AADLv2, a Domain Specific Language for the Modeling, the Analysis and the Generation of Real-Time Embedded Systems

Frank Singhoff+, Jérôme Hugues*

+University of Brest, Lab-STICC/CNRS UMR 6285, France
***ISAE, France**

About the presenters

- Jérôme Hugues: from ISAE, leads the Ocarina project, a AADL tool chain, member of the steering committee of SAE AS-2C, 8+ years on AADL

- Frank Singhoff: from UBO, leads the Cheddar project, a scheduling analysis tool, voting member of SAE AS-2C committee on AADL since 2008.

Resources for this tutorial

- Information on AADL
  - http://www.aadl.info : updates on AADL standard
  - http://www.openaadl.org : resources around Ocarina
  - http://beru.univ-brest.fr/~singhoff/cheddar/: Cheddar and real-time scheduling
- Materials for this tutorial (slides, examples, etc.) are on http://www.openaadl.org
- Feel free to contact us for more details

We focus on Real-Time, Critical, Embedded Systems

- *« The correctness of the system depends not only on the logical result of computation, but also on the time at which the results are produced »* Stankovic, 1988.
- Properties we look for:
  - Functions must be predictable: the same data input will produce the same data output.
  - Timing behavior must be predictable: must meet temporal constraints (e.g. deadline).

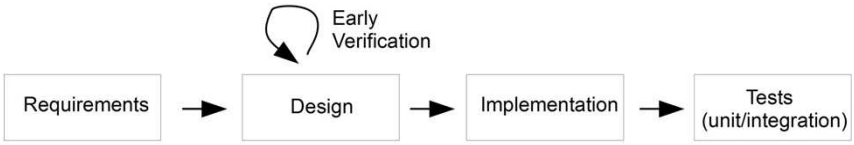
We focus on Real-Time, Critical, Embedded Systems

- **Critical real-time systems:** temporal constraints MUST be met, otherwise defects could have a dramatic impact on human life, on the environment, on the system,
- **Embedded systems:** computing system designed for specific control functions within a larger system.
  - Often with temporal constraints.
  - Part of a complete device, often including hardware and mechanical parts
  - Limited amount of resources.

We focus on Real-Time, Critical, Embedded Systems

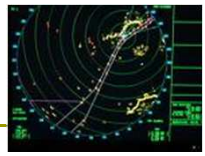
- Examples : aircraft, satellite, automotive, …
- Need to handle time. Concurrent applications.
- May have dramatic impact on human life, on the system, ...
- Do not allow software maintenance => difficult to correct erroneous software/bugs.
- High implementation cost : temporal constraints verification, safety, dedicated hardware/software

We focus on Real-Time, Critical, Embedded Systems



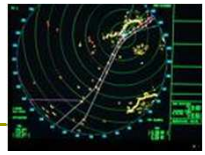
- **Specific software engineering** methods/models/tools to master quality and cost
  - Example : early verifications at design step

Objectives of this tutorial



- Issues
  - How to model/design a real-time critical embedded system that conforms to requirements?
  - How to verify the solution?
  - How to prototype/implement it?
- One solution amoung others: use an architecture description language
  - to model the system,
  - to run various verification,
  - and to automatically produce the system
- Focus on the AADL2.1 SAE standard

Objectives of this tutorial



- Goal: to model a simple radar system
- Let us suppose we have the following requirements

1. System implementation is composed by physical devices (Hardware entity): antenna + processor + memory + bus
2. and software entities : running processes and threads + operating system functionalities (scheduling) implemented in the processor that represent a part of execution platform and physical devices in the same time.
3. The main process is responsible for signals processing : general pattern: transmitter -> antenna -> receiver -> analyzer -> display
4. Analyzer is a periodic thread that compares transmitted and received signals to perform detection, localization and identification.
5. [..]

Outline

Goal: introduce model-based analysis of embedded real-time critical systems using the AADLv2 Architecture Description Language

- **Part 1: Introduction to AADLv2 core (about 60')**
  - Syntax, semantics of the language
- **Part 2: introducing a case study (about 30')**
  - A radar illustrative case study
- **Part 3: Scheduling analysis (about 45')**
  - Introducing real-time scheduling and its use with AADL
- **Part 4 : code generation (about 45')**
  - How to generate code from an AADL model and how to run it