Testing of distributed service-oriented systems

Faris Nizamic

Distributed Systems Group, Johann Bernoulli Institute,
University of Groningen, Nijenborgh 9, 9747 AG Groningen, NL
F.Nizamic@rug.nl

1 Introduction

We are experiencing an exponential growth of devices connected to the Internet and services offered through the web. Today, we are just a few mobile-clicks away using services which enormously simplify our life. Just think of how we are paying our bills, recharging our mobile pre-paid account, or how we buy tickets for the events we want to attend. It is all being done through web services. This increasing reliance on distributed service-oriented systems provided through the web places a high expectation on their reliability. To keep up with this growing trend that is embracing changes on a daily basis, the software development of the services has to be rapid and at the same time leaving not much space for software errors or failures.

In contrast with that, testing of distributed service-oriented systems is still a relatively unexplored area. In fully distributed environments, the costs of integration and testing are extremely high. The situation is even worse in large scale infrastructures where there is no single owner of the system and the various elements are highly decoupled. Sometimes even observing and monitoring the whole system is difficult.

In this research, we propose a framework where simulation environments are automatically built and maintained starting from existing process specifications of individual services, as well as available choreography and service behaviour knowledge. The proposed techniques, which are integral part of the software engineering process of building large scale, loosely coupled service-oriented systems, simplify the development by automating complex testing, verification and validation, thus resulting in more cost-effective reliable software-as-a-service solutions.

This paper is structured in the following way. In Section 2, we state the problem. Then, in Section 3 we define the research questions and research methodologies. Subsequently, in Section 4 we explain initial ideas and list the published work. In Section 5, we propose solution and then in Section 6 we discuss expected impact. In Section 7, we present preliminary results and the research plan. Finally, in Section 8 we list the envisioned research outcomes.

2 Problem Statement

Today, society's increasing reliance on services provided by web applications places a high demand on their reliability. Yet, significant amount of failures is

still being found [1]. That is due to poor engineering processes, unstandardised knowledge, and poor practices in software development and software testing processes. It is evident that the complexity of the software has drastically increased since the service-oriented architecture introduced. That was because the software became distributed across the network, which made many initial assumptions obsolete (shared memory and CPU, one owner of the system, etc.). Therefore, the new rules of the game are:

1. Software is distributed. Software systems are now distributed across different physical and/or logical locations. By means of network communication between system components (web services), certain functionality is being provided to end-users of a distributed service-oriented system. Yet, the testing of software still implicitly assumed full control over a system, which is not case in the reality.
2. System components are owned by different entities. To make the situation even more complicated, each service owner, be it individual, organization or company, uses its own development and testing standards, preferred technologies and methodologies, and has its own development schedules and policies towards their service consumers. Yet, from the end-user's perspective a final bill for eventual poor quality of provided service goes to overall service integrator (entity which provides a service by composing other services).
3. Testing tools and methodologies supporting automation of the software development life cycle are insufficient. Looking at software modelling tools going all the way to software testing tools, appropriate methodologies and tools supporting the automation of complete process of software delivery are rare or non-existing. That made communication and execution of tasks very difficult and imposed many issues for which the software engineering community did not find appropriate solutions yet.

3 Research questions and methodology

The main research question we therefore state is: *How can we simplify and automate testing of complex highly distributed and loosely coupled service-oriented systems to make them more reliable, robust and error-free?*

To answer this research question, a number of challenges needs to be addressed:

- **RQ1:** How to identify the constraints that limit the output of development, testing and integration processes for distributed service-oriented systems?
- **RQ2:** Which approach and techniques to use in order to tackle the identified constraints?
- **RQ3:** How to automate the creation and management of a testing environment?

In order to answer to **RQ1**, we plan to observe testing process of the newly developed distributed service-oriented system and that way to identify the constraints that industry encounters (high dependencies on external testing environment, not available test data, etc.) that slow the development process. To get

a realistic state from the software engineering field, besides literature research a survey may be conducted among the companies or organizations that are concerned with development and testing process of such service-oriented software systems.

In **RQ2**, we concentrate on the observed constraints from **RQ1** that may be solved or whose effects can be minimized using our proposed approach and techniques. Namely, using service virtualization (web service simulation) to mimic system components are not owned or under control of test performing body. Thus, within **RQ2** we construct models formally describing simulated services, define approaches for their automated creation and maintenance, and their validation and verification against service description or against real services. Additionally, we plan develop new testing techniques which can result in more cost-effective reliable service-oriented systems. Energy and Banking sectors are the main application fields for the project.

In **RQ3**, we focus on the automation process of fully integrated environment for simulation and testing of service-oriented systems. To achieve full automation, we combine and/or customize existing tools and techniques. Once automation is achieved, we evaluate the proposed solution by comparing expected and observed improvements, that may be reflected in energy savings (Energy sector), time or money savings (Banking sector).

4 Related work

In the paper *Testing for Highly Distributed Service-oriented Systems using Virtual Environments* [2], we show the risks of performing tests in production environment and propose how testing can be enabled using simulated services (virtual environment). To illustrate how this process, we use an example of a real system - WMO, based on the Dutch law for supporting people that have a chronic disease or disability.

Subsequently, in the paper *Policy-Based Scheduling of Cloud Services* [3], we address the questions of optimal scheduling for different partners competing for the available computing resources.

Finally, we propose a service-oriented system for making buildings more energy-efficient (*BernoulliBorg - The building of sustainability* [4]). The proposed project was awarded with a grant by the University of Groningen (Green Mind Award) and currently being implemented. This project being a source of valuable experience and inspiration, through which we have an opportunity to observe more constraints that encounter during development and testing of a service-oriented system and to experiment with possible solutions.

5 Proposed solution

Taking into consideration diversity of work done until now, the integration of the previous work with current work will be done with the following approach. Firstly, we will finalize the implementation of the mentioned service-oriented

system in order to gain the valuable knowledge from practice and to prepare the environment for experimentations to follow. Besides our own system, we will analyse other existing systems, not only to understand issues in external systems (i.e. banking applications), but also to understand differences and similarities among them. Furthermore, we will describe in detail how, by using the logic of theory of constraints [5], we can identify and remove the observed constraints. The evaluation will be done on our developed system [4], and potentially on one external system from industry. That way, we will gain the necessary insight and knowledge by going through the software development life cycle (SDLC) from the inside (by developing a service-oriented system), and from the outside (by observing a system developed by other company or organization).

Using gathered data, we propose a solution to solve some of the identified constraints in service design-time. To solve the identified constraints, we use use an approach of modelling and simulation techniques [6], [7] to substitute non-owned, non-developed or simply blocking parts of the system. That way, service under test can be properly tested before being deployed to production environment.

In our work, we first define the terminology of simulated services, explain the concepts and the processes around it. Subsequently, we propose the ways to create and model the simulated services and then we propose how to validate modelled services against the real services or against service description (interfaces and behaviour). Once modelled and validated, we show how executable simulated service should be modelled, maintained and deployed to both a design-time environment (for internal service consumers) and a run-time environment (for external service consumers). Further, we define a simulated service life cycle (SSLC), make a parallel with the SDLC, and propose how development cycles can be reduced and speed-to-market increased using service simulation. Finally, we introduce the a new testing technique, namely *Environment-based testing*, and present how it can make services under development to be more robust and error-free.

Finally, our proposed solution will be a process or technique for decoupling dependent services by providing simulated environment that supports automated changes. As a side effect, automated scripts or combined software solution supporting the process will be developed.

6 Expected impact

This research will have impact on several different fields, namely: global service-oriented research community, software engineering industry, as well as local community in the North of The Netherlands.

The impact on the global service-oriented research community will be reflected through the fusion of simulation, system modelling and production management knowledge with the software engineering knowledge. The expected results of this fusion are that techniques, methodologies and approaches of service

development will be proposed to reduce very high the complexity of service composition and integration.

Meanwhile, the impact on the global software engineering industry will be represented in decrease of time-to-market of service-oriented software products. This will be done by inclusion of simulated test environments auto-deployment within the overall test automation process. That way, there will be no constraints to use all currently unused time for automated testing in highly decoupled service-oriented systems.

Other side effects are expected as well. One of the side effects is that knowledge on automated simulation environments will suggest software engineers that concept like this can serve as a tangible communication mean to model services under development, but also to make an executable artefact which can be used for multiple purposes (e.g. documentation, traceability, environment state tracking, etc.).

Last but not least, the implementation of our energy-saving service-oriented system will definitely have a significant local impact on the University of Groningen. We expect that in case our developed system proves to fulfil the goals of energy saving, it will be deployed to more buildings of the North of The Netherlands.

7 Preliminary results and research plan

The preliminary results are showed in a proof of concept *Sustainable Buildings* service-oriented system that is, at the moment of writing of this document, being deployed at the Faculty of Mathematics and Natural Sciences building where the Distributed Systems research group is located. The pilot project includes 15 office spaces, of which there are nine private working rooms, one meeting room, one social corner, two hallways, and two restrooms. There are 15 people working at the area. Consumption measuring wireless devices are used for 42 appliances, providing the ability to measure the electricity consumption and to control the appliances. At the moment, we have available preliminary results of the experiments showing if system like this makes a building more energy efficient, and what the actual savings are.

Current focus is on completion of the implementation of the afore-mentioned proof-of-concept system. Undoubtedly, once the system is in place, our proposed approach and techniques will be tested on the developed system through a number of experiments. That will subsequently lead to data analysis and publishing of observations and findings. Final part of the project will be devoted to publishing of final results and writing the thesis.

8 Envisioned research outcomes

The main envisioned outcomes of the research project are:

- **Out1** An observation report of the constraints in software engineering process of service-oriented systems based on industrial experiences

- **Out2** Techniques, methods and theoretical foundations for an automated testing using simulated services
- **Out3** A pilot prototype of automated service-oriented system deployment and testing
- **Out4** An evaluation report of the proposed techniques and methodologies for testing service-oriented systems in different areas
- **Out5** A PhD thesis that consolidates all above-mentioned outcomes

As this work can be considered to be applied research, the experience gained may also be considered as the potential basis for setting up a commercial enterprise exploiting such unique knowledge on energy-efficient buildings. Additional exploitation may be seen in the software testing consulting with the special focus on web service providers.

Acknowledgement

The author would like to thank to Prof. Marco Aiello, Dr. Alexander Lazovik and Dr. Rix Groenboom for their invaluable support in this research project. Faris Nizamic is supported by the JoinEU-SEE grant and Green Mind Award project.

References

1. Offutt, J. and Papadimitriou, V. and Praphamontripong, U.: A case study on bypass testing of web applications. Empirical Software Engineering 2012. Springer US. 1–36 (2012)
2. Nizamic, F. and Groenboom R. and Lazovik A.: Testing for Highly Distributed Service-oriented Systems using Virtual Environments. Postproceedings of 17th Dutch Testing Day 2011. EEMCS, 23–25 (2012)
3. Nizamic, F. and Degeler, V. and Groenboom R. and Lazovik A.: Policy-Based Scheduling of Cloud Services. Journal on Scalable Computing: Practice and Experience, vol. 13, nbr. 3, pp. 187–199. (2012)
4. Nizamic F. and Nguyen, T. A.: BernoulliBorg - The building of sustainability (2012)
5. Cox, Jeff; Goldratt, Eliyahu M. The goal: a process of ongoing improvement. North River Press. (1986)
6. Sargent, R. G.: Verification and validation of simulation models. J. Simulation, vol. 7, nbr. 1 (2013)
7. Briand L. and Labiche, Y. and Wang Y.: Using Simulation to Empirically Investigate Test Coverage Criteria Based on Statecharts. In: ACM International Conference on Software Engineering. ICSE 2004, pp. 86-95 (2004)