1. OVERVIEW

Software systems that can automatically adapt to changes in their environments – also known as Dynamically Adaptive Systems (DAS) – play increasingly important roles in society’s everyday life. The growing demand for such systems, which range from simple mobile interactive gaming applications to complex crisis management systems, requires specific development methodologies that encompass the full software life cycle.

The EU-funded DiVA project (Dynamic Variability in Complex, Adaptive Systems) [1] aims at providing a tool-supported methodology for developing DASs. The DiVA framework addresses the management of dynamic variability through a combination of aspect-oriented and model-driven techniques.

Within the DiVA project context, we are interested in developing testing strategies that can adequately handle the issues involved in testing DASs. In particular, the combinatorial explosion of variant composition alternatives, many of which probably unforeseen at design time, and the runtime nature of system re-configuration, makes testing a daunting task for software developers.

We started our investigation by performing a systematic mapping study of testing approaches for DASs. The goals of our survey were two: (RQ1) identifying the challenges authors have described so far in the context of DASs testing; and (RQ2) identifying candidate testing approaches which can be customised to address the challenges within the scope of the DiVA project.

In regard to the research question (RQ1), we enumerated the following challenges:

1. How to guarantee the correctness of systems that have never been tested in advance [5, 6, 13];
2. How to detect and avoid incorrect system configurations defined at runtime [5, 6];
3. How to test adaptive systems that run in a distributed environment [8, 7];
4. How to deal with user interference in the system configuration [7];
5. How to anticipate all the relevant context changes and when they could impact the behaviour of context-aware applications [12];
6. How to deal with context-dependent control and data flow in DASs (e.g. how to avoid “noisy contexts” resulting from sensor interference) [3, 4];
7. How to deal with the exponential growth of system configurations that should be tested [13].

Note that some challenges can be more adequately tackled with approaches other than testing (e.g. challenge #2 is more about rule-based reasoning). Some others are domain-specific (e.g. challenge #3 that regards distributed systems and #6 that regards sensor-based systems). Nevertheless, the current state of the art shows that improving these systems’ reliability through testing still requires large effort from the research community and software practitioners.

In regard to the research question (RQ2), some approaches address the reduction of the associated costs through built-in testing strategies [5, 6] or through the prioritisation of components to be tested [13]. Other approaches include: applying model-checking algorithms for automatic detection of faults related to distribution [8, 7]; analysing the control and data flow focusing on context-aware data [12, 11, 3, 4, 2]; and analysing multi-layer context-aware adaptive applications [9].

The next step of our research is identifying how we can leverage the current testing approaches for DASs in order to compose an adequate testing framework for DiVA systems. An important property of the DiVA approach is that it is strongly oriented to model transformations. Along the full system life cycle, models are enriched and transformed with proper tooling support [10]. This shall be taken into account in our upcoming research initiatives.

Acknowledgments

This work is sponsored by the European Union as part of the DiVA project (ICT-215412).

References


