Towards AADL to SystemC mapping for partitioned systems

Michael Lafaye
Etienne Borde
Laurent Pautet
Marc Gatti
Presentation of a First Mapping Prototype:

**AADL to SystemC for Avionics Partitioned Systems**

Michael Lafaye  
lafaye@telecom-paristech.fr

Etienne Borde  
borde@telecom-paristech.fr

Laurent Pautet  
pautet@telecom-paristech.fr

Marc Gatti  
Marc-j.gatti@fr.thalesgroup.com
Objectives of this presentation

- Present the context in which this work started
- Describe first results and perspectives
- Give feedback in terms of usability of AADL with respect to simulation objectives
The work presented hereafter is realized in a partnership between THALES Avionics and TELECOM ParisTech.

The main objective is to improve early validation processes by the means of simulation techniques (early means here: without the concrete realization of the HW platform).

To be consistent with current industrial practices, the solution should be integrated in a Model Driven Engineering (MDE) approach.
Industrial Context: Avionics

IMA = integrated network architecture, composed of modules communicating through a deterministic network (ARINC664) and hosting some applications

+ reduces the number of modules of the platform and thus reduces space, weight and costs
- increases development and certification complexity

Need early platform model to:
- Allow customer and platform engineers to anticipate on platform performances
- check the compliancy between a proposal architecture and the system requirements
Problem Statement

- Current MDE approaches focus on the description of systems at high level of abstraction
  - System requirements
  - Functional decomposition
  - Physical decomposition (SW+HW)

- But at physical decomposition level, MDE often
  - focuses on software and deployment description, while hardware is modelled as black boxes with a few properties
  - performs static analysis, without correlation between components (SW/HW) interactions and applicative time frames

→ Problem: perform early dynamic performances analysis, with different levels of details for the execution platform model
General approach

Propose a simulation framework to:

1. Capture stimuli originated by software components onto hardware components (probabilistic, or based given a specific scenario, …)
2. Design libraries of hardware components (for simulation)
3. Integrate in a deployment model, software stimuli and hardware components
4. Simulate the final system and compare to system requirements
Virtual Integration Based on MDE

- Produce a simulation environment from a deployment model that enables to
  - Extend the model of the execution platform, (to answer the trades-off between simulation precision and simulation time).
  - Relate simulation results with execution time frames (to take more advantage of these simulation results)
AADL was chosen because

- It is dedicated to the description of Real-time and Embedded Systems
- It defines HW and SW components with extension mechanisms (property sets)
- It provides a specific annex (arinc653)
- It is particularly adapted for deployment modelling (Binding properties)
Simulation environment: SystemC/TLM

- IEEE standard for execution platform description
- some abstraction levels (functional, transaction, bit, transistors gate)
- widely used in industry
- particularly adapted for low-level hardware description
Simulation Construction Process

Study phase 3

- HW Platform data-sheets
- SW application characteristics
- Model transformation, profiling

Study phase 1: being finalized

- HW Platform AADL model
- SW Application AADL model
- Code generation

Study phase 2

- Component selection
- Component configuration
- Interconnected SystemC HW components
- Configurable SystemC SW instructions
- Code generation

Extensible SystemC HW components Repository

- Deployment AADL instance model
- Assembled SystemC (HW/SW) code

Simulation results related to time

AADL meeting, Paris

2011/06/07
HW SystemC model Generation

1. Repository content

2. Example

3. Configuration process

AADL meeting, Paris 2011/06/07
4. Refinement process

1. Add a SystemC component in the repository
2. Enrich AADL properties to configure it:
   a) `<component_type>_kind`
   b) `supported_<component_type>_kind`
   c) `Component kind dependent properties`

5. Example: DRAM

1. Add a SystemC DRAM component in the repository
2. Enrich AADL property sets for a memory component:
   a) `memory_kind => "DRAM"`;
   b) `Supported_memory_kind => {"DRAM,ROM,NVRAM"}`
   c) `Integer properties “refresh_period” and “refresh_time” represent constants in the SystemC component`
Avionic system made of 3 partitions

Physical isolation

Temporal isolation
Case-Study (physical structure)

- 1 physical system (board) made of
  - 3 memories
  - 2 processors
  - 5 I/O devices
Simulation Results

Partitions execution

Memory accesses and partitions execution

I/O accesses and partitions execution
Require to define few new properties for HW-PF modelling

Extensibility: `<component_type>_kind` is used to refine the component category `supported_<component_type>_kind` to list the type of supported refinement. For instance:

- `supported_memory_kind => (ROM, DRAM)`
- `Memory_kind=>ROM`

Processor Component: `cycle_per_instruction` : average number of clock cycles used to execute an instruction

We defined a first set of properties for the first version of the repository (DRAM, ROM, NVRAM, etc…)

To be evaluated when it comes to modeling low-level SW/HW interactions
Conclusion and Perspectives

- A first plug-in mapping AADL HW components to SystemC has been realized
  - The goal is not to integrate SystemC into AADL, but to use AADL for (i) integration of SystemC components, and (ii) producing interaction path between those components.

- A lot of work still has to be done
  - Deal with capture of software/hardware components interactions
  - Enrich the database with more precise SystemC HW components
  - Experiment the process on different HW platforms and applicative scenario
  - Compare simulation results with measurements on the real platform

- First experiments give promising results