MDE for Real-time Embedded Systems with Papyrus-RT

Nicolas Hili and Juergen Dingel

School of Computing
Queen’s University
Kingston, Ontario, Canada
{hili, dingel}@cs.queensu.ca

Abstract

The demo will sketch the use of Papyrus-RT for the model-driven development of C++ code for real-time embedded systems. The demo will show how to develop models expressed in a UML profile called UML-RT and how to use them to generate executable code. Extensions of Papyrus-RT to support animation, simulation, and software development for Raspberry Pi-based rovers will be briefly discussed.

Eclipse Papyrus for Real-Time (Papyrus-RT)

Papyrus-RT is based on the Papyrus/Eclipse platform and was designed to be extensible, allowing users to add, with relative ease, their own customizations or extensions. Its target audiences are industrial developers who want to build custom solutions, researchers who want to prototype and evaluate novel techniques, and educators who want to teach students the strengths and weaknesses of modelling and Model-Driven Engineering (MDE).

Fig. 1 gives an overview of Papyrus-RT. The central area is used for graphically editing models through several diagrams. Switching between diagrams can be done using the tab bar below the graphical editing area. A palette is used for creating and for adding model elements directly to the diagrams. It has been stripped down compared to the UML2 palette and only shows the relevant UML for Real-Time (UML-RT) concepts. Below the editing area, a property view can be used for editing properties of the selected model elements. On the left side, a model explorer is fully synchronized with the graphical editing area and displays a tree view of the currently edited model. It supports most of the basic actions such as element creation or deletion. Using the model explorer, code can be generated partially or entirely. Projects and generated code can be visualized in the project explorer.

Compared to previous UML-RT tools, Papyrus-RT provides fully synchronized graphical and textual editors for UML-RT, allowing developers to choose their preferred representation without having to commit. Users can move back
Figure 1: Papyrus-RT Workbench

and forth between the two representations and code generation is possible from both. Moreover, Papyrus-RT provides full code generation and a run-time system to deploy fully functioning applications on a target platform. Its code generator is incremental, so the entire code does not have to be regenerated every time a change is made to the model. Instead, the code generator isolates the part of the code affected by a change to the model in order to re-generate it. In addition, the code generator was designed to be customizable, for example to support code instrumentation or alternative code patterns. Additional features include code-assist and import capabilities of legacy models from proprietary tools. More detailed information about UML-RT and Papyrus-RT can be found at [1, 2, 3].

Acknowledgments: This work is supported by Ericsson Canada, EffiCiOS, and the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

