# Preface

Rosemary Monahan, Virgile Prevosto and José Proença

## Keynote: What is KeY's Key to Software Verification?

Wolfgang Ahrendt

## Experience Report: Towards Moving Things with Types - Helping Logistics Domain Experts to Control Cyber-Physical Systems with Type-Based Synthesis

Jan Bessai, Moritz Roidl and Anna Vasileva

## Automated Deductive Verification for Ladder Programming

Denis Cousineau, David Mentré and Hiroaki Inoue

## Deeply Integrating C11 Code Support into Isabelle/PIDE

Frédéric Tuong and Burkhart Wolff

## A Component-Based Formal Language Workbench

Peter D. Mosses

## An Integrated Development Environment for the Prototype Verification System

Paolo Masci and César A. Muñoz

## The TLA+ Toolbox

Markus Alexander Kuppe, Leslie Lamport and Daniel Ricketts

## Simulation under Arbitrary Temporal Logic Constraints

Julien Brunel, David Chemouil, Alcino Cunha and Nuno Macedo

## Tool Support for Validation of Formal System Models: Interactive Visualization and Requirements Traceability

Eduard Kamburjan and Jonas Stromberg
F-IDE 2019 is the fifth international workshop on Formal Integrated Development Environment, held on October 7, 2019 in Porto, Portugal, as part of the FM 2019 World Congress on Formal Methods.

High levels of safety, security and also privacy standards require the use of formal methods to specify and develop compliant software (sub)systems. Any standard comes with an assessment process, which requires a complete documentation of the application in order to ease the justification of design choices and the review of code and proofs. Ideally, an F-IDE dedicated to such developments should comply with several requirements. The first one is to associate a logical theory with a programming language, in a way that facilitates the tightly coupled handling of specification properties and program constructs. The second is to offer a language/environment simple enough to be usable by most developers, even if they are not fully acquainted with higher-order logics or set theory, in particular by making development of proofs as easy as possible. The third is to offer automated management of application documentation. It may also be expected that developments done with such an F-IDE are reusable and modular. Tools for testing and static analysis may be embedded within F-IDEs to support the assessment process. The workshop is a forum of exchange on different features related to F-IDEs.

We solicited several kinds of contributions: research papers providing new concepts and results, position papers and research perspectives, experience reports, tool presentations. The workshop was open to contributions on all aspects of a system development process, including specification, design, implementation, analysis and documentation. The current edition is a one-day workshop with eight communications, offering a large variety of approaches, techniques and tools. Each submission was reviewed by three reviewers.

We also had the honor of welcoming Wolfgang Ahrendt, from Chalmers University of Technology, who gave a keynote entitled What is KeY’s key to software verification?

The committees of F-IDE 2019 were composed of:

**Steering Committee:**
- Catherine Dubois, Samovar/ENSIIE
- Paolo Masci, HASLab/INESC-TEC and Universidade do Minho
- Dominique Méry, LORIA/Université de Lorraine

**PC Co-chairs:**
- Rosemary Monahan, Maynooth University
- Virgile Prevosto, Institut List, CEA, Université Paris-Saclay
- José Proença, HASLab/INESC-TEC and CISTER/ISEP, Portugal

**Program Committee:**
- Cinzia Bernardeschi (University of Pisa)
- José Creissac Campos (University of Minho)
- Paul Curzon (Queen Mary University of London)
- Damien Doligez (Inria)
- Andrea Domenici (University of Pisa)
- Carlo A. Furia (Chalmers University of Technology)
- Kenneth Lausdahl (Aarhus University)
- Stephan Merz (Inria Nancy)
- Stefan Mitsch (Carnegie Mellon University)
What is KeY's Key to Software Verification?

Wolfgang Ahrendt (Chalmers University of Technology, Sweden)

F-IDE Keynote

KeY is a deductive software verification approach and system, whose most elaborate version targets Java programs. In a recent KeY case study, which attracted attention also outside formal method circles, verification with KeY could reveal a bug in the main sorting routine of OpenJDK. While this talk will also cover the user interface of KeY, the focus of the discussion is more fundamental. KeY follows to a significant extent principles which are different from other deductive verification systems, on the level of the program logic, the proof calculus, the interaction with the prover, the transparency of proofs, and the usage of back-end solvers. In this talk, I will discuss the impact of these aspects, with a special focus on usability. In addition, we will look at how the design of the logic and calculus influenced the integration with other validation techniques, like test generation and runtime verification.