Study and definition of project attributes for selection of testing techniques for concurrent software

Student: Italo de Oliveira Santos[1]
Advisor: Simone do Rocio Senger de Souza[1]

Postgraduate Program in Computer Science and Computational Mathematics
Institute of Mathematics and Computer Science – University of São Paulo (USP)

italo.santos@usp.br, srocio@icmc.usp.br

Degree: Master
Entry year in the program: 03/2018
Expected date for final presentation: 02/2020
Date of approval of the thesis/dissertation proposal (qualification): 04/2019
Related Events: SBES, SAST

Abstract. ***[Context:]*** *The choice of testing technique to be adopted in a software testing project persists based on the tester's knowledge and often does not consider all of the testing techniques available in the industry or academia. In a previous project of the research group, a framework was defined to support the systematic selection of concurrent software testing techniques. Therefore, a characterization scheme was proposed and implemented in the SeleCTT tool, which is composed of a set of attributes that considers characteristics of concurrent programs and they are used to calculate which of these attributes are suitable to guide the selection of testing techniques for a particular software project.* ***[Objective:]*** *The selection of the testing technique at each stage of a software's life cycle depends on many factors, such as resources, schedule, cost of the project, among other attributes. This work will extend the previously defined attributes, taking into account other characteristics that may be considered, for example, information from previous similar projects. Considering that the testing techniques are complementary, another goal is to allow a set of testing techniques to be selected and not just one.* ***[Methodology:]*** *To achieve this goal, a systematic mapping study was conducted to identify and analyze papers that represent the current state of the literature about testing techniques selection. Moreover, we surveyed software testing practices carried in Brazil software companies. The survey identified the testing practices in Brazilian industries, for the purpose of knowing and having an overview on the latest testing techniques, tools and metrics used, the challenges faced by testers and the selection testing technique process.* ***[Results and Conclusions:]*** *With this study, it is expected to specify project attributes that can be used to improve the existing recommendation system in the SeleCTT tool and propose ways of combining testing techniques, contributing to industry and academia, and bring insights on the context of testing techniques selection.*

Keywords: Concurrent software testing, Project attributes, Testing techniques selection.

1. Problem Characterization

Software testing is one of the verification, validation, and test (VV& T) activities that have been used to increase the software quality and developed model [Myers et al. 2011]. This activity is becoming increasingly difficult due to the evolution that has been taking place in the field of computing, which has a wide variety of programming languages, operating systems, hardware platforms, and electronic devices that evolving daily. Testing in this complex environment requires a great effort. In this way, concurrent software testing emerges. In addition to the specific challenges of the testing area of sequential programs, it adds new challenges arising from this context as non-deterministic behavior.

The choice of adopting software testing techniques in project relays on the knowledge of the test designer and often does not consider all of the testing techniques available in the industry or academia [Vegas and Basili 2005]. Therefore, [Melo 2018] proposed a body of knowledge that brings together, in an integrated way, information relevant to the decision making process on which testing technique to apply to a particular software project, a framework was developed to support the characterization and systematic selection of concurrent software testing techniques. To build the body of knowledge, the author gathered information about the decision-making process on which testing technique would best apply to a software project. In order to systematize the process of testing technique selection, a characterization scheme was defined that considers the main characteristics of the concurrent programming that influence the software testing activity and calculates the adequacy of these attributes to the attributes of the project in development.

Concerning to allow the community to interact with the proposed framework, in the work of [Moura et al. 2018] the SeleCTT[1] (An infrastructure for Selection of Concurrent Software Testing Techniques) tool was developed a computational infrastructure that allows access to the body of knowledge and automates the selection process. A characterization scheme was built by [Melo 2018] to support decision-making during the process of selecting suitable testing techniques for a software project. The characterization scheme is composed of 21 attributes is shown in Table 1. These attributes are used to calculate the adequacy of the techniques to the test design and consequently the selection of the most appropriate technique, and they were classified into four categories according to the type of information they offer, as described below:

- **Programming model:** describes attributes that represent the characteristics related to the implementation of the software that will be tested by the technique;
- **General testing characteristics:** gathers attributes concerning testing techniques in general like: the type of technique, the approach for generating test data, test level and others;
- **Tool support:** describes information about the tools that support the technique;
- **Concurrent testing characteristics:** describes aspects of specific testing technologies for concurrent programming.

Based on the characteristics of the software project under test and the attributes of the testing techniques presented in the body of knowledge, the SeleCTT tool suggests the best testing technique. However, only one technique is suggested, and the ideal would be to check the possibility of combining more than one technique for improvement of results. Besides, the choice does not take into account information from previous projects.

[1] http://www.labes.icmc.usp.br/ selectt/home

Table 1. Attributes of testing techniques implemented in SeleCTT [Melo 2018].

Categories	Attribute	Example values
Programming model	1. Platform on which the software under test is executed	Linux, Windows, Android
	2. Development paradigm adopted	Shared memory, message passing, hybrid
	3. Programming Language/Runtime Libraries in which the software under test was developed	Java, C/MPI, C/Phtreads, Fortran, Ada, Erlang
General testing characteristics	4. Type of testing technique	Structural, functional, fault-based, model-based
	5. Testing level at which the technique was applied	Unit, integration, system, acceptance, regression
	6. Approach for the test data generation	Execution trace, reachability testing, threading schedules, location pairs
	7. Input expected for a test case	Code, requirements, models, bytecode
	8. Output expected for a test case	Percentage of coverage, execution time, bugs revealed
	9. Software quality characteristic which the technique is able to evaluate	Effectiveness, efficiency, performance, scalability, overhead
	10. Type of empirical study applied for the validation of the testing technique	Case study, controlled experiment
Concurrent testing characteristics	11. Type of analysis used by the technique	Static, dynamic
	12. Mechanism that forces the execution of different synchronization sequences	Reachability testing, noise injection
	13. Type of replay mechanism used for the re-execution of the program during testing	Monitoring, regression scheduler, record-replay, SYN-sequences
	14. Program representation that captures relevant information for testing	State space graph, petri nets, prediction model
	15. Context/Objective of the concurrent software	HPC, distributed applications, mobile, embedded
	16. Technique that treated the state explosion problem	Dynamic partial order reduction, fitness function, preemption bounding
	17. Type of concurrent bugs identified by the technique	Data race, deadlock, atomicity violation, livelock
Tool support	18. Name that identifies the supporting tool	Contest, CHESS, JavaPathFinder
	19. Cost associated with the tool	Academic, shareware, open source
	20. Programming Language/Runtime libraries supported	Java, C/MPI, C/Pthreads, C/OpenMP, Fortran, Ada, Erlang
	21. Execution platform on which the tool operates	Linux, Windows, Android

The main objective of this project is to answer the question: *"What project attributes can guide the selection of testing techniques for concurrent software?"*. This project intends to extend and improve the work developed by [Melo 2018, Moura et al. 2018], and investigate other project attributes that can be used to define a set of concurrent software testing techniques to be applied in a project considering specific attributes from the project.

2. Background

In this section, relevant concepts will be presented concerning the topics of software testing, concurrent programming, and concurrent software testing are addressed.

2.1. Software testing

According to [Myers et al. 2011] in each stage of the test process, the test activity must be performed in four distinct steps: (i) **planning:** in this step the test project plan is made; (ii) **test case design:** in this step the test cases are defined; (iii) **execution:** at that time the previously planned test cases are executed, and the results must be recorded; and (iv) **analysis**: after running the test cases, the generated results are analyzed in order to verify that the software behaves as defined in its specification. Among the steps presented, the one that deserves greater care is the stage of design of test cases. Testing techniques provide standards for the systematic design of test cases that exercise both the internal logic of software components and their input and output domains [Delamaro et al. 2017].

The test case quality is directly associated with the strength of the testing criterion used, and it defines which properties or requirements should be considered in the evaluation of the tests. According to [Ammann and Offutt 2016] testing techniques and criteria are fundamental in the selection of test cases because they can minimize the number of test cases, ensuring that the tests cover specific parts of the code.

2.2. Concurrent programming

A concurrent program has two or more processes or threads that work together to accomplish a task. A process is a running program composed of the program, a data set, and a descriptor record (the area of memory that stores the contents of the registers when the process is not running). A thread (or light process) is an independent control line within the process [Tanenbaum and Van Steen 2007]. Threading provides a mechanism for programmers to divide their programs into more or less independent tasks with the property that when one thread is blocked another thread can be run.

Concurrent programs are not error free, so they must be tested to ensure the quality of the software produced. However, the concurrent software testing is much more complex, considering that in this type of test there are characteristics not found in sequential programs such as: non-determinism, synchronization and communication, which make the test activity in this scenario even more costly and requires more effort of researchers and practitioners to bring solutions to improve testing in this environment.

2.3. Concurrent software testing

The software testing applied to concurrent programs aims to identify errors related to communication, parallelism, and synchronization. Unlike sequential programs, concurrent programs have non-deterministic behavior, a feature that makes the test activity even more complicated. Non-determinism allows different executions of a concurrent program with the same input value to produce different correct outputs [Souza et al. 2017]. Figure 1 exemplifies a scenario of non-determinism, in which we have a concurrent program with three parallel processes. The program is divided into a set of processes (Process 1, Process 2 and Process 3), which communicate through the exchange of messages using the send (S1, S2, and S3) and receive (R1, R2, and R3) primitives. A race condition can be observed between S1, S2, and S3 related to R1, R2, and R3. Each execution (Execution 1, Execution 2 and Execution 3) represents a synchronization sequence that can occur in that program. The objective of the testing activity is to identify all possible synchronization sequences and analyzing the outputs generated.

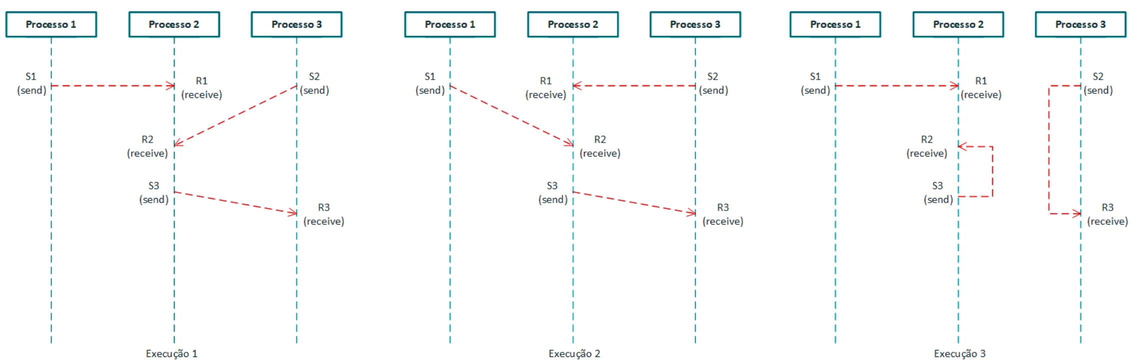

Figure 1. Example of non-determinism between processes.

3. Contributions

This project aims to evolve the attributes defined in the study by [Melo 2018], implement new attributes that can be used to improve the existing recommendation system in the SeleCTT tool. Also, this work aims to consider information from previous projects and the complementary aspect of the testing techniques to support the selection. This project expects to propose the combined selection of testing techniques, allow a set of testing techniques to be selected for the same software project and not just a single technique that SeleCTT tool currently offers. Therefore, contribute to the industry and academia, improve the context of testing techniques selection.

The contributions of this work will be: (i) identify relevant project attributes to aid the selection process; (ii) define and implement the combined selection of testing techniques; (iii) evolve the selection approach in the SeleCTT tool; and (iv) perform experiments to evaluate the selection approach proposed. Through these results, we expect to contribute to the literature identifying the best attributes used to support the process of testing techniques selection.

4. Current Status

This section details the planned activities to achieve the goal of this Master's project. Such activities are divided into the following steps:

1. **Study of the state of the art of research area:** This step involves the development of a systematic mapping study in order to identify, gather and analyze primary studies about testing technique selection. Therefore, the following research questions should be answered:
   (a) What strategies are employed to select testing techniques?
   (b) Which attributes are used for the selection of testing techniques and how they are extracted?
   (c) How are the approaches to select testing techniques evaluated?
2. **Design and conduct a survey about software testing practices:** this step intends to identify the testing practices in Brazilian industries, to know and having an overview on the latest testing techniques and better understanding the process of testing technique selection, which is the central concern of this Master's project.
3. **Study of the SeleCTT tool:** At this stage the SeleCTT tool will be investigated, trying to understand details of its implementation and define how to extend it with the results of this research project.
4. **Definition and implementation of the new attributes of testing techniques selection for concurrent programs:** Based on the results of the previous steps, this step will characterize the new attributes to support the testing technique's selection, implementing them in the SeleCTT tool.
5. **Evaluation of proposed new attributes:** In this step will be done the planning and conduction of an experimental study to evaluate the applicability of the proposed new attributes. For the development of the experiment, the experimental process defined by [Wohlin et al. 2012] will be followed.

It is currently being conducted the step 3, where the SeleCTT tool is being investigated so that it can be better understood how the tool was developed to continue the evolution of it where the selection approach proposed in this master's project will be implemented.

5. Description and Evaluation of Results

During the first year of masters, it was conducted a systematic mapping, with the objective of identifying and analyzing papers that represent the current state of the literature about testing technique selection. The results provide a discussion on: (i) the existing approaches proposed in the literature to select testing techniques; (ii) a classification schema of approaches for testing techniques selection; (iii) the types of empirical methods used to evaluate the proposals; (iv) a complete list of attributes that supports the the process of choose the best technique for software project; and (v) approaches which allow combined testing techniques selection. Finally, the mapping study helped us to identify the combined selection approaches proposed in the literature. We can observe the improvements that the combination of testing techniques can bring to the development process.

Then, we surveyed the software testing practices carried in Brazil software companies. The goal of our survey is to identify the testing practices in Brazilian industries, for the purpose of knowing and having an overview on the latest testing techniques, tools and metrics used, the challenges faced by testers and the process of testing technique selection, which is the central concern of this Master's project. Furthermore, we were able to identify through this study the software testing practices most used by testers. We can highlight in the results that the testers select a testing technique according to the scope of the project. The testing technique most used in software projects is the functional and structural testing techniques and the least used is mutation testing. Among testing tools, Selenium was indicated as the tool more used in testing activities. Finally, we identified the challenges faced by testers that could bring interesting topics to be explored more deeply by researchers.

Currently, we were analyzing the SeleCTT tool functioning that has the purpose of assisting the process of selecting the most appropriate technique for a specific software project. To meet this task, the user provides the tool with data that characterizes their software project. The tool operates by collecting this data, storing it in a list of attributes, and comparing this data with the information of the testing techniques already stored in the database. The classification of the techniques is based on the calculation of the Euclidean distance between vectors, based on the weights established for each of the analyzed attributes. The results are presented to the user in the form of an ordered list of techniques, where they can view and expand the results for each of the listed techniques.

6. Comparison with Related Work

Choosing the testing technique should not only be based on subjective knowledge, but it should also incorporate objective knowledge, guided by elements that favor a good choice [Victor and Upadhyay 2011]. The study described in [Wojcicki and Strooper 2007] consists of a framework that uses a systematic strategy focused on maximizing completeness and minimizing effort using metrics such as effort and defect-detection effectiveness. In the work of [Rothermel and Harrold 1996] is used a framework to analyzing and evaluated regression test selection techniques. This framework consists of four categories: inclusiveness, precision, efficiency, and generality.

The work proposed in [Dias-Neto and Travassos 2009] selects testing techniques in the context of model-based testing, and has developed the Porantim tool to support software engineers in the selection of model-based testing techniques for a software project.

In our work, we are interested in improving the testing techniques selection in the context of concurrent software testing.

Acknowledgements

The authors acknowledge FAPESP (São Paulo Research Foundation), for the financial support under process number 2018/10183-9.

References

Ammann, P. and Offutt, J. (2016). *Introduction to software testing*. Cambridge University Press.

Delamaro, M., Jino, M., and Maldonado, J. (2017). *Introduction to Software Testing*. Elsevier Brazil.

Dias-Neto, A. C. and Travassos, G. H. (2009). Evaluation of {model-based} testing techniques selection approaches: An external replication. In *Empirical Software Engineering and Measurement, 2009. ESEM 2009. 3rd International Symposium on*, pages 269–278. IEEE.

Melo, S. M. (2018). *A Framework for systematic testing techniques evaluation applied to concurrent programming*. PhD thesis, Universidade de São Paulo.

Moura, F. M., Melo, S. M., and Souza, S. R. S. (2018). Selectt: tool for automating the selection process of concurrent software testing techniques. *Universidade de São Paulo. Relatório técnico Nº 423.*

Myers, G. J., Sandler, C., and Badgett, T. (2011). *The art of software testing*. John Wiley & Sons.

Rothermel, G. and Harrold, M. J. (1996). Analyzing regression test selection techniques. *IEEE Transactions on software engineering*, 22(8):529–551.

Souza, S. R. S., Souza, P. S. L., Melo, S. M., Silva, R. A., and Vergilio, S. R. (2017). *Testing of Concurrent Programs*, pages 261–296. In: Delamaro, Marcio and Jino, Mario and Maldonado, José. (Org.). Introduction to Software Testing. Elsevier, Rio de Janeiro, RJ.

Tanenbaum, A. S. and Van Steen, M. (2007). *Distributed systems: principles and paradigms*. Prentice-Hall.

Vegas, S. and Basili, V. (2005). A characterisation schema for software testing techniques. *Empirical Software Engineering*, 10(4):437–466.

Victor, M. and Upadhyay, N. (2011). Selection of software testing technique: A multi criteria decision making approach. In *Trends in Computer Science, Engineering and Information Technology*, pages 453–462. Springer.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., and Wesslén, A. (2012). *Experimentation in software engineering*. Springer Science & Business Media.

Wojcicki, M. A. and Strooper, P. (2007). An iterative empirical strategy for the systematic selection of a combination of verification and validation technologies. In *5th International Workshop on Software Quality*, page 9. IEEE Computer Society.