Model based schedulability analysis with MAST and the UML Profile for MARTE

Julio Medina
Universidad de Cantabria
SPAIN

WATERS 2010.

6-july-2010

Partially supported by the FP7 NoE ARTIST Design

Focusing

"…creating a common ground and a community to collect methodologies, software tools, best practices, data sets, application models, benchmarks and any other way to improve comparability of results in the current practice of research in real- time and embedded systems."

- Various Schedulability Analysis Techniques
- Assumptions about the platform scheduling capabilities
- The model of computation and the semantic link to the design intent
- Characterization of the environment
- Expressing the constraints and expected results

→ Analysis Model ←

Outline

- Basic ideas in (RMA based) schedulability analysis
- The MAST suite: model and tools
- Modeling for schedulability analysis with MARTE
- Discussion:
  - Where we are, what do we need?
  - What are the real possibilities?

Key Ideas

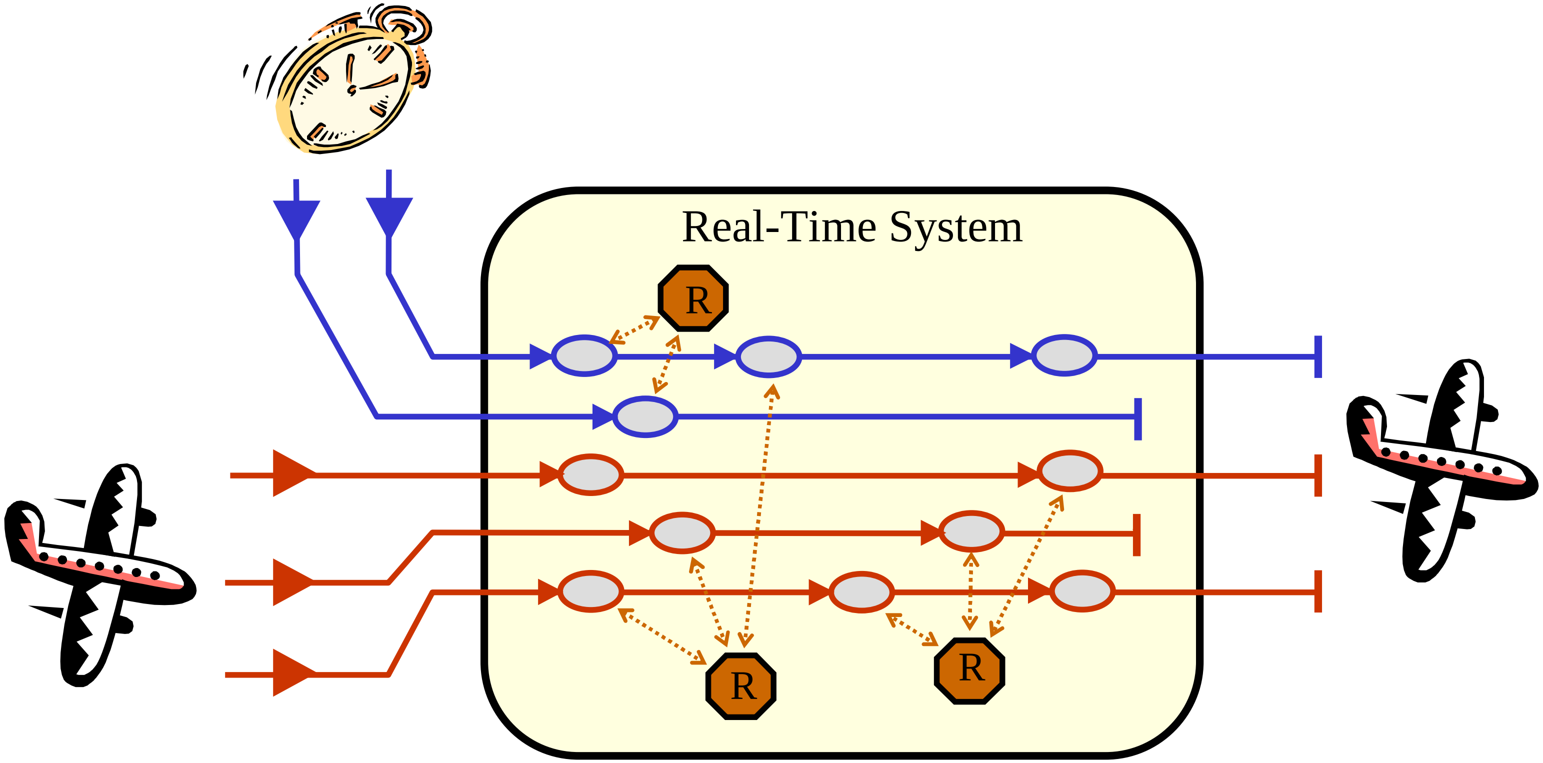
- Real-time goals are: predictability, guaranteed deadlines, and stability in overload.
- Rate monotonic analysis
  - based on rate monotonic scheduling theory
  - analytic formulas to determine schedulability
  - framework for reasoning about system timing behavior
  - separation of timing and functional concerns
- Provides an engineering basis for designing real-time systems

Key Ideas (cont.)

Two concepts help to build the worst-case condition:

- *Critical instant*. The worst-case response time for all tasks in the task set is obtained when all tasks are activated at the same time
- *Checking the first deadline*. When all tasks are activated at he same time, if a task meets its first deadline, it will always meet all of its deadlines

Transactional approach for analysis and design



Instance based (Classic RMA, MAST, SPT, MARTE,…)

The MAST suite: model and tools

Brief view of MAST

MAST:
A Timing Behavior Model for Embedded Systems Design/Verification Processes

By:

Michael González Harbour, José Javier Gutiérrez, José Carlos Palencia, José María Drake, Julio Medina, Patricia López

Universidad de Cantabria, Spain
`http://mast.unican.es/`

Workshop on Analysis Tools and Methodologies for Embedded and Real-time Systems; *WATERS 2010*, Brussels, July 2010.

1. Introduction: Background

Many real-time systems are now distributed

- Cyclic executives being replaced by run-time schedulers
- Fixed priority and EDF scheduling are most popular among the run-time scheduling policies

Schedulability analysis techniques have evolved a lot in the last decade

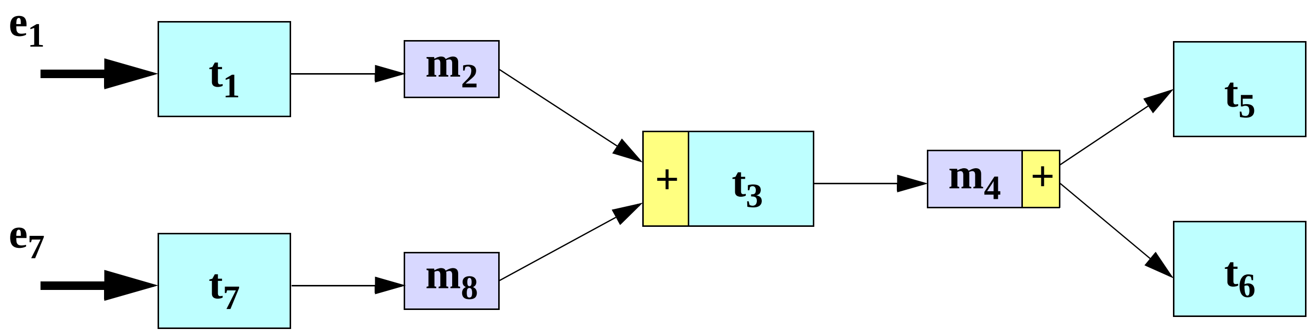
- Originally RM and DM priority assignment techniques, together with response-time analysis
- Extended to distributed systems (holistic analysis, HOPA)
- Offset-based analysis introduced (FP and EDF)
- Multiple-event synchronization handled

Motivation

The latest schedulability analysis techniques are difficult to apply by hand

Need for a rich and flexible model of the real-time system:

- distributed, multiprocessor, or single processor
- composable software modules
- separation of architecture, platform, and software modules
- rich set of event-driven patterns; e.g.:

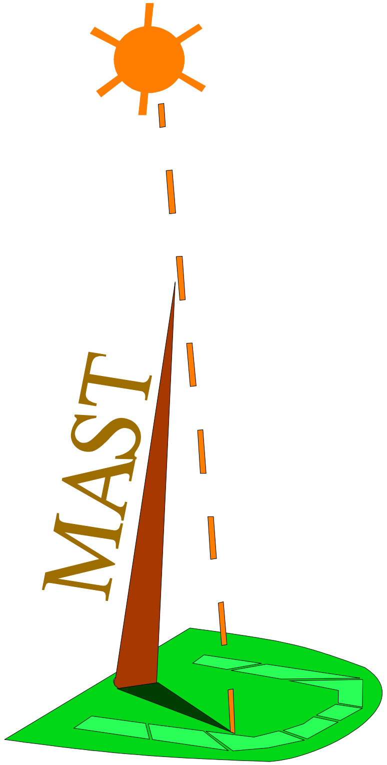


Objectives

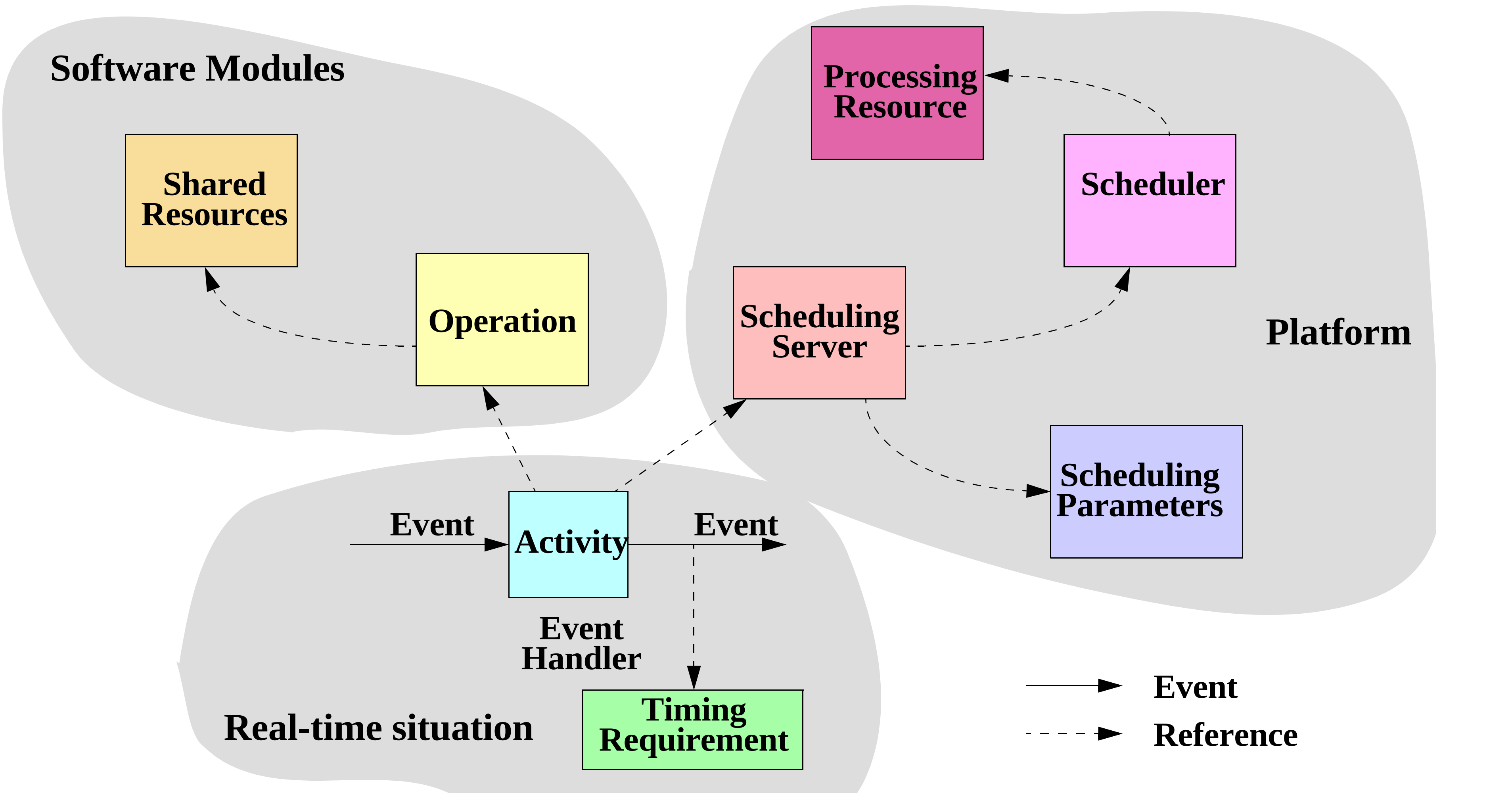
- Develop a *model* for describing the timing behavior of event-driven distributed real-time systems
- *Open model* that may evolve to include new characteristics or points of view of the system
- Develop a set of *tools* for analyzing the timing behavior of the application:
  - Schedulability analysis (hard real-time requirements)
  - Synchronization blocking calculation
  - Discrete-event simulation (soft real-time)
  - Priority assignment
  - Sensitivity analysis

MAST: A timing behavior model

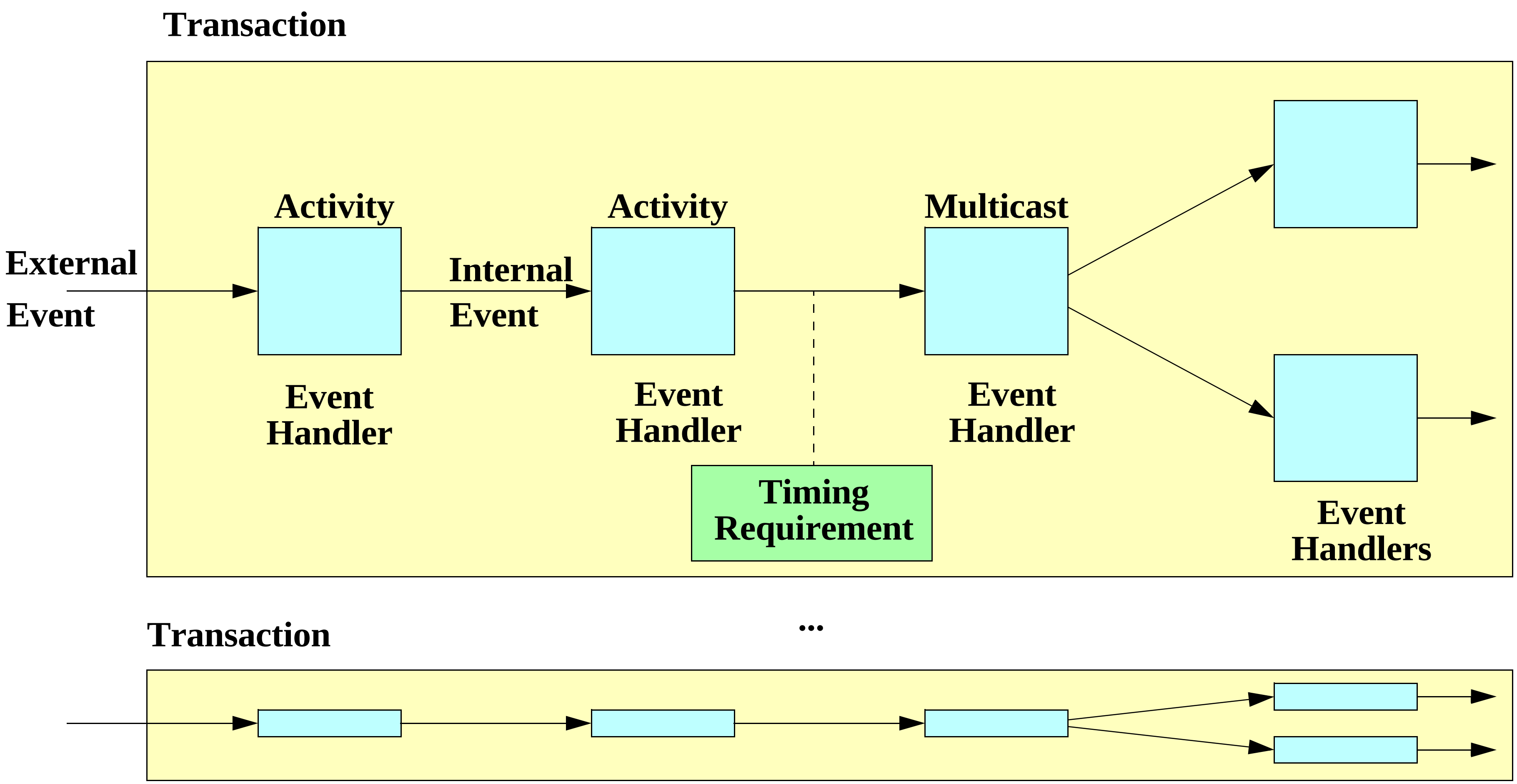
1. Introduction: Background, Motivation and Objectives
2. *Overview of the Real-Time Model*
3. Elements of the MAST Model
4. Integration into design processes
5. The MAST tool suite
6. Conclusions and Future Work



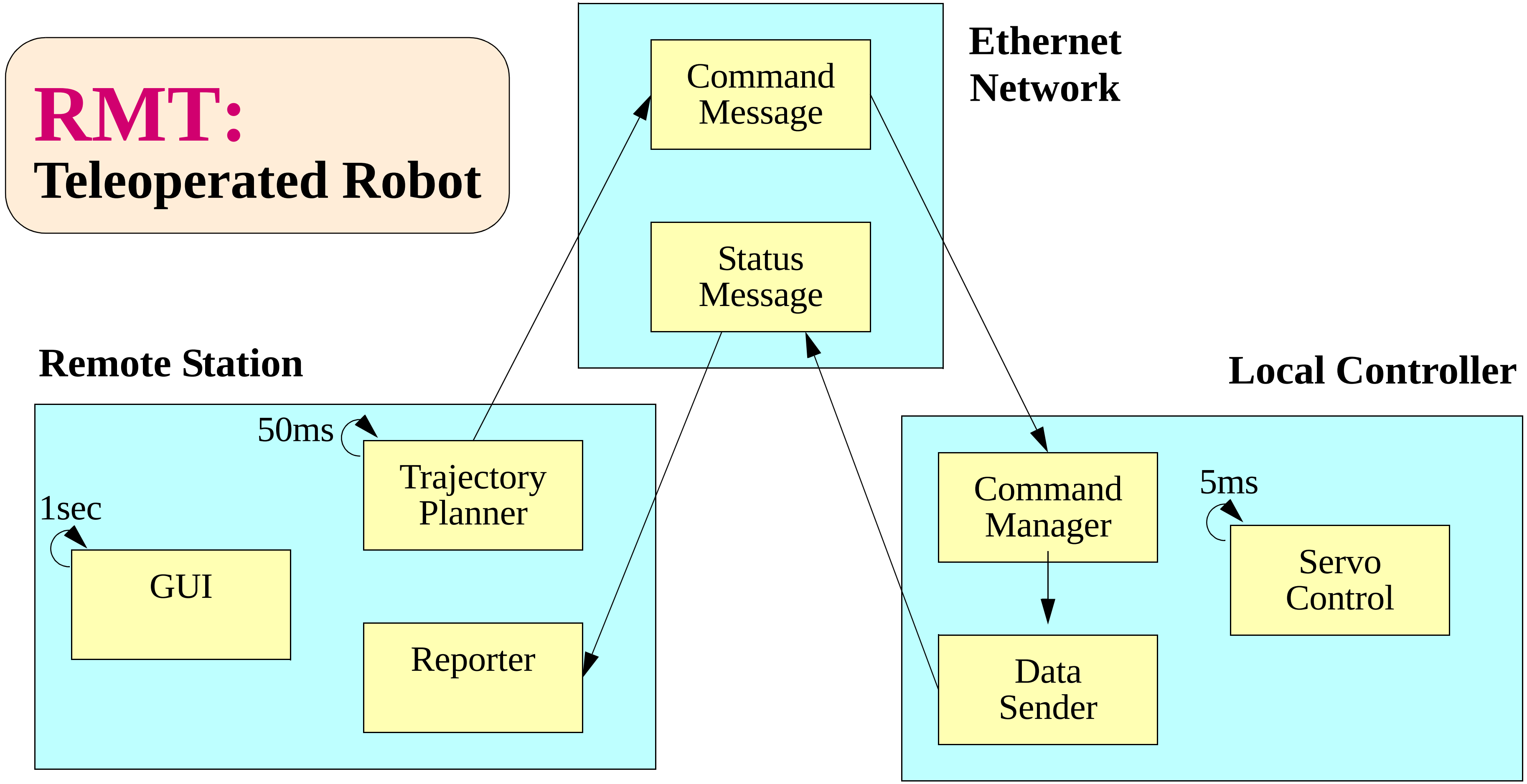
2. Real-Time Model: Overview



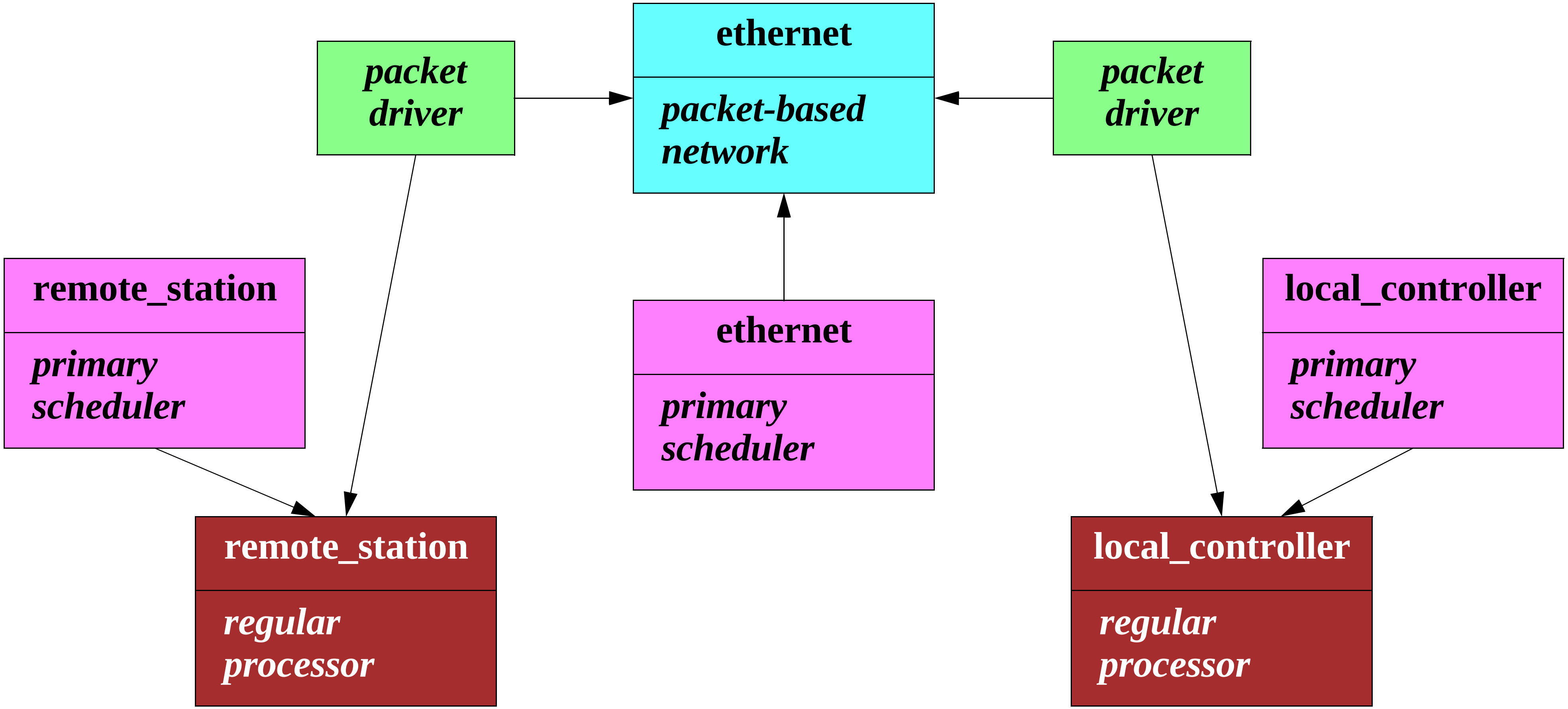
Real-Time Situation



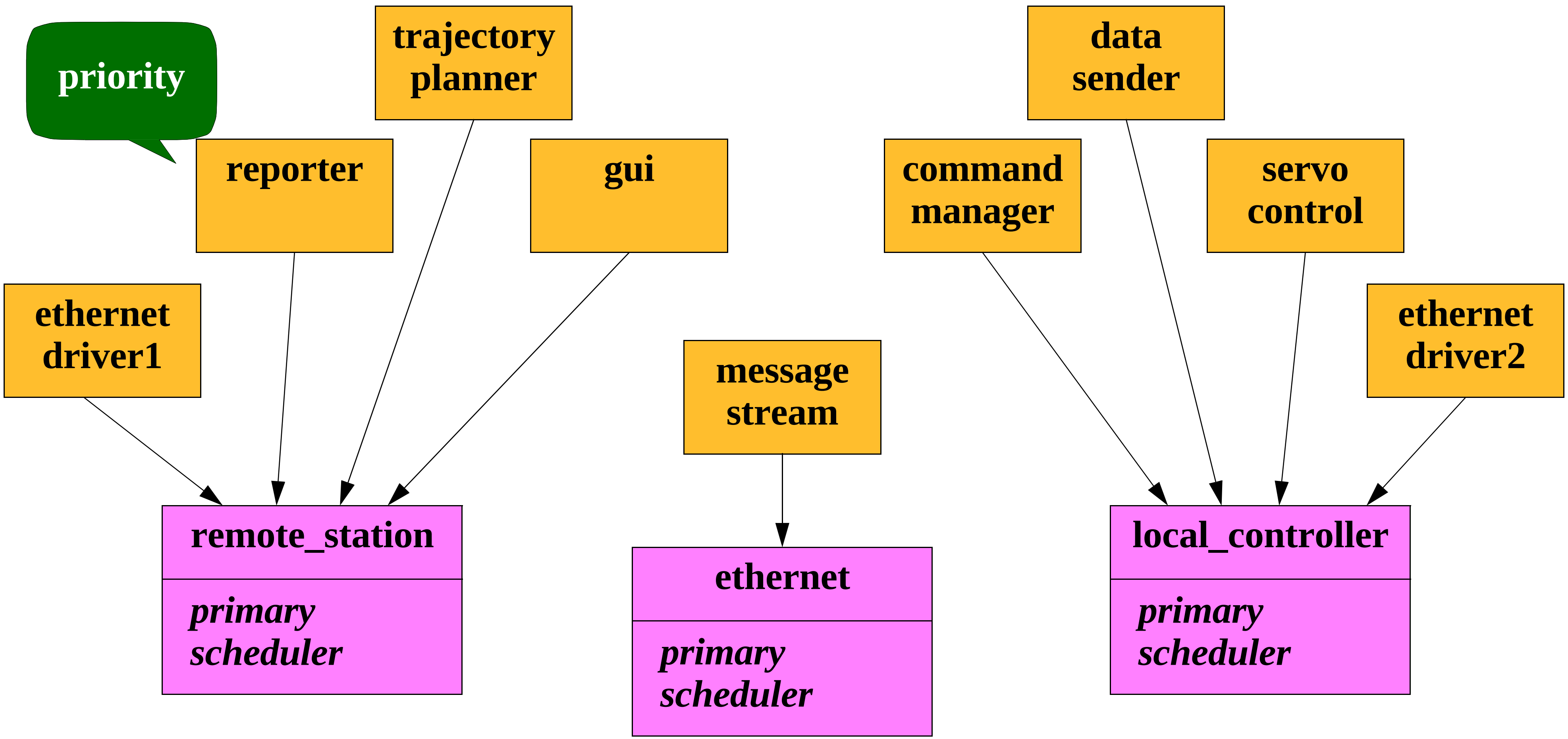
3. Elements of the MAST model: Example



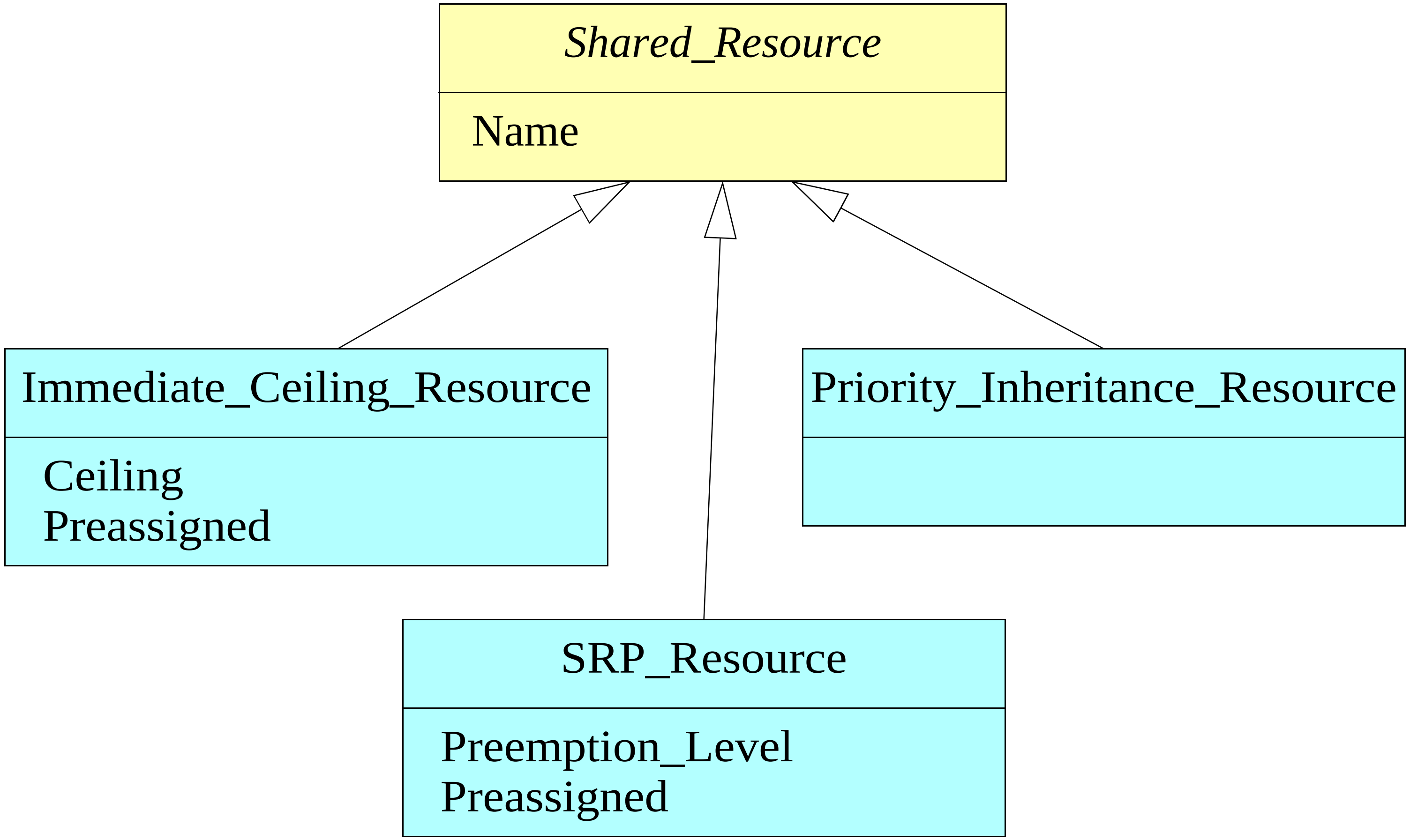
Processing resources, schedulers, drivers, and timers



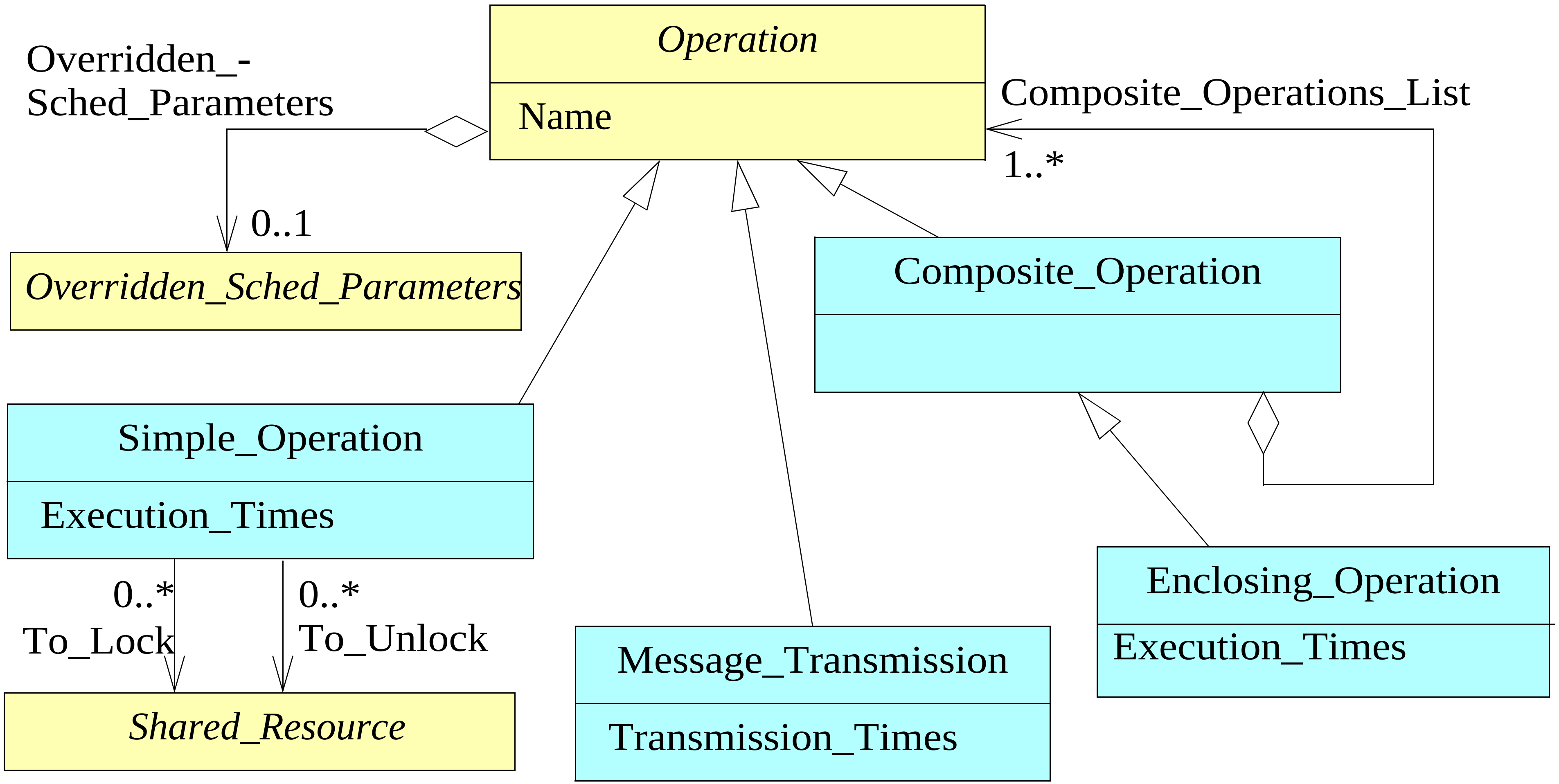
Scheduling servers



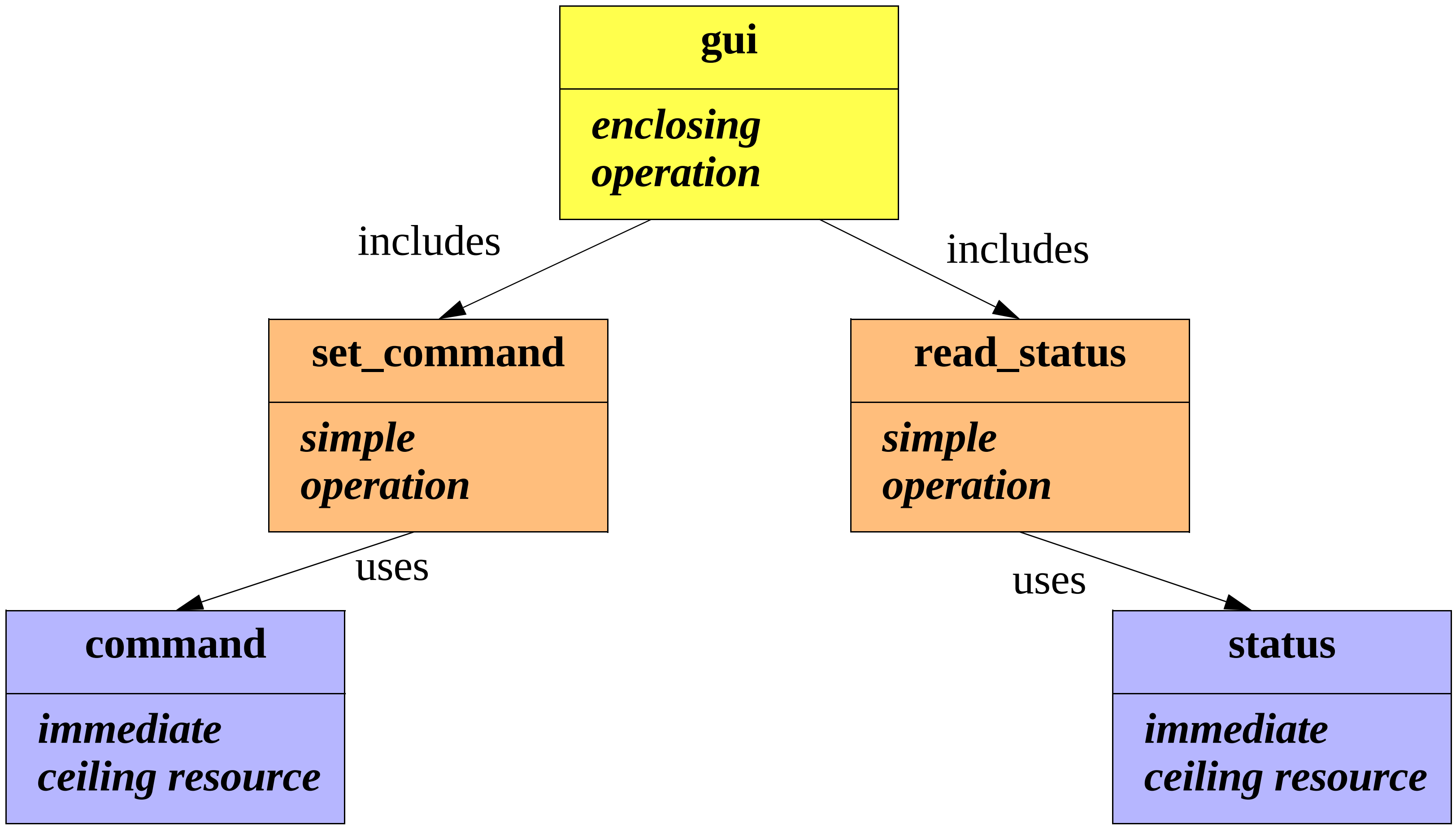
Shared resources



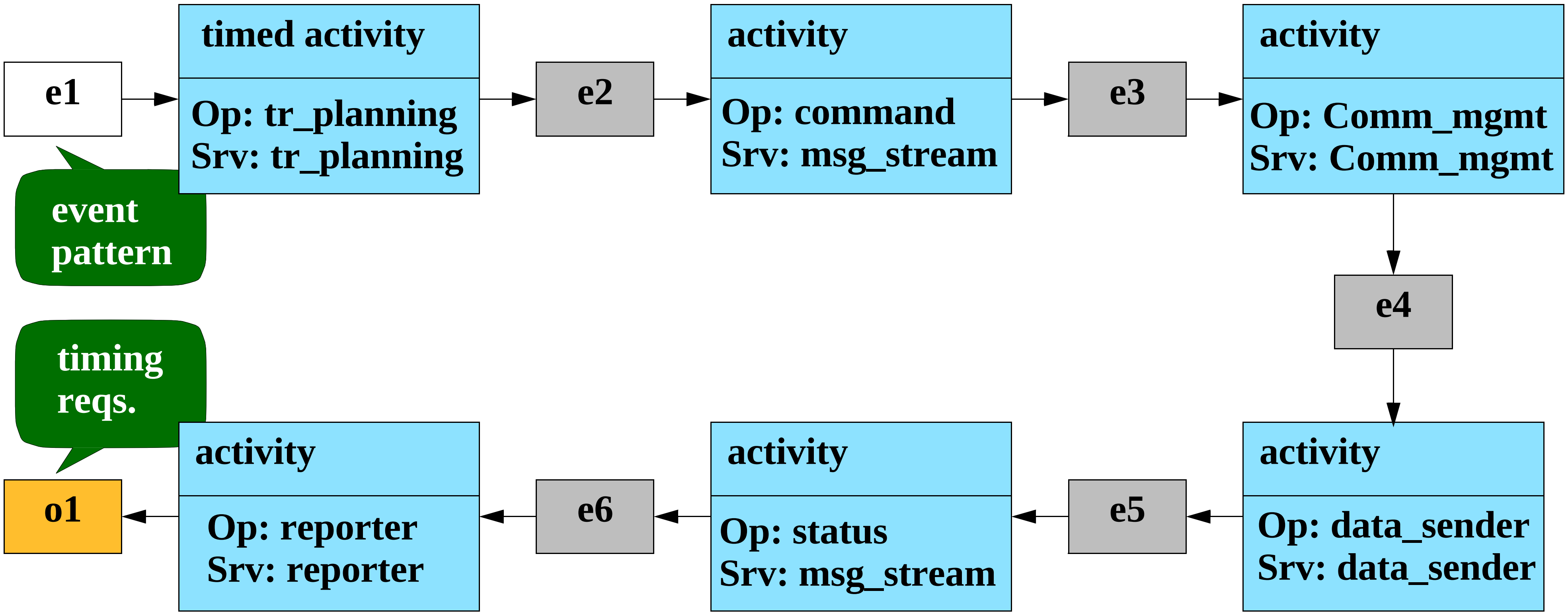
Logical Operations Model



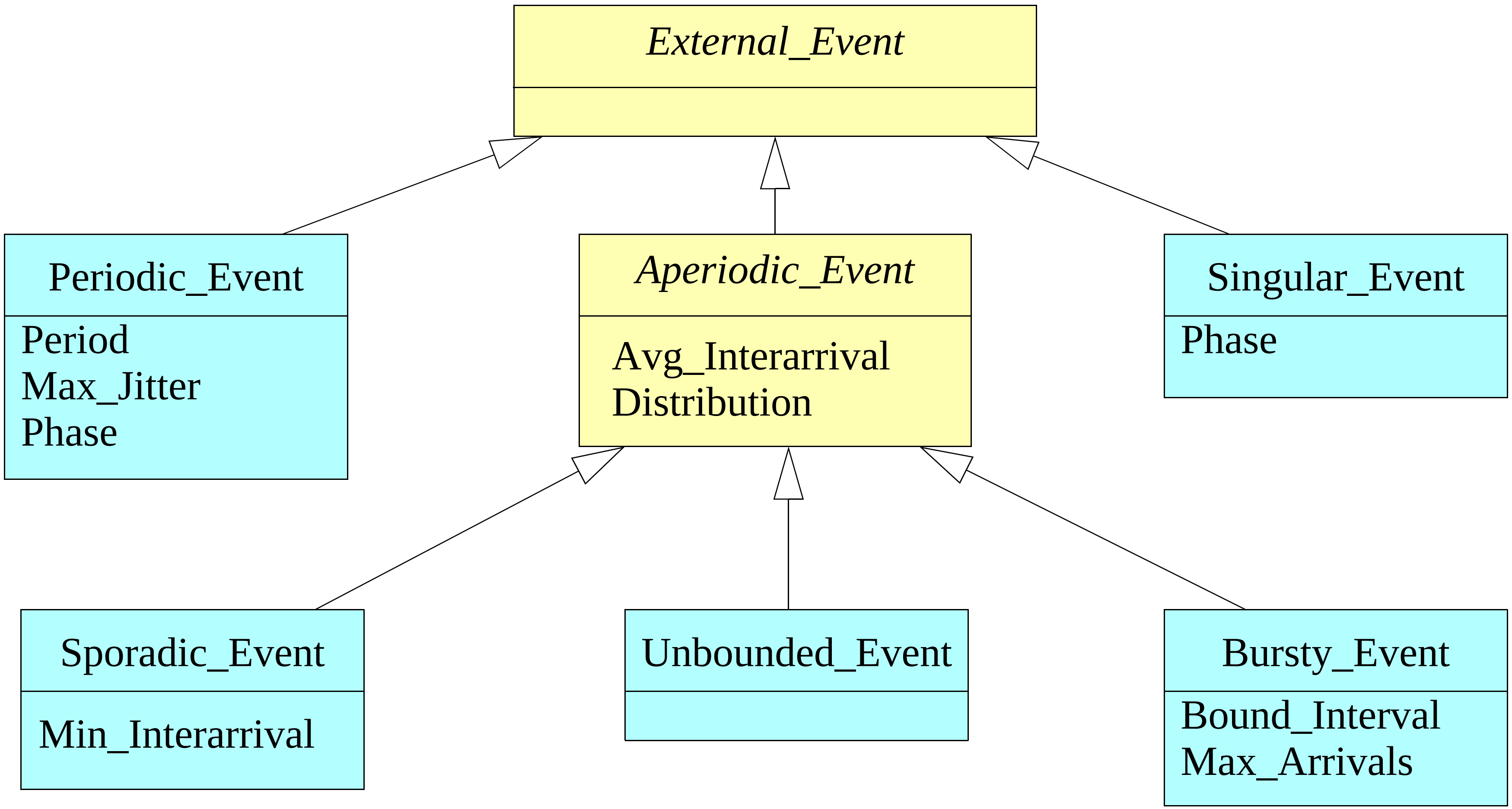
GUI operation in the example



Transactions: Distributed transaction in the example

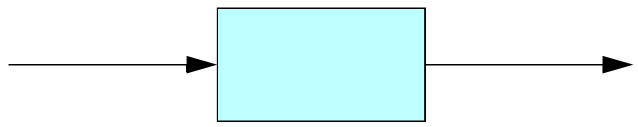


External Events

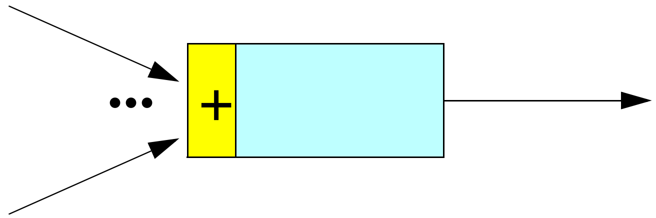


Event Handlers

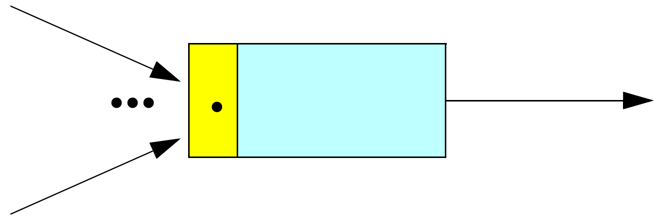
Activity / Rate Divisor / Delay / Offset



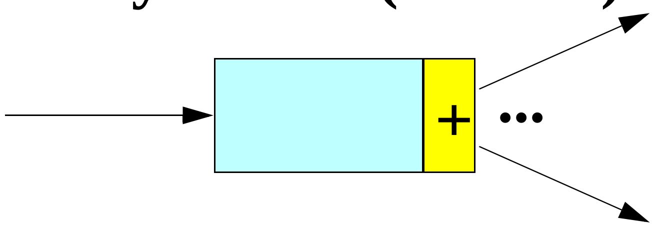
Concentrator (Merge)



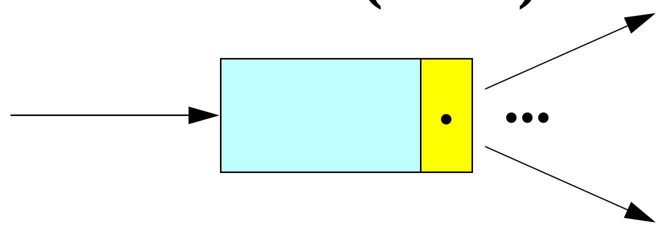
Barrier (Join)



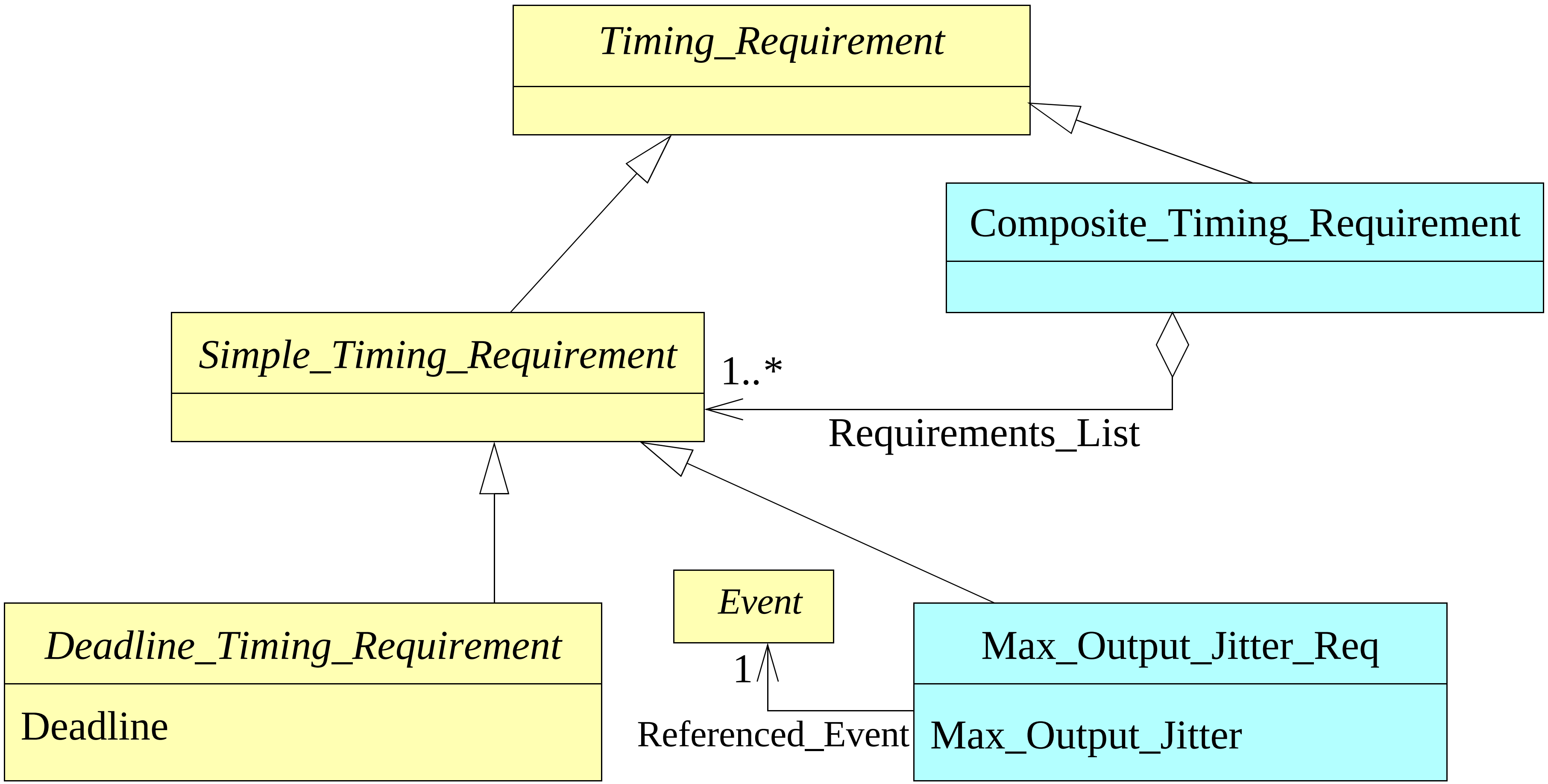
Delivery Server (Branch)



Multicast (Fork)



Timing requirements



4. Integration into the design process

UNIVERSIDAD DE CANTABRIA

Components built with their own timing behavior model

- passive components: operations and shared resources
- active components: single or multithreaded, distributed, ...

The model is parameterized

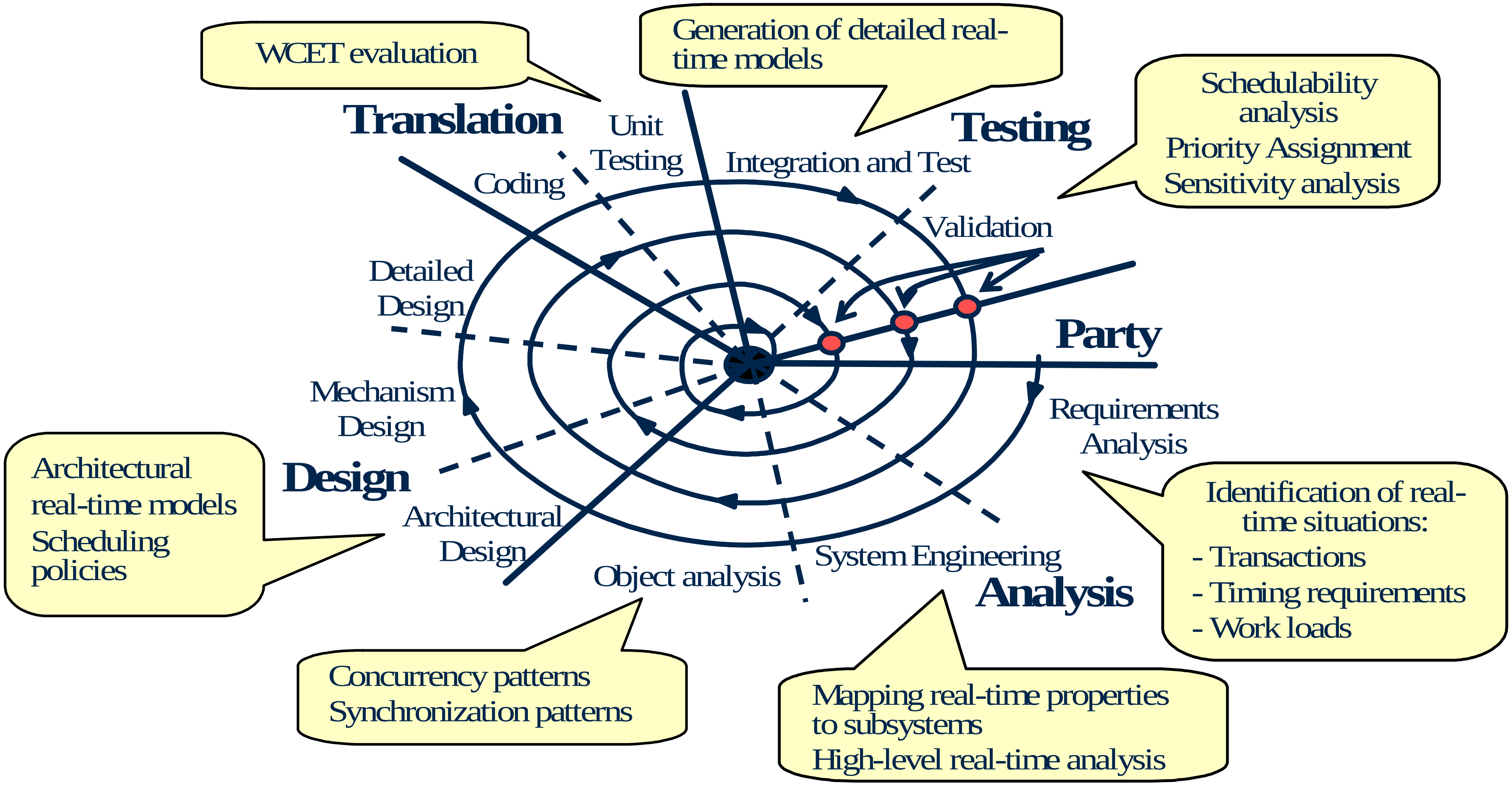
- i.e., actual data for WCETS

Deployment tool

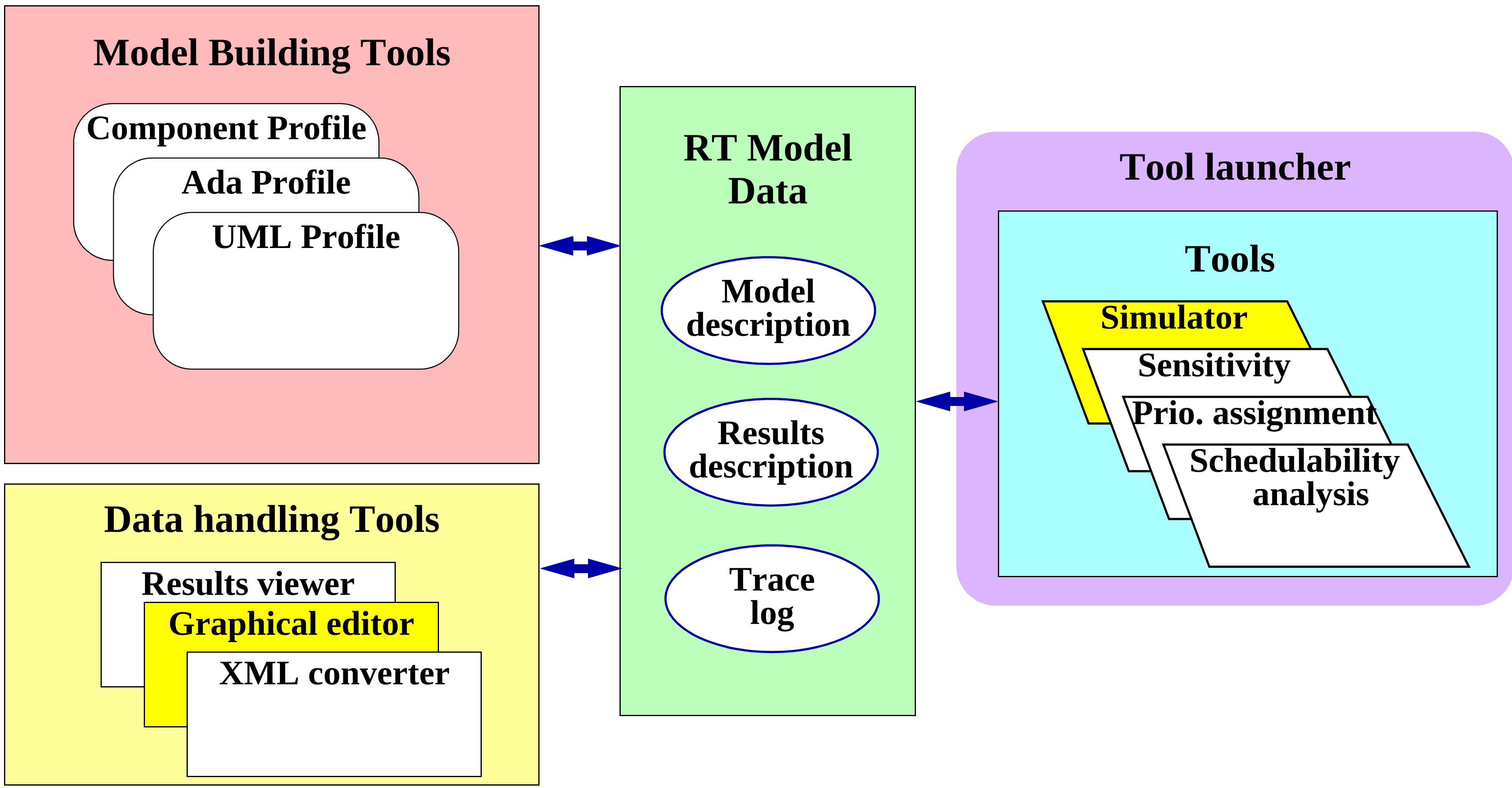
- instantiates the parameterized component models
- provides the platform model
- integrates them with the real time situation model

Automatic schedulability analysis is then made

Integration into the design process



5. The MAST Environment



6. Conclusions

UNIVERSIDAD DE CANTABRIA

MAST defines a model for describing real-time systems

- distributed and multiprocessor
- complex synchronization and event-driven schemes
- composable software modules
- independence of architecture, platform and modules

MAST provides an open set of tools

- hard and soft real-time analysis, FP and EDF
- automatic blocking times, priority assignment, sensitivity analysis, graphical editor...

XML specification and the conceptual modelling approach allows easy integration with other tools and formalisms like UML

Future Work: MAST 2.0

- Align MAST with MARTE. The elements in MARTE can be described using MAST, but the names are different. Since MARTE is an industry standard the names in both models should be harmonized.
- Add partitioned scheduling in processors and networks, and resource reservations such as FRESCOR contracts.
- Enhance the modelling capabilities of MAST, by adding two new elements: modelling the effects of mutual exclusion due to the use of a thread, even if it suspends itself, for instance in a synchronous remote procedure call (RPC); and modelling communication switches, such as AFDX switches.

MAST 2.0 (cont.)

- Enhance the timer overhead models.
- Enhance the expressiveness of average-case performance parameters, used by the simulator.
- Add new timing requirements, in particular analysis for maximum queue sizes.

URL

http://mast.unican.es

Key reasoning behind the chances of a standard language for conceptual modeling to be useful in the scheduling analysis of RTE Systems

- Consider the framework of a Model Based approach (MDD).
- MDD and UML have been broadly introduced and used in principle by the software engineering community, and have reached a significant number of practitioners and tool support.
- To take benefit of this in the RTE domain, They need to be capable of supporting the necessary (at least timing) verifications.
- These leads to the necessity of model based scheduling analysis techniques.
- As well as the necessity to have the modeling elements to describe the platform, the interacting environment and the timing requirements.

Modeling for schedulability analysis with MARTE

Tutorial on the analysis capabilities of the UML Profile for MARTE

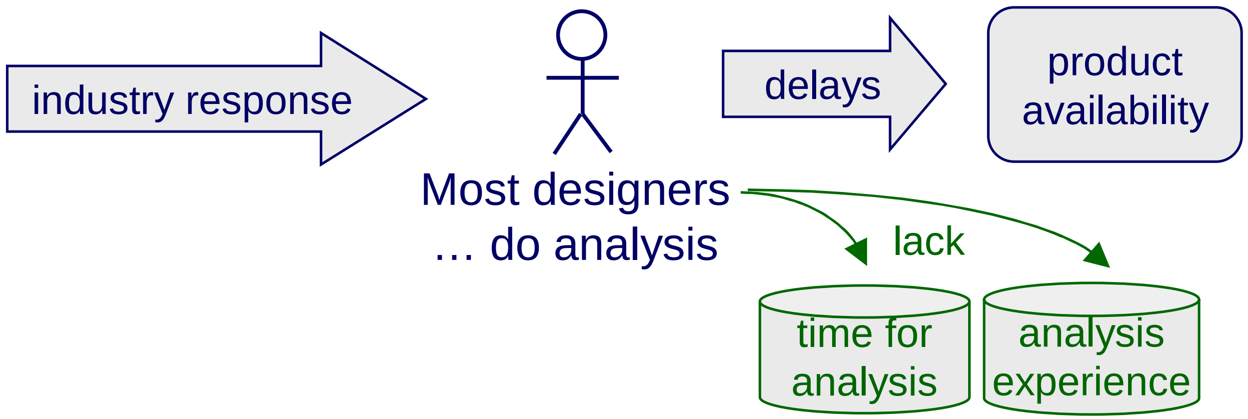
Agenda

(extract from the MARTE Tutorial)

- **Part 1**
  - Introduction to MDD for RT/E systems & MARTE in a nutshell
- **Part 2**
  - Non-functional properties modeling
  - Outline of the Value Specification Language (VSL)
- **Part 3**
  - The timing model
- **Part 4**
  - A component model for RT/E
- **Part 5**
  - Platform modeling
- **Part 6**
  - Repetitive structure modeling
- **Part 7**
  - **Model-based analysis for RT/E**
- **Part 8**
  - MARTE and AADL
- **Part 9**
  - Conclusions

Industry Landscape of Analysis Practice

- **Why "non-functional analysis" is important?**
  - ➔ Early: avoid design mistakes, asses design trade-offs…
  - ➔ Later: evaluate the impact of modifications,…
- **But,… doing analysis is hard and time-consuming!...**
  - ➔ From 40% to 50% of development costs [Hum02], …

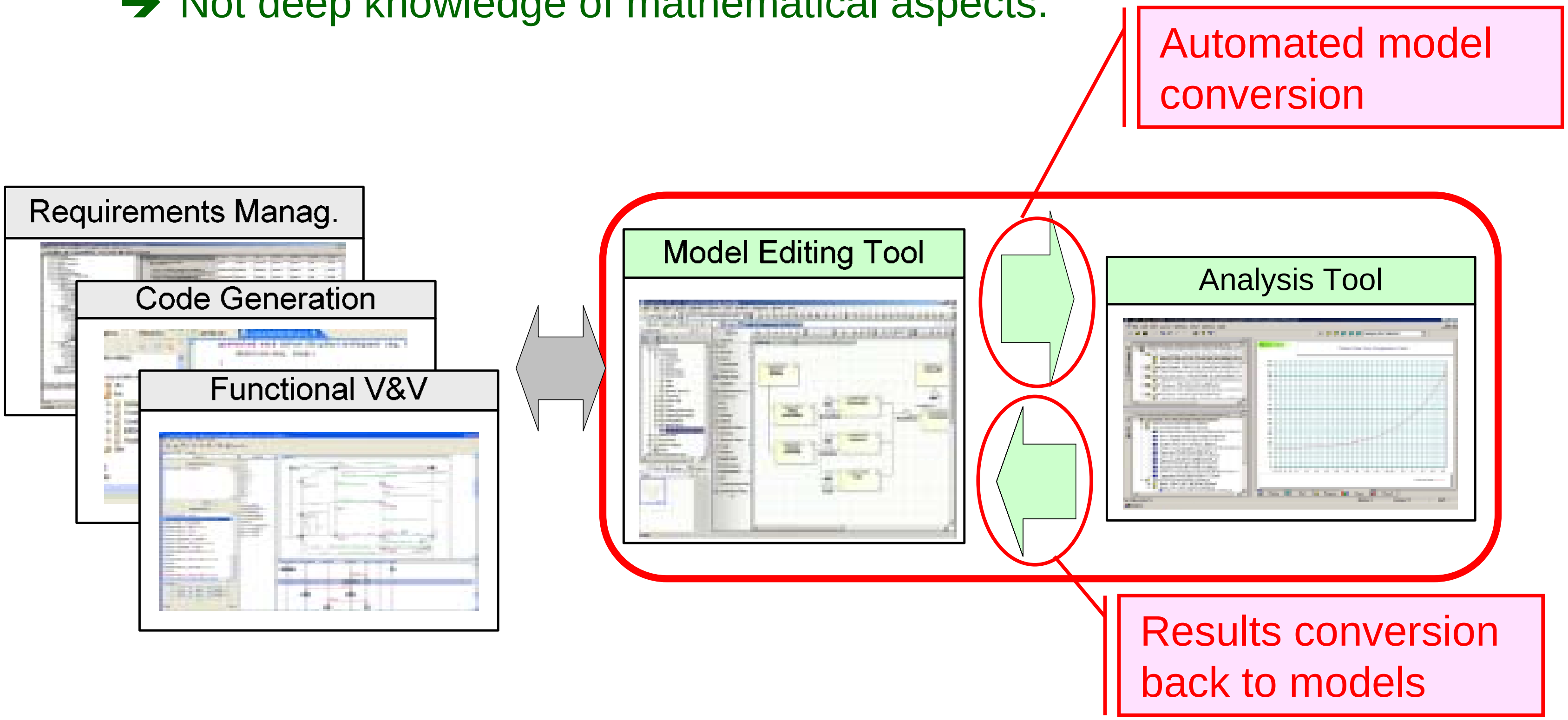


We need new approaches to cope with analysis complexity!

[Hum02] Humphrey, W. S., "The Payoff from Software Quality", In Computerworld, 2002

Goals for Model-Based RTES Analysis

- **Ability to specify non-functional information in models**
  - ➔ Automate the generation of analysis models (time savings!)
  - ➔ Not deep knowledge of mathematical aspects.



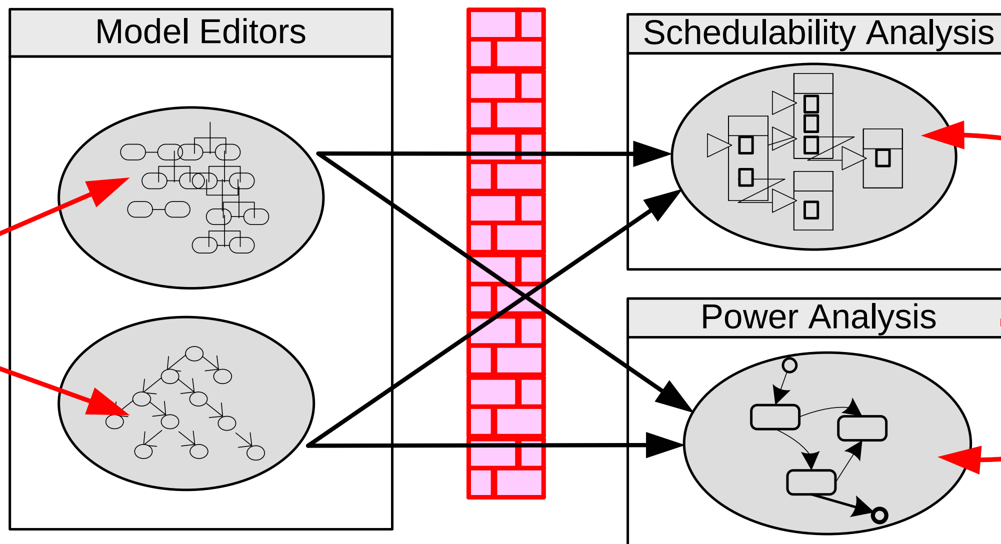
UML MARTE
www.omgmarte.org

Needs for Model-Based RTES Analysis

Design-Oriented Models
(code generation,...)

Semantic mismatch

Analysis-Oriented Models
(formal models)



How to specify unambiguous non-functional Information?

Trade-offs between different parameters?

Data-flow, Components, Classes, State-Charts,...

Holistic models, RMA-based, queuing theory...

Goals in Non-Functional (or Quantitative) Analysis

It offers a mathematically-sound way to calculate NFPs of interest based on other available NFPs and the system behavior

- **Different Goals for Evaluate & Verify System Architectures**
  - Point evaluation of the output NFPs for a given operating point defined by input NFPs
  - Search over the parameter space for feasible or optimal solutions
  - Sensitivity analysis of some output results to some input parameters
  - Scalability analysis: how the system performs when the problem size or the system size grow.

MARTE Features for Quantitative Analysis

- **Improvements w.r.t. SPT**
  - Extend implementation and scheduling models
    - e.g. distributed systems, hierarchical scheduling
  - Extend the set of analysis techniques supported
    - e.g. offset-based techniques
  - Extend timing annotations expressiveness
    - Overheads (e.g. messages passing)
    - Response times (e.g. BCET & ACET)
    - Timing requirements (e.g. miss ratios and max. jitters)
- **New features w.r.t. SPT**
  - Support for sensitivity analysis
  - Improve modeling reuse and component-based design.
  - Support of the “Y-chart” approach: application vs. platform models

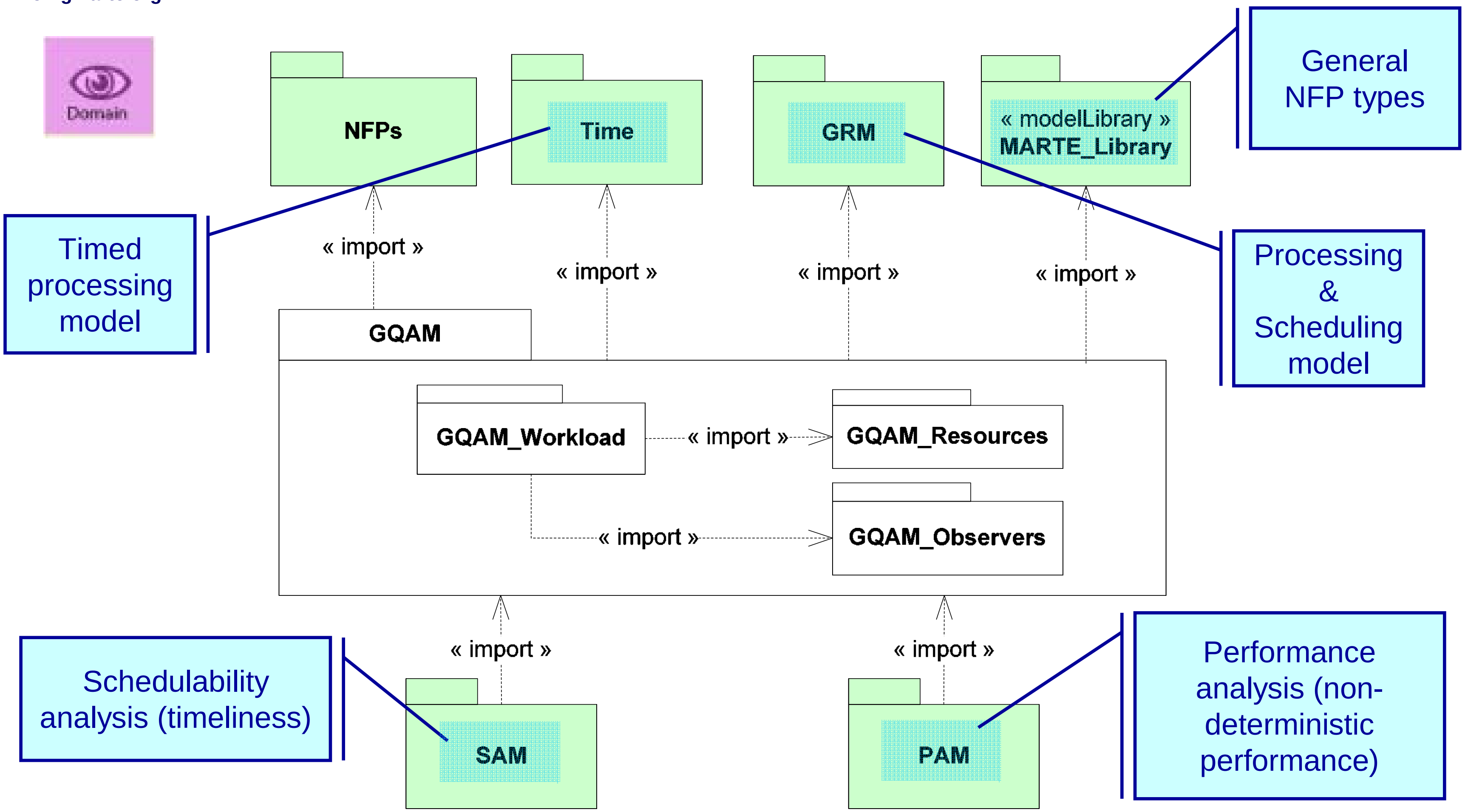
UML-Based Analysis Foundations

- **GQAM Profile factorizes common constructs and NFPs**
  - Stereotypes define “analysis” abstractions
    - workload events, scenarios,…
    - schedulable entities, shared resources, processing nodes, schedulers…
  - Stereotype attributes define pre-defined NFPs
    - e.g. event arrival patterns, end-to-end deadlines, wcet-bcet-acet,…
- **The analysis sub-profiles define model well-formedness rules**
  - It includes “constraints” to construct “analyzable” models, w.r.t…
  - ”Analysis Model Viewpoints” (e.g., schedulability analysis viewpoint)
  - Specialized constraints must be refined by technique-specific approaches

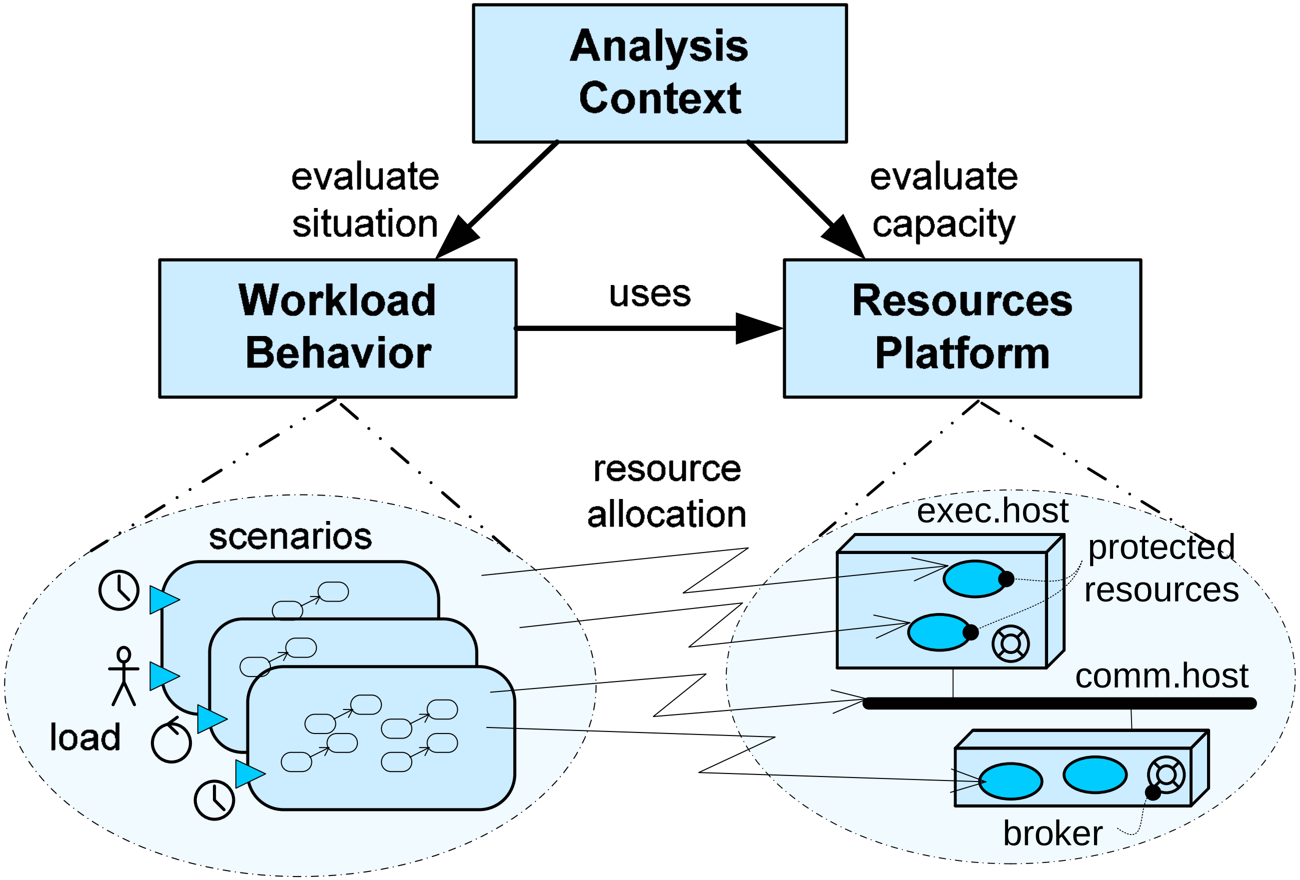
The MARTE analysis sub-profiles provide standard constructs to map UML models on well-established analysis techniques

➔MARTE “Foundations” and “GQAM” allow for extending to further techniques

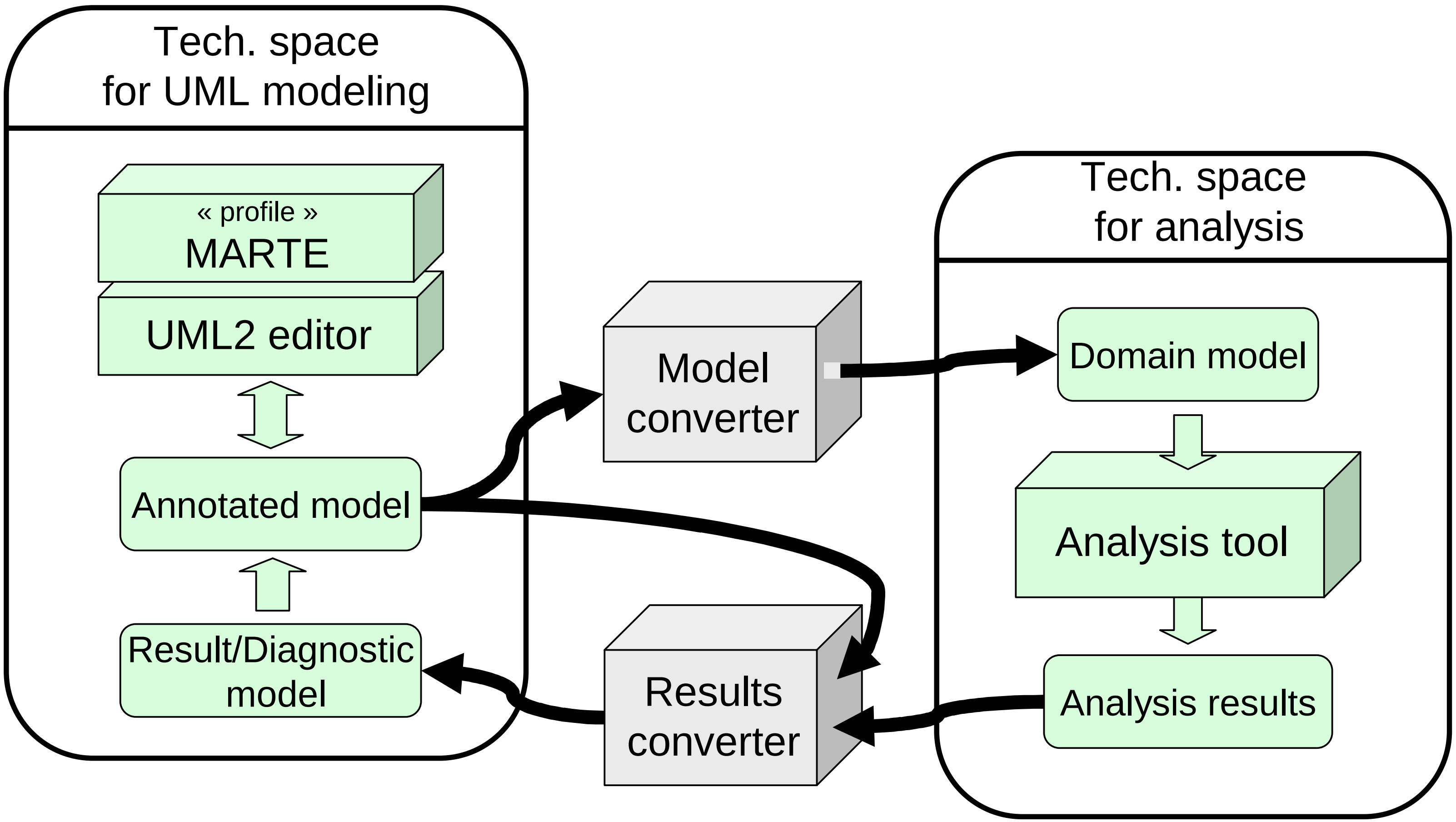
GQAM: Dependencies and Architecture



GQAM: Analysis Modeling Structure



Processing Schema for Analysis

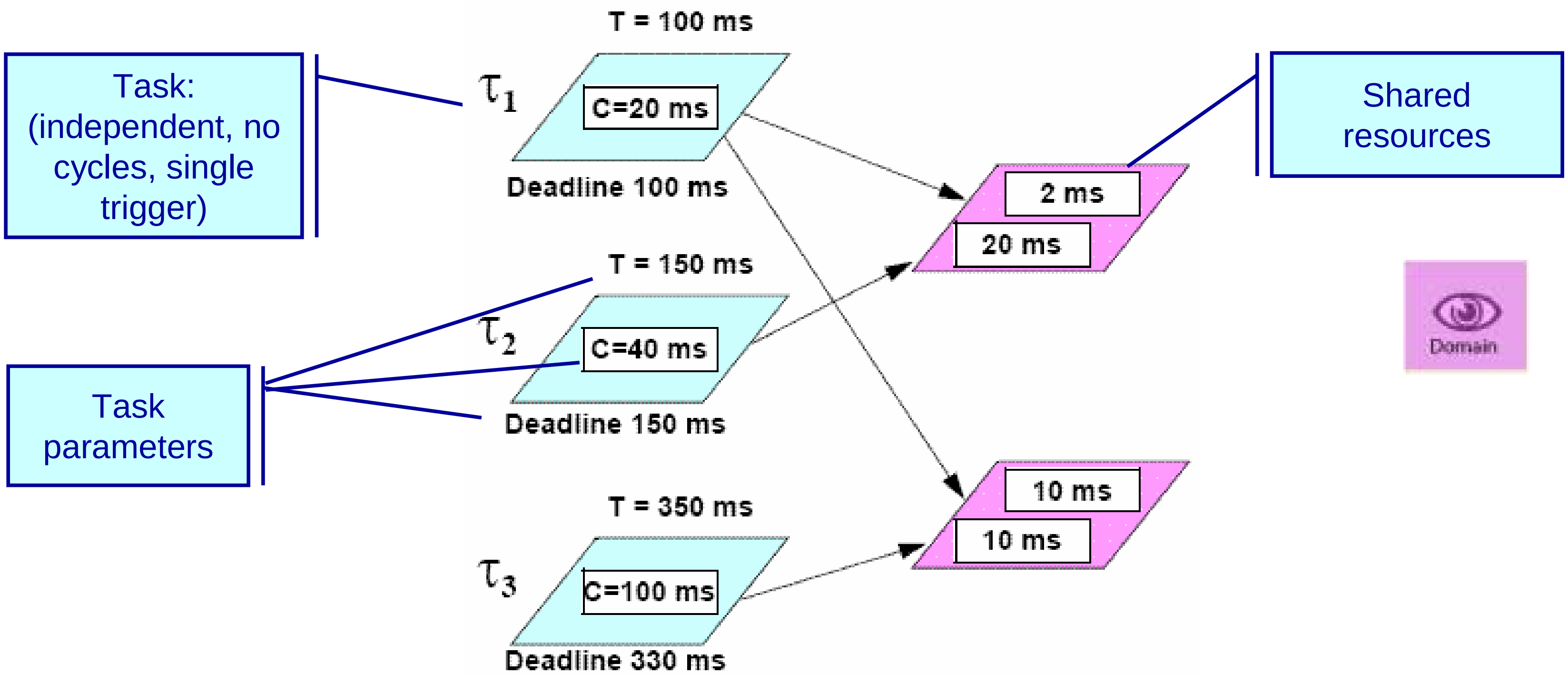


Schedulability Analysis

Provides the ability to evaluate time constraints and guarantee worst-case behavior of a system or particular piece of software

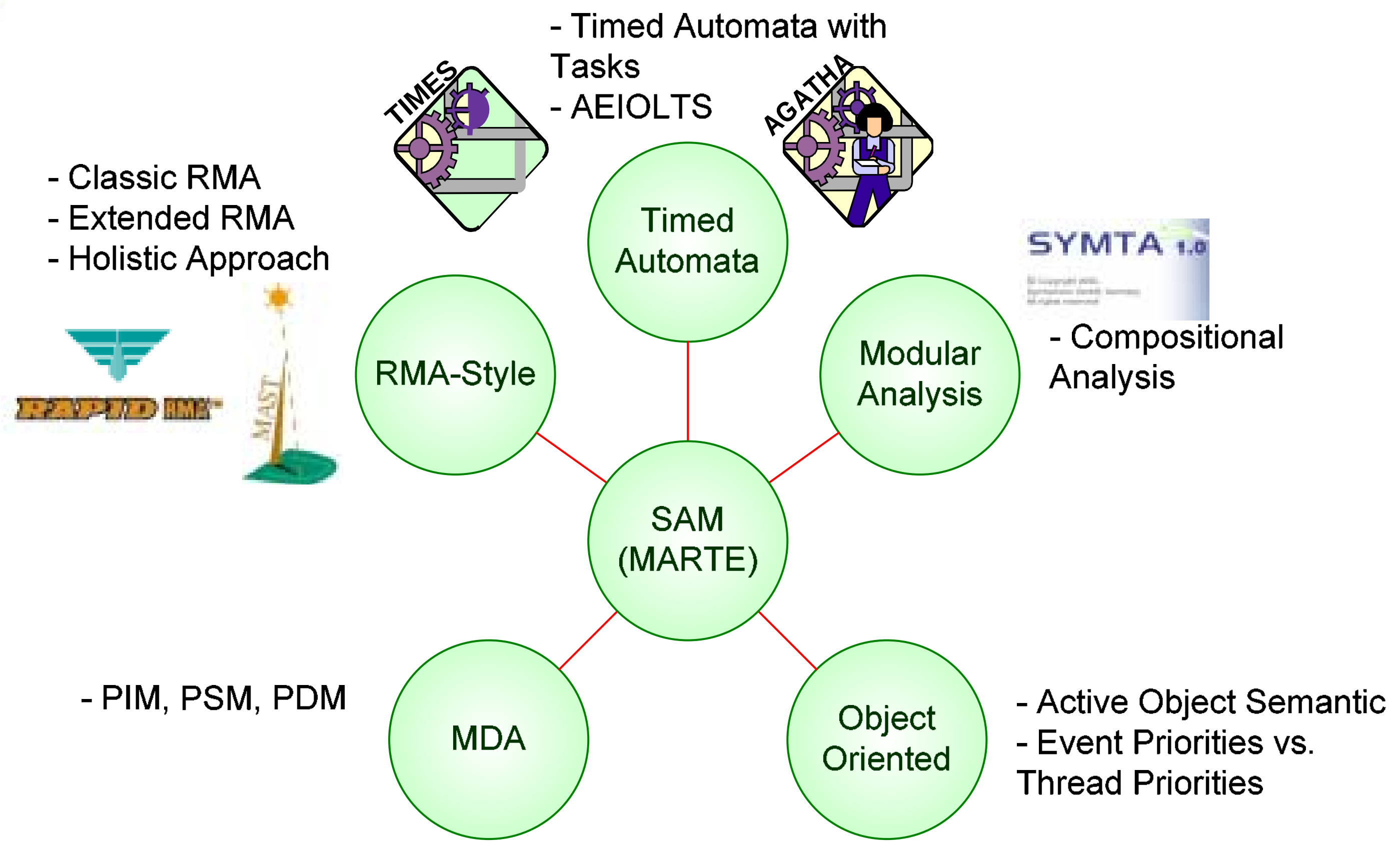
- **Schedulability analysis offers:**
  - Offline guarantees. E.g., worst-case latencies and worst-case resource usage.
  - At different development stages.
    - Early analysis: to detect potentially unfeasible real-time architectures.
    - Later analysis: to discover temporal-related faults, or to evaluate the impact of migrations (e.g., scheduling strategies).
- **Provide answer to questions such as for example...**
  - Will we miss any deadline if we switch a processor from a normal operation mode to a lower-consumption mode?
  - If yes, how can we modify task workloads for allowing our system to still work?

A Simple Example (Classical Scheduling Theory)



- **Three main analysis approaches for verify timeliness:**
  - Critical instant calculation
  - Utilization bound test
  - Response time calculation

SAM: Integration Different Approaches



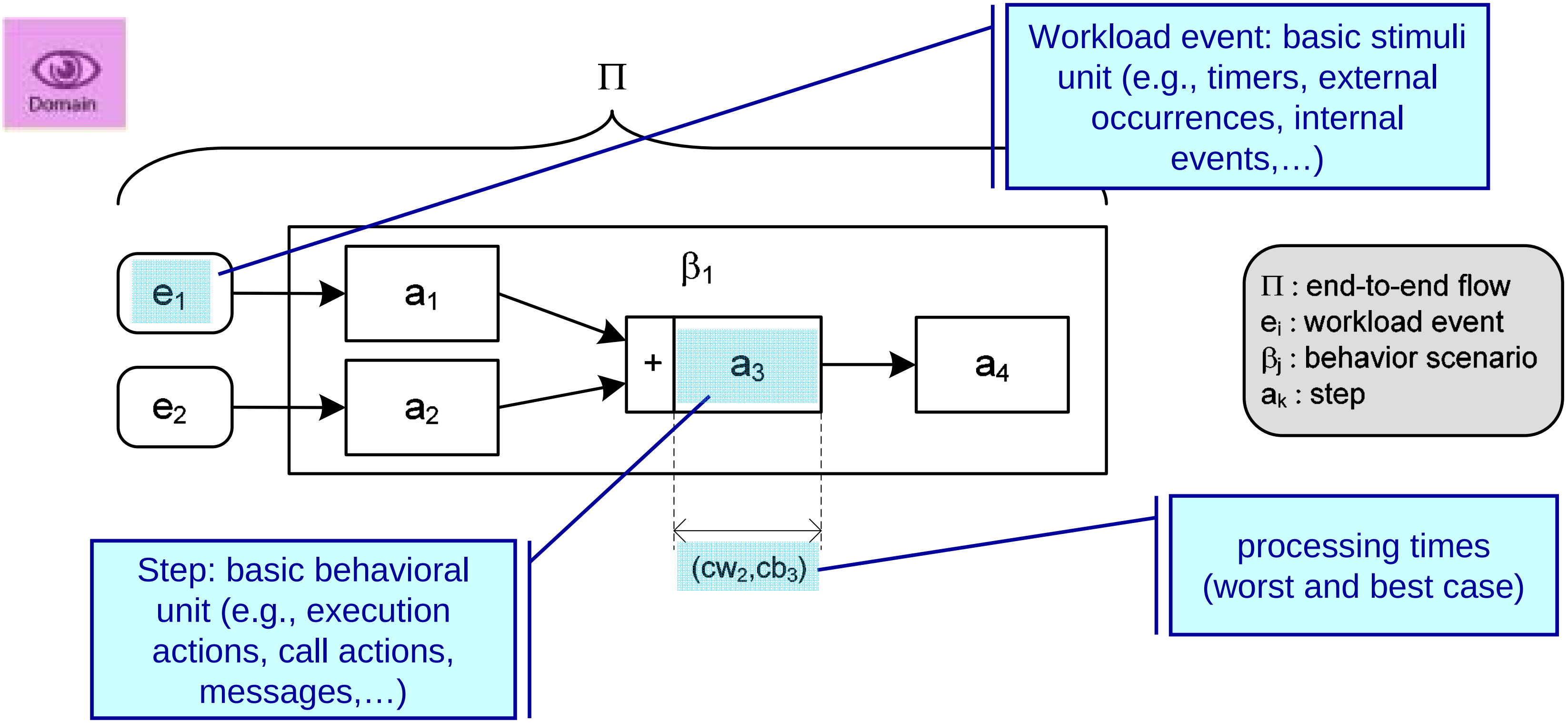
Other Sched. Analysis tools: Livedevices' Real-Time Architect, CoMET from VaST, Vector's CANAlyzer…

SAM: The Notion of End-To-End Flow

www.omgmarte.org

An "End-To-End Flow" is the basic workload unit to be evaluated by schedulability analysis tools.

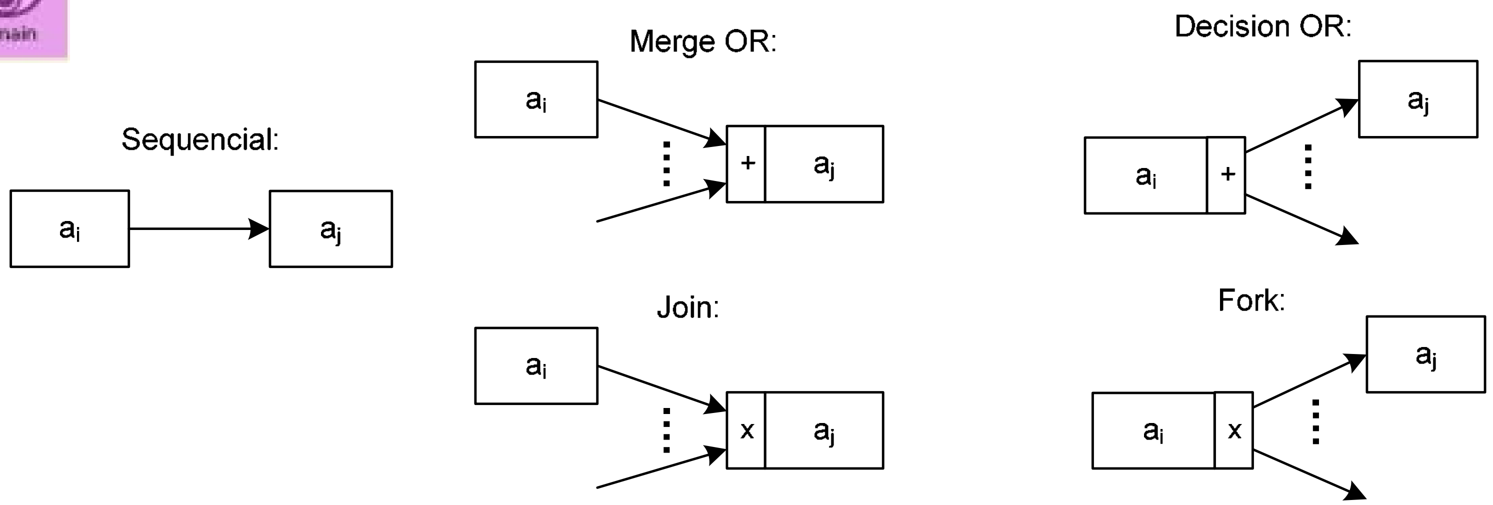
→ An end-to-end flow refers to the entire causal set of steps triggered by one or more external workload events.



SAM: Precedence Relations

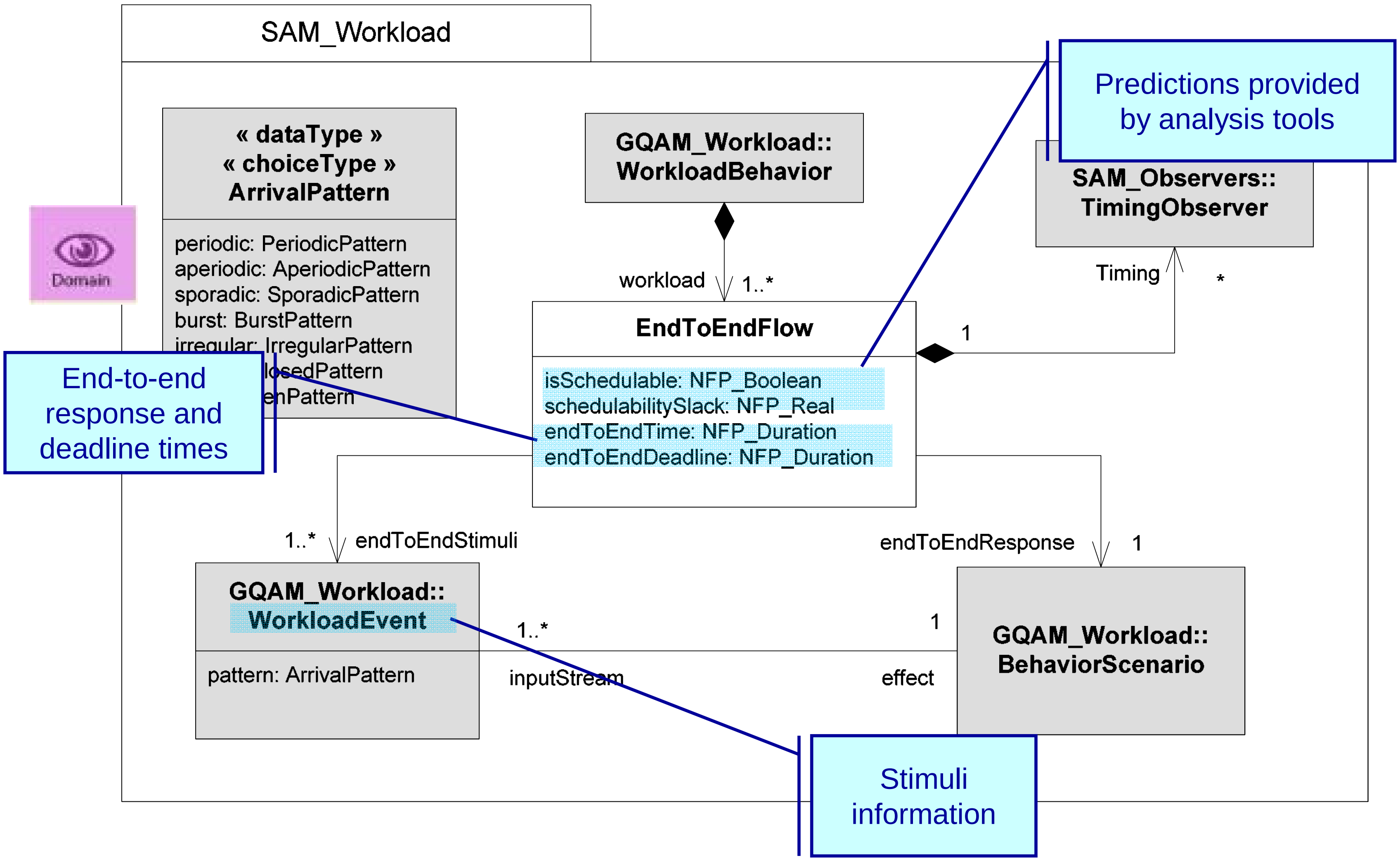
Execution and communication steps may be causally related by one of the following precedence relations:

Domain

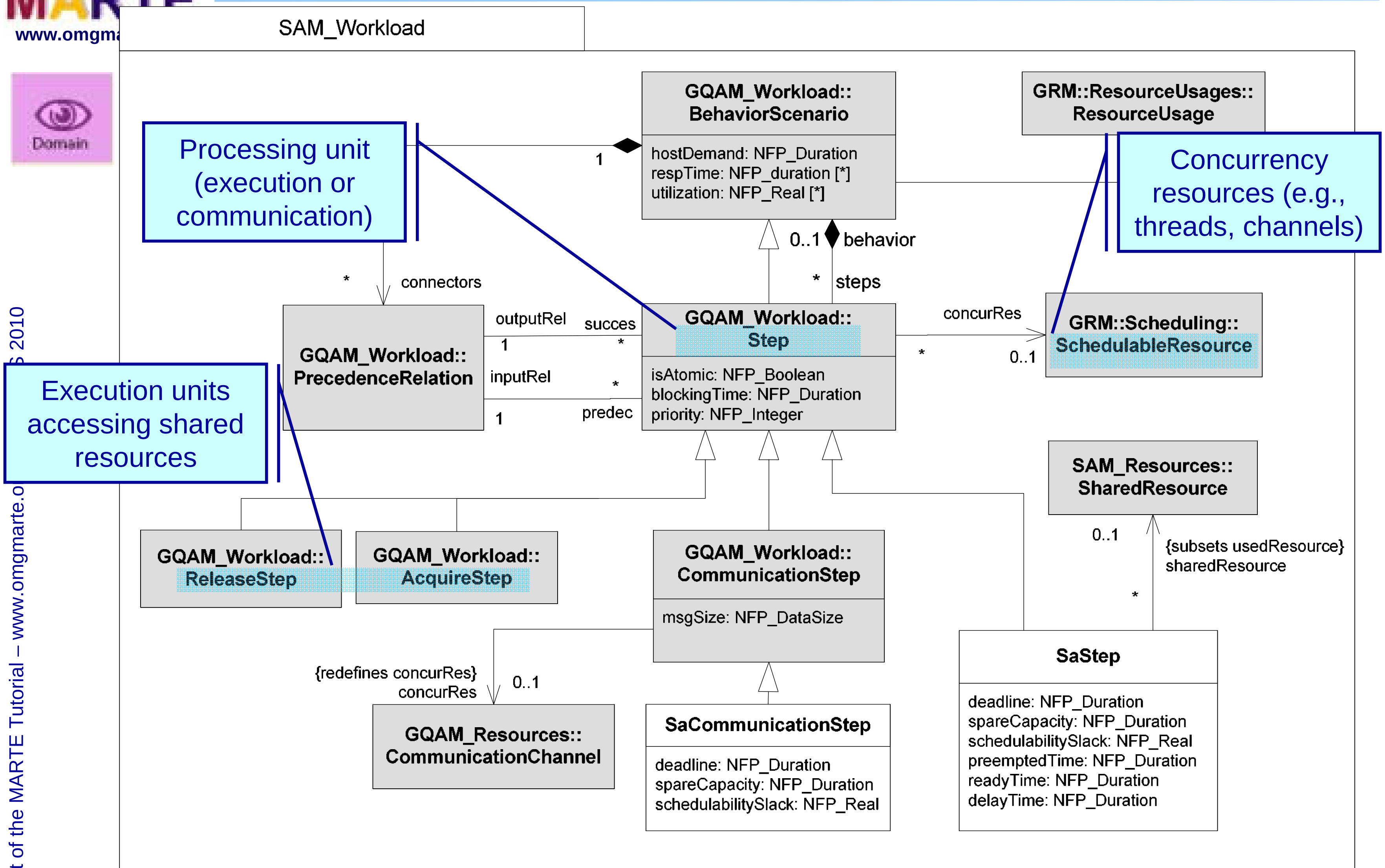


SAM: Workload Domain Metamodel (end-to-end)

www.omgmarte.org



SAM: Workload Domain Metamodel (detailed behav.)

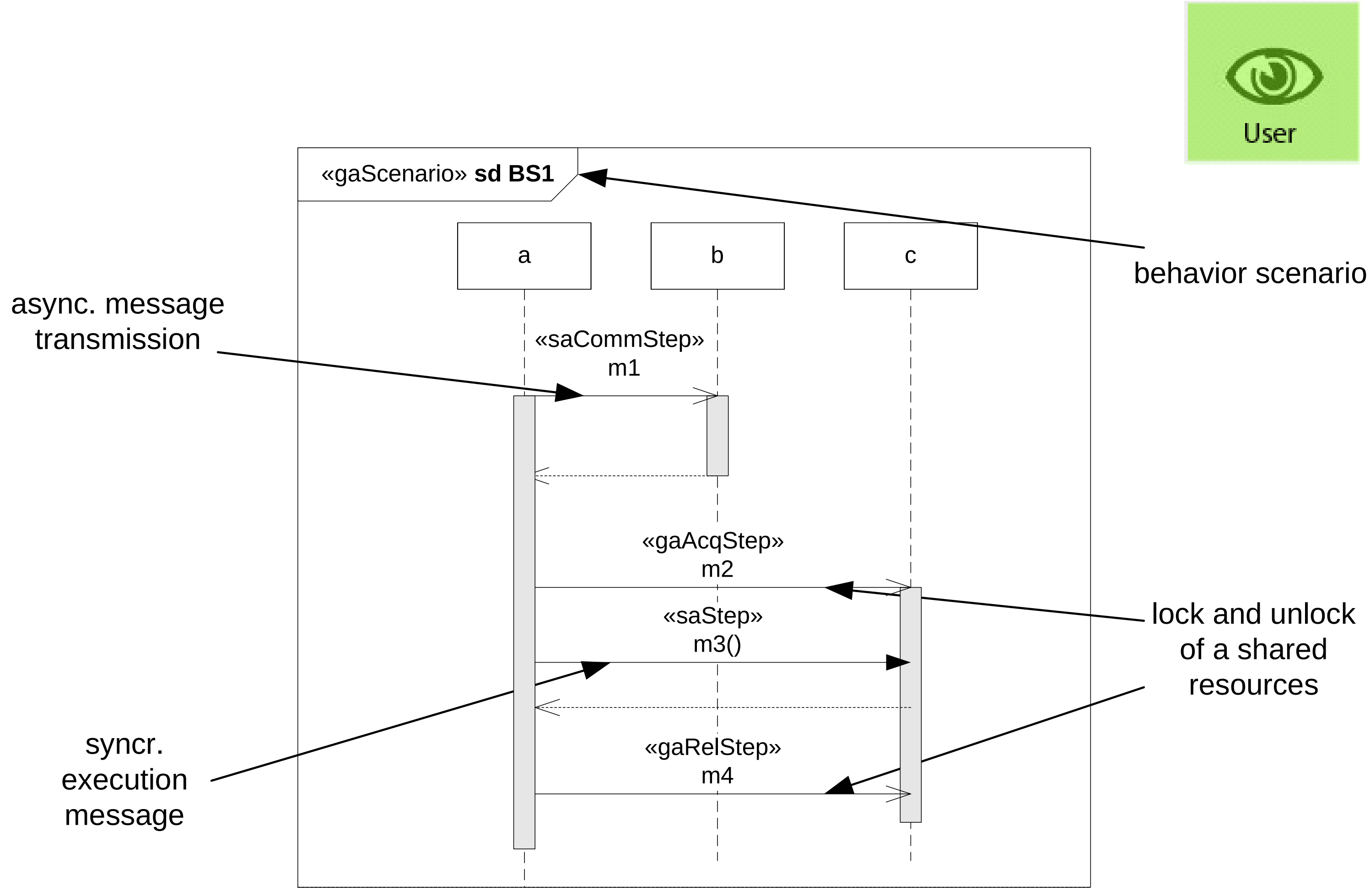


SAM: Example of Stereotype Extensions Usage

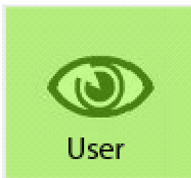
SAM Domain Model	SAM Stereotype	UML Metaclasses	Context
WorkloadBehavior	GaWorkloadBehavior	UML::Interactions::Fragments:: **CombinedFragments**	Modeled in a high-level interaction
EndToEndFlow	SaEnd2EndFlow	UML::Interactions::Fragments:: **InteractionOperand**	Modeled in a high-level interaction
WorkloadEvent	GaWorkloadEvent	UML::Interactions::BasicInteractions:: **Message**	Modeled in a high-level interaction
BehaviorScenario	GaScenario	UML::Interactions::BasicInteractions:: **Interaction**	Modeled as a low-level interaction nested within a higher-level interaction
Step **CommunicationStep** **ReleaseStep** **AcquireStep**	SaStep SaCommStep GaRelStep GaAcqStep	UML::Interactions::BasicInteractions:: **Message**	Messages in low-level interactions

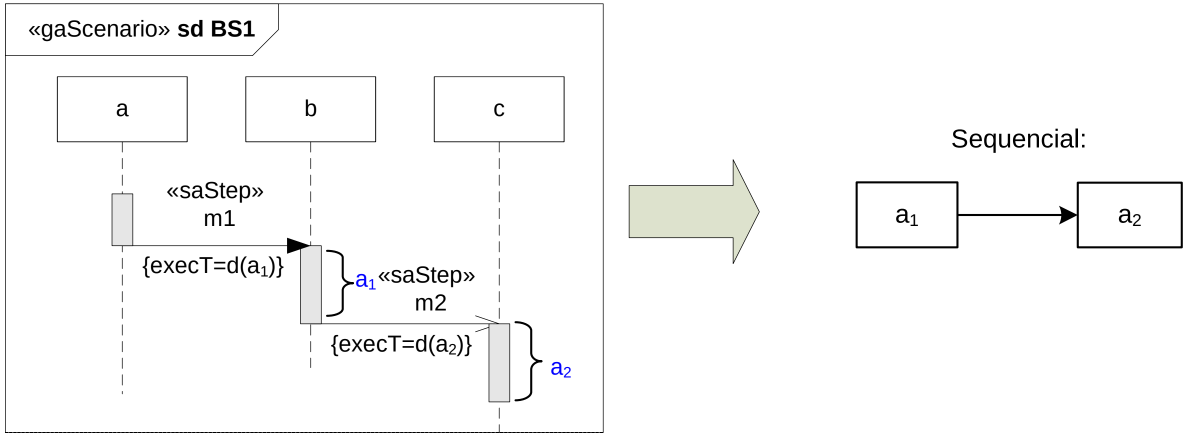
SAM: Examples of Behavior Annotations

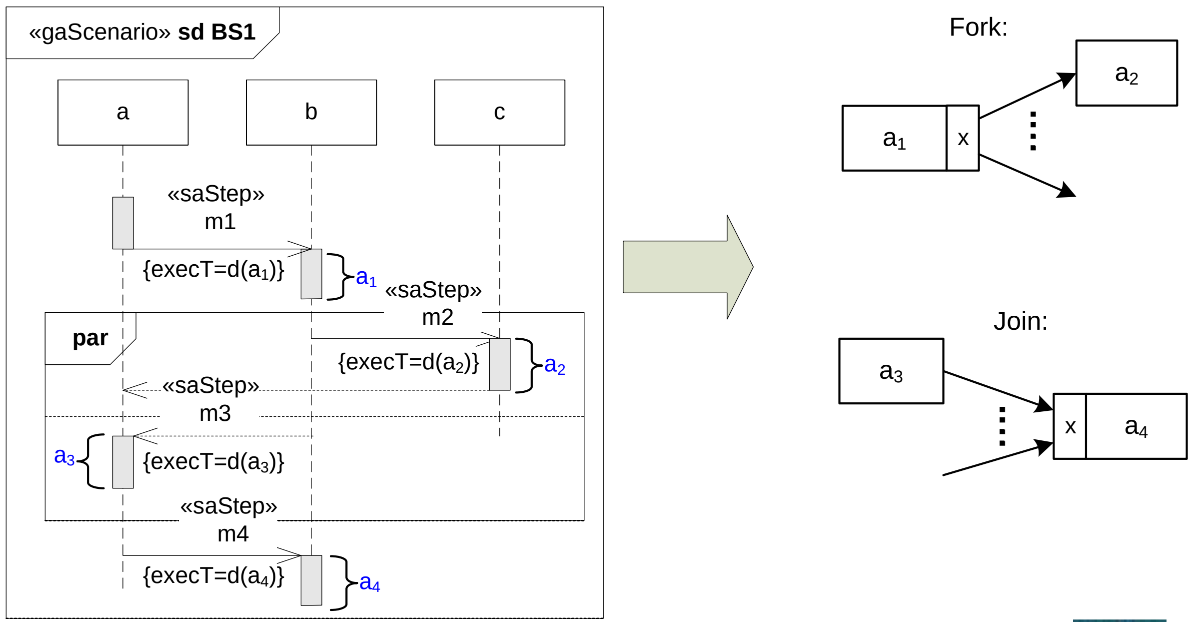
www.omgmarte.org



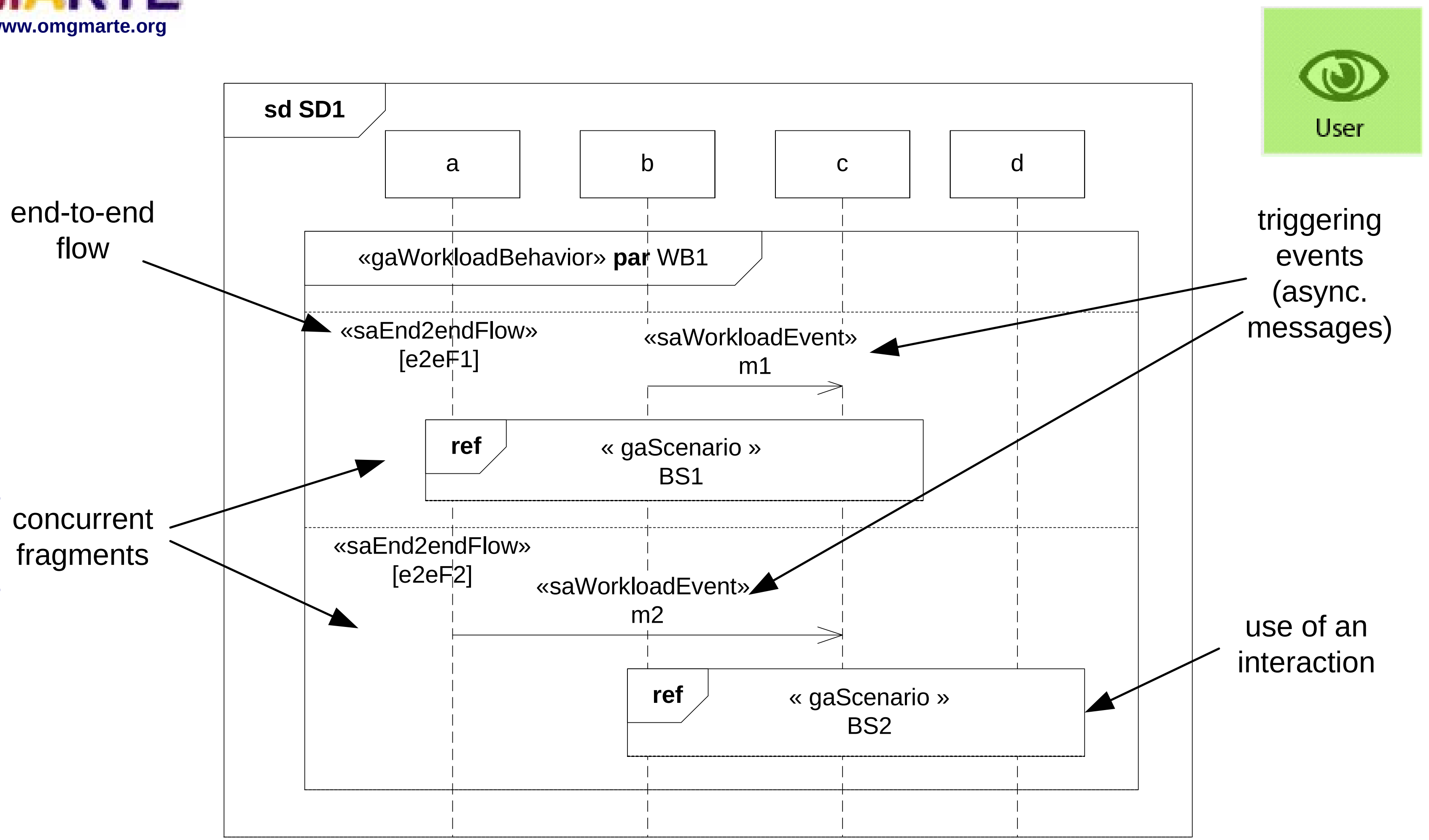
SAM: Example of Precedence Relations Annotation







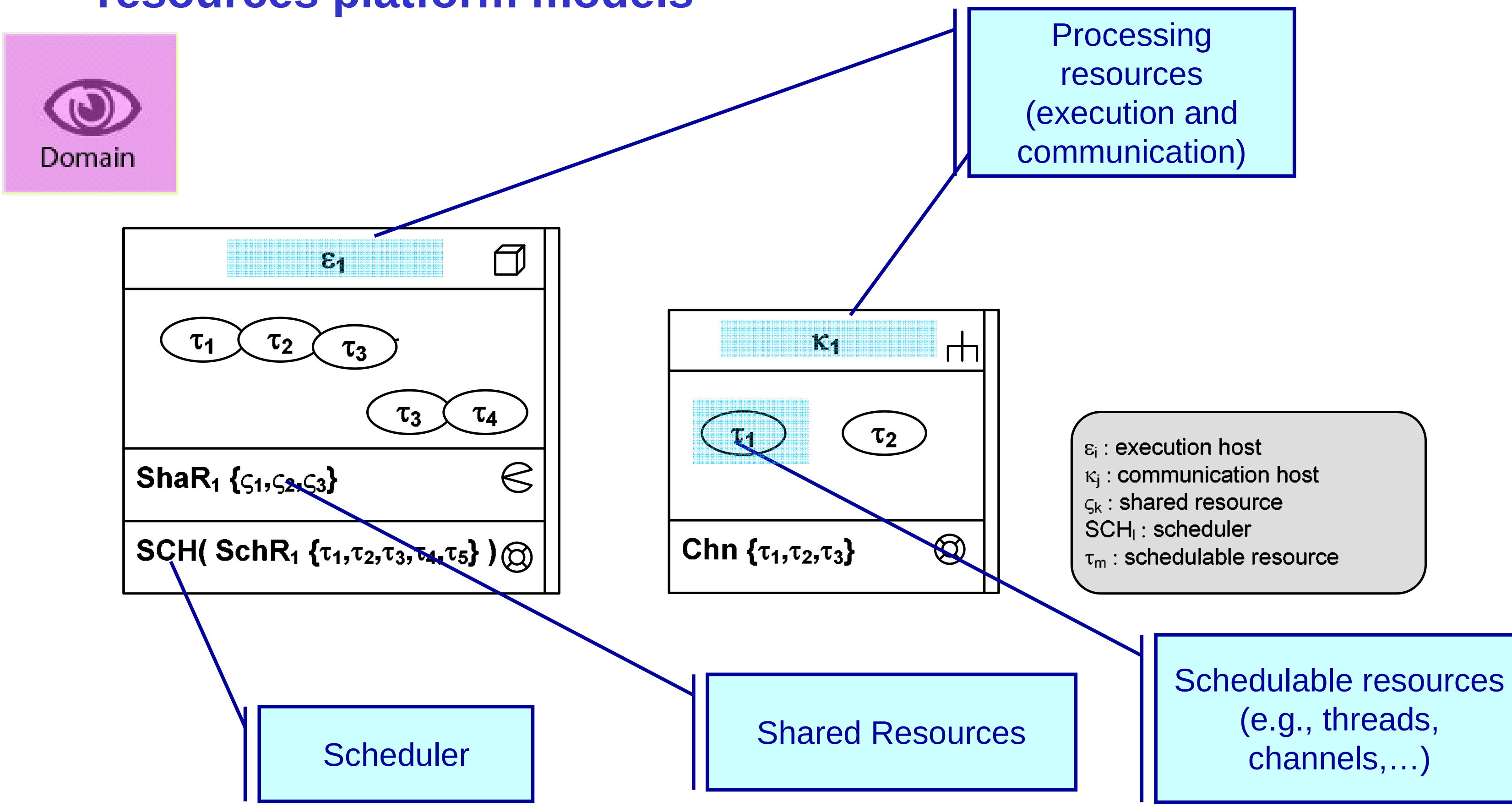
SAM: Example of Workload Annotations



UML MARTE
www.omgmarte.org

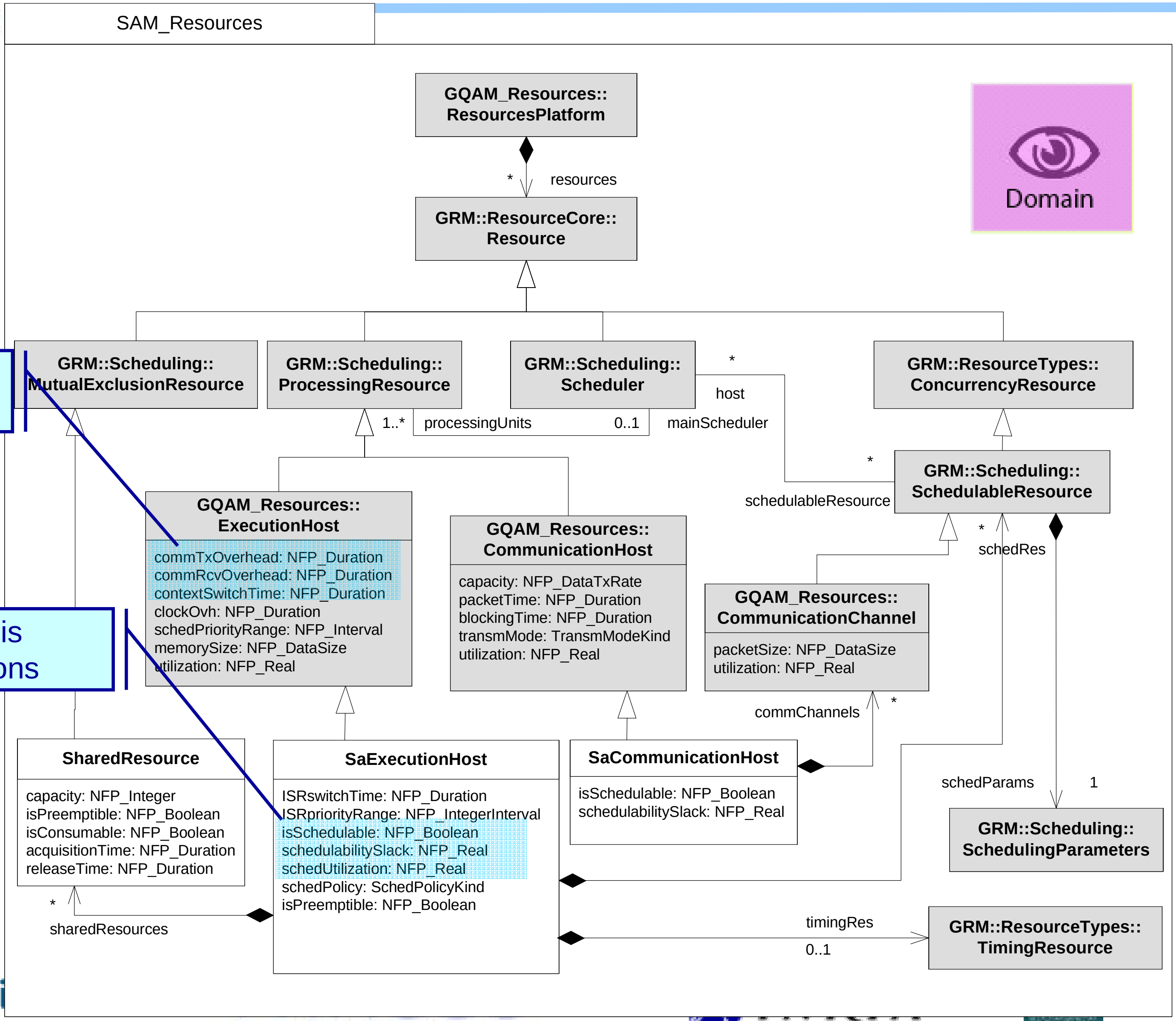
SAM: Resources Concepts

- **Provide additional (analysis-specific) annotations to annotate resources platform models**



cea list THALES INRIA UC

SAM: Resources Domain Metamodel



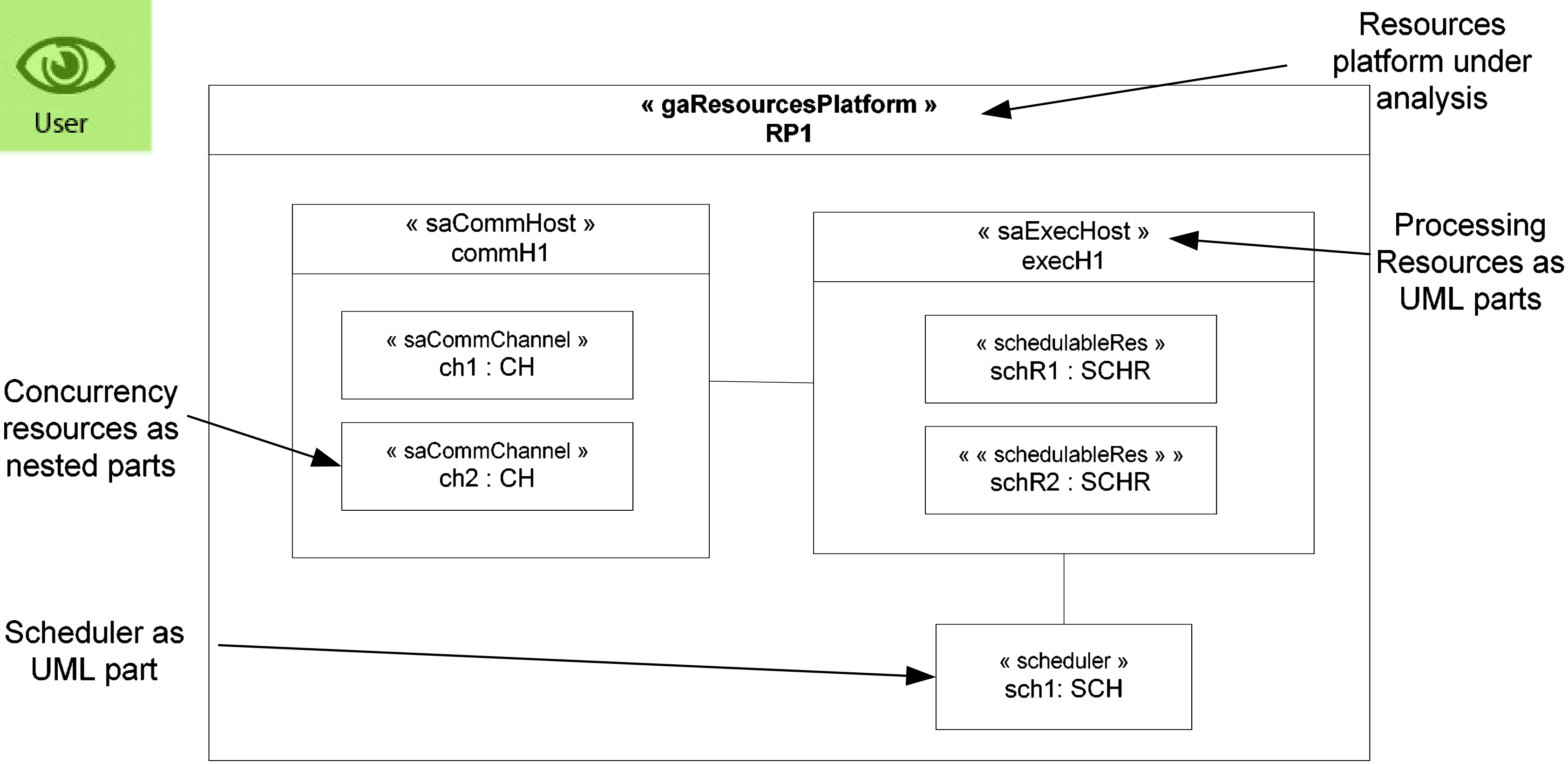
SAM: Examples of the Stereotypes Usage

UML

SAM Domain Model	SAM Stereotype	UML Metaclasses	Context
ResourcesPlatform	GaResourcesPlatform	UML::StructuredClasses:: **SctructuredClass**	Main container of resources
SaExecutionHost **SaCommunicationHost** **GRM::Scheduler**	SaExecHost SaCommHost Scheduler	UML:: StructuredClasses:: **Property**	Parts of the resources platform
GRM::SchedulableResource **SaCommChannel**	SchedulableRes SaCommChannel	UML:: StructuredClasses:: **Property**	Parts of processing resources

SAM: Example of Resources Stereotype Usage

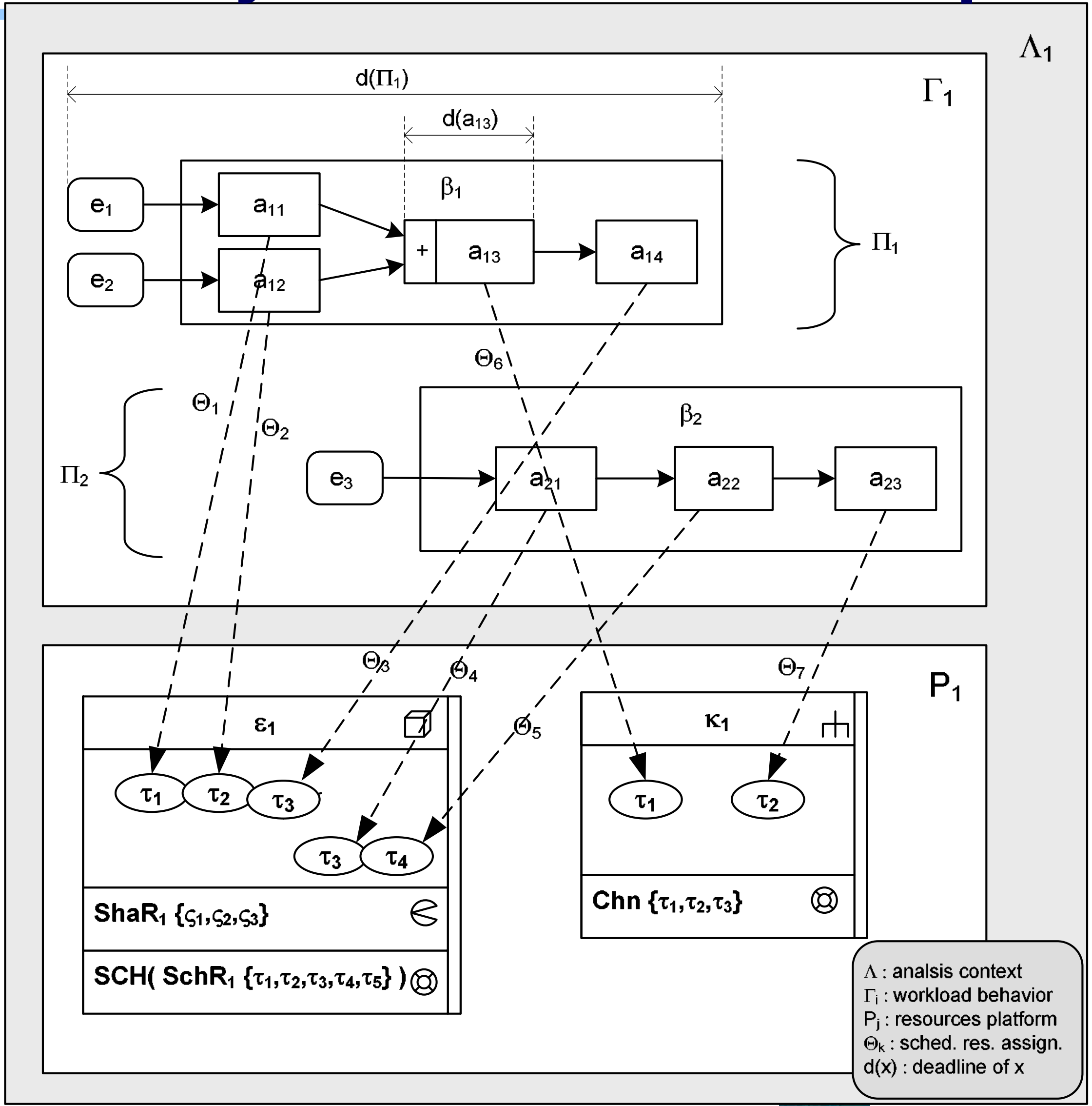
User



SAM: Analysis Context concepts

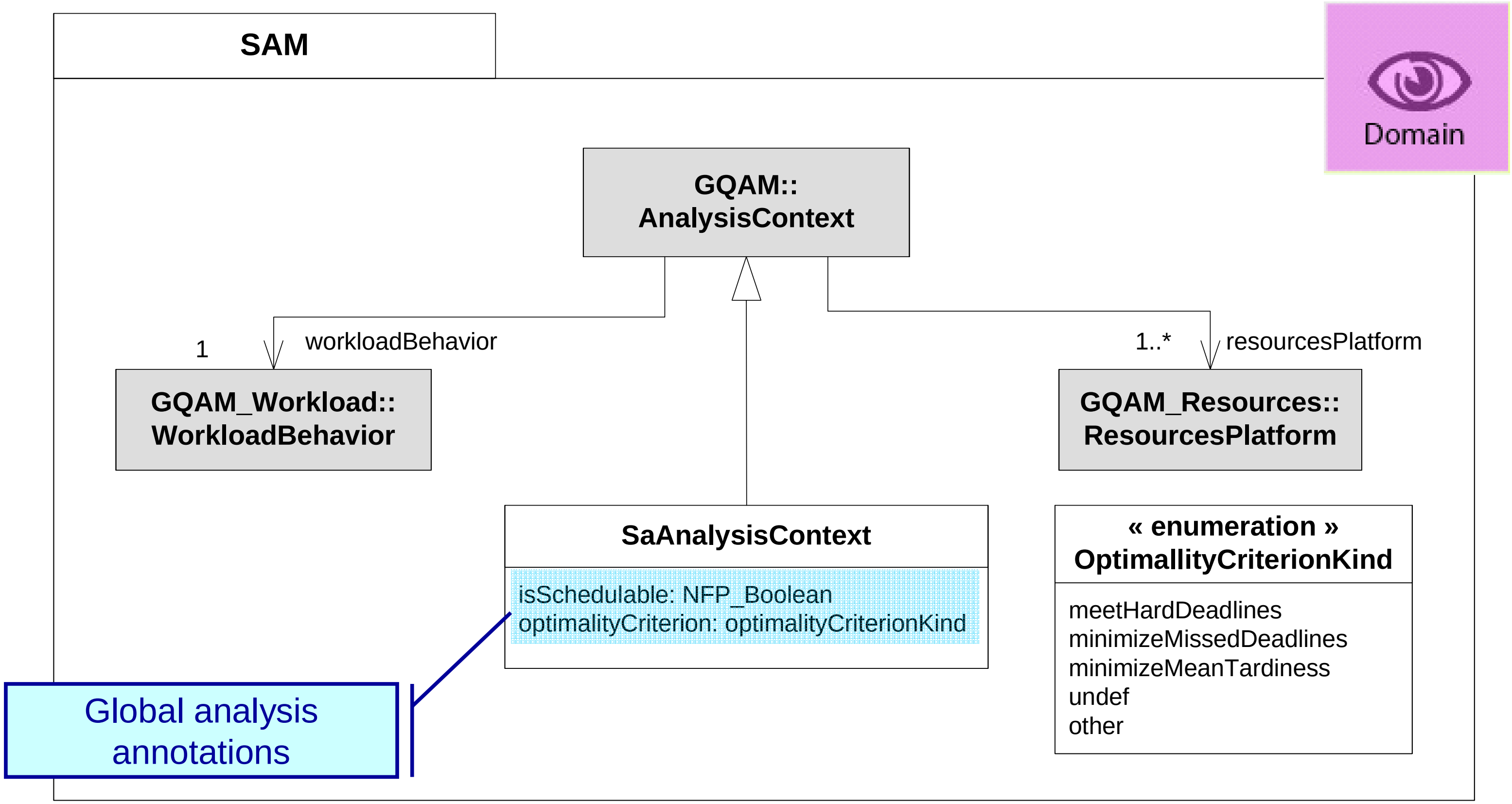
- An **analysis context** is the root concept used to collect relevant quantitative information for performing a specific analysis scenario.

- An **analysis context** integrates workload behavior models and resources platform models.

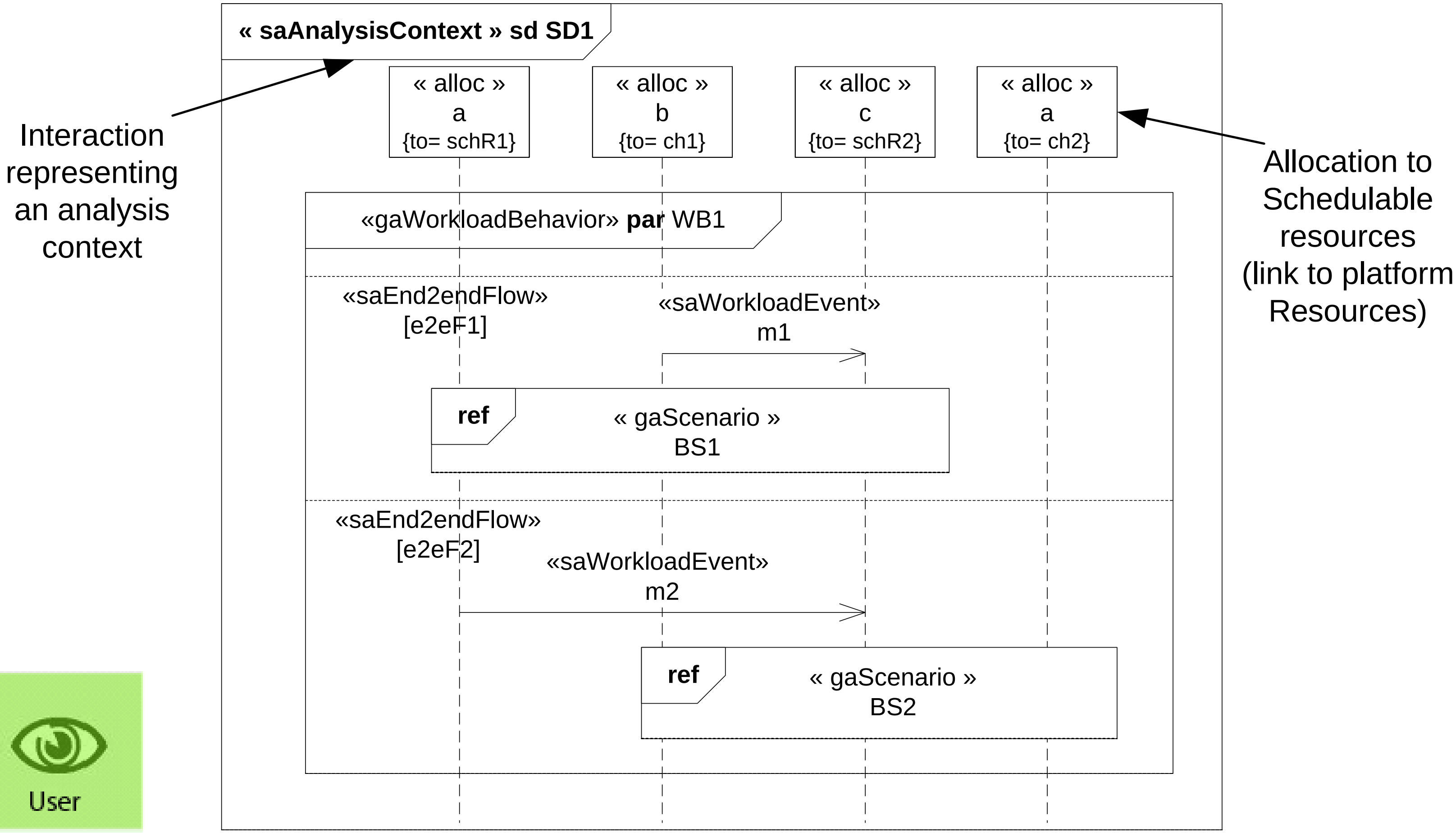


Domain

SAM: Analysis Context Domain Metamodel

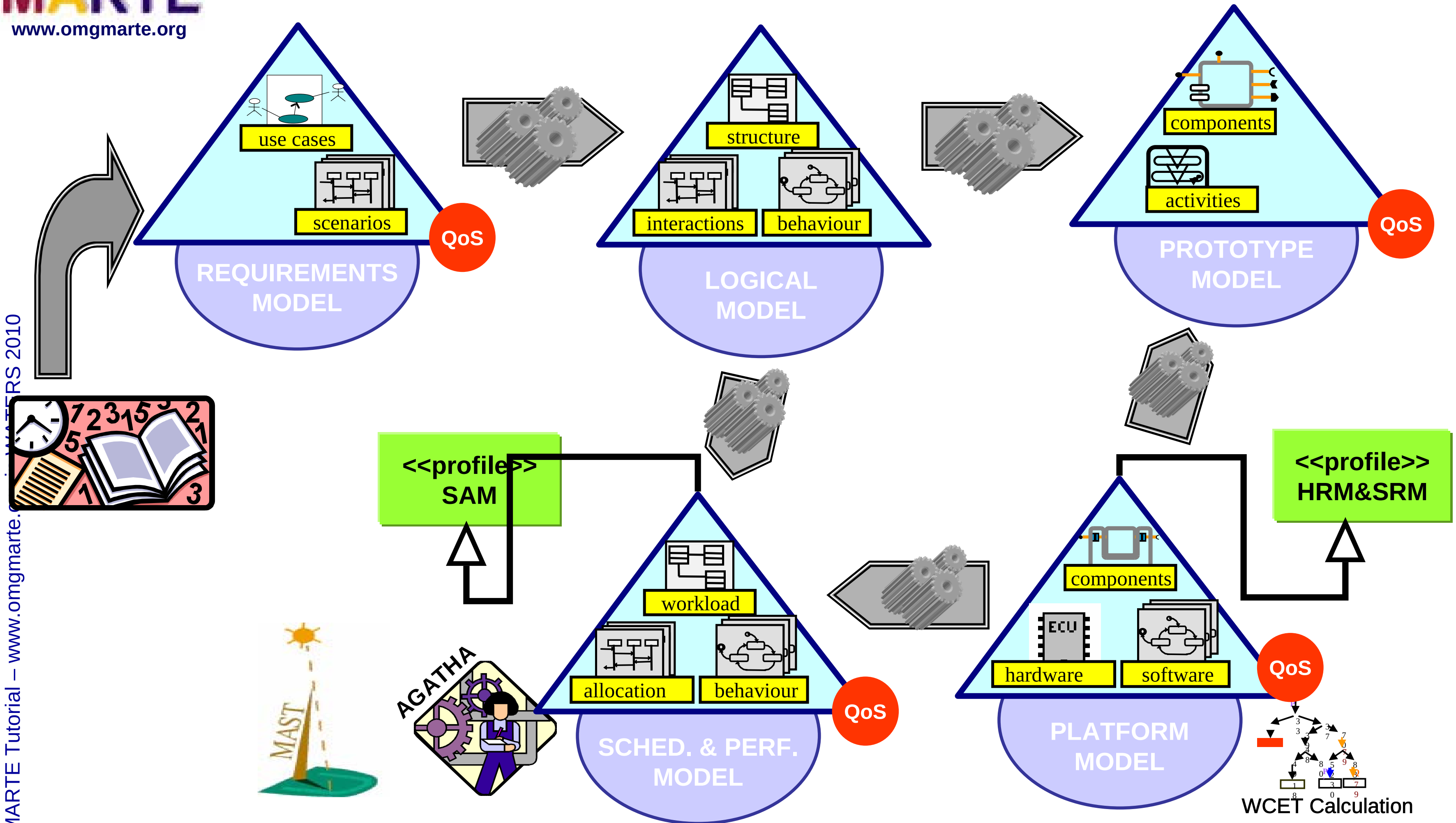


SAM: Example of Analysis Context Stereotype Applic.

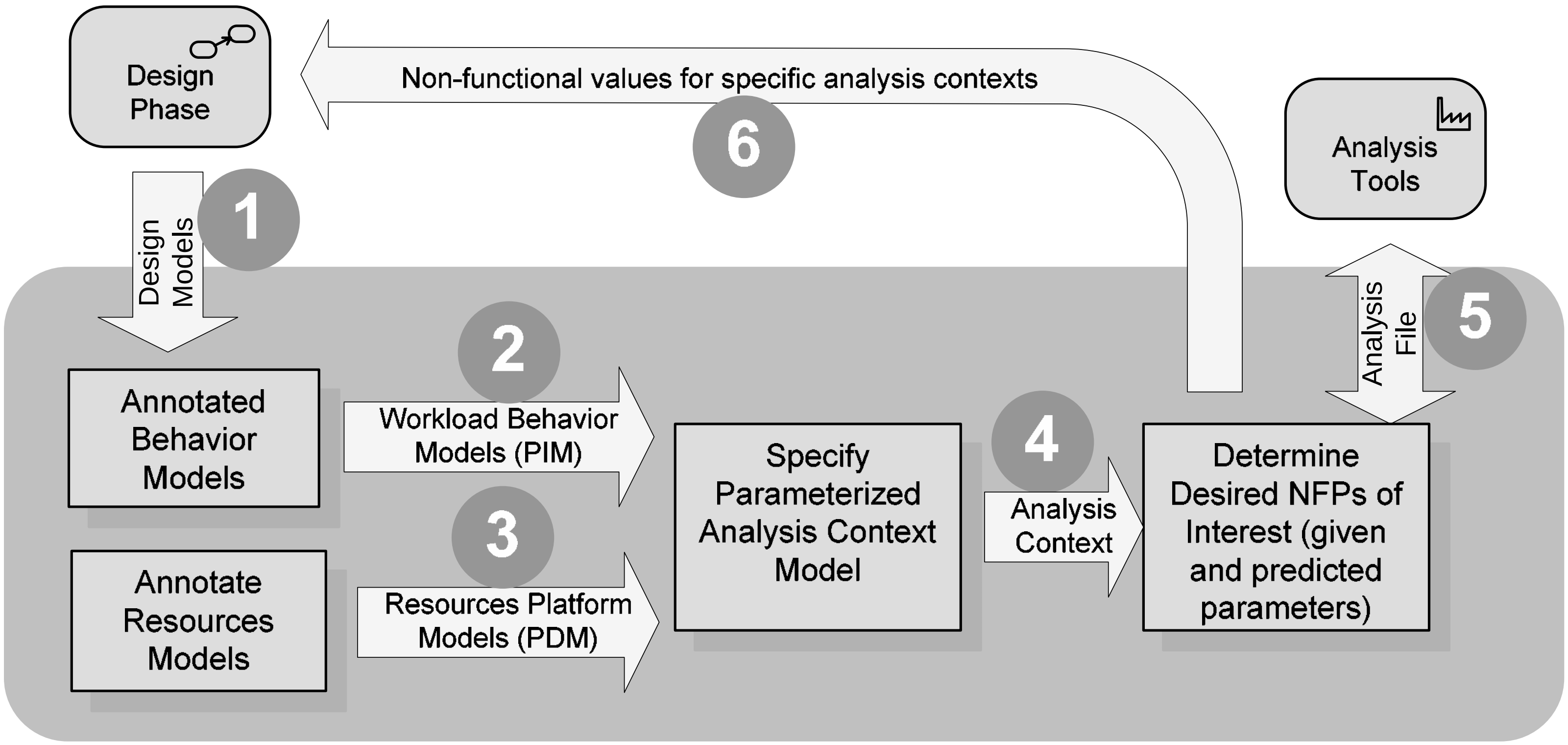


User

Example of Global Development Process



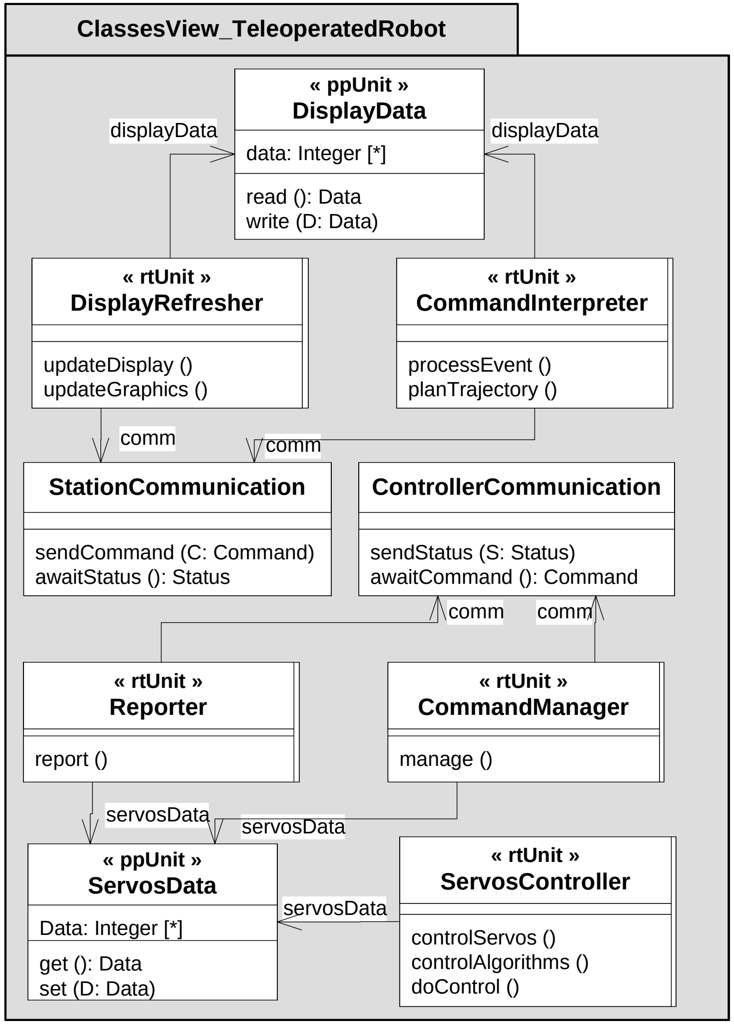
General Procedure to Use the SAM Profile

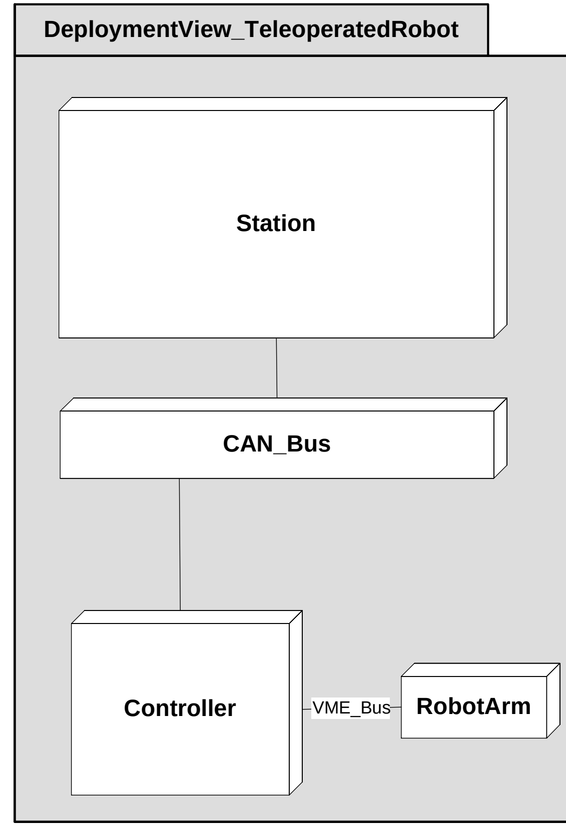


UML MARTE
www.omgmarte.org

Example: A Teleoperated Robot

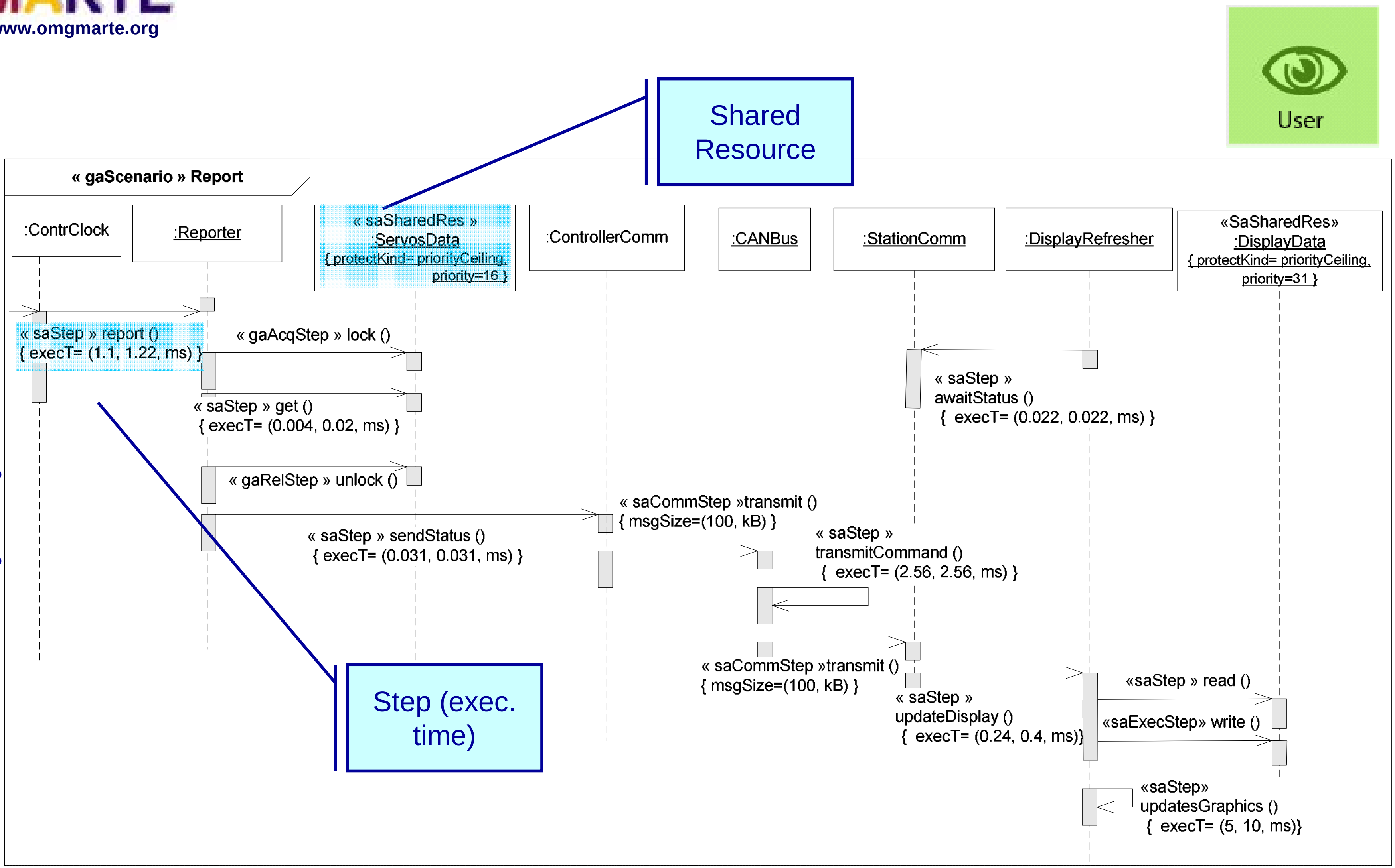
User





Example of Annotated Scenario with SAM

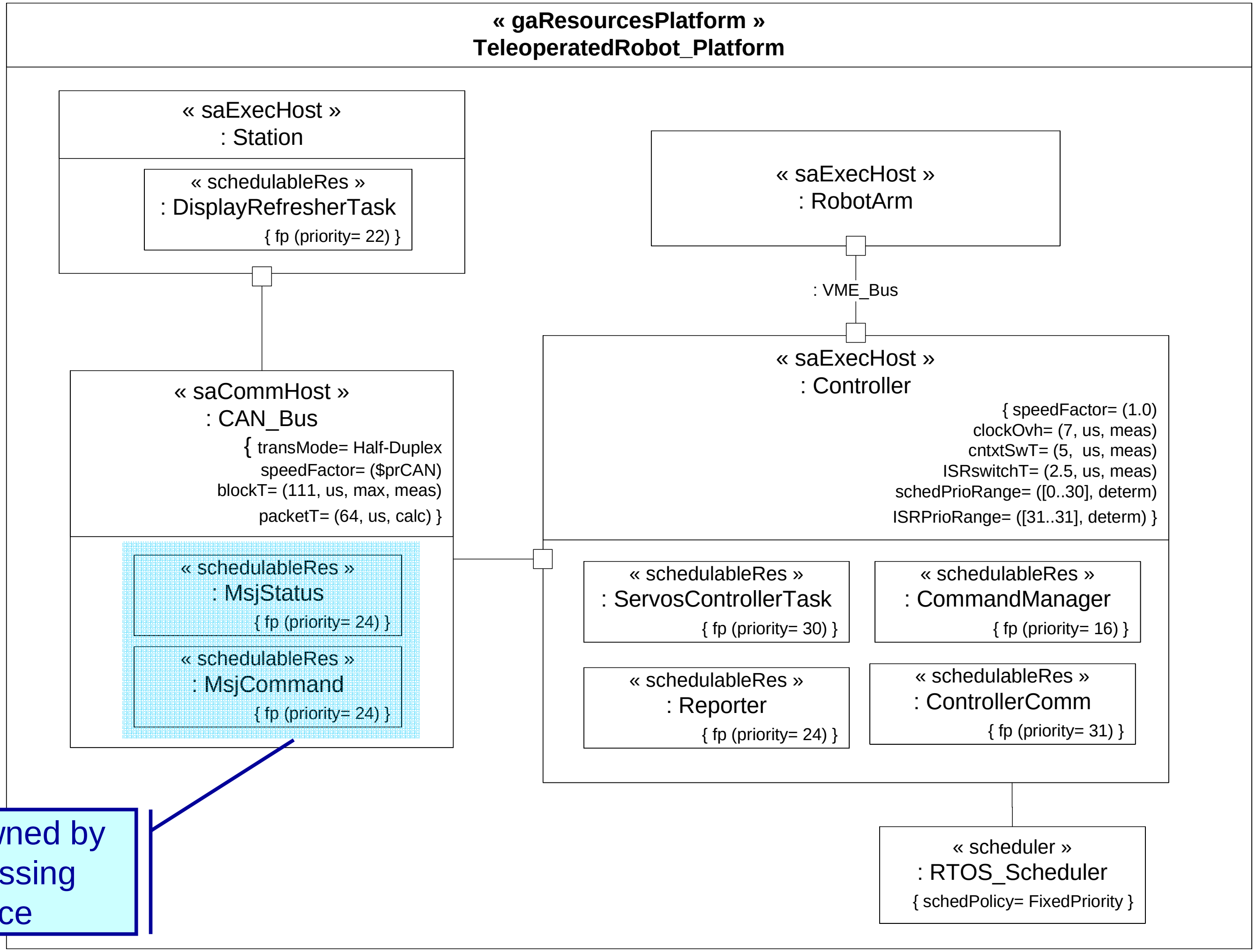
UML MARTE
www.omgmarte.org



cea list THALES INRIA UC

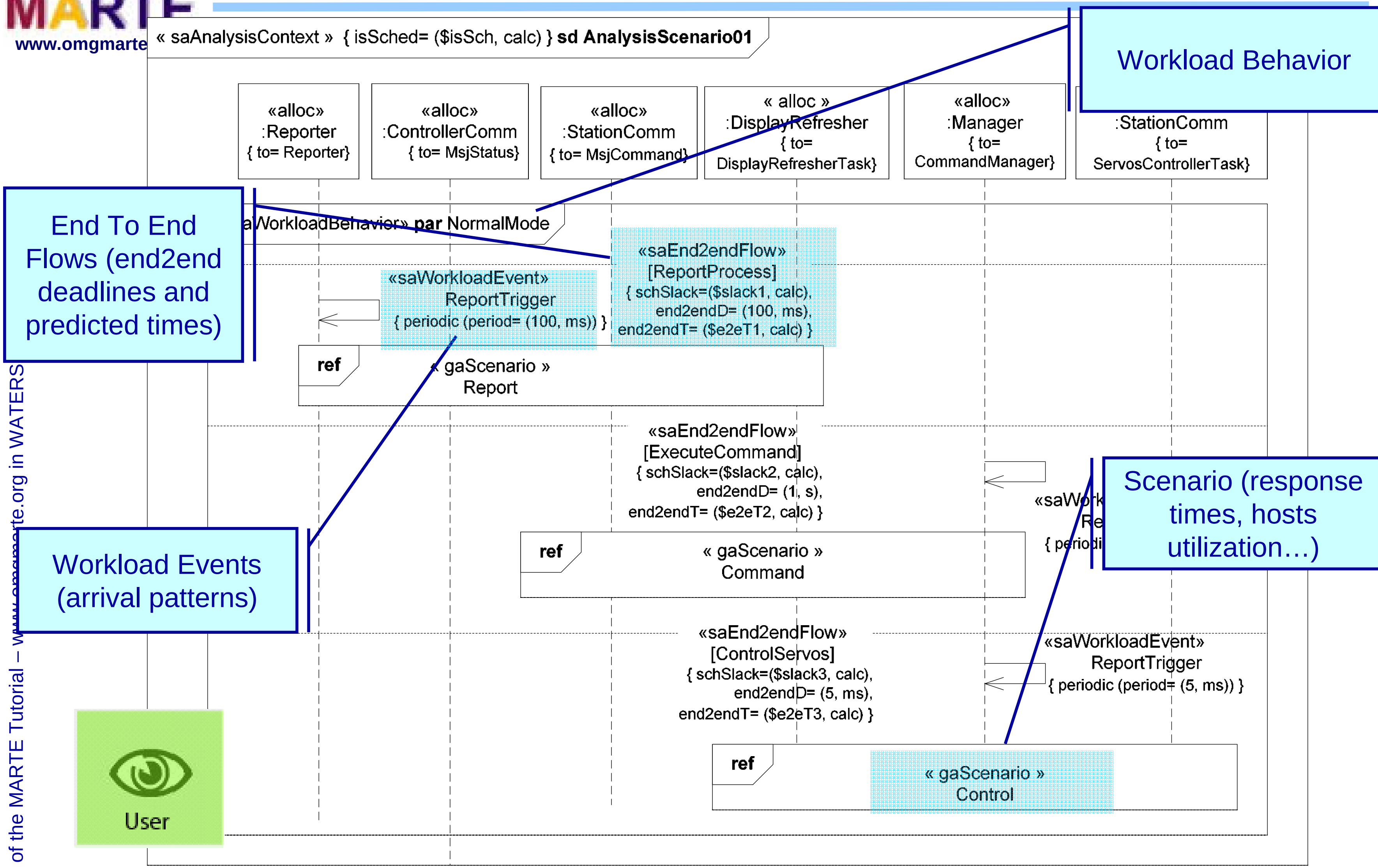
User

Example of Annotated Resources Model with SAM

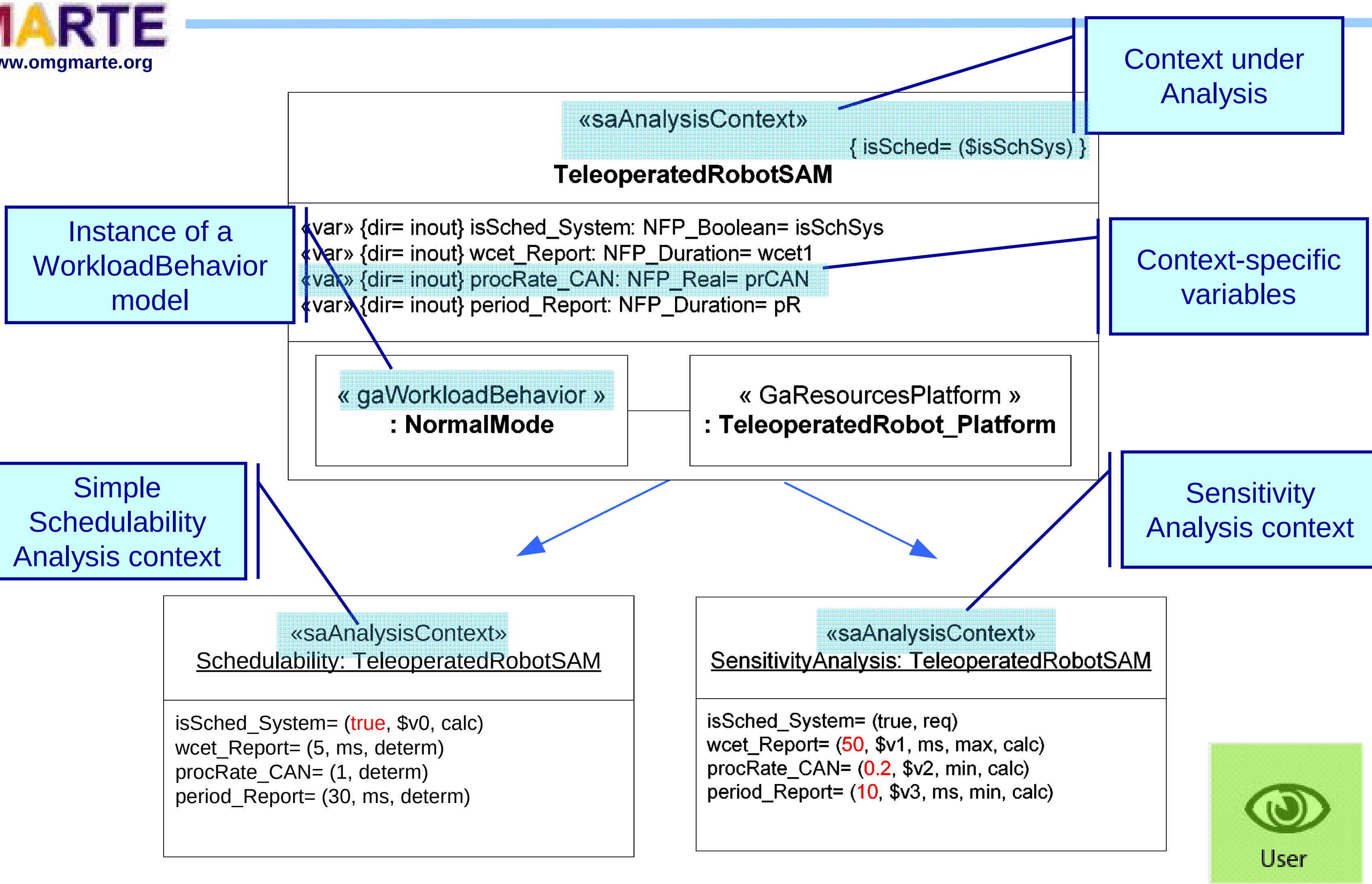


Threads owned by the processing resource

Example of Analysis Context Model



Example of Parametric Analysis Context



MARTE Tooling

- **Current Implementations supporting MARTE**
  - Full MARTE Profile & Libraries for Eclipse UML2
  - VSL edition assistant and type checker as a Eclipse plug-in for the UML Papyrus tool and RSA 7.0

- **On-going work:**
  - Eclipse plug-ins to transform UML models annotated with the SAM profile to input files of MAST, SymTA/S, Cheddar and RapidRMA tools

MARTE Open Source Implementation in

UML Papyrus: **www.papyrusuml.org**
IBM RSA: **www.omgmarte.org**

Conclusions on MARTE's Analysis

- **Industrial Use of V&V can benefits from MDE**
  - Analysis task must be cohesively integrated with Design tasks
  - Application of individual analysis techniques should be regarded as an essential part of an integrated V&V methodology

- **Methodological support is still under way:**
  - Complex analysis scenarios for Interface-Based Design, Multiobjective Design Space Exploration…
  - Means to manage NFP measurement models
  - Methods to map/transform MoCCs into analysis models

Elements for discussion

- Techniques comparison
- Models
  - Semantics
  - Syntax
  - Automation
- Tools
- Are all these domain specific?
  Or they are simply different flavors of complexity management