Proceedings of the 2019 Winter Simulation Conference
N. Mustafee, K.-H.G. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y.-J. Son, eds.

DEVELOPMENT OF A REAL-TIME DEVS KERNEL: RT-CADMIUM

Ben Earle
Kyle Bjornson
Gabriel Wainer

Department of Systems and Computer Engineering
Carleton University
1125 Colonel By Dr.
Ottawa, ON, Canada

ABSTRACT

With the rise of Internet of Things devices, there is an increasing demand for embedded control systems. Discrete-Event Modeling of Embedded Systems (DEMES) is a Discrete Event System Specification (DEVS) based model driven development methodology that increases reliability and improves time to market by simplifying the development and testing of embedded systems. There are problems with the existing toolchain that is being used to implement this design methodology which have been addressed and improved in Real-Time Cadmium (RT-Cadmium), which is a Real-Time (RT) DEVS kernel developed on top of the Cadmium DEVS Simulator. RT-Cadmium is a complete DEVS simulator and RT kernel, allowing users to simulate and deploy their project with ease. RT-Cadmium already supports MBed Enabled ARM microprocessors, however the process for porting it to new target platforms is simple and documented. RT-Cadmium can also handle asynchronous events, which do not exist in standard DEVS simulators.

1 INTRODUCTION

Discrete-Event Modeling of Embedded Systems (DEMES) is a Discrete Event System Specification (DEVS) based model driven development methodology that aims to simplify development and testing of embedded systems. In DEMES the user models their control system and its environment as DEVS models to be simulated before deploying the exact same models to control the hardware (Wainer and Castro, 2011). Therefore, DEMES requires a DEVS simulator that can use the same models in a bare metal RT DEVS kernel deployed on the target system.

The existing tool suite used for DEMES development uses CD-Boost as the DEVS simulator and ECD-Boost as the RT-DEVS kernel (Niyonkuru and Wainer, 2016). The (E)CD-Boost toolset has certain drawbacks, such as, limited message types, only being able to insatiate each atomic model once, and nonoptimized performance. These issues have all been addressed in a new DEVS simulator: Cadmium (Belloli et al., 2019). Additionally, ECD-Boost does not allow users to easily switch between simulation and deployment, it is not simple to port the tool to new physical targets, and it does not support asynchronous events. Real-Time Cadmium (RT-Cadmium) is a new RT-DEVS kernel based on Cadmium, that solves these issues.

2 RT-CADMIUM DESIGN GOALS

The RT-Cadmium kernel was designed with four main requirements: (1) it must be backward compatible with all existing Cadmium models, (2) it must conserve the DEVS simulator structure, (3) the models must have identical behavior when simulated and deployed on hardware, and (4) it must be easy compile the

project for the simulator and the RT target. The tool has already been configured to run in RT on both Linux and M-Bed enabled ARM microprocessors. Implementation details can be found in "Real-Time Cadmium Development Branch" (Earle and Bjornson).

Interchangeability of simulation and deployment: DEVS models are developed incrementally in an agile fashion. For this reason, it must be simple to switch back and forth between testing a project on the simulator and deploying it to hardware. This switch is not easy using the previous tool set. It was designed to be used in a waterfall development fashion, where the models are made and simulations are done in CD-Boost. Then, when the user is satisfied, they make a new ECD-Boost project that will reuse these models adding some features to make them compatible with the Target Platform. This workflow is not realistic and drives most users to skip simulating their system all together. RT-Cadmium is designed in such a way that encourages users to follow the DEMES workflow. In contrary, RT-Cadmium projects can be both simulated and deployed; the only difference is a compiler flag. RT-Cadmium also comes with visualizations tools to help debug simulated models, and runtime model verification (inherited from the Cadmium base).

Platform portability: To facilitate the growing number of embedded platforms, the RT-DEVS tool should be portable to many platforms. ECD-Boost was written solely for ARM platforms without considering portability. RT-Cadmium uses abstraction to remain hardware independent, requiring only three components to port it to a new a target system: A C++17 Compiler for the target, a RT Clock class relating simulation time to microseconds, and DEVS models that encapsulate IO drivers. To demonstrate the portability, these components have been developed for ARM MBed and standard Linux systems.

Asynchronous event handler: Any platform for embedded system development must support asynchronous events. Standard DEVS simulators (ECD-Boost included) do not support this, as all DEVS events are deterministic from the perspective of the simulator (i.e. they need to be defined at run time before the model goes to sleep). Any time a DEVS model goes to sleep, it will always return to the simulator the time when it should be woken up again. An asynchronous model, for example a model of an interrupting input, cannot tell the RT engine when it will generate its next output. RT-Cadmium uses an observer pattern to add support for asynchronous events, while maintaining the top down coordinator tree structure implemented in standard DEVS simulators. This solution lets the asynchronous atomic model notify the DEVS engine of the event at the top coordinator level.

3 CONCLUSION AND FUTURE WORKS

RT-Cadmium is a new RT-DEVS tool, designed to improve on ECD-Boost, as a platform for embedded systems development, following the DEMES methodology. Using the lessons learned from ECD-Boost, RT-Cadmium is more user friendly, easier to port, and supports asynchronous events. Existing models developed for this tool and are available in "Real-Time Cadmium Sample Models" (Earle, Bjornson).

The next steps for this research are to create a complex sensor driver workflow, explaining how to encapsulate advanced sensors in DEVS models. Additionally, we are interested in developing distributed DEVS systems that pass DEVS messages over serial connections / RF links.

REFERENCES

Belloli, L., D. Vicino, C. Ruiz-Martin, and G.Wainer. 2019. "Building DEVS Models with the Cadmium Tool". *In Proceedings of the 2019 Winter Simulation Conference*, edited by N. Mustafee, K.-H.G. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y-J. Son, eds. (In Press). National Harbor, Maryland: Institute of Electrical and Electronics Engineers, Inc.

Earle B., K. Bjornson. "Real-Time Cadmium Development Branch" Published on GitHub https://github.com/SimulationEverywhere/cadmium/tree/RT_DEVS_Development.

Earle B., K. Bjornson. "Real-Time Cadmium Sample Models". Published on GitHub https://github.com/SimulationEverywhere/RT-Cadmium-Models

Niyonkuru, D., and G. Wainer. 2016. "A Kernel for Embedded Systems Development and Simulation Using the Boost Library". *2016 Symposium on Theory of Modeling & Simulation (TMS/DEVS),* Pasadena, CA, USA Article No. 13. San Diego, CA, USA: Society for Computer Simulation International/

Wainer, G., and R. Castro. 2011. "DEMES: A Discrete-Event Methodology for Modeling and Simulation of Embedded Systems". *Modeling and Simulation Magazine*, April 2: 65–73