Noname manuscript No.
(will be inserted by the editor)

Introduction to the Special Issue on Runtime Verification

Yliès Falcone · César Sánchez

the date of receipt and acceptance should be inserted later

Abstract This article introduces the extended versions of selected papers from the refereed proceedings of the 16th International Conference on Runtime Verification (RV 2016) held in Madrid, Spain, in September 2016. Runtime Verification encompasses all aspects of monitoring and analysis of hardware, software, and system executions in general. Runtime verification techniques are lightweight dynamic techniques to assess and enforce correctness, reliability, and robustness during system execution. These techniques are significantly more powerful and versatile than conventional testing, and more practical than exhaustive formal verification (at the price of incomplete coverage).

1 Runtime Verification

Runtime Verification (RV) is the umbrella term to encompass all aspects of monitoring and analysis of hardware, software and the executions of systems in general. RV techniques are lightweight dynamic techniques to assess and enforce correctness, reliability, and robustness during system execution. These techniques are significantly more powerful, versatile and rigorous than conventional testing, and more practical than exhaustive formal verification. Foundational papers on RV include [1–5]; see also [6–11] for tutorial papers. Recently, a tutorial book on RV has

The editors of the special issue would like to thank the authors of all submitted papers, the members of the Program Committee of RV 2016, and the reviewers of this special issue for their exhaustive reviews and evaluations. The authors appreciated the support of the editorial team of Formal Methods in System Design. The authors acknowledge the support of the ICT COST Action IC1402 Runtime Verification beyond Monitoring (ARVI).

Yliès Falcone
Univ. Grenoble Alpes, CNRS, Inria, Grenoble INP, Laboratoire d'Informatique de Grenoble, Grenoble, France
E-mail: ylies.falcone@univ-grenoble-alpes.fr

César Sánchez
IMDEA Software Institute, Madrid, Spain
E-mail: cesar.sanchez@imdea.org

been released [12] presenting introductory and advanced topics, including the monitoring of cyber-physical systems [13], runtime failure prevention and reaction [14], monitoring concurrency errors [15], monitoring decentralized and distributed systems [16], and financial and transaction systems [17].

In the last decade, Runtime Verification, as a computer science field, has grown significantly. As an event, RV—which started initially as a workshop—became a conference in 2010 [18]. Three competitions on software for runtime verification have been organized in so far (see [19–21] and [22] for an extensive report of the first incarnation of the competition) as well as a workshop reporting reflections on past competitions [23]. Additionally, two graduate schools (see [24]) devoted to Runtime Verification have been recently organized in 2016[1] and 2018[2]. A European COST Action on Runtime Verification (IC 1402, Runtime Verification beyond monitoring[3]) is ongoing with the objectives of connecting Runtime Verification to other analysis techniques, and expanding the applicability of RV beyond software reliability. Workshops on particular aspects of Runtime Verification have also flourished: the AD-RV track [25,26] at ISoLA 2014[4] on general applications of Runtime Verification, the iAD-RV [27] track on industrial applications of Runtime Verification at ISoLA 2016[5], the PrePost workshop on pre and post deployment technique in 2016, and the recent RUME[6] and VORTEX[7] workshops on embedded systems and object-oriented languages, respectively.

As the area is getting increasingly mature, deeper results are generated tackling more complex problems. This special issue contains extended versions of selected papers from the RV 2016 conference, as a witness to this maturity.

2 Summaries of the Selected Articles

In this section, we briefly summarize the articles contained in this special issue. All articles are significantly extend the corresponding papers from the refereed proceedings [28] of the 16th International Conference on Runtime Verification, RV 2016, held in Madrid, Spain, in September 2016.

Goubault-Larrecq and Lachance [29] use monitoring for intrusion detection via their tool Orchids. Their paper addresses the problem of determining the complexity of monitoring an Orchids signature (a specification) and provide a linear-time algorithm that determines whether the number of monitors for this specification is linear or exponential in the number of events.

Shi et al. [30] validate at runtime wireless protocol using wireless sniffers to avoid device instrumentation. They address the problem of losses caused by wireless propagation which prevents the construction of complete traces. Protocols are expressed as a state machine which encodes the uncertainty of sniffers by adding non-determinism. The validation problem is framed as a decision problem, shown

[1] `rv2016.imag.fr/?page_id=128`
[2] `www.cost-arvi.eu/?page_id=1163`
[3] `www.cost-arvi.eu`
[4] `www.cs.uni-potsdam.de/gsse/www.isola-conference.org/isola2012/`
[5] `isola-conference.org/isola2016/`
[6] `beru.univ-brest.fr/RUME18.html`
[7] `conf.researchr.org/track/vortex-2018/vortex-2018-papers`

to be NP-complete, and they provide an algorithm for exhaustively exploring mutated traces. The algorithm can be enhanced by protocol-oblivious heuristics to select most likely mutated traces.

Kauffman et al. [31] infer abstractions of event streams produced by telemetry systems of the Curiosity rover on Mars. Their approach introduces a hierarchy of event abstractions that can be queried and visualized. Event abstractions are expressed as a rule-based system inspired by Allen's temporal logic. The approach is implemented in both the C and the Scala programming languages and the specification formalism as both internal and external domain-specific languages.

Jakšić et al. [32] monitor the physical behavior of cyber-physical systems. They provide a quantitative semantics of Signal Temporal Logic by introducing a weighted edit distance. The decision procedure is a dynamic programming algorithm which allows quantifying the similarity between the system and specification behaviors. Hardware-based monitors are implemented on an FPGA, assessed on automotive benchmarks, and used on a magnetic sensor of modern cars.

Moreno and Fischmeister [33] enforce the safety and security of embedded systems using power consumption. Their approach is non-intrusive and consists in analyzing the signal and the system using a spectral analysis that matches the input and output signal. The approach leverages the control-flow graph of the program for performance improvement. They present experiments on a SCADE application and a case study where anomalous executions are detected.

Roşu [34] introduces a sound and complete direct proof system for linear-temporal logic with (only) finite traces. The proof system consists of seven rules extending the proof system of propositional logic, six rules are rather expected, and one special rule is of coinductive nature. Roşu shows that this rule is strictly more powerful than the classical inductive rule used in existing proof systems for infinite and infinite-finite traces.

References

1. Moonjoo Kim, Mahesh Viswanathan, Hanêne Ben-Abdallah, Sampath Kannan, Insup Lee, and Oleg Sokolsky. Formally specified monitoring of temporal properties. In *Proc. of the 11th Euromicro Conference on Real-Time Systems (ECRTS 1999)*, pages 114–122. IEEE Computer Society, 1999.
2. Oleg Sokolsky, Sampath Kannan, Moonjoo Kim, Insup Lee, and Mahesh Viswanathan. Steering of real-time systems based on monitoring and checking. In *Proc. of the Fifth International Workshop on Object-Oriented Real-Time Dependable Systems, WORDS'99*, pages 11–18. IEEE Computer Society, 1999.
3. Klaus Havelund and Grigore Rosu. Monitoring programs using rewriting. In *Proc. of the 16th IEEE International Conference on Automated Software Engineering (ASE 2001)* [35], pages 135–143.
4. Dimitra Giannakopoulou and Klaus Havelund. Automata-based verification of temporal properties on running programs. In *Proc. of the 16th IEEE International Conference on Automated Software Engineering (ASE 2001)* [35], pages 412–416.
5. Mahesh Viswanathan and Moonzoo Kim. Foundations for the run-time monitoring of reactive systems - fundamentals of the MaC language. In Zhiming Liu and Keijiro Araki, editors, *Revised Selected Papers from the First International Colloquium on Theoretical Aspects of Computing (ICTAC 2004)*, volume 3407 of *LNCS*, pages 543–556. Springer, 2004.
6. Klaus Havelund and Allen Goldberg. Verify your runs. In Bertrand Meyer and Jim Woodcock, editors, *Revised Selected Papers and Discussions from the First IFIP TC 2/WG 2.3 Conference on Verified Software: Theories, Tools, Experiments, (VSTTE 2005)*, volume 4171 of *LNCS*, pages 374–383. Springer, 2005.

7. Martin Leucker and Christian Schallhart. A brief account of runtime verification. *J. Log. Algebr. Program.*, 78(5):293–303, 2009.
8. Yliès Falcone. You should better enforce than verify. In Barringer et al. [18], pages 89–105.
9. Yliès Falcone, Klaus Havelund, and Giles Reger. A tutorial on runtime verification. In Manfred Broy, Doron A. Peled, and Georg Kalus, editors, *Engineering Dependable Software Systems*, volume 34 of *NATO Science for Peace and Security Series, D: Information and Communication Security*, pages 141–175. IOS Press, 2013.
10. Yliès Falcone, Jean-Claude Fernandez, and Laurent Mounier. On the expressiveness of some runtime validation techniques. In Andrei Voronkov and Margarita V. Korovina, editors, *HOWARD-60: A Festschrift on the Occasion of Howard Barringer's 60th Birthday*, volume 42 of *EPiC Series in Computing*, pages 112–123. EasyChair, 2014.
11. Ezio Bartocci, Yliès Falcone, Adrian Francalanza, and Giles Reger. Introduction to runtime verification. In Bartocci and Falcone [12], pages 1–33.
12. Ezio Bartocci and Yliès Falcone, editors. *Lectures on Runtime Verification - Introductory and Advanced Topics*, volume 10457 of *LNCS*. Springer, 2018.
13. Ezio Bartocci, Jyotirmoy V. Deshmukh, Alexandre Donzé, Georgios E. Fainekos, Oded Maler, Dejan Nickovic, and Sriram Sankaranarayanan. Specification-based monitoring of cyber-physical systems: A survey on theory, tools and applications. In Bartocci and Falcone [12], pages 135–175.
14. Yliès Falcone, Leonardo Mariani, Antoine Rollet, and Saikat Saha. Runtime failure prevention and reaction. In Bartocci and Falcone [12], pages 103–134.
15. João M. Lourenço, Jan Fiedor, Bohuslav Krena, and Tomás Vojnar. Discovering concurrency errors. In Bartocci and Falcone [12], pages 34–60.
16. Adrian Francalanza, Jorge A. Pérez, and César Sánchez. Runtime verification for decentralised and distributed systems. In Bartocci and Falcone [12], pages 176–210.
17. Christian Colombo and Gordon J. Pace. Industrial experiences with runtime verification of financial transaction systems: Lessons learnt and standing challenges. In Bartocci and Falcone [12], pages 211–232.
18. Howard Barringer, Yliès Falcone, Bernd Finkbeiner, Klaus Havelund, Insup Lee, Gordon J. Pace, Grigore Rosu, Oleg Sokolsky, and Nikolai Tillmann, editors. *Proc. of the First International Conference on Runtime Verification (RV 2010)*, volume 6418 of *LNCS*. Springer, 2010.
19. Ezio Bartocci, Borzoo Bonakdarpour, and Yliès Falcone. First international competition on software for runtime verification. In Borzoo Bonakdarpour and Scott A. Smolka, editors, *Proc. of the Fifth International Conference on Runtime Verification (RV 2014)*, volume 8734 of *LNCS*, pages 1–9. Springer, 2014.
20. Yliès Falcone, Dejan Nickovic, Giles Reger, and Daniel Thoma. Second international competition on runtime verification CRV 2015. In Ezio Bartocci and Rupak Majumdar, editors, *Proc. of the 6th International Conference on Runtime Verification (RV 2015)*, volume 9333 of *LNCS*, pages 405–422. Springer, 2015.
21. Giles Reger, Sylvain Hallé, and Yliès Falcone. Third international competition on runtime verification - CRV 2016. In Falcone and Sánchez [28], pages 21–37.
22. Ezio Bartocci, Yliès Falcone, Borzoo Bonakdarpour, Christian Colombo, Normann Decker, Klaus Havelund, Yogi Joshi, Felix Klaedtke, Reed Milewicz, Giles Reger, Grigore Rosu, Julien Signoles, Daniel Thoma, Eugen Zalinescu, and Yi Zhang. First international competition on runtime verification: rules, benchmarks, tools, and final results of crv 2014. *International Journal on Software Tools for Technology Transfer*, Apr 2017.
23. Giles Reger. A report of rv-cubes 2017. In Giles Reger and Klaus Havelund, editors, *RV-CuBES 2017. An International Workshop on Competitions, Usability, Benchmarks, Evaluation, and Standardisation for Runtime Verification Tools, September 15, 2017, Seattle, WA, USA*, volume 3 of *Kalpa Publications in Computing*, pages 1–9. EasyChair, 2017.
24. Christian Colombo and Yliès Falcone. First international summer school on runtime verification - as part of the arvi COST action 1402. In Falcone and Sánchez [28], pages 17–20.
25. Yliès Falcone and Lenore D. Zuck. Runtime verification: The application perspective. In Tiziana Margaria and Bernhard Steffen, editors, *Proc. of the 5th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation Technologies for Mastering Change (ISoLA 2012), Part I*, volume 7609 of *LNCS*, pages 284–291. Springer, 2012.

26. Yliès Falcone and Lenore D. Zuck. Runtime verification: the application perspective. *STTT*, 17(2):121–123, 2015.
27. Ezio Bartocci and Yliès Falcone. Runtime verification and enforcement, the (industrial) application perspective (track introduction). In *Proc. of the 7th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation: Discussion, Dissemination, Applications (ISoLA 2016), Part II*, volume 9953 of *LNCS*, pages 333–338, 2016.
28. Yliès Falcone and César Sánchez, editors. *Proc. of the 16th International Conference on Runtime Verification (RV 2016)*, volume 10012 of *LNCS*. Springer, 2016.
29. Jean Goubault-Larrecq and Jean-Philippe Lachance. On the complexity of monitoring orchids signatures, and recurrence equations. *Formal Methods in System Design*, Oct 2017.
30. Jinghao Shi, Shuvendu K. Lahiri, Ranveer Chandra, and Geoffrey Challen. Wireless protocol validation under uncertainty. *Formal Methods in System Design*, Nov 2017.
31. Sean Kauffman, Klaus Havelund, Rajeev Joshi, and Sebastian Fischmeister. Inferring event stream abstractions. *Formal Methods in System Design*, Feb 2018.
32. Stefan Jakšić, Ezio Bartocci, Radu Grosu, Thang Nguyen, and Dejan Ničković. Quantitative monitoring of STL with edit distance. *Formal Methods in System Design*, Mar 2018.
33. Carlos Moreno and Sebastian Fischmeister. Non-intrusive runtime monitoring through power consumption to enforce safety and security properties in embedded systems. *Formal Methods in System Design*, Sep 2017.
34. Grigore Rosu. Finite-trace linear temporal logic: Coinductive completeness. *Formal Methods in System Design*, May 2018.
35. *Proc. of the 16th IEEE International Conference on Automated Software Engineering (ASE 2001)*. IEEE Computer Society, 2001.
36. Andrei Voronkov and Margarita V. Korovina, editors. *HOWARD-60: A Festschrift on the Occasion of Howard Barringer's 60th Birthday*, volume 42 of *EPiC Series in Computing*. EasyChair, 2014.
37. Zhiming Liu and Keijiro Araki, editors. *Theoretical Aspects of Computing - ICTAC 2004, First International Colloquium, Guiyang, China, September 20-24, 2004, Revised Selected Papers*, volume 3407 of *LNCS*. Springer, 2005.
38. *11th Euromicro Conference on Real-Time Systems (ECRTS 1999), 9-11 June 1999, York, England, UK, Proceedings*. IEEE Computer Society, 1999.
39. *Fifth International Workshop on Object-Oriented Real-Time Dependable Systems, WORDS Fall 1999, November 18-10, 1999, Monterey, California, USA. Proceedings*. IEEE Computer Society, 1999.
40. Bertrand Meyer and Jim Woodcock, editors. volume 4171 of *LNCS*. Springer, 2008.
41. Manfred Broy, Doron A. Peled, and Georg Kalus, editors. *Engineering Dependable Software Systems*, volume 34 of *NATO Science for Peace and Security Series, D: Information and Communication Security*. IOS Press, 2013.
42. Borzoo Bonakdarpour and Scott A. Smolka, editors. *Proc. of the 5th International Conference on Runtime Verification (RV 2014)*, volume 8734 of *LNCS*. Springer, 2014.
43. Ezio Bartocci and Rupak Majumdar, editors. *Proc. of the 6th International Conference on Runtime Verification (RV 2015)*, volume 9333 of *LNCS*. Springer, 2015.
44. Giles Reger and Klaus Havelund, editors. *RV-CuBES 2017. An International Workshop on Competitions, Usability, Benchmarks, Evaluation, and Standardisation for Runtime Verification Tools, September 15, 2017, Seattle, WA, USA*, volume 3 of *Kalpa Publications in Computing*. EasyChair, 2017.
45. *Proc. of the 7th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation: Discussion, Dissemination, Applications (ISoLA 2016), Part II*, volume 9953 of *LNCS*, 2016.
46. Tiziana Margaria and Bernhard Steffen, editors. *Proc. of the 5th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation Technologies for Mastering Change (ISoLA 2012), Part I*, volume 7609 of *LNCS*. Springer, 2012.