

# 日本語仕様文からの概念モデリングプロセス

—英語 7 文型と関数従属性に基づくクラス図の作成—

金田 重郎<sup>1,a)</sup> 井田 明男<sup>1</sup> 酒井 孝真<sup>1</sup>

**概要:** サピア=ウォーフの仮説は、母語が対象世界の認識に影響する可能性を示唆している。特に、ソフトウェア開発上流工程では、自然言語（母語）を用いるため、サピア=ウォーフの仮説の影響を受ける可能性がある。しかし、何故か、このことは、従来論じられて来なかった。そこで、本稿では、クラス図が英語の認知構造を反映していること、そして、そのことの認識なしでクラス図を理解しようとしても困難が生じる恐れがあること、を示す。ただし、著者らの従来研究では、日本では良く知られた英語 5 文型を用いてきた。5 文型は非英語圏出身者に英語を教えるためのツールであるが、現在の英国の文法書では、「所格」を必須とする 7 文型が推奨されている。そこで、本稿では 7 文型を用いてクラス図を理解する。これによって「もの—こと—もの」パターンの本質も理解できる様になる。更に、クラス図では、関数従属性によるモジュラリティの確保が必要であることを論じて、7 文型の視点をも併せ、クラス図作成のためのガイドラインを提示する。このガイドラインの下で、日本語特有の「存在文（『ある』型文）」表現ではなく、関数従属性を取り出しやすい「行為文（『する』型文）」を用いて、日本語記述文を単文化すれば、日本語圏に生まれた者であっても、見通し良くクラス図を作成できる。本稿で示したクラス図への理解によって、初学者はより易しくクラス図の本質を学び得ると確信している。

**キーワード:** クラス図, 英語, 認知言語学, 7 文型, 関数従属性, モジュラリティ

## Guidelines for Class Diagram Design based on English Sentence Patterns and Functional Dependency

SHIGEO KANEDA<sup>1,a)</sup> AKIO IDA<sup>1</sup> TAKAMASA SAKAI<sup>1</sup>

**Abstract:** The authors demonstrated previously that the structure of a class diagram reflects the cognitive structure of English, based on the results of cognitive linguistics. In that conventional study, the authors focused on the five sentence patterns of English, well known in Japanese English education courses. However, a contemporary English grammar book recommends seven sentence patterns for non-native education as a substitute for the five sentence patterns. Thus, this paper analyzes class-diagram structure using the seven sentence patterns. The analysis of seven sentence patterns clarifies the basis of the class diagram structure, and this study employs functional dependency to validate the class diagram structure. The combination of the seven sentence patterns and functional dependency clarifies some guidelines for class diagram creation. Also, the well-known “Mono Koto Mono Pattern” in Japan is properly understood with these guidelines. The resultant guidelines for designing a class diagram help us to create a class diagram and check the validity of the resultant classes in the requirement-analysis phase of software development.

**Keywords:** Class Diagram, English, Cognitive linguistics, Seven Sentence Patterns, Functional Dependency, Modularity

<sup>1</sup> 同志社大学大学院・理工学研究科, 〒 610-0321, 京都府京田辺市多々羅都谷, 1-3  
Graduate School of Science and Engineering, Doshisha Uni-

versity, 1-3, Miyakodani, Tatara, Kyotanabe-city, Kyoto-pref, 610-0321, Japan  
<sup>a)</sup> skaneda@mail.doshisha.ac.jp

## 1. はじめに

著者らは、UMLの静的モデル、即ち「クラス図」が、クラス図が生まれた国である英米の母語、英語の認知構造を反映していることを明らかにしてきた [1][2][3]。おそらく、英語ネイティブには、クラス図は（予備知識無しに）理解できるものと考えられる。しかし、英語の非ネイティブである我々は、クラス図が持つ対象への認知構造を理解した上で、学習・利用する必要があるのではないか、とするのが著者らのスタンスである。

但し、上記の従来研究では、英語の文構造として「5文型」を利用して来た。5文型は、もともと、非ネイティブに英語を教育するために生まれたものであるが、現在の英国の文法書で推奨されているのは7文型であり、5文型は忘れられている。7文型は、5文型とは異なり、文を構成する意味的に必要な要素を、前置詞が付いていても、文型の中に取り込んでいる。具体的には、第7文型で追加されたものは、**所格**（処格）（locative:場所を示す要素）である。もともと、ソフトウェア工学は、「本質的に関連性が強い要素同士を凝集しておけば、将来の改造に対して安定である」との視点に立ってデータ構造を設計してきた。従って、7文型は、5文型と比較して、より安定した構造に対応していると理解すべきであろう。

そこで、本稿では、7文型の立場から、クラス図の構造を分析する。これによって、「もの-こと-もの」パターンとして知られるケースも、自然にクラス図に取り込まれる。更に、本稿では、クラス図のモジュラリティを担保するために、クラス図では、関数従属性の観点が多用されていると考える。最終的に、7文型と関数従属性の観点からクラス図を眺め直すことによって、クラス図を作成するためのガイドラインが規定できる。本ガイドラインは、初学者が、クラス図を学ぶ上でも有効と考えている。

以下、2章では、サピア=ウォーフの仮説について触れる。3章では、7文型の観点からクラス図を再理解する。次に、4章では、関数従属性の観点から、「関連」の役割を再定義するとともに、日本語仕様記述からクラス図を生成するには、「存在文」表現を取る日本語を「行為文」\*1にリライトする必要があることを論じる。最後に、第5章は、本稿で示したガイドラインに基づいて、若干のサンプルからクラス図を生成した例を示す。第6章はまとめである。

\*1 日本語は「ある型」、英語は「する型」と呼ばれることがある [15][16]。この呼称では、動詞「ある」を使っていれば「ある型」と理解しなくなる。しかし、「磁石にはN極とS極がある」という日本語は、英語なら have に相当する「する型」である。主語から目的語へのエネルギーの流れがあるかどうかで判断したいので、日本語文を「ある型」とは呼ばず、「存在文」と呼ぶ。例えば、同じ「ある」でも、「信号がある」は、日本語的な存在文（ある型）である。

## 2. サピア=ウォーフの仮説

1920年代の終わり、サピアとウォーフは、「異なる言語間の差異は、単なる文法組織の違いを越えて、思考様式の深い差異に結び付く」と問題提起した。よく知られた、「サピア=ウォーフの仮説」である。「言語相対論」とも呼ばれる。この仮説については、1930年代にブームになった後、検証困難性が指摘され、近年は否定的に見られて来た [4][5]。しかし、1984年にケイとケンプトンが、メキシコのタラウマラ語の研究から、実証的にこの仮説を証明した。その後も、スタンフォード大学、マサチューセッツ工科大学、カリフォルニア大学ロサンゼルス校の研究者によって、検証研究が行われた結果、言語（母語）は、思考を限定するものではないが、話し手の心の習慣を形成し、認知・記憶・連想などに大きな影響を与えることが分かっている [5]。

上記のサピア=ウォーフの仮説は、ソフトウェア工学の立場から軽視し得ないと考えられる。なぜなら、

- 残念ながら、ソフトウェア工学の手法は、英米（=母語が英語）で開発されたと言って良い。
- 特に、要求分析、要件定義といった最上流工程では、ヒアリングを母語で行い、母語でモデル化する。

からである。しかし、不思議なことに、我が国のソフトウェア産業界では、このことについて、全く注意が払われて来なかった。

本稿では、ソフトウェア工学の最上流工程の手法が、英米の母語、即ち、英語の影響を受けていることを素直に認め、英語の認知構造からソフトウェア工学のツールを理解した方が、理解の早道だとする視点を提起する。具体的には、クラス図（概念クラス図）を取り上げる。クラス図の教科書としては、海外（=英語圏）の優れた教科書が日本語に翻訳されている。また、多くの日本で書かれた教科書も、説明方法としては、英米圏のテキストを参考としているケースが多い様に思える。しかし、サピア=ウォーフの仮説が正しいのであれば、英米圏で定評のあるテキストを日本語に訳しても、それは、必ずしも、日本人学生にとって理解容易なものとは言えない恐れがある。

クラス図はどこの大学の情報工学のコースでも教えられているだろう。しかし、現実には、我が国の受託開発ソフトウェアの開発においては、クラス図（特に上流工程の概念クラス図）は、必ずしも使われていない。そのひとつの原因は、実は、現場の担当SEが、クラス図に対して便利さを感じていなかったり、クラス図の書き方に自信を持っていないのではないだろうか。英米人には優れたテキスト（翻訳書）を学生に与えて、結果的にはあるが、「悟れ」と言って教えてしまっている事に、問題の一因があるのではないだろうか。クラス図は英語であるとみて、その上で、クラス図を理解するべきであるとするのが、サピア=ウォーフ

表 1 名詞・動詞とクラス図

品詞	種別	クラス図表現
名詞	可算名詞	クラス
	不可算名詞	属性, または属性値
動詞	動作動詞	オペレーション (メソッド)
	状態動詞	関連

の仮説に準拠した本稿の立場である。

### 3. 7文型にもとづくクラス図の理解

本章では、英語7文型とクラス図との対応を分析することで、クラス図作成に関する幾つかの「ガイドライン」を提示する。

#### 3.1 副詞類を除く7文型の理解

従来の Object 指向のテキストでは、Object のことを、日本語の「もの」とか「概念」という表現に翻訳して説明している。しかし、このように日本語にしてみると、本来の Object の意味ではなく、日本語名詞の意味が現れる。今井むつみは、認知言語学の立場から、Object には（英語の）物質名詞は含まれないことを論じている [6]。英米ネイティブの場合には、（英語の）可算名詞のみが「Object」となる。Concept にも同様の意味があり、数え得るインスタンスに共通的に成り立つ性質を示す。認知言語学を適用すれば、表1に示したように、クラス名は可算名詞の集合名であり、クラスが持つ属性名・属性値は、不可算名詞から取り出して利用されると理解される。クラスとはそもそも集合である\*2。一方、動詞は、動作動詞は原則、オペレーション（メソッド）であり、状態動詞が関連となる。表1にはこれをまとめて示しておく。ただし、英語の動作動詞は、永続的な Object 間の関係を含意する場合がある。従って、表1のみで表面的には処理できない。

なお、クラス名とその要素である Object（インスタンス）には、無標と有標の区別が可能である。有標とは、「太郎」「花子」の様な、個々の Object を識別するシンボルであり、無標とは、「人」のような、個々を識別しない、総称を表す言葉である。ただし、無標か有標かは、抽象化の段階によって変わる。たとえば、アムールトラは、動物園にいる個々のトラ個体からみれば無標であるが、哺乳動物というクラスから見れば、有標と見なし得る。仕様記述文の抽象化とは、このような無標、有標の視点から行うべきである。

では、仕様が日本語記述として与えられた時、どのようなルールで変換されてクラス図となるのだろうか。この問題を解く為には、英語の文章構造をモデル化する必要がある。

\*2 オブジェクト指向では、オブジェクトは、オブジェクトの属性とは別に、それ自身のアイデンティティを持つ [7]、とされるため、本稿の議論でも、クラスには暗黙裡に識別子となる一つの属性が存在していることを前提とする。

る。我々のもつとも身近にある文章構造は英語の「5文型」であろう。しかし、現在の英国の文法書では7文型が利用されている [8]。7文型を表2に示す。7文型は5文型の発展型であり、5文型と異なるのは第4文型 ( $S+V+A$ ) 及び第7文型 ( $S+V+O+A$ ) のみである。ここで、Sは主語、Oは目的語、Aは副詞類、Cは補語を示す。副詞類とは、動詞を形容して、場所、時間、方法、理由、程度などの情報を付け加える品詞類である。Cは、誰か/何かを性格づけるための質（量ではない）を表現している。注意すべきなのは、7文型で追加された要素が「所格」である点である。所格は、場所を示す格であり、日本語では、格助詞「に」「で」で表現されるものを含んでいる。

具体的に見てみよう。表2の第4文型、第7文型を、従来の5文型に無理やり当てはめると、“My office is”, “You can put the dish.”となる。これでは意味はなさないので7文型が導入された。7文型では、副詞類（所格）を追加して、意味的な必須要素を落とさないようにしている。7文型とクラス図の対応を確認したい。但し、第4文型、第7文型については後述する。

7文型の第1文型は自動詞であり、動詞はメソッドになる可能性が高い。Object が一個なので、「関連」とはなり得ない。表1のままとなる。

7文型の第2文型 (5文型の第3文型) は、最も良く知られた  $S+V+O$  型である。動詞が動作動詞ならメソッドに、状態動詞なら関連になる。認知言語学では、英語の場合、SからOへ、エネルギーの流れがあるとす。つまり、関連とは、Sが決まればOが決まるものであり、一種の関数従属性である。これら「関連」の役割については、後ほど考察する。

第3文型 (5文型の第2文型) について考える。第3文型の動詞は比較的限定されており、be動詞、become、look、seem等の動詞であり、そのままbe動詞 (is-a 関連) となるか、主語に対応する Object の属性となる。ただし、注意がいるのは、日本語で言えば「見なす」という動詞の存在である。英語の「is-a」階層とは、「見なす」とか「定義する」と言った意味を含意しており、決して、抽象的なより高度な概念との関係を表すものではない。

第5文型 (5文型の第4文型) について考える。一般に、認知文法では、1つの動詞が2つの目的語を取る場合 (二重目的構文)、二重目的語の間には、所有関係を含意する。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等がある。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する。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と等しい。第5文型が実行された結果が、「has-a 階層」である。ここから、「has-a」が関連名の中で特別扱いされる理由がわかる。もともとの仕様文には

表 2 英語の 7 文型 (S:Subject, V:Verb, O:Object, A:Adverb, C:Complement)

文型番号	文型	例文
1	$S + V$	The sun (S) is shining (V).
2	$S + V + O$	That lecture (S) bored (V) me (O).
3	$S + V + C$	Your dinner (S) seems (V) ready (C).
4	$S + V + A$	My office (S) is (V) in the next building (A)
5	$S + V + O + O$	I (S) must send (V) my parents (O) an anniversary card (O).
6	$S + V + O + C$	Most students (S) have found (V) her (O) reasonably helpful (C).
7	$S + V + O + A$	You (S) can put (V) the dish (O) on the table (A).

無い動詞 (have) で、クラス図を描く必要があるからである。つまり、仕様文中の動作動詞の名称は、そのままでは(動作動詞だから) 関連名としては使用できない。しかし、その動作動詞が現実社会に恒常的結果を残す場合、それをクラス図に静的な関係として表現しなければならない。その際に必要となる関連名が has-a である。

第 6 文型 (5 文型の第 5 文型) の例としては、「We call him Hide.」, 「She always makes me happy.」等がある。コアイメージは第 1 引数=第 2 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 1 引数がカバーする意味の空間の中に、第 2 引数が組み込まれるイメージである。第 5 文型は、「is-a 階層」(あるいはクラスの属性値) そのものである。この場合にも、元の仕様記述中の動詞は動作動詞なので、そのままでは関連名としては使用できない。そこで、特別な関連名「is-a」が必要となるのである。

以上見て来たように、「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。もともとの仕様文にはないが、意味的に静的な関係を描くツールとして、英語では、この 2 つが必須である。構文パターンがこれ以外には存在しないため、クラス図においても、is-a と has-a 以外に特別な関連名は不要となるのである。

### 3.2 副詞類の理解

前節では、従来からの著者らの研究成果を 7 文型に拡張した。本節では、残る、第 4 文型、第 7 文型について考察する。これらの文型は、場所を示す単語を持っている。所格である。これらの単語は、省略すると、文として意味をなさない。第 4 文型でも、第 7 文型でも同様である。

副詞類を伴う文型を分析する前に、そもそも「クラス図」とは何だったかを確認したい。クラス図は表 1 に示した要素を持つが、「時間軸」は捨象している。つまり、ある Object が生まれるとともにその Object の属性となり、その Object がこの世に存在する限り、その Object に付随し、その Object がこの世から消えるとともに当該属性も消えるものを「属性」として附属させている\*3。即ち、

\*3 この視点は、大木の研究と同じ意味を持つと思われる [9][10]

Object(正確には Object の識別子) と属性群は、第 3 正規形\*4をなす。では、「関連」とは何かということ、これも、「エネルギーの伝搬」なので、「主語が決まれば目的語も決まる」という意味で、関数従属性\*5を持つ。ただし、この関数従属性は、ドメインの時間の推移とともに、存在したり存在しなかったりする。関連とは、クラス間に存在する可能性がある関数従属性を、時間を無視して描くものと理解される。言いかえると、当該クラスのインスタンスの誕生から消滅まで常にリンクがある様なら、属性にするか、「合成」の「has-a」にすればよい。

そのような観点で、表 2 の、第 4 文型を見てみよう。必須の要素を含めている 7 文型では、副詞類 (A) は、所格、即ち「場所」を示している。表 2 の例では、ビルは、建物なので可算名詞であるため、Object となる。オフィスは隣のビルにある。動詞は存在動詞であるので、オフィスからビルに関連が張られる。つまり、もし、動詞が動作動詞でも、所格によって、静的な位置関係が生じるなら、それは関連化すべきことを暗示する。一方で、場所をオフィスの属性とすれば、next building は属性値として表現しても良い。

表 2 の第 7 文型も同様である。所格を導入しており、表 2 の文例では、動作動詞である。しかし、「動作動詞であっても、必須の格として所格を持っている以上、その結果は、何らかの作用がその場所に及ぶ」と考えるのが自然であろう。そうすると、クラス図はそれを記載する必要がある。

次に、この所格が関係するものとして、「もの-こと-もの」パターンと称せられるケースを見てみたい。たとえば、以下を考える。

The company ordered the toy to the manufacturer.

「The company ordered the toy.」「The company ordered to the manufacturer.」では何れも意味が完結しない。所格は、本来、英語では、場所を示す格であるが、日本語では、格助詞「に」「で」などで示されるものが含まれる。つ

\*4 後述する様に、正確には、ボイスコード正規形、あるいは、第 4、第 5 正規形であるが、ここでは、分かりやすいので、「第 3 正規形」と呼ぶ。

\*5 正確には、主語が決まると複数の目的語が限定されたため、一方の属性値が決まると他方の値が決まる関数従属性を含めて、「自明な多値従属」と言うべきと考える。本論文では、そのような意味で、関数従属性との用語を用いる。ただし、自明な多値従属性においても、推移性が含まれる事は望ましくないとと思われる。

表 3 認知言語学と関数従属性から導かれたクラス図作成ガイドライン (一覧)

項番	区分	ガイドライン内容	事例/補足説明
G1	全体	可算名詞がクラス名, 不可算名詞が属性名, 属性値, 状態動詞が関連, そして, 動作動詞がメソッドとなる. 日本語仕様記述を読み取る際には, この 4 種類の区別を意識する必要がある.	例: 自動車, 債務等は可算名詞, 砂, 朱色等は不可算名詞, 「歌う」は動作動詞, 「雇用している」は状態動詞. 日本語の状態動詞は「ある」「いる」しか無いが, この 2 つが付いていても, 状態動詞であることが釈然としない場合が多い.
G2		クラス図は, 時間的推移を捨象して, ビジネスの途中で, 一定期間に渡って成立する状態を表現するものである. クラス図には, 「時間的経過」の情報はない. 時間的経過は, 相互作用図などの動的モデルの担当である.	関係性が成立する期間とそうでない期間が存在する場合には, 当該関係性は「関連」とする. 関係が当該クラスのインスタンスが存在する期間中は時間的に変化しない関係は, クラス中の属性として表現するか, 又は「合成 (has-a)」関連として表現される.
G3		日本語は「存在文」を多用する言語であり, 英語は「行為文」中心である. 行為文 (する型) にしないと, 関数従属性が表現に現れ難いので, 日本語仕様文は, 単文化する際に, 日本語で行為文に書き換え, 主語を明確にするべきである. とりわけ, 日本語では少ない, 非生物主語を多用することになる.	「机には 4 本の足がある」→「机は 4 本の足を持つ。」クラス図は英語の認知構造であり, 特に, 「関連」はエネルギーの伝搬 (関数従属性) を意味するので, 日本語の存在文では, 関数従属性を取り出せない.
G4	クラス	クラスと属性の関係は, 当該クラスのインスタンスが生成されるとともに属性値が生まれ, インスタンスがこの世に存在する限り属性値が存在し, そのインスタンスがこの世から消えるとともに, 属性値も消える, そのようなライフサイクル一致の関係である. 結果, 必然的にクラスのすべての属性はクラスに関数従属することになるため, 第三正規形が保たれる.	「飲み物」の「値段」は, たとえ「メニュー」に書いてあったとしても, 値段はメニューの属性ではなく, 飲み物の属性である (但し, 複数の店を想定する場合には, 値段は, その提供者 (店) と提供物 (飲み物) の間にある「提供」の属性となる).
G5		メソッドも属性と類似した要素であり, 所属するクラスの識別子に対して, 推移的でない関数従属性を満足する.	「検索する」というメソッドは, データがある限りにおいて検索可能であるが, ユーザの誕生・死亡と検索は無関係である. 従って, 検索メソッドは, 被検索データにつく.
G6		関連とは関数従属性であり, 「このクラスが決まれば, 他方のクラスが決まるか否か」が関連を付ける必要性を判断する唯一の根拠である. 関連につながる一方のクラスが主語であると思っはならない.	「商品」と「メーカー」の間には明らかに関数従属性があるが, 「商品をメーカーに発注する」であり, 何れも主語ではない.
G7		クラスの名称は, 「無標」「有標」の観点から見た「無標」の名詞で表現される.	「花子」「太郎」は有標, 「人」は無標である. クラス名は集合の名称となる. 集合「人」の中に, 花子や太郎のインスタンスがある.
G8		対象ビジネス中にインスタンス一個だけあって, 当該ビジネスにおいて変化しない場合には, 概念クラス図でそれをクラス化する必要はない	特定の企業向けのアプリケーションでは, その日本語仕様に見える「会社」をクラスとして明示する必要はない.
G9		クラスには「種類」を表すクラスと, 個別のインスタンスを表すクラスがある.	プリウスでは, 1 台 1 台を区別して表すクラスと, トヨタ社内の車種を表すクラスが設けられるべきである. しかし, チョロ Q では, 1 台 1 台に管理番号を打つ必要はない.
G10	関連	動作動詞は, 結果として, 対象物の間に永続する関係性を含意することがある. この場合には, 動詞が動作動詞であっても, その永続的關係性を関連等として, 表現するべきである.	「チョコレートを花子に送る」は, 結果としてチョコレートは, 花子の所有物として永続的關係性を含意する. 英語の動作動詞では, しばしば, このような静的関係が含意されているためである.
G11		欧米人の抽象化とは, 「属性値が同じものを集める」という意味がある. 「一般的な大きな概念とものと具体的な概念との関係」とだけ捉えてはならない.	「～と見なす」「～と定義する」は is-a 階層の可能性が高い. is-a 階層とは, 「～と定義するための特別な関連」くらいに思った方が良い.
G12		「もの-こと-もの」パターンは, 本表の項番 6, 項番 10 である. 関連のいずれの側も主語ではない. 動作動詞 (行為) の結果, スタティックな関数従属性が生成される (多対多). このため「関連」は必要である. 但し, このままでは, (1) RDB に翻訳できない, (2) 関連に持たせるべき属性を書くべきクラスがない, と言った問題があり, 関連クラスの追加が必要であることが多い. これが「もの-こと-もの」パターンである.	「会社は A を B に発注する」では, 主語「会社」は暗黙の了解である. 一方, 「B に」は所格として必須である. A と B の間には関数従属性が生まれるから関連が生まれる. 但し, 発注に伴う「発注 ID」「発注日」などを記録するために, 第 3 正規形を構成する関連クラス「発注」が必要となる.

原文	ステップ1：単文化	ステップ2：存在文から行為文（所有文）への変換	ステップ3：無標化（なる型からする型への変換を含む）
日本のナンバープレートには数多くの種類が存在する。	(同左)	<b>ナンバープレートは多くの種類を持つ。</b>	(同左) 但し、原文で「種類」と言っているので、種類のクラスと個別インスタンスのクラスを分ける必要が取り出される。
自家用自動車は白地に緑文字で、事業用は緑地に白文字である。	自家用自動車のナンバープレートは白地に緑文字 事業用自動車のナンバープレートは緑地に白文字である。	(有標なので、次のステップで扱う)	<b>ナンバープレートは地の色、文字色の2つの属性を持つ。</b> <b>自家用自動車は、自家用自動車のナンバープレートを持つ。</b> <b>事業用自動車は、事業用自動車のナンバープレートを持つ。</b> <b>自家用自動車のナンバープレートの地の色は白、文字色は緑である。</b> <b>事業用自動車のナンバープレートの地の色は緑、文字色は白である。</b>
250ccを越える普通自動二輪車と大型自動二輪車は自家用には緑枠が付き、事業用には白枠が付く。	250ccを越える普通自動二輪車の自家用のナンバープレートには緑枠が付く。 大型自動二輪車の自家用のナンバープレートには緑枠が付く。 250ccを越える普通自動二輪車の事業用用のナンバープレートには白枠が付く。 大型自動二輪車の事業用のナンバープレートには白枠が付く。	(有標なので、次のステップで扱う)	<b>自動二輪車のナンバープレートは、枠の色という属性を持つ。</b> <b>250CCを越える普通自動二輪車のナンバープレートの枠色属性は緑である。</b> <b>250以下の普通自動二輪車のナンバープレートの枠色属性は「無し」である。</b> <b>大型自動二輪車のナンバープレートの枠色属性は「白」である。</b>
軽自動車は自家用が黄地に黒字、事業用が黒地に黄字である。	自家用の軽自動車のナンバープレートは、黄地に黒字である。 事業用の軽自動車のナンバープレートは、黒地に黄字である。	(有標なので、次のステップで扱う)	<b>軽自動車のナンバープレートには、地の色という属性と、文字色という属性を持つ。</b> <b>自家用の軽自動車のナンバープレートの地の色は黄色、文字色は黒である。</b> <b>事業用の軽自動車のナンバープレートの地の色は黒、文字色は黄色である。</b>
自動車には、普通乗用車と小型乗用車が存在する。普通乗用車と小型乗用車の区別は自動車の車幅や総排気量などで区別される。	自動車には、普通乗用車と小型乗用車が存在する。 普通乗用車と小型乗用車は、自動車の車幅や総排気量などで区別する。	<b>自動車には、普通乗用車と小型乗用車がある。</b>  (有標なので、次のステップで扱う)	(同左) <b>自動車は、車幅や排気量の属性を持つ。</b>

表 4 日本語仕様のクラス図への変換例（表中の太文字（朱書き）は行為文化を配意した部分）

まり、日本語で、「に」「で」を伴うものは、意味的に、この英語の第7文型に相当しており、広い意味での所格が省略できない。

一方で、上記の例で、「company」のクラス図上での扱いが問題となる。この company が、もし、特定企業のシステムの「当社」を示しているなら、クラス化する必要はない。クラス図上に載せるのは、一般的には、2個以上のインスタンスが存在し得るか、あるいは、何らかの関数従属性（=関連）が、当該 company との間に張られたり、張られなかったりするケースに限定すればよい。従って、日本語の「会社は商品をメーカーに発注する」では、当該企業のシステム構築なら、「会社」はクラス図から除外し、「商品」と「メーカー」の間に「発注する」という関連を張ることになる。ただし、このままでは、多対多の関連となるし、発注 ID とか、発注年月日などの属性は、第3正規形の観点から、どうしても、発注行為自体に付与する必要がある。このため、関連クラスとして「発注」と言ったクラ

スを作り、そこに発注 ID や発注年月日を入れることが必要となる。これが、「もの-こと-もの」パターンであろう。いいかえると、「もの-こと-もの」パターンも、英語とのアナロジーでいえば、少なくとも片方は、所格を含意していることが要求される。

## 4. 関数従属性と関連

### 4.1 関連と関数従属性

前節の「もの-こと-もの」パターンの例を見ても、「関連」の一方が必ず主語とは限らないことに注意が必要である。では、いったい、そもそも、「関連」とは何であろうか。クラス図は、時間的推移を捨象している。しかし、has-a 関連に「集約」「合成」の区別があることからみても、時間的に変化しない固定的な関係は、本来は、クラスと属性の関係とするか、「合成」を使うべきであり、時間的に変化する一時的な関数従属性をクラス間の「関連」として表現していると理解するべきである。「関連がありそうだから関連



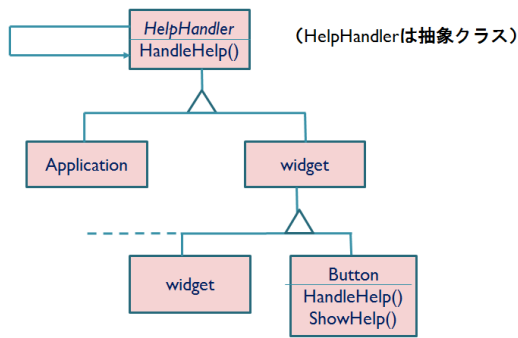


図 1 Chain of Responsibility パタンの例

を張る」と言う様な関連の作成には注意が必要だろう。なお、前述のように、関数従属性と言っても、1対多であるので、「自明な多値従属性」を指すと理解する。何れにせよ、「関連」として状態動詞を表現するべきかどうかを判断する基準は「一方のクラスのインスタンスが決まれば、もう一方が（多値を許容して）決まるか」である。これもひとつのガイドラインとなる。

更に、前章において7文型を分析した結果から、クラス図の関連とは、与えられた仕様文が動作動詞であっても、その影響が残る場合、あるいは、残したい場合には、結果として得られたデータ状態を、静的なクラス間の関係（関連）として表現しなければならない。これも、ひとつのガイドラインである。つまり、仕様記述に動詞として出現した動詞名を関連名にできないのは、has-a と is-a のケースだけであるだから、is-a と has-a はわざわざ、教科書に書いてあるのである。

注意がいるのは、第3文型、第6文型にみられるような、日本語言えば「見なす」と言った動作表現が is-a 階層として、静的なクラス間の関係になることである。is-a 階層については、欧米人の認識構造からみて、「同じ属性値をもつものは is-a 階層として汎化される」[11]。極論すれば、西欧人の「抽象化」とは、同じ属性値を持つものを集めたもの、位に捉えるべきであろう。従って、is-a 階層の上位にあるのは、より一般的なものとは限らない。たとえば、種々の花の上に、「まっすぐな茎をもつ花」と言った概念が is-a 階層の上部に設定されることがある。これも一つのガイドラインである。

GOF によるデザインパターン [12] の Chain of Responsibility パターンでも、アプリケーションの上位にヘルプハンドラーが置かれている例が示されている (図 1)。下位クラスのインスタンスであるためには、上位クラスのインスタンスでもあることが必要条件として満たされている必要がある。この例でも、アプリケーションのインスタンスはヘルプハンドラーとして用いることができる。最終的に、以上の議論から列挙されるガイドラインを表 3 にまとめる。尚、本稿では、紙幅の都合で、ガイドライン全てを説明できていない部分がある。

#### 4.2 ボイスコード正規形, 第 4 正規形, 第 5 正規形

本稿の前半部で、クラス図は、第 3 正規形と理解した。しかし、第 3 正規形は、候補キー (識別子) から候補キーに含まれない属性への (推移的ではない) 関数従属性は保証するが、候補キーではない属性から候補キーを構成する属性の一つへの関数従属性は妨げていない。クラスが持つべき性質から、必要なのは、「すべての事実、キーだけに関係する事実である [13]」である。従って、少なくともボイスコード正規形が望まれる。尚、候補キーを構成する属性が 1 つである場合には、ボイスコード正規形は第 4、第 5 正規形を意味する。

本稿では議論の対象外としたが、クラスの候補キー (識別子) が複数の属性から成る場合については、追加的分析が必要である。データベースの分野では、例えば、渡辺幸三が、エンティティをデザインする際に、第 4、第 5 正規形に注目する必要性を論じている [14]。但し、渡辺はあくまで、リレーショナルデータベース理論の中での議論を行っており、(自明ではない) 多値従属性の存在を問題視している。候補キーを構成する複数の属性間に (自明な) 多値従属性があったとしても、データベースの世界では、第 4 正規形化の必要性は生じない。しかし、クラス図の場合には、候補キーを構成する 2 つの属性に依存関係があることは、モジュラリティの観点からは望ましく無いかもしれない。今後の検討課題としたい。

#### 5. 存在文表現から行為文表現へ

表 3 を念頭に置いて、日本語仕様からのクラス図生成は、以下のステップを踏むことになる。ただし、以下のステップは、必ずしも順番に行われるものではなく、並行して行ったり、もとのステップに適宜戻っても良い。仕様記述は、主語と目的語の間に関数従属性の存在する文章に変換しないと、クラス図が取り出せない。この役割を担うのが、ステップ 2 の存在文から行為文への変換である [15][16]。行為文化することで主語を意識させ、しかも、主語から目的語への制御性を担保している。但し、存在文から行為文への書き直しは、ステップ 1 やステップ 3 で実行しても良い。

- ステップ 1: もとの日本語仕様を単文化する。
- ステップ 2: 存在文から行為文に変換する。
- ステップ 3: 有標から無標へ変換して、同じ意味を持つ単文を束ねて整理する。
- ステップ 4: クラス図を生成する。

以下の仕様文を用いて、クラス図を作成する。2 輪好きな院生に、ナンバープレートについてヒアリングしたものである。

【仕様記述文 (例)】日本のナンバープレートには数多くの種類が存在する。自家用自動車は白地に緑文字で、事業用は緑地に白文字である。これに加えて、250cc を越える普

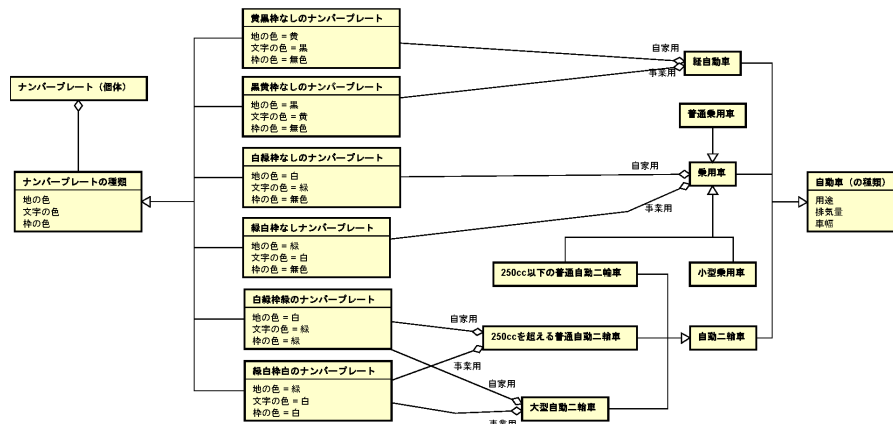


図 2 得られたクラス図

通自動二輪車と大型自動二輪車は自家用には緑枠が付き、事業用には白枠が付く。軽自動車は自家用が黄地に黒字、事業用が黒地に黄字である。また自動車には、普通乗用車と小型乗用車が存在する。普通乗用車と小型乗用車の区別は自動車の車幅や総排気量などで区別される。

分析結果の例を表 4 に示す。表 3 のガイドラインに準拠して、ひとつひとつの単文からクラス図を生成する。図 2 は最終的に得られたクラス図の例である。

## 6. おわりに

本稿では、日本語仕様記述からクラス図を作成するための分析ステップと、それに利用するガイドラインを整理した。本質的に「ある」型で、存在文形式で対象を表現する日本語から、「する」型で、行為文で対象を表現している英語への変換では、もとの日本語記述を単文に置き換えることが大きな意味を持っている。

同時に、表 3 に示したガイドラインは、一度、習熟すると、クラス図を作成する際に、その作業品質を向上させる。少なくとも著者らは、作成したクラス図が適切か否かを自由に議論できるようになったのは、表 3 のガイドラインを意識してからであった。もちろん、どのようなクラス図でも、表 3 の範囲で描ける保証はない。しかし、初学者がクラス図を学ぶ際には、十分に参考に資すると考えている。

なお、本稿で指摘したアプローチは、日本語仕様記述の個々の文に忠実なクラス図の作成である。実際のクラス図作成では、要求記述全体を見渡した分析も併せて行う必要がある。例えば日本語仕様記述には陽には出てこない抽象度のクラスを追加する必要があることもある。そのような視点は、本稿では考慮していない。今後の課題としたい。むしろ、日本語仕様記述をきちんと読み込む癖をつけるための、初学者向けのトレーニングの一助として、本稿の理解・手法をお役に立てて頂けることが、著者らの願いである。

## 参考文献

- [1] 金田重郎, 「認知文法に基づくオブジェクト指向の理解」, 同志社政策科学研究, Vol.13, No.2, pp.21-45, 同志社大学大学院総合政策科学研究科, 2012 年 3 月
- [2] 金田重郎, 世古龍郎, 「認知文法に基づくオブジェクト指向の理解」, 電子情報通信学会技術研究報告, 知能ソフトウェア工学, 111(396), pp.61-66, 2012 年 1 月
- [3] 金田重郎, 「日本文化に基づくソフトウェア工学の理解—オブジェクト指向と GTA を例として—」, 情報システム学会, 論文誌, Vol.7, No.2, pp.25-41, 2012 年 3 月
- [4] John B. Carroll (ed.), “Language, thought, and Reality – Selected Writings of Benjamin Lee Whorf”, Martino Publishing, Mansfield Center, CT, 2011.
- [5] ガイ・ドイッチャー (著), 椋田直子 (訳) 「言語が違えば世界も違って見えるわけ」, 合同出版 (株), 2012 年 12 月
- [6] 今井むつみ, 「ことばと思考」, 岩波書店, p.38, 2010 年 10 月
- [7] J. ランボー (他著) 羽生田栄一 (監訳), 「オブジェクト指向方法論 OMT モデル化と設計」, トッパン, 1992 年 7 月.
- [8] シドニー・グリーンバウム, ランドルフ・クワーク著, 池上嘉彦 (他) 著, 「現代英語文法 大学編 新版」, p.367, 紀伊国屋書店, 1995 年 8 月 (Sidney Greenbaum, Randolph Quirk “A Student’s Grammar of the English Language”, Langman, 1990)
- [9] 大木幹雄, 「ボトムアップな MVC アーキテクチャの概念的な構成方法に関する考察」, 情報処理学会研究報告, ソフトウェア工学研究会報告, 2000(104), pp.57-63, 2000 年 11 月
- [10] 大木幹雄, 秋山構平, 「概念モデリングにおける判断基準の提案とその有効性評価」, 電子情報通信学会論文誌, D-I, 情報・システム, J84-D-1(6), pp.723-735, 2001 年 6 月
- [11] 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」, ダイヤモンド社, 2004 年 6 月
- [12] エリック ガンマ (他著), 本位田真一 (他訳), 「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」, ソフトバンククリエイティブ, 改訂版, 1999 年 10 月
- [13] C. J. Date (著), 株式会社クイープ (訳), 「データベース実証講義-エンジニアのためのリレーショナル理論」, オーム社, 2006 年 2 月
- [14] 渡辺幸三, 「販売管理システムで学ぶモデリング講座」 翔泳社, 2008 年 5 月
- [15] 池上嘉彦 (著) 「「する」と「なる」の言語学—言語と文化のタイポロジーへの試論」, 大修館書店, 1981 年 1 月
- [16] 金谷武洋, 「日本語文法の謎を解く—「ある」日本語と「する」英語」, ちくま新書, 2003 年 1 月