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

Julien Delange, CMU/SEI, USA
Jérôme Hugues, ISAE, France

Software Engineering Institute
Carnegie Mellon

ISAE
Institut Supérieur de l'Aéronautique et de l'Espace

Copyright & Acknowledgments

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM-0002585

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**

- **Julien Delange: from CMU/SEI, author of the ARINC653 annex document of AADL and active contributor 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

- **Materials for this tutorial (slides and models) are on http://www.openaadl.org**

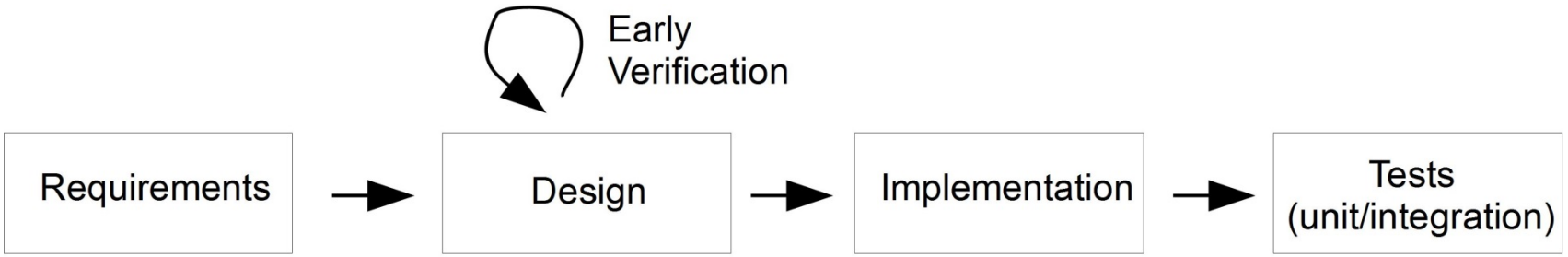
- **Feel free to contact us for more details**

Real-Time, Safety-critical, Embedded Systems

- ***"Real-time systems are defined as those systems in which 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.**
- **"A safety-critical systems is a computer, electronic or electromechanical system whose failure may cause injury or death to human beings."**
- **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),
  - Failure rates must be accounted for

Typical design flow

- **Rely on models and domain-Specific system/software engineering methods, processes and tools to master quality and cost so as to**
  - » Allow for early verifications at design step
  - » Reduce manual development efforts
  - » Ensure development consistency



Objectives of this tutorial

- **Issues**
  - How to model/design a real-time critical embedded system that conforms to requirements?
  - How to analyze the solution from a safety perspective
  - How to design and produce a *good* architecture?
- **One solution among 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 (2012)**

Objectives of this tutorial (cont'd)

- **Goal: to model an avionics-related system (ADIRU)**
- **Let us suppose we have the following requirements**
  - System implementation is composed by physical devices (i.e. Hardware entities) 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.
  - The main process is responsible for signals processing :
    - General pattern is sensing/processing/actuating
  - This system has to operate in an avionic platform, and demonstrate it is safe and secure, extends to both scheduling and safety analysis
  - The system is to be run on an ARINC653 OS
  - The system must be easy to design, understand and maintain

Outline

- **Goal: use AADLv2 for safety analysis and code generation**

- **Part 1: Introduction to AADLv2 core (~ 50')**
  - Syntax, semantics of the language
- **Part 2: introducing a case study (~ 20')**
  - A avionics case study (ADIRU)
- **Part 3: ARINC653 modeling with Code Generation (~ 45')**
  - Modeling ARINC653 systems with specific requirements
  - Generate Module and Partition Code for ARINC653 OS
- **Part 4: Reducing Architecture Complexity with AADL (~ 20')**
  - Detecting and avoiding complex architecture patterns using AADLv2
- **Part 5 : Error-Model Modeling (~ 45')**
  - Specify errors and faults propagations policy
  - Generate safety analysis documents (FHA, FMEA, FTA, etc.)