COST Action IC1402 Runtime Verification Beyond Monitoring

Christian Colombo[1(✉)], Yliès Falcone[2], Martin Leucker[3], Giles Reger[4], Cesar Sanchez[5], Gerardo Schneider[6], and Volker Stolz[7]

[1] University of Malta, Msida, Malta
christian.colombo@um.edu.mt
[2] Univ. Grenoble Alpes, CNRS, Inria, Grenoble INP, LIG, 38000 Grenoble, France
[3] Universtität zu Lübeck, Lübeck, Germany
[4] University of Manchester, Manchester, UK
[5] IMDEA Software Institute, Madrid, Spain
[6] University of Gothenburg, Gothenburg, Sweden
[7] Western Norway University of Applied Sciences, Bergen, Norway

Abstract. In this paper we report on COST Action IC1402 which studies Runtime Verification approaches beyond Monitoring. COST Actions are funded by the European Union and are an efficient networking instrument for researchers, engineers and scholars to cooperate and coordinate research activities. This COST action IC1402 lasted over the past four years, involved researchers from 27 different European countries and Australia and allowed to have many different working group meetings, workshops and individual visits.

1 Introduction

Runtime verification (RV) is a computing analysis paradigm based on observing a system at runtime to check its expected behavior. RV has emerged in recent years as a practical application of formal verification, and a less ad-hoc approach to conventional testing by building monitors from formal specifications. For tutorials and overviews of the field of Runtime Verification, we refer to [11,15,22,28].

There is a great potential applicability of RV beyond software reliability, if one allows monitors to interact back with the observed system, and generalizes to new domains beyond computers programs (like hardware, devices, cloud computing and even human-centric systems). Given the European leadership in computer-based industries, novel applications of RV to these areas can have an enormous impact in terms of the new class of designs enabled and their reliability and cost effectiveness.

COST Actions are a flexible, fast, effective and efficient networking instrument for researchers, engineers and scholars to cooperate and coordinate nationally funded research activities. COST Actions allow European researchers to jointly develop their own ideas in any science and technology field.

C. Colombo and M. Leucker (Eds.): RV 2018, LNCS 11237, pp. 18–26, 2018.
https://doi.org/10.1007/978-3-030-03769-7_2

This COST Action lasted from beginning of 2015 till the end of 2018. This paper describes its structure as well as the main results achieved in this action. Latest updates on this COST action can be found at https://www.cost-arvi.eu.

2 Working Groups

In this section, we briefly report on the activities carried by each of the working groups. Working groups served to structure and coordinate the work within the action.

2.1 Working Group 1: Core Runtime Verification

Working Group 1 (WG1) aimed at clarifying the dimensions of RV, its theory, algorithms and methods. These are the activities in which most of the work on RV has focused in the early stages of the discipline, with scattered results based on methods from other areas, notably formal methods and programming languages, and guided by application goals. Many outcomes from the other working groups posed new sets of problems and challenges for the core RV community. Specific activities of WG1 included research actions centered around establishing a common framework for RV, and challenges for new research and technology based on the other working groups. These activities led to several achievements, which are exposed in several publications and the report of WG1. We briefly summarize the achievements below:

- *A tutorial book* providing a collection of 7 lectures on introductory and advanced topics on RV [5].
- *A taxonomy of RV* aspects that "paves the road" to allow a classification and comparison of theoretical results, problems and techniques. The taxonomy has been published in [16].
- The identification of the *challenges and opportunities of instrumentation*, where the system under scrutiny is modified or harnessed to allow the monitoring process. The challenges are exposed in the report of this working group and in the introductory book chapter [7].
- A study of the *interplay* between RV and static analysis, between RV and model checking, and between RV and testing. All these activities usually serve to increase or assess system's reliability, but their interplay can potentially increase their applicability. The interplay study is exposed in the report of WG1.
- A study of potential applications of RV beyond system observation. This includes reflection to act upon the system, typically to control and prevent errors, or to replay allowing an error to be reproduced or even fixed. Potential applications beyond system observation are exposed in the report of this working group and in a chapter of the tutorial book dedicated to financial applications [14], and a chapter dedicated to runtime failure prevention and reaction [17]. We have also published a paper on the combination of reinforcement learning and RV monitors [29].

– To pose the *challenges in monitoring quantitative and statistical data*, beyond property violation. The challenges are exposed in the report of WG1 and in some chapters of the tutorial book, notably those on monitoring with data [25] and monitoring cyber-physical systems [2].

Additionally, WG1 has organized several events and coordinated publications to promote Runtime Verification as a field of research and favor the dissemination of the core aspects of the field. These events include *two tracks* on RV at IsoLA 2016 [30] and 2018 focused on industrial aspects [3,4], *two special issues* in Formal Methods in System Design [9,20], two successful *international schools* on RV attracting around 40 students each [12,13] with one organized alongside the 16th International Conference on Runtime Verification [19] and one as an independent event, *competitions* on Software for Runtime Verification [1,18,32] as well as an extensive report on the first edition [6].

2.2 Working Group 2: Standardization, Benchmarks, Tool Interoperability

This group aimed to clarify the landscape of formalisms and tools proposed and built for RV, to design common input formats and to establish a large class of benchmarks and challenges. We briefly summarise the main achievements of the working group:

– *Classification of Tools.* The taxonomy mentioned above (in working group 1) was developed alongside a classification of Runtime Verification tools and further refined with respect to this classification [16].
– *Exploration of Language Landscape.* The working group has encouraged a number of activities exploring the links between specification languages for Runtime Verification [24,35,36]. This has been both theoretically (defining translations between languages) and pragmatically (discussing topics such as usability).
– *Competitions.* Between 2014 and 2016 three competitions were carried out comparing Runtime Verification tools for monitoring C programs, Java programs, and log files. These competitions compared 14 tools using over 100 different benchmarks. Full accounts of the competitions have been published [1,6,18,32] and an ongoing account of these and future competitions can be found at https://www.rv-competition.org/.
– *Trace Formats.* A number of trace formats were introduced and refined in the above competitions including CSV, JSON, and XML formats. These have been the subject of further exploration and discussion [26,33].
– *Encouraging a Conversation.* One of the most important jobs of this working group was to get the different tool developers to talk to each other. We organised two events outside the Action to encourage this. Firstly, the RV-CuBES workshop [31,34] was held alongside the 17th International Conference on Runtime Verification [27]. This contained 11 short tool papers and 5 position papers discussing how RV tools should be evaluated [10,37,39], describing

challenges of using RV tools in industry [21], and encouraging the community to use open standards [26]. Secondly, a Dagstuhl seminar [23] considered various issues around behavioural specification languages, inviting researchers from outside the RV community to join the discussion.

The activities of the working group are ongoing. The above taxonomy and classification continues to be refined and extended. The landscape of Runtime Verification languages is still not fully understood and more work is being carried out in this area. The competition continues, with a challenge focusing on benchmarks coinciding with the 18th International Conference on Runtime Verification and the end of this Action.

2.3 Working Group 3: Challenging Computational Domains

The main goal of this group has been to studied novel and important (but challenging) computational domains for RV and monitoring, that result from the study of other application areas other than programming languages. The concrete objectives of this Working Group was to identify concrete challenges for RV and monitoring in the following application domains:

Distributed Systems: where the timing of observations may vary widely in a non-synchronized manner.
Hybrid and Embedded Systems: where continuous and discrete behaviour coexist and the resources of the monitor are constrained.
Hardware: where the timing must be precise and the monitor must operate non disruptively.
Security & Privacy: where a suitable combination between static and dynamic analysis is needed.
Reliable Transactional Systems: where data consistency and strong guarantees of concurrent execution must be provided at network scale.
Contracts & Policies: where the connection between the legal world and the technical is paramount.
Unreliable Domains and Approximated Domains: where either the systems is not reliable, or aggregation or sampling is necessary due to large amounts of data.

The study of these areas has involved expertise from more than one domain, and has been possible by attacking them cooperatively. The first concrete outputs of this Working Group is a series of documents that give a roadmap for the application of RV techniques to the areas listed above, identifying connections with established work in the respective sub-areas of computer science, and challenges and opportunities. A summary of the content of these works where consolidated into a paper (60 pages, 336 references) and will appear in journal survey publication, currently under submission [38]. Second, a concrete case study has been defined, aiming at a RV solution for multicore systems using dedicated monitoring hardware based on FPGAs to show the feasibility and general applicability of RV techniques (ongoing work).

2.4 Working Group 4: Application Areas (Outside "Pure" Software Reliability)

This group have studied the potential applications of RV to important application areas beyond software and hardware reliability, including medical devices and legal contracts. This task required the direct interaction with experts from the respective communities. For example, for the safe interoperability of medical devices, it was important to enrich the interface COST specifications with temporal properties about the intended interaction of two devices and to synthesize monitoring code for runtime. If monitoring identifies unwanted behavior, the systems might go into some fail-safe mode. Another interesting application area that has been explored was how to monitor legal e-contracts (e.g., computer-mediated transactions). Some efforts have recently been done to formalize legal contracts using formal languages, where skeletons of runtime monitors could be extracted from the formal semantics. Other applications included robotics and hybrid systems, monitoring for business models and systems security. Concrete output of this Working Group consisted on documents describing challenges and potential applications of RV to these application areas. Moreover, a concrete case study in the medical domain has been performed identifying the safety enhancements of medical devices by using RV techniques.

Main application areas studied by the working group:

- Medical devices
- Legal contracts
- Financial transactions
- Security and privacy
- Electrical energy storage

This Working Group have organized few workshops with invited experts from application domains:

- ARVI Workshop on Financial Transaction Systems (organized by Christian Colombo).[1]
- Workshop on Medical Cyber Physical Systems (co-organised by Ezio Bartocci and Martin Leucker).[2]
- ARVI Workshop on the Analysis of Legal Contracts (co-organized by Christian Colombo, Gordon Pace and Gerardo Schneider).[3]
- ARVI Workshop on Privacy & Security (co-organized by Leonardo Mariani and Gerardo Schneider).[4]

[1] https://www.cost-arvi.eu/?page_id=166.
[2] http://mlab-upenn.github.io/medcps_workshop/.
[3] https://www.cost-arvi.eu/?page_id=862.
[4] https://www.cost-arvi.eu/?page_id=1431.

3 Short-Term Scientific Missions (STSMs)

The COST actions also provided financial support for so-called short-term scientific missions. The idea is to support individual mobility, strengthening existing networks and fostering collaboration. The visits should contribute to the scientific objectives of the COST Action that means concentrate on topics investigated in one of the four working groups while at the same time, allow to learn new techniques, gain access to specific data, instruments, methods not available in their own organizations.

The applications for an STSM were carefully reviewed by the STSM committee, which consisted of Tarmo Uustalu (Reykjavik University, Iceland), César Sénchez (IMDEA Software, Spain) and Martin Steffen (University of Olso, Norway).

Within this COST action, a total of 23 STSMs were carried out while another 2 are currently planned. Overall, the STSMs strengthened our joint interaction and resulted in many high-quality scientific contributions.

4 IC1402 in Numbers

Grant period:	17.12.2014 – 16.12.2018
Participating COST countries:	27
COST international Non-European partner countries:	1 (Australia)
Participating scientists:	Over 90
STSMs completed:	23 (+2 expected)
Including for young scientists:	9
Including female scientists:	7
Meetings:	13 completed
Workshops:	5
Training schools:	2
ITC conference grants	1 (Serbia)
Publications:	Over 40
Book published:	Lectures on RV: Introductory and Advanced Topics, Springer 2017

References

1. Bartocci, E., Bonakdarpour, B., Falcone, Y.: First international competition on software for runtime verification. In: Bonakdarpour, B., Smolka, S.A. (eds.) RV 2014. LNCS, vol. 8734, pp. 1–9. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11164-3_1
2. Bartocci, E., Deshmukh, J., Donzé, A., Fainekos, G.E, Maler, O., Ničković, D., Sankaranarayanan, S.: Specification-based monitoring of cyber-physical systems: a survey on theory, tools and applications. In: Bartocci and Falcone [5], pp. 135–175
3. Bartocci, E., Falcone, Y.: RV-TheToP: runtime verification from theory to the industry practice (track introduction). In: Margaria T., Steffen B. (eds.) Leveraging Applications of Formal Methods, Verification and Validation: Discussion, Dissemination, Applications - 8th International Symposium, ISoLA 2018, Limassol, Cyprus, 30 October–13 November 2018 Proceedings, Part II. Lecture Notes in Computer Science (2018, to appear)
4. Bartocci, E., Falcone, Y.: Runtime verification and enforcement, the (industrial) application perspective (track introduction). In: Margaria and Steffen [30], pp. 333–338. https://doi.org/10.1007/978-3-319-47169-3_24
5. Bartocci, E., Falcone, Y. (eds.): Lectures on Runtime Verification - Introductory and Advanced Topics. LNCS, vol. 10457. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-75632-5
6. Bartocci, E., et al.: First international competition on runtime verification: rules, benchmarks, tools, and final results of CRV 2014. Int. J. Softw. Tools Technol. Transf. (2017)
7. Bartocci, E., Falcone, Y., Francalanza, A., Reger, G.: Introduction to runtime verification. In: Bartocci and Falcone [5], pp. 1–33
8. Bartocci, E., Majumdar, R. (eds.): RV 2015. LNCS, vol. 9333. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-23820-3
9. Bartocci, E., Majumdar, R.: Introduction to the special issue on runtime verification. Formal Methods Syst. Des. **51**(1), 1–4 (2017). https://doi.org/10.1007/s10703-017-0287-6
10. Bianculli, D., Krstic, S.: On the risk of tool over-tuning in run-time verification competitions (position paper). In: Reger and Havelund [34], pp. 37–40. https://easychair.org/publications/paper/N6cC
11. Colin, S., Mariani, L.: 18: Run-Time Verification. In: Broy, M., Jonsson, B., Katoen, J.-P., Leucker, M., Pretschner, A. (eds.) Model-Based Testing of Reactive Systems. LNCS, vol. 3472, pp. 525–555. Springer, Heidelberg (2005). https://doi.org/10.1007/11498490_24
12. Colombo, C., Falcone, Y.: First international summer school on runtime verification - as part of the ArVi COST action 1402. In: Falcone and Sánchez [19], pp. 17–20
13. Colombo, C., Falcone, Y.: Second school on runtime verification. In: Colombo, C., Leucker, M. (eds.) Runtime Verification - 18th International Conference, RV 2018, Limassol, Cyprus, 11–13 November, 2018, Proceedings. Lecture Notes in Computer Science, vol. 11237, pp. 27–32. Springer, Cham (2018)
14. Colombo, C., Pace, G.J.: Industrial experiences with runtime verification of financial transaction systems: lessons learnt and standing challenges. In: Bartocci and Falcone [5], pp. 211–232
15. Falcone, Y., Havelund, K., Reger, G.: A tutorial on runtime verification. In: Broy, M., Peled, D.A., Kalus, G. (eds.) Engineering Dependable Software Systems, NATO Science for Peace and Security Series, D: Information and Communication

Security, vol. 34, pp. 141–175. IOS Press (2013). https://doi.org/10.3233/978-1-61499-207-3-141
16. Falcone, Y., Kristc, S., Reger, G., Traytel, D.: A taxonomy for classifying runtime verification tools. In: Colombo, C., Leucker, M. (eds.) Proceedings of the 18th International Conference on Runtime Verification, Lecture Notes in Computer Science, vol. 11237, pp. 241–262. Springer, Cham (2018)
17. Falcone, Y., Mariani, L., Rollet, A., Saha, S.: Runtime failure prevention and reaction. In: Bartocci and Falcone [5], pp. 103–134
18. Falcone, Y., Nickovic, D., Reger, G., Thoma, D.: Second international competition on runtime verification CRV 2015. In: Bartocci and Majumdar [8], pp. 405–422. https://doi.org/10.1007/978-3-319-23820-3
19. Falcone, Y., Sánchez, C. (eds.): RV 2016. LNCS, vol. 10012. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46982-9
20. Falcone, Y., Sánchez, C.: Introduction to the special issue on runtime verification. Formal Methods Syst. Des. **53**(1), 1–5 (2018). https://doi.org/10.1007/s10703-018-0320-4
21. Hallé, S., Khoury, R., Gaboury, S.: A few things we heard about RV tools (position paper). In: Reger and Havelund [34], pp. 89–95. https://easychair.org/publications/paper/q246
22. Havelund, K., Goldberg, A.: Verify your runs. In: Meyer, B., Woodcock, J. (eds.) VSTTE 2005. LNCS, vol. 4171, pp. 374–383. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-69149-5_40
23. Havelund, K., Leucker, M., Reger, G., Stolz, V.: A shared challenge in behavioural specification (Dagstuhl seminar 17462). Dagstuhl Rep. **7**(11), 59–85 (2017). https://doi.org/10.4230/DagRep.7.11.59
24. Havelund, K., Reger, G.: Runtime verification logics a language design perspective. In: Aceto, L., Bacci, G., Bacci, G., Ingólfsdóttir, A., Legay, A., Mardare, R. (eds.) Models, Algorithms, Logics and Tools. LNCS, vol. 10460, pp. 310–338. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63121-9_16
25. Havelund, K., Reger, G., Thoma, D., Zalinescu, E.: Monitoring events that carry data. In: Bartocci and Falcone [5], pp. 61–102
26. Jakšić, S., Leucker, M., Li, D., Stolz, V.: COEMS - open traces from the industry. In: Reger and Havelund [34], pp. 96–105. https://easychair.org/publications/paper/QljX
27. Lahiri, S., Reger, G. (eds.): RV 2017. LNCS, vol. 10548. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-67531-2
28. Leucker, M., Schallhart, C.: A brief account of runtime verification. J. Logic Algebraic Program. **78**(5), 293–303 (2008). https://doi.org/10.1016/j.jlap.2008.08.004
29. Mallozzi, P., Pardo, R., Duplessis, V., Pelliccione, P., Schneider, G.: MoVEMo: a structured approach for engineering reward functions. In: Second IEEE International Conference on Robotic Computing (IRC 2018), pp. 250–257. IEEE Computer Society (2018). https://doi.org/10.1109/IRC.2018.00053
30. Margaria, T., Steffen, B. (eds.): ISoLA 2016. LNCS, vol. 9953. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-47169-3
31. Reger, G.: A report of RV-CuBES 2017. In: Reger and Havelund [34], pp. 1–9. https://easychair.org/publications/paper/MVXk
32. Reger, G., Hallé, S., Falcone, Y.: Third international competition on runtime verification - CRV 2016. In: Falcone and Sánchez [19], pp. 21–37
33. Reger, G., Havelund, K.: What is a trace? A runtime verification perspective. In: Margaria and Steffen [30], pp. 339–355. https://doi.org/10.1007/978-3-319-47169-3_25

34. Reger, G., Havelund, K. (eds.): RV-CuBES 2017. In: An International Workshop on Competitions, Usability, Benchmarks, Evaluation, and Standardisation for Runtime Verification Tools, Kalpa Publications in Computing, vol. 3. EasyChair (2017)
35. Reger, G., Rydeheard, D.: From parametric trace slicing to rule systems. In: Colombo, C., Leucker, M. (eds.) Proceedings of the 18th International Conference on Runtime Verification, Lecture Notes in Computer Science, vol. 11237, pp. 334–352. Springer, Cham (2018)
36. Reger, G., Rydeheard, D.E.: From first-order temporal logic to parametric trace slicing. In: Bartocci and Majumdar [8], pp. 216–232 https://doi.org/10.1007/978-3-319-23820-3_14
37. Rozier, K.Y.: On the evaluation and comparison of runtime verification tools for hardware and cyber-physical systems. In: Reger and Havelund [34], pp. 123–137. https://easychair.org/publications/paper/877G
38. Sánchez, C., et al.: A Survey of Challenges for Runtime Verification from Advanced Application Domains (beyond software) (2018, under submission)
39. Signoles, J.: Online runtime verification competitions: how to possibly deal with their issues (position paper). In: Reger and Havelund [34], pp. 157–163. https://easychair.org/publications/paper/m1vV