Model-Based Development of Integrated Computer Systems – Modeling the Execution Platform

Bernhard Huber and Roman Obermaisser

WISES 2007, Madrid, Spain

22-06-2007

Overview

- Problem Definition / Objectives
- DECOS Integrated Architecture
- Model-Based Design in DECOS
- Execution Platform Modeling
- Platform Model Editor
- Exemplary Application

Problem Definition / Objectives

Context:

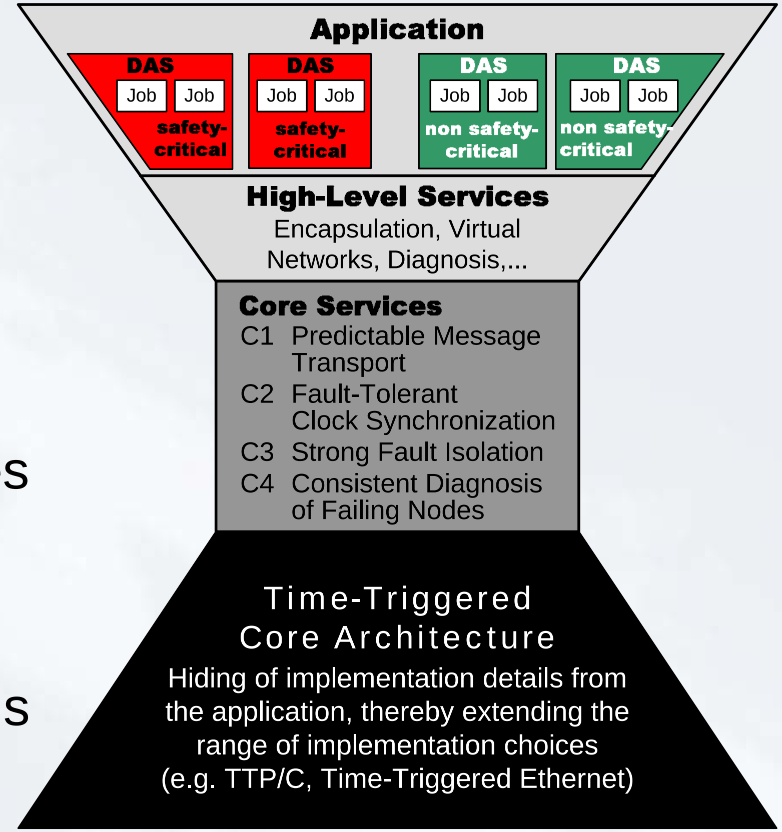
- Shorter development cycles of embedded systems despite increasing functionality and complexity
- Shift from federated to integrated systems in order to improve quality of control and improve resource utilization
- Model-based design methodologies separate application logic from underlying platform technology
  - facilitates reuse despite changing technologies

Objective:

- Facilitate a model-based design methodology for integrated architectures by providing methods, models and tools for capturing information on the available resources of the platform

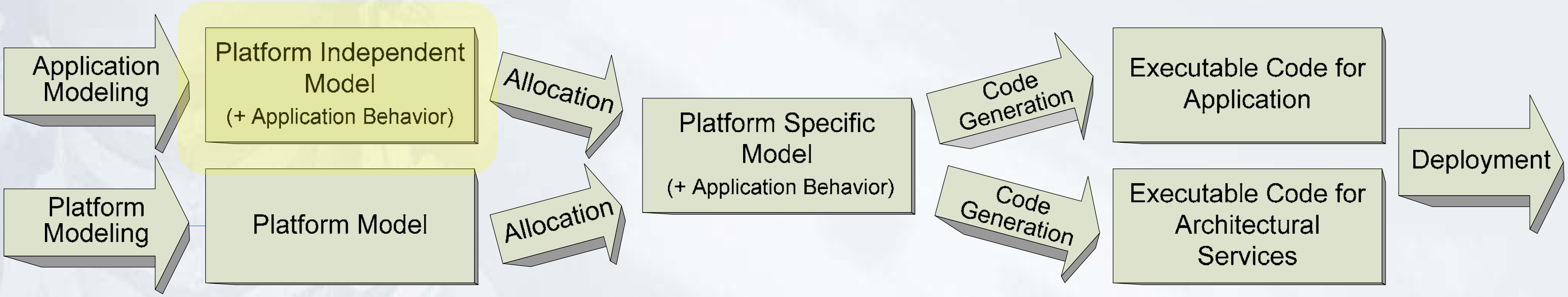
DECOS Integrated Architecture

- **D**ependable **E**mbedded **Co**mponents and **S**ystems (IP within 6th EU FP)
- Architecture for distributed embedded real-time systems mainly aimed at automotive and avionics domain
- An integrated architecture that combines the benefits of integrated and federated architectures
- Generic architectural services as a basis for application development



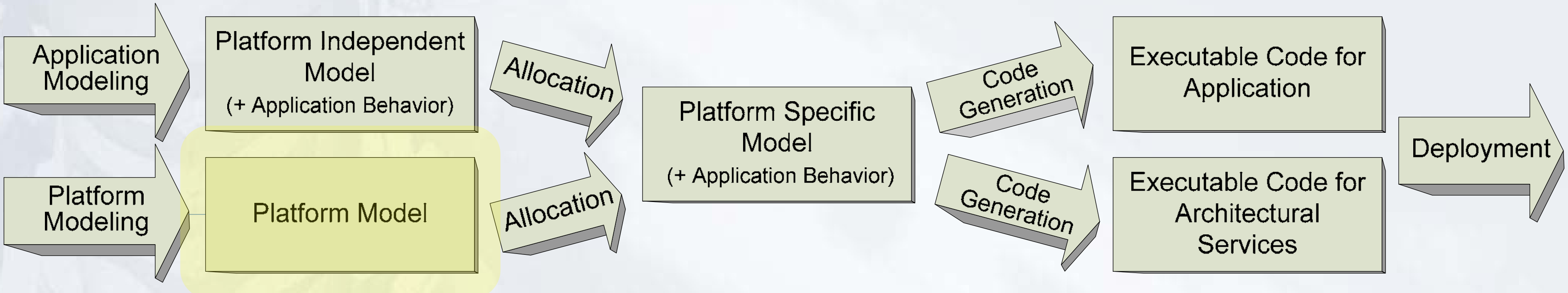
Model-Based Design in DECOS

PIM: Formal specification of the structure and function of a system that abstracts away technical details.



Model-Based Design in DECOS

PIM: Formal specification of the structure and function of a system that abstracts away technical details.



Platform Model: Specification of the available resources on the hardware platform implementing the DECOS architecture.

Model-Based Design in DECOS

PIM: Formal specification of the structure and function of a system that abstracts away technical details.

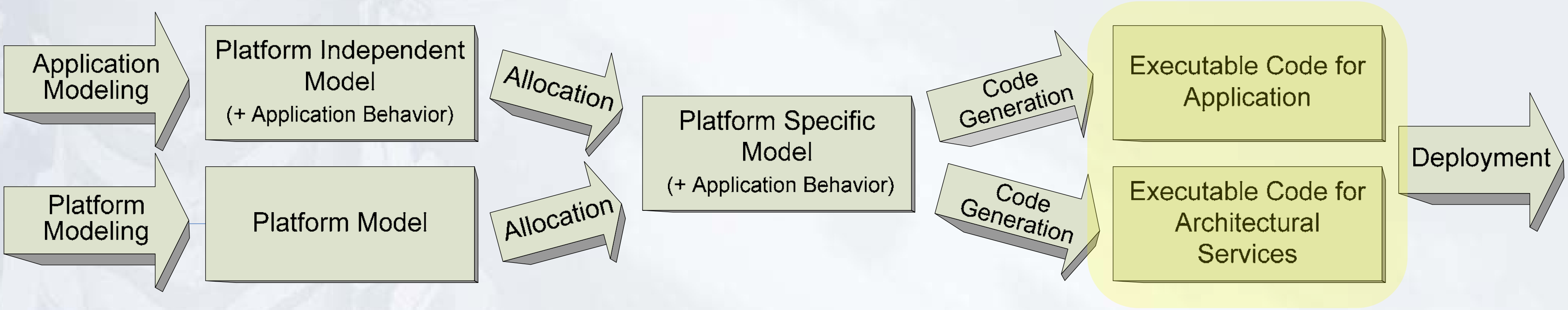
PSM: Extension of the PIM covering the details how the integrated system uses the available resources.

Platform Model: Specification of the available resources on the hardware platform implementing the DECOS architecture.

Model-Based Design in DECOS

PIM: Formal specification of the structure and function of a system that abstracts away technical details.

PSM: Extension of the PIM covering the details how the integrated system uses the available resources.



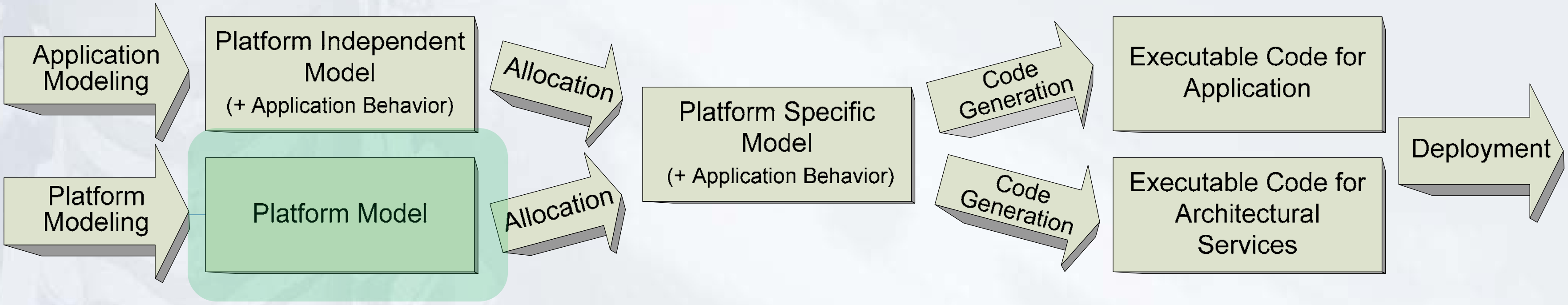
Platform Model: Specification of the available resources on the hardware platform implementing the DECOS architecture.

Executable Code: Resulting from code generation for application jobs and architectural services.

Model-Based Design in DECOS

PIM: Formal specification of the structure and function of a system that abstracts away technical details.

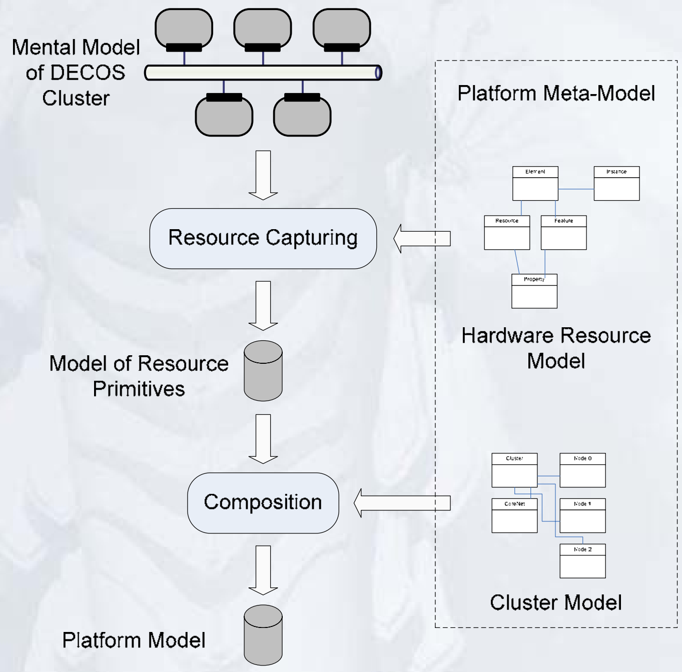
PSM: Extension of the PIM covering the details how the integrated system uses the available resources.



Platform Model: Specification of the available resources on the hardware platform implementing the DECOS architecture.

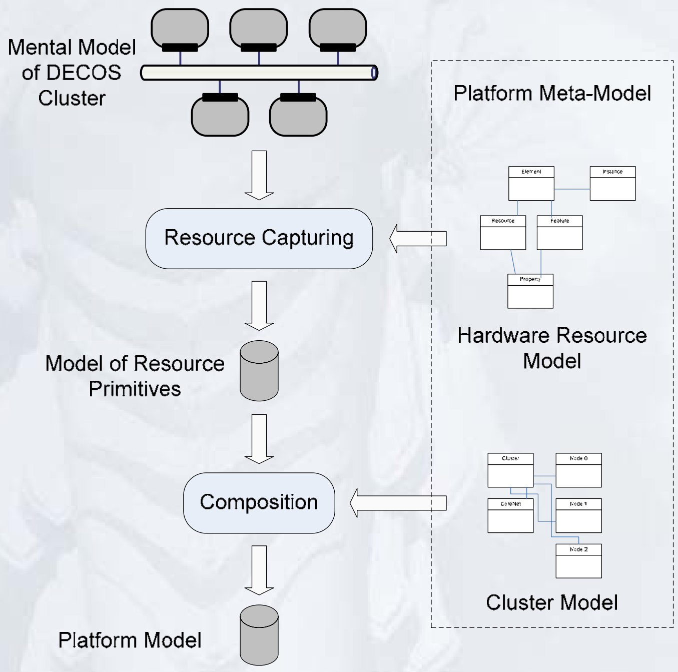
Executable Code: Resulting from code generation for application jobs and architectural services.

Execution Platform Modeling: Platform Meta-Model



- Resource Capturing:
  - Computational Resources
  - Special Purpose HW
  - Communication Resources
- Hardware Resource Model:
  - Meta-model that defines a predefined core set of resource types and resource characteristics
  - Facilitates extensibility and evolvability by the use of “technical dictionaries“

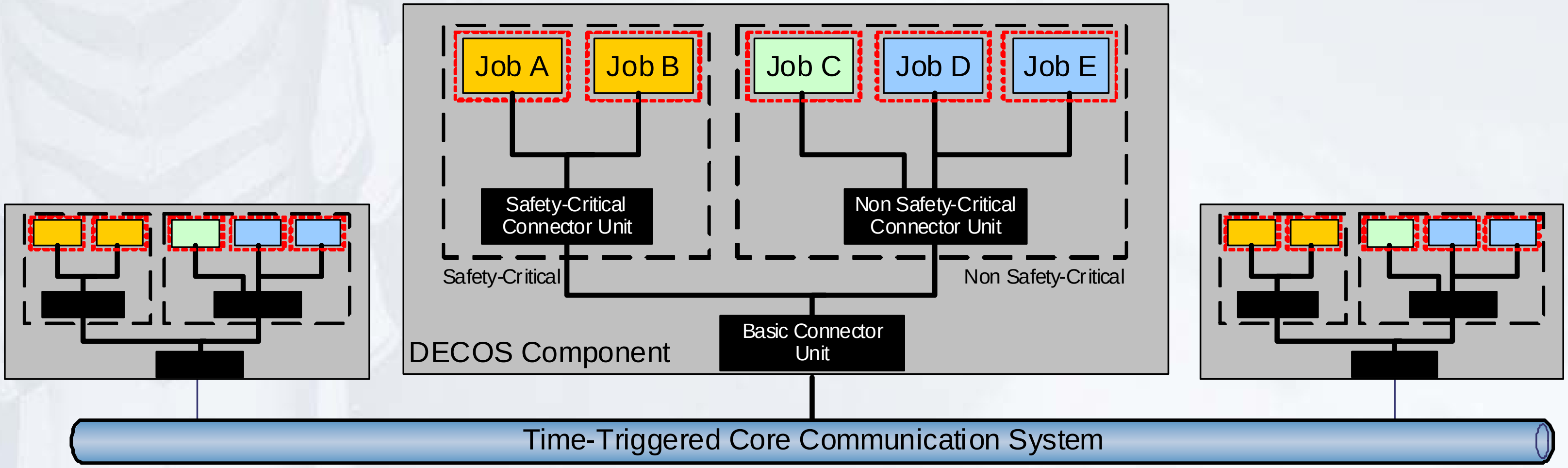
Execution Platform Modeling: Platform Meta-Model



- Resource Composition:
  - Specification of internal setup of individual nodes and the structure of the entire cluster
  - Specification of physical networks for inter- and intra-node communication
- Cluster Model:
  - Meta-model that reflects the setup of DECOS nodes and clusters
  - Additional constraints (OCL) for node and cluster composition that are hard to express in UML

Execution Platform Modeling: DECOS Node/Cluster Setup

- Jobs of different DASs hosted on the same node computer
- Support for mixed criticality applications
- Partitions as encapsulated execution environments for each job
- Encapsulated virtual network for each DAS

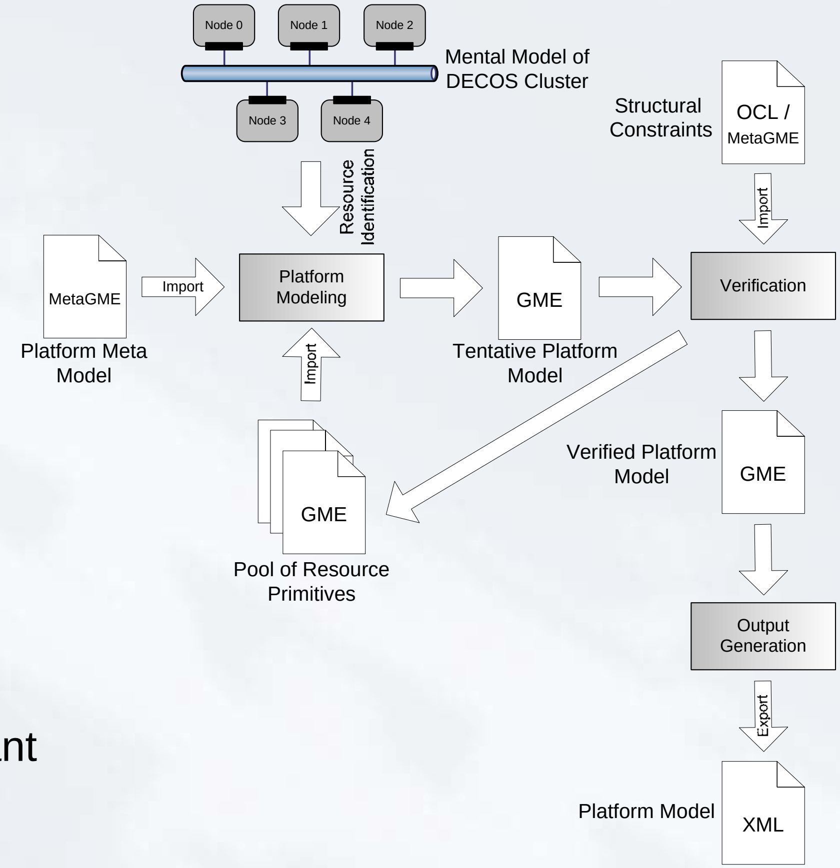


Platform Model Editor

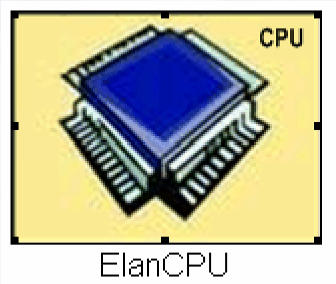
- Based on Generic Modeling Environment (GME), which has been developed at Institute of Software Integrated Systems, Vanderbilt
- Graphical domain-specific editor tailored at modeling hardware platforms for DECOS clusters
- Single tool covering the entire functionality required for modeling the execution platform
- Mental effort for creating and understanding platform models is reduced
- Automatic generation of XML output that is compatible with the DECOS tool chain

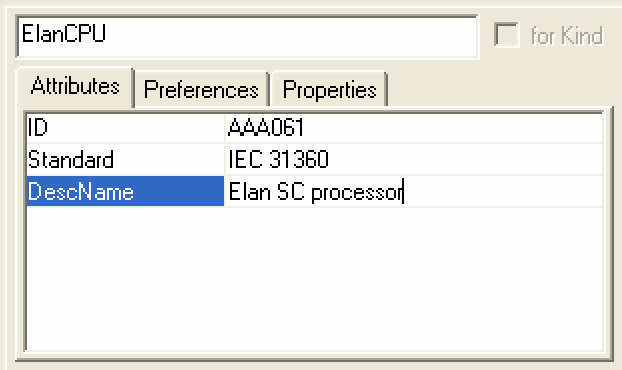
Platform Model Editor: Supported Workflow

- Platform Modeling:
  - Identification of resource primitives
  - Composition to nodes and cluster
- Validation:
  - Compliance check against platform meta-model
  - Compliance check against Object Constraint Language (OCL) constraints
- Output Generation:
  - Automatic transformation to XML (compliant to interface of DECOS tool chain)



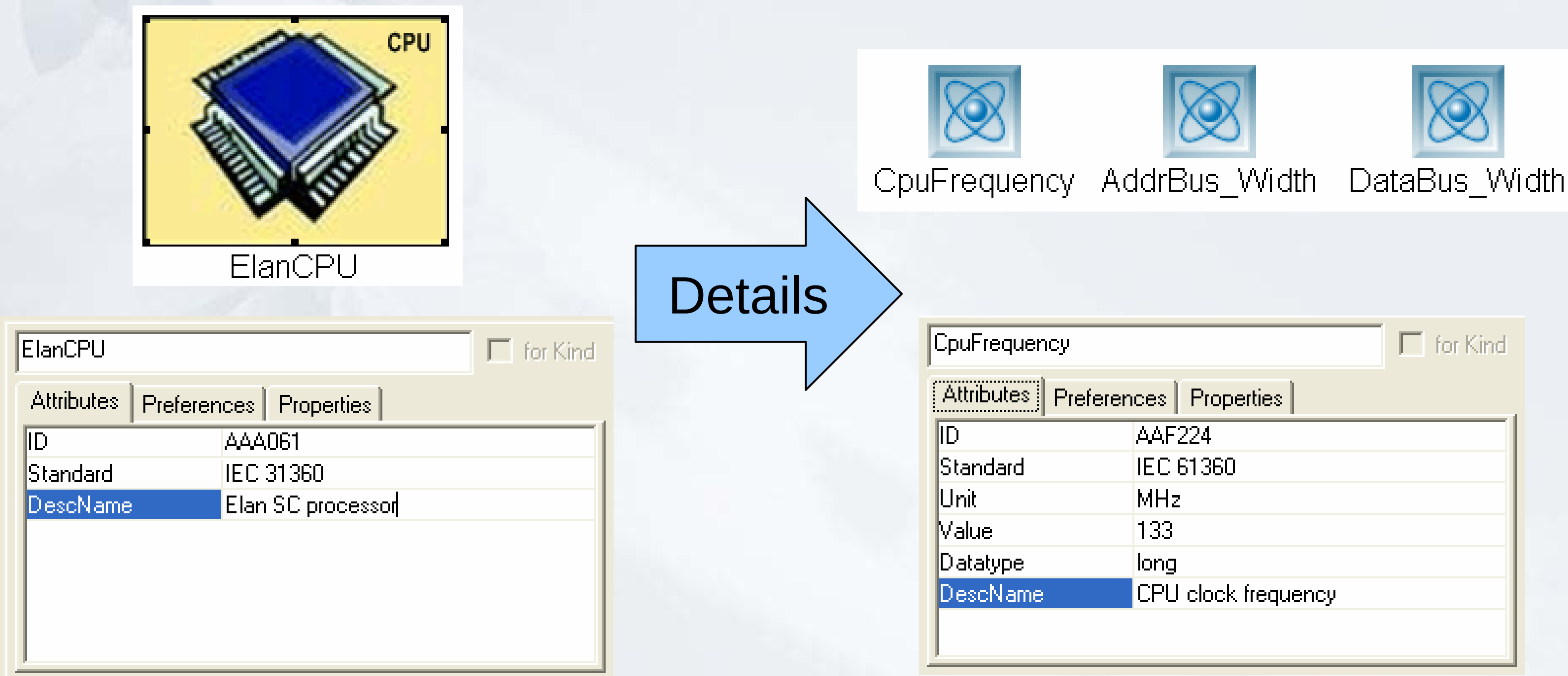
Exemplary Application: Resource Primitives





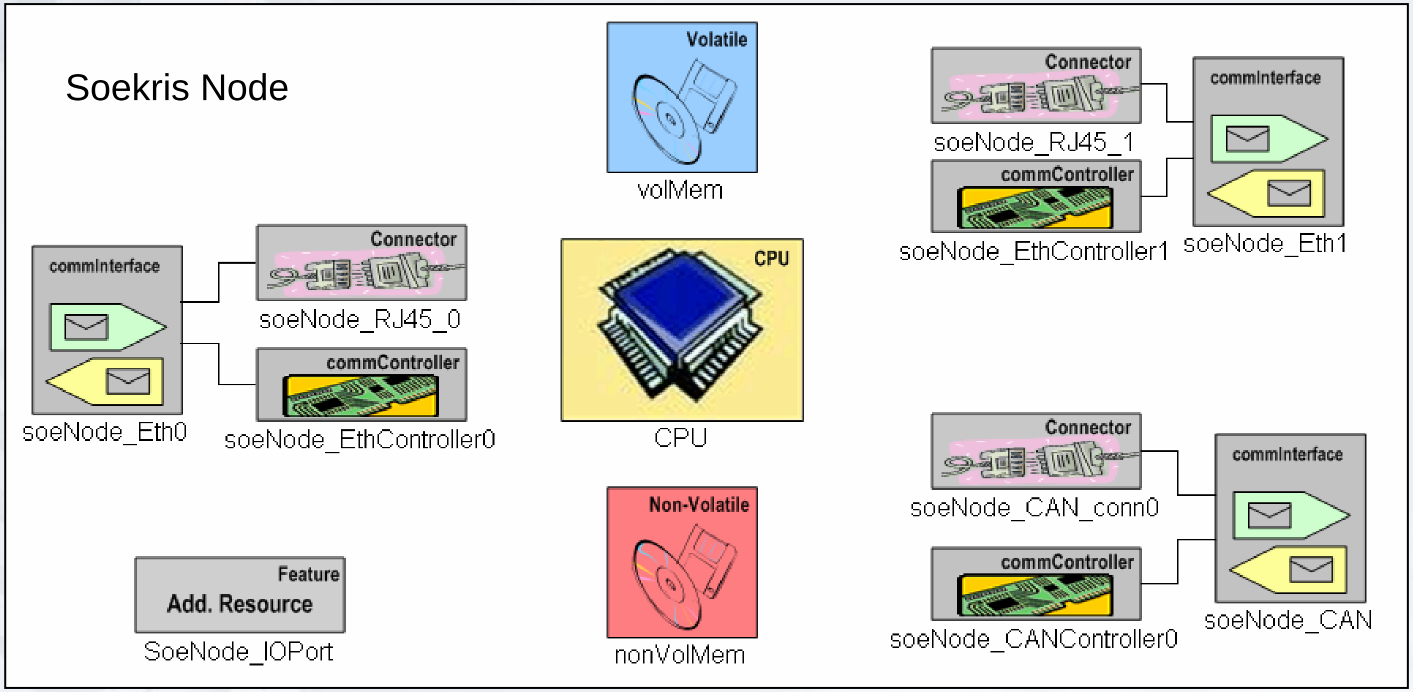
- Identification of resource primitives (e.g. CPU, memory, communication interface, ...)

Exemplary Application: Resource Primitives



- Identification of resource primitives (e.g. CPU, memory, communication interface, ...)

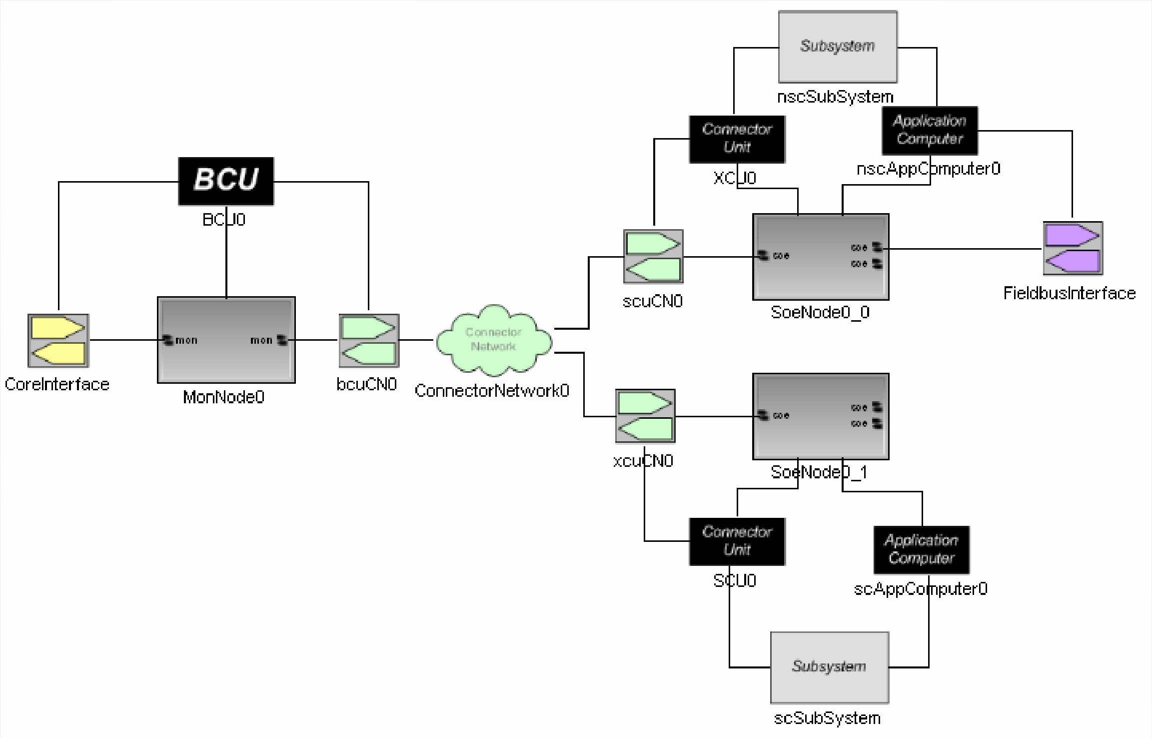
Exemplary Application: Resource Primitives



- Identification of resource primitives (e.g. CPU, memory, communication interface, ...)
- Grouping of several resource primitives to composite resources – so called „hardware elements“ (e.g. embedded computer node)

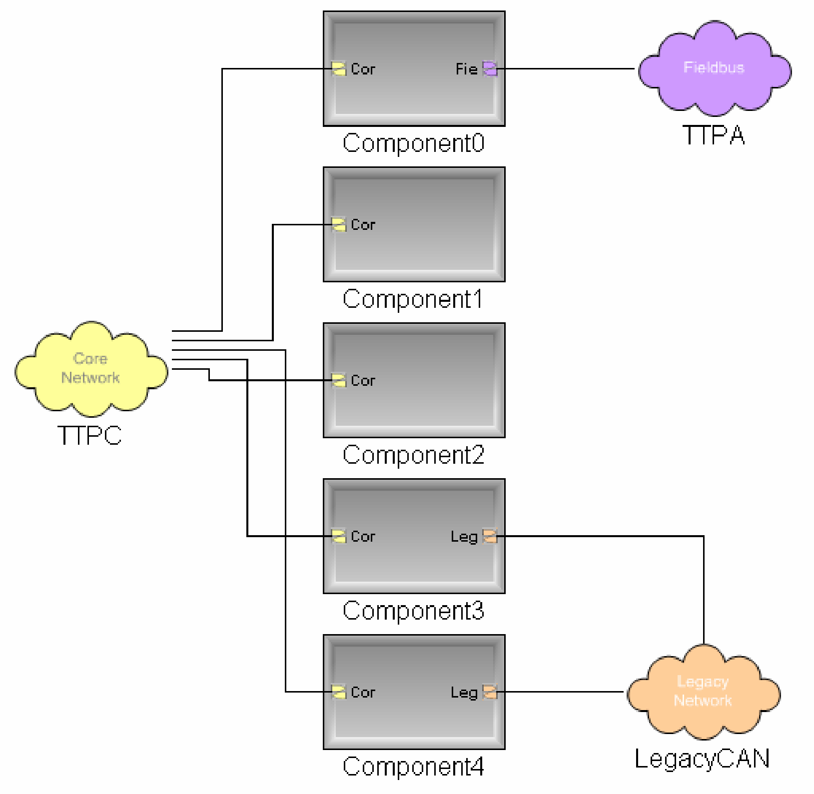
Exemplary Application: DECOS Node Computer

- Composition of a DECOS node computer out of several hardware elements
- Re-use of already modeled hardware elements
- Internal setup of node computer constraint according to DECOS node computer model



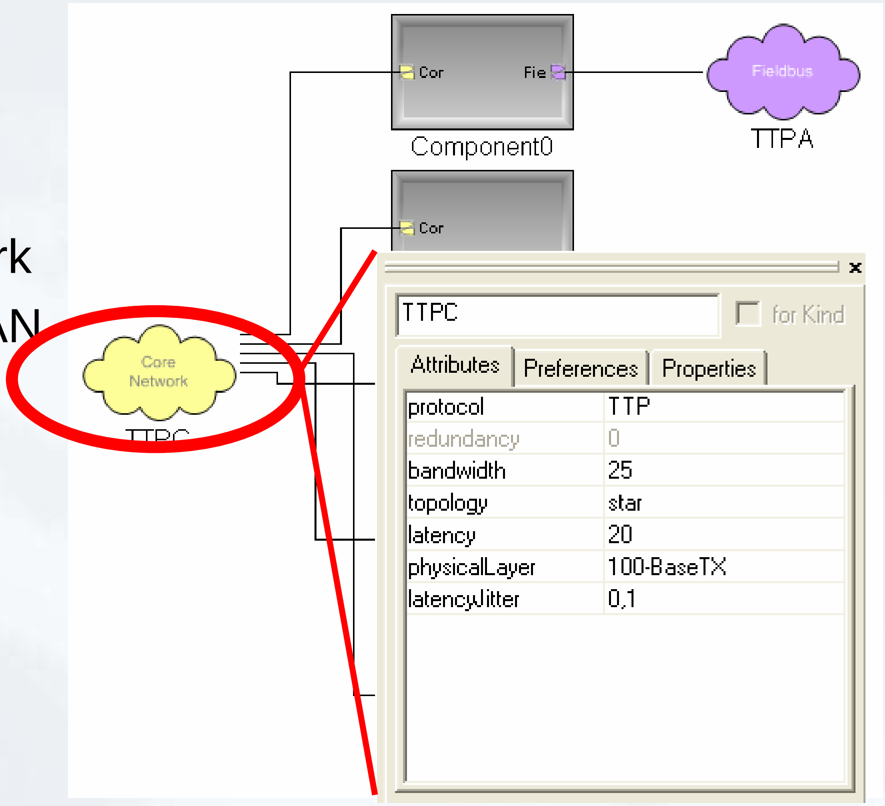
Exemplary Application: DECOS Cluster

- High-level view of a platform model describing a DECOS cluster comprising:
  - Five integrated nodes
  - TTP/C as time-triggered core network
  - Physical interfaces to TTP/A and CAN



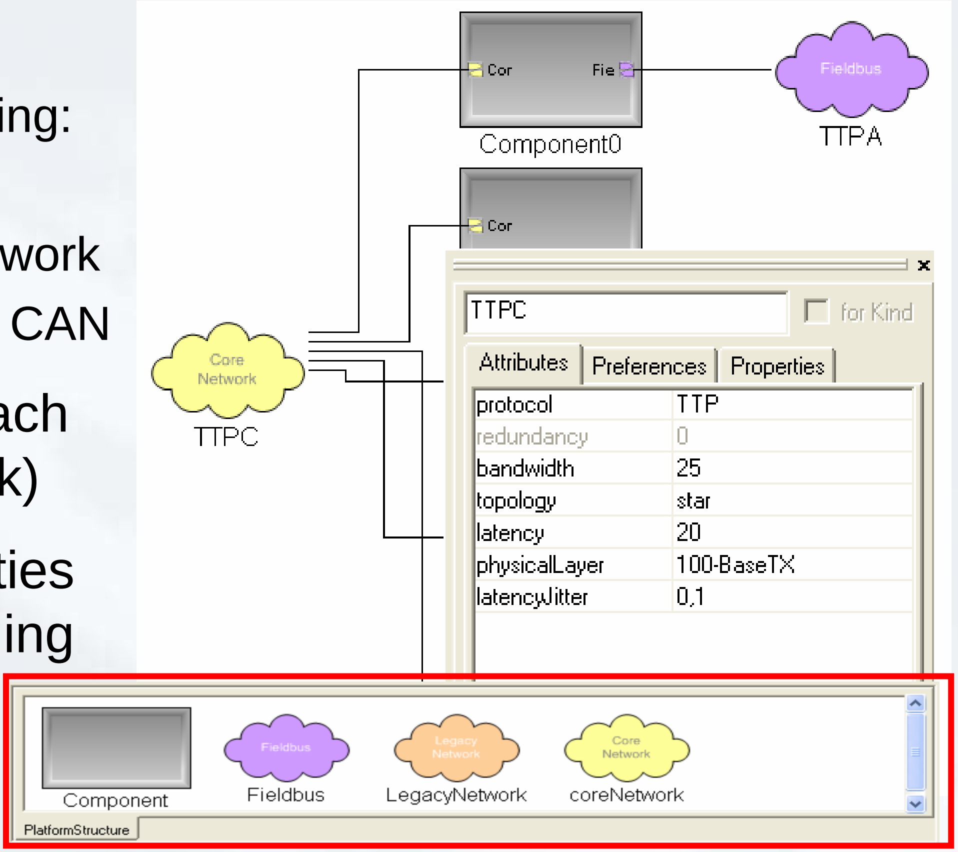
Exemplary Application: DECOS Cluster

- High-level view of a platform model describing a DECOS cluster comprising:
  - Five integrated nodes
  - TTP/C as time-triggered core network
  - Physical interfaces to TTP/A and CAN
- Customized attribute dialog for each modeling entity (e.g. CoreNetwork)



Exemplary Application: DECOS Cluster

- High-level view of a platform model describing a DECOS cluster comprising:
  - Five integrated nodes
  - TTP/C as time-triggered core network
  - Physical interfaces to TTP/A and CAN
- Customized attribute dialog for each modeling entity (e.g. CoreNetwork)
- Pre-selected set of modeling entities reducing the effort for understanding the meta-model



Conclusion

- A precise specification of the available resources at a reasonable level of detail is a mandatory prerequisite for a model-based development of a DECOS system
- The transformation of the PIM to the PSM is guided by the platform model
- Extensibility of the meta-model to be suited for a wide spectrum of hardware platforms is important (technical dictionaries for common understanding)
- Graphical platform editor with intuitive user interface reduces the mental effort for creating and understanding platform models and the occurrence of design faults

Thank you for your attention !

Any Questions?