



## The Real-Time UML Standard

Ben Watson  
Tri-Pacific Software Inc.

# The Real-Time A&D Group in OMG

- An OMG working group
  - mission: to investigate and issue requests (RFPs) for standard ways and means to apply UML to real-time problems
- Three principal areas of investigation:
  - Time-related modeling issues
  - General quality of service/fault tolerance modeling issues
  - Architectural modeling issues
- Status:
  - first RFP issued (April 1999)
  - second RFP being drafted

# Overview

- *The Real-Time UML Profile*
  - *Background*
  - Modeling Resources and Quality of Service
  - Modeling Time
  - Modeling Concurrency
  - The Schedulability Analysis Sub-Profile
  - The Performance Analysis Sub-Profile
  - The Real-Time CORBA Schedulability Analysis Sub-Profile
  - Real-Time CORBA UML Model
  - Model Processing Cycle
- Summary

# The Real-Time UML RFP

- *“UML Profile for Schedulability, Performance and Time”*
  - First in a series of real-time specific RFPs
  - RFP issued in March 1999
  - Initial proposal submitted in August 2000 (ad/2000-08-04)
  - Standard adopted in September 2001
- Standard methods for UML modeling of:
  - Physical time
  - Timing specifications
  - Timing services and mechanisms
  - Modeling resources (logical and physical)
  - Concurrency and scheduling
  - Software and hardware infrastructure and their mapping
  - ..including specific notations for the above where necessary

# Response to the RFP

- Just one submission throughout
- Consortium team:
  - ARTiSAN (UML tool vendor)
  - I-Logix (UML tool vendor)
  - Rational (UML tool vendor)
  - Telelogic (UML tool vendor)
  - TimeSys (RT tool and technology vendor)
  - Tri-Pacific Software (RT tool vendor)
- In consultation with many of the top real-time system experts (tool builders, analysis technique experts, academics)

# RT Profile: Guiding Principles

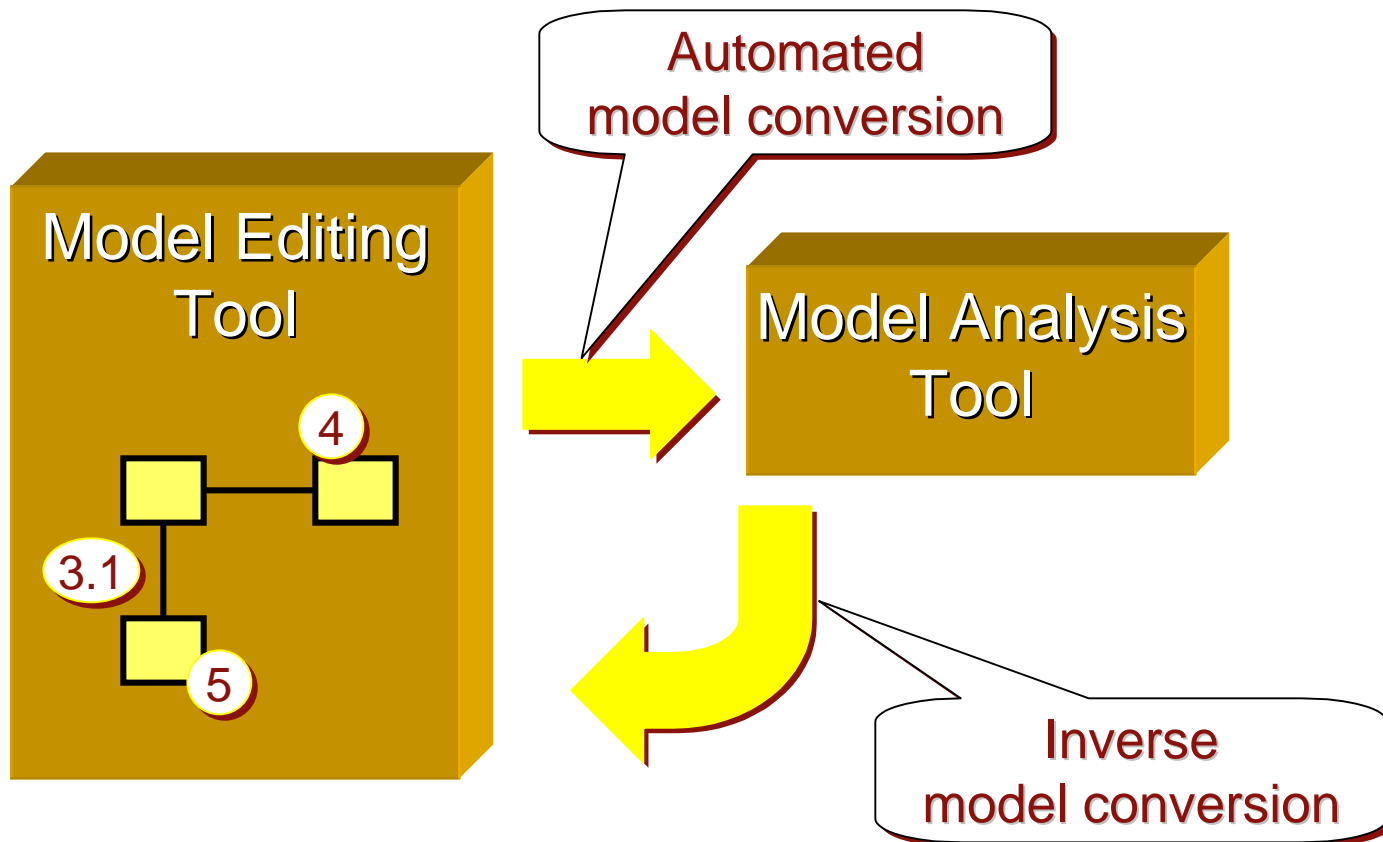
- Ability to specify quantitative information directly in UML models
  - key to quantitative analysis and predictive modeling
- Flexibility:
  - users can model their RT systems using modeling approaches and styles of their own choosing
  - open to existing and new analysis techniques
- Facilitate the use of analysis methods
  - eliminate the need for a deep understanding of analysis methods
  - as much as possible, automate the generation of analysis models and the analysis process itself

# Quantitative Methods for RT Systems

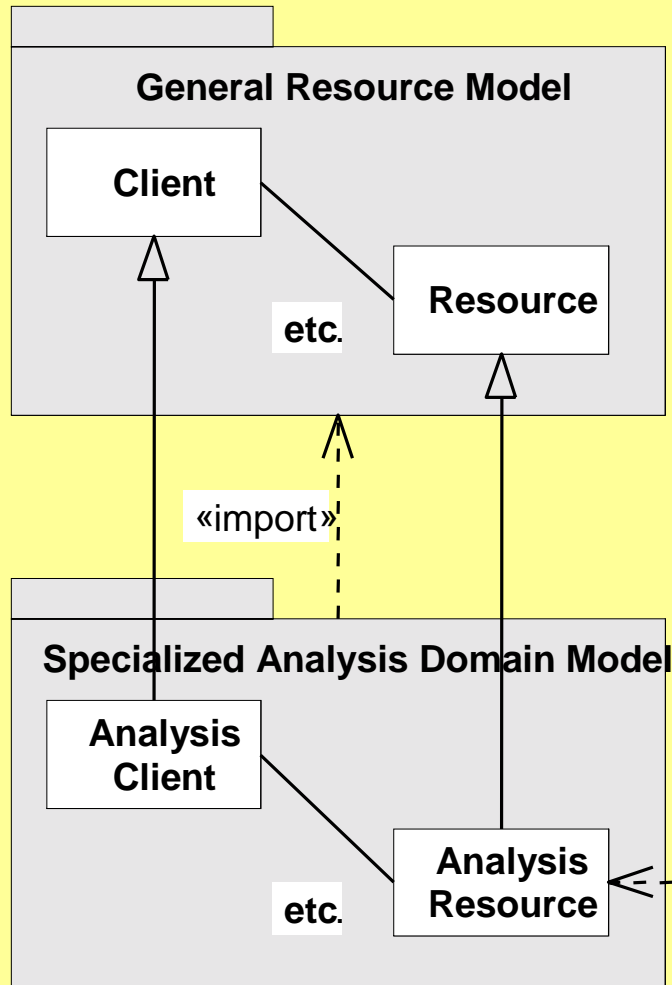
- Once we have included QoS information in our models, we can use *quantitative methods* to:
  - predict system characteristics (detect problems early)
  - analyze existing system
  - synthesize elements of the model
- Methods considered for the profile:
  - **Schedulability analysis**  
*will the system meet all of its deadlines?*
  - **Performance analysis** based on queueing theory  
*what kind of response will the system have under load?*

# Desired Development Model

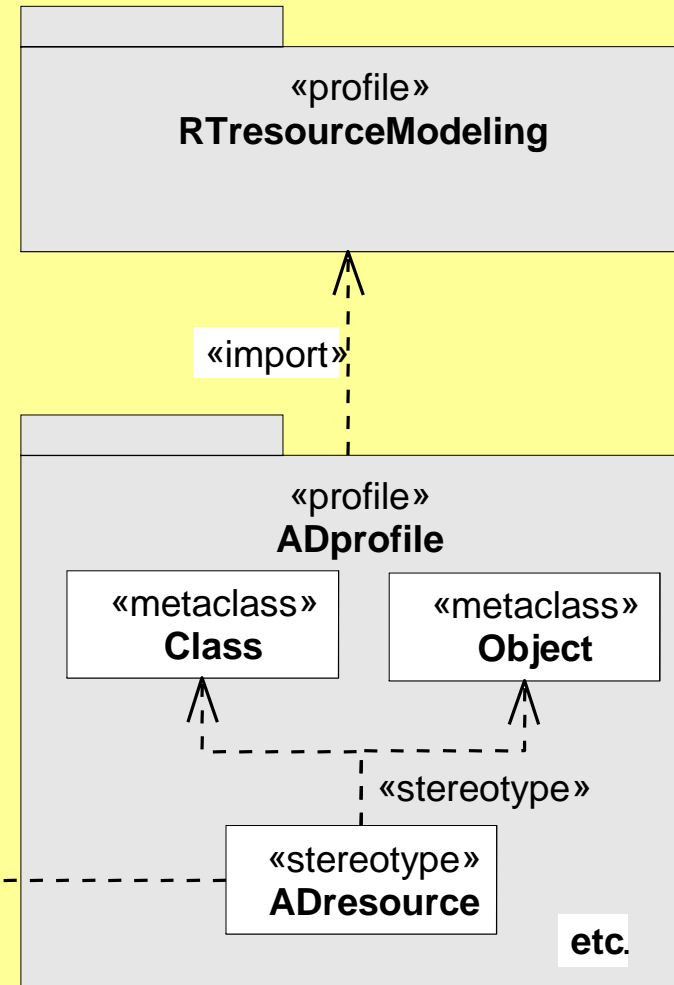
- Seamless integration of technologies and tools based on standards for real-time modeling



# Structure: Normative and Non-Normative Parts

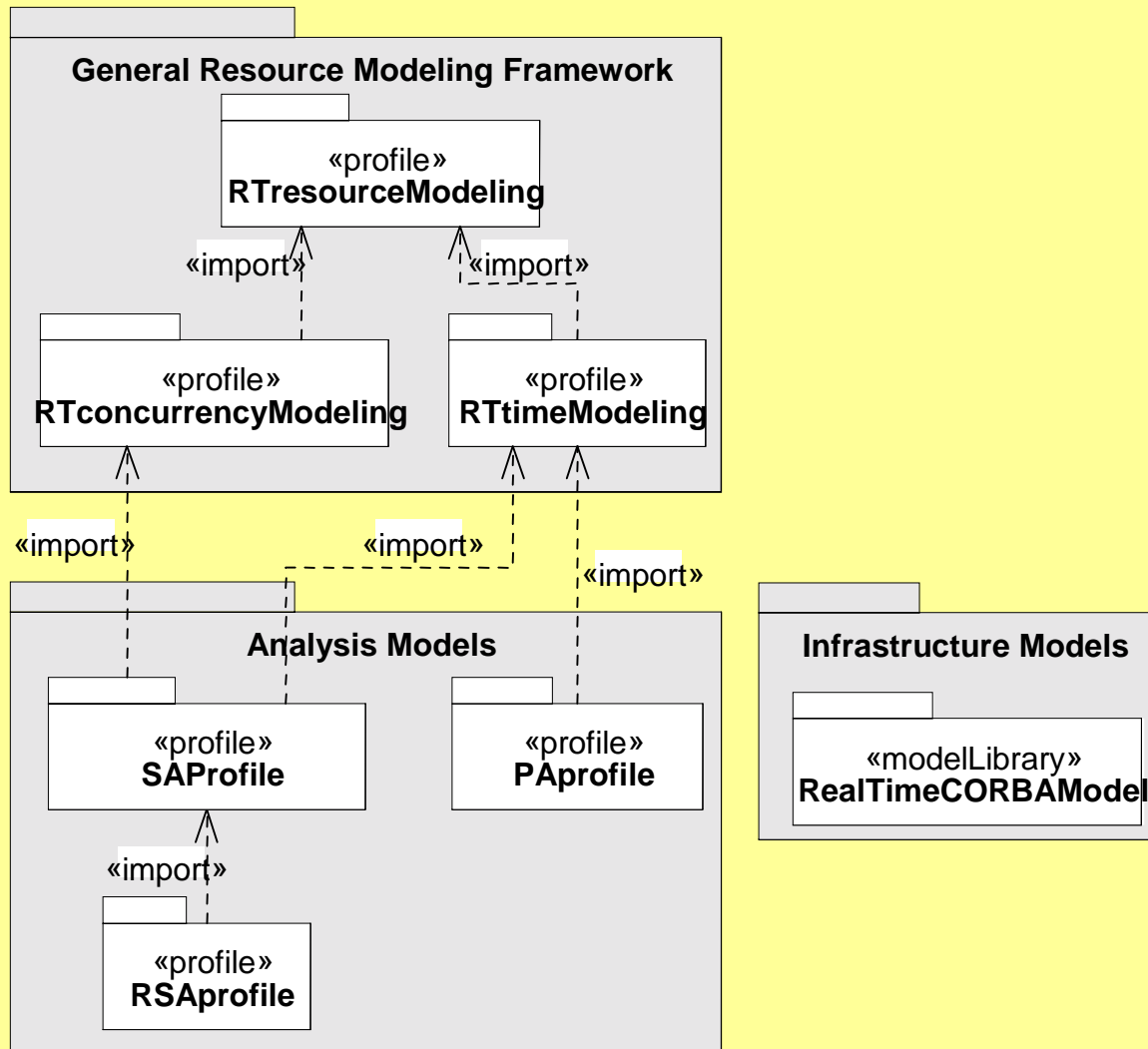


What (non-normative)



How (normative)

# UML Real-Time Profile Structure (Normative Part)



# Overview

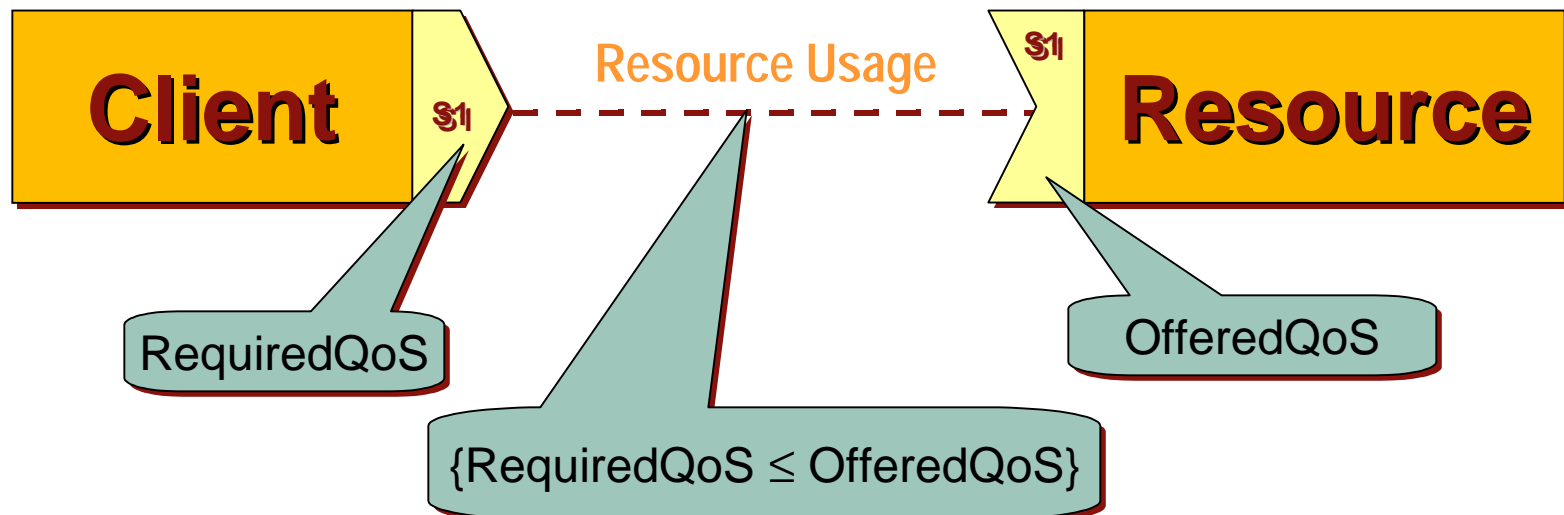
- The Real-Time UML Profile
  - Background
  - *Modeling Resources and Quality of Service*
  - Modeling Time
  - Modeling Concurrency
  - The Schedulability Analysis Sub-Profile
  - The Performance Analysis Sub-Profile
  - The Real-Time CORBA Schedulability Analysis Sub-Profile
  - Real-Time CORBA UML Model
  - Model Processing Cycle
- Summary

# Quality of Service Concepts

- An abstract, technology-independent representation of the engineering model can be specified using the general concept of *Quality of Service (QoS)*:
  - a specification (usually quantitative) of how a particular service is (to be) performed*
  - e.g. throughput, capacity, response time
- The specification of a model element can include:
  - *offered QoS*: the QoS that it provides to its clients
  - *required QoS*: the QoS it requires from other components to support its QoS obligations

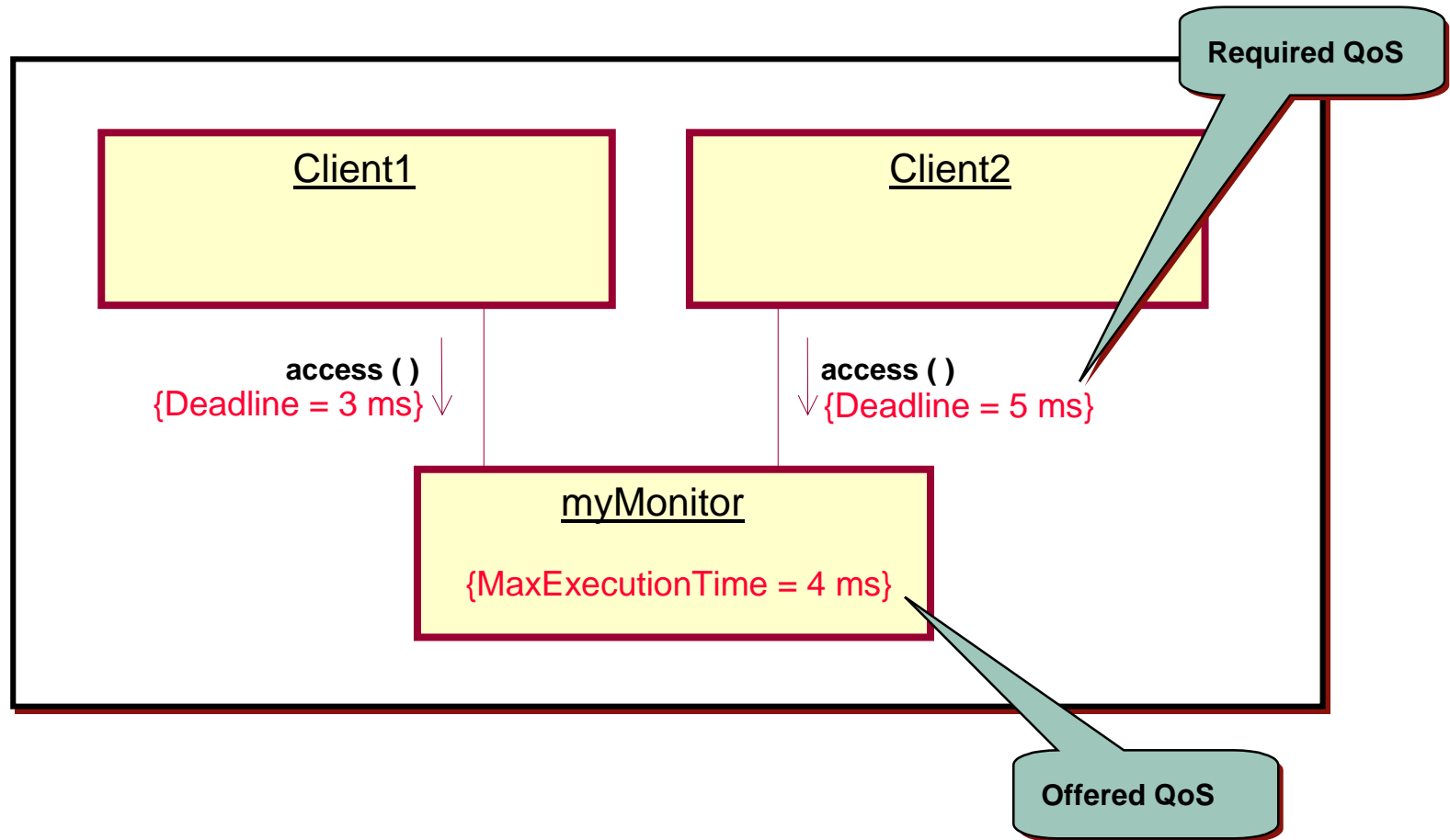
# Resources and Quality of Service

- **Resource**: an element whose service capacity is limited, directly or indirectly, by the finite capacities of the underlying physical computing environment
- The services of a resource are characterized by one or more **quality of service (QoS)** attributes
  - capacity, reliability, availability, response time, etc.

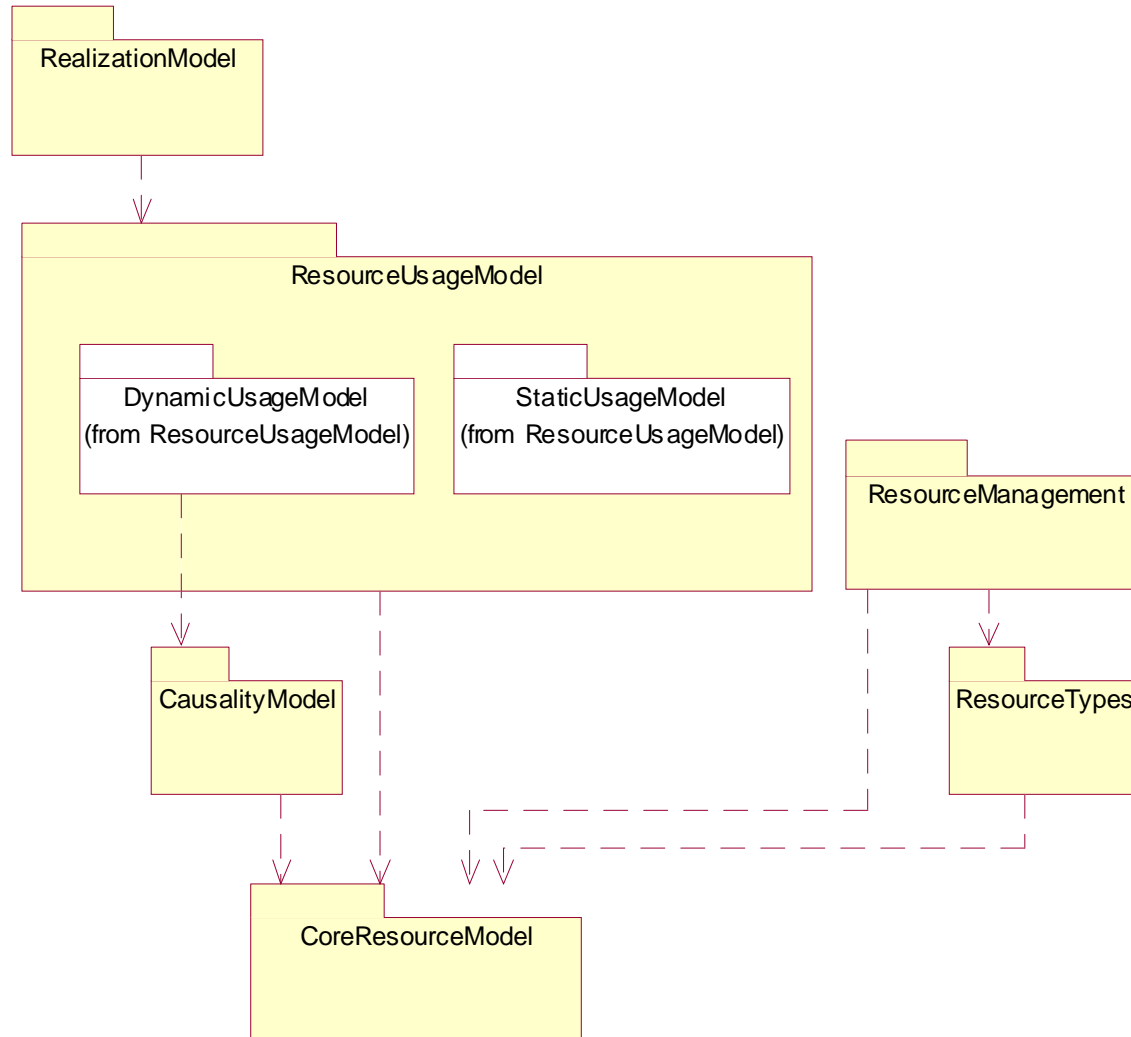


# Simple Example

- Concurrent tasks accessing a monitor with known response time characteristics



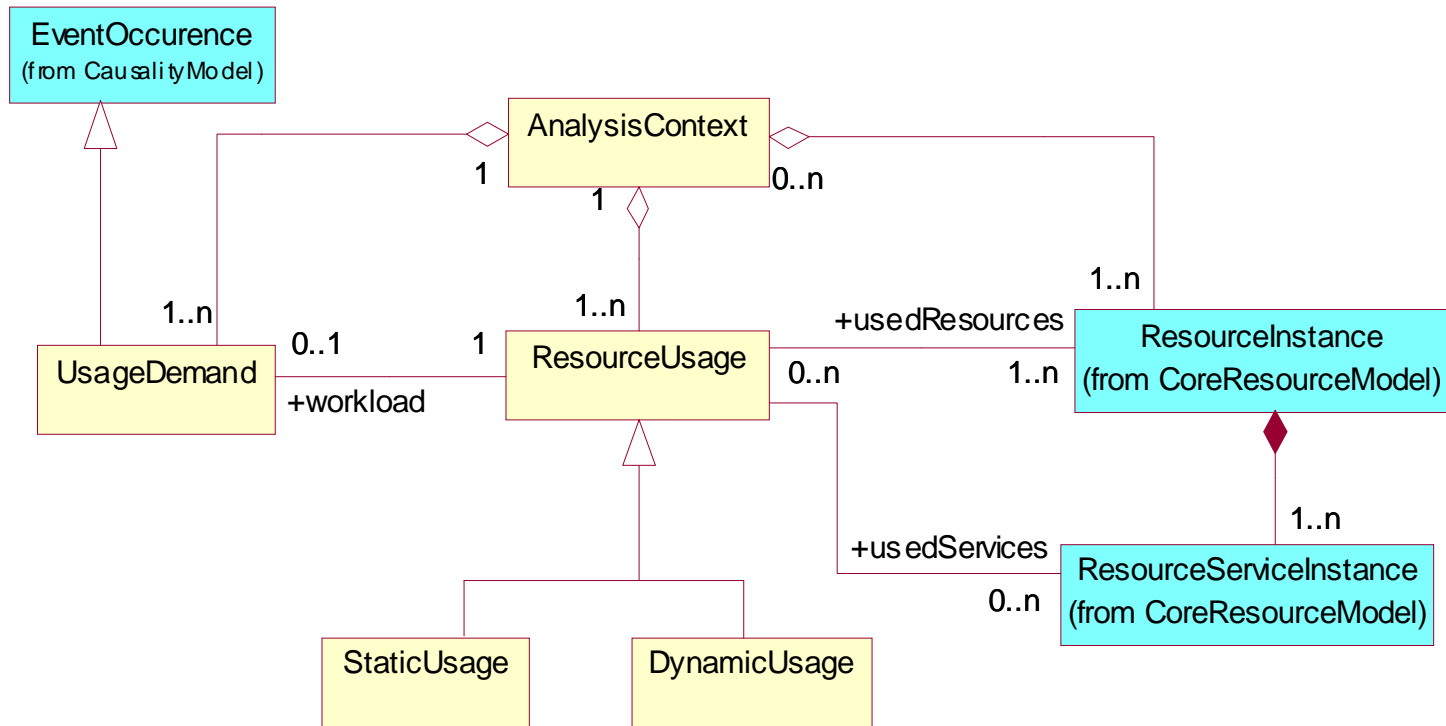
# The General Resource Model



# Instances Vs. Specifications

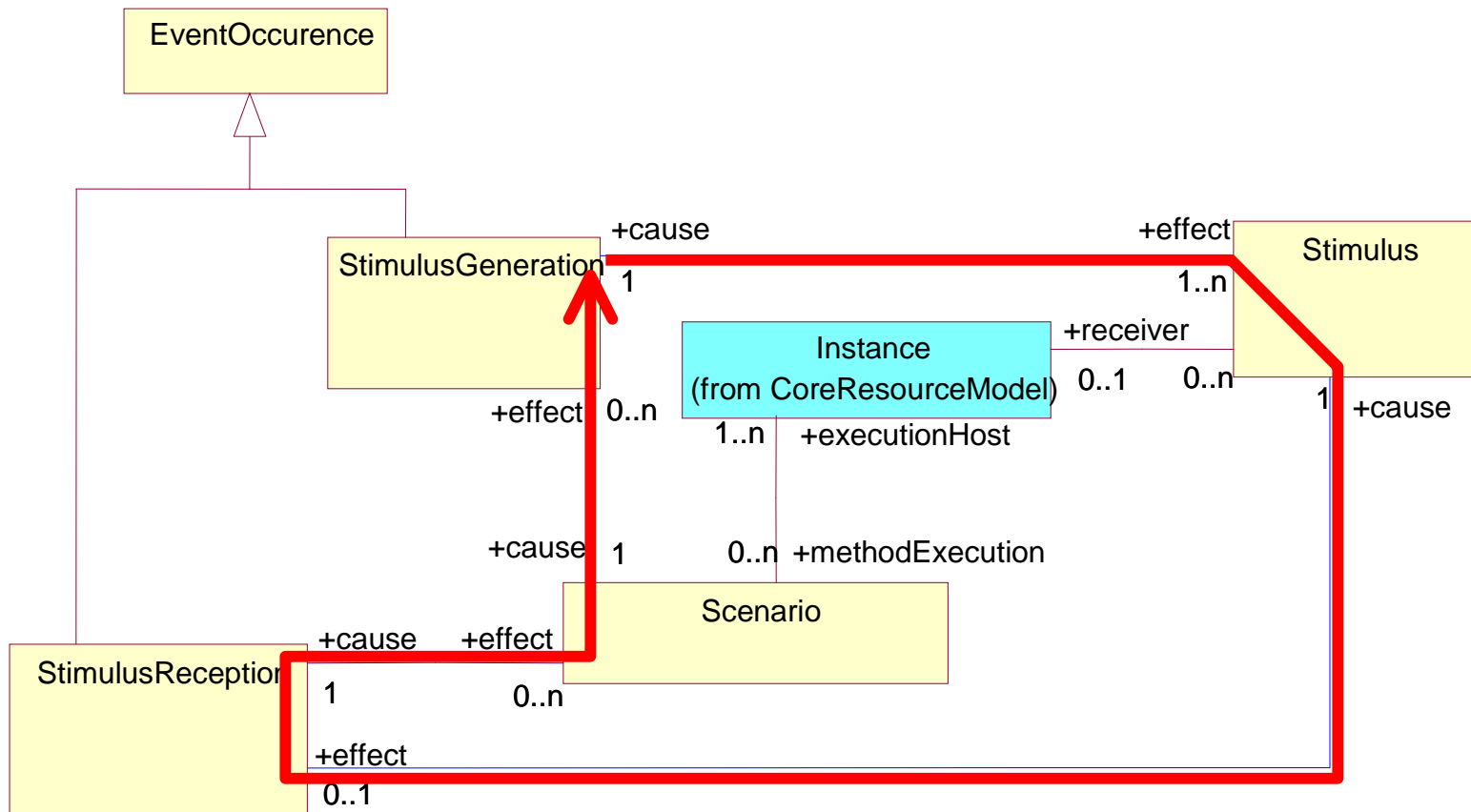
- Practically all analysis methods are concerned with instance-based models
  - E.g., class models are too abstract to be analyzable
  - E.g., two executions of the same action (specification) may have very different execution times
- However, it is often useful to associate QoS characteristics with classes
  - Used to define default values that may be overridden for specific instances
- Need to apply a stereotype to both spec elements and instance elements
  - E.g., Class and Object

# Basic Resource Usage Model



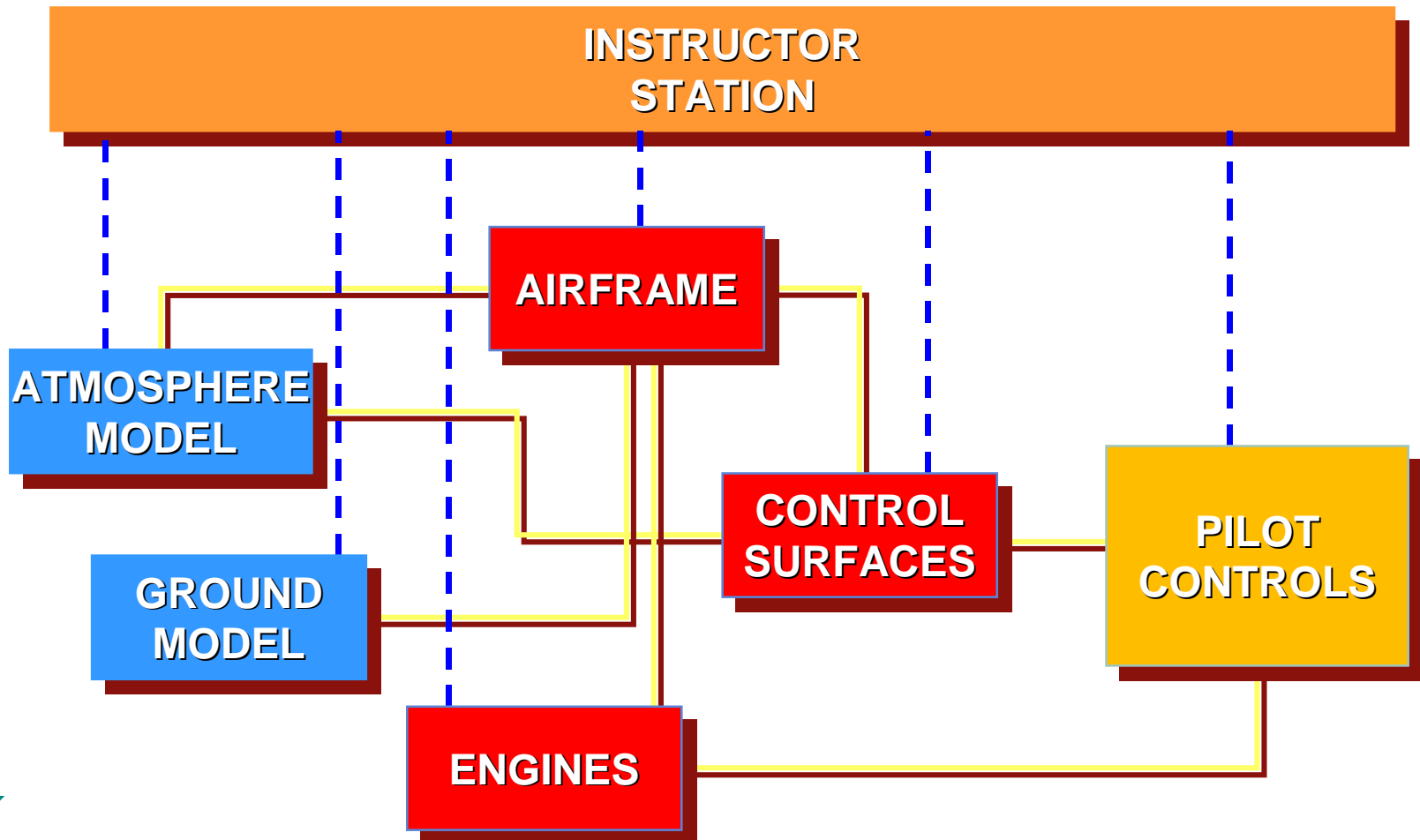
# Basic Causality Loop

- Used in modeling dynamic scenarios



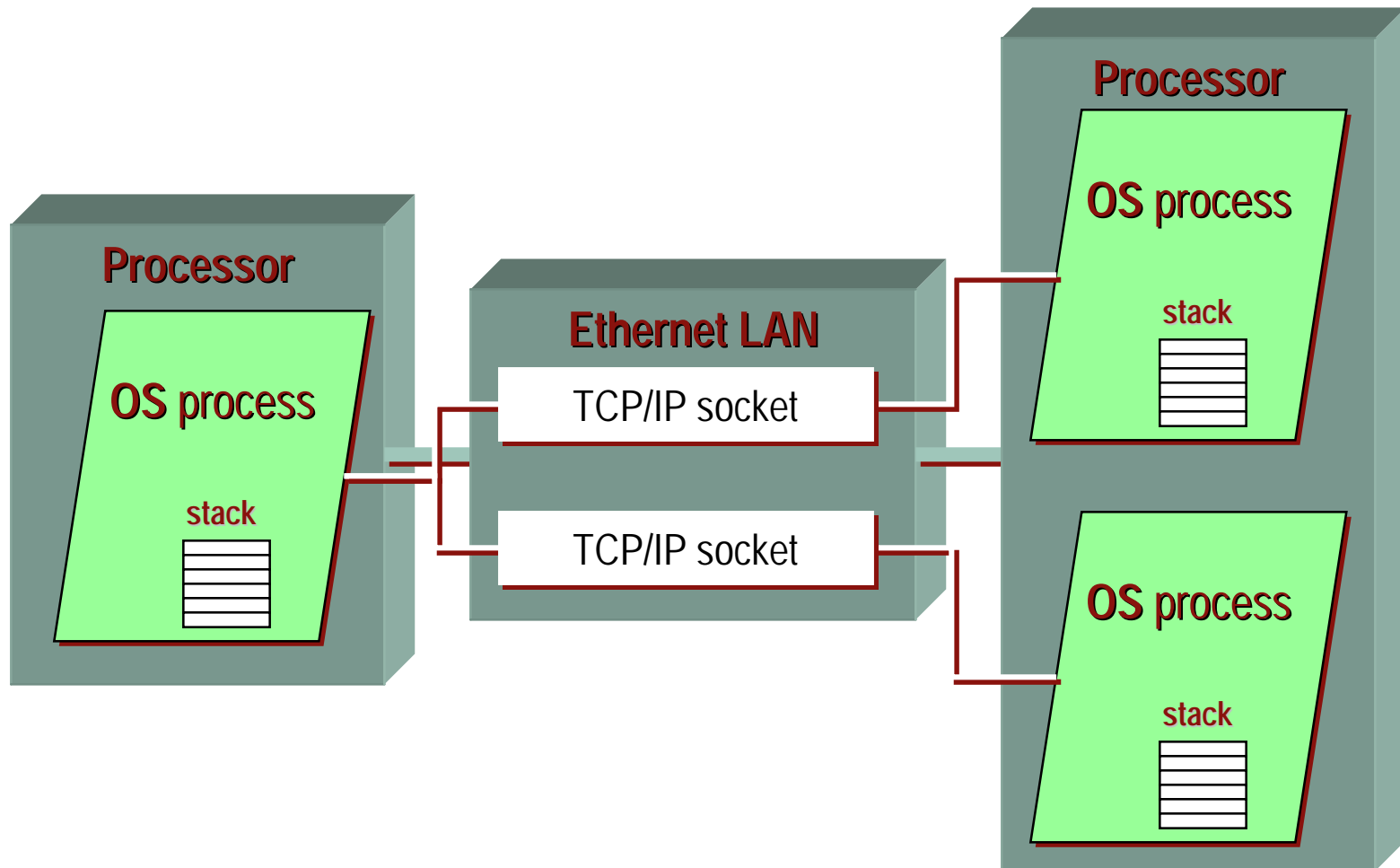
# Logical (Conceptual) Viewpoint

- A technology-independent view of the software
  - a “virtual” mechanism realized by a computer

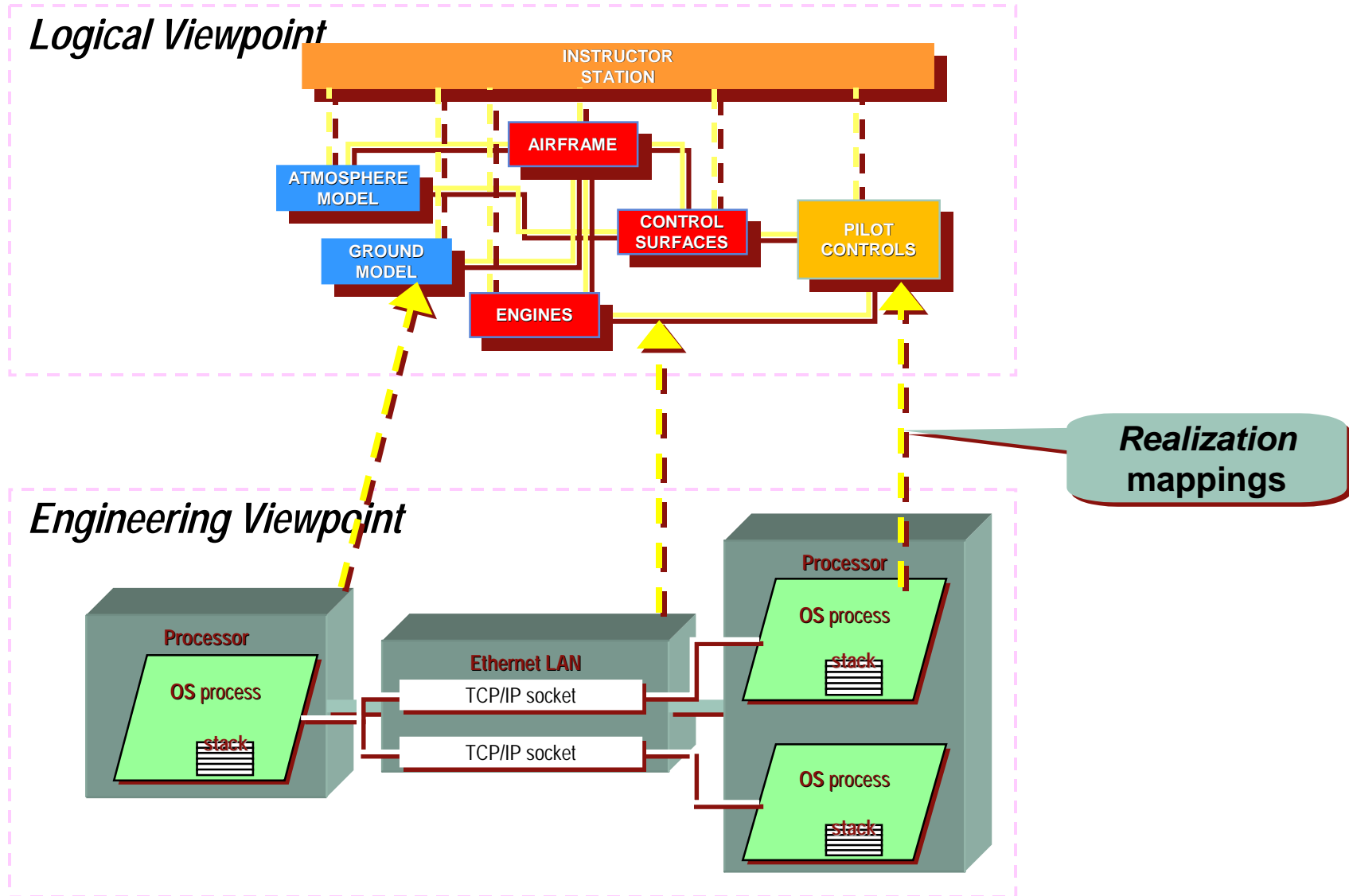


# Engineering (Realization) Viewpoint

- The realization of a specific set of logical components using facilities of the run-time environment

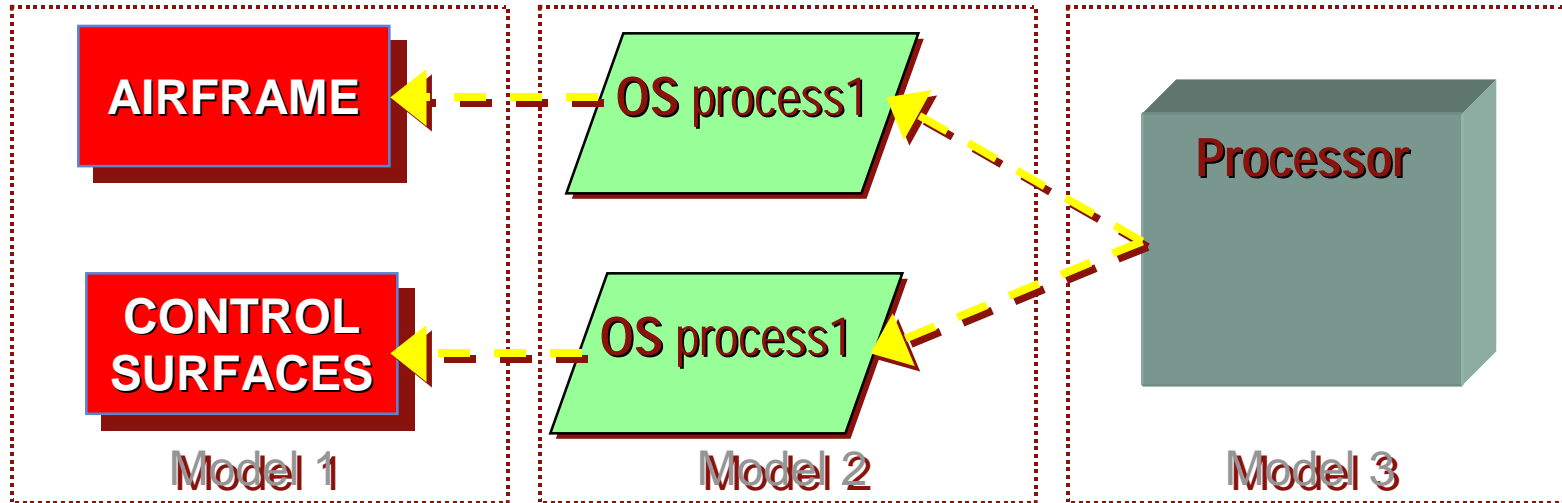


# Viewpoints and Mappings

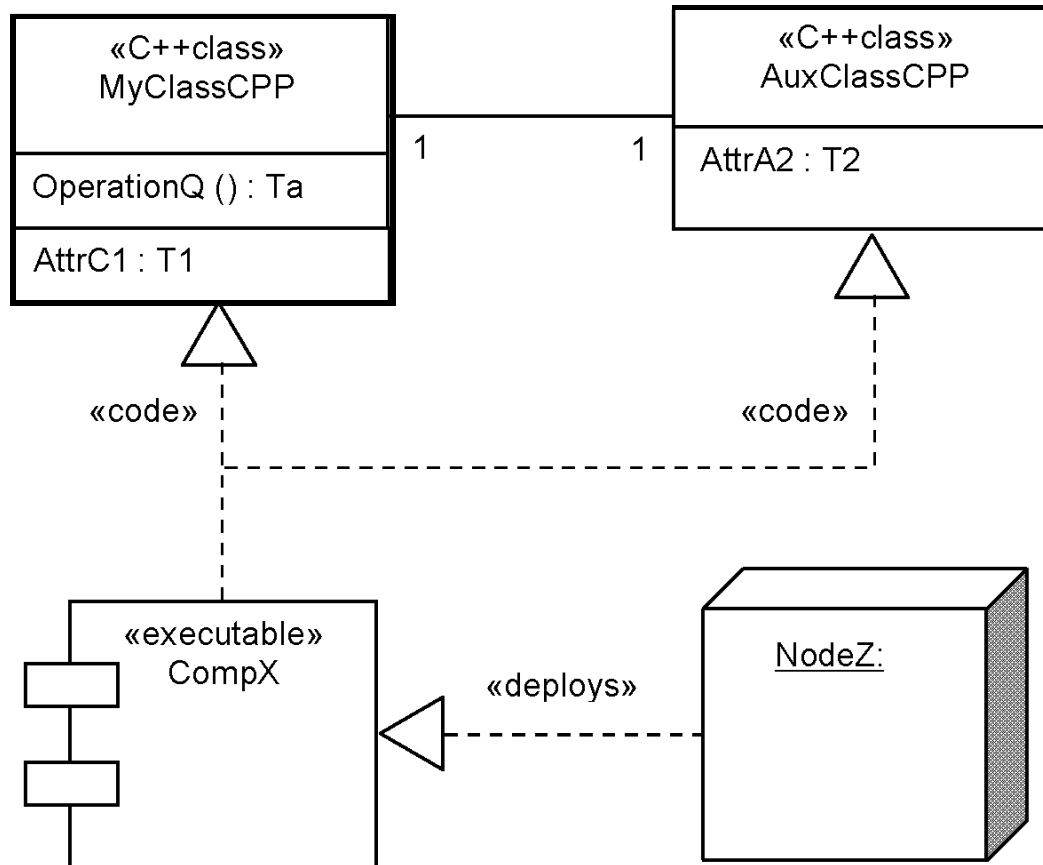


# Realization Mappings

- A correspondence between elements of two distinct models (logical and engineering)
- Semantics: the logical elements are *implemented* by the corresponding engineering model elements
  - logical elements can be viewed as “residing” on the corresponding engineering elements

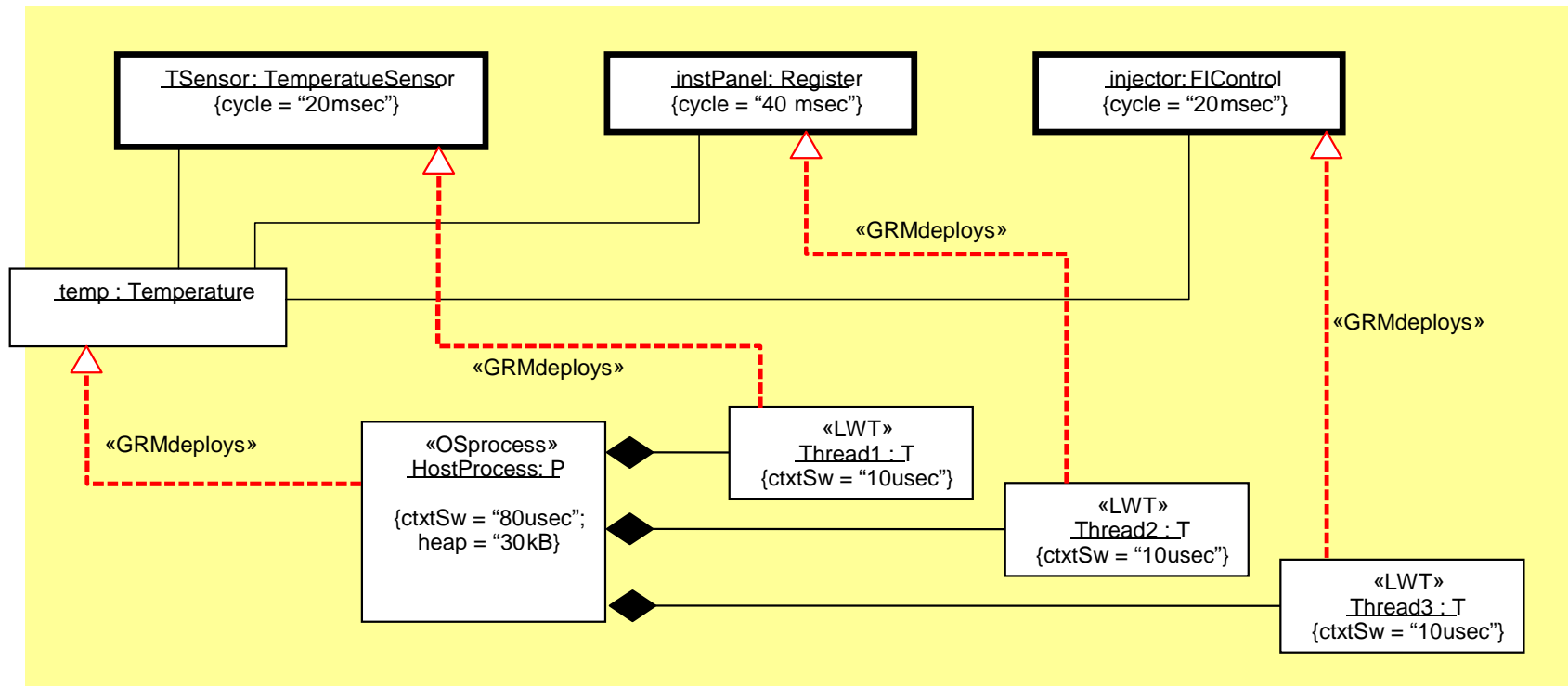


# Forms of Realization



# Realization Relationships

- For sophisticated multi-layer deployment modeling



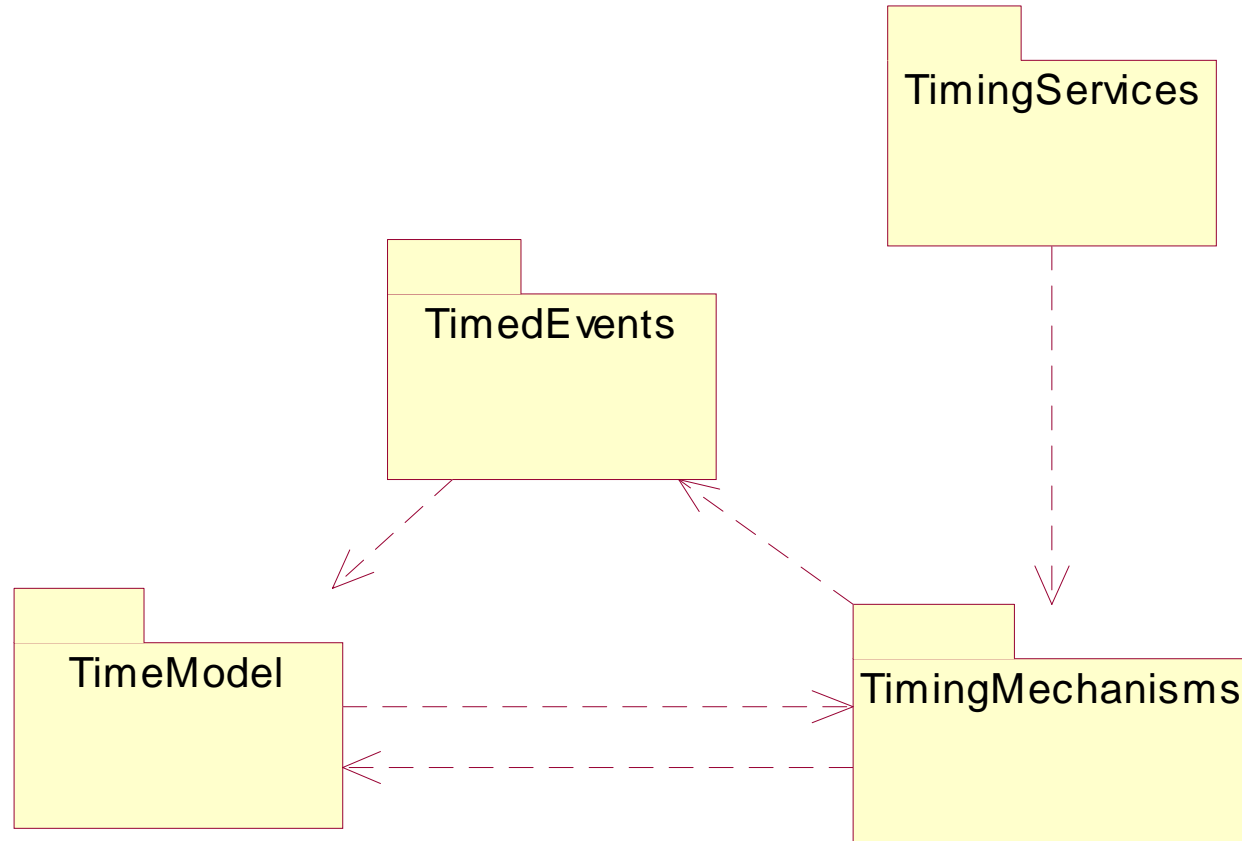
# Defined Stereotypes

Stereotype	Applies To	Tags	Description
«GRMacquire»	Stimulus, Message, Action, Operation, Reception...	GRMblocking [0..1] GRMexclServ [0..1]	An operation or invocation that acquires access rights to an exclusive resource
«GRMcode» (subclass of «GRMrealize»)	Abstraction	GRMmapping [0..1]	Relates a logical element model to the component that contains its code
«GRMdeploys» (subclass of «GRMrealize»)	Abstraction	GRMmapping [0..1]	Identifies where logical model elements are deployed
«GRMrealize»	Abstraction	GRMmapping [0..1]	A realization mapping
«GRMrelease»	Stimulus, Message, Action, Operation, Reception...	GRMexclServ [0..1]	An operation or invocation that releases an exclusive resource acquired previously
«GRMrequires» (subclass of «GRMrealize»)	Abstraction (subclass of «GRMrealize»)	GRMmapping [0..1]	A specification of a required environment for one or more logical model elements

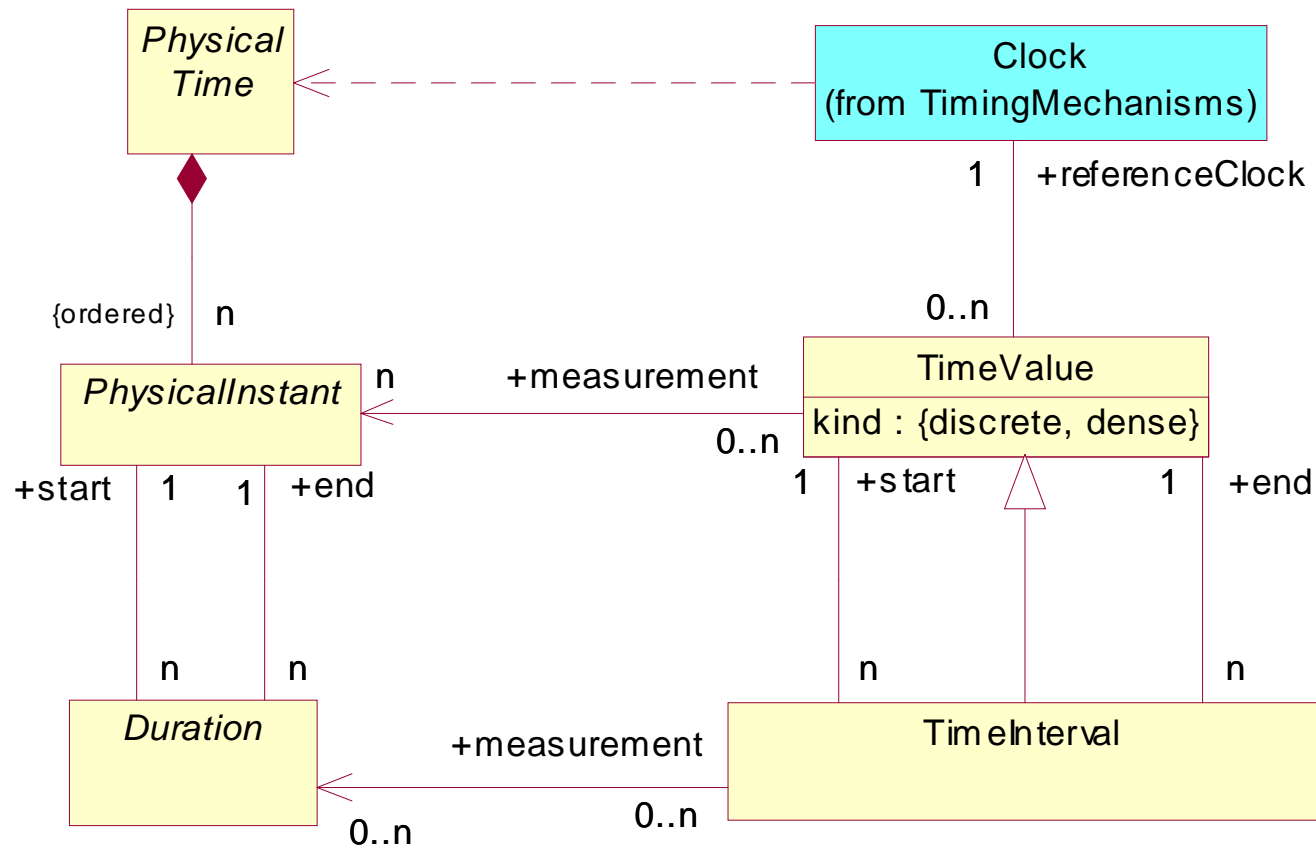
# Overview

- The Real-Time UML Profile
  - Background
  - Modeling Resources and Quality of Service
  - *Modeling Time*
  - Modeling Concurrency
  - The Schedulability Analysis Sub-Profile
  - The Performance Analysis Sub-Profile
  - The Real-Time CORBA Schedulability Analysis Sub-Profile
  - Real-Time CORBA UML Model
  - Model Processing Cycle
- Summary

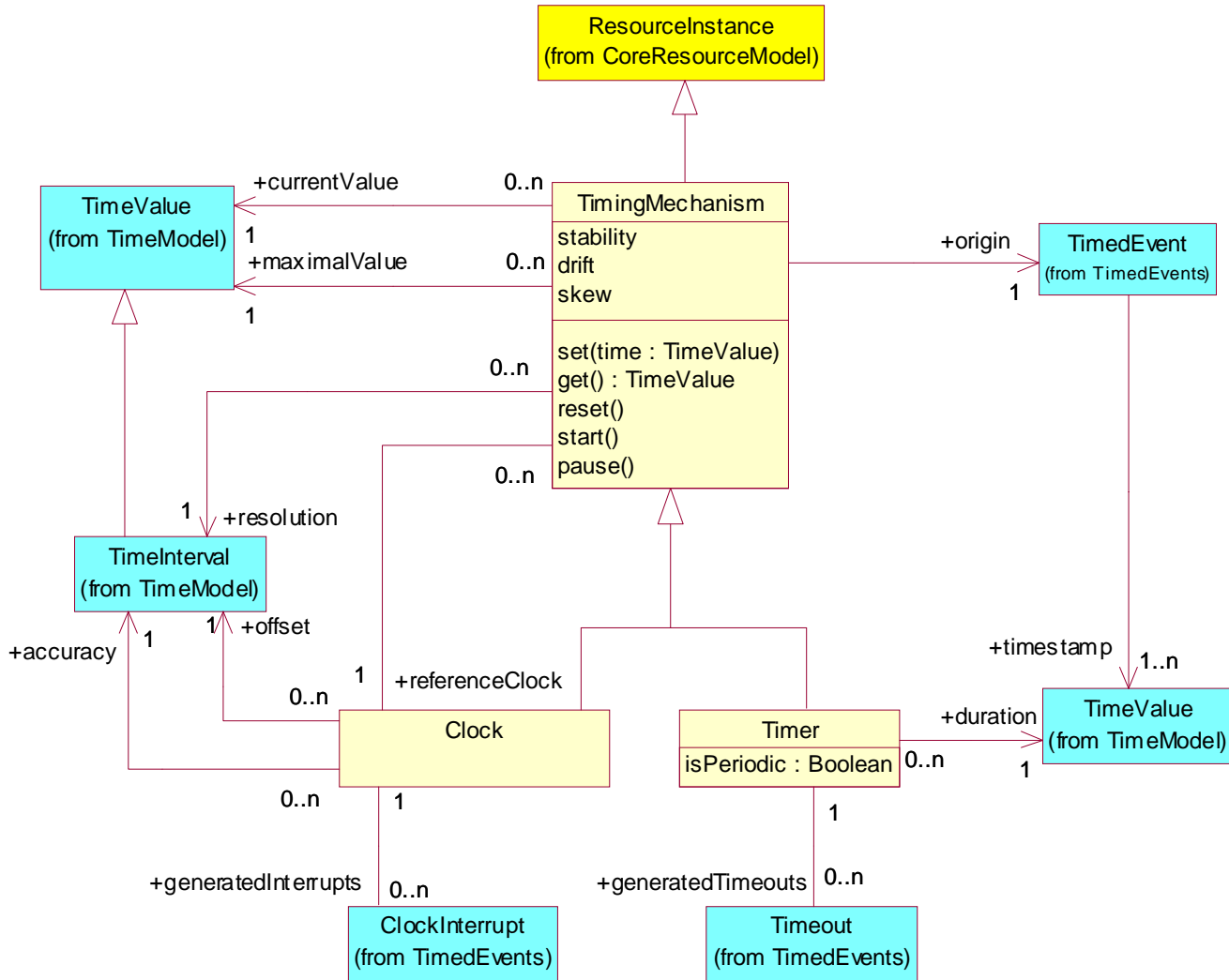
# General Time Model



# Physical and Measured Time



# Timing Mechanisms Model



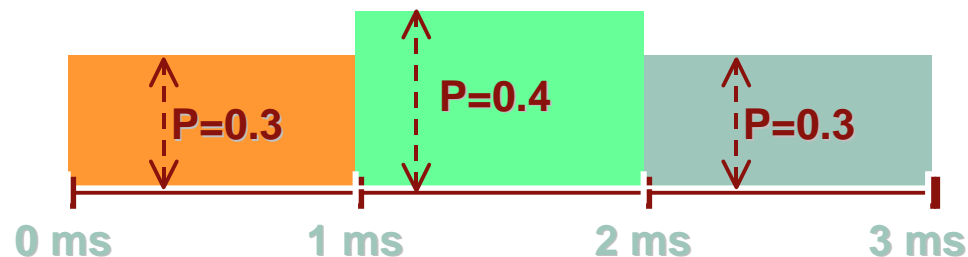
# Example Timing Stereotype

Stereotype	Base Class	Tags
«RTaction»	Action	RTstart RTend RTduration
	ActionExecution	
	Message	
	Stimulus	
	Method	
	ActionSequence	
	ActionState	
	SubactivityState	
	Transition	
	State	

Tag	Tag Type	Multiplicity	Domain Name
RTstart	RTtimeValue	[0..1]	TimedAction::start
RTend	RTtimeValue	[0..1]	TimedAction::end
RTduration	RTtimeValue	[0..1]	TimedAction::duration

# Specifying Time Values

- Time values can be represented by a special stereotype of Value («RTtimeValue») in different formats; e.g.
  - 12:04 (time of day)
  - 5.3, 'ms' (time interval)
  - 2000/10/27 (date)
  - Wed (day of week)
  - \$param, 'ms' (parameterized value)
  - 'poisson', 5.4, 'sec' (time value with a Poisson distribution)
  - 'histogram' 0, 0.3 1, 0.4 2, 0.3, 3, 'ms'



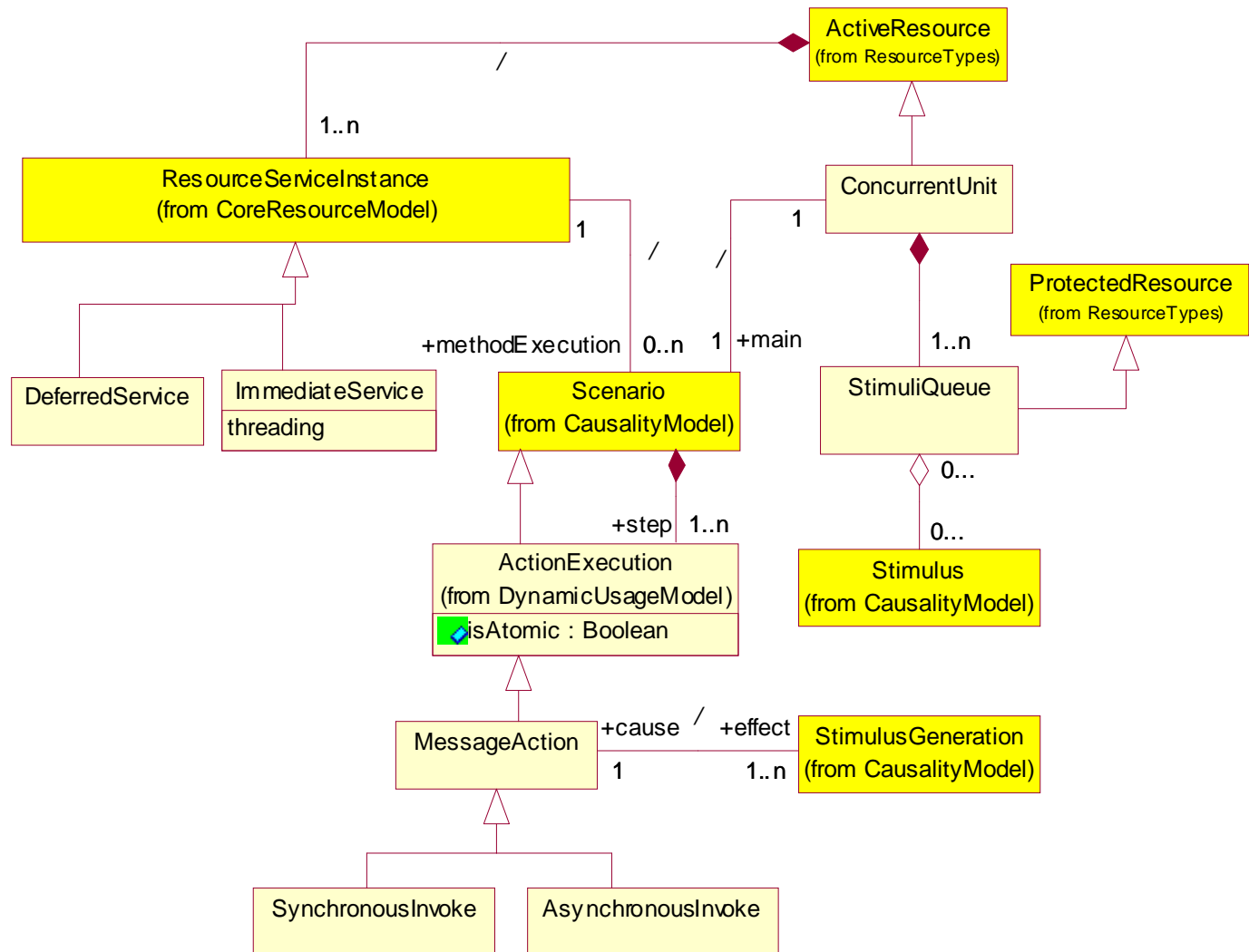
# Specifying Arrival Patterns

- Method for specifying standard arrival pattern values
  - **Bounded:** *'bounded', <min-interval>, <max-interval>*
  - **Bursty:** *'bursty', <burst-interval> <max.no.events>*
  - **Irregular:** *'irregular', <interarrival-time>, [*<interarrival-time>*]\**
  - **Periodic:** *'periodic', <period> [, <max-deviation>]*
  - **Unbounded:** *'unbounded', <probability-distribution>*
- Probability distributions supported:
  - Bernoulli, Binomial, Exponential, Gamma, Geometric, Histogram, Normal, Poisson, Uniform

# Overview

- The Real-Time UML Profile
  - Background
  - Modeling Resources and Quality of Service
  - Modeling Time
  - *Modeling Concurrency*
  - The Schedulability Analysis Sub-Profile
  - The Performance Analysis Sub-Profile
  - The Real-Time CORBA Schedulability Analysis Sub-Profile
  - Real-Time CORBA UML Model
  - Model Processing Cycle
- Summary

# General Concurrency Modeling



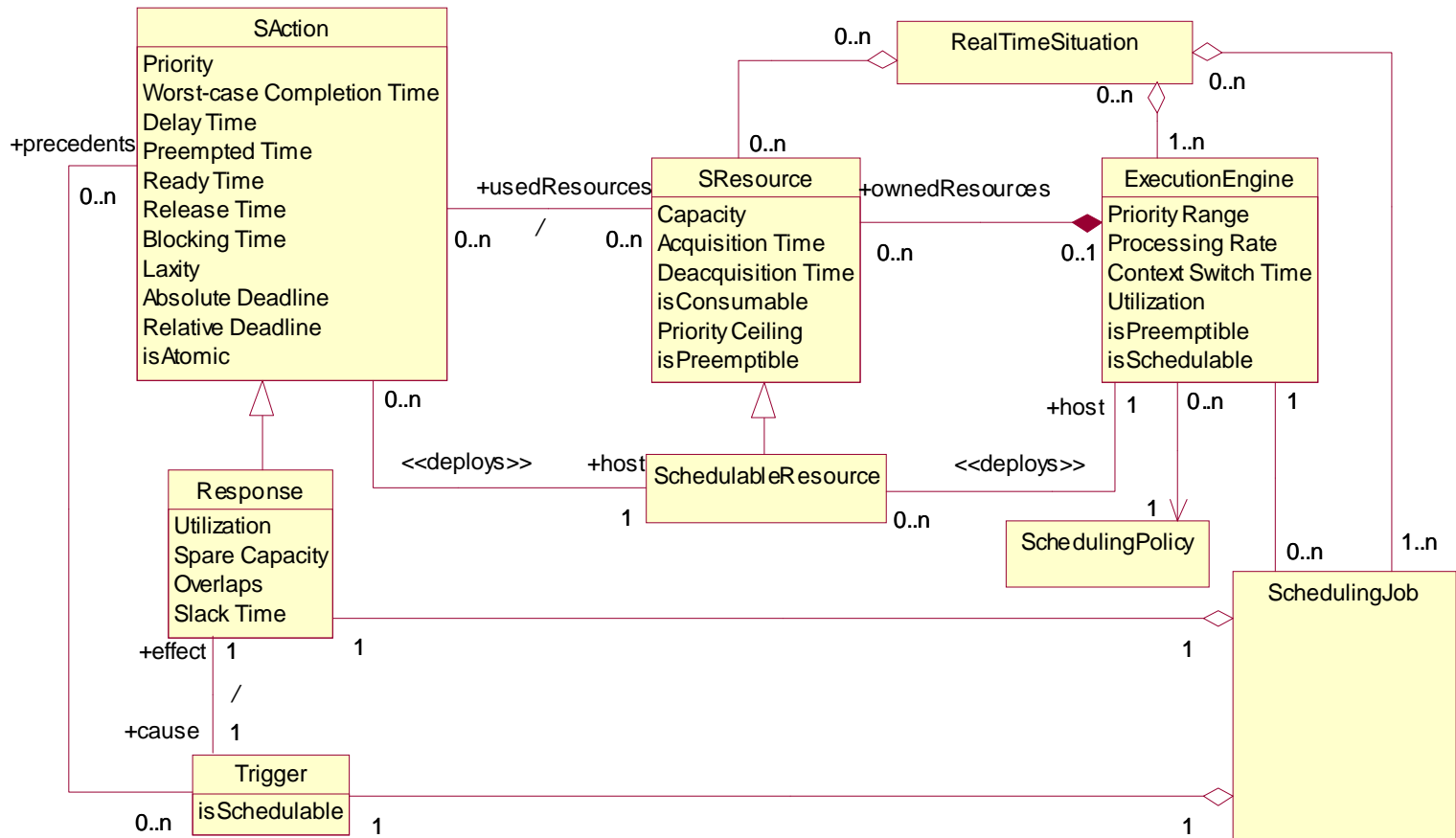
# Defined Stereotypes

Stereotype	Applies To	Tags	Description
«CRAction»	Action, ActionExecution, Stimulus, Action, Message, Method...	CRAtomic [0..1]	An action execution
«CRAsynch»	Action, ActionExecution		An asynchronous invocation
«CRConcurrent»	Node, Component, Artifact, Class, Instance	CRMain [0..1]	A concurrent unit concept
«CRContains»	Usage		A generalized usage dependency
«CRDeferred»	Operation, Reception, Message, Stimulus		A deferred receive
«CRImmediate»	Operation, Reception, Message, Stimulus	{remote, local} [0..1]	An instance of an immediate service
«CRmsgQ»	Instance, Object, Class, ClassifierRole		A stimuli queue
«CRSynch»	Action, ActionExecution		A synchronous invoke

# Overview

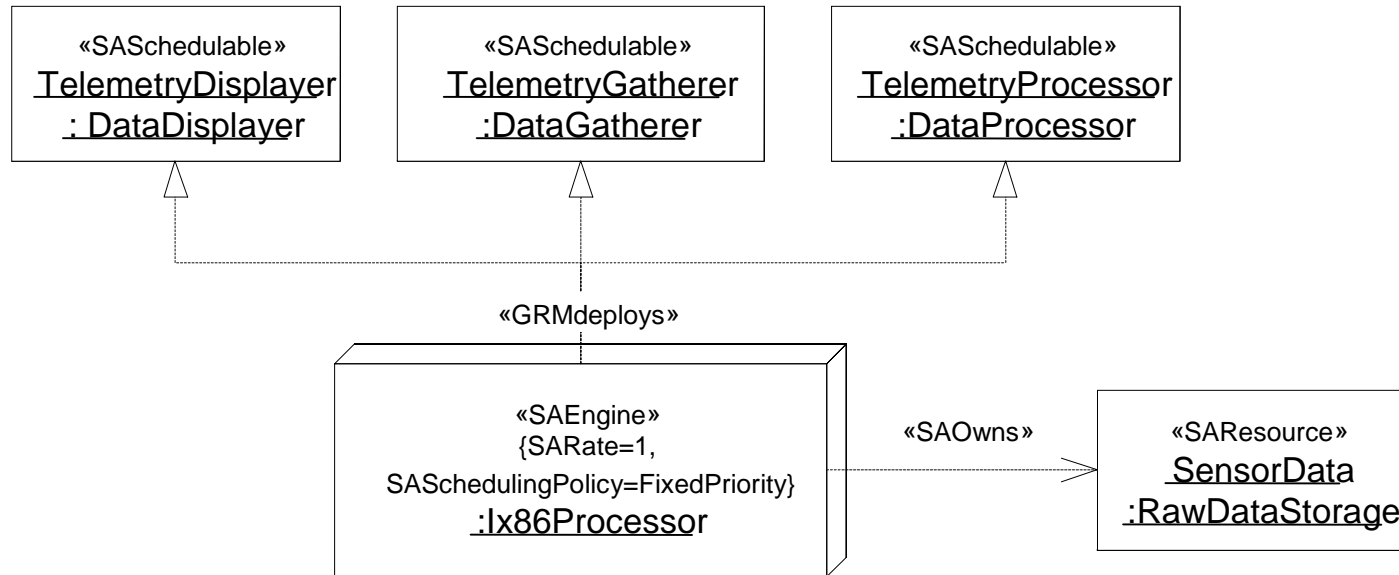
- The Real-Time UML Profile
  - Background
  - Modeling Resources and Quality of Service
  - Modeling Time
  - Modeling Concurrency
  - *The Schedulability Analysis Sub-Profile*
  - The Performance Analysis Sub-Profile
  - The Real-Time CORBA Schedulability Analysis Sub-Profile
  - Real-Time CORBA UML Model
  - Model Processing Cycle
- Summary

# Schedulability Analysis Sub-Profile





# Schedulability Example: Deployment



# Defined Stereotypes (1 of 3)

Stereotype	Applies To	Tags	Description
«SAAction» (subclass of «RTaction» and «CRAction»)	Action, ActionExecution, Stimulus, Action, Message, Method...	SAPriority [0..1] SAActualPty [0..1] SABlocking [0..1] SAReady [0..1] SADelay [0..1] SARelease [0..1] SAPreempted [0..1] SAWorstCase [0..1] SALaxity [0..1] SAPriority [0..1] SAAbsDeadline [0..1] SARelDeadline [0..1] SAusedResource [0..1] SAhost [0..1]	An action
«SAEngine»	Node, Instance, Object, Classifier, ClassifierRole	SASchedulingPolicy [0..1] SAAccessPolicy [0..1] SARate [0..1] SAContextSwitch [0..1] SAPriorityRange [0..1] SAPreemptible [0..1] SAUtilization [0..1] SASchedulable [0..1] Saresources [0..1]	An execution engine

# Defined Stereotypes (2 of 3)

Stereotype	Applies To	Tags	Description
«SAOwns» (subclass of «GRMrealize»)	Abstraction		Identifies ownership of resources
«SAPrecedes»	Usage		A precedence relationship between actions and triggers
«SAResource»	Classifier, ClassifierRole, Instance, Object, Node	SAAccessControl [0..1] SAConsumable [0..1] SACapacity [0..1] SAAcquisition [0..1] SADeacquisition [0..1] SAPtyCeiling [0..1] SAPreemptible [0..1]	A resource of some kind
«SAResponse» (subclass of «SAAction»)	Action, ActionExecution, Stimulus, Action, Message, Method...	SAUtilization [0..1] SASpare [0..1] SASlack [0..1] SAOverlaps [0..1]	A response to a stimulus or action
«SASchedulable» (subclass of «SAResource»)	Classifier, ClassifierRole, Instance, Object, Node		A schedulable resource

# Defined Stereotypes (3 of 3)

Stereotype	Applies To	Tags	Description
«SAScheduler»	Classifier, ClassifierRole, Instance, Object	SASchedulingPolicy [0..1] SAExecutionEngine [0..1]	A scheduler
«SAPrecedes»	Usage		A precedence relationship between actions and triggers
«SASituation»	Collaboration, CollaborationInstance, ActivityGraph		A schedulability analysis context
«SATrigger» (subclass of «SAAction»)	Message, Stimulus	SASchedulable [0..1] SASAPrecedents [0..1]	A trigger
«SAusedHost»	Usage		Identifies schedulable resources used for execution of actions
«SAUses»	Usage		Identifies sharable resources

# Policies Supported

## ■ Scheduling Policies:

- Rate Monotonic, Deadline Monotonic, HKL, Fixed Priority, Minimum Laxity First, Maximize Accrued Utility, Minimum Slack Time
- ...may be extended in the future

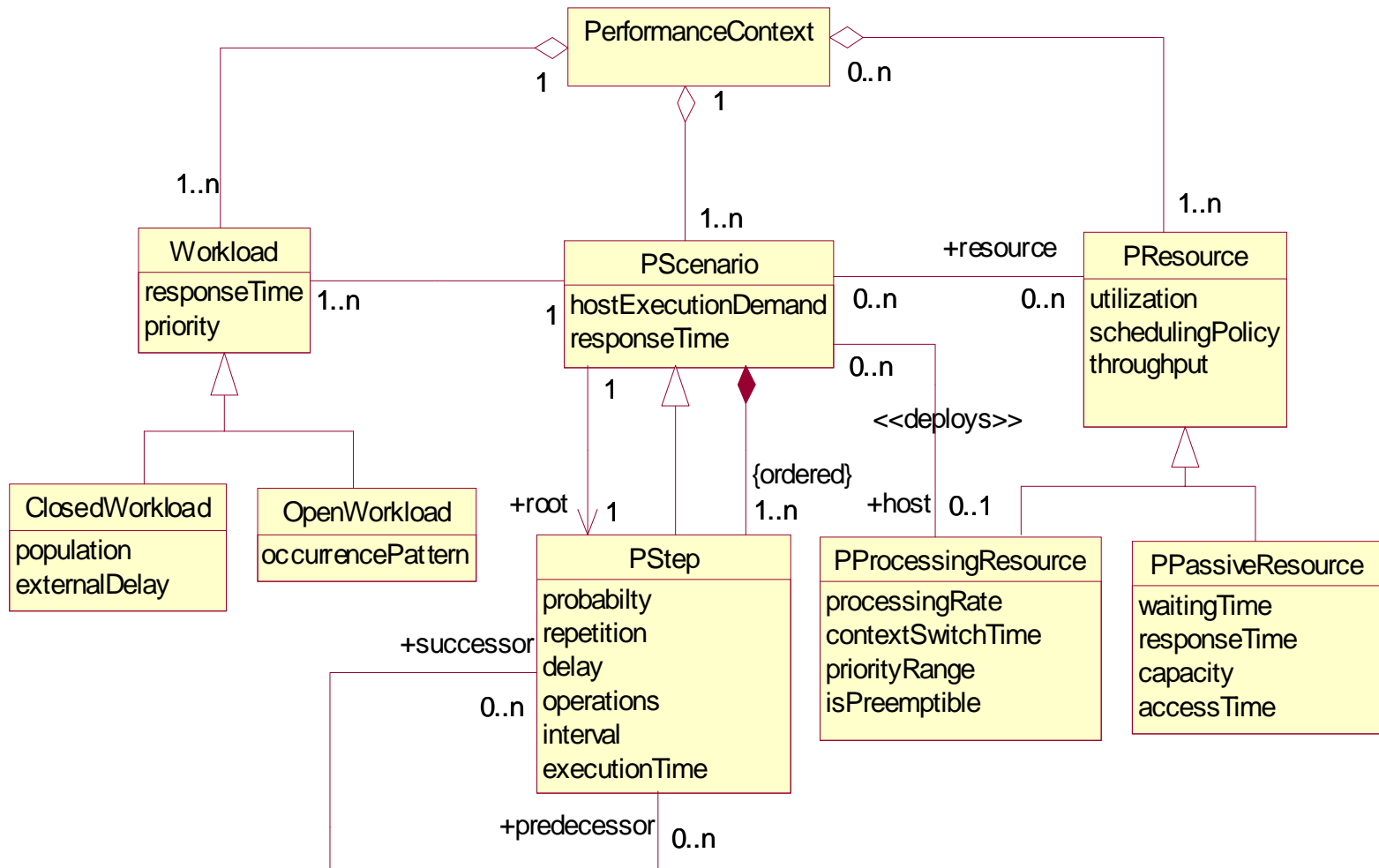
## ■ Access Control Policies:

- FIFO, Priority Inheritance, No Preemption, Highest Lockers, Priority Ceiling
- ...may be extended in the future

# Overview

- The Real-Time UML Profile
  - Background
  - Modeling Resources and Quality of Service
  - Modeling Time
  - Modeling Concurrency
  - The Schedulability Analysis Sub-Profile
  - *The Performance Analysis Sub-Profile*
  - The Real-Time CORBA Schedulability Analysis Sub-Profile
  - Real-Time CORBA UML Model
  - Model Processing Cycle
- Summary

# Performance Analysis Concepts



# Defined Stereotypes (1 of 2)

Stereotype	Applies To	Tags	Description
«PAClosedLoad»	Action, ActionExecution, Stimulus, Action, Message, Method...	PArespTime [0..*] PApriority [0..1] PApopulation [0..1] PAextDelay [0..1]	A closed workload
«PAcontext»	Collaboration, CollaborationInstanceSet, ActivityGraph		A performance analysis context
«PAhost»	Classifier, Node, ClassifierRole, Instance, Partition	PAutilization [0..*] PASchdPolicy [0..1] PARate [0..1] PActxtSwT [0..1] PAprioRange [0..1] PApreemptible [0..1] PAthroughput [0..1]	A deferred receive
«PAOpenLoad»	Action, ActionExecution, Stimulus, Action, Message, Method...	PArespTime [0..*] PApriority [0..1] PAoccurrence [0..1]	An open workload

# Defined Stereotypes (1 of 2)

Stereotype	Applies To	Tags	Description
«PAresource»	Classifier, Node, ClassifierRole, Instance, Partition	PAutilization [0..*] PASchedPolicy [0..1] PACapacity [0..1] PAMaxTime [0..1] PArespTime [0..1] PAwaitTime [0..1] PAtthroughput [0..1]	A passive resource
«PAstep»	Message, ActionState, Stimulus, SubactivityState	PAdemand [0..1] PArespTime [0..1] PAprob [0..1] PArep [0..1] PAdelay [0..1] PAextOp [0..1] PAinterval [0..1]	A step in a scenario

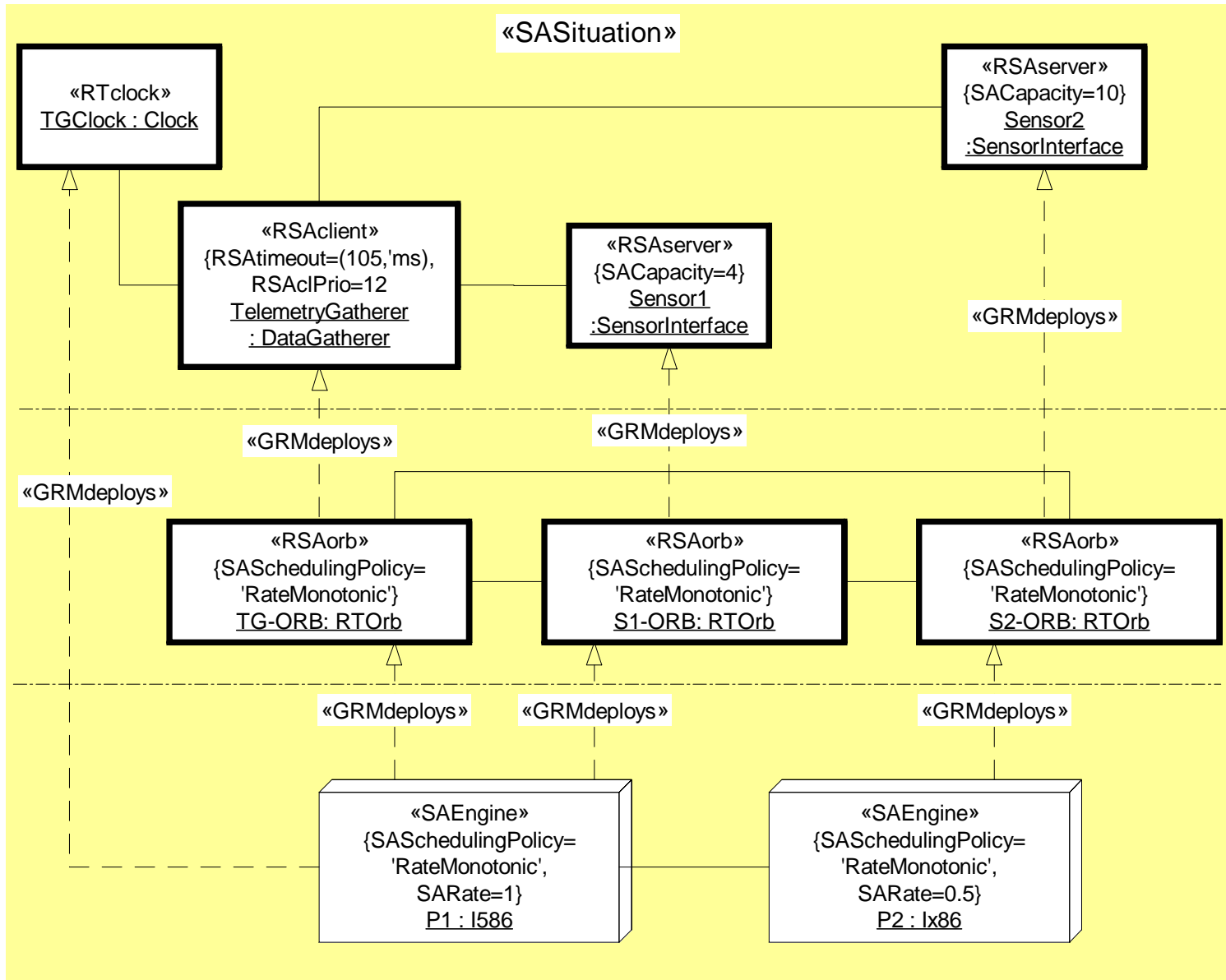


# Overview

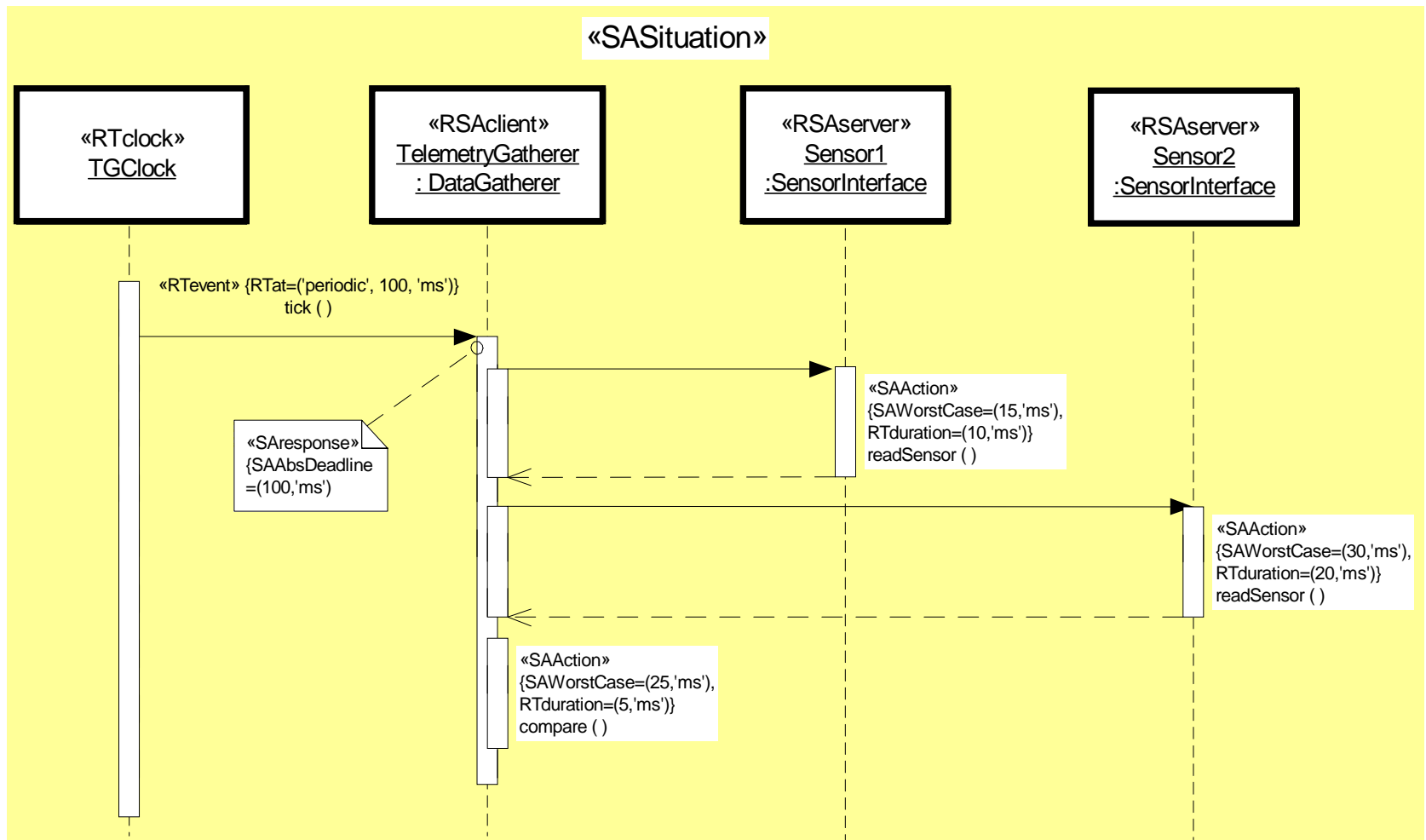
- The Real-Time UML Profile
  - Background
  - Modeling Resources and Quality of Service
  - Modeling Time
  - Modeling Concurrency
  - The Schedulability Analysis Sub-Profile
  - The Performance Analysis Sub-Profile
  - *The Real-Time CORBA Schedulability Analysis Sub-Profile*
  - Real-Time CORBA UML Model
  - Model Processing Cycle
- Summary



# Example: RT CORBA



# Example: RT CORBA Usage Scenario



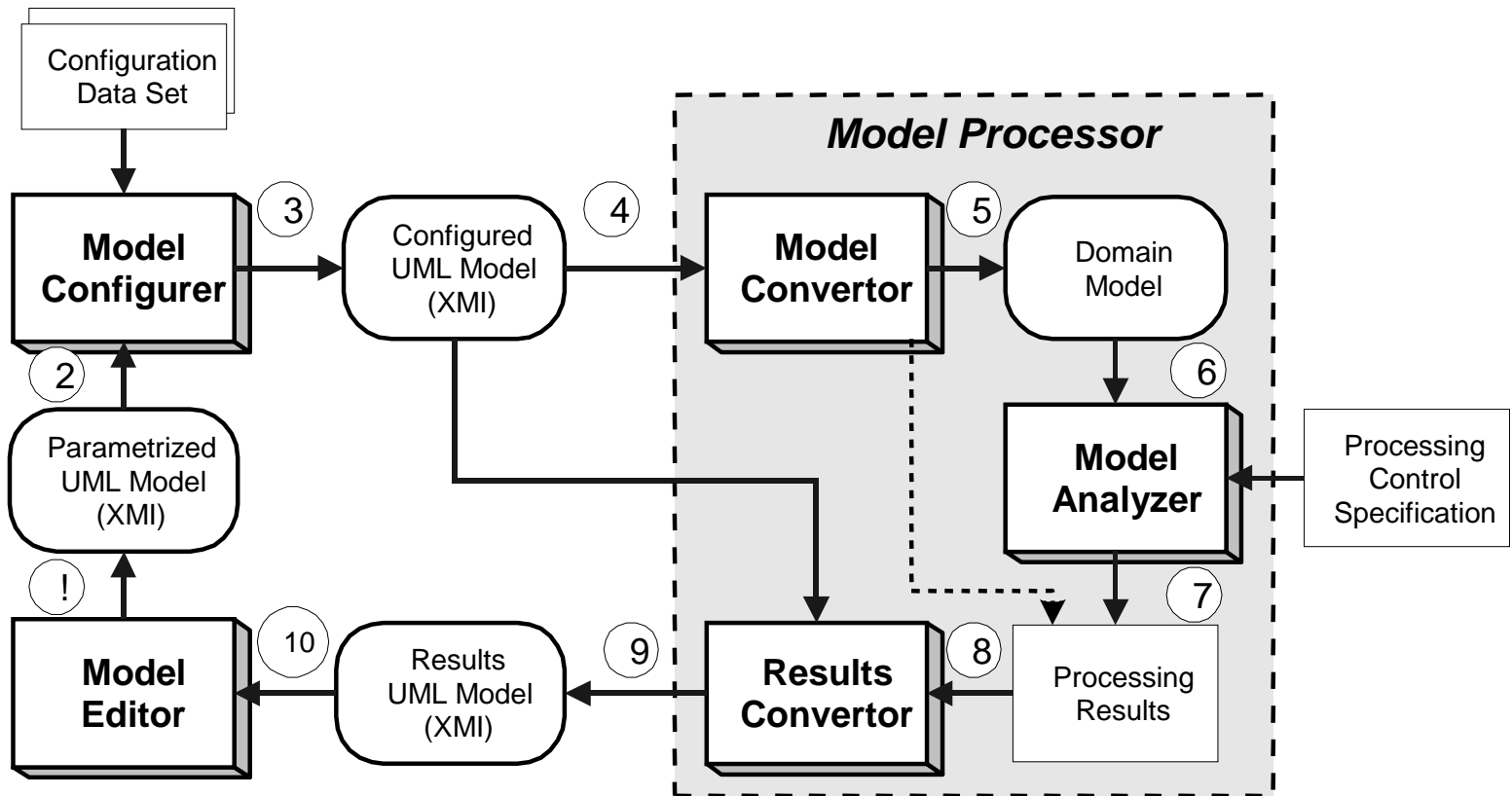
# Defined Stereotypes

Stereotype	Applies To	Tags	Description
«RSAclient» (subclass of «SASchedulable»)	Classifier, ClassifierRole, Instance, Object, Node	RSAtimeout [0..1] RSAclPrio [0..1] RSAprivate [0..1] RSAhost [0..1]	An RT CORBA client
«RSAconnection» (subclass of «SASchedulable» and «SAResource»)	Classifier, ClassifierRole, Instance, Object, Node	SAAccessControl [0..1] RSAshared [0..1] RSAhiPrio [0..1] RSAloPrio [0..1] RSAserver [0..1]	An RT CORBA connection
«RSAmutex» (subclass of «SAResource»)	Classifier, ClassifierRole, Instance, Object, Node	SAAccessControl [0..1] RSAhost [0..1]	An RT CORBA mutex
«RSAorb» (subclass of «SAResource» )	Classifier, ClassifierRole, Instance, Object, Node	SAschedulingPolicy [0..1]	An RT CORBA ORB
«RSAserver» (subclass of «SAResource» )	Classifier, ClassifierRole, Instance, Object, Node	RSAsrvPrio [0..1] SACapacity [0..1]	An RT CORBA server

# Overview

- The Real-Time UML Profile
  - Background
  - Modeling Resources and Quality of Service
  - Modeling Time
  - Modeling Concurrency
  - The Schedulability Analysis Sub-Profile
  - The Performance Analysis Sub-Profile
  - The Real-Time CORBA Schedulability Analysis Sub-Profile
  - Real-Time CORBA UML Model
  - *Model Processing Cycle*
- Summary

# Model Processing



# The Tag Value Language

- Tagged value format:  
    {<tag-name> = <tag-value>}
- Used to specify complex (structured) tagged values
- Based on a small proper subset of the freeware Perl language
  - Includes: variables, numbers, booleans, strings, lists, expressions (including conditionals), operators, and functions
- Suitable for:
  - expressing complex dependencies between values
  - writing processing scripts

# Summary (1 of 2)

- Defines a set of extensions for directly expressing real-time domain concepts and their key quantitative characteristics in UML:
  - resources
  - concurrency mechanisms
  - time and timing mechanisms
- Models constructed using the real-time UML standard can be exchanged with specialized analysis tools
  - automated model validation
  - eliminates need for expensive and rare expertise
  - predictive engineering models

# Summary (2 of 2)

- UML provides a common and standardized underpinning that supports all the components of our solution
  - for object-oriented modeling
  - for predictive QoS modeling (via the real-time profile)
  - for design analysis and synthesis (tool interchange)
  - for architectural definition
  - for implementation (through full automatic code generation)
- Furthermore, as a standard, it enables model interchange between specialized tools and is a basis for significant automation of the RT software development process