SAME 2009 Forum

Keynote Speaker

MARTE: The OMG UML2 Profile for Modeling and Analysis of Real-Time and Embedded systems

Frédéric Mallet
Aoste Team-project – INRIA/I3S
Université de Nice Sophia Antipolis

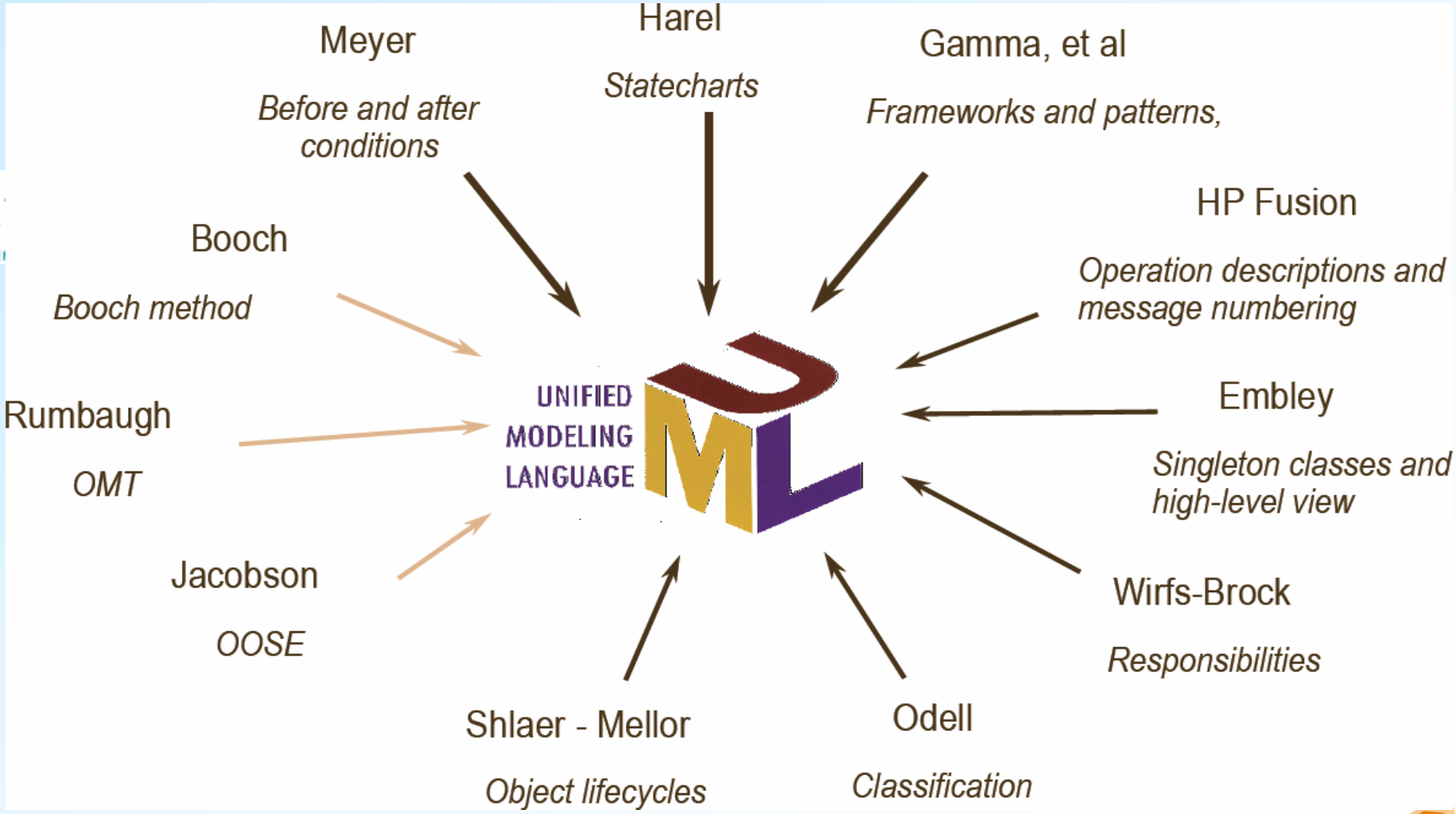
INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE
INRIA
centre de recherche SOPHIA ANTIPOLIS – MÉDITERRANÉE

cnrs

Université Nice SOPHIA ANTIPOLIS

i3s sophia antipolis

Unified Modeling Language (UML)



Domain-specific profiles

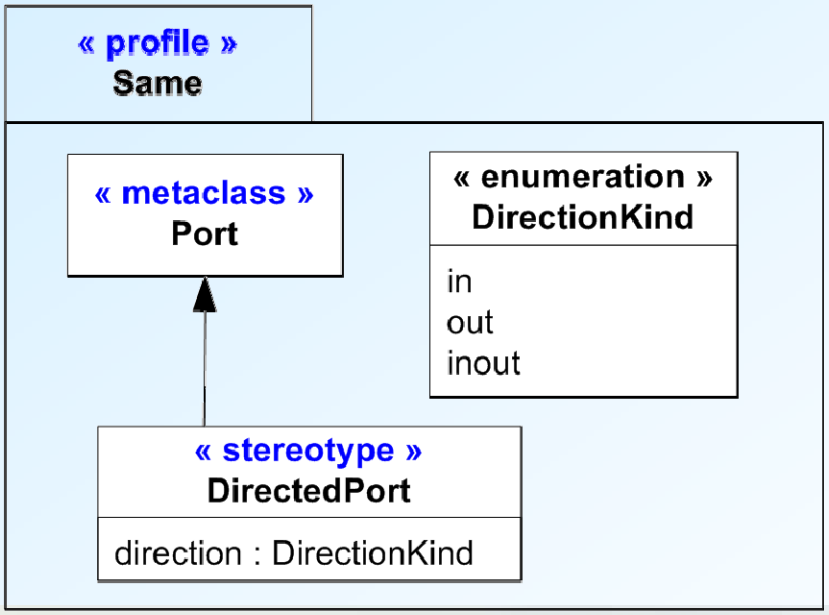
- **Advantages** of using UML profiles
  - Reuse of language infrastructure (tools, specifications)
  - Reduce training time of engineers
  - Allow for new (graphical) notation of extended stereotypes
  - A profile can define model viewpoints
    - E.g., UML activity diagram extended to specify multitask behavior
- **Disadvantages**
  - Constrained by the UML metamodel
  - 'light-weight‘ extensions

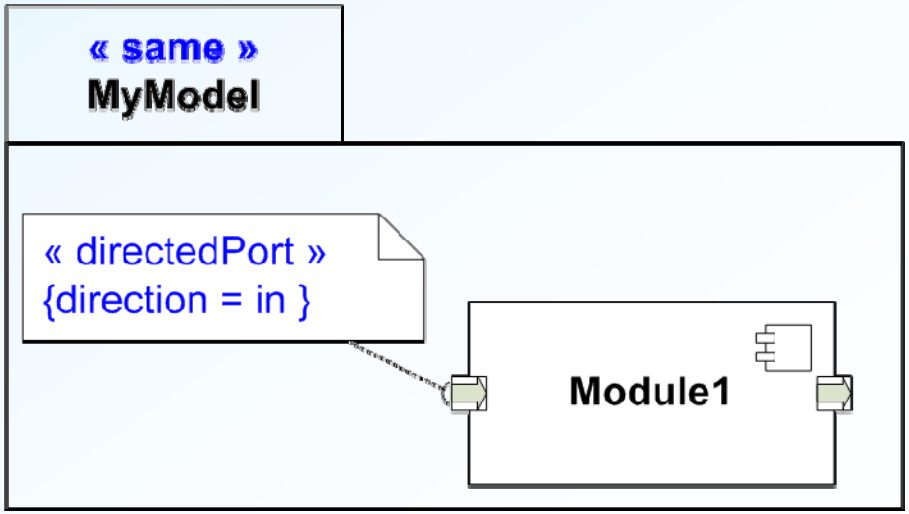
Profiles and stereotypes

- **Profiles**
  - Extend a metamodel for a specific platform or domain
  - Consist of stereotypes that extend the metamodel classes
- **Stereotypes**
  - Introduce platform-specific **terminology/syntax/notation**
    - E.g., IP or VC instead of Class/Component
    - E.g., Use an icon for a processor instead of a generic node
  - Provide **additional semantics**, but only for
    - Semantic restriction or clarification of existing concepts
    - New features (but compatible with existing ones !)
    - E.g., what happens with simultaneous events

Defining a stereotype: DirectedPort

- The **stereotype** DirectedPort refers to a **Port**
- It has only one **property**
  - The direction, whose type is specified by an Enumeration





SPT - Schedulability, Performance & Time

- **SPT was the first OMG UML profile for Real-Time Systems**
  - RMA-like Schedulability Analysis
  - Performance Analysis with Queueing Theory and Petri Nets
  - A model for “metric” Time and Time Mechanisms
- **Several improvements required**
  - Modeling HW and SW platforms, Logical Time, MoCCs
  - Alignment to UML2 (Simple Time), QoS&FT
  - SPT was considered too abstract and hard to apply

Hence, a Request For Proposal for a new profile was issued in 2005.

The ProMARTE Team

- **Industrials**
  - Alcatel*
  - Lockheed Martin*
  - Thales*
  - France-Telecom
- **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
- **Public website**

  www.omgmarte.org

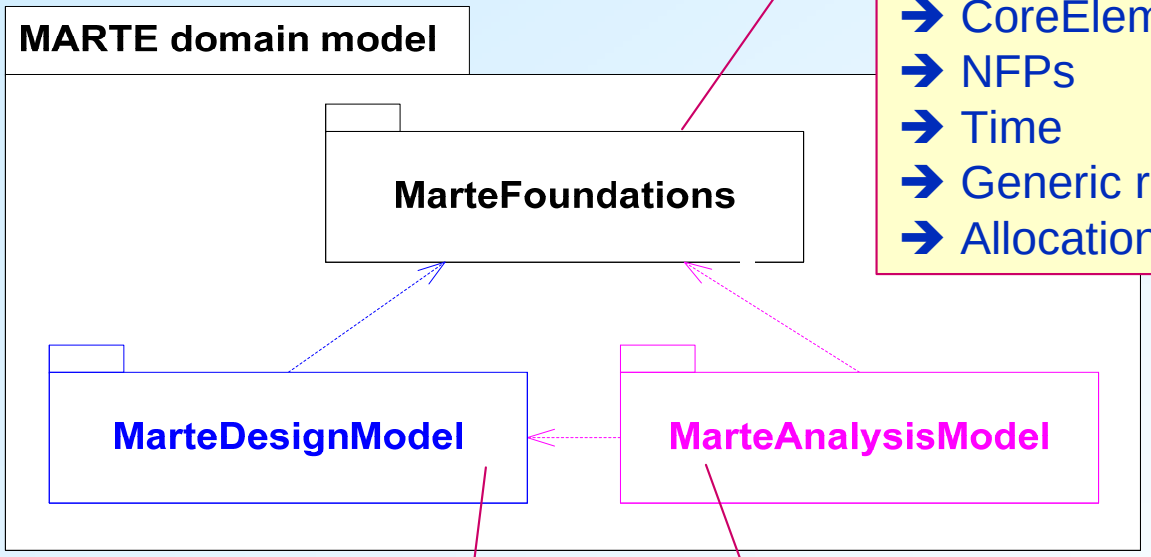
Relationships with other OMG Standards

- Relationships with generic OMG standards
  - **Extends** the UML2 superstructure meta-model
  - **Supersedes** UML Profile for SPT (Scheduling, Performance and Time)
  - **Uses** OCL2 (Object Constraint Language)
  - **Complements** the UML profile for Systems Engineering (SysML)
    - Specialization of SysML allocation concepts
    - Overlap of team members

Relationships with other OMG Standards

- Relationships with RT&E specific OMG standards
  - The UML profile for Modeling QoS and FT Characteristics and Mechanisms
    - Partially **covered by** MARTE NFP package
  - The UML profile for SoC (System On Chip)
    - **More specific** than MARTE purpose
  - The Real-Time CORBA profile
    - Real-Time CORBA based architecture can be annotated for analysis with MARTE

MARTE Overview



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

Specification and Design
- ➔ High-Level application modeling
- ➔ Generic component modeling
- ➔ Software resource modeling
- ➔ Hardware resource modeling

Analysis
- ➔ Generic quantitative analysis
- ➔ Schedulability analysis
- ➔ Performance analysis

Non-Functional Properties (NFP)

Includes VSL

(Value Specification Language)

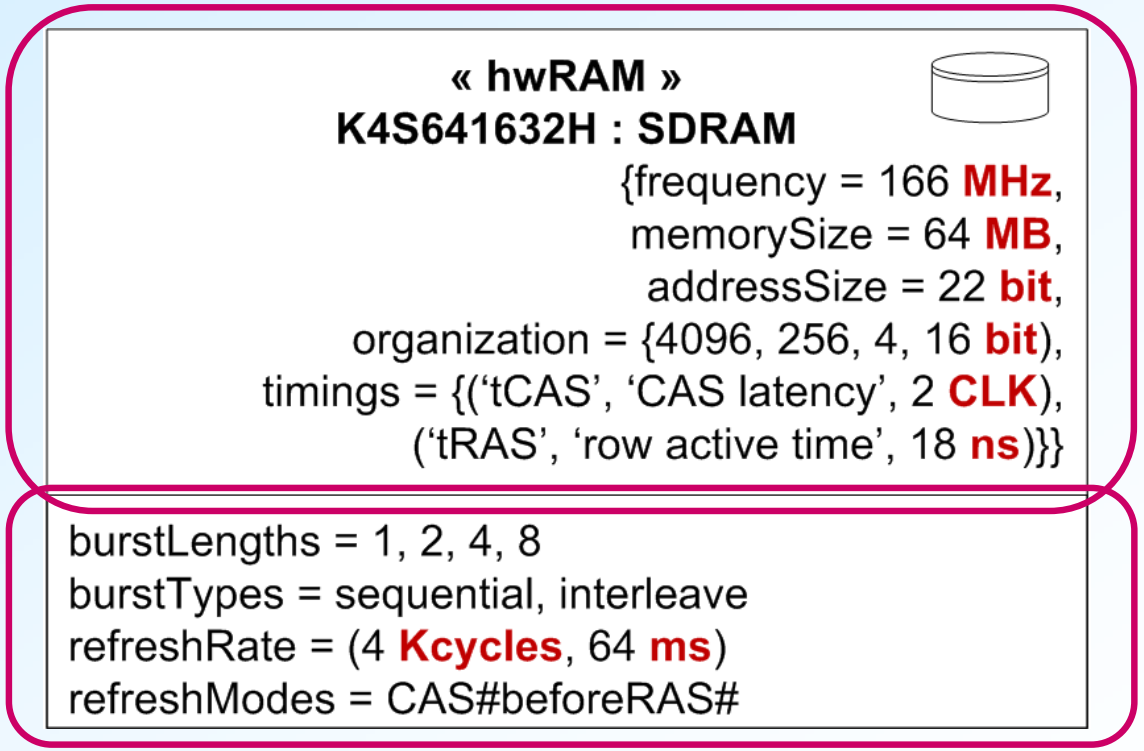
Non-Functional Properties

(E.g., performance, memory usage, power consumption)

- Nature of NFPs
  - Quantitative: magnitude + unit (E.g., energy, data size, duration)
    - Quantity-Unit-Dimension for dimensional analysis (from IEEE)
  - Qualitative (E.g., periodic or sporadic event arrival patterns)
- NFP values are qualified (required/provided/measured/estimated)
- NFP introduces variables and expressions
- NFPs have a precise syntax: ANTLR parser
  - Predefined NFPs (E.g., end-to-end latency, processor utilization)
  - User-specific NFPs (Parser generated by introspection)

Where to use NFP values ?

- Two kinds of non-function properties
  - Predefined stereotype properties
  - User-defined properties



predefined

User-defined

NFPs in Stereotype properties

Pre-defined in MARTE

1) Declare NFP types

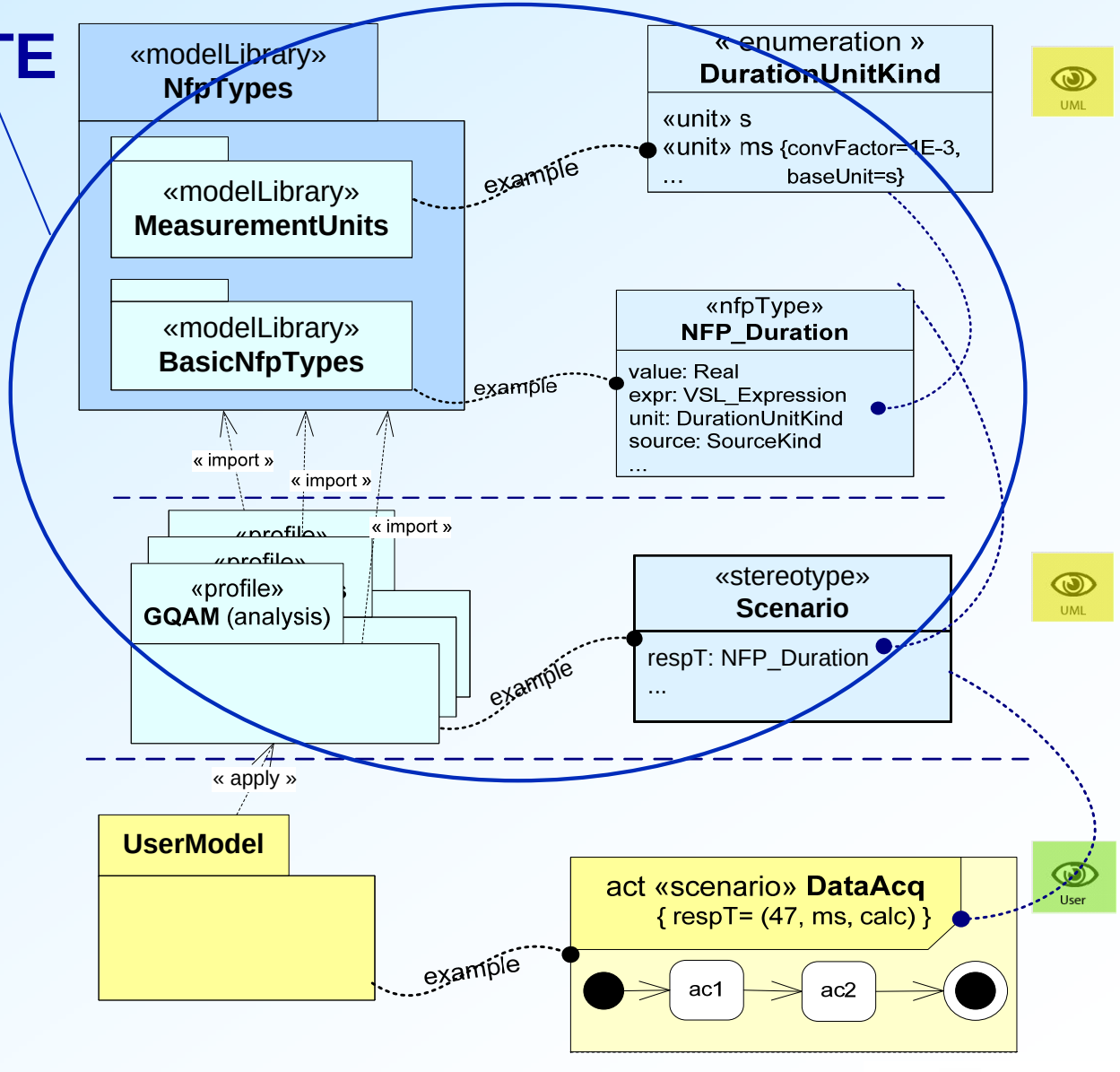
- Define measurement units and conversion parameters
- Define NFP types with qualifiers

2) Define NFP extensions

- Define stereotypes and their properties using NFP types

3) Specify NFP values

- Apply stereotypes and specify their properties using VSL



User-defined NFPs in slots

1) Declare NFP types

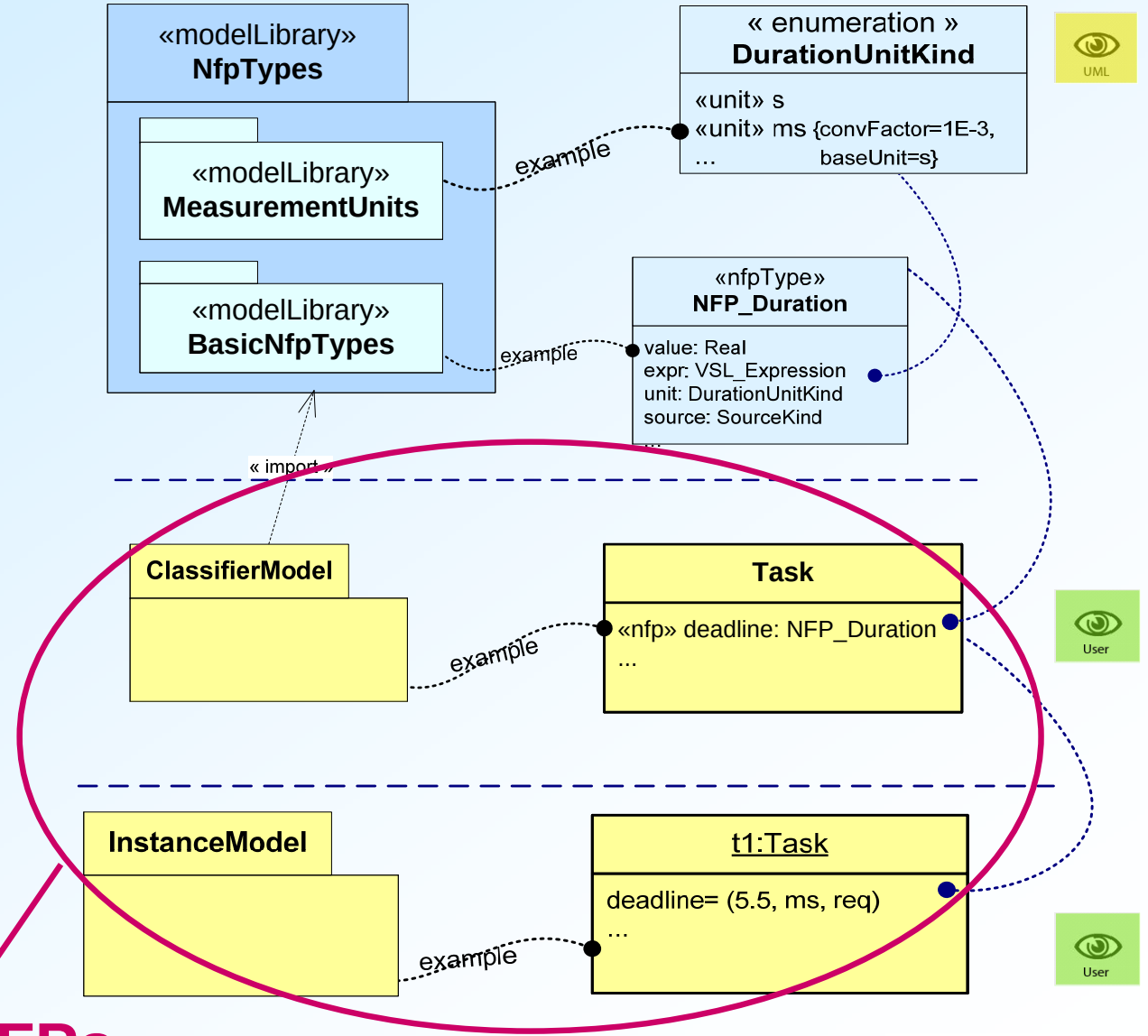
- Define measurement units and conversion parameters
- Define NFP types with qualifiers

2) Define NFP

- Define classifiers and their attributes using NFP types
- Apply the stereotype «nfp»

3) Specify NFP values

- Build instances and specify their slot values using VSL



Model-specific NFPs

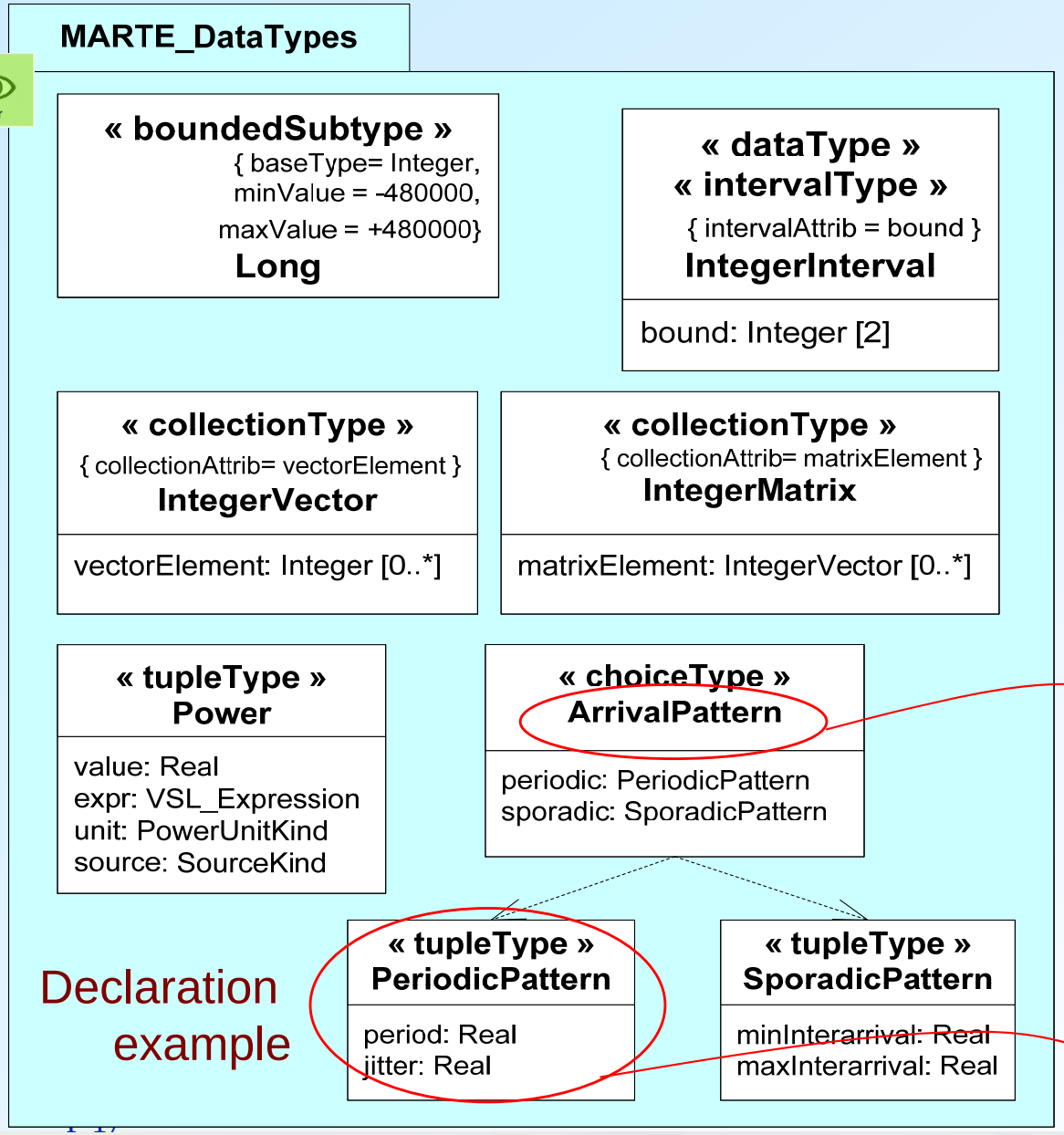
Basic Textual Expressions

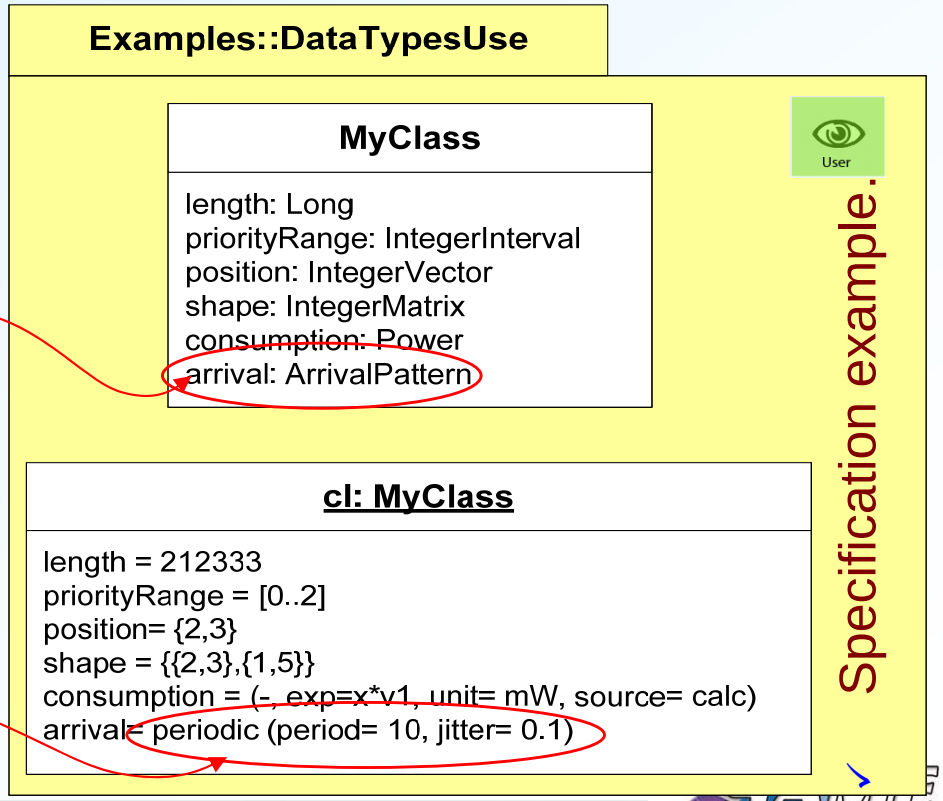
- Extended Primitive Values,
- Extended Composite Values,
- Extended Expressions

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





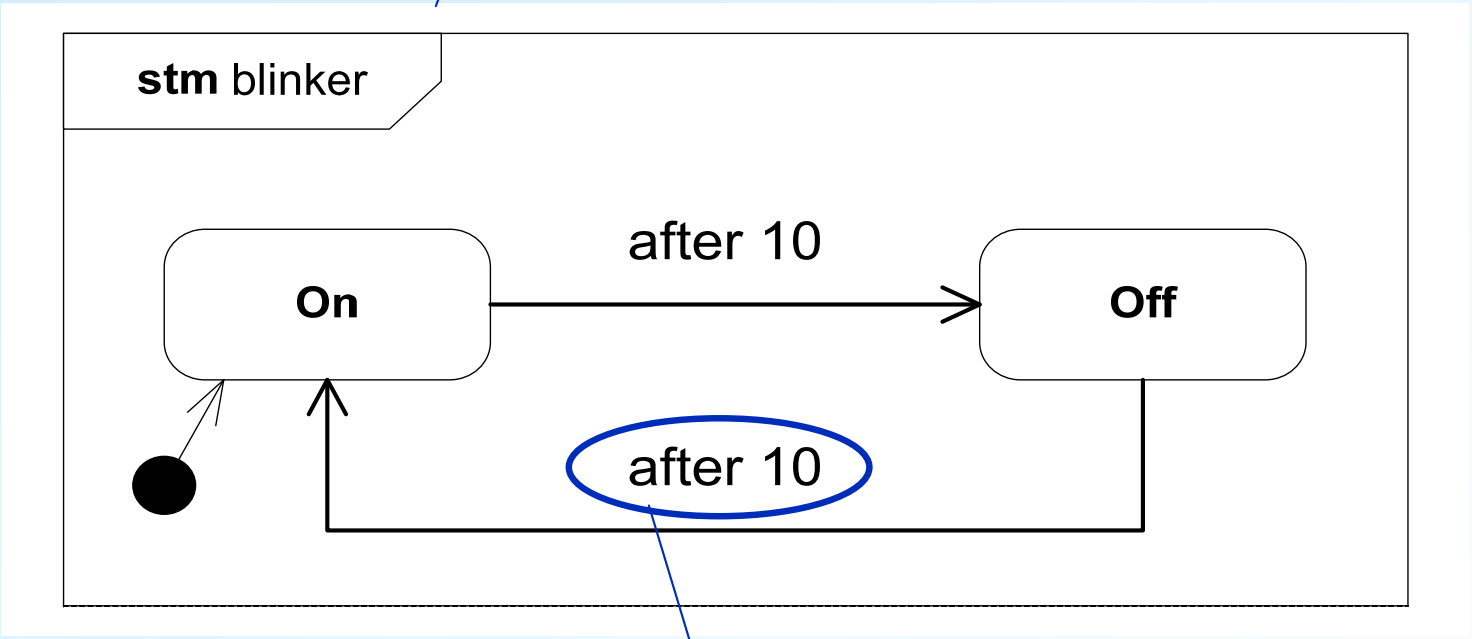
MARTE Time Model

UML::CommonBehaviors::SimpleTime

- UML2 adds **new metaclasses** to represent
  - Time
  - Duration
  - Observation (of time passing)
  - Some forms of time constraints
- **Simple** (even simplistic) model of time
- **Advice**: *Use a more sophisticated model of time provided by an **appropriate profile***, if needed.
  [UML superstructure, chapter 13]

TimeEvent – usage

UML state machine = behavior

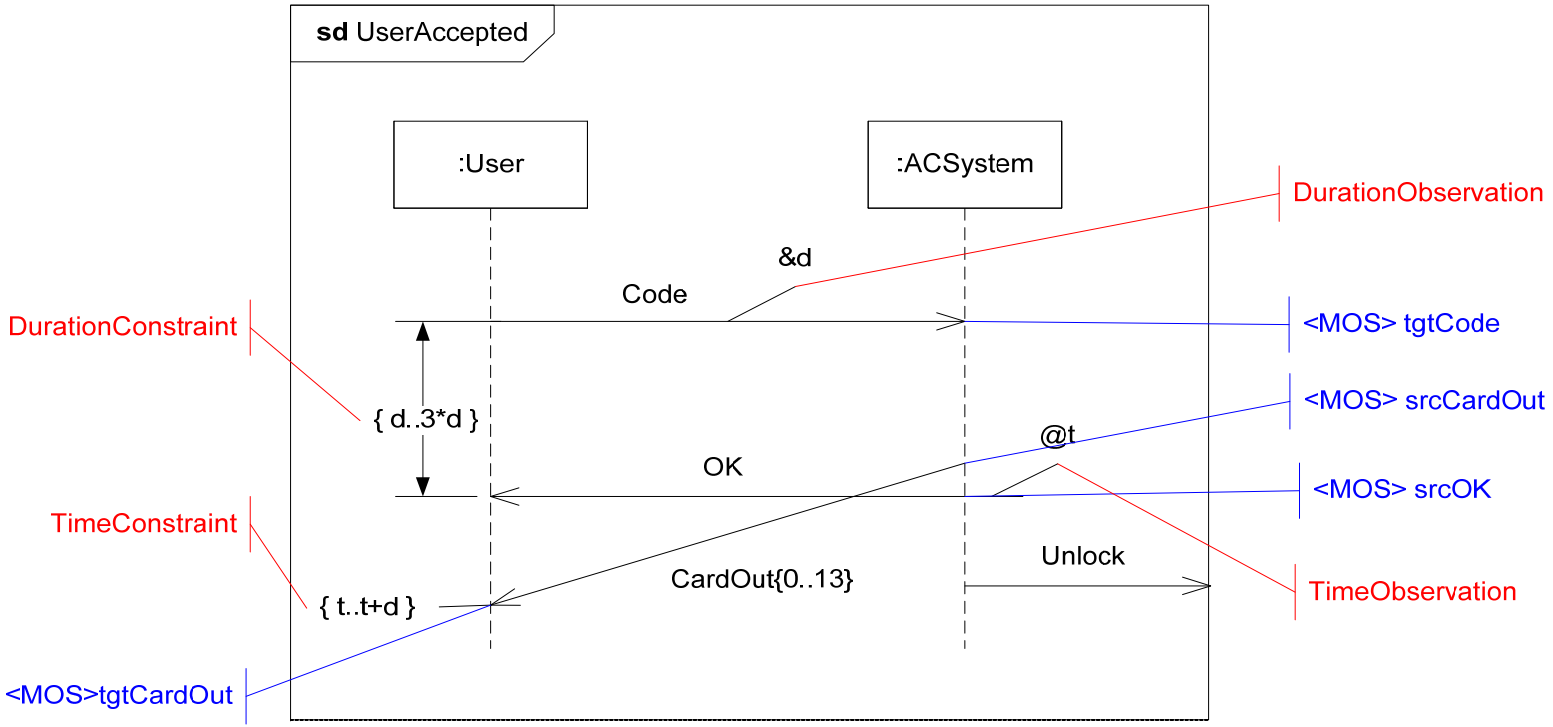


Specification of a time-trigger

Informal semantics

Observation – usage

Example of sequence diagram



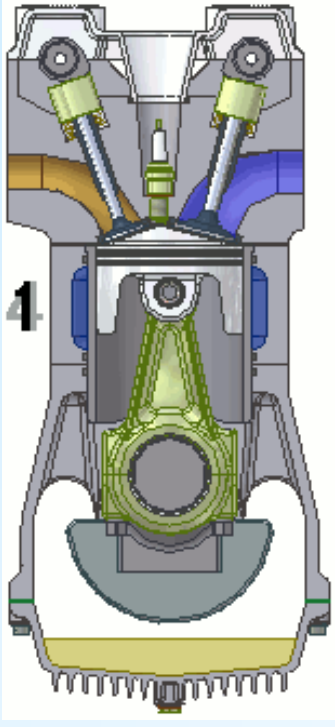
MOS stands for MessageOccurrenceSpecification

Note that red and blue annotations are not part of the UML notation.

Concepts in MARTE::Time

- **Time structure**: set of time bases + time structure relations
  → Partially ordered set of instants
- **Access to time** = Clock
  - Physical/logical discrete/dense multiform time
- **Principle**: associate Clocks with model elements
  - Behavioral elements → TimedEvent, TimedProcessing
  - Constraints → TimedConstraint
  - Data types and values → TimedValue

Logical multiform time (1/4)



- Modeling the “knock”
  - “time” properties expressed in rotation angles of the crankshaft
  - The actual “physical” time depends on the rotation speed and can be complex.

http://fr.wikipedia.org/wiki/Fichier:4-Stroke-Engine.gif

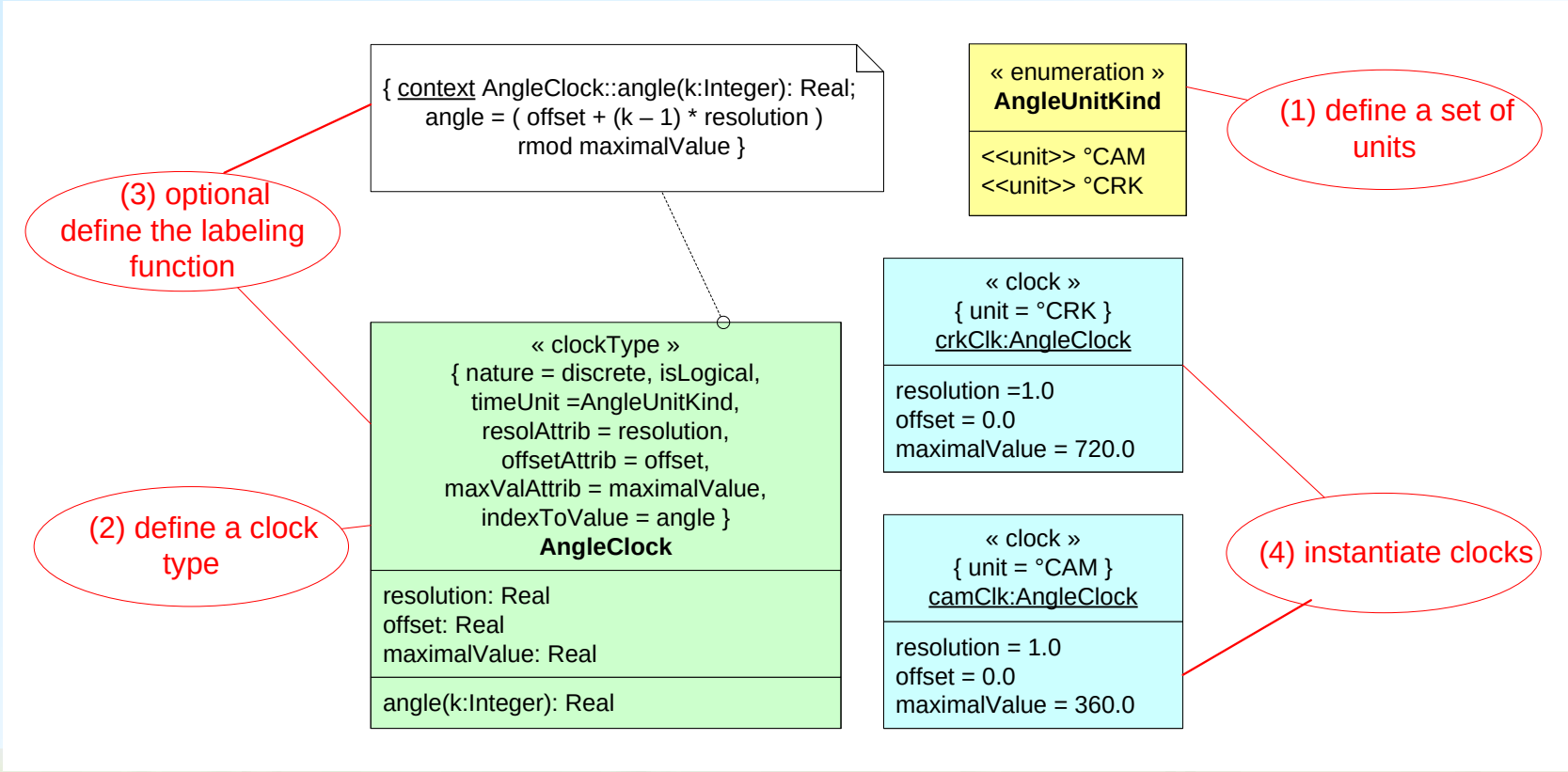
Multiform time (2/4)

An example of logical clocks

Automotive application

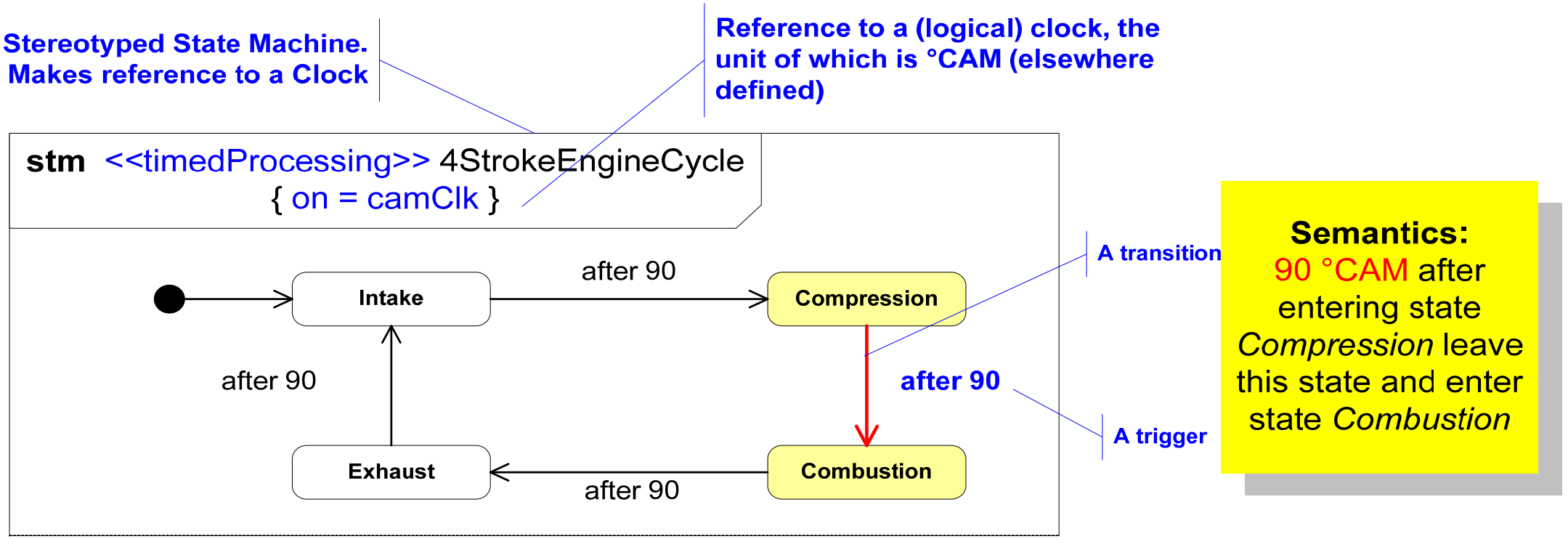
For ignition and injection, the position of the camshaft or the crankshaft is a “natural” reference frame for events and behaviors.

=> Define logical clocks dealing with angular positions.



Multiform time (3/4)

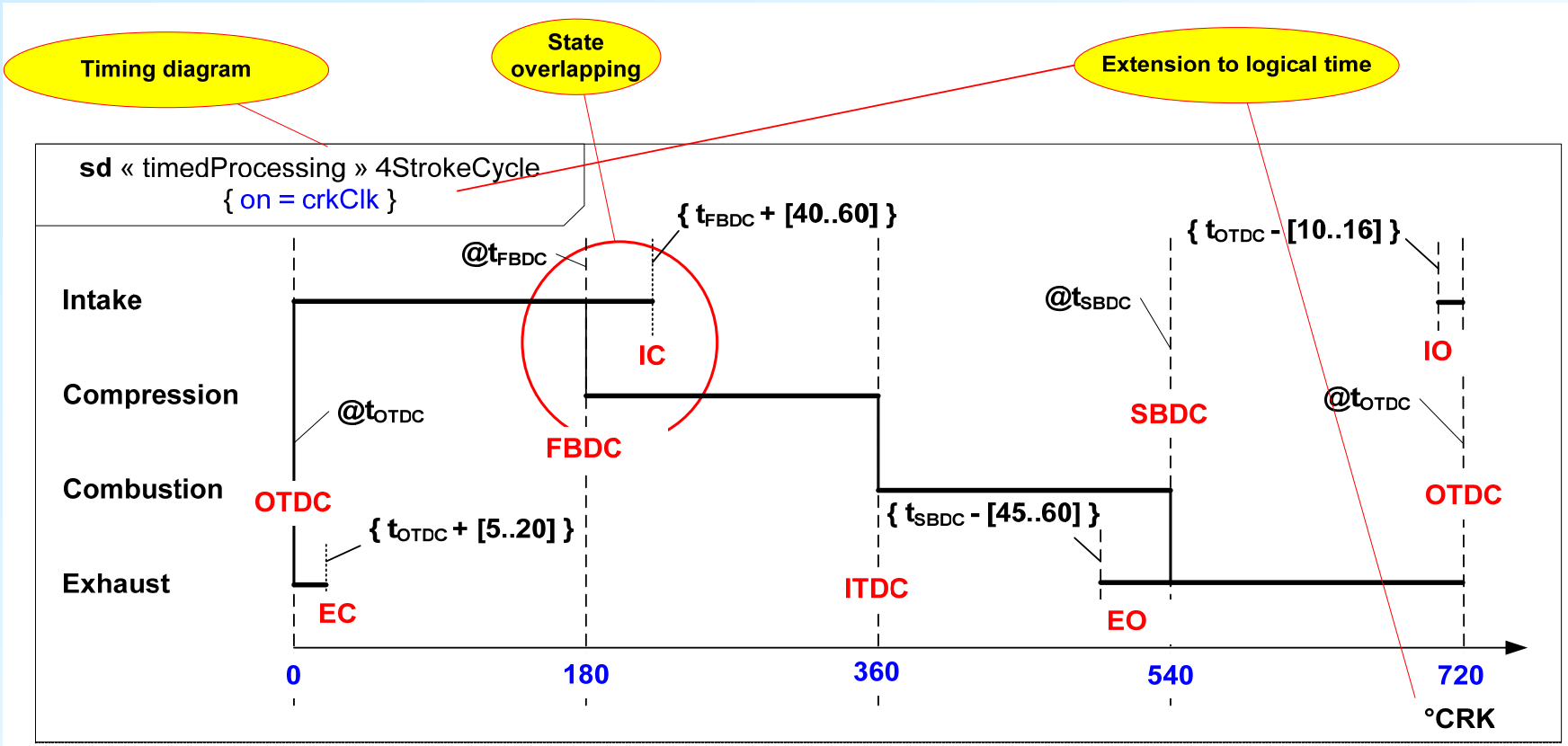
Example of usage of an “AngleClock”



Multiform time (4/4)

Another example of usage of an “AngleClock”:
Enhanced timing diagram used in specification

User



crkClk: crankshaft Clock
°CRK: degree crank

TDC: Top Dead Center
ITDC: Ignition TDC
OTDC: Overlap TDC

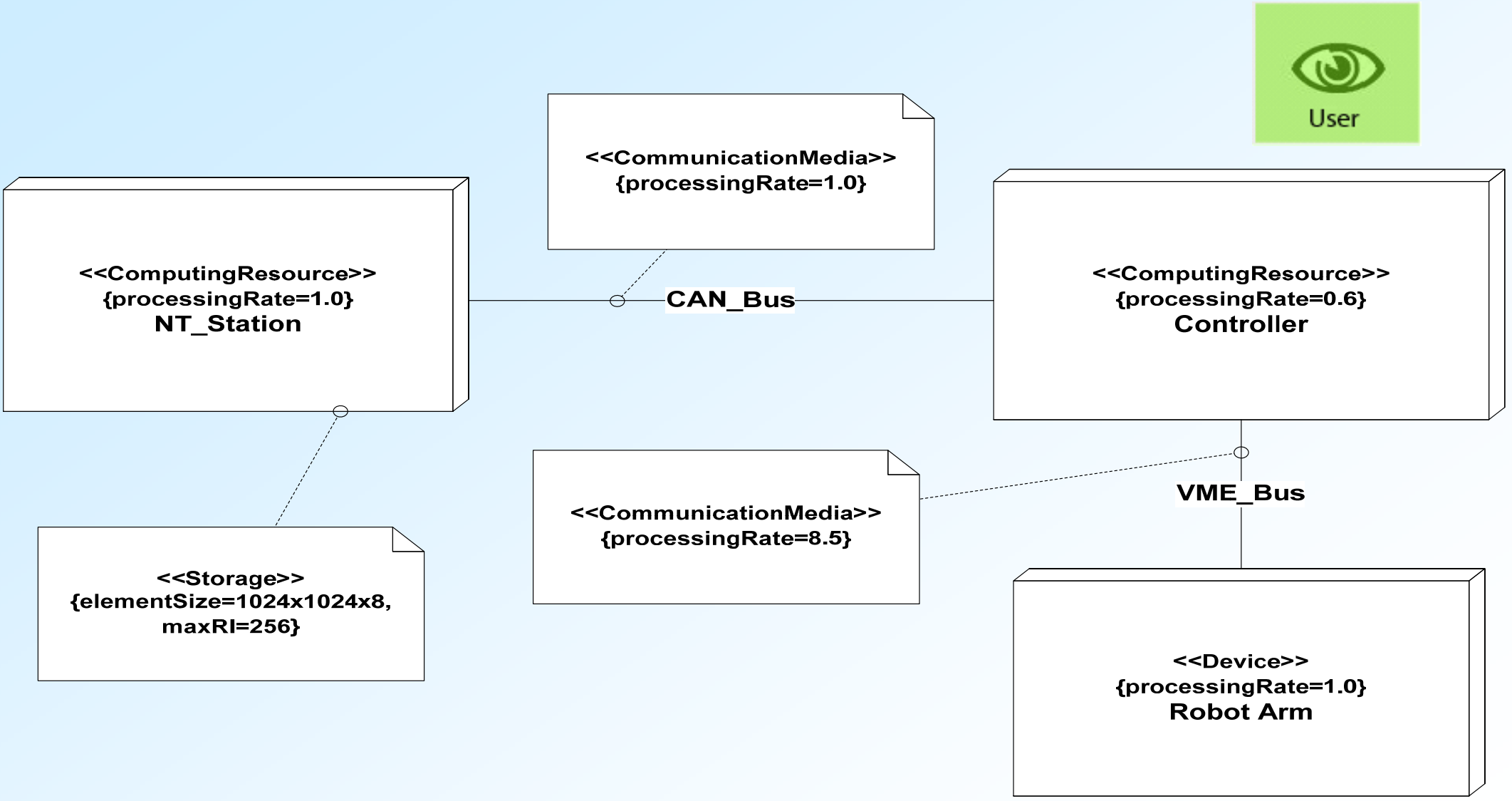
BDC: Bottom Dead Center
FBDC: First BDC
SBDC: Second BDC

IC: Intake closes
IO: Intake opens
EC: Exhaust closes
EO: Exhaust opens

Resource Modeling

- Generic Resource Modeling (**GRM**)
- Detailed Resource Modeling (**DRM**)
  - Software Resource Modeling (**SRM**)
  - Hardware Resource Modeling (**HRM**)

Generic resource modeling



Allocation & Refinement

- Allocation in MARTE
  - **Allocate** an application element to an execution platform element
    - **Application**: structure and behavior
    - **Execution platform**: structure and behavior # architecture
  - **Refine** a general element into one or several more specific elements
- Inspired from the SysML allocation
  - Can only allocate application to execution platform
  - Can attach NFP constraints to the allocation (cost)

Allocation in MARTE domain view

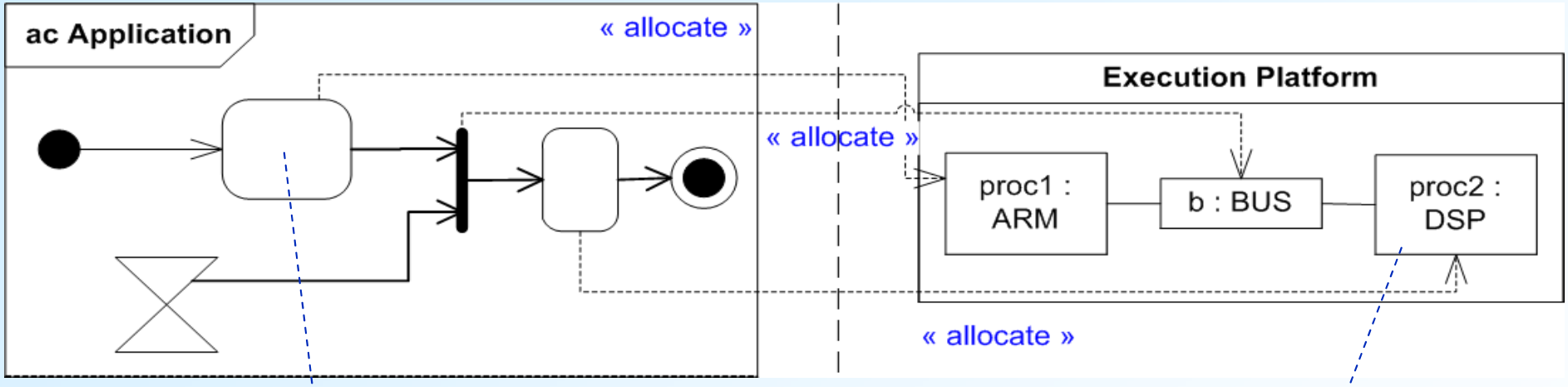
Allocation

Relationship between logical and physical time

Causality relations
Logical time

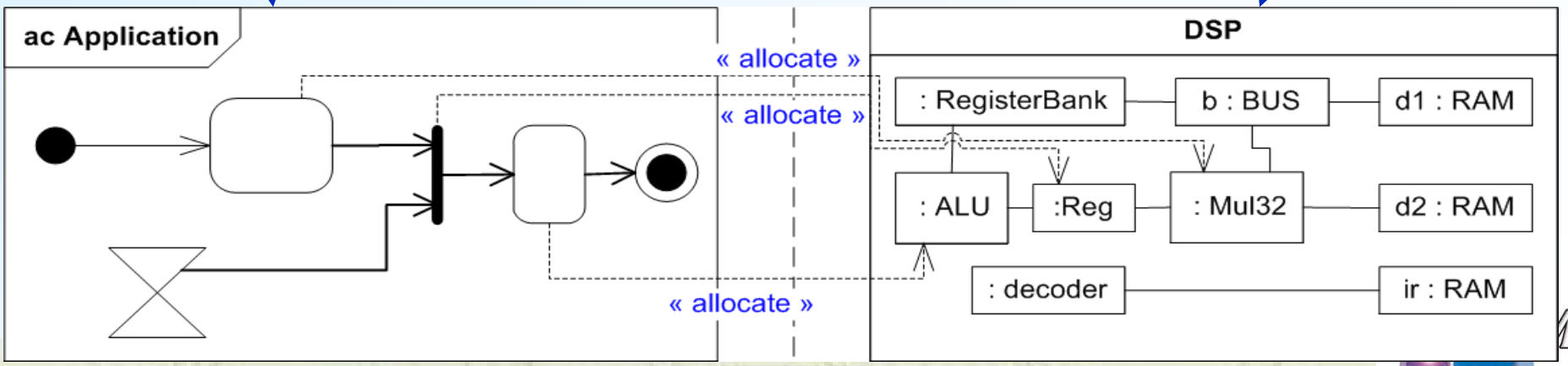
Physical time

Time/spatial refinement



« refinement »

« refinement »



MARTE Implementations

- **Currently available** (www.omgmarte.org)
  - Full MARTE Profile & Libraries for Eclipse UML2
  - VSL editor for UML Papyrus tool and RSA 7.0
  - TimeSquare
- **Transformations under development**
  - To MAST, SymTA/S, Cheddar, RapidRMA

MARTE Open Source Implementation in

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

Status

- **MARTE v1.0** has been adopted in June 2009
- **R**evision **T**ask **F**orce starts in July 2009 => v1.1
  - Implementation by tool vendors [within 2 years]
  - Up to now, only academic tools
  - Submit your issues to www.omg.org
- Expected results in July 2010
  - The RTF only addresses implementation issues

Perspectives

- Define methodologies to use MARTE
  - On-going for DoD
- Alignment with SysML
  - Allocation, NFP, Quantity-Unit-Dimension, Time
- Refinement for domain-specific usages
  - IP-Xact (IP interchange)
  - AutoSAR (Automotive)
  - AADL (Avionics)