

## ソフトウェア要求仕様記述の推敲法についての一考察

藤原 仁\* 林 貴之\* 屋部明彦\* 武内 惇\* 藤本 洋\* 椎野 努\*\*

\* 日本大学

\*\* 三重大学

あらまし 要求仕様書は、これから開発するソフトウェアの機能や実現条件などに誤りがないことをユーザに確認を依頼するために使用されるため、ユーザに理解しやすい自然言語(日本語)で記述されることが多い。日本語が持つ特性や文章表現法が記述者の自由に任せられていることから、記述者(開発者)の意図が正しく読者(ユーザ)に伝わらないという問題がある。この問題を解決するための方法として、記述者(開発者)が自分の意図を正確に読者(ユーザ)に伝えるため、作成した文書を自から推敲することが重要であると考えている。

本稿では、効率よく要求仕様書の推敲を行うために必要な要求仕様記述の評価法および制限日本語で記述された要求仕様記述を効率よく推敲するための方法について述べる。

## A Consideration about Elaboration Method for Software Requirement Description

Jin FUJIWARA\* Takayuki HAYASHI\* Akihiko YABE\* Atsushi TAKEUCHI  
Hiroshi FUJIMOTO\* Tsutomu SHIINO\*\*

\* Nihon University

\*\* Mie University

A software requirement specification is usually described in natural language for user to understand its content. Natural language makes difficult to detect some wrong descriptions by computer program. We propose a method to elaborate software requirement description document: metrics and procedure to evaluate of the document, and evaluation procedure for the document written in controlled Japanese.

## 1. はじめに

ソフトウェア要求仕様書（以下、要求仕様書と呼ぶ）は、コンピュータシステムのソフトウェアが実現すべき機能や実現条件などを記述したものである。要求仕様書は、一般にはコンピュータシステムの開発を要求する者（以下、ユーザと呼ぶ）から提示される機能や実現条件などに関する要件をもとに、コンピュータシステムを開発する者（以下、開発者と呼ぶ）が、不明確な要件についてユーザから聴取して作成されることが多い。

要求仕様書は、開発者からユーザに提示され、これから開発するソフトウェアの機能や実現条件などに誤りがないことをユーザに確認を依頼するために使用されるため、ユーザにも理解しやすい自然言語（日本語）で記述されることが多い。

本来日本語が持つ特性や文章表現法が記述者の自由に任せられていることから、記述者（開発者）の意図が正しく要求者であるユーザに伝わらないという問題がある。このことは、開発者が正しく要件を理解したとの誤解をユーザに持たせることもあり、結果的に、ユーザの意図が要求仕様反映されないことになる。

一方ソフトウェア開発において成果物を推敲することの重要性については既に報告した<sup>[1]</sup>。

一般に記述者の意図を伝える文書を作成する場合、記述者は作成した文書を推敲し、正確に自分の意図が伝わるように文書を改善している。筆者らは要求仕様書に関しても、記述者（開発者）が自分の意図を正確に読者（ユーザ）に伝えるため、作成した文書を自から推敲することが重要であると考えている。

本稿では、効率よく要求仕様書の推敲を行うために必要な、①要求仕様記述の評価法、②制限日本語で記述された要求仕様記述を効率よく推敲するための方法、について検討結果を述べる。

## 2. 要求仕様記述の問題と解決策

### 2. 1 要求仕様記述の問題

現在、ユーザと開発者の両者にとり有用な要求仕様書は、ソフトウェア開発企業内の規約や各種団体で制定された規約で記述すべき項目や目次の構成が明らかにされている<sup>[2]</sup>。

しかしながら、ユーザと開発者の両者にとって理解しやすい要求仕様書を実現するために必要な、文章の記述法に関しては充分検討はなされていない。日本語の特性（曖昧性や多義性、漠然性）や文章の表現法が記述者の自由に任せられていることから、以下の問題が生じる。

①記述された文章では、記述者（開発者）の意図を読者（ユーザや他の開発者）が正確には理解できないことがある。

②作成された文書を使用した記述内容のレビューには多大の時間を要し、誤解や不具合の検出漏れを起こす危険がある。

### 2. 2 推敲による問題の解決法

要求仕様の良否判定はユーザが行うため、要求仕様記述はユーザに理解しやすい自然言語で記述するのが一般的である。しかしながら自然言語で記述された文章は機械的に不具合を検出することが難しく、記述内容の不具合の検出はもっぱらユーザと開発者による記述内容のレビューに頼らざるをえない。そこでユーザと設計者による記述内容のレビューに先立ち、開発者が記述結果を推敲し、①仕様記述の不具合を削減すること、②レビュー参照するための記述の形態や内容に関する情報を抽出しておくことを提案する。

### 2. 3 推敲技術の要件

推敲を効率良く行うための主な要件を以下に示す。

(1) 記述の良否を判断するための評価基準

- ① 良い要求仕様記述の特性
- ② 良否の評価尺度とその計測法

(2) 要求仕様記述法

- ① 規定された記述項目に付いての記述法
- ② 記述内容を理解するために、ユーザは特別な技術を必要としない日本語の記述法
- ③ 記述の表現形式を限定し、一つの対象概念に関する表現は記述者（開発者）によらない記述法

(3) 推敲手順

- ① 記述の良否判定尺度を基に、機械的に記述の不具合の検出と修正を可能とする手順
- ② 記述の不具合を機械的に検出できない範囲を明らかにする手順

3. 要求仕様記述の評価法

3.1 評価尺度

要求仕様記述の評価法は様々であり、視点や要求仕様記述の何を評価するののかによってその尺度は変化する。そのため本稿では、推敲方式の有効性を計るための尺度を提案する。

- ① IEEE の要求仕様記述標準要綱<sup>[2]</sup>の“良い要求仕様記述の特性”における7つの特

表1 良い要求仕様記述の特性

名称	内容
非曖昧性 (Unambiguous)	記述がただ一つだけの解釈しか持たず、他の解釈を持たない
検証可能性 (Verifiable)	記述された要求が人間または計算機によって検証可能である
無矛盾性(Consistent)	記述された要求および要求間に矛盾が無い
完全性(Complete)	要求を表現するための項目がすべて揃っている
変更容易性 (Modifiable)	要求に変更があった場合に速やかに要求仕様記述の必要な部分を変更可能である
追跡可能性 (Traceable)	前後の工程で作成されたドキュメントとの関連を記述するための項目が揃っている
保守運用時適用性 (Usable during the Operation and Maintenance Phase)	保守と運用の段階で必要となる記述が、要求仕様記述に含まれる

性(表1)を採用した。

- ② 7つの特性を、記述のみから評価を行う表現上の尺度である表現尺度(表2)と、要求仕様に含まれるべき項目が有るかどうかの評価を行う内容上の尺度である内容尺度(表3)とに分けた。
- ③ 表現尺度の詳細化言語設計モデル(表4)を用いて表現尺度を詳細化することにより評価基準を明確化した。

以下に表現尺度と内容尺度について説明する。

表4 表現尺度の詳細化言語設計モデル

	内容
概念レベル	対象記述項目を決定する
意味レベル	対象記述項目の意味を表現するモデルを決定する
構文レベル	意味モデルで示された内容を表現する構文を決定する
語彙レベル	使用する語彙を制限する

3.2 表現尺度

表現尺度とは、要求仕様記述の記述法が適切であるかどうかを評価するための尺度であり、具体的には、要求を表現する項目、構文、語彙が適切であるかどうかを評価する。

表現尺度の計測は、尺度違反の数を数えることにより行い、尺度違反が無ければ、それは記述法の上では良い要求仕様書であるといえる。

表2 表現尺度項目

	非曖昧性	検証可能性	無矛盾性
概念レベル	一つ概念に対して複数の目次項目があるとき、当該目次項目の数	一つ目次項目の中で他の目次項目との関連を表わす文が無い目次項目の数	他の目次項目の文を含んでいる目次項目の数
意味レベル	一つモデルが複数の概念を表わしているとき当該モデルに基づいて表現されている文の数	他のモデルとの関連を示すモデル構成要素が欠落した文の数	一つ概念が複数のモデルで表されているときそれぞれのモデルを基に表現されている文の数
構文レベル	・代名詞を含む文の数 ・形容詞、接続詞、副詞が複数の係り先を持つ文の数	他の文との関連を示す記述要素が欠落した文の数	一つ概念が複数の文で表現されているとき当該文の数
語彙レベル	一つの用語が複数の意味で用いられているときの当該用語の数 (多義語の数)	用語が表わす概念の尺度が明確でない用語の数	一つ概念を複数の用語で表しているときの用語の数 (同義語の数)

表3 内容尺度項目

完全性	変更容易性	追跡可能性	保守運用時適用性
IEEE の標準記述項目が全て揃っていること	・目次 ・インデックス ・クロスリファレンスがあること	・前後のドキュメントとの関連を記述する項目があること	・保守 ・運用のための記述項目があること

### 3. 3 内容尺度

内容尺度とは、要求仕様記述に含まれるべき項目が含まれているかどうかを評価するための尺度であり、具体的には、必要な要求を表すための項目、要求仕様記述を参照するために必要な項目、作成されたシステムを運用していく上での必要な項目が記述されているかどうかを評価する。

内容尺度の計測は、項目が含まれているかどうかの判定により行う。

## 4. 制限日本語要求仕様記述の推敲法

### 4. 1 制限日本語の設計<sup>[3]</sup>

種々の要求仕様記述標準に規定されているように要求仕様記述項目は限定されていることから、それぞれの記述法(文章構文)を限定する。さらに文章構文を規定することにより、文章の記述要素の役割を規定することができることから、複数の文章において、文章の記述要素間の関係付けを行う。この関係を基に記述の不具合の検出を行うことができる。

2章で述べた要求仕様記述法の要件を実現するため、自然言語の記述法に制約を加えた、制限日本語を開発した。以下に制限言語の設計手順を述べる。

#### (1) 言語設計モデル

記述対象概念を限定し、その概念を表現するための意味モデル、構文、語彙を決定する。さらに対象概念を表現するための言語を決定することができる。表5に言語モデルの各段階の設計項目を示す。

表5 言語設計モデル

	内容
概念レベル	対象概念を決定する
意味レベル	対象概念の意味を表現するモデルを決定する
構文レベル	意味モデルで示された内容を表現する構文を設定する
語彙レベル	使用する語彙を制限する

#### (2) 概念レベル設計

IEEE の要求仕様記述標準要綱や既存の要

求仕様書を参照し、要求仕様記述の役割に注目し、表現対象概念を①目的、②機能、③構造、④制限および定義の4つに決定する。

### (3) 意味レベルの設計

表現対象概念の意味を表現するための視点を決定する。①目的、②機能、③構造、④制限および定義を表現するための意味モデルをそれぞれP(目的)表現モデル、F(機能)表現モデル、S(構造)表現モデル、C(制限)表現モデルと呼び、図1に示す。

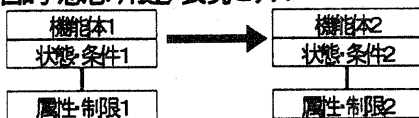
### (4) 構文レベルの設計

意味モデルを表現する構文を決定する。

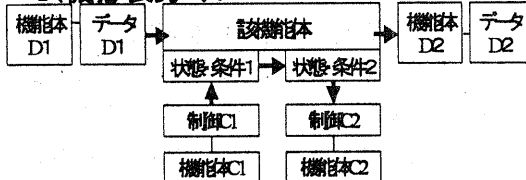
### (5) 語彙レベルの設計

語彙レベルでは、決定された構文に埋め込まれる語彙の制限を決定する。

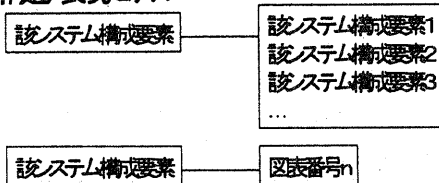
### P(目的意志・用途)表現モデル



### F(機能)表現モデル



### S(構造)表現モデル



### C(制限・定義)表現モデル

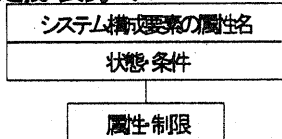


図1. 制限日本語の表現モデル構成図

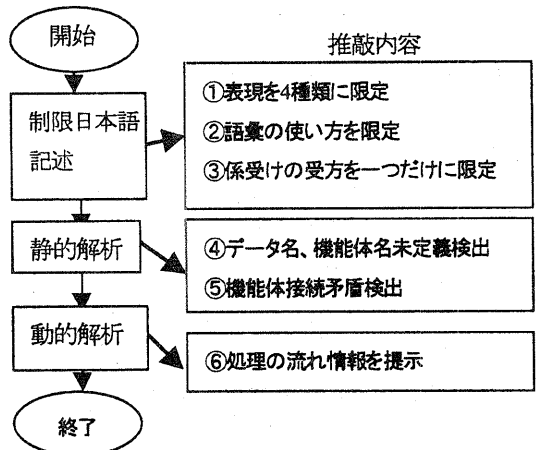


図2. 制限日本語による仕様記述手順と推敲内容

## 4. 2推敲法と記述の良否判定範囲

要求仕様書は図2に示す手順に従って作成する。作成した要求仕様記述に対して、計算機により不具合の検出を行い、人間がそれを修正することによって推敲を行う。不具合の検出は、計算機が自動的に行うものと、計算機による記述の解析結果を参照して人間が行うものに分けることができる。各解析によって検出される不具合と検出法を以下に示す。

### (1) 要求仕様記述

要求仕様を制限日本語の文法を用いて記述することにより、以下の項目を推敲する。

#### ① 表現を4種類に限定

要求仕様を目的、機能、構造、制限の4つに整理して記述するため、一つの要求項目を複数のモデルおよび構文で表現するという矛盾性の問題を解決できる。

#### ② 語彙の使い方の限定

モデル構成要素を表わす用語の役割(機能、データ、制御等)を限定することができること、代名詞や形容詞、副詞の使用を禁止していることから、曖昧性の問題を解決できる。

#### ③ 係り受けで受け方を一つだけに限定

接続詞、副詞は構文規則により使用を制限

しているため、接続詞、副詞の係り受けの曖昧性の問題を解決できる。

## (2) 静的解析<sup>[3]</sup>

各構文中の機能体、制御、データを抽出し、それらに必要な表現文が記述されているか、階層構造が合っているか、また、制御やデータの流れが正しく記述されているかの確認を計算機によって自動的に行うものである。それにより以下の項目を推敲する。

### ④ データ名、機能体名未定義検出

データ名や機能体名は図表を用いた定義の有無を確認する。図表の内容に関しては人間が確認する。これにより、曖昧性の問題、検証性の問題を検出できる。

### ⑤ 機能体接続矛盾検出

入出力データを用いて機能体間の接続関係を確認する。これにより、機能に関しては、他の記述と関連を示す記述が無い問題を解決できる。

## (3) 動的解析<sup>[4]</sup>

要求仕様記述から処理の流れ（業務フロー）を抽出し、処理の時間や同期、排他、分岐の各制御を付加してシミュレーションを行った結果から実行時の動作特性と記述されたシステムの動的な不具合の検出を行うものである。それにより以下の項目を推敲する。

### ⑥ 処理の流れの情報を提示

提示された処理の流れを解析することにより、処理の流れや制御が要求通りであるかを推敲する。また、処理の流れと実際の記述項目を照らし合わせることで、項目の分け方を見直す。このことにより、機能については複数の項目を一つの項目として扱っている問題と項目に関連の無い項目が含まれている問題と項目中の不必要な項目が含まれている問題を解決することが出来る。

推敲によって解決される表現尺度項目を表6に示す。

## 4. 3 制限日本語の課題

制限日本語を用いて、推敲を行えない項目についての推敲を行うための手法を制限日本語に付加することである。

### 4. 3. 1 語彙レベル

#### (1) 非曖昧性

制限日本語で用いられる機能体名、データ名以外の残された用語の定義を行えるようにし、全ての用語に定義が有るかどうかを静的解析によって調べることにより不具合の検出を可能とする。定義を用意すべき語彙は、表現モデルの構成要素の以下の用語である。

①状態・条件

②属性・制限

③制御

表6 推敲によって表現尺度項目

	非曖昧性	検証可能性	無矛盾性
概念レベル	機能：⑥（人間が判断する）により検出	機能：⑥（人間が判断する）により検出	機能：⑥（人間が判断する）により検出
意味レベル	②③により解決	機能：⑤と⑥（人間が判断する）より検出	①により解決
構文レベル	②③により解決	機能：⑤と⑥（人間が判断する）より検出	①により解決
語彙レベル	データ名、機能体名：④により検出	データ名、機能体名：④により検出	

## (2) 検証可能性

曖昧性と同様に、残された用語に対する定義を行えるようにし、定義の有無を静的解析段階で調べる。定義の内容を記述者が評価することにより不具合の検出を可能とする。

## (3) 無矛盾性

評価尺度を満足するためには、用語の持つ概念同士を比べる必要があり、これは用語とその定義の一覧を示し、記述者が用語とその定義を確認することにより不具合の検出を可能とする。

### 4. 3. 2 構文レベル

#### (1) 検証可能性

残された用語の定義を行うことにより、機能体の接続矛盾を検出するのと同じ方法での検出が可能であり、静的解析を用いて不具合の検出を可能とする。

### 4. 3. 3 意味レベル

#### (1) 検証可能性

構文レベルと同様に、残された用語の定義を行うことにより、機能体の接続矛盾を検出するのと同じ方法での検出が可能であり、静的解析を用いて不具合の検出を可能とする。

### 4. 3. 4 概念レベル

各表現尺度の概念レベルの推敲は、要求仕様記述の大部分を占める機能についての推敲が行えれば十分であると考えられるため、機能以外の項目については人間が行うものとする。

## 5. おわりに

自然言語で記述された要求仕様記述の問題点を提示し、記述の良さを計測する表現尺度と内容尺度を用いた推敲による問題解決のための手法を示した。推敲を系統的に行うための手法の実現を目指し、制限日本語による要求仕様記述について、静的解析および動的解析結果を基にした推敲手順と、この時確認可能となる表現尺度と内容尺度の範囲を検討した。

今後は、上記の静的解析および動的解析の範

囲を拡大し、確認可能となる表現尺度と内容尺度の範囲を拡大するための制限日本語の改善を進める予定である。

[1] 武内、藤本：ソフトウェア開発システムのインテリジェント化に関する一考察、情報処理学会、第52回全国大会 1996. 3

[2] ANSI/IEEE Std 830-1984

[3] 武内、他：制限自然言語による要求記述の有効性評価、情報処理学会ソフトウェア工学研究会、1995. 4

[4] 藤原、他：制限日本語で記述されたソフトウェア要求仕様記述の検証支援システム、情報処理学会ソフトウェア工学研究会、1997. 3