

# Automated Scenario-based Testing of Distributed and Heterogeneous Systems

Bruno Lima

INESC TEC and Faculty of Engineering, University of Porto

Rua Dr. Roberto Frias s/n, 4200-465 Porto, PORTUGAL

Email: bruno.lima@fe.up.pt

**Abstract**—In this document we outline a Ph.D. research plan and a summary of preliminary results on test automation for distributed and heterogeneous systems.

## I. MOTIVATION

Due to the increasing ubiquity, complexity, criticality and need for assurance of software based systems [1], testing is a fundamental lifecycle activity, with a huge economic impact if not performed adequately [2]. Such trends, combined with the needs for shorter delivery times and reduced costs, demand for the continuous improvement of software testing methods and tools, to make testing activities more effective and efficient.

Nowadays software is not more like simple applications but has evolved to large and complex system of systems [3]. A system of systems consists of a set of small independent systems that together form a new system. The system of systems can be a combination of hardware components (sensors, actuators, etc.) and software systems used to create big systems or ecosystems that can offer multiple different services. Currently, systems of systems capture a great interest from the software engineering research community. Testing these distributed and heterogeneous software systems or systems of systems, running over interconnected mobile and cloud based platforms, is particularly important and challenging. Some of the challenges are: the difficulty to test the system as a whole due to the number and diversity of individual components; the difficulty to coordinate and synchronize the test participants and interactions, due to the distributed nature of the system; the difficulty to test the components individually, because of the dependencies on other components.

An example of a distributed and heterogeneous system is the Ambient Assisted Living (AAL) ecosystem that was prototyped in the context of the nationwide AAL4ALL project [4]. The AAL4ALL ecosystem comprises a set of interoperable AAL products and services (sensors, actuators, mobile and web based applications and services, middleware components, etc.), produced by different manufacturers using different technologies and communication protocols (web services, message queues, etc.). To assure interoperability and the integrity of the ecosystem, it was developed and piloted a testing and certification methodology [5], encompassing the specification of “standard” interfaces and component categories, the specification of unit (component) and integration test scenarios, and the test implementation and execution on candidate compo-

nents by independent test laboratories. A major problem faced during test implementation and execution was related with test automation, due to the diversity of component types and communication interfaces, the distributed nature of the system, and the lack of support tools. Similar difficulties have been reported in other domains, such as the railway domain [6]. In fact, we found in the literature limited tool support for automating the whole process of specification-based (or model-based) testing of distributed and heterogeneous systems, as will be explained in Section III.

## II. GOALS

The goal of this work is to develop an approach and a toolset to automate the whole process of scenario-based testing of distributed and heterogeneous systems, with a focus on integration testing of end-to-end services, but supporting also unit (component) and system testing. The only manual activity should be the creation of the input model of the system under test (SUT). The aim is to significantly reduce the cost of testing such systems, from the standpoint of time, resources and expertise required, as compared to existing approaches.

To that end, the following sub-objectives will be pursued:

- Define appropriate modelling notations. To conciliate the needs of accessibility and executability, we intend to adopt different “frontend” and “backend” modelling notations, with automatic translation from a visual notation used by the user to a formal notation amenable for incremental execution at runtime.
- Define test generation and execution algorithms.
- Design and implement a support toolset, to fully automate the proposed test process and promote reuse and extensibility.

## III. POSITIONING IN THE STATE-OF-THE-ART

Although we didn’t find in the literature model-based testing (MBT) approaches supporting in an integrated fashion the whole test automation process for distributed systems, we found several works supporting parts of the process, that can help in the construction of an integrated approach and toolset. We also found in state of the art some concepts, architectures and frameworks from which we can take some ideas for our solution, namely the theoretical work on possible solutions to events synchronization in distributed systems testing, existing

MBT approaches to test standalone systems and the use of hybrid test architectures for distributed systems testing.

A more detailed state of the art analysis can be found in [7].

#### IV. SUMMARY OF RESULTS TO DATE

Until this moment it was conducted research work regarding the state of the art analysis, the state of the practice analysis, and the conception of the overall solution.

Regarding the state of the art analysis, it was conducted a systematic analysis of works on model based testing and distributed systems testing.

Regarding the state of the practice analysis, we conducted a survey on “Testing Distributed and Heterogeneous Systems – State of the practice”, that was distributed to the participants of two industry-oriented conferences in the software testing area (TESTING Portugal 2015 and UCAAT 2015) and was responded by 147 persons. This survey is available at <https://goo.gl/GExS2w>.

Regarding the conception of the overall solution, we conceived a preliminary approach and a toolset architecture for automating the testing of end-to-end services in distributed and heterogeneous systems. In the envisioned approach, the tester interacts with a visual modeling frontend to describe key behavioral scenarios based on UML sequence diagrams, invoke test generation and execution, and visualize test results and coverage information back in the model. The visual modeling notation is converted to a formal notation amenable for runtime interpretation in the backend based on extended Petri Nets. A distributed test monitoring and control infrastructure is responsible for interacting with the components of the system under test, as test driver, monitor and stub. At the core of the toolset, a test execution engine coordinates test execution and checks the conformance of the observed execution trace with the expectations derived from the visual model. This approach was fully described in [7].

#### V. EXPECTED BENEFITS

As compared to existing approaches, we expect that our envisioned approach will bring the following benefits:

- higher level of automation of the testing process - all phases of the process are supported in an integrated fashion; the only manual activity needed is the development in a user friendly notation of the model required as input for automatic test generation and execution; there is no need to develop test components specific for each SUT;
- higher fault detection capability - the usage of a hybrid test architecture allows the detection of a higher number of errors as compared to purely distributed or centralized architectures [8]; besides the interactions of the SUT with the environment, interactions between components in the SUT are also monitored and checked against the specification; with an incremental conformance checking algorithm, the execution state of the SUT can be captured as soon as a failure occurs, to facilitate subsequent fault

diagnosis; ability to test non-deterministic SUT behaviors, using an online, adaptive, test generation strategy; ability to test temporal constraints;

- easier support for multiple test levels - the same input model can be used to perform tests at different levels (unit, integration, and system testing), simply by changing the selection of observable and controllable events in the input model; a scenario-oriented approach simplifies the level of detail required in the input models;
- more efficient test execution - with a distributed conformance checking algorithm, communication overheads during test execution are minimized; usage of a state-oriented runtime model allows a more efficient model execution and conformance checking.

#### VI. VALIDATION METHODOLOGY

In order to validate the solution approach and toolset with respect to the research goals previously described, we intend to conduct real world case studies, one of which in the Ambient Assisted Living domain.

#### VII. CONCLUSION AND FUTURE WORK

It was presented a research plan and a summary of preliminary results of a Ph.D. work on test automation for distributed and heterogeneous systems. As next steps we intend to design and validate appropriate notations and algorithms for supporting the proposed approach described in [7].

#### ACKNOWLEDGMENT

The author would like to thank the Ph.D. advisor, Professor João Pascoal Faria for his support.

This work is financed by the ERDF – European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme within project «POCI-01-0145-FEDER-006961», and by National Funds through the FCT – Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) as part of project UID/EEA/50014/2013.

#### REFERENCES

- [1] B. Boehm, “Some Future Software Engineering Opportunities and Challenges,” in *The Future of Software Engineering*, S. Nanz, Ed. Springer Berlin Heidelberg, 2011, pp. 1–32.
- [2] G. Tasse, “The Economic Impacts of Inadequate Infrastructure for Software Testing,” NIST, Tech. Rep., 2002.
- [3] DoD, “Systems Engineering Guide for Systems of Systems,” Office of the Deputy Under Secretary of Defense for Acquisition and Technology, Systems and Software Engineering Version 1.0, Tech. Rep., 2008.
- [4] AAL4ALL, “Ambient Assisted Living For All,” <http://www.aal4all.org>, 2015.
- [5] J. P. Faria, B. Lima, T. B. Sousa, and A. Martins, “A Testing and Certification Methodology for an Open Ambient-Assisted Living Ecosystem,” *Int. Journal of E-Health and Medical Communications (IJEHMC)*, vol. 5, no. 4, pp. 90–107, 2014.
- [6] C. Torens and L. Ebrecht, “RemoteTest: A Framework for Testing Distributed Systems,” in *Software Engineering Advances (ICSEA), 2010 Fifth Int. Conf. on*, Aug 2010, pp. 441–446.
- [7] B. Lima and J. P. Faria, “An approach for automated scenario-based testing of distributed and heterogeneous systems,” in *Proc. of the 10th Int. Conf. on Software Engineering and Applications*, 2015, pp. 241–250.
- [8] R. M. Hierons, “Combining Centralised and Distributed Testing,” *ACM Trans. Softw. Eng. Methodol.*, vol. 24, no. 1, pp. 5:1–5:29, Oct. 2014.