

制限自然言語によるソフトウェア要求記述とその解析

原林利幸 河合敦夫 椎野 努 武内 惇
三重大学工学部情報工学科 日本大学工学部情報工学科

電子計算機システムの発注者や使用者とその設計者の間における、要求仕様に関するコミュニケーションの道具として、自然言語で記述された要求仕様書が用いられている。要求仕様記述の曖昧な記述や、矛盾した記述、記述漏れを検出するために、電子計算機を用いた要求仕様記述の解析が行われている。記述内容を解析するための方式には、①解析に必要な一般知識と分野知識を全て用意する方式、②解析上曖昧性を生じない程度に記述法に制限を設ける方式がある。あらかじめ必要な知識を全て用意することには限界がある。また制限を強めると、記述文章は読み難く、書き難くなる。本論では分野知識は使用せず、一般知識のみを使用して記述内容を解析することを前提とした、適切で明確な制限を設けた要求仕様記述用制限日本語を提案する。本論では、要求仕様記述の構成と表現パターン、記述文章の構文、解析法、記述例について述べる。

Software Requirement Specification in Controlled Natural Language and its Analysis

Toshiyuki HARABAYASHI Atsuo KAWAI Tsutomu SHIINO
Department of Computer Science Mie University
Tsu, Mie 514, Japan

Atsushi TAKEUCHI
Department of Computer Science Nihon University
Kooriyama, Fukushima 963, Japan

A software requirement specification document which is described in natural language has been used for communication between computer system's customer or user and designer. In order to detect ambiguous expressions, inconsistent expressions and omitted expressions in the requirement specification descriptions, it is often used to analyze them by computer system. To analyze the descriptions,

① general knowledges and particular knowledges referred to ordered system's environment (field knowledges) must be prepared. ② the styles of expression must be controlled until ambiguous expressions cannot be written. But in reality, it is difficult to prepare all field knowledges, and too much style control make the expression be difficult to read and write. In this paper we will propose controlled Japanese natural language for software requirement specification description. The expression style of this language is controlled so as to be analyzed by computer system, only using general knowledges, without using special field knowledges. We will also present the expression patterns of requirement specification descriptions, the syntax of this language, the method of analysis and the examples of description.

1. はじめに

ソフトウェア要求記述を形式化して、曖昧性や誤り、矛盾、抜けをなくし、計算機による検証を可能にしようとする試みは、これまで数多く行われてきた。これらを大別すると、主として処理構成やデータの流れ等の機能的側面に主眼をおいてモデル化したもの（データフロー図¹⁾、SADT²⁾、ジャクソン法³⁾等）と、処理の順序やタイミング等、動作的側面に主眼をおいてモデル化したもの（HIPO⁴⁾、HCPチャート⁵⁾等各種拡張フローチャート、状態遷移図⁶⁾、R-NET⁷⁾、ベトリネット⁸⁾等）に分けることができると考えられる。これらは、主眼とする側面からはある程度分かりやすい表現となるが、他の側面からは十分に表現しきれないか、あるいは表現できても複雑で分かりにくいものとなる傾向がある。また、これらの図表はソフトウェアの専門技術者には理解できても、発注者や使用者等のソフトウェアの専門外の人には理解しにくいという難点があり、設計者と発注者や使用者との間のコミュニケーションの道具という要求記述の持つ重要な役割が十分はたされない場合が多かった。

このような問題を解決するため、人間が日常使用していて理解しやすい自然言語に近い形で記述された要求仕様を、計算機で解釈したり、記述自体を支援したりするシステムの研究が行われている^{9, 10, 11)}。要求記述の解析は、機械翻訳のようにある程度の誤りは許容して実用に供するというようなことは許されず、全文が正確に解釈されるものでなければならない。そのためには、解釈に必要な知識を全て用意するか、解釈上曖昧性の生じない程度に記述に制限を設ける必要がある。これらは二者択一ではなく、一方を強化すれば他方は緩和できる。しかし、知識を用意することには自ら限界があり、また制限を強めれば読みにくく、書きにくくなる弊害が生じる。これまでの研究では、適切な制限という観点から考察したものは少なく、もっぱら知識の形式や知識構築可能な範囲等の観点から論じられ、その結果として解析

可能な文の制限が漠然と生じる場合が多かった^{10, 11)}。そのため、表現された文章に対する解析結果を保証することができず、常に人間の判断や支援を必要とするものであった。また、解析には分野知識を必要とするものが多く、要求記述を解析するために、予めその要求記述に関する知識を全て用意しなければならないという、ある種の矛盾を、含んだものであった。

本稿では、自然言語に近い形で記述された要求仕様を、人間の力を借りることなく、また分野に依存しない一般知識のみを使って、全文解釈することを前提に、そのための適切で明確な制限を設けた要求記述用制限自然言語を提案する。制限を設けるに当たっての条件は次のようなものである。

- ①分野知識なしで解析可能
- ②要求記述に必要な記述内容は全て記述可能
- ③できるだけ記述が稚拙にならず、冗長、煩雑にならない
- ④省略や言い換え、順序変更等の表層上の自由度を持つ
- ⑤単文のみではなく、文の連結や箇条書が可能
- ⑥図式モデルと対応できる
- ⑦図や表との対応、連結が可能

次章より本制限自然言語の概要と、その解析方式について述べる。

2. 要求記述の構成と記述内容のパタン化

標準的なソフトウェア要求仕様書の構成、すなわち記述されるべき順序と内容は、図1に示すようなものが良いとされている¹²⁾。これは、これまでに多くの人々によってなされた要求仕様記述に関する経験から、人間にとって最も分かりやすく、かつ間違いが少ない要求仕様書の構成として結論づけられたものである。要求仕様の内容を計算機で検証しようとする場合も、できれば人間に分かりやすい図1のような構成で記述されたものを、直接検証できることが望ましい。しかし、この構成では各項目内で論理が閉じている場合は少なく、また文章の表す内容（文章の働き）も項目内で一

様でなく、さまざまな文章が同時に一つの項目内に書かれる場合が多い。このような記述を計算機で検証するためには、図1に示した構造を手がかり

1. 序言
1.1 目的
1.2 範囲
1.3 用語定義, 同義語, 省略語
1.4 参照
1.5 関連事項
2. 一般事項
2.1 製品の用途
2.2 製品の機能
2.3 使用者の特性
2.4 前提事項, 他への依存事項
3. 詳細仕様
3.1 機能的要求
3.1.1 機能1
3.1.1.1 序言
3.1.1.2 入力
3.1.1.3 処理
3.1.1.4 出力
3.1.2 機能2
3.2 外部とのインタフェース
3.2.1 使用者とのインタフェース
3.2.2 ハードウェアとのインタフェース
3.2.3 ソフトウェアとのインタフェース
3.2.4 通信系とのインタフェース
3.3 性能に関する要求
3.4 設計上の制約
3.4.1 仕様規格
3.4.2 ハードウェアの制約
3.5 必要な性質
3.5.1 機密性
3.5.2 保守性
3.6 環境に関する要求
3.6.1 データベース
3.6.2 運用
3.6.3 現地据え付け, 組み立て, 調整
添付資料
索引

図1 標準的なソフトウェア要求仕様書の構成 (ANSI/IEEE Std.830 1984)

りとすることはできず、検証に適した別の構造ボタンを設定する必要がある。

筆者等は、このような観点から、これまで実際に作成された約200編のソフトウェア要求仕様書を調査し、検証に有効な手がかりとして、文章の表現する内容(文章の働き)によって文章を分類し、その分類に従って記述内容を検証する方式を考案した。

表1に示すように、ソフトウェア要求仕様書の記述内容は、5つの表現の種類(表現ボタン)に分類することができる。また各表現ボタンは、後述するようにそれぞれ特定の構文ボタンに集約することができる。図1の要求仕様書の項目との対応は表1に示す通りであり、この5つの表現ボタンにより図1の全項目の内容を表現することが可能である。表現の種類と、該当する図1の項目が1対1対応でなく、同じ項目が複数箇所に表れるのは、同じ項目の中に文章の表現する内容(文章の働き)の異なるものが同時に記述されている場合のあることを示す。このような表現ボタンを手がかりとすることにより、同じ表現ボタンの記述がどのような位置に記述されていようとも、全体としてその表現ボタンの記述に関する検証を行うことができる。例えば、機能の記述の検証を行うには、種々の項目の中に分散して記述されている機能に関する文を集めて、機能全体に関する論理

表1 表現の種類と内容, 図1の項目との対応

表現の種類	表現の内容(働き)	該当する図1の項目(番号)
目的, 用途記述	開発の目的, 対象, 範囲 システム, サブシステム, 処理の目的, 用途 等	1.1 1.2 2.1
制限, 定義記述	設計上の制約, 前提条件 外部とのインタフェース, 使用者の特性 性能上の制約 必要な性質 環境条件 用語定義 等	1.2 1.3 2.3 2.4 3.2 3.3 3.4 3.5 3.6
機能記述	システム, サブシステム, 処理等の機能 使用者の役割 その他外部とのインタフェースの機能 等	2.2 3.1 3.2
構造記述	システム, サブシステム, 処理等の機能 データの構造 環境の構造 等	1.4 1.5 3.1 3.2 3.6
その他	図, 表, 他の資料, 索引 等	

の完結性を調べる事が可能となる。

3. 表現の内容と構文パターン

図1の構成で記述された要求仕様書の文章から、表1の表現内容の種類別に文章を分類し、記述内容の検証を行うためには、まず文章を解析して、その文章がどの表現パターンに属するものかを判断し、かつ記述内容を解釈する必要がある。自然言語で自由に記述された文章を完全に解析することは現在の所不可能であり、わずかに正解率を高めるだけでも膨大な知識ベースを付加しなければならないことは良く知られている。これは自然言語の持つ曖昧性によるが、用途を限定した場合、その用途に支障をきたさない範囲内で、曖昧性を生じない構文上の制限が設定できれば、限定された知識で完全に解析可能な体系を構成することは可能である。このとき重要なことは、その構文上の制限のため、表現が分かりにくくなったり、稚拙になったり、冗長になったり、表現のために特別な工夫を強いたりするような体系にしないことである。

本稿では、前節で述べた各表現内容に対し、次に述べるような構文上の制限パターンを設定し、その制限に従って記述された文章に対しては、完全に解釈可能であるような文法を与えることにより、要求記述の表現内容別分類および記述内容の解釈を可能にする。

(1) 構文の共通的制限

文章は、名詞（サ変動詞の語幹を含む）と助詞、助動詞、拡張助詞（複合して助詞と同じ働きをするもの：例えば”のとき”、”のように”等）、拡張助動詞（複合して助動詞と同じ働きをするもの：例えば”である”、”となる”等）だけからなる。動詞は、いわゆる和語動詞は一般に意味が曖昧であるので、”名詞+助詞する”からなるサ変動詞のみとする。修飾句は名詞の内容を限定したり、特定したりするものとして、例えば”名詞+の（助詞）”、”名詞+が（助詞）”のような

形の修飾語（複数可）を伴って、名詞句となってもよい。また、各名詞（句）はANDを表す”と”、”および”、あるいは”、” さらにORを表す”または”を使って並列記述することができる。また主語が人間である場合は省略できる。

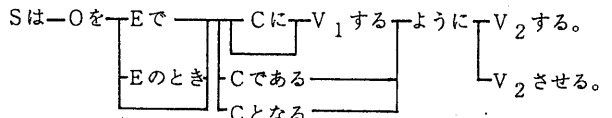
(2) 名詞の役割と記号

以下の構文説明で使用する記号を、名詞の役割に従って定義する。

- | | |
|---------------|-------------|
| S : 主語（動作主） | V : サ変動詞の語幹 |
| O : 目的語（処理対象） | C : 補語（属性） |
| E : 条件（状態、環境） | T : 制御信号 |
| I : 入力元 | D : 出力先 |
| N : 構成要素 | |

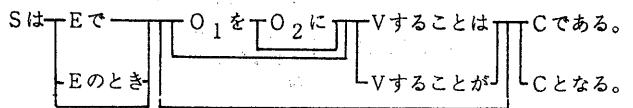
(3) 目的、用途表現に対する構文パターン

目的、用途を表現する文章は、ある目的のために対象物を変化させたり、使用することを表現するものであり、次の構文パターンで表現できる。



(4) 制限、定義表現に対する構文パターン

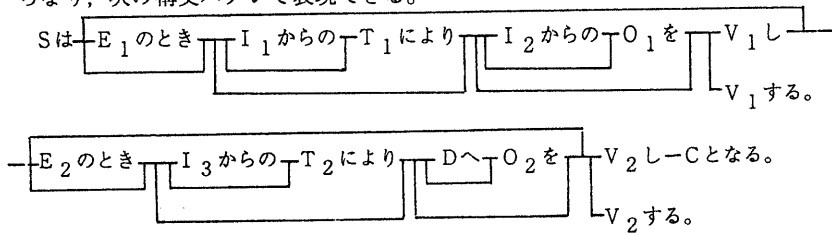
制限条件を述べたり、事物の定義を行う文章は、ある条件下での事実や制限、動作上の事実や制限を述べる文章であり、次の構文パターンで表現できる。



(5) 機能表現に対する構文パターン

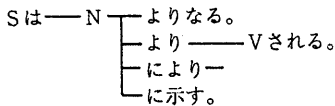
システムや処理プログラム等の機能を表す表現は、動作と、動作の対象となるものの流れ（入力元と出力先）、動作のきっかけとなる制御信号、および動作の前後における動作主の状態変化等か

らなり、次の構文パターンで表現できる。



(6) 構造表現に対する構文パターン

構造表現は、システムや処理プログラム、データ等の構成要素を示すもので、次の構文パターンで表現できる。



(7) 文の連結

(3) ~ (6) の各構文は、接続詞によって連結することができる。接続詞の中には複数の意味を持つものがあるので、ここでは曖昧性を除去するため、次のように一つの接続詞は一つの意味に限定して使用する。

- (文) ために, (文) : 目的
- (文) ので, (文) : 理由
- (文) あるいは, (文) : 排他
- (文) さらに, (文) : AND
- (文) また, (文) : AND
- (文) , (文) : AND
- (文) または, (文) : OR

4. 要求記述の例

次に、前章で述べた構文パターンに従って記述した要求記述の例について述べる。

(1) 目的, 用途表現の例

”システムAのユーザインタフェースをマルチウインドウでマウスによる作業が可能となるように変更する。また、操作が容易となるように改善する。

この例は、2つの目的文が並列の接続詞”また”で連結されている。前文、後文共主語が省略されているが、これは主語となるべき語が人間である場合、省略することができるためである。前文において目的となる名詞句は”ユーザインタフェース”を”システムAの”が修飾し、限定している。また”可能”という名詞を”マウスによる作業が”という句が修飾し、可能の内容を特定している。これらの修飾句+名詞からなる名詞句はそのまま全体を名詞句として認識する。後文では、目的句が省略されているが、接続詞で連結された複数文のうち、第1文以外は、第1文で記述された句の内容を継承する限りその句は省略できる。”操作が”は容易の内容を特定する修飾句である。文中下線部がこの構文パターンの予約語として識別される。

(2) 制限, 定義表現の例

”システムBは、一つのプロセスで全ての機能を実行することが可能であるので、一台のワークステーションで同時に複数のシステムを実行することは不可能である。”

この文は、理由の接続詞”ので”で連結された2つの制限表現文からなっている。下線のついた予約語に挟まれた各名詞句は必要に応じて修飾語で限定または特定されている。

後文の主語は前文の主語と同じであるため省略されている。

(3) 機能表現の例

”マンション監視システムは、センサからの室内状態データを評価し、監視室モニタ画面へ室内状態データと評価結果を表示する。また、異常状態

のとき、警報情報を監視室警報装置へ出力する。

この例も、並列の接続詞”また、”で連結された2文よりなる。”表示する”の目的句は”室内状態データ”と”評価結果”の2つの目的語が、接続詞”と”で並列されている。

(4) 構造の表現例

”処理Aは、処理 a、処理 b、処理 cよりなる。”

”システムPは、p、q、rより構成される。”

”システムPは、次より構成される。

p、

q、

r

”

”データXの構成は、図 xに示す。”

これら4つの例は、いずれも構造を表す表現の例である。各名詞句が修飾語を伴ってもよいのは、他の表現と同じであり、並列は”、”でも”と”および”の接続詞でもよい。

以上の例から推察されるように、構文パターンを制限することによって懸念される文章の書きにくさ、読みにくさ、あるいは稚拙さは、ほとんど問題とならないものとなり、自然な文章として表現理解できる。このことの検証のため、既存の要求仕様書の本記述言語に書き改める実験を数人の技術者で行った結果、図表を除いた文章部分は、特別の努力を要することなく、全て書き換えることができることが実証された。図表に関しては現在対象外としており、今後の課題と考えている。

5. 構文解析

構文解析の処理の流れを図2に示す。

5. 1 構文パターンの識別

入力文は、通常の形態素解析で単語に分解され、品詞、活用等の解析が行われた後、構文パターン識別処理が行われる。各構文パターンは、前述のように、必要に応じて語句の省略が可能となるように設定されているが、与えられた文がどの構造パターンに該当するかを識別するために最小限必要な語句は、キーワードとして省略不可としている。構

入力文

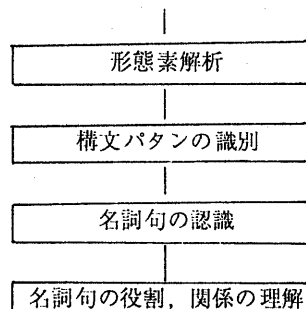


図2 構文解析の流れ

文解析では、まずこのキーワードをもとに構文パターンの識別を行う。各構文パターンのキーワードは次のようなものである。

- (1) 目的、用途表現
～を～ように～
- (2) 制限、定義表現
～である。
～となる。
- (3) 機能表現
～する。
～し、～となる。
- (4) 構造表現
～よりなる。
～(に)より～される。
～に示す。

本稿で取り扱う範囲では、予約語以外は内容に立ち入らず、全て名詞句と判断するため、形態素解析は通常のよりも簡略化されたものでよいが、将来名詞句内の解析を行ったり、機械翻訳を行ったりすることを考慮し、形態素解析は通常のものを使用する。

5. 2 名詞句の認識

構文パターンが決定されると、その構文パターンに対して決定されている予約語の認識が行われる。

助詞”は”や拡張助詞”からの”等がそれに相当する。先頭から予約語までと、予約語と予約語に挟まれた部分とが名詞句である。名詞句の中がひらがなで記述されている場合、形態素解析において部分的に予約語と同じ単語、品詞と誤認されることがあるが、その場合は識別が不可能となる。そのような危険性がある場合は、その部分を仮名名で記述するか、下線を付ける。形態素解析で未知語とされた部分は、全て名詞句と判断される。

5. 3名詞句の役割, 関係の認識

認識された名詞句が、文中でどのような役割を演じているかは、その名詞句に後続する予約語と、その順序によって一意的に決定することができる。例えば、機能表現においては、

” SはE₁のときI₁からのT₁によりI₂からのO₁をV₁し、E₂のときI₃からのT₂によりDへO₂をV₂しCとなる。

という記述は、次のように解析される。

S : 主語 (動作主)
 E₁ : V₁の条件 (状態)
 I₁ : T₁の入力元
 T₁ : V₁の起動信号
 I₂ : O₁の入力元
 O₁ : V₁の対象
 V₁ : 動作内容
 E₂ : V₂, Cの条件 (状態)
 I₃ : T₂の入力元
 T₂ : V₂の起動信号
 D : O₂の出力先
 O₂ : V₂の対象
 V₂ : 動作内容
 C : 条件 (状態) 変化

これらの解析は、記述に途中省略があっても可能であり、名詞句内の内容 (意味) が不明であっても可能である。したがって解析のために単語の内容や意味を照合するための特別な分野知識を必

要とせず、構文パタンの知識のみで解析可能である。

6. まとめ

本稿では、ソフトウェアの要求仕様を計算機によって検証する方法として、設計者が書きにくく、発注者や使用者が理解しにくい形式的な要求記述言語を設定するのではなく、人間が書きやすく理解しやすい自然言語に近い形で記述したものを計算機によって検証することを目的とした要求記述用制限自然言語を提案した。制限自然言語は、広く一般の文章を対象に考えると実用上問題が生じる場合が多いが、用途を限定すれば十分に実用に耐え得るものを構成できる可能性があると考えられる。本稿で提案した制限自然言語はソフトウェアの要求定義という目的に限定すれば、十分有効であり、書きやすさ、表現上の自然さも満足できる水準にあると考えている。解析に関しては、文中における各語句の役割、関係が一意的に決定できることをもって解析可能とした。今後は、この解析結果を用いて記述内容の検証、すなわち記述内容の誤り、矛盾、抜け等の解析を行う方式の開発を進める予定である。尚本制限自然言語は、構文パターンが決定されており、また名詞句内の修飾の係り受け関係も名詞句で閉じているために、文構造が細部に亘って一意に決定できる。このため機械翻訳等の自然言語処理を容易に行うことができる。今後この方面への展開も行う予定である。

参考文献

- 1) DeMarco, T. : Structured Analysis and System Specification, Yourdon Press(1978)
- 2) Ross, D.T. and Schoman Jr., K.E. : Structured Analysis for Requirement Definition, IEEE Trans. on Software Engineering, Vol.3, No.1, pp6-15(1977)
- 3) Jackson, M.A. : Principles of Program Design, Academic Press, 1975

- 4) Stay, P.J. : HIPO and Integrated Program Design,
IBM Syst. J. 15,2(1976)
- 5) 花田 収悦 : プログラム設計法, 企画センター
1983
- 6) Dasarathy, B. : Timing Constraints of Real-Time
Systems: Constructs for Expressing Them, Methods
of Validation Them, IEEE Trans. on Software
Engineering, Vol.11, No. 4, pp80-86(1985)
- 7) Alford, M.W. : A Requirements Engineering
Methodology for Real-Time Processing
Requirements, IEEE Trans. on Software
Engineering, Vol.3, No.1, pp60-69(1977)
- 8) Peterson, J.L. : Petri Net Theory and the Modeling
of Systems, Prentice Hall(1981)
- 9) 佐伯元司, 米崎直樹, 榎本 肇 : 自然言語の語
彙分割による形式的仕様記述, 情報処理学会論
文誌, Vol.25, No. 2, pp.204-214(1984)
- 10) 大西 淳, 阿草清滋, 大野 豊 : 要求フレーム
に基づくソフトウェア要求仕様化技法, 情報処
理学会論文誌, Vol.31, No. 2, pp.175-181(1990)
- 11) 日野克重 : 日常的世界観に基づく言語NAIVEに
よる実行可能な仕様記述, 情報処理学会研究会
報告, SE83-7, pp49-56(1992)
- 12) 松本吉弘 : ソフトウェア工学, 丸善株式会社
(1992)