

自然語によるプログラム仕様に対する 理解モデルの提案

大野 浩史(*) 菊野 亨(**) 鳥居 宏次(**)

(*)日本ユニシス(株), (**)大阪大学基礎工学部

本報告では、自然語(日本語)で書かれたプログラム仕様に対する理解モデルの提案を行い、具体的な問題への適用例を示す。提案するモデルでは、仕様記述中の操作“対象”に関する記述部分に注目し、その記述内容をラベル付き有向グラフとして表現する。

まず、プログラム仕様から“対象”に関する記述を単文として抽出する。次に、対象の間の2つの関係(kind-of, part-of)をKグラフとPグラフに表現する。最後に、これらのグラフを統合してFグラフを構成する。Fグラフは基本的には意味ネットワークと見なすことができるが、節点を2つの型(対象と操作)に明確に区分し、有向枝のラベルに固有の意味を割り当てている。

提案する理解モデルではグラフ表現を採用しているため、理解内容の追加、変更が容易である。更に、プログラム設計者の間でプログラム仕様に対する共通の認識を持ち易い。

A Graph Model to Represent Semantics of Program Specifications in Natural Language

Koji Ohno(*), Tohru Kikuno(**) and Koji Torii(**)

(*)Nihon Unisys Limited, (**)Faculty of Engineering Science, Osaka University

(*)Akasaka 2-17-51, Minatoku, Tokyo 107, Japan

(**)Machikaneyama 1-1, Toyonaka, Osaka 560, Japan

Abstract This paper proposes a graph model to represent semantics of a program specification written in Japanese. The graph model is constructed using a descriptive information concerning "object" in the program specification.

An outline of graph construction is summarized as follows : (1) extract simple statements, that describe objects, from a given program specification, (2) construct K-graph and P-graph based on two relationships (kind-of and post-of) between objects, and (3) construct F-graph by merging K-graph and P-graph. The F-graph is a so-called semantic network. The difference is that nodes in F-graph are classified into two categories(object-type and operation-type) and that arcs are attached labels denoting specific roles.

The graph model proposed in this paper will make it easy to add and modify information concerning object. Additionally, by using the model, it will be relatively easy for program designers to have common recognition of program specification.

1. まえがき

開発されたプログラムの正しさを議論する基準として、厳密に記述されたプログラム仕様に関心が集まっている^{[2][3][4]}。例えば、代数的記述言語を用いてプログラム仕様を形式的に記述する方法がある^{[2][6]}。実際に、形式的にプログラム仕様を与え、それを段階的に詳細化していった、プログラム生成を行なった例も報告されている^[2]。

しかし、開発すべきシステムが大規模化した時に、形式的な記述を（矛盾なく）与えることは容易でない。仮りに、それが与えられたとしても、その記述を設計者が（直観的なレベルで）理解するには多くの労力と時間を要する。

これに対し、開発現場での使いやすさを考慮して自然語で記述したプログラム仕様が広く利用されている。自然語による記述には次の(1)~(3)に示す問題点があることが指摘されている。

- (1) 意味のあいまいな記述が含まれる。
- (2) ある知識を前提として、記述の省略が行われる。
- (3) 実現に関する記述が含まれる。

同じ仕様書が与えられても、システム分析者や設計者毎に自然語の意図している内容の理解が異なり、複数の異なるプログラムが開発される可能性がある。その中の幾つかは、間違ったプログラムである危険性もある。従って、自然語による意図の正しい理解がプログラム開発の前提として重要となる。

本報告では、自然語（日本語）によるプログラム仕様に対する理解モデルの提案を行う。提案するモデルでは、操作“対象”に関する仕様記述部分に注目し、その記述内容をグラフとして表現する。

具体的には、仕様から“対象”に関する記述を単文として抽出し、“対象”の間の2つの関係（kind-of, part-of）を木グラフ（KグラフとPグラフ）に表現する。最終的に、これらの木グラフを統合してFグラフ（機能化モデル）を構成する。このFグラフが与えられた仕様の意味を表現する理解モデルになっている。

自然言語理解モデルの基礎となる意味構造の形式についてはこれまでに多くの提案がなされてきている^[5]。代表的なものに、意味ネットワーク形式、格フレーム形式、論理形式、等がある。本報告で提案するKグラフ、Pグラフ、Fグラフは基本的には意味ネットワーク形式である。ネットワークの節点を対象と操作に明確に区分していること、有向枝のラベルに固有の意味を割り当てていることが特徴である。

提案する理解モデルではグラフ表現を採用したため、直観的にかつ形式的な表原が可能となり、次の(1)、(2)の利点が生じる。

- (1) 理解内容の追加，変更が容易である。
- (2) 設計者の間で共通の認識が得られ易い。

2. 基本方針

プログラム仕様（自然語）の理解についての本報告における基本的な考えを図1に示す。仕様書を構成する記述内容は“機能”に関する記述と“対象”に関する記述から構成されている。ここではその内の対象に関する記述（具体的には、名詞、動名詞を指す）に注目し、その記述内容を表現するためのモデルを導入する。モデルは、“ある対象が他の対象を抽象化したものであるという関係（kind-ofとpart-of）”に基づいているため、抽象化モデルと呼ぶ。

更に、対象に注目して記述した抽象化モデル同士を統合するための機能化モデルも導入する。この機能化モデルを構成する過程で、与えられた仕様書の下で実現すべき機能が全て、隅に、列挙される。

従って、図1の点線部分で示すように、2つの記述の照合を行うことにより、仕様書の理解の妥当性の検討も可能となる。

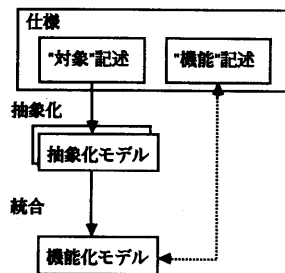


図1 基本方針

3. 意味表現

ここでは、仕様書を構成する文章を単文に分割し、各単文の意味を表現する。次に、それらを統合して仕様書の意味を表現する。

単文に変換した記述は、図2に示すように、機能に関する記述と対象に関する記述に分けられる。更に、機能の記述中には操作と（操作対象を表す）属性名に関する記述が含まれ、対象の記述中には属性名と属性値に関する記述が含まれるものとする。操作は変換操作と分類操作から構成される。各操作に対応する他動詞の具体的な例を表1にまとめる。

単文

- 機能の記述 = (属性名+操作) の記述
- 対象の記述 = (属性名+属性値) の記述

操作

- 変換操作
- 分類操作

図2 単文の構成

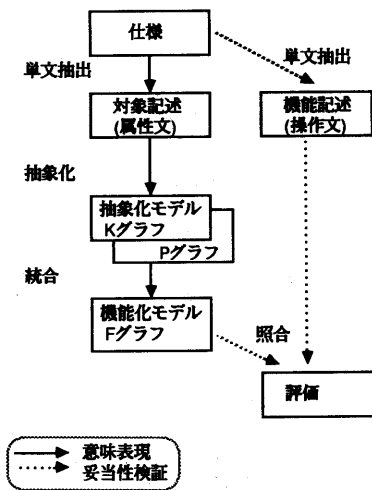


図3 意味表現の枠組み

意味表現を抽象化の視点から行う。図3に示すように抽象化モデルと機能化モデルの2つを導入する。実線部分に意味表現の流れを示す。求まる意味表現に対する機能から見た妥当性検証の流れを図3の点線部分に示す。

3.1 抽象化モデル

便宜上、対象の記述(従って、属性名、及び、属性値の記述)をしている単文を属性文と呼ぶ。一方、操作の記述をしている単文を操作文と呼ぶ。

対象記述の単文を表2に示す6種類の型に分類する(各型の詳細な説明は、4.1で述べる)。

表1 他動詞の分類

操作	動詞の型	例
変換操作	物理的な動作を指示する動詞	移動する, 置く, 変換する, 交換する, 作成する
	情報処理動作を指示する動詞	数える, 計算する, 加える, 引く, 書く, 記述する, 検索する, 比較する, 読む入力する, 出力する
分類操作	知覚動詞	知る, 把握する, 測る, 聞く, 調査する, 見る
	意思的動詞	決める, 予想する, 判断する, 仮定する, 推測する, 開始する終了する, 指示する

表2 対象記述の分類

番号	型名	形式	備考
1	attribute文	Aは性質Bをもつ	属性値種類 (親子関係) 部分全体関係 実現値
2	value文	性質Aは値Bをとる	
3	kind-of文	B, CはAの一つの種類である	
4	part-of文	AはB, Cからなる	
5	element文	AはBを要素にもつ	
6	equal文	AとBは同じである	

次に、これら6種類の単文の意味を表現する抽象化モデルとして2つのグラフ(木)KとPを導入する。

Kグラフの形式を図4に示す。図4からも分かるように、Kグラフはkind-of文, attribute文, equal文, value文に基づいて構成されるラベル付き木である。kind-of文「B, CはAの一つの種類である」に対して、Aを親とし、B, Cを子とする木が対応する。attribute文「Aは性質Bをもつ」に対して、Aを親としBを子とする木(直線グラフ)が対応する。equal文「AはBと同じである」, value文「性質Aは値Bをとる」に対しても、Aを親とし、Bを子とする木が対応する。

Pグラフは、図5に示すように、part-of文, element文, equal文に基づいて構成されるラベル付き木である。

3. 2機能化モデル

仕様に含まれる機能に関する意味を表現するため、図6に示す操作図を導入する。

図6(a)は、kind-of文に対応しており、入力された対象を(属性値に基づいて)幾つかのクラスに分類することを表す。一方、図6(b)は、part-of文に対応しており、入力された幾つかの対象に変換を行って1つの出力対象を求めることを表す。各操作に対応する他動詞の具体的な例は、既に、表1に示している。

図6の操作図は一つの単文に対するものである。そこで、単文の集合に対してその機能的意味を表現するため

に、図6(a)の出力対象名と図6(b)の入力対象名の同じものを重ね合わせる。結合結果の概念図を図7に示す。結合図の各構成要素には識別のための名前をつける。

この結合図を用いて機能に関する意味表現を行うための機能化モデルとして図8に示すFグラフを導入する。すなわち、接続リンク、及び、同じ対象名を利用して結合図同士を接続することにより意味表現を行う。

Fグラフの特徴を幾つかまとめておく。節点○に対しては入力枝が丁度1本、出力枝は高々1本である。節点□に対しては入力枝、出力枝ともに少なくとも一本である。(Fグラフの厳密な定義を与えること、一般的性質を明示することは今後の重要な課題である。)

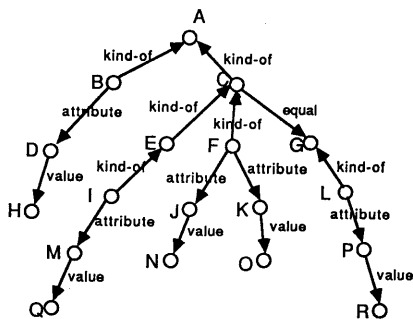


図4 Kグラフの例

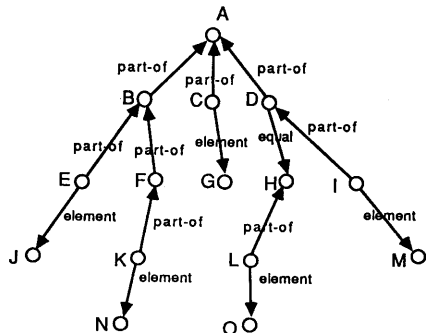


図5 Pグラフの例

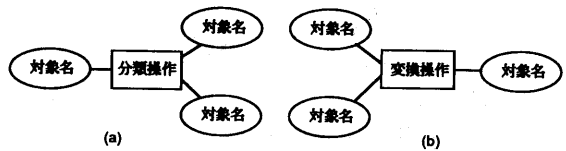


図6 操作図

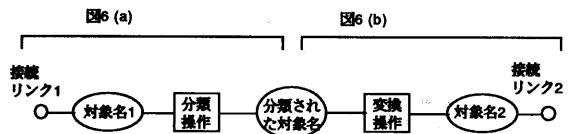


図7 結合図

リンク1	対象名1	分類操作	クラス名	変換操作	対象名2	リンク2
①	A	f1	E F	f4 f5	B C	② ③
②	B	f2	H	f6	D	④
③	C			f7	K	
④	D	f3	I J	f8 f9	A L	①

図8 Fグラフの例

4. 構成手順

抽象化に基づく意味表現の構成手順の概要を図9に示す。以下に、順次、各ステップの詳細について説明する。

4.1 Step 1 (単文への分解)

仕様書を構成する文中に含まれる動詞、名詞、形容詞に注目して、単文へと分解していく。更に、後の処理の都合上、各単文に意味上の補完を行う。

(A)操作文と属性文の抽出…まず、操作文について説明する。表1で示した他動詞に注目し、次の表3に示す規則1～規則6を適用して、「～を～する(=対象+操作)」の形式の単文を構成する。

次に、属性文について説明する。修飾語(形容詞)、及び、対象を指す語(名詞、動名詞)に注目し、表2に示した6種類のいずれかの形式の単文を構成する。ここで、6種類の形式の特徴を述べる。kind-of文が種類を表すのに対し、part-of文は構成を表す。例えば、語と課金される語の関係を記述した文はkind-of文と見なされ、電報と(電報を構成する)文字系列の関係を記述した文はpart-ofの文と見なされる。element文はpart-of文と一緒に用いられ、最小の(分割不可能な)要素を表す。その要素数は有限に限っている。一方、value文はkind-of文と一緒に用いられ、具体的な値を示す。

(B)操作文の補完…後のstep5での評価を容易にするために、表4に示す規則7～規則11を操作文に適用して、意味的な補完を行う。

4.2 Step 2 (Kグラフ, Pグラフの構成)

step1で求めた属性文に基づいた意味表現を求める。このステップでは、その準備として2つのグラフ(Kグラフ, Pグラフ)を構成する。

(A)属性グラフの構成…表2に示す6種類の型の属性文ごとに木グラフを構成する。

例えば、kind-of文「B, CはAの1つの種類である」に対して、Aを根とし、BとCを葉とする木を作る。B, CからAに向かう2本の有向枝を引き、ラベル kind-of を各有向枝につける。

attribute文とvalue文の合成と見なされる「Aの性質Bは値Cをとる」に対しては、Aを根とし、Aの子がBであり、Bの子がCである木を作る。CからBに向かうラベルvalueの有向枝と、BからAに向かうラベルattributeの有向枝をひく。

他のpart-of文, element文, equal文についても同様にして木を構成する(詳細は5.の適用例を参照)。こうして構成された木を属性グラフと呼ぶ。

Step 1 (単文への分解)

- 1.1 操作文と属性文の抽出
- 1.2 操作文の補完

Step 2 (Kグラフ, Pグラフの構成)

- 2.1 属性グラフの構成
- 2.2 Kグラフ, Pグラフの構成

Step 3 (機能の推定)

- 3.1 Kグラフに基づく操作図の構成
- 3.2 Pグラフに基づく操作図の構成

Step 4 (Fグラフの構成)

- 4.1 結合図の構成
- 4.2 意味的な補完

Step 5 (意味表現の評価)

- 5.1 機能推定の評価
- 5.2 再実行(必要ならば)

図9 意味表現手順の概要

表3 操作文の構成規則

規則	内容
1	受動態を能動態に変換する。
2	「～できる」は平常文に変換する。
3	否定文は肯定文に変換する。
4	「x AND y」, 「x OR y」を含む文はxとyの2つの文に分ける。
5	「AについてBをCする」は「AのBをCする」に変換する。
6	「DについてはEする」は「DをEする」に変換する。

(B)Kグラフ, Pグラフの構成…(A)で構成した属性グラフを結合して、図4, 図5で示したKグラフ, Pグラフを構成する。

まず、kind-of文, attribute文とvalue文に対応する属性グラフの集合に対し、ラベルが同じ節点同士を重ね合わせるにより、Kグラフを構成する。なお、equal文については、同じ(equalの関係にある)ラベルの節点がKグラフ中に含まれていれば、対応する属性グラフをKグラフに追加する。

次に、part-of文とelement文に対応する属性グラフの集合に対し、ラベルが同じ節点同士を重ね合わせることにより、Pグラフを構成する。必要ならば、equal文に対応する属性グラフもPグラフに追加する。

4.3 Step 3 (機能の推定)

Kグラフ、Pグラフに基づいて操作図を構成する。但し、操作図(図6参照)中の変換操作、及び、分類操作には識別のための記号を割り付ける。これらの記号を用いて、推定された(仮想的な)機能を表すことにする。

(A)Kグラフに基づく操作図の構成…Kグラフ中でラベルがkind-ofの有向枝が節点Aから節点Bに向かっていたとする。このとき、Bを入力対象名とし、Aを出力対象名とする変換操作図を作る。この場合の仮想的な変換操作には、直観的に、「見分ける」、「知る」、「判定する」等の機能が対応する。

一方、ラベルがvalueの有向枝が節点Aから節点Bに向かっていた場合、変換操作図を作る。但し、Aを入力対象名とする操作図とAを出力対象とする操作図の2つを作る(なお、Bは操作図中で用いない)。仮想的な変換操作には「入力する」、「出力する」、「記録する」等が対応する。

(B)Pグラフに基づく操作図…Pグラフ中でラベルがpart-ofの有向枝が節点Aから節点Bに向かっていたとする。このとき、Aを入力対象名とし、Bを出力対象名とする分類操作図を作る。この場合の(仮想的な)分類操作には「組み立てる」、「分解する」等が対応する。

4.4 Step 4 (Fグラフの構成)

Step 3で求めた操作図を利用してFグラフ(図8参照)を構成する。

(A)結合図の構成…図7中に示す(特に、図6(a),(b)との)対応に従って結合図を作る。

(B)Fグラフの構成…(A)で求めた結合図同士を、接続リンク、及び、有向枝で接続することにより、Fグラフを求める。なお、接続リンクについては、対象名1と対象名2が等しい時、かつその時に限り同じ番号を割当てる。有向枝による接続は、対応する操作図が存在する時に限って行う。

最後に、目的とする出力(対象名)に到る道が存在する節点以外の全ての節点、及び、それに接続する全ての有向枝をFグラフから削除する。

表4 操作文の補完規則

規則	内容
7	抽象的な操作をいくつかの具体的な操作に分ける。
8	(計算機上での意味が)不明な操作を具体的な操作に置き換える。
9	計算機上での操作と見なせないものを削除する。
10	操作の対象が欠落していれば補う。
11	操作の対象が明確でなければ、属性文を参考にして明示する。

4.5 Step 5 (意味表現の評価)

Fグラフ中の仮想的な操作とStep 1で求めた操作文とを比較することにより、求めた意味表現(Fグラフ)の妥当性の検討を行う。

(A)機能推定の評価…Fグラフ中に残っている(仮想的な)変換操作、及び、分類操作に対し、機能的に等価な他動詞を含む操作文がStep 1で求まっているか否かを調べる。Step 1で求まっていない場合には、元の仕様が不完全であるか、あるいは、Step 1~Step 4の実行が不適切であると判断する。

(B)再実行…必要ならば、Step 1から再実行する。

5. 適用例

電報解析問題^{[1][6][7]}に対するプログラム仕様に対し、

4. で述べた構成手順を適用した結果を以下に示す。文献[7]に記述されているプログラム仕様を図10に示す。なお、図10中の文番号1~7は後の説明のために用いる。

Step 1で求まる操作文を図11に、属性文を図12に示す。図11、図12中の文番号i-xは、その単文が図10の文iから作られたものであることを表す。(規則i,j)等はその単文を作るのに規則iと規則jが使われたことを表す。

Step 2.1で求まる属性グラフの一部を図13に示す。図13には対応する属性文の文番号も記述している。Step 2.2で求まるKグラフの一部を図14に、Pグラフの一部を図15に示す。

Step 3.1で求まる操作図を図16に示す。同様に、Step 3.2で求まる操作図を図17に示す。Step 4で求まるFグラフを図18に示す。最後に、Step 5.1で行った比較照合結果を表5に示す。

- (1) 電報の流れを処理するプログラムを求めたい。
- (2) この流れはある装置上の文字、数字、空白記号の記号の系列として表され、予め定められた大きさの区分でバッファ領域に転送できる。(3) 電報内の各語は空白記号の系列で区切られ、各電報は'ZZZZ'という語で区切られる。
- (4) 電報の流れは空電報(語が存在しない電報)が現れると終了する。(5) 各電報は課金される語数と長すぎる語の有無が調べられる。(6) 'ZZZZ'及び'STOP'の語は課金されず、13字以上の語は長すぎるとみなされる。(7) これらの処理の結果は各電報に課金される語数と長すぎる語の発生を示すメッセージとが添えられて、みやすい形のリスティングとして表示される。

図10 電報解析問題

- 1-1 電報の流れを転送、解析する(規則 7)
- 1-2 プログラムを作る(規則 8)
- 2-1 省略(規則 1,9)
- 2-2 電報の流れを予め定められた大きさの区分でバッファ領域に転送する(規則 2,10)
- 3-1 電報内の各語を空白記号の系列で区切る(規則 1)
- 3-2 各電報を'ZZZZ'という語で区切る(規則 1)
- 4-1 電報の流れを空電報で終了する(規則 6)
- 5-1 各電報の課金される語数を記録する(規則 1,4,5,6,8)
- 5-2 各電報の長過ぎる語の有無を記録する(規則 1,4,5,6,8)
- 6-1 'ZZZZ', 'STOP'以外の語を課金べき語と判定する(規則 1,3,6,8)
- 6-2 13字以上の語を長過ぎる語であると判定する(規則 1,3,5)
- 7-1 各電報を見やすい形のリスティングとして表示する(規則1,4,11)
- 7-1 各電報の課金される語数を見やすい形のリスティングとして表示する(規則 1,4,11)
- 7-1 各電報の長過ぎる語の発生を示すメッセージを見やすい形のリスティングとして表示する(規則 1,4,11)
- 7-2 省略(規則 9)
- 7-3 省略(規則 9)

図11 操作文(Step1)

- 2-a 電報の流れとは記号の系列のことである(規則 6)
- 2-b 記号の系列は記号からなる(規則 4)
- 2-c 文字、数字、空白記号は記号の一種である(規則 3)
- 2-d バッファ上の記号系列は記号系列の一部である(規則 4)
- 2-e バッファ上の記号系列には、予め定められた区分の大きさがあある(規則 1)
- 3-a 各語は文字、数字からなる(規則 4)
- 3-b 各電報は'ZZZZ'以外の語と空白系列からなる(規則 4)
- 3-c 電報の流れは各電報と'ZZZZ'とからなる(規則 4,5)
- 3-d 'ZZZZ'は語の一種である(規則 3)
- 4-a 空電報は電報の一種である(規則 3)
- 4-b 空電報の語数は0である(規則 2)
- 5-a 各電報は「課金される語」と「課金されない語」を部分に持つ(規則 4)
- 5-b 各電報は「長過ぎる語」と「長過ぎない語」を部分に持つ(規則 4)
- 5-c 「課金される語」は語の一種である(規則 3)
- 5-d 「課金されない語」は語の一種である(規則 3)
- 5-e 「長過ぎる語」は語の一種である(規則 3)
- 5-f 「長過ぎない語」は語の一種である(規則 3)
- 5-g 語には「語の長さ」という尺度がある(規則 1)
- 5-h 各電報には「長過ぎる語」の有無という尺度がある(規則 1)
- 6-a 'ZZZZ'は「課金されない語」である(規則 2)
- 6-b 'STOP'は「課金されない語」である(規則 2)
- 6-c 「長過ぎる語」の最小語長は13である(規則 3)
- 6-d 「長過ぎない語」の最大語長は12である(規則 3)
- 7-a 各電報には「課金される語」の数という尺度がある(規則 1)
- 7-b 「処理の結果」は「各電報」と「課金される語数」と「長過ぎる語の発生を示すメッセージ」からなる(規則 4)
- 7-c 「処理の結果」とは「リスティング」のことである(規則 6)
- 7-d 「リスティング」にはみやすさという尺度がある(規則 1)

図12 属性文(Step2)

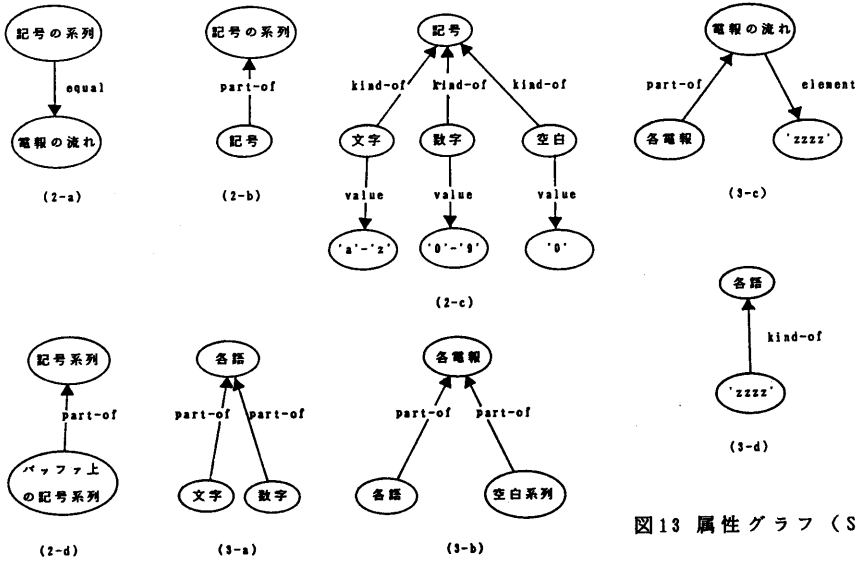


図13 属性グラフ (Step2.1)

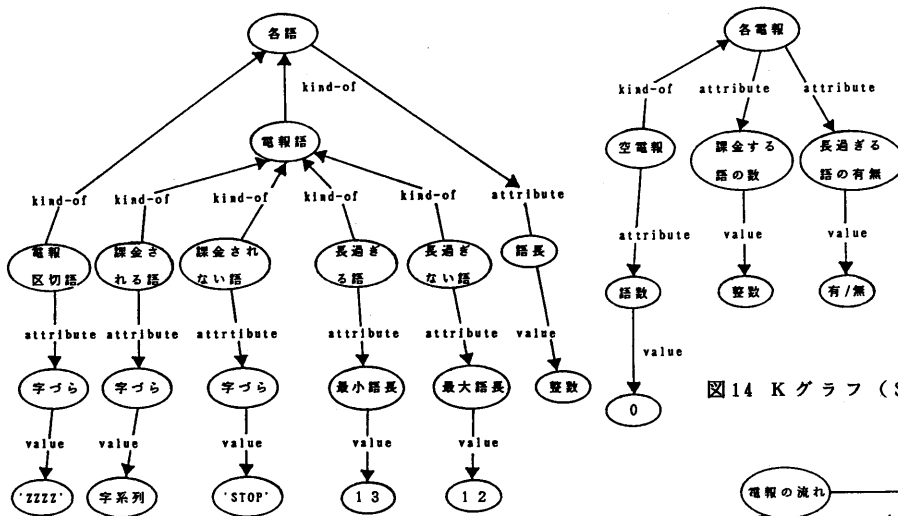
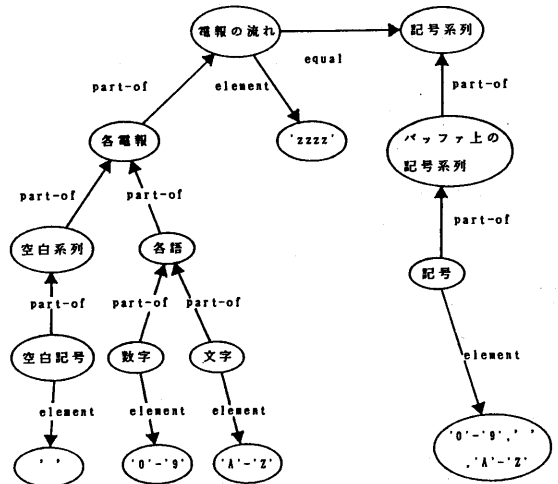


図14 Kグラフ (Step2.2)

図15 Pグラフ (Step2.2)



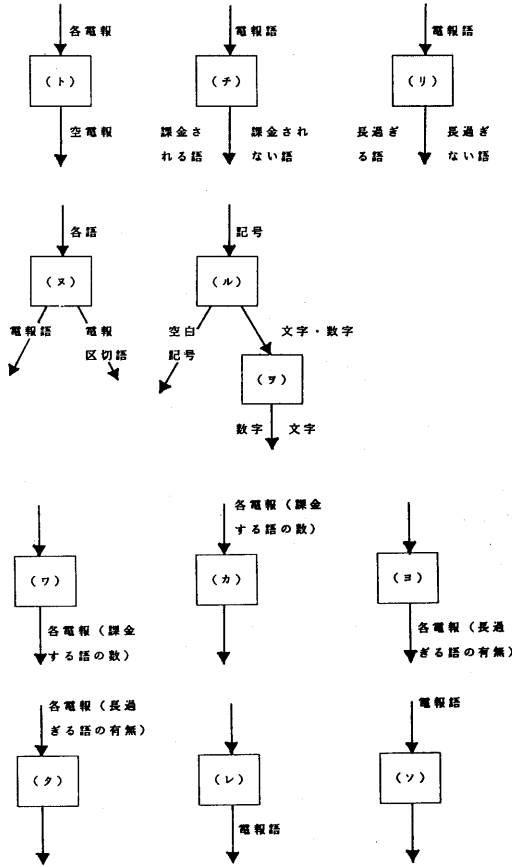


図16 操作図 (Step3.1)

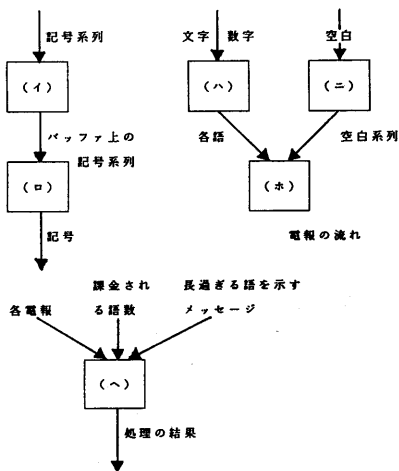


図17 操作図 (Step3.2)

6. むすび

本報告では、自然語（日本語）によるプログラム仕様に対する理解モデルの提案を行った。今回の提案は枠組みの説明に重点を置いており、用語の厳密な定義を与えるまでには至っていない。従って、理解した結果（グラフ）を設計段階にどのように利用するかについても触れていない。（これについては別の機会に発表する予定である）。

今後の主な課題を次にまとめる。

- ① 用語の厳密な定義を（具体的な例示も含めて）与えること。
- ② 表1、表2の分類を整備し、表3、表4の規則の充実を図ること。
- ③ Step1、4の手順の自動化、及び、Step5の内容の詳細化について検討すること。

謝辞：本報告の作成に当たり、ご協力頂いた大阪大学基礎工学部大学院生 飯田元君、楠本真二君、河野俊二君に感謝致します。

文献

- [1]Henderson,P. and Snowdon,R.A.: "An experiment in structured programming", BIT, Vol.12, No1, pp.38-53(1972).
- [2]東野, 関, 谷口: "代数的仕様から関数型プログラムの導出とその実行", 情報処理, Vol.29, No8, pp.881-896(1988).
- [3] 嵩, 谷口, 杉山: "代数的言語の設計と処理系", 榎本編: "ソフトウェア工学ハンドブック", オーム社(1986).
- [4]杉山健司: "知識処理的手法による自動プログラミング", 情報処理, Vol.28, No.10, pp.1297-1304 (1987).
- [5]田中, 辻井: "自然言語理解", オーム社(1988).
- [6]Torii,K.,Morisawa,Y.,Sugiyama,Y. and Kasami,T.: "Functional Programming and logical programming for telegram analysis", Proc. of 7th ICSE, pp.57-64(1984).
- [7]鳥居, 杉藤, 真野, 二木: "プログラム作成技術の現状に関する調査報告[1]", 電子技術総合研究所調査報告(1975).
- [8]辻井, 上原: "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).

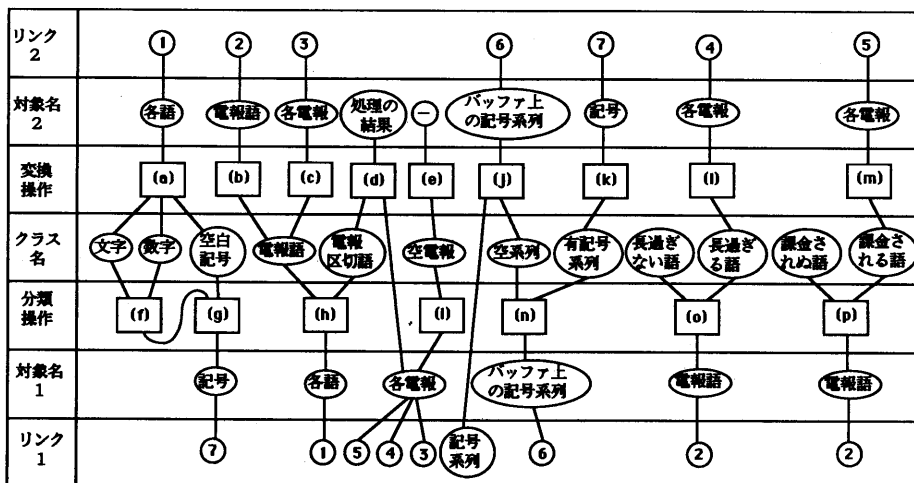


図18 F グラフ (Step4)

表 5 操作の比較照合

Fグラフ	操作文	対応する機能
(a)	3-1	各語を組み立てる
(b)		電報語を記録する
(c)	3-2	各電報を組み立てる
(d)	7-1	各電報、課金語数、長過ぎの有無を表示する
(e)	4-1	電報の流れを終了する
(f)		電報内の文字か数字かを見分ける
(g)	3-1	電報内の空白記号を見分ける
(h)	3-2	各電報を区切る
(i)	4-1	空電報を見分ける
(j)	2-2	電報の流れをバッファ領域に転送する
(k)		記号を取り出す
(l)	5-2	各電報の長過ぎる語の有無を記録する
(m)	5-1	各電報の課金される語数を記録する
(n)		バッファが空かどうかを見分ける
(o)	6-2	長過ぎる語を見分ける
(p)	6-1	課金すべき語を見分ける
	1-1	電報の流れを転送、解析する
	1-2	プログラムを作成する