

# Preface to the JOT special issue on VORTEX at ECOOP 2022: selected papers

**Davide Ancona<sup>\*</sup> and Adrian Francalanza<sup>‡</sup>**

<sup>\*</sup>University of Genova, Italy

<sup>‡</sup>University of Malta, Malta

**ABSTRACT** In this preface, the editors present an overview of the topics and scope of the workshop on *Verification and Monitoring at Runtime Execution (VORTEX)*, and describe the editorial and reviewing process for its edition at ECOOP 2022. The papers selected for publication are presented and briefly described.

**KEYWORDS** runtime verification, runtime monitoring

## 1. About VORTEX

The 5th Workshop on Verification and Monitoring at Runtime Execution (VORTEX 2022) brings together researchers working on all aspects of Runtime Monitoring (RM) with emphasis on integration with formal verification and testing.

RM is concerned with the runtime analysis of software and hardware system executions in order to infer properties relating to system behavior. Example applications include telemetry, log aggregation, threshold alerting, performance monitoring and adherence to correctness properties (more commonly referred to as Runtime Verification).

RM has gained popularity as a solution to ensure software reliability, bridging the gap between formal verification and testing: on the one hand, the notion of event trace abstracts over system executions, thus favoring system agnosticism to better support reuse and interoperability; on the other hand, monitoring a system offers more opportunities for addressing error recovery, self-adaptation, and issues that go beyond software reliability.

The goal of VORTEX is to bring together researchers contributing on all aspects of RM covering and possibly integrating both theoretical and practical aspects, with particular focus on hybrid approaches inspired by formal methods, program analysis, testing.

### JOT reference format:

Davide Ancona and Adrian Francalanza. *Preface to the JOT special issue on VORTEX at ECOOP 2022: selected papers*. Journal of Object Technology. Vol. 23, No. 2, 2024. Licensed under Attribution 4.0 International (CC BY 4.0) <http://dx.doi.org/10.5381/jot.2024.23.2.e1>

## 2. Submission and review process

At the workshop six short papers have been presented. After the event, four works have been invited to contribute to this special issue with extended versions of the papers submitted to the workshop. All manuscripts underwent one round of revision before being accepted for publication.

## 3. Accepted papers

- *Towards a Security Framework for Artifact-centric Workflows Leveraging Runtime Enforcement*. Gaurav Gupta, Saumya Shankar, and Srinivas Pinisetty.

In this paper the authors design and develop a framework for specifying lifecycle constraints on documents, and generate enforcement monitors out of them, to prevent unauthorized changes to the document, and assure the system's safety.

- *Tooling Offline Runtime Verification against Interaction Models: recognizing sliced behaviors using parameterized simulation*. Erwan Mahe, Boutheina Bannour, Christophe Gaston, Arnault Lapitre, and Pascale Le Gall

This paper proposes an algorithm to verify the conformity of local traces against formal specifications, called Interactions, to overcome the absence of a global clock, and, hence, of global traces in distributed systems. The algorithm is based on parameterized simulation to reconstitute unobserved behaviors.

- *Toward Using Fuzzers and Lightweight Specifications to Reveal Semantic Bugs*. Amirfarhad Nilizadeh, Gary T. Leavens and David R. Cok

The authors investigate the possible advantages of combining different kinds of fuzzing tools and runtime assertion checking (RAC) in a complementary manner to leverage their benefits, along with a preliminary study. Input test generation can be managed by a fuzzing tool, while a RAC tool can ensure that such generated inputs are valid and, then, check the results for semantic bugs.

- *VSMoN: Runtime Monitoring Based Data-driven Remote Vital Sign Monitoring System*. Rahul Bharadwaj Pendyala, Abhinandan panda, and Srinivas Pinisetty

This paper introduces a domain-specific language (DSL) to describe healthcare policies. To ensure the correctness of the monitoring system, monitoring code is synthesized by translating policies expressed in the DSL to the Timed Automata formalism. As a proof of concept of the proposed approach, a prototype implementation of a client-server-based mobile application has been developed.

## 4. Acknowledgments

We would like to thank all authors, the Program Committee of VORTEX and the reviewers selected from the PC who managed the review process of this special issue. A special thank goes to Alfonso Pierantonio and Juri Di Rocco for supporting us in the preparation of the camera-ready versions of the papers.

### About the authors

**Davide Ancona** is associate professor at the University of Genova (Italy). His research interests include programming languages, runtime monitoring and verification, formal methods, Internet of Things systems. You can contact the editor at [davide.ancona@unige.it](mailto:davide.ancona@unige.it) or visit <https://person.dibris.unige.it/ancona-davide/>.

**Adrian Francalanza** is professor at the Computer Science Department of the University of Malta (Malta). His research interests include programming language semantics, software verification and concurrency. You can contact the editor at [adrian.francalanza@um.edu.mt](mailto:adrian.francalanza@um.edu.mt) or visit <https://www.um.edu.mt/profile/adrianfrancalanza>.