UML MARTE
www.omgmarte.org

Máster en Ingeniería Informática

Modelado de Requisitos con UML y MARTE (El perfil UML para el modelado y análisis de sistemas empotrados y de tiempo real)

Julio Medina
(julio.medina@unican.es)

Universidad de Cantabria

cea list THALES INRIA Ctr UC

UML MARTE
www.omgmarte.org

Acknowledgment

- This presentation reuses and extends material prepared by the ProMARTE partners for the Object Management Group
- The initial presentation (realtime/07-03-14) is available to OMG members
- Requirements related material has been reused from *Using MARTE and SysML for Modeling Real-Time Embedded Systems*, presented by Huascar Espinoza in *International School for model-driven design for distributed real-time embedded systems*

Modeling Real-Time and Embedded systems in UML

- **UML is desired as a possible solution to address the Real-Time and Embedded systems modelling domain**
  - A large audience in the Software Engineering community
  - Steady semantic foundations
  - Extension capabilities through UML profiles (e.g. SysML)
  - But lacks key notions to fully address RTE specifics (time, resource, scheduling)
- **Previous attempts to adapt UML to the RTE domain**
  - Academic initiatives (e.g. ACCORD)
  - UML profile for Scheduling, Performance and Time (SPT)
    - The first OMG adopted specification in this domain
    - Defines annotation mechanisms to perform quantitative analysis
    - Required major improvements over time

In 2005, OMG called for a new UML profile for Modeling and Analysis of Real-Time and Embedded systems (MARTE)

Introducing MARTE

- **“The UML profile for MARTE addresses modeling and analysis of real-time and embedded systems, including their software and hardware aspects”**

- **Key features**
  - Provides support for non-functional property modeling
  - Adds rich time and resource models to UML
  - Defines concepts for software and hardware platform modeling
  - Defines concepts for allocation of applications on platforms
  - Provides support for quantitative analysis (e.g. scheduling, performance)
  - Complies with UML 2.1 and other existing OMG standards
  - Replaces the UML SPT profile 1.1

- **MARTE specification adopted in June 2007**
  - Current version available: http://www.omg.org/cgi-bin/doc?formal/2011-06-02
  - A Revision Task Force is currently updating it to its 1.2 version.
  - The community reflects now about requirements for a new MARTE 2.0 version

The ProMARTE partners

UML MARTE
www.omgmarte.org

Tool vendors

- ARTiSAN Software Tools*
- International Business Machines*
- Mentor Graphics Corporation*
- Softeam*
- Telelogic AB (I-Logix*)
- Tri-Pacific Software
- No Magic
- The Mathworks

Industrial companies

- Alcatel*
- France Telecom
- Lockheed Martin*
- Thales*

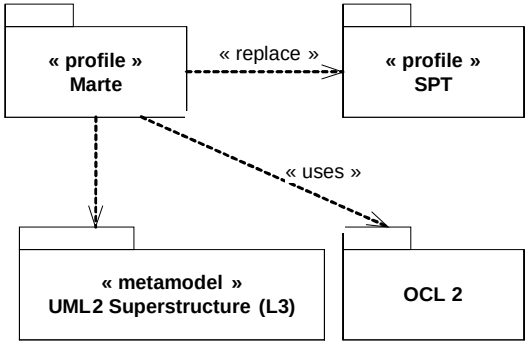
Academics

- Carleton University
- Commissariat à l'Energie Atomique
- ESEO
- ENSIETA
- INRIA
- INSA from Lyon
- Software Engineering Institute (Carnegie Mellon University)
- Universidad de Cantabria

* Submitters to the OMG UML for MARTE RFP

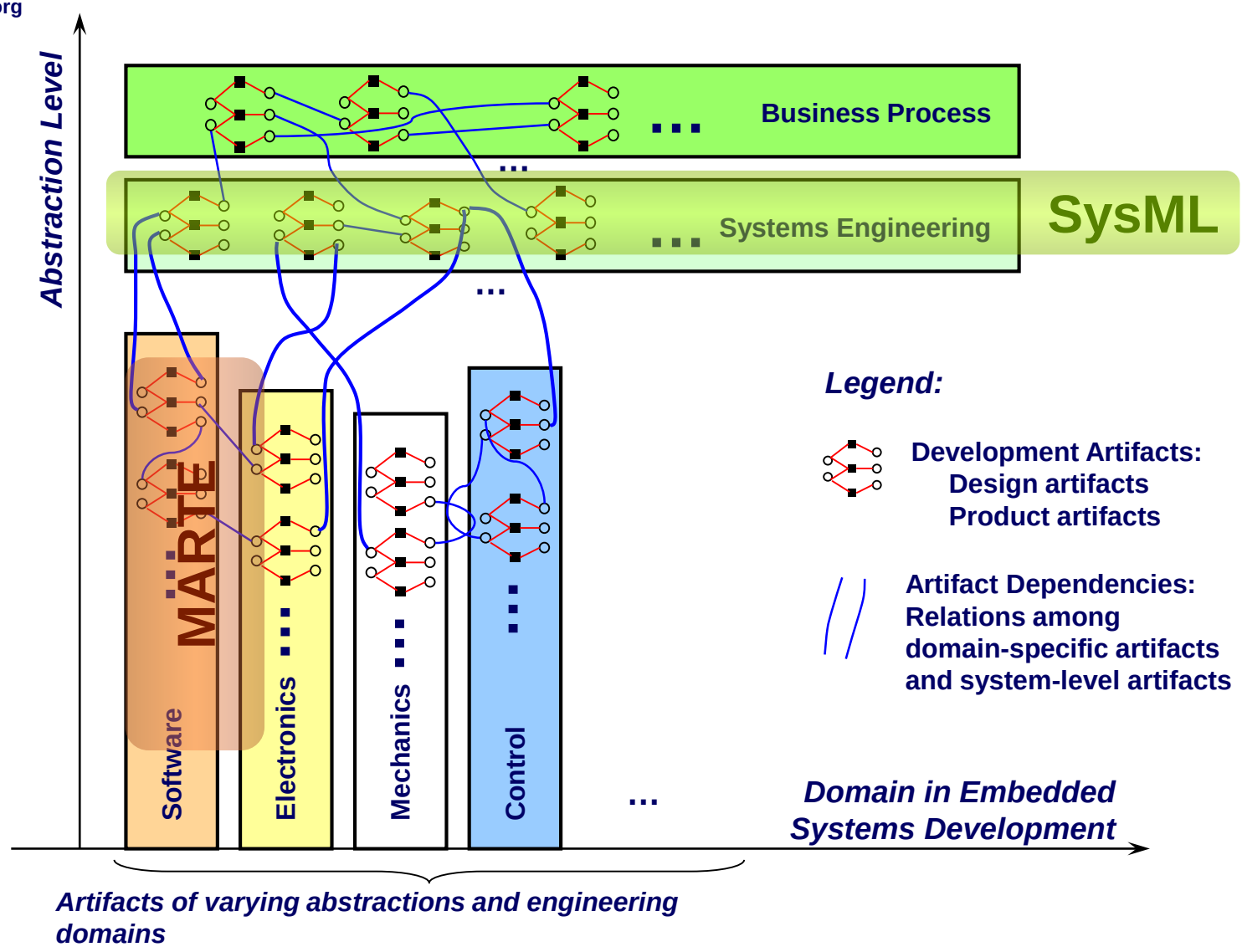
Relationships with other OMG standards

- **Relationships with generic OMG standards**
  - Profiles the UML 2 superstructure meta-model
  - Uses OCL 2 for description of domain constraints



- **Relationships with RTE specific OMG standards**
  - The UML profile for Modeling QoS and FT Characteristics and Mechanisms
    - Addressed through MARTE NFP package
  - The UML profile for SoC
    - More specific than MARTE purpose
  - The Real-Time CORBA profile
    - Real-Time CORBA based architecture can be annotated for analysis with MARTE
  - The UML profile for Systems Engineering (SysML)
    - Specialization of SysML allocation concepts and reuse of flow-related concepts
    - Ongoing discussion to include VSL in next SysML version
    - Overlap of team members

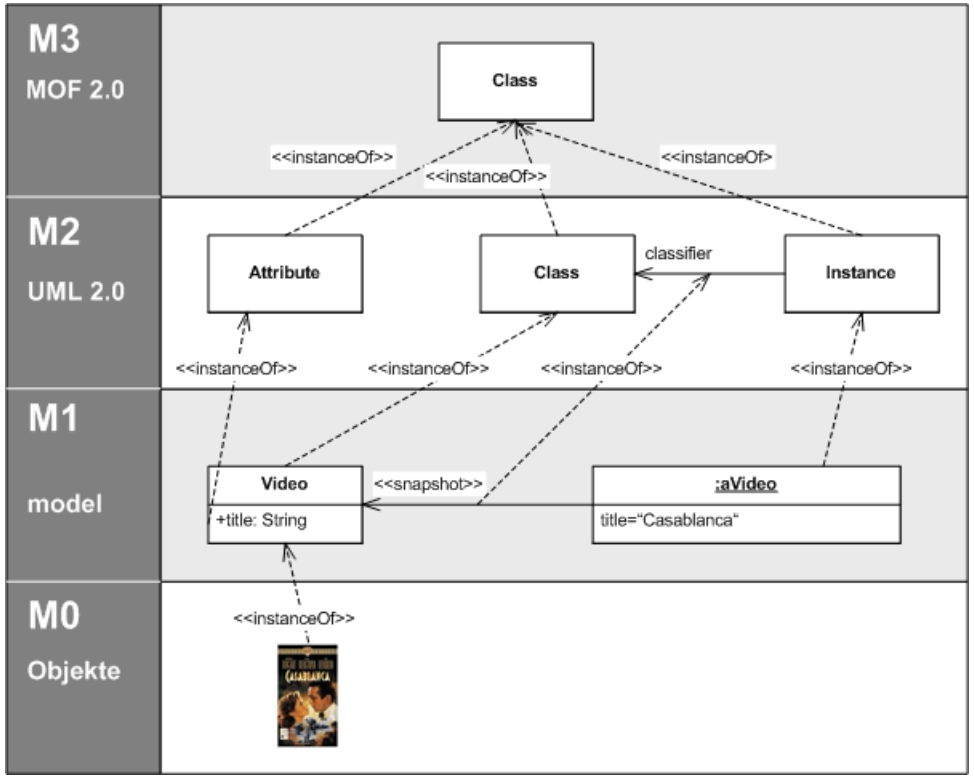
Identifying the System's level of abstraction?



Identifying the modelling level

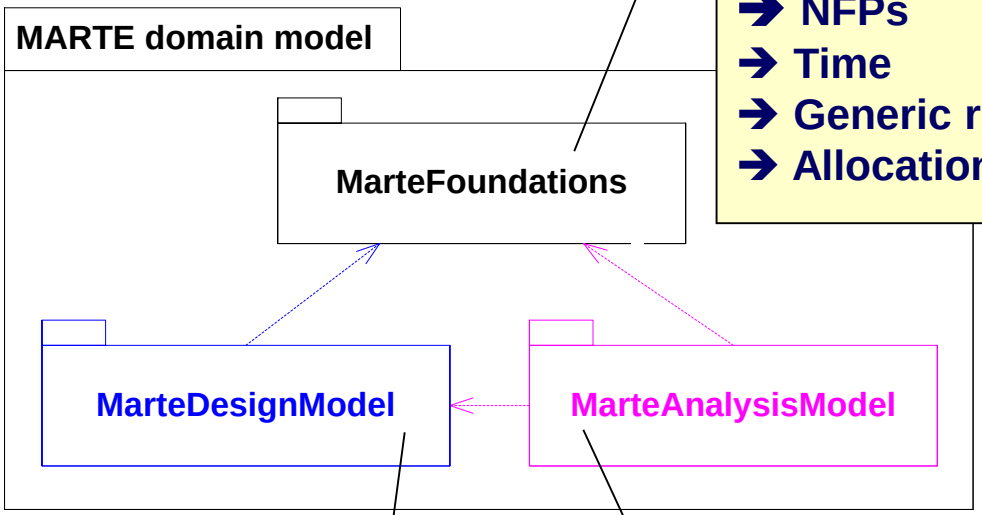
- **Meta object facilities**
  - Self-descripting sub-set
- **Meta-models**
  - Models of Languages and dialects
- **Models**
  - Abstract view of a reality sufficient to understand it
- **Objects**
  - Real (or close to real) representation of a physical entity

Profiles



MARTE is a Profile that Extends the UML Meta-model

MARTE Overview



Foundations for RT/E systems modeling and analysis:
- ➔ CoreElements
- ➔ NFPs
- ➔ Time
- ➔ Generic resource modeling
- ➔ Allocation

Specialization of MARTE foundations for modeling purpose (specification, design…):
- ➔ Generic component model
- ➔ High-level application modeling
- ➔ Software resource modeling
- ➔ Hardware resource modeling

Specialization of foundations for annotating model for analysis purpose:
- ➔ Generic quantitative analysis
- ➔ Schedulability analysis
- ➔ Performance analysis

UML MARTE
www.omgmarte.org

How to read, use, & implement MARTE

(see Section 2)

Extension Units

NFP	Non-Functional Properties	Section 8
Time	Enhanced Time Modeling	Section 9
GRM	Generic Resource Modeling	Section 10
Alloc	Allocation Modeling	Section 11
GCM	Generic Component Model	Section 12
HLAM	High-Level Application Modeling	Section 13
SRM	Software Resource Modeling	Section 14.1
HRM	Hardware Resource Modeling	Section 14.2
RTM	Real-Time objects Modeling	Section 13
GQAM	Generic quantitative Analysis	Section 15
SAM	Schedulability Analysis	Section 16
PAM	Performance Analysis	Section 17
VSL	Value Specification Language	Annex B
CHF	Clock Handling Facilities	Annex C
RSM	Repetitive Structure Modeling	Annex E
AADL	AADL models with UML	Section A.2

UML MARTE
www.omgmarte.org

Usage & Compliance Cases vs. Extension Units

Table 7.2 - Extension Units that must be supported in each Compliance Case

CASE	Level	GRM	NFP	VSL	Time	CHF	SRM	HRM	GCM	Alloc	HLAM	GQAM	PAM	SAM	RSM
Software	Base	X	X		X						X				
	Full			X		X	X		X						
Hardware	Base	X	X		X			X							
	Full			X		X			X	X					X
System	Base	X	X		X					X	X				
	Full			X		X	X	X	X						X
Performance	Base	X	X		X							X	X		
	Full			X		X									
Schedulability	Base	X	X		X							X		X	
	Full			X		X									
Infrastructure	Base	X	X		X		X								
	Full			X		X				X	X				
Methodologist	Base	X	X		X			X			X	X			
	Full			X		X	X		X	X			X	X	X

UML MARTE
www.omgmarte.org

Requirements Engineering and UML

- ***"The process by which the requirements for systems and software products are gathered, analyzed, documented, and managed throughout the development life cycle"***
- **UML has traditionally been used to document requirements by means of Use Case diagrams:**
  - **➔Lack of well defined semantics.**
  - **➔Limitations for managing traceability.**
  - **➔Useful for functional requirements, but not precise enough for non-functional requirements**
- **SysML and MARTE provide some improvements in these aspects…**

- From: ***Using MARTE and SysML for Modeling Real-Time Embedded Systems***, presented by Huascar Espinoza in ***International School for model-driven design for distributed real-time embedded systems***

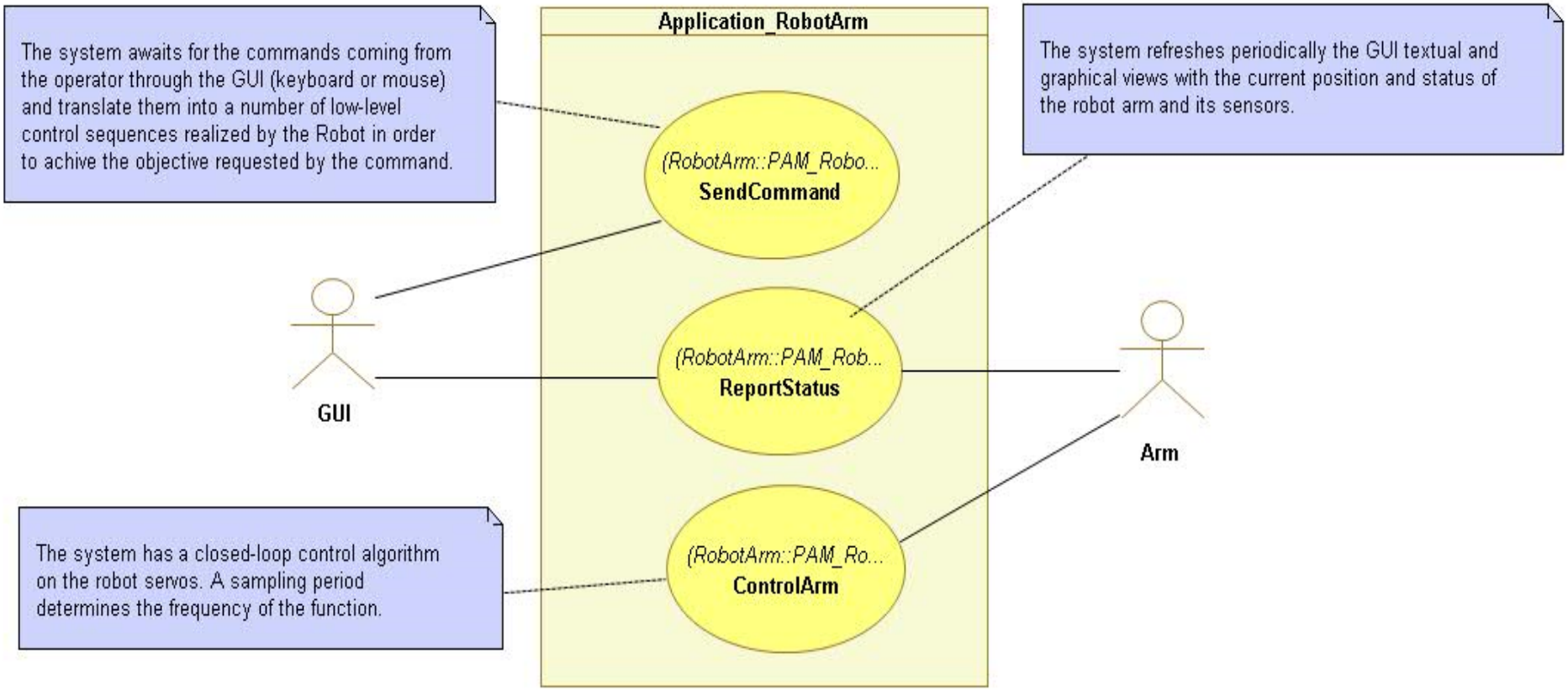
UML MARTE
www.omgmarte.org

Modeling Requirements with SysML

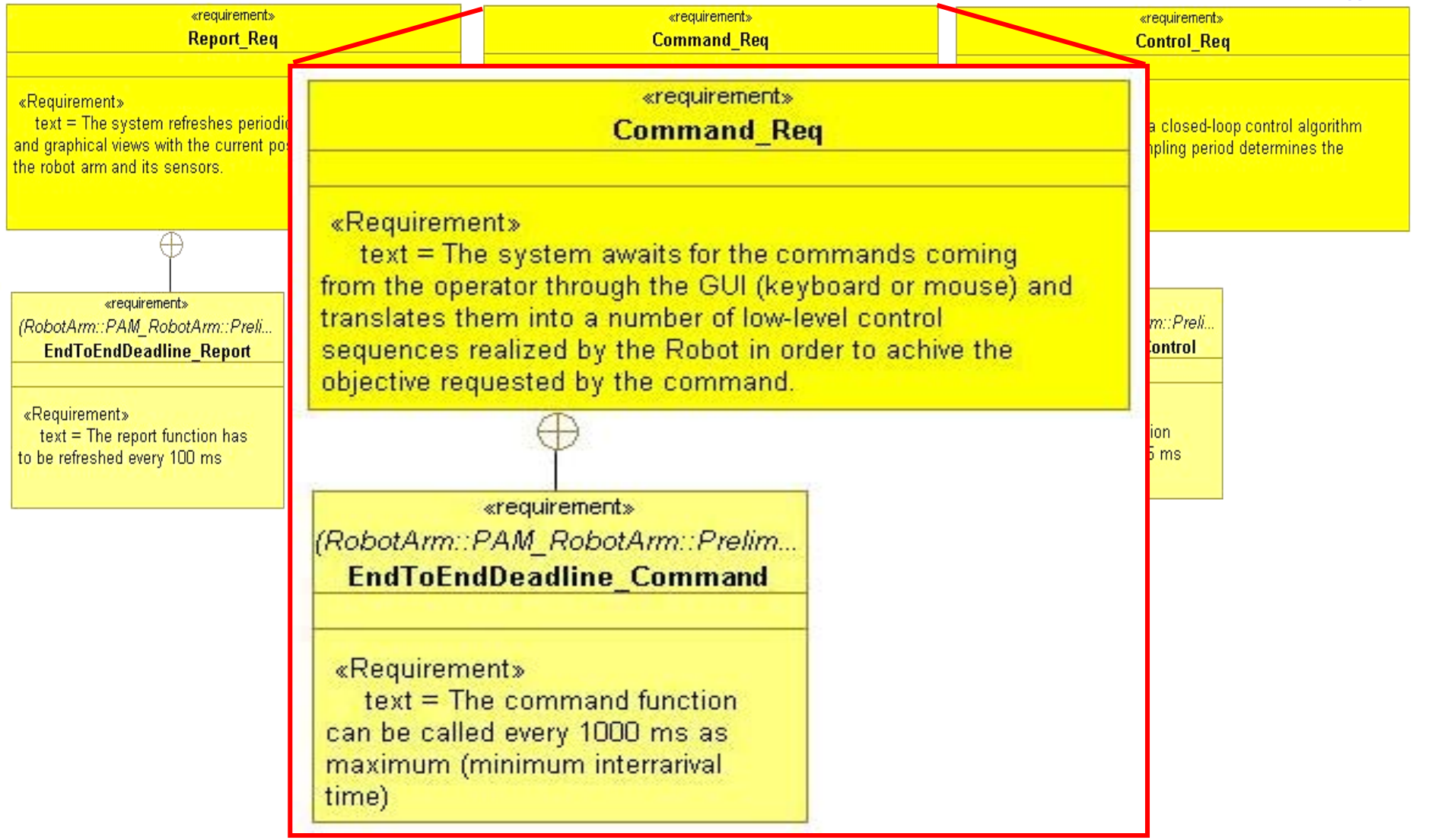
- **Explicitly model requirements relationships:**
  - ➔ **Between requirements (e.g. Refine)**
  - ➔ **Between requirements and architectural solutions (e.g. Satisfy)**

- **Keep traceability with specialized tools:**
  - ➔ **E.g. Requisite Pro, Rectify, or DOORS**
  - ➔ **Support for traceability analysis, flow-down, derivation, assignment, etc.**

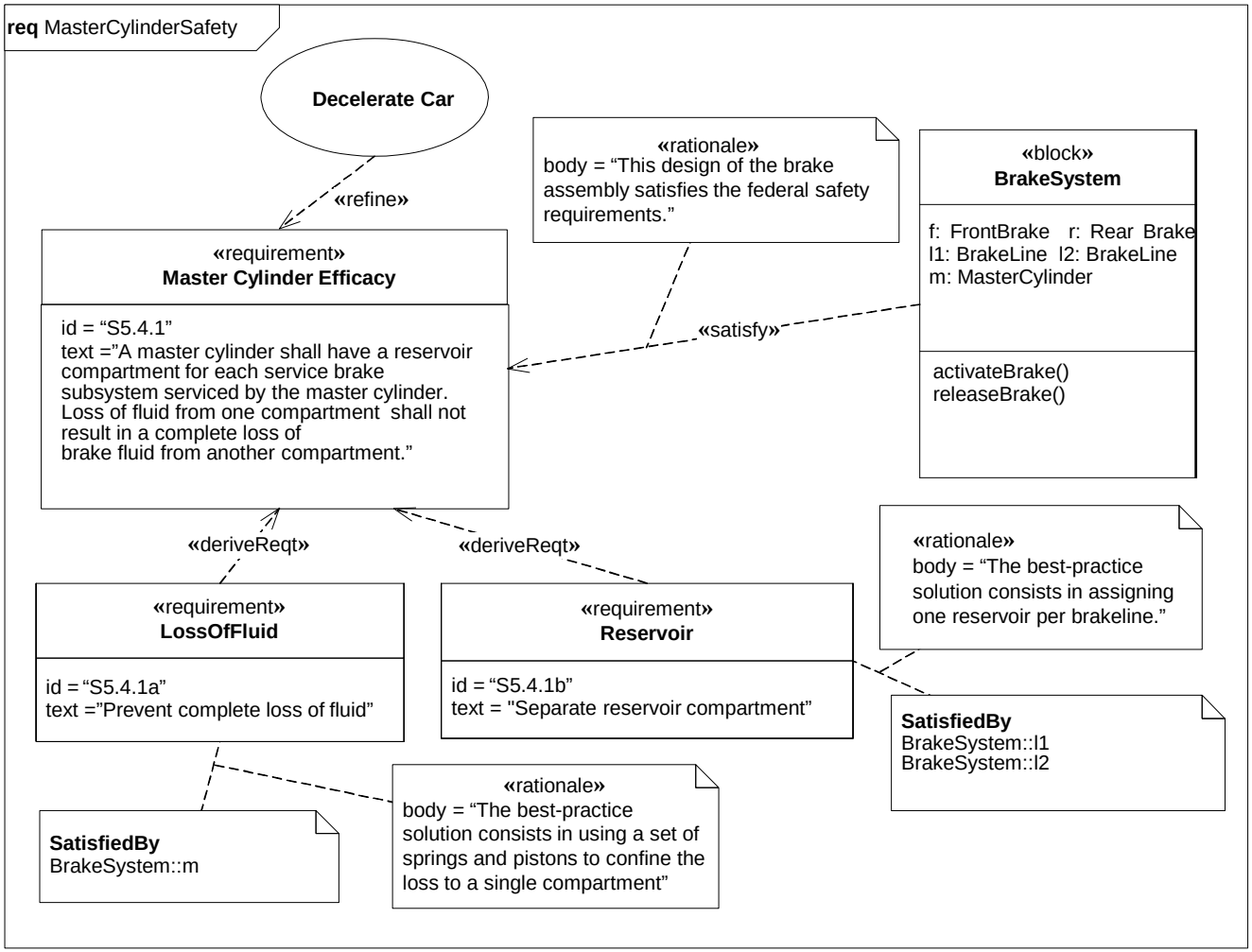
Basic Functional Requirements (Use Cases)



Basic Requirements (SysML)



Links between requirements and design (SysML Example)

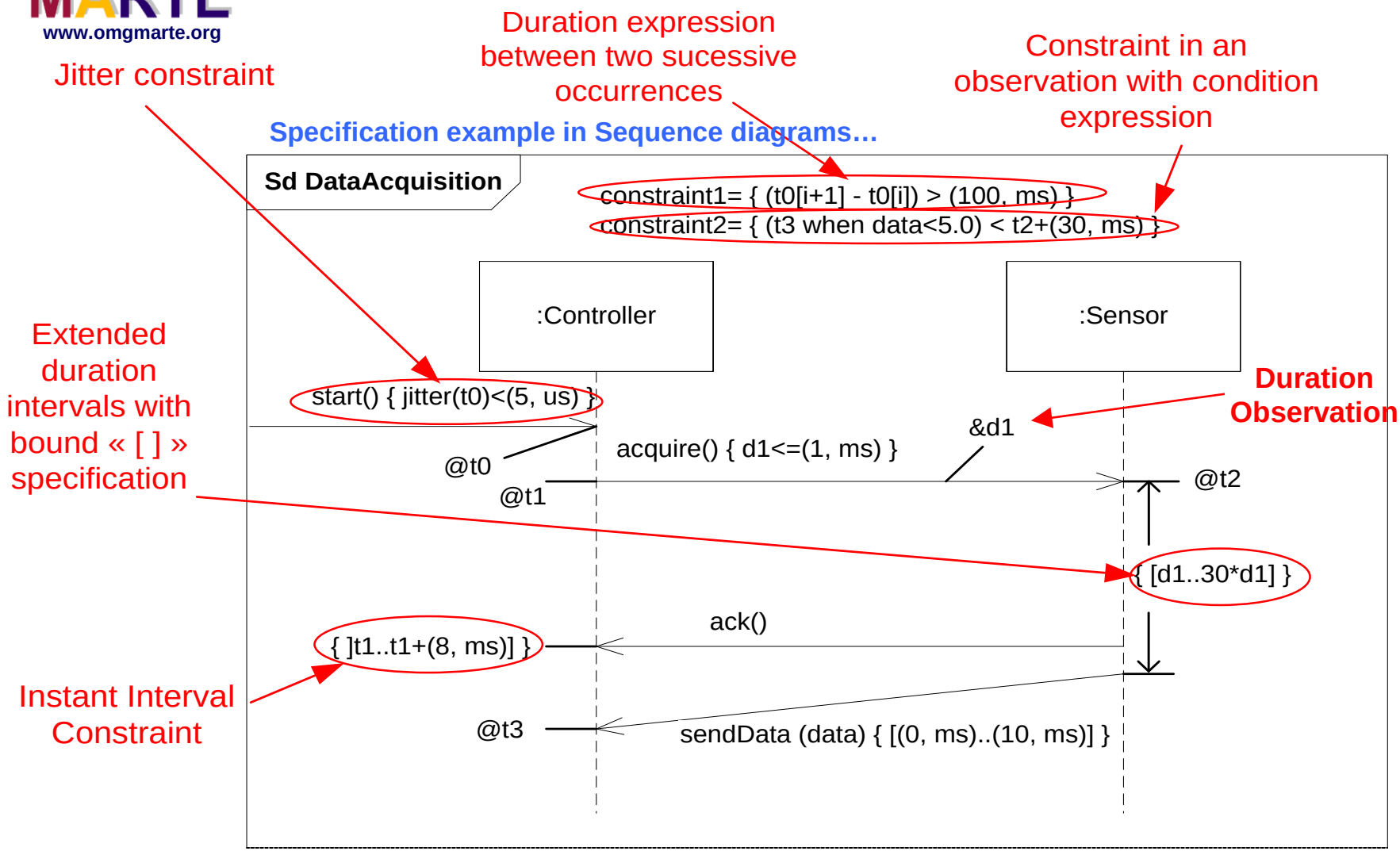


Taken from Figure 16.3 in SysML Specification – OMG Document formal/15-06-03.pdf

Modeling Non-Functional Aspects with MARTE

- **NFPs Modeling Framework:**
  - ➔ Measurements: magnitude + unit (e.g., energy, data size, duration)
  - ➔ Value Qualifiers: Value source, statistical measure, value precision,…
- **Value Specification Language (VSL):**
  - ➔ Mathematical expressions (arithmetic, logical,…)
  - ➔ Time expressions (deadlines, periods, trigger conditions,…)
  - ➔ Variables: placeholders for unknown parameters.
- **Why we need this level of formalization?**
  - ➔ Ability to support automated validation & verification.
  - ➔ Unambiguous understanding by stakeholders

Example: VSL Timing Constraints



Non-Functional Properties (NFP)

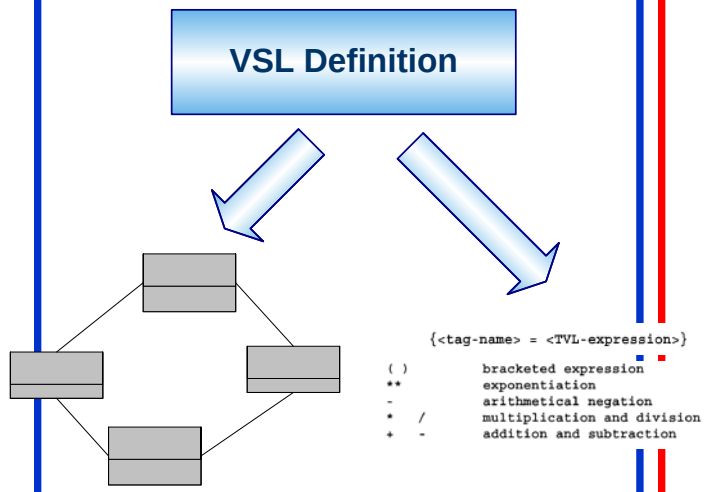
- **Formalize a number of ideas existing in SPT and QoS&FT**
  - From the SPT profile
    - e.g. Tag Value Language (variables, math. expressions) and time-related values
  - From the QoS&FT profile
    - e.g. Property Qualifiers
- **Add new modeling constructs required for MARTE**
  - e.g. tuple and choice values, time expressions and unit measurements conversion
- **NFP modeling required general extensions to UML tools**
  - e.g. value expressions editing and data type checking

  ➔ This is a key feature in DRES modeling that UML lacks

Organization of NFP constructs

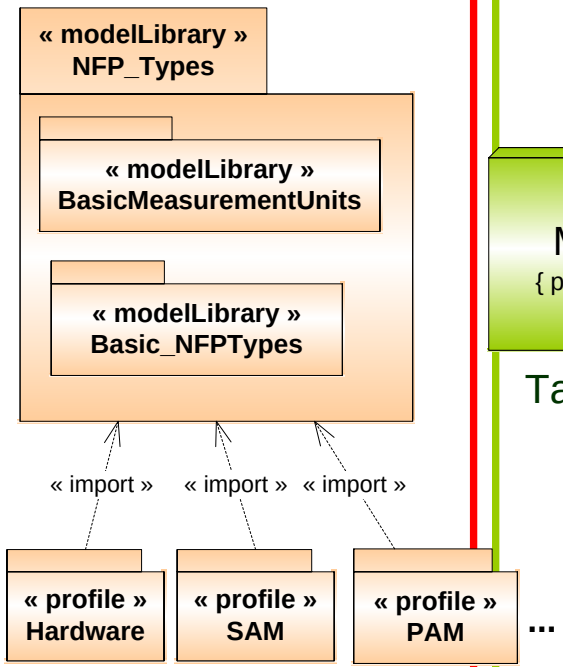
UML MARTE
www.omgmarte.org

1) Value Spec. Lang. (VSL)



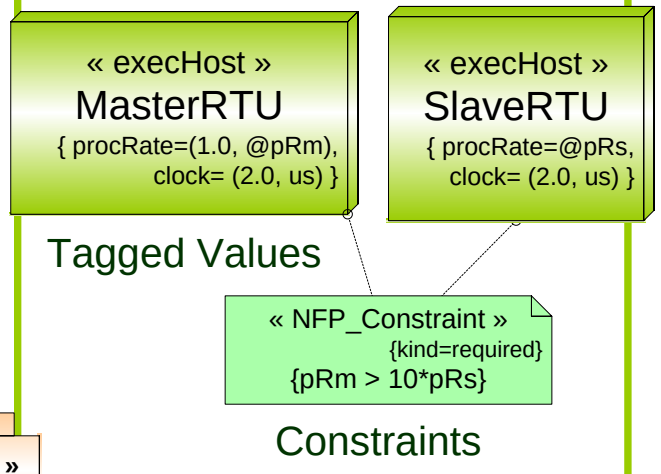
- Exact notation for values: extended Literals, Intervals, Tuples, Choices, Variables, Complex and Time Expressions

2) Declaration of NFPs



- To define, qualify measures (measurement units, source, statistical nature…) and organize NFPs

3) Annotation Mechanism



1. Tagged values
2. Constraints
3. Direct specification in UML models by using NFP types library

UML MARTE
www.omgmarte.org

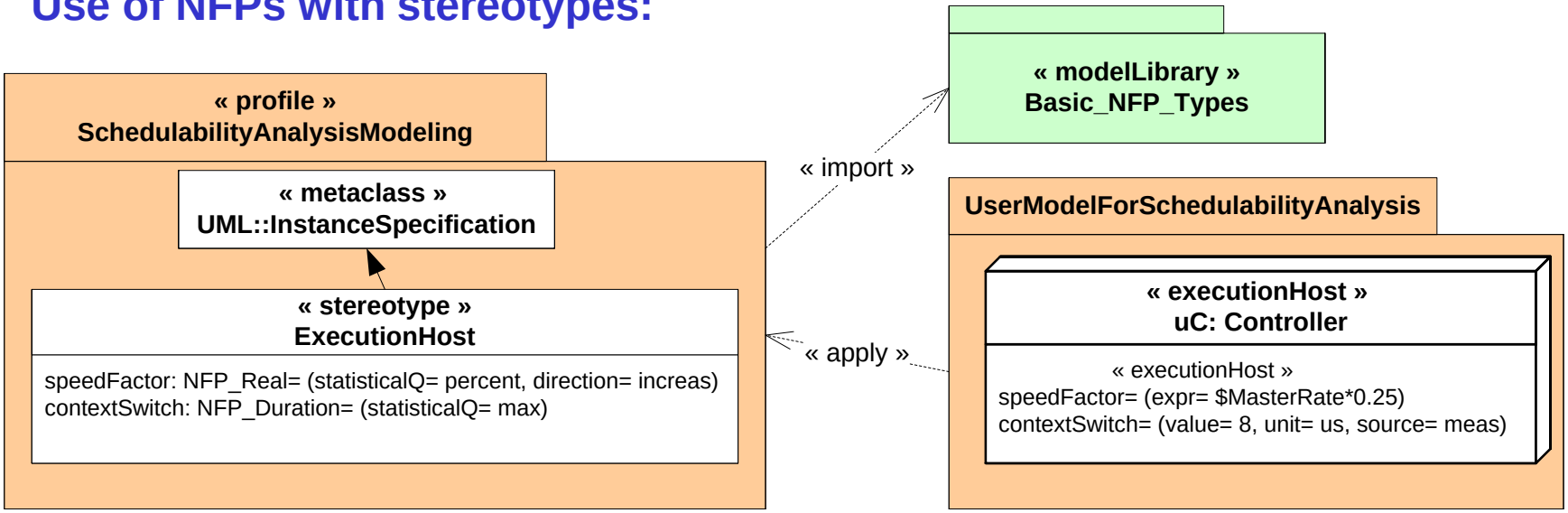
Examples of textual expressions

Value Spec.	Examples
Real Number	`1.2E-3 //scientific notation`
DateTime	`#12/01/06 12:00:00# //calendar date time`
Collection	`{1, 2, 88, 5, 2} //sequence, bag, ordered set..` `{{1,2,3}, {3,2}} //collection of collections`
Tuple and choice	`(value=2.0, unit= ms) //duration tuple value` `periodic(period=2.0, jitter=3.3) //arrival pattern`
Interval	`[1..251[ //upper closed interval between integers` `[@A1..@A2] //interval between variables`
Variable declaration & Call	`io@var1 //input/output variable declaration` `var1 //variable call expression.`
Arithmetic Operation Call	`+(5.0,var1) //"add" operation on Real datatypes` `5.0+var1 //infix operator notation`
Conditional Expression	`(($var1<6.0)?(10^6):1) //if true return 10 exp 6,else 1`

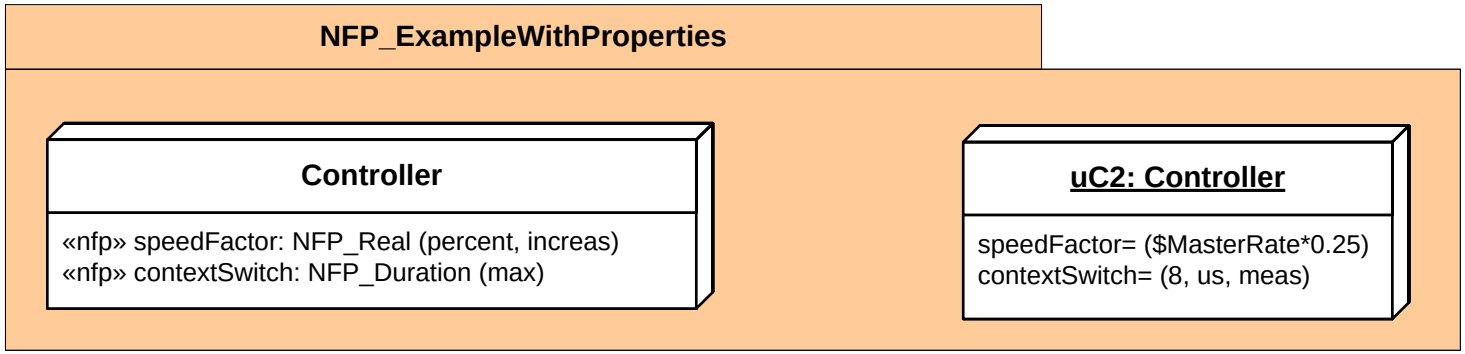
+ additional constructs to reference UML properties and time observations

Examples of NFP annotations

Use of NFPs with stereotypes:



Use of NFPs as M1 level properties:



UML MARTE
www.omgmarte.org

Time modeling

- **The Time model introduced in MARTE completes the features provided by the SimpleTime package of UML 2**

- **Basic ideas**
  - Any element related to time may explicitly refer to a clock
  - Time is multiform (not limited to “physical” time)
  - Support distribution, clock uncertainties
  - Design vs. Runtime clocks

- **What are the domain concepts?**
  - Events → TimedEvent
  - Behaviors and Actions → TimedProcessing
  - Constraints → TimedConstraint
  - Observations → TimedObservation

Time modeling (cont'd)

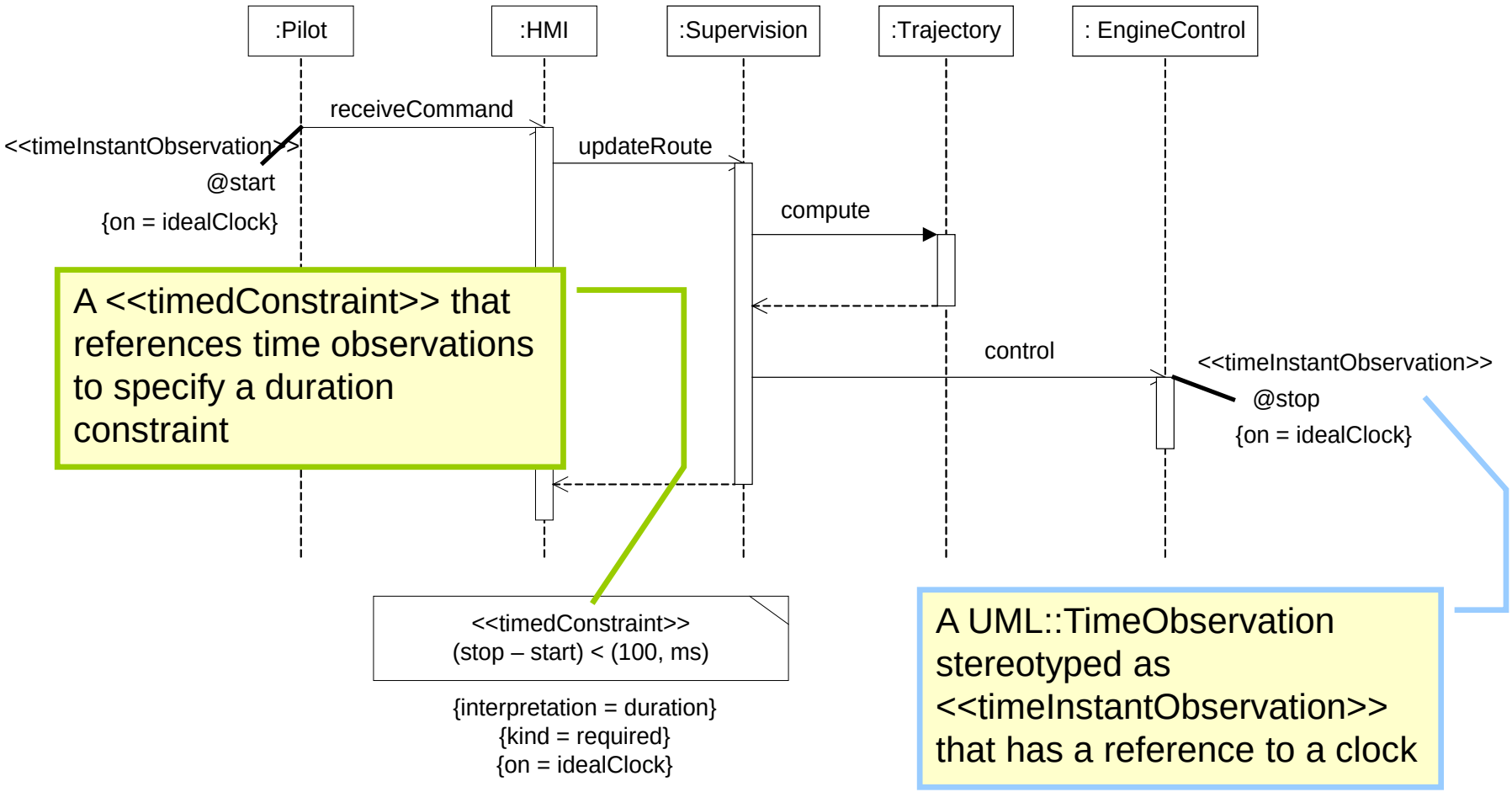
- **Time Structure**
  - Made of several clocks
- **Clock**
  - A totally ordered set of instants
  - Access to instant value and duration with units
- **Relations on Clocks**
  - Expression → ClockConstraint
  - Reflect causality (from algorithm and allocations)

nature / isLogical	discrete	dense
true	Logical clock	Not used
false	Chronometric clock	
	discrete	dense

Stereotype properties:
Special semantics

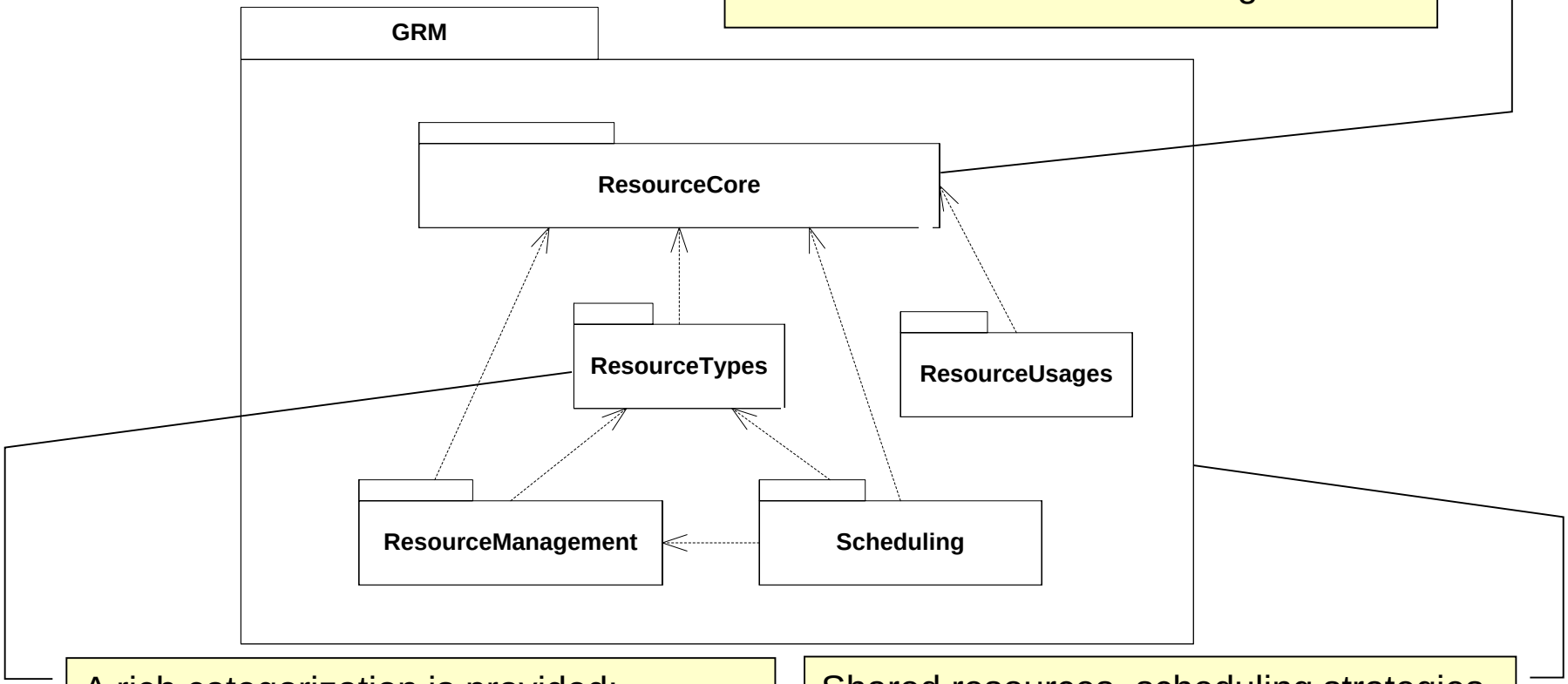
+ optional
- set of properties
- set of operations

Example of a time constraint



General Resource Modeling

Resource offers Services and may have NFPs for its definition and usage

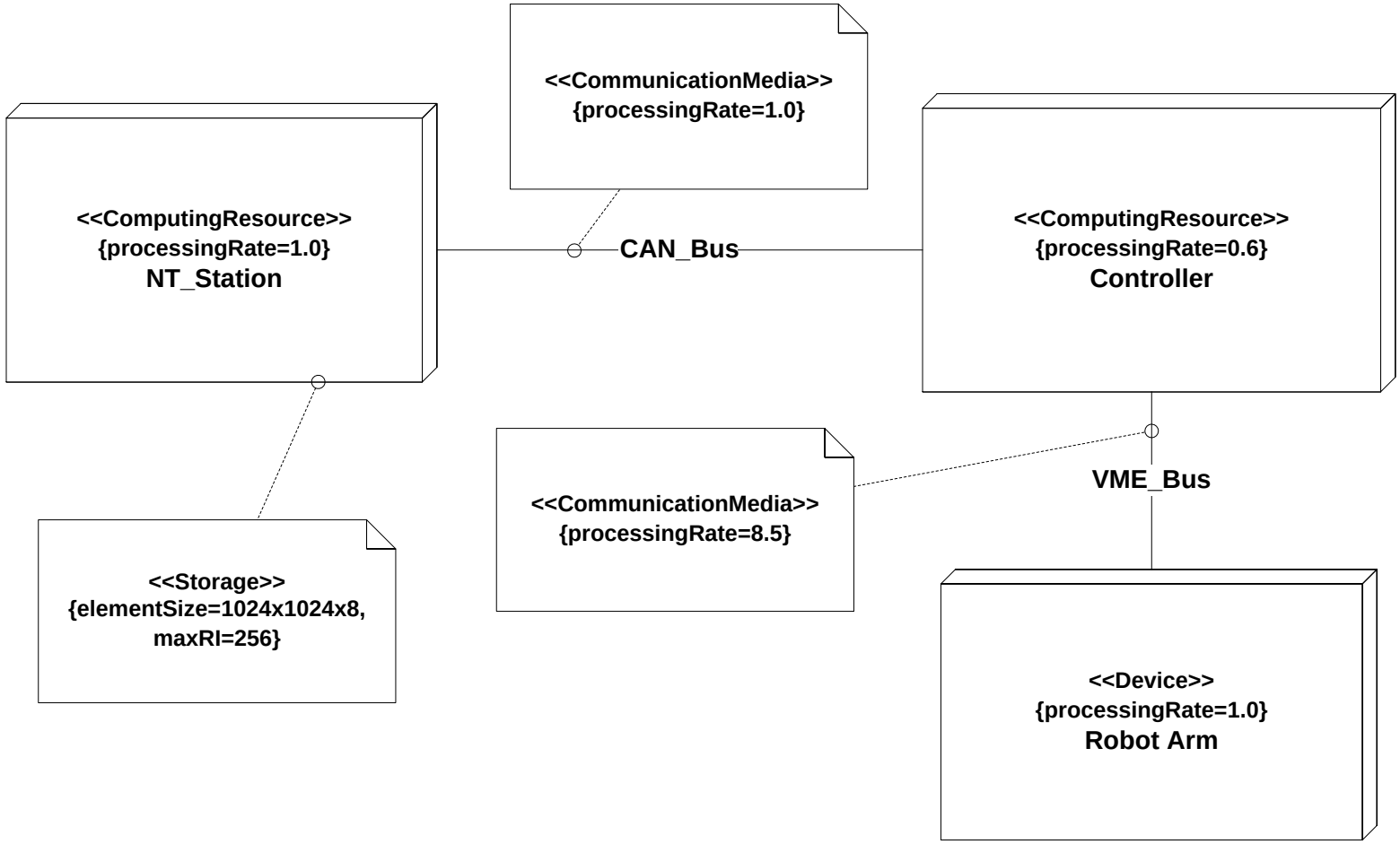


A rich categorization is provided: Storage, Synchronization, Concurrency, Communication, Timing, Computing, and Device Resources may be defined.

Shared resources, scheduling strategies and specific usages of resources (like memory consumption, computing time and energy) may be annotated.

UML MARTE
www.omgmarte.org

Example of resource modeling

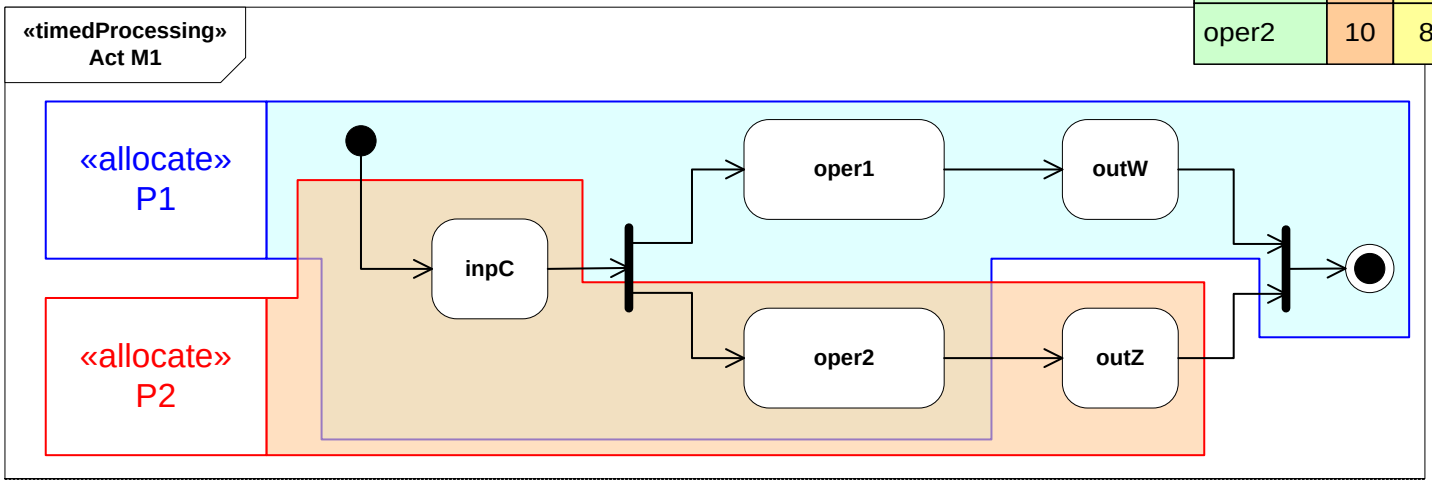


Allocation modeling

- **Basic ideas**
  - Allocate an application element to an execution platform element
  - Refine a general element into specific elements
  - Inspired by the SysML allocation
    - Can only allocate application to execution platform
    - Can attach NFP constraints to the allocation

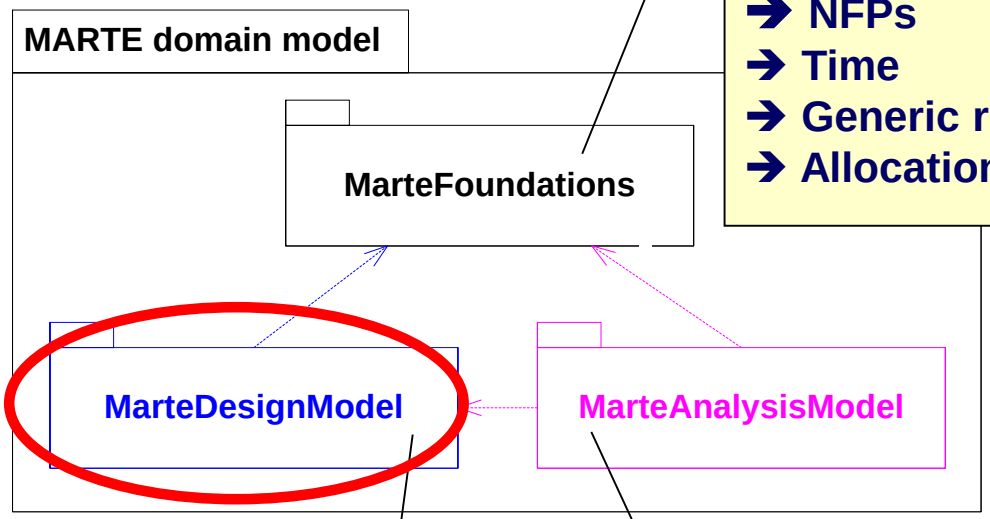
	P1	P2	Unique Alloc
inpC	4	6	true
outpW	4		true
outpZ		6	true
oper1	10		true
oper2	10	8	true

- **Example of allocation**



UML MARTE
www.omgmarte.org

Architecture of the MARTE specification



Foundations for RT/E systems modeling and analysis:
- ➔ CoreElements
- ➔ NFPs
- ➔ Time
- ➔ Generic resource modeling
- ➔ Allocation

Specialization of MARTE foundations for modeling purpose (specification, design…):
- ➔ Generic component model
- ➔ High-level application modeling
- ➔ Software resource modeling
- ➔ Hardware resource modeling

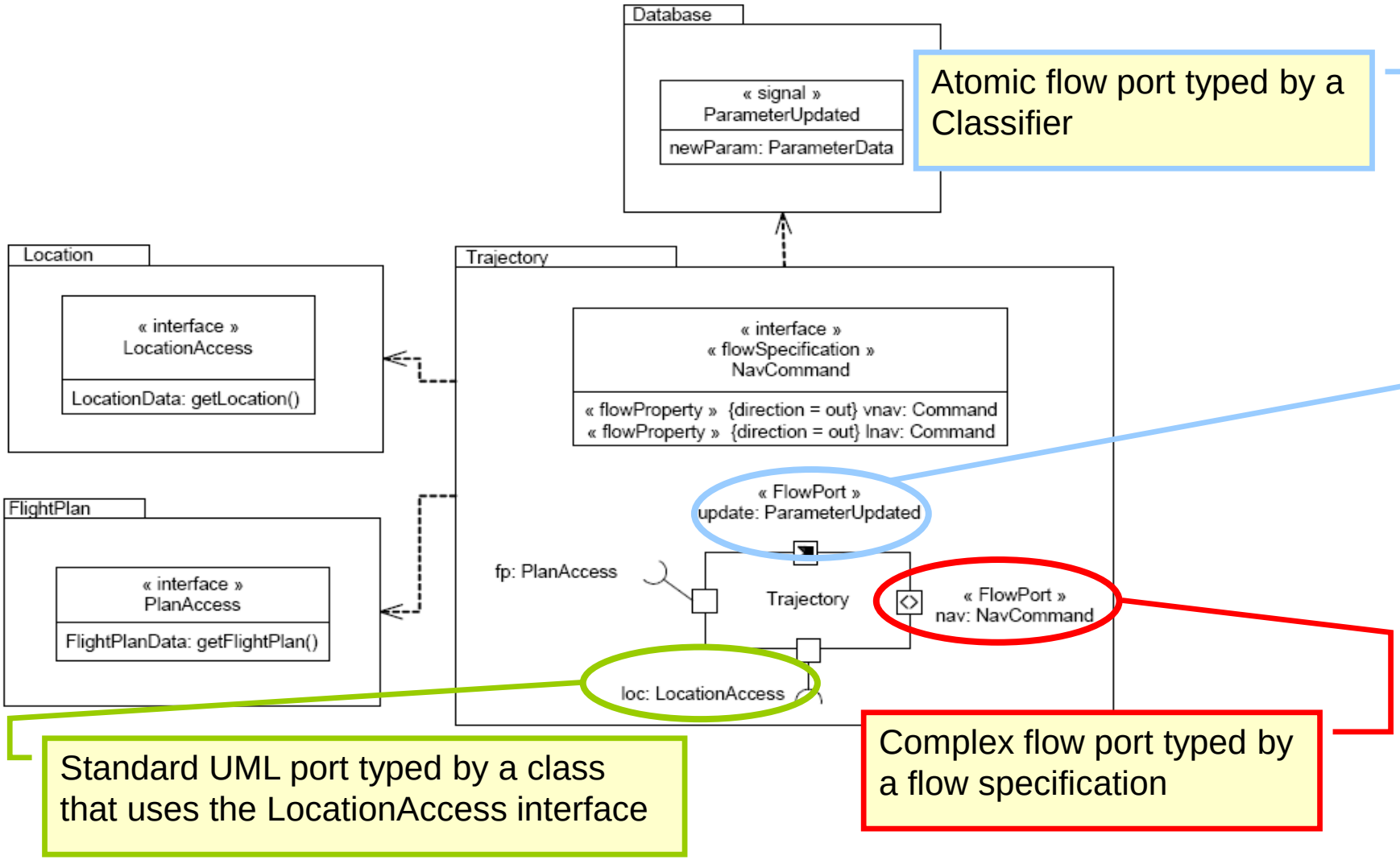
Specialization of foundations for annotating model for analysis purpose:
- ➔ Generic quantitative analysis
- ➔ Schedulability analysis
- ➔ Performance analysis

General Component Model

- **Introduced to cope with various component-based models**
  - UML2, SysML, Spirit, AADL, Lightweight-CCM, EAST-ADL2, Autosar…
- **Relies mainly on UML structured classes, on top of which a support for SysML blocks has been added**
  - Atomic and non-atomic flow ports
  - Flow properties and flow specifications
- **But also providing a support for Lightweight-CCM, AADL and EAST-ADL2, Spirit and Autosar**

UML MARTE
www.omgmarte.org

Example of component definition



UML MARTE
www.omgmarte.org

High-level application modeling

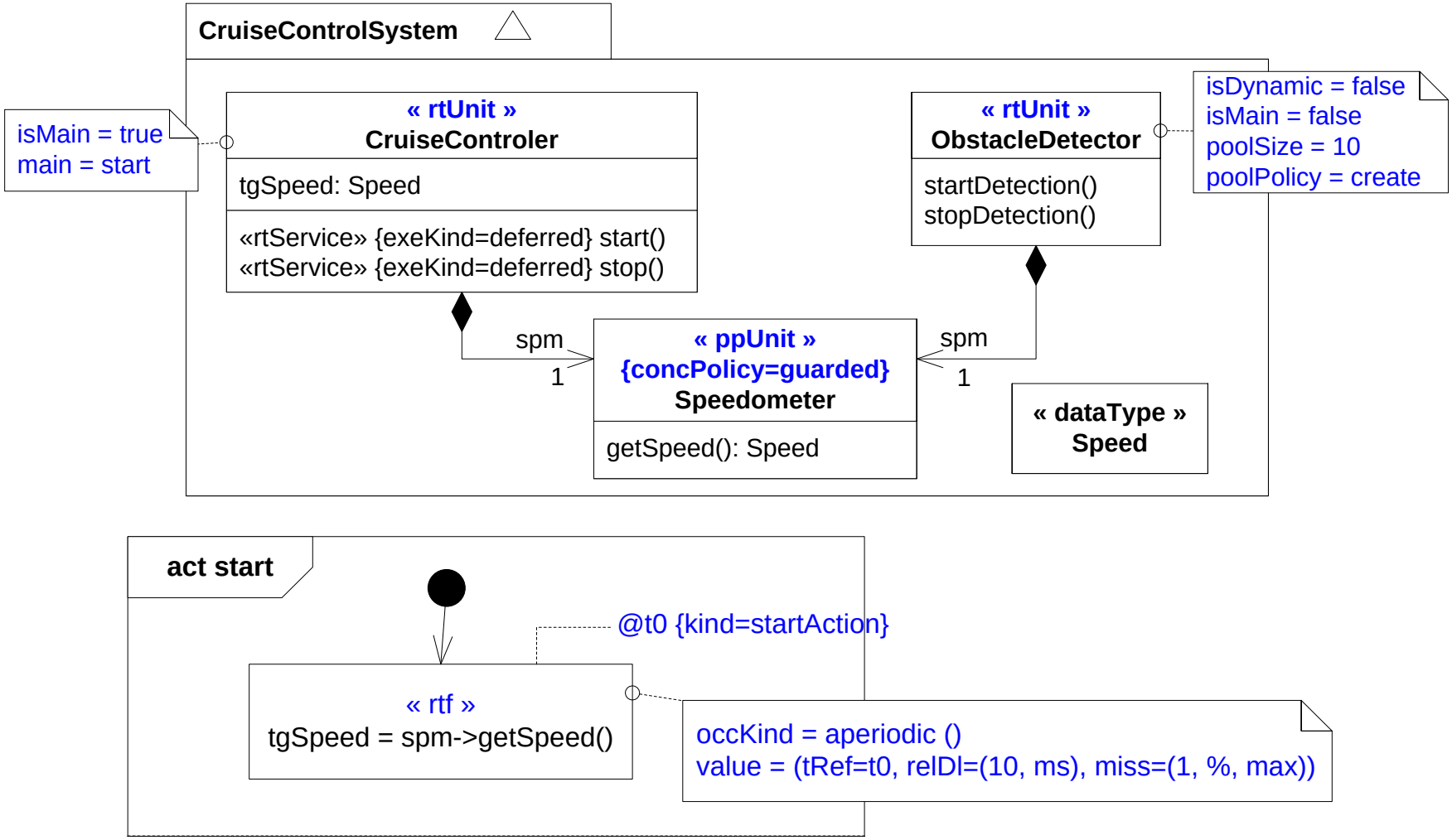
- **Provides high-level concepts for modeling qualitative real-time features**
  - Real-Time Unit (RTUnit)
    - Generalization of the Active Objects of the UML 2
    - Owns at last one schedulable resource
    - Resources are managed either statically (pool) or dynamically
    - May have operational mode description (similar to AADL concept)
  - Protected Passive Unit (PPUnit)
    - Generalization of the Passive Objects of the UML2
    - Supports different concurrency policies (e.g. sequential, guarded)
    - Policy are specified either locally or globally
    - Execution is either immediateRemote or deferred

UML MARTE
www.omgmarte.org

High-level application modeling (cont'd)

- **Provides high-level concepts for modeling quantitative real-time features**
  - Real-Time Behavior (RtBehavior)
    - Message Queue size and policy bound to a provided behavior
  - Real-Time Feature (RTF)
    - Extends UML Action, Message, Signal, BehavioralFeature
    - Relative/absolute/bound deadlines, ready time and miss ratio
  - Real-Time Connector (RteConnector)
    - Extends UML Connector
    - Throughput, transmission mode and max blocking/packet Tx time

Usage examples of the HLAM extensions



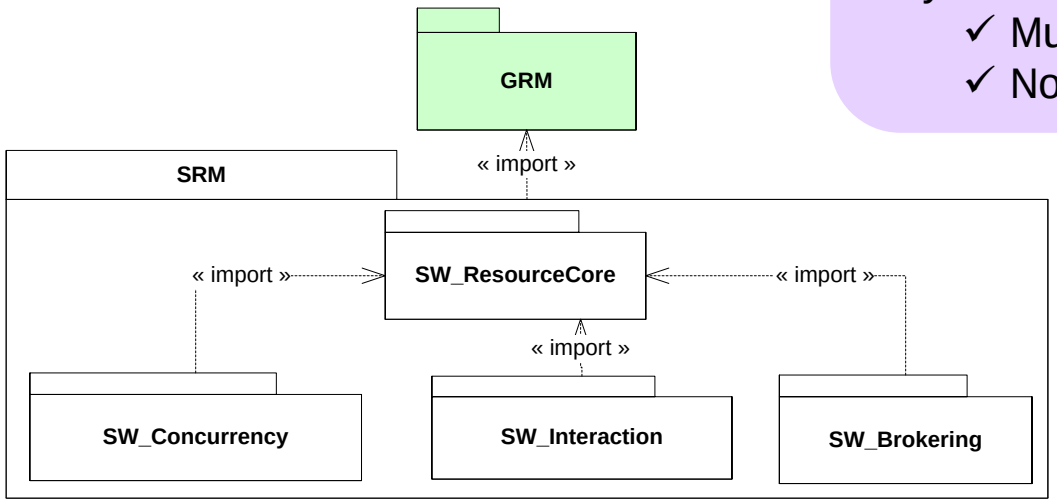
Outline of the Software Resource Model

Concurrent execution contexts:

- Schedulable Resource (Task)
- Memory Partition (Process)
- Interrupt Resource
- Alarm

Interactions between concurrent contexts:

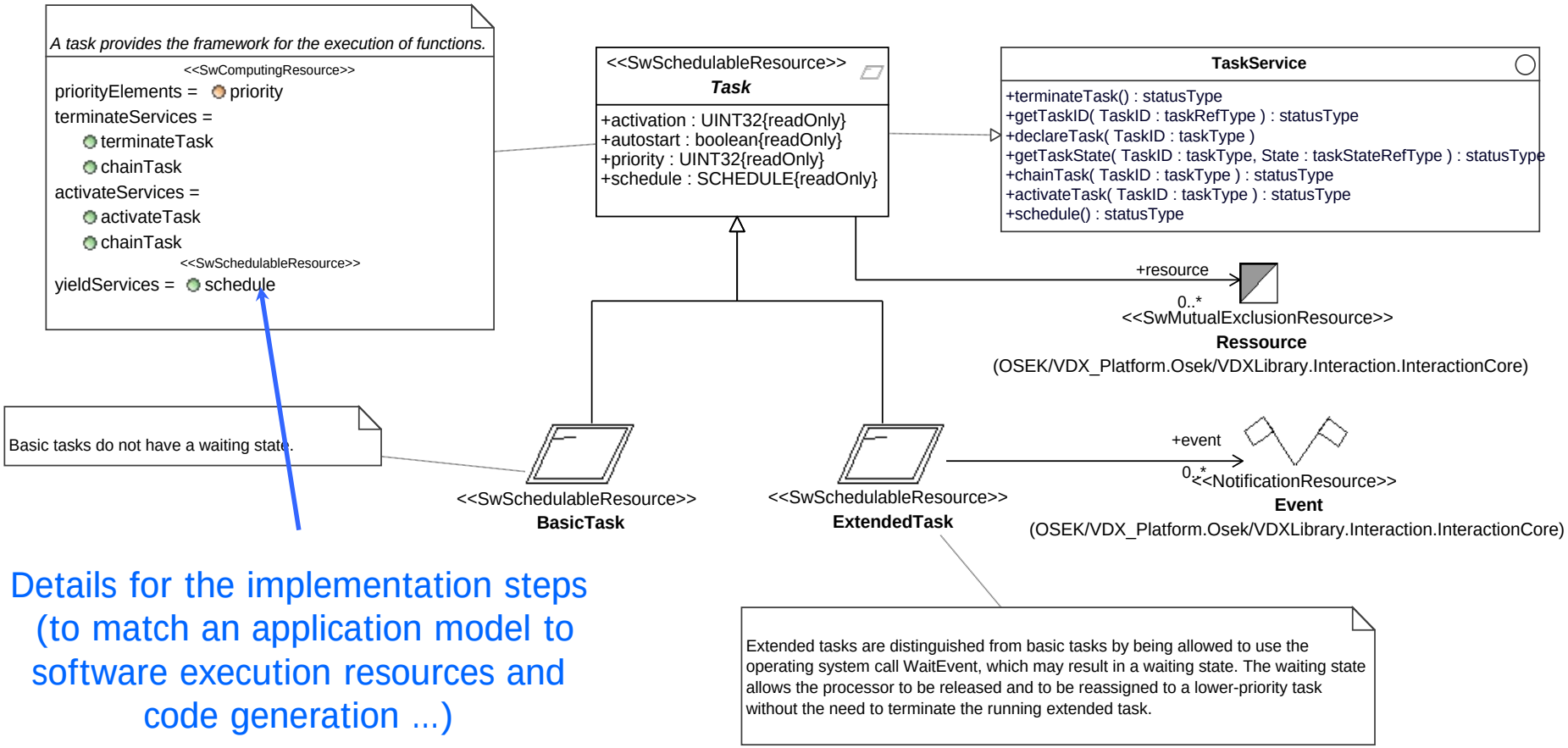
- Communication (Data exchange)
  - ✓ Shared data
  - ✓ Message (Message queue)
- Synchronization
  - ✓ Mutual Exclusion (Semaphore)
  - ✓ Notification (Event mechanism)



Hardware and software resources brokering:

- Drivers
- Memory management

OSEK/VDX modeled with SRM

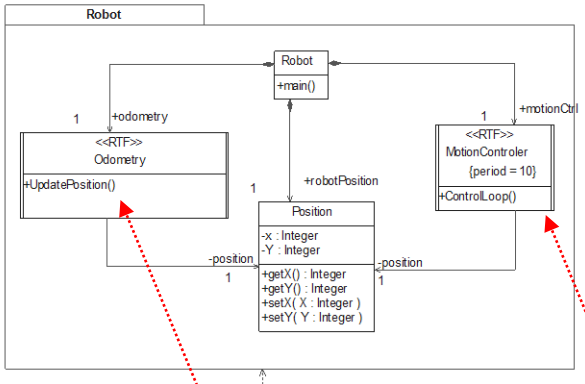


Details for the implementation steps (to match an application model to software execution resources and code generation ...)

SRM Usage example

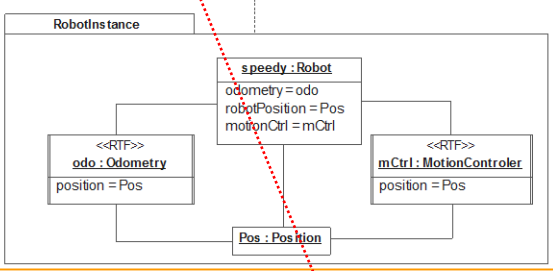
UML MARTE
www.omgmarte.org

Application Model



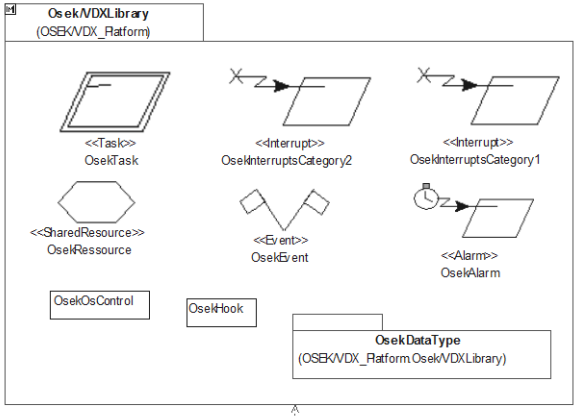
« entryPoint »

<<import>>

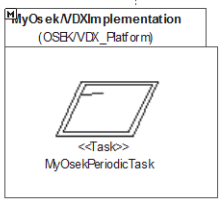


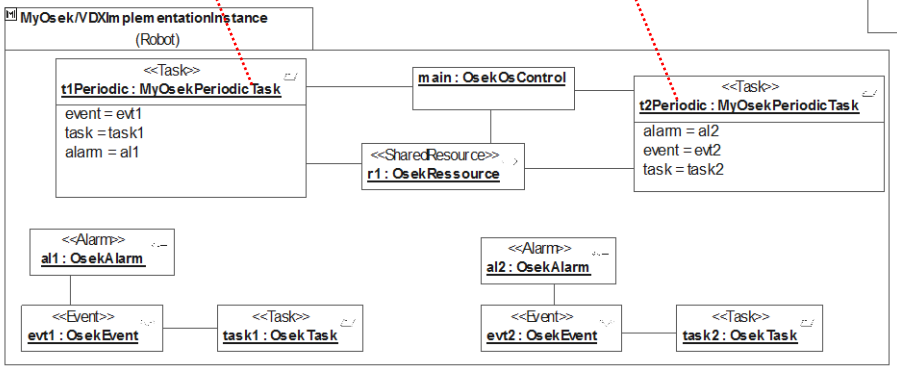
« entryPoint »

Execution Resources Model



<<import>>



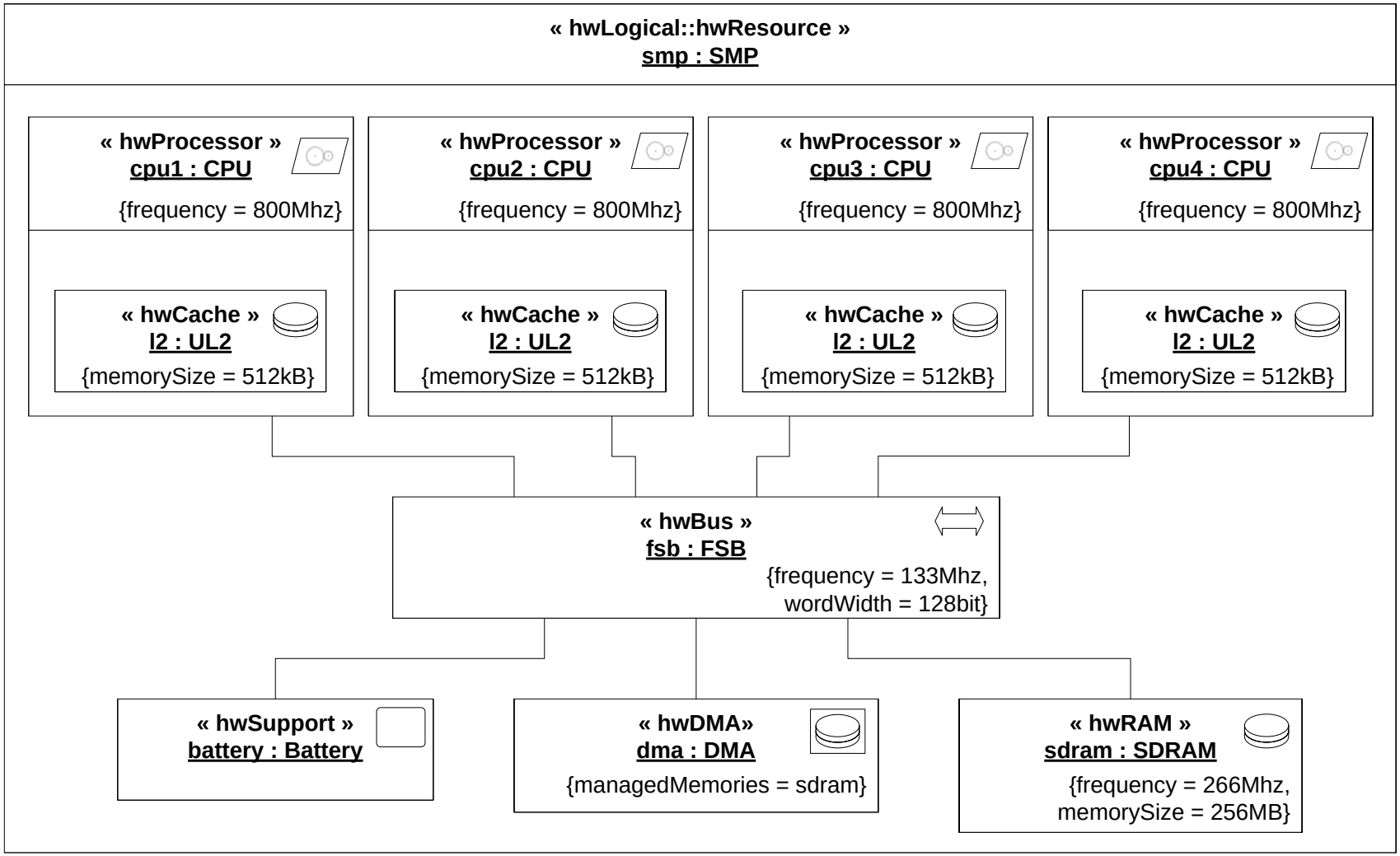


UML MARTE
www.omgmarte.org

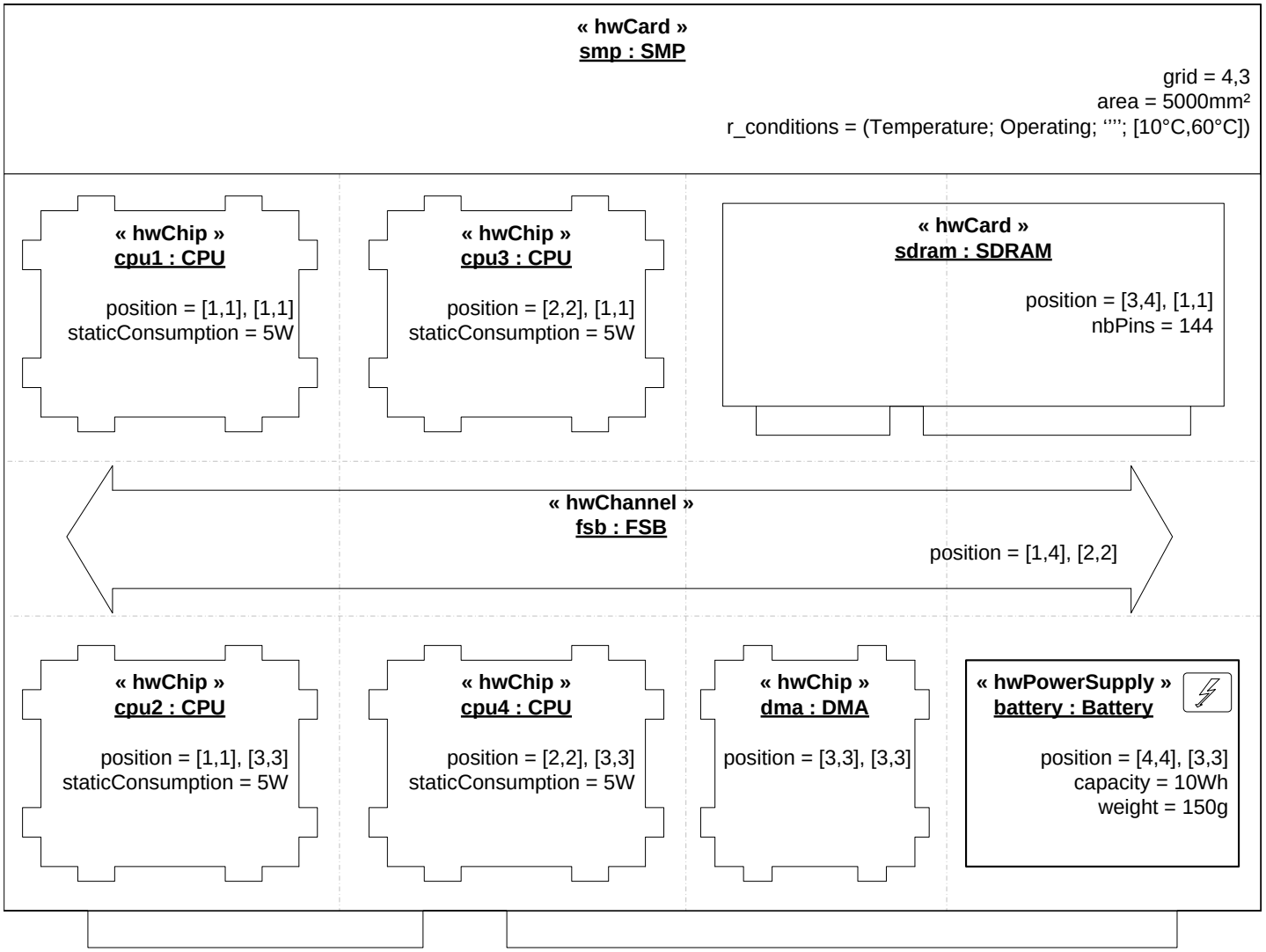
Hardware Resource Model

- **Logical view (functional modeling)**
  - Provides a description of functional properties
  - Based on a functional classification of hardware resources:
    - HwComputing resources
    - HwStorage resources
    - HwCommunication resources
    - HwTiming resources
    - HwDevice resources
- **Physical view**
  - Provides a description of physical properties
  - Based on both following packages:
    - HwLayout
    - HwPower

HRM usage example: Logical View

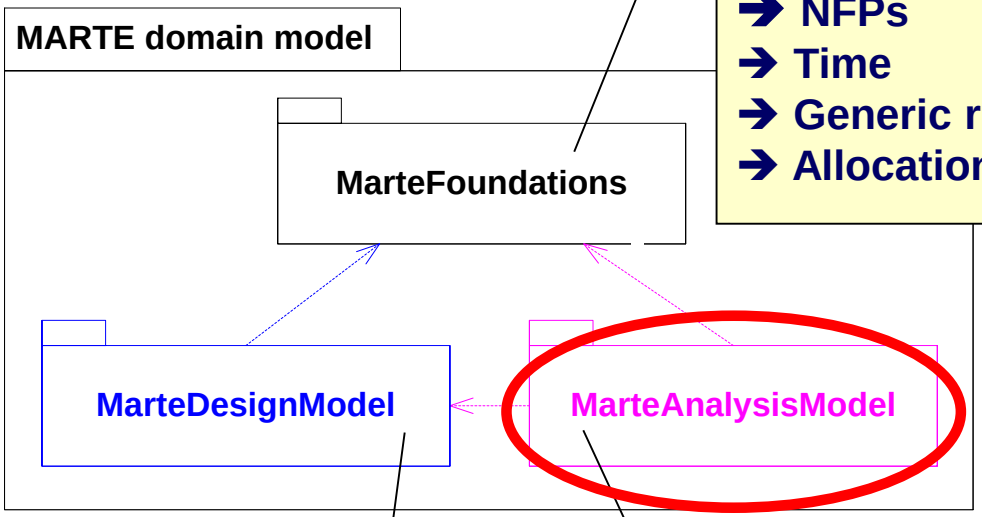


HRM usage example: Physical View



Analysis with MARTE

UML MARTE
www.omgmarte.org



Foundations for RT/E systems modeling and analysis:
- ➔ CoreElements
- ➔ NFPs
- ➔ Time
- ➔ Generic resource modeling
- ➔ Allocation

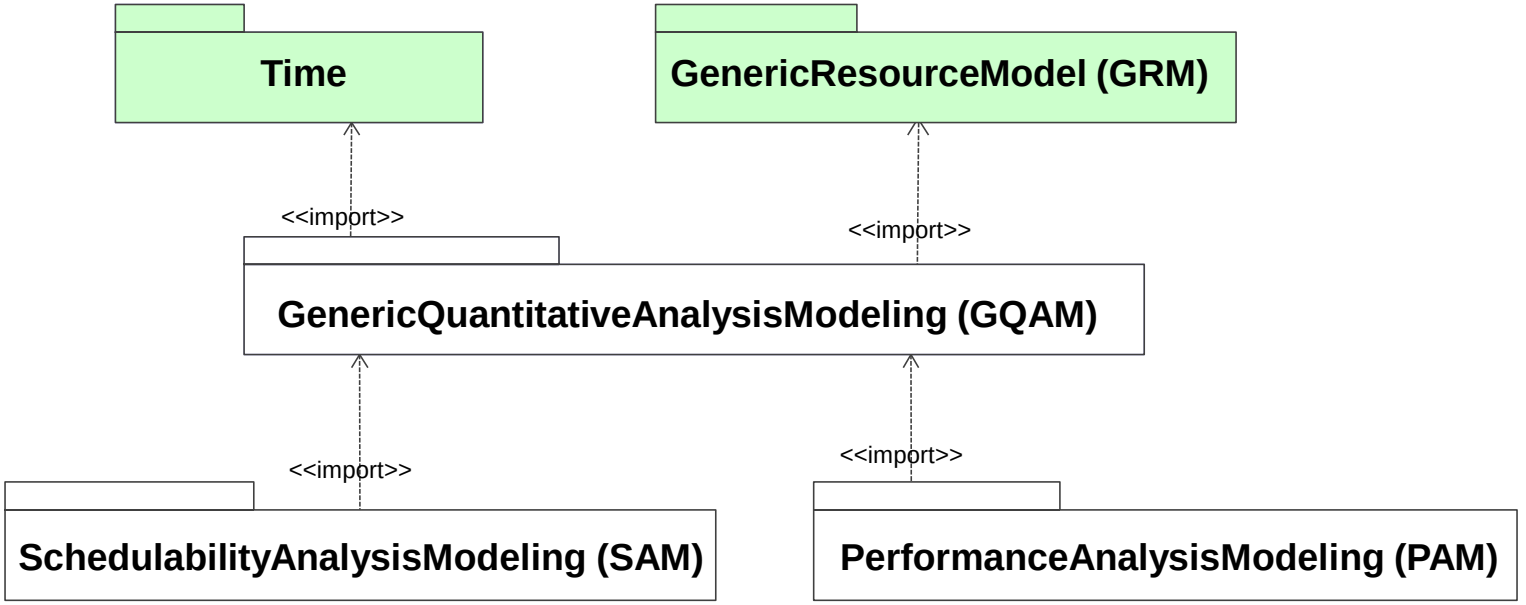
Specialization of MARTE foundations for modeling purpose (specification, design…):
- ➔ Generic component model
- ➔ High-level application modeling
- ➔ Software resource modeling
- ➔ Hardware resource modeling

Specialization of foundations for annotating model for analysis purpose:
- ➔ Generic quantitative analysis
- ➔ Schedulability analysis
- ➔ Performance analysis

General Quantitative Analysis Model

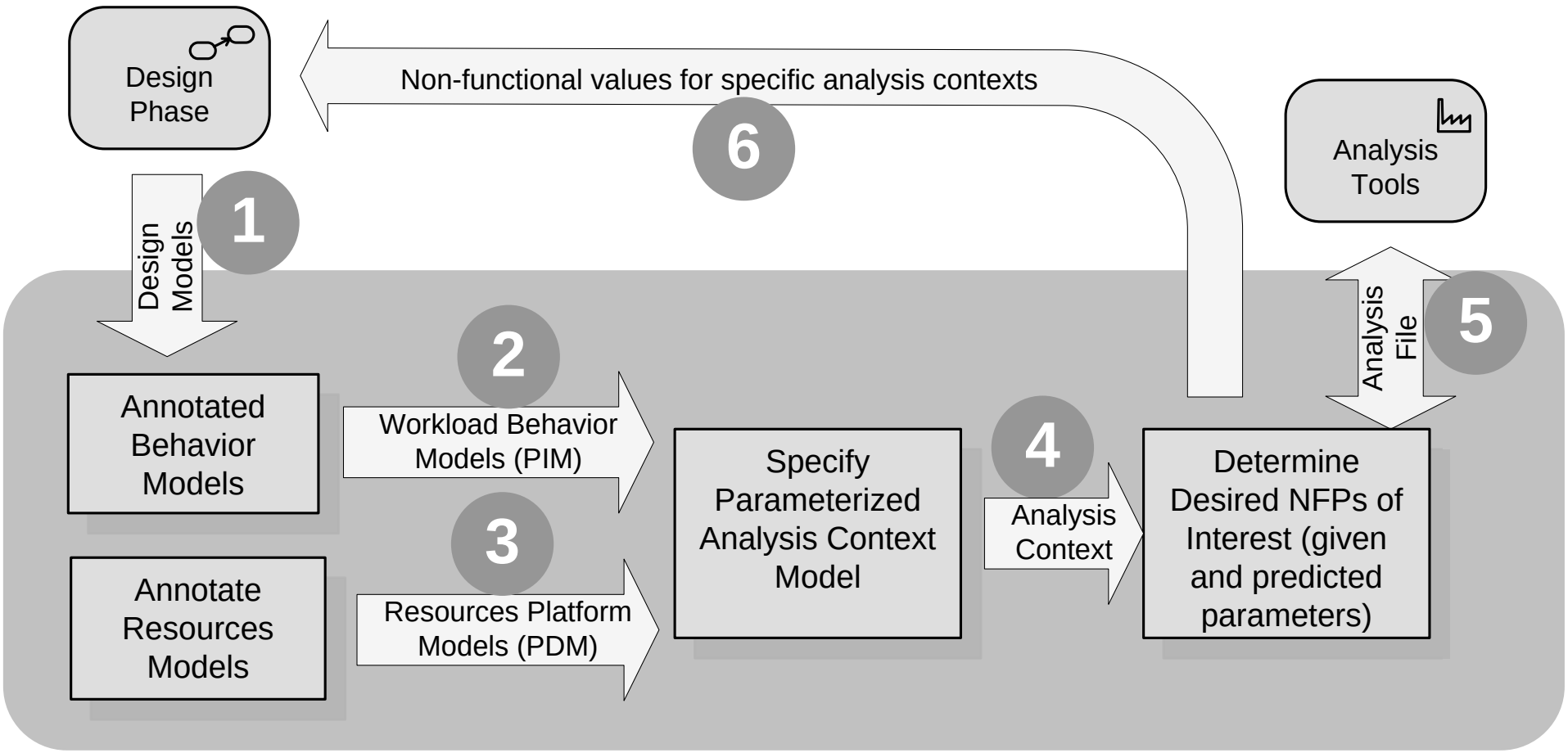
- **GQAM updates SPT**
  - Alignment on UML2
  - Harmonization between two SPT subprofiles: sched. and perf.
  - Extension of timing annotations expressiveness
    - Overheads (e.g. messages passing)
    - Response times (e.g. BCET & ACET)
    - Timing requirements (e.g. miss ratios and max. jitters)
- **Main concepts common for quantitative analysis**
  - Resources
  - Behavior
  - Workload
  - All embedded in an analysis context (may have analysis parameters)

Dependencies and architecture of GQAM



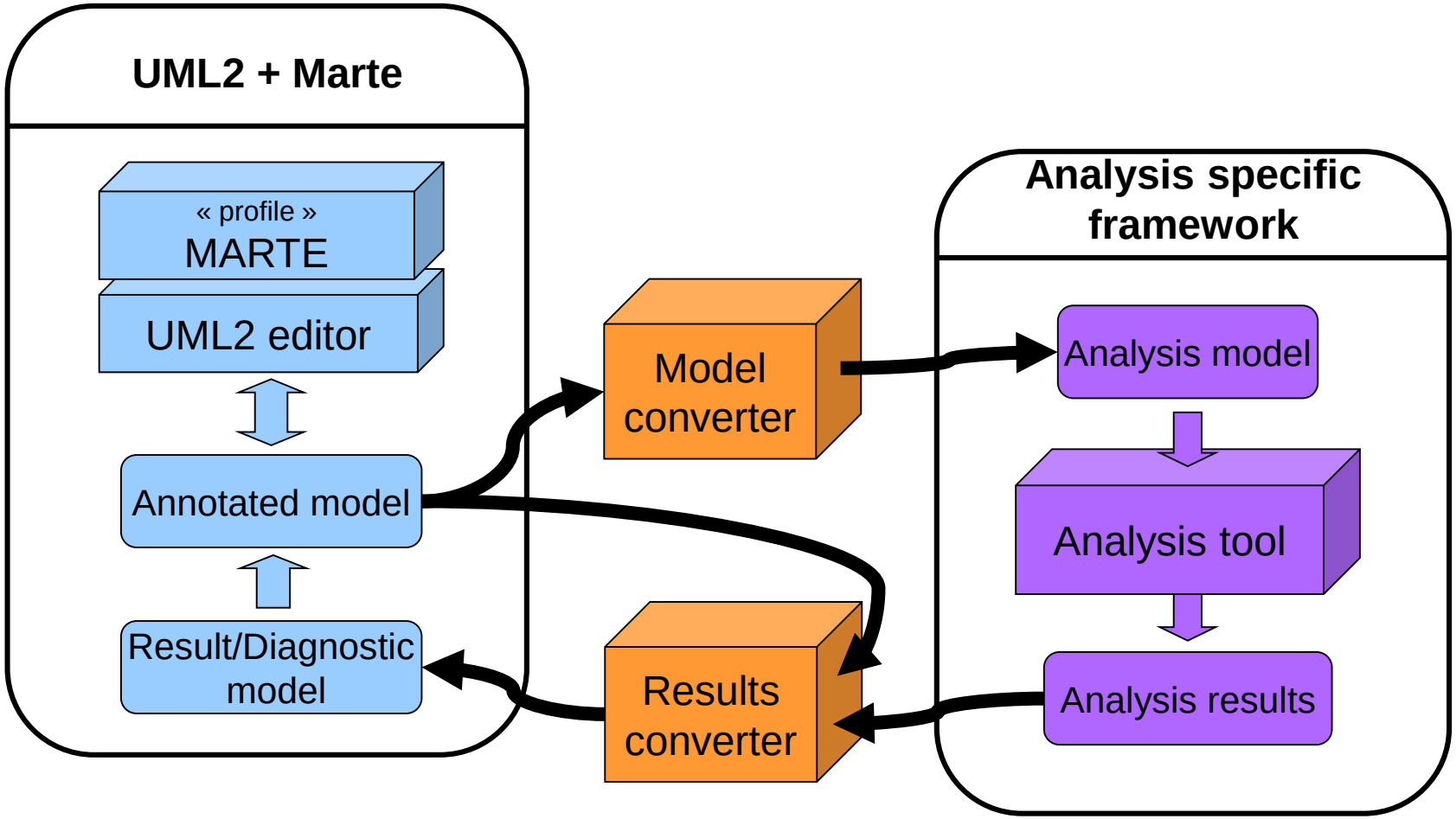
- **GQAM**
  - Common concepts to be used by analysis sub-profiles
- **SAM**
  - Modeling support for schedulability analysis techniques.
- **PAM**
  - Modeling support for performance analysis techniques.

From design to analysis



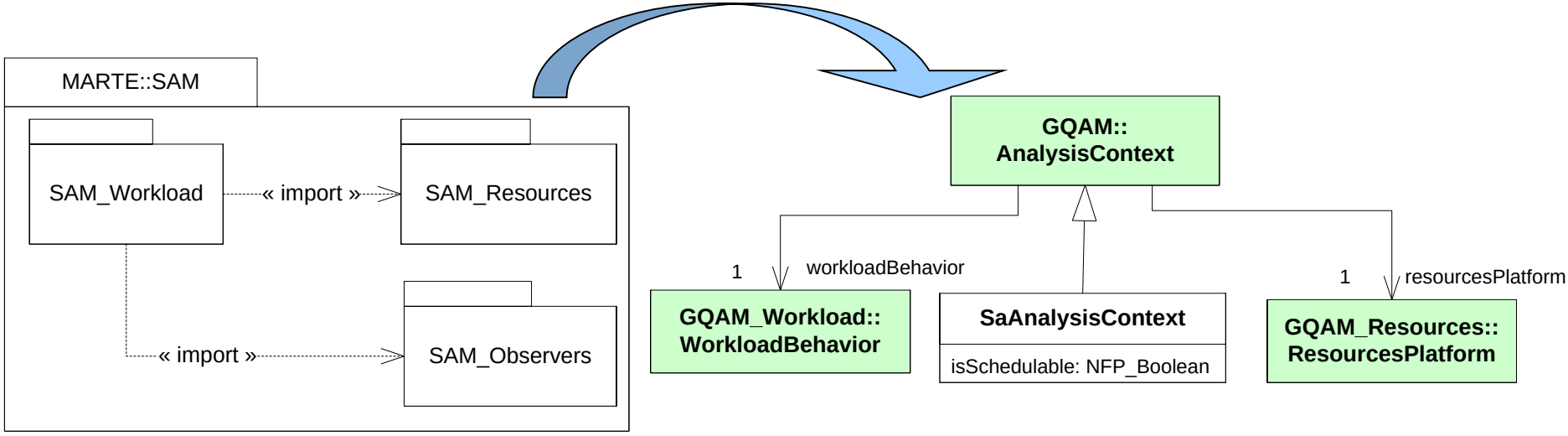
UML MARTE
www.omgmarte.org

Processing schema for model-based analysis



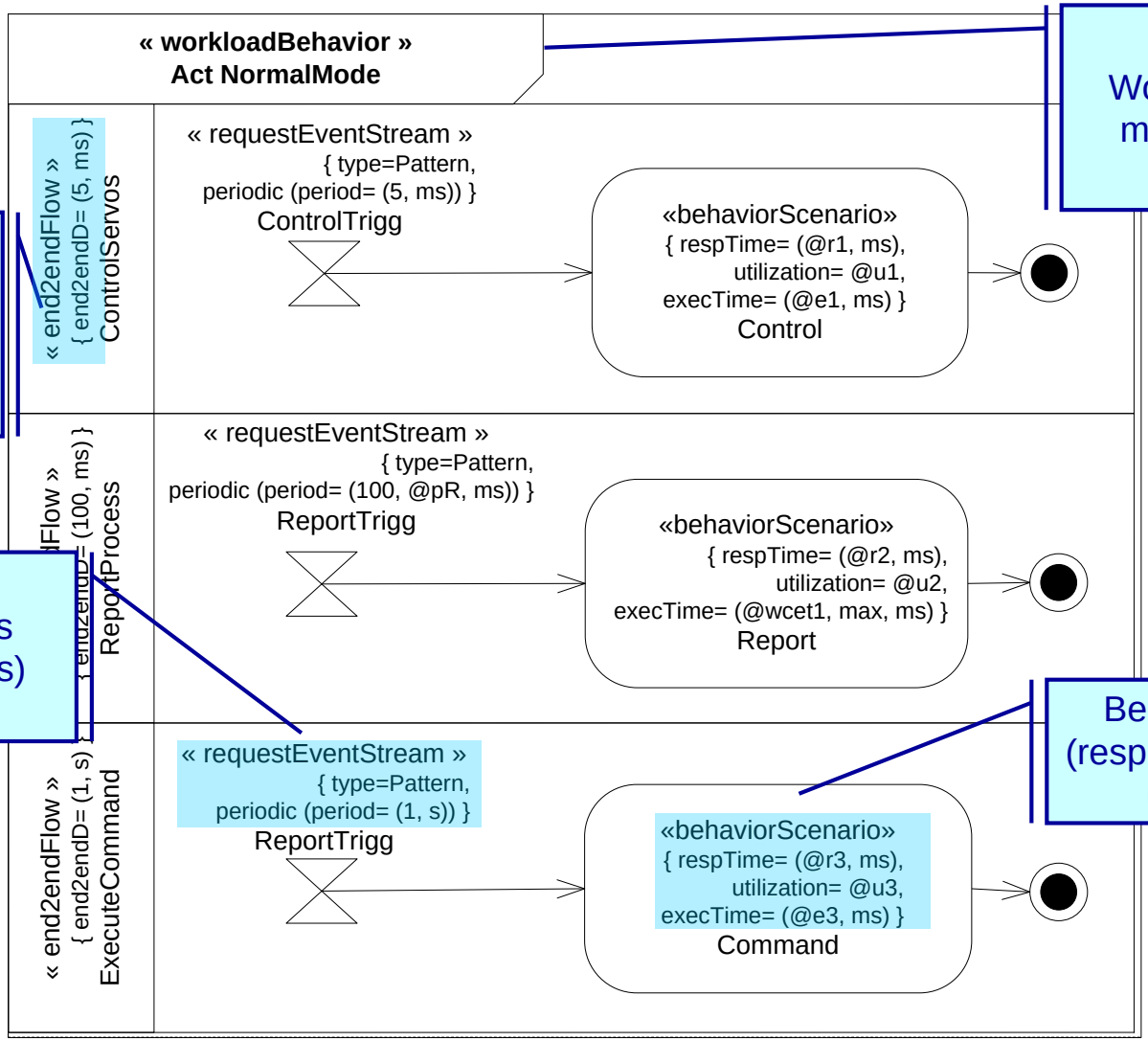
Schedulability Analysis Model

- **Modeling for analysis techniques taking into account scheduling aspects**
- **Provides high-level analysis constructs**
  - Sensitivity analysis, parametric analysis
  - Observers for time constraints and time predictions at analysis context level
- **Supports most common sched. analysis techniques**
  - RMA-based, holistic techniques and modular techniques



Workload Modeling Example

UML MARTE
www.omgmarte.org



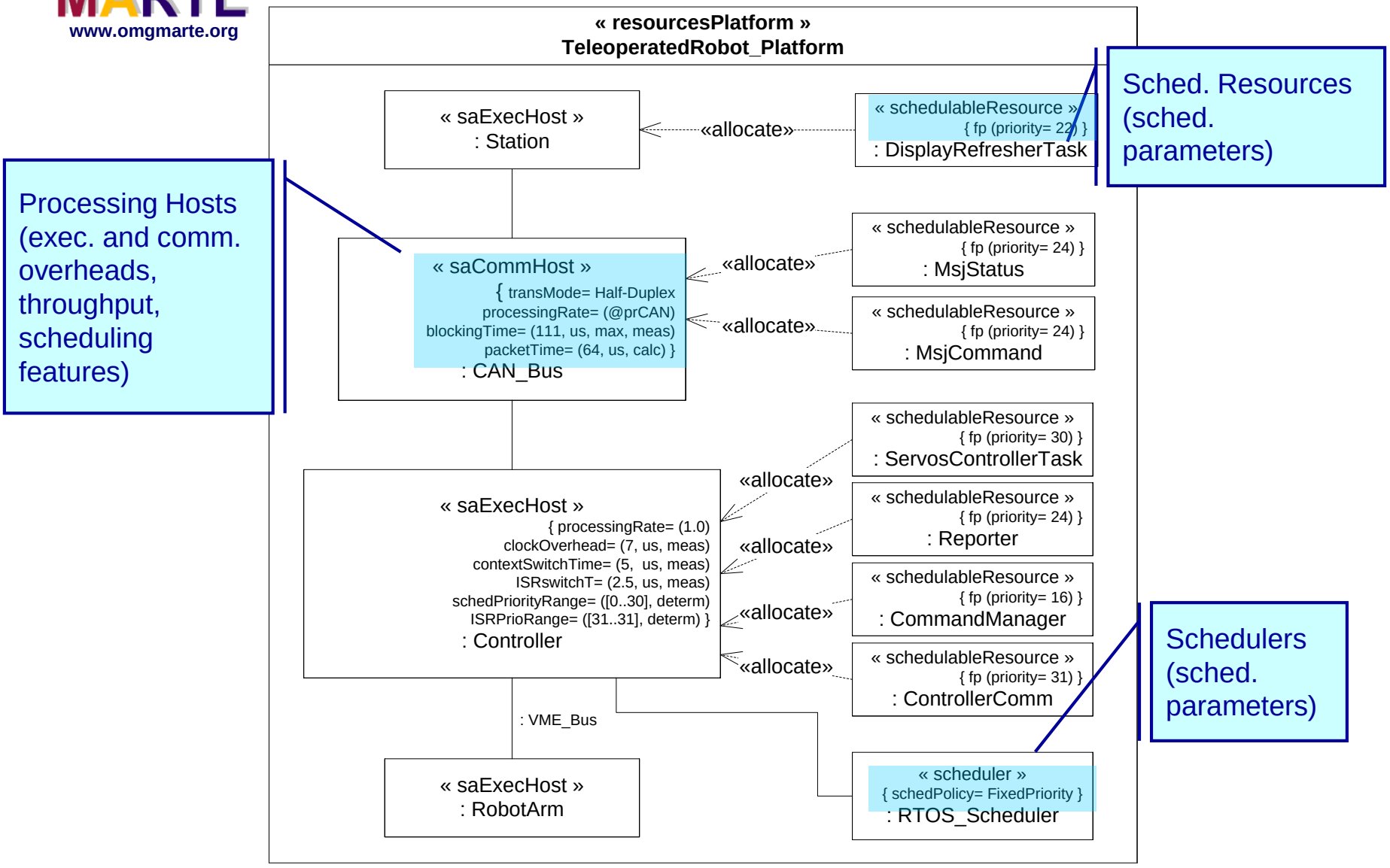
Workload Behavior maps to Behavior

EndToEndFlow (end2end deadlines and predicted times)

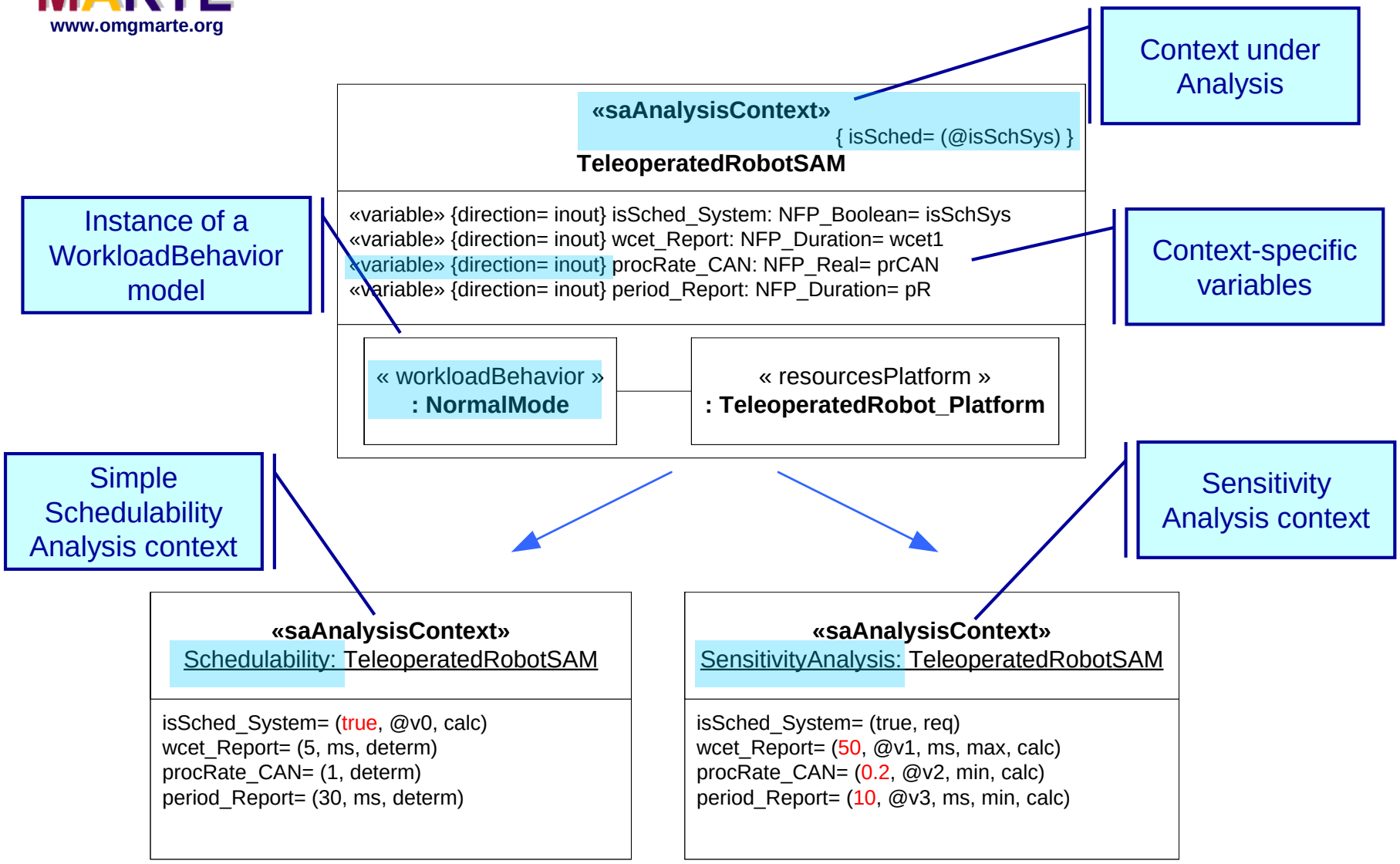
Event Streams (arrival patterns)

BehaviorScenario (response times, hosts utilization…)

Resources Platform Modeling Example



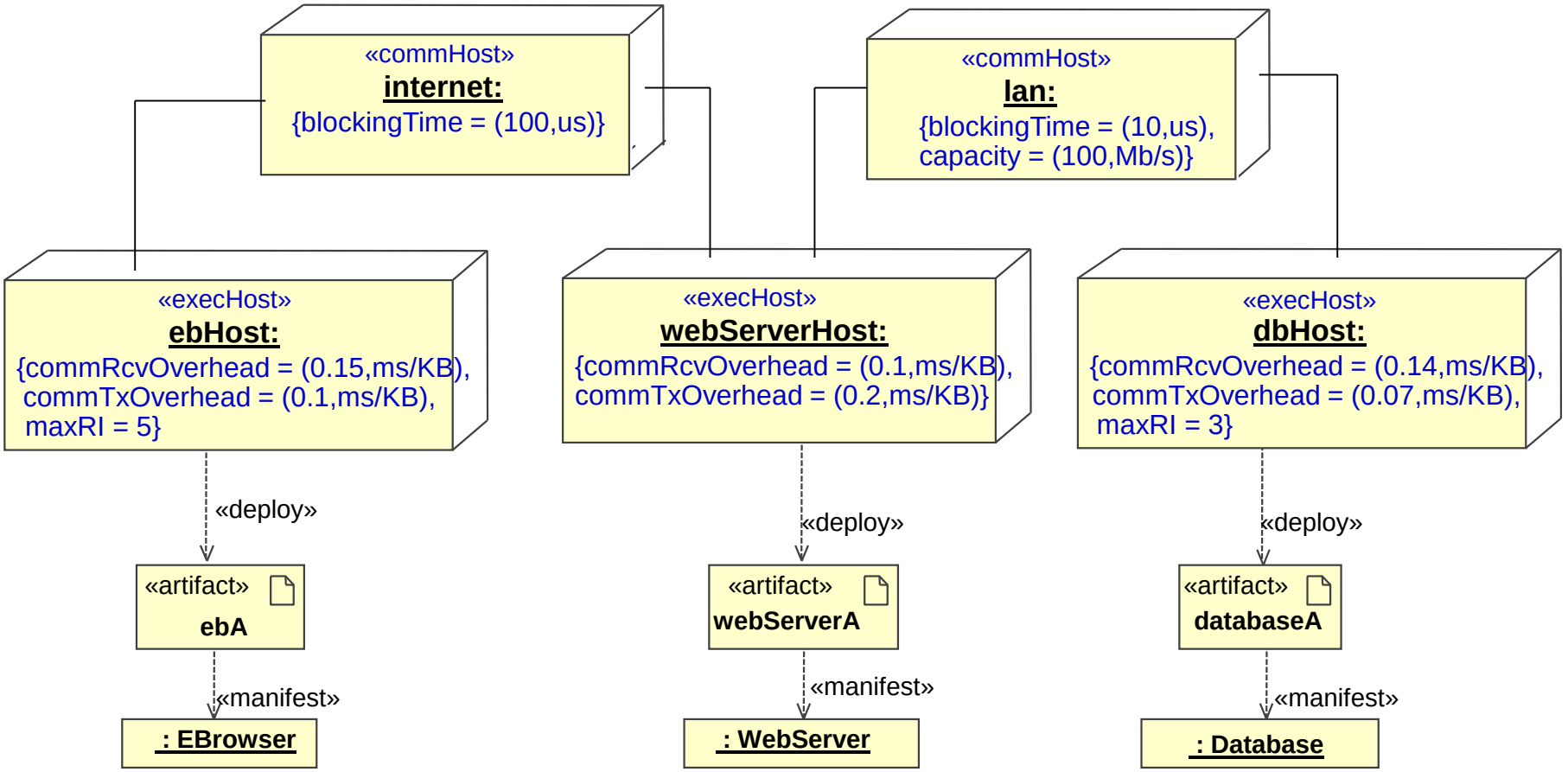
Sched. Analysis Context Example



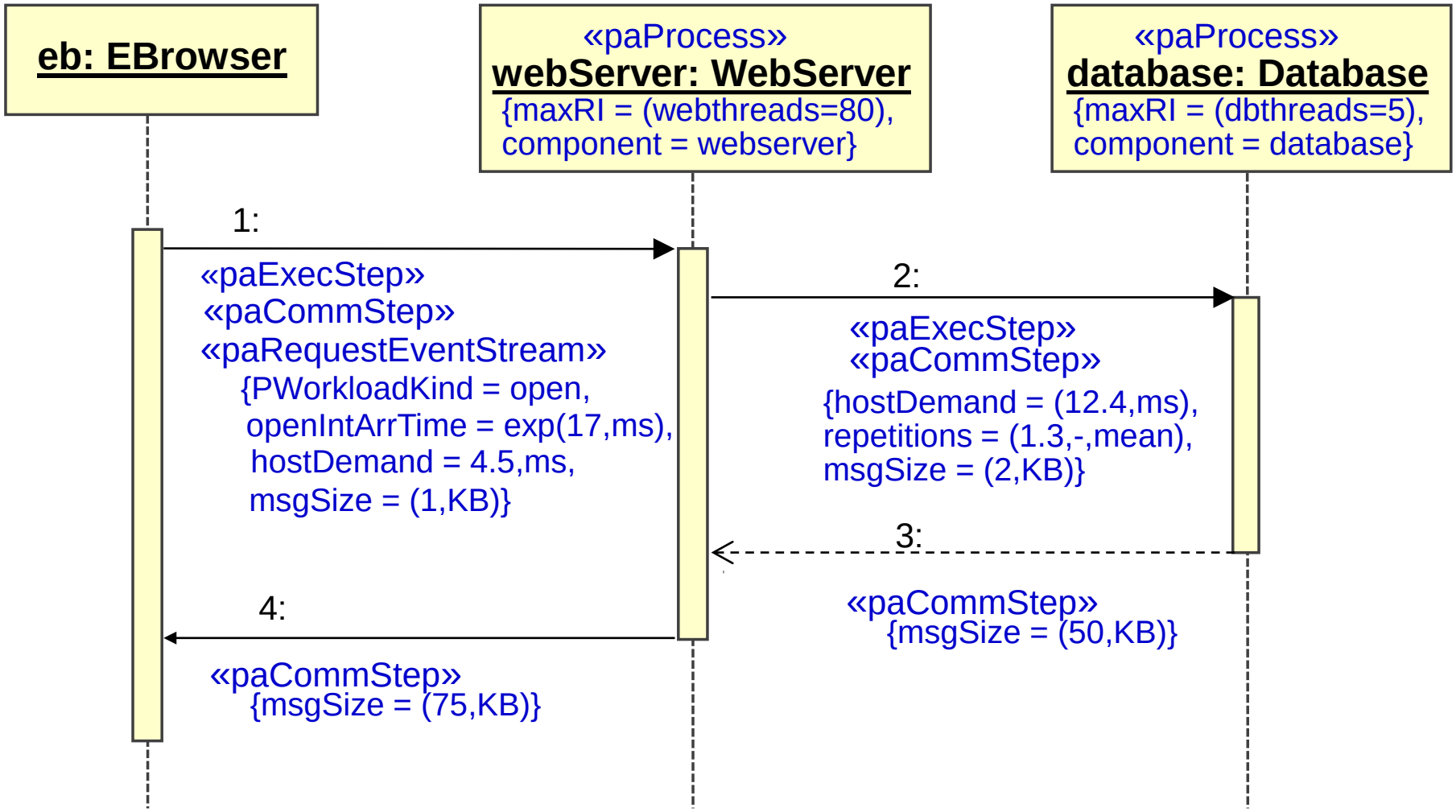
Performance Analysis Model

- **Specializes some GQAM stereotypes and reuses others**
  - Workload
    - specialized: PaRequestEventStream, PaWorkloadGenerator, PaEventTrace
  - Behaviour
    - reused: BehaviorScenario, AcqStep, RelStep
    - specialized: PaStep, PaExecStep, PaCommStep, ResPassStep, RequestedService
  - Resources
    - Reused: ExecHost, CommHost, CommChannel
    - Specialized: PaProcess
- **Supports most common performance analysis techniques**
  - Queueing Networks and extensions, Petri Nets, simulation
- **UML + MARTE models should contain**
  - Structural view: software architecture and deployment
  - Behavioral view: key performance scenarios

Example: deployment



Example: simple scenario



MARTE Annexes

- **Repetitive Structure Modeling**
- **Guidance for use of MARTE**
  - e.g. AADL-like models in MARTE
- **Value Specification Language (VSL)**
- **Clock Handling Facilities**
  - Clock Value Specification Language (CVSL)
  - Clock Constraint Specification Language (CCSL)
- **MARTE Library**

Conclusion

- **MARTE is the OMG specification for Modeling and Analysis Real-Time and Embedded systems**
- **It provides extensions to UML for modeling non-functional properties, time and resources, software and hardware execution platforms and allocations. MARTE enables model-based quantitative analysis, including schedulability and performance**
- **Eclipse-based open-source implementation is available**
  - Papyrus for MARTE (http://www.papyrusuml.org)
  - Marte to MAST (http://mast.unican.es/umlmast/marte2mast)
- **Ongoing efforts to align parts of MARTE with fUML**
- **It is possible to revise and enhance it→ MARTE 2.0 is coming… (The MARTE 1.3 Revision Task Force is still active)**

UML MARTE
www.omgmarte.org

References

- **OMG MARTE web site**
  - http://www.omgmarte.org
- **UML profile for MARTE**
  - http://www.omg.org/cgi-bin/doc?formal/2011-06-02
- **UML 2 Superstructure**
  - http://www.omg.org/cgi-bin/doc?formal/07-02-05