Formal methods and tools for industrial critical systems

Lluch Lafuente, Alberto; Mavridou, Anastasia

Published in:
International Journal on Software Tools for Technology Transfer

Link to article, DOI:
10.1007/s10009-022-00687-7

Publication date:
2022

Document Version
Peer reviewed version

International Journal on Software Tools for Technology Transfer manuscript No.
(will be inserted by the editor)

Formal Methods and Tools for Industrial Critical Systems

Alberto Lluch Lafuente · Anastasia Mavridou

Received: date / Revised version: date

Abstract Formal methods and tools have become well-established and widely applied to ensure the correctness of fundamental components of industrial critical systems in domains like railways, avionics and automotive. In this Introduction to the special issue, we outline a number of recent achievements concerning the use of formal methods and tools for the specification and verification of critical systems from a variety of industrial domains. These achievements are represented by four properly revised and extended versions of papers that were selected from the 26th *International Conference on Formal Methods for Industrial Critical Systems* (FMICS 2021).

Keywords Formal Methods · Tools · Critical Systems

1 Introduction

In industrial domains such as railways, avionics and automotive, critical (software) systems must comply with stringent dependability and safety requirements and standards. Therefore, formal methods and tools are commonly used for decades now when engineering such critical systems (cf., e.g., Craigen et al (1995); Clarke et al (1996); Hinchey and Bowen (1999); Woodcock et al (2009); Sztipanovits et al (2012); Gnesi and Margaria (2013); Sifakis (2013, 2014); Güdemann and Núñez (2017); Basile et al (2018); ter Beek et al (2018); Garavel et al (2020); Gleirscher and Marmsoler (2020); Margaria and Kiniry (2020); Ferrari and ter Beek (2022)), in particular in specific application domains (cf., e.g., Brat et al (2004); Campos et al (2014); ter Beek et al (2016); Voas and Schaffer (2016); Ozay and Tabuada (2017); Weyers et al (2017); ter Beek and Loreti (2018); Bonfanti et al (2018); Marko et al (2020); Michael et al (2021); Wing (2021); Kulik et al (2022); ter Beek and Ferrari (2022)).

Formal methods are rigorous and mathematics-based specification languages to describe (model) system behaviour (cf., e.g., Wing (1990); Hinchey et al (2010); Almeida et al (2011); Bowen and Hinchey (2014); Nielson and Nielson (2019)) that come with a precise semantics and with tools for automated formal verification (analysis) of these system models, based on techniques like theorem proving (cf. Robinson and Voronkov (2001)) and model checking (cf. Clarke et al (2009, 2018)), including probabilistic (cf. Baier and Katoen (2008)) and statistical (cf. Agha and Palmskog (2018)) approaches.

As in other engineering disciplines, the envisioned advantage of using formal methods and tools is the expectation that appropriate mathematical modelling and analysis contributes to the correctness of the developed systems by eliminating errors in the initial designs, i.e. well before implementation, and by guaranteeing robust and fault-tolerant systems that behave as specified even in uncertain environments.

This special issue dedicated to “Formal Methods and Tools for Industrial Critical Systems” contains a total of four papers that concern properly revised and extended versions of papers from the 26th *International Conference on Formal Methods for Industrial Critical Systems* (FMICS 2021).

Based on the original reviews and the subsequent discussions among the PC members and PC chairs of

Alberto Lluch Lafuente
DTU Compute, Kongens Lyngby, Denmark
E-mail: albl@dtu.dk

Anastasia Mavridou
NASA Ames Research Center, Moffett Field, USA
E-mail: anastasia.mavridou@nasa.gov

FMICS 2021 invited the authors of four papers to submit a revised and substantially extended version of their original conference paper to this special issue. The authors of all four papers decided to accept this invitation and based on a thorough reviewing process, by which each paper was reviewed by three reviewers of which at least two had not previously reviewed the conference version, the editors of this special issue decided accept all four papers for inclusion in this special issue.

2 FMICS

The aim of the FMICS conference series is to provide a forum for researchers who are interested in the development and application of formal methods in industry. In particular, FMICS brings together scientists and engineers who are active in the area of formal methods and interested in exchanging their experiences in the industrial usage of these methods. FMICS also strives to promote research and development for the improvement of formal methods and tools for industrial applications. FMICS is the annual conference of the ERCIM Working Group on Formal Methods for Industrial Critical Systems[1], and it is the key conference in the intersection of industrial applications and Formal Methods.

FMICS 2021 was held online during August 24–26, 2021 (the conference was originally scheduled to be held in Paris, France).

FMICS 2021 called for contributions on the following, non-exhaustive, topics of interest:

- Case studies and experience reports on industrial applications of formal methods, focusing on lessons learned or identification of new research directions.
- Methods, techniques and tools to support automated analysis, certification, debugging, descriptions, learning, optimisation and transformation of complex, distributed, real-time, embedded, mobile and autonomous systems.
- Verification and validation methods (model checking, theorem proving, SAT/SMT constraint solving, abstract interpretation, etc.) that address shortcomings of existing methods with respect to their industrial applicability (e.g., scalability and usability issues).
- Impact of the adoption of formal methods on the development process and associated costs. Application of formal methods in standardisation and industrial forums.

The proceedings of FMICS 2021 have been published in Springer's Lecture Notes in Computer Science series (cf. Lluch-Lafuente and Mavridou (2021)).

3 Selected Papers

In the remainder of this Introduction to the special issue, we briefly present the contributions of the papers that make up this special issue.

The paper *Automated Formal Analysis of Temporal Properties of Ladder Programs*, by Belo Lourenço et al (2022), the recipient of the FMICS 2021 Best Paper Award, describes a new approach for verifying the code running on Programmable Logic Controllers (PLCs). PLCs are industrial digital computers used as automation controllers in manufacturing processes. The software running on PLCs can be developed using Ladder Logic language, i.e., a graphical language that uses circuit diagrams of relay logic hardware to represent programs. Because of the widespread use of PLCs in industry, verifying that a given Ladder Logic program conforms to its expected behavior is of critical importance. The presented approach translates the Ladder Logic program into a Why3 program for deductive verification using automated theorem provers. The verification process, which is fully automated, either obtains a complete proof that verifies the Ladder program or a counterexample. The authors provide a prototype implementation of their approach, as well as experiments that demonstrate its effectiveness.

The paper *Verification and Synthesis of Co-Simulation Algorithms Subject to Algebraic Loops and Adaptive Steps*, by Hansen et al (2022) presents an approach to synthesize and verify orchestration algorithms for co-simulation scenarios of complex systems with algebraic loops, step rejections, adaptive couplings, and reactivity constraints. Such an approach is particularly useful for Cyber-Physical systems (CPSs), which are becoming increasingly complex. The authors argue that traditional simulation techniques are no longer sufficient to cope with the integrated development processes of CPS, which consist of heterogeneous subsystems typically developed using different tools and formalisms. The presented co-simulation approach was implemented in a tool called Scenario-Verifier, which is integrated with the orchestration engine Maestro 2. The integration enables users to seamlessly simulate their co-simulation scenarios. The authors demonstrate the effectiveness of their approach using two real-world case studies, which were simulated in Maestro 2 using a verified and synthesized orchestration algorithm from Scenario-Verifier.

The paper *Randomized Reachability Analysis in Uppaal: Fast Error Detection in Timed Systems*, by Kiviriga et al (2022) presents the implementation of randomized reachability analysis in the tool Uppaal. Randomized reachability analysis is a non-exhaustive technique for the detection of safety violations through repeated ex-

[1] `https://fmics.inria.fr/`

ploration of a model by means of random walks. It uses under-approximation for the quick falsification of models. The presented method analyses Timed Automata and Stopwatch Automata models. The authors demonstrate the scalability and effectiveness of their approach. by provideing an evaluation of the strengths and weaknesses of random reachability analysis. The authors show that their implementation outperforms in several cases existing model checking techniques by several orders of magnitude. The benefits are particularly notable when checking industrial sized systems, where traditional model-checking techniques often become intractable due to the state space explosion problem.

The final paper *SMT Solving for the Validation of B and Event-B Models* by Schmidt and Leuschel (2022) presents a new translation from B and Event-B operators to SMT-LIB. The authors also extend the interface of the PROB constraint solver to Z3 to run different solver configurations in parallel. Empirical results show that the new translation and workflow improves performance and coverage when compared to the prior PROB implementation by utilizing Z3's lambda functions. The integration of Z3 is also able to decide a lot of constraints with bounded and unbounded integer domains where PROB's solver times out. On the other hand, Z3 was not as effective for constraints that involve set cardinality or many quantifiers. To tackle such challenging constraints, the authors also developed a direct implementation of SMT solving in PROB using its constraint solver as a theory solver. The empirical evaluation performed by the authors showed that a diverse portfolio of constraint solving backends is beneficial for the B language, since no constraint solver is best for all types of constraints.

4 Discussion

We have briefly presented the four selected papers that constitute this special issue. The topics addressed in these papers cover a broad range of formal methods and tools, ranging from theorem proving and SMT solving to randomized reachability analysis and synthesis of co-simulation algorithms. Moreover, they contain applications to critical systems from industrial domains such as automation controllers and Cyber Physical systems. For future and more effective application in industry, the approaches described in the papers constituting this special issue require further development and implementation.

In the future, Belo Lourenço et al (2022) would like to improve the counterexample generation of their tool and augment the trust in translation from Ladder Logic to WhyML by developing a systematic and automatic validation process. Hansen et al (2022) plan to formalize the FMI 3.0 standard, integrate the Scenario-Verifier with other orchestration engines, and examine whether it is possible to synthesize an optimal orchestration algorithm for a given co-simulation scenario. Kiviriga et al (2022) intend to further investigate methods to improve efficiency (tokenized, coverage-based, and guided methods) and to enhance user experience (through automatic sanity checks). Finally, Schmidt and Leuschel (2022) plan to provide alternative translations for B sequences using lambda functions, compile other configurations in Z3 to run in parallel, and experiment with different SMT solvers to solve SMT-LIB models.

Acknowledgements We would like to thank all authors for their contributions to this special issue and the reviewers of FMICS 2021, and in particular those of this special issue, for their reviews.

References

Agha G, Palmskog K (2018) A survey of statistical model checking. ACM Transactions on Modeling and Computer Simulation 28(1):6:1–6:39, doi:10.1145/3158668

Almeida JB, Frade MJ, Pinto JS, Melo de Sousa S (2011) An overview of formal methods tools and techniques. In: Rigorous Software Development: An Introduction to Program Verification, Springer, pp 15–44, doi:10.1007/978-0-85729-018-2_2

Baier C, Katoen JP (2008) Principles of Model Checking. MIT Press, Cambridge, URL `http://mitpress.mit.edu/books/principles-model-checking`

Basile D, ter Beek MH, Fantechi A, Gnesi S, Mazzanti F, Piattino A, Trentini D, Ferrari A (2018) On the industrial uptake of formal methods in the railway domain. In: Furia CA, Winter K (eds) Proceedings of the 14th International Conference on Integrated Formal Methods (iFM 2018), Springer, Lecture Notes in Computer Science, vol 11023, pp 20–29, doi:10.1007/978-3-319-98938-9_2

ter Beek MH, Ferrari A (2022) Empirical Formal Methods: Guidelines for Performing Empirical Studies on Formal Methods. Software 1(4):381–416, doi:10.3390/software1040017

ter Beek MH, Loreti M (2018) Guest editorial for the special issue on FORmal methods for the quantitative Evaluation of Collective Adaptive SysTems (FORECAST). ACM Transactions on Modeling and Computer Simulation 28(2):8:1–8:4, doi:10.1145/3177772

ter Beek MH, Clarke D, Schaefer I (2016) Editorial preface for the JLAMP special issue on Formal Methods for Software Product Line Engineering. Journal of Logical and Algebraic Methods in Programming 85(1):123–124, doi:10.1016/j.jlamp.2015.09.006

ter Beek MH, Gnesi S, Knapp A (2018) Formal methods for transport systems. International Journal on Software Tools for Technology Transfer 20(3):237–241, doi:10.1007/s10009-018-0487-4

Belo Lourenço C, Cousineau D, Faissole F, Marché C, Mentré D, Inoue H (2022) Automated Formal Analysis of Temporal Properties of Ladder Programs. International Journal on Software Tools for Technology Transfer In this issue

Bonfanti S, Gargantini A, Mashkoor A (2018) A systematic literature review of the use of formal methods in medical software systems. Journal of Software: Evolution and Process 30(5):e1943:1–e1943:18, doi:10.1002/smr.1943

Bowen JP, Hinchey MG (2014) Formal Methods. In: Gonzalez TF, Diaz-Herrera J, Tucker A (eds) Computing Handbook, CRC Press, chap 71, pp 71–25

Brat GP, Drusinsky D, Giannakopoulou D, Goldberg A, Havelund K, Lowry MR, Pasareanu CS, Venet A, Visser W, Washington R (2004) Experimental Evaluation of Verification and Validation Tools on Martian Rover Software. Formal Methods in System Design 25(2-3):167–198, doi:10.1023/B:FORM.0000040027.28662.a4

Campos J, Seatzu C, Xie X (eds) (2014) Formal Methods in Manufacturing. CRC, doi:10.1201/9781315216140

Clarke EM, Wing JM, et al (1996) Formal methods: State of the art and future directions. ACM Computing Surveys 28(4):626–643, doi:10.1145/242223.242257

Clarke EM, Emerson EA, Sifakis J (2009) Model checking: algorithmic verification and debugging. Communications of the ACM 52(11):74–84, doi:10.1145/1592761.1592781

Clarke EM, Henzinger TA, Veith H, Bloem R (eds) (2018) Handbook of Model Checking. Springer, doi:10.1007/978-3-319-10575-8

Craigen D, Gerhart S, Ralston T (1995) Industrial Applications of Formal Methods to Model, Design and Analyze Computer Systems: An International Survey. Advanced Computing and Telecommunication Series, William Andrew, doi:10.1016/C2009-0-20452-1

Ferrari A, ter Beek MH (2022) Formal methods in railways: a systematic mapping study. ACM Computing Surveys doi:10.1145/3520480

Garavel H, ter Beek MH, van de Pol J (2020) The 2020 expert survey on formal methods. In: ter Beek MH, Ničković D (eds) Proceedings of the 25th International Conference on Formal Methods for Industrial Critical Systems (FMICS 2020), Springer, Lecture Notes in Computer Science, vol 12327, pp 3–69, doi:10.1007/978-3-030-58298-2_1

Gleirscher M, Marmsoler D (2020) Formal methods in dependable systems engineering: a survey of professionals from Europe and North America. Empirical Software Engineering 25(6):4473–4546, doi:10.1007/s10664-020-09836-5

Gnesi S, Margaria T (eds) (2013) Formal Methods for Industrial Critical Systems: A Survey of Applications. John Wiley & Sons, Inc., Hoboken, doi:10.1002/9781118459898

Güdemann M, Núñez M (2017) Preface of the special issue on formal methods in industrial critical systems. International Journal on Software Tools for Technology Transfer 19(4):391–393, doi:10.1007/s10009-017-0455-4

Hansen ST, Thule C, Gomes C, Pol Jvd, Palmieri M, Oguz IE, Madsen F, Alfonso J, Castellanos JA, Rodriguez JM (2022) Verification and Synthesis of Co-Simulation Algorithms Subject to Algebraic Loops and Adaptive Steps. International Journal on Software Tools for Technology Transfer In this issue

Hinchey M, Bowen JP, Vassev E (2010) Formal methods. In: Laplante PA (ed) Encyclopedia of Software Engineering, Taylor & Francis, pp 308–320, URL http://www.crcnetbase.com/doi/abs/10.1081/E-ESE-120044313

Hinchey MG, Bowen JP (eds) (1999) Industrial-Strength Formal Methods In Practice. Formal Approaches to Computing Information Technology, Springer, doi:10.1007/978-1-4471-0523-7

Kiviriga A, Larsen KG, Nyman U (2022) Randomized Reachability Analysis in Uppaal: Fast Error Detection in Timed Systems. International Journal on Software Tools for Technology Transfer In this issue

Kulik T, Dongol B, Larsen PG, Macedo HD, Schneider S, Tran-Jørgensen PWV, Woodcock J (2022) A Survey of Practical Formal Methods for Security. Formal Aspects of Computing 34(1):5:1–5:39, doi:10.1145/3522582

Lluch-Lafuente A, Mavridou A (eds) (2021) Proceedings of the 26th International Conference on Formal Methods for Industrial Critical Systems (FMICS 2021), Lecture Notes in Computer Science, vol 12863, Springer, doi:10.1007/978-3-030-85248-1

Margaria T, Kiniry J (2020) Welcome to formal methods in industry. IT Professional 22(1):9–12, doi:10.1109/MITP.2020.2968715

Marko N, Möhlmann E, Ničković D, Niehaus J, Priller P, Rooker M (2020) Challenges of engineering safe and secure highly automated vehicles: Whitepaper. arXiv 2103.03544 [cs.AI], URL https://arxiv.org/abs/2103.03544

Michael JB, Drusinsky D, Wijesekera D (2021) Formal methods in cyberphysical systems. IEEE Computer 54(9):25–29, doi:10.1109/MC.2021.3089267

Nielson F, Nielson HR (2019) Formal Methods: An Appetizer. Springer, doi:10.1007/978-3-030-05156-3

Ozay N, Tabuada P (2017) Guest editorial: special issue on formal methods in control. Discrete Event Dynamic Systems 27(2):205–208, doi:10.1007/s10626-017-0246-9

Robinson JA, Voronkov A (eds) (2001) Handbook of Automated Reasoning. Elsevier and MIT Press

Schmidt J, Leuschel M (2022) SMT Solving for the Validation of B and Event-B models. International Journal on Software Tools for Technology Transfer In this issue

Sifakis J (2013) Rigorous System Design. Foundations and Trends in Electronic Design Automation 6(4):293–362, doi:10.1561/1000000034

Sifakis J (2014) Rigorous System Design. In: Proceedings of the 33rd ACM Symposium on Principles of Distributed Computing (PODC 2014), ACM, p 292, doi:10.1145/2611462.2611517

Sztipanovits J, Koutsoukos XD, Karsai G, Kottenstette N, Antsaklis PJ, Gupta V, Goodwine B, Baras JS, Wang S (2012) Toward a Science of Cyber–Physical System Integration. Proceedings of the IEEE 100(1):29–44, doi:10.1109/JPROC.2011.2161529

Voas JM, Schaffer K (2016) Insights on formal methods in cybersecurity. IEEE Computer 49(5):102–105, doi:10.1109/MC.2016.131

Weyers B, Bowen J, Dix A, Palanque P (eds) (2017) The Handbook of Formal Methods in Human-Computer Interaction. Human–Computer Interaction Series, Springer, doi:10.1007/978-3-319-51838-1

Wing JM (1990) A specifier's introduction to formal methods. IEEE Comput 23(9):8–24, doi:10.1109/2.58215

Wing JM (2021) Trustworthy AI. Communications of the ACM 64(10):64–71, doi:10.1145/3448248

Woodcock J, Larsen PG, Bicarregui J, Fitzgerald JS (2009) Formal methods: Practice and experience. ACM Computing Surveys 41(4):19:1–19:36, doi:10.1145/1592434.1592436