

# The WOODDES<sup>1</sup> Project: Building Better Embedded Systems

M. Oliver Möller

BRICS<sup>2</sup> Århus

PhD student

omoeller@brics.dk

---

<sup>1</sup>Workshop for Object-Oriented Design and Development of Embedded Systems

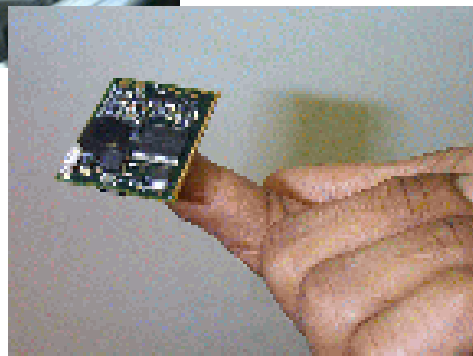
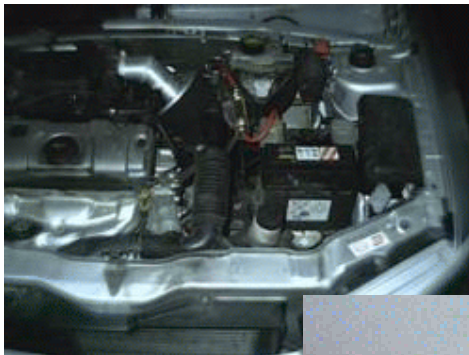
<sup>2</sup>Basic Research In Computer Science

# Embedded Systems



## Embedded:

mixture of hard- and software;  
strong resource limitations;  
interaction with environment

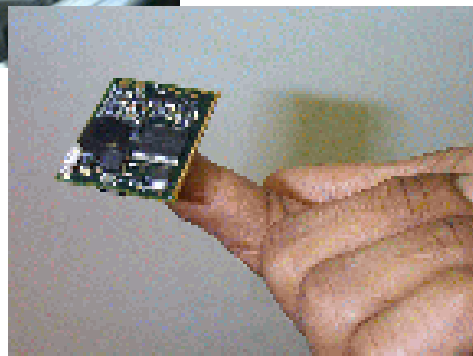
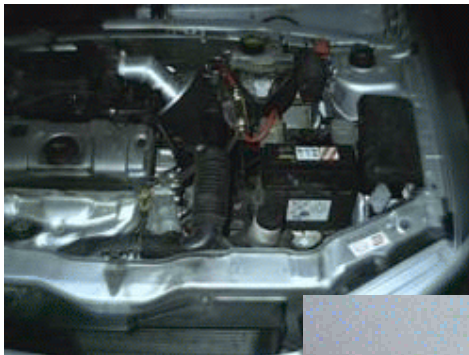


# Embedded Systems



## Embedded:

mixture of hard- and software;  
strong resource limitations;  
interaction with environment



## Real-Time:

Correctness not only dependent  
on the logical order of events,  
but also on their timing

# Challenges in Embedded System Design

---

- **mixture of technologies**
- **market pressure**
- **complex behavior**

# Challenges in Embedded System Design

---

- **mixture of technologies**  
hardware + software + communication protocols
- **market pressure**
- **complex behavior**

# Challenges in Embedded System Design

---

- **mixture of technologies**  
hardware + software + communication protocols
- **market pressure**  
too late = out of business
- **complex behavior**

# Challenges in Embedded System Design

---

- **mixture of technologies**

hardware + software + communication protocols

- **market pressure**

too late = out of business

- **complex behavior**

difficult to test, expensive to fix

# Outline

---

- 1** Introduction to WOODDES  
scope, objectives, partners, status
- 2** The UML-RealTime profile  
context, motivation, and notation
- 3** Methodology for Developing Embedded Systems  
basic tasks and iterations
- 4** Tool Support  
overview, model interchange, small demo
- 5** Expected Outcomes  
lessons to learn, case studies to validate



# AIT-WOODDES: project No. IST-1999-10069

---

Advanced  
Information  
Technology

-

Workshop for  
Object-Oriented  
Design and  
Development of  
Embedded  
Systems

## Objective:

improve *process, methods, and tools*  
for developing embedded systems

## Industrial Gain:

- reduce *cost*
- reduce *time-to-market*
- improve *quality*

## Academic Gain:




- develop notion of *Real-Time Object*
- contribute to *standardization*
- *apply* verification technology

# WOODDES Partners

---

Workshop for Object-Oriented Design and Development of Embedded Systems

**End Users:**

 PSA	<i>automotive</i>
 Mecel	<i>automotive</i>
 Intracom	<i>telecommunication</i>





---

**Tool Providers:**

 I-Logix	<i>Rhapsody</i>
 SOFTEAM	<i>UML Objecteering</i>

---

**Academic:**

 CEA	<i>nuclear energy (safety)</i>
 Offis	<i>formal methods</i>
 Uppsala	<i>real time</i>
 Aalborg	<i>real time</i>

# Scope & Timeline

---

<b>Start (delayed)</b>	November 1999
<b>Duration</b>	3 years
<b>Man-Month</b>	337,5

---

<b>WP0</b>	Project Management (PSA) (coordinated by PSA)	20
<b>WP1</b>	<b>Common Methodology</b>	72
<b>WP2</b>	<b>Tool Interaction Mechanisms</b>	48,5
<b>WP3</b>	<b>Validation of Real-Time Systems</b>	41,5
<b>WP4</b>	<b>Case Studies</b>	84
<b>WP5</b>	<b>Exploitation and Dissemination</b>	28

# Outline

---

- 1** Introduction to WOODDES  
scope, objectives, partners, status
- 2** The UML-RealTime profile  
context, motivation, and notation
- 3** Methodology for Developing Embedded Systems  
basic tasks and iterations
- 4** Tool Support  
overview, model interchange, small demo
- 5** Expected Outcomes  
lessons to learn, case studies to validate

# Unified Modeling Language (UML)

---

Born from unification of other methods (*Booch, OMT, OOSE*)

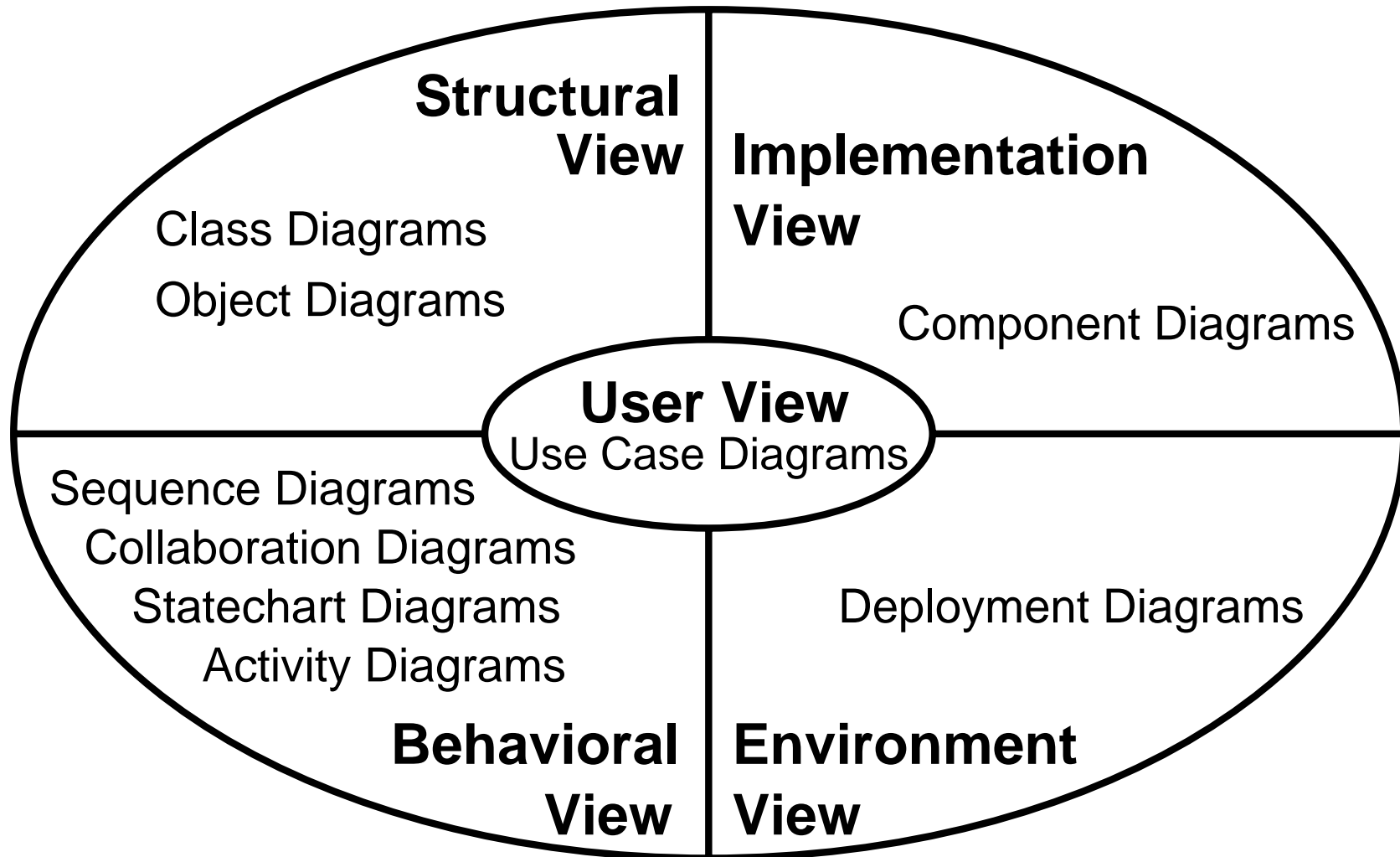
Key Elements:

- Object Orientation
- Visual Formalisms
- Well-Defined Notations

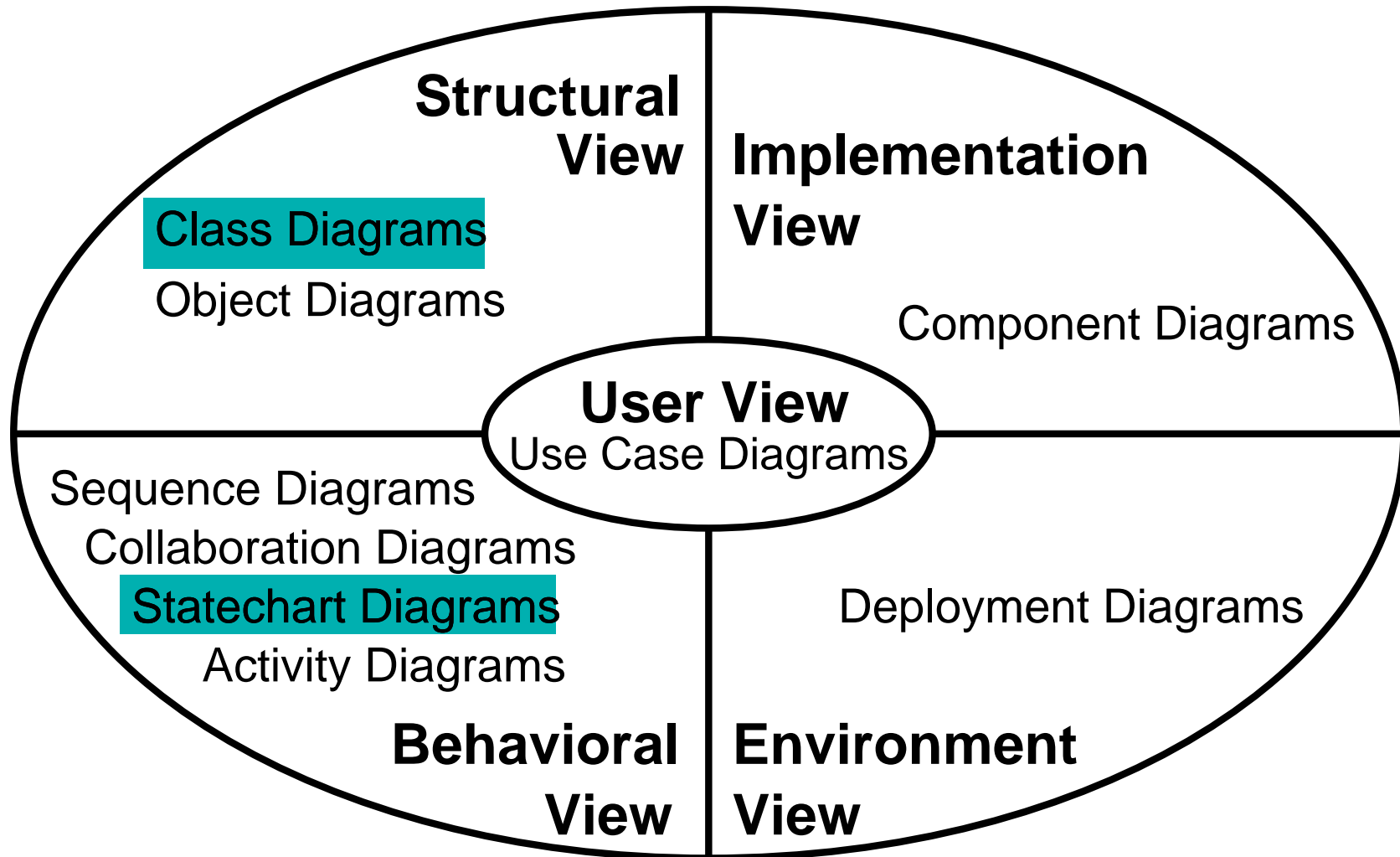
*An evolving standard:*

1.3	finished 2000
1.4	finished 2001
2.0	work in progress (4 RFP issued May/Sept)

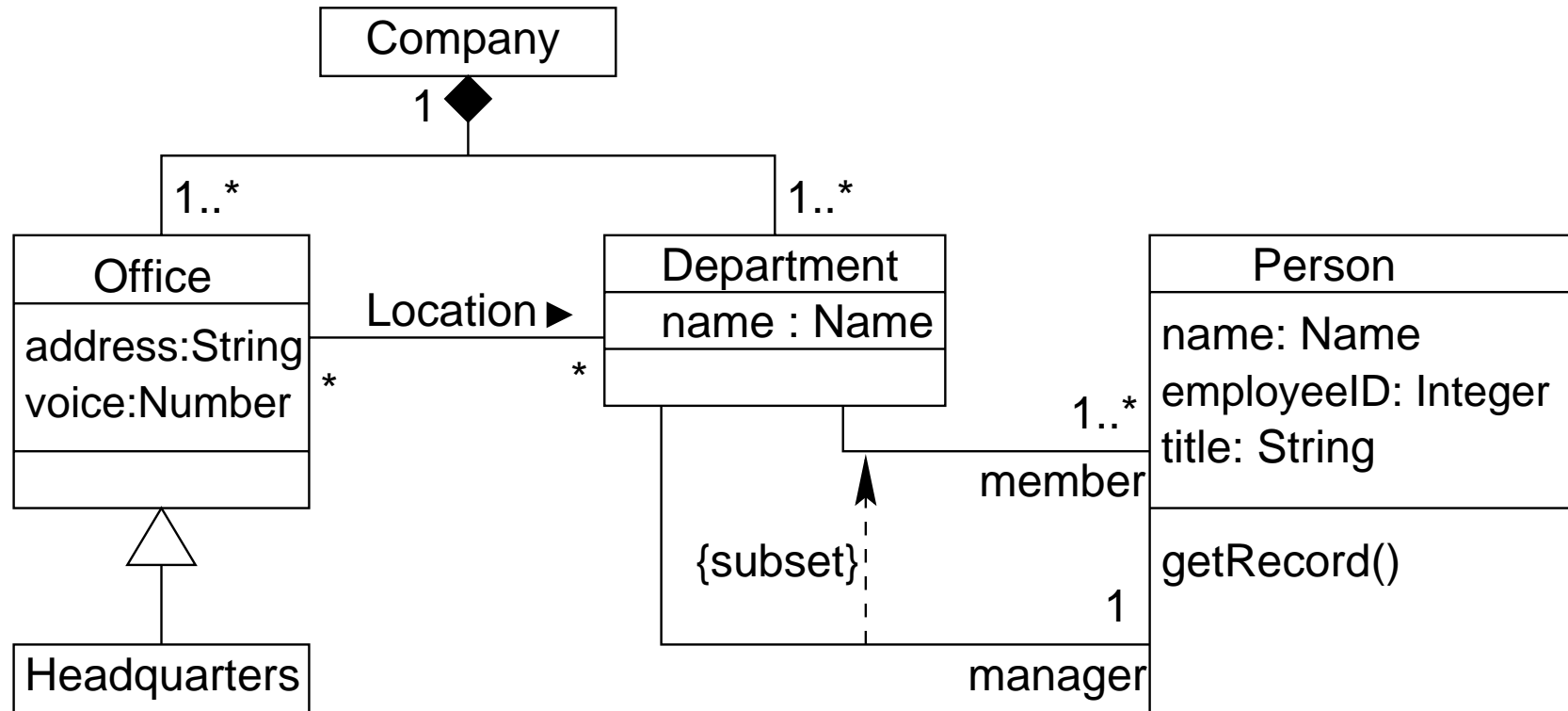
# Organization of the UML



# Organization of the UML



# Class Diagrams



◆ Aggregation

1..\* multiplicity

△ generalization

{subset} constraint

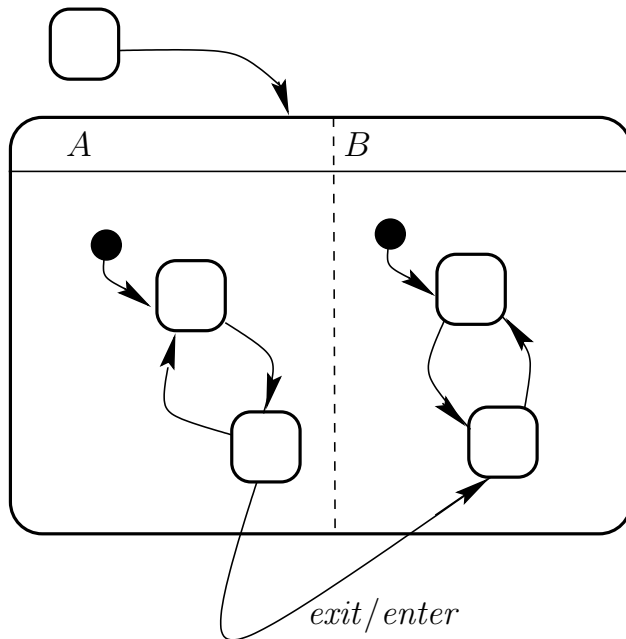
title: String attribute

getRecord() operation



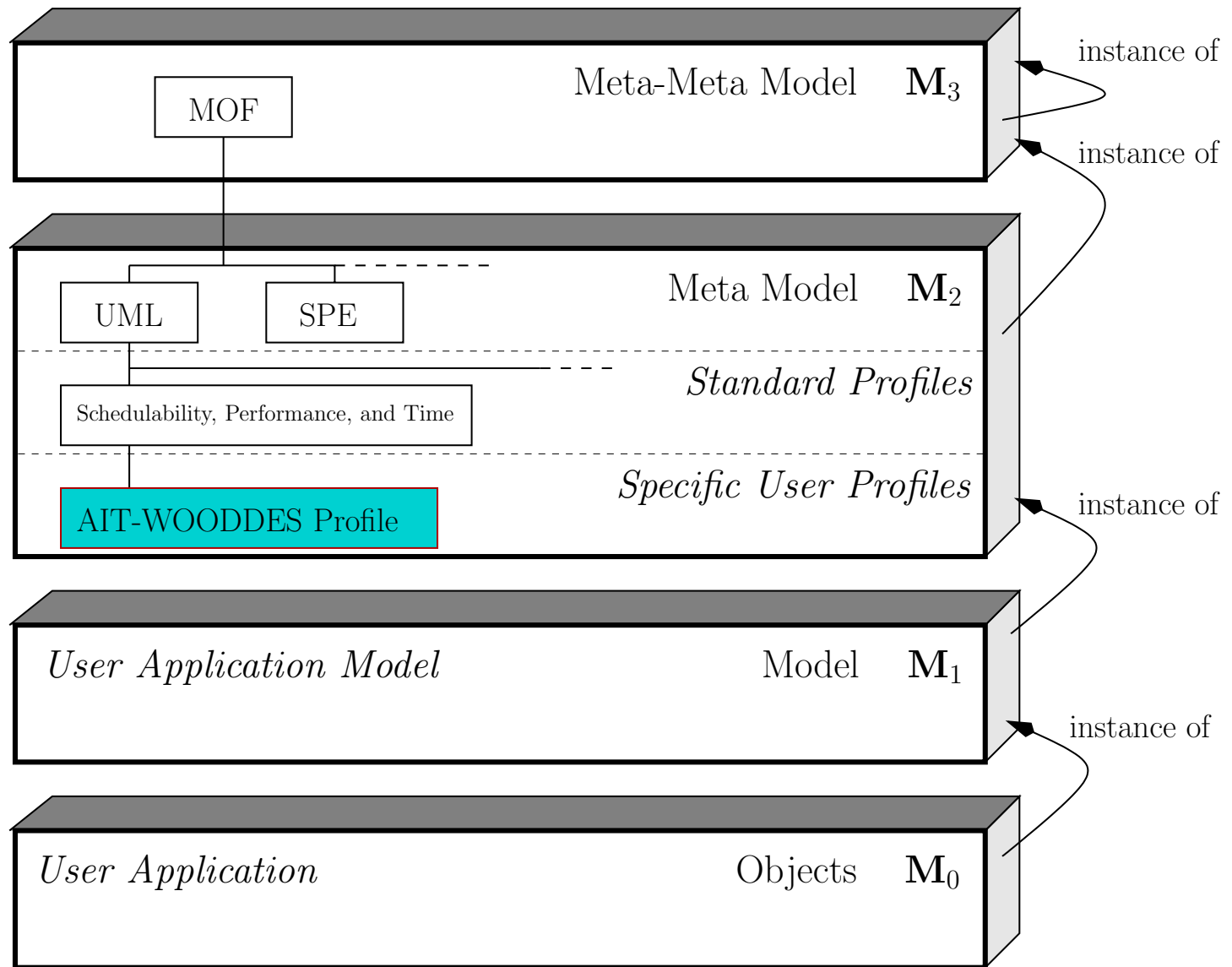
# The Statechart Formalism

## Features



- hierarchical state machines
- parallelism (on any level)
- history
- event communication
- powerful synchronization mechanisms
- inter-level transitions
- actions that are dependent on states
- actions on entry/exit
- ...

# Extensibility: UML Metamodel Architecture



# WOODDES UML-RealTime Profiles

---

**profile:** standard way to extend UML  
*leightweight*

**consists of:** stereotypes  
tagged values     ... & gives meaning to it  
constraints

*Supports the WOODDES methodology:*

- emphasizes crucial modeling concepts
- defines additional modeling elements, that tools should support

# Outline of the WOODDES UML-RealTime profile

---

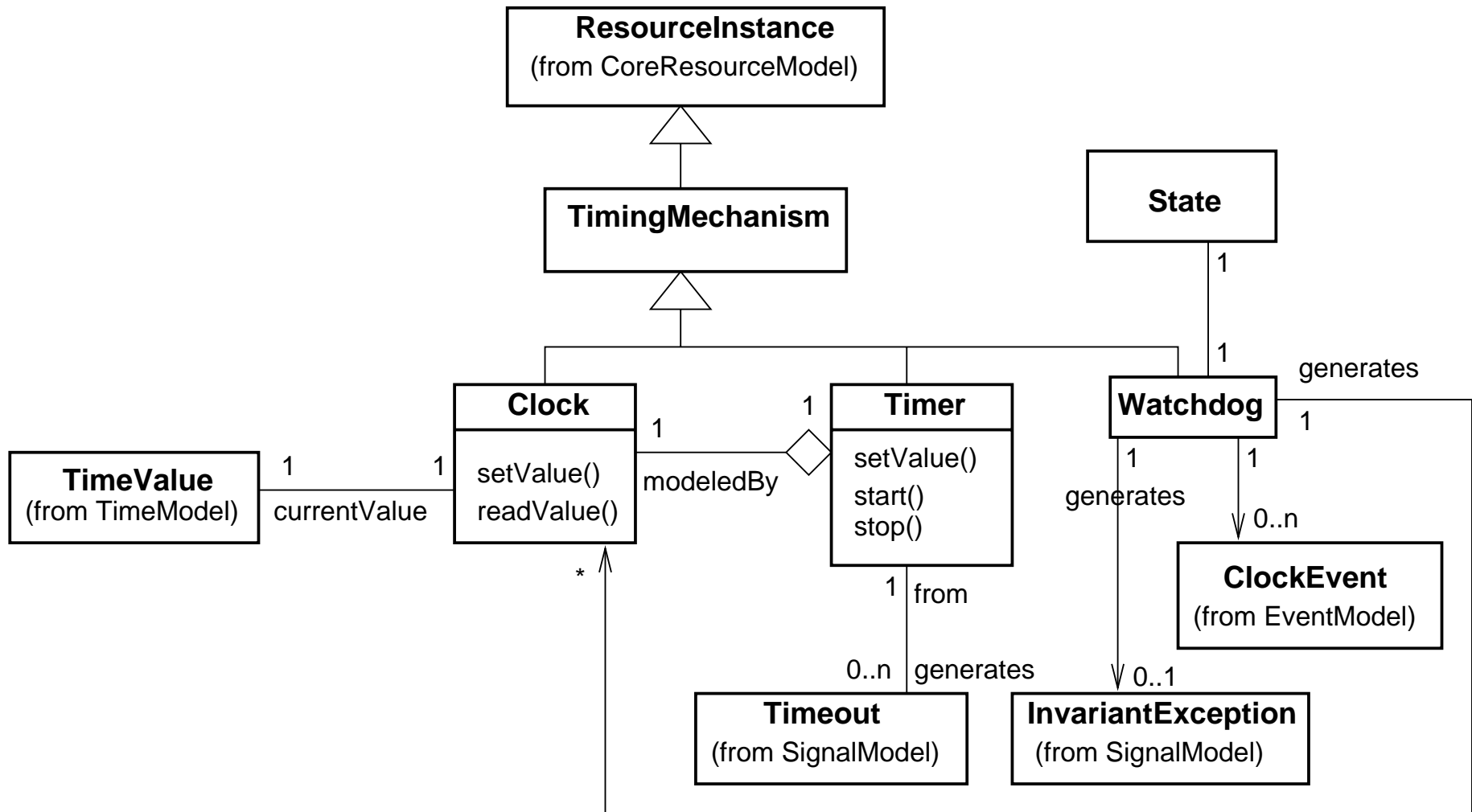
## Major Concepts:

- timed transitions
- timeouts
- relationship physical time  $\leftrightarrow$  clock time

## Main Elements:

- Clocks
- Watchdogs
- ClockEvents
- invariant-Exception
- sync-Actions

# Timing Mechanisms



# Example: Model Element Definition

---

## Clock

Clocks have standard operations such as reset, read, compare. However, as the model is aimed at modeling real systems, actual values used are integer, i.e. resets and tests are done with integers, as a controller would do with milliseconds 1 .....

.....

At any point in time, a clock value represents how much time elapsed since the last clock reset (with respect to the reset value). A clock can not be stopped.

Textual  
Description

### Operations:

*setValue()*      resets the clock to a given time value

*readValue()*    reads the value of a clock;  
This operation returns a TimeValue

Connections  
with other  
Modeling  
Elements

### Associations:

*currentValue*    the time value (real number)

# Outline

---

- 1** Introduction to WOODDES  
scope, objectives, partners, status
- 2** The UML-RealTime profile  
context, motivation, and notation
- 3** Methodology for Developing Embedded Systems  
basic tasks and iterations
- 4** Tool Support  
overview, model interchange, small demo
- 5** Expected Outcomes  
lessons to learn, case studies to validate

# WOODDES Methodology

---

**Objective:** give guidelines and detailed prescriptions, that define the layout of a design project  
in particular, define a *process* for development, that makes effective use of the UML-RT profile

**Context:** other object-oriented development methodologies like *SPEM*, *RUP*, *ROPES*, etc...

**Distinction:** focused on embedded real-time control tailored to the UML extension supported by a specified tool platform

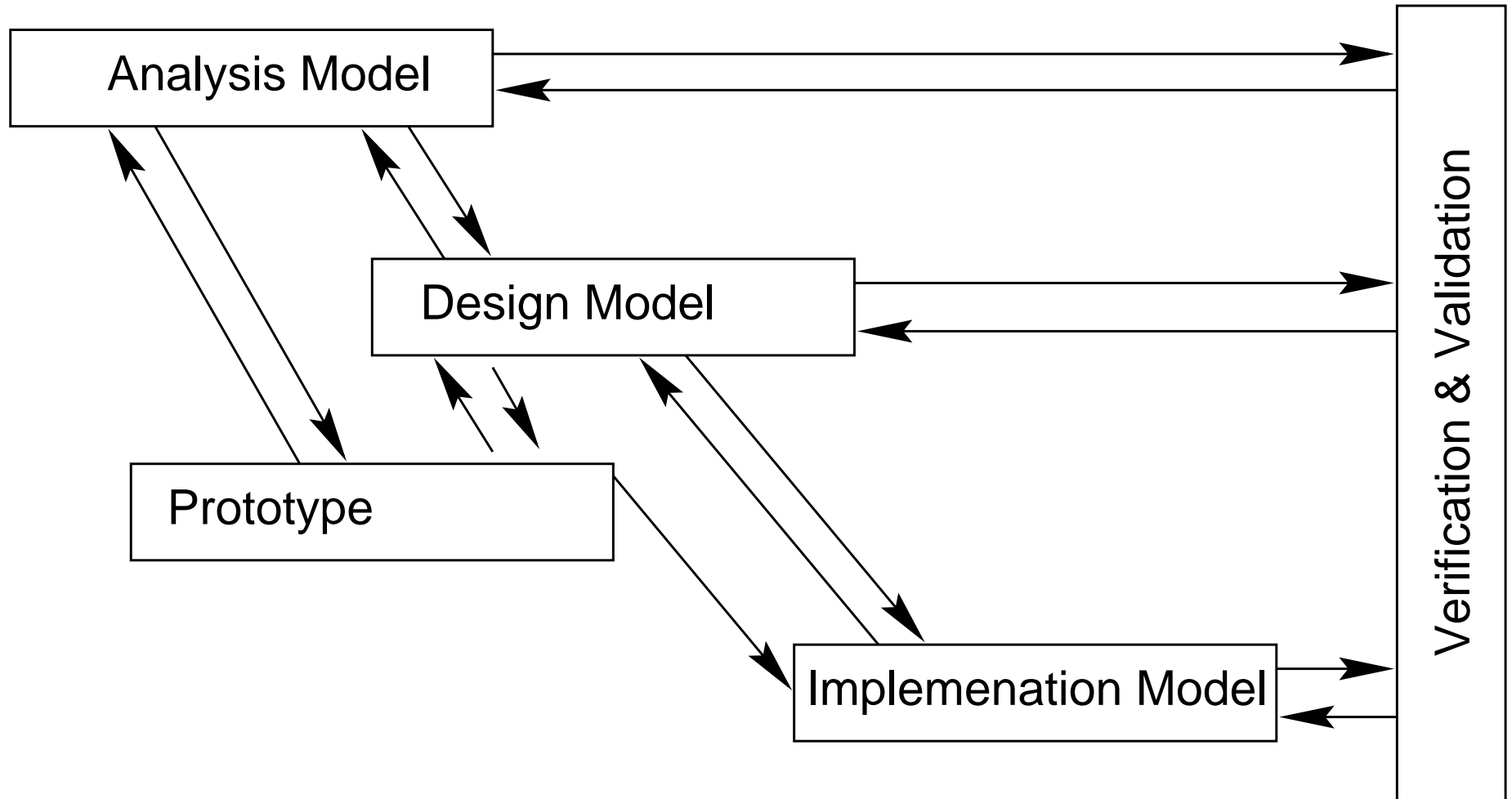


# Emphasis in the Methodology

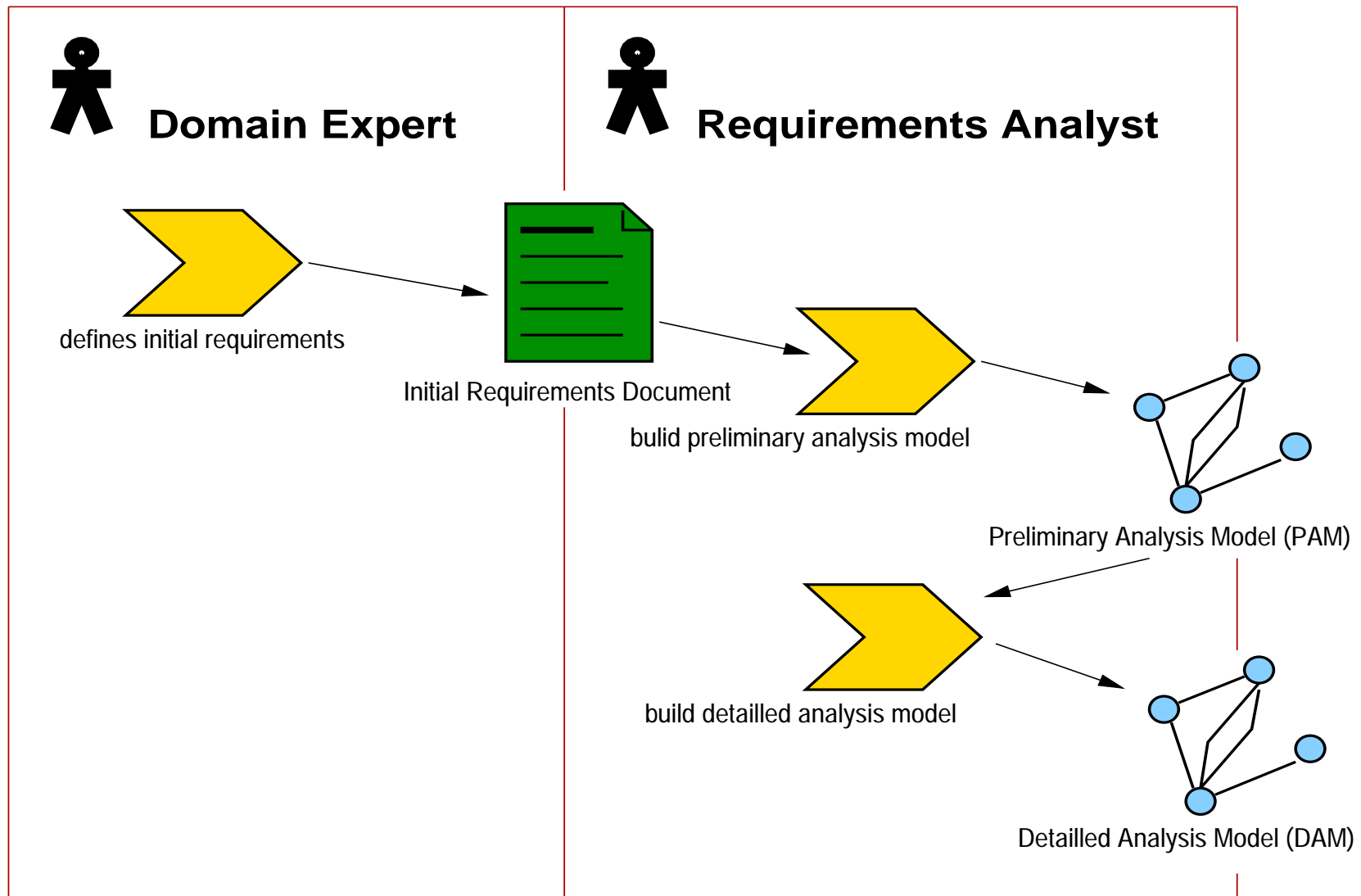
---

- Model-Based** UML description is *central* up to implementation  
increasing granularity during the process
- Continuous** stepwise refinement of the system model  
verification/validation integrated part
- Documented** artifacts of the process are reusable  
communication within large and heterogeneous teams

# Overview of the WOODDES Process



# Example: Build Analysis Model



# Verification/Validation Methods

	Analysis	Design	Implementation
User-guided Simulation	✓	✓	
Glass Box Testing			✓
Exhaustive Test Case Generation			(✓)
Program Slices (trace acceptance)			wrt. design
Safety Properties		✓	
Deadlock Detection		✓	(✓)
Time Stop/Zeno Behavior		✓	
Model Checks (properties)	✓	✓	
Conformance Check		wrt.analysis	wrt. design
Dead Code Detection	✓	✓	partial

# Outline

---

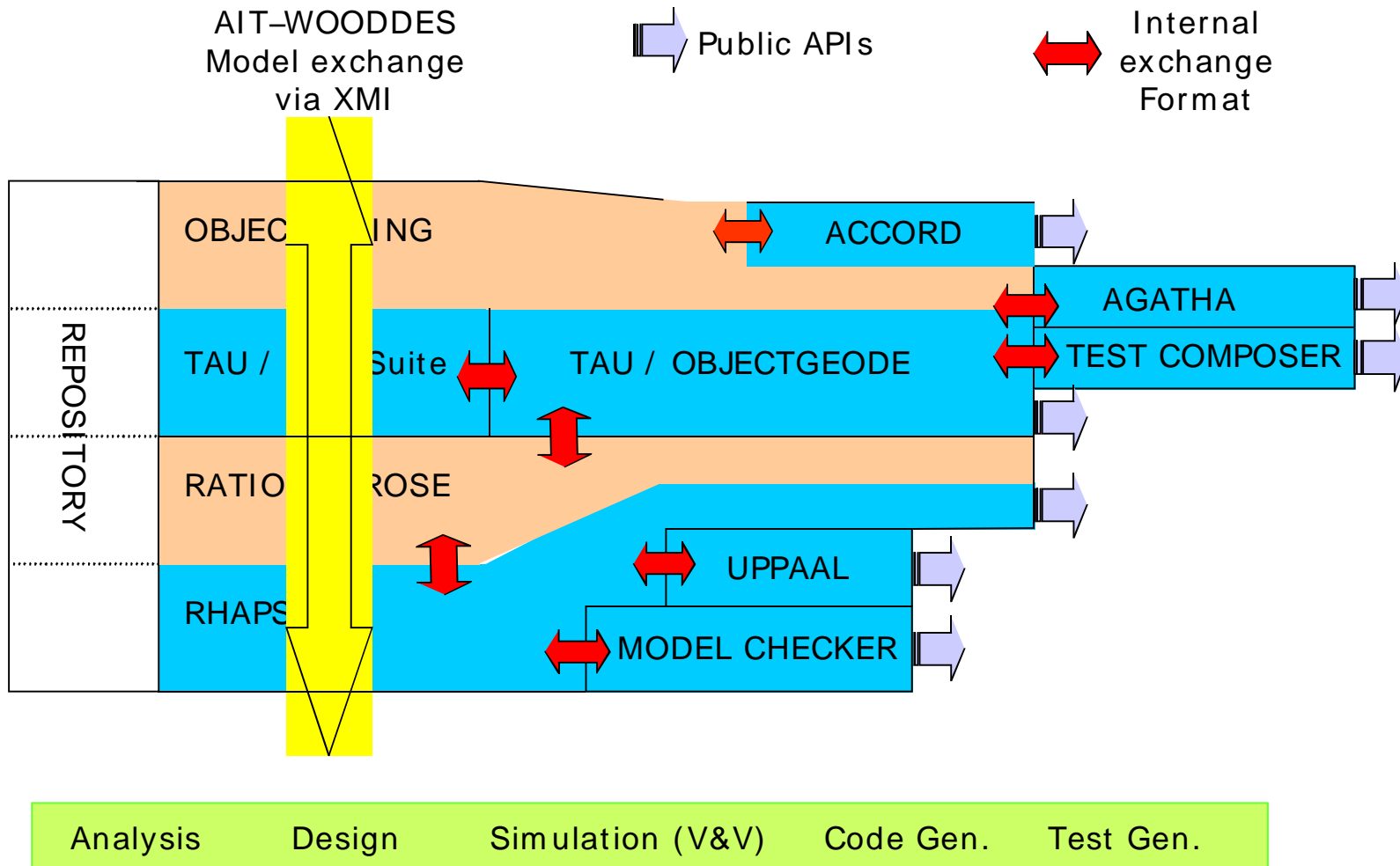
- 1** Introduction to WOODDES  
scope, objectives, partners, status
- 2** The UML-RealTime profile  
context, motivation, and notation
- 3** Methodology for Developing Embedded Systems  
basic tasks and iterations
- 4** Tool Support  
overview, model interchange, small demo
- 5** Expected Outcomes  
lessons to learn, case studies to validate

# The WOODDES Tool Platform

---

- Vision:** use *same data* throughout design process  
avoid duplications  
avoid mistakes
- Object Orientation:** high-level concepts  
step-wise refinement
- Realization:** define interfaces between tools  
use model exchange technologies (like XMI)  
implement semantic translations, if appropriate

# WOODDES Tool Interactions



In            tools owned by project partners

# Model Checking

---

$$M \stackrel{?}{\models} \varphi$$

$M$  : description of the system

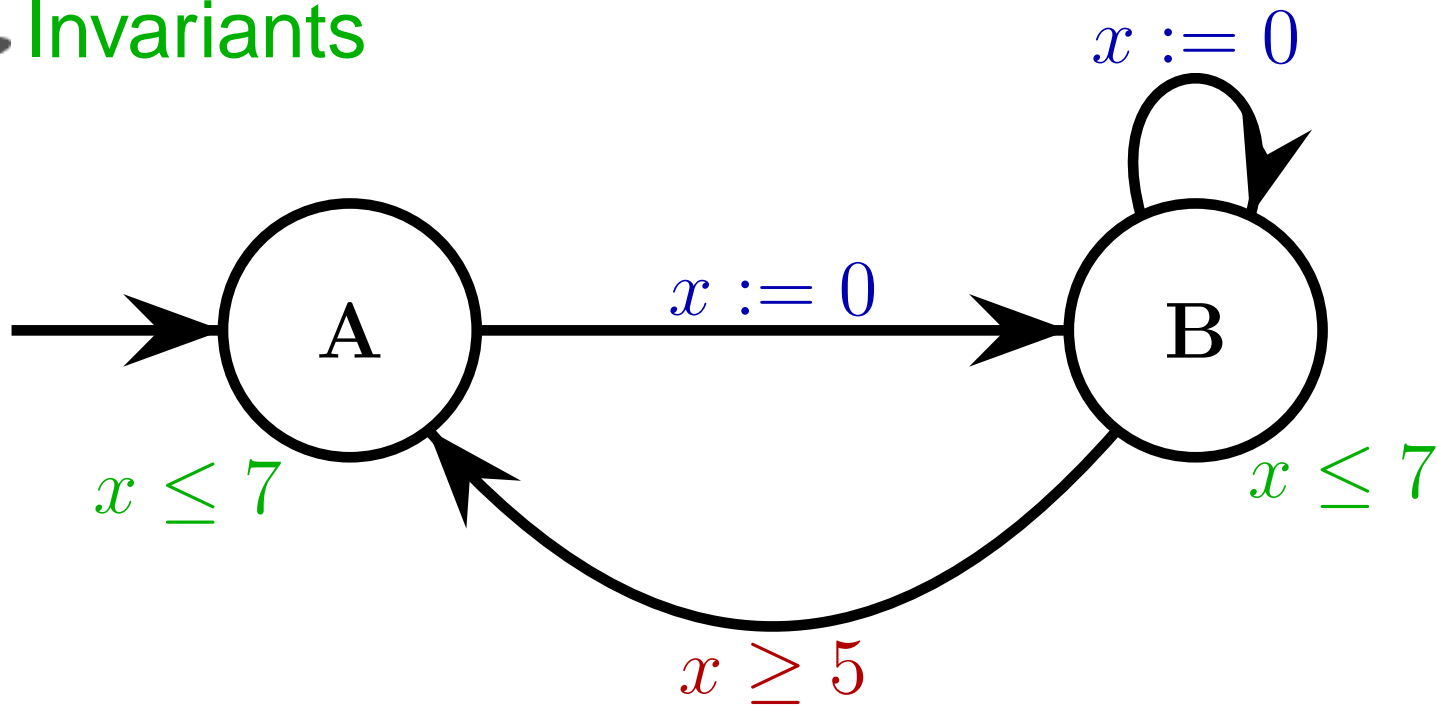
$\varphi$  : desired property

- easier than proving a general theorem
- completely automatic ('yes' or counterexample)
- *efficient* algorithms tailored for classes of problems

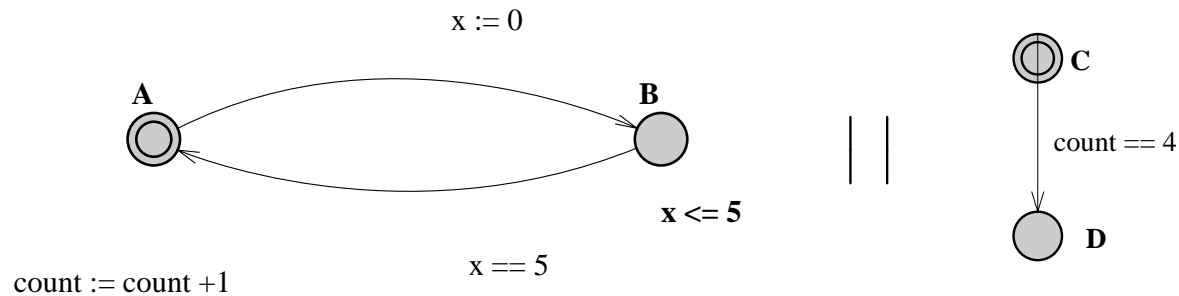


# Timed Automata Model

- Clocks
- (timed) **Guards**
- **Invariants**



# Real-Time Model Checking with UPPAAL



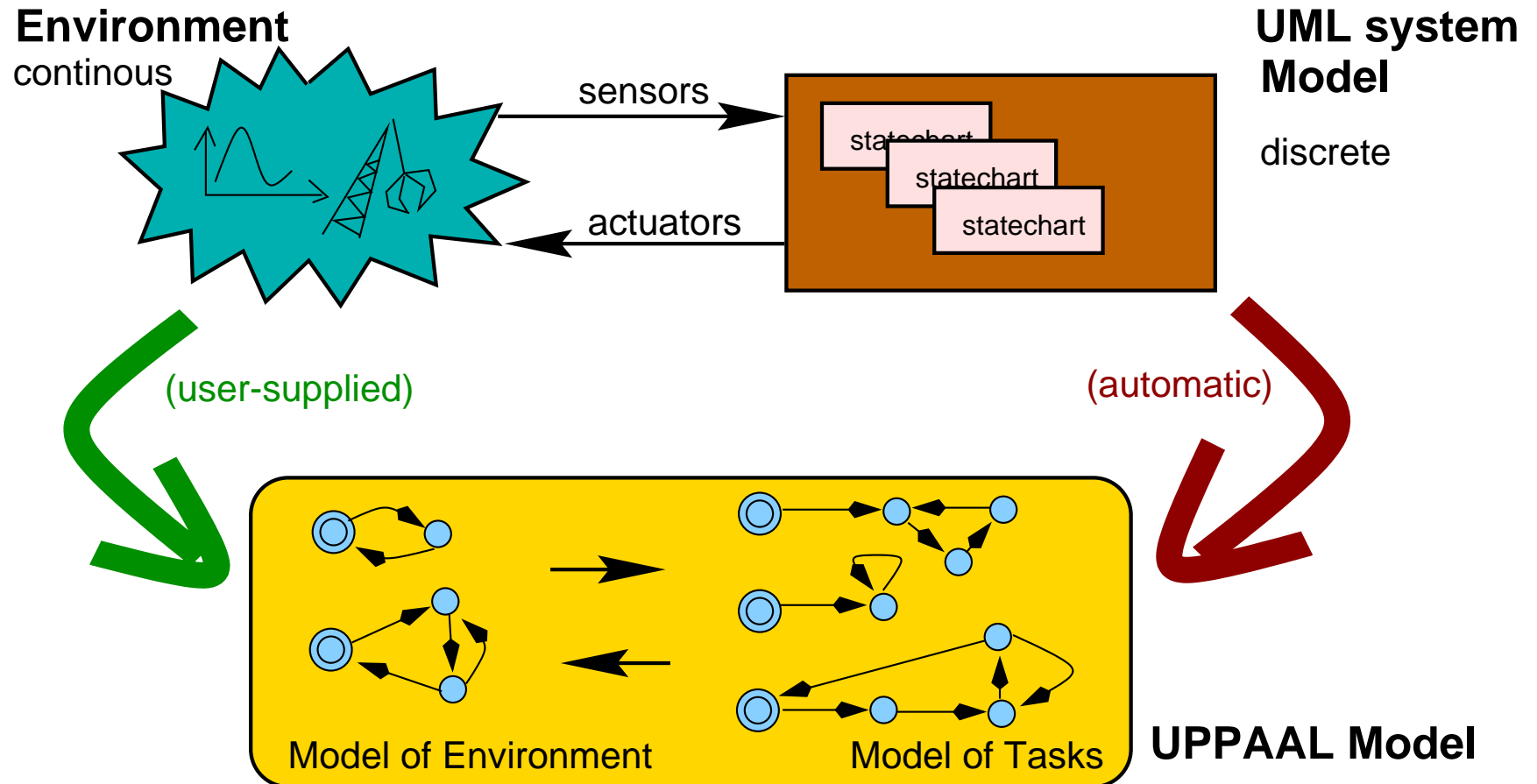
```
clock x; int count
```

Subset of timed computation tree logic (TCTL) supported:

- $E \langle \rangle \varphi$  reachability
- $A [] \varphi$  safety (invariantly  $\varphi$ )
- $E [] \varphi$  possibly always  $\varphi$
- $A \langle \rangle \varphi$  inevitably  $\varphi$
- $A [] \varphi \Rightarrow A \langle \rangle \psi$  unbounded response

$\varphi, \psi$  : propositional formula over locations and (existing) clocks

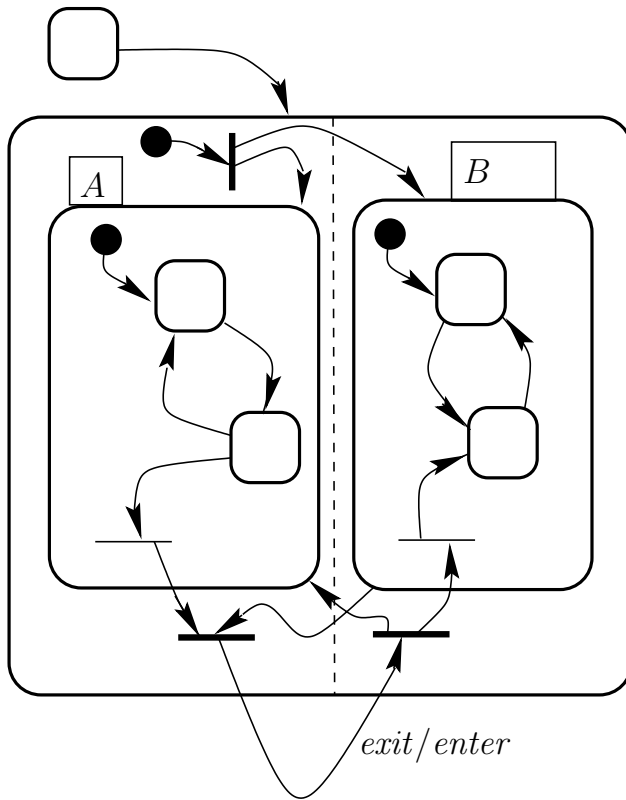
# Composing the Embedded System Model



# Restricted Statechart Formalism

## (Currently) restricted features

- hierarchical state machines ✓
- parallelism (on any level) ✓
- history ✓
- **no** event communication
- **no** sync states
- **no** inter-level transitions
- **no** actions that are dependent on states
- **no** actions on entry/exit



## instead:

- hand-shake style synchronization
- shared variables

# From (timed) Statecharts to UPPAAL

---

*Rhapsody timed Statechart*

hierarchical model

informal description



*HTA model*

$M_H$

TA-close hierarchy

formal semantics



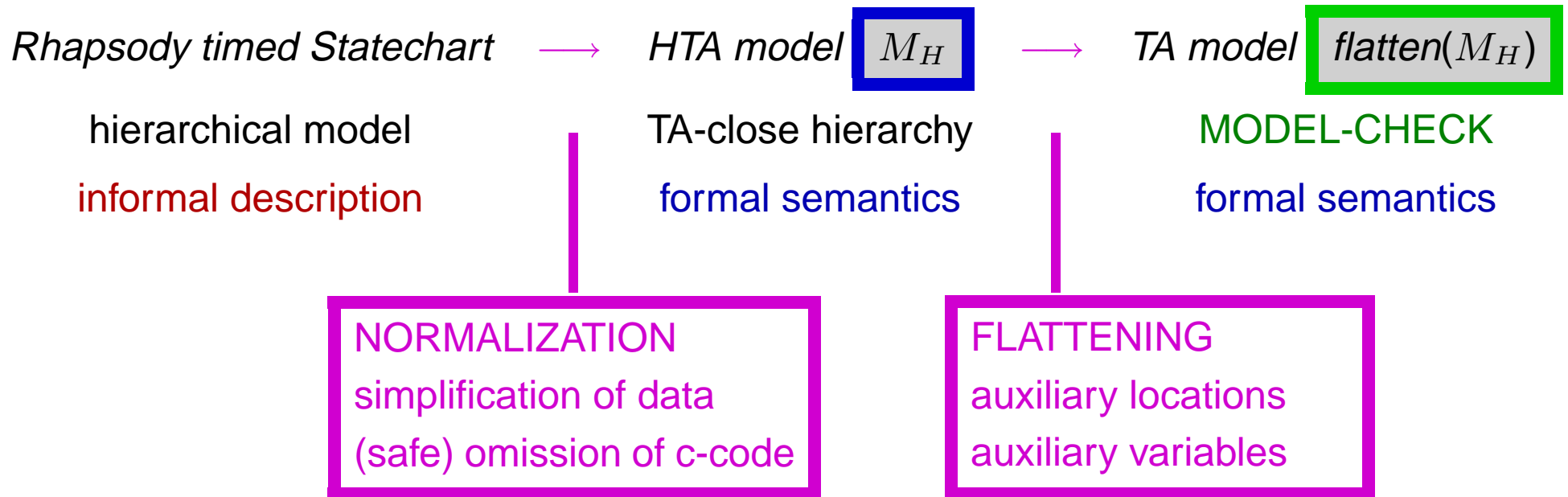
*TA model*

$\text{flatten}(M_H)$

MODEL-CHECK

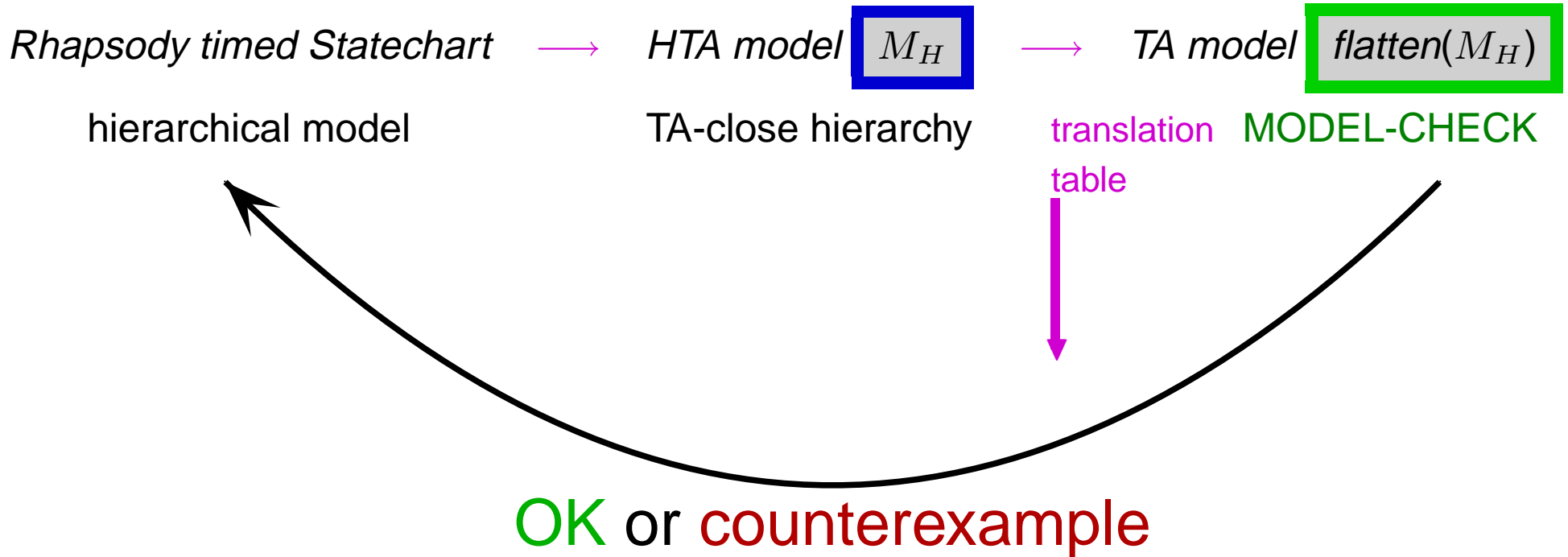
formal semantics

# From (timed) Statecharts to UPPAAL



**Guiding Principle:** Make it easy to adjust to small changes

# From UPPAAL back to (timed) Statecharts



For safety properties:

counterexample = trace

$\Rightarrow$  use simulator to analyze

$\Rightarrow$  correct model

# A word on semantics

---

## Semantics gives

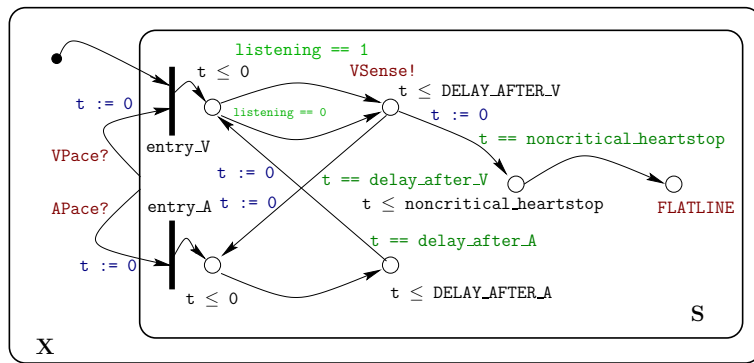
- an *unambiguous* description, of what can happen
- a *mathematical* model rather than a physical one

## Why is an 'implementation' not good enough?

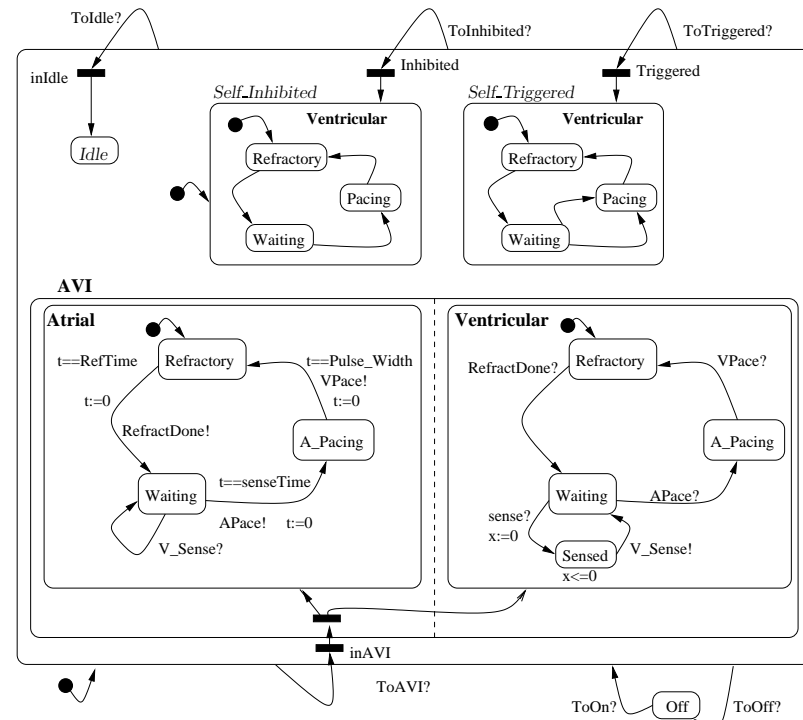
- no level of abstraction
- specification messy (if available at all)
- (often) implicit assumptions
- dependent on low-level hardware
- analysis needs *mathematical* rather than physical models



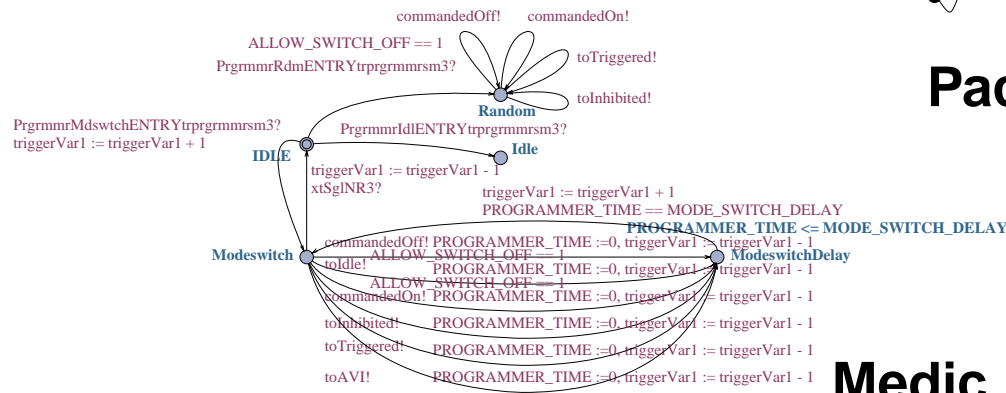
# Example: Cardiac Pacemaker Model



Human Heart



Pacemaker



Medic

# Model-Checking the Pacemaker

---

- DEADLOCK:  
possible (if heart stops)
- SAFETY:  
 $A[] \neg \text{heart stops}$   
only true for 'good' medic
- LIVENESS:  
 $A[] V \text{contract} \Rightarrow A\langle \rangle A \text{contract}$

# Model-Checking the Pacemaker

---

- DEADLOCK:  
possible (if heart stops)
- SAFETY:  
 $A[] \neg \text{heart stops}$   
only true for 'good' medic

- LIVENESS:  
 $A[] V\text{contract} \Rightarrow A\langle\rangle A\text{contract}$

## Parameters:

REFRACTORY\_TIME = 50  
SENSE\_TIMEOUT = 15

DELAY\_AFTER\_V = 50  
DELAY\_AFTER\_A = 5

HEART\_ALLOWED\_STOP\_TIME = 135

MODE\_SWITCH\_DELAY = 66

For  $\text{MODE\_SWITCH\_DELAY} = 65$ ,  $A[] \neg \text{heart stops}$  is violated

# Outline

---

- 1 Introduction to WOODDES  
scope, objectives, partners, status
- 2 The UML-RealTime profile  
context, motivation, and notation
- 3 Methodology for Developing Embedded Systems  
basic tasks and iterations
- 4 Tool Support  
overview, model interchange, small demo
- 5 Expected Outcomes  
lessons to learn, case studies to validate

# Expected Benefits

---

## End-users

- decreased design, prototyping and validation **time**
- derive **multiple specific models** from one UML high-level model
- decreased **feasibility and analysis** time;
- increased product **quality** through better reliability and safety, validation, and reuse

## Tool providers

- **integrated design toolset** that takes as input UML models, validates the system design and automatically generates the executable model (i.e. the target code)
- (step towards) **standardization** of UML notations;  
UML standard improvements compatible with the project solutions

# Documented Results

---

- RT-Profile** as an UML extension, that is standardized and tailored for **embedded system development**
- Methodology** As a concise collection of **rules, tools**, and **design steps** to be used
- Process** as **concrete** part of the methodology
- Assessment** of **applicability** of the methodology

# Find Out More...

---

[wooddes.intranet.gr](http://wooddes.intranet.gr)

# References

---

- [AD94] R. Alur and D.L. dill. A Theory of Timed Automata. In *Theoretical Computer Science*, number 125, 1994
- [vdB94] Michael von der Beeck. A Comparison of Statechart Variants. In de Roever Langmaack and Vytopil, editors, *Formal Techniques in RealTime and Fault-Tolerant Systems*, volume 863 of *Lecture Notes in Computer Science*, pages 128–148. Springer-Verlag, 1994.
- [D99] Bruce Powel Douglass. Real-Time UML, Second Edition - Developing Efficient Objects for Embedded Systems. *Addison-Wesley*, 1999
- [DM01] Alexandre David and M. Oliver Möller. From Hierarichcal Timed Automata to UPPAAL. Research Series RS-01-11, BRICS, Department of Computer Science, University of Aarhus, March 2001. see <http://www.brics.dk/RS/01/11/index.html>.
- [OMG] Unified Modeling Language, version 1.4. Download from <http://www.omg.org>
- [WOODDES] WOODDES web page: <http://wooddes.intranet.gr>