Proceedings of the 2018 Winter Simulation Conference
M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, eds.

SIMULATION-DRIVEN EMBEDDED CONTROL OF ROBOTIC SYSTEMS BASED ON MODEL CONTINUITY

Ezequiel Pecker Marcosig

Departamento de Ingeniería Electrónica
FI, UBA and CONICET
Av. Paseo Colón 850
Buenos Aires, C1063ACV, ARGENTINA

ABSTRACT

Designing hybrid controllers for cyber-physical systems raises the need to interact with embedded platforms, where robotic applications are a paradigmatic example. This can become a difficult, time consuming and error-prone task for non-specialists as it demands for background on low-level software/hardware interfaces often falling beyond the scope of control designers. We propose a simulation-driven methodology and tool for designing hybrid controllers based on a model continuity approach. The simulation model of a controller evolves transparently from a desktop-based mocking up environment until its final embedded target without the need of intermediate adaptations. We rely on the Discrete EVent Systems Specification (DEVS) framework for robust modeling and real-time simulation of hybrid controllers, and on the Robotic Operating System (ROS), a middleware for flexible abstraction of sensors and actuators. We successfully tested our approach in scenarios where custom-made robots are designed concurrently with their controllers.

1 INTRODUCTION AND MOTIVATION

The efficient development of embedded controllers for Cyber-Physical Systems (CPS) is under unprecedented pressure, driven by an upsurge of markets combining the Internet of Things and flexible production systems. Salient properties in this setting are the uncertainty and rapid evolution of requirements, pushed by a pervasive availability of cheap, yet powerful technology for sensors, actuators, and embedded computing platforms.

For decades the software engineering community devoted tremendous efforts in creating formal methods and tools to develop controllers for embedded systems, in particular for those of hybrid nature and with real-time constraints. Modern design methods for hybrid controllers tend to rely on unified modeling frameworks, which capture and combine together the expressive power of well-known modeling techniques. Yet, most existing methods are still heavy, expensive and hard to scale up for real applications. M&S techniques offer increasingly attractive capabilities to allow for rapid, yet robust prototyping and final delivery of embedded controllers. In particular the Discrete EVent Systems Specification (DEVS) combines discrete event, discrete time, and continuous dynamics under a mathematically sound way. DEVS-based methodologies for M&S-driven engineering have been successfully implemented by integrating software development best practices to offer product life-cycle control. This becomes particularly relevant in scenarios where it is difficult to predict systems behavior as changes are introduced very frequently.

Hybrid controllers for resource-constrained CPSs running on embedded platforms call for balancing the allocation of resources between so called *cyber* and *physical* controllers. This can be performed by varying the execution rate of both tasks. However, this adaptation requires guarantees on the closed-loop stability, which demands controller gains adaptations (Pecker Marcosig et al. 2014).

Our approach aims at an end-to-end methodology for designing hybrid controllers based on *model continuity* for DEVS. Therefore, the developed DEVS simulation controller model does never leave the simulator and evolves transparently from a desktop-based mocking up environment until its final embedded target without the need of intermediate recoding nor reimplementation. In this context, the DEVS simulation engine plays the role of a ”simulation virtual machine” for the DEVS model. Key advantages of the DEVS engine are its compactness and efficiency, making it suitable for real-time scenarios. Model continuity is a sound approach to mitigate the introduction of errors in the development process of controllers. The model continuity strategy as well as the DEVS framework are being recognized as candidate technologies for CPSs (recently validated by Pecker Marcosig et al. 2017).

Yet, we believe that the current weakest link between DEVS and embedded simulation lies in the connection between models and lower level sensors and actuators. This link represents a key interface layer between software abstractions and the physical platform. Typically, device drivers should cope with this requirement. Yet, being low-level platform-specific artifacts, and given the changes of platform inherent to model continuity, reprogramming or replacing device drivers can be a heavy process demanding too specialized skills (often beyond those found in control designers).

2 THE DEVS-OVER-ROS (DOVER) FRAMEWORK

We proposed and developed DoveR, a conceptual and practical framework helping to remove the previous obstacle, by combining real-time DEVS simulation with the Robot Operating System (ROS). ROS is a massively adopted middleware library for developing robotic applications. We chose the ROS middleware to deal with low level intricacies at the hardware/software interface, and make it as modular, reusable, and transparent as possible to both a DEVS simulator and a DEVS modeler.

We chose the PowerDEVS simulator for hybrid systems modeling and real-time simulation. PowerDEVS is also suitable for non-DEVS experts, hiding away the internals of the formalism when needed. The control engineer can then focus on the performance of the controller being designed, without a need of being a DEVS expert or a ROS developer.

The extensions developed for the PowerDEVS engine abide by the standards of a DEVS real-time abstract simulator, assimilating smoothly the message-based communication with ROS nodes via UDP sockets. Therefore, ROS messages play the role of external events in the DEVS framework. We successfully verified the efficacy and flexibility of our methodology by applying it to a real, custom-made robotic platform (Pecker Marcosig et al. 2018).

3 CONCLUDING REMARKS AND FUTURE RESEARCH

DoveR proved a software framework that fosters simulation-model continuity to develop hybrid embedded controllers for cyber-physical systems, with a focus in robotic applications. The framework is accompanied with a sample end-to-end methodology where the simulator plays the role of a real-time virtual machine for the simulation models. Future research includes a systematic characterization of DoveR latencies, a study of communication alternatives between PowerDEVS and ROS, and a better harnessing of ROS tools.

REFERENCES

Pecker Marcosig, E., F. Felicioni, and A. Zanini. 2014. “Self-Triggering Based on Lyapunov with Adaptive Control Law for WNCS”. *IEEE Latin America Transactions* 12(7):1242–1249.

Pecker Marcosig, E., J. I. Giribet, and R. Castro. 2017. “Hybrid Adaptive Control for UAV Data Collection: A Simulation-based Design to Trade-off Resources Between Stability and Communication”. In *Proceedings of the 2017 Winter Simulation Conference*, edited by W. K. V. Chan et al., 1704–1715. Piscataway, NJ, USA: IEEE.

Pecker Marcosig, E., J. I. Giribet, and R. Castro. 2018. “DEVS-over-ROS (DoveR): A framework for Simulation-Driven Embedded Control of Robotic Systems based on Model Continuity”. In *Proceedings of the 2018 Winter Simulation Conference*, edited by M. Rabe et al. Piscataway, NJ, USA: IEEE.