USING AADL TO SUPPORT AVIONICS ARCHITECTURE REFINEMENT AND SELECTION

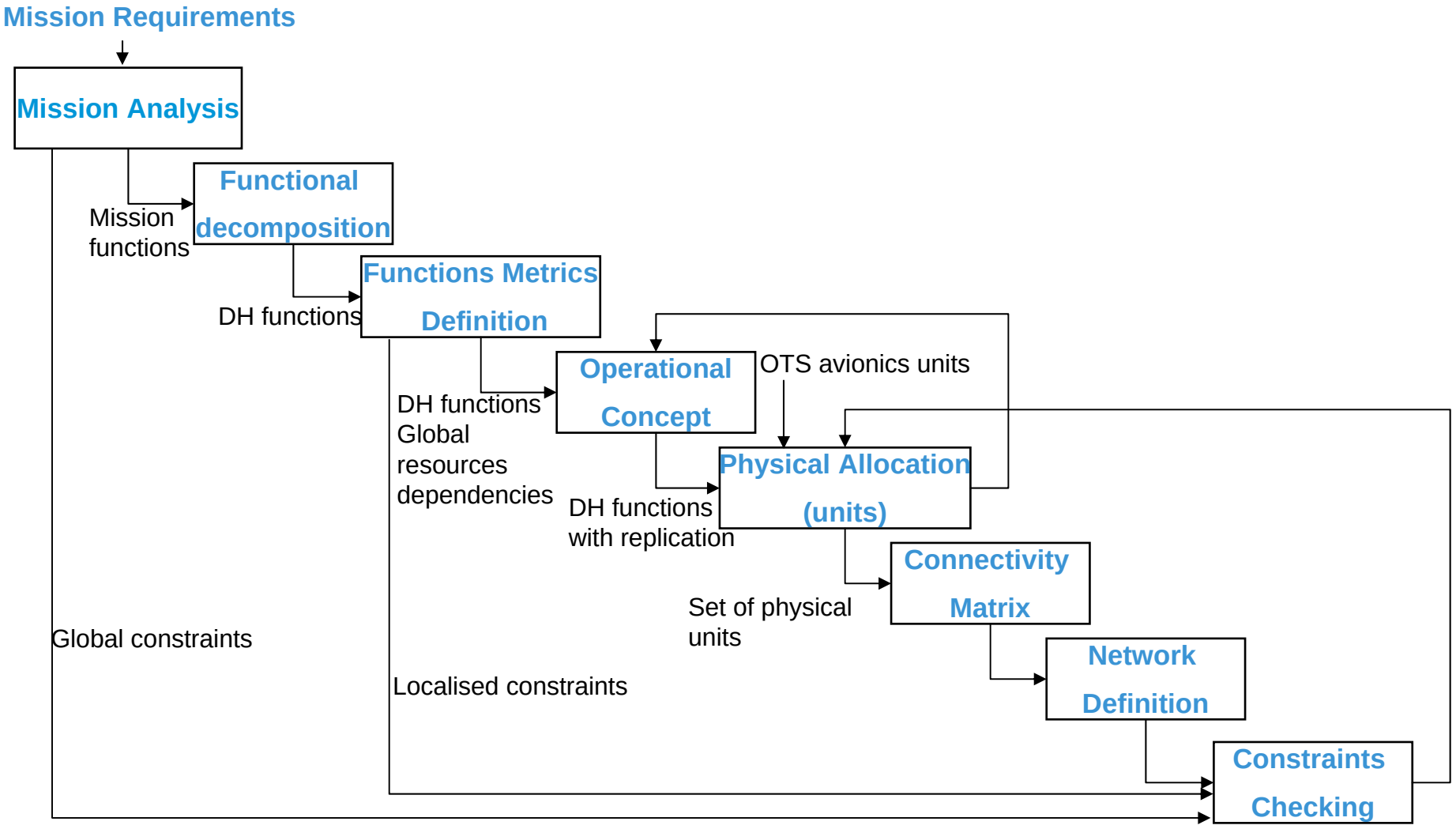
Julien Delange, Andreas Jung,

Ana-Elena Rugina, Jean-Francois Soucaille

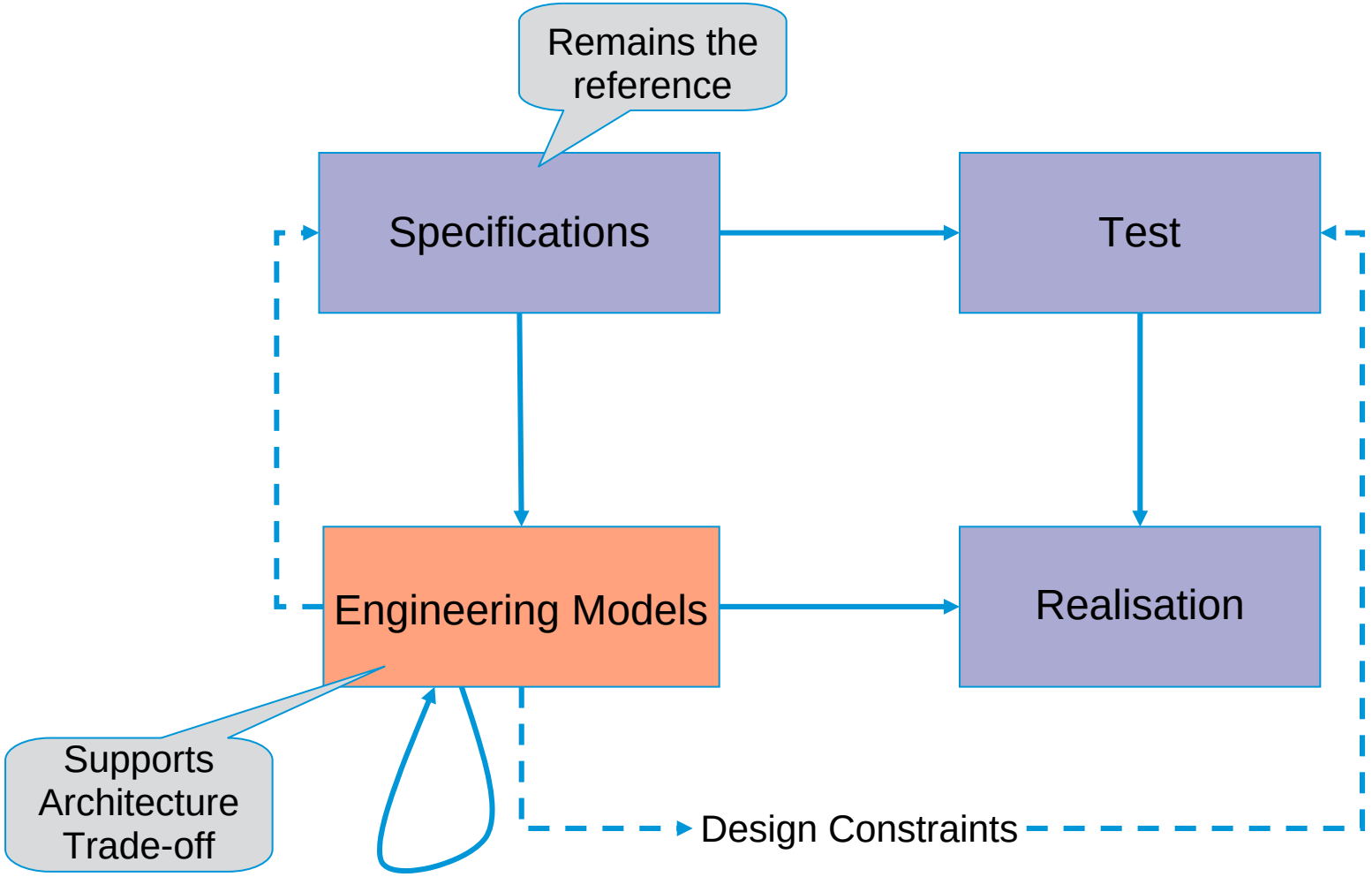
Outline

- The method for avionics architecture selection
- AADL
- “ARAM” Study
- “Guidelines for the Selection of Architectures” Study
- Feedback

Avionics Architecture Selection Method



Process Considerations



AADL Overview

- Architecture description language standardized by the SAE
  - v1 in 2004, v2 in 2009
- AADL model
  - 14 component categories in v2 (10 categories in v1)
    - Software: process, subprogram, subprogram group, data, thread, thread group
    - Platform: processor, virtual processor, memory, device, bus, virtual bus
    - Composite: system
    - Abstract: abstract
  - Textual and graphical syntax
  - Composition, interconnection
  - Two levels of description: type and implementation
  - Reconfiguration through operational modes
  - Mechanisms for refinement & architectural patterns
    - Inheritance
    - Abstract components and features
  - Extensible through properties and annexes
    - For analysis, code generation, …

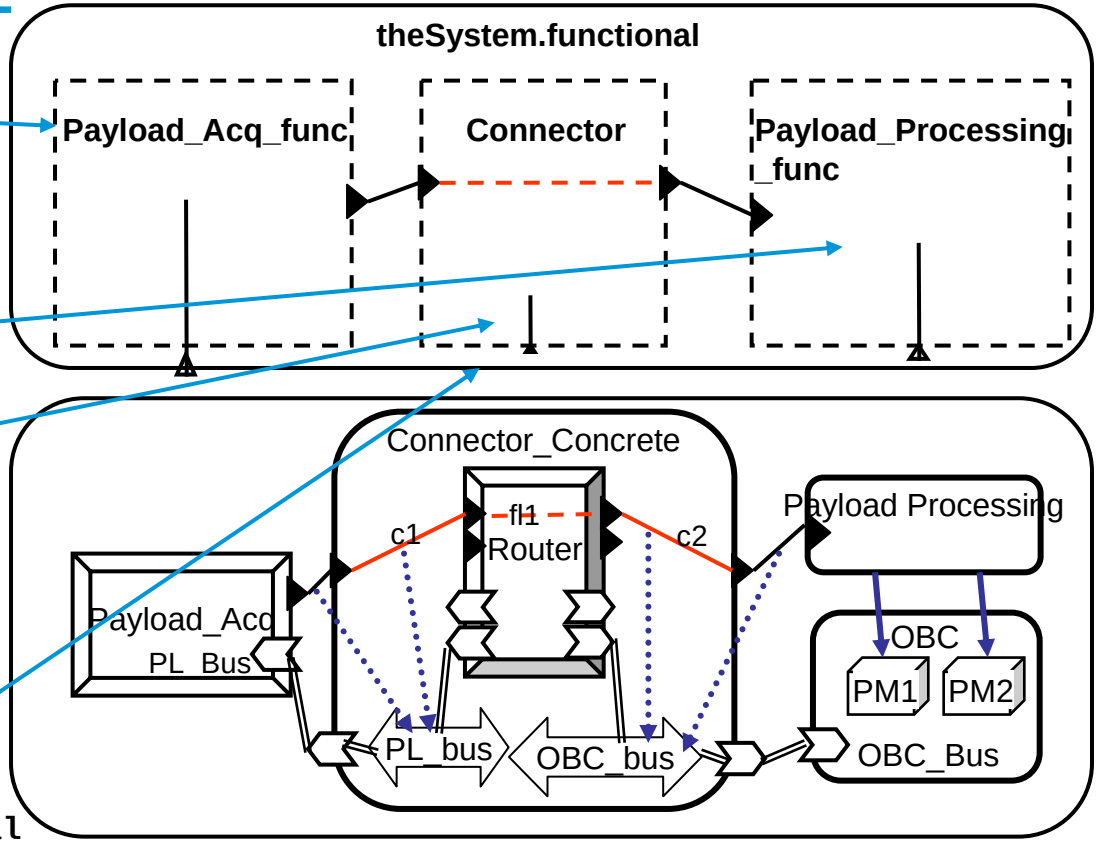
Example in AADL

```
ABSTRACT Payload_Acq_func
  FEATURES
    MissionData: OUT DATA PORT;
END Payload_Acq_func;

ABSTRACT Payload_Processing_func
  FEATURES
    MissionData: IN DATA PORT;
END Payload_Processing_func;


ABSTRACT Connector
  FEATURES
    MissionDataIn: IN DATA PORT;
    MissionDataOut: OUT DATA PORT;
  FLOWS
    fl1: FLOW PATH MissionDataIn
                   -> MissionDataOut;
END Connector;

SYSTEM IMPLEMENTATION theSystem.functional
  SUBCOMPONENTS
    Payload_Acq : ABSTRACT Payload_Acq_func;
    Payload_Processing : ABSTRACT Payload_Processing_func;
    Connector: ABSTRACT Connector;
  CONNECTIONS
    C1: PORT Payload_Acq.MissionData -> Connector.MissionDataIn;
    C2: PORT Connector.MissionDataOut -> Payload_Processing.MissionData;
END theSystem.functional;
```



ARAM

- Objectives
  - Support for the SAVOIR group
    - Consortium of ESA and space industry
    - Definition of avionics architecture reference
  - Investigation of AADL modelling
    - How to represent the reference architecture in AADL (generic architecture patterns)
    - Assessment of existing tools (modelling & validation)
- Case study
  - GAIA Payload
- Tools
  - REAL (Requirements Enforcement and Analysis Language)
    - Definition of theorems
    - Computation engine to inspect the AADL model and check it against the theorems
  - Objective to develop gateways to external validation tools (FDIR, RAMS, resources dimensioning) → COMPASS

Guidelines for the Selection of Architectures

- Objectives
  - Formalization of a process for DH architecture refinement and selection
  - Implementation of the defined process using AADL
  - Demonstration on real-scale case study
- Case study
  - Complete Solar Orbiter Data Handling Architecture
- Tools
  - Used: ADELE4.3.1 (graphical editor), OSATE2.0 (textual editor), Topcased-Req (traceability between textual requirements and model)
  - Identified
    - Power & mass analysis
    - Reliability and availability analyses
    - Resources analysis (memory and CPU load)
    - Data latency: coarse (based on latency annotations on flow paths) and fine-grained (based on behavioural descriptions of components)
    - Bus load: coarse (based and bandwidth budgets on flow paths and capacities) and fine-grained (to support for example the frame definition)
    - Design consistency and correctness checks

Feedback

- AADL language
  - Clean support for avionics architecture refinement
    - Including allocation of abstract functions to HW or SW components
  - Annotations for analyses
    - Proposals of sets of AADL properties to support the suggested analyses
  - Identification of design patterns for each step in the process
  - Identification of necessary improvements
    - Inheritance mechanisms
    - Flow paths

- Tools
  - The editors are not mature enough
  - Constraints checking tools for AADL models are not sufficient
  - “AADL-like” languages focusing on specific analysis aspects → bridges?, merging?

- Conclusions and open questions
  - Currently AADL cannot be operationally used for avionics architecture refinement and trade-off analyses (no library of characterized components, not mature enough tools)
  - If the above issues are solved, AADL is perceived as a good candidate for capturing in a central repository all the information necessary for performing architectural tradeoffs
  - How to transition to the SW and HW design processes?