# The UML Profile for MARTE: modelling predictable real-time systems with UML

Julio Medina
Universidad de Cantabria, Spain
julio.medina[at]unican.es

(Based on extracts from the MARTE Tutorial:
http://www.omgmarte.org/sites/default/files/TutorialMARTE-Final_version1_1_0.pdf

# General Requirements of the MARTE RFP

- Proposals shall support Modeling and Analysis of Real-Time and Embedded (in short MARTE) systems including its software and hardware aspects.
  - The Proposals will define a metamodel and its underlying UML profile.
  - It shall be possible to use independently software and hardware parts of the profile.
  - It shall comply with existing standards (UML2)
  - It shall update the SPT profile 1.1

Ref. of MARTE RFP (realtime/05-02-06):
http://www.omg.org/cgi-bin/doc?realtime/05-02-06

# The ProMARTE Team

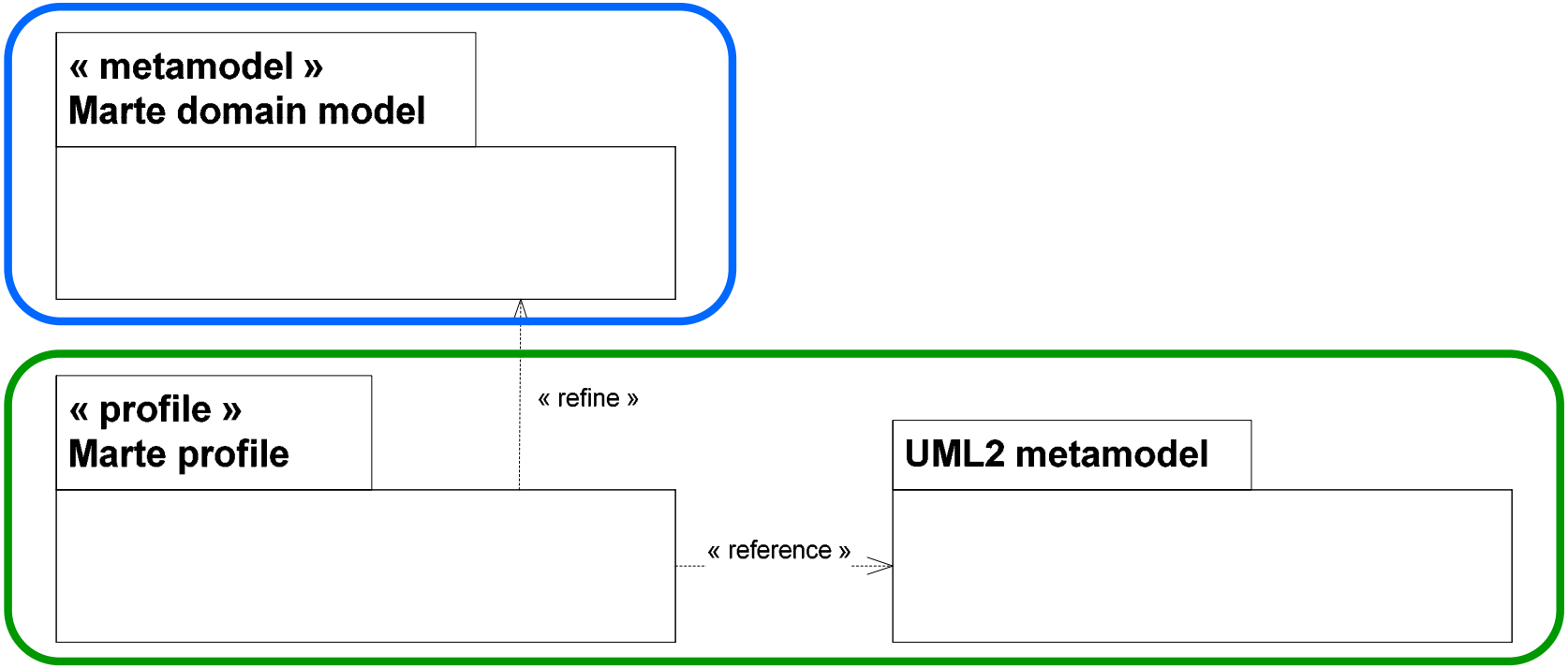
- Partners
  - Industrials
    - Alcatel*
    - Lockheed Martin*
    - Thales*
  - Tool vendors
    - ARTISAN Software Tools*
    - International Business Machines*
    - Mentor Graphics Corporation*
    - Softeam*
    - Telelogic AB (I-Logix*)
    - Tri-Pacific Software
    - France Telecom
    - No Magic
    - Mathworks
  - Academics
    - Carleton University
    - Commissariat à l'Energie Atomique
    - ESEO
    - ENSIETA
    - INRIA
    - INSA from Lyon
    - Software Engineering Institute (Carnegie Mellon University)
    - Universidad de Cantabria

* Submitter to OMG UML Profile for MARTE RFP

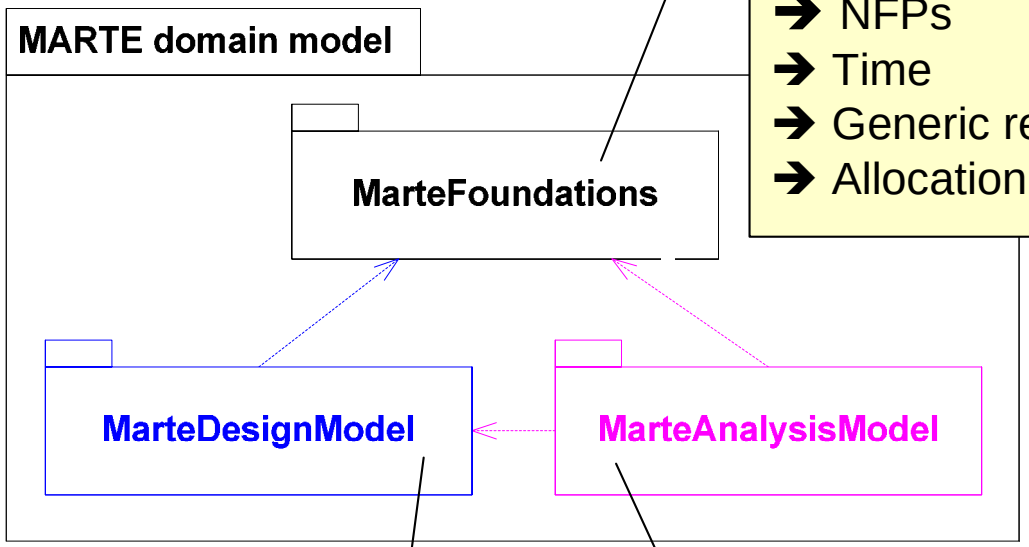
# Design Pattern Adopted for the MARTE Profile

- Stage 1 → Description of MARTE domain models
  - Purpose: Formal description of the concepts required for MARTE
  - Techniques: Meta-modeling

- Stage 2 → Mapping of MARTE domain models towards UML2
  - Purpose: MARTE domain models design as a UML2 extensions
  - Techniques: UML2 profile



# 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

# How to read, use, & implement MARTE

(see Section 2)

Extension Units

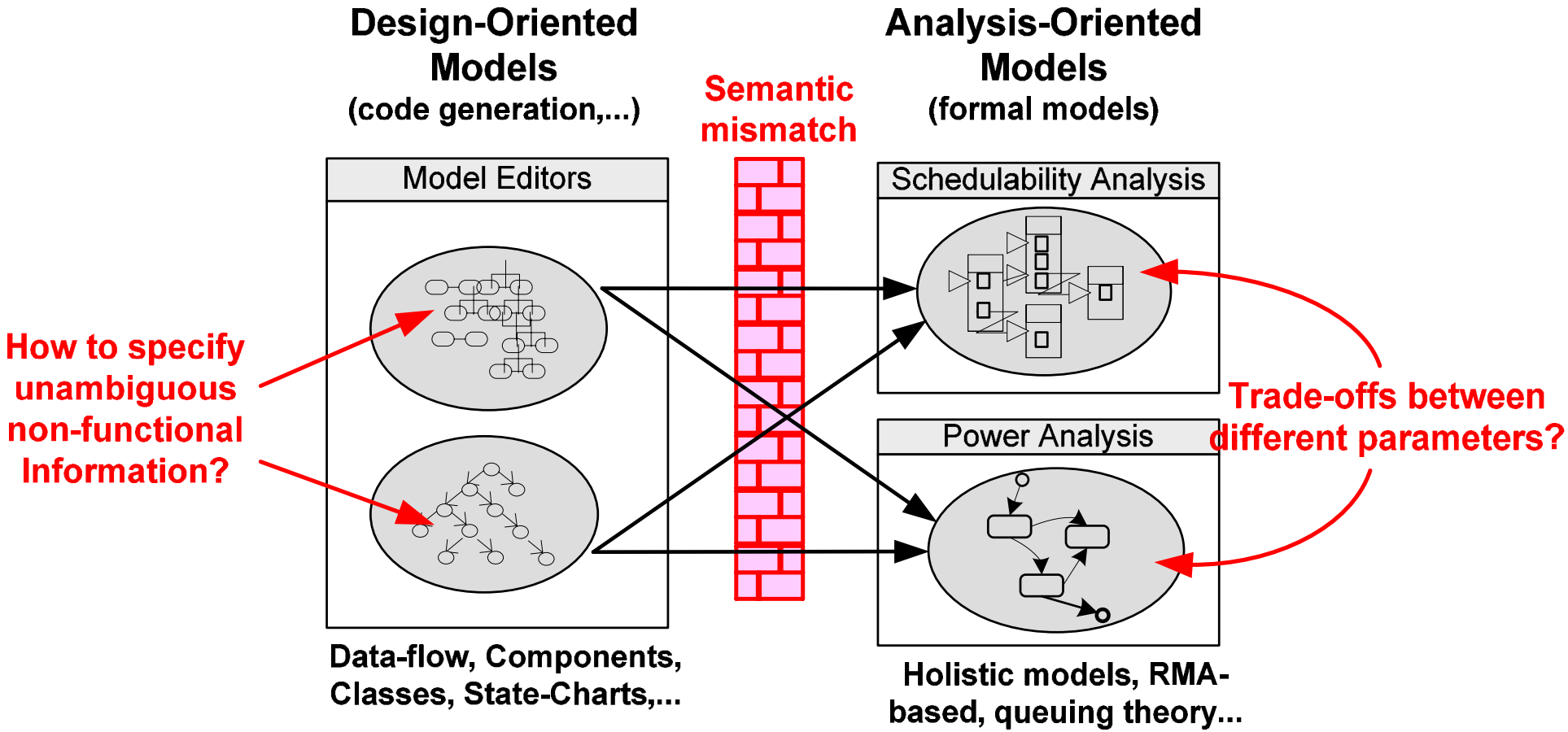
| | | |
|---|---|---|
| NFP | Non-Functional Properties | Section 8 |
| Time | Enhanced Time Modeling | Section 9 |
| GRM | Generic Resource Modeling | Section 10 |
| Alloc | Allocation Modeling | Section 11 |
| GCM | Generic Component Model | Section 12 |
| HLAM | High-Level Application Modeling | Section 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 |

# 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 |

# Needs for Model-Based RTES Analysis



# 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

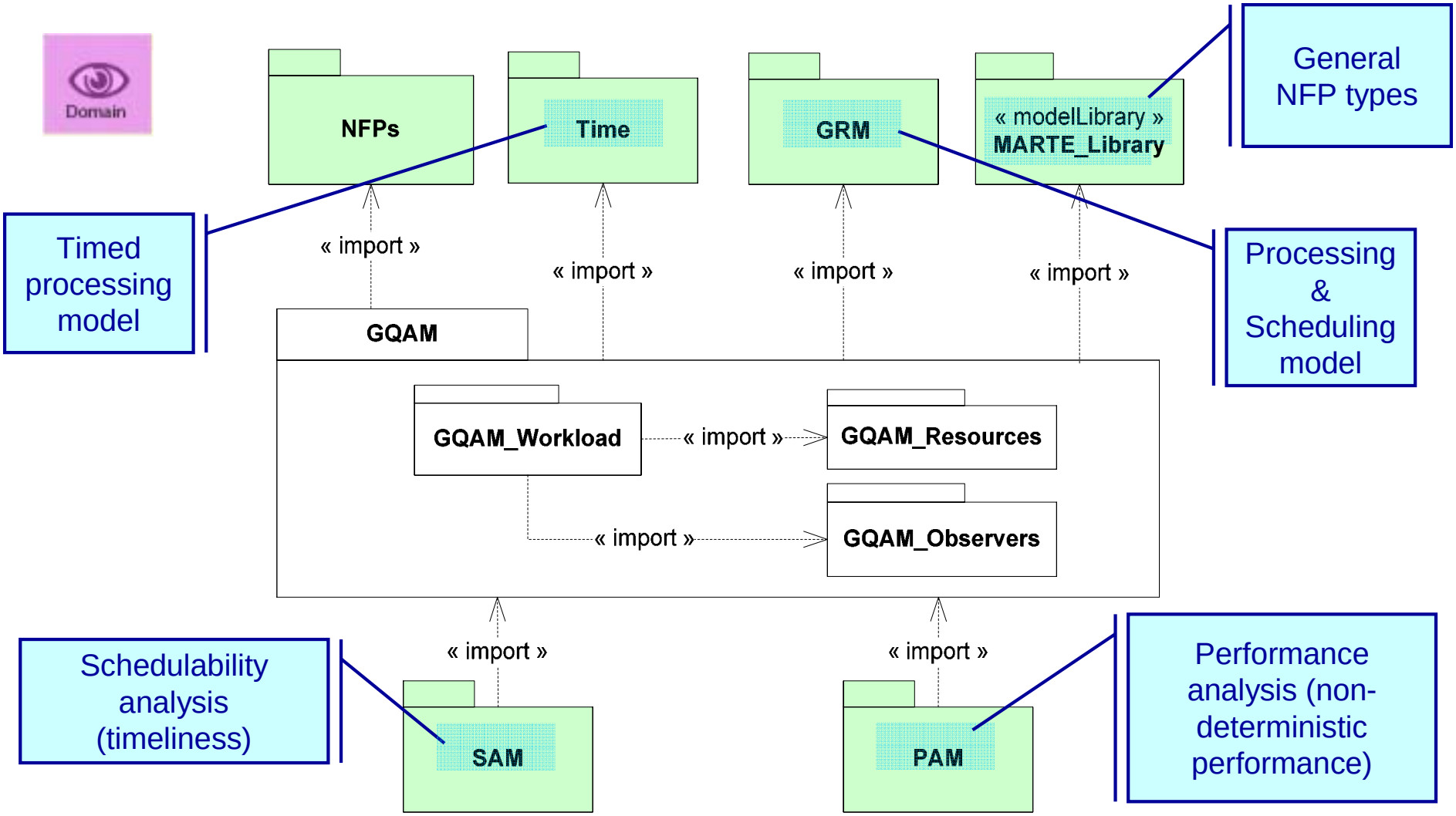
# UML-Based Analysis Foundations

- 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

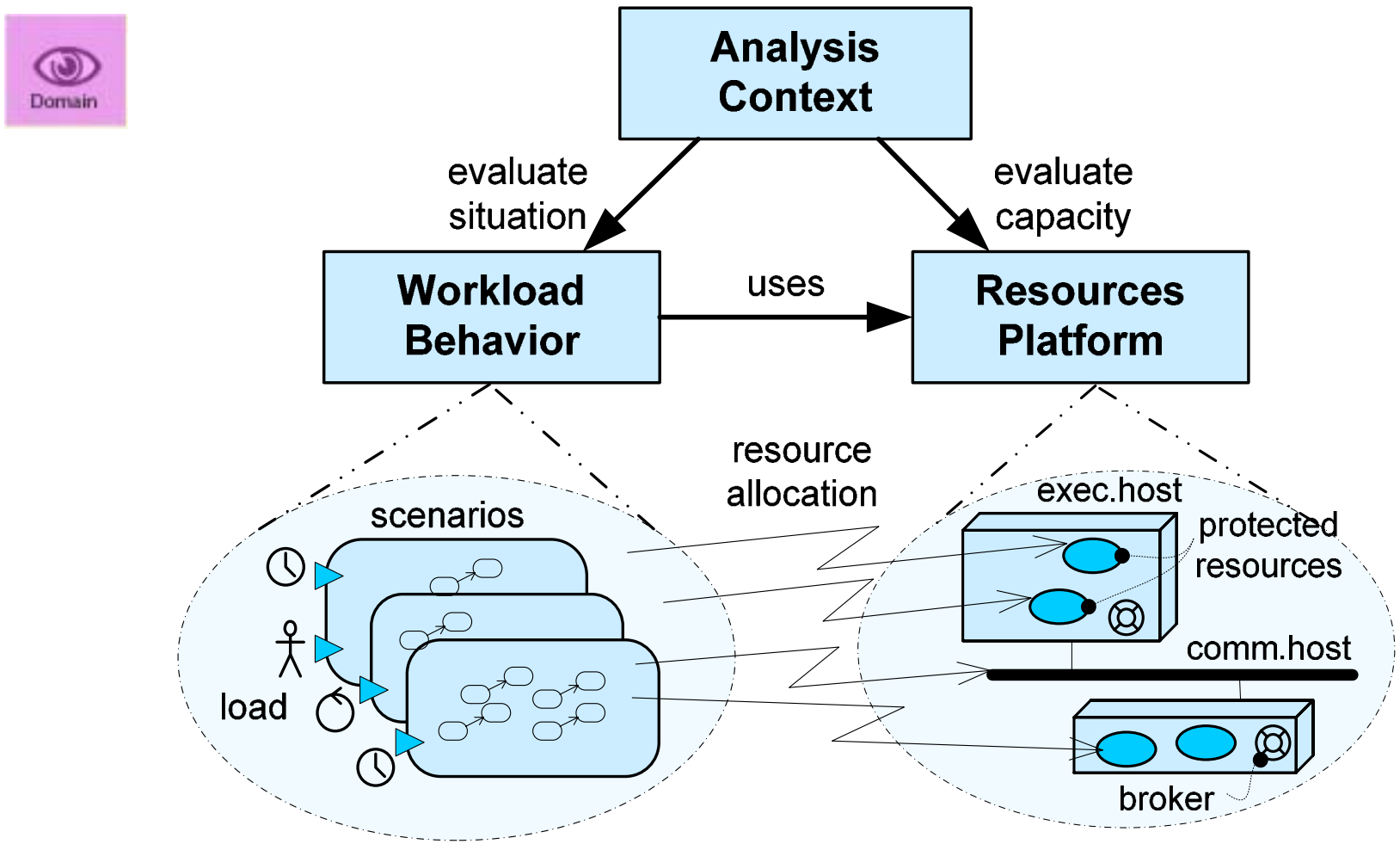
**The MARTE analysis sub-profiles provide standard constructs to map UML models on well-established analysis techniques**

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

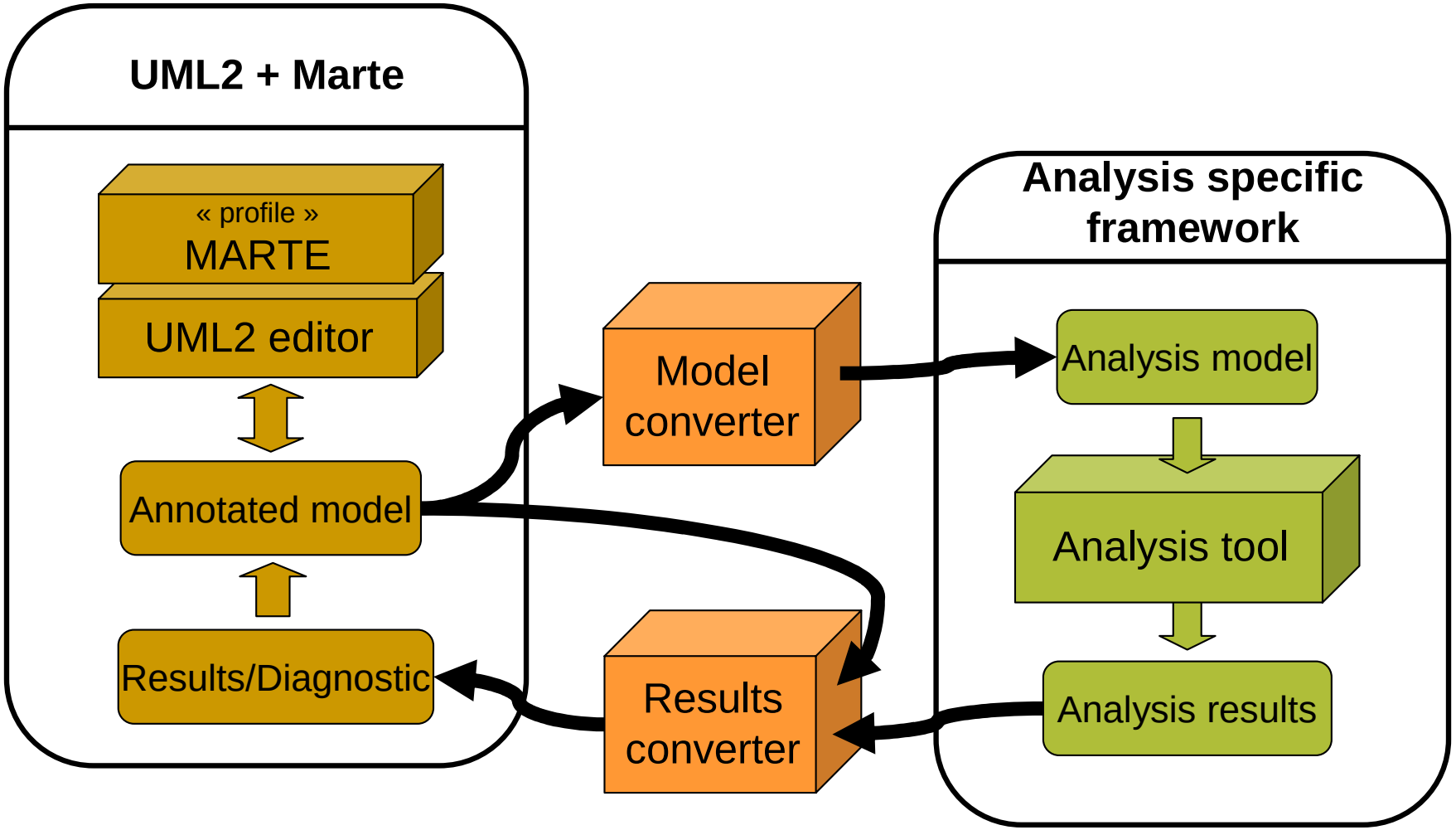
# GQAM: Dependencies and Architecture



# GQAM: Analysis Modeling Structure



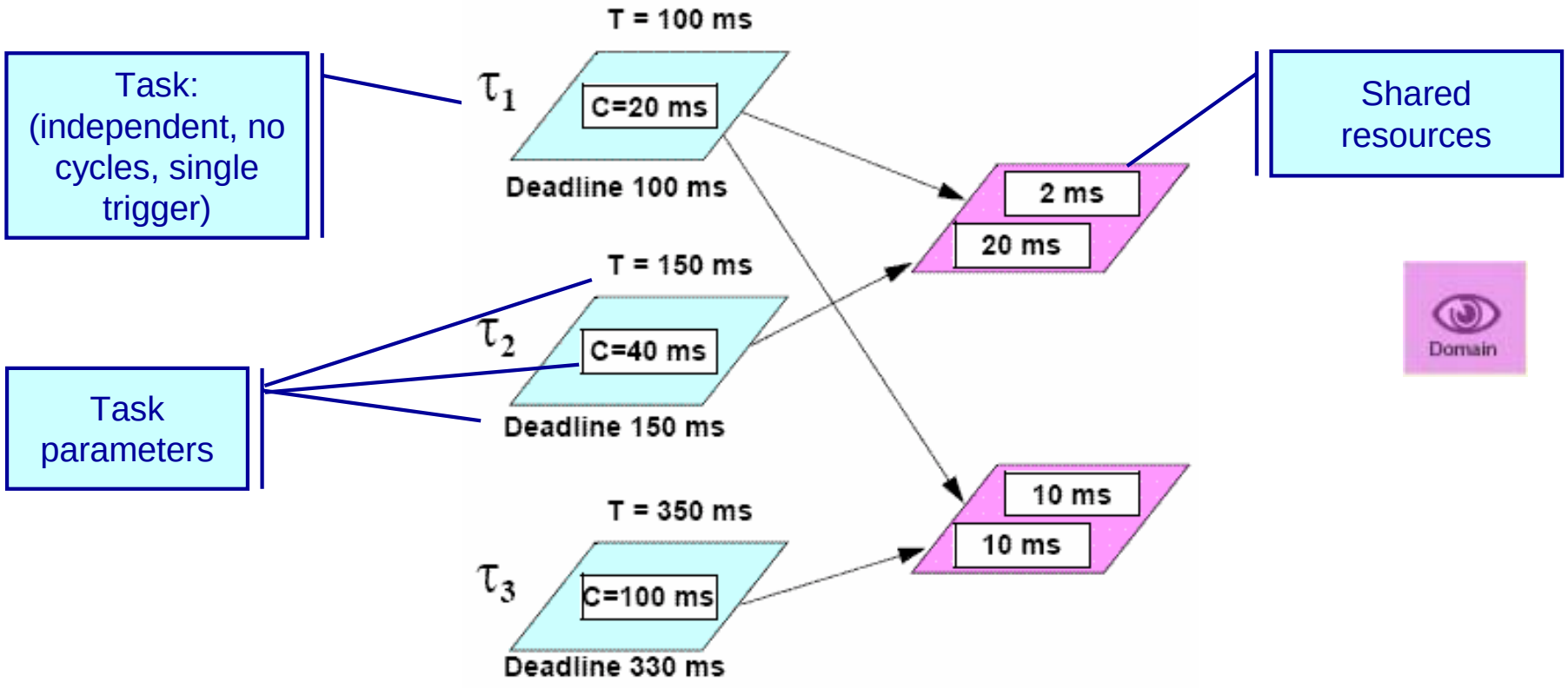
# Processing schema for model-based analysis



# Key reasoning for the use of UML in the scheduling analysis of RTE Systems

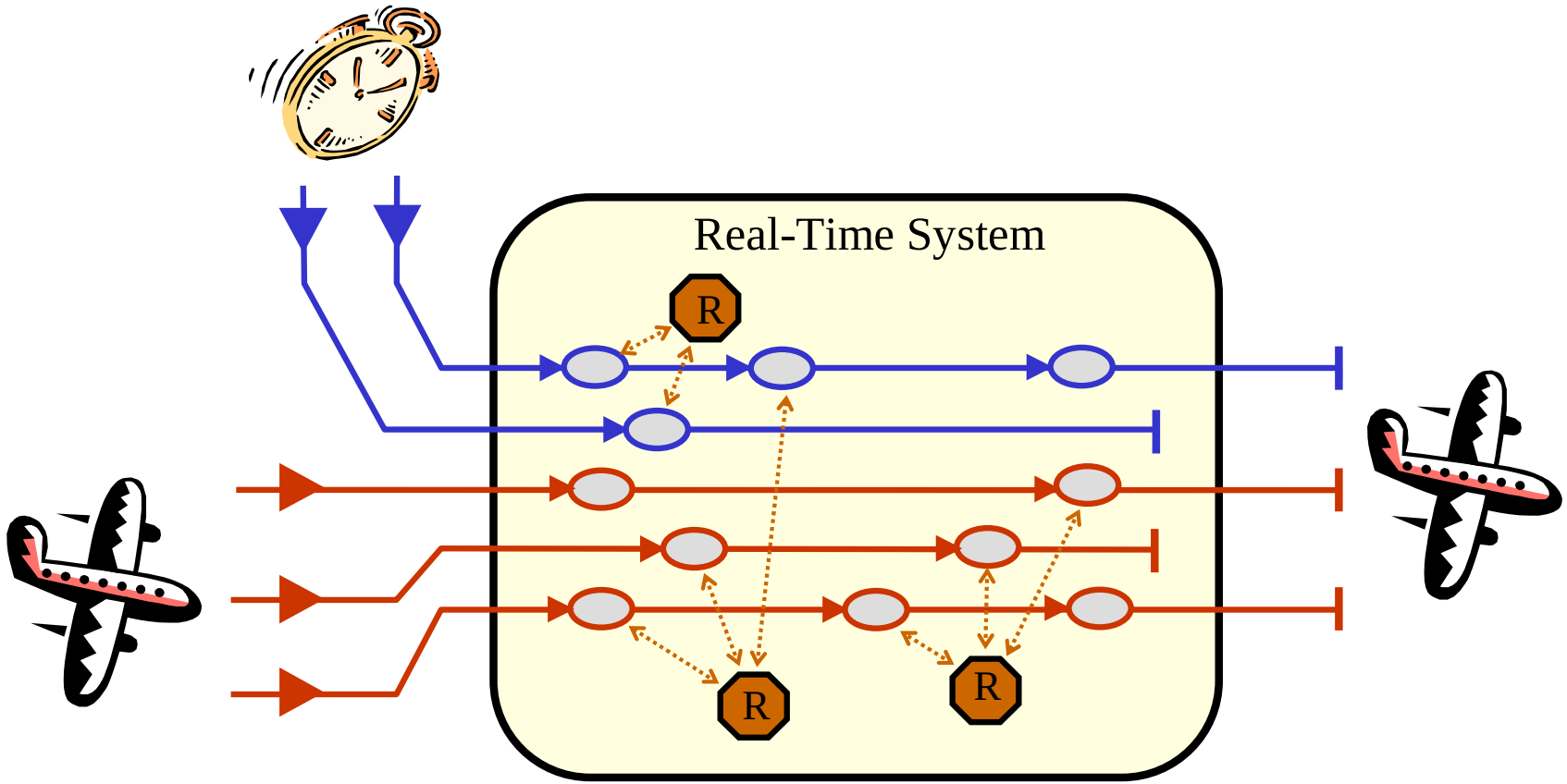
- UML is a standard semi-visual language for conceptual modeling that enables the usage of a Model Based approach for software and system engineering.
- MDD and UML have been broadly introduced and used in principle by the software engineering community, and have reached a significant number of practitioners 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.
- This 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.

# A Simple Example (Classical Scheduling Theory)



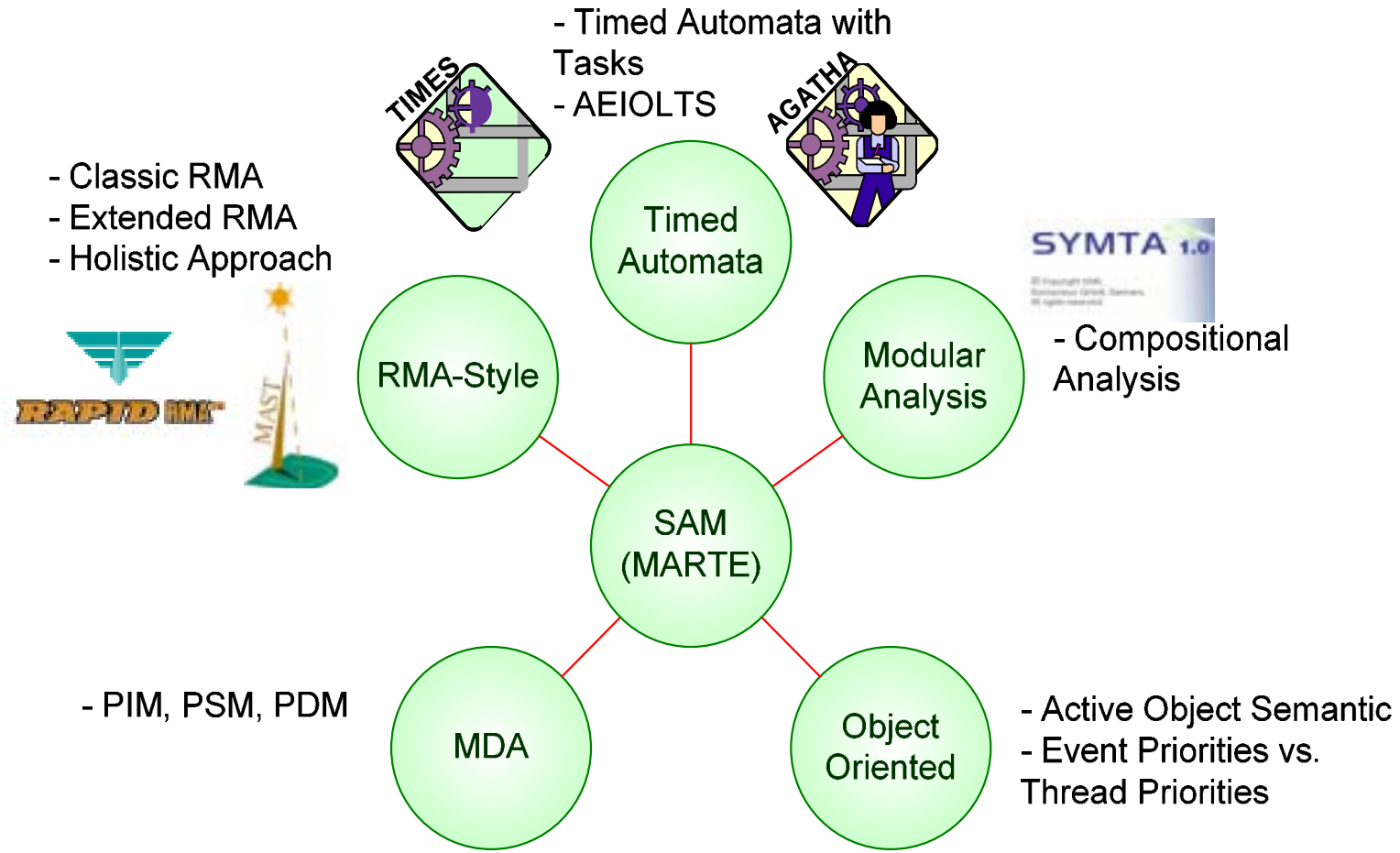

- Typically analyzed with RMA:
  - Critical instant calculation
  - Utilization bound test or Response time calculation for the first deadline.

# More general approach



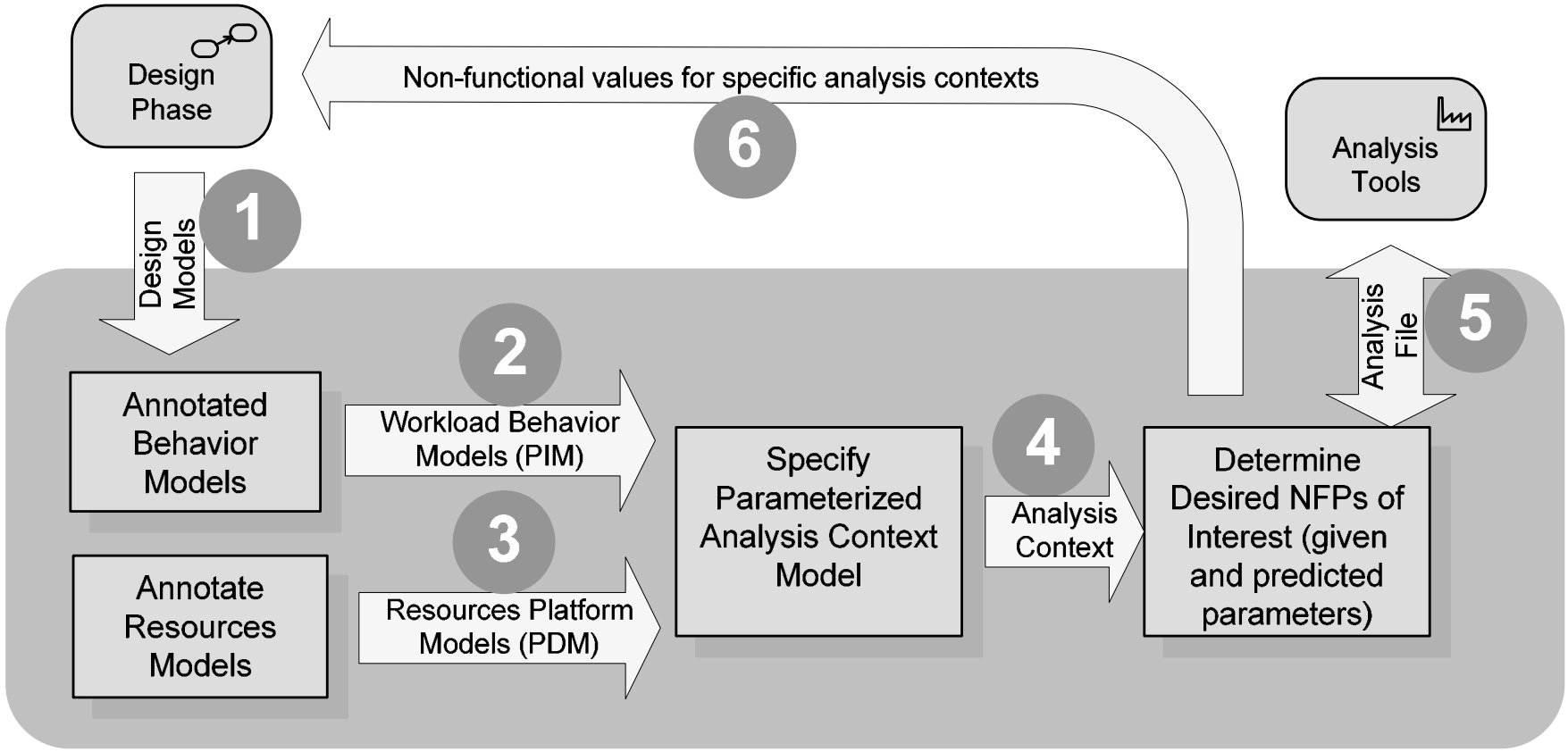

**Scenario(instance) based, distributed, control-flow dependencies**

# SAM: Integration Different Approaches




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

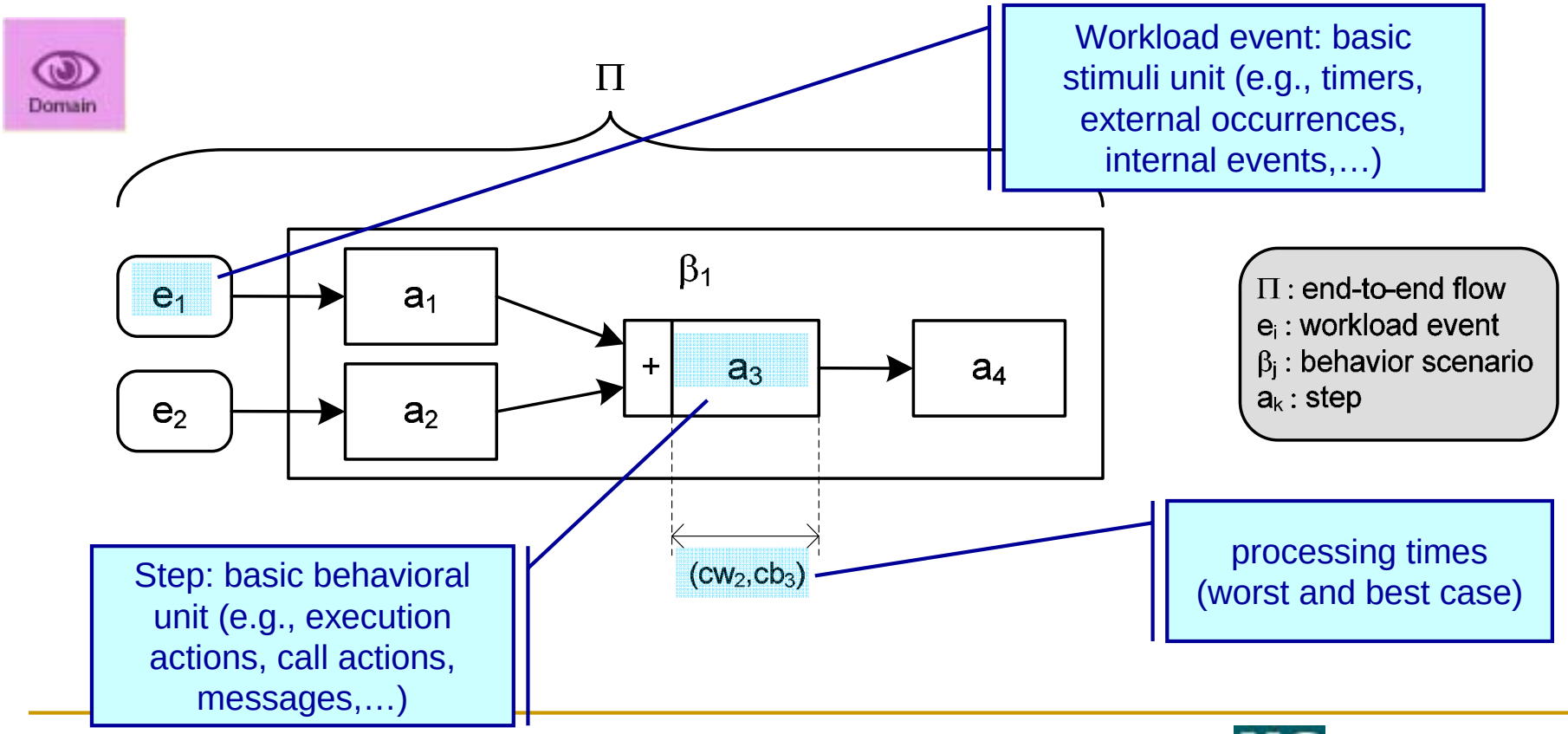
# General Procedure to Use the SAM Profile



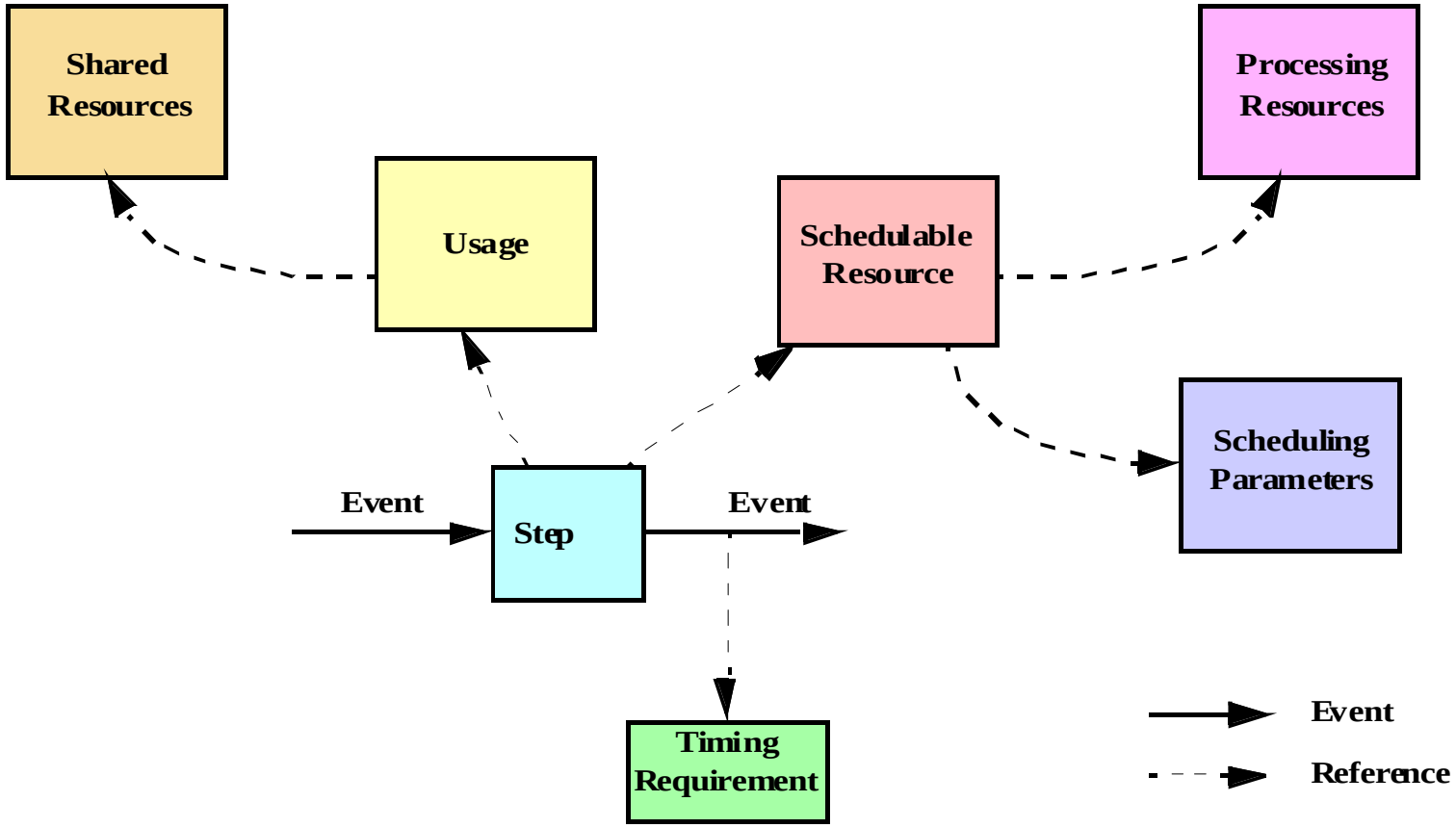
# SAM: The Notion of End-To-End Flow

**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.

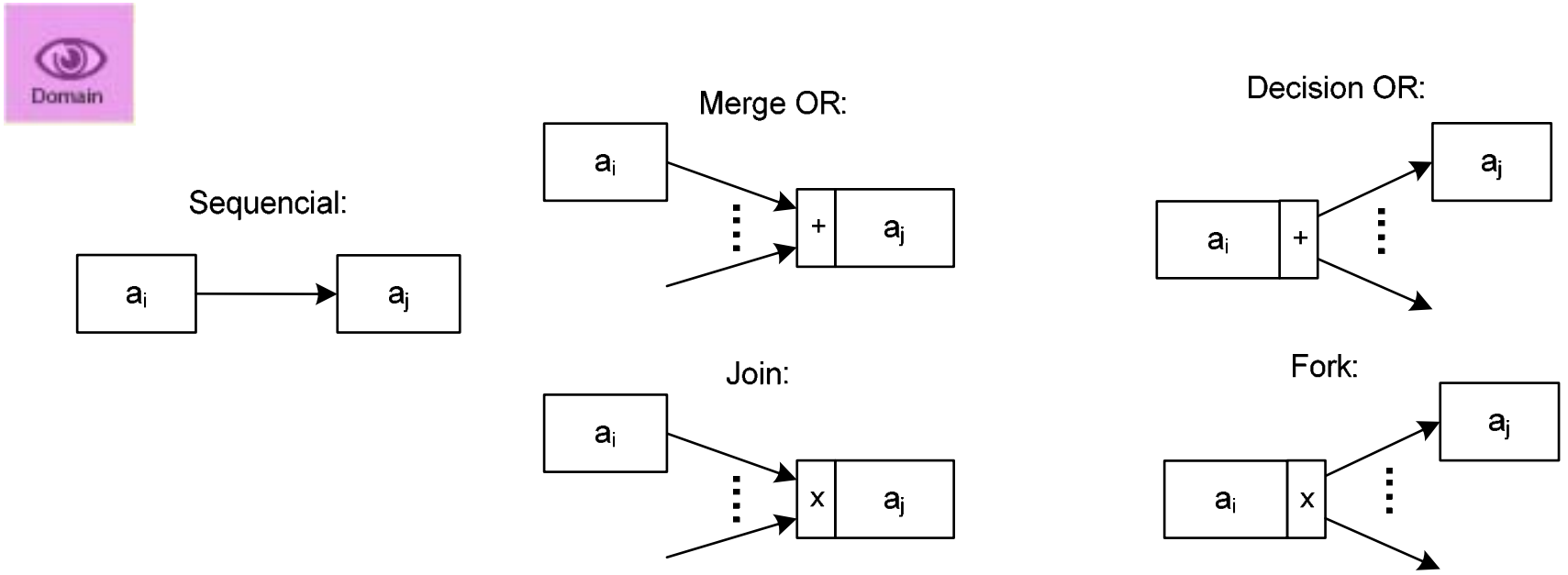


# End-to-end flows (from MAST)

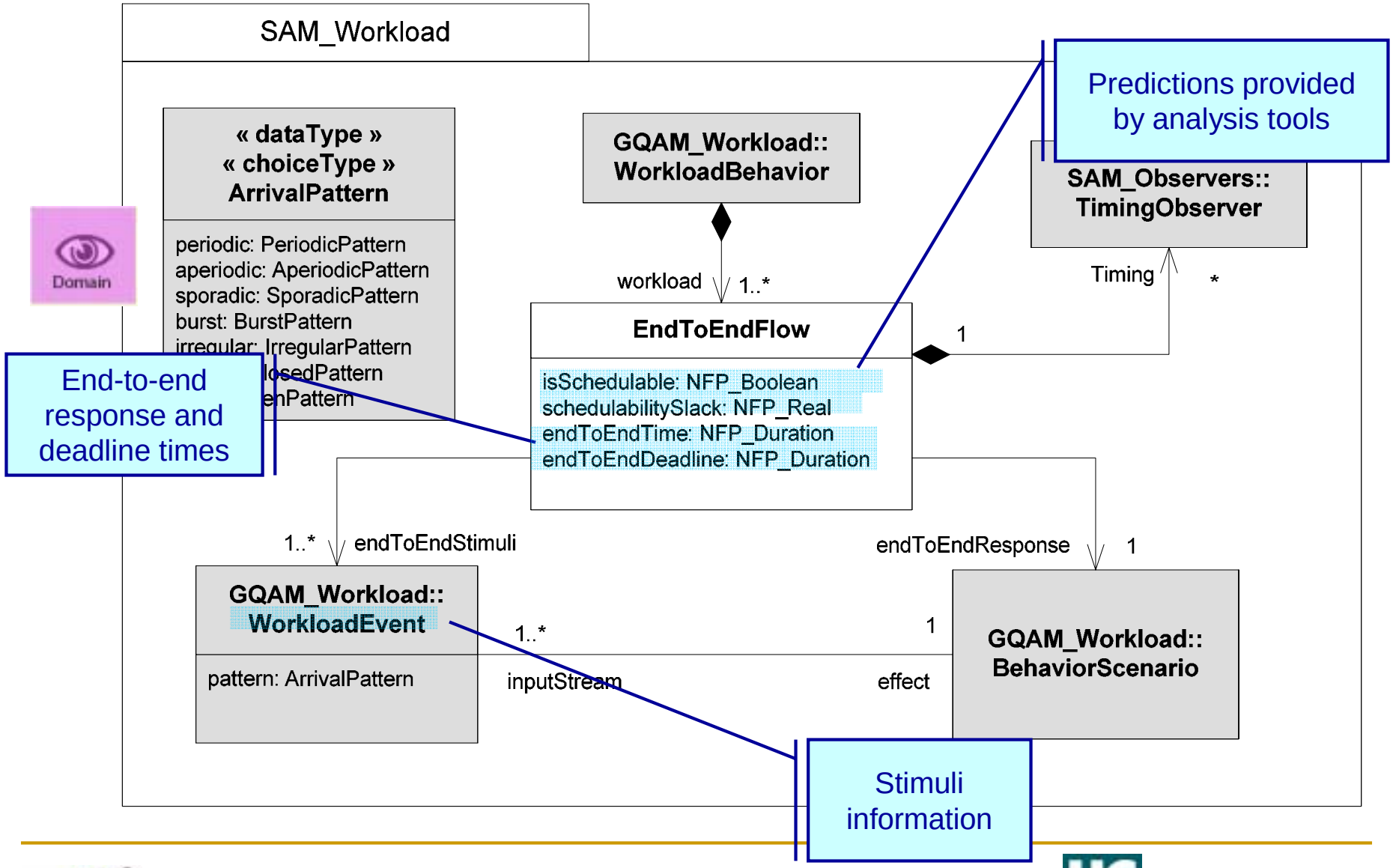


# SAM: Precedence Relations

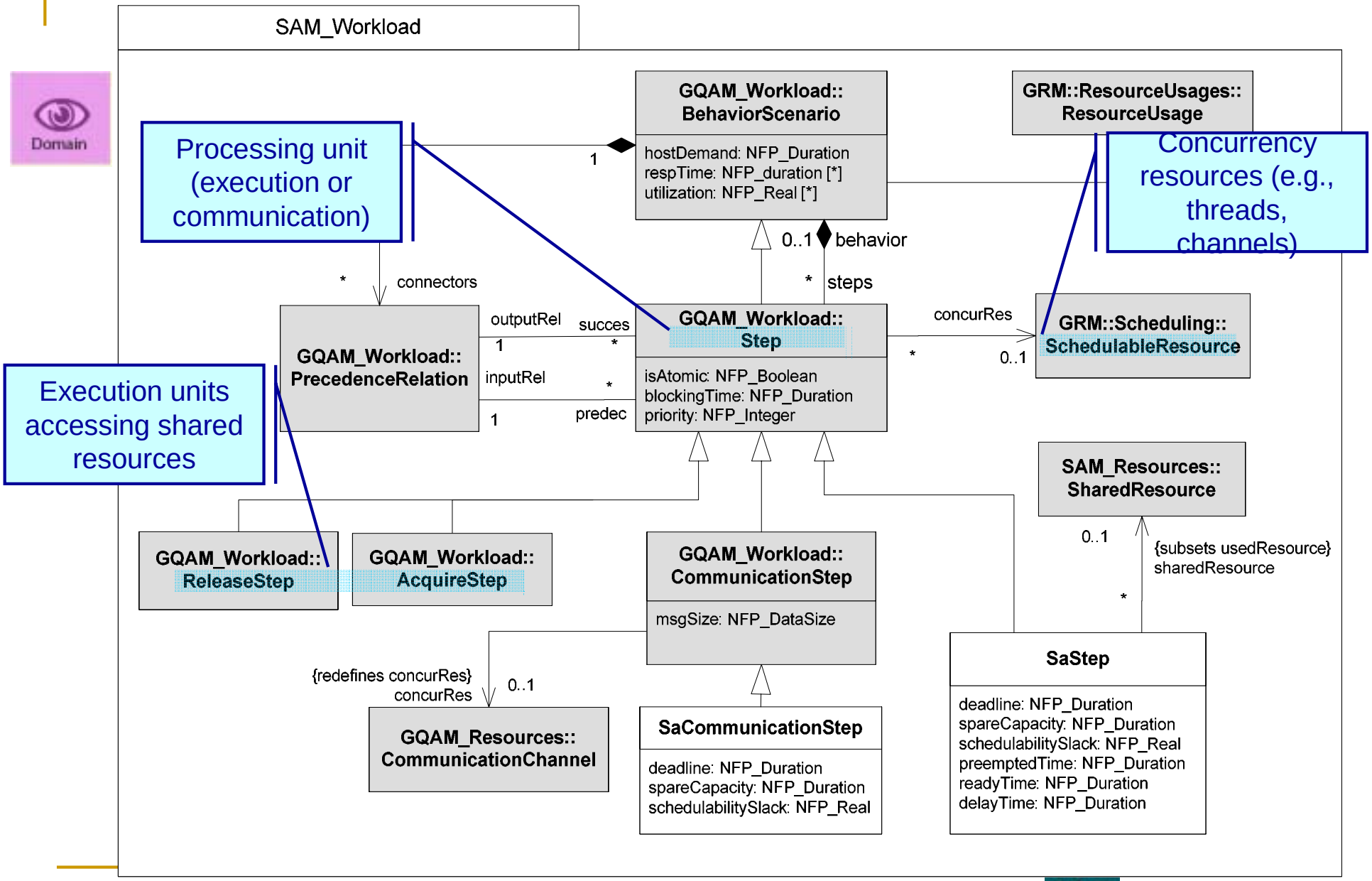
**Execution and communication steps may be causally related by one of the following precedence relations:**



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



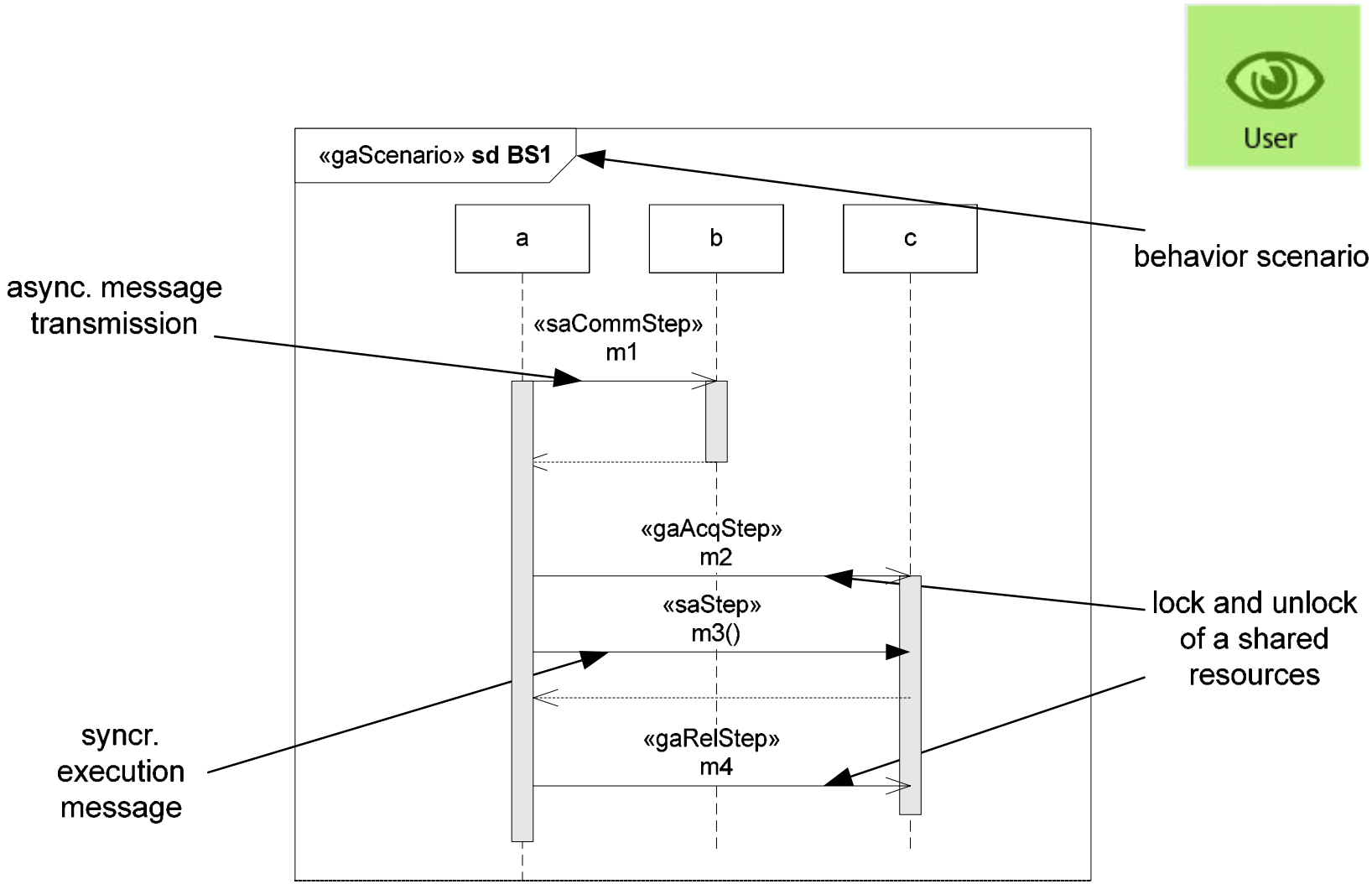
# SAM: Workload Domain Metamodel (detailed behavior.)



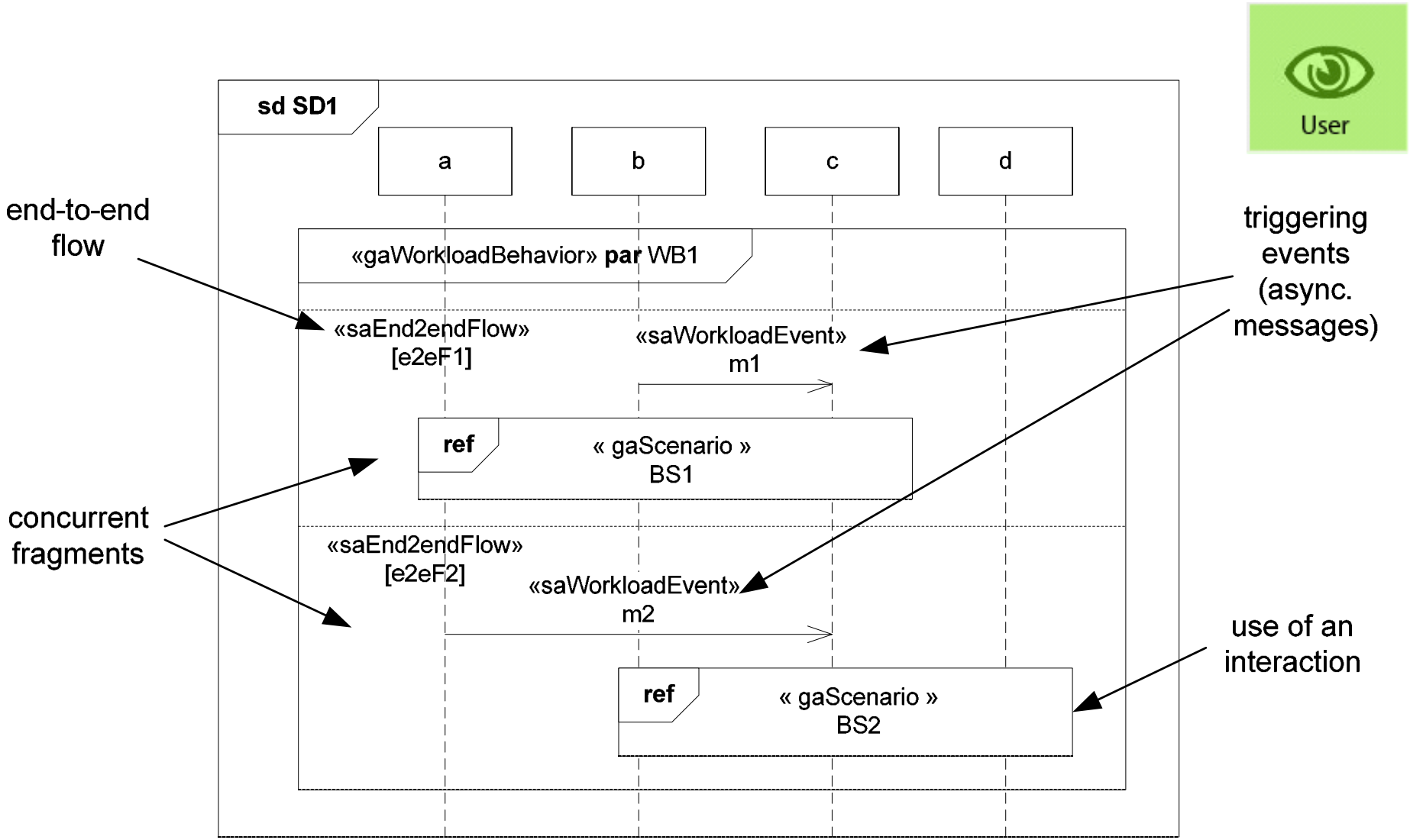
# SAM: Example of Stereotype Extensions Usage


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

# SAM: Examples of Behavior Annotations

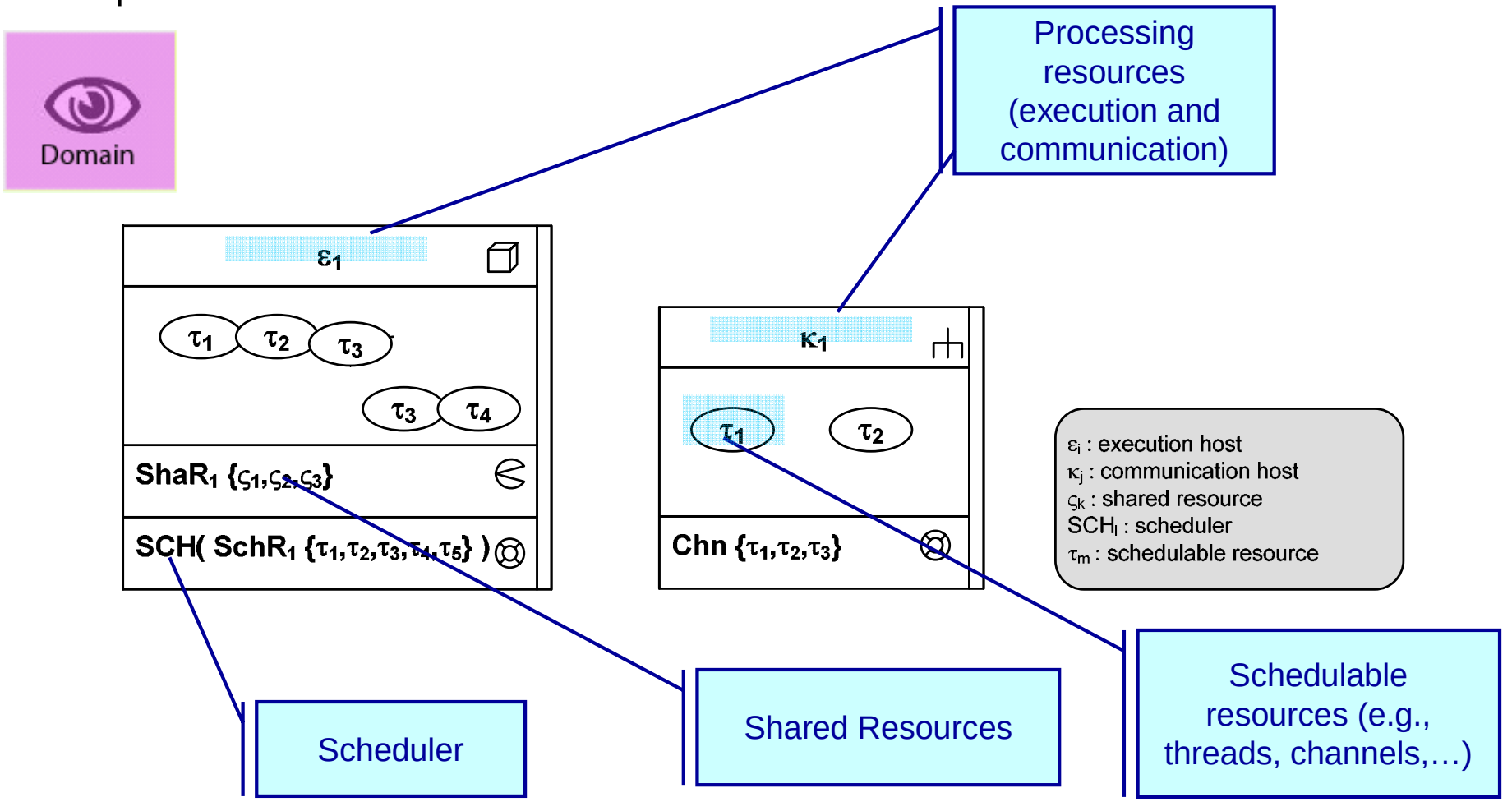


# SAM: Example of Workload Annotations

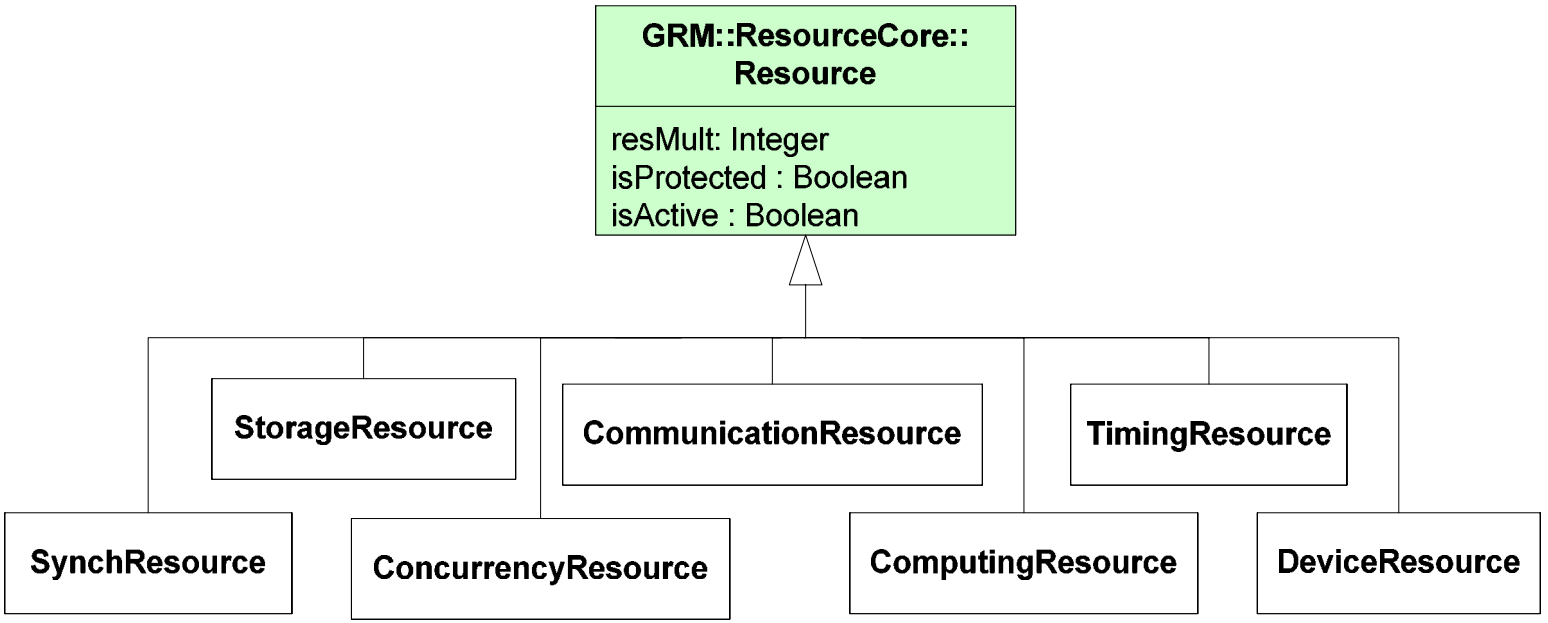


# SAM: Resources Concepts

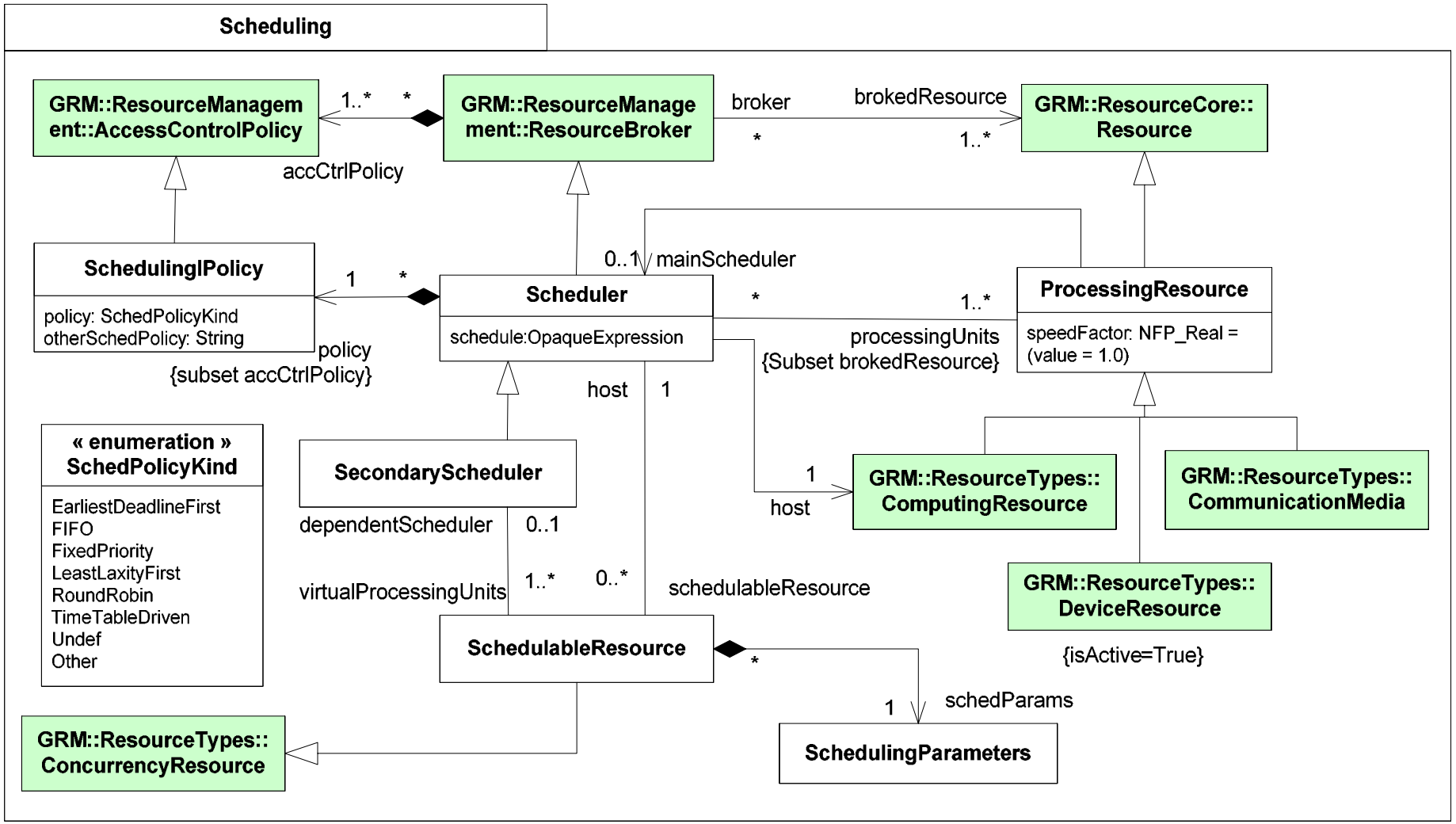
- Provide additional (analysis-specific) annotations to annotate resources platform models



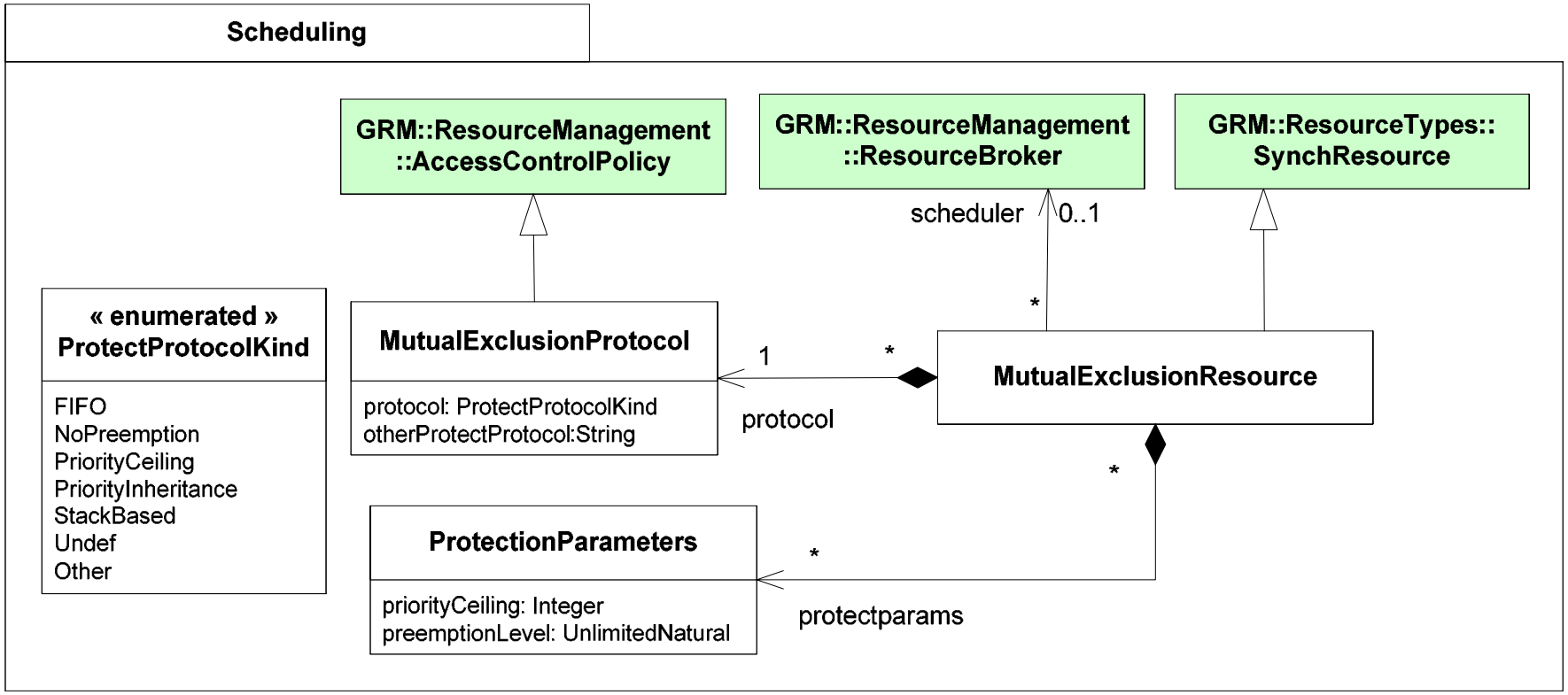
# Resources (from GRM)



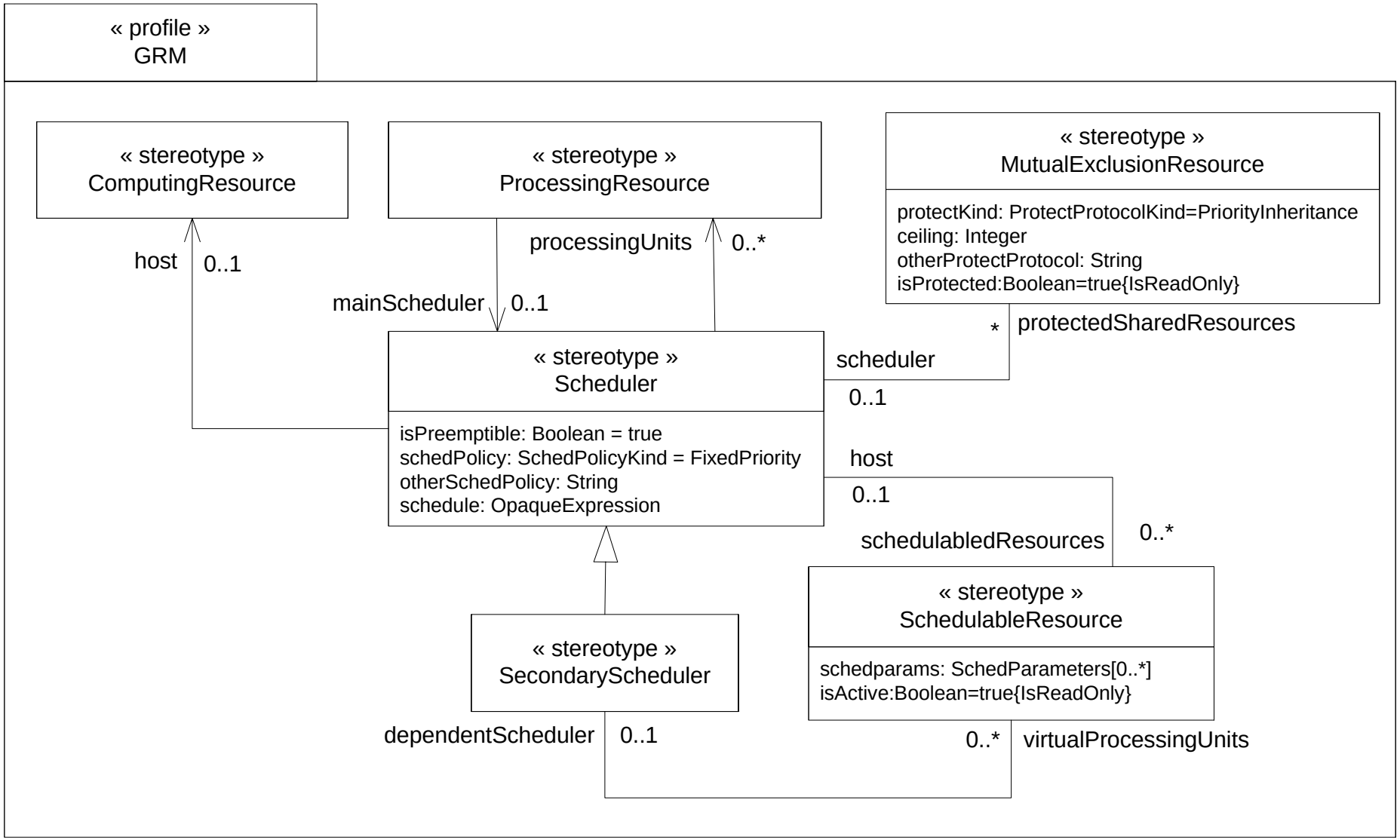
# Domain model for Schedulable resources



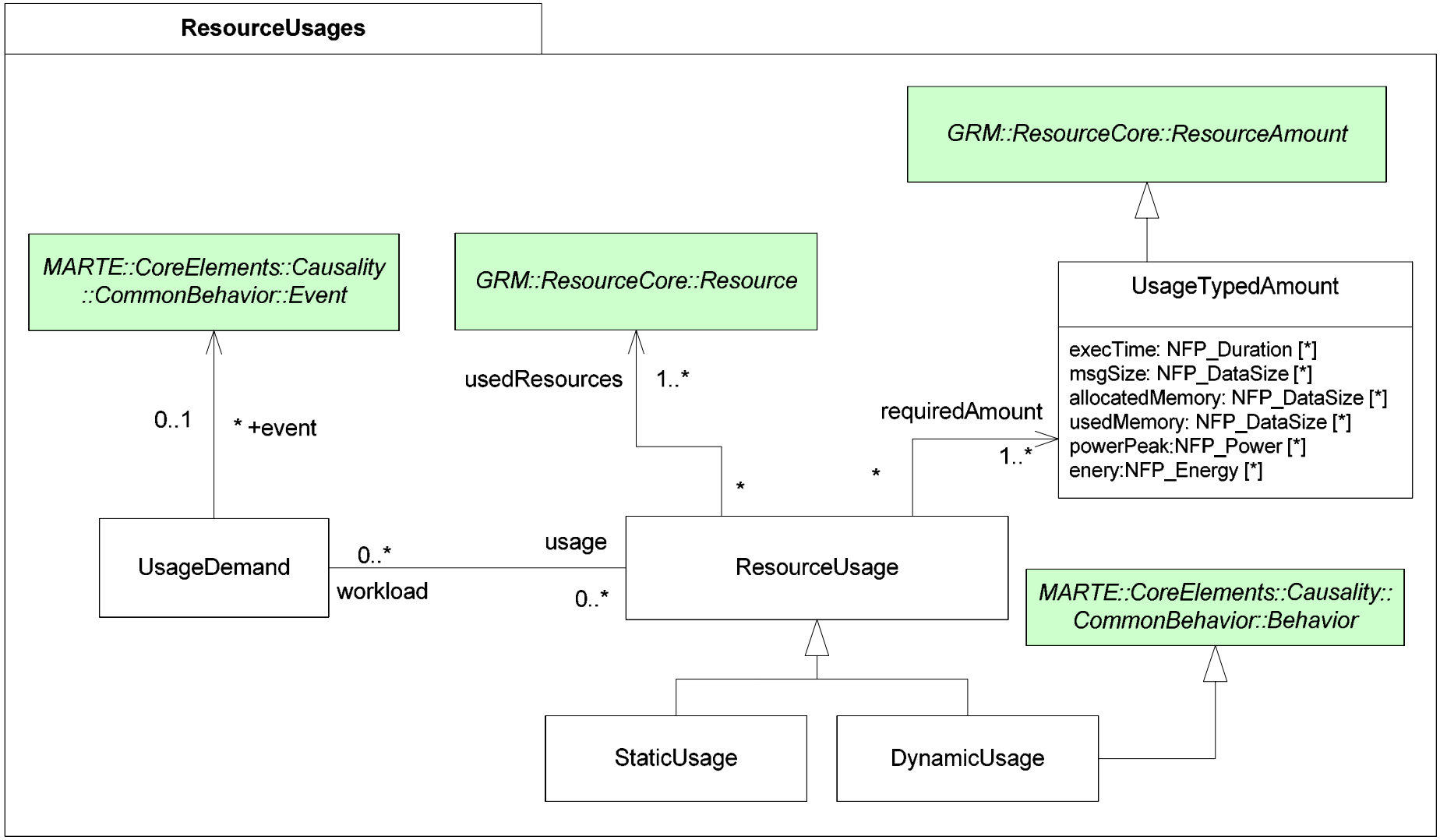
# Shared resources

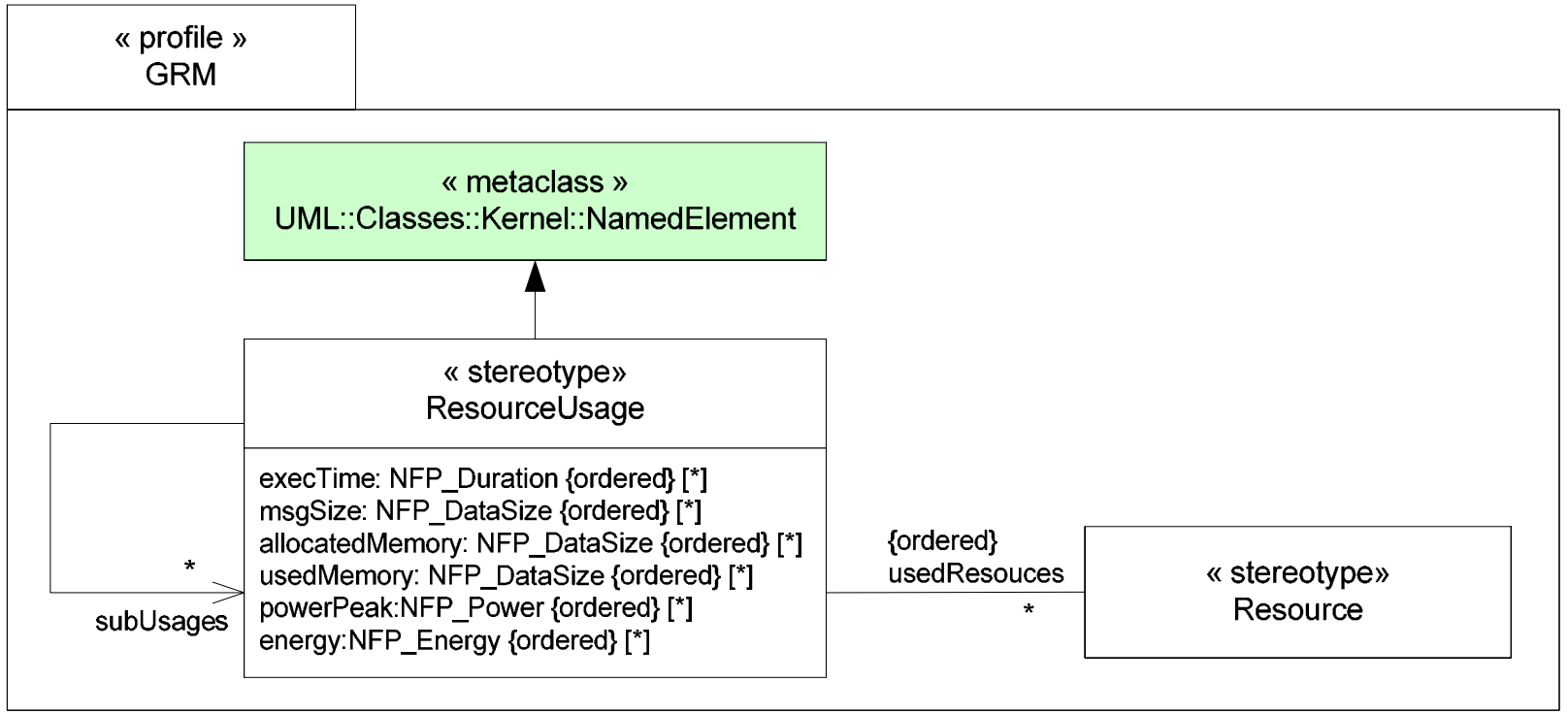


# Extensions for scheduling



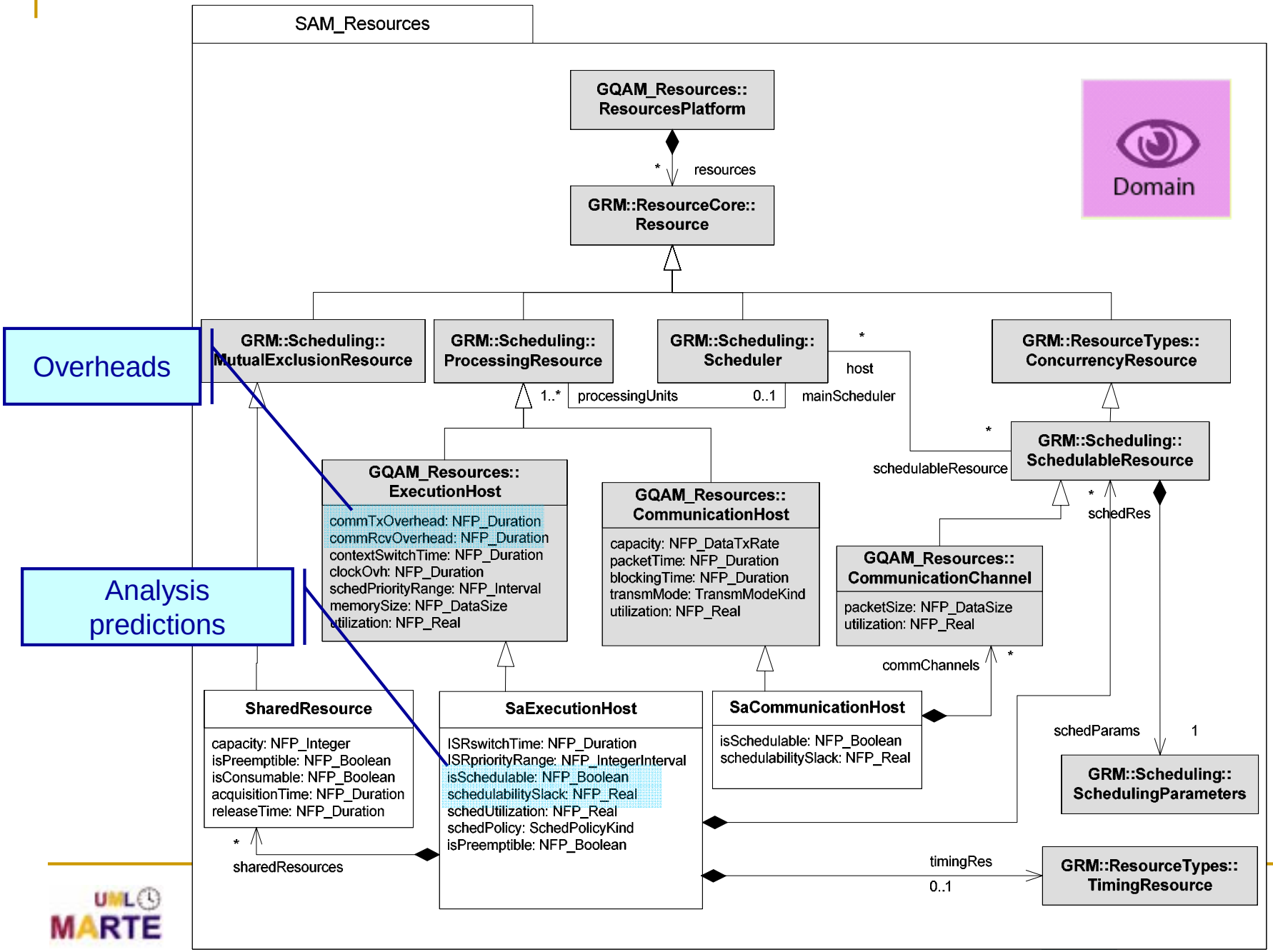
# Resource Usage






1. If the list usedResources is empty the list subUsages should not be empty and viceversa..
2. If the list usedResources has only one element, all the optional lists of attributes refer to this unique Resource and at least one of them must be present.
3. If the list usedResources has more than one element, all of the optional lists of attributes that are present, must have that number of elements, and they will be considered to match one to one.
4. If the list subUsages is not empty, and any of the optional lists of attributes is present, then more than one annotation for the same resource and kind of usage may be expressed. In this case, if the annotations have also the same source and statistical qualifiers they will be considered in conflict, and hence the ResourceUsage inconsistent.

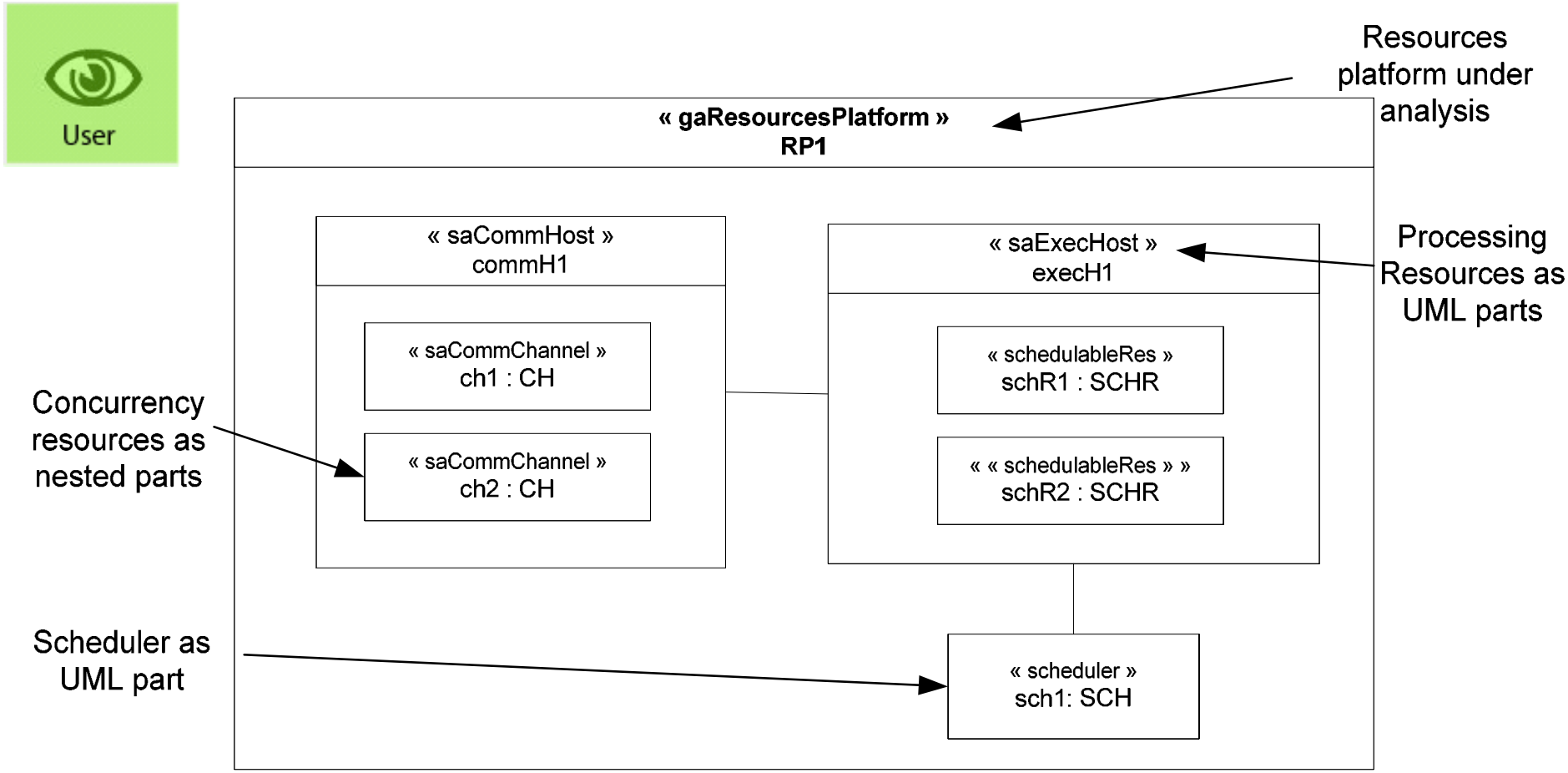
# SAM: Resources Domain Metamodel



# SAM: Examples of the Stereotypes Usage


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

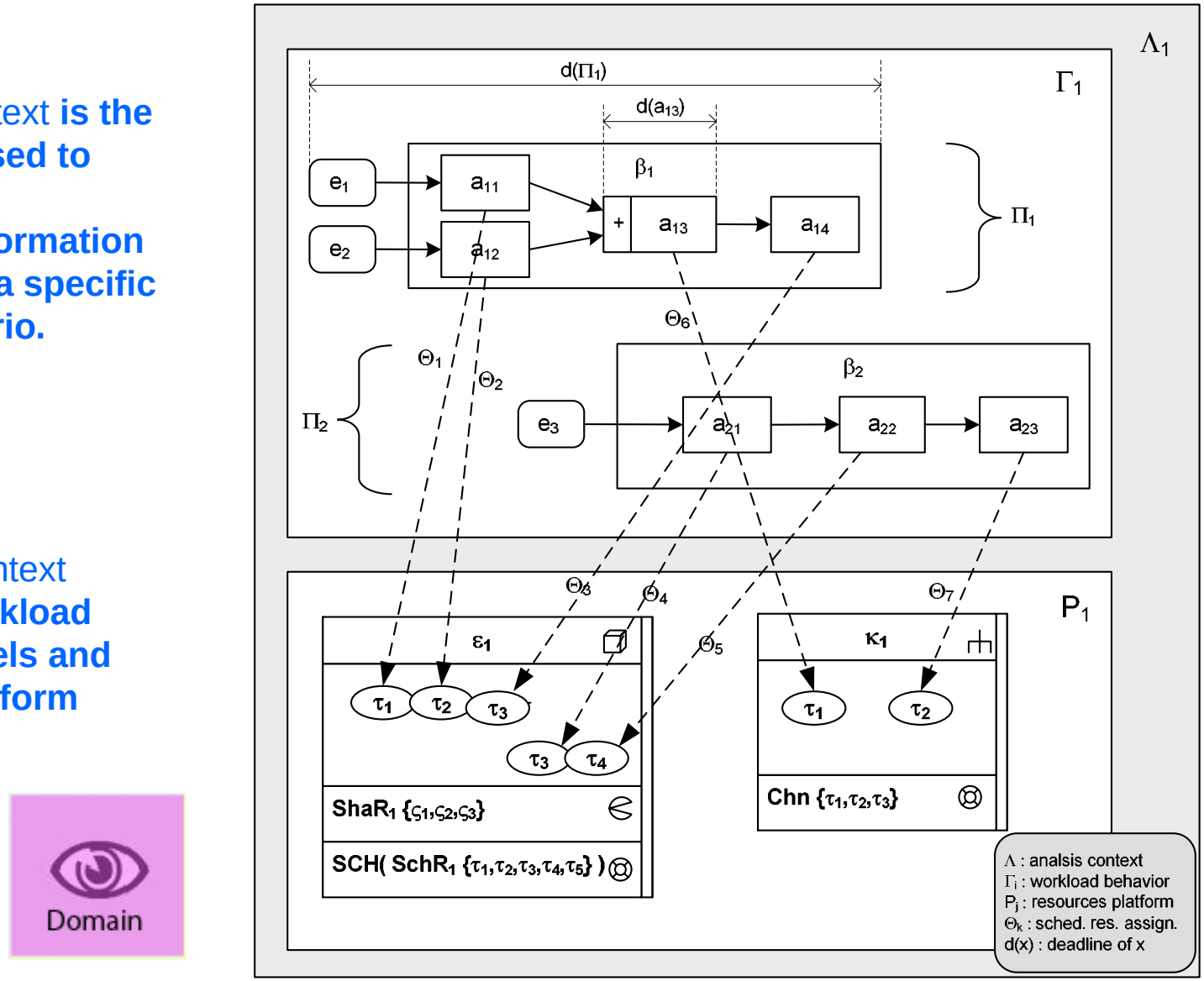
# SAM: Example of Resources Stereotype Usage



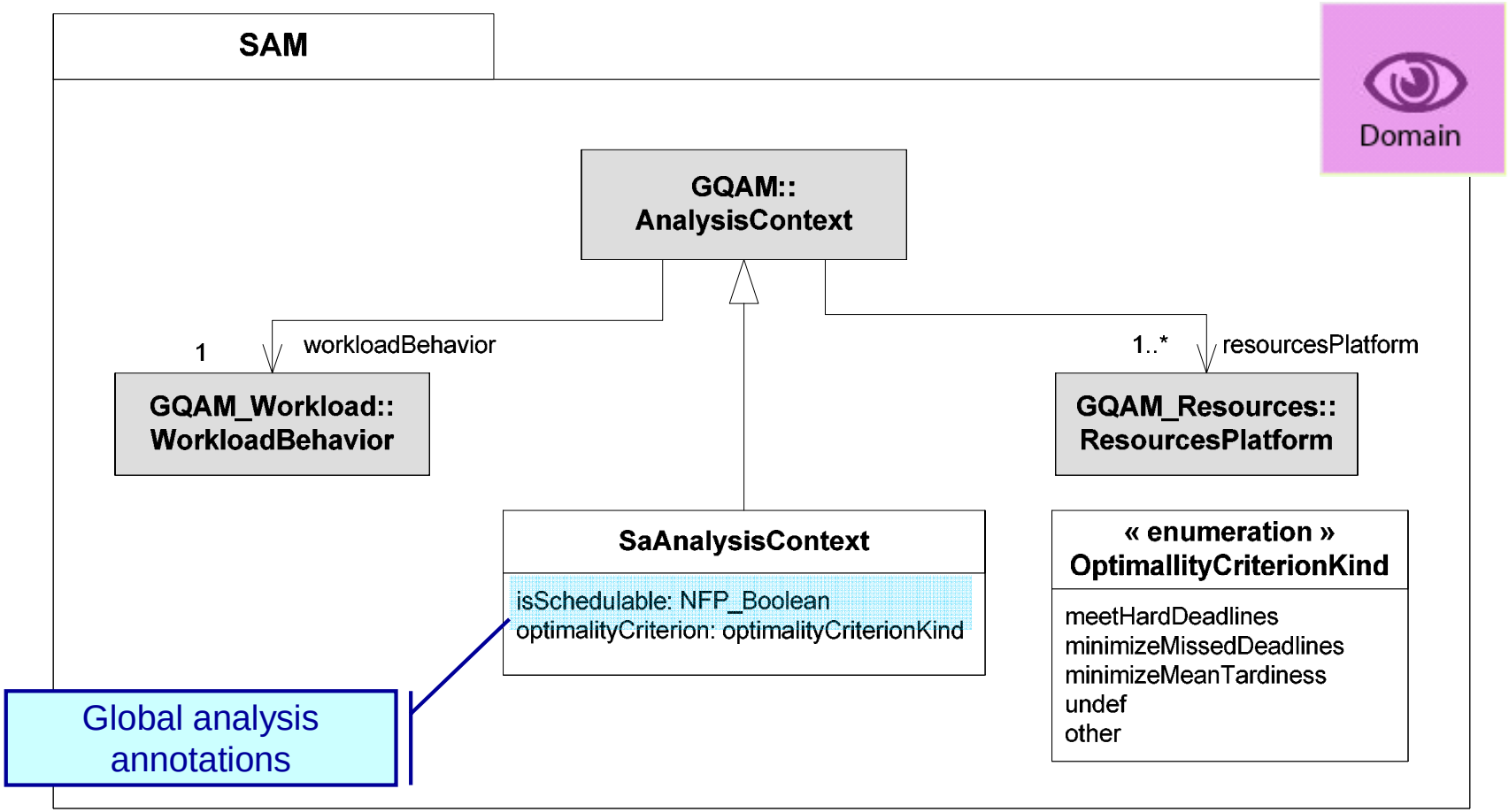
# SAM: Analysis Context concepts

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

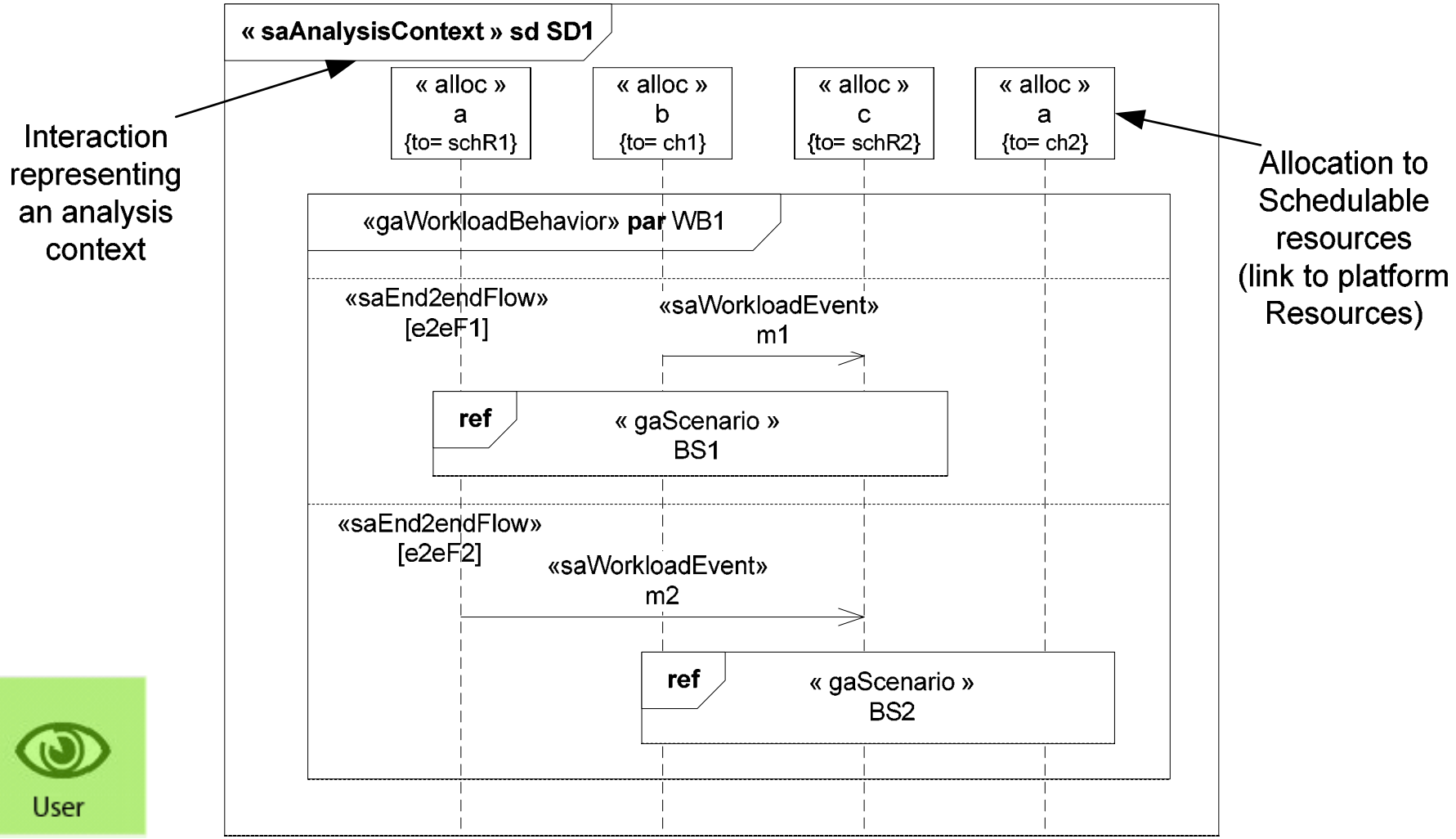
- **An** analysis context **integrates workload behavior models and resources platform models.**



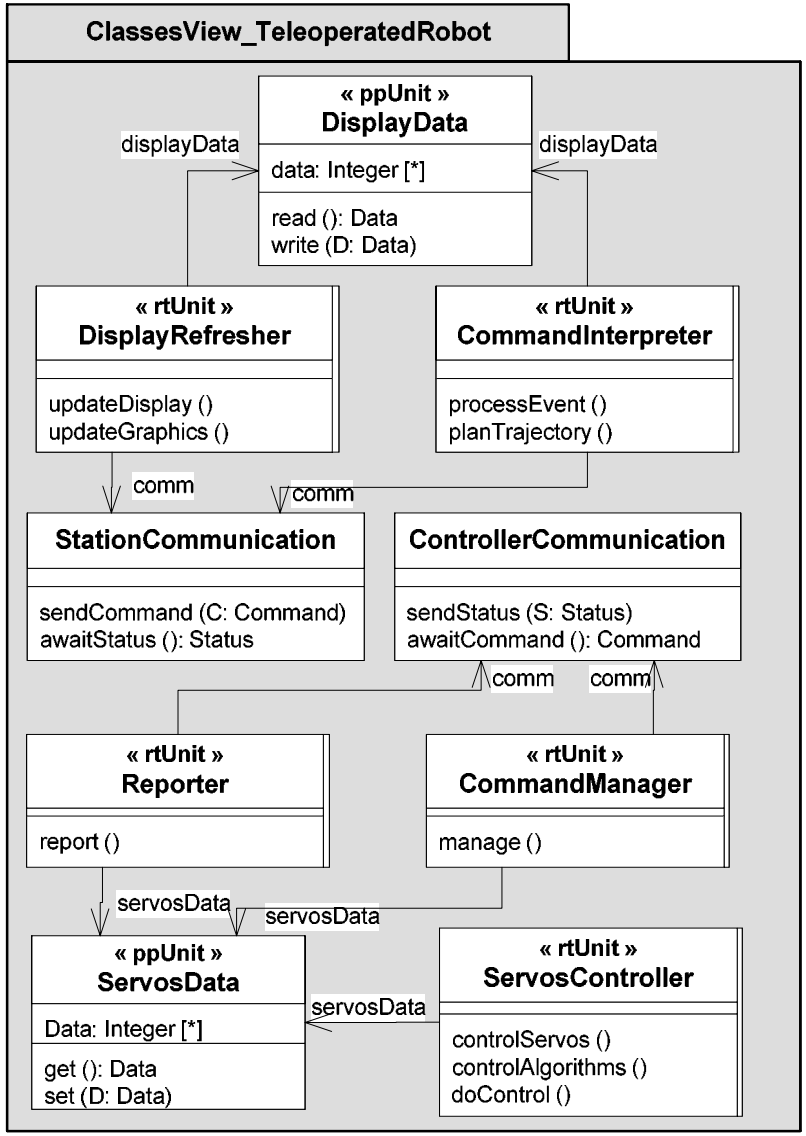
# SAM: Analysis Context Domain Metamodel

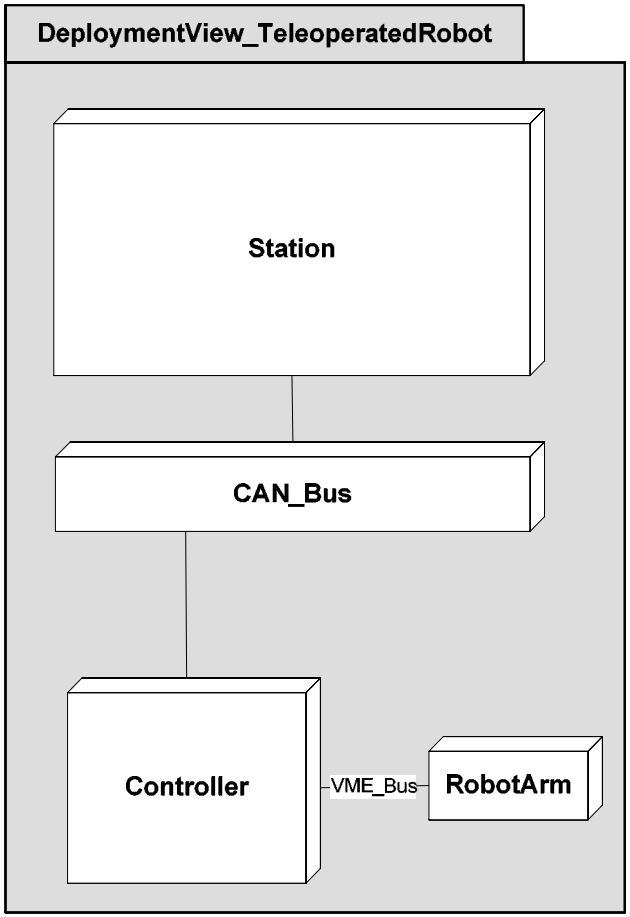


# SAM: Example of Analysis Context Stereotype Applic.

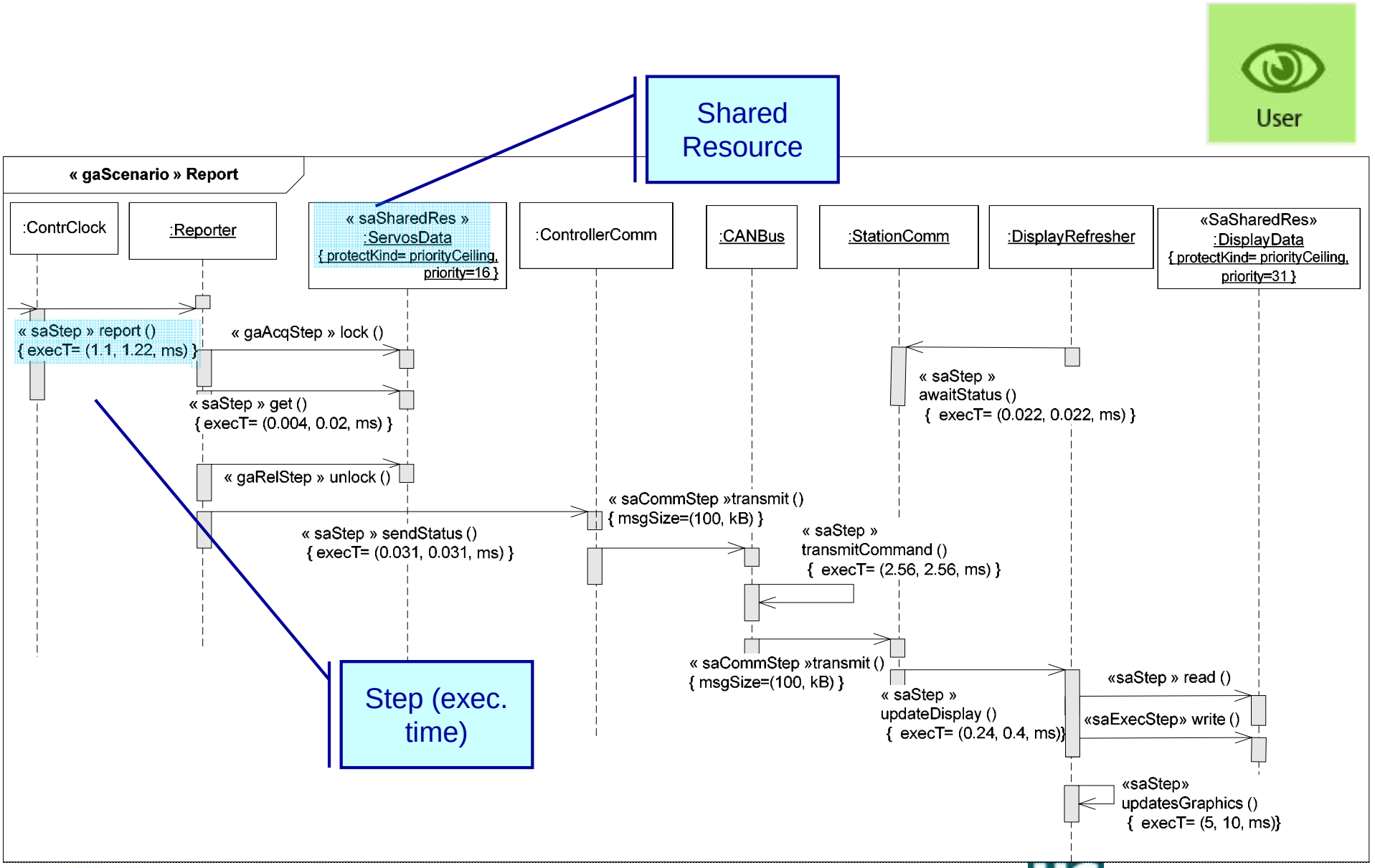


# Example: A Teleoperated Robot

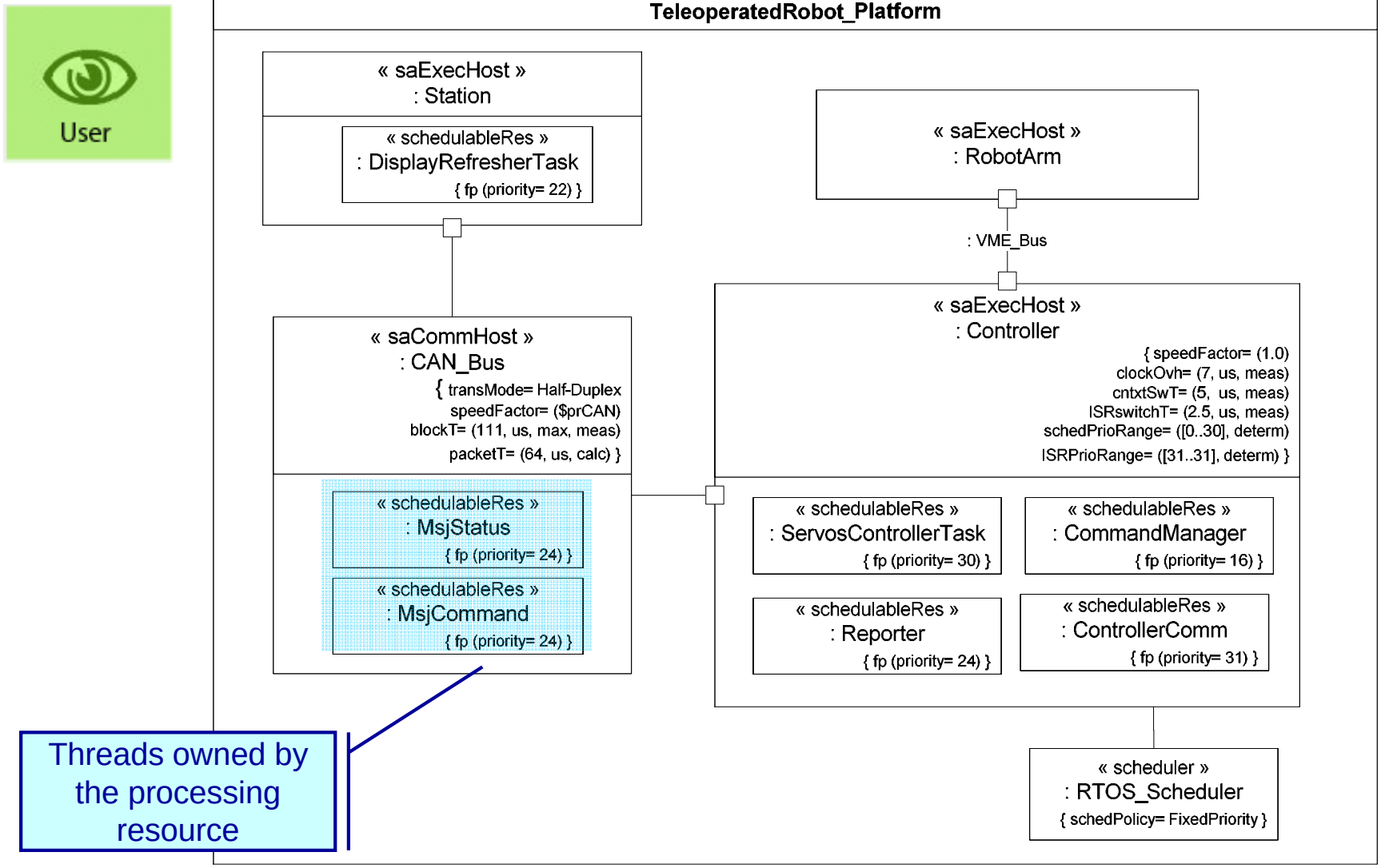



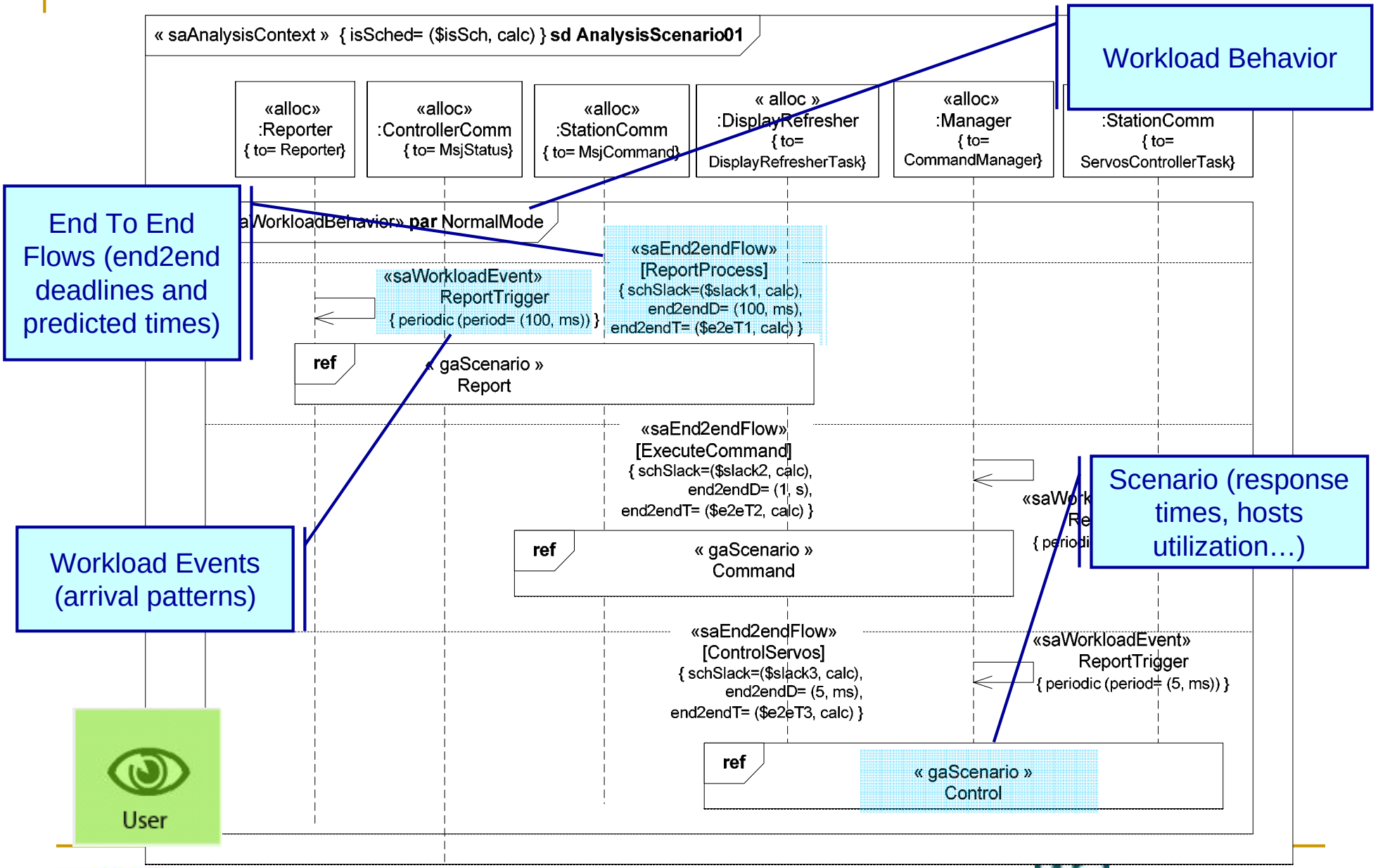
# Example of Annotated Scenario with SAM



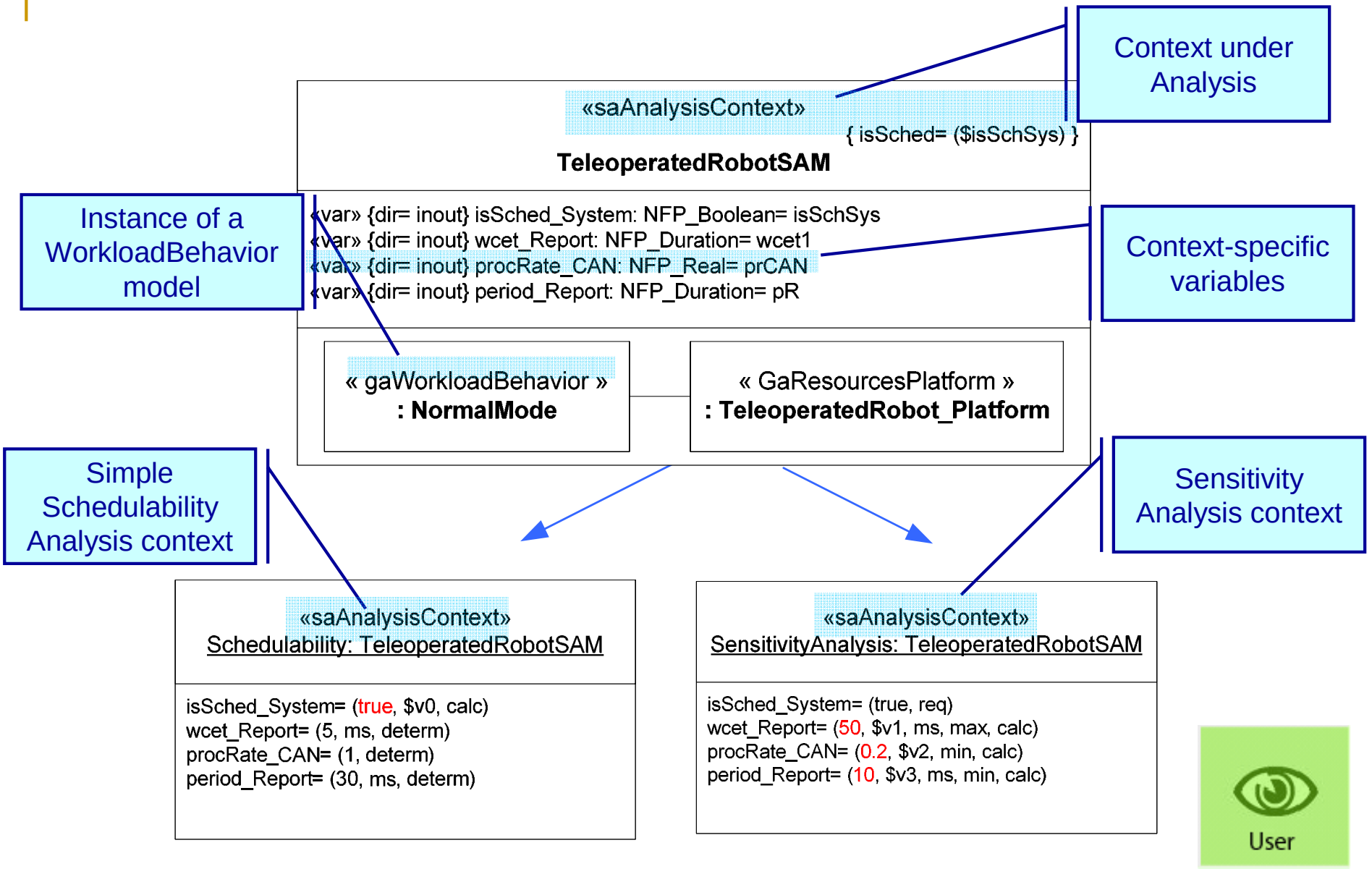
# Example of Annotated Resources Model with SAM



# 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
  - Implemented by RSA, Magicdraw, Raphsody, PapyrusUML,…
- Links to other tools:
  - Eclipse plug-ins to transform UML models annotated with the SAM profile to input files of MAST, SymTA/S, Cheddar and RapidRMA
  - TimeSquare: handling of Clock Constraint Specification Language

The official OMG web page holds links to al these tools**:**
**www.omgmarte.org**

MARTE Open Source Eclipse based implementation in UML Papyrus: **www.papyrusuml.org**

# Conclusions on analysis capabilities of MARTE

- 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

# MARTE Frontiers and Challenges

- MARTE defines the language constructs only!
  - Common patterns, base building blocks, standard NFP annotations
  - Generic constraints that do not force specific execution models, analysis techniques or implementation technologies
- It does not cover methodologies aspects:
  - Interface-Based Design, Design Space Exploration
  - Means to manage refinement of NFP measurement models
  - Concrete processes to storage, bind, and display NFP context models
  - Mapping to transform HLAM's into analysis models

**MARTE is to the RTES domain as UML to the System & Software domain: a family of well known and open specification formalisms!**

# Thanks for your attention..!

## Questions ?

- www.omgmarte.org
  - The MARTE web site
- www.papyrusuml.org
  - On open source Eclipse plug-in for UML2 graphical modeling