自然言語仕様からモジュール構造を得る手法について†

佐伯元司†† 蓬莱尚幸††† 榎本肇††††

本論文では，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設計す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，その動詞が取りうる格構造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（たとえば，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．

1. はじめに

代数的仕様化技法[1]に代表される形式的仕様記述法やプロトタイピング手法[2]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．プロトタイププログラムや形式的仕様は読解性や修正のしやすさを考慮して，自然な発想に基づいた自然な構造をしていることが望ましい．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未記述の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)]．また，このような手がかりを用いて構築された形式的仕様は，元の自然言語仕様の構造を反映し，理解しやすいものとなっていることが予想される．このように，自然言語記述中の要素と形式的仕様中の記述要素との間になんらかの意味的な関係をつけておくことは，開発者間の意志伝達において重要なことである．

一般に自然言語文中に出現する名詞は，データやオブジェクトといった「もの」を，動詞は「もの」に加えられる「操作」を表す[3),8)]とし，この対応を形式仕様への手がかりとしている．しかし，これだけでは文中に出現する多くの語句から「もの」や「操作」となる語句を選択し，それらをソフトウェアのコンポーネントとしてどのようにまとめ上げるかには不十分であり，形式仕様構築のための人間の負担度はほとんど改善されていない．また，「もの」や「操作」を表す語句以外にも重要な手がかりとなる語句が自然言語文中に含まれており，それらを活用すべきである．

本論文では，より強力な手がかりとして，自然言語文(英語)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の構造を設計していくかについて考察する．ここでは，形式化のモデルとしてオブジェクトモデルを使用する．そのため，具体的にはクラスモジュール，そのクラスのメソッドモジュール，インスタンスオブジェクトが持つ属性といった構造を得ることを目的とする．モジュール構造の設計手順については，人間がどのようにして形式的仕様を構築していくかという立場をできるだけ考慮する．というのは人間が使用する手がかりおよび手法を用いたほうがわかりやすい形式的仕様ができるためである．

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，その本動詞の果たす役割はきわめて大きい．ここでとった手法は，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モジュール設計の手がかりにしようというものである．なお自然言語による非形式的仕様の分析には文献 6)～8)の仕様記述文例集を用いた．

† On Extracting Module Structures from Natural-language Specifications by MOTOSHI SAEKI (Department of Electrical and Electronic Engineering, Tokyo Institute of Technology), HISAYUKI HORAI (IIAS-SIS, Fujitsu Limited) and HAJIME ENOMOTO (Shibaura Institute of Technology).

†† 東京工業大学工学部電気・電子工学科
††† 富士通(株)国際情報社会科学研究所
†††† 芝浦工業大学

本論文で対象とする自然言語仕様は，制限のない自然言語で記述された仕様である．その記述内容が完全なもの（記述もれがなく，矛盾もない）になっているということは要求しない．したがって，本論文で述べる手法は，コンピュータによる完全自動化を目指したものでなく，構造化設計法[14]，ジャクソン法[15]，Boochのオブジェクト指向設計法[8]といった従来のソフトウェア設計法と同様に，人間が系統的にソフトウェア設計を行うための手法の1つである．コンピュータ処理可能なように自然言語の構文，意味を制限する研究[10]も報告されているが，これについては第6章で議論する．

2. 基本方針

2.1 オブジェクトモデル

ソフトウェアシステムを形式的に記述する際，どのような観点からモデル化するかということが重要である．ここでは，形式化のためのモデルとして，以下のようなオブジェクト指向型のモデル[8]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操作を加えることによって，その属性値が変化する．Smalltalk 80[12]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操作を加えることは，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．操作が加えられたことによって，どのようにオブジェクトの属性値，つまり状態が変化するかを記述したものがメソッドである．メッセージには受信オブジェクトの状態変化に必要なデータをつけて送信することもできるし，変化の結果得られたデータを送信オブジェクトへ返すこともできる．つまり，メッセージには入出力引数を付加することができる．

このようなモデルのもとで，オブジェクト，クラス，属性，メソッドといったソフトウェアコンポーネントが自然言語による仕様文中のどの要素，つまりどの語句に対応するかを考察する．

2.2 オブジェクトモデルと語句との対応

2.2.1 オブジェクトと操作

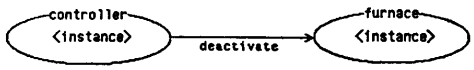
一般に，オブジェクトは名詞に，操作は動詞に対応するといわれている．しかし，すべての名詞や動詞がこれらに対応するわけではなく，実際にはその一部だけである．これら以外の語句の中にも他の種類のソフトウェア構成要素に対応するものもある．たとえば，Central Heating System[6]の仕様文中，

> Once the home temperature reaches tr+2 degrees, the controller deactivates the furnace. (2.1)

において，名詞「controller」，「furnace」はオブジェクトに，動作を表す動詞「deactivate」はfurnaceオブジェクトに対する操作に対応するが，それ以外の名詞「temperature」や動詞「reach」は各々オブジェクトや操作に対応するとするのは不自然である．「temperature」はhomeが表すオブジェクトの1つの属性，「reach」はその属性値がどうなっているか，つまりオブジェクトの内部状態を表していると考えたほうが自然である．したがって，名詞および動詞をオブジェクトモデル上でのソフトウェア概念に沿って，整理・分類する必要がある．これについては，3.1と3.2節で述べる．

2.2.2 動作を表す動詞とメッセージとの対応

一般的には，動作を表す動詞があるオブジェクトに対する操作を表していると考えられ，その動詞を含む文がメッセージの送信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．一般的には，その動作の動作主が送信オブジェクトを，動作対象が受信オブジェクトを表すものとするのが自然である．つまり，動作主オブジェクトが動作対象オブジェクトにメッセージを送り，それを受けた動作対象オブジェクトの状態が，その動詞が表している動作の意味に従って変化すると解釈できる．構文的には，通常は主語が動作主，目的語が動作対象となる．たとえば，(2.1)中の動作を表す自然言語文をオブジェクトモデル上で解釈すると，**図1**のようにな



the controller deactivates the furnace

図1 自然言語文とオブジェクトモデルとの対応
Fig. 1 Correspondence between natural-language sentences and object model.

る.

2.2.3 動作主と動作対象

実際の自然言語文においては，必ずしも図1のような対応がつくわけではない．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であるため，構文上の目的語よりも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．たとえば，

The buoys collect air and water temperature through a variety of sensors[8]. (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でもある) は collect 動作を行った結果得られる値（出力値）であると考えたほうが自然である．つまり前置詞句の目的語が動作対象となっている．また，

(The control mechanism) moves a lift between floors[6]. (2.3)

においては，move 動作によって lift と2つの floor (出発階と到着階) の3つのオブジェクトの状態が変化する．このように動作によって状態が変化するオブジェクトが文中に複数個出現したり，自動詞などのように主語や動作主自身が動作対象であったりする場合がある．さらに，動作主や動作対象を指す語句が省略されている場合もある．したがって，まず動作を表す動詞を選択するとともに，動作主と動作対象の特定が必要である．また，それらを特定することは，(2.2) の文例のようにその動作に対応するメッセージの入出力引数を決めるのにも役立つ．

2.2.4 送信オブジェクト・受信オブジェクトの決定

自然言語文における動作主を送信オブジェクトに，動作対象を受信オブジェクトにそのままましてもよいとは限らない．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の関係（たとえば時間的な順序関係など）も規定している．したがって，個々のメッセージに対応する動詞ごとに各々の動作主と動作対象から送受信オブジェクトを決定していっても，動作間の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，動作間の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ければならない．

2.3 全体の流れ

以上のような考察から，モジュール構造を抽出するまでの動詞に関するステップの流れの概略は，

1) 動作を表す動詞の抽出
2) 抽出した動詞の主語，目的語の抽出
3) 動作主，動作対象の抽出
4) 動作間関係の抽出
5) 送信，受信オブジェクトの決定

となる．

3. 名詞・動詞の分類

3.1 名　　詞

名詞の種類は以下のように大別される．

1) オブジェクトやオブジェクトの集合（クラス）を指すもの（以下，クラス名詞という）
2) オブジェクトの属性などの値や値の集合を指すもの（以下，値名詞という）
3) オブジェクトに行われる動作を指すもの（以下，動作名詞という）

1)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．2)の例として，値の集合を表す名詞 integer のほかに，'temperature of the room' 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，クラス名詞の修飾を受ける形で文中に出現することが多い．3)の例は，動作を表す動詞から派生した名詞（たとえば 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]. (3.1)

のように使役動詞 cause の補語がある．

3.2 動　　詞

簡単のために，自然言語文中で，動作を表す動詞の受動態表現は，すべて能動態に直してこれからの議論を進める．また，前置詞句であるかどうかの区別は 2.2.3 項で述べたように本質的ではないので，分類項目を少なくするためにも，以下では前置詞句の目的語もその前置詞句の役割に従って補語（例文 (3.2) の場合）や目的語（例文(2.2)，(2.3)）として扱う．

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，動詞が導く文型といった構文面だけでなく，動詞の意味として，その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意味面からの分析も行った．意味の分析は，オブジェクト指向モデルを使用するため，オブジェクトの「関係」，「状態」，「動作」といった概念と名詞に対する機能とを結びつけることによって行った．以上のような分析により，動詞を以下の(1)～(4)のカテゴリに分けて考える．

(1) 関係動詞

関係動詞は，オブジェクト間の関係やオブジェクトと属性の関係（たとえば，所有関係など）を表す動詞で，必ず主語と目的語をとる．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，所有関係を表している．

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．have 以外には，contain, consist, provide などがこのカテゴリの動詞として使用されている例があった．

このカテゴリの動詞の主語はクラス名詞，目的語はクラス名詞や値名詞となっている．上の例では，主語がクラス名詞の lift，目的語がクラス名詞の button である．

(2) 状態動詞

状態動詞は，オブジェクトの属性値が現在どのようになっているかを表すのに使用される．その代表的な例は be 動詞である．状態動詞は，主語と補語をとり，主語はクラス名詞もしくは値名詞，補語は値を表す語句（値名詞，形容詞，前置詞句など）がくる．たとえば，

The lift should remain
at its final destination[6]. (3.2)

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す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．ここでは，属性値の変化するオブジェクトが文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．変化の内容については考慮しない．語句の果たす構文上の役割より，文の格構成要素を主語，目的語，補語に分類し，さらに構成要素となっている名詞（句，節）が 3.1 節のどの分類カテゴリに属するかに注目して分類する．分類結果を**表1**に示す．action, value, object は，各々動作名詞，値名詞，クラス名詞を示している．クラス名詞がきている場合は，動作によって，その語が表すオブジェクトに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 object は #，そうでないものは後に-をつけて示した．何もつけられていないものは，どちらでもよいことを示している．たとえば，I-5 のカテゴリに属する動詞は，主語がクラス名詞で，その動作では主語が表すオブジェクトの変化は起こらない．変化の起こるオブジェクトを表すクラス名詞は目的語として必ず1つ持つ．それ以外に目的語を持っているのであれば，残りの目的語は値名詞か変化のないオブジェクトを表すクラス名詞で

表 1 動作動詞の分類
Table 1 Classification of action verbs.

分類番号		主語	目的語 #1…#n	補語
I	1	object#	(value | object-)*	(value | object-)*
	2	value		
	3	object-	(value)+	
	4	object-	object-, (value | object-)*	
	5	object-	object#, (value | object-)*	
	6	object-	object#, (object#)+, (value | object-)*	
	7	object#	(object#)+, (value | object-)*	
	8	object-	object#	value
II	1	object	action	
	2	object	object	action
	3	action		

$(\alpha|\beta)$ は α または β, α^* は0個以上の α, α^+ は1個以上の α からなっていることを表す．

表 2 動作関係動詞の分類
Table 2 Classification of action relational verbs.

分類番号		主語	目的語	補語
II	1	object	action	
	2	object	object	action
	3	action		
	4	action	action	
	5	action	object	
	6	action	object	action

ある．分類番号が II で始まる動詞は，動作名詞を含んでいることを表す．

（4） 動作関係動詞

動作の間の関係を表す動詞である．文 (3.1) の例では，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るよりも，動作間関係を表すと考えたほうが自然である．つまり cause は因果関係という動作間の関係を表している．これらの動詞は，動作名詞を格要素として含む．動作関係動詞の文型を**表 2** に示す．このうち，II-1, II-2, II-3 は動作動詞のカテゴリと同じ型で，このカテゴリに属する動詞は動作動詞，動作関係動詞の両方のカテゴリに属する可能性がある．文 (3.1) の cause は，II-6 の例である．

1つの動詞が(1)～(4)までの唯一のカテゴリにしか属さないというのではなく，同じ動詞でも文中で異なった意味で使用されているときは，各々の出現が異なるカテゴリに属することがある．つまり1つの動詞が複数の分類カテゴリを持つこともある．

4. モジュール構造の設計作業

本章では，前章で行った分類をもとに，名詞・動詞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設計するかについて述べる．1つの文のみを例にとった作業の流れの概略を**図 2** に示す．

4.1 名詞，動詞の抽出・分類

自然言語による仕様が与えられたときに，最初に行う作業は，ソフトウェアの構成要素を表すと思われる名詞と動詞の抽出・分類である．名詞は 3.1 節の3つのカテゴリ，動詞は 3.2 節の4つのカテゴリに分類する．

動詞については，細分類を行うために，その主語，目的語，補語を抽出し，格構造を調べなければならない．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，格構造を求めるには，これらを正確にする作業も必要である．この作業は，コンピュータでは不可能で，設計者が文章を理解することによってのみ行うことができる．分類作業を行った結果

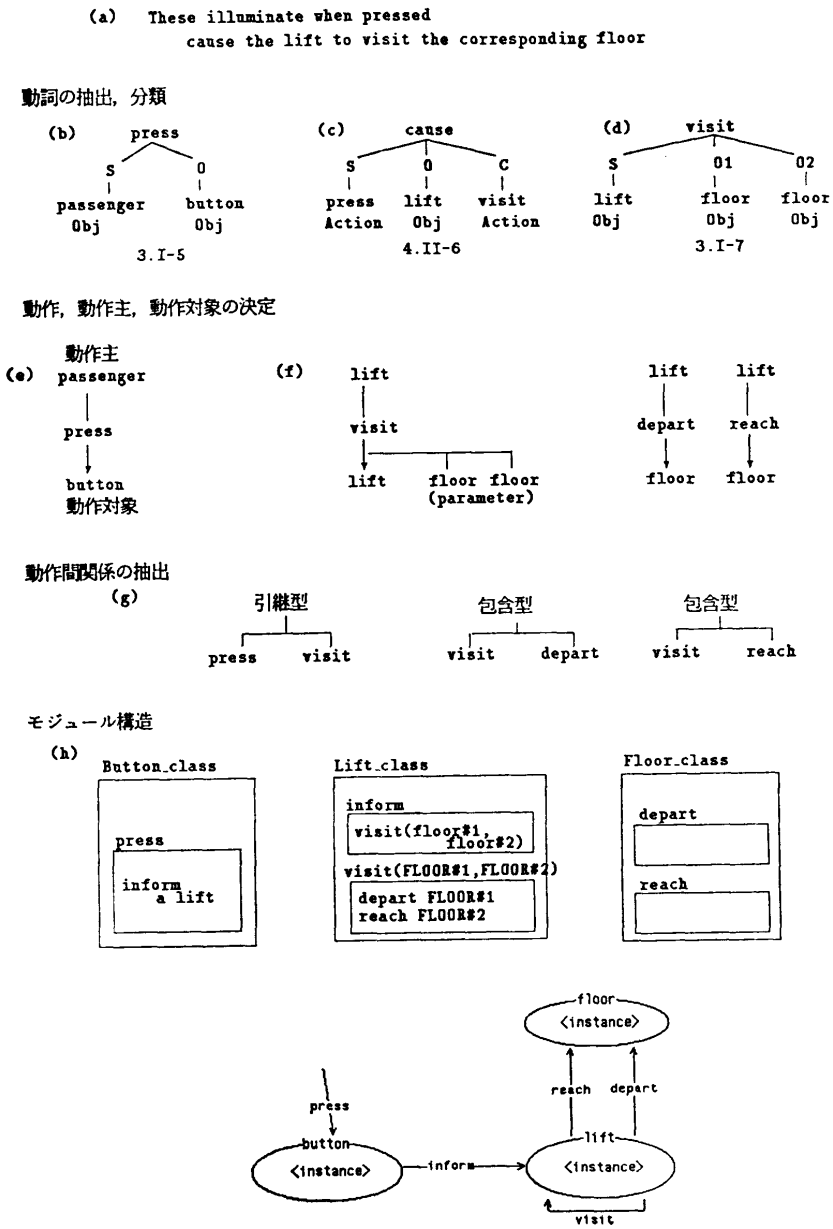


図 2 1つの文における作業の流れ
Fig. 2 Activity flow for a sentence.

が3.2節で述べたパターンや表1，表2以外のパターンになった場合は，省略や取り違えが補正されていない可能性がある．

特に，動作動詞と分類された動詞に対しては，動作主の候補となる主語，動作対象の候補となる目的語が省略されている場合がよくあるので，十分に注意を払い，これらを補ってやる．また，変化のあるオブジェクトを表すクラス名詞を目的語として含んでいないような型（I-1, I-2, I-3 など）で出現している動作動詞について，ほかに目的語が省略されていないかどうかをチェックする．

たとえば，図2の例文中の動作動詞 press については，能動態での主語が省略されているので，通行人が button を押すと解釈し，主語として passenger を補う．press の格構造は，図2（b）のようになる．主語の passenger も目的語の button もクラス名詞であり，しかも button は press 動作によって内部状態が変化する．図中の obj はクラス名詞の意味である．これと表1の分類表より，press は動作動詞（カテゴリ3）の I-5 に分類される．

この抽出・分類作業は，誰がやっても結果が唯一になるというのではなく，本手法中の他のステップの作業よりも個人による違いが現れやすい．しかし，人間の名詞と動詞の意味解釈能力を基にしているため，他の設計手法[8),11),14),15)]に比べて，個人による違いは少ないのではないかと思われる．また，どのような作業結果が得られても，4.4 節で述べる規則によって，論理的にシステムを記述できないようなモジュール構造ができてしまうことはない．

4.2 動作主と動作対象の決定

格構造を求めた動作動詞について**表3**に示した規則に従って，動作主，動作対象を決定する．図2（d）の動

表3 動作動詞の動作主・動作対象決定規則
Table 3 Decision rules for agents and objectives of action verbs.

分類番号		動作主	動作対象	引数
I	1	主語	主語	目的語，補語
	2	値名詞 value を属性として持つ object	動作主と同じ	
	3	主語	値名詞 value を属性として持つ object で状態が変化するもの 変化する object が複数個あるときは，これらに対し6の処理を行う	残りの目的語
	4 1) 動作間関係があるとき	主語	目的語の中の object-のどれか1つ	残りの目的語
	2) 動作間関係がないとき	動作ではないとして削除する value を object-の属性とする		
	5	主語	目的語の中の object#	残りの目的語
	6	主語	目的語の中の object# のどれか1つ（これをAとする）	残りの目的語
		A以外の目的語の残りの各 object# B_i について以下のような動作を追加する．		
		A	B_i	A, B_i 以外の目的語
	7	主語	主語，目的語の中の object# のどれか1つ（これをAとする）	
		A以外の主語，目的語中の残りの各 object# B_i について以下のような動作を追加する．		
		A	B_i	A, B_i 以外の目的語
	8	主語	目的語	補語
II	1	主語	目的語が表している動作の動作主	
	2	主語	目的語	
		補語が表している動作については		
		目的語	補語が表している動作の動作対象	
	3	主語が表している動作の動作主	主語が表している動作の動作対象	

作動詞 visit は，I-7 タイプの動作動詞で，状態が変化するオブジェクトを複数個(lift, floor, floor)持っている．状態変化の起こるオブジェクトの中からまず主語となっている lift を選び，visit の動作主，動作対象をともに lift オブジェクトとし，残りの 2 つの floor は visit 動作の入力引数とする．さらに，この 2 つの floor について各々新しい動作 depart (lift の出発階の状態を変える)，reach (lift の到着階の状態を変える) を導入し，動作主をともに lift，動作対象を各々 2 つの floor にする．この規則の適用によって，図 2 (f)のような動作に関する構造が得られる．depart, reach 動作は，lift オブジェクトが visit 動作を実行中に，lift から発動される．したがって，visit と depart, visit と reach には，因果関係 (4.3.2 項のカテゴリ iii)) があることになる．以上のように表 3 の I-6, I-7 のカテゴリの処理は，その動作によって状態が変化するオブジェクトが複数個あるために，1 つずつになるように動作を分解するものである．

表 4 動作関係動詞の動作主・動作対象決定規則
Table 4 Decision rules for agents and objectives of action relational verbs.

分類番号		関係に参与している動作＃1		関係に参与している動作＃2	
		動作主＃1	動作対象＃2	動作主＃2	動作対象＃2
II	1	(省略されている) 主語	目的語の動作主	目的語 目的語の動作主	目的語の動作対象
	2	(省略されている) 主語	目的語	補語 目的語	補語の動作対象
	3	主語 主語の動作主	主語の動作対象	(省略されている)	
	4	主語 主語の動作主	主語の動作対象	目的語 目的語の動作主	目的語の動作対象
	5	主語 主語の動作主	目的語	(省略されている) 主語の動作主	目的語
	6	主語 主語の動作主	主語の動作対象	補語 目的語	補語の動作対象

4.3 動作間の関係

動作間関係の種類を分類し，分類結果をもとに 4.1 節で決められた動作主，動作対象から送受信オブジェクトを求める戦略について述べる．動作間の関係は，スケジューリングや割り込みなどのあるシステムについては，種々複雑なものまで考えられるが，本稿ではモジュール構造を設計するという目的により，単純でかつ primitive な関係である 2 つの動作間の因果関係[11]のみを対象とする．

4.3.1 動作間関係を表す動詞

動作間の関係は，副詞句・節以外に動詞によっても記述されることがある．表 2 に分類された型に従って用意されている規則を用いて，関係に関与している動作およびその動作主・動作対象を求める．その規則を**表 4** に示す．たとえば文 (3.1) の場合，図 2 (c)のように II-6 の規則より，関係に関与する動作は，主語となっている press，補語として出現している visit で，visit の動作対象は cause の目的語の lift となる．

4.3.2 動作間関係の種類

動作間の因果関係を 2 項関係に分解してとらえると，以下の 4 つのカテゴリに大別される．

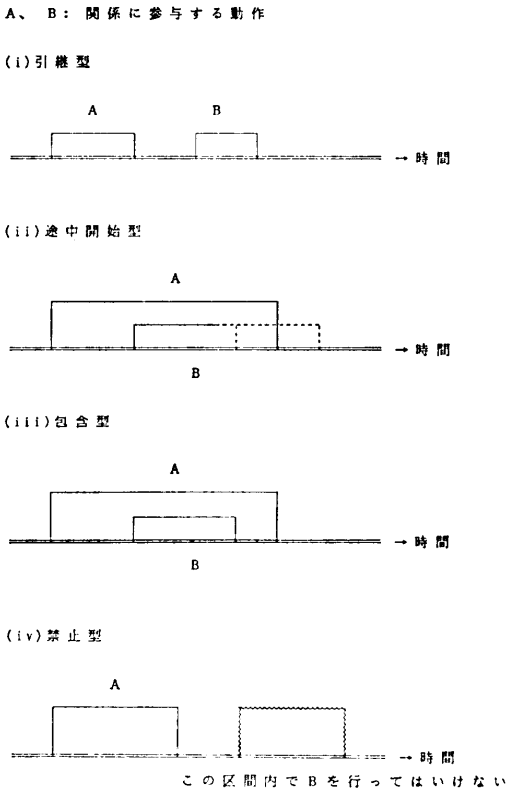


図 3 因果関係の分類
Fig. 3 Classification of causal relations.

i) 引継型：先行する動作が終了してから，次の動作が始まる (**図 3**(i))．たとえば，

The controller deactivates the furnace by first closing the oil valve and then, after 5 seconds,

stopping the motor[6]. (4.1)

という文は，close 動作が終了してから5秒後に stop 動作が行われることを表している．

ii）途中開始型：先行する動作が行われている最中に，次の動作が始められる．後続の動作が先行する動作よりも先に終了してもよいし，また終了しなくともよい（図3（ii））．

iii）包含型：先行する動作が行われている最中に，次の動作が始まり，かつ終了する．つまり後続の動作が完全に含まれている場合である．(4.1)の文例では，deactivate 動作に2つの動作 close, stop 動作が含まれていることが述べられている（図3（iii））．

iv）禁止型：あるインタバル（たとえば，動作中や動作の終了時など）において，その動作が絶対に行われない関係である（図3（iv））．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，前の動作の終了から5分間，start 動作を禁止することを表している．

4.4 送受信オブジェクトの決定

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4参照）がある．図4中で，楕円はオブジェクト，矢印は動作を表す．矢印の先が動作対象，根元が動作主オブジェクトを表している．これら4つの場合と前節の動作間関係の種類によって，望ましくないパターンが生じないように，送受信オブジェクトを決定していく規則を用意する．たとえば，d)のパターンが出現した場合，動作V1とV2の定義は，各々オブジェクトO1とO2の記述中にカプセル化されるため，V1とV2の関係を記述すべきモジュールがないことになる．つまり，d)のようなパターンは望ましくないパターンである．送受信オブジェクトを決定していく際には，できるだけ動作対象オブジェクトはそのまま受信オブジェクトになるようにする．また，関係を記述するためだけに新しいオブジェクトを導入するのはできるだけ避けることにする．

問題となるのは，i）引継型やii）途中開始型の因果関係がある場合で，メッセージを送信したオブジェクトが受信オブジェクトの動作が終了するまで待たされるような機構だけでは，うまくこれらを記述できない．Smalltalk 80 の Process クラスの fork メッセージ，ABCL[13] の Past 型メッセージのように受信オブジェクトの動作終了とは無関係に送信オブジェクトの動作が続けられる，もしくは LOOPS[9] の active value などのようにメッセージの受信以外に動作が起

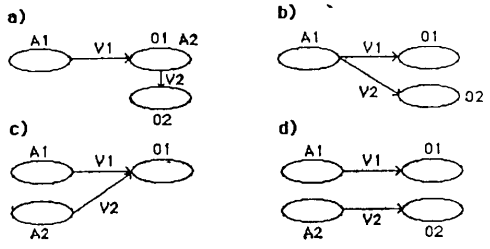


図4 2つの動作とオブジェクトの関係
Fig. 4 Relationship between two actions and objects.

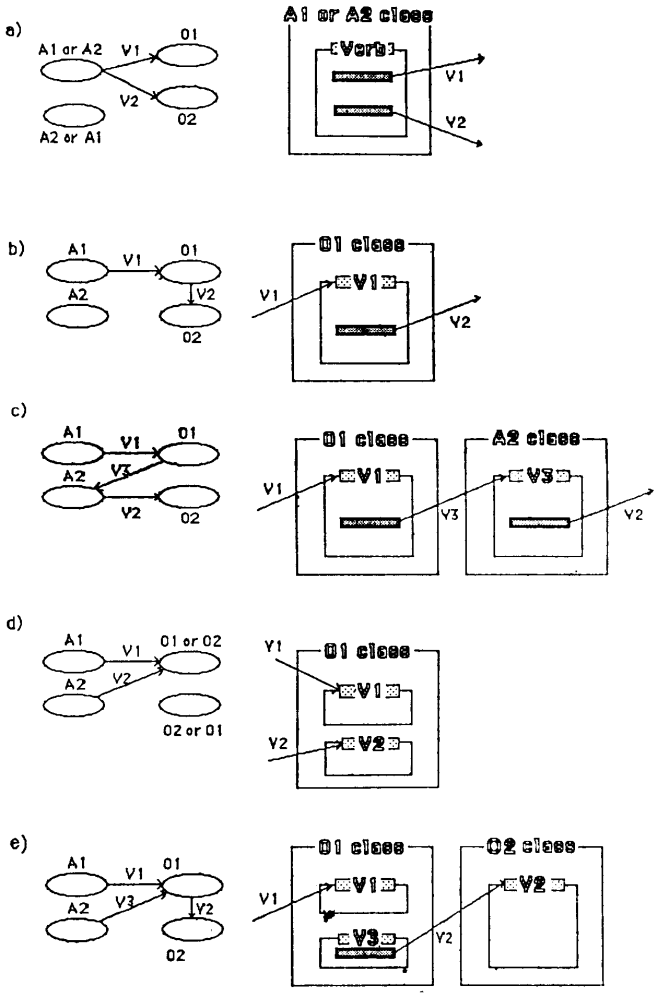


図5 送信・受信オブジェクトの決定
Fig. 5 Decision of senders and receivers.

動されるような機構がなければならない．本稿では，使用するオブジェクト指向の仕様記述言語に ABCL の Past 型のメッセージ機能が備わっていると仮定する．この選択は，本手法では本質的ではなく，上記のような機構さえあれば本手法が適用できる．

2つの動作 V1，V2 間の因果関係を記述するには，次のような5つの方法があり，これらのうちのどれが使用できるかは，因果関係の種類によって異なる．以下の(1)～(3)において V1 を先行する動作，V2 を後続する動作とする．また，**図5**は図4の(d)のパターンに基づいている．したがって，図4(a)のパターンはもともと片方の受信オブジェクトともう一方の送信オブジェクトが共通であるため，以下の(2)の操作（片方の送信オブジェクトともう一方の受信オブジェクトを共通にする操作）を施しても双方の送信，受信オブジェクトは変わらない．

(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（図5(a)参照）．ただし，この規則を使用するには2つの動作 V1，V2 を含む動作がなければならない．

(2) V2 の送信オブジェクトを，V1 の受信オブジェクトと同じにする（図5(b)）．

(3) V1 の受信オブジェクトから V2 の送信オブジェクトへ向かう新しい動作 V3 を追加する（図5(c)）．

(4) V1 と V2 の受信オブジェクトを同じになるようにする（図5(d)）．

(5) V2 の送信オブジェクト A2 から V1 の受信オブジェクト O1 へ向かう新しい動作 V3 を追加し，V2 の送信オブジェクトを新たに O1 とする（図5(e)）．

表5に各タイプの因果関係に対して，上記のどの操作を施せばよいかを示す．たとえば，タイプ i) の因果関係を記述するためには，図5(a)もしくは(c)のようにする．(c)のようにした場合，新しい動作 V3 を導入し，先行する動作 V1 を受けたオブジェクトが V3 を送信し，V3 を受信したオブジェクトが後に続く V2 を送信する形で記述される．

図2(g)は抽出された動作間関係を表したものである．press と visit 間の関係は cause の格構造（図2(c)）に表3の II-6 を適用して得られたもの，visit と depart および visit と reach の関係は，visit に

表5 因果関係に基づく送信・受信オブジェクトの決定／変更
Table 5 Decision of senders and receivers based on causal relations.

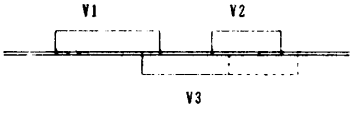
カテゴリ	変換規則（図5のどれを用いるか）
I	(a) ただし、2つの動作を包含する1つの動作がなければならない
	(c) V3の送信をPast型のメッセージ送信にする 
II	(b) V2の送信をPast型のメッセージ送信にする
	(c) V2もしくはV3の送信をPast型のメッセージ送信にする
III	(b)
	(c)
IV	(d)
	(e) V1とV3の禁止関係を記述する

表3の I-7 を適用して得られたものである．これら3つの動作間関係に表5の規則を適用し，図2(h)のモジュール構造を得る．これは，どのクラスモジュールにどの動作がメソッドとしてカプセル化されるかを表した図である．press と visit の関係はカテゴリ I であり，これに対して適用できる規則は，表5より(a)と(c)の2つがある．どちらを使用するかは設計者の判断による．この例では(c)を適用し，新しい動作 inform を図5(c)の動作 V3 として導入する方法を選択した．これは，2つの動作の動作主を同じオブジェクトにして，lift が button を press する，あるいは passenger が lift を visit すると解釈するよりも自然であると判断したためである．このように，送信オブジェクトを主語に，動作を動詞に，受信オブジェクトを目的語とする自然言語表現を作り，自然な表現であるかどうかを吟味するのも，規則の選択基準の1つになりうるであろう．

カテゴリ iv)の関係を記述するには，2つの動作が同じオブジェクトに施されていなければならない．2つの動作の定義を行っているクラスモジュール内で禁止条件が記述される．図5(e)は，新しい動作 V3 が V2 を発動する形になっており，V1 と V3 の禁止関係を定義することによって，V1 と V2 の関係が規定されたことになる．

5. 応用例と評価

本章では，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[6]で，非形式

仕様の原文を**図6**に示す.

5.1 名詞・動詞の抽出と分類

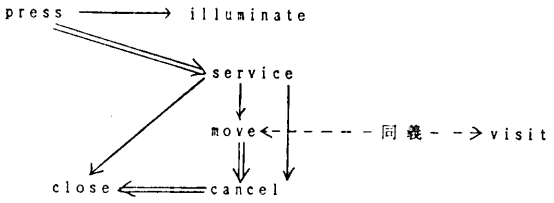
本手法を図6の lift の制御問題に適用し，形式仕様を構築する例について述べる．まず各文に含まれている動詞を，省略されている主語，目的語を補いながら，第2章であげたカテゴリに分類していく．たとえば，文 6) 中では，動作動詞 press に対する主語，目的語，補語のいずれも出現していないため，どのような語句が省略されているかを判別し，それらを補った後で分類を行う．この場合は，主語として passenger，目的語として button を補った．表1中のカテゴリにないパターンの動作動詞が出現している場合は，なんらかの語句が省略されている場合である．また，各文の本動詞だけでなく，節や句中に出現している動詞の中で，システムの構成要素となりう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よる部分がきわめて大きく，必ずしも結果が唯一になるとは限らない．

分類後に表3の規則に従って，動作動詞の動作主・動作対象を決定していく．図6には分類結果および動作主 (agent)・動作対象 (obj) も示されている．分類番号の1桁目の数字は，3.2 節の動詞の4つの分類カテゴリの番号を表す．したがって，1,2,4 で始まる動詞は，動作を表しているのではないと判断された動詞である．A_{sub}, A_{obj} は，主語，目的語として動作名詞が出現していることを表し，それに続く括弧内の sub, obj が各動作名詞の動作主，動作対象を表している．たとえば，14の文中に出現している動詞 service は動作動詞（分類番号3）の I-1 のカテゴリに分類される．これは，意味上の主語が control mechanism と考えられ，目的語は request である．名詞 request を値名詞と判定したのは，文 13)中で request が関係動詞 have (lift オブジェクトとの所有関係を表す) とともに使用されていることによる．すなわち，文 13)より，名詞 request はオブジェクト lift の属性を表していると判断した．I-1 の規則より，動作主は control mechanism，動作対象は lift で

```
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. I-6  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. I-1  agent, obj:button)  (3. I-5  agent:passenger  obj:button)
    and cause the lift to visit the corresponding floor. --- 6)
   (4. II-6  A_sub:press(agent:passenger obj:button)  A_obj:visit)
                          (3. I-7  agent:lift  obj:floor)
   The illumination is cancelled
         (3. I-3  agent:lift  obj:button)
     when the corresponding floor is visited by the lift. --- 7)
                       (3. I-7  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. II-1  agent:passenger  A_obj:up-lift(agent, obj:lift))
  (3. II-1  agent:passenger  A_obj:down-lift(agent, obj:lift))
  These buttons illuminate when pressed. --- 9)
      (3. I-1  agent, obj:button)  (3. I-5  agent:passenger  obj:button)
   The illumination is cancelled
      (3. I-3  agent:lift  obj:button)
    when a lift visits the floor
            (3. I-7  agent:lift  obj:floor)
    and is either moving in the desired direction,
             (3. I-1  agent, obj:lift)
         or has no outstanding requests. --- 10)
            (1)
   In the latter case, if both floor buttons are pressed,
                                  (3. I-5  agent:passenger  obj:button)
                     only one should be cancelled. --- 11)
                       (3. I-5  agent:lift  obj:button)
   The algorithm to decide which to service
   should minimize the waiting time for both requests. ---12)
              (3. II-1  agent:control mechanism  obj:request(obj:lift))
3- When a lift has no requests to service,
            (1)
   it should remain at its final destination with its door closed
            (2)                              (3. I-5  agent:lift  obj:door)
   and await further request (or model a 'holding floor'). --- 13)
      (2)
4- All requests for lifts from floors must be serviced eventually,
    with all floors given equal priority. --- 14)
       (3. I-1  agent:control mechanism  obj:lift)
5- All requests for floors within lifts must be serviced eventually,
       (3. I-1  agent:control mechanism  obj:lift)
    with floors being serviced sequentially in the direction of travel. --- 15)
       (3. I-7  agent:control mechanism obj:floor)
6- Each lift has an emergency button which, when pressed
           (1)                          (3. I-5  agent:passenger  obj:button)
 causes a warning signal to be sent to the site manager. --- (16)
 (4. II-6  A_sub:press(agent:passenger obj:button)  A_obj:send)
                          (3. I-5  agent:lift  obj:site manager)
   The lift is then deemed 'out of service'. --- (17)
                 (2)
   Each lift has a mechanism to cancel its 'out of service' status. --- (18)
      (3. I-1   agent:lift  obj:lift)
```

図 6 Lift 問題と動詞の分類

Fig. 6 Informal specification of lift control problem and classification of their verbs.



⇒は i)、→は iii)カテゴリの関係を表す。

図 7 抽出した動作間関係
Fig. 7 Extracted action relations.

あるとした．図6の 14), 15) の文より，request は floor からのもの (from_floor) と lift からのもの (within_lift) と2種類あると考えられる．なお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．

この過程で，図6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．button オブジェクトについても，その機能に応じて区別し，lift button, up button, down button, emergency button とした．

5.2 モジュール構造の設計

動詞の分類後，動作を表す動詞に対して，他の動詞と因果関係があるかどうかを調べ，もしあれば分類する．16)の文中のタイプ II-6 の動詞 cause は，press 動作と send 動作間のカテゴリ iii)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．動作間関係の抽出・分類作業も，人間でなければ行えない作業である．抽出した動作間関係の一部 (up button の press 動作について) を**図7**に示す．次に不要な名詞，動詞を消去していく．たとえば，request (passenger から lift への) と press, visit と move は同義であるため，各々の前者を削除する．この過程の結果得られた関連図を**図8**に示す．

残った動作動詞と動作関係について，表5の規則をもとに動作とそれに関与しているオブジェクトとの関係を再検討する．たとえば press 動作と service 動作は，カテゴリ i)の因果関係にあるため，図5(c)の方法を用いて各 button から control mechanism への新しい動作 inform を追加する．このとき inform は Past 型の送信が行われる．以上のような手順によっ

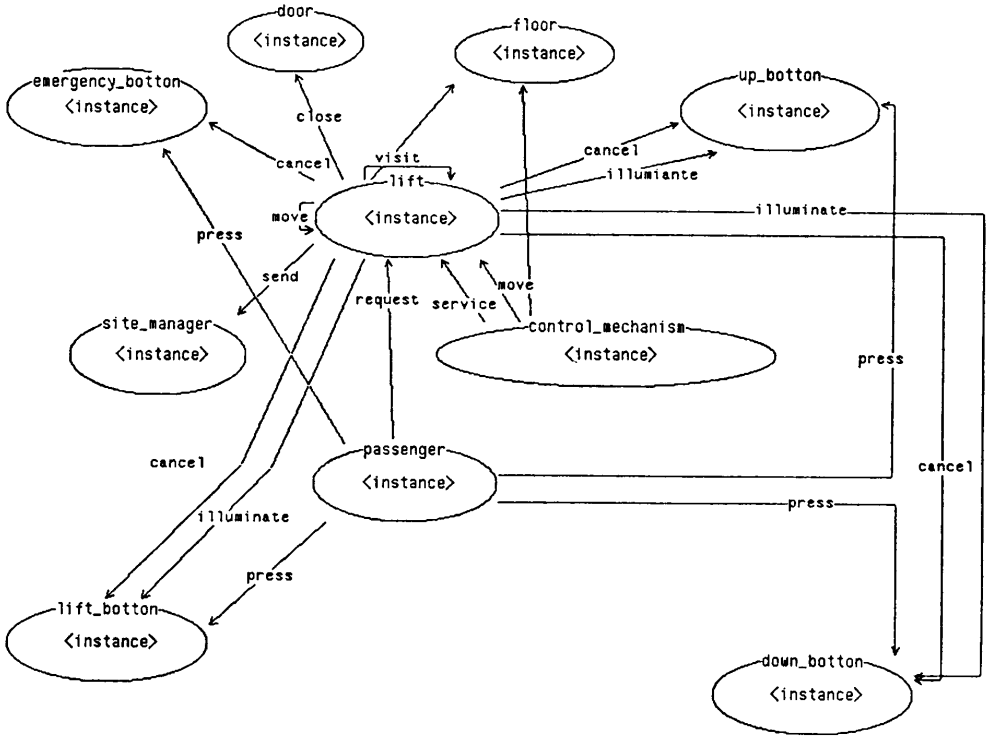


図 8 オブジェクトと動作の関連
Fig. 8 Relationship between objects and actions.

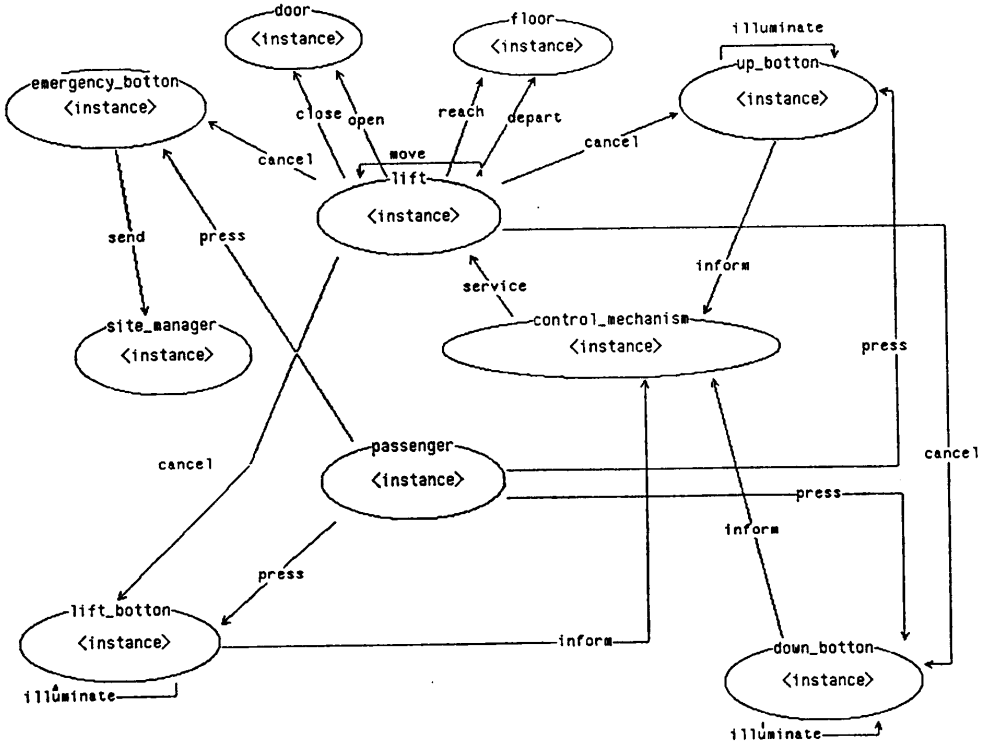


図 9 オブジェクトと動作の関連—最終版
Fig. 9 Relationship between objects and actions—Final version.

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 9** の関連図が得られる．**図 10** は，設計したモジュールの内容を非形式的に述べたものである．

図 10 では，lift, lift button, emergency button などがクラスモジュールである，lift は location, direction, status, requests を属性として持っている．':' 以下が各属性値の型を表している．lift に対する動作，つまり lift のメソッドは，service, move があり，その内容は動作前，動作中，動作後のオブジェクトの状態つまりその属性値がどうなっているかと，その動作中に発動する動作が書かれている．

図 6 の 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応じて，つまり lift から見て近い階から順に service されていくことを述べている．このような 1 つのオブジェクトに作用する複数の動作の関係（排他制御や優先順位関係）の要求は，そのオブジェクトのクラスモジュール内に書かれなければならない．図 10 の自然言語文で記述された内容を形式化すれば，オブジェクトモデルに基づいた形式的仕様が得られる．得られた形式仕様，および形式的記述言語については本論文の趣旨からはずれるので，紙面の都合上割愛する．

6. 議　　論

6.1 自動化の可能性

本手法でのすべての処理をコンピュータで自動的に行うのは，現在の自然言語処理技術から考えても不可能である．コンピュータ処理できるように，自然言語の構文や意味を制限し，完全な仕様書（記述もれなく，矛盾もない）のみを対象とすることも考えられるが，現実のソフトウェア開発現場において，制限自然言語で完全な要求仕様書を書くのは非常に難しい作業になる．したがって，制限のない自然言語で記述された仕様を入力として，設計者と対話形式で作業を行っていく支援ツールが現実的である．その際，名詞・動詞の候補の抽出，第 4 章で述べた規則の適用といった作業は，自動化が行えると思われる．

6.2 オブジェクト指向設計法[8]との比較

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，Booch のオブジェクト指向設計法と異なり，オブジェクトやクラスを表す名詞ではなく，動詞に注目してその文型を細分類する

```
Control mechanism

  A button informs a control mechanism of a request
    行う動作
      1) service the request with a lift.
           --- Past 型のメッセージ送信
end  control mechanism ;

Lift
  属性 location : floor
       direction : 'up', 'down', or 'stopped'
       status : 'on service' or 'out of service' ---- 18)
       requests : set of request    ---- 10),13)のhave 動詞

  動作間の相互作用 (優先順位など)
   [service 動作同士の順序]
     1) request がともにwithin_lifts からきたものであるときは,
        request のdestination がlift に対してより近いほうの
        service 動作が先に終了する.          ---- 15)

  A control mechanism services a request with a lift
     動作前
        1)lift のstatus が'on service'である.
     動作中
        1)lift のrequests に, 対応するrequest が含まれている.
     動作後
        1)そのrequest はlift のrequests から無くなる.
        2)そのrequest がfrom_floor からきたものであれば,
             lift のdirection とそのrequest のdirection は
             同じ値となっている.
        3)lift のrequests が空になっていれば,
             lift のdirection はstopped になっている.
             --- 13)のremain at its final destination の部分

     行う動作
       1) move the lift to the destination of the request.
       2) lift のlocation 、direction が
            ともにそのrequest を満たしたなら、
          => open the door.
       3) そのdoor がopen したなら、
          => close the door.
          => cancel the button.  ---- 7)、10)の前半部
  end  service ;

  A lift move a lift to a floor
     動作中
      1)そのfloor(目的階)が動作前にいたlocation よりも
        下の階であるならば,
          lift がその間(動作前の階と目的階の間)にいるときは,
                そのlift のdirection はdown になっている.
      2)そのfloor が動作前にいたlocation よりも上の階であるならば,
          lift がその間にいるときは,
                そのlift のdirection はup になっている.
     動作後
       1) lift のlocation がそのfloor である.
     行う動作
       1) depart from 動作前にいたlocation.
       2) reach the floor(目的階).
  end  move ;

end  lift ;
```

```
Up button
   属性 status : 'on' or 'off'
        illumination : 'on' or 'off'.
        floor : floor

   A passenger press an up button
     動作前
       1) up button のstatus が'off'である.
     動作後
       1) up button のstatus が'on'になる.
     行う動作
       1) up button の動作が'on'になったら、
          => inform a controller of a request
          => illuminate the up button
   end  press ;

   A lift cancel an up button
     動作前
       1) up button のstatus が'on'である.
     動作後
       1) up button のillumination が'off'になる.
       2) up button のstatus が'off'になる.
   end  cancel ;
   ...

end  up button ;

Request
    within_lift からくるもの  or
    from_floor からくるもの

          within_lift からくるもの
               属性 destination : floor
          from_floor からくるもの
               属性 destination : floor
                    direction : 'up' or 'down'
end  request ;

Floor
  属性  lifts : set of lift
        number : integer.

  A lift reaches a floor
    動作後
      1) そのlift はfloor のlifts に追加された.
  end  reach ;

  A lift departs from a floor
   ...
  end  depart ;
  ...
end  floor ;

end lift system ;
```

図 10 モジュール内容の非形式的記述 (一部)
Fig. 10 Informal specification of contents of modules.

ことにより, モジュール構造を設計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) 自然言語文の意味の中で, 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, 設計者がより正確な自然言語理解を得ることができる.

という利点が考えられる. しかし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

しきれるかどうかの問題も残されている．

スーパクラス・サブクラスといったクラスモジュールの階層構造の抽出は，Booch の手法と同様に，ここでは行わなかった．というのは，一般に人が新しく仕様を書くとき，まずクラス間の階層性に注目するのではなく，オブジェクトとメソッドを抽出し，その後に階層性を考えるという手順を踏むからである．その意味で，階層構造の抽出は，本手法の後段に位置する作業と考えられる．

6.3 自然言語仕様の質

最初に与えられる自然言語仕様中で，不可欠な事項が未記述のままであったり，記述が矛盾していたりする場合がある．本手法は自然言語文中に出現している語彙をソフトウェアの要素に対応づけるやり方をとっているため，モジュール構造の構築作業を進めるには，対応した語彙の出現が必要である．したがって，文中に出現している記述のみを反映した構造しか得られない．本手法を適用し設計を進めていく段階で，各ステップで適用する規則がない，適用しても不自然になったなどの理由により，自然言語記述の不備を設計者が検出することもある程度は可能である．設計者が気づかなかった記述の論理的な不備を検出するのは，図9，図 10 のような実際に得られたモジュール設計仕様書よりプロトタイププログラムを作成し，検査していくのよりも効果的な手法はない．検出された不備は，自然言語記述の上で修正を行い，その上で修正の影響が及ぶ箇所の設計を本手法の手順に従ってやり直す．

自然言語による非形式仕様では，読者が常識的に理解できるような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，このような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みなすことができる．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，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．したがって，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り，「記述が省略されている」という短所を逆に自然な階層構造の構築に生かせることになる．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．

7. おわりに

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文献 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．その意味で最上位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分を埋め込むためのスロットを作りだしていると見ることもできる．

本論文では，動詞の扱いを中心的に述べてきたが，他の語，たとえば，名詞，形容詞，副詞などのいっそうの分析が必要である．また，動詞の分類はオブジェクトモデルをベースに行ったが，モデル化の方法（たとえば，データフロー法など）が異なれば，分類方法や注目すべき品詞も異なってくるであろう．抽出する動作間関係においても，スケジューリングや割り込みなどがあるシステムについてはより複雑な分析が必要であろう．今後は，運用経験を積んで，より洗練された統合的なガイドシステムの構築が必要である．

参考文献

1) 有澤 誠：ソフトウェアプロトタイピング，近代科学社 (1986).
2) Guttag, J. V., Horowitz, E., and Musser, D. R.: Abstract Data Types and Software Validation, *CACM*, Vol. 21, No. 12, pp. 1048-1064 (1978).
3) Abbot, R.: Program Design by Informal English Descriptions, *CACM*, Vol. 26, No. 11, pp. 882-894 (1983).
4) Balzer, R., Goldman, N. and Wile, D.: Informality in Program Specification, *IEEE Trans. Softw. Eng.*, Vol. 4, No. 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, *Proc. of 4th International Workshop on Software Specification and Design*, 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. Softw. Eng.*, Vol. 12, No. 2, pp., 211-221 (1986).
9) Østerbye, K.: Active Objects: An Access Oriented Framework for Object-Oriented Languages, *Journal of Object Oriented Programming*, June/July, pp. 6-10 (1988).
10) 辻井潤一, 上原邦昭: ソフトウェア工学と自然言語処理, 情報処理, Vol. 28, No. 7, pp. 913-921 (1987).
11) Potts, C. and Finkelstein, A.: Structured Common Sense: A Requirements Elicitation and Formalization Method for Modal Action Logic, FOREST Deliverable Report 2 (1986).
12) Goldberg, A. and Robson, D.: *Smalltalk-80: The Language and Its Implementation*, Addison-Wesley (1983).
13) 鈴木則久編: オブジェクト指向, 共立出版 (1985).
14) Yourdon, E. and Constantine, L.: *Structured Design: Fundamentals of a Discipline of Computer Program and System Design*, Yourdon Press (1977).
15) Jackson, M.: *Principles of Program Design*, Academic Press (1975).

(昭和63年10月24日受付)
(平成元年9月12日採録)

佐伯 元司（正会員）

昭和31年生. 昭和53年東京工業大学工学部電気電子工学科卒業. 昭和58年同大学院情報工学専攻博士課程修了. 工学博士. 同年より東京工業大学工学部情報工学科助手. 昭和63年同大学工学部電気電子工学科助教授. ソフトウェア工学, マンマシン・システムなどの研究に従事.

蓬莱 尚幸（正会員）

昭和36年生. 昭和59年東京工業大学工学部情報工学科卒業. 昭和61年同大学院情報工学専攻修士課程修了. 同年富士通(株)入社. 現在, 国際情報社会科学研究所に勤務. 仕様記述言語と方法論, プロトタイピングなどの研究に従事.

榎本 肇（正会員）

昭和23年東京工業大学電気卒業. 郵政省電波管理局, 国際電電(株)を経て, 昭和42年東京工業大学教授. 現在, 同大名誉教授, 芝浦工業大学教授, 富士通(株)顧問. 電波のフェージング, 雑音理論, 計算機要素, 計算機システムとその応用, 画像処理の研究に従事. 最近はサービス工学を提案し, マン・マシンインタフェースとしての言語の研究に興味をもっている. 本会論文賞2回, 電子情報通信学会論文賞2回, 同会業績賞1回, テレビ学会論文賞2回各受賞.