

MARTE Infoday: The timing Model

An OMG standard:
UML profile to develop Real-Time and Embedded systems

(F. Mallet, INRIA Sophia Antipolis)
12/12/2007

- **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

- **From antiquity to nowadays**
 - Several models of time (philosophy, religion, science)
- **In Real-time and Embedded Systems**
 - Multiple time domains (multicore, NoC, low power)
 - Highly heterogeneous: unify multiple models in a single framework (MoCC)
 - ⇒ Theory of tag systems [Lee & Sangiovanni-Vincentelli]
- **In MARTE**
 - Simple mechanisms to deal with “classical” time issues
 - Advanced mechanisms to build your own MoCC

- **SPT, UML 2 and Time**
 - UML::CommonBehaviors::SimpleTime
- **the MARTE Time domain view**
 - a.k.a. the MARTE Time meta-model
 - Concepts and relationships
- **the MARTE Time sub-profile**
 - a.k.a. UML view
- **Usage of the Time sub-profile**

- **OMG UML profile formal/05-01-02 (v1.1)**

- **Based on UML 1.4**

To be aligned to UML 2

- **Dealing with time and resources**

- **Quantitative time information**

Metric time

- **Concepts**

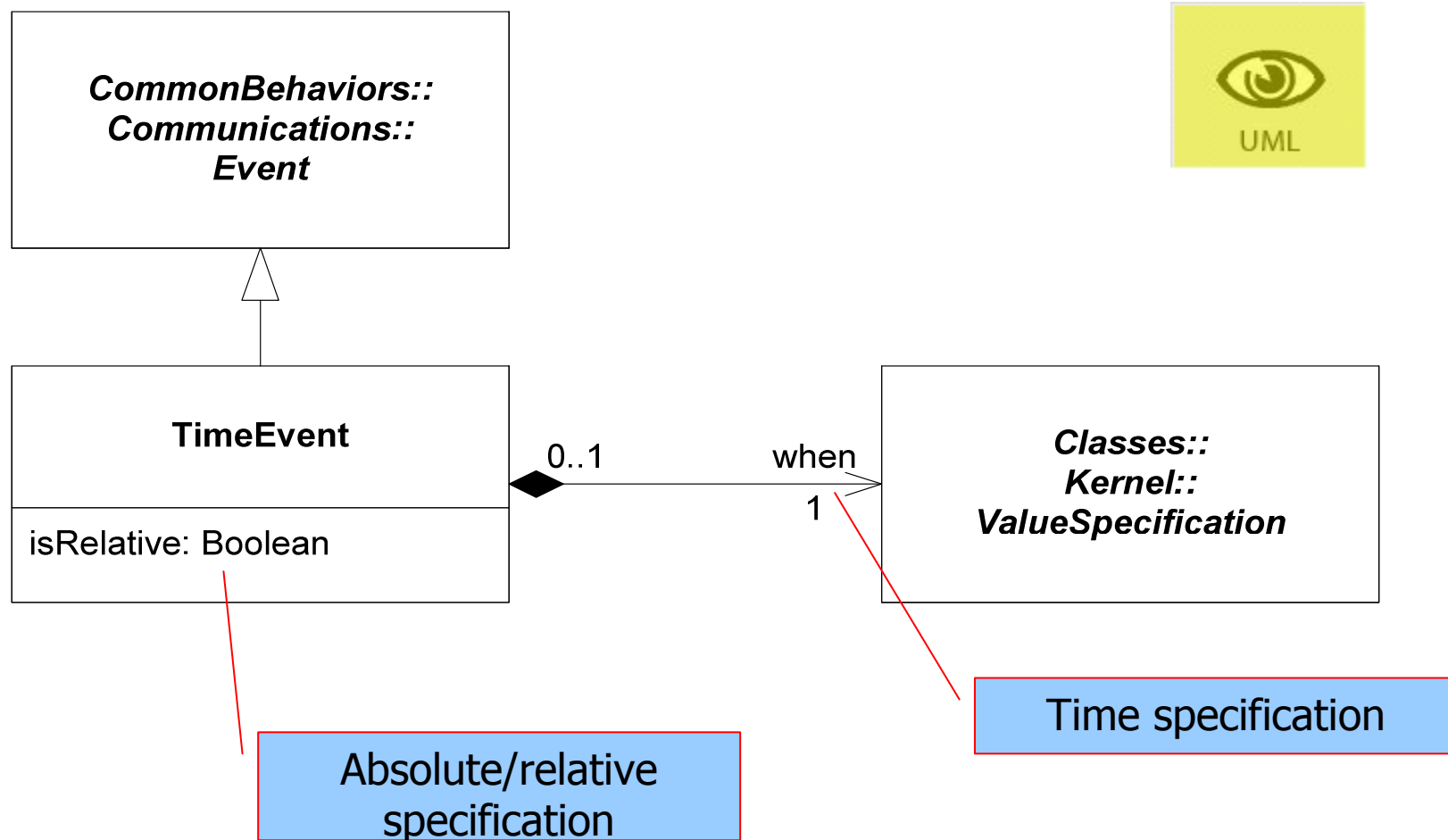
- Instant, duration
- Event bound to time, stimuli

- **Timing mechanisms & services**

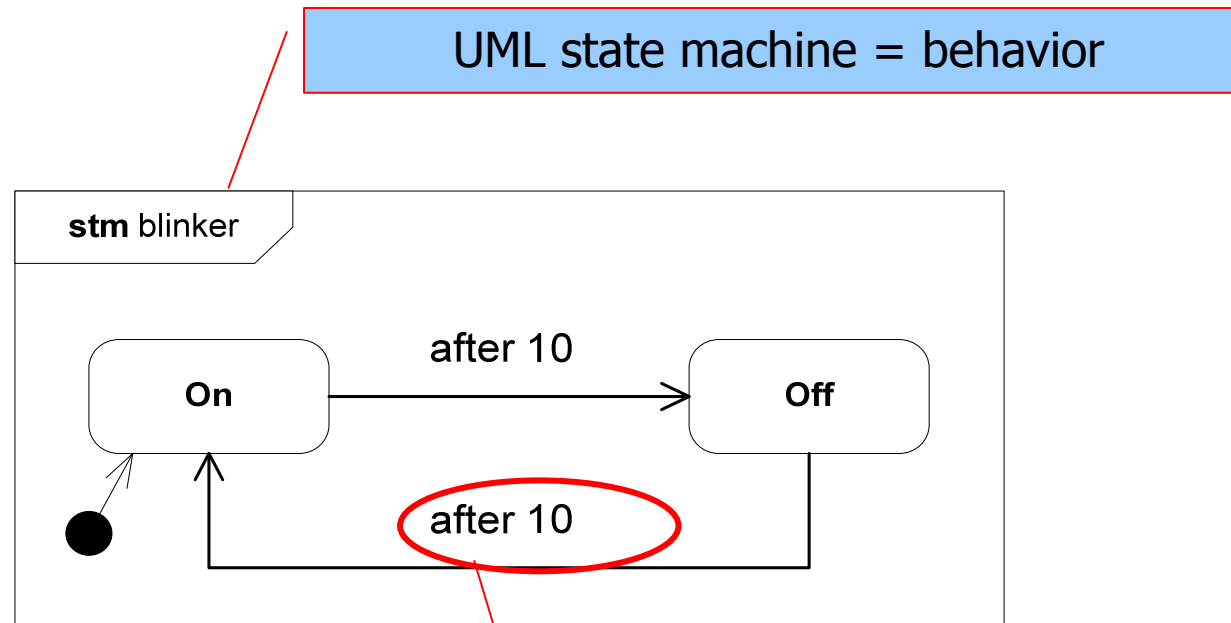
- UML2 adds new metaclasses to represent
 - Time (TimeEvent)
 - Duration
 - Observation: TimeObservation and DurationObservation
 - Some forms of time constraints
- Simple (even simplistic) model of time
- Advice: *Use a more sophisticated model of time provided by an appropriate profile, if needed.*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

e.g., MARTE

SimpleTime::TimeEvent



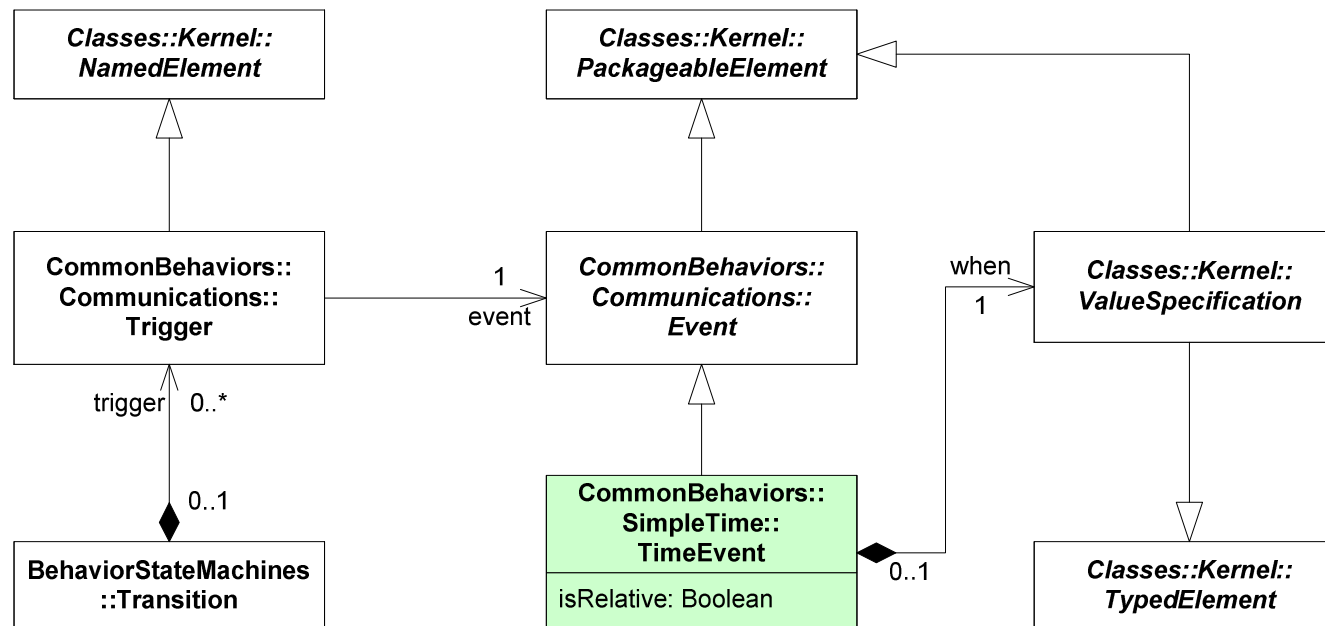
TimeEvent – usage (1)



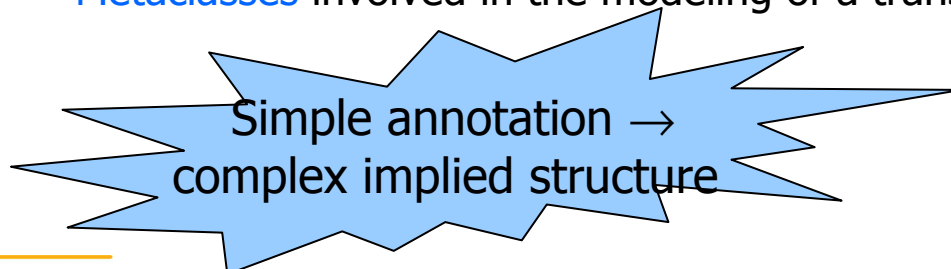
Specification of a time-trigger

Informal semantics

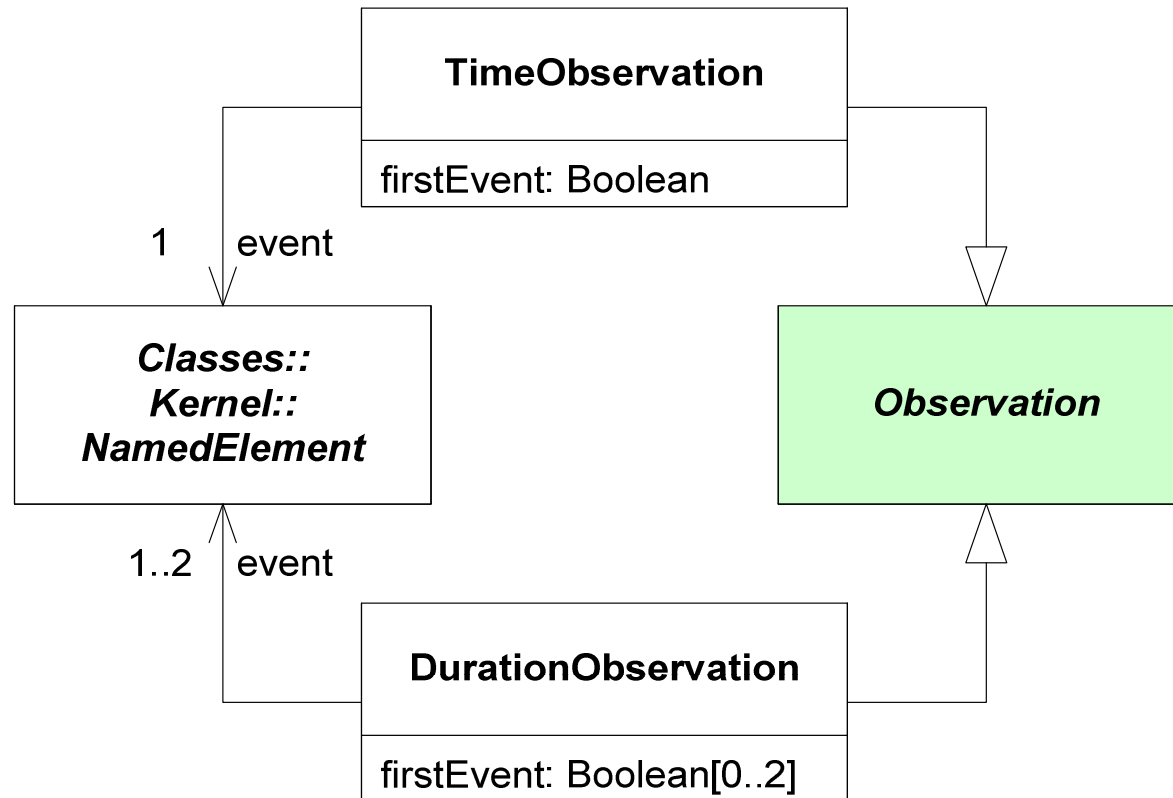
Meaning of “after 10”



Metaclasses involved in the modeling of a transition triggered by a TimeEvent

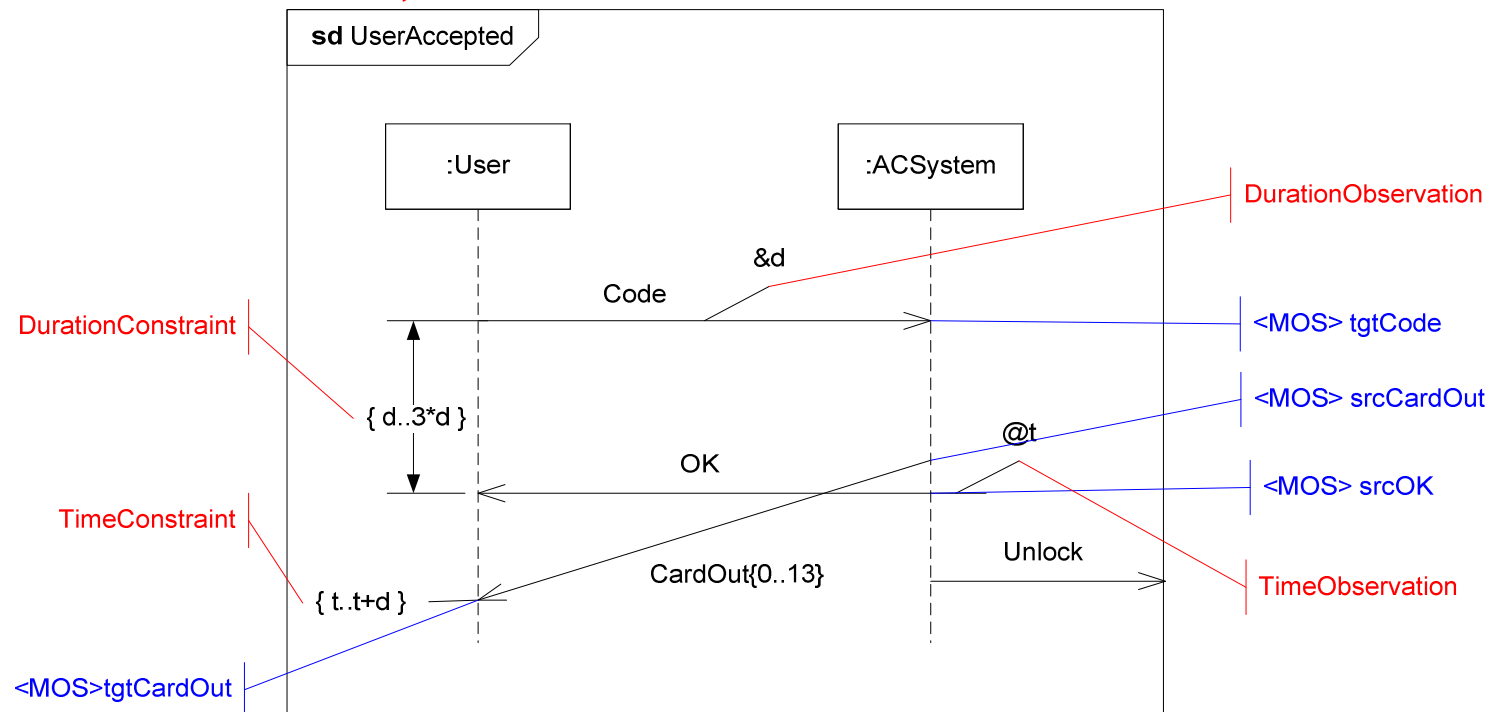


SimpleTime::Observation



Observation – usage (1)

Example of sequence diagram

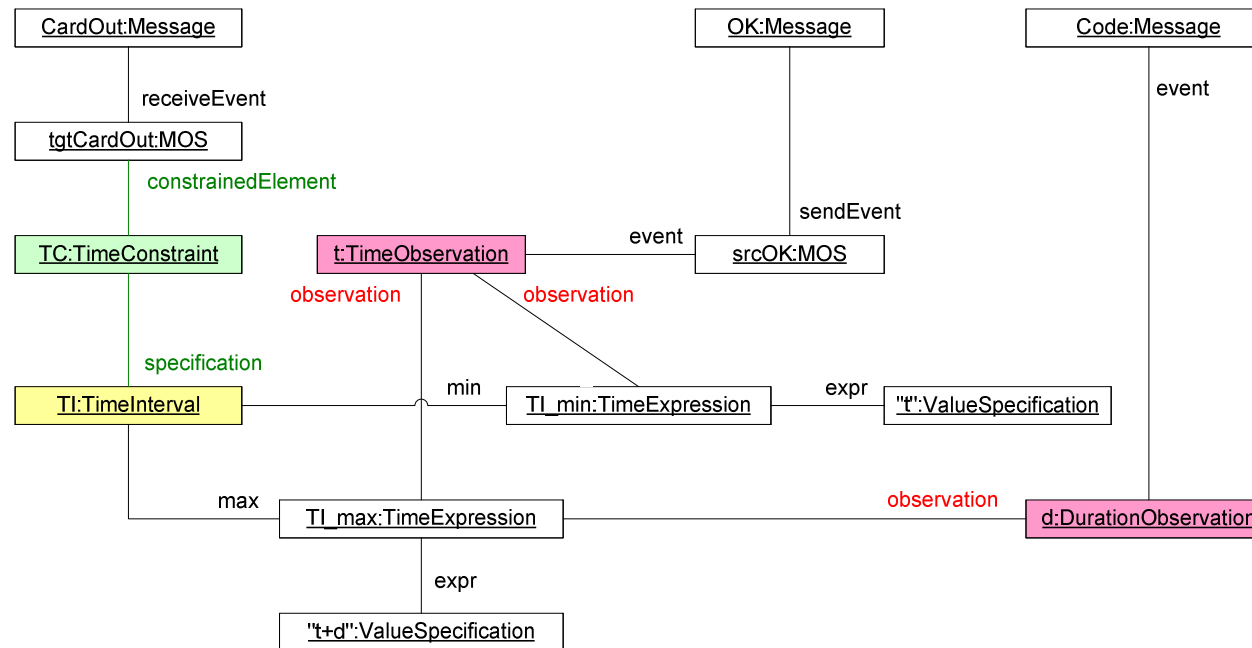


MOS stands for MessageOccurrenceSpecification

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

Observation – usage (2)

Instance model of the **time constraint**: receive CardOut in {t .. t+d}



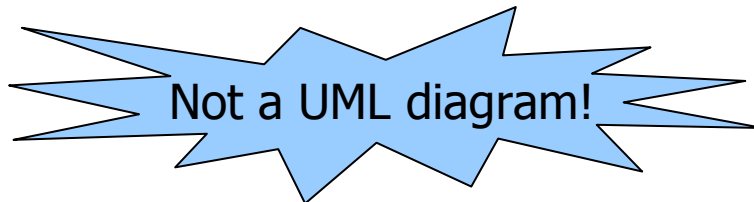
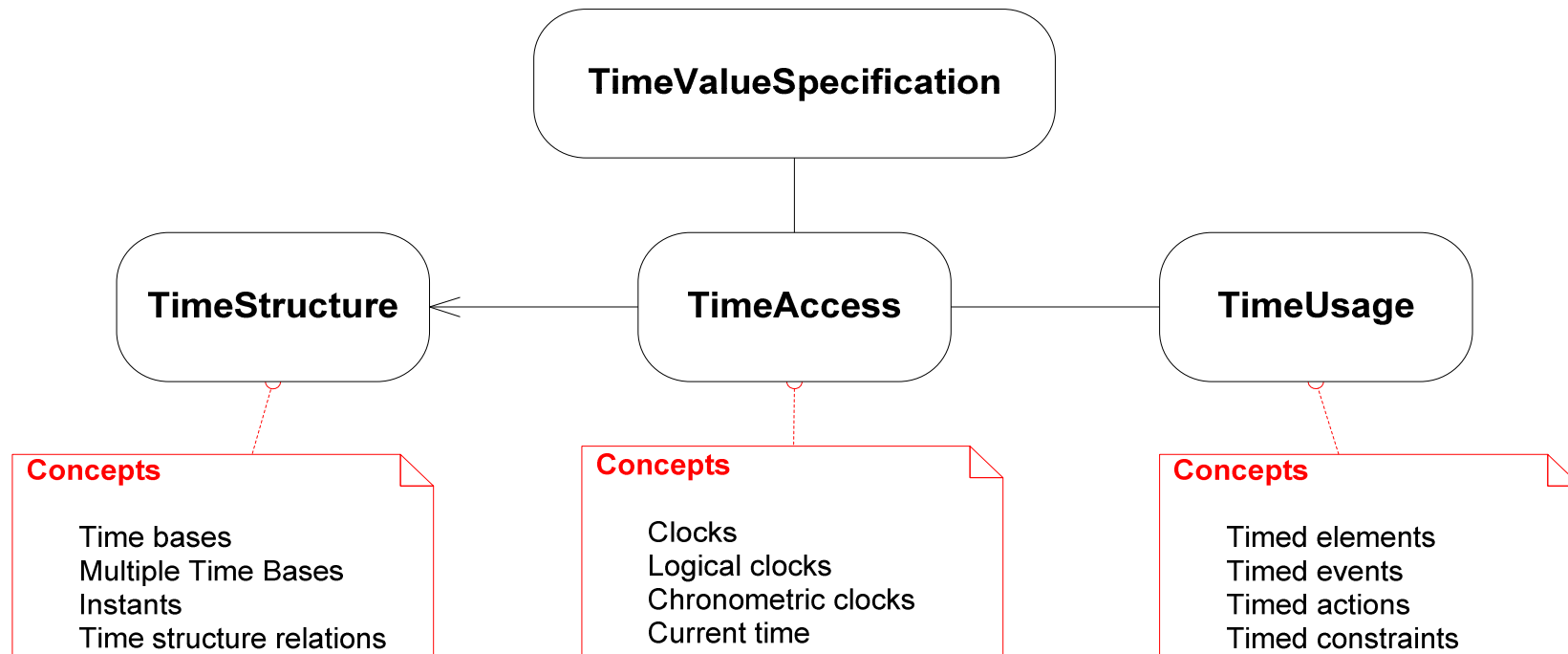
Simple annotation →
complex implied structure

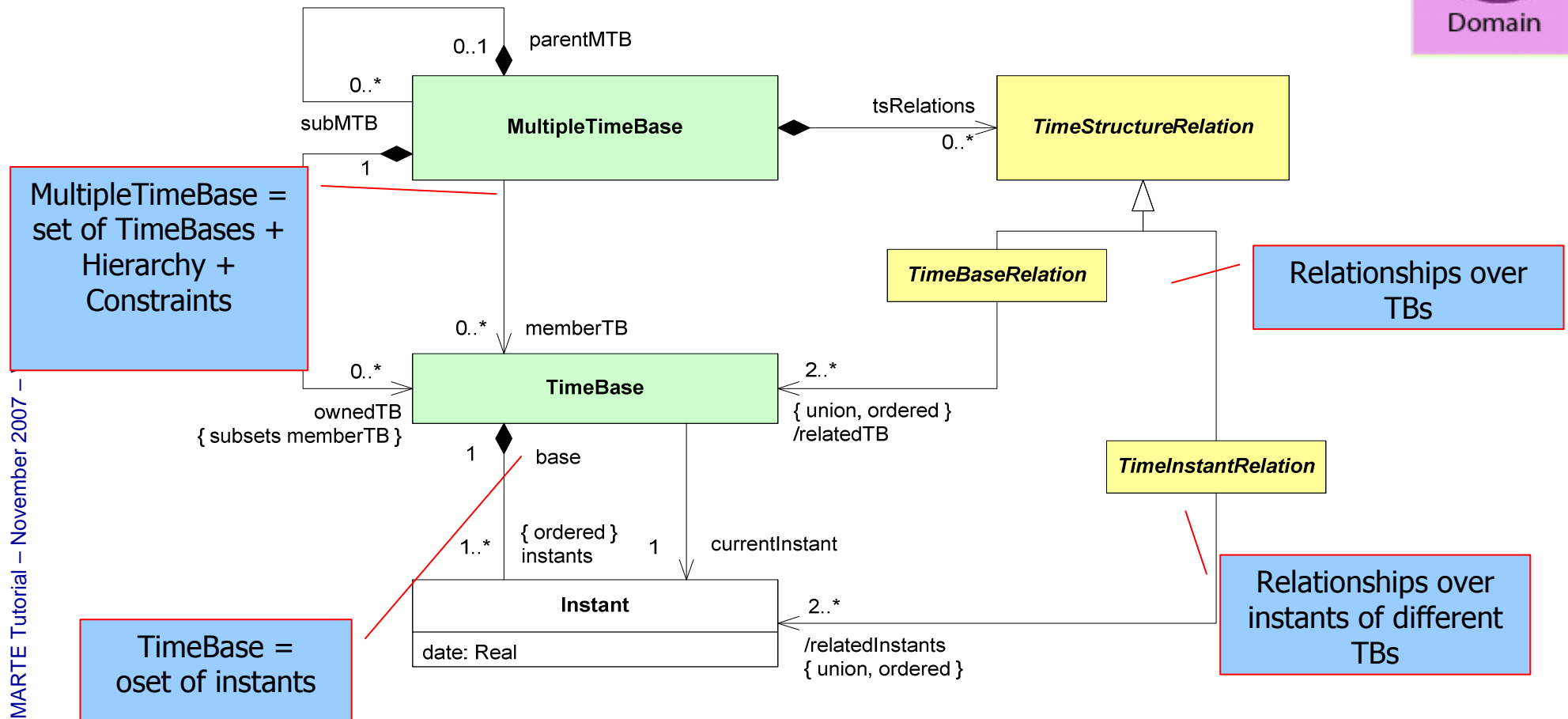
- SPT, UML 2 and Time
 - UML::CommonBehaviors::SimpleTime
- **the MARTE Time domain view**
 - a.k.a. the MARTE Time meta-model
 - Concepts and relationships
- the MARTE Time sub-profile
 - a.k.a. UML view
- Usage of the Time sub-profile



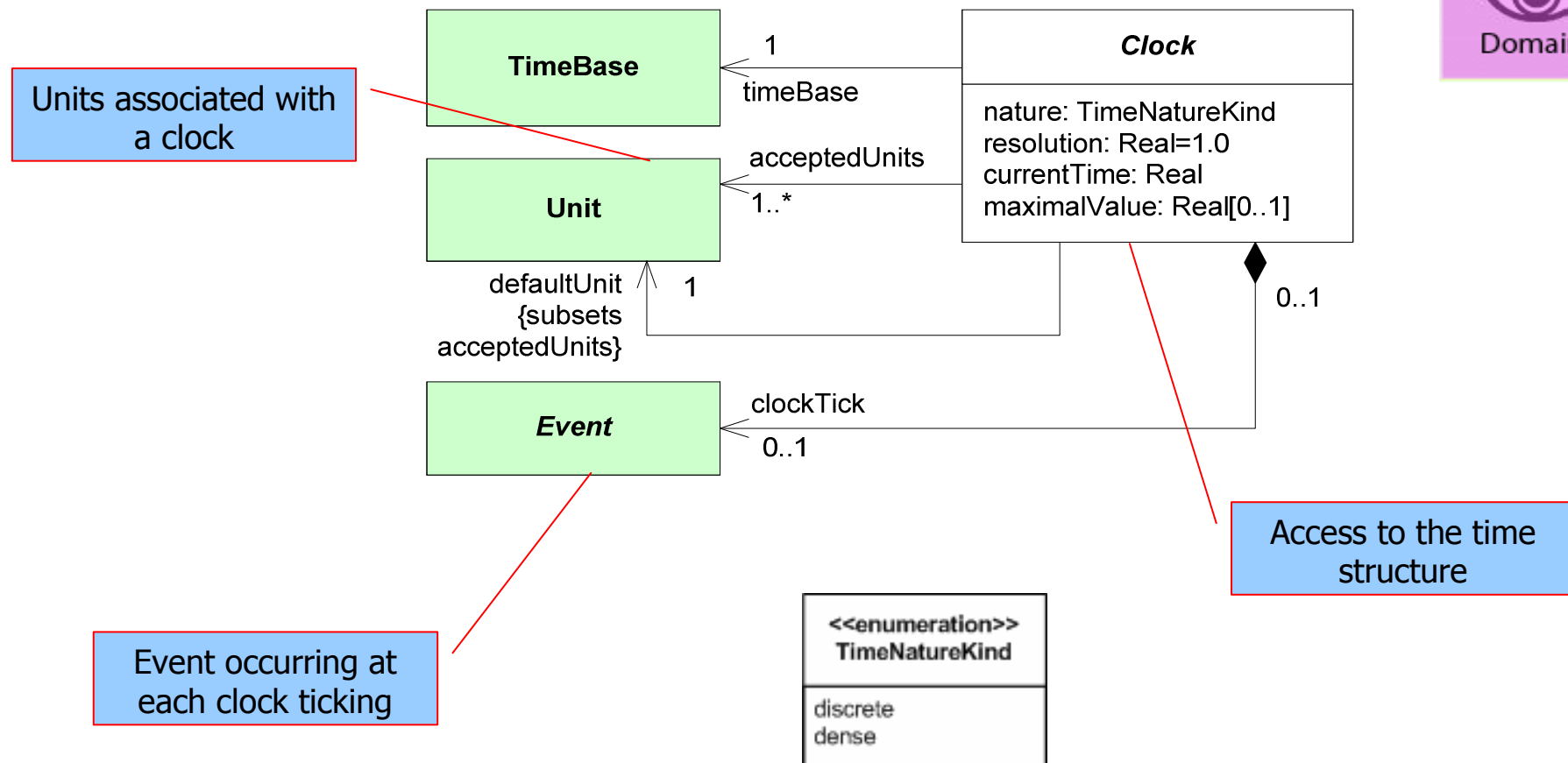
- **Time structure =**
set of time bases + time structure relations
→ Partially ordered set of instants
- **Access to time = Clock**
- **Principle:** associate Clocks with model elements
 - Behavioral elements → TimedEvent, TimedProcessing
 - Constraints → TimedConstraint
 - Data types and values → TimedValue

Main concepts introduced in Time modeling



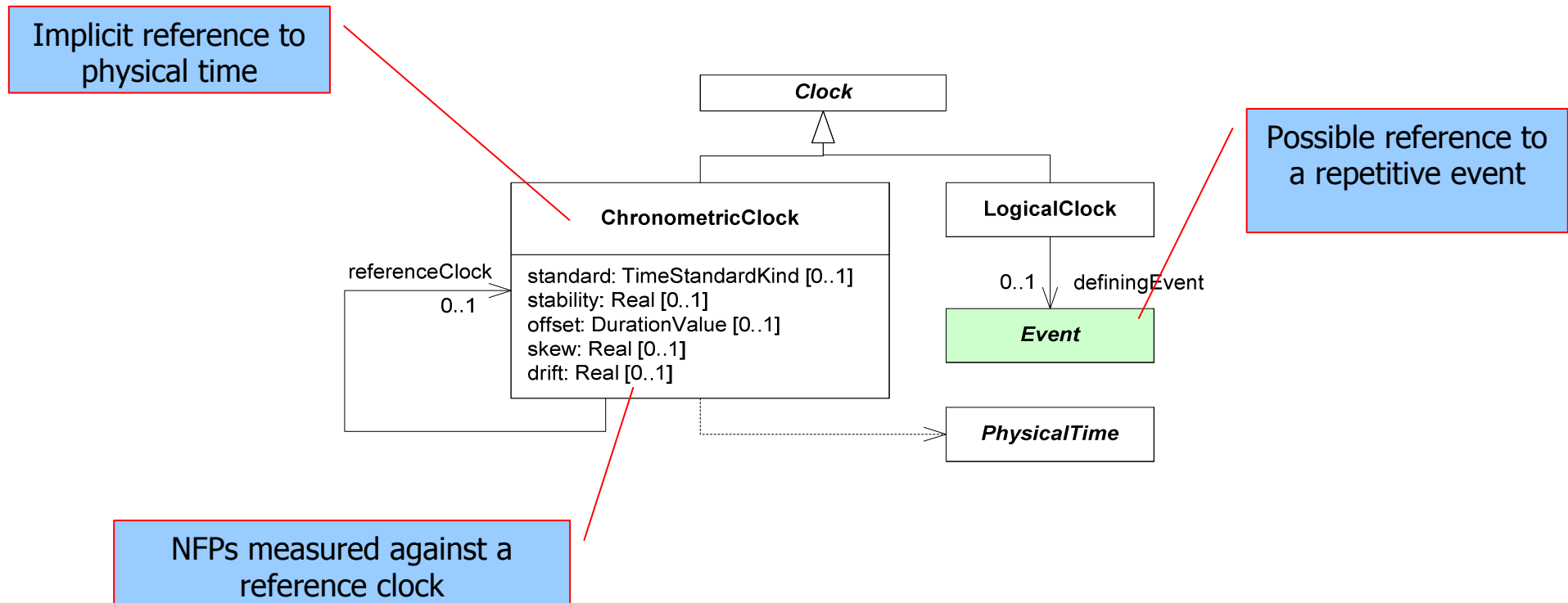


Access to Time: Clock



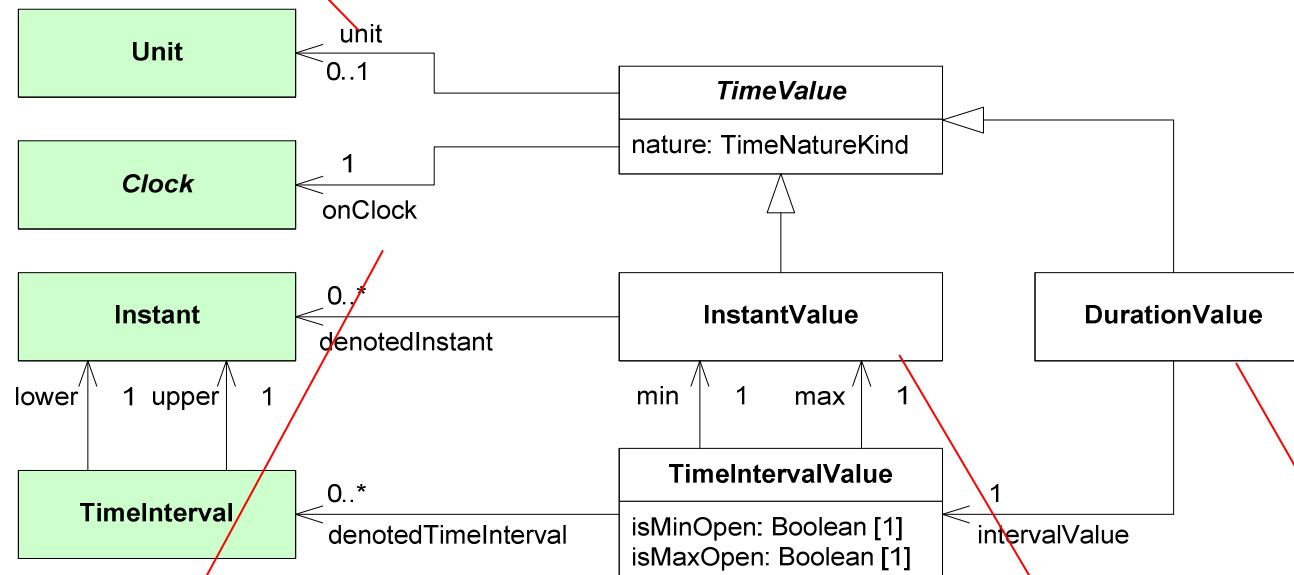
Chronometric/Logical Clocks

Two kinds of clocks





A TimeValue has a unit
(default= clock unit)

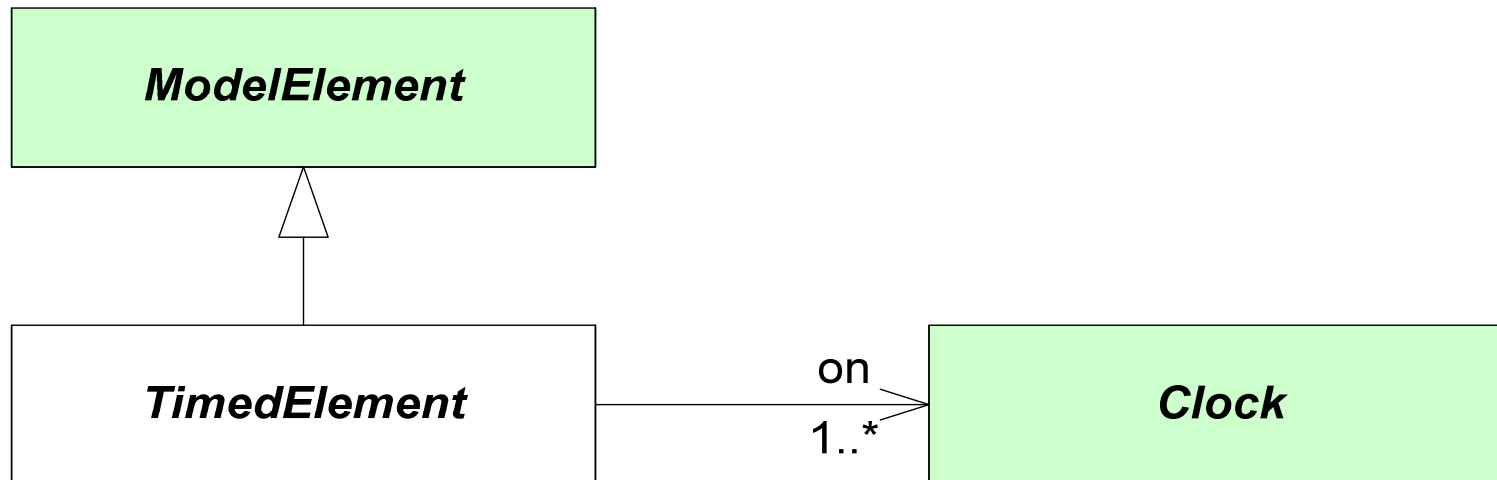


A TimeValue must
reference a clock

Instant/Duration two
distinct concepts

Timed Entities: TimedElement

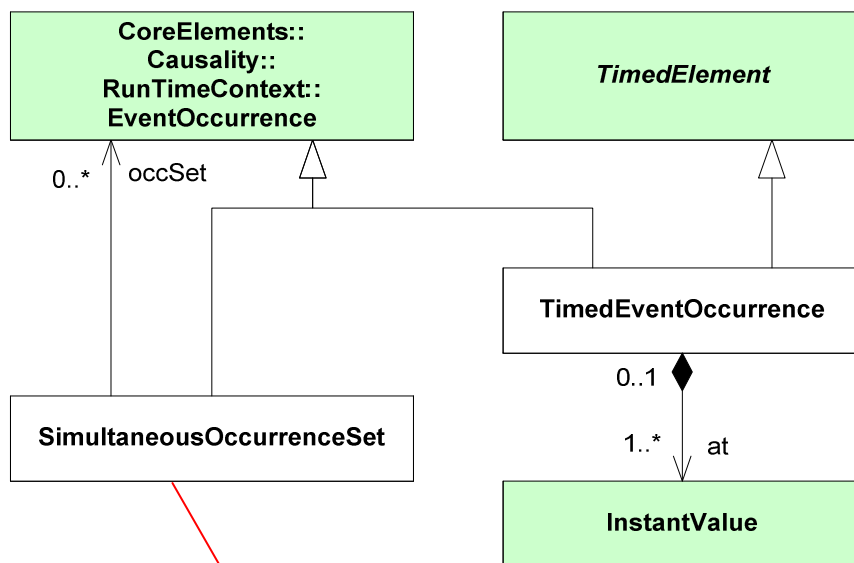
The unifying concept: a **TimedElement** = a **ModelElement** + a **Clock**



Timed Entities: TimedEvent

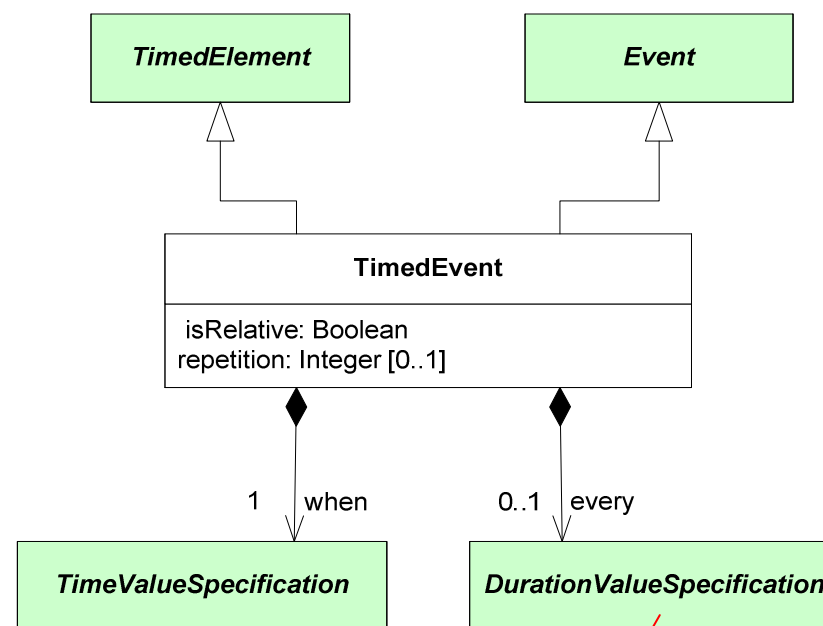


occurrences



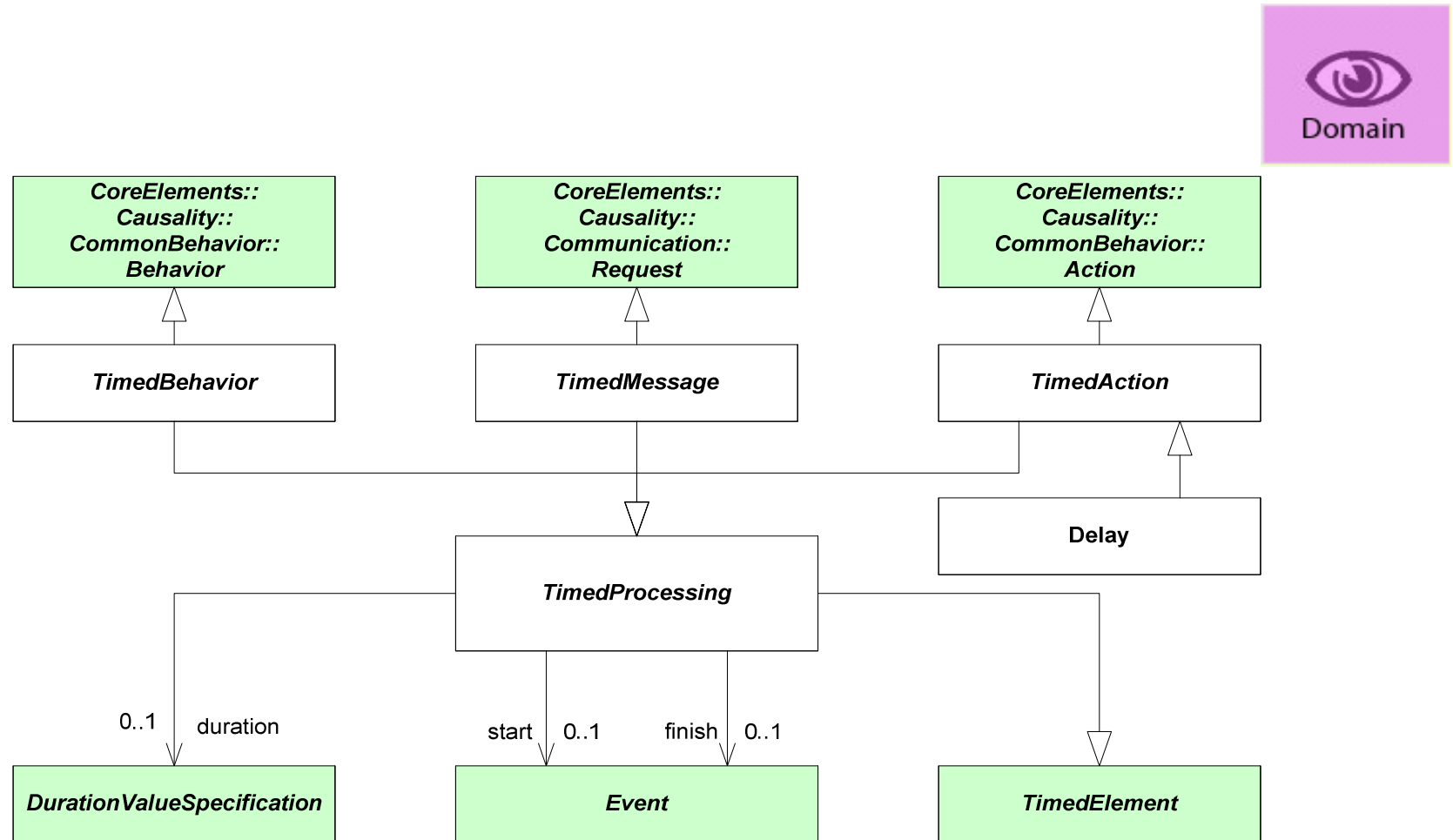
Provision for
simultaneity

events

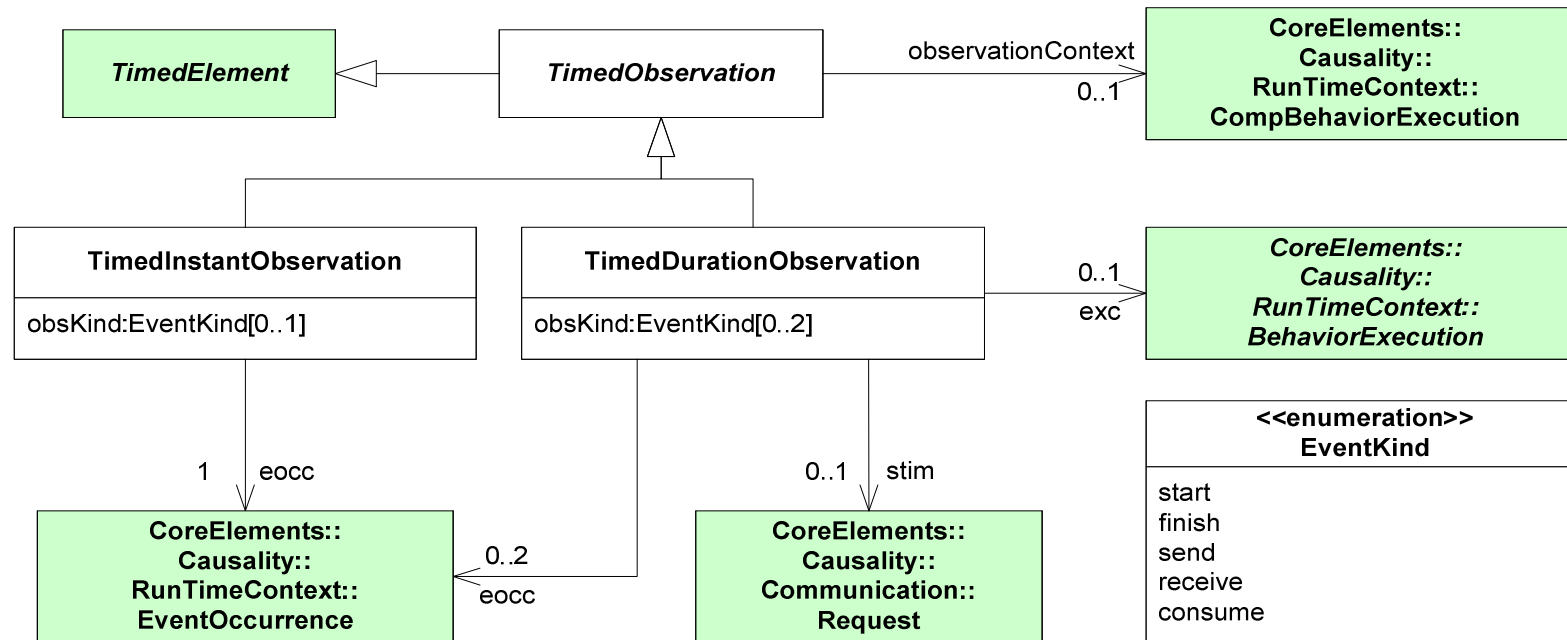


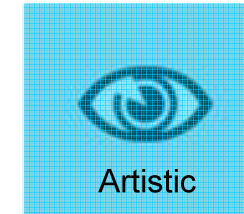
Facility to specify
multiple occurrences

Timed Entities: TimedProcessing



Timed Entities: TimedObservation

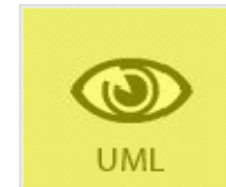




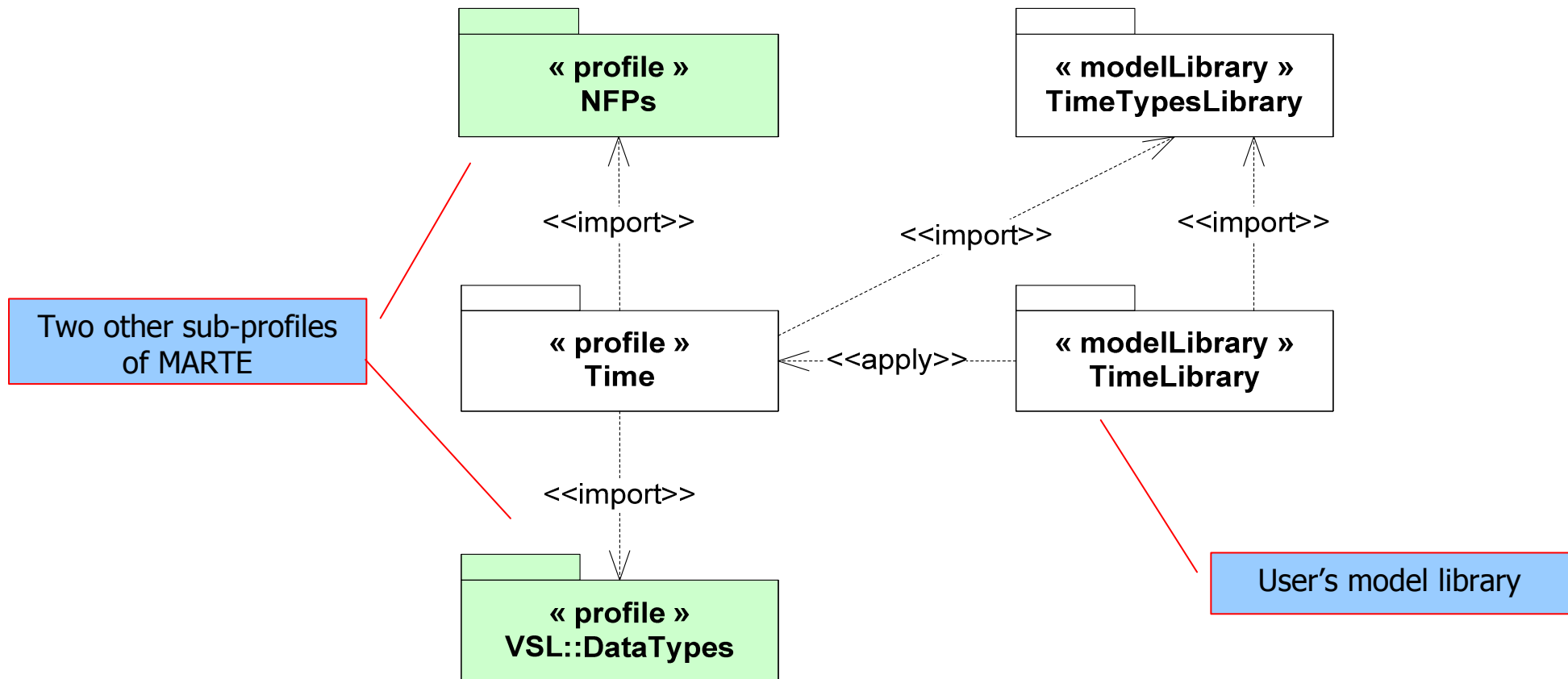
See:

http://en.wikipedia.org/wiki/Image:The_Persistence_of_Memory.jpg

- **SPT, UML 2 and Time**
 - UML::CommonBehaviors::SimpleTime
- **the MARTE Time domain view**
 - a.k.a. the MARTE Time meta-model
 - Concepts and relationships
- **the MARTE Time sub-profile**
 - a.k.a. UML view
- **Usage of the Time sub-profile**



- **Through a UML profile**
 - New Stereotypes
- **Facilities**
 - Model libraries
 - Dedicated languages (especially for expressions)
 - Clocked Value Specification Language (CVSL)
 - Clock Constraint Specification Language (CCSL)



Chronometric clock → "physical " time; units $\in \{s, ms, us, \dots\}$

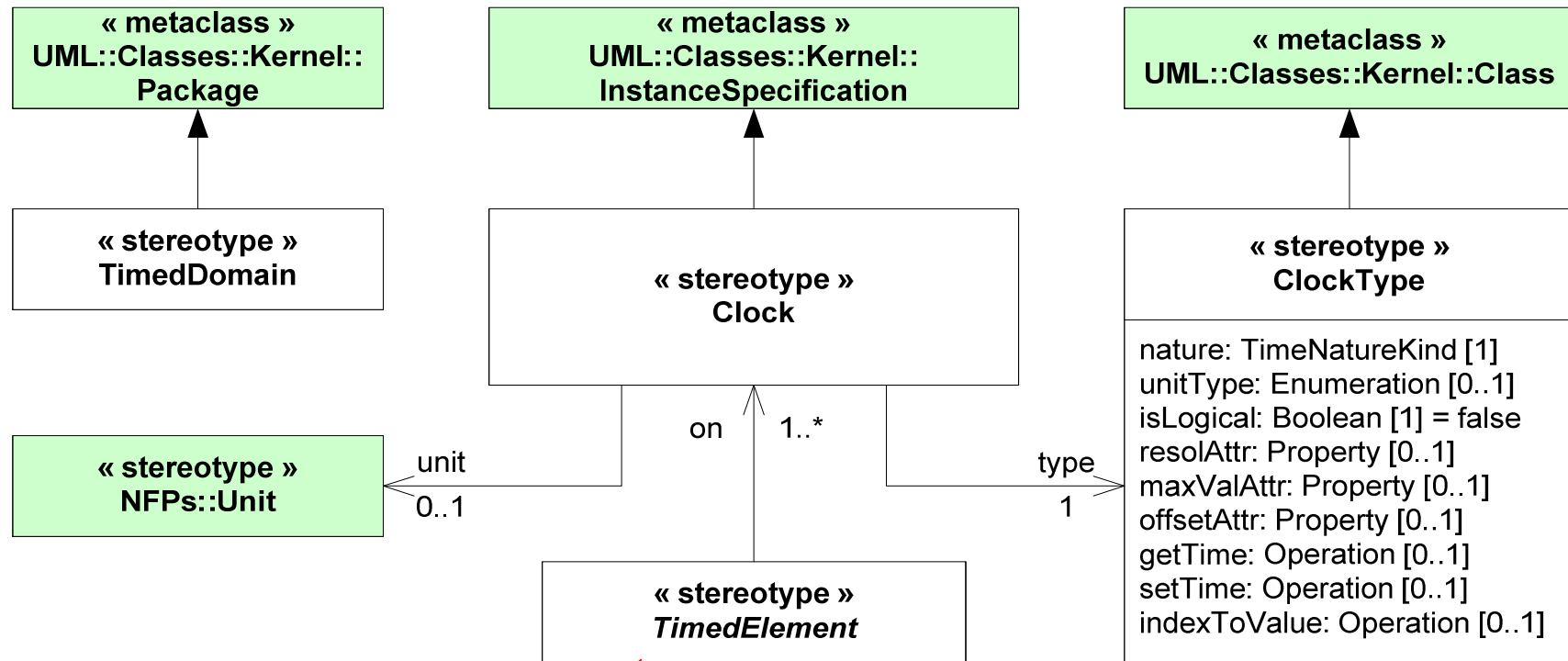
Logical clock → any repetitive event; units $\in \{tick\} \cup \text{PhysicalUnits}$

- **Accepted units**
- **Default unit**

Stereotype properties:
Special semantics

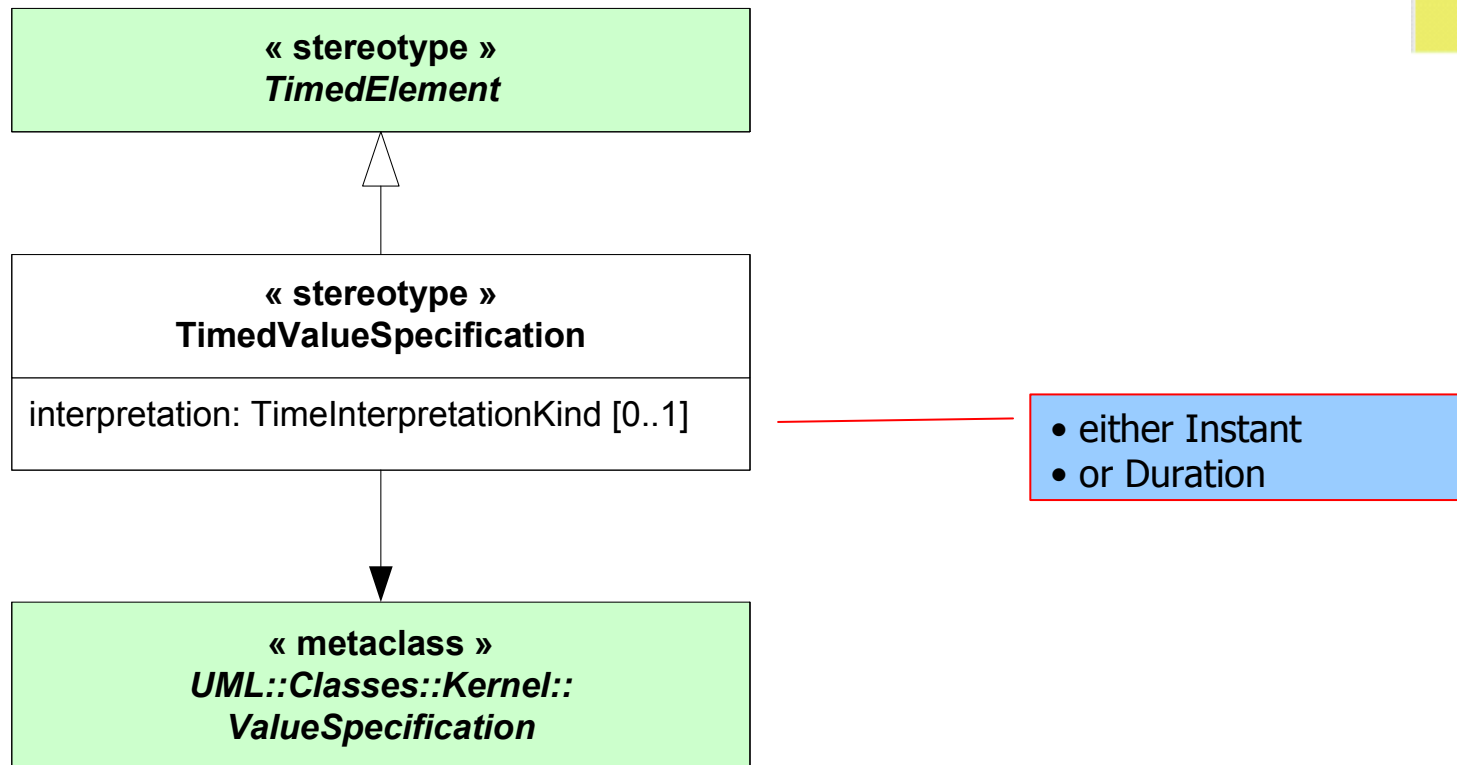
- + **optional**
- **set of properties**
- **set of operations**

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



Notice that this abstract stereotype has no base metaclass

TimedValueSpecification



Examples of Clocked value expressions

Simple time values

(value=3.5, unit=ms, onClock='idealClk');
3.5 ms on idealClk;

tuple, *a la* VSL

short form

Homogeneous expressions

(value=1.5, unit=ms, onClock='idealClk') +
(value=150, unit=us, onClock='idealClk');
→ (value=1650, unit=us, onClock='idealClk')

Can be evaluated,
because convFactor
between units

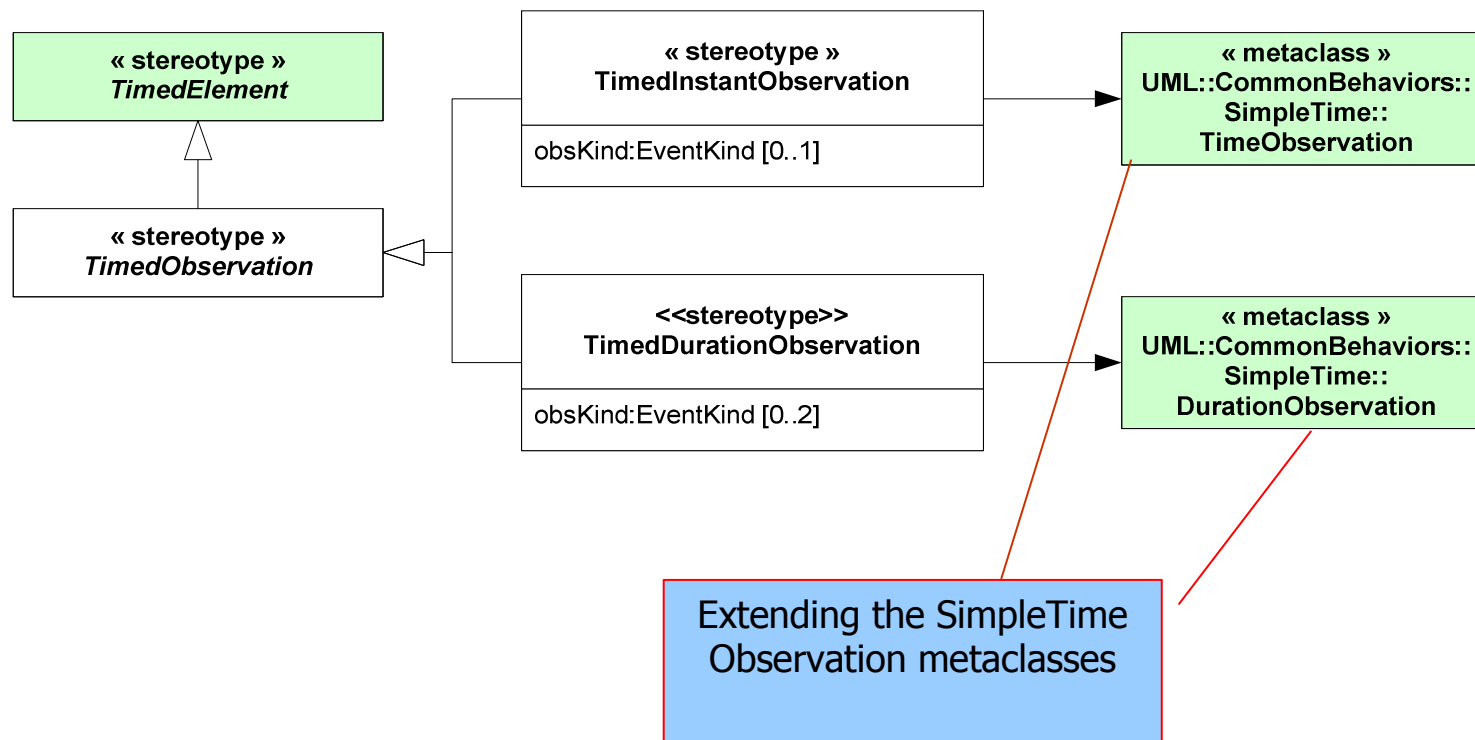
Heterogeneous expressions

min (15 tick on prClk, 5 ms on idealClk);

Clock relation between
prClk and idealClk must
be provided

Additional capabilities with VSL

- Occurrence number, jitter,...
- but implicitly on idealClk



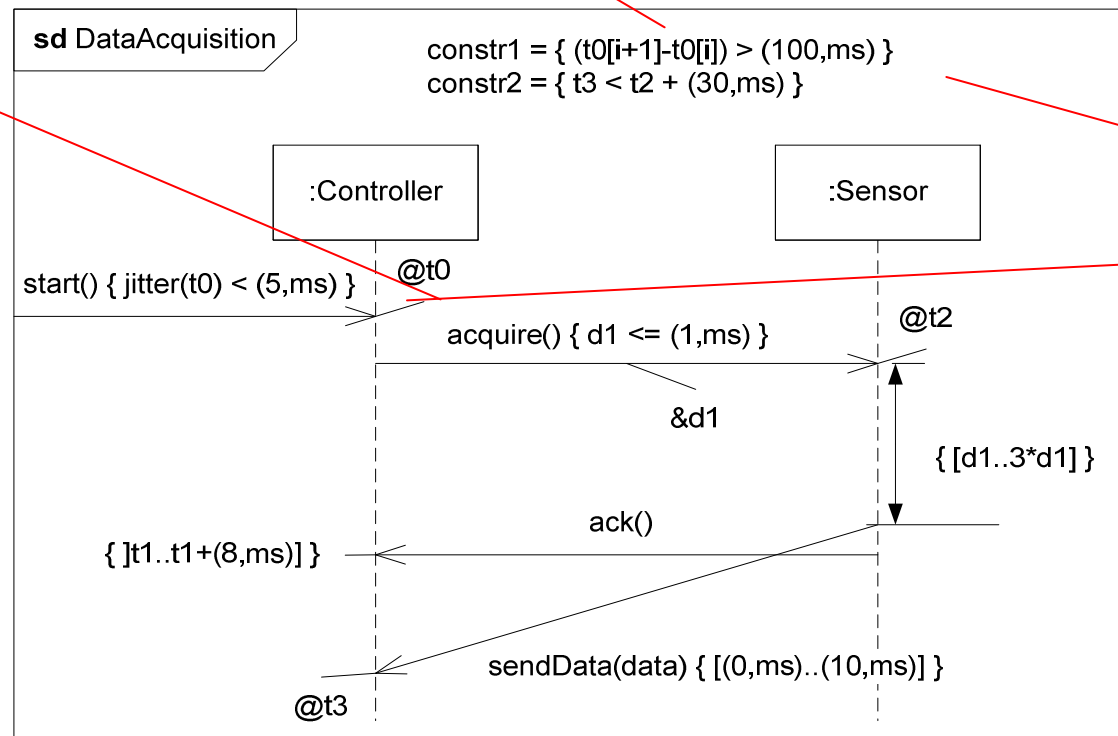
Time specific languages: VSL Time Constraints

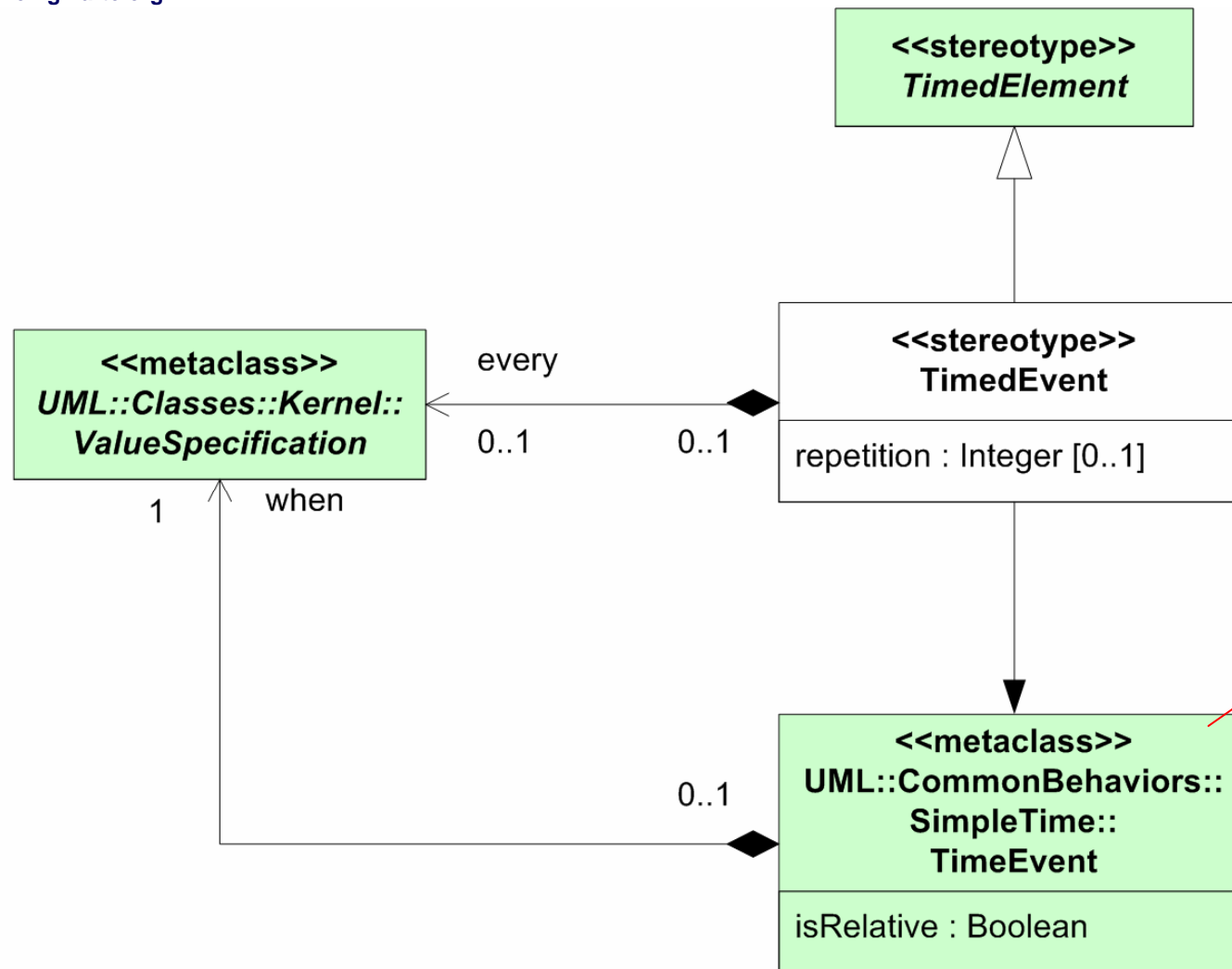
$t0[i]$ denotes the
i-th occurrence
of



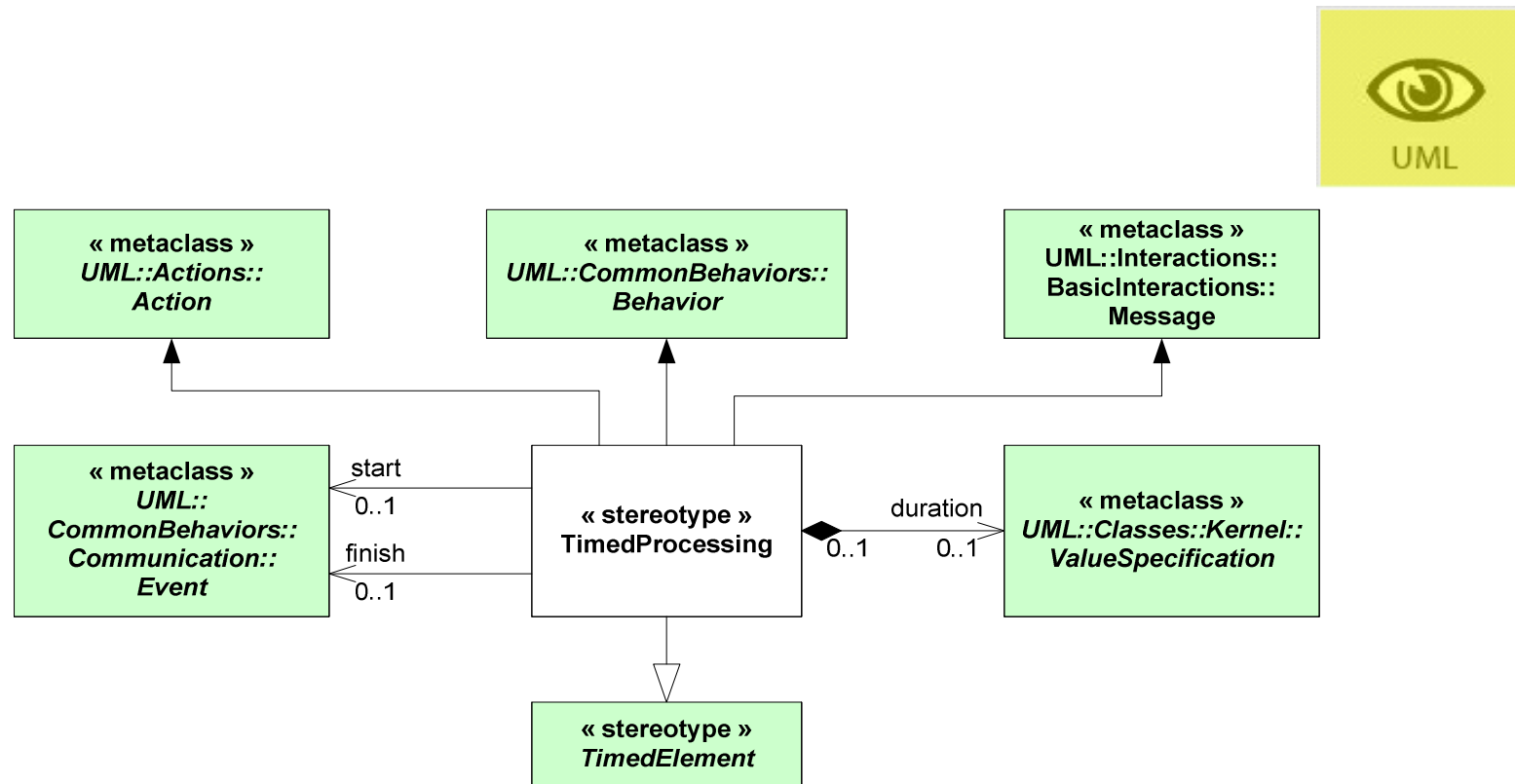
$t0$: observation
of the message:
start

$t0$ is periodic,
period 100ms
with a jitter less
than 5ms

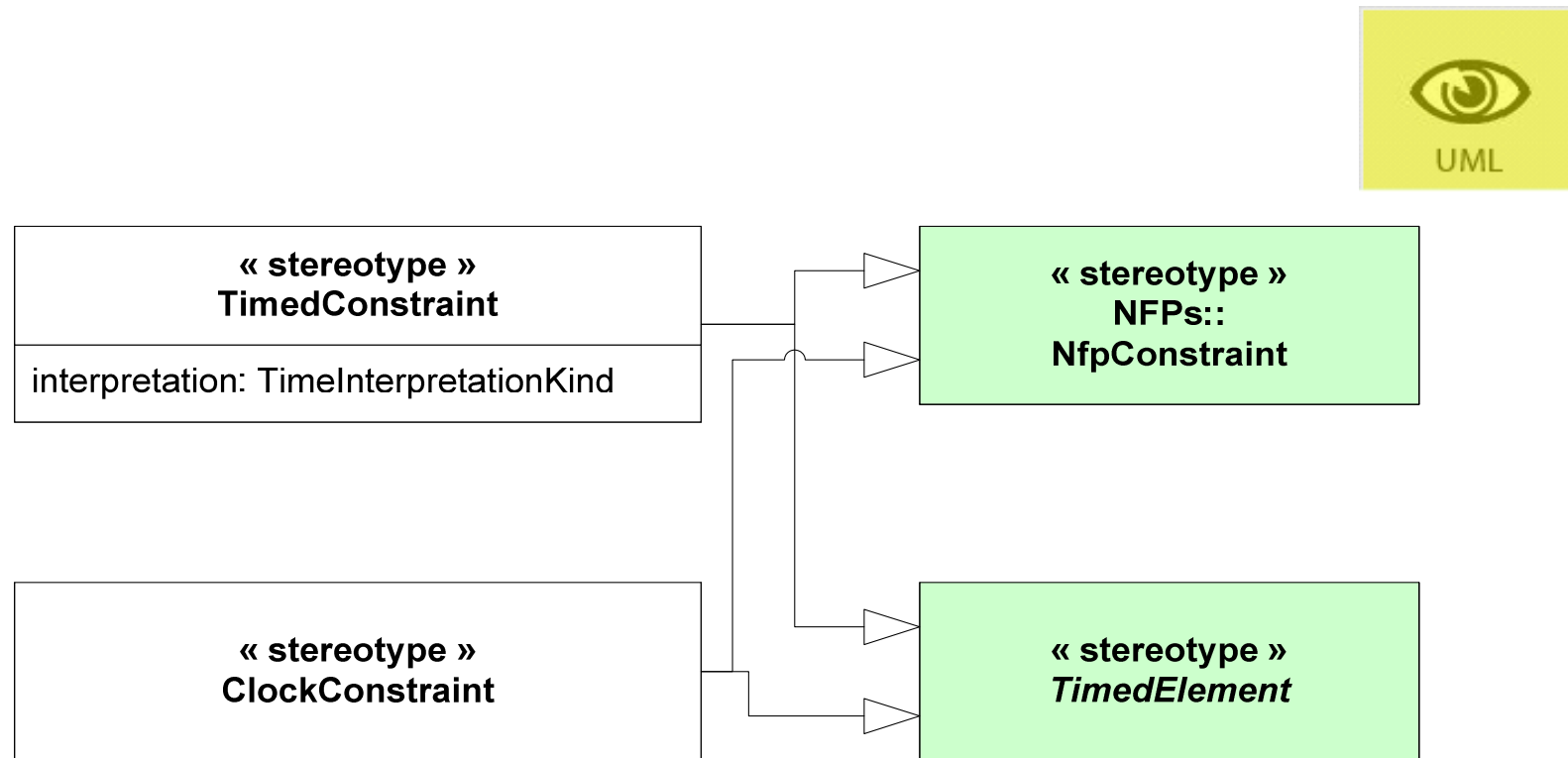




Extending the
TimeEvent metaclass of
SimpleTime

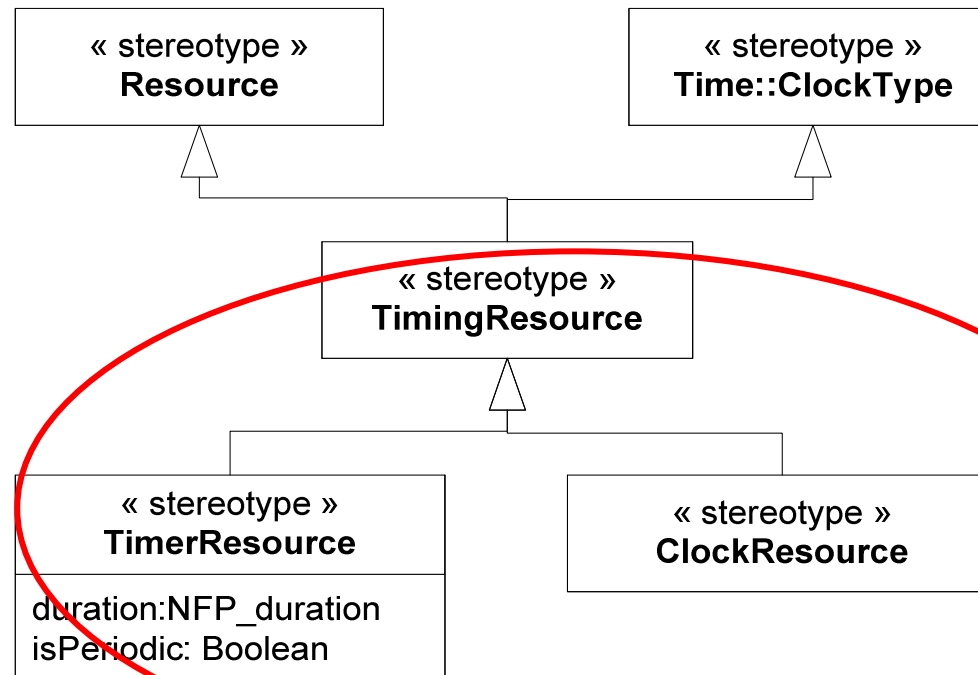


TimedConstraint & ClockConstraint



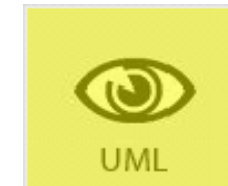
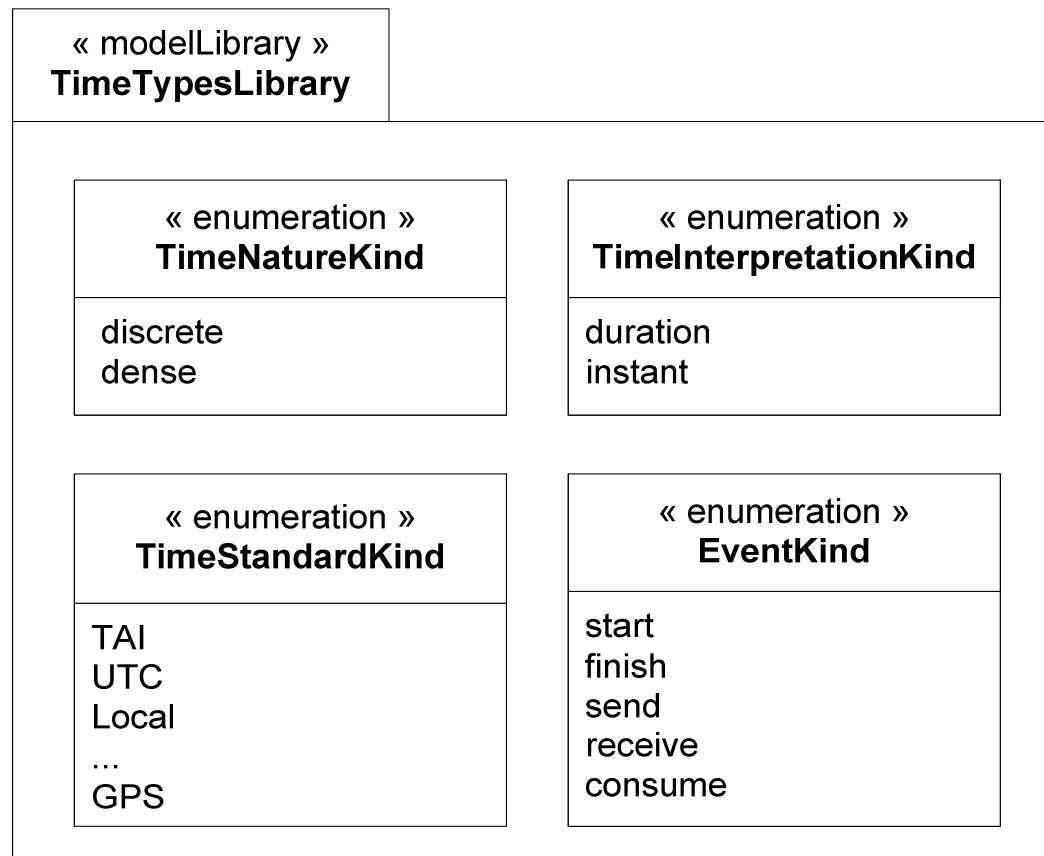
Time-related GRM stereotypes

Stereotypes defined in the [Generic Resource Modeling](#) sub-profile



Resources for time management

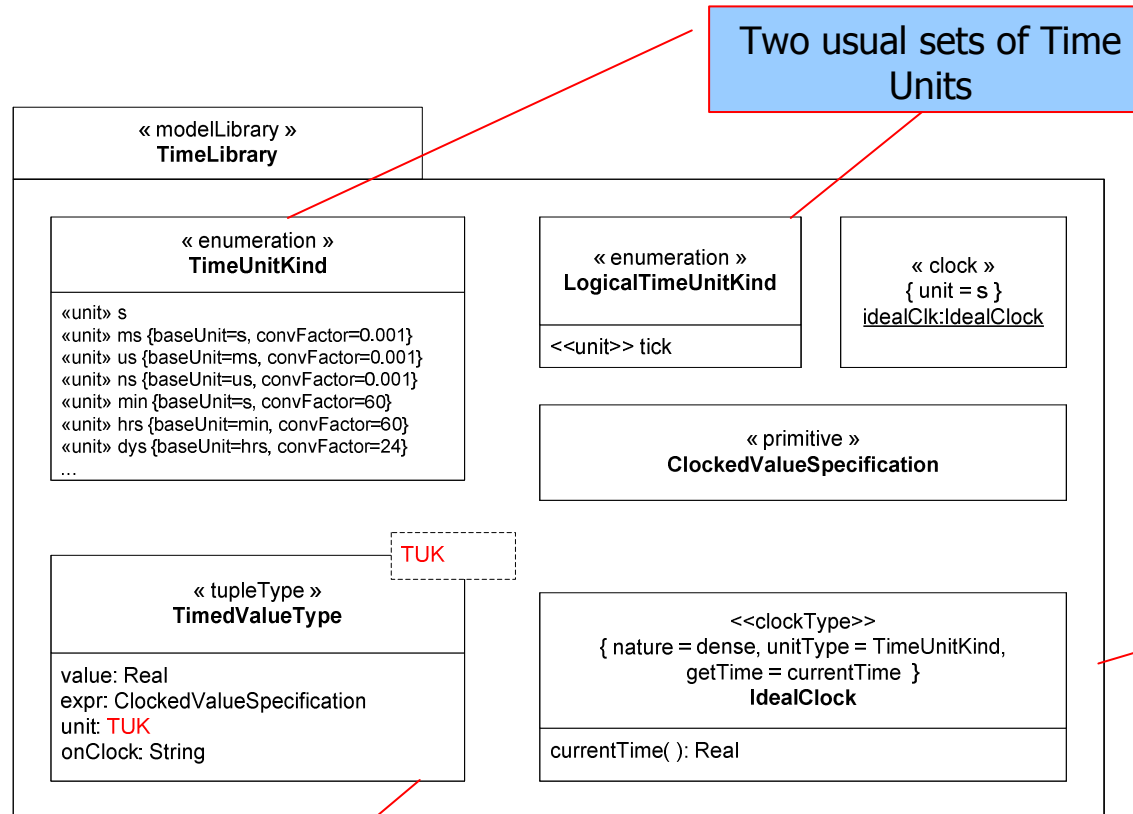
Time-related libraries: TimeTypesLibrary

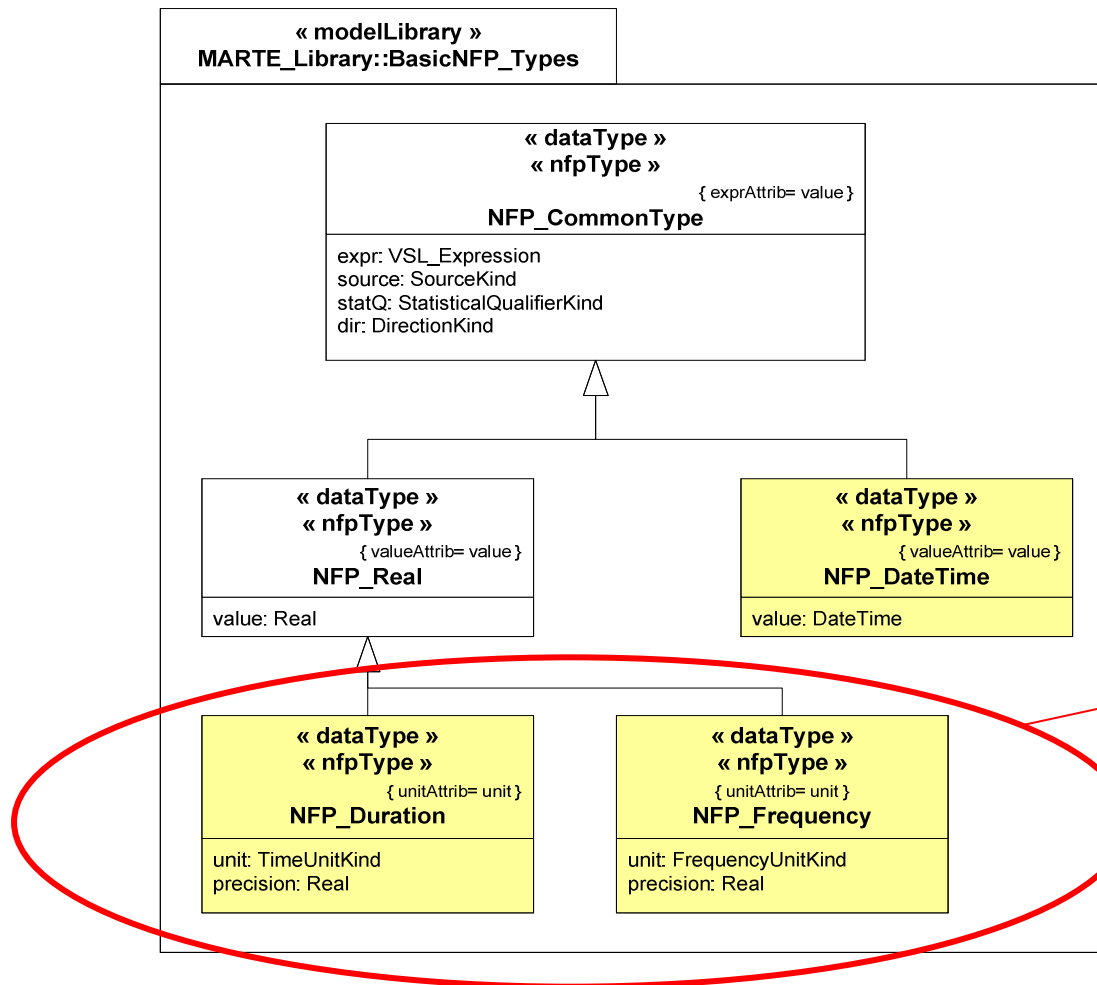


&



Time-related libraries: TimeLibrary

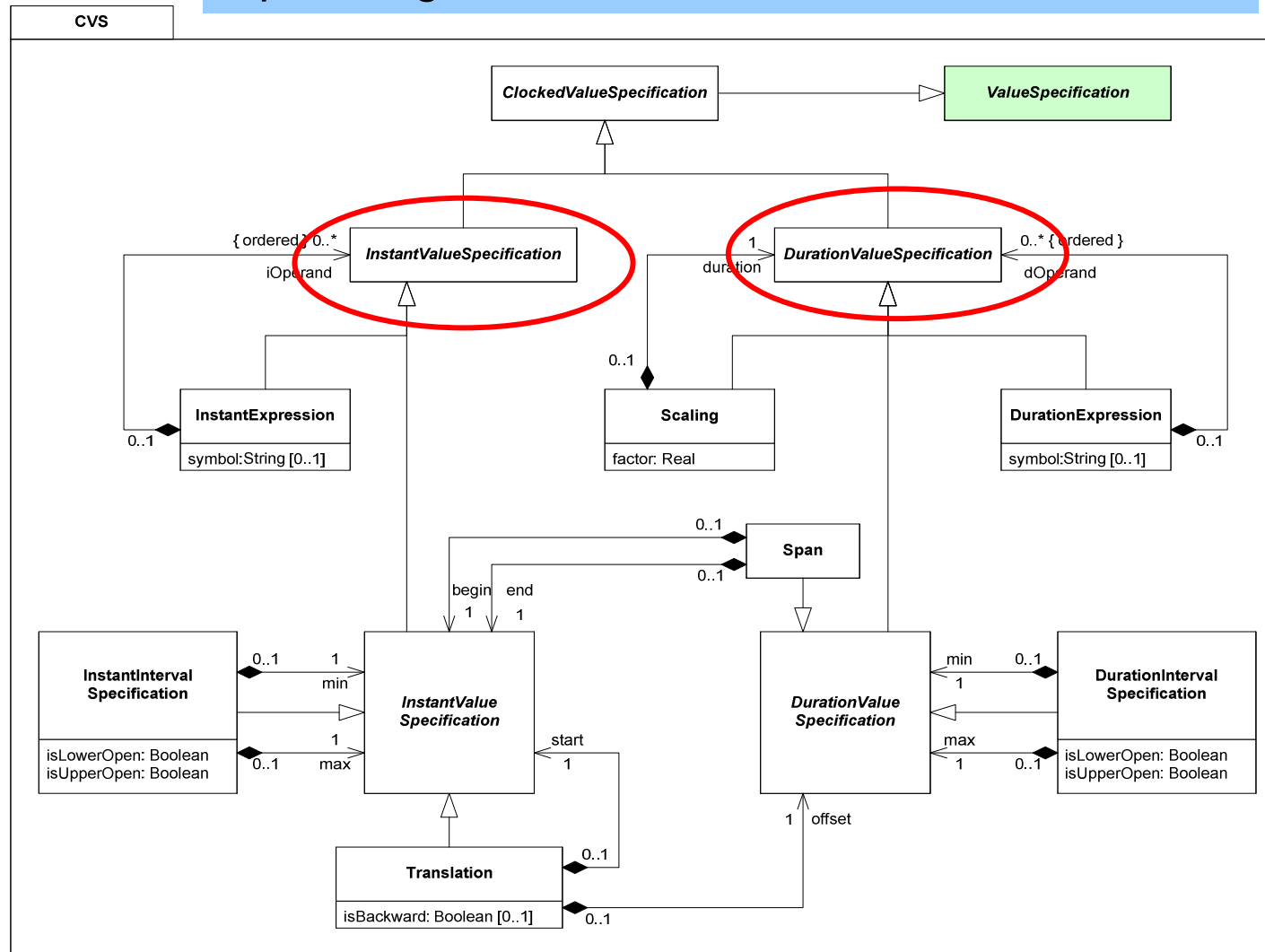




Time-related types.
Often used.

Time specific languages: Clocked Value Specification

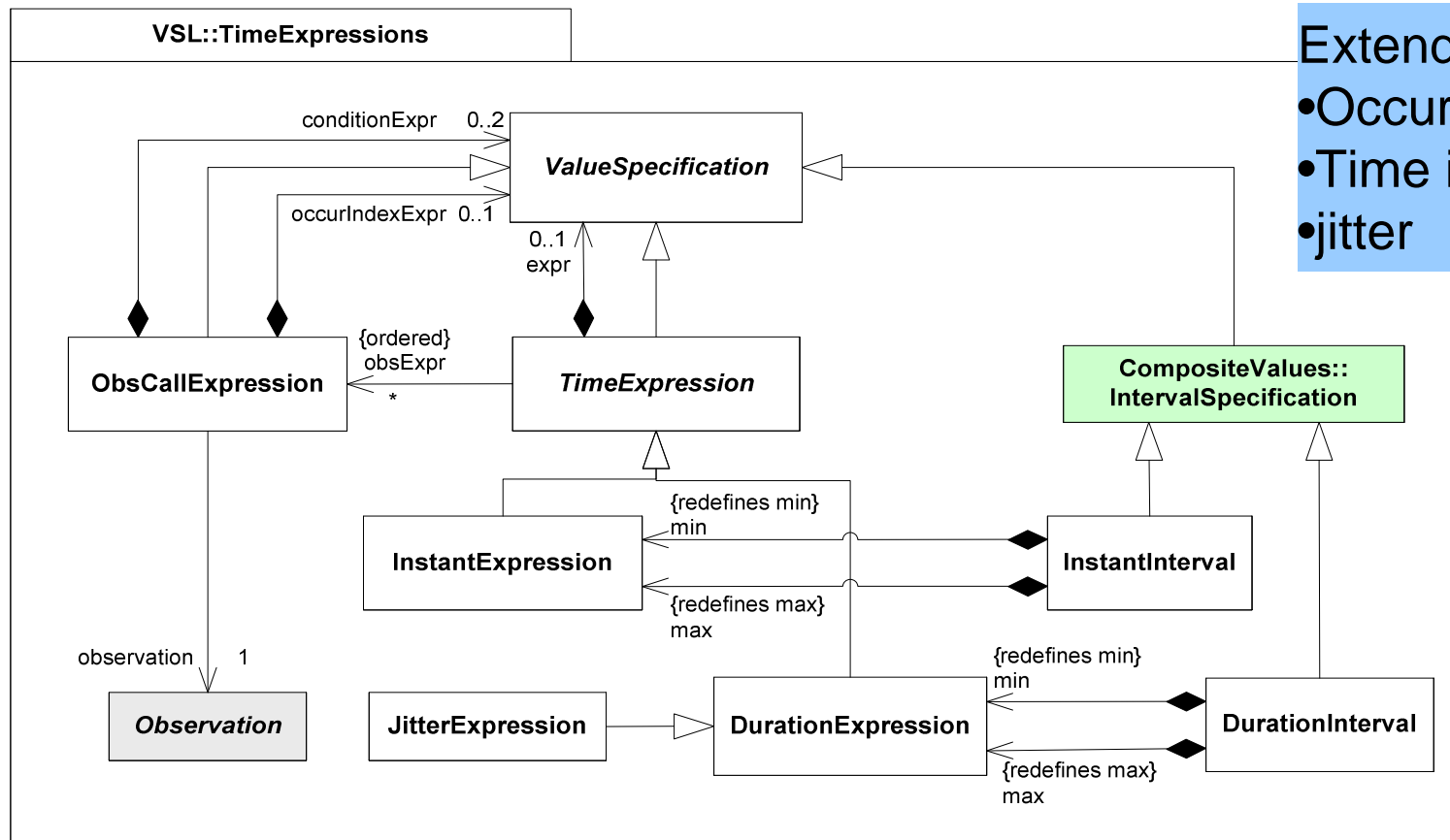
Expressing time values with EXPLICIT clocks



Instant
≠
Duration

Time specific languages:VSL Time Expressions

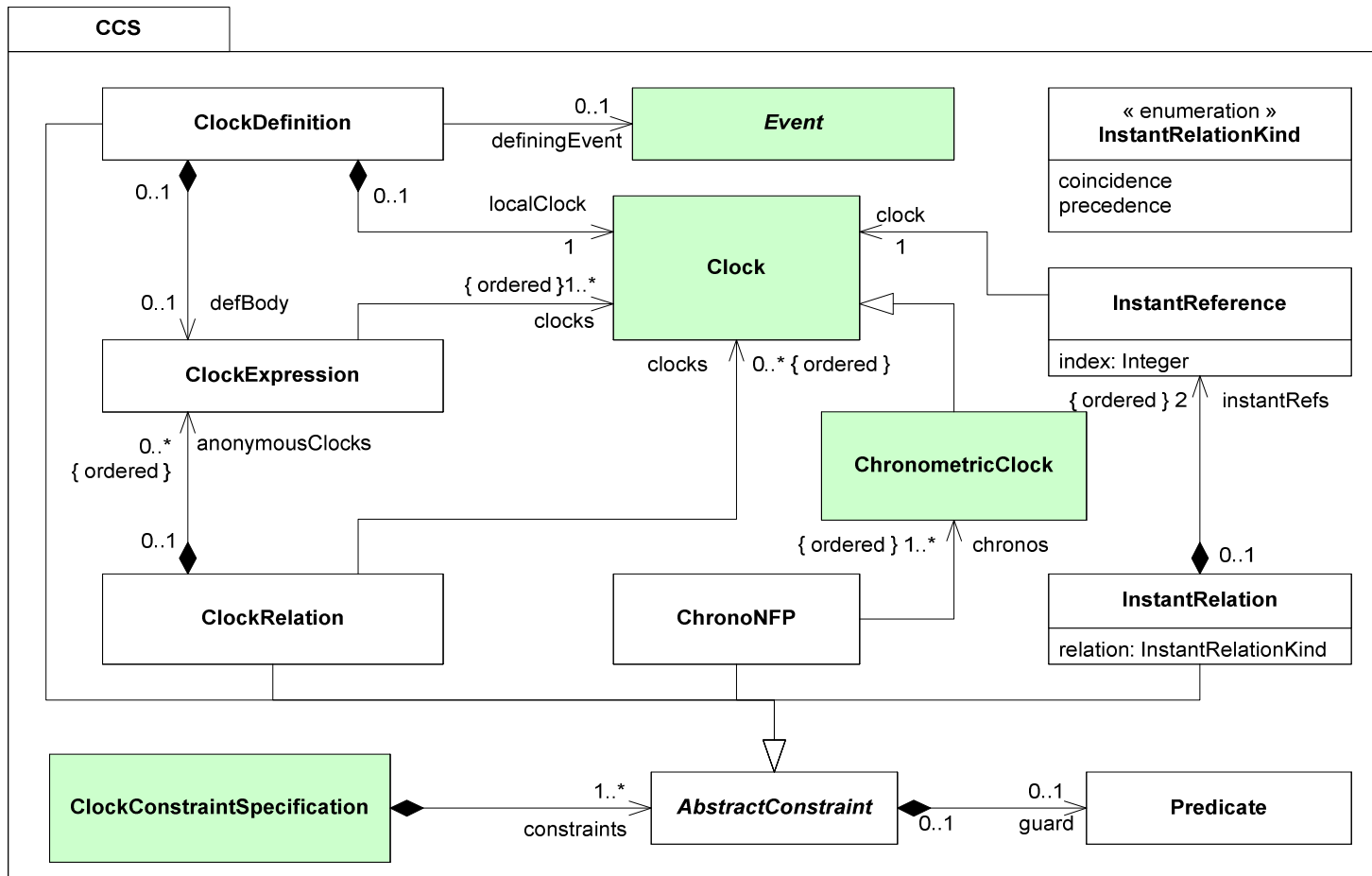
Expressing time values with EXPLICIT clocks



Extended capabilities:

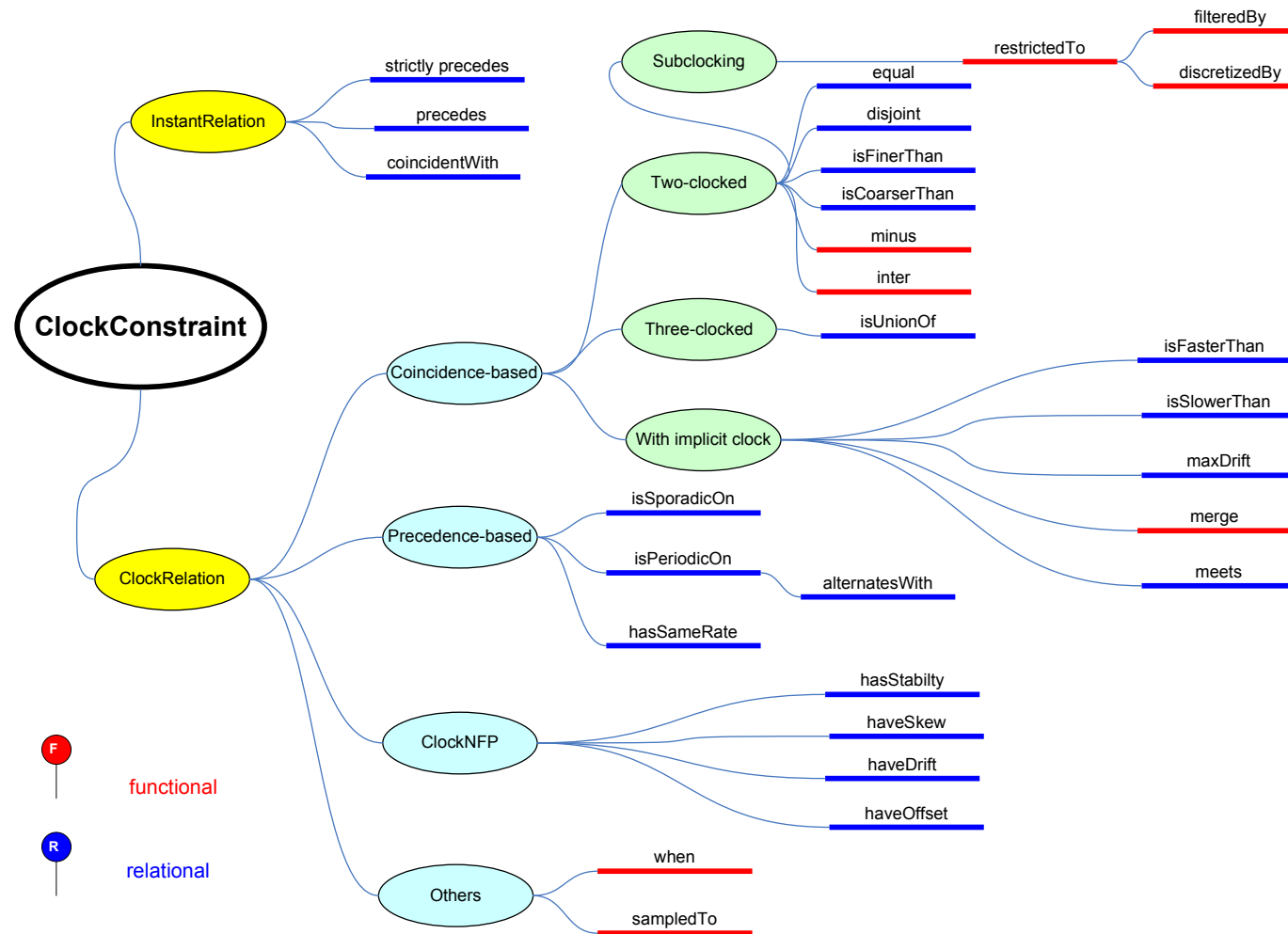
- Occurrence index
- Time intervals
- jitter





Expression of
Clock
dependencies

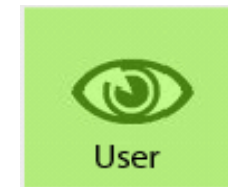
Clock Constraint Specification



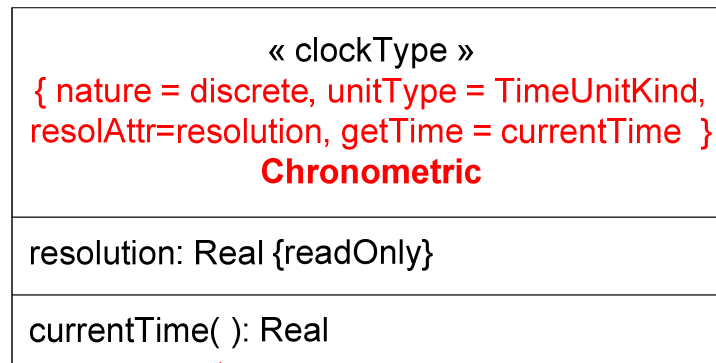
Pre-defined
Clock
Constraints

Each relation
has a
mathematical
specification

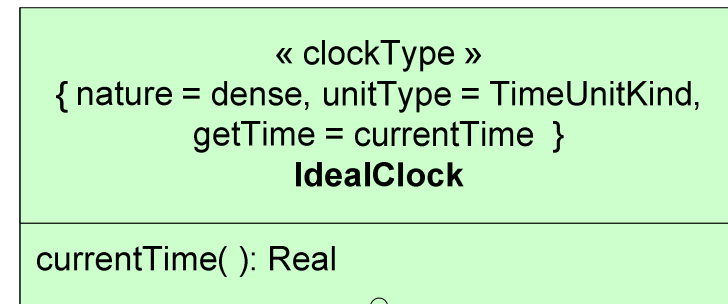
- **SPT, UML 2 and Time**
 - UML::CommonBehaviors::SimpleTime
- **the MARTE Time domain view**
 - a.k.a. the MARTE Time meta-model
 - Concepts and relationships
- **the MARTE Time sub-profile**
 - a.k.a. UML view
- **Usage of the Time sub-profile**



How to specify chronometric clocks



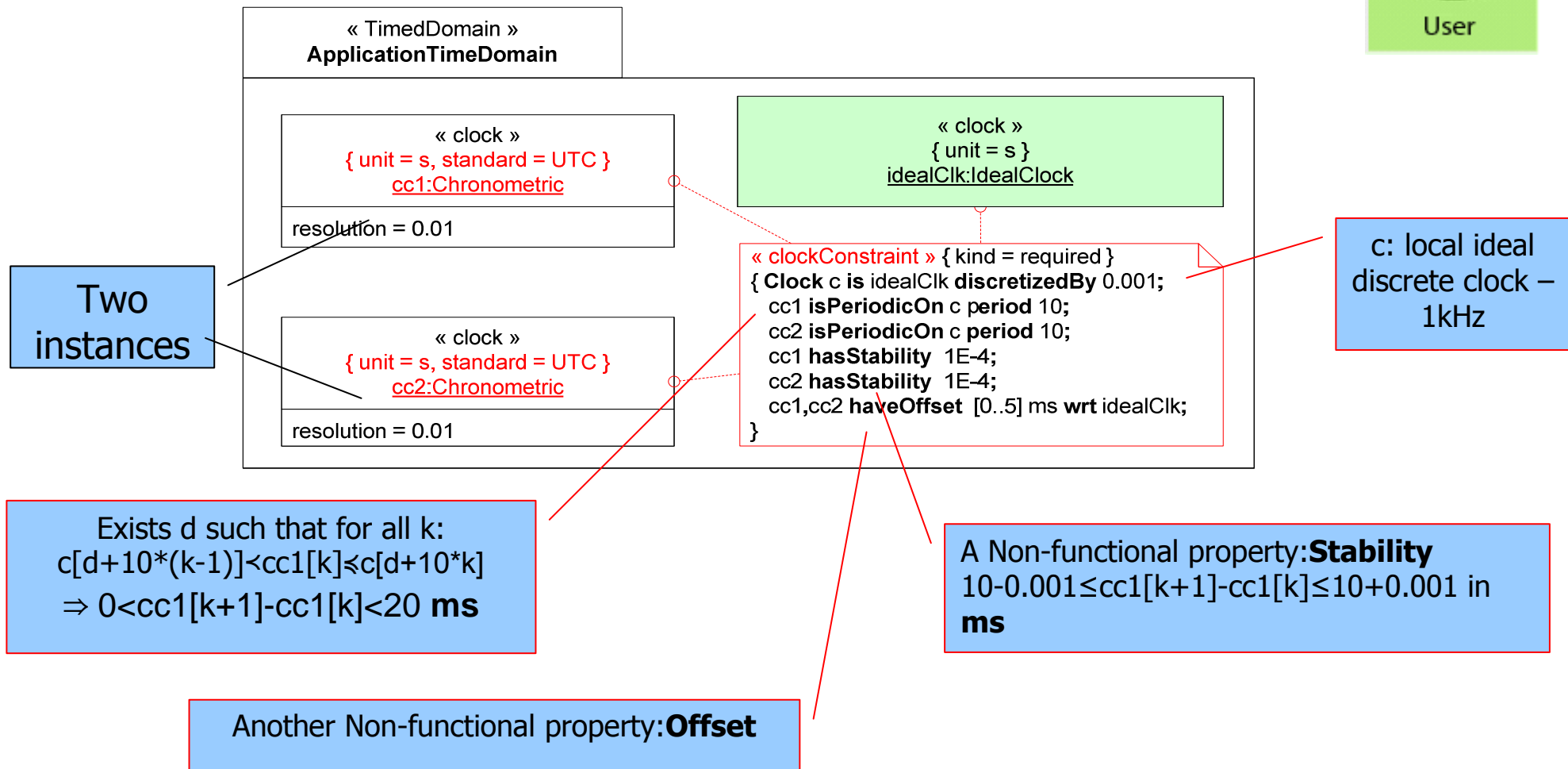
An user's defined
ClockType



Imported from
MARTE::TimeLibrary

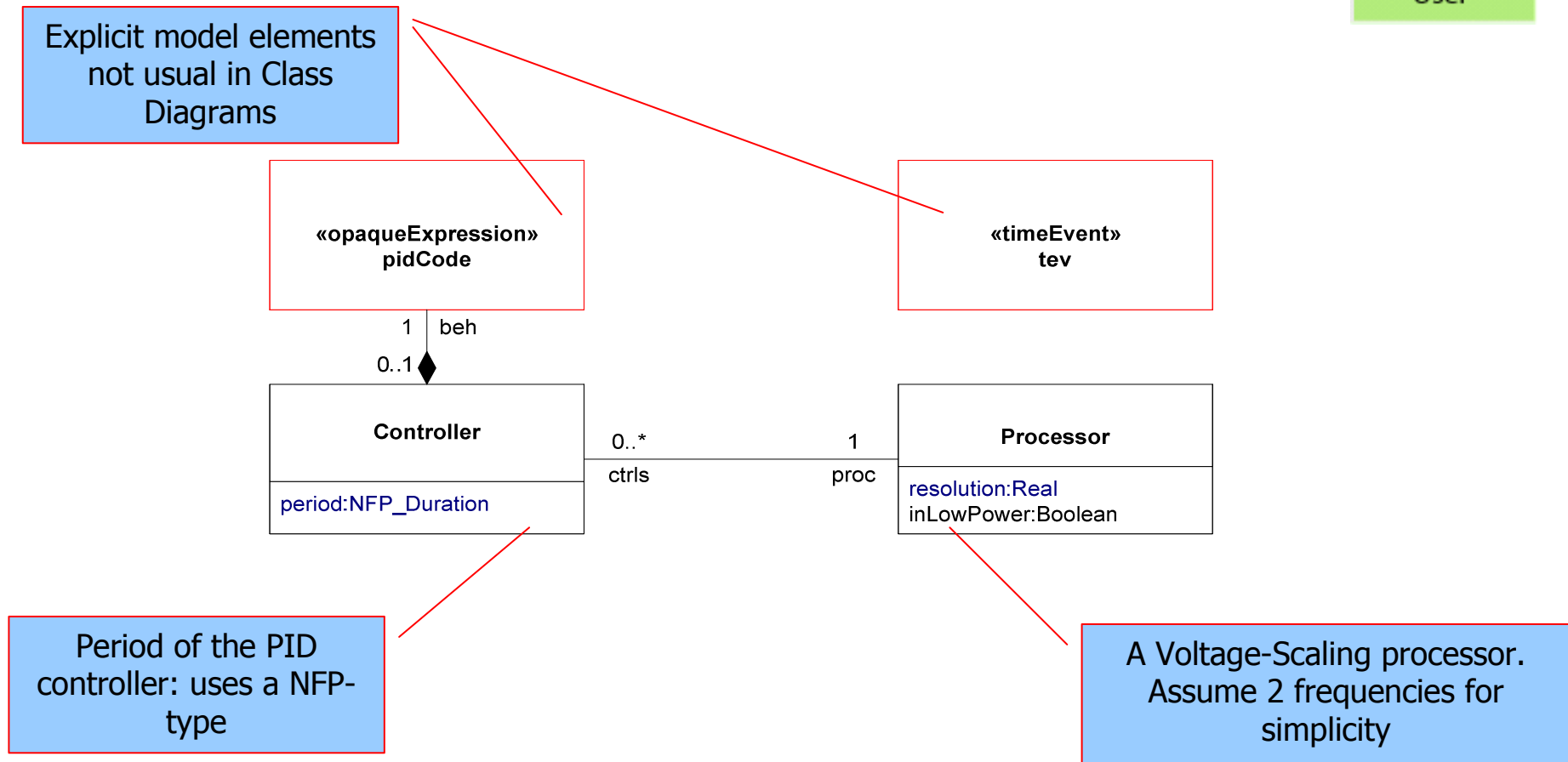
Chronometric Clocks (2)

Specifying NFP of (non ideal) chronometric clocks

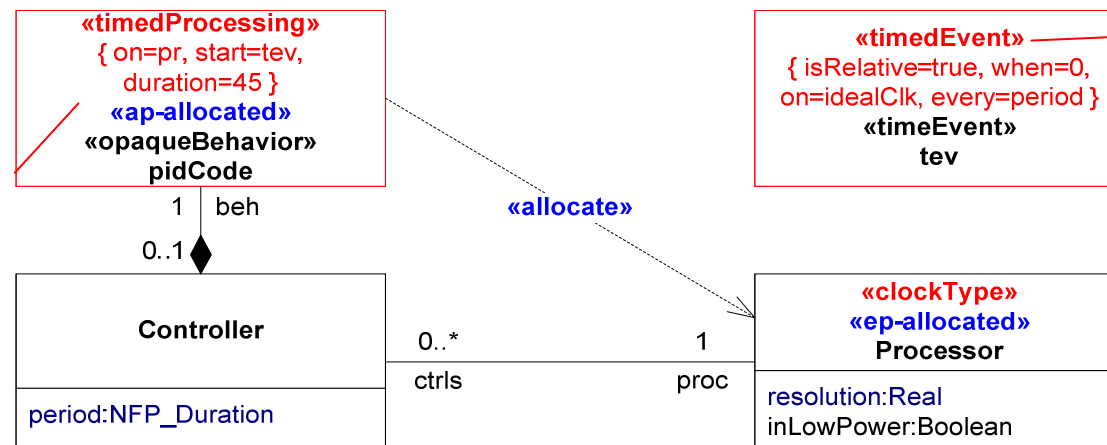


How to specify logical clocks:

1) Start with a standard UML class diagram



2) Apply MARTE stereotypes



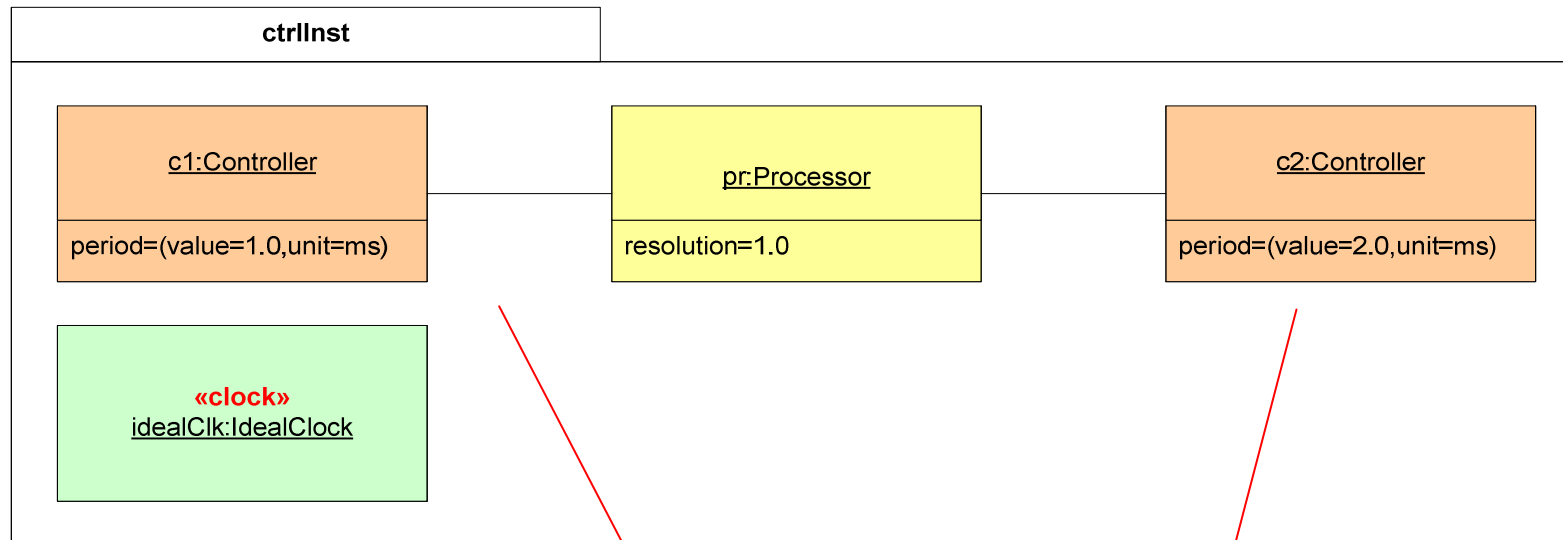
The pid code is triggered by tev and takes 45 cycles of Processor

Event tev is periodic on idealClock, the period is the value of the controller's attribute

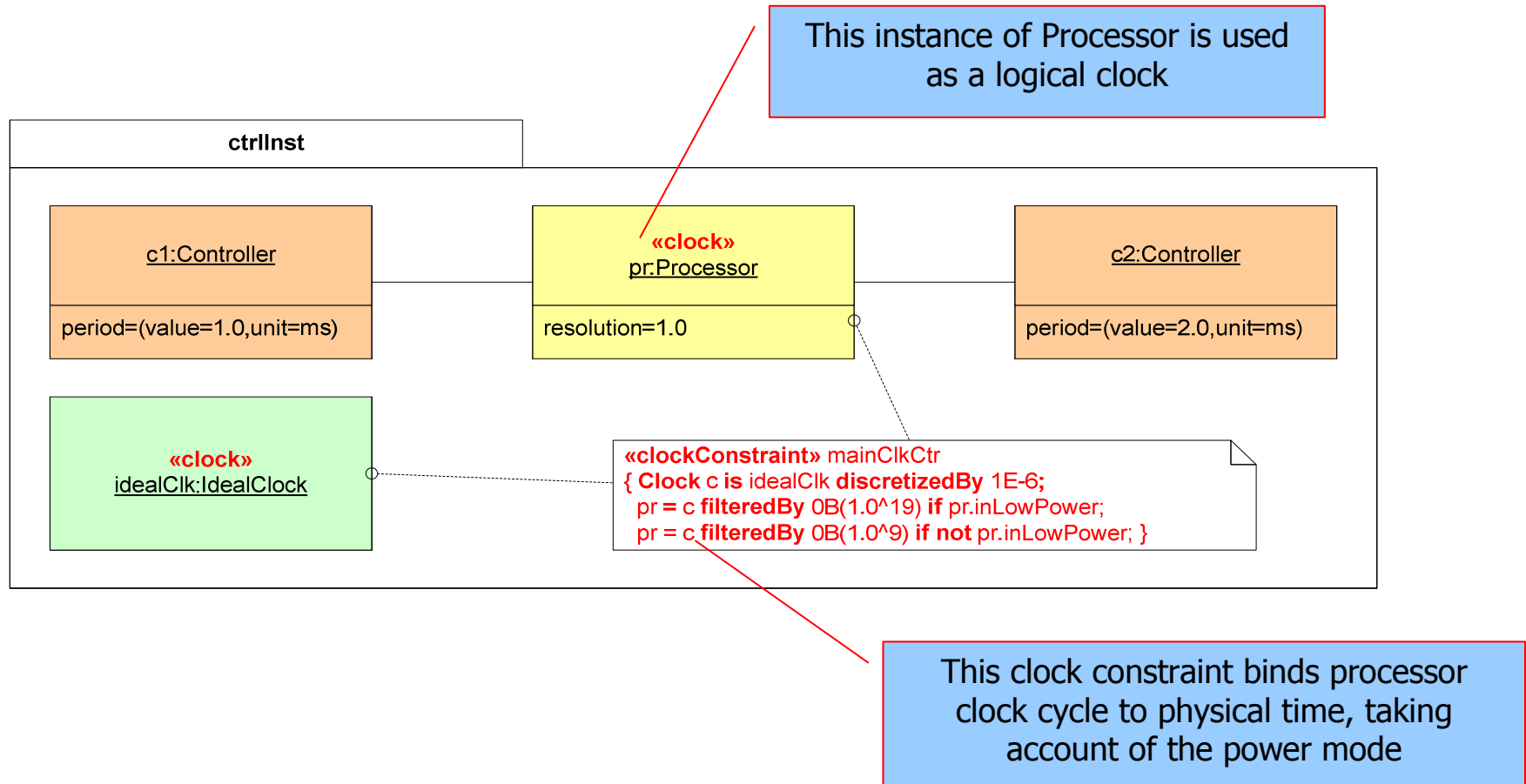
The class Processor is stereotyped by ClockType

3) Instantiate user's model elements

An **instance of the system** with an instance of Processor supporting two instances of Controller



4) Introduce clock (by stereotyping)

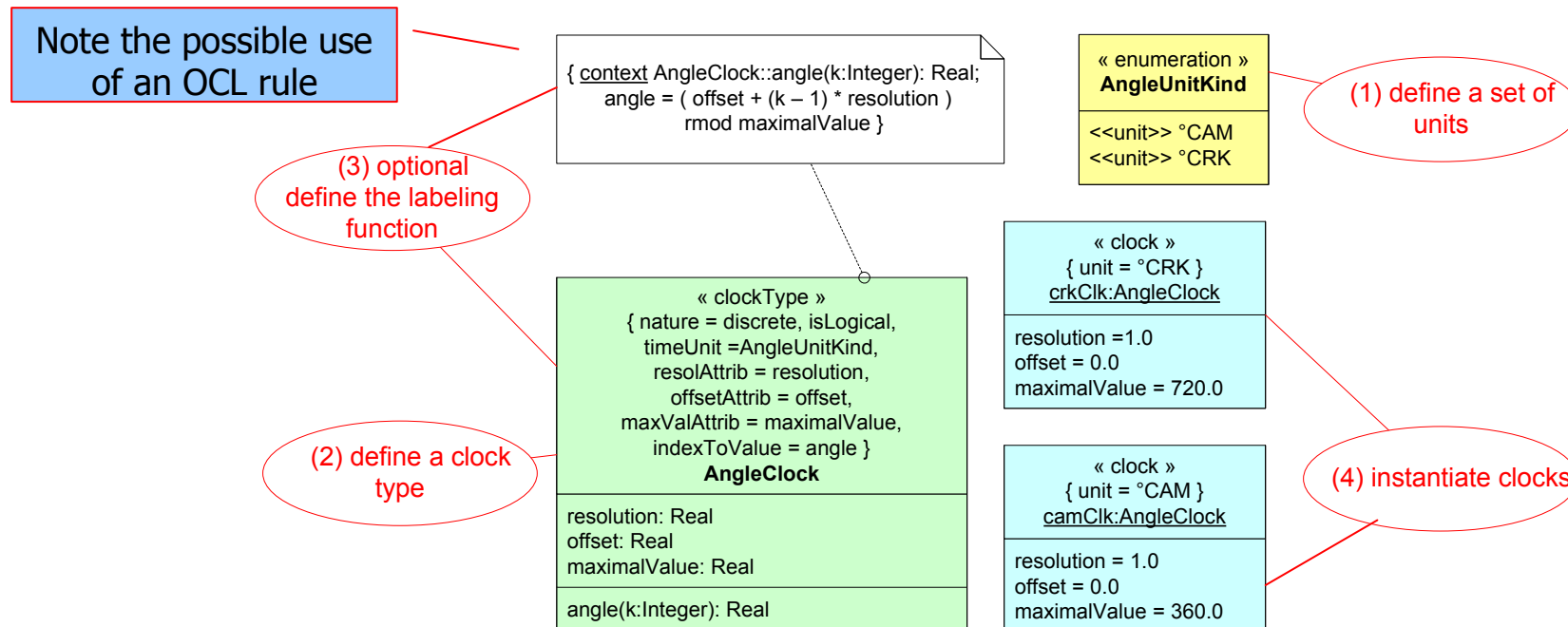


Another example of logical clocks

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.

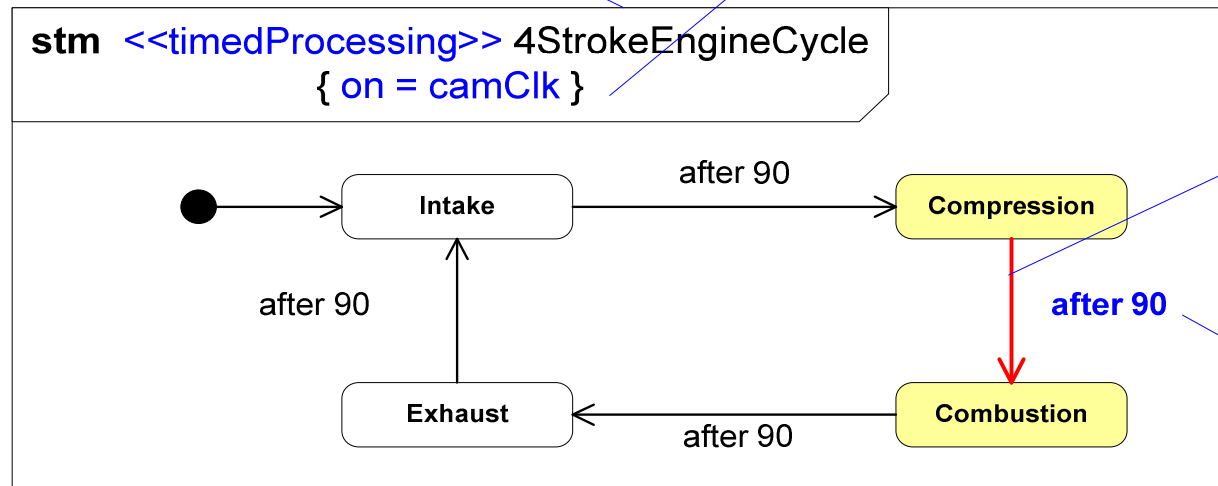


Example of usage of an "AngleClock"



Stereotyped State Machine.
Makes reference to a Clock

Reference to a (logical) clock, the
unit of which is °CAM (elsewhere
defined)



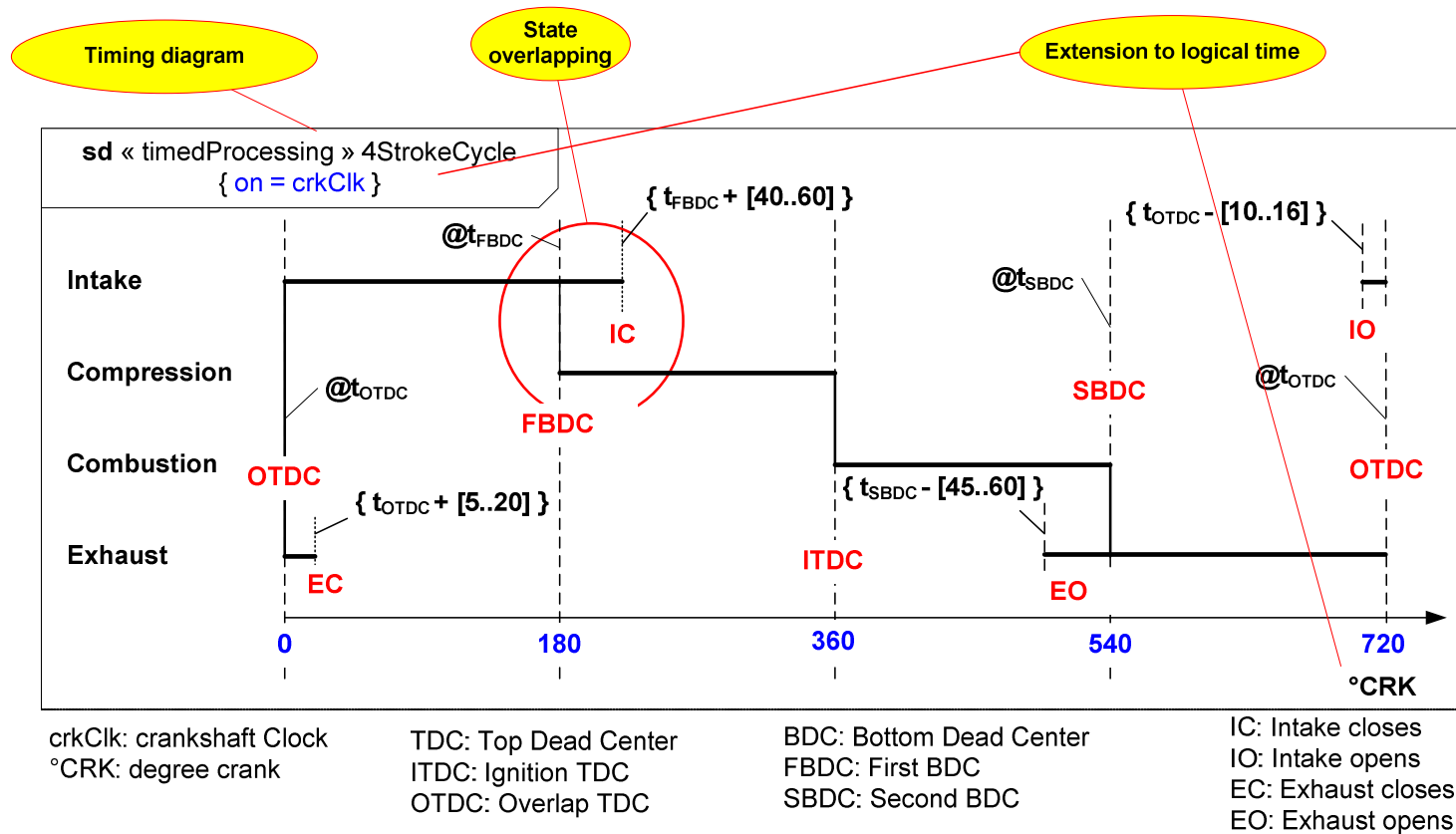
A transition

after 90

A trigger

Semantics:
90 °CAM after
entering state
Compression leave
this state and enter
state *Combustion*

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





Combining logical clocks:

ck is an AngleClock used to specify the ignition of a cylinder
c is the clock used to specify ignitions in a 4-cylinder engine

