

On Extending JGrafchart with Support for FMI for Co-Simulation

Alfred Theorin Charlotta Johnsson

Department of Automatic Control, Lund University, Lund, Sweden

Sequential Function Charts (SFC) is one of the IEC 61131-3 PLC standard languages for sequential, parallel, and general state-transition oriented automation applications. SFC is supported by most large industrial automation systems, for example 800xA by ABB, SIMANTIC S7 by Siemens, RSLogix 5000 by Rockwell Automation, DeltaV by Emerson, and CENTUM CS by Yokogawa. SFC is widely used and accepted for industrial automation, but is a low level programming language and thus implementing larger applications in SFC is inconvenient.

Grafchart has been developed for automation applications with focus on scalability. It extends SFC with high level features such as hierarchical structuring, reusable procedures, and exception handling. This makes it convenient to implement large applications that are overviewable and maintainable. Grafchart and SFC have the same graphical syntax with steps, possible application states, and transitions which change the application state.

JGrafchart [1] is a free integrated development environment for the Grafchart language. It is a research and education tool which is used for instance in the EU/GROWTH project CHEM for control in process industry, the EU FP7 project ROSETTA for robotic assembly, a master's thesis for modeling of avionics systems, and laboratory exercises on sequential and batch control.

JGrafchart can be connected to external environments through a multitude of customizable input/output (I/O) integration capabilities and can thus be used to control external real and simulated processes. As there is no built-in support for any particular simulation environment there is much potential for improvement in terms of effort for specifying the simulated model, quality of the models, support for inspecting simulation results, and time required to simulate, especially for more complicated physical systems.

Functional Mock-up Interface (FMI) is a standard which aims at combining dynamic models developed in various tools. A tool can export a model as a Functional Mock-up Unit (FMU) which can then be combined with other FMUs to compose the whole system. The FMI standard consists of two parts, namely *FMI for Model Exchange* and *FMI for Co-Simulation*. The difference is that for *FMI for Co-Simulation* a FMU also includes an individual solver to simulate its behavior. *FMI for Co-Simulation* enables simulation of coupled technical systems with focus on time-dependent problems.

Extending JGrafchart with built-in support for *FMI for Co-Simulation* gives more and better opportunities to connect JGrafchart to other simulation capable tools. It also makes a state machine related language with roots in industrial automation and with high level language features such as object orientation, hierarchical structuring, code reuse, and exception handling available for FMI co-simulation.

In this paper adding *FMI for Co-Simulation* support to JGrafchart is conceptually evaluated. It is discussed how JGrafchart fits into the *FMI for Co-Simulation* framework and potential ways to implement this are discussed.

References

- [1] Lund University. *JGrafchart*. URL: <http://www.control.lth.se/Research/tools/grafchart.html> (visited on 2014-01-26).