

自然言語仕様からモジュール構造を抽出する手法について

佐伯 元司^{*} · 蓬萊 尚幸^{**} · 榎本 肇^{**}

^{*}東京工業大学

^{**}富士通国際研

本報告では、自然言語（英語）で記述された非形式仕様から形式仕様のモジュール構造を得る手法について考察する。得られるモジュール構造は、オブジェクト指向的な設計法に基づいたトップレベルのモジュール構造である。

オブジェクトモデルにおけるソフトウェア構成要素であるクラス、オブジェクト、属性、メソッドといった概念を自然言語文の構成要素、つまり語句にどのように対応づけるかが問題である。本手法では、自然言語文の意味において重要な役割を果たす動詞に注目し、その動詞が取り得る格構造（動詞型と呼ぶ）をオブジェクトモデルの視点から分類した。さらに、システム内の動作同士の関係（例えば、因果関係など）も分類した。動詞型とその動作が関与する動作間関係とに基づいて、文中の名詞や動詞がどのようなソフトウェア構成要素に対応するかの規則を与えた。この規則をLiftの制御問題に適用し、本手法を評価した。このような仕様化の過程を形式化しようとする考えは、ソフトウェアプロセスを形式化しようとする考え、つまりプロセスプログラミングの考えでもある。

On the Method for Extracting Module Structures from Natural Language Specification

Motoshi SAEKI[†], Hisayuki HORAI[‡], Hajime ENOMOTO[‡]

[†]Tokyo Institute of Technology, [‡]IIAS-SIS, FUJITSU LIMITED

[†]2-12-1, O-okayama, Meguro-ku, Tokyo, 152, Japan

[‡]1-17-25, Shin-Kamata, Oota-ku, Tokyo, 144, Japan

The method to extract the structure of software modules based on object oriented model from natural-language specification is presented. We concentrate on types of verb patterns occurring in sentences and on types of relations among actions represented by verbs. In principle, each word such as nouns and verbs in a natural-language specification corresponds to a software concept in object oriented model, e.g. "class", "attribute", and "method". Choosing candidate words for software components, determining kinds of them, and relating modules to other modules hierarchically depend on types of verb patterns and on those of the relations. These rules for extracting module structures are collected and applied to the simple example "Lift Control Problem". We discuss our approach using the example. This is an example of a part of a formal specification of software development process.

1. まえがき

プロトタイプ手法¹⁾や代数的仕様化技法²⁾に代表される形式的仕様記述法は、発注者の要求意図に即した高品質の要求仕様を作り上げるための1つの有用な手段である。しかし、これらの手法はいずれも生成物を検査するという形式をとるため、なんらかの形で入力となるプロトタイププログラムや形式的仕様を作り上げた後でないと効果をあげることができず、発注者の意図からいかにこれらを作り上げるかについては、依然として問題になっている。発注者の意図は、発注者自身が曖昧であっても、自然言語で非形式的仕様として記述することは可能である。もちろん、このような仕様には矛盾や不完全な部分といった誤りを含んでいることもある。しかし、このような記述には誤りのない形式的仕様を導くための重要な手がかりが含まれているはずである³⁾⁻⁵⁾。

本稿では、手がかりとして、自然言語文の中のどのような特徴に注目すればよいか、さらにその手がかりを使ってどのように形式的仕様を構築していくかについて考察する。具体的には、自然言語の文中に出現する動詞に注目し、その格構造や格要素となっている語句の素性を形式的仕様構築の手がかりにしようというものである。本論文では、非形式的仕様として英語による記述を、形式的仕様を記述するための言語として Pure TELL^{5), 9)}を用いる。Pure TELLの言語仕様については別稿にゆずるが、本手法が適用可能であるように設計されている。

2. 品詞の分類

仕様は、一般には平叙文の集まりとして記述され、その意味は文同士の連結関係と各文の意味によって規定される。1つの文の意味において、その本動詞の意味の果たす役割はきわめて大きい。仕様記述中に本動詞として使用される動詞や動詞が導く文型(以下、動詞型という)は限定されている。従って、仕様記述文から動詞を収集し、その動詞型を分類しておくことは、記述内容の理解だけでなく、形式的仕様導出の際の手がかりを探すのにも役立つと考えられる。

様々な種類の動詞を、ソフトウェアの言葉として統一的に取り扱うために、以下のようなオブジェクト指向型のモデル⁶⁾を導入する。現実世界は、ある実体(オブジェクトと呼ぶ)の集合からなる。共通の性質を持ったオブジェクト同士をまとめてクラスとすることができる。各オブジェクトは、種々の属性を持っており、その属性値の直積によってオブジェクトの内部状態が表される。属性値は、時刻の進行に応じて変化するものもあれば、そうでなく不変なものもある。外部からオブジェクトに動作を加えることによって、その属性値が変化する。Smalltalk80のメッセージパッシングモデル上で考えると、あるオブジェクトに動作を加えることは、そのオブジェクトにメッセージを送信することに対応する。このようなモデルのもとで、オブジェクト、クラス、属性、メッセージといったソフトウェアの構成要素が自然言語文中のどの要素、つまりどの語句に対応するかが問題となる。以下の節では、仕様文中に出現する名詞と動詞がどのような対応を持つかについて分析する。分析には文献(6)-(8)の仕様記述文例集を用いた。

2.1 名詞

名詞の種類は以下のように大別される。

- 1) オブジェクトを表すもの
- 2) オブジェクトの集合を表すもの
- 3) オブジェクトの属性を表すもの
- 4) オブジェクトが行う動作を表すもの

1), 2)は、最も一般的な用法で、固有名詞や定冠詞のついた普通名詞などが1つのオブジェクトを表し、普通名詞がある共通の性質を持ったオブジェクトの集合、つまりクラスを表すのに使用される。3)の例として 'temperature of the ro

om'や'speed of the motor'のtemperature (roomの属性)、speed (motorの属性)があげられる。このタイプの名詞は、オブジェクトを表す名詞の修飾を受ける形で文中に出現することが多い。4)の例は、動作を表す動詞から派生した名詞(例えばtransmission, receiptなど)で、動名詞やto不定詞の形で出現することも多い。to不定詞の例として、

These (pressing the buttons) cause the lift
to visit the corresponding floor.⁶⁾ ... 2.1)

のように使役動詞causeの補語がある。

2.2 動詞

1つの動詞は、ある限定された文型(もちろん唯一とは限らない)で用いられ、その文型のスロットにうめられる名詞の種類も制限されている。ここでは、動詞がスロットにうめられた名詞に対して、どのように機能するかという立場から、以下の1), 2), 3)のタイプに分けて考える。

1) 関係動詞

関係動詞は、オブジェクト間の関係を表す動詞で、必ず主語と目的語をとる。両者ともオブジェクトを表す名詞が使用される。オブジェクト間の関係は、オブジェクトの生成や消滅などのように動的に変化することもある。以下の例は、haveが関係動詞として用いられた例で、part_of関係を表している。

Each lift has a set of buttons,
one for each floor.⁶⁾

この文は、1つのliftオブジェクトが複数個のbuttonオブジェクトと関係づけられ、しかもそれらの中のbuttonオブジェクトは各々floorオブジェクトに関係づけられていることを述べている。

2) 状態動詞

状態動詞は、1)オブジェクトが自分の内部状態を表すものとしてどのような属性を持っているかや、2)その属性値がどのようなになっているかを表す。前者の役割を持つ動詞は、いわばあるオブジェクトのクラスが持つ属性の宣言を行っており、必ず目的語をとる。主語はオブジェクトを表す名詞、目的語は属性名を表す名詞をとる。この動詞の例としてhaveなどがあげられる。このように、1つの動詞は1)~3)までの唯一のカテゴリに属するというわけではない。後者の動詞は、1つのオブジェクトの属性値が現在どうなっているかを表し、その代表的な例はbe動詞である。これらは、主語と補語をとり、主語はオブジェクトもしくは属性名を表す名詞、補語は属性値を表す語句(名詞、形容詞、前置詞句など)がくる。例えば、

The lift should remain at its final destination.⁶⁾

中の動詞remainは、liftの所在を表す属性値がどうなっているかを述べている。

3) 動作動詞

動作を表す動詞は、格要素となっている語句が表しているオブジェクトの属性値を変化させたり、オブジェクト間の関係を変化させたりする動詞である。変化後の状態が前の状態と変わらなくても、変化させようという意志があれば、それは動作と見なす。ソフトウェアの世界では、原則的に動作主のいない動作は存在しない。というのは、ソフトウェアには「自然に状態が変化する」という概念がなく、必ずあるモジュールが演算を行って状態を変化させていくからである。自然言語による動作の記述では、動作主がいなかったり、明示されていないことが多い。従って、このような記述に対して動作主を補ってやるのが形式化への前段階ということになる。以下では、動作主は常に存在し、それは構文上は主語として現れることを想定し、議論を進める。従って、動作を表す動詞の受動態は能動態に直し、動詞型の分類を行っていく。

動作主に対する動作対象とは、その動作によって属性値や関係が変化するオブジェクトと考えられる。従ってオブジェクトモデルにおいては、動作に対応するメッセージの宛先が動作対象で、送信元が動作主に対応すると考えられる。一般的には動作対象を表す語句は、構文上は目的語として出現するが、このようなモデルでは他の語が動作対象を表すと考えたほうがよい場合もある。例えば、

The buoys collect air and water temperature through a variety of sensors.³⁾ ... 2.2)

という文では、むしろcollectの動作対象がsensorオブジェクトで、目的語temperature (air, waterオブジェクトの属性名)はcollect動作の出力パラメータと考えたほうが自然である。以上より動作対象を表す語は、構文上目的語または必須格となっている前置詞句の目的語(他動詞として扱う)として現れたり、主語として現れたり(この場合は本稿ではその動詞を自動詞として扱う)現れたりすると考える。もちろん、1つの動詞型に複数個の動作対象が出現したり、動作主自身もその動作によって属性値や関係が変化することもある。ここでは、変化するオブジェクトが動詞型の中でどの語となって出現しているかに注目して分類を進める。変化の内容、つまり動詞の意味までは考慮しない。以下、目的語と称する語句は動作対象となっている前置詞句の目的語を含む。構文上の語句の果たす役割より、格構成要素を主語(Subject)、目的語(Objective)、補語(Complement)に分類して、動詞型とそれらの変化性を表したものを図1に示す。action, attrib, objectは、各々動作を表す名詞、属性を表す名詞、オブジェクトもしくはオブジェクトの集合を表す名詞を示している。図中の数字は分類番号、空白はそのようなパターンが対象とした文例には出現しなかったことを表す。オブジェクトもしくはその集合を表す名詞は、変化があるかどうかでさらに分類した。属性値や関係の変化がある名詞は+、そうでないものは-をつけて示した。例えば、1-1のカテゴリに属する動詞は、動作主である主語しか格要素としてとらず、しかもそれが動作の結果、変化することを表している。どのような属性値や関係に変化するかは、その動詞の意味に含まれている。これに対して、1-3の文型はS+V+Cの文型に属するものであり、主語が表しているオブジェクトが動作後にど

のような属性値となるかは、補語によって表されている。それゆえ、1-1の動詞型は、1-3の型で代替することも可能である。図中の()はそのような関係を表している。

動作対象を複数個とする動詞は、S+V+O+Oの文型のように複数個の目的語をとる動詞型となる。この動詞型は、1-5や1-6の動詞型の拡張として考えられる。つまり、1-5、1-6の動詞型にオブジェクトを表す名詞が目的語として追加され、これらのオブジェクトはすべて動作によって変化する。分析の対象となった仕様文には、複数個目的語をとる動詞型はこのようなパターンしか現れなかった。

3. 自然言語仕様とオブジェクトモデルとの対応

Balzerらも指摘しているように⁴⁾自然言語文は、省略や語句の取り違えなどの種々の非形式的な不正確性を含んでいるため、まずこれらを正確にする操作が必要である。特に本手法では、動詞で表された動作の動作主と動作対象を明確にしておかなければならない。また、自然言語仕様は、オブジェクトモデルをあらかじめ想定して記述しているわけではないため、名詞がオブジェクトに、動詞がメッセージに直接対応するとは限らない。本節では、前節で行った分類をもとに、単語をどのようにしてオブジェクトモデル中の各構成要素に対応づけ、形式的仕様のモジュール構造を求めるかについて述べる。

3.1 格要素とメッセージ送受信

動作動詞がメッセージを表すとすると、その動詞を含む文は、Smalltalk80のメッセージ式に対応すると考えることができる。このとき、文中の格要素として出現する名詞が表すオブジェクトのうち、どれがメッセージの送信先で、どれが受信側となるかが問題である。もちろんこれらの語句が文中では省略されていることもありうる。主語が動作主、目的語が動作対象であるという規則のもとでは、動詞がタイプ3.1-5であり、動作対象に該当する目的語が1つの場合に限り、主語がメッセージの送信側、目的語が受信側と考えられる。しかし、動作対象を表す目的語が複数個ある場合や他のタイプの動詞については問題である。また、メッセージと共にデータが運ばれるため、データの流れが動詞に関してどうなっているかも問題である。例えば、前節2-2)の自然言語文例では、動作主が主語のbuoy、動作対象がsensorであると考えられるが、データ(water temperatureなど)の流れは、sensorからbuoyである。これは、buoyがsensorにデータ要求のメッセージを送り、それを受信したsensorオブジェクトがデータをbuoyオブジェクトに送信するという一連のメッセージ交換を表していると解釈することも可能である。また、データフロー通りsensorが動作主となって、動作対象buoyにデータを送信する動作と解釈することも可能である。どちらの解釈をとるかは、'collect'で表されるデータを受け取る操作の必要なタイミングをどちらのオブジェクトが知っているかによる。例えば、値を必要とするタイミングをbuoyのみが知っているのであれば前者を、sensorのみが知っている、もしくは相手のオブジェクトの状態に関係なく自由に送ってよいならば後者を選択する。以上のように、状態変化が起こるオブジェクトを表す名詞とその間のデータの流れを考慮して、動詞がどのようなメッセージ送信に対応するかを判断する。図1中の分類番号が2で始まるカテゴリの動作動詞、つまり動作を表す名詞を格要素として含んでいる動詞については、次節で考察する。以下に1のカテゴリの動詞についての対応のさせ方を述べる。

タイプ1.1

主語が送信側であり、かつ受信側となる。
(自分自身にメッセージを送る)

タイプ1.2

主語となっている属性を持っているオブジェクトが送信側

		subject				complement	
		action	attrib.	object			
				-	+		
ob- ject ive	action	2-3 (2-1,4)	1-2 (1-3)		1-1 (1-3)	action attrib.	1-3
	attrib.	2-4 (2-6)		2-1			
	-			1-7 (1-4,9)	1-8 (1-3)		
	object			1-4			
				1-5	1-6		
com- ple- ment		2-5 (2-6)					
		action	2-6		2-2		
			attrib.			1-9	

図1 動作動詞の分類

で、かつ受信側となる。タイプ1. 3のオブジェクトが文中に出現していない形と見ることができる。

タイプ1. 3

主語が送信側であり、かつ受信側となる。属性値を表している補語をメッセージのパラメータとするか、それとも動詞の定義内に含めてしまうかの選択がある。

タイプ1. 4

以下の2通りの解釈がある。

- a)状態が変化するオブジェクトが含まれていないため、メッセージとはしない。
- b)その動作に関する動作間関係が存在するならメッセージとする。後で述べる送受信オブジェクト決定規則Dに従って送信オブジェクト、受信オブジェクトを決める。

タイプ1. 5

- a)状態が変化する目的語が1つのとき
主語、状態が変化する目的語が送信オブジェクト、受信オブジェクトとなる。どちらが送信側になるかは、決定規則Dに従って判定する。
- b)状態が変化する目的語が複数あるとき
主語、目的語の中から決定規則Dに従って、送信オブジェクトを1つ決め、残りのオブジェクトをすべて受信オブジェクトとする。送信オブジェクトから1つの受信オブジェクトへのメッセージへ分解する。ただし、送信オブジェクトも状態変化を起こすときは、送信オブジェクト自身へ送信するメッセージを追加する。分解した動作がどのタイプに属するかを判定し、各々の処理を行う。

タイプ1. 6

タイプ1. 5と同じ。ただし、主語も状態変化を起こすため、主語を受信オブジェクトとするメッセージが追加される。例えば、

Users check out a copy of book from the library^o

という文に対して、送信オブジェクトをuserとし、

1)userからlibraryへのメッセージ

パラメータはbookで、libraryの所有本からbookを削除する効果を与える。

2)userからbookへのメッセージ

bookの状態をnot available、もしくは所有者をlibraryからuserにする。

3)userからuserへのメッセージ

userの状態をbookを借りている状態にする。

の3つに分割する。

タイプ1. 7

目的語で表された属性を持つオブジェクトを目的語として補い、そのオブジェクトの状態変化の有無によって、タイプ1. 4(変化がないとき)、または1. 9(変化があるとき)の処理を行う。

タイプ1. 8

1. 3の処理を行う。このタイプの動詞型は、目的語で表された属性を持つオブジェクトが主語と同じオブジェクトとなっている場合しかなかった。

タイプ1. 9

主語を送信オブジェクト、目的語を受信オブジェクトとする。補語の扱いについては、タイプ1. 3と同様である。

[送受信オブジェクト決定規則D]

タイプ1. 4~7の動詞型では、前述のように構文要素のみからは、送信オブジェクト、受信オブジェクトを特定できない。メッセージのパラメータとなるデータの流れを考慮して、送信オブジェクト、受信オブジェクトを以下のように判断する。

a)データの流れがないときは、主語を送信オブジェクト、目的語を受信オブジェクトとする。

b)データが主語で表されるオブジェクトから目的語へと流れる場合は、主語を送信オブジェクト、目的語を受信オブジェクトとする。

c)データが目的語から主語へと流れる場合は、この節の前半で述べたcollectの例のように2つの解釈のうちのどれかとする。

分類番号が2で始まる動詞は、動作間関係を表す場合もあるため、次節で述べる。

3. 2 動作間関係

実際の仕様記述では、動作の内容だけでなく、システム中の動作間関係(例えば時間的な順序関係など)も規定している。従って、個々のメッセージに対応する動詞ごとに送受信オブジェクトを決定していても、動作間関係を記述できるようなモジュール構造とはならないことがある。オブジェクトモデルでは、動作の定義はオブジェクトごとにまとめられるため、動作間関係を記述するには、それらの動作に関与する共通のオブジェクトが存在しないと、記述できない。本節では、動作間関係の種類を分類し、それに応じて、3.1で決められた送受信オブジェクトを変更する戦略について述べる。

3. 2. 1 動作間関係を表す動詞

動作間関係は、副詞句・節以外に動詞によっても記述されることがある。図1の分類番号2の動詞(格要素に動作を表す名詞をとる)は、動作を表す以外に動作間関係を表す可能性のある動詞である。特に、2-6のように、動作を表す名詞が主語(動作主)になっているような動詞は、動作間関係を表す場合がほとんどである。例えば、前節2-1)の文例では、press動作がvisit動作を引き起こすことを表しているが、本動詞causeが動作を表す動詞であると考えられるよりも、動作間関係を表すと考えたほうが自然である。これをもとにタイプ2に属する動詞型について以下のような戦略をとる。

1. 主語がオブジェクトを表す名詞のときは、動詞は動作、つまりメッセージを表すものと解釈する。

タイプ2. 1

送信オブジェクトは主語、受信オブジェクトは目的語によって表されている動作の動作対象(動作の目的語)である。

タイプ2. 2

送信オブジェクトは主語、受信オブジェクトは目的語Objである。補語で表されている動作Compもメッセージを表し、その送受信オブジェクトはCompのタイプに応じて3.1の規則によって決められる。送信オブジェクトはObj、受信オブジェクトはCompの動作対象となるのが普通である。

2. 主語が動作を表す名詞のときは、原則として動詞はメッセージではなく、メッセージ間関係を表しているものとする。

タイプ2. 3

タイプ2. 1の変形か、タイプ2. 4の変形としてとらえる。タイプ2. 1の変形の場合は、主語となっている動作の動作主から動作対象となっているオブジェクトへのメッセージとなる。タイプ2. 4ととらえるときは、他の動作を表す語(目的語となる)が省略されているとみる。

タイプ2. 4

主語である動作(以下、A_{sub}と略す)と目的語として出現している動作(A_{obj})の関係を表しているとし、メッセージとは解釈しない。動作間関係の処理については3.2.2で述べる。

タイプ2. 5

タイプ2. 4の目的語A_{obj}が本動詞として格上げされた動詞型とみなすことができる。従って、本動詞は動作A_{obj}と主語で示された動作A_{sub}との時間的な関係、およびA_{obj}の内容の両方の意味を含んでいる。処理は、タイプ2. 4と

同じであるが、一般的には目的語ObjがそのAobjの動作対象オブジェクトとなる。

タイプ2. 6

タイプ2. 4のAobjの動作主オブジェクトが目的語Objとして明示された動詞型である。従って、タイプ2. 4と同じ処理をする。

以上のような手順により、分類した動詞型より、メッセージとその送信オブジェクト/受信オブジェクト（の候補）、パラメータを決定する。

3. 2. 2 動作間関係の種類

動作間関係は、以下の3つに大別される。

1)因果関係

因果関係は、動作の時間的な順序関係を表す。つまりある動作が行われたとき、もう一つの動作が必ずその後に行われるという関係である。例えば、

Speed readings are taken every 30 seconds.
 という文は、read動作同士の因果関係を表している。

2)禁止関係

あるインターバルにおいて、その動作が絶対に行われない関係である。相互排他などがこの例である。

Minimum time for furnace restart
 after prior operation is 5 minutes.⁶⁾
 は、前の動作から5分間、start動作を禁止することを表している。

3)優先関係

割り込みなどの動作間の優先順位を表す関係である。

This (broadcasting informations in response to requests from passing vessels) takes priority over the periodic broadcast.⁸⁾
 という文は、ある種のbroadcast動作が定期的なbroadcast動作よりも優先することを述べている。

動作間関係を2項関係に分解してとらえたときに、2つの動作に関与するオブジェクトの関係は、a)一方の動作の受信オブジェクトと他方の送信オブジェクトが同じ、b)送信オブジェクトが同じ、c)受信オブジェクトが同じ、d)無関係の4つの場合(図2参照)がある。これら4つの場合と上記の動作間関係の種類によって、送受信オブジェクトを変更したり、望ましくないパターンを消去したりする規則を用意する。変更の際には、できるだけ受信オブジェクトを変更しないようにする。因果関係の場合は、図2のように1)送信オブジェクトが同じオブジェクトになるようにするか、2)時間的に後続する動作の送信オブジェクトを、先行する動作の受信オブジェクトと同じにする方法がある。2)の場合は、先行する動作を受けたオブジェクトが後に続く動作を発動する形で、因果関係が記述される。他の2つの関係は、受信オブジェクトが同じになるようにする。

3. 3 抽出過程

モジュール構造抽出の過程をまとめると以下ようになる。

- 1)自然言語仕様から動詞を抽出し、2節に従って分類する。このとき、動作主、動作対象候補となる主語、目的語(もしあれば)を明確にしておく。名詞の分類は、動詞の分類と同時にされる。
- 2)仕様中から動作間関係を抽出し、分類する。このとき、タイプ2の動詞が動作を表すか、動作間関係を表すかにも注目する。
- 3)不要な名詞や動詞(例えば、類義語など同じことを述べている語)を消去する。
- 4)動作を表す動詞について、3.1, 3.2.1で述べた規則を適用し、送信、受信オブジェクトを決める。
- 5)各動作間関係について、3.2.2で述べた規則を適用し、その関係に関与している動作の送信、受信オブジェクトを変更したり、動作間関係を変更したりする。

4),5)においては、適用可能な規則が複数個存在するため、規則の選択、および適用の是非は人間の判断にゆだねられる。この抽出過程を、各規則までも含めて英語による仕様で表現し、この手法自身を適用し、形式的な仕様で記述することも試みた。さらにこの仕様から抽出過程を支援するツールのプロトタイプを作成した。これらの詳細は、紙面の都合上別稿にゆずる。

4. 応用例と評価

本節では、実際に本手法を適用した例について述べる。例として用いたのはliftの制御問題で、非形式仕様の原文を図3に示す。

4. 1 Pure TELLの言語仕様

形式的仕様記述言語Pure TELLの記述の単位は、限定された自然言語(英語)の単文である。その詳細化手法は、記述中に出現する問題固有の単語がモジュールに対応すると見なし、これらの単語の意味を限定自然言語で次々に定義していくという手法を用いている。従って、モジュール記述は自然言語の語句を定義する形式となっている。モジュールが1)静的であるか動的であるか2)オブジェクトであるか機能/動作であるかという観点からとらえ、各々用意された記述形式に従って記述していく。このようなとらえ方により4つの語句定義形式が用意されており、これらは以下のように定義する単語の構文カテゴリを反映している。

	データ、オブジェクト、クラス	機能/動作
静的	普通名詞(クラス) 固有名詞(値)	名詞、形容詞
動的	普通名詞(動的クラス) 固有名詞(オブジェクト)	一般動詞

本節で関連が深いのは、このうち動的クラス定義と一般動詞を定義する動作定義である。

動的クラス定義は、内部状態を持つオブジェクトをある共

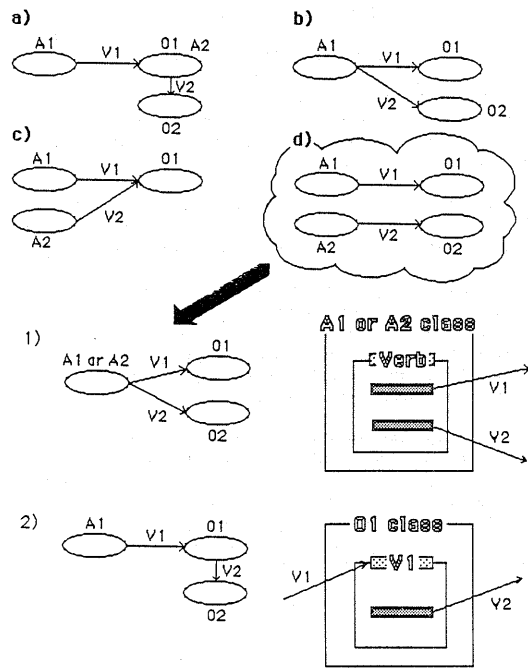


図2 因果関係の処理

通の性質に着目してクラスとして定義するための記述形式である。その構文は、以下のようにになっている。

クラス名の宣言

- { 属性の宣言の並び }
- { 動作定義の並び }
- { オブジェクト生成時に行う動作の記述 }
- { 相互作用の記述 } ({ } は省略可能を表す)

動的クラス定義中にネストされている動作定義が、このクラスに属するオブジェクトに対して許されている動作を定義する部分であり、Smalltalk80のメソッドに該当する。相互作用記述部は、これらの動作関係のうち 2), 3) のカテゴリに属するものを記述する。例えば、1つのオブジェクトに同時に複数の動作要求があったとき、それらの優先順位や排他制御といった制約を記述する。

4. 2 lift問題⁶⁾の記述

本手法を図3のliftの制御問題に適用し、形式仕様を構築する例について述べる。まず各文に含まれている動詞を、省略されている主語、目的語を補いながら、第2章であげた型に分類していく。各文の本動詞だけでなく、節や句中に出現している動詞の中で、システムの構成要素となり得ると思われるものについても分類を行う。動詞や名詞の分類は、人間の自然言語理解の能力に依る部分が極めて大きく、必ずしも結果が唯一になるとは限らない。図3に分類結果を示す。図中のagentはその動詞に対する主語を、objは目的語を示す。この過程で、図3の文中には全く出現していない語passengerを常識的な判断によって導入した。このpassengerや、仕様中に出現していてもその動作や性質の全く述べられていないsite managerがlift systemを取り巻く外部環境となる。一般的にこのような語が表すオブジェクトが外部環境の候補になると思われる。分類番号の付けられていない動詞は、ソ

フトウェアモジュールとはならないと判断された動詞である。

分類中に問題となったのは、requestである。requestは、図3の8)の文には動詞として出現し、10)や12)など他の文では名詞として出現しており、これらrequestの指す意味は同じである。requestを動作を表す動詞(タイプ3の動詞)としてとらえるか、それともオブジェクトや属性を表す名詞としてとらえるかが問題である。ここでは、requestを3.2-1のタイプの動詞(目的語はup-liftなどのliftの移動動作)としてとらえ、そのrequestを目的語として使用している動詞serviceを、request動作が行われた後に実際にrequestが求めている動作を実行する動作であると考えた。図3の14), 15)の文より、request動作はfloorからのものとliftからのものと2種類あり、各々動作主はpassengerであると考えられる。

動詞の分類後、動作を表す動詞に対して、他の動詞と動作間関係があるかどうかを調べていく。16)の文中のタイプ2-6の動詞causeは、press動作とsend動作の因果関係を表している。このように語句として出現している以外に、文中では明示されていない関係についても注意を払う。例えば、visitやmove, move動作は、必ずservice動作の後に行われる。つまり、これらには因果関係がある。ここまでの結果を図5に示す。楕円がオブジェクトを、矢印が動作を表す。矢印の出所が動作主、先が動作対象オブジェクトを表す。次に不要な名詞、動詞を消去していく。例えば、requestとpress, moveとserviceは同義、moveはvisitに含まれているため、各々の前者を削除する。残った動作動詞について3.1, 3.2.1の規則に基づいて送受信オブジェクト(の候補)を決定する。visitは、1-6の規則に基づいて、move(動作対象がlift)とreach(動作対象がfloor)の2つの動作に分割される。次に動作間関係について、3.2.2の規則をもとに動作とオ

An n lift system is to be installed in a building with m floors. — 1)

(1)

The lifts and the control mechanism are supplied by the manufacturer. — 2)

The internal mechanisms of these are assumed (given). — 3)

The problem concerns the logic

to move lifts between floors according to the following constraints : — 4)

(3.1-5 agent : control mechanism obj:lift, floor)

1- Each lift has a set of buttons, one for each floor. — 5)

(1)

These illuminate when pressed

(3.1-1 agent, obj:button) (3.1-5 agent:passenger obj:button)

and cause the lift to visit the corresponding floor. — 6)

*(3.2-6 A_{sub}:press(agent:passenger obj:button) A_{obj}:visit)

(3.1-6 agent:lift obj:floor)

The illumination is cancelled

when the corresponding floor is visited by the lift. — 7)

(3.1-7 agent:lift obj:button) (3.1-6 agent:lift obj:floor)

2- Each floor has two buttons (except ground and top floor),

(1)

one to request an up-lift and one to request an down-lift. — 8)

(3.2-1 agent:passenger A_{obj}:up-lift(obj:lift))

(3.2-1 agent:passenger A_{obj}:down-lift(obj:lift))

These buttons illuminate when pressed. — 9)

(3.1-1 agent,obj:button) (3.1-5 agent:passenger obj:button)

The illumination is cancelled

(3.1-7 agent:lift obj:button)

when a lift visits the floor

(3.1-6 agent:lift obj:floor)

and is either moving in the desired direction.

(3.1-1 agent,obj:lift)

or has no outstanding requests. — 10)

(2)

In the latter case, if both floor buttons are pressed,

(3.1-5 agent:passenger obj:button)

only one should be cancelled. — 11)

(3.1-8 agent:floor obj:button)

The algorithm to decide which to service

should minimize the waiting time for both requests. — 12)

(3.2-1 agent:control mechanism obj:request(obj:lift))

3- When a lift has no requests to service,

(2.2)

it should remain at its final destination with its door closed

(2.2)

(3.1-5 agent:lift obj:door)

and await further request (or model a 'holding floor'). — 13)

(2.2)

4- All requests for lifts from floors must be serviced eventually,

with all floors given equal priority. — 14)

(3.2-1 agent:control mechanism A_{obj}:request(obj:lift))

5- All requests for floors within lifts must be serviced eventually,

(3.2-1 agent:control mechanism A_{obj}:request(obj:lift))

with floors being serviced sequentially in the direction of travel.

— 15)

(3.1-5 agent:control mechanism obj:lift(obj:request))

6- Each lift has an emergency button which, when pressed

(1)

(3.1-5 agent:passenger obj:button)

causes a warning signal to be sent to the site manager. — 16)

*(3.2-6 A_{sub}:press(agent:passenger obj:button) A_{obj}:send)

(3.1-5 agent:lift obj:site manager)

The lift is then deemed 'out of service'. — 17)

(2.2)

Each lift has a mechanism to cancel its 'out of service' status. — 18)

(3.1-3 agent:lift obj:lift)

図3 Lift問題と動詞の分類

プロジェクトとの関係を再検討する。例えばpress動作とservice動作は、因果関係にあるため、serviceの動作主を各buttonに置き換える。以上のような手順によって、各動作およびそれらの関係をstepwiseに明確化していくと、図5の関連図が得られる。この関連図をもとに各オブジェクト及びそれらのオブジェクトに許されている動作を形式的に定義していく。その結果、図6のPure TELLによる仕様を得る。

ここで問題となったのは、12)と15)の文で示された要求の形式化である。上、下どちらのpress動作を先に選んでserviceするかは、その時点での状態のみからは決定できず、未来の状態の情報、つまり将来どのようなpress動作がはいってくるかという情報も必要とする。ここでは、この制限を無視し、任意に処理すべきpress動作が選択されるとする。

15)の文は、service動作はpress動作が行われた順に処理されていくのではなく、liftの動いている方向にも応じてserviceされる順序が決っていくことを述べている。従ってこの制約を形式化するには、liftごとに要求されたservice動作を記憶しておく必要がある。そのため、図6に示すように、図5の下位レイヤのオブジェクトとしてlift request queueを導入して形式化する。これによってservice動作は、queueへの登録動作enqueに置き変わる。

図3の文中には、floorからのpress動作に対してn個のうちのどのliftがserviceされるかに関する記述が含まれていないが、常識的に待ち時間ができる限り少なくなるように考える。ここでは、pressの出たfloorに向かうlift(停止しているliftも含める)のうち最も近いものを選択する(chosenの定義)。このことは、liftは要求されたserviceがすべて実行されるまで、現在の移動方向を変えないという制約を置いていることになる。しかしこのような選択方式は、必ずしも待ち時間が最小になるというわけではない。

4.3 議論

本手法では、ソフトウェアをオブジェクトモデルでとらえ

ているのにもかかわらず、オブジェクトを表す名詞ではなく、動詞に注目してその動詞型を分類することにより、システムを構成するオブジェクトを抽出するという手法をとっている。これは、

1)動詞を分析する過程でその格要素となる名詞も分析しているため、各々独立した分析を行うのよりも、相互の分析結果を利用し合い、より正確な分析が可能となる。

2)自然言語の意味の中で、動詞は最も重要な比重を占めているため、動詞を中心に分析することによってより正確な自然言語理解が得られる。

という利点が考えられる。しかし名詞の分析は、あくまでも動詞型の分類の中に埋め込まれているため、名詞の分類の処理が単純になるという反面、多彩な自然言語の表現に対処しきれぬかどうかの問題も残されている。

自然言語による非形式仕様では、システムの詳細な部分の記述、特に常識的と思われる事項は省略されるのが常である。文中に出現している記述は、省略されている記述よりもより上位の概念と考えることができる。省略されている記述は、読者には隠蔽されている記述と見なす。本手法では、省略されている部分の記述の解析は後にまわされ、記述中に出現している、つまり上位レベルのオブジェクトと動作がまず抽出され、各オブジェクト内部を仕様化していく過程で省略されている部分が補われ、下位レベルのオブジェクトや動作が導入される。lift問題では、下位レベルのオブジェクトとしてlift request queueが導入された。従って本手法では、人間の自然言語記述と同じイメージでの自然な階層レイヤ構造が得られることになる。もちろんこのような階層構造は、自然言語仕様の書き方にも依存しているが、その記述が自然でわかりやすい構造をしていれば、本手法で得られる構造も自然でわかりやすいものになると考えられる。

5. むすび

自然言語で記述された仕様から実行可能な仕様、もしくはは

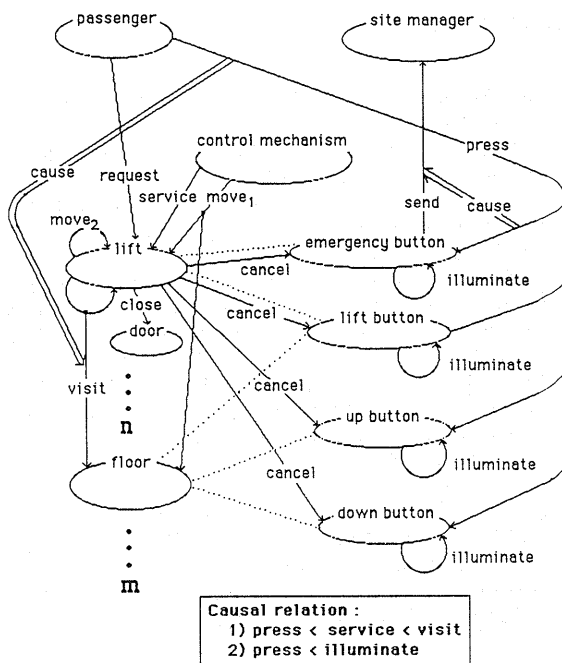


図4 オブジェクト、動作と動作間関係

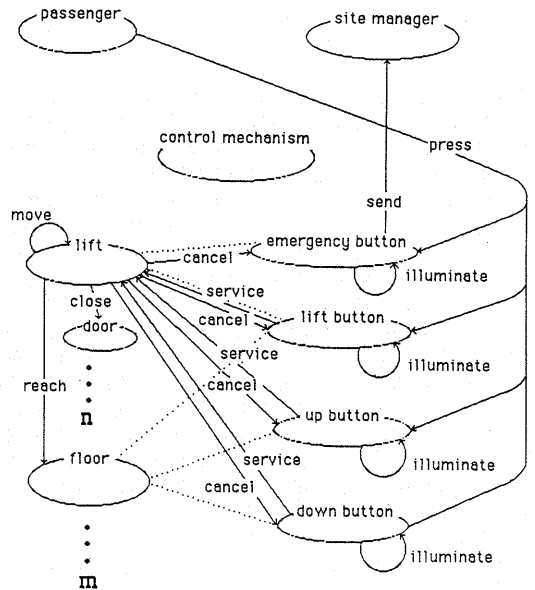


図5 処理後

Lift
 has location : floor
 direction : up, down, or stopped
 status : out of service or on service

Lift L moves
 means that

- 1) Repeat until the lift-request-queue of lift L is empty or the status of lift L is out of service
- 1-1) If <floor F, direction D> is included in the lift-request-queue of L, then
 - 1-1-1) Lift L stops.
 - 1-1-2) Lift L opens the door of lift L.
 - 1-1-3) Lift L closes the door of lift L.
 - 1-1-4) Lift L dequeues <floor F, direction D> from the lift-request-queue of L.
 - 1-1-5) Lift L cancels the lift-button for floor F of lift L.
 - 1-1-6) If direction D is up, then Lift L cancels the up-button of floor F.
 - 1-1-7) If direction D is down, then Lift L cancels the down-button of floor F.
- 1-2) Lift L moves at 1 step.
- 1-3) where
 - 1-3-1) floor F is the location of lift L.
 - 1-3-2) direction D is the direction of lift L.
- 2) The direction of lift L becomes stopped.
- 3) If the illumination of the up-button of the location of lift L is off, then lift L cancels down-button of floor F.
- 4) If the illumination of the up-button of the location of lift L is on, then lift L cancels up-button of floor F.
- 5) Lift L closes the door of lift L.

Lift L moves at 1 step
 means that

- 1) The location of lift L becomes floor Fnext.
 - 2) Lift L reaches floor Fnext.
 - 3) where floor Fnext is a next floor for lift L.
- end move at 1 step ;
 ...

end lift ;

Lift-request-queue is queue of <floor, direction>

Lift-button of lift L

has illumination : on or off

Someone presses lift-button B of floor F
 means that

- 1) If the illumination of lift-button B of floor F is off, then
 - 1-1) Lift-button B of floor F enqueues floor F to the lift-request-queue of lift L.
 - 1-2) The illumination of lift-button B of floor F becomes on.

end press ;

Someone cancels lift-button B of floor F
 means that

- 1) The illumination of lift-button B of floor F becomes off.

end cancel ;

end lift-button ;

Up-button of floor F

has illumination : on or off

Someone presses up-button B

means that

- 1) If the illumination of up-button B of floor F is off, then
 - 1-1) Up-button B enqueues <floor F, up> to the lift-request-queue of lift L, where lift L is chosen for going upward to floor F.
 - 1-2) The illumination of lift-button B of floor F becomes on.

end press ;

...

Lift L is chosen for going upward to floor F

means that

- 1) The location of lift L is nearest to floor F.
- 2) The direction of lift L is up or stopped.

end chosen ;

end up-button ;

図6 Lift問題の形式仕様(一部)

プログラムを得るための研究は種々行われているが¹⁰⁾、これらはいずれも使用する自然言語を計算機処理可能であるように限定し、自然言語特有の省略表現を、あらかじめ用意した対象領域固有の知識を用いて補うという手法をとっている。固有の知識を簡単に形式化できるような対象領域では有効な手法であるが、必ずしも対象領域が、単純にその固有の知識を形式化できるもののみであるとは限らない。本論文で提案する手法は、対象領域に依存しないことをねらいとしている。さらに本手法と前述の10)などの手法を組み合わせることにより、より高品質の形式的仕様を構築することも可能であると思われる。その意味でトップレベルのモジュール構造を得る本手法は、知識ベースに格納されている仕様部品を埋め込むためのスロットを作りだしているとも見られる。

本稿では、動詞の扱いを中心に述べてきたが、他の語、例えば、名詞、形容詞、副詞などのいっそうの分析が必要である。今後は運用経験を積んで、より洗練されたガイドシステムの構築を目指したい。

参考文献

- (1)有澤：ソフトウェアプロトタイプリング、近代科学社、(1986)
- (2)Gutttag, J.V., Horowitz, and Musser, D.R. : Abstract

- Data Types and Software Validation, CACM, 21-12, pp.1048-1064 (1978)
- (3)Abbot, R. : Program Design by Informal English Descriptions, CACM, 26-11, pp.882-894 (1983)
- (4)Balzer, R., Goldman, N., and Wile, D. : Informality in Program Specification, IEEE Trans. Soft. Eng., 4-2, pp.94-103 (1978)
- (5)Saeki, M., Horai, H., Toyama, K., Uematsu, N., and Enomoto, H. : Specification Framework Based on Natural Language, Proc. of 4th International Workshop on Software Specification and Design, pp.87-94 (1987)
- (6)Problem Set, ibid, pp.ix-x (1987)
- (7)Gehani, N. and McGettrick, A.D.(eds.) : Software Specification Technique, Addison Wesley (1986)
- (8)Booch, G. : Object-Oriented Development, IEEE Trans. Soft. Eng., 12-2, pp.211-221
- (9)植松, 佐伯, 外山, 蓬萊, 榎本 : 自然言語と図を用いた仕様記述言語の構造化エディタの開発, 第33回情報処理学会全国大会, pp.761-762 (1986)
- (10)辻井, 上原 : ソフトウェア工学と自然言語処理, 情報処理学会誌, 28-7, pp.913-921 (1987)