TESTDAS: Testing Method for Dynamically Adaptive Systems

Ismayle de Sousa Santos*
Federal University of Ceará
Fortaleza, Ceará
ismaylesantos@great.ufc.br

Rossana Maria de Castro Andrade†
Federal University of Ceará
Fortaleza, Ceará
rossana@ufc.br

Pedro de Alcântara dos Santos Neto‡
Federal University of Piauí
Teresina, Piauí
pasn@ufpi.edu.br

CCS CONCEPTS

• **Software and its engineering → Software verification and validation**;

KEYWORDS

Dynamically Adaptive Software, Dynamic Software Product Line, Software Testing, Model Checking

ACM Reference Format:
Ismayle de Sousa Santos, Rossana Maria de Castro Andrade, and Pedro de Alcântara dos Santos Neto. 1997. TESTDAS: Testing Method for Dynamically Adaptive Systems. In *Proceedings of SBQS - CTDQS (SBQS'18),* Jennifer B. Sartor, Theo D'Hondt, and Wolfgang De Meuter (Eds.). ACM, New York, NY, USA, Article 4, 1 page. https://doi.org/10.475/123_4

1 MOTIVATION

Dynamic systems exploit contextual information to adapt at runtime, according to changes in their surrounding environment. Such dynamic behavior is typically designed using adaptation rules, and many of the pervasive and software-intensive systems that use context properties are based on a Dynamic Software Product Line (DSPL) approach [1].

Despite having some works addressing the testing and model checking of DSPL, there is a lack of work addressing the effects of the adaptation rules on the feature status. Therefore, there is a lack of testing method focused on the actions (*i.e.*, activation and deactivation of system features) of adaptation rules to validate the DSPL adaptations triggered by context changes.This is corroborated by the systematic review performed on context-aware application testing [4]. Also, there is a lack of a formalism and model checking focused on the adaptation rules and their effects on the DSPL products.

2 TESTDAS OVERVIEW

In a nutshell, the proposed method, called TestDAS, receives as input the DAS feature model with adaptation rules, and a context variation model. Then, it provides an approach to DAS model checking and generates tests for validating the DAS adaptive behavior.

Firstly, the Software Engineer should specify the DAS adaptive behavior using the DFTS [3], which is the formalism proposed to models the changes of the DAS configurations according to context changes and the triggered adaptation rules. In this work, the DFTS is created using Promela, which is the language of the SPIN model checker tool. To support the DAS model checking, a set of five general properties related to adaptation fault patterns were proposed [2, 3]. Besides that, other domain-specific properties can be checked. If any of the properties is violated, then the DAS has design faults in the adaptive behavior specification. Those faults should be fixed and once the DAS design is correct, the Tester selects the test coverage criteria that define the required coverage. After that, the Tester must generate test cases ("adaptation test sequences") to satisfy the test criteria.

A supporting tool, called TestDAS tool, was implemented to generate the test sequences. Also, a library called CONTroL was implemented to support the test sequences execution.

To assess the TestDAS and the supporting tools, three kinds of evaluations were performed. The mutant analysis gathered data that showed that the model checking approach proposed is effective in the identification of behavioral fault patterns. In the controlled experiment, the results indicate that TestDAS generates more tests and achieves a better coverage than the experience-based testing. Besides, the observational study indicates that the TestDAS tool and CONTroL support the TestDAS activities successfully.

3 CONCLUSIONS

This thesis proposed the TestDAS, which is a DAS testing method that includes: (i) a set of test coverage criteria used to generate test sequences to validate the adaptive behavior; (ii) a model, called Dynamic Feature Transition System (DFTS), to specify the DAS adaptive behavior; (iii) a set of behavioral properties that DAS should satisfy; and (iv) a model checking approach that can identify design faults; and (v) two supporting tools. From the results of this thesis, the main future research directions are to provide support to runtime testing and handle the context changes during the testing.

REFERENCES

[1] R. Capilla, O. Ortiz, and M. Hinchey. 2014. Context Variability for Context-Aware Systems. *Computer* 47, 2 (Feb 2014), 85–87. https://doi.org/10.1109/MC.2014.33
[2] Ismayle S. Santos, Rossana Maria Castro Andrade, Lincoln Souza Rocha, Santiago Matalonga, Kathia Marcal de Oliveira, and Guilherme Horta Travassos. 2017. Test Case Design for Context-aware Applications: Are We There Yet? *Inf. Softw. Technol.* 88, C (2017), 1–16. https://doi.org/10.1016/j.infsof.2017.03.008
[3] Ismayle S. Santos, Lincoln S. Rocha, Pedro A. Santos Neto, and Rossana M. C. Andrade. 2016. Model Verification of Dynamic Software Product Lines. In *Brazilian Symposium on Software Engineering*. ACM, New York, NY, USA, 113–122.
[4] Ismayle S. Santos, M. L. J. Souza, M. L. L. Carvalho, T. A. Oliveira, E. S. Almeida, and R. M. C. Andrade. 2017. Dynamically Adaptable Software is All about Modeling Contextual Variability and Avoiding Failures. *IEEE Software* (2017).

*Scholarship sponsored by Capes during Ph.D..
†Researcher scholarship - DT level 2, sponsored by CNPq
‡Researcher scholarship - DT level 2, sponsored by CNPq

SBQS'18, Setembro 2018, Curitiba, ParanÃą

ACM ISBN 123-4567-24-567/08/06.
https://doi.org/10.475/123_4