ESI
An initiative of industry, academia and TNO

NWO

MASCOT Messenger

Issue 2, July 2020

Editorial

This is the second issue of the MasCot Messenger, the newsletter that brings you news and updates on the projects in the MasCot programme, that is jointly set up by NWO-TTW and ESI (TNO).

In the first issue we presented the projects DSE and SAM-FMS. In this issue, we present the other two MasCot programme projects: “Programming and Validating Software Restructurings” and “Testing in Times of Continuous Change” (TiCToC).

You will also find the announcement of the first MasCot Programme Day in this issue.

Finally, you may have noticed some differences in the appearance of issues 1 and 2 of the MasCot Messenger. We are still tweaking the layout, and if you have suggestions on how to improve please send an email at harm.munk@tno.nl. If you missed the first issue and would like to receive it, mail to the same address.

First MasCot Programme Day on October 27, 2020 in Eindhoven.

On October 27, 2020, ESI and NWO will organize the first MasCot day. This first edition will be held in Eindhoven on the High Tech Campus.

MasCot days will be organized annually, with the purpose to exchange information and explore synergies between the four MasCot programme projects, and between the project members and ESI’s industrial partners.

This first MasCot day’s agenda and form is yet to be finalized. We strive to make this into a physical meeting, adhering to the covid-19 measures, and possibly augmented with on-line availability.

The agenda will include presentations of the research plans and their industrial relevance by the four projects. This will be followed by interactive sessions, creating a meeting place for the various projects to connect and build synergy and coherence, and discuss how to optimally benefit from the MasCot programme.

Programming and Validating Software Restructurings

Main applicant:

Prof. dr. Eelco Visser, Delft University of Technology

Co-applicant:

Prof. dr. ir. Jan Friso Groote, Eindhoven University of Technology

Research Summary

Legacy software is the result of years of accumulated investment, leading to code bases that are complex, poorly documented, and hard to modify. The maintenance effort for legacy code bases is significant. In fact, maintenance dominates the software development budget over the lifetime of a software system. To keep up the rate of innovation, software engineers should be enabled to make large-scale code restructurings efficiently and systematically to reduce complexity of the code. Modern integrated development environments such as Visual Studio and Eclipse support a fixed set of automated refactorings, i.e., transformations that improve the structure of existing code while preserving its behavior. Such refactorings are useful, but they are insufficient for the majority of maintenance restructurings, which require the ability to define custom, project specific transformations. Moreover large-scale restructurings do not have to be universally applicable, software engineers should instead be able to validate their correctness in the context of the project under consideration.

The goal of this project is to develop a language parametric framework that supports software engineers to define custom, project-specific software restructurings that can be automatically applied at scale to large code bases, and that supports the validation of the correctness of (the result of) restructurings. We will combine our expertise in language engineering and verification to develop a framework consisting of the following components: (1) A language parametric program model with graph-based semantic models including data-flow, control-flow, and name binding for executing large-scale restructurings of software projects with extraction from and rendering to program texts. (2) A domain-specific language for defining custom restructurings in terms of the concrete syntax of the language under transformation, operating on the semantic program model. (3) Techniques for validating the correctness of (the results of) restructurings, including static checks of safety properties on restructuring specifications, and post-translation validation of structural and behavioral properties of the program after restructuring. We will evaluate the framework

by instantiating it to the C++ language, and applying it to restructuring case studies from the C++ code base of Philips.

Utilisation Summary

The project will be conducted by Delft University of Technology (TUD) and Eindhoven University of Technology (TU/e) in collaboration with the Philips IGT (Image Guided Therapy) group and ESI (TNO). The proposed project follows an industry-as-lab setting at Philips IGT, which provides access to realistic industrial case studies to drive the research. A user committee will be installed, consisting of ESI, Philips (as Carrying Industrial Partner), Océ, Thermo Fisher Scientific, and Verum.

The project will iteratively develop a language parametric framework for programming and validating software restructurings, instantiate the framework to the C++ language, and make it applicable to software beyond the selected case studies. We will make these tools publicly available as open source software allowing them to be used both by other research groups, but also by companies. The project team will give demonstrations and presentations at national and international events such as conferences, ESI symposia, and special interest groups such that the existence of the methods and tools becomes known.

In addition, ESI intends to experiment with these tools in its projects with other industrial partners, such as Thermo Fisher Scientific. This guarantees the widest possible dissemination of the results in both academic and industrial environments, and helps reduce the tremendous complexity of contemporary code.

TiCToC - Testing in Times of Continuous Change

Main applicant:

Dr. ir. Jan Tretmans, Radboud University Nijmegen

Co-applicant:

Prof. dr. Mariëlle Stoelinga, University of Twente

Research Summary

Modern software systems come in many variants, customized for various users, often referred to as variability, product lines, 'mass customization', or 'series-of-one'. These systems evolve over time to adapt to changing requirements and contexts. Variability and evolution make testing of such systems a major effort, since the number of different variants and versions grows exponentially with the number of variation points.

The TiCToC project will investigate formalisms, techniques, methods, and tools, to manage and reduce the combinatorial explosion of testing caused by high variability and evolution. TiCToC will follow a model-based, compositional approach to testing: (*i*) decompose a whole system into parts; (*ii*) test these parts individually; (*iii*) reason formally under which conditions, or with which additional (integration) tests, systems built from these parts are correct; and (*iv*) use model-based impact analysis to argue which parts are affected after a change, and thus should be re-tested.

Two approaches to decomposition will be investigated: components and features. Components are the replaceable building blocks from which a system is constructed. A feature is a distinctive, user-visible part or aspect of the behavior of the whole system, i.e., a slice of user-visible system behavior, that is usually implemented across several components. Features are commonly used to describe variability. The goal of TiCToC is to develop a feature-driven and component-based testing framework, such that the above steps (*i-iv*) can be performed and argued, supported by sound algorithms and a well-defined semantics.

The methods that TiCToC aims to develop shall satisfy some additional requirements. First, considering our application domain of very complex high-tech systems, it is very unlikely that complete, monolithic, and sufficiently detailed behavioral models of whole systems and their variants can ever be developed. Therefore, the methods must be able to cope with partial information, in particular, with partial models, under-specification, and models with only a subset of features or components. Secondly, the framework shall be scalable and cope with distribution, concurrency, nondeterminism, and complex interfaces, which are ubiquitous in the domain of complex high-tech systems. Thirdly, the concepts and methods shall not only be elaborated on a formal, semantic level, but

they shall also be represented in an intuitive and human accessible way for industrial testers. Therefore, a concise, modular, user-friendly, and accessible Domain Specific Language (DSL) shall be developed. The DSL should have a clear and well-defined semantics in the formal domain.

Utilisation Summary

Systematic testing is the most important and widely used technique to verify and validate systems and to check their quality and reliability. The number of possible different variants and versions, however, grows exponentially and testing all different variants and versions is totally impossible. The aim of TiCToC is to develop methods to cope with this combinatorial explosion in such a way that not all possible variants have to be tested, and not all versions have to be completely re-tested after a modification, while still providing high, argued test coverage and confidence in the quality of the whole system. Improvements in testing systems with variability and evolution have an enormous, multi-MEURO impact on the Dutch economy, both directly, by making the work of testers more effective and more efficient, and indirectly, by faster and cheaper development of better tested products. Without such improvements, there is a great danger that future generations of ever larger and more complex cyber-physical systems cannot be tested fast and thoroughly enough, which will seriously hamper innovation.

TiCToC is a largely problem-driven project: the research topics directly emanate from problems experienced in industry. Application, utilization, and transfer of the project results are very important, and will be organized along different lines. First, TiCToC will work on case studies in close cooperation with ASML and Océ. At ASML, TiCToC will work on testing in the *Virtual Fab*, a 'digital twin', where features, variability, components (actually systems in a systems-of-systems context), component interoperability, and evolution play a role. At Océ, the first case study will concentrate on embedding feature-driven, model-based testing in their test automation framework. Second, since many high-tech companies have very similar testing problems whereas they do not compete with each other on testing, a couple of these companies have been invited to the User Committee to foster discussion, dissemination, and feedback. Also the extensive ESI network of high-tech and embedded systems companies, testing companies, and academia will be supportive in this respect. Third, to establish that the results will eventually be incorporated in professional test tools, and will be accessible for other companies, a number of companies offering software testing services will participate in the User Committee. Finally, an important vehicle for utilization and transfer will be the prototype tool set that will be developed by extending the existing, open source tool TorXakis, and that can be used freely by parties interested in experimenting with component-based and feature-driven MBT.