Journal of Software Engineering for Robotics 8(1), December 2017, 1-2
ISSN: 2035-3928

Editorial to the Special Issue on Robotic Computing

Davide Brugali[1] Bruce A. MacDonald[2]
[1] University of Bergamo, Italy
[2] University of Auckland, New Zealand

INTRODUCTION

The First IEEE International Conference on Robotic Computing (IRC 2017) was held in Taichung, Taiwan, April 10-12, 2017. The new IRC conference series addresses the synergetic interaction of computing technologies and robotic technologies. While it is true that computing technologies are integral parts of any autonomous robotic system, they are often considered ancillary to their development and find little attention in traditional robotic conferences. Much of the value of robotic systems is realized by computing technologies that grow more complex and present significant design, certification, and operational challenges for researchers and engineers to address.

The IRC conference series aims at providing a forum where experts in the fields of Computer Science and Robotics can exchange experiences and ideas on the effective interplay of Robotic Computing technologies.

On one hand computers continue to be humanized and a large number of cyber-physical systems are being developed to act upon the physical world. On the other hand the robotic community is looking ahead to robots of the 21st century that are versatile computing machines with high social impact potential, such as enhancing transportation safety, reducing agricultural pesticide use, providing human like companionship, and improving public safety and crime-fighting efficacy, among other things.

The barriers that restrain robotic system diffusion significantly correlate to the complexity of designing, developing and operating their software control systems, which must be reliable, maintainable, intelligent, and safe. While the popularity and expectations of robotic systems grow rapidly on the current wave of technology improvements, robotic computing design tools can be significantly improved if they are to well support large scale robot deployment.

IRC covers broad and diverse topics from a variety of Computer Science research areas that are relevant for Robotics, such as Software Engineering, Knowledge Engineering, Computer Security, Semantic Computing, Human-computer interaction.

This special issue specifically focuses on the interplay between Software Engineering and Robotics.

Editorial paper *Digital Object Identifier: 10.6092/JOSER_2017_08_01_p1*

OVERVIEW OF CONTRIBUTIONS

Eleven full papers and two short papers presented at IRC 2017 have been selected for this special issue of the International Journal of Software Engineering for Robotics (JOSER). The selected papers have been extended with at least 30% new and unpublished material and underwent a rigorous extra refereeing and revision process.

There are there important areas addressed by the papers in the special issue. **Model driven engineering (MDE)** is an active area for research and expected to be an essential capability for engineering robotic software as robots become more widely used in more diverse tasks in offices, homes, factories and outdoors. Carefully devised models will be converted automatically to robust, reliable, adaptable software for applications in all these areas. More high level and sophisticated software **frameworks** will be required for integrating complex libraries, components, middleware and systems. Rich **middleware** will be required for connecting a variety of diverse software capabilities across diverse software and hardware environments.

Five papers examine different aspects of MDE. Adam *et al* propose designs to enable a modular toolchain for, to enhance compatibility between MDE tools, and promote reuse at the model level. Ingibergsson *et al* consider safety certification of robot software, a growing concern and an essential requirement for practical robots increasingly operating in human spaces. The focus is on expressing safety constraints for vision systems in a domain specific language, so that perception system faults can be managed for safety. Bardaro *et al* propose a generalised architecture analysis and design language for modelling ROS nodes, which can be extended to other architectures. Wigand *et al* present a method for composition

of domain specific languages, to create an MDE system, and promote reuse of the different domain specific code segments. Ramaswamy *et al* argue for a more coherent practice with some formalization and standardization of MDE architectures, to reduce barriers to creating tools and frameworks.

Five papers are more focused on framework design. Wong *et al* show how to transform a high level task description in to a correct–by–construction ROS application. Sandretto *et al* show how to manage uncertainties based on an interval analysis framework, enabling formal verification of robotic controllers. Romano *et al* present an abstraction for whole–body controllers that separates the control software from the implementation details, where no assumptions are made about the control law. Schierl *et al* address geometric modelling that enables different robots to have a common approach to the surrounding scene geometry estimation, and separating logical relationships. The method facilitates multiple robot coordinated manipulation. Ahn *et al* present a robotic software manager that enables integration across multiple and disparate robotic software frameworks, improving reuse and flexibility.

Three papers are focused on the middleware level. Doose *et al* describe the design of a new middleware that focuses on software reconfiguration in real time according to the prevailing constraints, thereby increasing the robustness of the robotic system. Reichardt *et al* address the importance of quality in robotic software, as tradeoffs across quality constraints, with a modular design. Muratore *et al* introduce a new middleware that meets real time requirements, and provides a flexible API for integration with different frameworks.

These papers illustrate the breadth, depth and complexity of issues to be addressed in the software engineering areas of robotic computing. In all cases further research is warranted and it may be some time before a large and coherent tool set coalesces from the broad and diverse research fronts that are being advanced by researchers.

ACKNOWLEDGEMENTS

We would like to thank all authors who submitted papers, and the reviewers for their efforts and high quality reviews. Finally, we would like to thank the library staff of the University of Bergamo for their excellent support to the archiving of the articles in this special issue.

Please enjoy the articles in this special issue, and feel free to distribute its contents as far and widely as possible.