

# Consistency of UML/SPT Models

---

Abdelouahed Gherbi and Ferhat Khendek  
Concordia University  
Montreal, Canada

# Outline

---

- Introduction
- Issues
- Background
- Framework for UML/SPT models consistency
- Concurrency-related Consistency
- Time Consistency
- Conclusions

# Introduction

---

- ❑ Embedded real-time systems (ERTS) are used in a large set of applications
- ❑ ERTS functionality is increasingly software based
- ❑ It is also increasingly complex
- ❑ Low-level optimization techniques of code are no more enough

# Introduction

---

- Modeling is a fundamental engineering activity
  
- Models of high-level of abstraction
  - Increase the visibility and control over systems complexity.
  - Help in understanding the problem
  - Communicate
  - Reason about the model
  - Automate (verification, implementation synthesis)
  
- Model-driven engineering approach:
  - MDA
  - UML, MOF, QVT, XMI, etc.

# Introduction

---

- ❑ UML: *de facto* Standard software modeling language.
- ❑ UML is intuitive because it is a visual language.
- ❑ UML adopt a multi-view modeling approach
  - Provides many modeling elements and diagrams
- ❑ UML is adaptable and customizable
  - Profiles or domain specific modeling languages
- ❑ However, UML faces the consistency issue
  - Lack of formal semantics
  - Multi-view approach

# Introduction

---

- UML can be used to model real-time systems
  
- UML profiles for embedded and RT systems
  - UML/SPT
  - MARTE
  
- Real-time software systems have in addition specific characteristics
  - Time constraints
  - Concurrency
  
- Consequently, the consistency issue is more complex in UML models of ERTS

# Introduction

---

- In this paper
  - Focus on a general definition of the consistency issue in UML/SPT models.
  
- Contributions
  - A framework addressing incrementally the consistency issue of UML/SPT models
  - Focus on the time consistency using an approach based on schedulability analysis

# Background

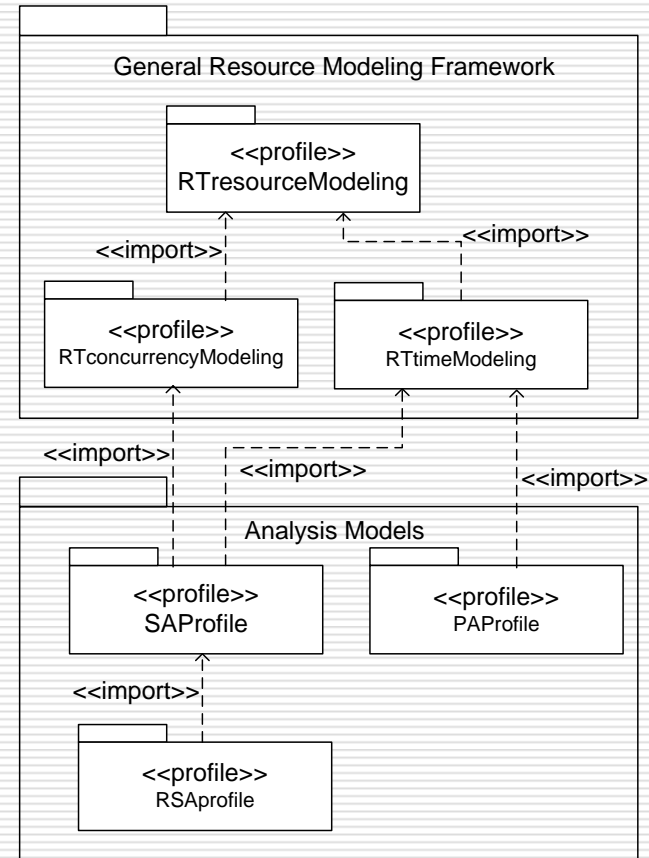
---

- ❑ UML/SPT: UML profile for Schedulability, Performance and Time.
- ❑ OMG current standard and it is being updated (MARTE)
- ❑ The objective: Construction of *predictive UML models* – Introduce quantitative information in the models and predict key properties (timing) early before any costly implementation.
- ❑ UML/SPT provides:
  - ❑ A set of domain models encapsulating the concepts resource and quality of service; time; concurrency; performance; and schedulability modeling.
  - ❑ Is mapped to UML through a set of stereotypes to annotate its UML models.



# Background

- General resource modeling framework
  - Resource and QoS
  - Concurrency
  - Time and time-related mechanisms
- Analysis modeling
  - Schedulability analysis
  - Performance analysis



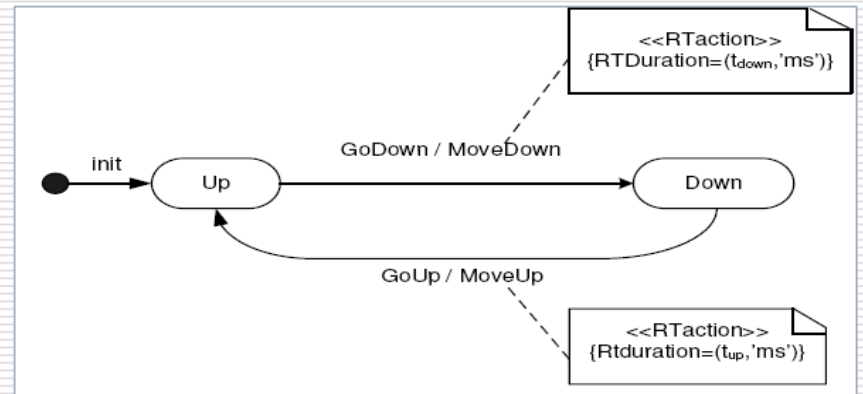
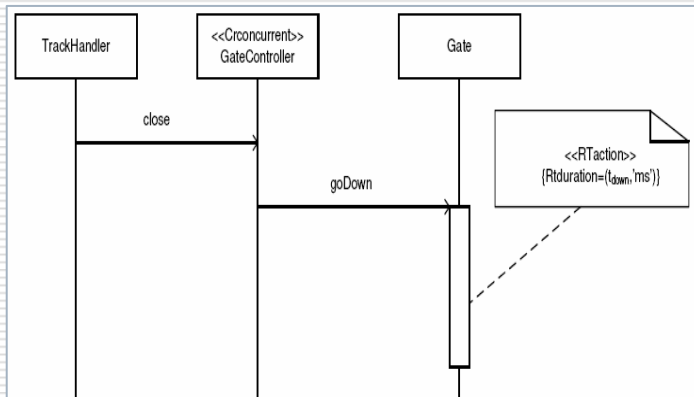
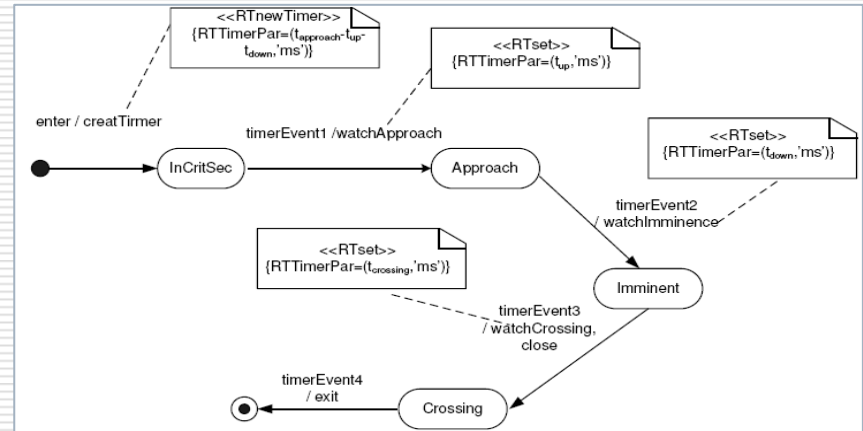
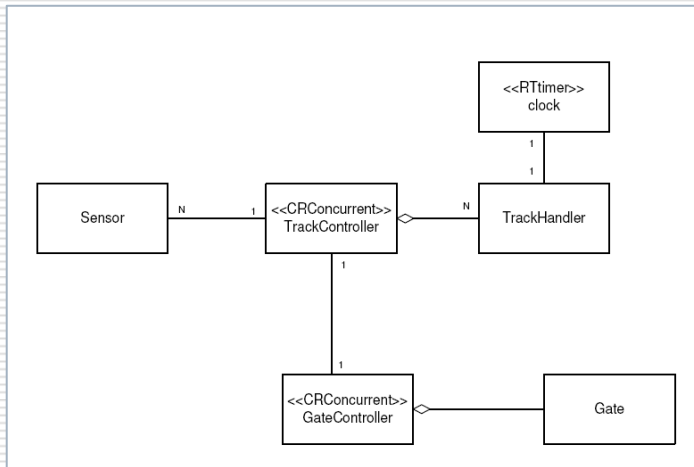
# Background

---

## □ Sample of UML/SPT Stereotypes

Streotype	Applies to	Tagged values
<<RTAction>>	Action, Actionexecution, Message, Method ...	RTstart, RTend RTduration
<<RTEvent>>	Action, Actionexecution, Stimulus, Message ...	RTat
<<RTtimer>>	Instance, Object	RTduration RTperiodic
<<CRsynch>>	Action, ActionExecution	
<<SAAction>>	Action, Actionexecution, Stimulus, Message, Method ...	SAPriority SAWorstCase SAAbsDeadline...
<<SAEngine>>	Node, Instance, Object...	SAschedulingPolicy SAContextSwitch SAPriorityRange
<<SAResource>>	Node, Instance, Object...	SAptyCeiling SApreemptible

# Example



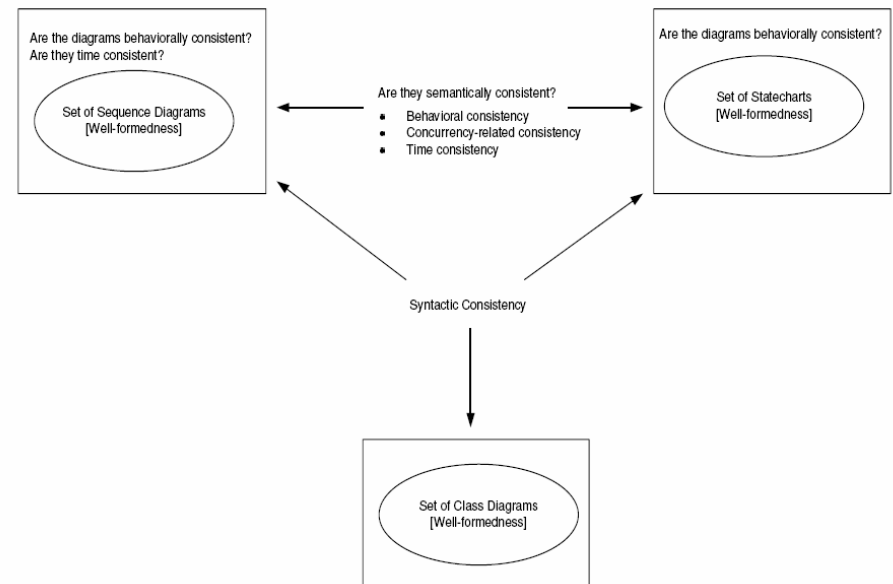
## Framework for UML/SPT models consistency

---

- UML/SPT model is:
  - A UML model → several UML diagrams
  - Capture RT relevant features (Time constraints, concurrency) using stereotypes cross-cutting the different UML diagrams.
- What is a consistent UML/SPT model?
  - No straightforward definition.
  - Incremental approach is appropriate.

# Framework for UML/SPT consistency

- One way to consider this question is to an incremental approach
- Syntactic level
  - Static property
  - Intra diagram:
    - Well-formedness rules in OCL
  - Inter-diagram
- Semantic Level
  - Dynamic property
  - Behavioral consistency: Inter-diagram consistency used for behavior modeling (sequence diagrams, statecharts)
  - Concurrency-related consistency
  - Time Consistency
    - Logical time consistency
    - System time consistency

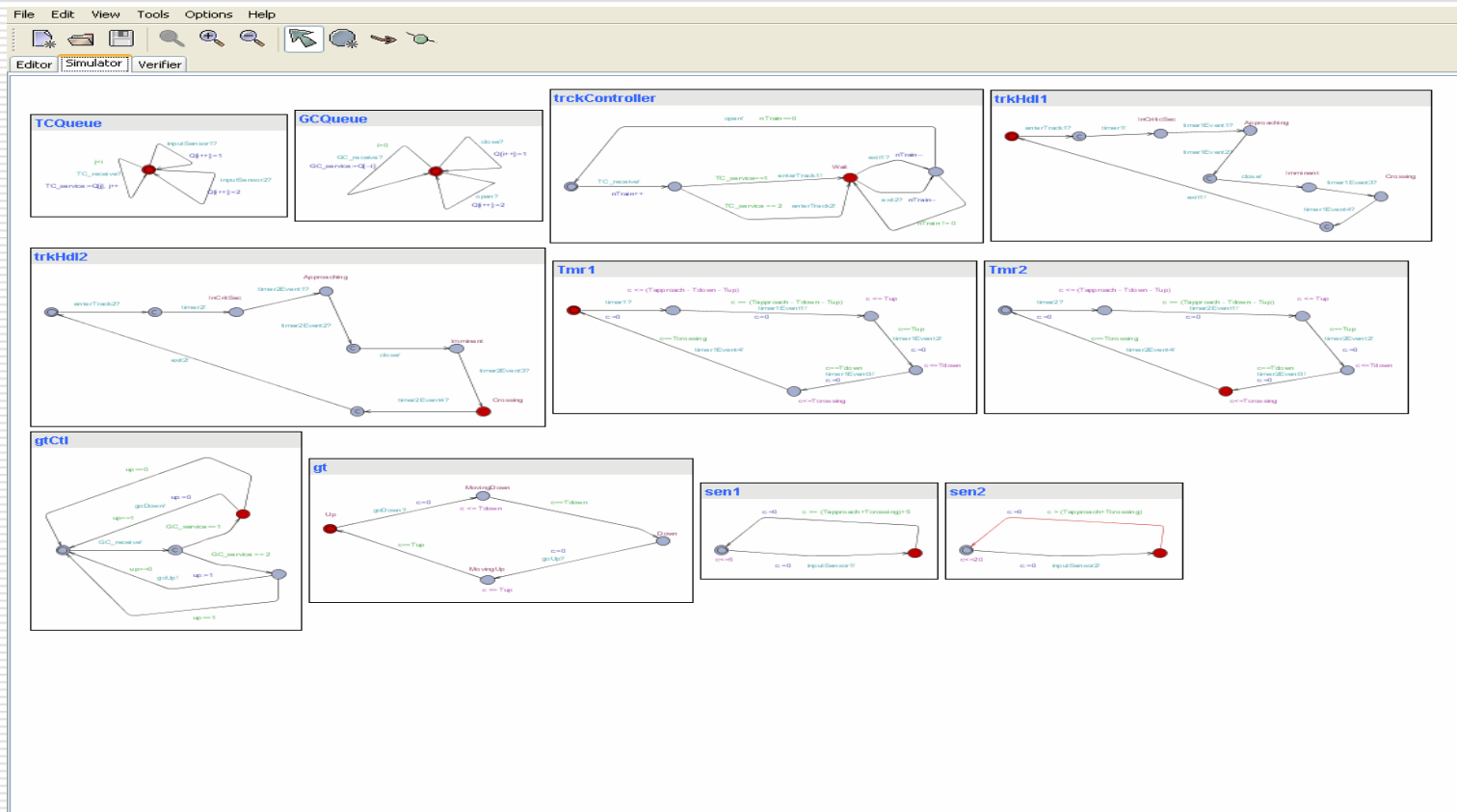


# Concurrency-related consistency

---

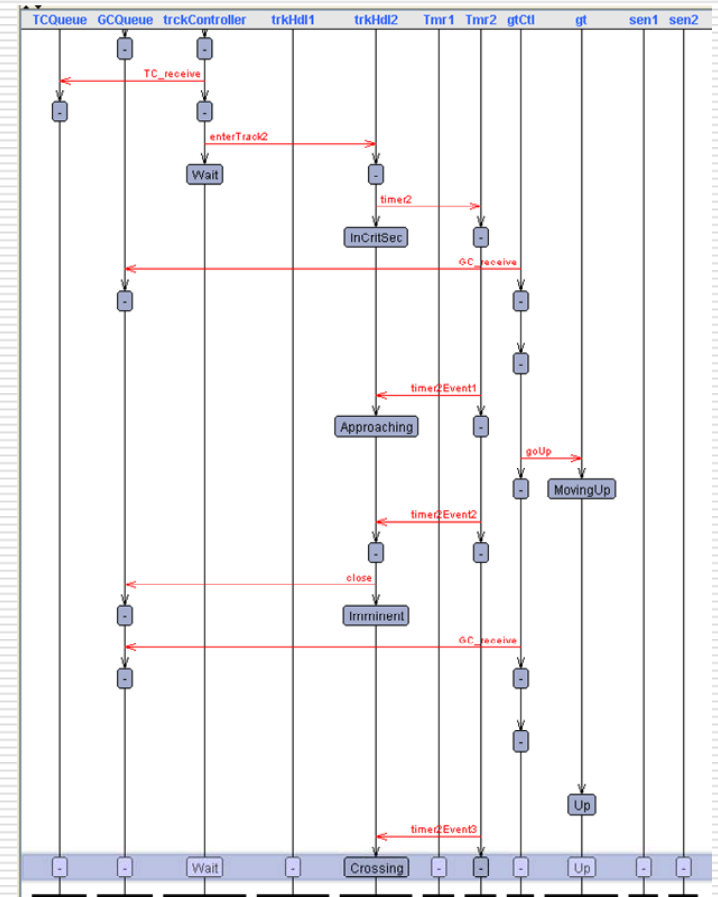
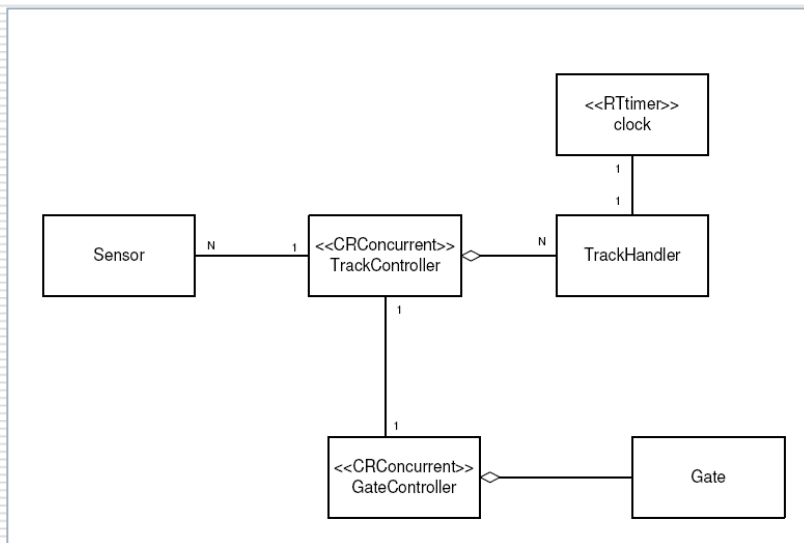
- ❑ Focus on the concurrency choices expressed using UML/SPT stereotypes
- ❑ Design choices in terms of concurrency allow for an efficient resources use and to meet the time constraints.
- ❑ But, may lead to issues (e.g., deadlock and race conditions)
- ❑ UML/SPT
  - defines a concurrency domain model
  - provides a set of stereotypes to use on a UML model
- ❑ Timed automata semantics for this concurrency domain model
- ❑ Model checking techniques can then be used to check a UML/SPT model and detect concurrency related issues.

# Concurrency-related consistency



# Concurrency-related consistency

$\exists \diamond (\text{TrkHdl1.Crossing or TrkHdl1.Crossing}) \text{ and } \text{gt.Up}$





# Time Consistency

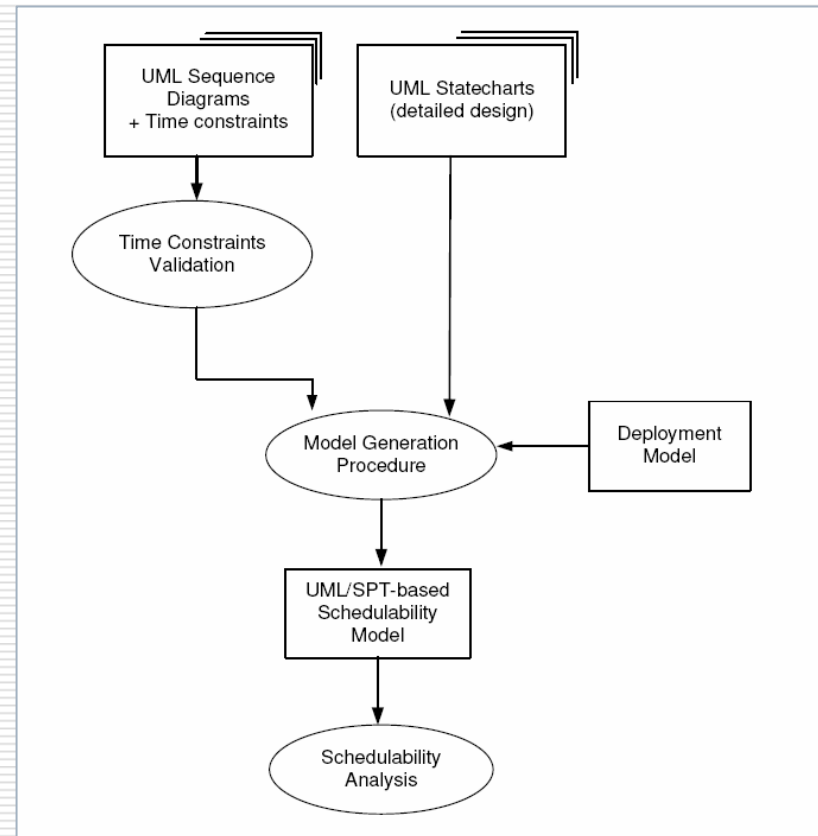
---

- Focus on the time constraints expressed with UML/SPT time stereotypes.
  
- Two particular distinctions:
  - Logical time consistency of sequence diagrams
  - System time consistency (sequence diagrams, statecharts and deployment constraints)

# UML/SPT time consistency

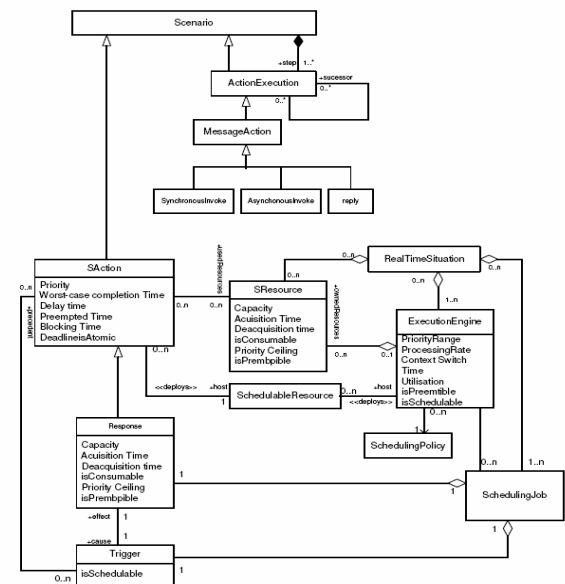
## □ Approach:

- Use schedulability analysis to check time consistency of statecharts with sequence diagrams
- Generation an UML/SPT model for schedulability analysis from:
  - A set of sequence diagrams (each one is assumed to model a time constraints on an end-2-end system transaction )
  - A set of statechart: The detailed design that should satisfy the time constraints considering the deployment model
  - A deployment model: Information on the CPU characteristics, threads, priorities, WCET, etc.



# UML/SPT time consistency

- ❑ UML/SPT SA model generation
- ❑ UML/SPT can be used to support schedulability analysis.
- ❑ This is achieved using the *SAProfile* package.
- ❑ This define a SA domain model and a set of stereotype



# UML/SPT time consistency

## □ UML/SPT SA model generation procedure

- Input:  $SeqD \langle O, E, V, Label \rangle$  be a sequence diagram
- Step 1:  $SC = \{o_i.sc \mid \forall o_i \in O\}$  be a set of associated statecharts

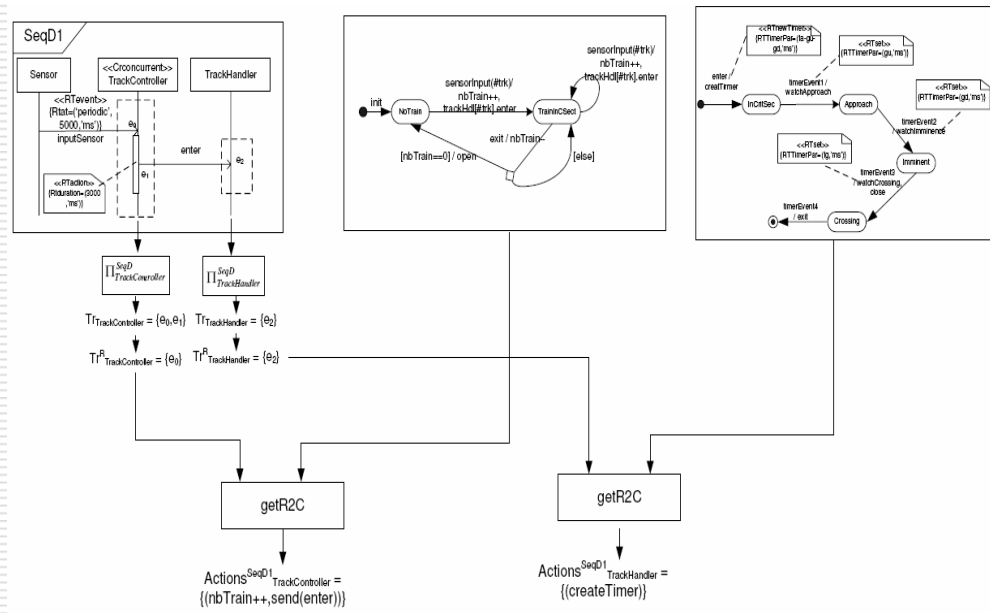
```

for all  $o_i \in O$  do
  Step 1.1: Event partition
    let  $tr_{o_i} \leftarrow \Pi_{o_i}^{SeqD} = \{e_{o_i1}, e_{o_i2}, \dots, e_{o_im}\}$ 
  Step 1.2: Event restriction to receptions
    let  $tr_{o_i}^R \leftarrow tr_{o_i} \cap R = \{e_{o_i1}^r, e_{o_i2}^r, \dots, e_{o_ik}^r\}$ 
  Step 1.3: Run to completion steps
    let  $Action_{o_i} \leftarrow \cup_{j \leq k} \{getR2C(o_i.sc, e_{o_ij}^r)\}$ 
end for
  
```

- Step 2:  $\xi = \{(a_j, a_k) \mid a_j, a_k \in Actions \wedge \exists o_i \in O \wedge \exists e_{o_ij}^r, e_{o_ik}^r \in tr_{o_i}^R \wedge e_{o_ij}^r \preceq e_{o_ik}^r\}$
- Step 3:  $\zeta = \{(a_i, a_j) \mid a_i, a_j \in Actions \wedge \exists e, e' \in E \wedge (e, e') \in V \wedge gen(a_i, e) \wedge trigger(e', a_j)\}$
- Step 4: for all  $a_i \in Actions$  do  
     let  $(a_i.wcet, a_i.priority, a_i.thread, \dots) \leftarrow deploys(a_i)$   
 end for

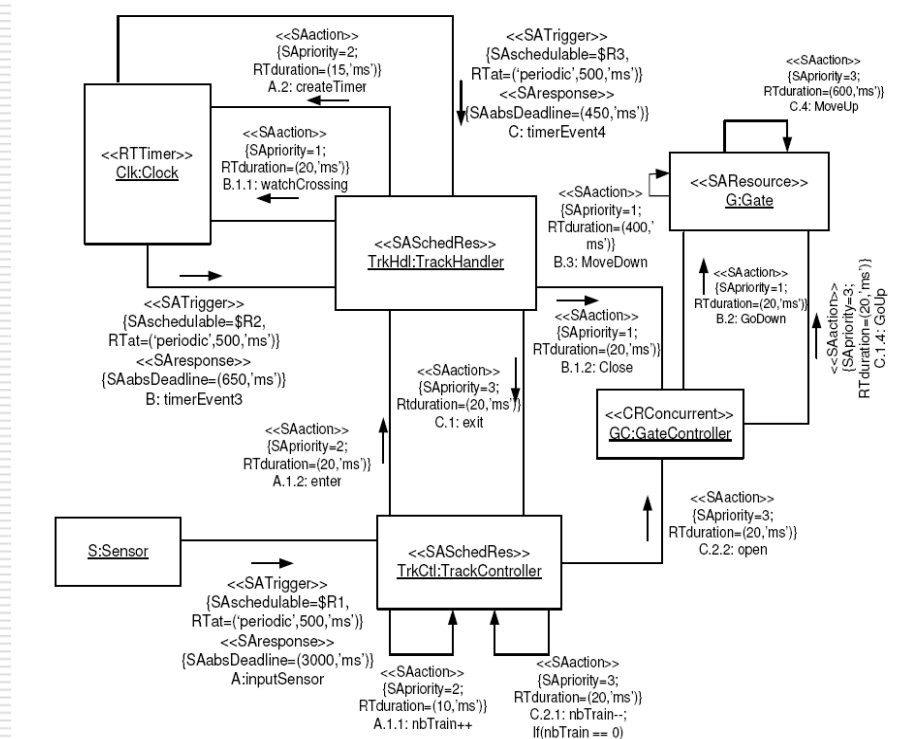
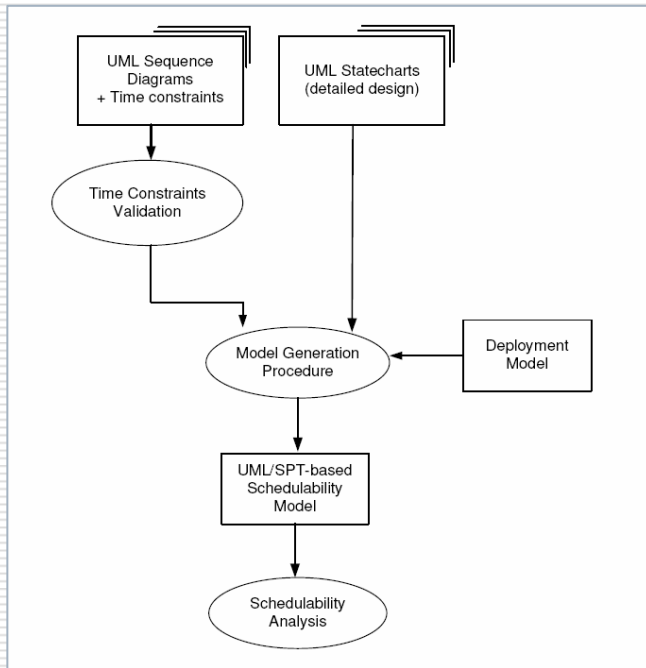
# UML/SPT time consistency

## □ Example



# UML/SPT time consistency

## Generated UML/SPT SA model



# Conclusion

---

- ❑ An UML/SPT model is a UML model enriched with RT aspects such as time constraints and concurrency
- ❑ UML/SPT model consistency is challenging
- ❑ A straightforward definition is difficult
- ❑ An incremental approach is more appropriate
- ❑ Schedulability analysis of UML/SPT model can be used to check time consistency of statecharts with sequence diagrams

---

Thank you!