機械学習を用いた自然言語仕様書を対象とした VDM++仕様書のクラスとインスタンス変数定義の自動生成手法の提案

菅　健将 a)・片山　徹郎 b)

Proposal of an Automatic Generation Method of Class and Instance Variable Definitions in VDM++ Specification from Natural Language Specification Using Machine Learning

Kensuke SUGA, Tetsuro KATAYAMA

Abstract

The natural language contains ambiguity, so specifications written in natural language can cause software bugs.VDM is one of the formal methods to write the specification without ambiguity. Writing VDM++ specification is difficult because it has a strict syntax and requires writing data types and system invariant conditions. Our laboratory proposed a method for automatically generating VDM++ specifications from natural language specifications using machine learning. However, the existing method is not useful because it only supports type definitions and constant definitions in the VDM++ specification. This paper proposes a method to generate classes and instance variable definitions in the VDM++ specification from natural language specification. The superordinate and subordinate relationships between words are quantified, and then they are used as new parameters for machine learning. It is confirmed that the proposed method has given more useful results than the existing method.

Keywords: Natural language specification, Machine learning, VDM++, Automatic generation

1.　はじめに

社会におけるソフトウェアの重要性はますます高まっており、ソフトウェアのバグは社会に大きな影響を与えている[1)]。ソフトウェアのバグの原因の 1 つに、ソフトウェア開発の上流工程で自然言語を使用していることが挙げられる。自然言語には曖昧さが含まれているため、自然言語で書かれた仕様書をプログラマが読むと、仕様書の作者の意図とは異なる解釈をしてしまうことがある。プログラマが、本来の仕様書の意図とは異なる実装を行った結果、ソフトウェアにバグが混入してしまう可能性がある。

この問題を解決する 1 つの方法として、ソフトウェア開発の上流工程において、形式手法を用いるということが挙げられる。形式手法を用いたソフトウェアの開発は、数理論理学をベースとした形式仕様記述言語によって記述されるため、自然言語の持つ曖昧さを排除した、厳密な仕様を作成することが可能となる。

開発現場向けのライトウェアな形式手法として、VDM（Vienna Development method）が存在する[2)]。また、オブジェクト指向に基づいたモデル化を扱えるように文法を変更した VDM++も存在する[2)]。VDM++のような形式仕様記述言語は、厳密な文法を持ち、かつ、データ型やシステムの不変条件などを書く必要があるため、記述が困難である。従来この作業は、プログラマ個人の経験に依存しており、属人性が高いという問題点がある。

そこで執行氏は、自然言語仕様書内の単語に着目し、機械学習を用いて VDM++仕様書を自動生成する手法を提案した[3,4)]。既存手法は、自然言語で書かれた仕様書から抽出した単語を、VDM++仕様書における型定義と定数定義に分類することを可能としている。しかし、クラスやその他のブロック定義に分類することはできないため、既存手法が生成する VDM++仕様書は、クラスやその他の定義ブロックに対応していない。そのため、既存手法は対応している VDM++の構文が少なく、有用性が低いと言える。

そこで本論文は、既存手法の有用性の向上を目的として、自然言語仕様書から、クラスとインスタンス変数定義に対応した VDM++仕様書を生成するための手法を提案し、既存手法に適用する。なお、本研究は、日本語の文書を対象とする。

2.　研究の準備

提案手法のために、必要となる前提知識を説明する。

a) 工学専攻機械・情報系コース大学院生
b) 情報システム工学科教授

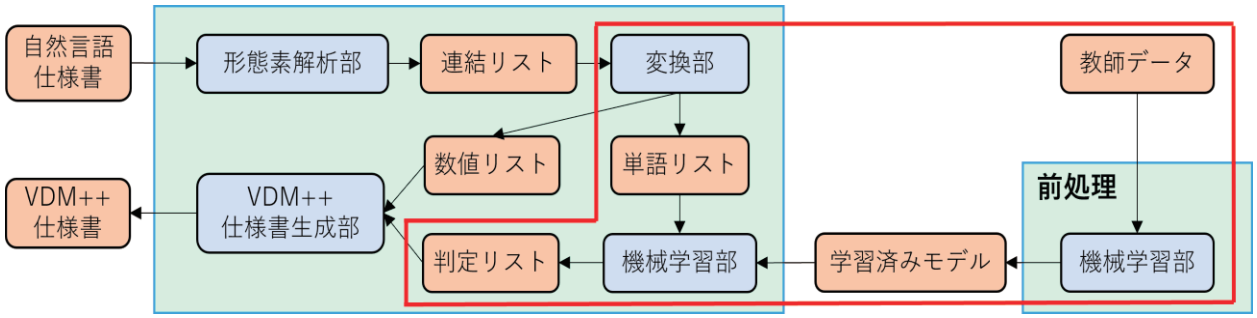

図 1.　本研究の手法の構造

単語	判定結果	TF-IDF値	出現回数	優先値	連結回数
ユーザ認証	1	0.67186	3	2.2	0
教員	1	0.31718	4	1	0
ユーザid	1	0.44688	1	2	0
パスワード	1	0.35217	3	2.2	0
学生登録	1	0.46335	1	2	0
企業	1	0.47214	1	1	7
企業id	1	0.43043	1	1.8	0
システム	0	0.46335	1	2	0
学生	1	0.39363	4	1	2
学生id	1	0.44688	1	1.8	0
利用	0	0.43692	1	1	0
企業担当者	1	0.32638	2	2.5	1
企業担当者id	1	0.46849	1	3.3	0
インターンシップ名	1	0.3263	2	4.2	0

図 2.　既存手法で用いる教師データ

2.1 VDM

形式手法の 1 つに VDM（Vienna Development Method）がある [2]。VDM は、1970 年代に IBM のウィーン研究所にて PL/I コンパイラの正しさを検証するために形式手法として開発された。VDM++は、VDM-SL を基にオブジェクト指向拡張した言語であり、現在 VDM の中では主流である [2]。本研究では VDM++仕様書を自動生成する。VDM++は、VDMTool[5]や VDMJ[6]などの支援ツールが揃っており、他の形式手法に比べ仕様の検証がしやすい。

2.2 WordNet

WordNet は、1980 年代にプリンストン大学で開発された、名詞の類義語、上位語、下位語など、英語の名詞同士の意味的な関係に基づいて作成された辞書である [7]。WordNet を、日本語に対応できるように拡張したものとして、日本語 WordNet がある [8]。日本語 WordNet は、57,238 個の概念と、93,834 の単語を持つ。

本研究では、VDM++におけるクラスの候補となる単語を、自然言語で書かれた仕様書から抽出する際に、日本語 WordNet を使用する。

3.　既存手法

既存手法は、自然言語で書かれた仕様書と、事前に作成した教師データを基に生成した学習済みモデルを入力として、機械学習を用いて自然言語仕様書内の単語を、VDM++仕様書に必要である単語と、必要でない単語に分類する。その後、VDM++仕様書を自動生成する。本研究の手法の構造を、図 1 に示す。既存手法の処理の流れを、以下に示す。

1. 前処理として、事前に作成した教師データを基に、学習済みモデルを生成する。
2. 形態素解析部で、自然言語で書かれた仕様書の各文を形態素解析し、解析後の文を格納した連結リストを生成する。
3. 変換部で、連結リスト内の文の単語に着目し、各単語に対して TFI-DF（Term Frequency Inverse Document Frequency）値、出現回数、優先値、連結回数の 4 つの値を、機械学習に必要なパラメータとして追加し、パラメータを追加した単語を格納した単語リストと、単語の内、数字である単語を格納した数値リストを生成する。
4. 機械学習部で、単語リストと前処理で生成した学習済みモデルから、単語リスト内の各単語の分類を行い、分類結果を格納した判定リストを生成する。
5. VDM++仕様書生成部で、機械学習部で生成した判定リストと、変換部で生成した数値リストから、VDM++仕様書を自動生成する。

既存手法で用いる教師データを、図 2 に示す。既存手法で用いる教師データは、1 列目に単語名、2 列目に判定結果、3 列目以降に変換部で各単語に追加した TFI-DF 値、出現回数、優先値、連結回数を説明変数として持つ。判定結果は 0 か 1 のいずれかであり、0 は 1 列目の単語が VDM++仕様書に必要でない単語であることを表し、1 は 1 列目の単語が VDM++仕様書に必要な単語であることを表す。

既存手法は、自然言語の形態素解析と、機械学習の分類によって VDM++仕様書に必要な単語を抽出し、VDM++仕様書を自動で生成することができる。しかし、既存手法は、自然言語で書かれた仕様書から抽出した単語を、型定義と定数定義に分類できるが、クラスやその他のブロック定義に分類することができないため、既存手法が生成する

単語	判定結果	TF-IDF値	出現回数	優先値	連結回数	概念レベル
ユーザ認証	B	0.67186	3	2.2	0	15.1
教員	C	0.31718	4	1	0	24
ユーザid	B	0.44688	1	2	0	13.6
パスワード	B	0.35217	3	2.2	0	0
学生登録	B	0.46335	1	2	0	31.3
企業	C	0.47214	1	1	7	136.5
企業id	B	0.43043	1	1.8	0	68.2
システム	A	0.46335	1	2	0	323.5
学生	C	0.39363	4	1	2	25
学生id	B	0.44688	1	1.8	0	14.1
利用	A	0.43692	1	1	0	39.1
企業担当者	C	0.32638	2	2.5	1	69
企業担当者id	B	0.46849	1	3.3	0	46
インターンシップ名	B	0.3263	2	4.2	0	18.4

図 3.　　提案手法で用いる教師データ

VDM++仕様書は、クラスやその他の定義ブロックに対応していない。そのため、既存手法は、対応している VDM++の構文が少なく、有用性が低いと言える。

本論文では、既存手法の有用性の向上のため、VDM++仕様書におけるクラスとインスタンス変数定義に対応するための手法を提案し、既存手法に適用する。

4.　提案手法

提案手法は、既存手法で対応している VDM++における型定義と定数定義に加えて、クラスとインスタンス変数定義を記述したVDM++仕様書を自動生成することに着目している。本論文では、図 1 の赤枠内に示す機能および出力を改良する。VDM++におけるクラスの候補となる単語を、自然言語で書かれた仕様書から抽出する際に、WordNet を使用する。提案手法の流れを、以下に示す。

1. 変換部において、WordNet を用いて、解析対象である単語に意味的に関係する単語を、木構造で表現する。さらに、木構造のノード数と根の深さを用いて、本研究で新たに定義する概念レベルを計算する。また、概念レベルをパラメータとして各単語に追加し、単語リストに格納できるように、単語リストを改良する。
2. 機械学習部において、単語リスト内の単語を、VDM++仕様書に必要でない単語、VDM++仕様書に必要であるが、クラスの候補ではない単語、VDM++仕様書に必要であり、かつ、クラスの候補である単語の 3 つに分類し、分類結果を判定リストに格納できるように判定リストを改良する。以降、VDM++仕様書に必要でない単語を WordA、VDM++仕様書に必要であるが、クラスの候補ではない単語を WordB、VDM++仕様書に必要であり、かつ、クラスの候補である単語を WordC と表現する。
3. 機械学習部において、WordA である単語と、WordB である単語の、自然言語で書かれた仕様書内での関係から、WordB の単語の内、インスタンス変数の候補となる単語を抽出し、WordA の単語のいずれかに分類する。
4. VDM++仕様書生成部において、3.で分類した結果と判定リストから、型定義と定数定義に加えて、クラスとインスタンス変数定義を記述したVDM++仕様書を生成する。

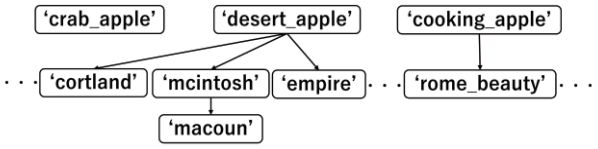

図 4.　　“りんご”を入力した際の木構造の例.

提案手法で用いる教師データを、図3に示す。提案手法で用いる教師データは、図2に示す既存手法の教師データに、提案手法で新たなパラメータとして算出した概念レベルを説明変数として追加する。提案手法の教師データの判定結果はA、B、Cのいずれかであり、Aは1列目の単語がWordAの単語であることを、Bは1列目の単語がWordBであることを、Cは1列目の単語がWordCであることを表す。

本論文では、上記に示す手順の内、手順1と手順2に着目している。手順3と手順4に示す、インスタンス変数の候補となる単語の抽出および抽出した単語のクラスの候補となる単語への分類と、改良したVDM++仕様書の生成は、今後の課題である。

4.1 概念レベルの計算

既存手法は、変換部において、各単語にTFI-DF値、出現回数、優先値、連結回数の4つのパラメータを追加した後、単語リストを出力する。我々は、各単語に概念レベルを新たなパラメータとして追加し、単語リストに格納できるように単語リストを改良する。

概念レベルの計算を行う際に、日本語WordNetを使用し、各単語に意味的に関係する単語を木構造として表現した。単語の木構造の例を、図4に示す。図4は、”りんご”の文字列を入力した際の木構造の例である。日本語WordNetは、日本語での入力に対応しているが、出力する概念を表す単語は英語表記であるため、木構造を構成する単語も英語表記となっている。図4の木構造の場合、りんごの概念を持つ単語を最上位のノードとし、その下位概念である単語を子ノードとして表現する。概念レベルの計算式を、以下に示す。

$$\text{概念レベル} = \sum_{i=1}^{n}\sum_{j=1}^{m}\frac{1}{n} \qquad (1)$$

$$\begin{pmatrix} m\text{: 同じ根の深さのノード数} \\ n\text{: 根の深さ} \end{pmatrix}$$

ユーザ認証:教員は、ユーザIDとパスワードでユーザ認証を行う。学生登録:教員は、システムを利用する学生の情報を登録できる。登録する情報は、学籍番号、氏名とする。企業登録:教員は、インターンシップを提供する企業を登録できる。登録する情報は、企業名であり、登録後、企業IDを発行する。エントリ登録:教員は、インターンシップに参加を希望する学生のエントリを登録することができる。ユーザ認証:企業担当者は、企業担当者IDとパスワードでユーザ認証を行う。インターンシップ登録:企業担当者は、インターンシップ情報を登録することができる。登録する情報は、インターンシップ名、実施日、実施日数とする。ユーザ認証:学生は、学生IDとパスワードでユーザ認証を行う。インターンシップ情報閲覧:学生は、インターンシップ情報を確認することができる。確認する項目は、インターンシップID、インターンシップ名、企業名、実施開始日、実施終了日、実施日数とする。

図 5. 適用例で用いる自然言語で書かれた日本語の仕様書

単語	TF-IDF値	出現回数	優先値	連結回数	概念レベル
ユーザ認証	0.67186	3	2.2	0	15.1
教員	0.31718	4	1	0	24
ユーザid	0.44688	1	2	0	13.6
パスワード	0.35217	3	2.2	0	0
学生登録	0.46335	1	2	0	31.3
企業	0.47214	1	1	7	136.5
企業id	0.43043	1	1.8	0	68.2
システム	0.46335	1	2	0	323.5
学生	0.39363	4	1	2	25
学生id	0.44688	1	1.8	0	14.1
利用	0.43692	1	1	0	39.1
企業担当者	0.32638	2	2.5	1	69
企業担当者id	0.46849	1	3.3	0	46
インターンシップ名	0.3263	2	4.2	0	18.4

図 6. .改良した単語リスト

単語	判定結果	Probability of A	Probability of B	Probability of C
ユーザ認証	B	0.362433	0.489462	0.148103
教員	C	0.239892	0.297224	0.462883
ユーザid	B	0.396748	0.444751	0.158501
パスワード	B	0.303368	0.367307	0.329323
学生登録	B	0.399443	0.448536	0.152019
企業	C	0.287615	0.176724	0.53566
企業id	B	0.39726	0.43816	0.164578
システム	A	0.43733	0.413029	0.14964
学生	C	0.195058	0.204736	0.600204
学生id	B	0.396771	0.444731	0.158497
利用	A	0.351601	0.39905	0.249338
企業担当者	C	0.347326	0.284968	0.367705
企業担当者id	B	0.40069	0.449343	0.14996
インターンシップ名	B	0.340109	0.386199	0.273691

図 7. 改良した判定リスト

提案手法は、各単語を、WordA、WordB、WordCのいずれかに分類するために、概念レベルを計算し、パラメータとして各単語に追加する。

概念レベルの値と、自然言語で書かれた仕様書内の単語には、以下のような特徴が見られた。

- 概念レベルの値が大きすぎる単語(“情報”、”データ”、”物体”など)は、VDM++仕様書に必要でない単語の可能性が高い。
- 概念レベルの値が小さすぎる単語(“番号”、”ID”など)は、VDM++仕様書に必要でない単語である可能性が高い。しかし、クラスの候補となる単語に接続する場合、そのクラスが持つインスタンス変数となる可能性が高い。
- 大きすぎる概念レベルの値と、小さすぎる概念レベルの値の中間程度の概念レベルの値である単語の内、概念レベルの値が大きい単語は、クラスの候補である単語の可能性が高い。
- 上記の項目にあてはまらない単語は、クラスの候補でない単語の可能性が高い。

提案手法は、上記の特徴に基づいて、各単語をWordA、WordB、WordCのいずれかに分類する。

4.2 単語の分類

本研究の手法は、機械学習部において、ロジスティック回帰モデルを用いて単語の分類を行い、判定リストを出力する。

既存手法は、二項ロジスティック回帰分析[9]を用いて、自然言語で書かれた仕様書内の単語を、VDM++仕様書に必要でない単語と、VDM++仕様書に必要な単語のいずれかに分類する。

提案手法は、多項ロジスティック回帰分析[10]を用いて、自然言語内の単語を、WordA、WordB、WordCのいずれかに分類する。これにより、既存手法のVDM++仕様書に必要でない単語と、VDM++仕様書に必要な単語に加え、VDM++仕様書に必要であり、かつ、クラスの候補となる単語の分類が可能となる。

5. 適用例

本論文では、既存手法における、変換部と機械学習部を拡張し、出力である単語リストと判定リストを改良した。提案手法の適用例で用いる、自然言語で書かれた日本語の仕様書を、図 5 に、提案手法の変換部および機械学習部で出力する単語リストと判定リストの一部を、それぞれ図 6 と図 7 に示す。

図 6 に示す提案手法で改良した単語リストから、単語リストに新たなパラメータとして、概念レベルを追加できて

表 1.　VDM++仕様書に必要であり、かつ、
クラスの候補となる単語の分類精度

仕様書	適合率	再現率	F 値
仕様書 A	0.8	0.8	0.8
仕様書 B	0.6	0.86	0.71

いることが確認できる。図 7 に示す提案手法で改良した判定リストから、図 5 の仕様書内の名詞である、’教員’、’企業’、’学生’、’企業担当者’といった名詞を、VDM++仕様書に必要であり、かつ、クラスの候補となる単語として分類できていることが分かる。図 5 から図 7 より、提案手法は、自然言語で書かれた仕様書の単語を、WordA、WordB、WordC のいずれかに分類ができていることを確認できた。

6.　評価

提案手法の有用性を評価するため、インターンシップオンライン提出システム仕様書と ET ロボコン 2020 競技規約 [11] の 2 つの仕様書を用いて、VDM++仕様書に必要であり、かつ、クラスの候補となる単語の分類精度に関する実験を行う。以降、2 つの仕様書を、それぞれ仕様書 A、仕様書 B と表現する。評価では、機械学習部において、仕様書 A を用いて学習済みモデルを構築する。各仕様書から生成した判定リストに対して、F 値を用いて評価する。実験結果を、表 1 に示す。

表1より、提案手法は、VDM++仕様書に必要であり、かつ、クラスの候補となる単語の分類精度に関して、仕様書Aの場合はF値が0.8、仕様書Bの場合はF値が0.71と、いずれも高い精度で分類ができた。よって提案手法は、既存手法のVDM++仕様書に必要でない単語と、VDM++仕様書に必要な単語の分類に加え、VDM++仕様書に必要であり、かつ、クラスの候補となる単語の分類ができるといえる。そのため、提案手法は、既存手法の有用性の向上を達成できたといえる。

7.　関連研究

大森氏らは、自然言語仕様書の品質改善のため、自然言語仕様書と形式モデルの相互変換をサポートし、対応付けを辞書として管理する辞書ツールを開発することによって、自然言語仕様書から VDM++仕様書を生成することを可能とした [12]。辞書ツールに自然言語仕様書内の単語を辞書として登録し、登録した単語を類義語やグループ分けの定義や状態として定義することにより、自然言語仕様書に含まれる曖昧さをなくすことができる。また、入力となる形式的定義と出力する形式的種別を辞書ツールに登録しておくことで、VDM++仕様書として出力することができる。

さらに、辞書ツールを使用した形式仕様の作成から実装までを行うことで形式仕様への変換の手順の改良を提案した [13]。これによって、辞書ツールを使用した仕様書から形式仕様の手順を適用することで発生する問題点と、この手順を適用する際に考慮すべき点が発見できた。

大森氏らが開発した辞書ツールが、自然言語仕様書から VDM++仕様書の作成を支援するツールに対し、本研究の手法は、機械学習を用いて自然言語仕様書から VDM++仕様書を自動生成することができる。さらに、本論文の提案手法は、既存手法の型定義と定数定義に加え、クラスの分類が可能である。

8.　おわりに

本論文では、自然言語の仕様書から VDM++仕様書を自動的に生成する既存手法の有用性の向上を目的として、型定義と定数定義に加え、クラスに対応するための手法を提案し、既存手法に適用した。これは、本論文における提案手法(4 章参照)の、手順 1 と手順 2 に該当する。自然言語で書かれた仕様書を用いた評価実験の結果、提案手法を用いることで、仕様書 A の場合は F 値が 0.8、仕様書 B の場合は F 値が 0.71 と、いずれも高い精度で VDM++仕様書に必要であり、かつ、クラスの候補となる単語を分類することができた。そのため、提案手法は、既存手法の有用性の向上を達成できたといえる。

今後の課題を、以下に示す。

- インスタンス変数定義への対応
  提案手法は、手順 3 における、インスタンス変数定義の候補となる単語の抽出、および、抽出した単語のクラスの候補となる単語への分類ができない。これにより、インスタンス変数定義を記述した VDM++仕様書の生成ができないという問題がある。この問題は、形態素解析を用いて、抽出した単語の内、クラスの候補となる単語と、その他の単語との自然言語で書かれた仕様書内での接続関係などを分析することによって解決できると考える。

- 型定義と定数定義の要素の、クラスの候補となる単語への分類
  提案手法は、既存手法で生成した型定義と定数定義ブロックの要素を、クラスの候補となる単語へ分類することができないため、クラスごとに VDM++仕様書を生成することができない。これにより、厳密な仕様書を作成することができなくなるため、対応する必要があると考える。

- 関数定義と操作定義への対応

提案手法は、VDM++を構成するブロックである関数定義と操作定義には対応していない。この問題については、形態素解析を用いて、自然言語で書かれた仕様書内の単語を解析し、サ変名詞である名詞を抽出することによって、動詞となりうる単語を抽出できる。さらに、抽出した動詞の候補となる単語と、仕様書内の他の単語との関係を分析し、分析した情報を新たなパラメータとして機械学習に適用することによって、対応できると考える。

参考文献

1) Evan Marcus , Hal Stern: Blueprints for High Availability, Wiley Publishing, 2003.
2) International Organization for Standardization: ISO/IEC 13817-1:1996, Information technology - Programming languages, their environments and system software interfaces –Vienna Development Method – Specification Language - Part 1: Base language, 1996.
3) Tetsuro Katayama and Yasuhiro Shigyo et al: Proposal of an Algorithm to Generate VDM++ Specification Based on its Grammar by Using Word Lists Extracted from the Natural Language Specification, Journal of Robotics, Networking and Artificial Life, vol7(3), pp. 165-169, 2020.
4) Yasuhiro Shigyo and Tetsuro Katayama: Proposal of an Approach to Generate VDM++ Specifications from Natural Language Specification by Machine Learning, 2020 IEEE 9th Global Conference on Consumer Electronics (GCCE), pp. 292-296, 2020.
5) VDMTools, http://fmvdm.org/vdmtools/index.html, (Accessed: 2022-2-12).
6) VDMJ, https://github.com/nickbattle/vdmj, (Accessed: 2022-2-12).
7) WordNet, https://wordnet.princeton.edu/ (Accessed 2022-2-12).
8) 日本語 WordNet, http://compling.hss.ntu.edu.sg/wnja/ (Accessed 2022-2-12).
9) Microsoft: 2 クラスロジスティック回帰コンポーネント, https://docs.microsoft.com/ja-jp/azure/machine-learning/component-reference/two-class-logistic-regression(Accessed 2022-2-12).
10) Microsoft: マルチクラスロジスティックコンポーネント , https://docs.microsoft.com/ja-jp/azure/machine-learning/component-reference/multiclass-logistic-regression (Accessed 2022-2-12).
11) ET ロボコン 2020 委員会: ET ロボコ 2020 シュミレーター競技規約, https://docs.etrobo.jp/rules/2020/ETRC2020_rules(sim)_1.0.1.pdf (Accessed 2022-2-12).
12) 大森洋一, 荒木啓二郎: 自然言語による仕様記述の形式モデルへの変換を利用した品質向上に向けて, 情報処理学会論文誌プログラミング(PRO), vol.3, no.5, pp.18-28, 2010.
13) 井上心太, 大森洋一, 荒木啓二郎: ツールを使用した形式仕様の事例研究, 情報処理学会研究報告ソフトウェア工学(SE), vol.2012-SE-175, no.8, pp.1-8, 2012.