Excellence in Variant Testing

Holger Schlingloff
Fraunhofer FOKUS, Berlin, Germany

Peter M. Kruse
Expleo GmbH, Berlin, Germany

Mehrdad Saadatmand
RISE Research Institutes of Sweden

ABSTRACT

In this short paper, we report on the motivation, background and ambition of the ITEA3 project XIVT – excellence in variant testing. We describe a work flow and tool chain for testing of configurable and highly-variant embedded systems in various domains.

ACM Reference Format:
Holger Schlingloff, Peter M. Kruse, and Mehrdad Saadatmand. 2019. Excellence in Variant Testing. In *Proceedings of 14th International Working Conference on Variability Modelling of Software-Intensive Systems (Vamos2020).* ACM, New York, NY, USA, 2 pages.

1 INTRODUCTION

Due to increasing market diversification and customer demand, most industrial products today are available in many different variants. The increasing functionality of embedded systems makes validation and testing highly complex. For cyber-physical systems (CPS) which contain configurable or variant components, the design of adequate test suites is a major problem. In such a setting, not just one system, but many different, 'similar' systems must be tested. In a product line with, say, just 10 different features which can be present or not, there are already more than 1000 different combinations to be tested. With the increasing modularity and interconnectivity of smart devices, the variability rises as well. Incremental development and reuse of components can easily result in a system with thousands of configuration parameters. Current test design and selection methods as well as test tools fail to deliver adequate test suites for such a setting. This significantly hinders deployment and evolution of respective products in the market, especially when it comes to systems with high quality demands. Configurable and variant-rich systems are heavily being used in all domains where the innovation is mainly realized by software. The market in several industrial domains is driven by a strong demand for enhanced functionality. This includes personalization and customization, increased interconnectivity and interoperability, higher modularity, faster design cycles, and higher demands on safety and security. The resulting high complexity of software integration and communication requires capabilities for variant intensive systems testing methodologies.

In this context, the ITEA3[1] project XIVT (pronounced "shift", 2019-2022) will develop methods and tools for the design of test suites and the selection of test cases. It targets highly configurable, variant-rich embedded systems in the automotive, rail, industrial production and telecommunication domains. This enables a highly effective, cost-efficient quality assurance, allowing the shift to autonomous, flexible and adaptive applications in these domains. Base of the method is a knowledge-based analysis of requirements formulated in natural language, and a model-based test generation on the product-line level. The XIVT project will result in modules and open-source packets for the enhancement of existing domain-specific testing tools, a reference implementation of the resulting method and toolchain, and an ecosystem of services around the proposed methodology.

[1]ITEA is a transnational and industry-driven Research, Development and Innovation programme in the domain of software innovation, involving around 40 countries globally: https://itea3.org/about-itea.html

2 CHALLENGES AND APPROACHES

Within the project, the following challenges are being addressed.

- **Requirements hierarchies:** Defining appropriate requirements for a product can be a complex task, depending on the system under analysis. The increase of functionality allows new variants of a product base, changing the system requirements in a such way that a product-based requirements analysis takes excessive time. In a family-based approach, a set of products with similar features and requirements must be analysed. We propose to use a hierarchical structure of system requirements with different levels of abstraction. Common domain requirements and their correlation are identified by a schematic analysis of several use case documents. Individual product requirements are associated with distinguishing features of a product. Thus, we define a hierarchy of system requirements and correlations between them, which is abstracted from the application domain. This hierarchy allows to characterize different concrete products and create test cases accordingly.
- **Knowledge-based test optimization:** Industrial product lines are characterized by a myriad of features that can be associated with several system requirements each. Thus, analysing such products by humans is difficult or even infeasible, especially with highly configurable products and systems. We propose to deal with this complexity by automatically processing the relations between features and requirements, using machine learning techniques and map-reduce algorithms. We use different forms and combinations of processing big data, e.g., data mining and clustering algorithms for the aggregation of similar features / requirements. Also, we are considering whether natural language processing can be applied to a chain of requirements in order to find out which features are involved. The result is an automatic prioritisation and ranking of requirements for test case selection.
- **Abstract test generation:** Test cases validate a set of functionalities of a product with respect to a chain of requirements. Test generation for configurable products is challenging since each concrete product realizes a subset of all features, but not all combinations of features are realized. For testing product lines, however, it is mandatory that all features are thoroughly tested. We propose to generate abstract test cases, based on feature models and the hierarchy

Vamos2020, February 5–7, 2020, Magdeburg, Germany
2019.

of requirements. These test cases are built taking into account the selected and prioritized requirements, components and features. Our method automatically derives abstract test cases from base models and feature models, which can be instantiated for concrete configurations and products.
- **Test assessment:** The quality of a test suite is its potential to detect errors. Assessing a test case for a given product in a specific domain means to check if the test is able to verify the system requirements and product features for which it was designed. We propose to create tests following the abstraction levels defined above, but under the premise that they have to fit in a real environment and are able to test the product. Then, test monitoring is used to assess the generated test suites based on requirements and features, and the specific product under test.
- **Security testing:** Despite several techniques used to test the security of developed software, such as fuzzing / attack injection and static and dynamic analysis, there is not a single technique able to cover all software security aspects and discover all software vulnerabilities. Therefore, we propose an integrated approach, where regression and mutation techniques are applied to identify vulnerabilities in software. We design a 'cluster' of security tools that employ a combination of software testing techniques and machine learning techniques, for processing large amounts of data and helping in the detection of software flaws and its correction.
- **Orchestration platform for variant testing:** There currently is no end-to-end automation platform and tool chain for variant testing that is readily available and easy to use. Therefore, we are bringing together the various innovations, approaches and techniques described in this paper in a holistic manner to make it easy for industrial users to utilize the outcome of our project. The intention is to ensure that there are no market barriers to the adoption of the developed innovations and resulting outcomes. The orchestration platform is based on Eclipse, integrates various of the tools mentioned above, and is open for extensions and customizations. The core tool chain will be available in the public domain, with additional tools being commercially licensed.

3 WORKFLOW AND TOOLCHAIN

In this section, we sketch the workflow and toolchain which currently is under development within the project. Figure 1 shows the overall workflow with respect to the different techniques that are considered in the project to answer the variability testing challenges of XIVT industrial partners and domains. The implemented techniques and developed solutions in the project are then encapsulated and provided as services that constitute the overall project framework and tool chain, referred to as project reference platform. These services are categorized as: core services which are common and needed in all of the domains and use-cases of the project, and also customized services (e.g., as extensions and tailored versions of the core services) to address specific needs of use-cases from the four target domains of the project; namely, automotive, railway, industrial production, and telecommunication.

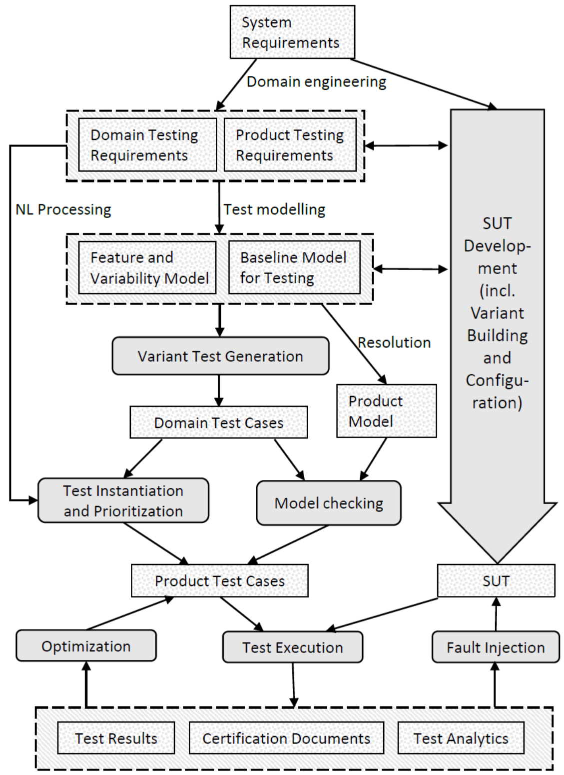

Figure 1: Workflow and toolchain

From another perspective, the services in the XIVT reference platform can also be categorized as main versus supporting services. For instance, the services related to test optimization in XIVT currently include services for test case selection and test case prioritization (as main services), which are supported by similarity analysis and change impact analysis services as supporting ones. Similarity analysis is needed to identify features and components that are re-used across different projects and products. The result of this analysis will then help towards making optimal decisions for selection and prioritization of test cases. In a similar fashion, change impact analysis is also required to optimize the use of testing resources in variant-intensive products where it is needed to determine which features to choose to test as test objects. For instance, if a fault is detected in a component in the product baseline, we need to decide if it is necessary to (re-)execute the test cases for each product variant, a subset of variants, or only the baseline. Change impact analysis becomes also important to make optimal decisions for regression testing when a bug is fixed in one product variant or in the baseline.

More information on the XIVT project and the results mentioned here can be found at https://www.xivt.org/.

Acknowledgement: We would like to acknowledge the work of all authors who contributed to the XIVT proposal, notably Peter Watkins, Ibéria Medeiros, Eduard Enoiu, Noel Kirthiraj, Thomas Bangemann, Jan Krause, Heiko Dörr, Ferhat Erata, and others.