

# EARS2TF: A Tool for Automated Planning Test from Semi-formalized Requirements

Hui Liu<sup>†</sup>, Yunfang Li<sup>†</sup>, Zhi Li<sup>†\*</sup>

<sup>†</sup>School of Computer Science and Engineering, Guangxi Normal University, Guilin, Guangxi, P. R. China

Email: huiliu729@gmail.com, zhili@gxnu.edu.cn

**Abstract**—Software testing is critical to the integrity of the software implementation. The test engineers must design tests under the premise of ensuring the conformance of software system with respect to stakeholder requirements. Generally, requirements documents are large and requirements specifications are represented by NL(natural language) which can be error-prone, so test plans tend to consume more time and effort. In this paper, we propose a tool, EARS2TF, which supports test framework generation from NL requirements specifications. Requirements specifications are represented by a requirements template EARS (Easy Approach to Requirements Syntax), which is easy to use and capable of representing stakeholder requirements with less ambiguity. To save testing costs, we provide a technique and tool that allows requirements to be written, edited, and checked for conformance to existing requirements and EARS syntax, while allowing test engineers to test directly without validating the requirements specification. A demo video of this tool is available at <https://youtu.be/fmk4xSRh40k>.

**Index Terms**—test plan, EARS, natural language requirements, requirements engineering

## I. INTRODUCTION

Software testing is the process of review of software requirements analysis, design specifications and coding throughout the whole software development lifecycle, as well as verifying the quality of the software by measuring and evaluating the quality of the software to meet stakeholder requirements [1] [2]. The test plan serves as a blueprint for testing, describing information needed to perform software product testing such as the test strategy, objectives, etc. and can help determine the information and work needed to verify the quality of the system under test.

The creation of a test plan requires a number of tasks that are currently mostly done manually and require a lot of time and effort. Test engineers need to fully understand and extract test-related information from requirements, most of which are written in natural language (NL) [3] [4] [5]. Natural language contains the inherent characteristic of ambiguity and requirements are often cumbersome, especially in large software projects, and requirements elicitation is challenging as they become complex [6]. The complexity and lack of precision of the requirements causes test planning to become a time-consuming and error-prone process, and the lack of automation in the process also leads to additional maintenance costs.

The expression of requirements has a significant impact on the creation of the testing process. Requirements that are not formally described are usually imprecise and vague, which can hinder the testing process. Creating requirements according to requirements templates, also known as Constrained Natural Language (CNL), can reduce such problems, and EARS [5] is one of the more popular templates in terms of their use in industry and their availability to practitioners [7]. The template specializes the generic requirements syntax into five types to describe requirements, is concise and clear, and demonstrates its usability with industrial examples.

In this paper, we use EARS to describe requirements to ensure their accuracy, and then automatically generate a test framework from EARS requirements to reduce the cost of testing while ensuring consistency between requirements and tests, and to provide the following tool: EARS2TF. This paper is presented as follows: Section II describes the features of EARS2TF. Section III shows the evaluation results of three cases. Section IV discusses related work. We conclude the paper in Section V.

## II. TOOL FEATURES

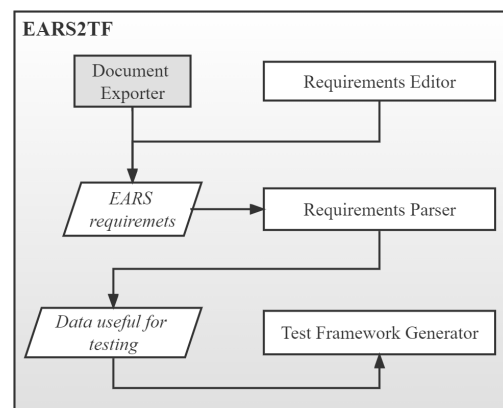


Fig. 1. EARS2TF tool architecture. Grey boxes show third-party components, while white boxes denote EARS2TF components

### A. Tool Overview

EARS2TF is a testing aid tool based on Eclipse Xtext, which consists of three main parts: requirements editor, requirements parser, and test framework generator. Fig. 1 shows

\*corresponding author: zhili@gxnu.edu.cn  
DOI reference number: 10.18293/SEKE2022-179

the tool architecture. We design our own EARS syntax rules, requirements conforming to EARS syntax rules can be imported from existing files with the .ears suffix or obtained by using out requirements editor. EARS requirements can be parsed by the requirements parser to obtain data useful for testing, and then the test framework generator generates a test framework corresponding to the requirements based on the obtained data.

### B. Requirements Editor

In general, requirements are often written with some errors that can lead to testing errors. To make requirements descriptions more accurate, EARS2TF provides a requirements editor that describes the syntactic structure of EARS using the Xtext framework. Our requirements editor enables the written requirements to conform to the EARS syntax and provides a more convenient use experience.

### C. From Requirements to Test Framework

We designed the test framework by referring to some testing standards, such as ISO/IEC 29119 [8], UTP (UML Testing profile) [9], etc. A test framework is a plan for the testing process, which contains the test object, test architecture, etc. In order to narrow the gap between requirements and tests, also to avoid incomplete understanding of requirements leading to missing tests, EARS2TF can convert EARS requirements into test frameworks by designing algorithms to read key information of EARS requirements and then parse them into corresponding test information, and finally the test framework can be displayed in the form of text. Testers can then start testing directly from the test framework, avoiding spending too much time on requirements and missing test information.

## III. EVALUATION

We evaluated EARS2TF by using two examples from a dataset of public requirements document-PURE [10], which containing 44, 192 requirement statements, respectively. We checked and analyzed the results, and almost all of the generated test plan messages are correct except for 13 messages. The reason for these error messages is that there are 6 requirement statements that are untestable assumptions or semantically incorrect.

## IV. RELATED WORK

A lot of existing research has been devoted to automated testing, such as automatic generation of test cases, test scripts, etc from NL requirements or restricted requirements [11] [12] [13] [14]. However, none of these tasks can guarantee correct results, and some require manual addition of files or manual intervention to get the correct results. Some approaches focus on generating test model or test plan. Fischbach et al [6] generate test model from semi-structured requirements by extracting Cause-Effect-Graphs. Lukose et al [15] proposed a tool for guiding a software tester in generating test plans. All these approaches either do not automatically generate test plans or the results obtained cannot plan test. To the best,

EARS2TF is the only approach that automates the test planning from requirements comply with template, and also allows requirements editing.

## V. CONCLUSION AND FUTURE WORK

In this paper, we propose a automated planning test tool from EARS requirements, which includes a requirements editor allowing requirements editing, writing and checking. Future work includes the extension of our tool to check the semantic of requirements and generate test case.

### ACKNOWLEDGMENT

This work is partially supported by the National Natural Science Foundation of China (61862009), Guangxi Natural Science Foundation (2018GXNSFAA281314, Guangxi “Bagui Scholar” Teams for Innovation and Research, the Project of the Guangxi Key Lab of Multi-source Information Mining Security (Director’s grant 19-A-01-02), Innovation Project of GuangXi Graduate Education(JXXYYJSCXXM-2021-005).

### REFERENCES

- [1] J. A. Whittaker, “What is software testing? and why is it so hard?” *IEEE software*, vol. 17, no. 1, pp. 70–79, 2000.
- [2] P. Ammann and J. Offutt, *Introduction to software testing*. Cambridge University Press, 2016.
- [3] C. Ribeiro and D. Berry, “The prevalence and severity of persistent ambiguity in software requirements specifications: Is a special effort needed to find them?” *Science of Computer Programming*, vol. 195, p. 102472, 2020.
- [4] M. Luisa, F. Mariangela, and N. I. Pierluigi, “Market research for requirements analysis using linguistic tools,” *Requirements Engineering*, vol. 9, no. 1, pp. 40–56, 2004.
- [5] A. Mavin, P. Wilkinson, A. Harwood, and M. Novak, “Easy approach to requirements syntax (ears),” in *2009 17th IEEE International Requirements Engineering Conference*. IEEE, 2009, pp. 317–322.
- [6] J. Fischbach, M. Junker, A. Vogelsang, and D. Freudenstein, “Automated generation of test models from semi-structured requirements,” in *2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*. IEEE, 2019, pp. 263–269.
- [7] K. Arora, M. Sabetzadeh, L. Briand, and F. Zimmer, “Automated checking of conformance to requirements templates using natural language processing,” *IEEE transactions on Software Engineering*, vol. 41, no. 10, pp. 944–968, 2015.
- [8] ISO, “Ieee/iso/iec 29119-1-2021,” 2021.
- [9] OMG, “Uml testing profile, version 2.1,” 2019.
- [10] A. Ferrari, G. O. Spagnolo, and S. Gnesi, “Pure: A dataset of public requirements documents,” in *2017 IEEE 25th International Requirements Engineering Conference (RE)*. IEEE, 2017, pp. 502–505.
- [11] C. Wang, F. Pastore, A. Goknil, and L. C. Briand, “Automatic generation of acceptance test cases from use case specifications: an nlp-based approach,” *IEEE Transactions on Software Engineering*, vol. 48, no. 2, pp. 585–616, 2022.
- [12] T. Yue, S. Ali, and M. Zhang, “RTCM: A natural language based, automated, and practical test case generation framework,” *2015 International Symposium on Software Testing and Analysis, ISSA 2015 - Proceedings*, pp. 397–408, 2015.
- [13] J. Fischbach, J. Frattini, A. Vogelsang, D. Mendez, M. Unterkalmsteiner, A. Wehrle, P. R. Henao, P. Yousefi, T. Juricic, J. Radduenz et al., “Automatic creation of acceptance tests by extracting conditionals from requirements: Nlp approach and case study,” *arXiv preprint arXiv:2202.00932*, 2022.
- [14] D. Flemström, T. Gustafsson, and A. Kobetski, “Saga toolbox: Interactive testing of guarded assertions,” in *2017 IEEE International Conference on Software Testing, Verification and Validation (ICST)*. IEEE, 2017, pp. 516–523.
- [15] K. Lukose, S. Agarwal, V. N. Rao, and J. Sreevalsan-Nair, “Design study for creating pathfinder: A visualization tool for generating software test plans using model based testing,” in *VISIGRAPP (3: IVAPP)*, 2018, pp. 289–300.