Demo

Modelling Avionics Applications with CONCERTO

Andrea Baldovin

Alessandro Zovi

Eduardo Tovar*

Geoffrey Nelissen*

Tullio Vardanega

*CISTER Research Center

CISTER-TR-141211

2014/12/02

Modelling Avionics Applications with CONCERTO

Andrea Baldovin, Alessandro Zovi, Eduardo Tovar*, Geoffrey Nelissen*, Tullio Vardanega

*CISTER Research Center
Polytechnic Institute of Porto (ISEP-IPP)
Rua Dr. António Bernardino de Almeida, 431
4200-072 Porto
Portugal
Tel.: +351.22.8340509, Fax: +351.22.8321159
E-mail: baldovin@math.unipd.it, emt@isep.ipp.pt, grrpn@isep.ipp.pt
http://www.cister.isep.ipp.pt

Abstract

No abstract, one page paper.

Modelling Avionics Applications with CONCERTO

Andrea Baldovin, Alessandro Zovi, Tullio Vardanega
Department of Mathematics
University of Padova
{baldovin, azovi, tullio.vardanega}@math.unipd.it

Geoffrey Nelissen, Eduardo Tovar
CISTER/INESC-TEC, ISEP
Polytechnic Institute of Porto
{grrpn, emt}@isep.ipp.pt

The demonstration presents some highlights of the CONCERTO toolset [1], a model-driven development tool environment developed in the CONCERTO project funded by the Artemis JU. CONCERTO builds on the foundations of the CHESS project [2] to target multicore platforms for a large set of industrial domains. See http://www.concerto-project.org/ for a round-up of the CONCERTO vision.

The demonstration focuses on the avionics domain, where conformance to the IMA (Integrated Modular Avionics) [3] reference architecture uses time and space partitioning to achieve sufficient isolation for applications sharing physical resources. The demonstration shows the support offered to the CONCERTO user for modelling IMA partitions and for their deployment on a multicore processor with reduced effort when compared to current industrial practice. This is showcased through the following steps:

- *Partition modelling.* System partitions are defined in IMA as stand-alone applications; that is they aggregate processes (the equivalent of threads in ARINC-653 speak) which concur to implement part of the whole system behaviour. To this end, CONCERTO introduces the concept of component hierarchy, not only to encapsulate pure functional logic but also to aggregate finer-grained components recursively. This approach promotes a hybrid development process by allowing users to define partitions as either an aggregate of component instances (bottom-up) or as a top-down artefact that can be seen as the parent of an aggregate of child components that can be further instantiated as needed within an active partition. The interesting result here is that the modelling of IMA partitions in CONCERTO is shown to be fully compatible with the component model supported in CHESS.
- *Partition decoration.* Partitions are enriched with additional information to express extra-functional properties that have to apply to their deployment and execution, notably execution-time budgets and space constraints (i.e., core assignment and memory partition). In this step, the user is requested a minimal set of basic inputs, which the CONCERTO tool environment treats to automatically calculate as good configuration parameters as possible.
- *Partition schedule generation.* Taking the system specification defined in the former steps as an input, CONCERTO automatically computes a possible (feasible and efficient) partition schedule. The end user, who is ultimately responsible for the system under design, is given the possibility of accepting or iterating over the schedule generation process.
- *Partition allocation to cores.* In a partitioned multicore scenario, partitions should be statically assigned for execution to available cores. This choice has direct influence on system performance and may incur inefficient utilisation of system resources. For this reason, CONCERTO provides guidance by suggesting a possible allocation of partitions to cores, which maximises system utilisation. Similarly to the schedule generation step, the end user is given the option to either accept the proposed allocation or to modify it according to needs or preferences.
- *Model analysis and back propagation.* After the design and the semi-automatic system configuration, an analysis step is performed to check the system model for inconsistencies that may undermine its validity. For example, schedulability analysis may reveal deadline overruns incurred by time-critical operations. Results are propagated back to the model level and its individual entities to highlight the areas of highest criticality, where user intervention is required to fix misconfigurations and to ensure system coherence before actual deployment. To this end, the MAST scheduling analysis tool [5] is connected to the modelling environment. MAST has been enhanced to support multicore platforms and the IMA hierarchical scheduling model. This modified version of MAST calculates useful information such as the worst-case response time and the worst-case blocking time suffered by the scheduled processes. Those values are then compared with the timing constraints of the application.
- *Code generation and execution.* As a last step, after the model has been validated, the target-specific code for the system is generated from the platform specific model (PSM). Execution is eventually performed on TiCOS [4], a time-composable ARINC-compliant operating system developed at the University of Padova.

References

[1] CONCERTO Project Homepage, *http://www.concerto-project.org/*, 2014.
[2] CHESS Project Homepage, *http://www.chess-project.org/*, 2014.
[3] Radio Technical Commission for Aeronautics, *Integrated Modular Avionics (IMA) Development Guidance and Certification Considerations*, 2005.
[4] A. Baldovin, E. Mezzetti and T. Vardanega, *A time-composable operating system*, in Proceedings of the 12th International Workshop on Worst-Case Execution Time Analysis, 2012.
[5] MAST Homepage, *http://mast.unican.es/*, 2014.

Work supported by: ARTEMIS/0003/2012, JU grant nr. 333053 (CONCERTO)