「を」のみを利用しているが、「に」「から」などを利用して問題ない。また、多対多のために新たに作ったクラスは、関連の説明文をそのまま複合名詞化した。しかし、前述したように、「動詞の語幹+単位（と言う言葉）」としても良い。

なぜ、最初からこうしなかったのか、一見すると不思議である。ただし、使われないのには、一定の理由があると思われる。例えば、「昨日太郎が見かけた男に会った」では、連体修飾節「昨日太郎が見かけた」を省略すると、意味がおかしくなる。「男に会った」では、何が言いたいかわからない。これは、「限定的連体修飾節」と呼ばれる[26]。一方、「疲れ切った太郎はヘタリ込んだ」では、「太郎はヘタリ込んだ」が主題であり、「疲れ切った」は解説である。これを「非限定的連体修飾節」と呼ぶ。

「学生が履修する科目」と言った時、「科目」は一般的な名詞であるから、限定的連体修飾節に見える。しかし、科目エンティティ（クラス）

は一般的なものであり、限定的に連体修飾することには意味的には違和感がある。学生が選択していない科目だってあり得る。この限定性は、あくまでも、ここで議論している関連に関するものである。それにもかかわらず、この日本語は、視点が最後に、主名詞「科目」に流れて行く。ここに不自然さがあるように思われる。図12のような表現方法は、助詞を補う必要がないので、セルフドキュメント性を有する。しかし、日本語としては、文末のクラス（名詞）を強く限定するように読める。関連が記述する対象は、あくまでも動詞の筈なので、読み易い「AがBするC」という表現が避けられ、「Bする」や「B」が採用されて来たのであろう。

以上のように見てくると、日本語関連名については、以下のような理解をすべきと考える。

1. 日本語では、2つのクラス（エンティティ）間に動詞による関連名を付与しても、助詞が定まらないので、読みこなせない。そうであるなら、むしろ「会社は」というような「主題」を視点として設定し、その前提で、適切な動詞（あるいは名詞）を選択すべきである。この場合、主題の粒度（例えば、会社が主題か、担当者が主題か）によって、取り出されるクラスの粒度をコントロールできる可能性がある。これは、英語とは異なる日本語のメリットである。
2. もともと、日本語には状態動詞が少ないので、「～している」という表現を用いないと状態動詞化は難しい。しかし、これでは、長すぎるので、「している」を取り去り、サ行変格活用動詞の語幹だけで、関連名を表現するのは、現実的である。言い換えると、「している」を補って関連名を理解することが望ましいケースが多いと予想される。結果的に、このことは、「ものーことーもの」パタンの形、即ち、「アクションの記録」の形で、関連をクラス化（エンティティ化）するケースが多いことを意味していると思われる。
3. 英語では、S+V+O パタンは、本来、「エネルギーがSからOに行く」コアイメージがある。英語の場合には、行為の主体で

あるSがクラス的一方である可能性が高い。しかし、日本語の場合には、上記(1)のような意味で、「関連が張られたクラス間に関連があることは事実」だとしても、それが、因果関係を必ずしも表現していないケースが生じやすいと推定される。例えば、S + V + O で言えば、「自治体は住民に自動車税を課税する」という表現を考える。「自治体は」の助詞「は」が係助詞か否かのニュアンスは微妙であるが、因果関係の主体というイメージは薄い。あくまでも、課税原因は、「所有しているという事実」である。日本語の場合には、関連を張る前に、それが、結果として関係性が生じるのか、本来の因果関係としてエネルギーがクラス間に伝わっているかは十分に吟味する必要がある。

以上見てきたように、英語ではS + V + Oの形で状態動詞を関連として描くことで、ある程度、因果関係は担保される可能性が高い。しかし、日本語の場合、主題を表す「主語」は通常は別にある。関連が示すクラス間の関係は、S + V + Oの形を取らないものが多くなる。この場合、1) (英文のS + V + Oと同様に) 本質

的な「エネルギー伝搬」「情報伝搬」の関係性がクラス間に存在するののか、あるいは、2) クラスは相互に結果的に関連しているだけにすぎないのか(例：参照属性のような場合は、対象世界の因果関係を見て、意味的に判断せざるを得ない。日英翻訳のように、一度、日本語の世界で、主語となり得る「もの」を探し、因果関係を確認しなければならないことを意味している。その意味でも、クラス図の作成は、日英翻訳に近いと考えている。

7. 第3正規形に基づく静的モデル

7.1 概念データモデリングの静的モデル

以上の議論から、次に、MASP アソシエーションによって提案された「概念データモデリング」の「静的モデル」を分析したい。概念データモデリングは多数のモデルを利用するが、本稿では、「静的モデル」「動的モデル」そして「組織間連携モデル」に着目する。概念データモデリングについては、文献[1]を参照されたい。ただし、この文献[1]には、本質論は記載されているが、細かい手続きは掲載されていない。概

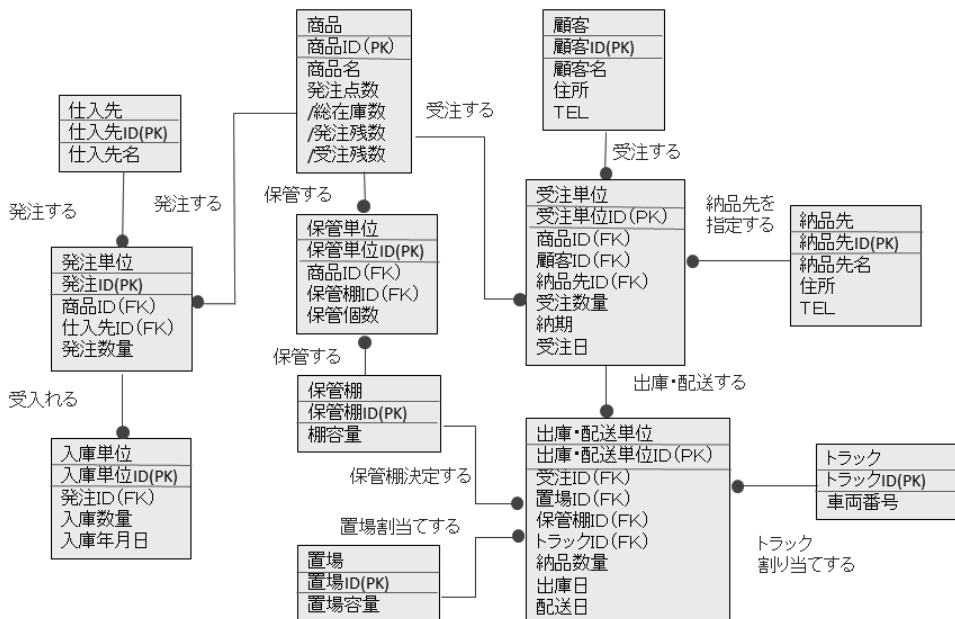


図 13 静的モデルの例

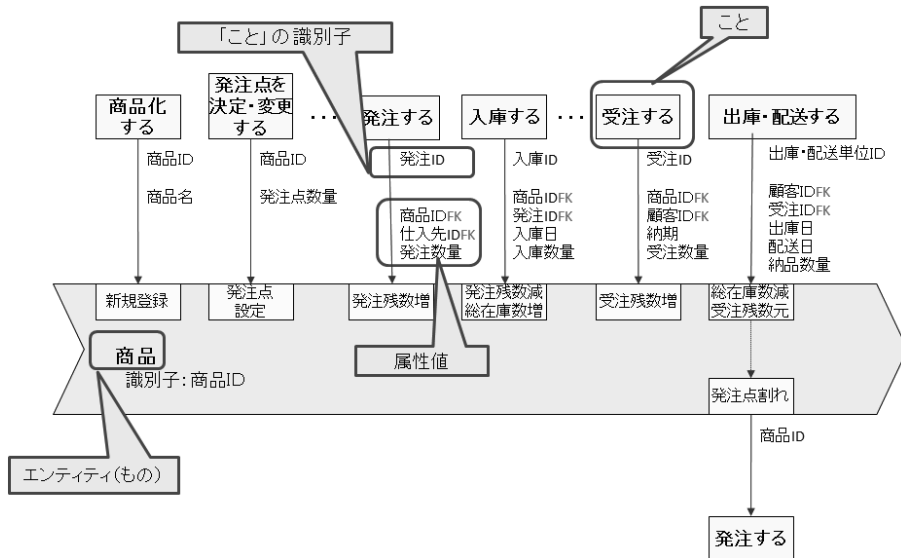


図 14 動的モデルの例

念データモデリングの概要は、拙稿にも紹介している [13] [12]。概念データモデリングのモデルの中で、静的モデルは、対象ビジネスドメインに存在する「要のもの」と「要のこと」を取り出すツールである。概念クラス図であるため、関連名は動詞で描くが、すべて現在形である。従って、その関連が時間の経過とともに消滅するようなケースについては、この段階では考えていない。静的モデルでは、時間関係は無視されている。対象ビジネスにおけるエンティティ（概念データモデリングにおける静的モデルは、クラス図よりも、メソッドを持たない ER 図に近い。このため、本章では、あえて、「クラス」と呼ばずに「エンティティ」と呼ぶ。）とその間の関連から構成される。エンティティの取り出しには、以下の基本ルールが適用される。

- 対象ビジネスに存在する「もの」の中で、属性値が書き換えられる「もの」のみを取り出す⁷。たとえば、組織、担当者、書類などは、データが通過してゆく可能性があるが、当該データ自体が、そのエンティティの属性ではないので、モデリングの対象と

はならない。これによって、要（かなめ）のエンティティを取り出す。

- エンティティは、当該エンティティがこの世に出現したときに必然的に値が付与される属性であって、当該エンティティが存続する限り（値は変更されるかもしれないが）その属性値が存在し、そのエンティティが消えるとともに、世の中から、その属性が消えてなくなるような属性のみを属性とする。また、プライマリー・キーに対しては、そのエンティティ中の全属性は、Codd の第 3 正規形をなす必要がある。

MASP の概念データモデリング手法における静的モデルは、ドメインのユーザ（業務担当者）も加わったワークショップに適用できるように考えられている。しかし、そのことが、あいまい性を持つ原因になっている面を否定できない。そこで、本稿では、以下のように、関係モデルデータベースの初歩的知識を前提とした表現方法を用いる。図 13 は、静的モデルの例である。このモデルを作成するに際しては、以

⁷ レイコフは、他動的事態には、少なくとも 2 つの参加者が関係し、その一方が動作主 (Agent)、もう一方が被動作主 (Patient) と言う意味役割を担い、動作主がエネルギーを被動作主に与えることで、その位置・状態が変化する場合が、プロトタイプであるとした [7]。ここで、プロトタイプとは、対象を認識してカテゴリー化する際に、それを核として、その周りに様々な成員を位置づけることで、全体を構造化するものを言う。属性値が書き換えられるエンティティのみを扱う事は、認知言語学の立場からも支持されると思われる。

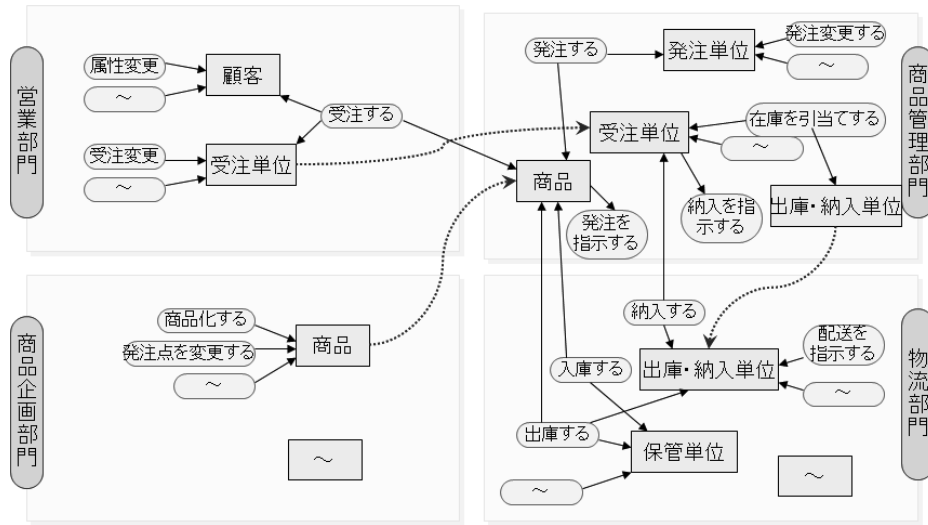


図 15 組織間連携モデルの例

下の原則に従っている。

- 属性値は、その属性が存在するのに最もふさわしいエンティティにのみ記録される。例えば、「商品名」は、商品エンティティの属性である。現実には、仕入先に発注したり、お客さんの注文を受けたときに、必ず商品名は明示されるはずである。しかし、このような場合でも、必要であれば、FK（外部キー）によって次々とエンティティを辿り、最も根源的なエンティティから、データを持ってきて表示すれば良い。
- 多対多の関係は、存在しない。すべて、エンティティ間に別のエンティティを設定して、1対多+多対1の関係で表現する。図13でいえば、「受注単位」「発注単位」などがその新設されたエンティティに相当する。P.コードとE.ヨードンは、「記憶される事象」を中間に作れといている [27]。いわゆる「もの」—「こと」—「もの」パターン [3] である。静的モデルは、「もの」—「こと」—「もの」パターンを多用している。ただし、概念データモデリングの紹介でよく利用される「受注品」のようなエンティティ名は図12では用いていない。後述の組織間連携図に流すために、ひとつは、このような具体的なイメージの湧きやすい

「もの」の名称が付与されていると思われるが、本稿では、「××単位」を用いている。ただし、前章までの議論の通り、「受注」としても問題は無い。

本稿では、「関連名」を問題としている。クラス図では、主題の係助詞「は」を想定して、粒度のコントロールをすべきことを論じた。以下のような種々のパターンが考えられる。

会社は、(エンティティ1)を、(エンティティ2)に××する。

会社は、(エンティティ1)を、(エンティティ2)から××する。

会社は、(エンティティ1)を、(エンティティ2)として××する。

「××する」としては、使役動詞も歓迎される。ただし、上記3種類で限定されるという意味ではない。図13でも、原則的には、関連名はこのパターンで作成されている。これは、あくまで、主語である企業の立場から、何が起きているかという「こと」を表現している。静的モデルは、「要のもの」とともに、その「もの」が最初に作られた際の「要のこと」、即ち、エンティティの誕生や、属性値の書き換えが生じた「こと」をリストアップする手段である。

7.2 動的モデルと組織間連携モデル

図 14 は動的モデルの例である。説明は要しないと思われる。ひとつのエンティティの状態変化を左から右に描いている。最上部に書かれたのがビジネスに生じた「こと」の名称であり、それに伴って、エンティティの属性が一部書き換えられる。下向きの矢印は、このエンティティの内部状態がある条件を満足した際に、他のエンティティに対して、なんらかのアクションが引き起こされることを示している。

静的モデル(図 13)を描く際には、図 14 を参照しつつ進めなければならない。日本語では、対象世界の記述(テキスト)な説明の中に、アクターが陽に出てこないことがあるからである。また、日本語では、無生物が主語になることは比較的少ない。これに対して、Object 指向では、そこに情報の流れがあり、依存関係があるのであれば、「もの」から「もの」に情報が流れる。そして、静的モデルでは、情報の流れの中にあっても、情報がスルーするだけの、たとえば、帳票、画面、そして、担当者といったエンティティは、最初から分析に含めない。これらの分析作業は、日本語の特性としても、静的モデルだけを見ては不十分であり、動的モデルにおいて、隠れたアクターを探し、アクター間の情報の流れを分析することが求められる。その意味で、後述の組織間連携モデルを

含めて、静的モデル、動的モデルの間で、相互チェックを繰り返しながら、モデル構築を進める必要がある。

図 15 は組織間連携モデルの例である。組織間連携モデルでは、「もの」であるエンティティの各組織の管轄権限を明確化する。「もの」自体、あるいは、そのデータが組織間をまたがって流れてゆく。ビジネスの視点から見た「こと」は種々の組織にばらまかれる。そして、「こと」が関係したデータは、もっとも根源的なソースである属性値を記述したエンティティから FK を使って持って来るしかない。それが、「こと」から「もの」への実線矢印である。

8. 考察と課題

本稿では、認知文法の知見を活用しながら、Object 指向について日本語でモデリングする際の課題について分析した。なお、認知文法・言語学と Object 指向については、すでに、詳細な議論が提起されている [32] [31]。それらとは異なり、本稿の視点は、日英翻訳として、モデリングを捉えた点に特徴があると考えている。但し、クラス図の作成には、日英翻訳と同じような作業が必要であるとの本稿の主張は、今後、定量的な評価によって確認されなければならない。

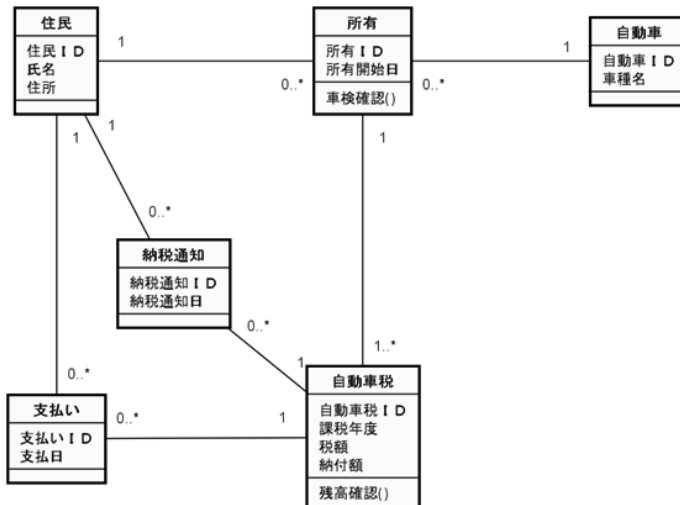


図 16 自動車税のモデル (日本語)

最後に関連と派生属性との関係について確認する。図 16 は、自動車税をモデル化した例である。ここで、納税通知書の送付、自動車税の支払いは、1回で終わるとは限らず、また、納税通知書の送付は、自動車税と住民との関連したイベント、自動車税の支払いは、住民と自動車税との関連したイベントなので、それぞれ、関連がクラス化している。本来、関連は、一種のポインタである。それがクラス化した「納税通知」クラスが「自動車税」と「住民」の間に設定されている。納税通知は、この2つに意味的に関連するからである。ただし、この納税通知書クラスについて言えば、そこには、「どこに送るのか」という具体的な属性は無い。つまり、納税通知クラスには、その送り先を示す属性「住民 ID(FK)」が陽には存在しない。理由は、納税通知の送り先は、「自動車税」→「所有」→「住民」をたどってゆけば分かるからである。納税通知の属性名として、「住民」属性は、少なくとも、モデル作成段階では不要である。入れてしまうと、冗長な「派生属性」(他の属性から計算できる属性値)となる。

実装段階のことを考えても、派生属性は望ましいものではない。納税通知に、送り先の「住民」属性を追加してしまうと、派生属性のソースである属性の値が書き換えられた場合には、派生属性値も書き換える必要が生じる。これは、余分なコーディングを生む。また、ソフトウェア改造時には、派生属性値を計算するルーチンまで、保守の対象となる。本来は、モデリングレベルの第3正規形に整理されたクラスが、そのまま実装レベルのクラスとなり、それがそのまま Codd の第3正規形のリレーショナルデータベースとして写し取られるべきであろう。

現実には、派生属性を入れることは、クラスの実装段階ではあり得る。アクセス速度を上げたいからである。HDD上に存在するデータベースシステムにアクセスして、各クラスをFK(外部キー)を頼りに次々と探ってゆくことは効率的ではない。派生属性により値を保存しておけば、関連先の Object が素早くわかる。また、実装段階では、第3正規形を第2正規形に変換して、アクセス速度を稼ぐこともしばしば行われる。しかし、これもシステムの保守性から言えば、望ましいことではない。

近年、OS の 64bit 化と DRAM 高集積化によ

り巨大な実メモリ空間をアプリケーションが利用可能となっている。また、MRAM(磁気記録メモリ)等の不揮発性 RAM の実用化により、DRAM と同等の速度を有し、しかも、不揮発性のメモリ空間が実用の射程に入ってきている。データベースを RAM 上に載せたり(RAMDB)、巨大な DRAM 空間にアプリケーションが持つ全 Object を実装することも技術的には可能となっている。このような状況では、データベースシステムの応答は従来の HDD による場合よりも、大幅に高速化される。将来的には、派生属性や第2正規形をつかってアクセスを速くする発想から自由となり、モデル作成段階から実装レベルまでが、一貫した第3正規形で管理されることを期待するものである。

なお、ひとつの仮説であって、検証されたものではないが、上記のような、派生属性とでも言うべき関連(ポインタ)がクラスに張られるとすれば、「直接に対象クラスを指摘しないで、周囲の状況から入ってゆく」日本文化の影響がソフトウェアシステムの内部に、「関連という名前のポインタ情報」として、影響を与えている可能性を棄却できない。図 17 は、そのような日本におけるソフトウェアの内部構成をイメージしている。ソフトウェアの中心には、第3正規系で書かれた、英語的な因果関係をもつクラス図があるとす。しかし、ユーザはそのようなクラスのアクセスを期待していないとすれば、ソフトウェアの内部構造はどうなるだろうか。図 17 に示したように、本来のきれいな英語形式のクラスの周りに、日本的気配りのための関連、すなわちリンクが貼られる恐れなしとは言えない。

日本のソフトウェア開発では、従来より、以下のことが主張されてきた。

- 日本のソフトウェア産業では、欧米に比べて、パッケージソフトの割合が低く、単品の受託開発ソフトの割合が大きい。たとえ、海外の ERP パッケージを導入しても、結局、細かいカスタマイズが要求され、画面の修正等のコストがかかる。
- モデリングを行えばよいことは分かっているが、なかなか実行されない。実行しても「分析のための分析」となり、実際のコー

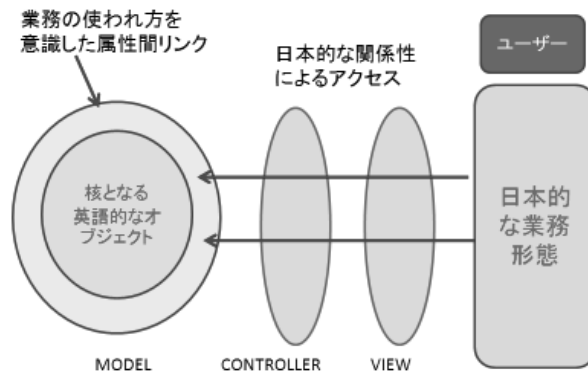


図 17 日本文化と Object 指向のミスマッチ

ディングにそれが生かされないことがある。

- 関数仕様書は、実際には、プログラミングが終わってから、逆に仕様書に転記していることが多い。
- 一方で、GOF のデザインパターンについては、自分のプログラミングに取り込みたいと念じている。

以上の状況がなぜ、生じるかは単純ではない。しばしば言われるように、我が国における発注から納期までの短さも大きな原因であろう。しかしながら、本稿で述べた、言語と文化のギャップも、その一因ではないだろうか。

海外からの輸入品であるデザインパターンは、西洋人のカテゴリーの概念、トップダウンに対象を捉えてゆく文化によって形成されている。一生懸命練習してモデル化した対象ドメインのモデルは、英語流である。そして、それをコーディングするプログラミング言語もコンテキスト・フリー・グラマーが対象とする世界である。即ち、トップダウンな問題解決を要求する。プログラムの世界は、すべて、欧米の世界なのである。

これに対して、ユーザの要求、利用者のインタフェースへの「使い心地」は英語で判断されるのであろうか。言うまでも、ここでは、アクション記述よりも、状態・状況表現から入ることを好む、日本語の世界が展開する。そうであるなら、インタフェースにうるさいのは、日本

人の現場ユーザの頑固さが原因であるとは思えない。一番、おそれるのは、プログラムの核の部分には、欧米流の美しいモデルや、GOF のデザインパターンで固められているとしても、その周辺では、日本的なユーザの要求をストレートに表現してしまった日本型のコードが取り巻いているのかもしれない。

もし、上記の仮説が正しいなら、日本のアプリケーションは、中心部分に一部のモデラーが作成した「美しい」モデルがあったとしても、そのモデルは、応用プログラムからは、日本的な気配りのアクセスパスを持っている事を要求される。アクセスパスは、結局のところ、モデルの上に、切り貼りされた「関連」として、実装されることになる。そうであるなら、欧米流の設計論をいくら持ちこんでも、シンプルで改造が容易なアプリケーションプログラムは生まれにくい。言語の問題は、今後の調査にまたねばならないが、大きな問題を内包している可能性は感じている。

9. 終わりに

本稿では、認知言語学の知見に従えば、Object 指向の ER 図・クラス図は、英語の認知構造そのものであることを示した。この見方が正しいとするなら、英語圏では、SE (ソフトウェア・エンジニア) にとって、ユースケース記述などの対象世界を記述したテキストから、名詞と動詞を抜き出せば、ストレートフォワードな手順で、ER 図・クラス図を作成できることに

なる。その意味において、RUP (Rational Unified Process) [35] の妥当性は保証される。

しかし、そのプロセスを日本語による Object 指向に、そのまま適用して良いかどうかは、疑念が残る。日本語の認知構造は、英語とは大きく異なっているからである。動詞の種別も異なっている。日本語で書かれたヒアリング結果から、ストレートフォワードに名詞・動詞を抜き出しただけでは、ER 図・クラス図には対応しない恐れがある。日本語テキストから ER 図等を作成することは、一種の日英翻訳となるからである。日本語テキストが伝える対象世界について、その内部に含まれる因果関係を冷静に分析する作業が必要である。

以上に従えば、「Object 指向とは、対象世界に存在する『もの』に着目して、対象世界をモデル化する手法である」と言った類の説明は、初学者の SE (ソフトウェアエンジニア) に混乱をもたらす恐れがある。認知言語学の教える通り、「可算な名詞」が Object であると最初から説明するべきである。そして、日本語テキスト記述から ER 図・クラス図を作る場合には、テキスト記述に存在するアクターや、アクターからアクターへの原因結果の因果関係の流れについて、一度、きちんと考察する癖をつけるべきである。それは、主題優先言語で、環境や結果を記述しがちな日本語表現を、主語優勢言語である英語の表現に、心の中で読みかえる作業に等しい。

上記の観点からすれば、グローバルスタンダード UML の書き方さえ学べば、海外の SE と共同作業できる、と考えるのは早計であろう。英語をコアイメージレベルで理解できないと議論は難しい。認知文法は、英語教育には既に導入されている [29] [28]。認知文法を参照する教育法の有効性は、英語教育だけではなく、Object 指向教育にも当てはまる可能性はある。その検証は、今後の課題である。

謝辞

本稿に盛られた議論については、同志社大学嘱託講師・井田明男様から頂いたコメントが寄与している部分があります。深謝の意を表しま

す。ただし、本稿に含まれる誤り・誤解の責任はすべて、著者にあります。

参考文献

- [1] 手島歩三, 松井洋満, 南波幸雄, 安保秀雄, 小池俊弘, 「働く人の心をつなぐ情報技術—概念データモデルの設計」, 白桃書房, 2011年6月
- [2] マーチン・ファウラー (著), 羽生田栄一 (訳), 「UML モデリングのエッセンス (第3版)」, 翔泳社, 2005年6月
- [3] 児玉公信, 「UML モデリングの本質」, 日経 BP 社, 2004年4月
- [4] 今井むつみ, 「ことばと思考」, 岩波新書, 2010年10月
- [5] 金田重郎, 「『中空均衡構造論』に基づく情報システムの要求分析に関する一考察」, 情報処理学会, 情報システムと社会環境研究会, SIG-IS, 2010年9月.
- [6] L・ベンジャミン・ウォーフ (著), 池上嘉彦 (訳), 「言語・思考・現実」, 講談社学術文庫, 1993年5月
- [7] 河上誓作 (編著), 「認知言語学の基礎」, 研究社, 1996年11月
- [8] 今井隆夫, 「イメージで捉える感覚英文法」, 2010年10月
- [9] 金田重郎, 「認知文法に基づくオブジェクト指向の理解」, 情報処理学会, 情報システムと社会環境研究会, SIG-IS, 2011年9月
- [10] 金田重郎, 永田健, 「日本語特性に着目した日本語 GTA (J-GTA) の提案」, 電子情報通信学会, 知能ソフトウェア研究会, SIG-KBSE, 2011年7月
- [11] 金田重郎, 「三上文法に基づく ER 図関連名に対する一考察—関連の説明における視点移動と概念データモデリング (CDM) —」, 電子情報通信学会, 知能ソフトウェア研究会, SIG-KBSE, 2011年3月
- [12] 金田重郎, 中川隆広, 一瀬邦継, 「概念データモデリングの実施手順について」, 情報処理学会, 情報システムと社会環境研究会, SIG-IS, 2010年12月
- [13] 金田重郎, 「プログラマティズムに基づく概念データモデリングの再構築—オブジェクト指向の哲学的背景について—」, 電子情報通信学会, 知能ソフトウェア研究会, SIG-KBSE, 2010年5月
- [14] P.P. Chen, "The Entity-Relationship - Toward Unified View of Data", ACM Trans-actions on Database, Vol.1, No.1, pp.9-36, March, 1976.
- [15] T. Teorey et al., "Database Modeling and Design, 5th Ed.", Elsevier Inc, 2011.
- [16] マーチン・ファウラー (著), 堀内一 (訳), 友野晶夫 (訳), 児玉公信 (訳), 大脇文雄 (訳) 「アナリシスパターン—再利用可能なオブジェクトモデル— (新装版)」ピアソンエデュケーション, 2002年4月
- [17] エリック・ガンマ (著), ラルフ・ジョンソン (著), リチャー

- ド・ヘルム (著), ジョン・プリシディース (著), 本位田真一 (訳), 吉田和樹 (訳), 「オブジェクト指向における再利用のためのデザインパターン (改訂版)」, ソフトバンククリエイティブ, 1999年10月
- [18] リチャード・E・ニスベット (著), 村本由紀子 (訳), 「木を見る西洋人森を見る東洋人—思考の違いはいかにして生まれるか—」, ダイアモンド社, 2006年6月
- [19] 増田貴彦, 「ボスだけを見る欧米人・みんなの顔まで見る日本人」, 講談社, 2010年11月
- [20] 金田一春彦編, 「日本語動詞のアスペクト」, むぎ書房, 1976年
- [21] 金谷武洋, 「日本語に主語はいらない」, 講談社選書メチエ, 2002年1月
- [22] Sharon Allen, Evan Terry, "Beginning Relational Data Modeling, SECOND EDITION", Apress, 2005.
- [23] 須賀一好, 早津恵美子, 「動詞の自他 (日本語研究資料集 (第1期第8巻))」, ひつじ書房, 1995年8月
- [24] 三上章, 「象は鼻が長い—日本文法入門」, くろしお出版, 1963年10月
- [25] 三上章, 「続・現代語法序説 - 主語廃止論」, くろしお出版, 1992年10月
- [26] 益岡隆志, 「連体節の表現と主名詞の主題性」, pp.139-153, 益岡隆志, 野田尚史, 沼田善子 (編), 「日本語の主題ととりたて」, くろしお出版, 1995年5月
- [27] ビーター・コード, エドワード・ヨードン (著), 富士ゼロックス情報システム株式会社 (訳), 羽生田栄一 (監), 「オブジェクト指向分析 (OOA) 第2版」, トッパン, 1993年, P.155.
- [28] Takao Imai, "A Consideration of How to Teach Peripheral Examples of be + V-ing from the Viewpoint of Cognitive Grammar", 瀬木学園紀要, Vol.4, pp.5-13, 愛知みずほ大学, 2010年
- [29] 今井隆夫, 「学校文法における可算/不可算名詞及び動作/状態動詞の教え方を認知文法の考え方を参照して検討する」, 瀬木学園紀要, Vol.3, Vol.3, p.27-45, 愛知みずほ大学, 2009年3月
- [30] 上野景文, 「現代日本文明論—神を呑み込んだカミガミの物語 (はなし)」, RBA 新書, 第三企画, 2006年9月
- [31] 長谷部陽一郎, 「コンピュータ・アナロジー再考—認知言語学とオブジェクト指向の観点から」, 日本認知言語学会論文集, 日本認知言語学会, Vol.5, pp.126-136, 2005年
- [32] 羽生田栄一の「オブジェクト論」【後編】, 日経バイト 2004年6月号, 2004年9月
- [33] G.レイコフ (著), 池上嘉彦他 (訳), 「認知意味論」, 1993年1月
- [34] 三浦つとむ, 「日本語とはどういう言語か」, 講談社学術文庫, 講談社, 1976年6月
- [35] フィリップ・クルーシュテン (著), 藤井拓 (訳), 「ラショナル統一プロセス入門第3版」, アスキー, 2004年11月