Electronic Communications of the EASST
Volume 77 (2019)

Interactive Workshop on the Industrial Application of Verification and Testing, ETAPS 2019 Workshop (InterAVT 2019)

Scalable Software Testing and Verification for Industrial-Scale Systems: The Challenges

Anila Mjeda and Goetz Botterweck

7 pages

Guest Editors: Anila Mjeda, Stylianos Basagiannis, Goetz Botterweck
ECEASST Home Page: http://www.easst.org/eceasst/ ISSN 1863-2122

Scalable Software Testing and Verification for Industrial-Scale Systems: The Challenges

Anila Mjeda* **and Goetz Botterweck†**

Lero–The Irish Software Research Centre
University of Limerick
Limerick, Ireland
name.surname@lero.ie

Abstract: In this position paper, we argue that more collaborative research is needed to increase the use of research-led verification and testing techniques in industrial-scale projects. We focus on the a) *practical applicability* and *scalability* of verification and testing techniques in industrial projects, and b) to *autonomous systems*. We identify the challenges involved and bring forward some initial suggestions.

Keywords: testing; verification; industrial-scale systems

1 Introduction

Modern verification and testing techniques are highly relevant for industrial software-intensive systems. While recent technological trends increase the need for industrial-scale robust approaches, issues such as education of practitioners, insufficient tools, and specifics of the industrial environment [6] create barriers for the application of modern verification and testing techniques in industrial practice.

We direct our focus on two areas: the *practical applicability* and *scalability* of verification and testing techniques in realistic industrial projects and the application of verification and testing to *autonomous systems*. In this position paper, we (1) argue that to overcome such barriers more *collaborative* research is needed and (2) identify research areas that show potential on this regards.We conclude the paper with an overview of the current challenges aiming to sketch potential areas for collaborative future work.

2 Scalability and Practical Applicability

In this section, we focus on the scalability of verification and testing techniques and applicability to industrial projects. In particular, we are looking at (1) model-based testing, (2) the integration of model-checking and testing, and (3) metaheuristic approaches.

Model-based Testing (MBT). MBT approaches typically rely on a specification-based model of the system under test. The model is used to generate tests according to some previously

* Supported, in part, by Science Foundation Ireland grant 13/RC/2094.
† Supported, in part, by Science Foundation Ireland grant 13/RC/2094.

specified test generation and selection criteria. MBT is reportedly cost-effective; it provides a platform for repeatable and traceable results which link tests to the requirements (one of the key needs for certification); and, can be used to automatically generate tests to achieve coverage criteria (such as MC/DC). MBT approaches can be distinguished by the kind of model they employ (e.g., deterministic/non-deterministic, discrete/continuous/hybrid, etc.); the test generation technology (e.g. graphs search algorithms) and test selection criteria (e.g., structural model coverage); the test execution options (e.g., MiL, SiL, HiL); and, the test evaluation approach (e.g., signal-feature based) [23]. For a review on MBT see [29].

Integration of Model-Checking and Testing. Model checking (MC) 'reasons' in terms of the states a system can be in, with state-space explosion as a fundamental challenge. Mitigation strategies typically aim to (i) consider only a subset of all possible states (e.g., probabilistic MC) or (ii) raise the level of abstraction for the state representation (e.g., symbolic MC).

Callahan et al. [4] and Engels et al. [8] were the first to suggest deriving tests from counterexamples found by a model checker, e.g., generate a test that drives the system towards a known property violation (see the survey in [13]). Such approaches can be broadly categorised into: (i) ones where the test criteria/test objectives are mutated into their logical complement (also called *trap properties* [13]); and, (ii) ones where the *model-under-analysis is mutated* [13] (e.g., by reversing the logic of transition guards in a statechart). There are a number of industrial evaluations for test generation via MC (e.g., [10]), nevertheless, it is still difficult to scale up for testing complex systems. Bounded model checking [3] and directed model checking [7] have been proposed as promising approaches in the field of test generation [12].

Metaheuristic Search and Optimization. Metaheuristic approaches mitigate the scalability problem by translating testing or verification into (automated) search and optimization techniques [5, 22, 17, 1]. Quite often the optimization problems are too complex, they could be NP-complete or NP-hard, for which optimal solutions would be impractical to compute. Metaheuristics are a subset of techniques (such as evolutionary algorithms, particle swarm optimization) that solve optimization problems via the use of non-exhaustive search algorithms; and while they do not provide guarantees to find the best solution, they return a good nearly-optimal solution at a reasonable computational cost. For an overview of the field see [17].

3 Testing and Verification of Autonomous Systems

The correct behaviour of autonomous systems requires automated correct decisions taken in real time. The decision-making is typically driven by a machine learning technique such as neural networks where the 'learning' phase mimics empirical learning based on a set of training data. These systems tend to be opaque to humans, e.g. a human typically cannot read or deduce the decision rules that have been learned by a fully trained neural network [19]. The classic testing approaches of these systems typically revolve around randomly selected test inputs and scenario simulations and can leave corner-case behaviours completely untested (e.g. autonomous driving in poor visibility). **Metamorphic techniques and new coverage metrics** – Metamorphic testing of autonomous behaviour is pursued by a number of researchers, such as [24]. Pei et al.

propose using neuron coverage (defined as the number of unique neurons that get activated for given inputs over the total number of neurons [25]) as a new testing metric applicable to neural networks. Tian et al. [28] use neuron coverage for guiding test generation and metamorphic relations to identify erroneous behaviours. **Adversarial techniques**– A number of techniques use adversarial examples such as the introduction of small distortions or noise in the inputs to detect possible erroneous behaviour in some corner cases (e.g., image distortion) [11, 20]. **Verification**– Katz et al. [18] propose using an SMT solver to verify safety properties in deep neural networks and are working to improve on the scalability of their technique for real-world systems.

4 High-Potential Industry-Academia' Collaborations

A large and growing body of literature has reported on the need for more collaboration between industry and academia and has analysed the solutions needed to address it.

For example, Bertolino [2] analysed the discordance between the state of practice and state of art in testing and argued for the need of more empirical research in industrial software testing. Garousi et.al [15] conducted a systematic literature review on the challenges of industry-academia collaborations and concluded that different focus areas are the key impediment to better collaboration.

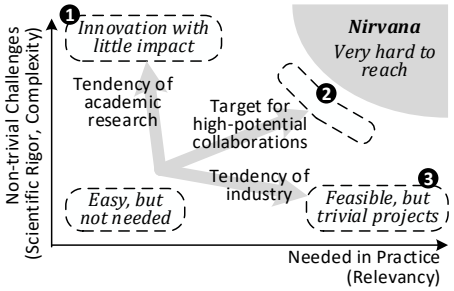

Figure 1: Conflicting objectives.

Whereas, Engström et. al. [9] proposed a taxonomy for supporting industry-academia collaboration and Gorschek at. al. [16] proposed a model for technology transfer to happen in practice.

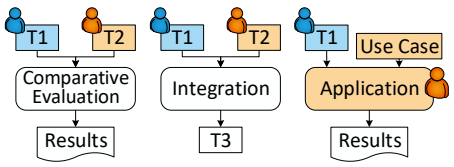

Figure 2: Models of collaboration.

Yet, this industry-academia viewpoint' divide remains and its current span damages both 'parties'. One could discuss, whether there is an *inherent* conflict between addressing challenges that are attractive for both academic research and industry. Typically, academic researchers tend to be attracted by scientifically challenging complex problems [14] (illustrated by ❶ in Figure 1); can often use artificial examples when validating the (albeit complex) research; and, can often focus on generic problems while not addressing the domain constraints [26]. Whereas, industry tends to be interested in challenges that could be scientifically trivial but tangibly relevant and feasible for them (e.g. improve effectiveness and efficiency of testing) and not particularly interested in scientifically challenging techniques that are too complex to implement in practice [14] ❸.

Here, *collaborative* projects [21] could serve as a motivation to leave one's comfort zone and tackle problems that are both scientifically novel *and* relevant in practice ❷ while industry could benefit from access to novel techniques and innovation potential. The *collaboration* can follow a number of different models such as (1) comparative evaluations of techniques, (2) integration of

techniques, or (3) the application of novel techniques in a realistic setting (Figure 2). This calls for joined efforts throughout the stages of scientific collaboration (e.g, foundation, formulation, sustainment, conclusion [27]). In collaborations, the involved can learn from each other, boost innovation, and enrich their own knowledge by relating different world-views.

5 Challenges and Conclusions

Scalable Testing and Verification. A close analysis of the state-of-the-art identifies a gap in addressing the formal testing and verification of industrial-scale software systems. The majority of the proposed solutions are based on a number of assumptions on the system under test, which can prove too restrictive for real-world systems. Moreover, the sometimes different *world-views* of researchers and industry and unfamiliarity of practitioners with prototype tools proposed by researchers create a real or perceived barrier to industry application.

Even though there are some industrial evaluations on model checking for test generation (e.g., [10]) there remain many issues when attempting to scale up these approaches for testing complex systems. At the core of the issues remains the fact that model checkers have not been originally designed for test generation [12] and the state-explosion problem that typically accompanies model-checking.

Scalability remains one of the biggest challenges and the main conclusions to take on board are: (1) *scalability analysis* should be considered in early stages of any project's design; (2) currently, the most direct route is to aim for *combinations* of testing and verification approaches that complement each other, i.e., mitigate each other's shortcomings; and, (3) collaborate to *understand each other's world view*, to compare or combine techniques, and to apply and evaluate techniques in practice.

Testing and Verification of Autonomous Systems. The complex nature of autonomous systems translates into developing methods of *testing the unknown* and *the unpredictable*. Some promising emerging techniques include neuron coverage used as a classic black-box testing technique (testing the unknown)[25], grey-box test generation led by a neuron-coverage metric and oracle generation via metamorphic testing techniques [28], leveraging adversarial techniques to detect erroneous behaviour [11, 20]; and, using solvers to verify (all be it not very rich) safety properties [18].

Collaborative Research. In this position paper, we argue that, to overcome barriers to the application of verification and testing techniques in industry, we need more *collaborative* research which is both scientifically novel *and* relevant in practice (Figure 1). This calls for *joined efforts* throughout the stages of scientific collaboration which could follow cooperation models such as (1) comparative evaluations of techniques, (2) integration of techniques, or (3) the application of novel techniques in a realistic setting (Figure 2).

Researchers could use this to avoid an over-emphasis of internal validity (e.g., experiments that are easy to run but irrelevant) and mitigate bias towards selecting evaluations that artificially confirm their techniques and all the involved can learn from each other, boost innovation, and

enrich their own knowledge by relating different world-views. Last but not least, many current challenges require interdisciplinary approaches and, hence, collaborations.

Bibliography

[1] Shaukat A., L. C Briand, H. Hemmati & R. K. Panesar-Walawege (2010): *A systematic review of the application and empirical investigation of search-based test case generation*. *TSE* 36(6), pp. 742–762.

[2] Antonia Bertolino (2004): *The (im) maturity level of software testing*. *ACM SIGSOFT Software Engineering Notes* 29(5), pp. 1–4.

[3] Armin Biere, Alessandro Cimatti, Edmund M. Clarke, Ofer Strichman & Yunshan Zhu (2003): *Bounded Model Checking*. *Advances in Computers* 58, pp. 117–148.

[4] John Callahan, Francis Schneider, Steve Easterbrook et al. (1996): *Automated software testing using model-checking*. In: *Proc. of 1996 SPIN workshop*, 353.

[5] J. Clarke, J. J. Dolado, M. Harman, R. Hierons, B. Jones, M. Lumkin, B. Mitchell, Mancoridis et al. (2003): *Reformulating software engineering as a search problem*. *IEE Proc.-Sw.* 150(3), pp. 161–175.

[6] J. A. Davis, M. Clark, D. Cofer, A. Fifarek, J. Hinchman, J. Hoffman, B. Hulbert, S. P. Miller & L. Wagner (2013): *Study on the Barriers to the Industrial Adoption of Formal Methods*, pp. 63–77. Springer, Berlin.

[7] Stefan Edelkamp, Alberto Lluch Lafuente & Stefan Leue (2001): *Directed explicit model checking with HSF-SPIN*. In: *Proc. of 8th SPIN workshop on Model Checking of Software*, Springer, pp. 57–79.

[8] André Engels, Loe Feijs & Sjouke Mauw (1997): *Test generation for intelligent networks using model checking*. *Tools and Algorithms for the Construction and Analysis of Systems*, pp. 384–398.

[9] Emelie Engström, Kai Petersen, Nauman bin Ali & Elizabeth Bjarnason (2017): *SERP-test: a taxonomy for supporting industry–academia communication*. *Software Quality Journal* 25(4), pp. 1269–1305.

[10] Eduard P. Enoiu, A. Čaušević, T. J. Ostrand, E. J. Weyuker, D. Sundmark & P. Pettersson (2016): *Automated test generation using model checking: an industrial evaluation*. *STT* 18(3), pp. 335–353.

[11] Ivan Evtimov, K. Eykholt, E. Fernandes, T. Kohno, Bo Li, A. Prakash, A. Rahmati & D. Song (2017): *Robust physical-world attacks on machine learning models*. *arXiv preprint arXiv:1707.08945*.

[12] Gordon Fraser, Franz Wotawa & Paul Ammann (2009): *Issues in using model checkers for test case generation*. *Journal of Systems and Software* 82(9), pp. 1403–1418.

[13] Gordon Fraser, Franz Wotawa & Paul E. Ammann (2009): *Testing with model checkers: a survey*. *Software Testing, Verification and Reliability* 19(3), pp. 215–261.

[14] Vahid Garousi, Michael Felderer, Marco Kuhrmann & Kadir Herkiloğlu (2017): *What industry wants from academia in software testing?: Hearing practitioners' opinions*. In: *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering*, ACM, pp. 65–69.

[15] Vahid Garousi, Kai Petersen & Baris Ozkan (2016): *Challenges and best practices in industry-academia collaborations in software engineering: A systematic literature review*. *Information and Software Technology* 79, pp. 106–127.

[16] Tony Gorschek, Per Garre, Stig Larsson & Claes Wohlin (2006): *A model for technology transfer in practice*. *IEEE software* 23(6), pp. 88–95.

[17] Mark Harman (2007): *The current state and future of search based software engineering*. In: *2007 Future of Software Engineering*, IEEE Computer Society, pp. 342–357.

[18] Guy Katz, Clark Barrett, David L Dill, Kyle Julian & Mykel J Kochenderfer (2017): *Reluplex: An efficient SMT solver for verifying deep neural networks*. In: *CAV 2017*, Springer, pp. 97–117.

[19] Alex Krizhevsky, Ilya Sutskever & Geoffrey E Hinton (2012): *Imagenet classification with deep convolutional neural networks*. In: *Advances in neural information processing systems*, pp. 1097–1105.

[20] Alexey Kurakin, Ian Goodfellow & Samy Bengio (2016): *Adversarial examples in the physical world*. *arXiv preprint arXiv:1607.02533*.

[21] Satoshi Masuda (2017): *Software testing in industry and academia: A view of both sides in japan*. In: *2017 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, IEEE, pp. 40–41.

[22] Phil McMinn (2004): *Search-based software test data generation: a survey*. *Software testing, Verification and reliability* 14(2), pp. 105–156.

[23] Anila Mjeda (2013): *Standard-compliant testing for safety-related automotive software*. Ph.D. thesis.

[24] Anh Nguyen, Jason Yosinski & Jeff Clune (2015): *Deep neural networks are easily fooled: High confidence predictions for unrecognizable images*. In: *Proc. of the CVPR 2015*, pp. 427–436.

[25] Kexin Pei, Yinzhi Cao, Junfeng Yang & Suman Jana (2017): *Deepxplore: Automated whitebox testing of deep learning systems*. In: *Proc.of the 26th Symp. on Operating Systems Principles*, ACM, pp. 1–18.

[26] Rick Rabiser, Klaus Schmid, Martin Becker, Goetz Botterweck, Matthias Galster, Iris Groher & Danny Weyns (2018): *A study and comparison of industrial vs. academic software product line research published at SPLC*. In: *Proceeedings of the 22nd International Conference on Systems and Software Product Line-Volume 1*, ACM, pp. 14–24.

[27] Diane H Sonnenwald (2007): *Scientific collaboration*. *Annual review of information science and technology* 41(1), pp. 643–681.

[28] Yuchi Tian, Kexin Pei, Suman Jana & Baishakhi Ray (2018): *Deeptest: Automated testing of deep-neural-network-driven autonomous cars*. In: *Proc. of ICSE'18*, ACM, pp. 303–314.

[29] Mark Utting, Alexander Pretschner & Bruno Legeard (2012): *A taxonomy of model-based testing approaches*. *Software Testing, Verification and Reliability* 22(5), pp. 297–312.