Evolutionary Generation of Metamorphic Relations for Cyber-Physical Systems

Jon Ayerdi
jayerdi@mondragon.edu
Mondragon University
Mondragon, Spain

Valerio Terragni
v.terragni@auckland.ac.nz
University of Auckland
Auckland, New Zealand

Aitor Arrieta
aarrieta@mondragon.edu
Mondragon University
Mondragon, Spain

Paolo Tonella
paolo.tonella@usi.ch
Università della Svizzera italiana (USI)
Lugano, Switzerland

Goiuria Sagardui
gsagardui@mondragon.edu
Mondragon University
Mondragon, Spain

Maite Arratibel
marratibel@orona-group.com
Orona
Hernani, Spain

ABSTRACT

A problem when testing Cyber-Physical Systems (CPS) is the difficulty of determining whether a particular system output or behaviour is correct or not. Metamorphic testing alleviates such a problem by reasoning on the relations expected to hold among multiple executions of the system under test, which are known as Metamorphic Relations (MRs). However, the development of effective MRs is often challenging and requires the involvement of domain experts. This paper summarizes our recent publication: "Generating Metamorphic Relations for Cyber-Physical Systems with Genetic Programming: An Industrial Case Study", presented at ESEC/FSE 2021. In that publication we presented GAssertMRs, the first technique to automatically generate MRs for CPS, leveraging GP to explore the space of candidate solutions. We evaluated GAssertMRs in an industrial case study, outperforming other baselines.

CCS CONCEPTS

• **Computer systems organization → Embedded software**; • **Theory of computation → Assertions**; • **Software and its engineering → Software testing and debugging**; • **Computing methodologies → Genetic programming**.

KEYWORDS

cyber physical systems, metamorphic testing, quality of service, oracle generation, oracle improvement, evolutionary algorithm, genetic programming, mutation testing

ACM Reference Format:
Jon Ayerdi, Valerio Terragni, Aitor Arrieta, Paolo Tonella, Goiuria Sagardui, and Maite Arratibel. 2022. Evolutionary Generation of Metamorphic Relations for Cyber-Physical Systems. In *Genetic and Evolutionary Computation Conference Companion (GECCO '22 Companion), July 9–13, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3520304.3534077

GECCO '22 Companion, July 9–13, 2022, Boston, MA, USA

1 INTRODUCTION

Cyber-Physical Systems (CPSs) integrate digital cyber technologies with physical processes. Most of the functionalities of CPSs are implemented through software. As many other complex software-driven systems, many CPSs suffer from the *oracle problem* [4], which refers to the problem of determining whether a software test execution is correct or not. The oracle problem is exacerbated when automatically generated tests are involved. Indeed, it is infeasible to manually define oracles for many automatically generated tests.

Metamorphic testing [5] alleviates the oracle problem by checking whether multiple test executions fulfill certain necessary properties called Metamorphic Relations (MRs). Instead of verifying the correctness of each individual program execution, metamorphic testing exploits known input and output relations (MRs) that should hold among *multiple* executions of the program [5, 8].

More formally, a test case τ_i is represented as a pair $\langle I_i, O_i \rangle$, where I_i is the input for the system under test, and O_i is the output observed when executing the system under test with I_1. Given two test cases, $\tau_s = \langle I_s, O_s \rangle$ and $\tau_f = \langle I_f, O_f \rangle$, whenever a given input relation r_I holds between the two inputs, a corresponding output relation r_O is expected to hold between the outputs. Hence, a metamorphic relation can be expressed as:

$$r_I(I_s, I_f) \Rightarrow r_O(O_s, O_f)$$

where $\Rightarrow$ is the "implies" Boolean operator. τ_s is called *source* test case, while τ_f is called *follow-up* test case.

When combined with automated test generation, MRs are useful because the same MR can be applied to multiple test cases. Whenever two test cases satisfy the input relation, they also need to satisfy the output relation, otherwise a test failure is reported.

Although metamorphic testing was proposed around 25 years ago [5], recently it is gaining a lot of attention. Large companies (such as Meta - Facebook) are now relying on MRs to test their software systems [1]. An example of the application of metamorphic testing in an industrial CPS was proposed by Ayerdi et al. [2]. They applied metamorphic testing to the traffic manager of a system of elevators developed by Orona [7], one of the leading elevators companies in Europe. For example, one of the inputs of such a CPS is a set of elevators (E), one of the output is the Average Waiting Time (AWT) for the passengers. A MR manually defined with the help of Orona engineers is:

$$E_f = E_s \cup E \Rightarrow AWT_s \geq AWT_f$$

Intuitively, this metamorphic relation specifies that the average waiting time should be reduced if more elevators are added in the input to the traffic manager.

In order to apply metamorphic testing, MRs need to be defined. However, the definition of effective MRs is possible only with in-depth knowledge of the domain and the system under test [8]. As a result, the development and maintenance of effective MRs required a heavy involvement of domain experts, which is a high cost to pay.

2 GASSERTMRS

To avoid the manual cost of defining MRs, we recently proposed at ESEC/FSE 2021 **GAssertMRs** [3], the first technique to automatically generate MRs for CPS. GAssertMRs is built on top of GAssert [9] a technique to automatically improve/generate assertion oracles. Assertion oracles (also known as program assertions) are executable Boolean expressions that predicate on program variable values at specific program points. An assertion oracle should pass (return true) for all correct executions and fail (return false) for all incorrect executions Similarly to assertion oracles, MRs are Boolean expressions that predicate on multiple executions. The difference is that MRs predicate on the inputs and outputs of multiple test cases, while assertion oracles on program variables.

GAssert presents a novel **co-evolutionary algorithm** to effectively explore the space of possible assertions to find one that "improves" an initial assertion. The definition of "improvement" is based on the concept of false positives and false negatives [6].

Definition 1. *A **false positive (FP)** of an oracle is a correct program state in which the assertion fails (but should pass).*

Definition 2. *A **false negative (FN)** of an oracle is an incorrect program state in which the assertion passes (but should fail).*

GAssert leverages evolutionary computation to explore the space of possible assertions to find one with fewer FPs and FNs.

GAssert was demonstrated to be effective in improving assertion oracles [9]. This motivated us to extend the GAssert approach to generate effective metamorphic relations that minimize the number of FPs and FNs. We call this extension GAssertMRs. This extension employs the same oracle generation/improvement algorithm as GAssert, adapting it to the context of metamorphic testing.

The current implementation of GAssertMRs assumes that the inputs and outputs of MRs have numeric types. For cases where not all inputs are numeric, which is also the case of our Orona case study, we defined domain-specific functions to extracts numeric features from the inputs/outputs. When GAssertMRs explores the space of possible MRs it explores numerical expressions in the form $F(O_s, I_s, I_f)$. Every time GAssertMRs evaluates the fitness of an individual it constructs the full Boolean expression $O_f \ [operator] \ F(O_s, I_s, I_f)$, where the output variables (O_s and O_f) and relational operator ($[operator]$) are input parameters.

GAssertMRs takes as inputs samples of *correct* and *incorrect* executions of the system under test. By using genetic programming, GAssertMRs automatically generates MRs that minimize the number of false positives (FPs) over the *correct* samples and the number of false negatives (FNs) over the *incorrect* ones. As such, GAssertMRs obtains MRs that can predict the correctness of an outcome as accurately as possible.

Internally, GAssertMRs explores the huge space of candidate MRs with a co-evolutionary algorithm that formulates the oracle improvement problem as a multi-objective optimization problem with three competing objectives: (i) minimizing the number of FPs, (ii) minimizing the number of FNs, and (iii) minimizing the size of the generated MRs. GAssertMRs evolves in parallel two distinct populations of MRs with two competing fitness functions that reward fewer FPs and fewer FNs, respectively. Both fitness functions consider the remaining objectives only in tie cases. These populations periodically migrate their best individuals to exchange genetic material useful to improve the secondary objectives.

We evaluated GAssertMRs with an industrial case study at Orona. The results show that our approach generates MRs that are comparable with MRs manually defined by domain experts. Moreover, GAssertMRs outperformed a random baseline, showing that evolutionary computation is a valuable methodology to effectively navigate the search space of possible MRs.

3 CONCLUSION

This paper summarizes our recent industrial case study on automatically generating Metamorphic Relations (MRs) for a CPS in an industrial setting. Towards this goal, we proposed GAssertMRs [3], the first technique to automatically generate MRs for CPSs using evolutionary computation. GAssertMRs implements an evolutionary algorithm aiming to minimize the false positive and false negative rates of the generated MRs. An important future work is to evaluate GAssertMRs with other CPSs in order to assess the effectiveness of our co-evolutionary algorithm also in different application domains.

ACKNOWLEDGMENT

This publication is part of a project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 871319. This work has also been partially supported by the H2020 project PRECRIME, funded under the ERC Advanced Grant 2017 Program (ERC Grant Agreement n. 787703).

REFERENCES

[1] John Ahlgren, Maria Eugenia Berezin, Kinga Bojarczuk, Elena Dulskyte, Inna Dvortsova, Johann George, Natalija Gucevska, Mark Harman, Maria Lomeli, Erik Meijer, et al. 2021. Testing Web Enabled Simulation at Scale Using Metamorphic Testing. In *ICSE-SEIP*.

[2] Jon Ayerdi, Sergio Segura, Aitor Arrieta, Goiuria Sagardui, and Maite Arratibel. 2020. QoS-aware Metamorphic Testing: An Elevation Case Study. In *ISSRE*. 104–114.

[3] Jon Ayerdi, Valerio Terragni, Aitor Arrieta, Paolo Tonella, Goiuria Sagardui, and Maite Arratibel. 2021. Generating metamorphic relations for cyber-physical systems with genetic programming: an industrial case study. In *ESEC/FSE*. 1264–1274.

[4] Earl T Barr, Mark Harman, Phil McMinn, Muzammil Shahbaz, and Shin Yoo. 2014. The oracle problem in software testing: A survey. *IEEE TSE* 41, 5 (2014), 507–525.

[5] T. Y. Chen, S. C. Cheung, and S. M. Yiu. 1998. *Metamorphic Testing: A New Approach for Generating Next Test Cases.* Technical Report. HKUST-CS98-01.

[6] Gunel Jahangirova, David Clark, Mark Harman, and Paolo Tonella. 2016. Test oracle assessment and improvement. In *ISSTA*. 247–258.

[7] Orona. 2021. Orona Group. https://www.orona-group.com/.

[8] S. Segura, G. Fraser, A. Sanchez, and A. Ruiz-Cortés. 2016. A Survey on Metamorphic Testing. *IEEE TSE* 42 (2016), 805–824.

[9] Valerio Terragni, Gunel Jahangirova, Paolo Tonella, and Mauro Pezzè. 2020. Evolutionary Improvement of Assertion Oracles. In *ESEC/FSE*. 1178–1189.