



HAL
open science

Second School on Runtime Verification, as part of the ArVi COST Action 1402 Overview and Reflections

Yliès Falcone

► **To cite this version:**

Yliès Falcone. Second School on Runtime Verification, as part of the ArVi COST Action 1402 Overview and Reflections. RV 2018 - 18th International Conference on Runtime Verification, Nov 2018, Limassol, Cyprus. pp.1-5. hal-01882413

HAL Id: hal-01882413

<https://inria.hal.science/hal-01882413>

Submitted on 27 Sep 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Second School on Runtime Verification, as part of the ArVi COST Action 1402

Overview and Reflections

Yliès Falcone

Univ. Grenoble-Alpes, CNRS, Inria, LIG, F-38000 France
yliès.falcone@univ-grenoble-alpes.fr

Abstract. This paper briefly reports on the second international school on Runtime Verification, co-organized and sponsored by Inria and COST Action IC1402 ArVi. The school was held March 19-21 2018, in Praz sur Arly (near Grenoble) in the French Alps. Most of the lectures dealt with introductory and advanced topics on Runtime Verification from the first tutorial book on Runtime Verification [2]. Additional lectures were given on cutting-edge research topics. We report the context and objectives of the school, overview its program, and propose outlooks for the future editions of the school.

1 Context and Objectives

Runtime Verification (RV) is the umbrella term to refer to the study of languages, (lightweight) techniques, and tools related to the verification of the executions of software and hardware systems against behavioral properties (see [5,10,11,14,7,4] for tutorials and overviews). Runtime Verification is a very effective technique to ensure that a system is correct, reliable, and robust. Compared to other verification techniques, RV is more practical than exhaustive verification techniques (e.g., model-checking, static analysis), at the price of losing completeness. Compared to conventional testing, RV is more powerful and versatile.

As a field of research, RV is endowed with a yearly conference¹, which exists since 2000. The field is getting more mature and diverse and the community is building documentation and lecture material to help students and practitioners entering the field. The international school on Runtime Verification constitutes one key element to facilitate the adoption of RV. This edition of the school shared the same objectives as the first edition [6] (which was held in Madrid as part of RV 2016 [9]):

- to present the foundations of the techniques;
- to expose participants to cutting-edge advances in the field;
- to provide a balance on theoretical and practical aspects of Runtime Verification;
- to adopt a hands-on approach and expose participants with the basics of building an RV tool.

¹ runtime-verification.org.

Table 1. Program overview - Day 1 - Monday 19th March

Time slot	Topic	Lecturer
08:00 10:00	<i>An Introduction to Runtime Verification and Monitorability</i> slides video	A. Francalanza
10:30 12:00	<i>Monitoring Cyber-Physical Systems</i> slides video	A. Donze
13:00 14:30	<i>The Java Modeling Language a Basis for Static and Dynamic Verification</i> slides video	W. Ahrendt
14:30 15:30	<i>Foundations on Runtime Verification</i> video	M. Leucker
16:00 16:45	<i>Monitoring Data Minimization</i> slides video	G. Schneider
16:45 17:30	<i>Runtime Assertion-Based Verification for Hardware and Embedded Systems</i> slides video	L. Pierre

Table 2. Program overview - Day 2 - Tuesday 20th March

Time slot	Topic	Lecturer
08:00 10:00	<i>Combined static and dynamic analyses in Frama-C: An Overview</i> slides video	N. Kosmatov
10:30 12:00	<i>A Hands-On Introduction to Building a Runtime Verification Tool</i> slides video	C. Colombo
14:00 17:00	Social event: Outing to Mont Blanc	

2 Presentation

The second edition of the school on Runtime Verification was sponsored by COST Action IC1402 ArVi², Inria³, and Persyval-Lab⁴.

The school was organised over three days with a series of lectures from international experts (see Tables 1-3). Lectures at the school ranged from the fundamentals of runtime verification to more practical aspects, but also covered cutting-edge research. All lectures were fully recorded. In Tables 1-3, below each lecture title, one can find 3 clickable links to the slides and videos.

In the remainder, we report on some of the lessons learned from the organization of the school and make suggestions to future organizers of the school.

² www.cost-arvi.eu

³ www.inria.fr

⁴ persyval-lab.org

Table 3. Program overview - Day 3 - Wednesday 21st March

Time slot	Topic	Lecturer
08:00 10:00	<i>Discovering Concurrency Errors</i> slides video	J. Loureno
10:30 12:00	<i>Stream Runtime Verification</i> slides video	C. Sanchez
13:00 14:30	<i>Industrial Experiences with Runtime Verification of Financial Transaction Systems: Lessons Learnt and Standing Challenges</i> slides video	C. Colombo

3 Reflections

We summarize the most frequent comments obtained from the participants. The balance between practice and theory was really appreciated by the participants (with application-oriented lectures very welcome). However, we note that some participants (legitimately) found the practice sessions too short and that monitoring for concurrency errors was under-represented (only 2 hours). Participants appreciated the format of the sessions: lectures last between one and two hours and focused on a topic. However, some participants with experience in RV would have preferred to opt for slightly more technical lectures with a greater focus on state of the art approaches, especially during the last sessions (as was actually the case with the first edition of the school). The participants appreciated the opportunities to discuss with experts and lecturers (thanks to the long breaks and social events); these opportunities allowed new connections to participants. Some participants suggested organizing additional group discussions in the form of panels dedicated to cutting-edge topics lead by the lecturers.

4 Outlooks

Given the great success of the last two editions, we hope that the school will proceed in the future. We make recommendations for future editions of the school. These recommendations are based on the experience gained from the organization of the two editions and the feedback received from the participants.

- First, we recommend that future editions of the school last longer: 3 to 4 full days would be the appropriate duration to dedicate one day to basics, one day to advanced topics, one day to cutting-edge research and recent results, and one day dedicated to tool construction (which is of importance for such a pragmatic technique as RV).
- Regarding practice sessions, we believe that sessions such as the ones organized during the first edition of the school would be more effective by allowing students to really address implementation issues. To save time, setup and configurations of the tools involved should be provided to the participants ahead of time. Using technologies such as virtual machines and containers is to be considered.

- Moreover, we would like to see a lecture dedicated to tool evaluation, detailing the methodology to compare a tool and assess its relevance.
- Furthermore, we would suggest future organizers to prepare overview slides providing participants with a big picture of the presented techniques. This shall help connect methods, techniques, and tools, and better see the complementarities. Similar to this is the concern of participants in seeing when to use RV compared to other techniques. Providing concrete example situations (upfront during the first lecture) where non-RV techniques fail and where RV techniques are complementary would clarify the position of RV in the big picture of verification techniques.
- To put the approaches in a better perspective, it would be nice to compare different approaches addressing the same problem. After an exposition of the approaches, a common interactive session could serve see the limits of each approach and study possible cross-fertilization between them.
- Finally, we would like to organize a small competition between the tools constructed by the students, in the same spirit as the competitions on Runtime Verification, usually held during the conference [1,8,13,3,12].

Acknowledgment

The organizer would like to warmly thank all the researchers for their lectures and all the participants to the winter school for their feedback.

The school lectures were based upon work from COST Action ARVI IC1402, supported by COST (European Cooperation in Science and Technology) and from the material in the tutorial book on introductory and advanced topics on Runtime Verification [2].

The organizer is grateful to several institutes who sponsored the school: the COST association, Inria, PERSYVAL-lab, and the Laboratoire d'Informatique de Grenoble.

The organizer is also much in debt with several fellows who largely contributed to making the school a successful event: Sophie Azzaro, Imma Presseguer, and Carmen Contamin for helping with logistic organization, and Djamel Hadji and Gilles Gardès for recording the lectures and handling all the audiovisual matters during the school.

References

1. Bartocci, E., Bonakdarpour, B., Falcone, Y.: First international competition on software for runtime verification. In: Bonakdarpour, B., Smolka, S.A. (eds.) Proc. of the Fifth International Conference on Runtime Verification (RV 2014). LNCS, vol. 8734, pp. 1–9. Springer (2014)
2. Bartocci, E., Falcone, Y. (eds.): Lectures on Runtime Verification - Introductory and Advanced Topics, Lecture Notes in Computer Science, vol. 10457. Springer (2018)
3. Bartocci, E., Falcone, Y., Bonakdarpour, B., Colombo, C., Decker, N., Havelund, K., Joshi, Y., Klaedtke, F., Milewicz, R., Reger, G., Rosu, G., Signoles, J., Thoma, D., Zalinescu, E., Zhang, Y.: 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)

4. Bartocci, E., Falcone, Y., Francalanza, A., Reger, G.: Introduction to runtime verification. In: Bartocci, E., Falcone, Y. (eds.) *Lectures on Runtime Verification - Introductory and Advanced Topics*, Lecture Notes in Computer Science, vol. 10457, pp. 1–33. Springer (2018)
5. Colin, S., Mariani, L.: Run-time verification. In: Broy, M., Jonsson, B., Katoen, J., Leucker, M., Pretschner, A. (eds.) *Model-Based Testing of Reactive Systems, Advanced Lectures* [The volume is the outcome of a research seminar that was held in Schloss Dagstuhl in January 2004]. Lecture Notes in Computer Science, vol. 3472, pp. 525–555. Springer (2004)
6. Colombo, C., Falcone, Y.: First international summer school on Runtime Verification - as part of the ArVi COST action 1402. In: Falcone, Y., Sánchez, C. (eds.) *Runtime Verification - 16th International Conference, RV 2016, Madrid, Spain, September 23-30, 2016, Proceedings*. Lecture Notes in Computer Science, vol. 10012, pp. 17–20. Springer (2016)
7. 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)
8. Falcone, Y., Nickovic, D., Reger, G., Thoma, D.: Second international competition on runtime verification CRV 2015. In: Bartocci, E., Majumdar, R. (eds.) *Proc. of the 6th International Conference on Runtime Verification (RV 2015)*. LNCS, vol. 9333, pp. 405–422. Springer (2015)
9. Falcone, Y., Sánchez, C. (eds.): *Runtime Verification - 16th International Conference, RV 2016, Madrid, Spain, September 23-30, 2016, Proceedings*, Lecture Notes in Computer Science, vol. 10012. Springer (2016)
10. Havelund, K., Goldberg, A.: Verify your runs. In: Meyer, B., Woodcock, J. (eds.) *Verified Software: Theories, Tools, Experiments, First IFIP TC 2/WG 2.3 Conference, VSTTE 2005, Zurich, Switzerland, October 10-13, 2005, Revised Selected Papers and Discussions*. Lecture Notes in Computer Science, vol. 4171, pp. 374–383. Springer (2005)
11. Leucker, M., Schallhart, C.: A brief account of runtime verification. *Journal of Logic and Algebraic Programming* 78(5), 293–303 (may/june 2008)
12. Reger, G.: A report of rv-cubes 2017. In: Reger, G., Havelund, K. (eds.) *RV-CuBES 2017. An International Workshop on Competitions, Usability, Benchmarks, Evaluation, and Standardisation for Runtime Verification Tools*, September 15, 2017, Seattle, WA, USA. *Kalpa Publications in Computing*, vol. 3, pp. 1–9. EasyChair (2017)
13. Reger, G., Hallé, S., Falcone, Y.: Third international competition on runtime verification - CRV 2016. In: Falcone and Sánchez [9], pp. 21–37
14. Sokolsky, O., Havelund, K., Lee, I.: Introduction to the special section on runtime verification. *STTT* 14(3), 243–247 (2012)