SUGISAKA MASANORI

Journal of Advances in Artificial Life Robotics
Vol. 2(4); March (2022), pp. 189-194
ON LINE ISSN 2435-8061; ISSN-L 2435-8061
https://alife-robotics.org/jallr.html

Research Article

# Proposal of a Method to Generate Classes and Instance Variable Definitions in VDM++ Specification by Using Machine Learning

Kensuke Suga[1], Tetsuro Katayama[1], Yoshihiro Kita[2], Hisaaki Yamaba[1], Kentaro Aburada[1], Naonobu Okazaki[1]
[1]*Department of Computer Science and Systems Engineering, Faculty of Engineering, University of Miyazaki, 1-1 Gakuen-kibanadai nishi, Miyazaki, 889-2192 Japan*
[2]*Department of Information Security, Faculty of Information Systems, Siebold Campus, University of Nagasaki, 1-1-1 Manabino, Nagayo-cho, Nishi-Sonogi-gun, Nagasaki, 851-2195 Japan*

ARTICLE INFO

*Article History*
Received 25 November 2021
Accepted 28 March 2022

*Keywords*
Natural language specification
Machine learning
VDM++ specification
Automatic generation

ABSTRACT
Writing VDM++ specifications is difficult. The existing method can automatically generate only type and constant definitions in VDM++ specification from natural language specification by using machine learning. This paper proposes a method to generate classes and instance variable definitions in VDM++ specification from natural language specification to improve the usefulness of the existing method. From the evaluation experiment by using F-values, it has been confirmed that the proposed method can improve the usefulness of the existing method.



## 1. Introduction

The importance of software in society is increasing, and software bugs have a huge impact on our society. One of the causes of software bugs is the use of natural language in the upstream process of software development. Because natural language contains ambiguity, programmers may interpret the specifications to differ from the intent of the specification writers. As a result, the software contains bugs. One way to solve this problem is to design software using formal methods in the upstream process. The development of software using formal methods is written by a formal specification description language based on mathematical logic. This allows writing specifications without the ambiguity of natural language.

VDM(Vienna Development Method) [1] is one of the formal methods. VDM++ [1] is the extension of VDM's syntax to handle object-oriented type-based modeling. A formal specification description language such as VDM++ is difficult to write because it has a strict syntax and requires writing data types and system invariant conditions. Traditionally, this task has depended on the experience of each programmer and has the problem of high dependency. For this reason, we proposed a method for automatically generating VDM++ specifications using machine learning by focusing on words in natural language specifications [2], [3]. The existing method can classify words that are extracted from natural language specifications, into type definitions and constant definitions in VDM++ specifications. However, the existing method is unable to classify words into classes or other block definitions, so the generated VDM++ specification can only output type definitions and constant definitions. Therefore, the existing method is less useful.

In this paper, we propose a method to generate classes and instance variable definitions in VDM++ specification by using machine learning and apply it to the existing method in order to improve its usefulness. Here, this study focuses on specifications written in Japanese language.

*Corresponding author's E-mail: suga@earth.cs.miyazaki-u.ac.jp, kat@cs.miyazaki-u.ac.jp, kita@sun.ac.jp, yamaba@cs.miyazaki-u.ac.jp, aburada@cs.miyazaki-u.ac.jp, oka@cs.miyazaki-u.ac.jp*

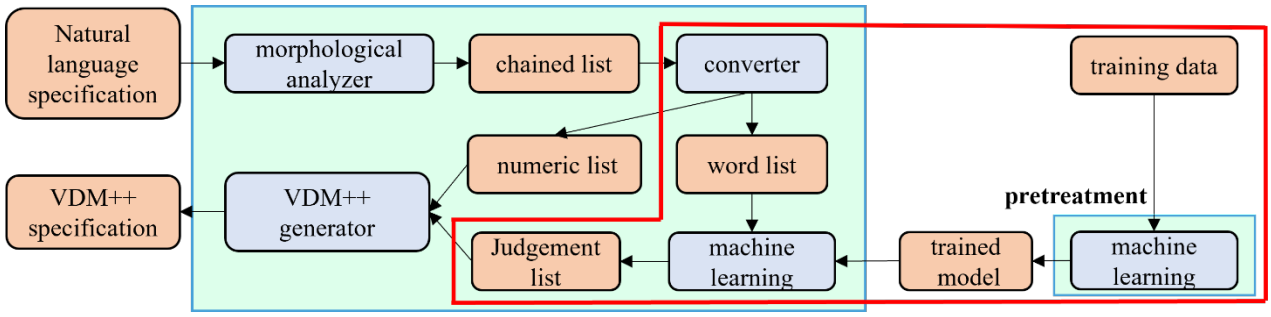


Fig. 1. Flow of the method in this research

| Word | Judgment result | TF-IDF value | Number of occurances | Preferred value | Number of connections | Concept lebel |
|---|---|---|---|---|---|---|
| 教員(teacher) | Word C | 0.31718 | 4 | 1 | 0 | 24 |
| パスワード(password) | Word B | 0.35217 | 3 | 2.2 | 0 | 0 |
| 企業(company) | Word C | 0.47214 | 1 | 1 | 7 | 136.5 |
| 企業id(company id) | Word B | 0.43043 | 1 | 1.8 | 0 | 68.2 |
| システム(system) | Word A | 0.46335 | 1 | 2 | 0 | 323.5 |
| 学生(student) | Word C | 0.39363 | 4 | 1 | 2 | 25 |
| 学生id(student id) | Word B | 0.44688 | 1 | 1.8 | 0 | 14.1 |
| 利用(use) | Word A | 0.43692 | 1 | 1 | 0 | 39.1 |

Fig. 2. Part of the training data used in this study

## 2. Existing Method

Fig. 1 shows the flow of the method in this research. The existing method automatically generates a VDM++ specification from a natural language specification and a trained model generated based on training data. The steps of the existing method are shown below.

1. As a pretreatment, a trained model is generated based on training data by machine learning.
2. The morphological analyzer morphologically analyzes each sentence of the natural language specification and generates a chained list that contains each sentence after analysis.
3. The converter focuses on the words of the sentences in the chained list and adds the parameters necessary for machine learning to each word. In addition, it generates a word list that contains the words and a numeric list that contains the words that are numbers.
4. The machine learning part classifies the words in the word list and generates a judgment list containing the results of the classification.
5. VDM++ generator generates a VDM++ specification using the numerical list generated in Step 2 and the words classified in Step 3.

Fig. 2 shows a part of the training data used in this study. The training data used in the existing method has the word name in the first column and the judgment result in the second column. The third and succeeding columns contain explanatory variables added in the converter. The explanatory variables in the existing method are the TF-IDF value, the number of occurrences, the preferred value, and the number of connections. The judgment result in the existing method is a binary value indicating whether the word in the first column is necessary or not in the VDM++ specification.

The training data used in the proposed method has the concept level defined in this study in addition to the explanatory variables in the existing methods. In addition, the result of the judgment for the word in the first column is indicated by Word A, Word B, or Word C. Word A indicates that the word is not necessary for the VDM++ specification, Word B indicates that the word is necessary for the VDM++ specification but is not a candidate for a class, and Word C indicates that the word is a candidate for a class.

The existing method can only classify words extracted from natural language specifications into type definitions and constant definitions. It is less useful because it cannot output classes and other block definitions. In order to improve the usefulness of the existing method, this paper proposes a method to generate classes and instance variable definitions in VDM++ specification by using machine learning and apply it to the existing method. First, we output a word list adding new parameters by extending the converter. Next, the machine learning part classifies the words in

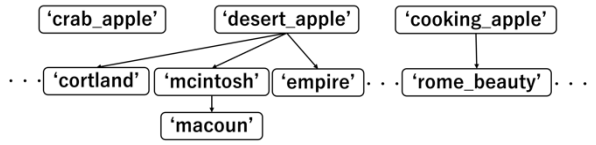


Fig. 3. Example of a tree structure of a words

the word list into Word A, Word B, or Word C. Finally, the machine learning part outputs a judgment list containing the classification results.

## 3. Proposal Method

The proposed method improves the functions and intermediate data circled in red color in Fig. 1. The proposed method supports not only type definitions and constant definitions, but also classes and instance variable definitions, and automatically generates the VDM++ specification. We adopt WordNet [4] to extract candidate words for classes in VDM++ specification from the natural language specification. WordNet is a dictionary created based on the semantic relationships between nouns, such as synonyms, superlatives, and subordinates. The steps of our proposed method are shown below.

1. The converter uses WordNet to generate a tree structure of words that are semantically related to words. In addition, the number of nodes and the root depth of the tree structure are used to calculate the concept level, which is newly defined in this research, and added to each word as a parameter.
2. The machine learning part extracts words and classifies them into Word A, Word B, or Word C.
3. In the machine learning part, the words extracted in Step 2 that are Word B are classified into words that are necessary for the VDM++ specification and are candidates for classes.
4. The machine learning part extracts words that are instance variables based on the relationship between the words classified in Step 3 and the words that are Word C in the natural language specification.

In this paper, we focus on the above steps 1-2. The classification of each word into classes and the dealing with instance variable definitions in VDM++ specification in steps 3-4 are future works.

### 3.1 *Concept Level Calculation*

The existing method outputs a word list after adding four parameters to each word in the converter: TF-IDF value, number of occurrences, priority value, and number of concatenations. We extend the word list by adding a concept level for each word as a new parameter. In calculating the concept level, we use WordNet to generate tree structures of words that are semantically related to the word. Fig 3 shows an example of a tree structure when the string "apple" is entered. Eq.1 shows the formula for calculating the concept level.

$$conceptLevel = \sum_{i=1}^{n}\sum_{j=1}^{m}\frac{1}{n} \quad (1)$$

$$\begin{pmatrix} m: Nodes\ with\ the\ same\ root\ depth \\ n: Depth\ of\ roots \end{pmatrix}$$

In order to classify each word into word A, word B, or word C, the proposed method adds a concept level value as a parameter to each word.

It is found that the concept level value and words in the natural language specifications had the below characteristics.

- Words with too large a concept level value (data, object, etc.) are likely to be words that are not necessary for the VDM++ specification.
- Words with too small a concept level value ('number', 'ID', etc.) are likely to be words that are not necessary for the VDM++ specification. However, if it is connected to a word that is a candidate for a class, it is most likely to be an instance variable that the class has.
- Among the words that have a concept level value between too large and too small, words with a larger concept level value are more likely to be candidates for the class.
- Otherwise, words are more likely not candidates for the class.

Based on the above features, The proposed method classifies each word into word A, word B, or word C.

### 3.2. *Word Classification*

The method in this study uses the logistic regression model to classify words in the machine learning part and outputs a judgment list.

The existing method uses the binomial logistic regression analysis [5] to classify words in specifications written in natural language into words that are not necessary for the VDM++ specification or words that are necessary for the VDM++ specification in the machine learning part.

| |
|---|
| 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。 |
| User authentication: Teachers can authenticate themselves with a user ID and password. Student Registration: Teachers can register the information of students who use the system. The information is to be registered in the student number and name. Company Registration: Teachers can register their companies that offer internships. The information to be registered is the company name. Teachers will be issued a company ID after registering the company name. Entry Registration: Teachers can register the entry of students who wish to participate in the internship. User authentication: Company staff can authenticate themselves with their company ID and password. Internship Registration: Company staff can register their internship information. The information to be registered in the name of the internship, the date of the internship, and the number of days of the internship. User authentication: Students can authenticate themselves with their student ID and password. Viewing Internship information: Students can check the internship information. The items to check are internship ID, internship name, company name, start date, end date, and a number of days of implementation. |

Fig. 4. Japanese specification and its English translation

| Word | TF-IDF value | number of occurances | Preferred value | Number of connections | concept level |
|---|---|---|---|---|---|
| 教員(teacher) | 0.31718 | 4 | 1 | 0 | 24 |
| パスワード(password) | 0.35217 | 3 | 2.2 | 0 | 0 |
| 企業(company) | 0.47214 | 1 | 1 | 7 | 136.5 |
| 企業id(company id) | 0.43043 | 1 | 1.8 | 0 | 68.2 |
| システム(system) | 0.46335 | 1 | 2 | 0 | 323.5 |
| 学生(student) | 0.39363 | 4 | 1 | 2 | 25 |
| 学生id(student id) | 0.44688 | 1 | 1.8 | 0 | 14.1 |
| 利用(use) | 0.43692 | 1 | 1 | 0 | 39.1 |

Fig. 5. Part of the word list

| Word | Judgment result | Probability of Word A | Probability of Word B | Probability of Word C |
|---|---|---|---|---|
| 教員(teacher) | Word C | 0.239892 | 0.297224 | 0.462883 |
| パスワード(password) | Word B | 0.303368 | 0.367307 | 0.329323 |
| 企業(company) | Word C | 0.287615 | 0.176724 | 0.53566 |
| 企業id(company id) | Word B | 0.39726 | 0.43816 | 0.164578 |
| システム(system) | Word A | 0.43733 | 0.413029 | 0.14964 |
| 学生(student) | Word C | 0.195058 | 0.204736 | 0.600204 |
| 学生id(student id) | Word B | 0.396771 | 0.444731 | 0.158497 |
| 利用(use) | Word A | 0.351601 | 0.39905 | 0.249338 |

Fig. 6. Part of the judgment list

The proposed method uses the multinomial logistic regression analysis [6] to classify words in natural language into Word A, Word B, or Word C. This enables the classification of words that are necessary for the VDM++ specification and candidates for the class, in addition to words that are necessary or not for the VDM++ specification in the existing method.

## 4. Application Example

In this paper, we extend the converter and machine learning part of the existing method and improve the output word list and judgment list. The specifications used in the application of the proposed method and its English translations are shown in Fig. 4 and part of the word list and judgment list output by the converter and machine learning part are shown in Fig. 5 and Fig. 6, respectively.

We can see that we have been able to add the concept level as a new parameter to the word list in Fig. 5. The results shown in Fig. 6 show that the nouns in the specification of Fig. 4, such as “teacher”, “company”, and “student”, can be classified as necessary and candidate class words for the VDM++ specification.

From Fig. 4 to Fig. 6, we can confirm that the proposed method is able to classify words in natural language specification properly into Word A, Word B, or Word C.

## 5. Evaluation Experiment

In order to evaluate the improvement of the usefulness of the proposed method, we experiment on the classification accuracy of words using two specifications: the Internship Online Submission System Specification and the ET Robot Contest 2020 competition Rules [7]. From now on, we refer to the two specifications as Specification A and Specification B. In the evaluation experiment, the machine learning part builds a trained model using Specification A. We evaluate the model by using F-values for the judgment lists generated from each specification.

### 5.1. *Classification Accuracy for Words that are Candidates for the Class*

Experimental results on the classification accuracy of words that are candidates for the class are shown in Table 1. Table 1 shows that the proposed method achieves a high F-value in classifying words that are necessary for the VDM++ specification and are candidates for classes, with an F-value of 0.8 for Specification A and an F-value of 0.71 for Specification B. Therefore, the proposed

Table 1. Classification accuracy of words that are candidates for the class

| specification | precision | recall | F-value |
|---|---|---|---|
| Specification A | 0.8 | 0.8 | 0.8 |
| Specification B | 0.6 | 0.86 | 0.71 |

Table 2. Classification accuracy of words that are necessary or not for the VDM++ specification for the proposed method.

| specification | precision | recall | F-value |
|---|---|---|---|
| Specification A | 0.67 | 0.7 | 0.68 |
| Specification B | 0.63 | 0.54 | 0.58 |

Table 3. Classification accuracy of words that are necessary or not for the VDM++ specification for the existing method.

| specification | precision | recall | F-value |
|---|---|---|---|
| Specification A | 0.58 | 0.71 | 0.64 |
| Specification B | 0.48 | 0.54 | 0.51 |

method can classify words that are necessary for the VDM++ specification and are candidates for classes, in addition to the existing method.

### 5.2. *Classification Accuracy for Words that are Candidates for a Class*

Classification accuracy of the proposed and existing method for words that are necessary or not in the VDM++ specification is shown in Tables 2 and 3, respectively. Tables 2 and 3 show that the classification accuracy of the proposed method achieves higher F-values for both Specification A and Specification B than the existing method.

The above shows that the proposed method can classify candidate words for the classes without reducing the accuracy of the existing method in the classification of words that are necessary or not for the VDM++ specification. Therefore, the proposed method can achieve the improvement of the usefulness of the existing method.

## 6. Conclusion

In this paper, we propose a method to generate classes and instance variable definitions in VDM++ specification by using machine learning and apply it to the existing method in order to improve its usefulness. This corresponds to steps 1-2 of the proposed method in chapter 3.

As a result of evaluation experiments using natural language specifications, the proposed method can classify words that are necessary for the VDM++ specification and are candidates for classes with an accuracy of F-value 0.8 for Specification A and F-value 0.71 for Specification B without reducing the classification accuracy of the existing method. Therefore, it can be said that the proposed method achieves the improvement of the usefulness of the existing method.

Our future tasks are shown below.

- Classification of words necessary for the VDM++ specification into extracted classes.
- Dealing with instance variable definitions.

## 7. Reference


1. 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.
2. 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.
3. 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.
4. “Japanese wordnet” http://compling.hss.ntu.edu.sg/wnja/index.en.html (Accessed 2022-3-26)
5. “Two-Class Logistic Regression component” https://docs.microsoft.com/en-us/azure/machine-learning/component-reference/two-class-logistic-regression (Accessed 2022-3-26)
6. “Multiclass Logistic Regression component” https://docs.microsoft.com/en-us/azure/machine-learning/component-reference/multiclass-logistic-regression (Accessed 2022-3-26)

7. “ET Robocon 2020 Simulator Competition Rules” (in Japanese) https://docs.etrobo.jp/rules/2020/ETRC2020_rules(sim)_1.0.1.pdf (Accessed 2022-3-26)

## Authors Introduction

Kensuke Suga

He received a Bachelor's degree in engineering (computer science and systems engineering) from the University of Miyazaki, Japan in 2021. He is currently a Master's student in the Graduate School of Engineering at the University of Miyazaki, Japan. His research interests are natural language processing, machine learning, and formal specification.

Tetsuro Katayama

He received a Ph.D. degree in engineering from Kyushu University, Fukuoka, Japan, in 1996. From 1996 to 2000, he has been a Research Associate at the Graduate School of Information Science, Nara Institute of Science and Technology, Japan. Since 2000 he has been an Associate Professor at the Faculty of Engineering, Miyazaki University, Japan. He is currently a Professor with the Faculty of Engineering, University of Miyazaki, Japan. His research interests include software testing and quality. He is a member of the IPSJ, IEICE, and JSSST.

Yoshihiro Kita

He received a Ph.D. degree in systems engineering from the University of Miyazaki, Japan, in 2011. He is currently an Associate Professor with the Faculty of Information Systems, University of Nagasaki, Japan. His research interests include software testing and biometrics authentication.

Hisaaki Yamaba

He received the B.S. and M.S. degrees in chemical engineering from the Tokyo Institute of Technology, Japan, in 1988 and 1990, respectively, and the Ph D. degree in systems engineering from the University of Miyazaki, Japan in 2011. He is currently an Assistant Professor with the Faculty of Engineering, University of Miyazaki, Japan. His research interests include network security and user authentication. He is a member of SICE and SCEJ.

Kentaro Aburada

He received the B.S., M.S, and Ph.D. degrees in computer science and system engineering from the University of Miyazaki, Japan, in 2003, 2005, and 2009, respectively. He is currently an Associate Professor with the Faculty of Engineering, University of Miyazaki, Japan. His research interests include computer networks and security. He is a member of IPSJ and IEICE.

Naonobu Okazaki

He received his B.S, M.S., and Ph.D. degrees in electrical and communication engineering from Tohoku University, Japan, in 1986, 1988 and 1992, respectively. He joined the Information Technology Research and Development Center, Mitsubishi Electric Corporation in 1991. He is currently a Professor with the Faculty of Engineering, University of Miyazaki since 2002. His research interests include mobile network and network security. He is a member of IPSJ, IEICE and IEEE.