

# Systematic mapping on structural testing criteria for concurrent programs considering dynamic aspects

Silvia M. D. Diaz<sup>1</sup>, Paulo S. L. de Souza<sup>1</sup>

<sup>1</sup>Institute of Computer and Mathematics Sciences - University of Sao Paulo (USP)  
Sao Carlos - SP - Brazil

smdiazdiaz@usp.br, pssouza@icmc.usp.br

***Abstract.** As there are several approaches to define structural testing criteria for concurrent programs, it is necessary to describe the state of knowledge in this topic in order to identify the gaps that require to be addressed. Thus, it is fundamental to learn about who is investigating, which approaches and what purposes are there in current research, to evaluate which is the most effective path to follow to fill in divergences. Therefore, the goal of this systematic mapping is to find studies related to the testing of concurrent programs, that allows evaluating what is the best option to guide a research in the improvement of concurrent software quality through structural criteria. The results of the systematic mapping showed that there is indeed a gap in the definition of criteria for concurrent programs considering dynamic aspects.*

## 1. Introduction

The main objective of this secondary study is to find related studies to a research in structural testing of concurrent programs that considers dynamic aspects, in order to expose the gap among the current mentioned research and results of previous investigations on the topic, and thus allow the clarification of its contributions. The planning and definition of research questions and establishment of search string are fundamental for every secondary study, in order to obtain consistent results [Kitchenham and Charters 2007]. This systematic mapping protocol summary presents the scope and goals of the study to be executed based on guidelines [Petersen et al. 2008] that along research questions require selection of primary studies, definition of inclusion and exclusion criteria and the classification of the selected studies.

## 2. Planning

The goals of this study oriented the definition of the research questions, and represent the expected results that aim to establish a categorization of the primary studies.

1. Who is investigating about structural criteria for concurrent programs that base on their dynamic aspects (who is interested)?
2. How is the problem being addressed?
3. Which approaches have been considered in these researches?
4. How are the approaches being implemented?
5. Who is using symbolic execution, and if not why is it anybody using it?
6. Where are those studies being published (country, journal, events, groups, etc.)?
7. Are those studies applicable? In addition, are they academic or practically focused?

The main terms of the search string consisted of the topics related to *concurrent programs*, *structural testing* and *dynamic analysis*, including their synonyms. It was also considered the calibration of the string and its implementation in the databases (relevant to the research area and recognized sources of publications: ACM Digital Library, IEEE Xplore, Scopus, ScienceDirect - Elsevier, SpringerLink and Web of science).

### 3. Execution

The execution consists of the selection and evaluation of the studies for data extraction. The selection of primary studies obtained from the execution of the search strings in the databases, threw a total of **1599**, disregarding the duplicates. The strategy to evaluate the papers consisted of two main stages: The first applied exclusion and inclusion criteria to the titles and abstracts of the papers, when none of those elements allowed clarifying whether the paper should be included or not, it was considered for evaluation in the second stage. Second stage consisted of two phases in order to optimize the selection of the final papers. An early phase considered the reading of the introduction and conclusion sections of the papers (included from the first stage), extracting punctual information that allowed deciding whether the contributions were relevant or not. And the other phase consisted of a full-paper reading of the remaining studies that had the highest quality contributions, and that were used to extract the data for the analysis of results and categorization of studies.

### 4. Results

Stage one left **217** papers after implementing the criteria, showing that a large quantity of studies were discarded considering the exclusion criteria and that did not correspond to computer science areas. Figure 1a shows the percentage of included papers selected in the first stage. Figure 1b shows the percentage of papers excluded by the criteria. The IC3 criterion refers to approaches that use dynamic analysis along with static analysis to define criteria, while EC3 corresponds to papers that use concurrent or parallel programming for optimization of computations, or viceversa; being these the most common in both cases.

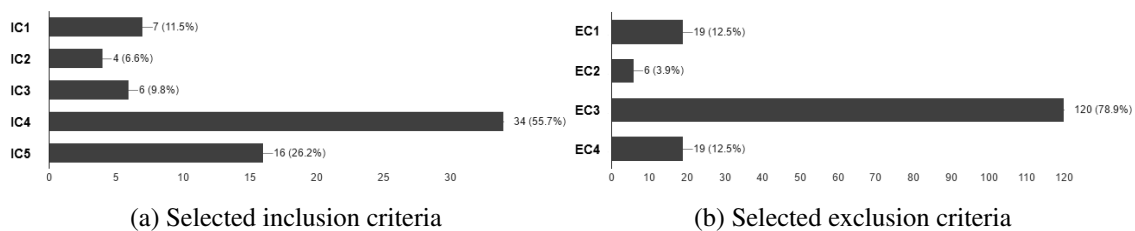


Figure 1. Number of papers per inclusion and exclusion criteria

For stage two, the included papers were read in completeness in order to obtain the required and necessary information to be classified. There were found studies that did not referred to concurrent programs or that used different representations; however, some characteristics, methods, tools or metrics were found useful for the current research in structural testing of concurrent programs. Most of the papers considered different approaches for testing concurrent programs, and some others focused on different areas of Computer Science such as hardware testing methods. All of the references were checked

and verified in order to ensure the reliability of the source. The second stage left a total of **61** references and related work papers. Figure 2 shows the number of papers published each year from the results in the second stage. It can be seen that the last past years have had an increase in investigations related to the area. Figure 3 presents the percentage of papers in each category. Although the majority of papers classify in C1, few papers considered the three main concepts in the research objective of the systematic mapping.

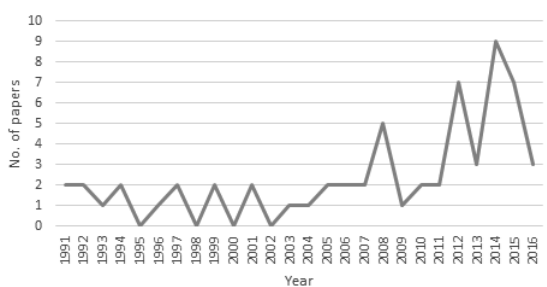


Figure 2. Number of papers per year

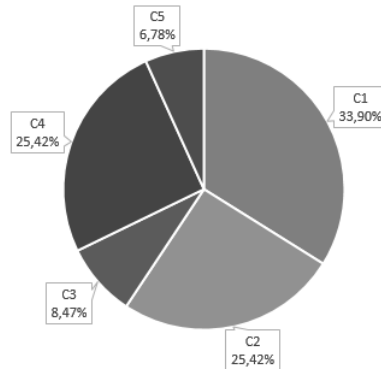


Figure 3. Percentage of papers per category

## 5. Analysis

For the inclusion criteria in stage two, the ideal was that most of the papers corresponded to IC1, since it involves directly the objective and research questions; however, dynamic analysis inclusion criteria (IC4) presented a high quantity of papers. Regarding exclusion criteria, the most frequent is EC3. The majority of studies considered static and dynamic analysis techniques for testing both concurrent and sequential programs, but not necessarily working the techniques accordingly. Few papers treated the three concepts in the research objective and some of them implemented symbolic execution methods for test data generation. Categories for the classification of papers were identified considering the evaluation and data extracted in stage two and the content of each one of the papers, as follows:

- C1.** Dynamic and static analysis combined in software testing.
- C2.** Other approaches on testing concurrent programs.
- C3.** Frameworks and characteristics for testing distributed systems and concurrent programs.
- C4.** Methods for software testing that implement one of the concepts.
- C5.** Test data generation or metrics definition for concurrent programs.

The research question resolution exposes the results of the systematic mapping and the accomplishment of its objectives:

1. *Who is investigating about structural criteria for concurrent programs that base on their dynamic aspects (who is interested)?* The greatest quantity of papers were published by universities, events, journals, conferences or books from USA, Germany and the UK, which could also be confirmed during the revision of the references. On the other hand, it was found that studies are more oriented to propose new methods that use different approaches to reveal concurrency errors in programs, together with dynamic analysis. Most of the papers belong to journal publications and conference proceedings, as well as lecture notes. Given the current status of constant research of this topic, there are few book chapters that address this area.

2. *How is the problem being addressed?* There were found different approaches to test concurrent programs, as seen in classification C2. Many of them consider dynamic analysis along with static analysis techniques, since it was observed that they can complement each other because of their strengths and limitations.
3. *Which approaches have been considered in these researches?* As mentioned before, the considered approaches correspond to reachability testing, symbolic execution, data race detection techniques, path selection methods, coverage-based testing, noise injection techniques, genetic algorithm based methods and path selection techniques; these were classified in C2 since they treated different approaches with a relation with structural testing static and dynamic analysis techniques.
4. *How are the approaches being implemented?* Through the implementation of frameworks, metrics according to different testing criteria, optimization of existing testing tools and realization of case studies to validate the theories.
5. *Who is using symbolic execution, and if not why is it anybody using it?* In the classification C4 methods for software testing that recognized symbolic or concolic execution, as well as model checking, were included these papers. There are indeed several researches that base on this technique, since it provides an advantage in test data generation; however, one of its limitations is the cost of implementation and the few quantity of studies for concurrent programs.
6. *Where are those studies being published (country, journal, events, groups, etc.)?* Most of the papers have been published in the USA, being the majority from conference events.
7. *Are those studies applicable? In addition, are they academic or practically focused?* The studies are applicable since most of them are verified through case studies, existing testing tools (with modifications according the requirements of the researches) and reference to other papers. In its greater quantity, they belong to academic papers that intend to contribute to the testing of concurrent programs area, but some interesting studies from the NASA have been published about the tool Java PathFinder, which at the same time has been utilized by several researches that implement symbolic execution.

## 6. Conclusion

The systematic mapping allowed finding related studies in the research field, and it could be confirmed that there is indeed a gap in the definition of structural testing criteria for concurrent programs that considers dynamic aspects. However, there are researches that utilize different techniques to implement static and dynamic analysis methods, such as symbolic and concolic execution; also, some of them focus on the coverage criteria to guide the use of techniques and tools to test (or to generate test data) concurrent programs. This work presented the results of a systematic mapping and how they relate to the current research in structural testing of concurrent programs by categorizing them.

## References

- Kitchenham, B. and Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering. Technical Report EBSE 2007-001, Keele University and Durham University Joint Report.
- Petersen, K., Feldt, R., Mujtaba, S., and Mattsson, M. (2008). Systematic mapping studies in software engineering. In *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering, EASE'08*, pages 68–77, Swindon, UK. BCS Learning & Development Ltd.