System-Level Modeling and Simulation with Intel® CoFluent™ Studio

Anthony Barreteau

Abstract Intel® CoFluent™ Studio is a visual model-driven development (MDD) solution for creating executable specifications of complex systems. It can be used at any point of the project lifecycle for modeling and validating any electronic or information systems in any application domain: hardware block, software stack, System-on-Chip (SoC), mixed hardware/software embedded system, networked/distributed system, end-to-end Internet-of-Things (IoT) infrastructure and Big Data networks. Intel CoFluent Studio can predict performance data from the application and use cases model execution on a multicore/multiprocessor platform model. Intel CoFluent Studio is a system modeling and simulation toolset based on Eclipse. Models are captured in graphical diagrams using Intel CoFluent optimized domain-specific language (DSL) or standard UML notations—a combination of SysML and the MARTE profile. ANSI C or C++ is used as action language to capture data types and algorithms. Non-functional system requirements or model calibration data such as execution durations, power, or memory values, are added through model attributes. Models are translated into transaction-level modeling (TLM) SystemC code for execution. The SystemC code is instrumented and generates traces that can be monitored with various analysis tools. Fast host-based simulations allow designers to observe the real-time execution of their application models on multiprocessor/multicore platform models. Performance figures such as latencies, throughputs, buffer levels, resource loads, power consumption, memory footprint, and cost can be extracted.

We will present this system-level technologies and associated methodology with a poster. The scope of the poster is related to the two following topics in technical and scientific methods:

A. Barreteau (✉)
CoFluent Technology Center, 3 Rue Alfred Kastler, Immeuble 'Le Saphir',
Nantes 44300, France
e-mail: anthony.barreteau@intel.com

G. Auvray et al. (eds.), *Complex Systems Design & Management*,
DOI 10.1007/978-3-319-26109-6_32

– Systems architecture (needs capture, requirements development, systems modelling, simulation, optimization, sizing and specification, architectural frameworks).
– Systemic tools (configuration management, system behaviour analysis tools, modeling and simulation tools, test management).

Keywords System-level modeling · Executable specifications · Use-cases modeling · Performance prediction