



Quality-of-Service Modeling and Analysis of Dependable Application Models



András Balogh
András Pataricza
BUTE-DMIS-FTSRG

DECOS

<http://www.decos.at/>



Outline

- Introduction
- Target application domains
- Application models
- QoS modeling in PIM
- Platform models
- Model integration
- Analysis of PSMs
- Conclusions

Introduction

- Software Engineering
 - Paradigm evolution (object-oriented)
 - Improve quality
 - Maintain the gap between software and problem domains
- Current effort
 - Model-driven development
 - Higher abstraction level
 - Reduced manual coding
- Problem
 - QoS aspects are not handled systematically

Target application domains

■ Business systems

- Large, distributed multi-tier software
- Heterogeneous platforms
- Rapid evolution of requirements
- Service-Oriented Architecture
 - Components as loosely coupled web services
 - Process orchestration components
- Business-critical functions
 - Performance, security, availability is a key aspect

Target application domains

■ Embedded systems

- focus: automotive, aerospace
- Distributed, networked systems
- Heterogeneous hardware
 - 32 bit embedded processor boards with rich set of resources
 - Small, intelligent sensor/actuator nodes
- Safety critical functions
 - X-by-wire
 - Timeliness, availability is a key aspect

Modeling of applications

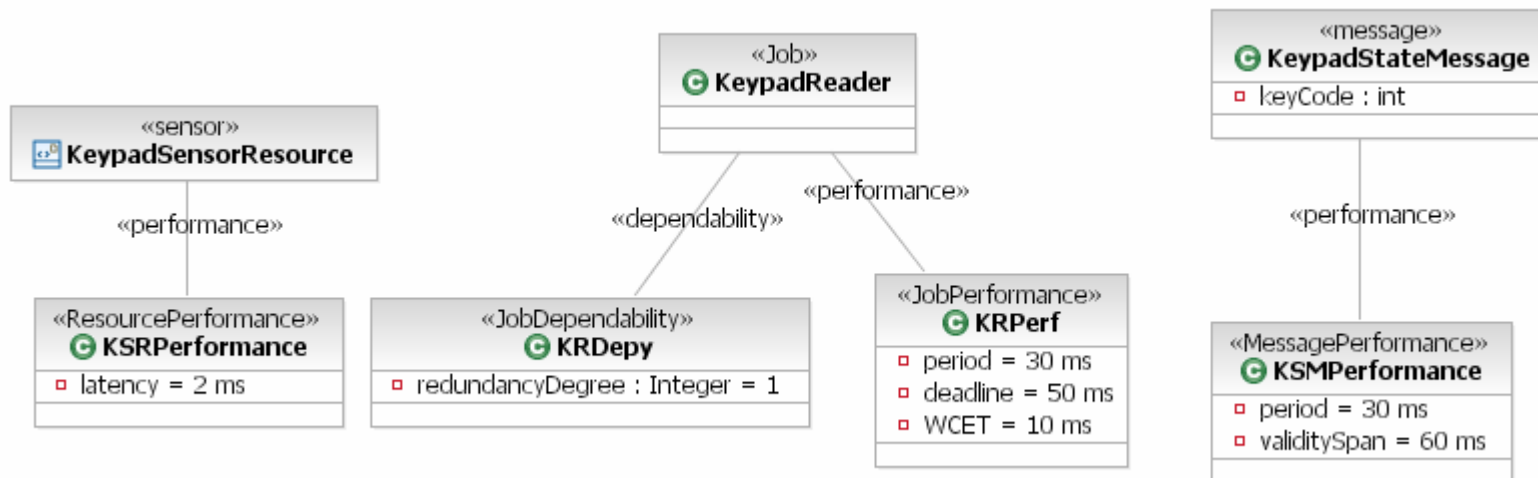
■ Properties of applications

- Loosely coupled,
- Communicating components
- Distributed to a set of computing nodes

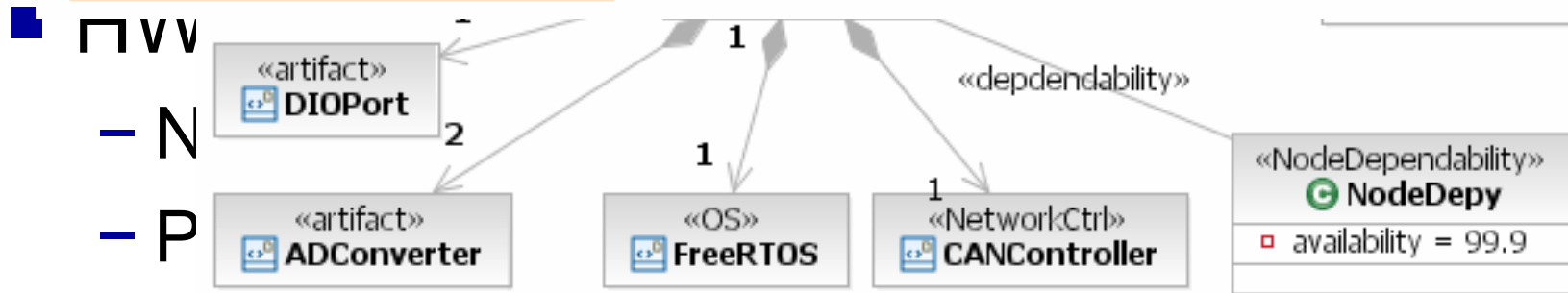
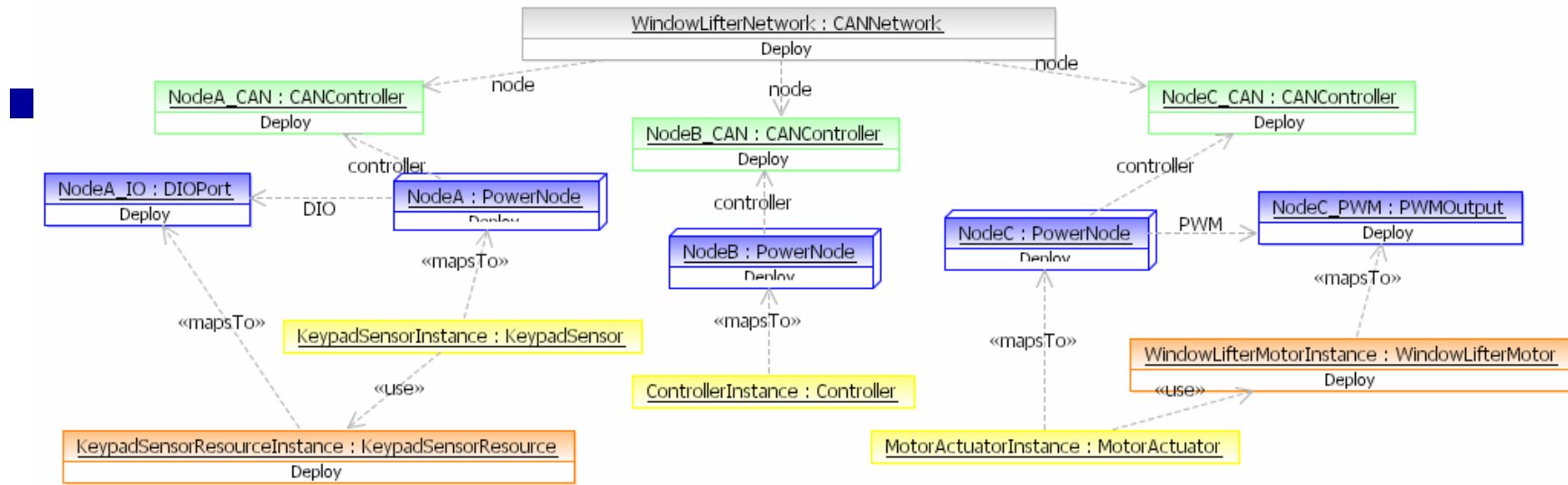


QoS modeling of PIM components

- An extension to standard UML needed
 - Profile for modeling QoS and FT characteristics
 - Profile for schedulability, performance and time
- Compact form
 - Selected attributes in stereotyped classes
 - Job: period, deadline, WCET



Platform modeling



– Network links

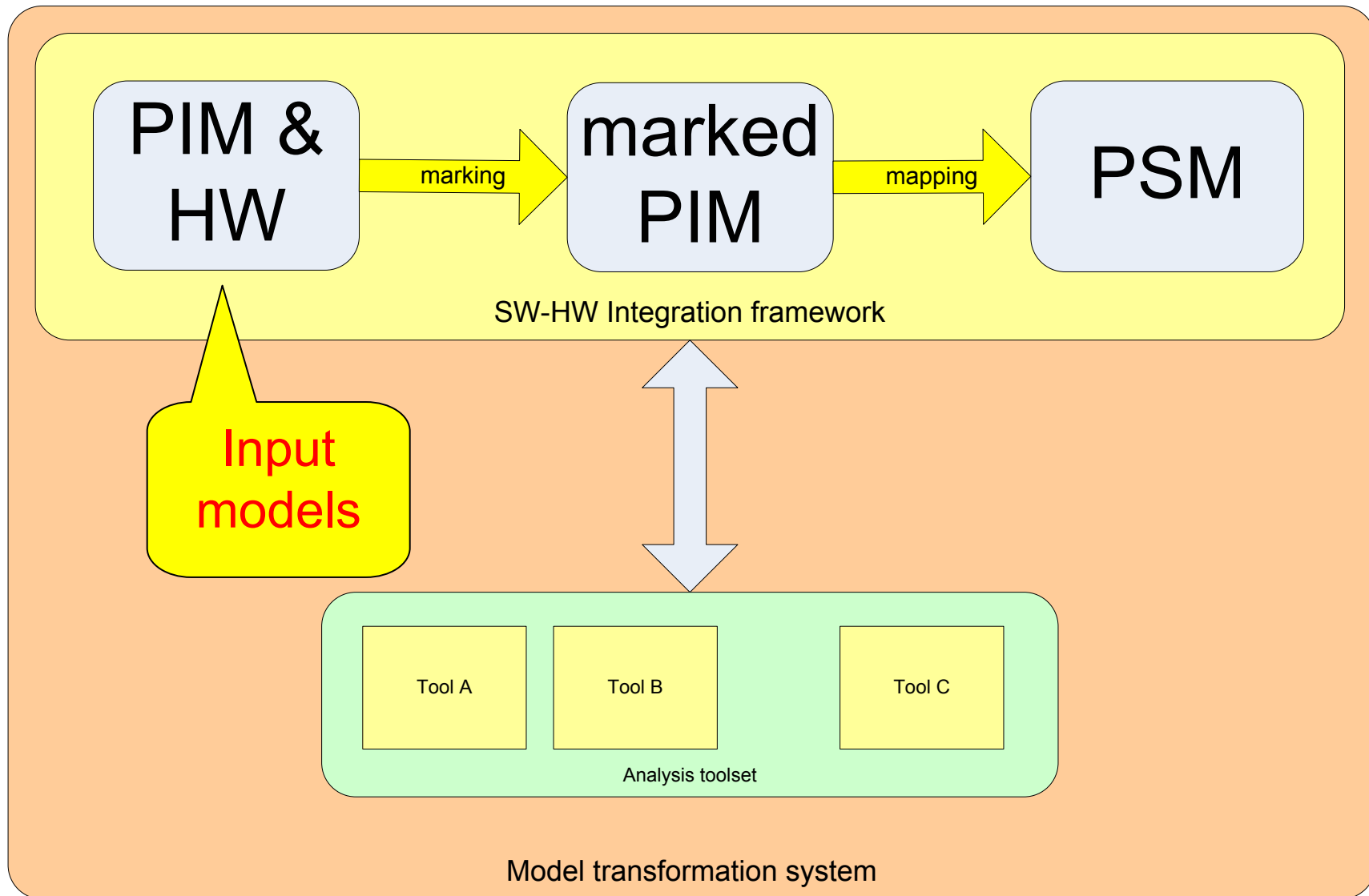
HW-SW Integration (PIM-PSM mapping)

- Traditional MDA
 - Inputs: PIM, platform model
 - Concept: highly automated process
- Embedded systems
 - Several additional constraints
 - Designer wants to take manual decisions
 - → PIM marking is needed
 - Additional information from the system designer
 - ▣ job-node pre-allocation
 - ▣ Logical-physical resource pairing
 - ▣ ...

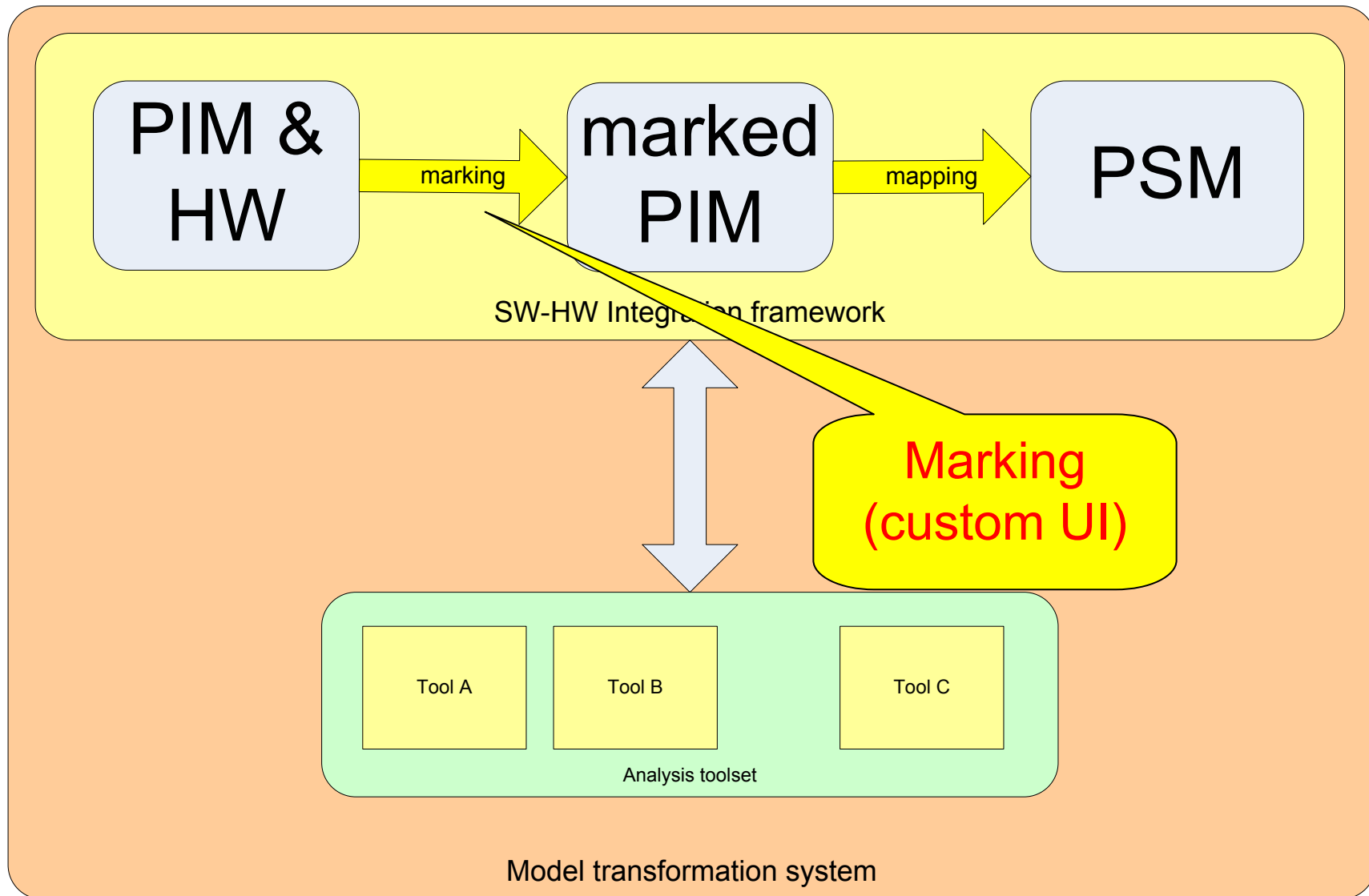
HW-SW Integration (PIM-PSM mapping)

- Complex PIM-PSM
 - Manual marking
 - Automatic mapping
 - Job allocation
 - Comunication schedule
 - Job schedule
 - Approach: integration of existing tools by model transformations
 - Problem:
 - QoS analysis needs to be completed

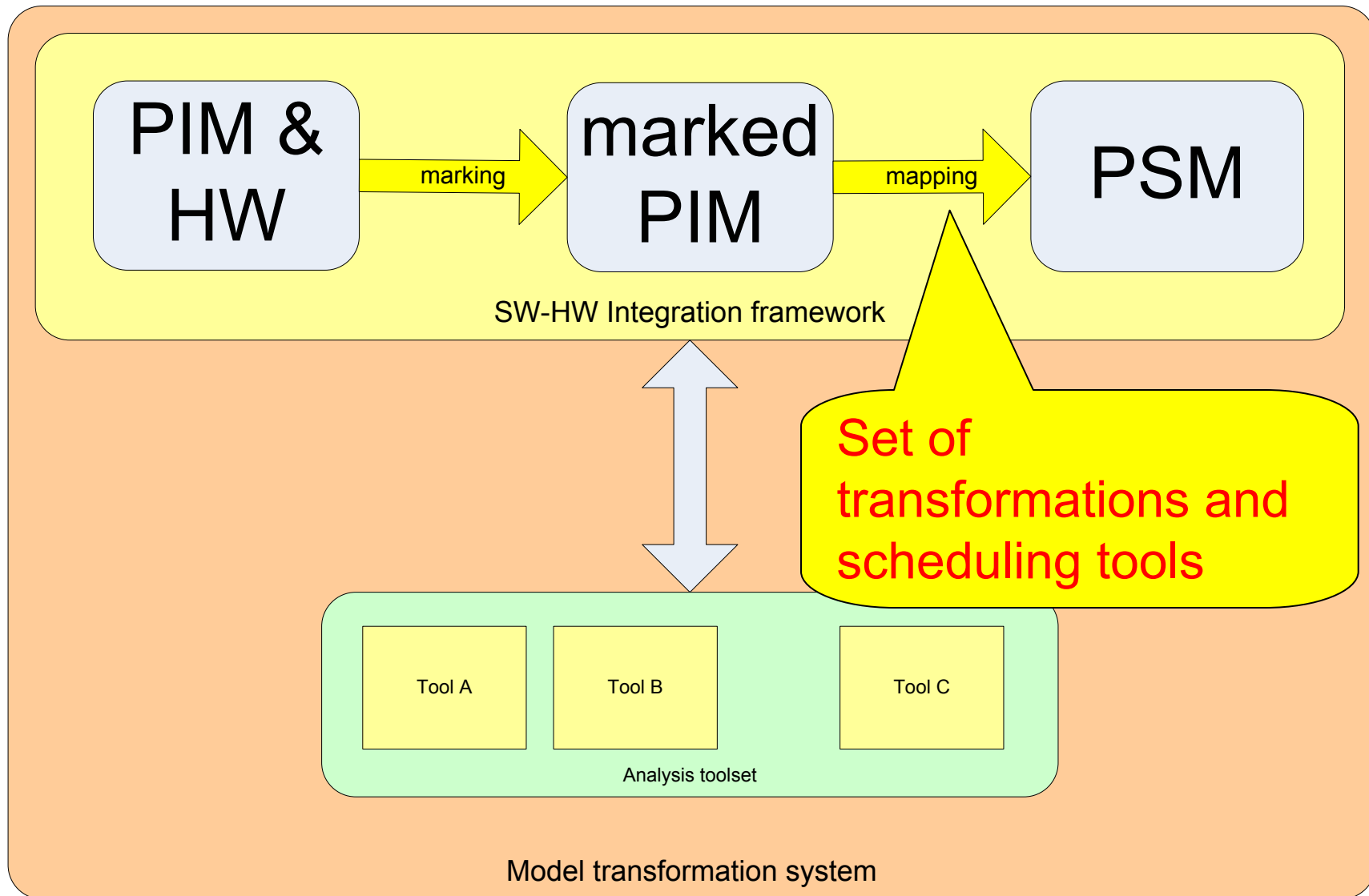
HW-SW integration workflow



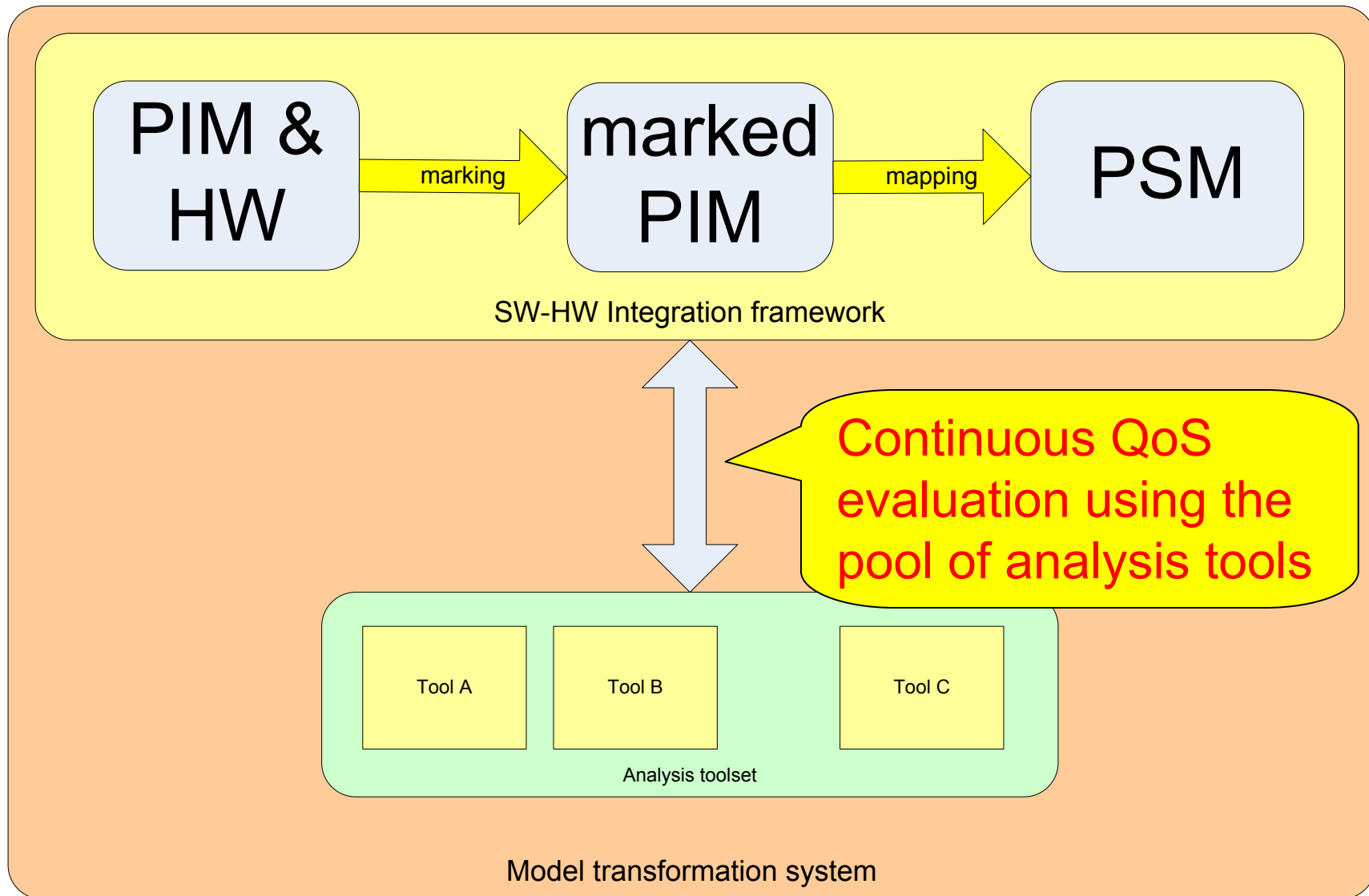
HW-SW integration workflow



HW-SW integration workflow



HW-SW integration workflow



Continuous QoS evaluation

■ Advantage

- Early feedback to the designer
 - During marking
 - During the automatic mapping

■ Technique

- Static analysis of the intermediate models
 - Analyze completed aspects (eg. Availability after allocation)
 - Using constraint solvers to solve constraints with open variables

Model analysis

- Most important aspects
 - Timeliness
 - Performance (end-to-end)
 - Availability
 - Error propagation/ error containment

Timeliness/performance analysis

■ Hypothesis

- Jobs are black boxes
 - Internal behavior modeled and generated by other tools
- worst-case values are calculated
- The developer is interested in an end-to-end value
 - ES: sensor to actuator
 - BS: high level request to response
- Platform parameters are known

Timeliness/performance analysis 2.

■ Steps

- Transformation from UML to stochastic queuing networks
- Calculating the values using a mathematical tool
 - Java Modeling Tools – Politecnico di Milano
- Results are compared to PIM requirements
 - If a constraint is violated the system generates an error message
- Results can also be imported to the model

From UML to queuing networks

UML Element	QN Element
User request/input data	Source element
Sensor	Delay node
Job	Processor node
Network link	Processor node
Actuator	Delay node
Response/output data	Sink element

Availability analysis

■ Hypothesis

- The availability of a given service is interesting
 - ES: The availability of the actuator output
 - BS: The availability of response message to the user
- The availability of platform is known (measurements)
- The implementation of the jobs is correct
 - Automatic code generation and model verification

Availability analysis 2.

- A tree is created
 - nodes: hw and sw components
 - Edges: dependencies
 - Job *runs on* node
 - JobA *sends message* to jobB
- Evaluation: simple model transformation
 - Calculates transitive dependencies for all nodes
 - Availability of a node equals the production of availabilities of hardware nodes connected to it

Implementation results

- Business systems
 - Automatic QoS-driven allocation of web services to servers (ISAS 2005 paper)
- Embedded systems
 - DECOS HW-SW Integration Framework
 - EU FW 6 Integrated Project
 - Our framework contains the concepts introduced here

Conclusions

- Our approach
 - Integrates MDA with analysis methods
 - Supports early recognition of design problems
 - Works in several application domains
- Limitations
 - Platform parameters must be known
 - Jobs are handled as black boxes