7th Collaborative Workshop on Evolution and Maintenance of Long-Living Systems

Reiner Jung[2], Marco Konersmann[3], Eric Schmieders[4]
[2]*Christian-Albrechts-Universität zu Kiel*, Kiel, Germany, reiner.jung@email.uni-kiel.de
[3]*University of Koblenz-Landau*, Koblenz, Germany, konersmann@uni-koblenz.de
[4]*Information und Technologie Nordrhein-Westfahlen (IT.NRW)*, Düsseldorf, Germany

Long-living systems are used and changed over a period of time. *Living* means that software is subject to change in requirements and context, creating the demand to evolve the software. Long-loving software is software used for some years and continuously changed, but also software that run with little change for decades and must undergo migration to a new platform or language today.

In recent years, the EMLS workshops have dealt with the challenges of the transition between development phases for long-living software. The workshops aim is to promote the exchange on the challenges, solutions and experiences in evolution and maintenance of long-living software systems. Experiences with specific technologies and solution strategies are just as much demanded for the workshop as challenges and evaluation strategies. The focus of the workshop is on **intensive discussions**. It is a **forum** in which the participants work on common topics collaboratively. We strive for an exchange between research and industry. Therefore, accepted contributions and summaries of the discussions are published via different outlets including post proceedings and the workshop wbesite.

I. Keynote

The workshop was opened this year with a keynote by Dr. Eric Schmieders from *Information und Technologie Nordrhein-Westfahlen* (IT.NRW), the IT service provider of the federal state Northrhine-Westphalia in Germany. The keynote **Digitalisierung und Architekturmanagement** dealt with the challenges and solutions in the digitalization of administrative processes. Digitalization is a social and technological transformation. It does not only maps analog to digital processes, but also transforms processes fundamentally. German laws demand the digitalization of administrative processes within a specific time frame, affecting thousands of processes. Not only do they have to be digitally operational, but they also need to be evolvable and maintainable for a long time. This induces contraints on the development and maintenance of the software stack, including time to market, software quality and personnel availability. A good compromise between individual developed and bought software seems, therefore, to be promising for the IT.NRW. Central for the IT.NRW is a good requirement engineering processes which does identify shared functionality, optimizes for costs, development personnel, and current and future personnel for maintenance and evolution.

II. Workshop Process

This year's workshop had three accepted contributions. 16 participants attended the workshop. Each session begun with a presentations of a contribution (10-15 minutes) concluding with questions to discuss. The major part of each session was a detailed discussion on the topic at hand by the authors and participants. Each discussion was guided by a moderator utilizing state of the art discussion material (e.g., flip charts and discussion cards) to guide the discussion and to prepare the presentation of the results. At the end of each session, the discussion's results were collected and presented in a poster. All presentations in this year's workshop will be made available on the website https://rgse.uni-koblenz.de/emls/2020/.

III. Workshop Contributions

The first contribution **Model-driven Development of Evolving Secure Software Systems** by Sven Peldzsus describes 3 challenges that arose during the implementation of *GraViTY*, a framework that supports model-driven evolution for secure software systems. The first challenge are transformations between models of different granularity levels. We discussed related challenged from other fields and solution ideas for the specific challenge. The second challenge is to propagate changes in interrelated models, when trace links are lost during the initial change. We discussed possibilities to reduce the workload required by the current implementation. The third challenge is the maintenance of networks of transformations, which is a challenging task in general [2]. We discussed possible solutions, taking the pragmatics of the transformation domain into account, and considering solutions to related problems, e.g., of the database or RE domains.

Whenever an application is migrated, the development of test cases for the migrated application causes significant costs especially when test cases are developed from scratch. To tackle this issue the second contribution **Test Case Co-Migration Method Patterns** by Ivan Jovanovikj, Enes Yigitbas, Stefan Sauer, and Gregor Engels proposed method patterns to co-migrate existing test cases of the application that is subject of migration. During the workshop, we discussed two crucial questions. Firstly, we discussed on how to assess the usefulness of the proposed method patterns in four dimensions: the derivation of "usefulness" into empirical and measurable units, the preconditions of the approach's applicability, the extension of the approach towards decision

Copyright © 2020 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR-WS.org/Vol-2581/emls2020preface.pdf

transparency, and adequate evaluation setups. Secondly, we identified and discussed factors that affect the usage of change patterns for concrete application migrations. We identified over 10 factors and classified them into factors that characterize the approach itself, e.g. efforts for applying a method pattern, and factors that characterize the migrated application, e.g. code coverage of the existing test cases.

The third contribution **Maintenance of Long Living Smart Contracts** by Matthias Lohr and Sven Peldzsus discuss the role of evolution and maintenance in smart contracts, especially in the Ethereum blockchain [1]. The concept contradicts the idea of immutable smart contracts on the block chain, but has shown to be existent in practice, but not systematically. Also, existing smart contracts that try to implement evolvability, such as proxy contracts, undermine the trustworthiness of smart contracts.

The workshop concluded with a summary of the contributions and discussions. We thank all authors and participants of the EMLS'20, and the organizers of the SE'20.

IV. Program Committee

Our program committee consist of members from industry and academia. They rigorously reviewed the three contributions and provided insights regarding the discussion.

a) Moritz Balz: Ista International GmbH
b) André van Hoorn: University of Stuttgart
c) Heiko Koziolek: ABB Corporate Research
d) Klaus Krogmann: Citrix Systems GmbH
e) Dušan Okanović: University of Stuttgart
f) Volker Riediger: University of Koblenz-Landau
g) Stefan Sauer: University of Paderborn
h) Jochen Quante: Robert Bosch GmbH
i) Bastian Tenbergen: State University of New York

V. Workshop Summary Notes

Participants 16 including the authors
Contributions 3, covering topics from model-driven development, software migration and smart contracts.

References

[1] Ethereum. Ethereum: a global, open-source platform for decentralized applications. https://ethereum.org/.

[2] Perdita Stevens. Maintaining consistency in networks of models: bidirectional transformations in the large. *Software and Systems Modeling*, 19(1):39–65, 2020.