

From the Guest Editors of the Special Issue on Best Practice in Robotic Software Development

Nico Hochgeschwender¹ Roland Philippsen²

¹ Bonn-Rhein-Sieg University, Sankt Augustin, Germany

² Halmstad University, Halmstad, Sweden

INTRODUCTION

We are happy to present you the special issue on Best Practice in Robot Software Development of the Journal on Software Engineering for Robotics! The spark for this special issue came during the eighth workshop on Software Development and Integration in Robotics (SDIR) at the 2013 IEEE International Conference on Robotics and Automation. The workshop focused on Robot Software Architectures, and the fruitful discussions made it clear that the design, development, and deployment of robot software is always an interplay between competing aspects. These are often couched in antagonistic pairs, such as dependability versus performance, and prominently include quality attributes as well as functional, non-functional, and application requirements.

A considerable amount and variety of experience has been accumulated to master this interplay. All too frequently, however, these insights remain hidden behind the scenes. Frequently, avoidable mistakes thus get repeated, and “the wheel” is reinvented. A perceived lack of proper venues for publication and valorization appears to be one of the causes. The SDIR workshop series, and particularly the related open-access peer-reviewed Journal of Software Engineering for Robotics, aim to fill this gap and promote the synergy between software engineering and robotics.

This special issue reports on behind-the-scenes insights from software engineering in robotics projects whose success is demonstrated in scientific articles or other reusable format, such as tools, frameworks, systems, or products. The seven contributions present robotic software engineering knowledge in reusable form, such as descriptions of lessons learned, patterns and anti-patterns, general guidelines, and re-usable models.

OVERVIEW OF CONTRIBUTIONS

Coleman et al. contribute two key elements for **producing reusable complex robotic software**, focusing on reducing the barrier to entry. The work is based on the experience of creating, maintaining, and extending a framework for motion planning, manipulation, 3D perception, kinematics, control and navigation. The paper provides insights into why robotic software is particularly challenging in this respect, how these challenges can be overcome with the help of the proposed design principles, as well as a description of a tool that automatically tunes a concrete complex software pipeline for arbitrary robots.

The contribution by Vanthienen, Klotzbuecher, and Bruyninx is based on extensive experience with **designing, using, maintaining, and finally re-developing a framework** for constraint-based robotic task specification, execution, and monitoring. Among several lessons learned, they particularly emphasize and discuss the importance of adhering to the established design principle of separating concerns, and as a core contribution propose a composition pattern that has proven helpful for designing application-specific system architectures.

The paper on software reuse for in-hand manipulation by Walck et al. describes the **lessons learned from leveraging an existing middleware and package ecosystem** in order to build and control a hand and arm system. It summarizes core concepts, relates them to component-based design patterns and their application, and describes best practices to take advantage of the provided flexibility, interoperability, and reconfigurability. A detailed discussion of integration choices rounds off the paper, with an emphasis on interfacing, coordinating, and configuring software components for a dexterous robot.

A generic multi-sensor fusion framework for localization and odometry is presented by Cucci and Matteucci. They focus on a modular formulation based on factor graphs, with the aim

of incorporating differing rates, arbitrary sensors, and various target applications. They **discuss design choices and implementation trade-offs** with respect to ad-hoc solutions, and introduce an architecture with decoupled interfaces between key aspects such as solvers, state variables, and error models.

The paper on self-driving miniature cars by Berger provides a detailed study of how small-scale experimental platforms can bridge the gap between simulation-based prototyping and resource intensive implementation and testing on full-sized vehicles. It illustrates the advantage of **clearly separating and leveraging appropriately chosen tools** for algorithm design, validation, implementation, and real-world experimentation. The main contributions lie in a systematic investigation of design decisions for a particular miniature car competition, and the related push toward an experimental platform with standardized hardware and software interfaces.

Paikan et al. describe a mechanism to help strike the proper **balance between reusability, generality, and application specific needs**. Port monitors and port arbitrators are introduced to encapsulate coordination and related data processing. They rely on runtime scripting and first-order logic, to transparently insert coordination aspects into a component. This significantly increases the support for reusability, and reduces the communication requirements of decision making.

Ramaswamy et al. describe a **systematic, model-based approach for dealing with the uncertain problem space** and huge solution space appearing in the development of robot architectures. They introduce a Solution Space Modeling Language which supports not only specifying the solution space, but also helps to identify variation points and to make design decisions explicit and tractable. All models are demonstrated and validated by modeling and realizing the solution space of a real-world vehicle tracking system.

ACKNOWLEDGEMENTS

We thank everyone who contributed, starting with the participants of the SDIR workshop where our community clearly agreed not only on the need, but also signalled its readiness to participate in the effort, of this special issue and JOSER in general. Together, we are shaping the methods, formats, and priorities of a community which spans many interests and backgrounds. We thank the authors of all submitted manuscripts for their effort and trust that those whose papers which did not make the cut will find the review comments a valuable help. We are extremely grateful to the reviewers who took the time to thoroughly assess the contributions and provide such detailed and helpful comments to the authors.