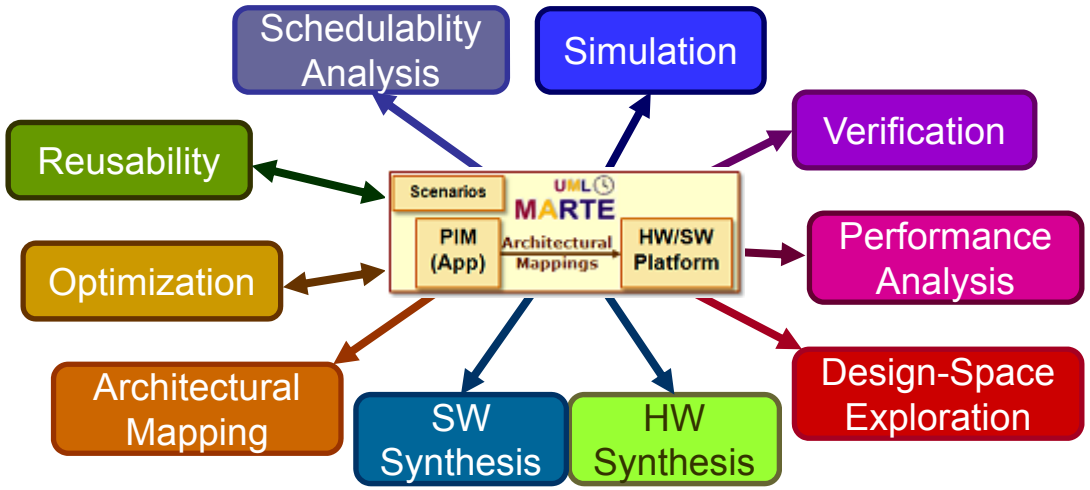
A Model-based, Single-Source approach to Design-Space Exploration and Synthesis of Mixed-Criticality Systems



Eugenio Villar
Fernando Herrera
Pablo Peñil

University of Cantabria
Spain

Agenda

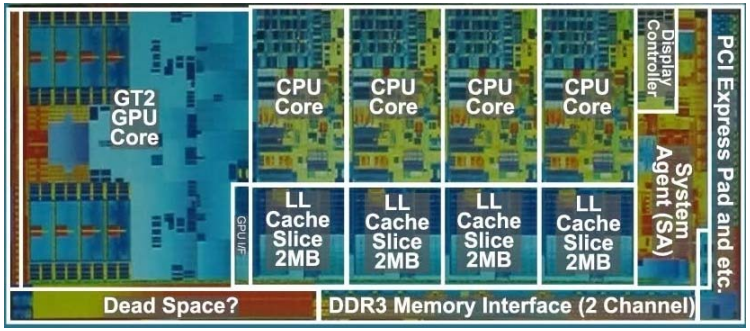
- Motivation & Introduction
- Contrex Modeling Methodology
  - SW Synthesis
  - Modeling of Mixed-Criticality Embedded Systems
  - Modeling for Design-Space Exploration
- Future work
- Conclusions

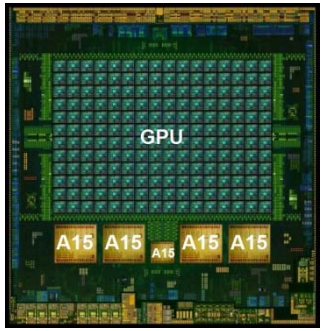
Agenda

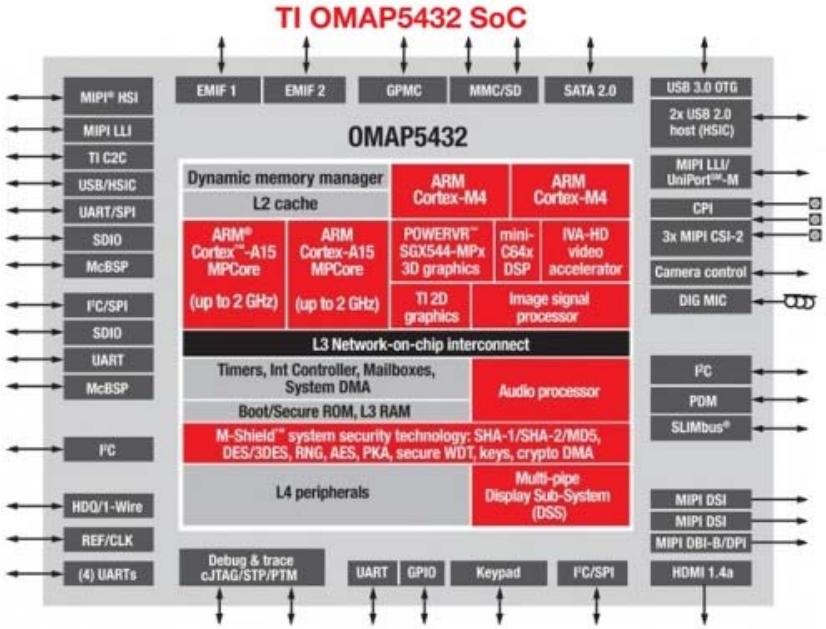
- Motivation & Introduction
- Contrex Modeling Methodology
  - SW Synthesis
  - Modeling of Mixed-Criticality Embedded Systems
  - Modeling for Design-Space Exploration
- Future work
- Conclusions

Motivation

- Design productivity gap
  - Raising the abstraction level
- Multi-Processing & Heterogeneous platforms



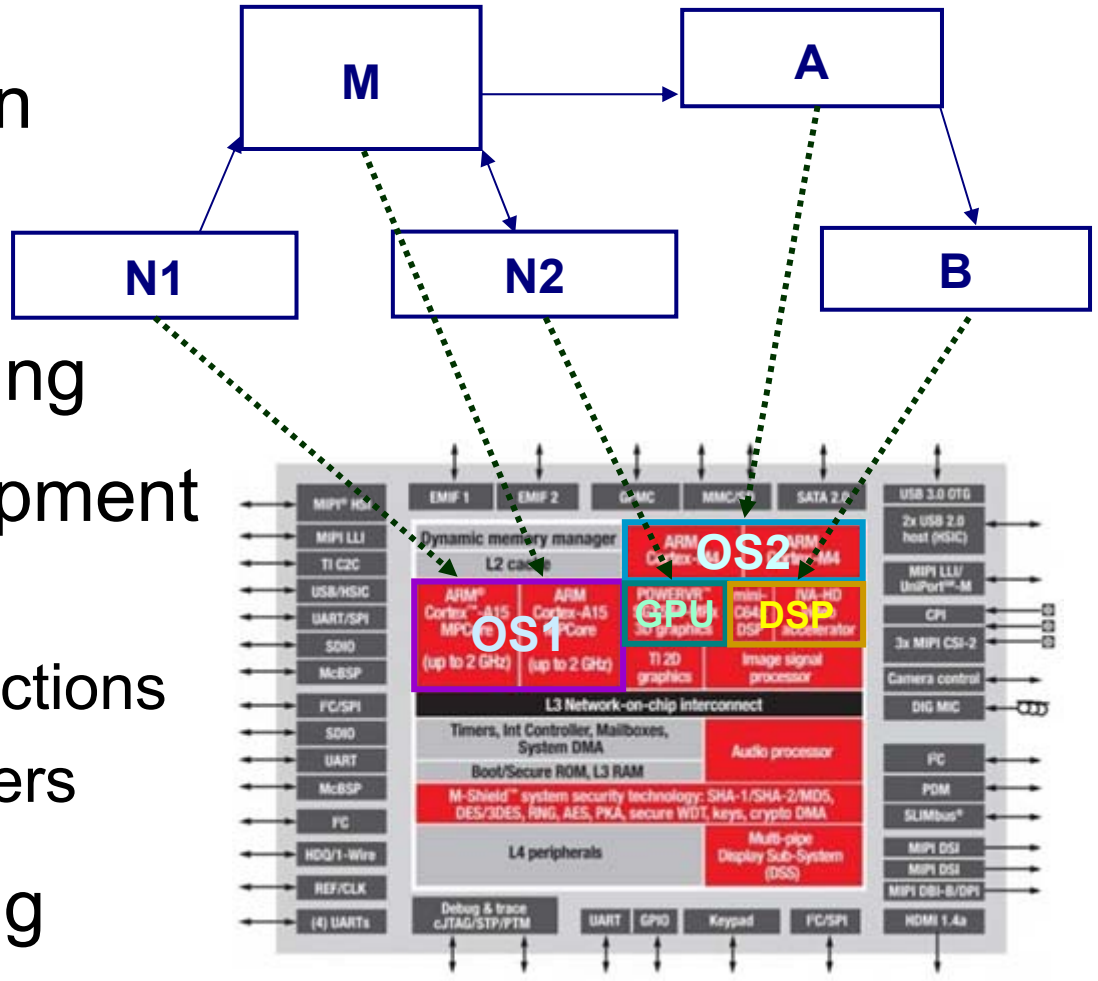




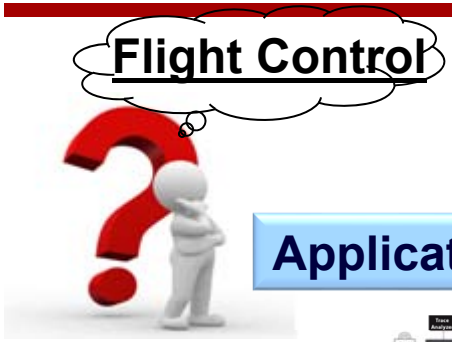
- Increasing SW content
  - SW-centric design methodologies

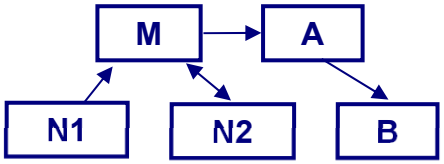
Usual SW development flow

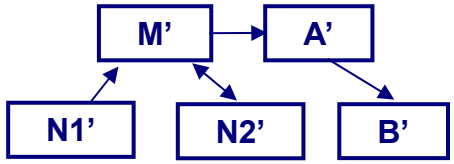
- Architectural Design
- HW/SW platform
- Architectural mapping
- Ad/Hoc SW development
  - System calls
  - Communication functions
  - I/O functions & drivers
- Verification & Debug
- Costly fixing of wrong design decisions



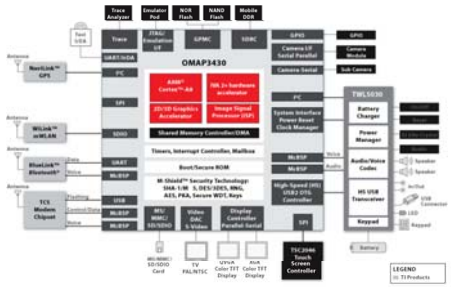
Reusability

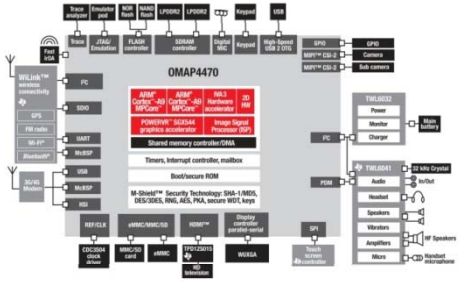


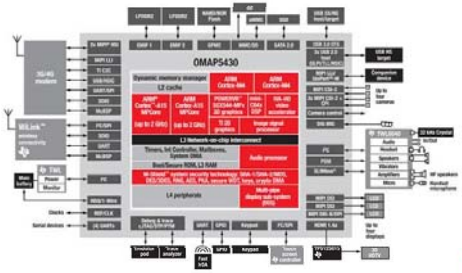




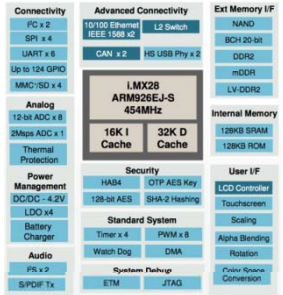
Application Evolution Time

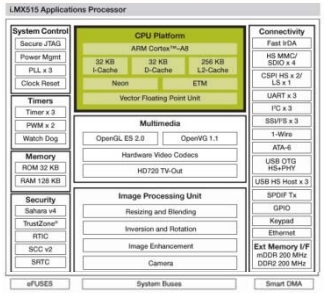


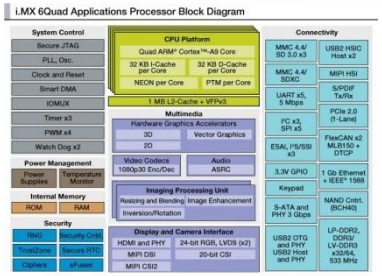




Texas Instruments OMAP Evolution Time







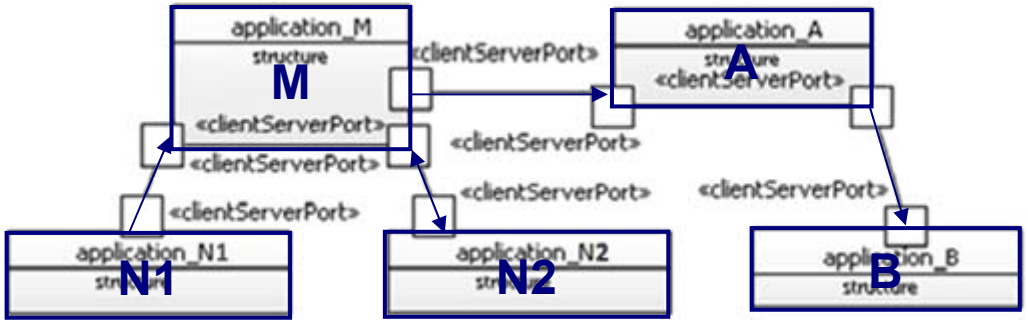
Freescale IMX Evolution Time

Introduction

- Model-Driven Design (MDD)
  - High-abstraction level
  - Mature SW engineering methodology
- UML language
  - Application to embedded systems design

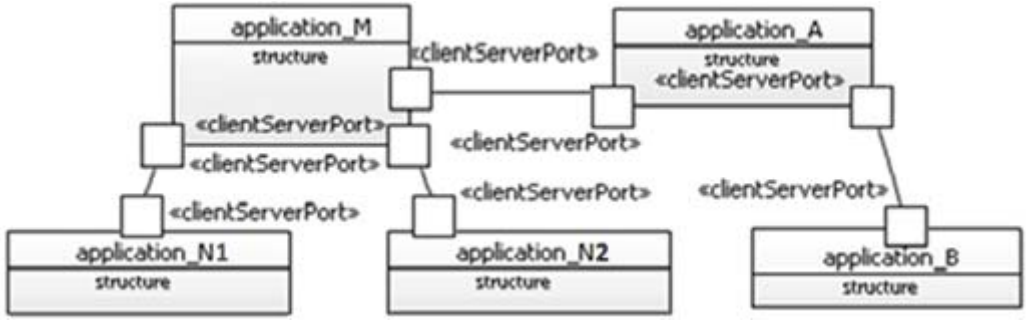
Introduction

- Why UML?
  - Natural way to capture system architecture
  - Standard way



Introduction

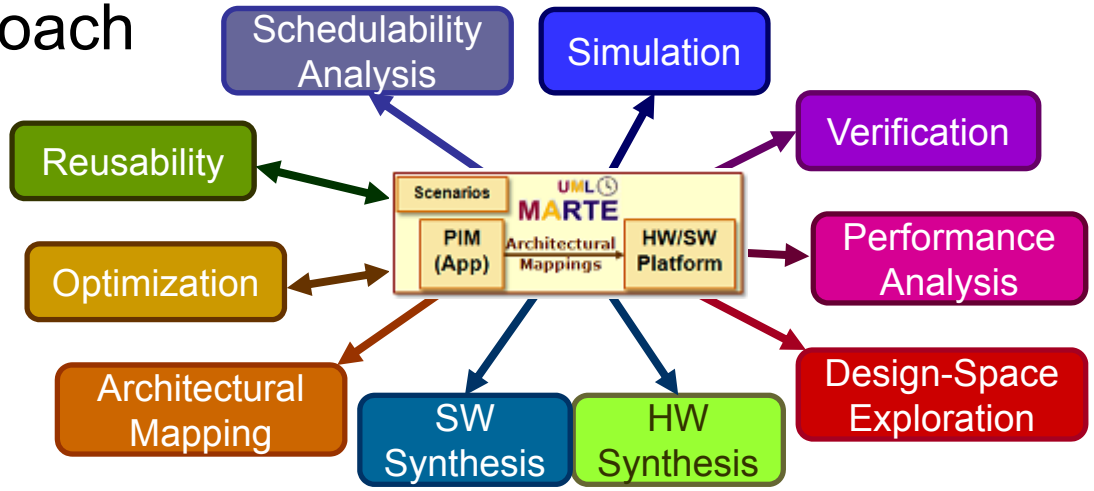
- Why UML?
  - Natural way to capture system architecture
  - Standard way



- UML language
  - Semantics lacks
    - What is each component?
    - What kind or interaction each link actually means?
  - Domain-specific profiles
    - UML/MARTE

Introduction

- MARTE
  - Standard UML profile for real-time embedded systems
    - Platform-Independent Model (PIM)
    - Platform Description Model (PDM)
    - Platform-Specific Model (PSM)
  - Rich semantics content
  - Single-source approach



Agenda

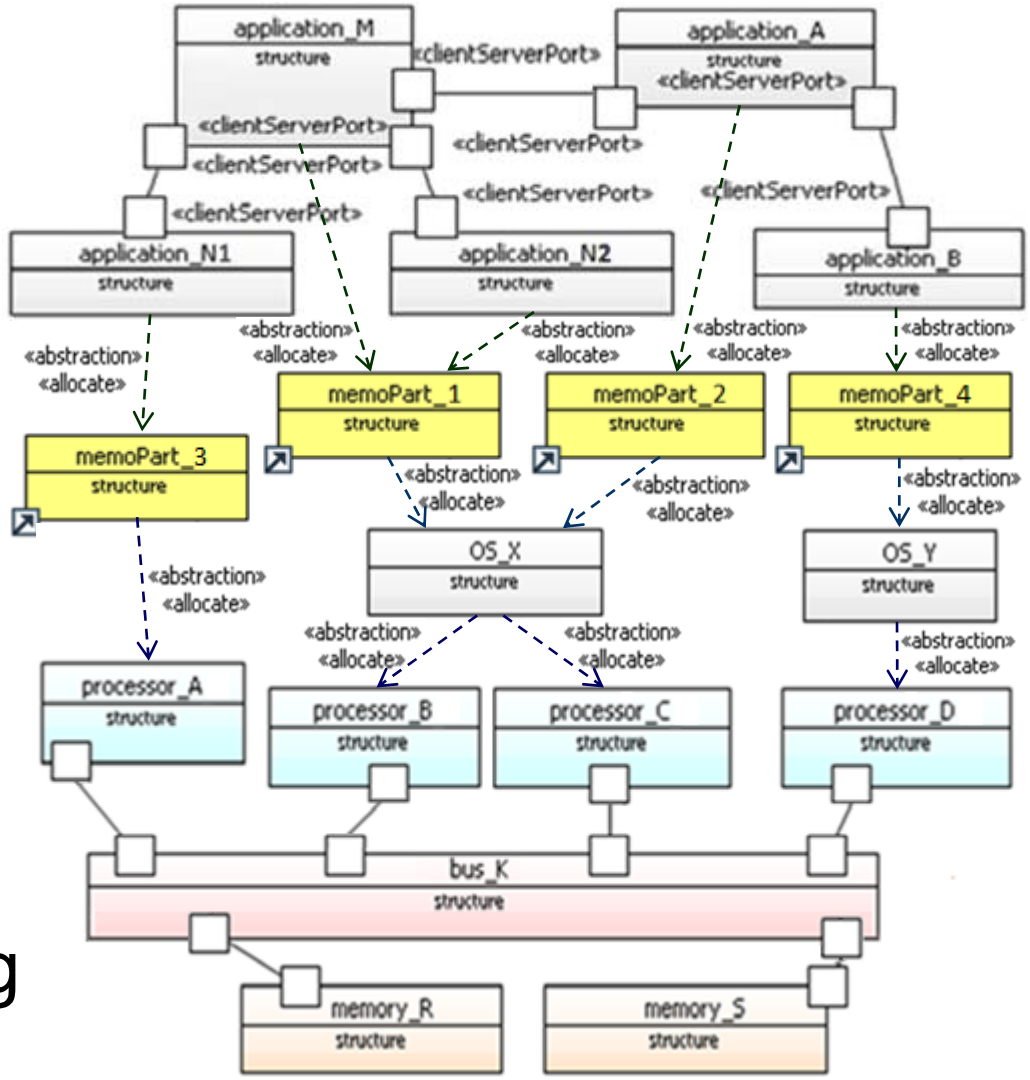
- Motivation & Introduction
- Contrex Modeling Methodology
  - SW Synthesis
  - Modeling of Mixed-Criticality Embedded Systems
  - Modeling for Design-Space Exploration
- Future work
- Conclusions

CONTREX Modeling Methodology

- Main features
  - MDD support
  - Component-Based Engineering approach
  - SW centric
  - Standard
    - MARTE profile
  - SW synthesis
  - Supporting Mixed-Criticality Modeling
  - Supporting Design-Space Exploration

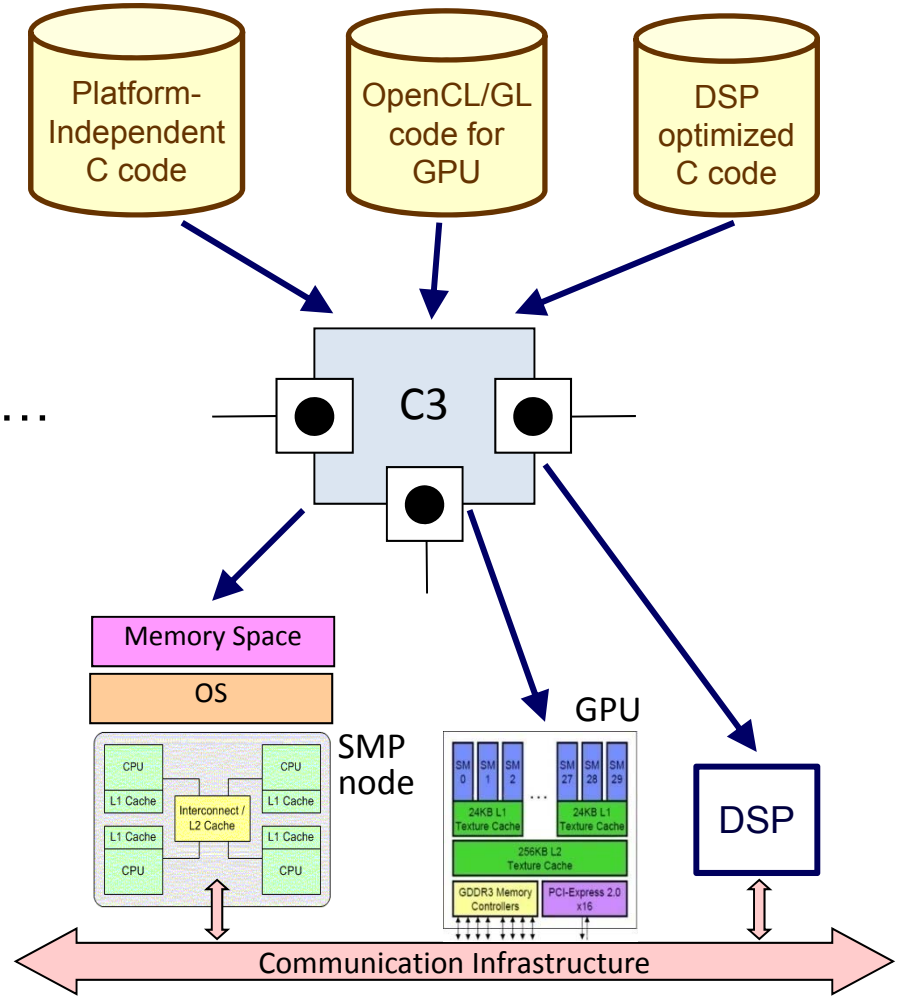
CONTREX Modeling Methodology

- Architectural Design
- Code reuse and/or development
  - platform independent
- HW/SW platform
- Architectural mapping



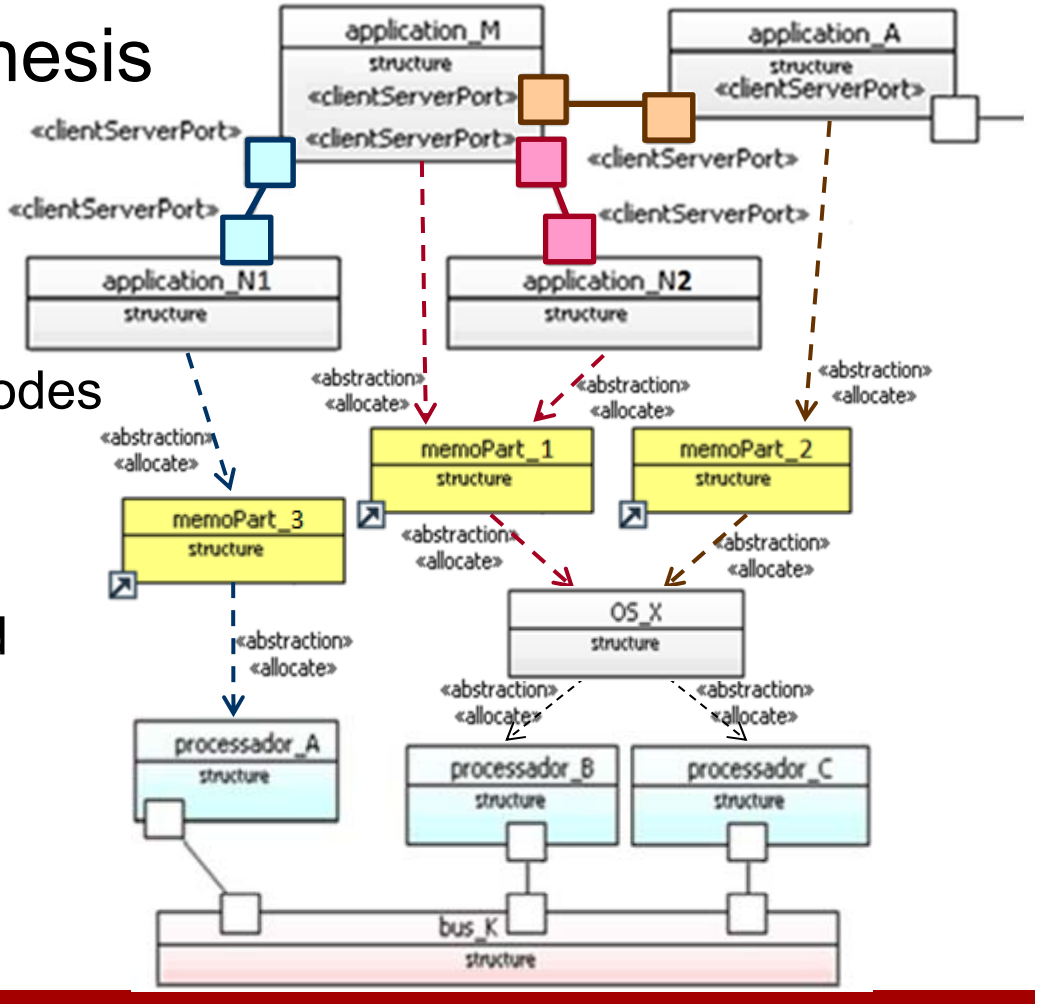
SW Synthesis

- Functional synthesis
  - Platform-Specific code
    - Optimized C code for DSPs
    - OpenCL/GL for GPUs
    - C/C++ & OpenMP for SMPs…



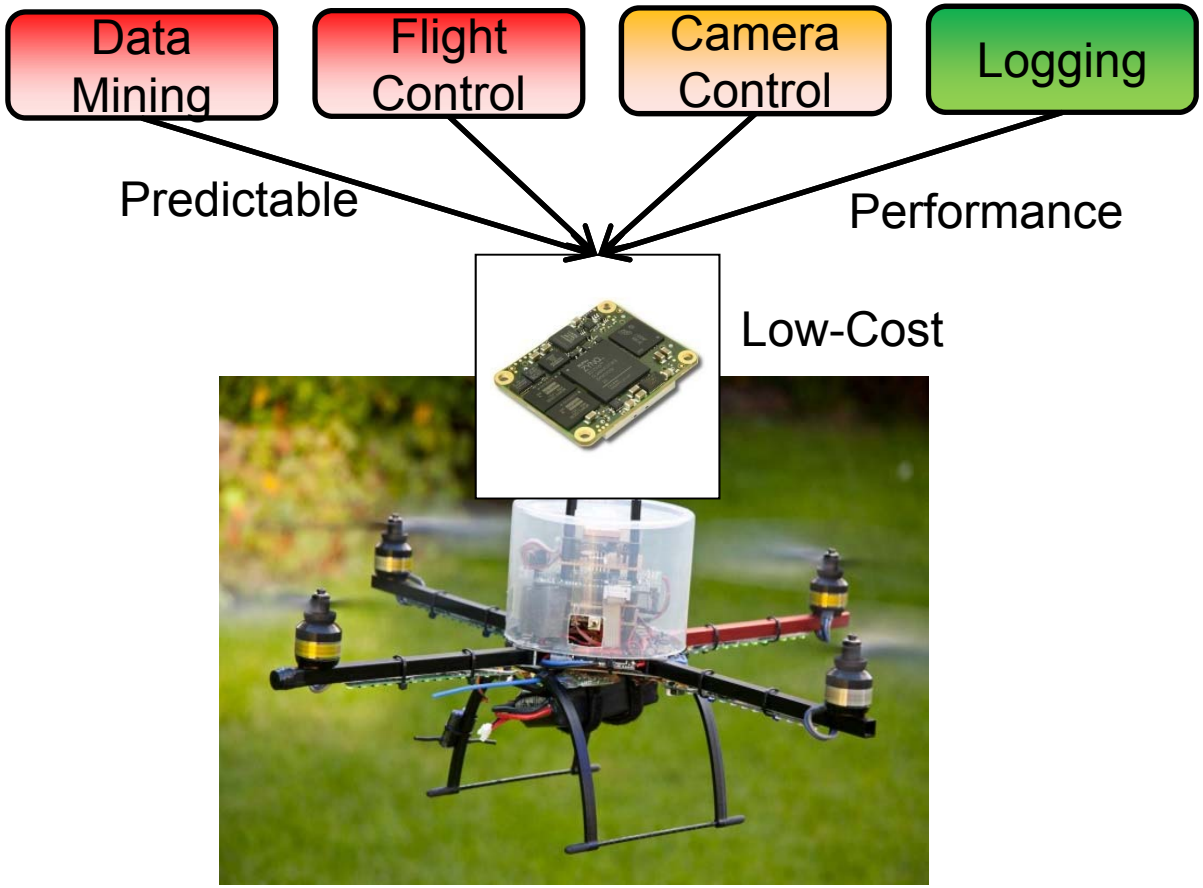
SW Synthesis

- Communication synthesis
  - Architectural mapping
    - Same memory space
    - Same OS
    - Different processing nodes
  - Benefits / Drawbacks
    - Communication Speed
    - Memory protection
    - Memory/cache use
    - Scheduling
    - Parallelism…



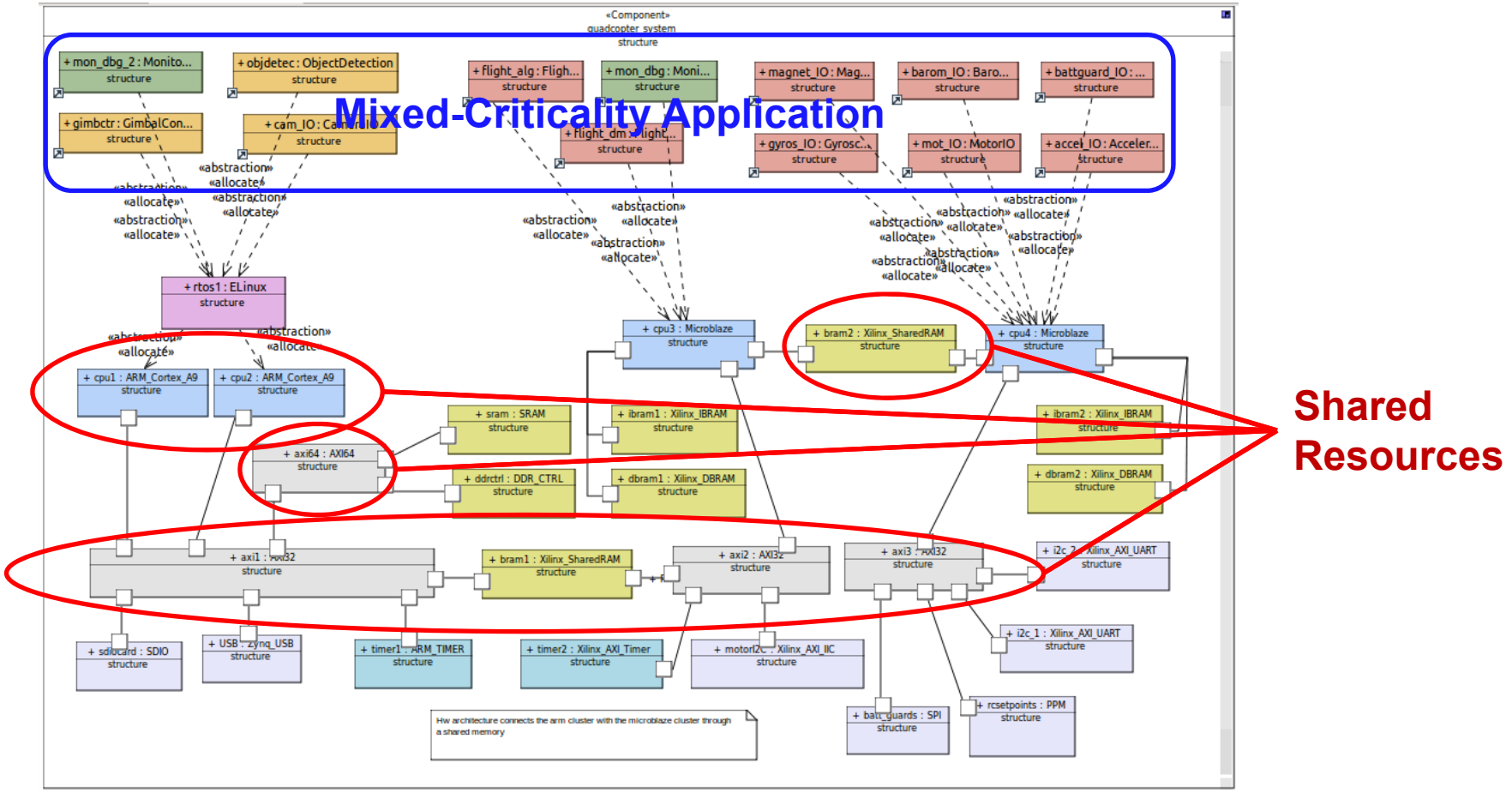
CONTREX Modeling Methodology

- Mixed-Criticality approach



CONTREX Modeling Methodology

- Mixed-Criticality

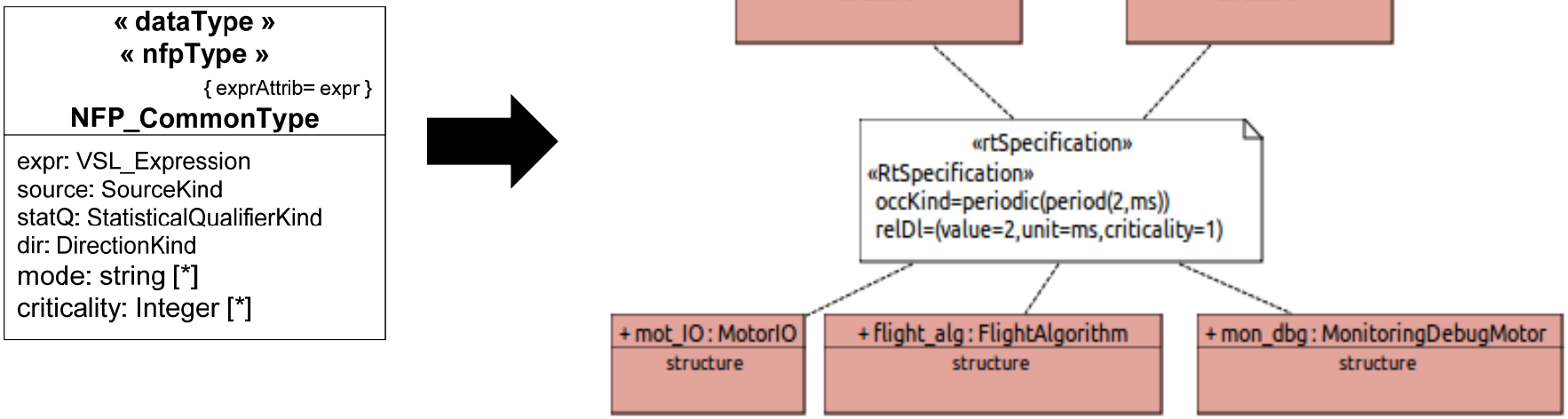


CONTREX Modeling Methodology

- Mixed-Criticality
- Criticality
  - Integer Level of importance
    - Functional & Extra-Functional Requirements
  - Implications on analysis and development
- In-lined with usual definitions
  - Level of assurance against failure [Burns&Davis, 2015]
  - Safety Standards
    - IEC/EN 61508 (SIL)
    - DO-178B
    - ISO 26262 (ASIL)

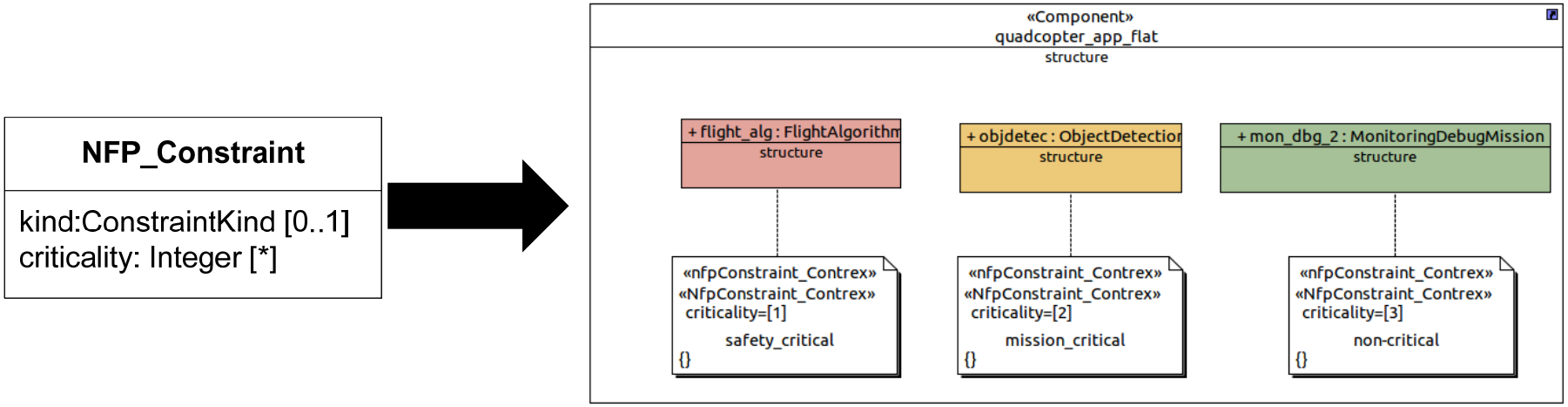
CONTREX Modeling Methodology

- Criticality of Value Annotations
  - Synthetic description of criticalities
  - MC-aware schedulability analysis
    - WCET = F(Criticality)
    - Probabilistic WCET analysis techniques



CONTREX Modeling Methodology

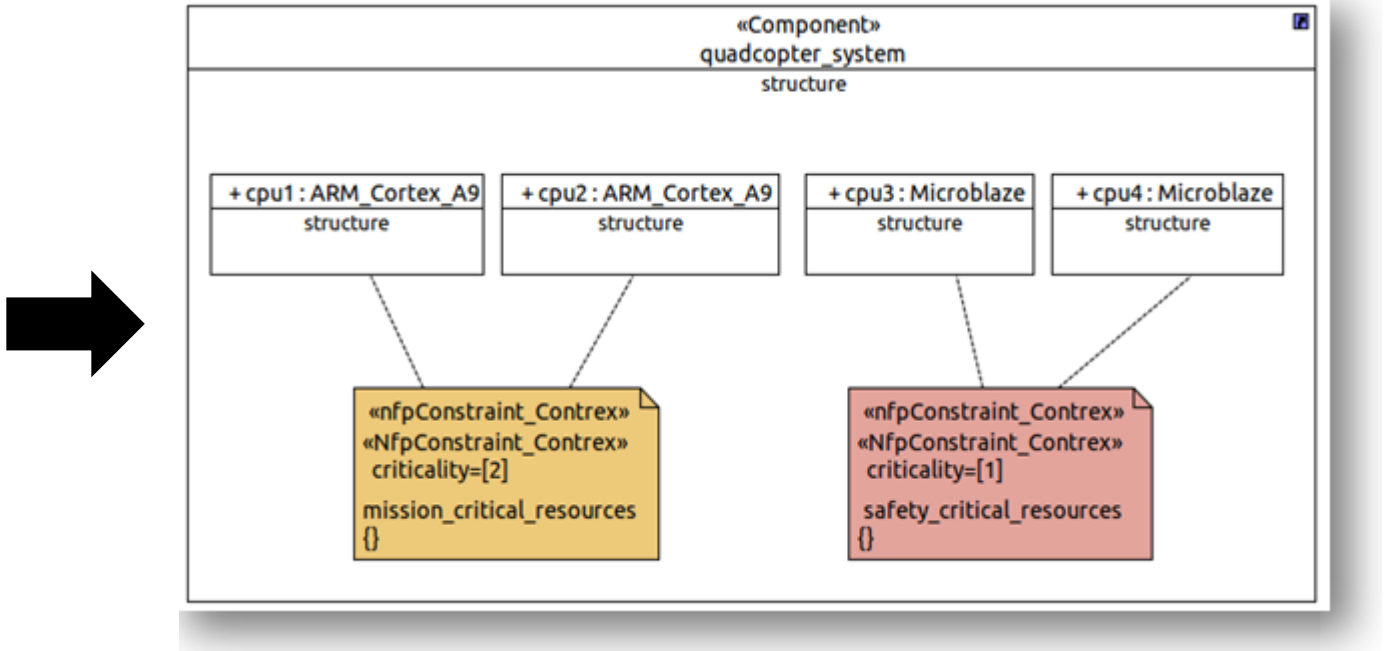
- Criticality of Application Components
  - For imposing conditions on the software development
  - Associate criticality to all the related constraints and sub-components



CONTREX Modeling Methodology

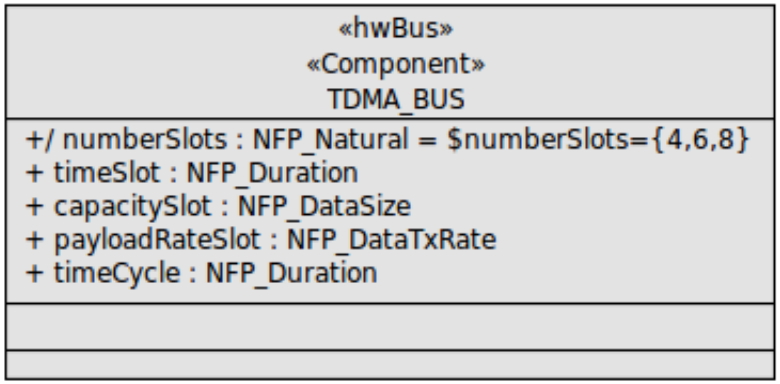
- Criticality of Platform Components
  - HW constraints derived from the criticality level
    - Imposing conditions on the hardware development
    - Coherence of application to platform component mapping

NFP_Constraint
kind:ConstraintKind [0..1] criticality: Integer [*]

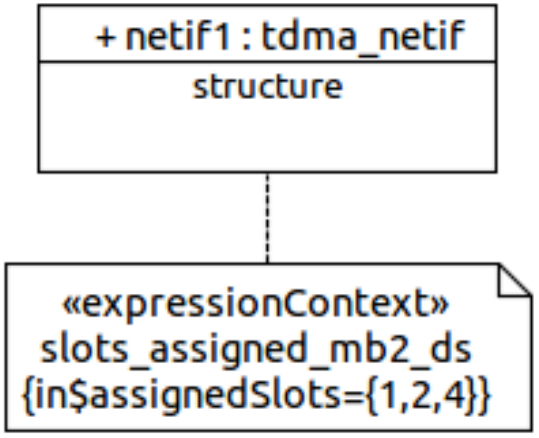


CONTREX Modeling Methodology

- Design Space Exploration
  - A single model for describing the Design Space
  - DSE parameters: declared as VSL expressions



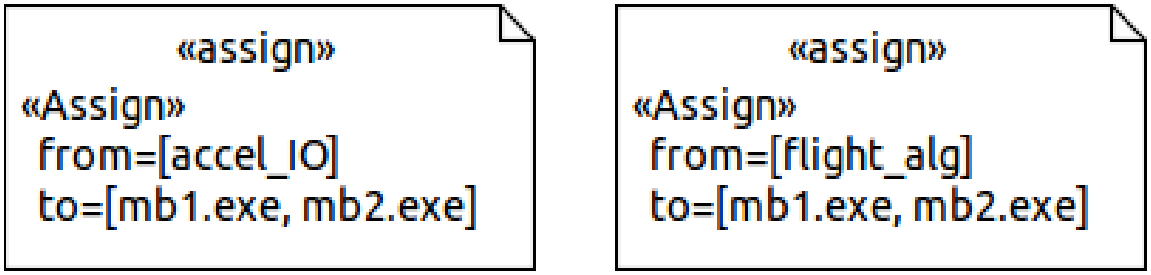
within an attribute of
a component declaration



Through a constraint associated
to a component instance

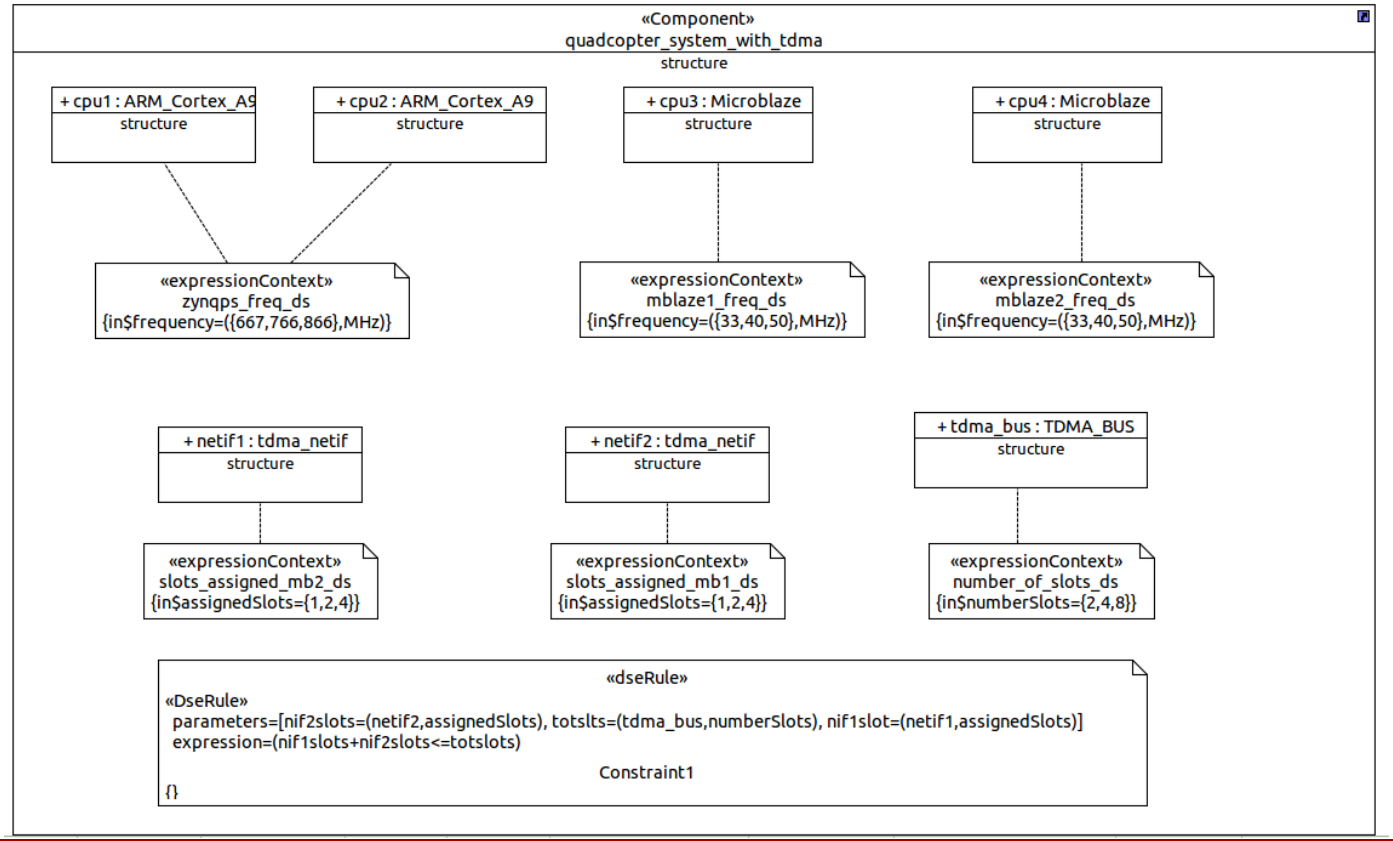
CONTREX Modeling Methodology

- Design Space Exploration
  - Mapping Exploration



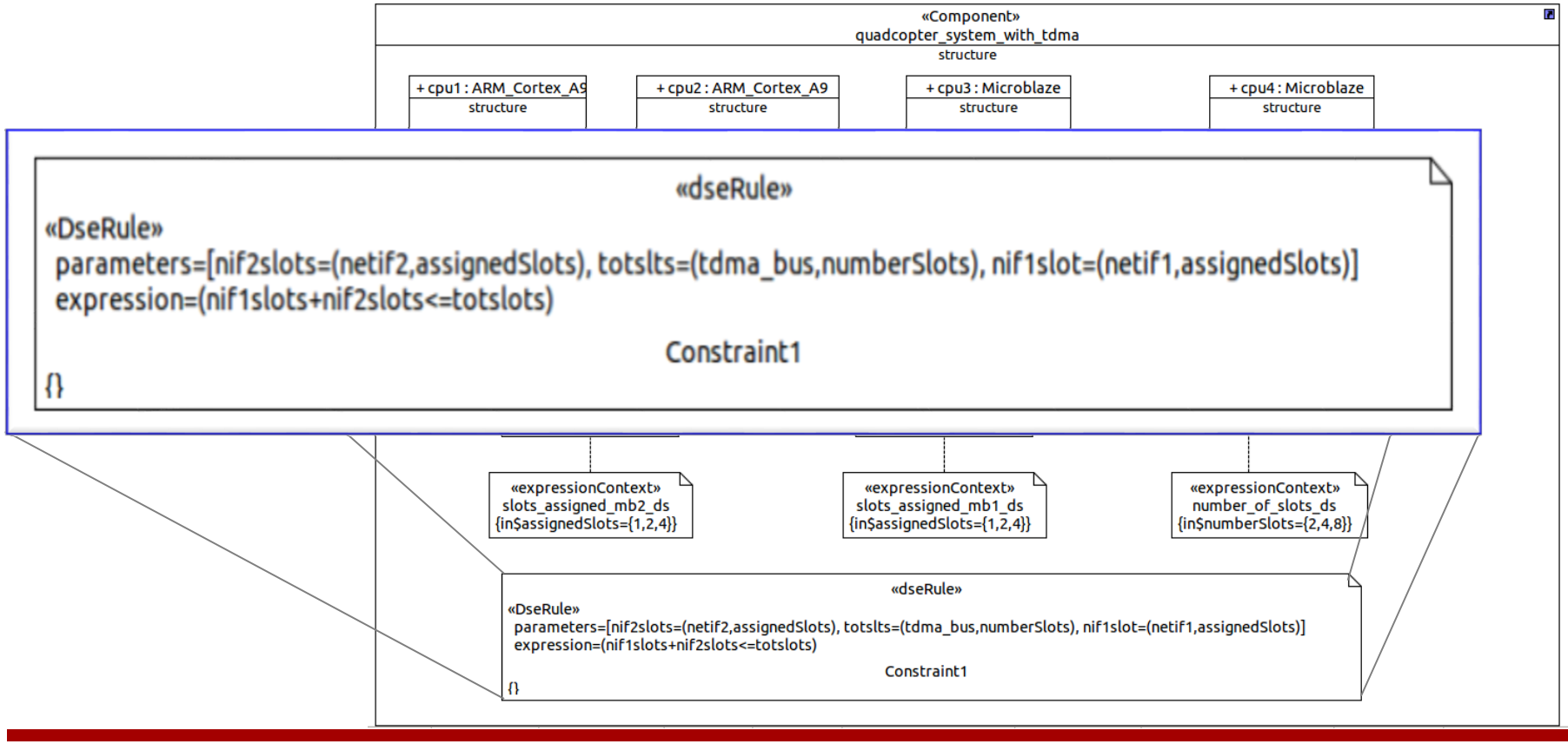
CONTREX Modeling Methodology

- Design Space
  - a N-dimensional cube ($3^6 = 729$)



CONTREX Modeling Methodology

- DSE rules
  - Constrain the N-dimensional cube

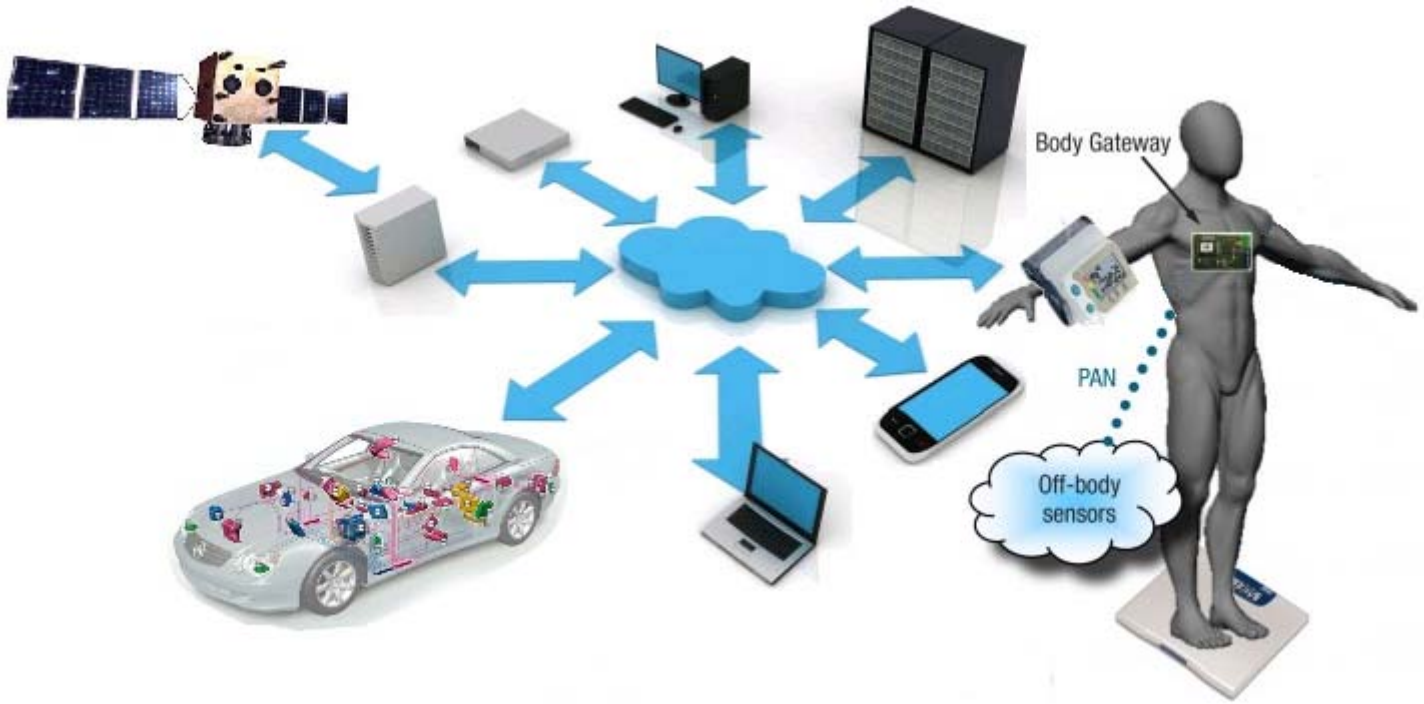


Agenda

- Motivation & Introduction
- Contrex Modeling Methodology
  - SW Synthesis
  - Modeling of Mixed-Criticality Embedded Systems
  - Modeling for Design-Space Exploration
- Future work
- Conclusions

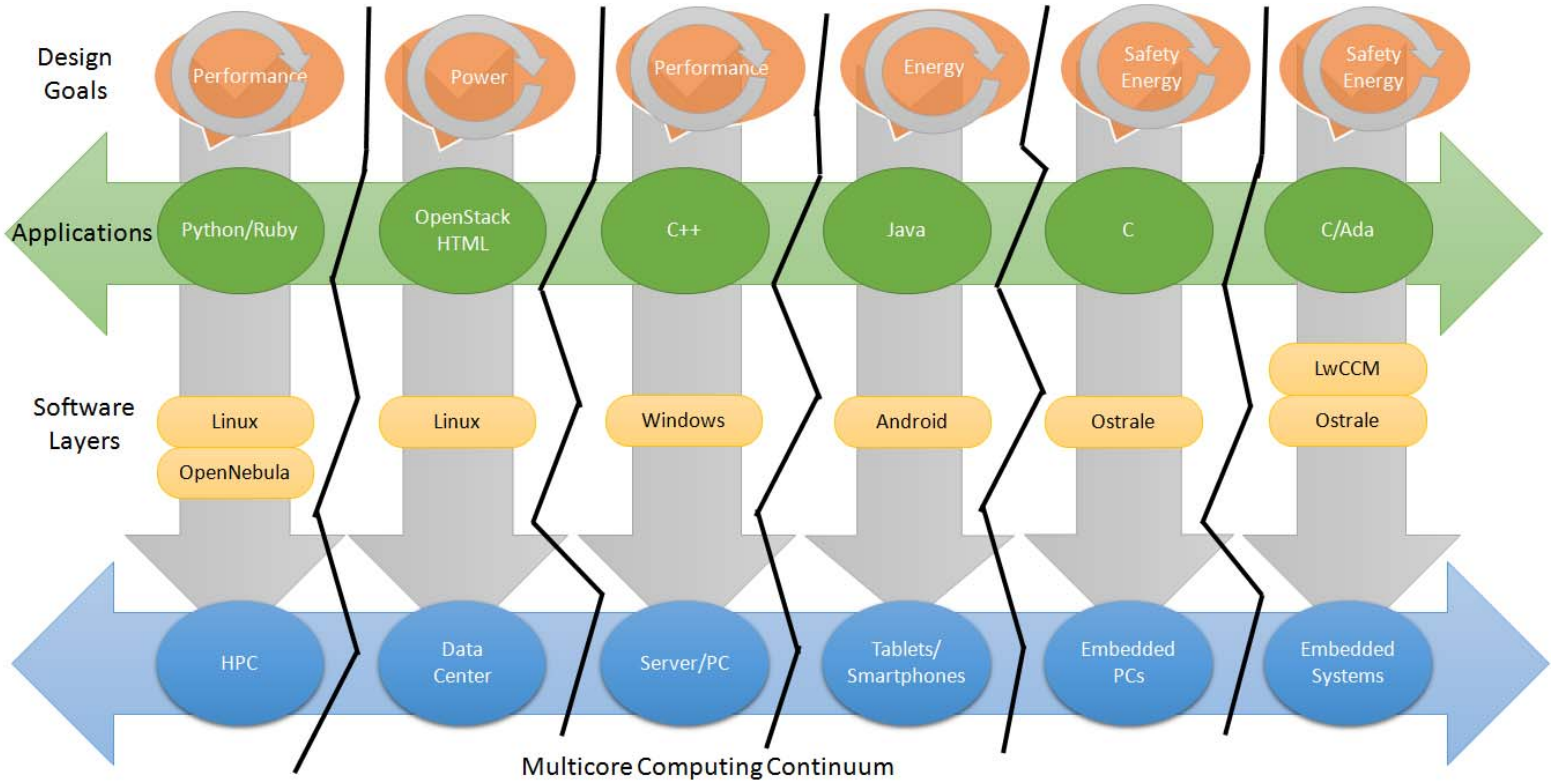
Future Work

- Programming the Computing Continuum
  - Spanning computing platforms of many kind



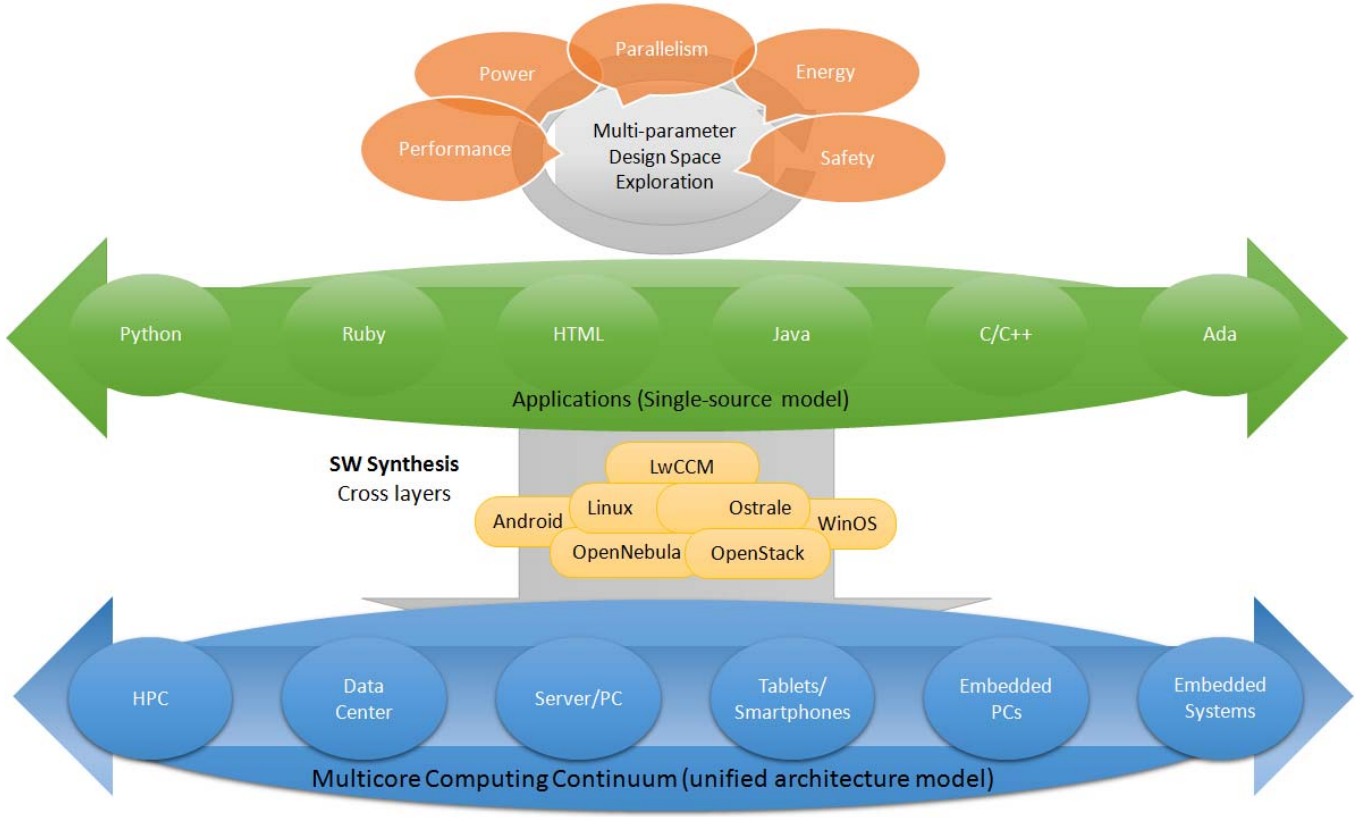
Future Work

- MDE as a powerful approach
  - …but based on Domain-Specific Languages & Tools



Future Work

- MDE as a holistic system engineering approach
  - Commonalities across domains



Agenda

- Motivation & Introduction
- Contrex Modeling Methodology
  - SW Synthesis
  - Modeling of Mixed-Criticality Embedded Systems
  - Modeling for Design-Space Exploration
- Single-source design & programming framework
- Future work
- Conclusions

Conclusions

- Contrex UML/MARTE Modeling Methodology
  - Powerful Single-Source approach
  - Reusability
  - Component-Based Engineering approach
  - SW centric
  - DSE-oriented
  - Supporting Mixed-Criticality Design
  - SW synthesis
  - Extensible to distributed applications