Empowering Testing Activities with Modeling
Achievements and Insights from Nine Years of Collaboration with Cisco

Shaukat Ali[1,*], Marius Liaaen[3], Shuai Wang[1] and Tao Yue[1,2]
[1]*Simula Research Laboratory, Oslo, Norway*
[2]*The University of Oslo, Oslo, Norway*
[3]*Cisco Systems, Oslo, Norway*
{shaukat, shuai,tao}@simula.no, marliaae@cisco.com

Keywords: Testing, Modeling, Video Conferencing Systems, the Unified Modeling Language, the Object Constraint Language, Feature Model.

Abstract: Research-Based Innovation (RBI) aims at bringing research-driven innovative solutions to the industrial problems identified from the industry in close collaboration with researchers. This paper focuses on presenting one such instance of RBI between Cisco Systems, Norway and the Software Engineering department of Simula Research Laboratory, Norway over the period of last nine years. The main topic of the collaboration was related to improving the current testing practice at Cisco with the use of models. We present a brief overview of various model-driven testing projects, and lessons learned from such RBI collaboration from researchers' perspective.

1 INTRODUCTION

To observe a quicker impact of research in the industry, the research problems must be identified directly from the industry and innovative solutions to those problems must be developed in a close collaboration with the industry. Such an approach referred as Research-Based Innovation (RBI) is applicable to a wide variety of engineering disciplines including software engineering as highlighted by Briand et al (Briand et al., 2012).

In this paper, we present our experience, results, and lessons learned from a long-term collaboration of such RBI of Simula Research Laboratory (SRL), Norway with Cisco Systems, Norway over the period of last nine years. More specifically, we focus on projects of testing that aimed at solving problems whose solutions required the use of models. The branch of Cisco we collaborate with was previously known as Tandberg—the world leading Norwegian company developing high-quality Video Conferencing System (VCSs). Tandberg was later bought by Cisco and their core business still focuses on developing a wide variety of hardware and software-based (including cloud-based solutions) telepresence devices, such as VCSs.

First, we present a brief overview of mode-driven testing projects followed by commenting on our collaboration by presenting some of the lessons learned. Notice that the purpose of this paper is to summarize our collaboration with Cisco and not to present the new unpublished research. Thus, we refer to appropriate references where readers can find the details on the projects and research results. Furthermore, experiences and lessons learnt are solely from researchers' perspective.

The rest of the paper is organized as follows: Section 0 presents an overview of our collaboration with Cisco including various modeling notations we tried in different projects. Section 3 presents the projects related to model-drive test generation, Section 4 presents projects for model-driven product line testing, and Section 5 provides an overview of various model-driven test optimization projects. Section 6 presents some research work based on restricted natural language-based notations that had a comprehensive metamodel behind it. Section 7 presents lessons learned from our collaboration. Finally, we conclude the paper in Section 8.

[*]All the authors are alphabetically ordered.

Ali, S., Liaaen, M., Wang, S. and Yue, T.
Empowering Testing Activities with Modeling - Achievements and Insights from Nine Years of Collaboration with Cisco.
DOI: 10.5220/0006216105810589
In *Proceedings of the 5th International Conference on Model-Driven Engineering and Software Development (MODELSWARD 2017)*, pages 581-589
ISBN: 978-989-758-210-3

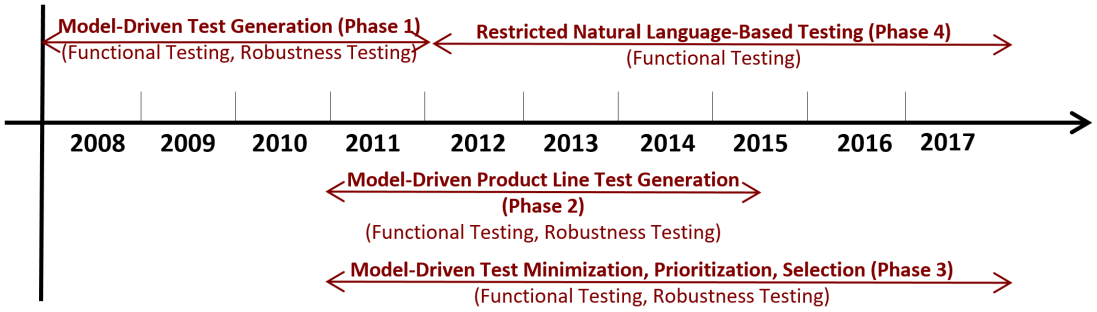

Figure 1: A timeline of collaboration with Cisco.

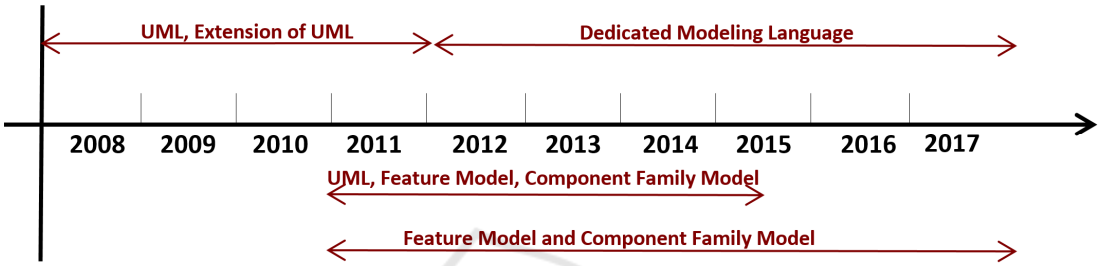

Figure 2: A timeline of modelling languages tried.

2 AN OVERVIEW OF THE COLLABORATION

The research scientists from the *Software Engineering* department of Simula Research Laboratory[1] (SRL), Norway initiated collaboration with Cisco Systems[2], Norway in the beginning of 2008. The tagline for the Software Engineering department was *"The Industry Is Our Lab"* (Sjøberg, 2010) and thus the interest was to build long-term collaborations with industry on various topics in software engineering. The research interests of the sub-group in the software engineering department were on verification and validation of software-intensive systems. The collaboration initiated with one of the testing groups at Cisco focused on testing software for one of the Video Conferencing Systems (VCSs) being developed at Cisco. The testing team was led by the test manager (the second author on this paper), who is still after nine years the main contact of the collaboration of SRL with Cisco. Later on, in 2011, the Cisco and SRL's collaboration was further strengthened, when a new verification and validation center was established at SRL with Cisco as one of the key user partners. The center is named as Certus Software Verification and Validation Centre[3] and has a duration of eight years and Cisco has been actively participating in several sub-projects in the center.

An overview of the timeline of our collaboration is shown in Figure 1. The timeline shows various phases of model-driven testing projects together with Cisco ranging from model-driven test generation (Phase 1), model-driven product line testing (Phase 2), model-driven test optimization (Phase 3), and requirements modelling-driven test generation (Phase 4). Over the time of collaboration, we tried out a variety of modeling notations for the above-mentioned testing problems as shown in Figure 2 including UML-based models, feature models, and dedicated natural language based modeling notations. Based on the timeline of the collaboration, we will provide a brief introduction to the testing problems we solved (Section 3 to Section 6) followed by key lessons learned from our collaboration in Section 7.

[1] www.simula.no
[2] http://www.cisco.com/c/no_no/index.html
[3] www.certus-sfi.no

3 MODEL-DRIVEN TEST GENERATION (PHASE 1)

Our initial collaboration with Cisco was focused on the use of models to improve the current testing practice. We set the objective of collaboration at a very high level with the ultimate aim of working together with Cisco to elicit interesting research-based innovation problems that we can solve together. Based on several meetings and workshops, we decided to automate the generation of test scripts based on the models since test scripts were manually coded by test engineers at the time of collaboration. The test scripts were then executed using the already developed test execution system.

This very first project included two Ph.D. students, a master student, and two scientists from SRL. Initially, the two Ph.D. students and the master student worked together to develop models and model-driven test case generation tool, called TRUST(Ali and Hemmati, 2014, Ali et al., 2010) to support functional testing of a Videoconferencing System (VCS) called as *Saturn*. The aim was to set up a base to build the Ph.D. theses on. The master student's thesis was focused on the development of the tool.

Once the support for functional testing was established, two parallel tasks were initiated. Each Ph.D. student exclusively worked on his respective tasks; however, there was intense collaboration among the topics. Given the fact that a large number of test cases can be generated from models, our natural need was to devise scalable methods to reduce the number of test cases to be executed without compromising their effectiveness. This was the focus of the first thesis (Hemmati et al., 2013). The second thesis focused on testing the robustness of the VCS in the presence of a variety of faults in the environment (Ali et al., 2011a, Ali et al., 2012a).

During the first phase of collaboration, the focus was testing one system at that time; however other VCSs were used with the System Under Test, i.e., *Saturn* to assist test execution. Our choice of modeling was the Unified Modeling Language (UML) and more specifically, we used UML class diagrams to model testing APIs of *Saturn* that was implemented in the system and was used to automate testing. UML state machines were used to model the expected behavior of the system based on the APIs defined in the class diagrams. For automated test data generation, constraints in the Object Constraint Language (OCL) were modeled on the transitions as guard conditions that were solved automatically to generate data to fire transitions (Ali et al., 2014, Ali et al., 2011b, Ali et al., 2012b). Test oracles were also implemented as OCL constraints that were used to check actual system states during the test execution with the ones specified in the models(Ali et al., 2011a, Ali et al., 2012a).

Our in-house prototype, i.e., TRUST (Ali and Hemmati, 2014, Ali et al., 2010) was developed that take the models developed as input and generated test scripts that can be executed to test the system. TRUST had several other integrated tools including state machine flattener to flatten hierarchy, test data generation from OCL constraints, and runtime constraint checking (Ali and Hemmati, 2014, Ali et al., 2010, Ali et al., 2014, Ali et al., 2011b, Ali et al., 2012b).

Test ready models in UML were created together with a research engineer for *Saturn* and were discussed with several others and the test manager (Ali et al., 2011a). This part of collaboration was research-oriented and eventually results were transferred only in the form of knowledge.

4 MODEL-DRIVEN PRODUCT LINE TESTING (PHASE 2)

The *Phase 1* collaboration was focused on testing only one system at a time. Based on further investigation, we discovered that Cisco develops product lines of VCSs and thus by applying methodologies from Product Line Engineering (PLE), we can enable systematic reuse of test ready models for the products belonging to a product line instead of creating models from scratch for each product to be tested. However, such reuse comes at the expense of a methodology with the tool support to configure models for each product.

Our first attempt (Ali et al., 2012c) was developing configurable UML class diagrams, UML state machines, and their associated OCL constraints (product line models). On the top of the models, we developed a methodology with the tool support (Ali et al., 2012c) that allows to configure the models for each new product and then use the configured models for test case generation using the existing version of our prototype test case generation tool, i.e., TRUST (Ali and Hemmati, 2014, Ali et al., 2010).

Over the time, we further learned that the test engineers were reluctant in using UML models. They rather prefer coding test scripts, and thus we decided to further raise the level of abstraction of the product lines models, where we introduced Feature

Models as the main frontend to select and configure UML models for a particular product. Such an approach (Wang et al., 2013b) hides the UML Models from users (e.g., test engineers in our case) who rather focuses on the Feature Models. All the configuration of models took place using the feature model, where the users only need to select relevant features followed by selecting various values of attributes and the result was configured UML models for a VCS product (Wang et al., 2013b) to be used for test case generation.

5 MODEL-DRIVEN TEST OPTIMIZATION (PHASE 3)

Motivated by the industrial needs of testing product lines of VCSs coined as *Saturn*, we further managed to identify three test optimization problems together with the test engineers at Cisco in the context of product line testing, which included: 1) Test case selection; 2) Test suite minimization and 3) Test case prioritization. Our previous works have addressed the above-mentioned test optimization problems by employing systematic and automated approaches with models (e.g., feature models) and search algorithms (e.g., genetic algorithms) (Ali et al., 2009). We briefly present each problem along with proposed approaches as below together with the references.

5.1 Model-Driven Test Case Selection

The key goal for test case selection was to automatically and systematically select a subset of test cases for testing a particular VCS product from the test suite developed for testing the entire product line. To achieve this goal, we proposed an automated test selection approach using feature model together with the test engineers at Cisco (Wang et al., 2013a, 2016a, 2015). More specifically, a Feature Model for Testing (FM_T) was defined to capture commonalities and variabilities across a VCS product line while a Component Family Model for Testing (CFM_T) was designed to model the test case repository that is used to test the entire product line. Moreover, restrictions were employed to link FM_T, CFM_T and test cases in the repository and thus test engineers are able to perform automated test case selection at a higher level of abstractions (i.e., through FM_T) using our approach.

5.2 Model-Driven Test Suite Minimization

The aim of test suite minimization was to eliminate the redundant test cases from the selected test suite (obtained from the test case selection activity) in order to 1) reduce the cost of testing (e.g., execution time) while 2) preserve high effectiveness (e.g., coverage of testing functionalities modeled as features in the feature model). With such goal in mind, we formulated such test suite minimization problem as an optimization problem followed by proposing a search-based test minimization approach (Wang et al., 2014a, 2013a). The approach defined a fitness function that took five cost-effectiveness objectives (e.g., fault detection capability) into account and integrated the fitness function with a multi-objective search algorithm (e.g., Random-Weighted Genetic Algorithm) with the best performance based on an extensive empirical evaluation. In addition, a tool named TEst Minimization using Search Algorithms (TEMSA) was also designed and implemented for supporting test suite minimization activity using our search-based approach.

5.3 Model-Driven Test Case Prioritization

The goal for test case prioritization was to schedule the minimized test suite (obtained from the test suite minimization activity) into an optimal order before executing with the aim to achieve high effectiveness (e.g., fault detection capability) as early as possible with the minimum cost (e.g., execution time). To address such challenge, a search-based multi-objective approach was proposed to prioritize a given set of test cases in a cost-effective manner when there is a limited budget (e.g., time and test resources) (Wang et al., 2014b, 2016b, Pradhan et al., 2016, Wang et al., 2016c). The approach defined a fitness function by considering multiple cost and effectiveness measures (e.g., resource usage and feature pairwise coverage on the features modeled in the feature model). Moreover, the approach empirically evaluated more than ten multi-objective search algorithms and integrated the best one for prioritizing test cases in practical applications.

Notice that the above-mentioned approaches have been proposed together with test engineers at Cisco with the aim to cost-effectively testing VCS products in the context product line. It is worth mentioning that the proposed approaches are also applicable separately in other contexts (e.g.,

regression testing).

6 TEST CASE GENERATION FROM REQUIREMENTS-BASED TEST SPECIFICATION (PHASE 4)

In this approach, we made another attempt to automate test script generation from a dedicated test case specification language, called Restricted Test Case Modeling (RTCM) (Yue et al., 2015). With RTCM, one can specify test case specifications using restricted natural language. The RTCM specification language is composed of a set of keywords (e.g., VERIFIES THAT) for the purpose of reducing ambiguities inherent in natural language based specifications and facilitating automated analyses and generations such as test scripts. RTCM specifications are automatically formalized into instances of TCMeta, the metamodel designed particularly for the RTCM methodology. From TCMeta instances, test scripts can be automatically generated.

In the context of this research stream, we have developed an editor dedicated for specifying test case specifications. With this editor (with the RTCM test case specification template implemented), one can directly embed testing APIs of the system under test into test case specifications by simply making a selection from a drop list that lists all the available APIs, while typing. Using the easy-to-use template provided with RTCM, test specifications are specified in natural language combined with testing APIs related to making calls, checking state variables, and configuring a VCS. Once the test case specifications are written, our test generator (integrated with the RTCM editor) automatically outputs test scripts using various coverage criteria such as *All Branch* and *All Condition* coverage criteria (Yue et al., 2015).

This work is ongoing and is mostly of research nature with a prototype implementation of the editor and generator. It has been validated on the VCS systems at SRL available for experimentation and provided by Cisco.

7 LESSONS LEARNED

In this section, we present a set of lessons learnt extracted from the fruitful collaboration with Cisco. The aim for presenting these lessons learnt is to provide some guidelines for the future practitioners who plan to conduct industry-oriented research.

7.1 Willingness of the Industrial Collaborators to Try out New Things is Critical for Research-based Innovation

Based on our nine-year collaboration with Cisco, we observe that the willingness of learning new knowledge plays a key role in the adoption of research findings (e.g., approaches and prototype tool) into industrial practice. In the worst case, the door of collaboration would be closed if industrial companies are not open and do not like to depart from what they are already doing. In our case, we are very fortunate to have Cisco as our industrial partner who always expresses a strong desire to receive new knowledge that could be used for improving their testing practice. More specifically, one test quality assurance manager and around 5-6 test engineers have been involved in our collaboration and these people have shown high degree of openness and motivation for acquiring diverse theories and developing innovative approaches. For instance, when the collaboration started in 2007, there were no concepts of model-driven testing (even though some models were used sometimes for illustrating concepts, e.g., test cases) or product line testing (even though several VCS products already existed at that time). The test engineers at Cisco were also not aware of the state-of-the-art with respect to these research fields. With our collaboration, one of the key outcomes is that several concepts have been well introduced along with the proposed approaches throughout the testing group, e.g., model-drive test case generation and test case selection using feature models. Notice that the test manager/test engineers who we are collaborating with managed to 'advertise' the knowledge and approaches they gained to the whole company through department meetings or workshops and thus more practitioners at Cisco have opportunities to learn fresh knowledge that have potentials to be employed for solving other challenges. From the long-term view, such learning circle is essential for companies to be innovative and competitive since a diversity of knowledge is always brought in.

7.2 Productive Collaborations Require Collaborators Holding a Long-Term Vision

Another of our key observations is that it is crucial

for industrial collaborators to have a long-term vision when there is an ambition to mature research findings into practical applications. Such observation is extracted due to several factors based on our experience with Cisco. The first factor is that it usually takes years to successfully deploy research achievements (e.g., approaches and prototype tools) into the industry. With respect to model-driven testing, it even takes more time since models (e.g., UML and feature model) are not well known in industry and practitioners have to first acquire relevant modeling knowledge (e.g., UML and feature models) before digging into specific approaches and tools. Therefore, it requires that industrial companies have a long-term vision for future markets when establishing collaborations with research institutions. For instance, Cisco has a strong ambition to fully automate their testing process of VCS products (which is a very challenging goal) and thus they are willing to investigate significant effort on research that can be employed for their ambition. In the past nine years, on average two meetings (around 1.5 hours) per month have been arranged between researchers in SRL and engineers at Cisco for discussing potential challenges, designing and revising approaches with tool support. With such large amount of effort we spent, a set of testing challenges have been identified faced by Cisco (e.g., test case generation) and a number of cost-effective model-based approaches have been proposed and developed together with tool support (Section 3 to Section 6). We have ambitions to mature all these research approaches into the current testing practice of Cisco with the aim of improving the efficiency of testing.

7.3 Seeking 'champions' in the Industry to Support Collaboration

In most of the industrial companies, technical employees (e.g., test engineers in our case) do not hold the power to make final decisions for the adoption of new approaches or tools. Such decisions usually need to be submitted and discussed through the level of the management team. Based on our experience, it is also infeasible for technical staff to decide and establish collaboration between research institutions (e.g., SRL) and industrial companies (e.g., Cisco). Such process also requires involving one or more high-level persons from the management team. Therefore, it is of paramount importance to seek champions from industry who can play as backbones for strongly supporting during the entire process of collaboration, e.g., identifying problems and realizing solutions. Such champions usually hold several key characteristics including 1) being highly motivated for improving the current practice; 2) being open to new knowledge and techniques; 3) having a high influence on the decisions of the management team; and 4) working stably in the company for a long time. In our case, we are lucky in having such strong champion at Cisco, who plays as test quality assurance manager and have been working in the testing department of Cisco for more than 15 years. He is always motivated to tackle difficult challenges that exist in the VCS testing process, has an open mind for learning new theories/approaches and is active to provide us with real testing dataset used for evaluating the research approaches. In general, finding such champion is essential for research-industrial collaboration to produce useful outcomes that can have an impact on practical applications.

7.4 Efficient Teamwork with Good Attitude are the Key Factors for the Success of Collaboration

This lesson learned underlines the importance of efficient teamwork and good attitudes for both researchers and industrial collaborators. More specifically, efficient teamwork and good attitudes refer to two angles. First, researchers and engineers (e.g., test engineers in our case) should communicate and understand each other in a good way. This is particularly important since researchers and engineers sometimes use different terminologies, which could mislead the understanding (e.g., for problems or approaches). In our case, we did a mapping job to unify the terminology between research and industry, which will be used throughout the collaboration. For instance, we tried to map the term *feature* in feature model to the term *testing functionality* at Cisco and thereby avoiding misunderstanding of terminology. Second, efficient teamwork and good attitudes denote that both researchers and engineers should be willing and active to work together. From the perspective of research, researchers need to be positive for communicating with industrial practitioners, identifying potential challenges, proposing approaches and revising the approaches based on the feedback provided by industrial practitioners. From the view of the industry, practitioners should be active to learn new knowledge (e.g., models), help researchers to realize the proposed approaches, provide real dataset (e.g., VCS testing results) for empirical evaluation and eventually deploy the

approaches into the practice. Notice that all these activities required by researchers or industrial practitioners have a lot of overlap, which requires an efficient teamwork and optimistic attitudes from the both sides. In our case, researchers from SRL frequently communicate with test engineers at Cisco with different means (e.g., video call, physical meetings and workshops) to ensure that challenges have been identified and addressed in a proper way. Test engineers also like to share ideas and provide their valuable feedback to further improve the approaches. Sometimes, both researchers and test engineers work together for the integration of the research approaches. All these activities can guarantee the collaboration is productive and fruitful from a long-term perspective.

7.5 Introducing Modeling Notations to Industry in a Smart Way

This lesson learned emphasizes that industrial practitioners usually prefer simple modeling notations in terms of learning and employing model-driven testing approaches. Based on our experience, modeling notations with high complexity would make industrial practitioners get lost and lose their interests in a short time and thereby resulting in a failure of productive collaboration. In our case, when introducing UML notations to Cisco, we first investigated what would be the sufficient subset of UML modeling nations for VCS testing based on several discussions with the test engineers at Cisco. We observed that most of the UML notations were not needed for VCS testing and therefore we only chose several key notations (e.g., classes and state machines) and presented to the test engineers. With such way, we found that the test engineers were able to acquire the UML modeling notations in an efficient way and some of them even tried to model VCSs with these notations by themselves.

In terms of variability modeling, we chose feature model (Section 5.1) since 1) the notations of feature model are sufficient for capturing the commonalities and variabilities of VCS product line, and 2) the notations of feature model is simple and quite easy to understand by the test engineers even without modeling background. We also tried to map the notations of feature model to the terminology used in VCSs, which again made the modeling notations easily understood. In our previous work (Wang et al., 2016a), we also conducted a controlled experiment for soliciting the views of the experts of VCS testing (e.g., test manager and test engineers) in terms of understanding and using the modeling notations of feature model. The results showed that industrial practitioners at Cisco was able to gain a good understanding on the modeling notations we introduced and they were positive to employ our proposed model-driven testing approaches (Section 5.1) into their practice.

7.6 User-friendliness of a Tool is a Key Factor for the Success to the Adoption of Research Results

One of the major gaps between academics and industry is that academics usually do not put particular focus on tool support, which is, however, the key concern from the perspective of the industry. To bridge such gap, researchers should push themselves to spend more effort on maturing research approaches into tool support, which can be eventually employed by industrial practitioners. In our case, we designed and developed a set of user-friendly tools together with the test engineers at Cisco, which integrated the proposed approaches, e.g., TRUST (Ali and Hemmati, 2014, Ali et al., 2010) for model-based test case generation (Section 3), TEMSA (Wang et al., 2014a) for model-driven test suite minimization (Section 5.2). Based on our experience, we also notice that industry usually prefers to use commercial tools instead of open-source tools. For instance, there are currently a number of tools available for feature modeling, e.g., FeatureIDE[4] and pure::variants[5]. After we discussed with the test engineers at Cisco, we decided to choose pure::variants for feature modeling since: 1) it is a commercial tool that holds good stability and reliability; 2) it has a good and friendly user interface that can be learned and used quickly; 3) it supports plugin development and we can implement our functionalities on the top of it; and 4) technical support is usually efficient for commercial tools.

8 CONCLUSION

In this paper, we summarized and reported our nine-year collaboration between the *Software Engineering* department (Simula Research Laboratory, Norway) with Cisco Systems, Norway, which focus on employing models to cost-effectively test Video Conferencing Systems (coined as model-driven testing). With such long-term productive collaboration, we managed to gain

[4] http://fosd.de/fide/
[5] http://www.pure-systems.com

achievements (with numerous research publications and tool support) from four angles together with industrial practitioners at Cisco, which include: *model-based test generation*, *model-driven product line testing*, *model-driven test optimization* and *test case generation from requirements-based test specification*. With respect to each angle, we reviewed the challenges identified together with approaches (with tool support) proposed to deal with these challenges. Furthermore, based on our experience, we extracted and shared a set of lessons learned from researchers' perspective with the aim of providing guidance to future practitioners who plan to work on industrial-oriented research, particularly for the research related with model-driven testing.

It is also worth mentioning that the collaboration between Simula with Cisco is still ongoing and will be continued from a long-term perspective in the future. We are currently working together to address new challenges. We believe that the outcomes from such collaboration (i.e., research-based innovation) will be beneficial to both academics and industry.

ACKNOWLEDGMENTS

Tao Yue and Shaukat Ali are supported by RCN funded Zen-Configurator project, the EU Horizon 2020 project Testing Cyber-Physical Systems under Uncertainty, RFF Hovedstaden funded MBE-CR project, RCN funded MBT4CPS project, and RCN funded Certus-SFI. Shuai Wang is supported by RFF Hovedstaden funded MBE-CR project and RCN funded Certus-SFI.

REFERENCES

Ali, S., Briand, L., Arcuri, A. & Walawege, S. 2011a. An Industrial Application Of Robustness Testing Using Aspect-Oriented Modeling, UML/MARTE, And Search Algorithms. *ACM/IEEE 14th International Conference On Model Driven Engineering Languages And Systems (Models 2011).*

Ali, S., Briand, L. C. & Hemmati, H. 2012a. Modeling Robustness Behavior Using Aspect-Oriented Modeling To Support Robustness Testing Of Industrial Systems. *Software And Systems Modeling,* 11**,** 633-670.

Ali, S., Briand, L. C., Hemmati, H. & Panesar-Walawege, R. K. 2009. A Systematic Review of the Application And Empirical Investigation of Search-Based Test Case Generation. *Ieee Transactions On Software Engineering,* 99.

Ali, S. & Hemmati, H. Model-Based Testing of Video Conferencing Systems: Challenges, Lessons Learnt, And Results. 2014 IEEE Seventh International Conference On Software Testing, Verification And Validation, March 31 2014-April 4 2014 2014. 353-362.

Ali, S., Hemmati, H., Holt, N. E., Arisholm, E. & Briand, L. 2010. Model Transformations As A Strategy To Automate Model-Based Testing - A Tool And Industrial Case Studies, *Simula Research Laboratory, Technical Report* (2010-01).

Ali, S., Iqbal, M. Z. & Arcuri, A. 2014. Improved Heuristics For Solving OCL Constraints Using Search Algorithms. *Proceedings of The 2014 Conference On Genetic and Evolutionary Computation.* Vancouver, Bc, Canada: ACM.

Ali, S., Iqbal, M. Z., Arcuri, A. & Briand, L. 2011b. A Search-Based OCL Constraint Solver for Model-Based Test Data Generation. *Proceedings of The 11th International Conference on Quality Software (Qsic 2011).* IEEE Computer Society.

Ali, S., Iqbal, M. Z., Arcuri, A. & Briand, L. 2012b. Generating Test Data from OCL Constraints With Search Techniques. Simula Research Laboratory.

Ali, S., Yue, T., Briand, L. & Walawege, S. 2012c. A Product Line Modeling and Configuration Methodology to Support Model-Based Testing: An Industrial Case Study. *In:* France, R., Kazmeier, J., Breu, R. & Atkinson, C. (Eds.) *Model Driven Engineering Languages and Systems.* Springer Berlin Heidelberg.

Briand, L., Falessi, D., Nejati, S., Sabetzadeh, M. & Yue, T. 2012. Research-Based Innovation: A Tale of Three Projects In Model-Driven Engineering. *In:* France, R. B., Kazmeier, J., Breu, R. & Atkinson, C. (Eds.) *Model Driven Engineering Languages And Systems: 15th International Conference, MODELS 2012, Innsbruck, Austria, September 30–October 5, 2012. Proceedings.* Berlin, Heidelberg: Springer Berlin Heidelberg.

Hemmati, H., Arcuri, A. & Briand, L. 2013. Achieving Scalable Model-Based Testing Through Test Case Diversity. *ACM Trans. Softw. Eng. Methodol.,* 22**,** 1-42.

Pradhan, D., Wang, S., Ali, S., Yue, T. & Liaaen, M. 2016. Stipi: Using Search To Prioritize Test Cases Based On Multi-Objectives Derived From Industrial Practice. *In:* Wotawa, F., Nica, M. & Kushik, N. (eds.) *Testing Software And Systems: 28th IFIP WG 6.1 International Conference, ICTSS 2016, Graz, Austria, October 17-19, 2016, Proceedings.* Cham: Springer International Publishing.

Sjøberg, D. I. K. 2010. The Industry is our Lab — Organisation and Conduct Of Empirical Studies In Software Engineering at Simula. *In:* Tveito, A., Bruaset, A. M. & Lysne, O. (eds.) *Simula Research Laboratory: By Thinking Constantly About it.* Berlin, Heidelberg: Springer Berlin Heidelberg.

Wang, S., Ali, S. & Gotlieb, A. Minimizing Test Suites In Software Product Lines Using Weight-Based Genetic Algorithms. Proceeding of The Fifteenth Annual Conference on Genetic and Evolutionary Computation

Conference, 2013a. ACM, 1493-1500.

Wang, S., Ali, S. & Gotlieb, A. 2014a. Cost-Effective Test Suite Minimization In Product Lines Using Search Techniques. *Journal of Systems And Software*.

Wang, S., Ali, S., Gotlieb, A. & Liaaen, M. 2015. Automated Product Line Test Case Selection: Industrial Case Study and Controlled Experiment. *Software & Systems Modeling***,** 1-25.

Wang, S., Ali, S., Gotlieb, A. & Liaaen, M. 2016a. A Systematic Test Case Selection Methodology for Product Lines: Results And Insights from An Industrial Case Study. *Empirical Software Engineering,* 21**,** 1586-1622.

Wang, S., Ali, S., Yue, T., Bakkeli, Y. & Liaaen, M. 2016b. Enhancing Test Case Prioritization In An Industrial Setting With Resource Awareness And Multi-Objective Search. *Proceedings of The 38th International Conference on Software Engineering Companion.* Austin, Texas: ACM.

Wang, S., Ali, S., Yue, T., Li, Y. & Liaaen, M. 2016c. A Practical Guide To Select Quality Indicators For Assessing Pareto-Based Search Algorithms In Search-Based Software Engineering. *Proceedings Of The 38th International Conference on Software Engineering.* Austin, Texas: ACM.

Wang, S., Ali, S., Yue, T. & Liaaen, M. Using Feature Model To Support Model-Based Testing of Product Lines: An Industrial Case Study. 2013 13th International Conference on Quality Software, 29-30 July 2013 2013b. 75-84.

Wang, S., Buchmann, D., Ali, S., Gotlieb, A., Pradhan, D. & Liaaen, M. Multi-Objective Test Prioritization In Software Product Line Testing: An Industrial Case Study. Proceedings of The *18th International Software Product Line Conference-Volume* 1,2014b.ACM, 32-41

Wang, S., Gotlieb, A., Ali, S. & Liaaen, M. 2013a. Automated Test Case Selection Using Feature Model: An Industrial Case Study. *In:* Moreira, A., Schätz, B., Gray, J., Vallecillo, A. & Clarke, P. (Eds.) *Model-Driven Engineering Languages And Systems*. Springer Berlin Heidelberg.

Yue, T., Ali, S. & Zhang, M. 2015. Rtcm: A Natural Language Based, Automated, And Practical Test Case Generation Framework. *Proceedings Of The 2015 International Symposium on Software Testing And Analysis.* Baltimore, Md, Usa: ACM.