A MDD Methodology for Specification of Embedded Systems and Automatic Generation of Fast Configurable and Executable Performance Models

Int. Conf. on HW/SW codesign and HW synthesis

(**CODES-ISSS 2012**)

Embedded System Week

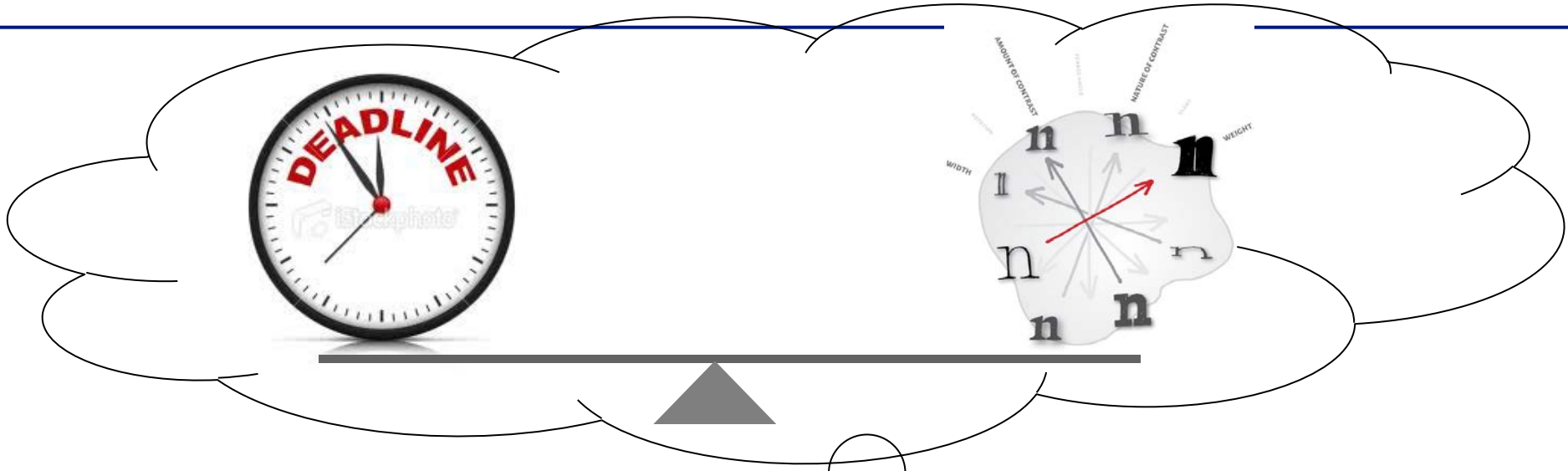
(**ESWeek 2012**)

Tampere, Finland

F.Herrera, H.Posadas, P. Peñil, E. Villar

F. Ferrero, R. Valencia

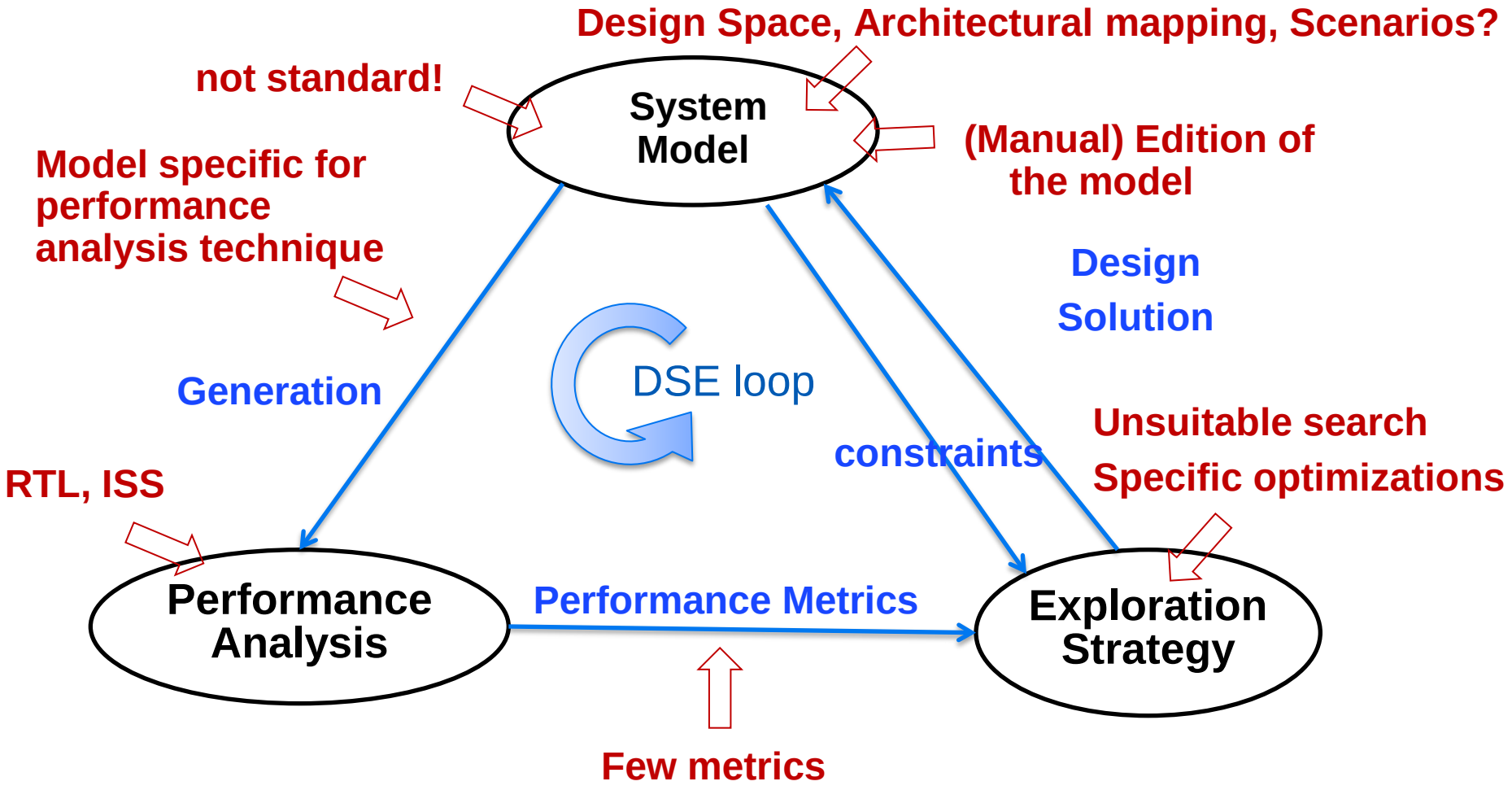
2 Introduction: The challenge for the System Designer



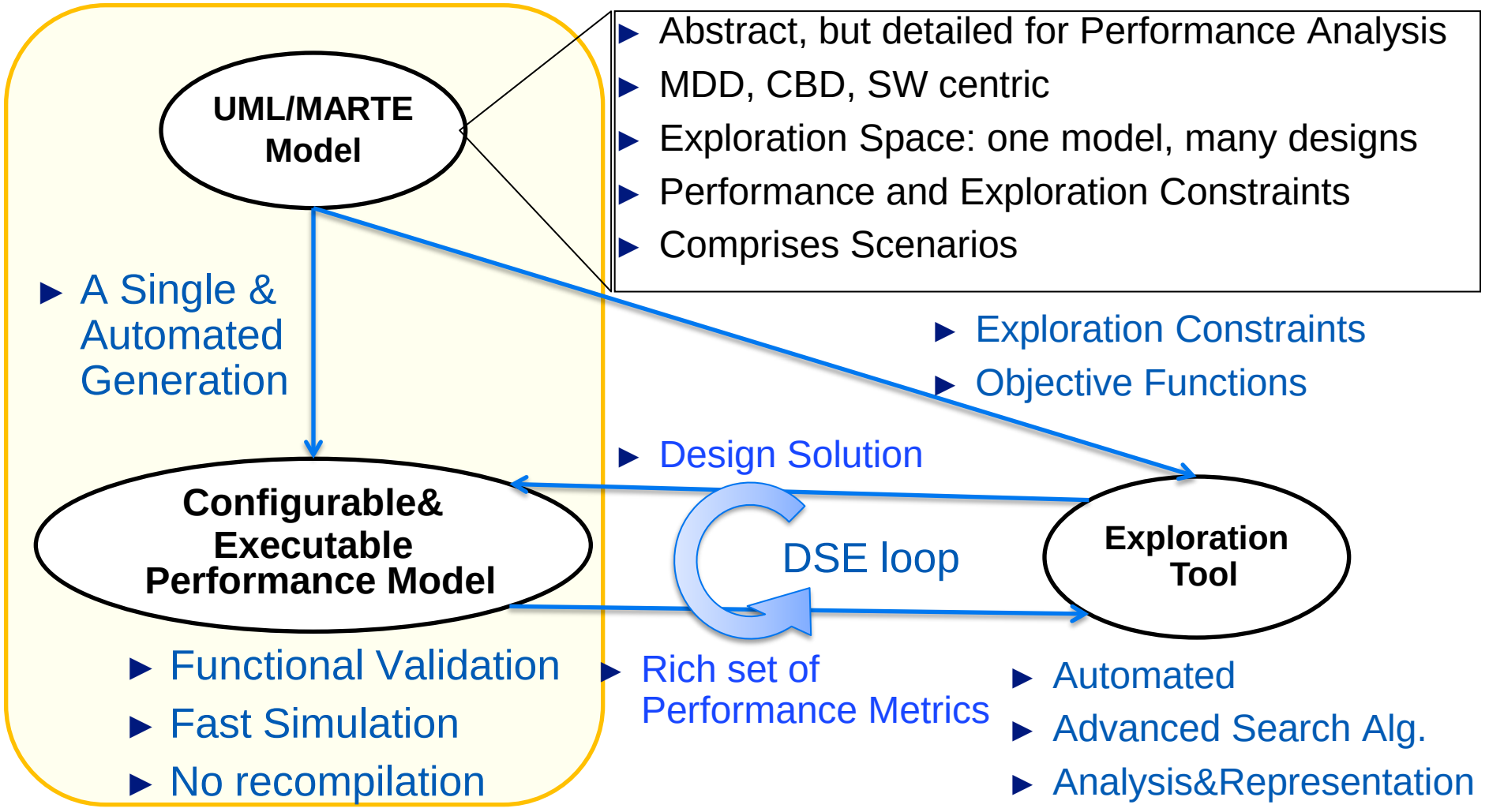
- **A competitive product**
  - Optimum Design
- **Short time to Market**
  - Successful Design ≠ Successful Product

- **Complex systems**
  - **Functionality..**
  - …and **Performance!**
- **Complex Design Problem**
  - Application, Platform, Mapping, …

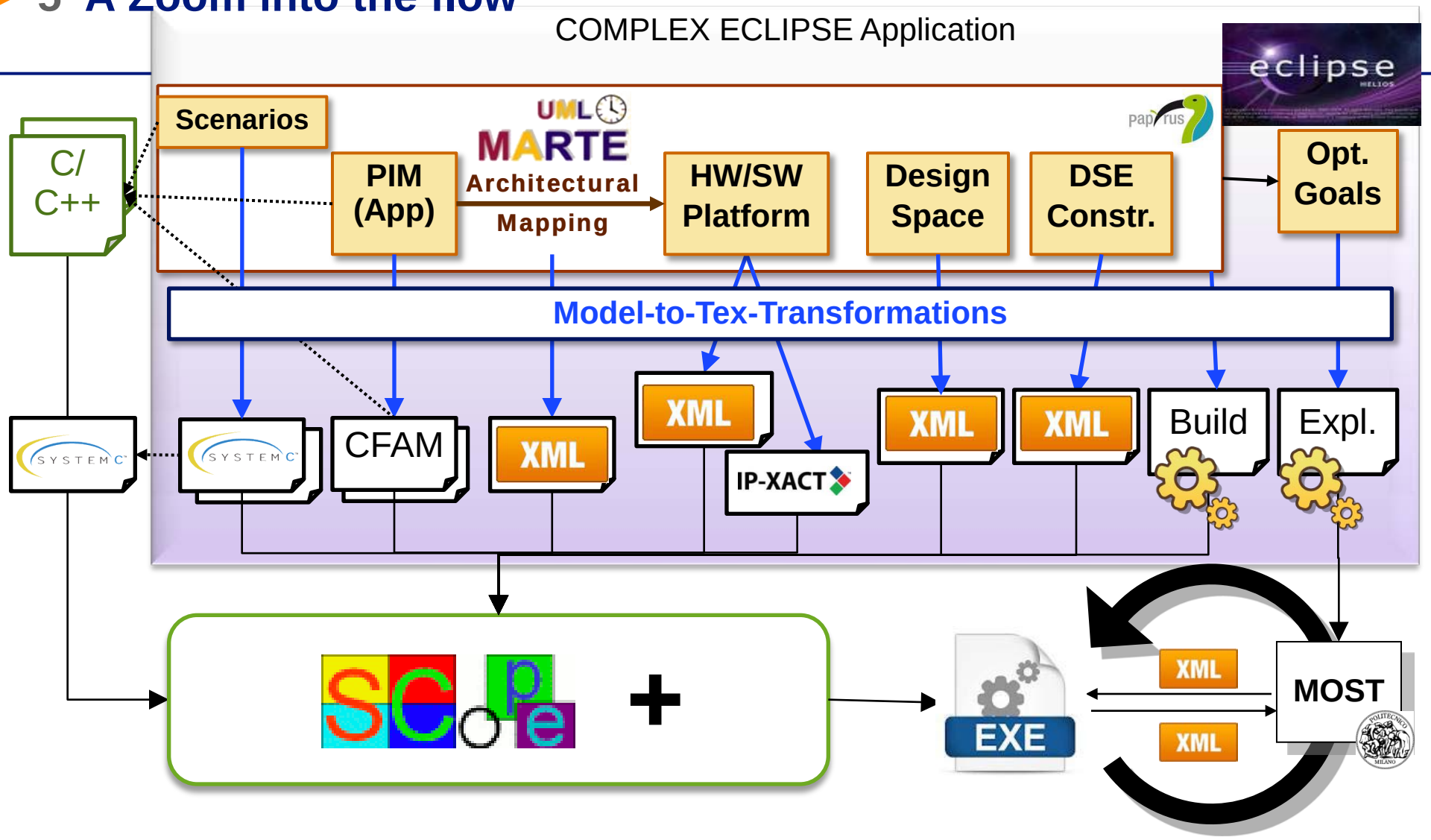
Classic slow DSE approach



4 Introduction: The UML/MARTE COMPLEX DSE Solution



5 A Zoom into the flow

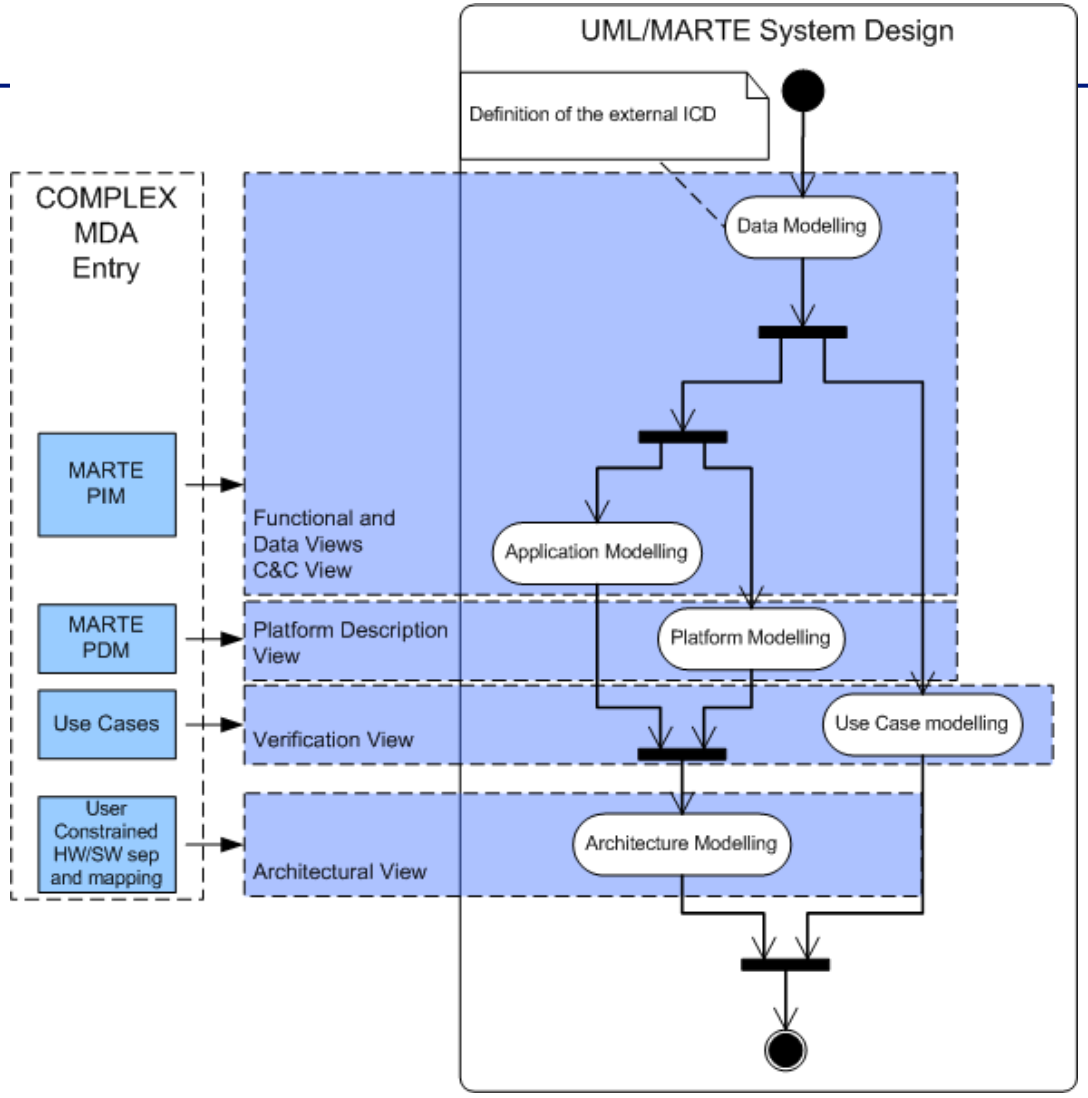


6 COMPLEX UML/MARTE Modeling Methodology: Main Features

- **MDD** concepts: **Separation of Concerns**
- **CBE:** Component-Based Engineering approach
- **SW centric**
- **DSE oriented**
- **UML-based**
  - **MARTE profile:** Capture most of the RTE required semantics
  - **COMPLEX profile**:
    - Defines DSE specific aspects not covered by MARTE (by any other profile)

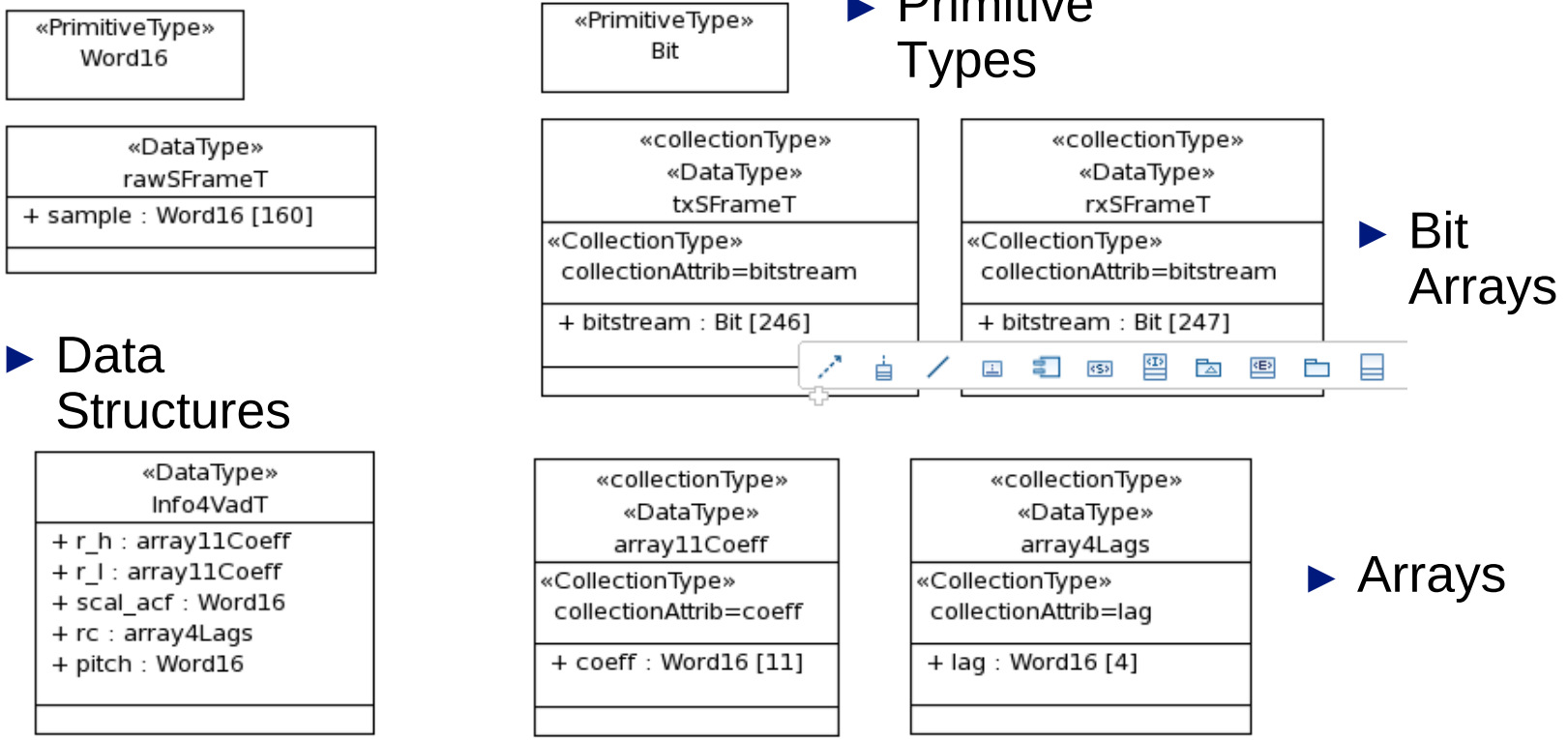
7 Modeling Methodology

- The modeling methodology states a well-defined flow
- Fulfill Industrial needs:
  - The flow exposes dependencies and independencies among modeling tasks (some views can be captured in parallel and by different specialist of the team)



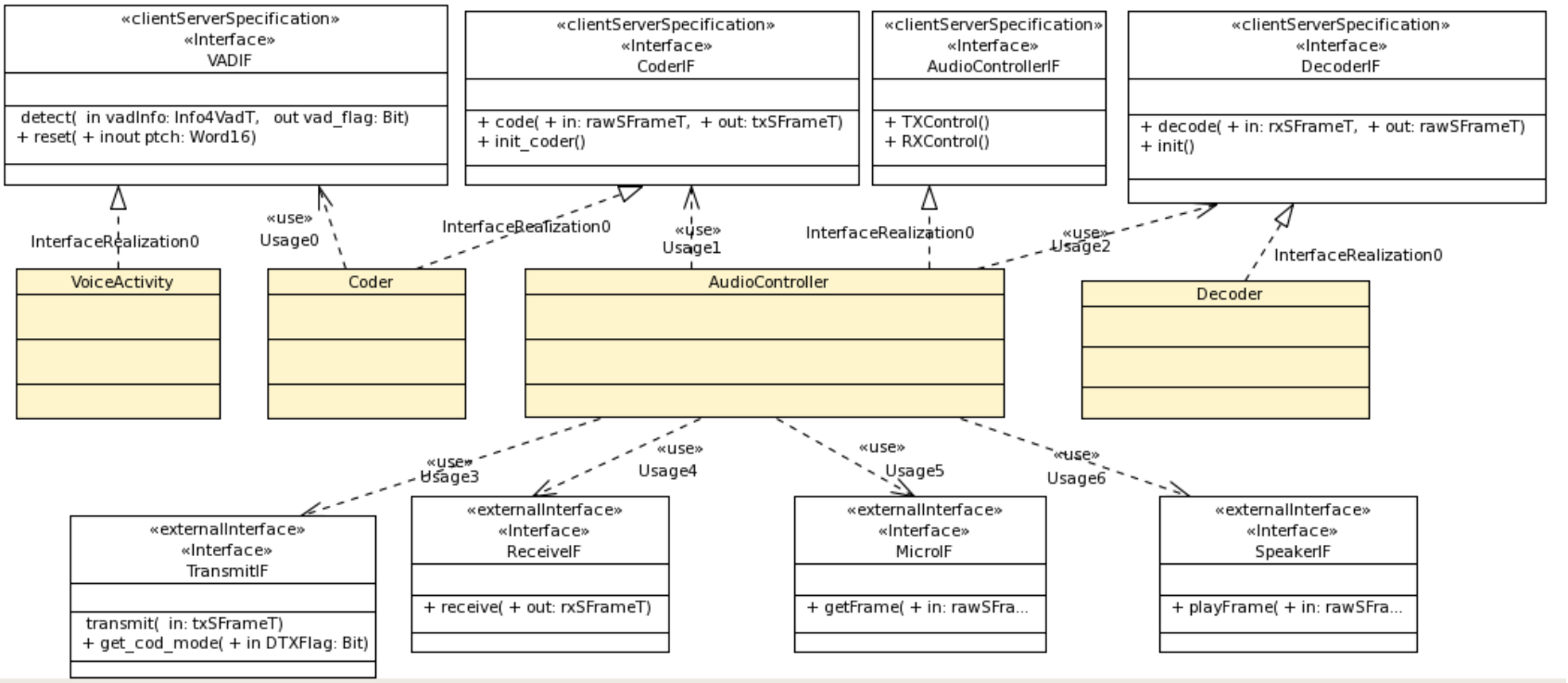
8 PIM Modeling I : Data View and Functional View

- **Data View:** Declare Data Types for Communication Interfaces



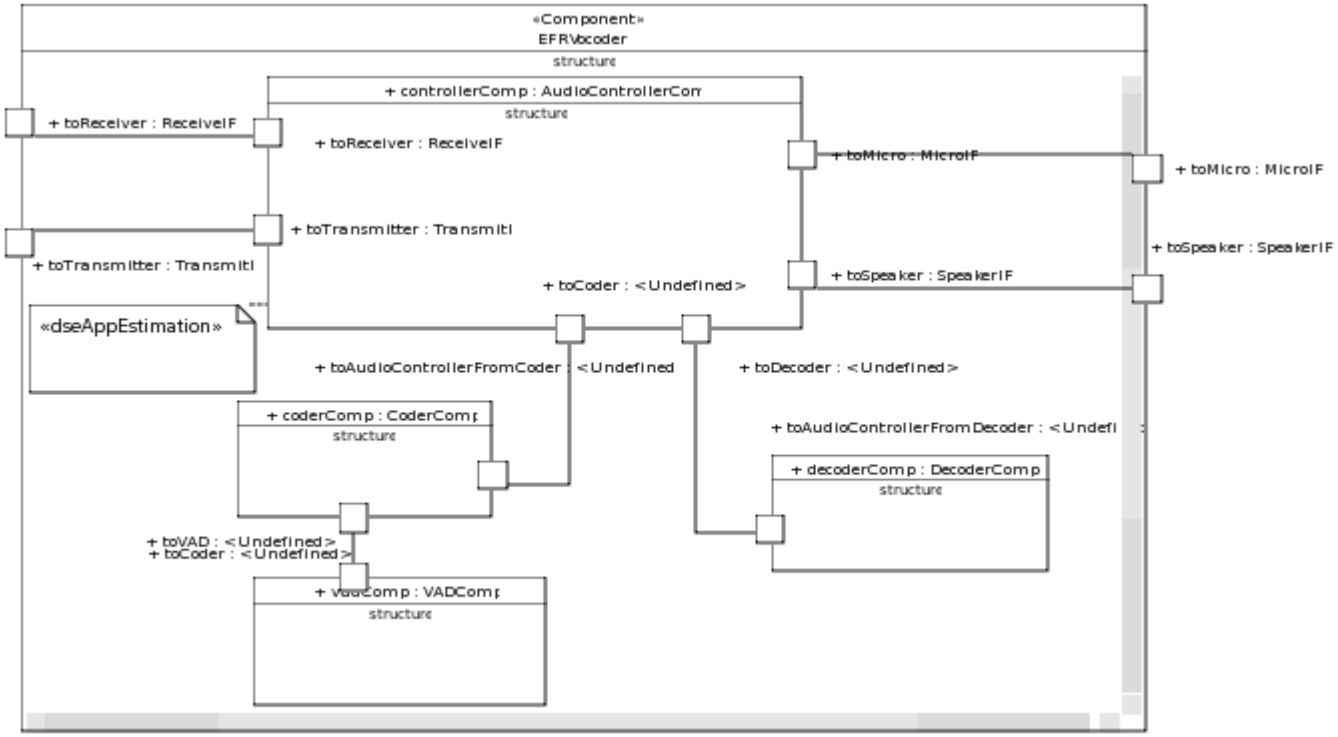
9 PIM Modeling I : Data View and Functional View

- **Functional View**
  - Declare **Component Interfaces** and **Functional Classes**
  - Classes **implement** Interfaces and **require** the services of other interfaces



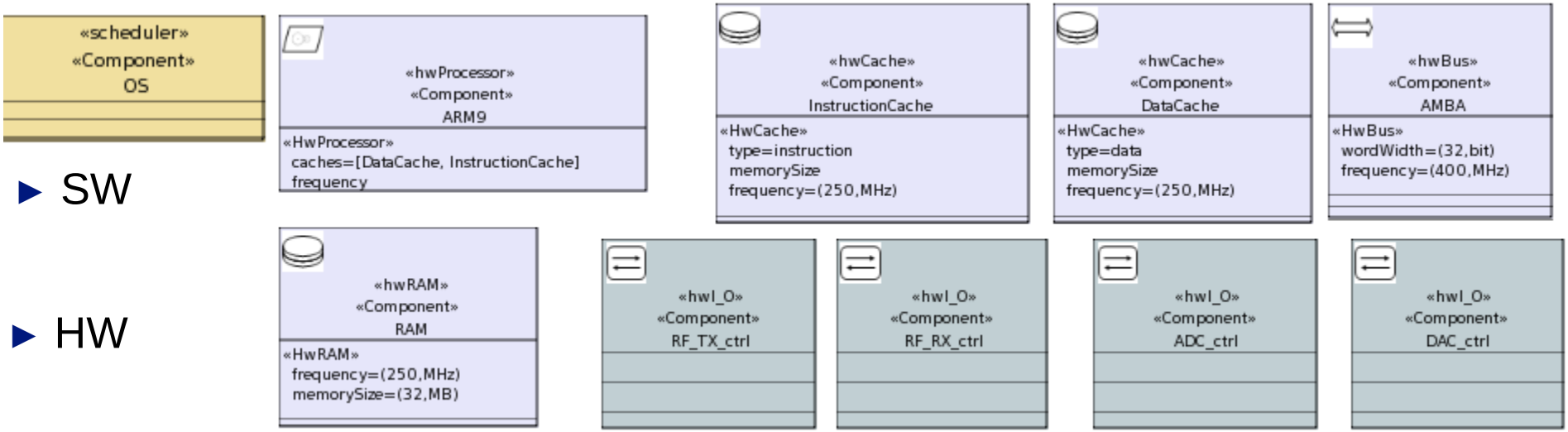
10 PIM Modeling III: Communication & Concurrency View

- Captures **Application Component Architecture** (Composite diagram)
  - Application components: provided and required operations (in SW centric)
  - Component instances declared in C&C view (UML properties) to build Application Architecture (PIM)



11 Platform View

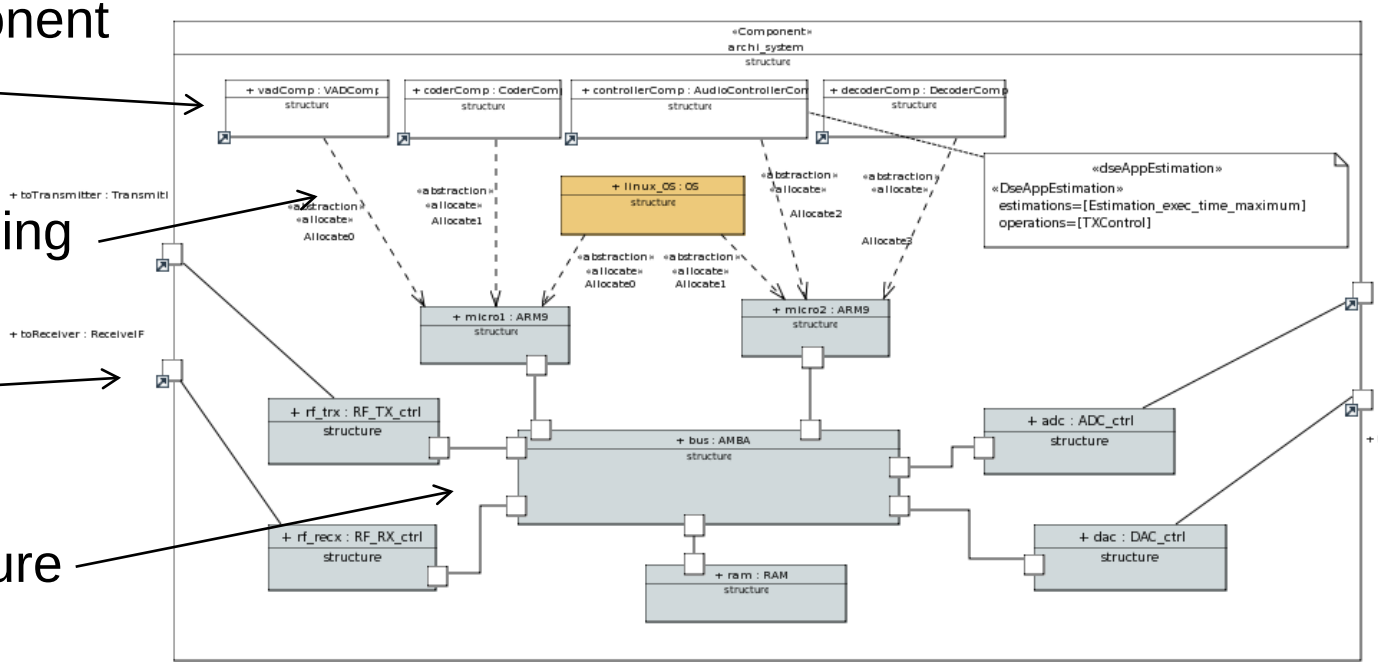
- Declares the main components of the platform
  - Software Components: OS, Drivers, …
  - Hardware Components: Processors, Memories, Buses, Custom HW, I/O
- Modelling entities: Components with MARTE stereotypes



- SW
- HW

12 Architectural View

- System Component (UML Component) representing the PSM
- Composite Diagram reflecting:
- Application Component instances
- Architectural Mapping
- System I/O
- Platform Architecture



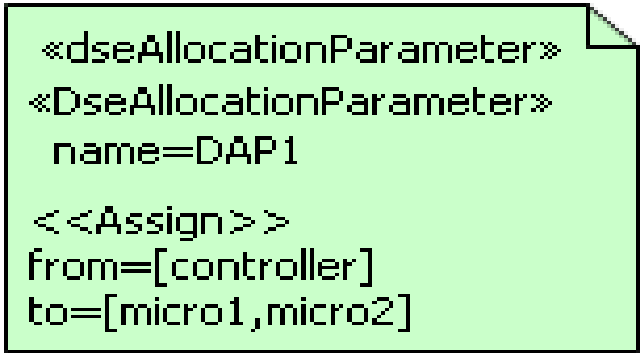
DSE Features

- Capturing the **Exploration Space** in a single model
- Defining a set of **Scenarios**
  - which allows the selection of the scenarios to be explored
- Capturing the **Output metrics**
  - which will be used as input for selecting the next experiment
  - finally determining the Pareto points

- The **Design Space is** composed of
  - A set of Architectural Mappings
  - A set of configurable attributes for Platform Components
  - A set of Platforms
  - A set of **DSE Constrains** and **rules**

Modeling the Design Space: Architectural Mapping Space

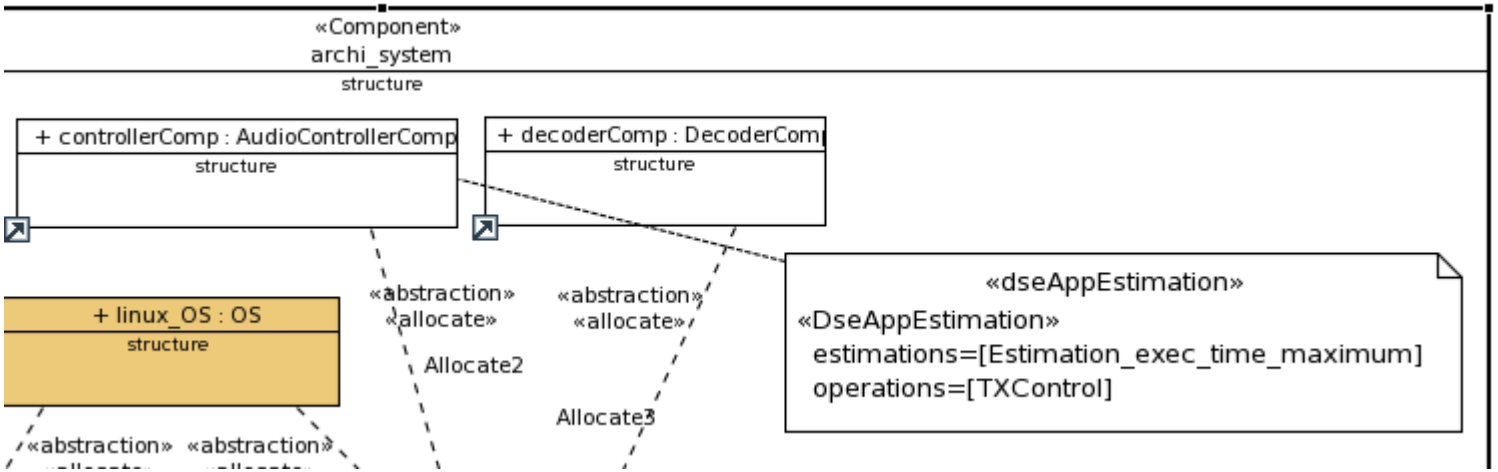
- Mapping of an application component to **several** platform components
  - UML Comment
  - MARTE <<Assign> stereotype
    - in the Architectural view



- Defining **exclusive** architectural mappings
- **Several** platform architectures in the platform view
  - COMPLEX <<dseAllocationParameter>>
    - Let assign a name to the allocation

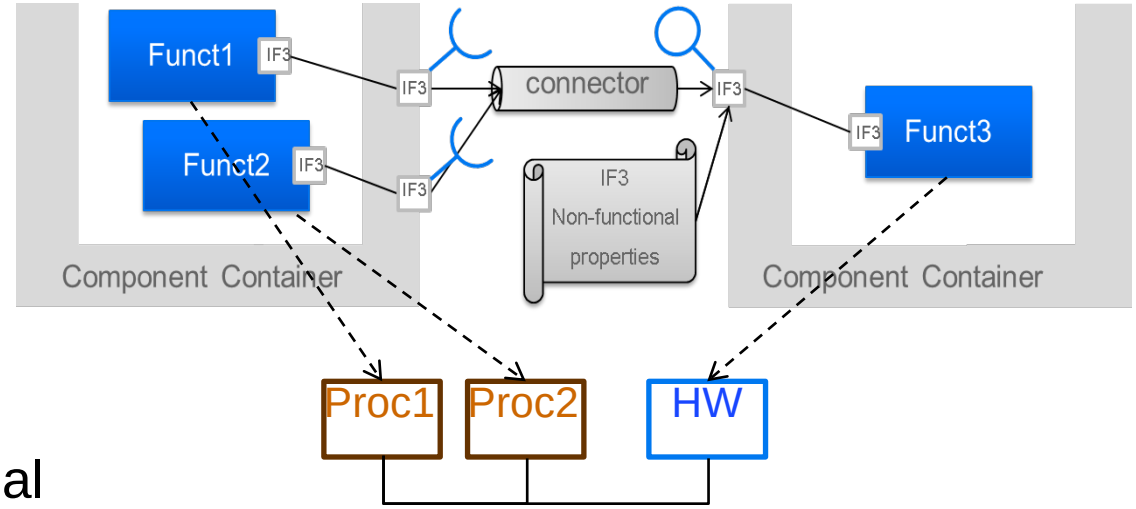
15 Output Metrics

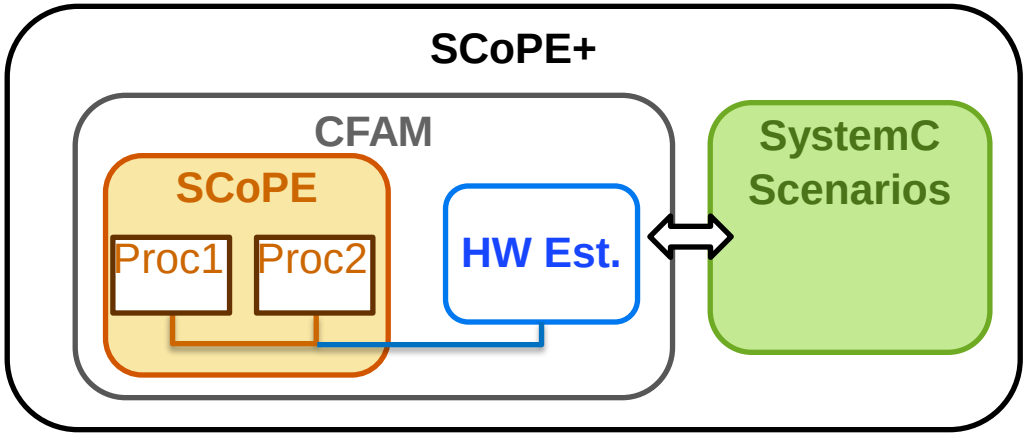
- Output metrics used by the goal functions in DSE
  - General application metrics on the platform architecture
- Definition of application-dependent metrics
  - COMPLEX <<dseAppEstimation>>
    - e.g, “get the maximum execution time for receiving, coding and sending a voice subframe”



SCoPE+ performance model

- SCOPE
  - Native simulation
  - SW estimation
  - Performance figures
- **SCoPE+:**
  - CFAM API
    - multiple computational models &
    - Architectural mappings
  - Integration of SW&HW estimations
  - Communication Impacts
  - Synchronization with a SystemC environment





Conclusions

- **COMPLEX UML/MARTE modeling methodology**
  - Support the development of models of COMPLEX systems for DSE
- **Automated generation of the executable and configurable performance model** relying on:
  - A text-based representation which improves extensibility
- **COMPLEX Eclipse Application integrates the high-level estimation and exploration tools**
  - Executable and configurable performance model
    - Avoids UML/MARTE model refactoring for the exploration
    - Enables an automated steering by the exploration tool