Modeling Non-Functional Properties with MARTE

Huascar Espinoza, PhD
CEA LIST, LISE Lab - France

MARTE Information Day

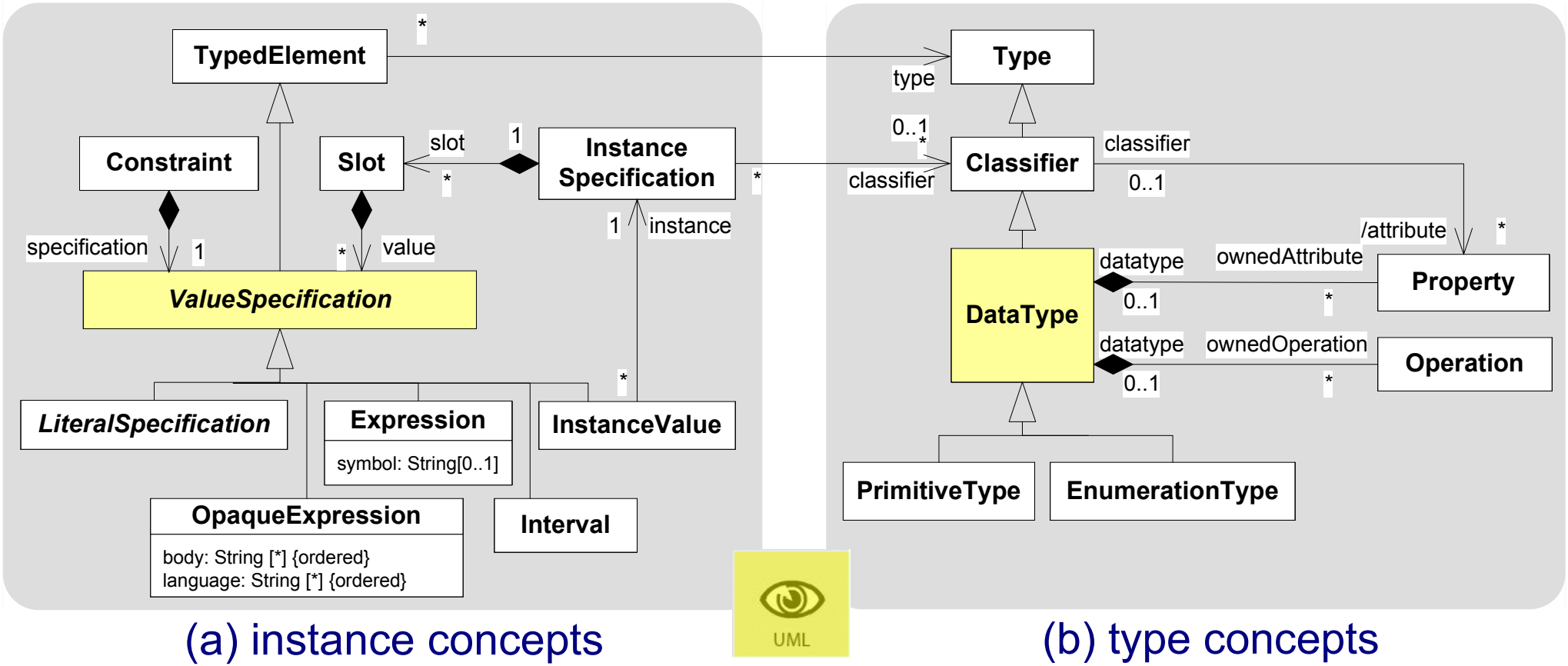
December 12th, Burlingame CA

Non-Functional Properties

An information of a modeled system, or system part, which is not related to its functional purpose
(e.g., latencies, memory size, power consumption)

- Information used to:
  - ➔ Static V&V: performance prediction, resource usage evaluation, …
  - ➔ Dynamic configuration: resource management, mode changes, …
- Should include design process information:
  - ➔ How information was obtained (estimations, measurement,…)?
  - ➔ How information was calculated (derived parameters, refinement)?
  - ➔ How accurate information is?

UML-Related Concepts



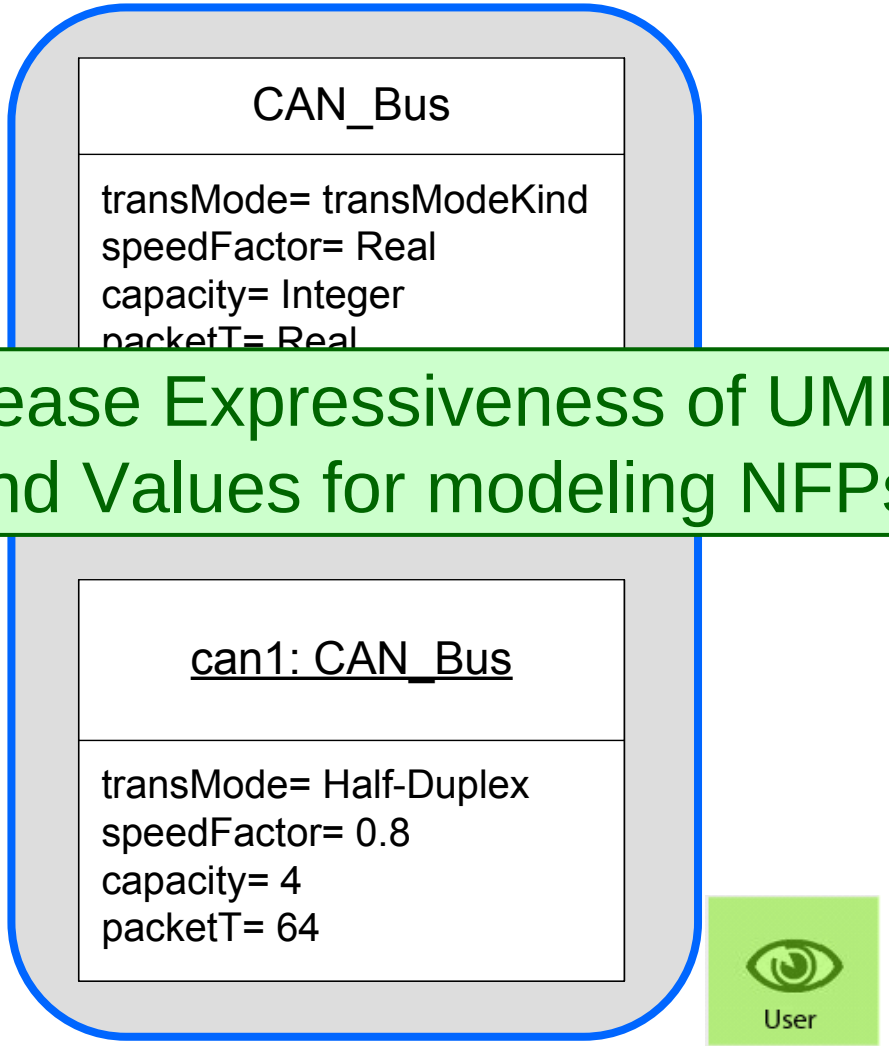
(a) instance concepts

(b) type concepts

UML lacks modeling capabilities for non-functional modeling !!

Extending UML Expressiveness for NFPs

CAN Bus with **Pure UML**



Our Goal: Increase Expressiveness of UML Data Types and Values for modeling NFPs

The NFP Modeling Framework

A set of extensions to specify semantically rich non-functional annotations

1. NFPs sub-profile:
   - ➔ Measurements: magnitude + unit (e.g., energy, data size, duration)
   - ➔ Qualifiers: Value source, statistical measure, value precision,…
2. Value Specification Language (VSL):
   - ➔ Mathematical expressions (arithmetic, logical, …)
   - ➔ Time expressions (delays, periods, trigger conditions,…)
   - ➔ Variables: placeholders for unknown analysis parameters.

The NFP Modeling Framework

- UML lacks modeling capabilities for NFPs !!

- Value qualifiers?
- Measures?
- NFP Libraries?
- Annotation mechanism?

- And UML expression syntax is also not sufficient!!

- Variables?
- Structured Values?
- Data Type System?
- Complex time expressions?

Annotating NFPs in Property/Slots

1) Declare NFP types

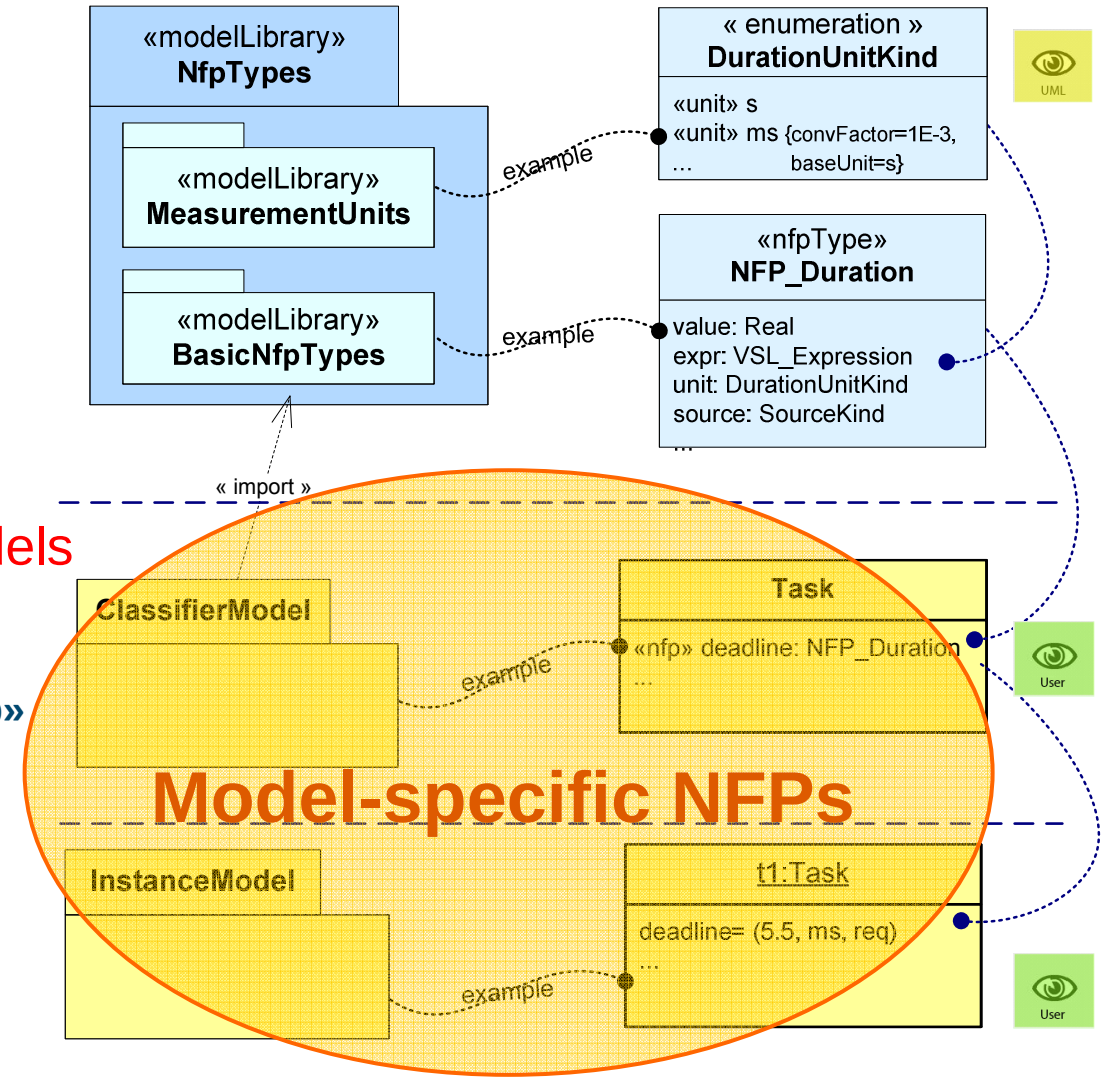
- **Define measurement units and conversion parameters**
- **Define NFP types with qualifiers**

2) Declare NFPs in user models

- **Define classifiers and their attributes using NFP types**
- **Such attributes are tagged as «nfp»**

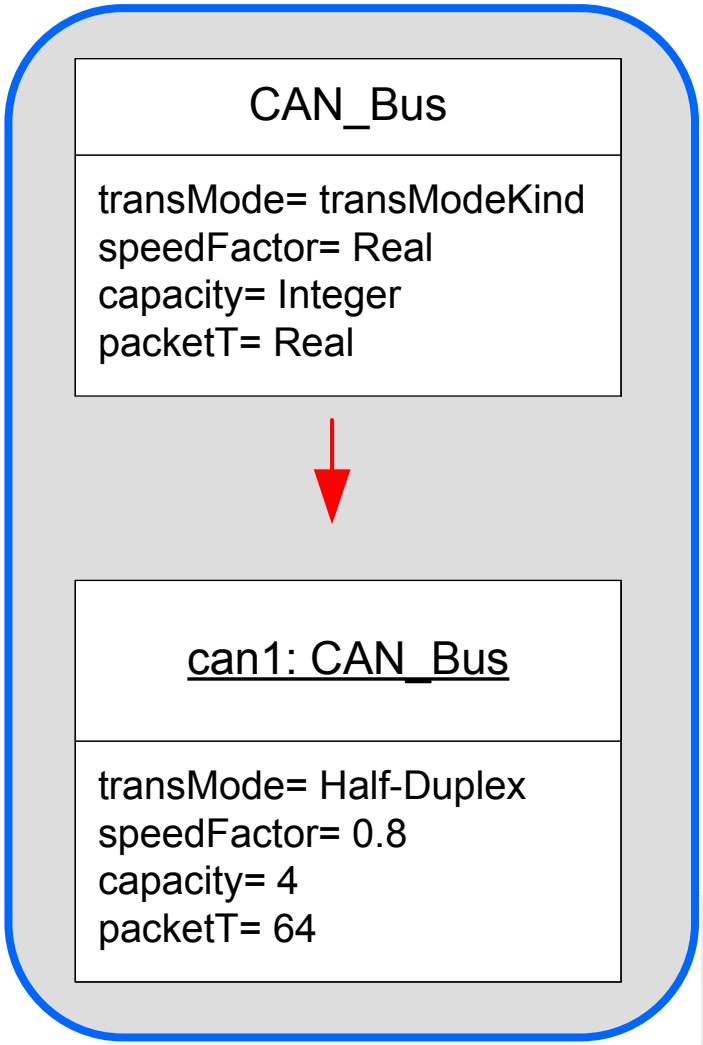
3) Specify NFP values

- **Instantiate classifiers and specify their slot values using VSL**

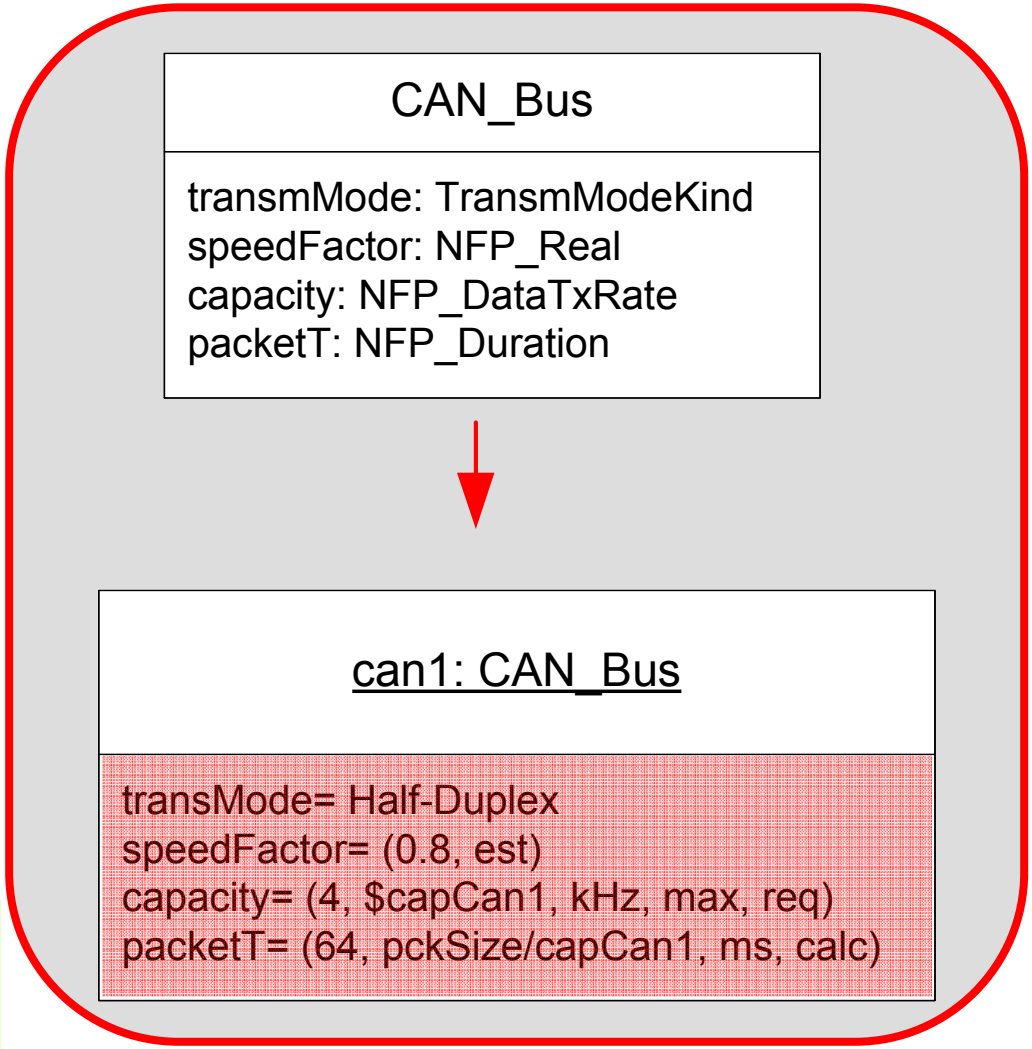


Extending UML Expressiveness for NFPs

CAN Bus with **Pure UML**



CAN Bus with **MARTE's NFP**



Annotating NFPs in Tagged Values

1) Declare NFP types

- Define measurement units and conversion parameters

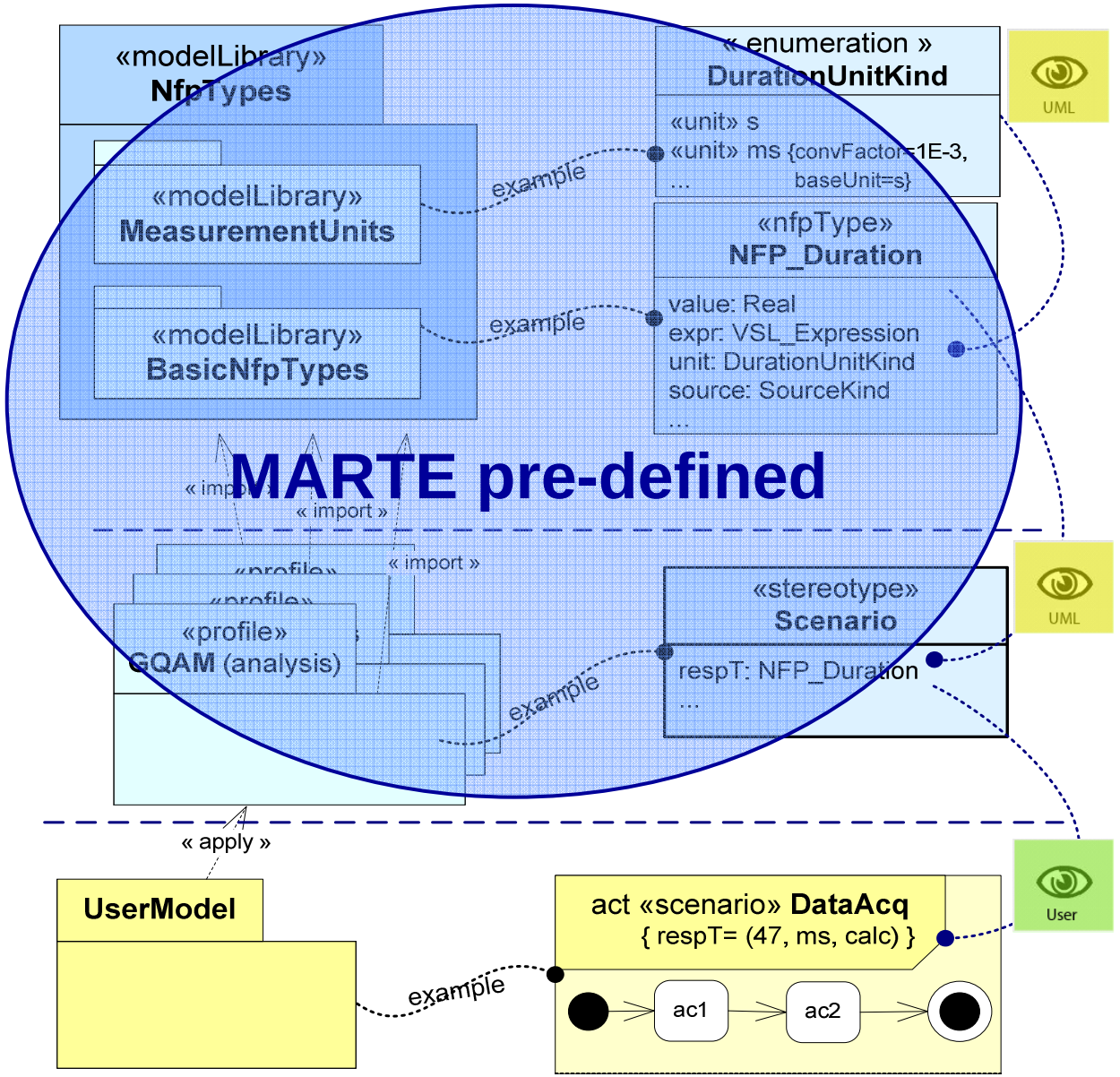
- Define NFP types with qualifiers

2) Define NFP-like extensions

- Define stereotypes and their attributes using NFP types

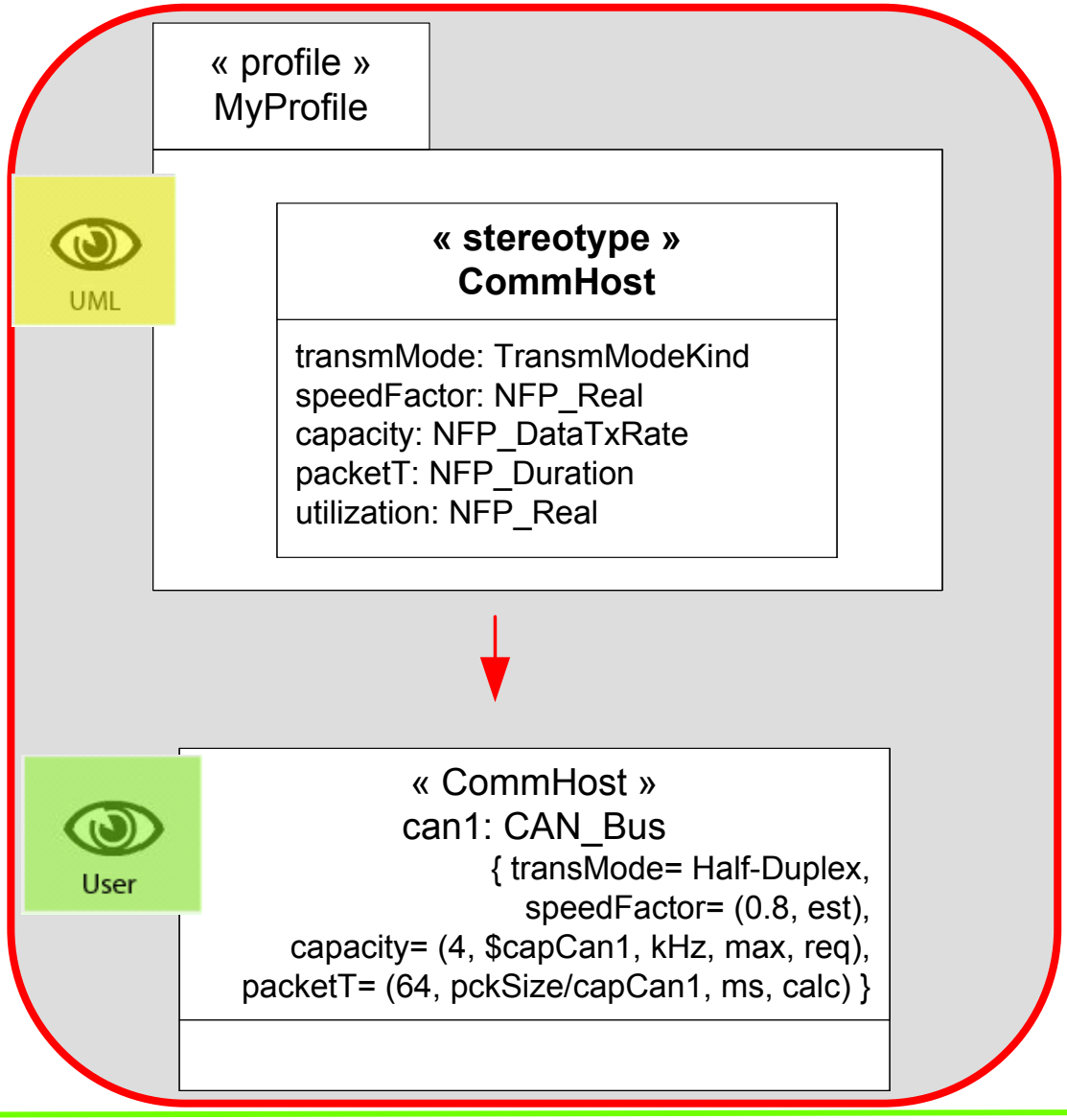
3) Specify NFP values

- Apply stereotypes and specify their tag values using VSL



Annotating NFPs in Tagged Values: Example

CAN Bus annotated with Stereotypes



Annotating NFPs in Constraints

1) Declare NFP types

- Define measurement units and conversion parameters

- Define NFP types with qualifiers

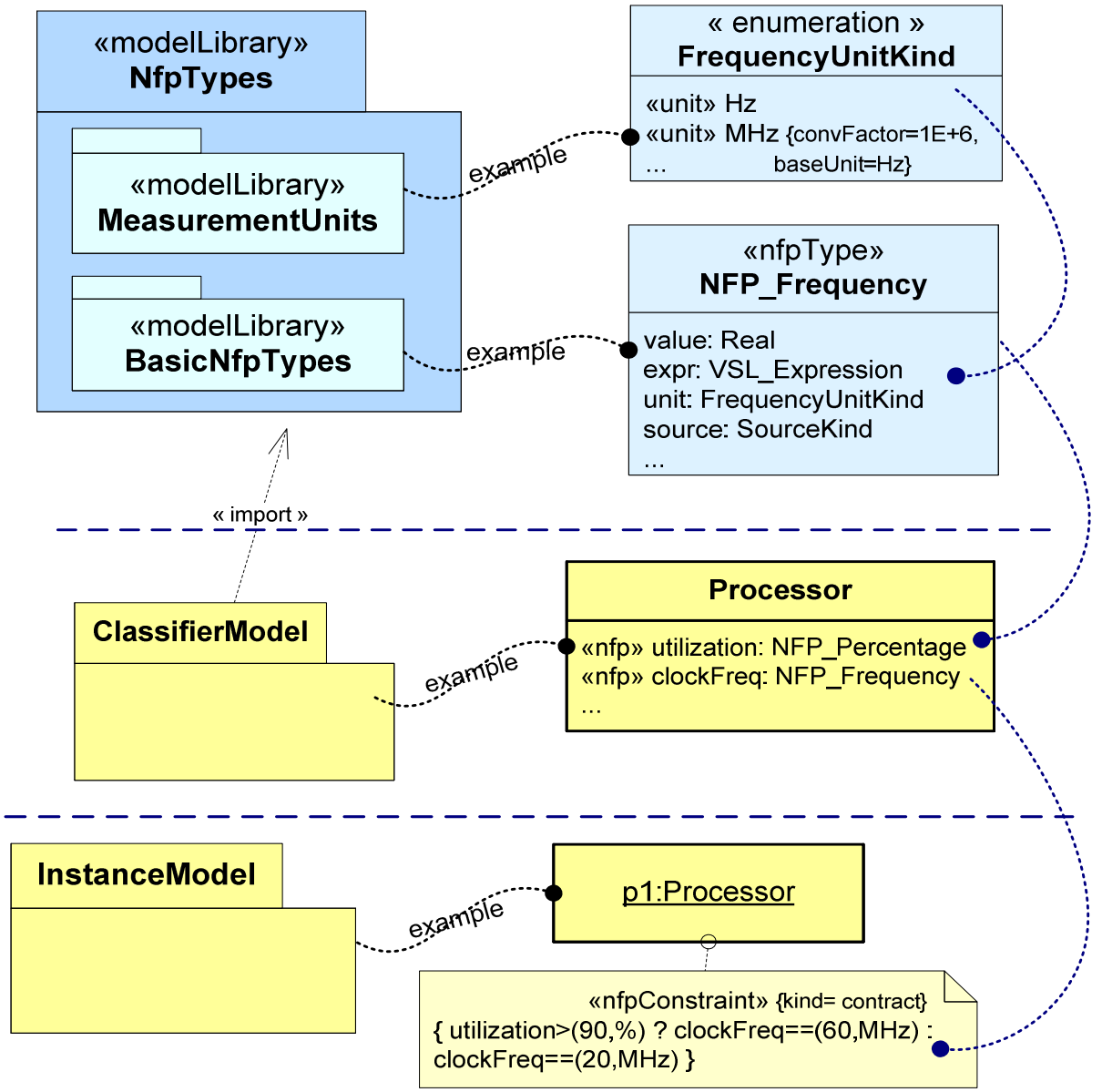
2) Declare NFPs

-Define classifiers and their attributes using NFP types

3) Specify NFP values

-Create Constraints to define assertions on NFP values using VSL

-«nfpConstraint» is a *required*, *offered*, or *contract* constraint of NFPs

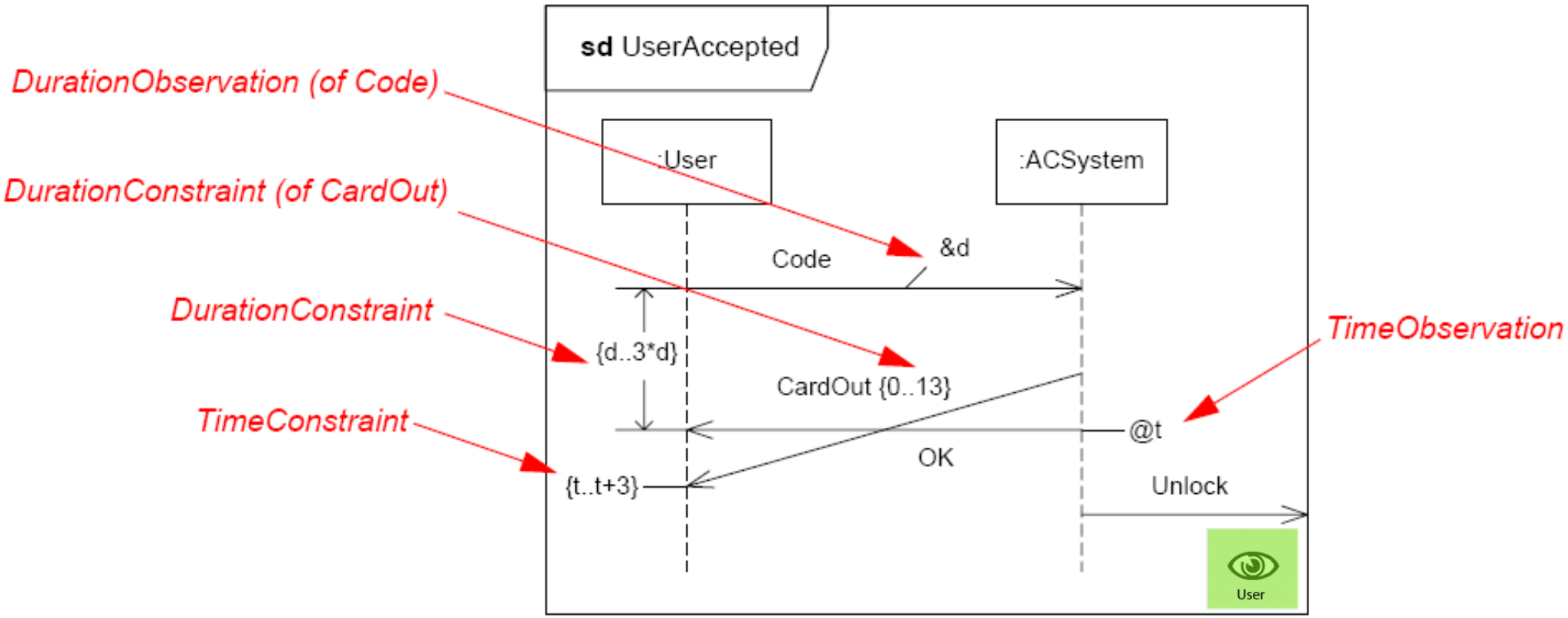


UML (native) Observation Concept

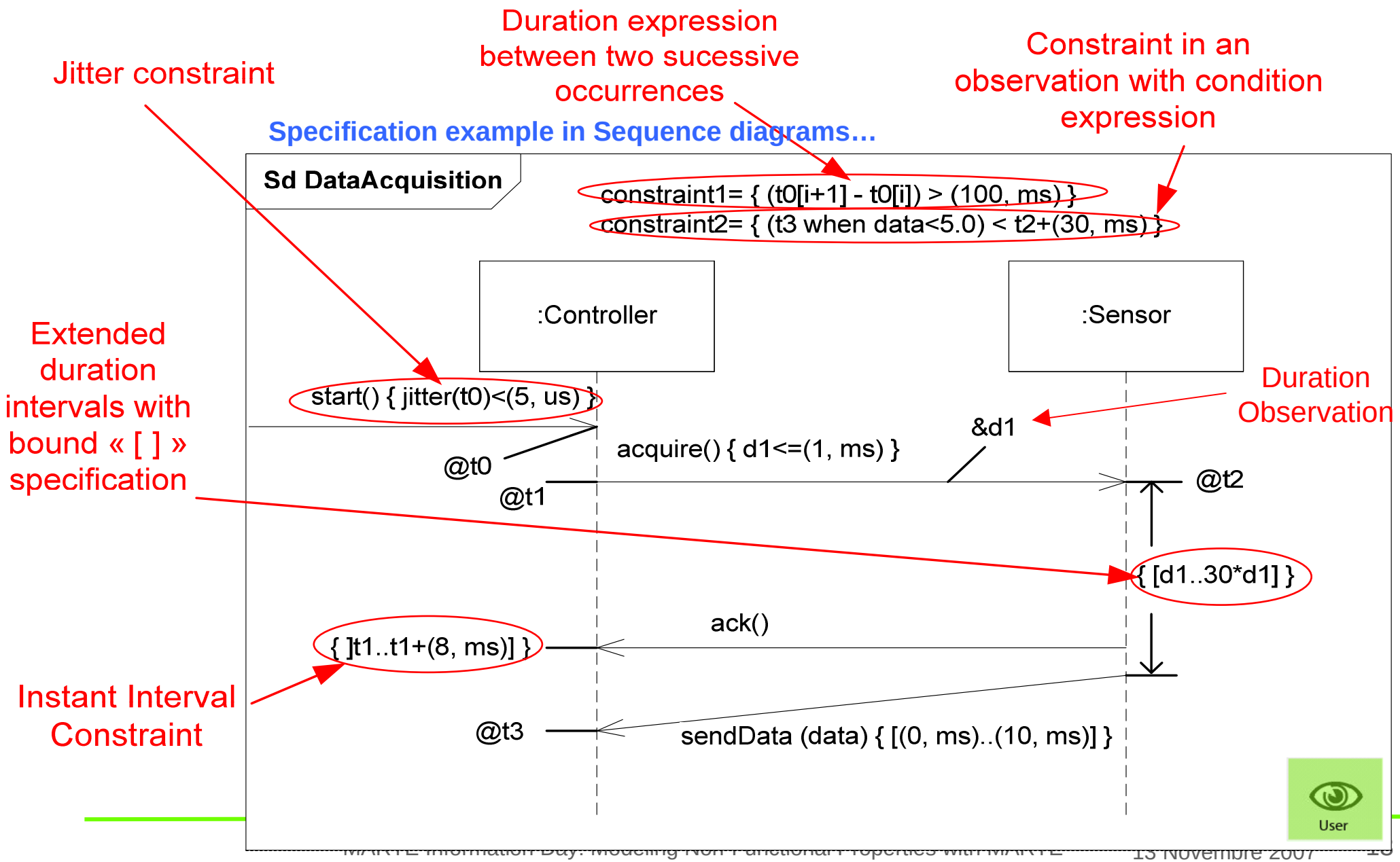
An time observation is a reference to a time instant during an execution.

An duration observation is a reference to a time interval during an execution.

Specification example in Sequence diagrams…



Time Expressions with VSL

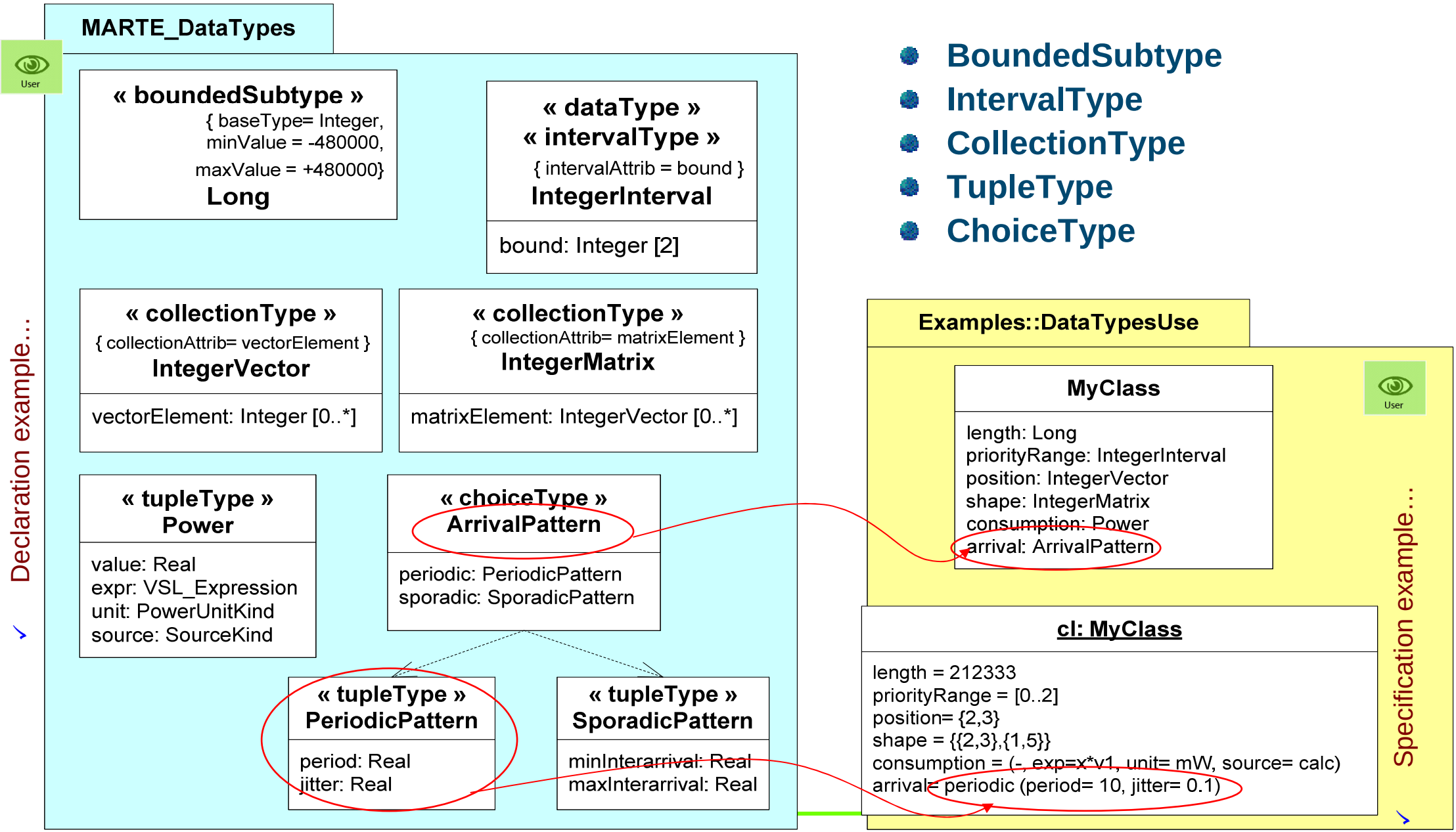


Basic Textual Expressions in VSL

- **Extended Primitive Values**
- **Extended Composite Values**
- **Extended 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 opened 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`

VSL Extended Data Types



- BoundedSubtype
- IntervalType
- CollectionType
- TupleType
- ChoiceType

VSL Tool Support

- Plugin For UML2 Eclipse (CEA LIST):
  - ➔ Completion assistant, syntax checker and type checker
  - ➔ Open Source for Papyrus Graphical Editor (www.papyrusuml.org).
- VSL for RSA (Thales TRT):
  - ➔ Completion assistant, syntax checker and type checker
  - ➔ Open source: www.omgmarte.org

Summary of the NFP Modeling Framework

1. Usability vs. Flexibility:
   - ➔ Three annotation mechanisms: Stereotypes, Properties and Constraints
   - ➔ Stereotypes: predefined NFPs (e.g., end-to-end latency, processor utilization)
   - ➔ Properties & Constraints: user-specific NFPs (but still unambiguously interpreted)
2. Synthesis of best modeling practices…
   - ➔ Reuse OCL constructs: grammar for values and expressions
   - ➔ Formally defined by abstract (metamodel) and concrete (grammar) syntaxes➔ Can be implemented as non-UML based language
   - ➔ VSL supports time expressions (occurrence index, jitters,...)

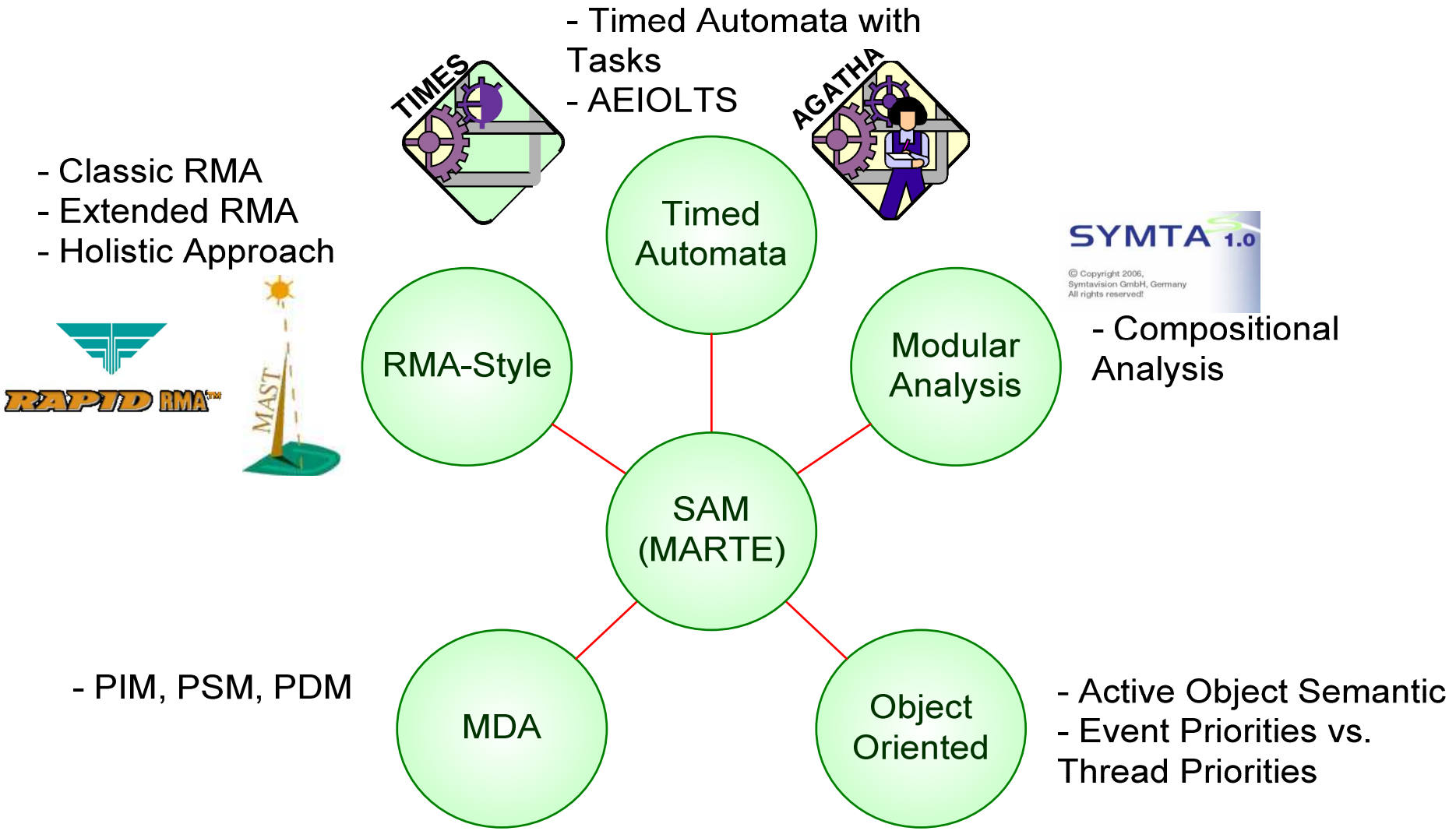
Thank you!

Questions?

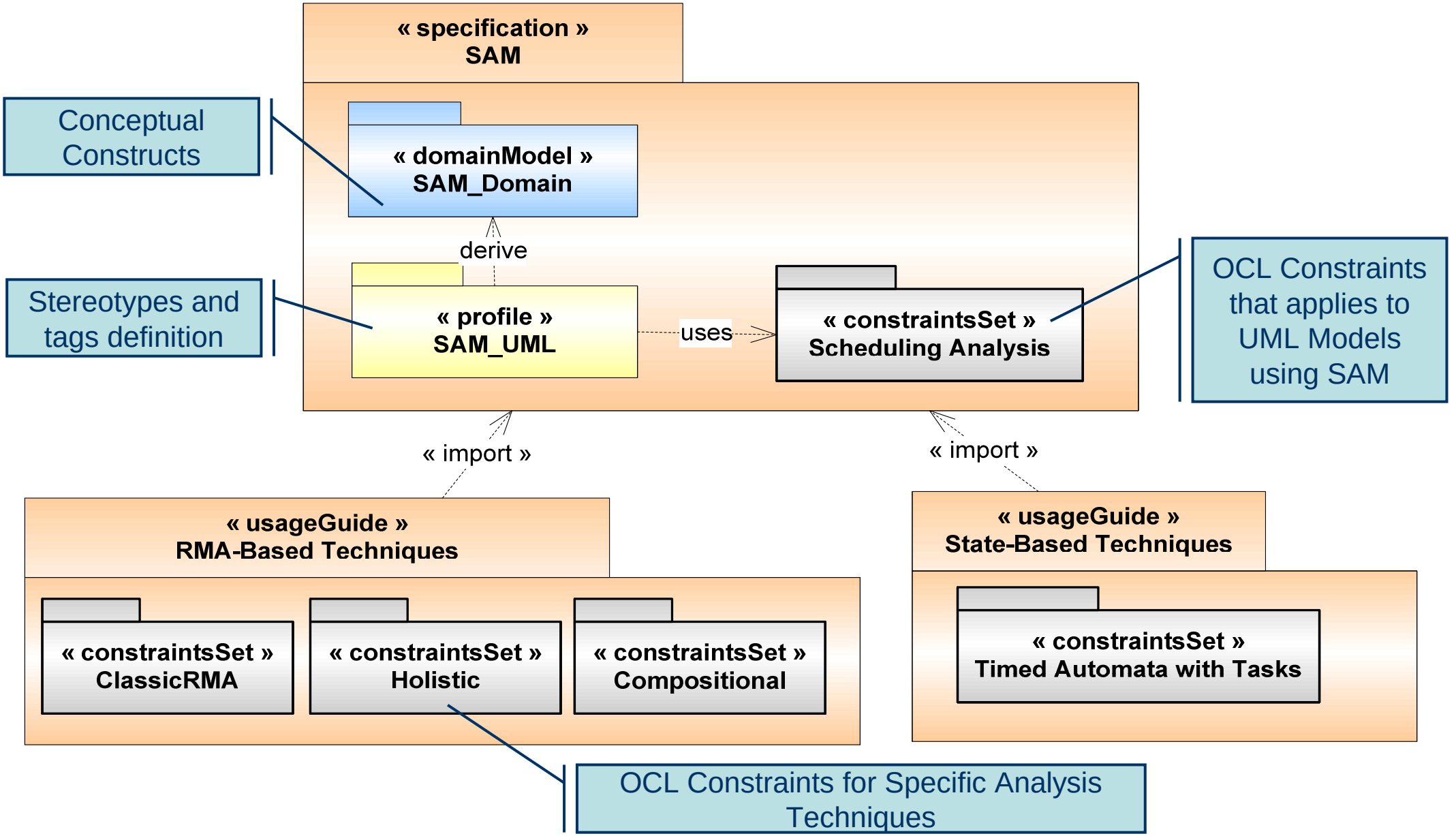
Demo

Schedulability Analysis Modeling with MARTE

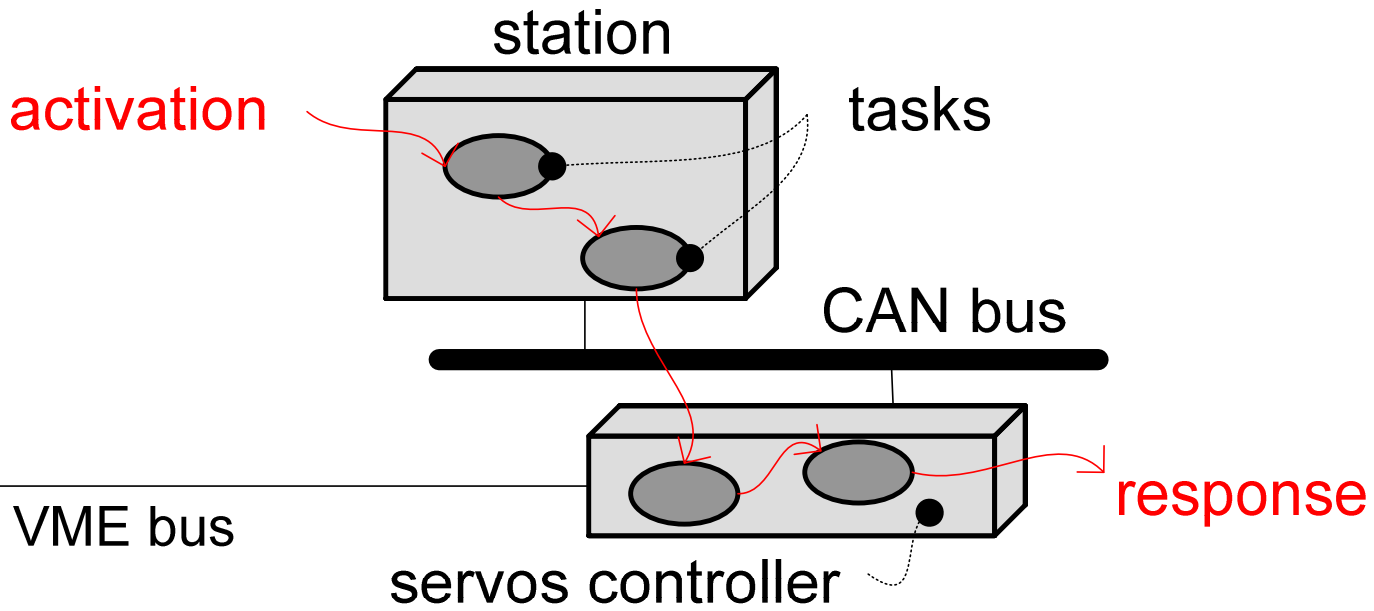
Schedulability Analysis Modeling Profile



Constraining SAM for Specific Techniques



Example: A Teleoperated Robot



- Industrial Robot Controllers project (University of Cantabria)
- 3 end-to-end flows
  - ➔ Refresh Report, Control Arm, Send Command
  - ➔ Periodic activation (5 ms, 100 ms, 1 s)
- Evaluate end-to-end deadlines
  - ➔ Deadlines = activation period

General Method to Use the SAM Profile