

## 流れ図文の性質一文記述の違いに着目した分析

佐藤 匡 正†\*

プログラム流れ図の処理や註釈の記述には自然語が使われている。自然語は親しみやすく、解かりやすいという特徴がある反面、表現の曖昧さや暗号的なため難解で誤解を招くことも少なくない。こうした問題点は記述方法の改善によって軽減される。その改善の一つの方向として、多彩な言い廻しの標準化が挙げられる。この標準化では自然語による記述の特徴を失わないように留意せねばならない。そのためには、現状の文がどのように記述されているか、どのような言い廻しの違いがあるかを把握する必要がある。本論文では、流れ図の処理記述および註釈記述の文について使用されている構文および述語を明らかにし、重文や註釈における文脈を分析する。更に文の変型の要因を探求し、その体系化を試みた。また、この分析を通じて処理記述の標準化について検討し、この可能性のあることを示した。

## Characteristics of Flow Diagram Sentences —An Analysis Focusing on Descriptive Variations of Sentences

TADAMASA SATOU †\*

This paper presents an analysis on sentences in program flow diagrams. Natural language description has the merits of being familiar and easy to understand, although it has some risks of being misleading and difficult to understand. These risks may be reduced by some improvement of description methods. Such improvement may be done by the standardization strategy for variations in sentences. To do so the study will be required to know how to describe processes and how many variations of sentences are possible. The paper includes grammatical analysis and description variation analysis on sentences in both text and notes described in program flow diagrams. In the former analysis it is studied syntax and predicate analysis for all sentences, and context analysis in compound sentences and also context analysis between texts and notes. In the latter one, variation factors are studied to find and try to systematize them. Through these analyses it is shown that the description standardization is possible.

### 1. 序 論

プログラム処理の記述には様々な図による記述方法(図法)が用いられている。こうした図法の改善策の一つとして木構造図法があり、多様な提案がある<sup>1)</sup>。更に、この木構造図法の普及と発展を目的として、多様化している図法の支援ツール間につながりを設けて全国規模で図における技術交流の輪を広げる検討も進められている<sup>2)</sup>。木構造図法では制御構造などは形式化して図で示すが、要素処理の記述は従来の流れ図と同様に自然語による記述である。

自然言語での処理記述については、数学的な背景をもつ形式記述に比べて厳密ではなく方法としては不十

分と指摘されている<sup>3)</sup>。しかし、ソフトウェア開発の実務の観点から言えば、非形式の故に、なじみやすく技術者がその経験によって培われた勘や知識を活用できるという特徴がある。形式記述を理想としても一気にこの方法に転換するのではなく、段階的な方法として、このような非形式記述の特徴を生かした記述方法を確立することも考えられる。つまり、現状の記述方法を現状に添って標準化して形式記述方法に近づけるのである。

こうした考えは理想的とはいえないまでも現実的な方策としては有意義と考えられる。設計や保守における文書化技術としてはもちろん、リバース・エンジニアリングにおける資産活用のための基礎技術となし得る。流れ図文を対象として、その性質を解明しようとする研究は小規模な調査結果が報告されている<sup>4)</sup>が、これは予備的な研究の報告であり本格的な研究の前段階に位置づけられる。本格的には、現状の記述特性を

† 日本情報通信コンサルティング株式会社海外事業部  
Overseas Telecom Consulting Division, NIPPON  
TELECOMMUNICATIONS CONSULTING CO., LTD.

\* 現在、情報処理コンサルタント

Presently, consultant of information processing

解時した上でその標準化の可能性を検討する。この場合、記述特性の解明に難しさが有り、自然言語における記述の自由度が問題となる。流れ図文を自然語とみたときの特徴は記述の簡略指向にある。これは文を一定の枠内に記述するという制約や書き手の記述量を最小限にしたいという欲求から生ずるものと推定される。

この簡略化には文成分の省略や結合などの様々な方法が含まれており一意には決まらない。同一の意味をもつ文が、複数の類似表現の文や全く別の文になることはよく経験する。この違いはその一部の違う些細なものから全面的なものまで含まれるが、自然語記述の一つの特徴であり、同時に技術的には分析を難解にしている要因である。従って、記述特性の解明ではこうした多様な言い廻しの違いの要因を見極め整理することが主題となる。

本論文では実際の流れ図文の記述特性を分析し、その標準化の可能性について検討する。記述特性の分析では、使用されている構文や述語の性質を示し、重文や註釈における文脈について述べ、言い廻しの違いが生ずる要因を体系化する。更に、この結果を踏まえて、標準的な記述方法の可能性について検討する。

## 2. 分 析

### 2.1 準 備

#### 2.1.1 用 語

##### (1) 基本用語

本論文の議論では日本語文法の基本的な知識を用いる。日本語文法に関しては山田文法や時枝文法など様々な学説があるが、高等学校以下の国文法では橋本進吉の説（橋本文法）に準じて教育されている。その定義や解釈には異論も多いのだが、議論での用語を確定させるという点からは橋本文法で十分と考えられる。文や文節など基本的な用語は橋本文法に添ったものを用いる。

##### (2) 流れ図文

流れ図の記述は線図と文に分けられる。線図の箱の中・外に記述されている自然語の文章もしくは記号列を流れ図文と呼ぶ。流れ図文を、その文の構造および記述されている内容で次のように分類する。なお、メタ記号として「」を用いる。

##### ① 文構造による分類

自然語におけるのと同様に、文の構造から単文、重文、複文に分ける。橋本文法に従えば主語と述語の関係によって節が構成される。この節が一組だけの文は単文である。複文は複数の節があり、これらが従属な

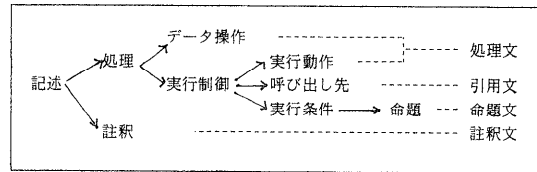


図1 記述内容からみた文の種類

Fig. 1 Sentence classification by its description.

どの対等でない関係の文である。重文は複数の対等な節、つまり対立節をもつ文であるが、これだけに限定せず箱内の複数の単文も含むように拡張する。

##### ② 記述内容による分類

この流れ図文を記述内容から分類する。流れ図文では通常、箱の内側には処理が、外側には註釈が記述される。流れ図の図形と対応づけ、長方形の箱に記述されている文を処理文、変形長方形の箱で呼び出し先を記述されている文を引用文、菱形の箱で実行条件を記述している文を命題文、および註釈を記述している文を註釈文と名付けることとする。註釈には文だけではなく、表や図も用いられるが、ここでの分析の対象は文に限定する。

##### 2.1.2 分析の対象

分析対象に一般性がなければ、その結果の一般性は保証されない。一般性の要求条件として、①組織的に記述され整備されている、②試作でない、③多様な処理を含む、が挙げられる。この③の条件に関しては、どのような処理を選択するかが問題となる。一般に、処理は計算機向き処理と業務向き処理に大別でき、業務向きの処理は適用対象によって多様であるが、計算機処理の方は比較的共通性がある。

これらの条件を勘案して、計算機向き処理に絞りオペレーティング・システム[OS]の実行管理サブシステムの一部を分析の対象とする。これはOSの基本機能の一つであり、タスクの生起・消滅および実行や同期などの制御を担う。この処理には、データ構造の扱いからI/O処理に至るまでの基本的なプログラム処理が含まれていて一般性があると考えられる。

この流れ図の記述形式はJIS流れ図規定に準拠している。記述の水準は、詳細化されていてプログラミングを対象として記述されている。量はA3用紙で352枚である。分析の対象は、流れ図文のうち文の性質が類似している処理文、引用文、およびこれらに対する註釈文を分析の対象とする。命題文については、これらの文とその性質が異なるので稿を改めて論ずることとする。

流れ図文の出現頻度を表1に示す。全体で4940文の

うち 3285 文が分析の対象で、文全体の約 65% を占める。

## 2.2 分析の方法

ここでの分析では、文の整形→構文解析→変型の分析→文脈の分析の手順で進める。このうち、整形方法と構文の表現方法についてここで述べ、変型の分析方法および文脈については節を改めて述べる。

### 2.2.1 文の整形

文の整形は、表現上の軽微な違いを吸収し、以降の分析を円滑に進めるためのものである。文の単文化および表現の正規化の方法を次に示す。

#### (1) 単文化

重文および複文には、文を形成する複数の節が含まれている。これらの節を次のように切り出す。

##### ① 重文

一文に複数の対等節が含まれる場合は、節を単文として独立させる。また、対等節間で共用している文成分または品詞は展開する。

##### ② 複文

一文に従属節が含まれている場合、その節を原文から切り出し、文として独立させる。例えば、「UDBLU をクリアし、BRUNIT が指す 2 バイトを UDBLU にセットする」は、「UDBLU をクリアする」、「2 バイトを UDBLU にセットする」、「BRUNIT が指す」の三つの単文になる。

##### ③ 「として」

「として」は重文として扱う。文法書によっては、これを資格や立場あるいは状況を表す助詞とすること

もあるが、この立場はとらない。重文の前半の節は「とする」で終わる単文であり、後半はその対格が省略されている単文とする。例えば、「リターン・コードとして  $\alpha$  値をレジスタ<sub>1</sub> に設定する」は、「 $\alpha$  値をリターン・コードとする。（この  $\alpha$  値をレジスタ<sub>1</sub> に設定する）」と解釈する。

#### (2) 表現の正規化

文成分の省略や文節もしくは複合節のみの表現などの記述があるが、こうした不完全な文に対しては、不足している成分の補足や語幹の動詞化などの措置によって、元の文と意味の等価な文に整形する。このように整形した文を正規化文という。この正規化の措置では表現上の変化を最小限に抑え原文表現を保持するために表 2 に示す原則を設ける。

#### (3) 述語の正規化

述語に用いられる動詞の表現は多様であり、正規化して統一する。この多様化の要因は、①形式、②活用形、および③類似表現に分けられる。①の要因では仮名や漢字の日字、英字が混在して使用されている。実例を引けば、「よみあげる、読上げる、読み上げる」とか「アペンドする、APPEND する、'APPEND' する」といった具合である。②の要因に関しては、時制、態、助動詞、否定形が問題となる。これらは終止形に統一して扱う。

③の要因では、微妙なニュアンスの違う動詞の存在で生ずる多様さである。これに関しては強調表現、翻訳語、同義語がある。実例を引けば、「つなぐ」に対して「つなぎ込む」といった強調表現、「セットする」に対して「設定する」といった翻訳語、「(値を) 引く」に対して「減ずる」といった同義語が挙げられる。動詞の正規化の様子を表 3 に整理する。

### 2.2.2 構文の表記方法

#### (1) 基本

構文の形式的な表現方法の一つとして、文成分の種類とその存否を示す方法が考えられる。文の成分とし

表 1 流れ図文の出現頻度

Table 1 Frequency numbers of flow diagram sentences.

記述	処理文	註釈文	計
処理	1816	575	2391
引用	644	250	894
命題	1313	342	1655
合計	3773	1167	4940

表 2 文の正規化措置項目

Table 2 Sentence regularization.

正規化項目	措置	ねらい	例
動詞語幹の句表現	動詞化	成分の明確化	リターン・コードの設定 →「～を設定する」
省略成分	補充	文の標準化	上例で「*に」を補う
単語のみの表現	文表現化	〃	BR 7 →「*は BR 7 である」
コード値表現	標準化	分析の容易化	C 84 C 0602, '...'H, '...'X →「X'...'」の形式に統一

表 3 動詞の正規化  
Table 3 Verb regularization.

正規化の対象	基準		例	記事
表現の統一	文字の種類	漢字を原則*)	よみあげる, 読上げる, … →「読み上げる」	*) 英単語は片仮名 マクロ名は英字
	送り仮名	標準		
活用形	時制	現在	読んだ →「読む」	
	態	能動態	指定された →「指定する」	
	接続助詞	削除	完了している→「完了する」	
	否定助動詞	削除	転送せず →「転送する」	
類似表現	強調表現	標準語に統一	つなぎ込む →「つなぐ」	
	翻訳語	日本語に統一	セットする →「設定する」	
	同義語	標準語に統一**)	(値を)引く →「減ずる」	

て存在するものを記号#とその格を代表する助詞で、またこの文節が表現上で省略された場合は、記号\*と助詞で表す。特に、述語は「V する」で示す。この表記方法の例として、「処理数を JFB の先頭 MDB に記憶する」は与格と対格のみからなる構文なので「#を#に V する」と表記する。この記述方法による構文の分類を構文の型という。

#### (2) 式の扱い

文として、記号←、→や＝で右辺と左辺を結んだ式が用いられる。この場合は値の設定元を右辺成分、設定先を左辺成分に標準化し、記号を述語とする。設定先を左辺成分に標準化し、記号を述語とする。例えば、「(ICBAD)→退避アドレス」では左辺成分は「退避アドレス」、右辺成分は「(ICBAD)」で、述語は「→」である。また、式中に算術式が含まれることがあるが、このような場合はこの式全体を文の成分として扱う。例えば、「A+B を C に設定する」では「A+B を」が対格である。

### 2.3 変型の分析方法

変型を分類し、それぞれの変型の分析内容について述べる。

#### 2.3.1 変型の種類

文の変型を形式及び意味の上から、次の三種類に分類する。

##### ① 形式変型

構文の型が同一で、かつその文成分のすべてについて直接対応のある変型をいう。文成分の入れ替わる転置や文成分の省略などによって生ずる。正規化による変型は、ここに含まれる。

##### ② 同成分変型

構文の型は異なるが、そのすべての文成分が対応するような変型をいう。成分間の結合、述語の言い換え、

態の変化などで生じる。

##### ③ 言い廻し変型

文成分の対応は不明確だが文全体の意味が同じ変型をいう。記述の観点が異なるために生ずる。例えば、「ファイル名は全字空白文字である」と「ファイル名が指定されていない」では、文成分の対応付けはしにくい。同じ意味を表す文同士である。前者は、後者に対する判断方法を記述している。

#### 2.3.2 形式変型

##### (1) 文成分の省略

一つ以上の文成分の省略による変型である。これは、記述の簡略化をねらいとするものと想定される。

##### (2) 転置

文成分の記述順序が入れ替わった変型をいう。日本語では、基本的に文成分の順序についての制約は弱い。後述のように、動詞と結びつきが強い文成分のある場合は記述順序は確定して転置が不自然な表現となることもある。自然語では文成分の統計的な傾向からその並び順は「時の修飾語→場所の修飾語→主格→与格→対格→述語」としている<sup>5)</sup>。これに対して流れ図処理文では、主格の省略は多いが、修飾語の順序よりも与格と対格の順序が問題となる。

##### (3) 正規化

第 2.2.1 項で述べた正規化は形式変型に含まれる。

#### 2.3.3 同成分変型

##### (1) 対格統合

対格に他の文成分を統合して生ずる変型である。統合には助詞「の」が用いられる。統合される文成分には、修飾語と述語がある。

##### ① 場所の修飾語

修飾語の文節「#から」「#だけ」「#で」などは本来、述語の動詞を修飾している。こうした修飾語は対格と

連結して対格を修飾するような構文に変型でき、異なる型の構文となる。「#だけ」や「#から」の成分は「#{だけ, から}の#を」といったように与格に吸収される。例えば、「入力パラメータよりFCBアドレスをEDB域に得る」という文ではFCBアドレスの所在を「#から」成分で示している。この文は対格の修飾語とした「入力パラメータのFCBアドレスをEDB域に得る」という変型もある。このような変型では転置が生じやすい。これは対格の記述が長くなるためと考えられる。これは長い修飾語は短いものより前に置くという自然語における指摘<sup>9)</sup>と符合する。

## ② 述 語

述語が、動作を示す語幹と接尾語「する」からなる動詞の場合はこの語幹と対格の統合が生ずる。構文「#<sub>1</sub>に#<sub>2</sub>をVする」は「#<sub>1</sub>に#<sub>2</sub>のVをする」の構文に変型し得る。この構文の動詞は形式動詞と呼ばれ、「する」の他に「行う」が使用される。

x に y 値を設定する  
 ↑↓  
 x に y 値の設定を行う

### (2) 与格の動詞化

値を扱う文において「#<sub>1</sub>を#<sub>2</sub>にVする」の型の構文では変型が想定できる。与格が値または動作を表す単語でかつ述語が「行う」とか「する」といった形式動詞では与格が動詞化された変型「#<sub>1</sub>を#<sub>2</sub>する」が生ずる。

#### 2.3.4 言い廻し変型

##### (1) 文 意

データ操作の処理文において型は同一で意味上からは与格と対格が互いに逆なことがある。この変型は「#を#にVする」構文だけで生ずる。この型の次の二つの文は動詞のニュアンスによる微妙な差はあるが同値としてよい。

x に y 値を設定する  
 ↑↓  
 x を y 値にする

両者は形式的には与格と対格の位置が入れ替わっている転置変型だが、意味上からは両者の対格と与格の役割が互いに逆である。なお、下の文は転置変型が不自然となる例である。

##### (2) 修飾の方法

修飾のし方によって変型が生ずる。修飾には、状況に応じて文節や複文、重文が用いられる。文成分と結び付けた対等節で修飾すれば重文構造に変型するし、従属節とすれば複文構造に変型する。重文構造では、修飾語と被修飾語が対等節に分けられその修飾関係が

暗示される。例えば、単文「#<sub>1</sub>から#<sub>2</sub>を#<sub>3</sub>に回復する」における場所の修飾語と対格を別個の対等節に分けた重文「#<sub>1</sub>から#<sub>2</sub>を得て#<sub>3</sub>に回復する」に変型できる。前半対等節の動詞としては値を求める動作の動詞である「求める」、「算出する」などが使用される。後半の対等節における動詞は「設定する」、「回復する」、「回避する」などが使われる。例えば、「PXAよりSVTアドレスをレジスタ<sub>1</sub>に設定する」の変型は「PXAよりSVTアドレスを得てレジスタ<sub>1</sub>に設定する」である。

複文構造では、重文構造における修飾節が対格を修飾する従属節に変換できる。ここで従属節の文体や構文は文意による。上の例では「PXAより」という修飾語を対格「SVTアドレスを」を修飾するように変えて、「PXAより得た」、あるいは「PXAに回避してある」という修飾節にした変型がある。

##### (3) フラグ処理記述文の変型

言い廻し変型の代表的なものとしてフラグに対する処理記述がある。制御プログラムではフラグが多用される。フラグは構造をもたぬデータ型としても、また関連する複数個を組にしたレコード型としても使用される。この場合、上位層はフラグと呼ばれ、下位層は表示、スイッチ、標識、インディケータなどと様々に呼ばれるが、ここでは表示で代表される。いずれの場合にしてもフラグはオン/オフの二値(ビット)をとる。この二値によって、ある事態の生じたこととそうでないことを表わす。プログラム処理では、このフラグを様々なモジュールで適当に値の参照・設定が行えるような仕掛けを作る。

このフラグに対する処理記述は言い廻し変型が生ずる。この変型を整理する考えとしてフラグを抽象データ型と捉えて、くくりつけられた操作(演算)に注目する方法がある。操作は大別して設定と参照がある。参照操作は命題文で表わされるが、これについては別稿に譲り、ここでは設定操作に絞って分析する。フラグのオン値設定操作で用いられる構文を整理すると、表9のように10種類に整理できる。

表9の変型の要因は「オン」をどの格へ位置づけるかによるものと推定される。この位置づけによって関連する文節同士が接合され変型が生じている。対格の場合は値そのものに注目しているのであり、与格および述語において値を変えることに注目している。

#### 2.4 文脈の分析方法

文脈の分析については重文と註釈に関するものに限って分析する。

##### (1) 重文の文脈

重文の対等節間の文脈、つまり対等節間における文

節の結びつきについて分析する。重文に含まれている対等節の、一方に含まれる文節が他方の節と意味上で結びついて構文が変化することがある。例えば、「#か

ら#を得て#に設定する」という重文において、前半では与格が、後半では対格がない。単文の構文ではそれぞれの成分が必要となるのに対して重文では「得て」の対格と「設定する」の与格の結びつきが重要となる。共通な文成分の省略された形式変型と捉えることもできるが、ここでは、重文固有の文脈として分析する。

(2) 註釈文の文脈

註釈は処理文および引用文の処理記述（以下、本文という）に対する補追である。補追をその対象について整理すると、文全体に対するものと文の成分に対するものとに分けられる。この様子を図2に示す。文全体に関するものはその記述に対して不足している処理目的 (what) やこの実現手段 (how) の追記が中心で

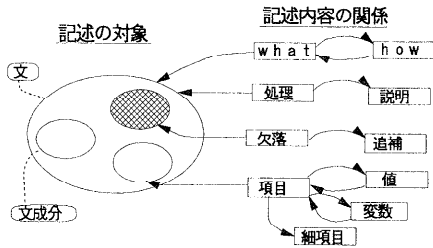


図2 本文と註釈の関係

Fig. 2 Relationship between text and its notes.

表4 構文の型の出現頻度

Table 4 Frequency numbers of sentence construction types.

分類	構文の型					処理文	註釈文	複文	[%] 計	動詞数 <sup>*)</sup>	変形数
	左辺	右辺	たは	かめ	だにけ						
無											
	主格										
主格											
式											

\*) 原形動詞

ある。一方、文成分に関しては、欠落している文成分の追補、項目の具体的な値、項目に対する変数名、項目の構成細項目の追記などである。これらの状況を分析する。

### 3. 分析結果

#### 3.1 構文の分析

##### (1) 出現頻度

処理文および註釈文における構文の型の出現頻度を表4に示す。全体で35種の型が観察された。

これらの型を分類する。形式上から文形式と式形式に分けられ、文形式は更に主格の有無で無主格と有主格に分ける。無主格の型は動作の記述に使用されるが、その主格は計算機（もしくはプログラム）に固定されているので省略される。有主格の型は説明の記述に使用され、説明の対象によって主格が変わる。「#は#である」の型が主であるが、「#は#に設定されている」といった受動態も使用される。

この分類による出現頻度は、無主格の型では25種類、有主格の型では8種類、式形式は2種類である。

表5 正規化要因別件数

Table 5 Numbers of verbs classified by regularization factors.

項目		動詞種別	原形	正規化
対象の動詞数			110	37
正規化要因	表現方法	文字	18	12
		仮名	20	16
	活用形	態	6	6
		他	4	4
	類似語			26

表6 動詞と構文の型

Table 6 Verbs and sentence construction types.

動詞	型の種類	処理文	註釈文	計
交換する	3	5	—	5
呼び出す		2	5	7
オンする		27	9	36
ゼロ・クリアする		69	15	84
求める		3	3	6
減ずる		21	—	21
更新する		33	4	37
復帰する	26	5	31	
設定する		438	105	543
行なう	4	—	7	7
辿る			3	12
する			13	75
得る	6	218	64	282

すべての文に対しての出現比率はそれぞれ、75.0%、12.0%、13.1%である。

無主格の型での出現頻度の高い型は、「#に#をVする」が55.9%、「#をVする」が15.3%、「#から#に#をVする」が12.5%の順である。三つの型の合計は83.7%で集中しているといえる。

有主格の型では「#は#である」の型に集中して88.6%を占める。この型での受動態の出現頻度は2%以下である。式形式は二文節の「#→#」が99.4%であり、述語の記号には→、←、=の三種類が使用される。

##### (2) 述語の正規化

原形動詞は201種類使用されている。この原形動詞に対して表3で示す措置によって正規化した結果、128種類に削減できた。この様子を正規化の要因に分類して表5に示す。正規化の対象は原形動詞のうち110種類であった。これらを、表現に関して38種、活用形10種、類似表現語26種について正規化した結果、37種類に集約された。

n個の原形動詞が一つの正規化動詞に集約できるということは、nは動詞表現の表現方法の多様さを表しているものと捉えられる。この観点から変型数を観察する。この最大値は「減ずる」における6である。次は「つなぐ」、「呼び出す」、「オフする」、「オンする」の4である。なお、動詞の出現数の合計は同一の動詞が複数の型において使用されることがあるので動詞の種類とは一致しない。

##### (3) 動詞と構文の型

動詞がどのような構文の型をとるかに着目する。複文を除く無主格の型について観察すると、一つの動詞が3以上の型をとるものは12種類存在する。この様子を表6に示す。出現頻度の多い「設定する」と「得る」に注目する。「設定する」の型には「#に#を」、「#に#で#を」、「#から#に#を」があるが、先頭の型が大部分(98.7%)である。

これに対して「得る」は表7に示すように六つの型

表7 「得る」の構文の型

Table 7 Sentence construction types of 'uru'.

文の成分				出現頻度 [%]
から	だけ	を	で	
#	#			27.6
#	#	#	#	0.3
#	#	#		0.3
#		#		1.4
#	#	#		68.5
#	#	#	#	0.3

が使用されていて、「#から#に#を」と「#に#を」の二つの型に集中している(96.1%)ことが分かる。

3.2 変型分析の結果

(1) 文成分省略の形式変型

表4の変型数の欄に構文の型ごとにその変型の出現数を示す。この変型は形式変型の成分省略によるものである。この中で最も多い変型数をもつ構文の型について、無主格の型における文成分省略の形式変型の様子を表8に整理する。

表8は、2成分をもつ構文における変型についての二事例、および3成分をもつ構文での変型事例を示す。2成分構文ではどちらの事例でも省略の全パターンが

出現している。しかし、「#を#で」の型では「で」成分の省略された「#を\*で」が72%であるのに対して、「#に#を」の型では成分省略のない「#に#を」が52.6%出現している。この違いは用いられる動詞によるものである。

前者の型では「更新する」がよく用いられる。この動詞では手段を示す「で」成分はよく省略される。これに対して後者の型では「設定する」がその代表であり、対格と与格の両方を明に記述する。

三成分の型では、そのすべての成分を省略したパターンを除いた7パターンが出現している。「\*から\*に#を」と「#から\*に#を」が同数出現し両者で大部分(89.2%)を占める。前者の型での代表的な動詞は「復帰する」であり、場所を示す修飾語「から」の成分はよく省略される。後者は「求める」が代表的であり、対格の省略は、「から」修飾語によって推定されるためである。

有主格の型については、最高頻度の「#は#である」について注目する。主格の省略された「\*は#」の型が81.2%で、主格をもつ「#は#」が18.8%である。前者は註釈としてリターン・コードやメッセージなどに関して具体的な値を示す場合によく用いられる。主格が省略されるのは述語(#である)の記述からこれが容易に推定できるためと考えられる。

(2) 転置の形式変型

文成分の転置に関して、出現頻度が多く転置可能な動詞「設定する」を用いた構文を観察する。対格と与格の両方をもつ構文は62.3%あり、「与格→対格」が18.7%を、「対格→与格」が43.6%である。これから流れ図文の文成分の順序は、「時の修飾語→場所の修飾語→対格→与格→述語」が主で転置は30%程度生じているといえる。この転置の生ずる要因として次に述べる

表8 形式変型(文成分省略)

Table 8 Format variations (abbreviation of sentence component).

文の成分	出現頻度 [%]
から に を で	
* *	2.0
* #	2.0
# *	72.0
# #	24.0
* *	0.3
* #	37.9
# *	9.2
# #	52.6
* * #	44.6
* # *	0.4
* # #	1.6
# * *	2.1
# * #	44.6
# # *	1.2
# # #	5.4

表9 フラグの設定操作の構文

Table 9 Sentence constructions in assignment sentences in flag mech.

データ型	対 格	与 格	述 語
無構造	$\alpha$ フラグ	—	オンする
	↑	オン	する*)
	オンの設定 オン	$\alpha$ フラグ ↑	行なう, する 設定する
レコード データ型	$\beta$	↑	表示する
	$\beta$ 表示	↑	オンする, 設定する**)
	$\alpha$ フラグの $\beta$ 表示	—	↑
	↑ オンの設定 オン	オン $\alpha$ フラグの $\beta$ 表示 ↑	する*) 行なう, する***) 設定する

\*) 「と」の文成分を含む。

\*\*) 「立てる」を含む。

\*\*\*) 「 $\alpha$  フラグに  $\beta$  表示の設定を {行なう, する}」を含む。



対格の文成分統合が挙げられる。

(3) 同成分変型

与格の動詞化変型を起こす単語は少なく、「表示」, 「オン/オフ」が代表的な例である (表9 参照)。

(4) フラグ処理記述の変型

表9のレコード型の構文の出現頻度の目安として述語の出現頻度を見る。これを表10に示す。これによれば最も簡潔な構文となる「表示する」が60%で最もよく使用されている。

(5) 変型の体系化

同成分変型や言い回し変型は単独の変型もあるが、これらが組み合わせられ複数の変型が一つの体系をなすことがある。この体系は変型の要因によって変型間の関係が図示できる(変型体系図)。表9のフラグ設定操作の変型体を体系化した変型体系図を図3に示す。図3では表9の無構造データ型の文を矩形で囲んで示し、矢付き線で変型の方向を示している。このとき、この変型要因を線横の[ ]内に示してある。

3.3 文脈分析の結果

(1) 重文の文脈

処理記述における重文89文から重複した記述を除いた81文について考察する。このうち、45文は上のよ

うな文脈をもつ文である。文脈の分析結果を表11に示す。表では前後の対等節において関係をもつ文節を示している。この表から、重文は前対等節の対格と後節の対格または与格に文脈をもつ構成の多いことが分かる。

(2) 註釈文の文脈

註釈率(註釈文数/記述文数)は処理文が31.7%で、引用文が38.8%である。次に、註釈の用途を8項目に分類し処理文に対するものと引用文に対するものとに分け、これらの出現頻度を表12にまとめる。

文全体に関する註釈について、処理文では約半数(55%)であるのに対して、引用文ではそのすべてである。これは引用文の本文はルーチン名だけのためである。註釈には、引用する処理の働き(what 記述であり、95.3%出現)やパラメータや呼び出し規約などの引用方法(how 記述で4.7%出現)が記述される。一方、処理文の註釈では、どのように実現するかを記述するhow 記述が77.4%で、処理目的の what 記述が22.6%である。これは本文はコードに対して what 向きの記述であることを意味している。

文成分の追記の註釈は処理文に対してだけである。追記の内容は処理の説明とコーディングへの配慮である。この類いの註釈は全体の11.3%で、このうち項目の補追は7.51%(66.5%)である。残りは本文中に記述されている値についての説明である。一方、コード

表10 述語の出現頻度  
Table 10 Frequency numbers of predicates.

述語	表示	フラグ
初期設定する	1	1
リセットする	14	2
オフする	9	—
ゼロ・クリアする	3	3
取り消す	8	—
消す	1	—
消去する	1	—
立てる	—	1
オンする	7	3
表示する	68	1
設定する	27	1
する	6	4

表11 重文の文脈  
Table 11 Context in a compound sentence.

前対等節		後対等節			出現数	
修飾	対格	与格	修飾	対格		与格
		#		*		4
	#				#	2
	#				*	9
	#			#		3
	#			*		14
	#		*			1
#				#		12

表12 註釈の用途の分類  
Table 12 Classification of notes by their usages.

註釈の用途	処理文	引用文
how (実現手段)	42.7%	4.70%
what (処理目的, 働き)	12.5	95.3
欠落文節, 値の説明	11.3	—
具体的な値の明示	14.1	—
条件付け具体化	11.1	—
変数名の明示	8.34	—

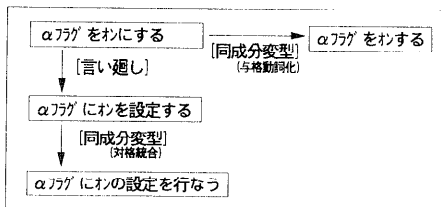


図3 フラグ設定操作の変型体系図

Fig. 3 A variation transition diagram of assignment sentences in flag mech.

ィングへの配慮は、変数名やコード値など具体的な値の明示によるもので、全体の33.5%を占める。上述のhowもコーディングに関するものと解釈できる。これを含めれば全体としては76.2%を占め、註釈の大部分はコードへの配慮と言える。

#### 4. 考 察

ここでは、これまでの分析結果を踏まえて処理記述の標準化の可能性について考察する。

##### (1) 変型の統一

処理記述は画一的な堅い記述方法であるよりも、現実添って規定に幅をもつものが効果的と考えられる。次の論拠からこのような記述方法を制定できると期待される。

##### ① 正規化表現

記述の正規化によって形式変型の削減できることは第3.1.2項で示した。表3の措置による正規化で201種類が128種類まで削減されたが、標準化であれば更に削減は可能である。基本的には、基準となる動詞を限定し、この構文の型と語順を制定する。

動詞の限定において留意すべきは拡大解釈の点である。一般に、意味を包含している動詞への置換は、元の意味を大まかには保持できる。しかし、適切な動詞を使うことが原則であり、汎用的な動詞を基準に選ぶと問題を生ずることがある。例えば、「x 値を y に回復する」において、「回復する」の代わりにより一般的な意味の動詞「設定する」を基準とすると文脈ニュアンスは消える。「x 値を y に設定する」という文は「y の値が x 値になる」点では等価であるが、「回復する」で暗示されている「x 値が退避されている」ことは消える。

##### ② 省略方法の確立

処理の記述に省略は有効である。省略をせず全成分を記述した文は教育などには有効なこともあるが、まわりくどい印象を与え実務には向かない。記述の手間がかかるだけでなく、文章が冗長かつ生硬になる。分かりやすいとはいえない。

省略が可能なのは書き手と読み手に共通する知識が備わっていることが前提である。書き手の省略は読み手のもつ、こうした共通知識、つまり常識に基づいて文意が理解される。処理記述における与格の省略、更に動詞の語幹化といった短縮記述や註釈記述における主格の省略は、読み手が容易に省略内容を推定できるからである。しかし、読み手にこの常識が欠けていれば文全体の意味は分からなくなり、誤解を招いたり難解になったりする。

文成分の省略方法の確立にはこうした常識の確立が必要である。現状では第3.2.1項で述べたように構文の型や用いている動詞によって特定の省略パターンに集中している。これは、おおむね常識の枠があり省略方法として整理できることを意味している。

##### ③ 基準の型の制定

単純な変型では基準の型の決定は容易である。一方、同成分変型と言い廻し変型が組み合わされ体系化している変型では、その基準の文は確定しにくい。この方法として第3.2.4項で述べた変型体系図が利用できる。図3のフラグの設定操作の変型体に対しては、「 $\alpha$  フラグをオンにする」という表現が基準の文であることが分かる。

##### (2) 註釈記述

註釈は本文と一体化してその記述方法を考えた方が自然である。処理記述は、what-how の組で記述されていると分かりやすい。what は処理の目的や用途を表し、how はその実現方法を表す。更に、このhow が次の層における what となる階層構造を成し、最も抽象化された最上位層から最下位層のコード・レベルに詳細化されて記述されるのが一つの理想である<sup>7)</sup>。しかし、流れ図にはこうした階層の記述機能は備わっていないので処理記述では、本文と註釈の関係でこれを実現することができる。

この場合、what-how のどちらを本文とするかが問題となる。本文を what 表現とし、註釈を how 表現とするのが自然である。なぜなら、what 表現は、how 表現よりも簡潔で分かりやすく人間向きである。しかも、変型が生じ難い特徴があるため標準化に向いている。例えば、第2.3.1項で述べたファイル名の指定の例で、「ファイル名が指定されていない」が what 表現で、この実現手段の「ファイル名は全字空白文字である」は how 表現である。

現状においても第3.3.2項の分析から、処理記述の註釈率は3割強であり、註釈をもつ文に関してはその8割弱がコードへの配慮である。これから現状は本文で what、註釈で how といえる。これは、本文の用途を人間への処理説明用とし、註釈はこのコード化用としているためと解釈できる。

#### 5. 結 論

流れ図の処理記述および註釈記述の3285文について使用されている構文および述語、重文や註釈における文脈について分析を行った。更に文の変型の要因を明らかにし体系化を試みた。その結果、次が得られた。

(1) 記述特性として次を明らかにした。

1) 使用されている構文は処理記述では無主格の型であり、註釈記述では有主格の型と無主格の型の両方が用いられる。無主格の型の構文は25種類あり、この出現比率は全体の75%である。最も出現頻度の高い構文は「#に#をVする」で全体の56%である。なお、式形式は13.1%出現している。

2) 述語については原形で出現した201種類の動詞のうち、55%が表現形式や類語の整理による正規化の対象であり約1/3の種類に削減できた。

3) 構文と述語の関係に関しては、複数の構文で使用される動詞でも特定の構文の型に集中する(95%以上)。

4) 重文の文脈では前対等節の対格と後節の与格に関係をもつことが観察された。

5) 処理記述の註釈率は3割強である。註釈の文脈としてはコードに関するものが最も多く76.2%である。これは処理記述がコードよりも人間向きのためである。

6) 文の変型を三種類に整理した。変型には単一要因だけでなく、要因の複合も生ずるが、こうした変型体は変型体系図で体系化することによって、基本文が確定できることを示した。

(2) 処理記述の標準化に対してその可能性を示した。正規化表現の方法、基準となる省略の方法、基準の型の制定方法の三点から、また註釈の標準的な記述方法に関して検討した結果、その可能性のあることを示した。

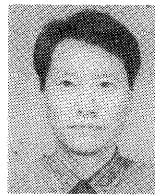
## 参 考 文 献

- 1) Toripp, L. L.: A Survey of Graphical Notations for Program Design —An Update, *ACM SIGSOFT SOFTWARE ENGINEERING NOTES*, Vol. 13, No. 4, pp. 39-44 (1988).
- 2) 長野宏宣ほか: 木構造図用 CASE ツール間のデータ交換言語: DXL, *情報処理*, Vol. 35, No. 5, pp. 341-349 (1994).
- 3) Sommerville, I.: *SOFTWARE ENGINEERING 4th ed. (7. Formal Specification)*, Addison-Wesley Publishing Company (1992).
- 4) 佐藤匡正: プログラム処理記述文に関する分析, 第30回情報処理学会全国大会論文集, pp. 735-736 (1985).
- 5) 佐伯哲夫: 現代文における語順の傾向—いわゆる補語のばあい, *言語生活*, 111 (1960).
- 6) 本田勝一: 日本語の作文技術, 朝日新聞社 (1976).
- 7) 佐藤匡正: 解法を表わすプログラム図法の実現性分析, *情報処理学会論文誌*, Vol. 27, No. 11, pp. 1129-1136 (1986).

(平成6年8月4日受付)

(平成7年2月10日採録)

### 佐藤 匡正 (正会員)



昭和42年横浜国立大学工学部電気工学科卒業。工学博士。昭和42年日本電信電話公社(現、日本電信電話㈱)入社。電気通信所勤務。昭和44年から同47年にかけてデータ通信本部において各種データ通信システムの開発に従事。この経験に基づき、昭和47年から横須賀電気通信研究所において一貫してソフトウェア開発技法の研究・実用化に従事、主導。昭和63年退社。平成元年より横浜創英短期大学に勤務、同6年退職。現在、情報処理コンサルタント。専門は、ソフトウェア工学で、特に、実践的な技法に興味をもち、現在はリエンジニアリングに関心がある。情報処理論文賞受賞(昭和56年度)。プログラム図法 HCP 考案者。主要な著書「ソフトウェア・デザイン・レビューの実践技法(ソフト・リサーチ・センター刊(1990))」, 訳書「ソフトウェアエンジニアリング(フジ・テクノ・システム刊(1993))」。