Towards Runtime Testing of Dynamically Adaptive Systems based on Behavioral Properties

Erick Barros dos Santos[1]*, **Rossana Maria de Castro Andrade**[1]†, **Ismayle de Sousa Santos**[1]

[1]Department of Computer Science
Federal University of Ceará - (UFC) - Fortaleza, Ceará, Brazil

`{erickbarros,rossana,ismaylesantos}@great.ufc.br`

Abstract. *Dynamically Adaptive Systems (DAS) support adaptations to deal with changes in the user requirements and the environment's constraints at runtime. A DAS can have high dynamicity of its configurations at runtime, so a major challenge is to perform quality assurance activities. In the literature, approaches that perform runtime testing mostly uses the DAS operational context to generate test plans, then missing other relevant runtime data (e.g., violation of behavioral properties) in the process. Therefore, this master thesis proposes an approach that monitors behavioral properties and use its results to improve the selection and execution of tests at runtime. Thus, the main contribution is the identification of failures in the adaptation mechanism during the DAS execution.*

1. Introduction

Currently, systems are capable of obtaining information about its environment, the own system, and its goals to perform self-adaptation in its behavior or structure [Lemos et al. 2013]. This information defines the system context, which is any information that can be used to characterize the situation of an entity (person, place or object), including the user and applications themselves [Abowd et al. 1999].

These context-triggered adaptations are necessary for the system to increase the dependability or to deal with the rapid evolution of requirements [Lahami et al. 2016]. However, considering the state explosion problem of DAS [Lim et al. 2015], testing all configuration at design time could be infeasible. Then, the DAS adaptation mechanism must accomplish Verification and Validation (V&V) tasks during runtime to detect adaptation failures, where techniques such as the verification of properties and software testing can take place [Tamura et al. 2013]. Thus, this master thesis proposes an approach that uses the runtime information from the DAS behavioral properties to refine the generation of a runtime test plan, aiming to increase its fault detection capability.

2. Problem and Motivation

The inherent complexity related to the DAS characteristics can hinder the V&V activities in this kind of system. Studies have identified challenges such as the testing of DAS configurations that exponentially grow [Lim et al. 2015] and identification of wrong configurations at runtime [Xu et al. 2012]. Therefore, the application of traditional V&V methods during design time may not be suitable to ensure the correctness of the DAS adaptations, thus requiring the creation of specific methods for runtime [Tamura et al. 2013].

*Master Researcher scholarship - sponsored by CNPq
†Researcher scholarship - DT Level 2, sponsored by CNPq

Studies in the literature have been dealing with the problem of DAS V&V at runtime. The work of Fredericks and Cheng [Fredericks and Cheng 2015] proposes a requirements-driven approach to generate adaptable unit test plans based on the DAS operational context. Similarly, the work of Lahami et al. [Lahami et al. 2016] presents a solution of adaptable black-box test plans but focusing on the available resources to its execution. Lim and Bae [Lim et al. 2015] target the runtime testing through the analysis of traces to select test cases that verify the quality properties after the adaptation. Xu et al. [Xu et al. 2012] present a model-based approach to the runtime verification of adaptation properties in context-aware applications through assertions in the source code.

The aforementioned approaches mostly focus on the generation of test plans based on the current context, DAS resources or are restricted to the verification of properties without testing the features. Nevertheless, they are limited, because the focus on the operational context may select test cases that do not properly verify the DAS features throughout its sequences of adaptations, in which the satisfaction of behavioral properties can help to improve this selection to reveal adaptation failures. For instance, given a feature that is not activated when required (violating the *Feature Liveness* property [Santos et al. 2016]), then, more test cases must be selected in this feature to verify its correct behavior when active. In this sense, Santos et al. [Santos et al. 2016] defines behavioral properties that can be used as test coverage criteria in addition to the current operational context to generate the runtime test plans. However, to the best of our knowledge, there is a lack of approaches that make use of this kind of criterion to select test cases at runtime and focused on the DAS adaptation mechanism.

3. Approach Overview

This paper presents an initial proposal of an approach to test the DAS through runtime selection and execution of test plans to verify structural or behavioral adaptations of DAS. One of the characteristics of the proposed approach is to make use of a variability model that maps the active DAS features to a given context, allowing the verification of the behavioral properties and also the test case selection from a high-level representation of the system. Also, we will propose a mechanism capable of executing tests during design time as well as runtime.

The proposed approach will be supported by a framework based on our previous work [Santos et al. 2018], in which was presented CONTroL (CONtext-variability-based software Testing Library), a tool for instrumenting a DAS and executing test sequences to verify the reconfigurations of its features. With regards to the system model, DFTS (Dynamic Feature Transition System) [Santos et al. 2016], which models the DAS features and context according to the DAS feature model and adaptation rules, will be used.

Figure 1 summarizes the approach activities which requires the following input artifacts: (i) DFTS model; and (ii) the DAS unit test cases. The details of the proposed approach are presented as follows.

1. **DAS Instrumentation**: the first activity comprises the analysis of the inputs artifacts by the tester, which will provide the information to instrument the DAS using an annotation-based mechanism. The instrumentation enables the monitoring of the operational context and the status (activated/deactivated) of the features. It also has the purpose of mapping at runtime the DAS state to the DFTS model, thereby

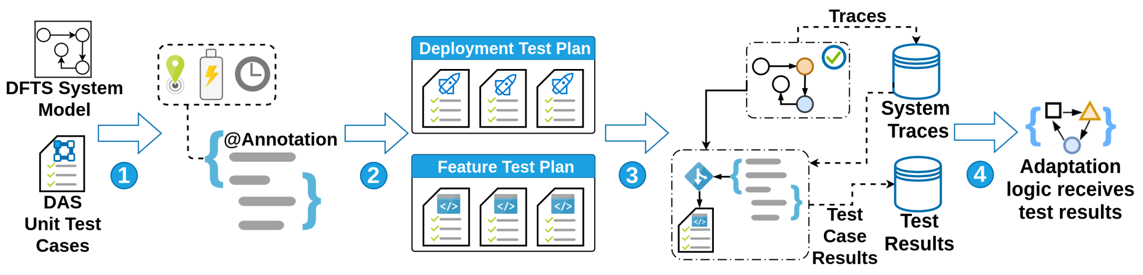

Figure 1. Approach Overview

allowing the monitoring of the system traces. To perform the monitoring, the framework intercepts the execution flow through aspect-oriented programming.

2. **Specification of Test Plans**: the second activity consists of the automatic generation of the deployment test plan based on the DFTS model. This test plan has the purpose of testing the DAS adaptations through the execution of test sequences. Furthermore, the tester must specify a test plan that involves the unit test cases to continually test the features during the DAS execution.
3. **Runtime Testing**: the third activity is the execution of the DAS and the test plans. The execution of the deployment test plan begins soon after the DAS deploy, manipulating context and verifying the correct (de)activation of features. After that, the framework continually monitors the DAS to assert the properties defined in [Santos et al. 2016] after each adaptation. The results of the runtime verification, the current context, and the DFTS traces are used to select the features test plan.
4. **Result Feedback**: the last activity comprises the feedback of the test results to the DAS. The framework must send these results to the DAS after the execution of the deployment test plan at design time, and also during the monitoring and feature test at runtime. With the test results, the DAS's adaptation mechanism then could decide for a new adaptation plan or not.

4. Evaluation

To evaluate the proposed approach, we intend to perform controlled experiments following the guidelines in [Wohlin et al. 2012]. To these experiments, two adaptive systems for mobile devices were selected: GREat Tour[1] and PhoneAdapter[2]. To measure the failure detection capability of the approach, we plan to insert faults in the adaptation rules of the selected DAS and then collect the number of detected ones. Besides that, to measure the performance of our framework, we will also collect the following metrics: generation and execution time of the test plans, and the usage of processing and memory resources.

5. Preliminary Results

The current master's work is in its initial stage. Thus, it was performed a literature review, in which were identified the work dealing with the analyses in DAS runtime assurance, the methods, and taxonomies that motivate the formulation of the proposed approach. Currently, the CONTroL tool is being evolved with new annotations to instrument context variables and adaptation reactions in the DAS source code. Additionally, it was developed an algorithm for the generation of the deployment test plan.

[1]http://mobiline.great.ufc.br/

[2]http://sccpu2.cse.ust.hk/afchecker/phoneadapter.html

6. Final Remarks

The main expected contribution of this master thesis is a new approach that combines monitoring of properties over the DAS behavior and testing to verify the adaptations of DAS features at runtime, thus helping testers to identify not only violation of properties but also failures related to the features. The next activities involve: (i) to search for ways to specify the proposed test plans; (ii) to evolve the CONTroL tool to conduct runtime verification; and (iii) to implement the test selection and execution.

References

Abowd, G. D., Dey, A. K., Brown, P. J., Davies, N., Smith, M., and Steggles, P. (1999). Towards a better understanding of context and context-awareness. In *International symposium on handheld and ubiquitous computing*, pages 304–307. Springer.

Fredericks, E. M. and Cheng, B. H. (2015). Automated generation of adaptive test plans for self-adaptive systems. In *Proceedings of the 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, pages 157–168. IEEE Press.

Lahami, M., Krichen, M., and Jmaiel, M. (2016). Safe and efficient runtime testing framework applied in dynamic and distributed systems. *Science of Computer Programming*, 122:1–28.

Lemos, R. d., Giese, H., Müller, H. A., Shaw, M., Andersson, J., Litoiu, M., Schmerl, B., Tamura, G., Villegas, N. M., Vogel, T., et al. (2013). Software engineering for self-adaptive systems: A second research roadmap. In *Software Engineering for Self-Adaptive Systems II*, pages 1–32. Springer.

Lim, Y. J., Jee, E., Shin, D., and Bae, D.-H. (2015). Efficient testing of self-adaptive behaviors in collective adaptive systems. In *2015 IEEE 39th Annual Computer Software and Applications Conference*, volume 2, pages 216–221. IEEE.

Santos, I. d. S., Santos, E. B. d., Andrade, R. M. d. C., and Neto, P. d. A. d. S. (2018). Control: Context-based reconfiguration testing tool. In *Brazilian Conference on Software – Tools Session*, page 6. ACM.

Santos, I. S., Rocha, L. S., Neto, P. A. S., and Andrade, R. M. C. (2016). Model verification of dynamic software product lines. In *Proceedings of the 30th Brazilian Symposium on Software Engineering*, SBES '16, pages 113–122, New York, NY, USA. ACM.

Tamura, G., Villegas, N. M., Müller, H. A., Sousa, J. P., Becker, B., Karsai, G., Mankovskii, S., Pezzè, M., Schäfer, W., Tahvildari, L., et al. (2013). Towards practical runtime verification and validation of self-adaptive software systems. In *Software Engineering for Self-Adaptive Systems II*, pages 108–132. Springer.

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

Xu, C., Cheung, S., Ma, X., Cao, C., and Lu, J. (2012). Adam: Identifying defects in context-aware adaptation. *Journal of Systems and Software*, 85(12):2812–2828.