2021 IEEE/ACM 43rd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)

2021 IEEE/ACM 43rd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion) | 978-1-6654-1219-3/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICSE-Companion52605.2021.00118

Finding Metamorphic Relations for Scientific Software

Xuanyi Lin*, Zedong Peng*, Nan Niu*, Wentao Wang†, and Hui Liu‡
* University of Cincinnati, USA
† Oracle, USA
‡ Beijing Institute of Technology, China
{linx7, pengzd}@mail.uc.edu, nan.niu@uc.edu, wentao.ww.wang@oracle.com, liuhui08@bit.edu.cn

***Abstract*—Metamorphic testing uncovers defects by checking whether a relation holds among multiple software executions. These relations are known as metamorphic relations (MRs). For scientific software operating in a large multi-parameter input space, identifying MRs that determine the simultaneous changes among multiple variables is challenging. In this poster, we propose a fully automatic approach to classifying input and output variables from scientific software's user manual, mining these variables' associations from the user forum to generate MRs, and validating the MRs with existing regression tests. Preliminary results of our end-to-end MR support for the Storm Water Management Model (SWMM) are reported.**

***Index Terms*—Scientific software, metamorphic relation identification, Storm Water Management Model (SWMM)**

I. Introduction

Wiese *et al.* [1] recently summarized the pain points in an online survey of 1,577 scientific software developers, showing that one of the most difficult technical problems was software testing. One survey respondent revealed [1]: "*It's frequently difficult to test scientific software, since you might not even know in advance what the answer should be.*" Given an input, not knowing the expected output of the software under test (SUT) is called the *oracle problem*.

An emerging method of alleviating the oracle problem in scientific software is *metamorphic testing* (MT) [2]. Rather than focusing on the correctness of output from a *single* execution of the SUT, e.g., sine(x), MT checks whether a relation, e.g., sine(x)=sine($\pi - x$), holds among *multiple* executions.

Properties like sine(x)=sine($\pi - x$) are known as *metamorphic relations* (MRs), which are the essence of MT. However, identifying MRs remains manual effort. In this poster, we address the concern by proposing an end-to-end MR identification framework, and report the preliminary results on the Storm Water Management Model (SWMM) developed and maintained by the U.S. Environmental Protection Agency (EPA), showing the feasibility of our approach.

II. End-to-End MR Identification

A. Overview and Assumptions

Our objective is to leverage the community knowledge embedded in scientific software's user forums, as well as the software documentation and numerical regression tests, to automatically construct MRs. Our central conjecture is that "turning a large set of software usage posts into I/O association patterns" is viable to automated extraction of hidden predictive information for MR discovery. The overview of our approach is shown in Figure 1 where the output of one tool is connected with another. We refer to the automation pipeline depicted in Figure 1 as an end-to-end support for MR identification.

- We use machine learning (ML) to locate the variables, and further classify them into input or output variables, from the scientific software's user manual.
- We mine the I/O variables' associations from the scientific software's user forum, and the resulting candidate MRs are in the form of $\{\Delta \mathrm{I}\} \Rightarrow \{\Delta \mathrm{O}\}$ where an association rule's antecedent is the changing input(s) and the rule's consequent is the changing output(s).
- We validate the candidate MRs with existing regression tests, and following Zhang *et al.* [3], if at least 95% of the test inputs support an MR, then the MR is of high quality and passes the validation.

The final output is a ranked list of validated MRs. Our approach assumes no pre-existing MRs, but does assume the availability of user manual, forum, and regression tests of a scientific software system. While such availability is supported by the literature [4], [5], we next present a preliminary study applying our approach to EPA's SWMM.

B. Finding MRs for SWMM

The U.S. EPA's SWMM [6] is a dynamic rainfall-runoff simulation model that computes runoff quantity and quality from primarily urban areas. The development of SWMM began in 1971 and since then the software has undergone several major upgrades. We examined SWMM version 5.1.13 which has 46,291 lines of C code and a 353-page user manual. Our I/O variable classification builds on noun and noun-phrase tagged in the user manual. We further define such ML features as: including "init" or "initial", having summary terms ("final", "ave", "average", and "total"), etc. Based on an answer set prepared by the research team, the accuracy of classifying variables from SWMM's user manual is reported in Table I. Among the five ML algorithms considered, random forest best implements our variable classifier. This result is in line with Ibarguren *et al.*'s experience that random forest almost always has lower classification error in handling uneven data sets [7].

While the I/O variables learned from the user manual are comprehensive, our MR finder of Figure 1 aims to discover

978-1-6654-1219-3/21/$31.00 ©2021 IEEE
DOI 10.1109/ICSE-Companion52605.2021.00118

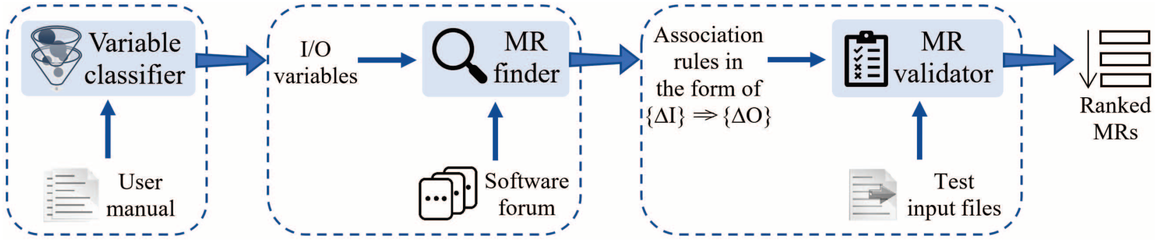

Fig. 1. Our automatic approach is composed of a variable classifier, an MR finder, and an MR validator.

TABLE I
ANSWER SET SIZE AND F-MEASURE COMPARISONS (AVERAGE ACROSS TEN-FOLD VALIDATION)

	non-variable	input (I) variable	output (O) variable	both I and O
|answer set|	13,665	807	164	53
logistic regression	0.883	0.381	0.397	0.591
support vector machine	0.900	0.381	0.425	0.634
decision tree	0.857	0.285	0.253	0.436
random forest	0.900	0.396	0.499	0.652
feedforward neural net	0.866	0.351	0.385	0.637

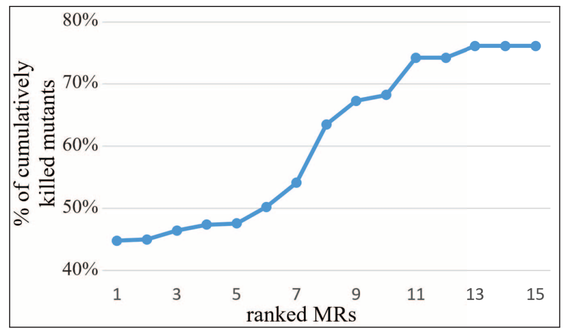

Fig. 2. MR effectiveness measured by mutation scores.

the association rules of the I/O variables from the software forums where large amounts of usage data are continuously collected. OpenSWMM [8], for example, is one of the most prominent forums for EPA's SWMM users. The forum has 26 years of shared knowledge, more than 1,900 contributors, and over 17,000 posts [8]. We treat every single post as a unit to construct one transaction, and leverage the well-known Apriori algorithm to mine $\{\Delta I\} \Rightarrow \{\Delta O\}$ association rules.

To ensure the high quality of identified association-rule candidates, we run a validator and further rank the validated MRs. Peng *et al.* [4] recently reported more than 1,600 unit and regression tests developed and released by SWMM developers. Our MR validator thus uses these regression tests to filter out an association-rule candidate if more than 5% of the test inputs violate the candidate's I/O changes. The validated MRs are ranked by the *confidence* that determines the relative amount of the given consequence across all alternatives for a given antecedent. For example, {increase depression_storage} ⇒ {increase runoff} has a 0.55 confidence, and hence is ranked higher than {decrease surcharge} ⇒ {decrease flow} whose confidence value is 0.54.

Effectiveness of our approach is currently assessed by the mutation score measuring the MRs' fault detection capabilities. We apply only the "traditional" mutation operators (arithmetic operator replacement, relational operator replacement, etc.) [9] in Visual Studio to generate the mutants, each of which contains a single fault. In total, 500 mutants are created for SWMM. Figure 2 shows that the cumulative mutation score could reach over 70% when the top-10 MRs are considered. We attribute the quality of MRs to the validation step where 52% (32/62) SWMM MRs are validated by using the existing regression tests. Note that regression tests themselves suffer from the oracle problem, which is alleviated by MRs.

III. CONCLUDING REMARKS

Manually devising MRs for scientific software is challenging, partly due to the numerous variables in the large input space. In this poster, we have presented an automatic MR identification approach, and showed the approach's applicability via a study on SWMM. Our future work includes improving the MR validation rate with more test inputs and carrying out more empirical studies on other scientific software systems.

REFERENCES

[1] I. Wiese, I. Polato, and G. Pinto, "Naming the pain in developing scientific software," *IEEE Software*, vol. 37, no. 4, pp. 75–82, July/August 2020.

[2] X. Lin, M. Simon, and N. Niu, "Exploratory metamorphic testing for scientific software," *Computing in Science and Engineering*, vol. 22, no. 2, pp. 78–87, March/April 2020.

[3] J. Zhang, J. Chen, D. Hao, Y. Xiong, B. Xie, L. Zhang, and H. Mei, "Search-based inference of polynomial metamorphic relation," in *International Conference on Automated Software Engineering (ASE)*, Västerås, Sweden, September 2014, pp. 701–712.

[4] Z. Peng, X. Lin, and N. Niu, "Unit tests of scientific software: A study on SWMM," in *International Conference on Computational Science (ICCS)*, Amsterdam, The Netherlands, June 2020, pp. 413–427.

[5] X. Lin, M. Simon, and N. Niu, "Scientific software testing goes serverless: Creating and invoking metamorphic functions," *IEEE Software*, vol. 38, no. 1, pp. 61–67, January/February 2021.

[6] United States Environmental Protection Agency, "Storm Water Management Model (SWMM)," https://www.epa.gov/water-research/storm-water-management-model-swmm, 2021, Last accessed: March 2021.

[7] I. Ibarguren, J. M. Pérez, J. Muguerza, I. Gurrutxaga, and O. Arbelaitz, "Coverage-based resampling: Building robust consolidated decision trees," *Knowledge Based Systems*, vol. 79, pp. 51–67, May 2015.

[8] Open SWMM, "SWMM Knowledge Base," https://www.openswmm.org, 2021, Last accessed: March 2021.

[9] H. Liu, F.-C. Kuo, D. Towey, and T. Y. Chen, "How effectively does metamorphic testing alleviate the oracle problem?" *IEEE Transactions on Software Engineering*, vol. 40, no. 1, pp. 4–22, January 2014.