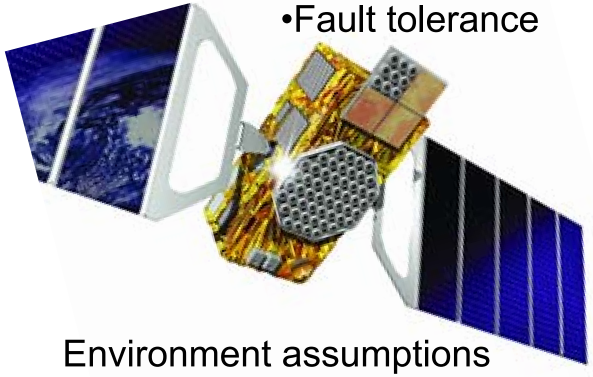
Quasimodo

Quantitative System Properties in Model-Driven-Design of Embedded Systems

Kim G. Larsen & Brian Nielsen

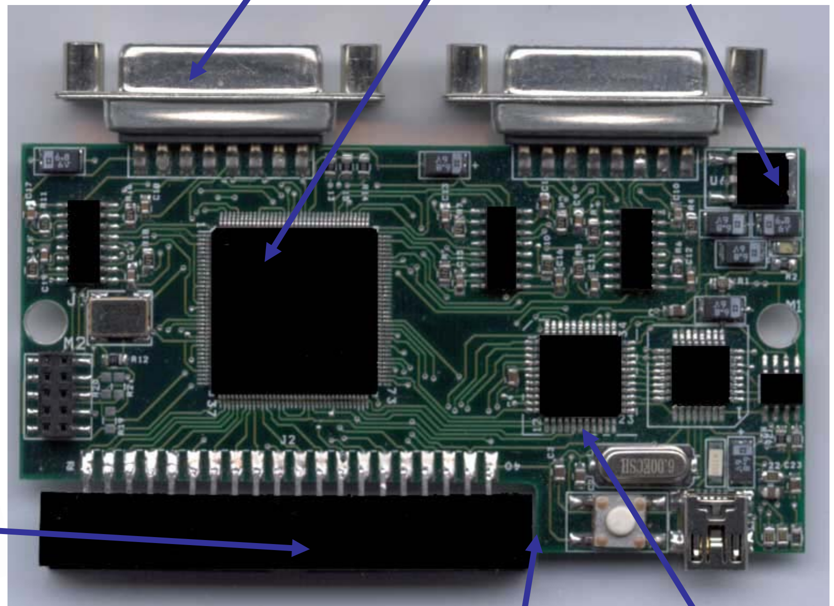
Aalborg University, DK

Quantitative System Properties



Service requirements
- QoS
- Availability
- Fault tolerance

Environment assumptions
- Timing constraints
- Hybrid behavior
- Arrival rates



Communication bandwidth

Computation resources

Power consumption

Costs

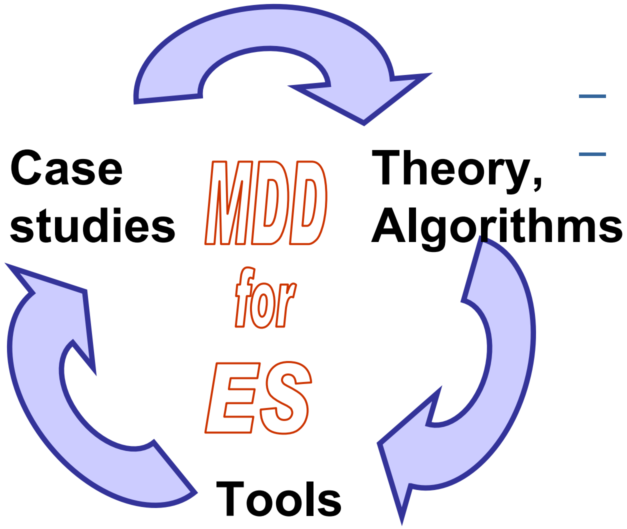
Memory usage

Quasimodo Research Goals

We apply (techniques/tools) them to **several complex industrial case studies.**

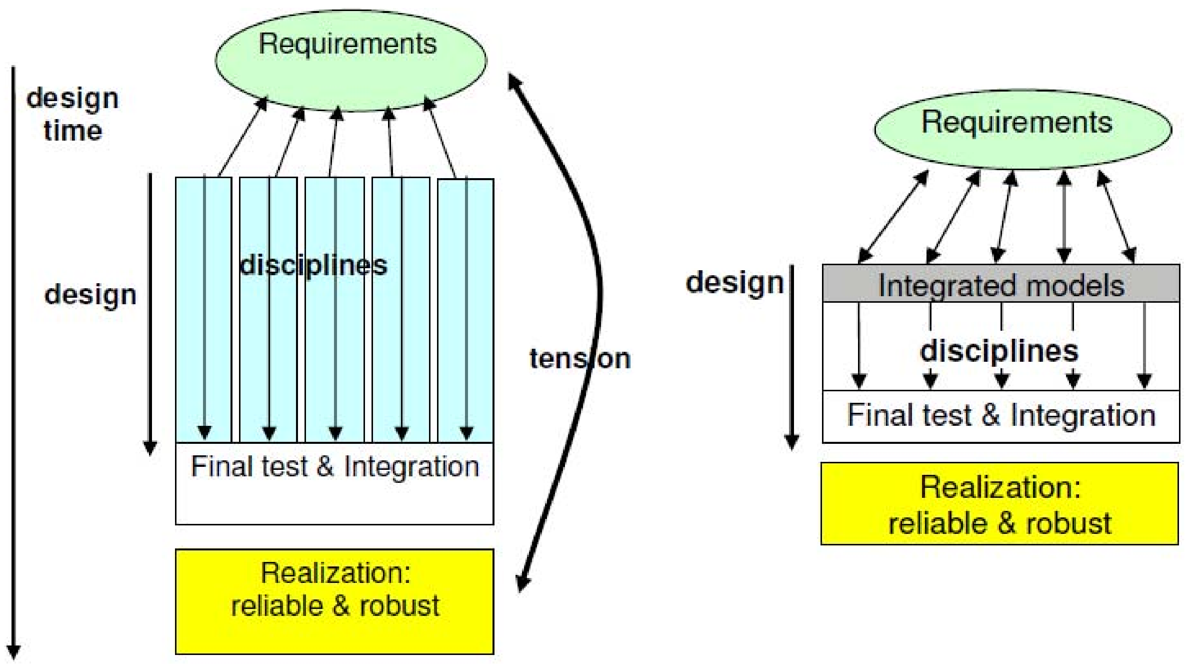
Improve **Quantitative** aspects

- Modeling, notation, semantics
- Analysis/Abstraction techniques
- Code/controller synthesis
- Test generation



- **Provide unique tool components** to be used as plug-ins in **industrial tools / tool chains**
- Create first prototypes of an integrated **tool environment**
- **Disseminate**

Early Integration & validation



(a) Traditional (b) Integrated

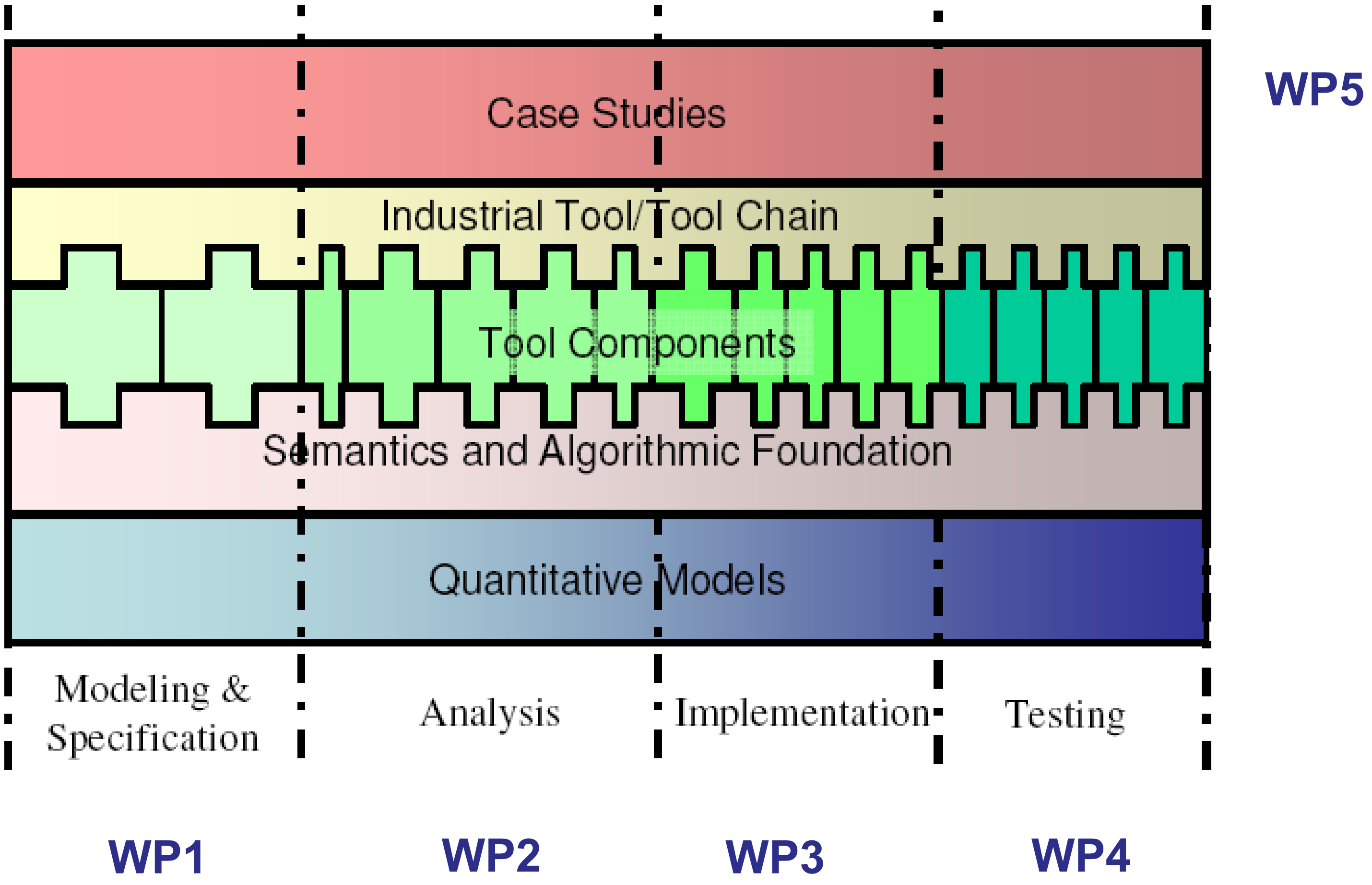
Improve competitiveness of European companies that rely on the design and integration of embedded systems in their products by reducing costs and time to market

Workplan Strategy

"Well documented API's & exchange formats"

xLTS

StateCharts & timed, hybrid, stochastic, priced, automata



Partners

Quasimodo

- Aalborg University / CISS
- Terma Space A/S

- Embedded Systems Institute
  - Twente University
  - Radboud University
- CHESS

- Université Libre de Bruxelles (CFV)

- CNRS (LSV, ENS Cachan)

- Saarland University
- RWTH Aachen
- Hydac Electronic Gmbh

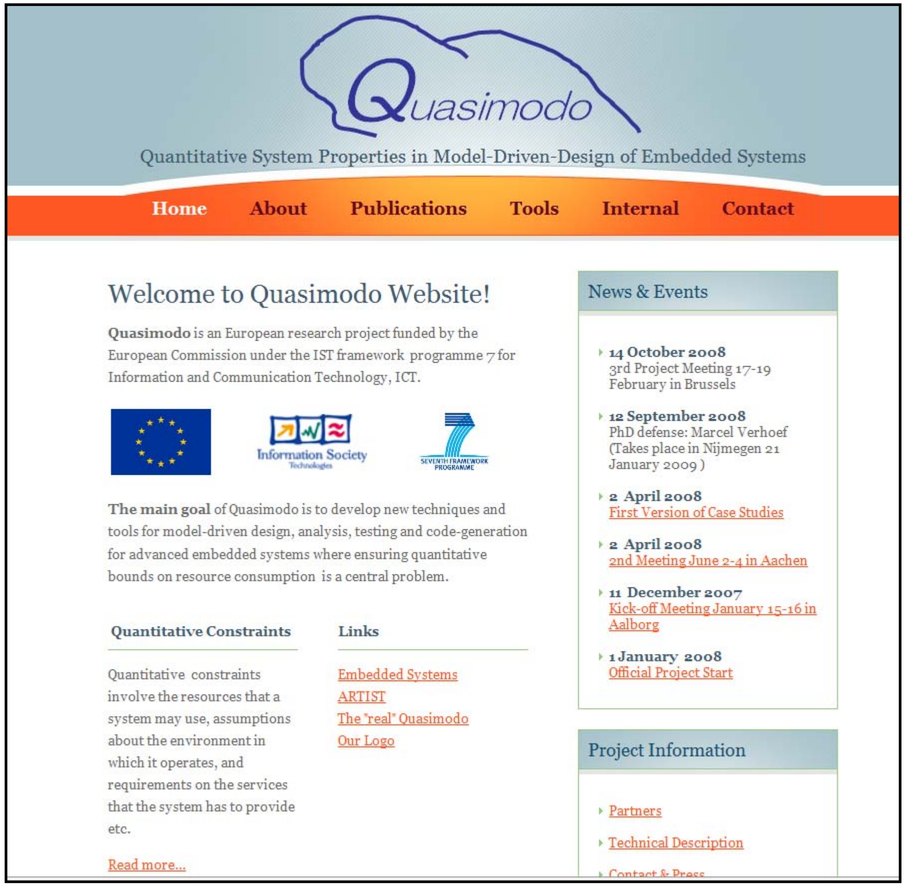
Work Tasks

WP1 Modeling and Specification	**WP2** Analysis	**WP3** Implementation	**WP4** Testing	**WP5** Case Studies, Tools, Disse-mination and Exploitation
T1.1 Model Process Improvement	T2.1: State Space Representation and Model Checking	T3.1: Controller synthesis and scheduling	T4.1: Test Generation	T5.1: Case Studies
T1.2: Modeling of Quantitative System Aspects	T2.2: Abstraction, Refinement and Compositionality	T3.2: Implementability and code generation	T4.2: Approximate Testing	T5.2: Tool Plugins and Tool Chain Integration.
T1.3: Design Notation and Tools	T2.3: Approximate Analysis Techniques			T5.3: Dissemination and Exploitation

The Hunch-backs

Quasimodo

www.quasimodo.aau.dk



Logo

WP1: Modeling and Specification

Leader: Frits Vaandrager (ESI/RU)

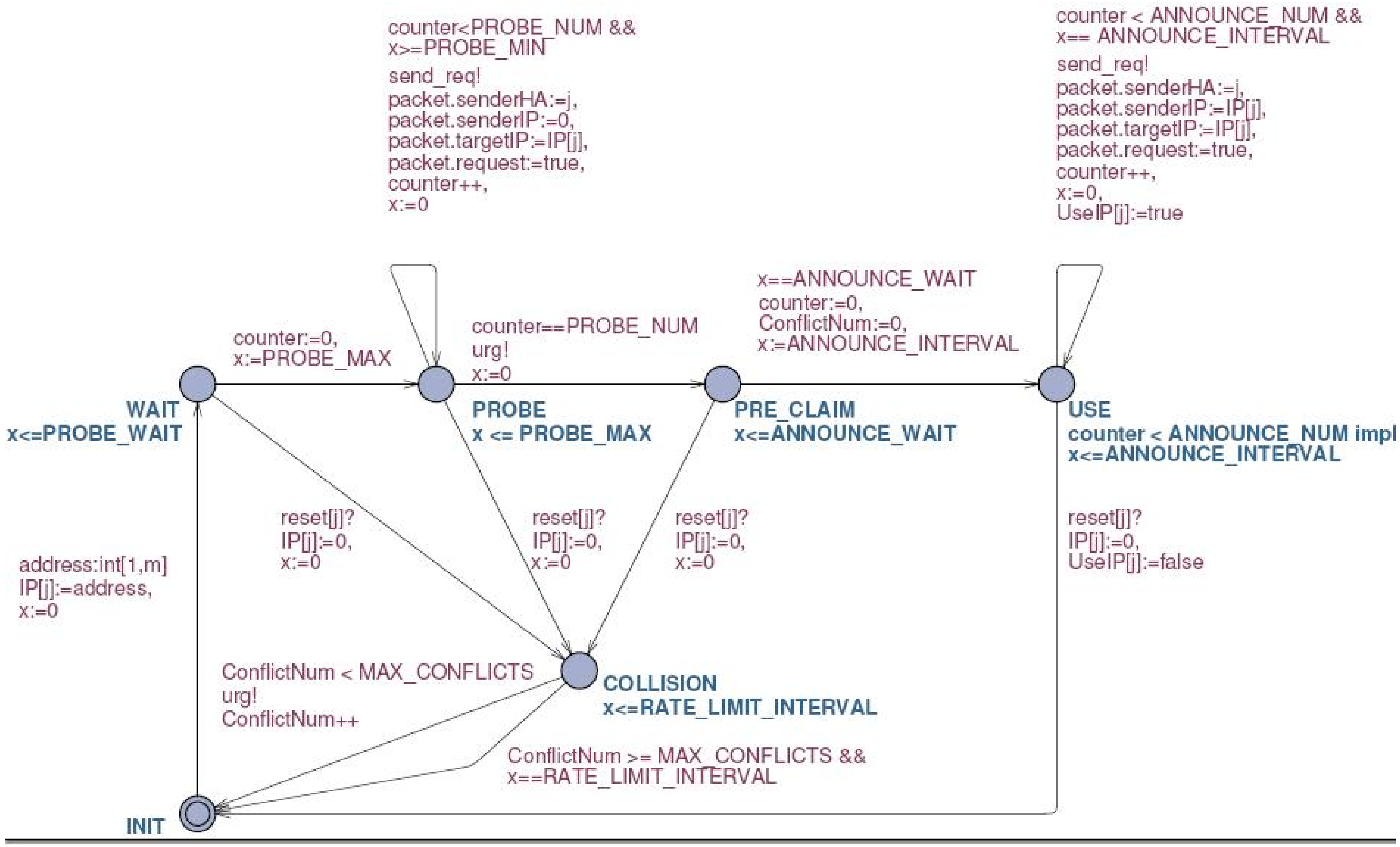
- How to obtain adequate and faithful models of embedded systems?
  - Modeling formalisms integrating timed, hybrid, cost and stochastic aspects
  - Overarching design notation for embedded systems with accompanying tool support?
  - Modeling process and model management

T1.1: Model Process Improvement

T1.2: Modeling of Quantitative System Aspects

T1.3: Design Notation and Tools

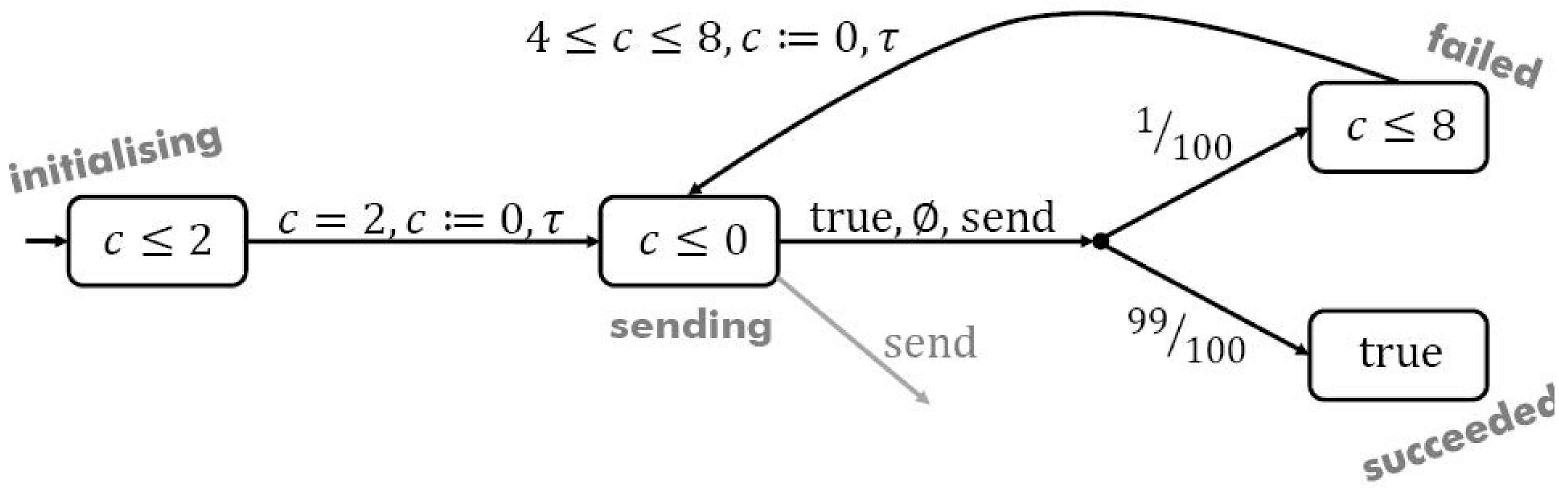
Timed Automata



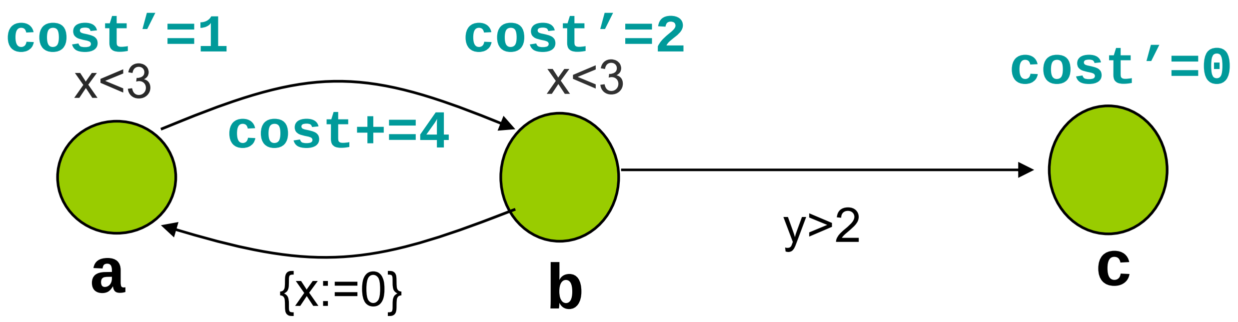
Probabilistic Timed Automata

Timed automata

+ probabilistic branching finite



Priced Timed Automata



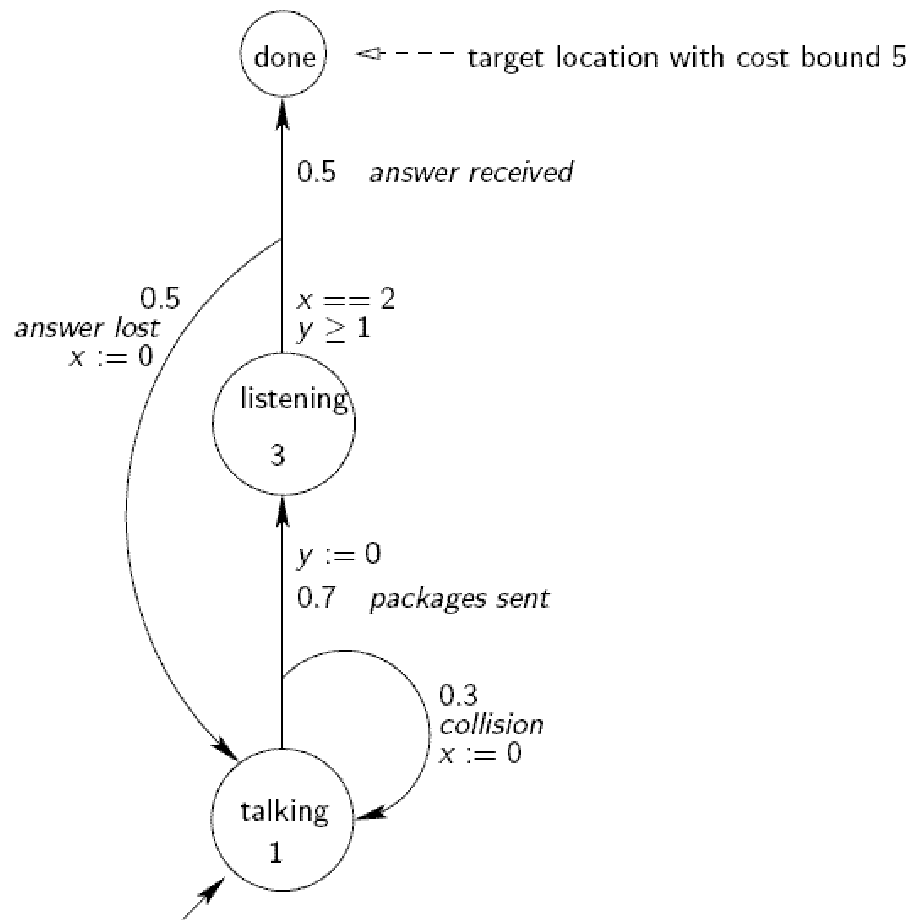
- Timed Automata + Costs on transitions and locations
  - Cost of performing transition: Transition cost
  - Cost of performing delay **d**: (**d** x location cost)

$$(\mathbf{a},x=y=0) \xrightarrow[4]{} (\mathbf{b},x=y=0) \xrightarrow[2.5 \times 2]{\varepsilon(2.5)} (\mathbf{b},x=y=2.5) \xrightarrow[0]{} (\mathbf{a},x=0,y=2.5)$$

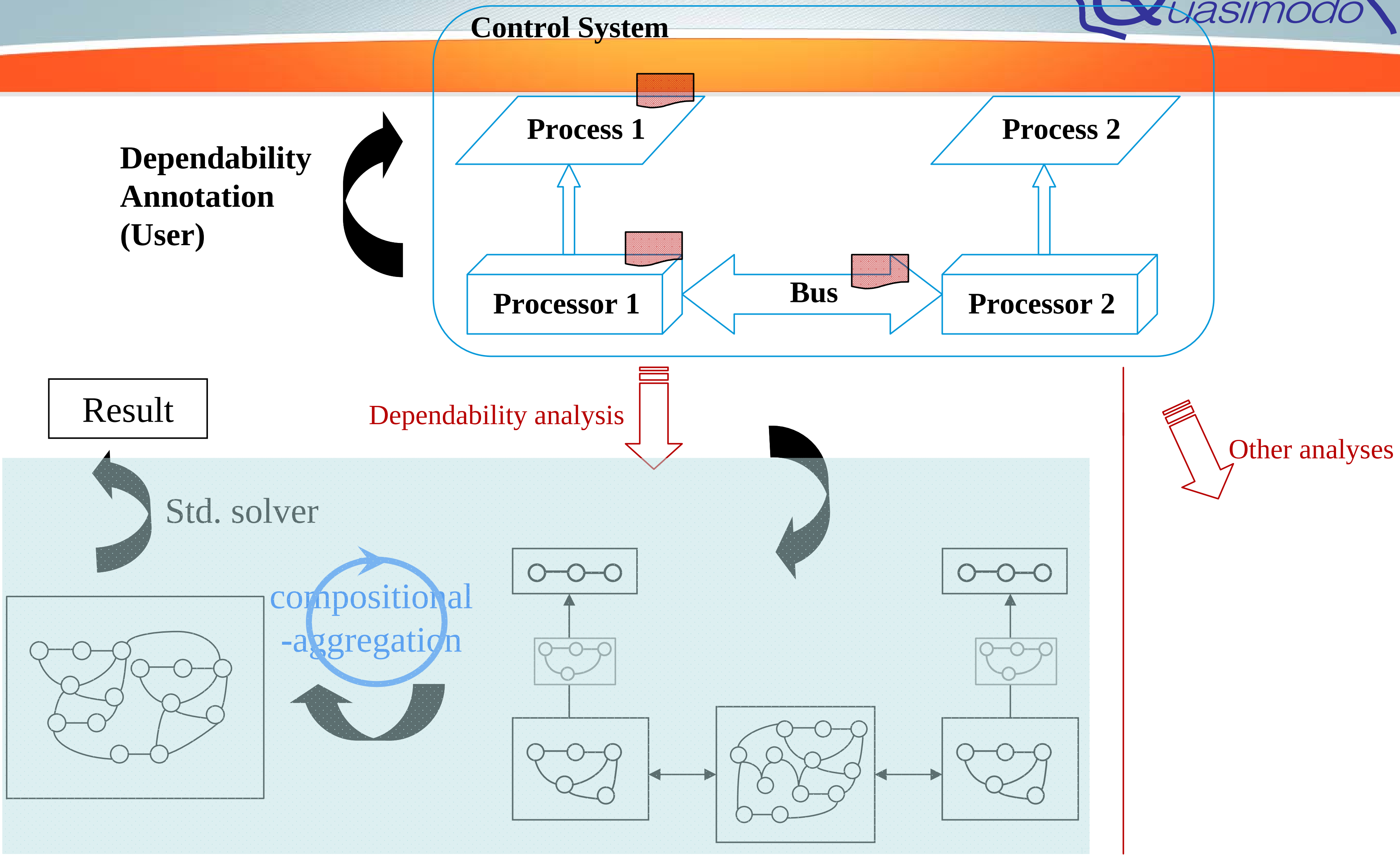
- Cost of Execution Trace: Sum of costs: **4 + 5 + 0 = 9**

Problem: Find minimum cost of reaching location **c**

Priced Probabilistic Timed Automata



Arcade (SU/UT)

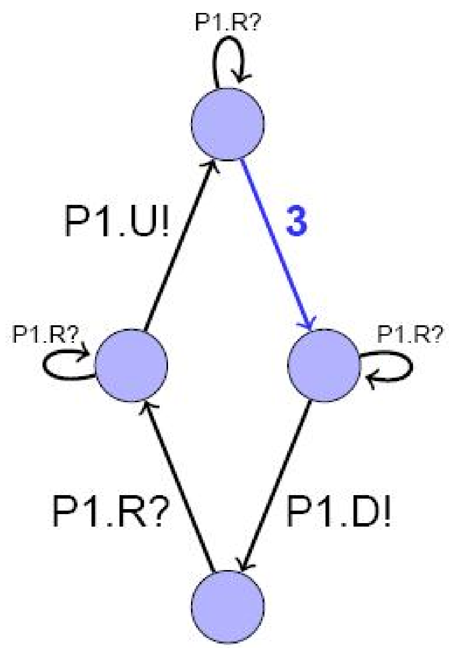


Control System
Process 1
Process 2
Processor 1
Processor 2
Bus
Dependability
Annotation
(User)
Result
Dependability analysis
Other analyses
Std. solver
compositional
-aggregation

I/O Interactive Markov Chains

Input/Output Interactive Markov Chains

- Combination of I/O automata and Markov chains, close to IMCs
- Discrete state space
- Labelled transitions
  - Markovian transitions (**3**)
  - Input transitions (?)
  - Output transitions (!)
  - Internal transitions (;)
- Input enabled, outputs cannot be delayed
- Parallel composition, hiding, renaming, lumping, etc.



WP2: Analysis

Leader: Joost-Pieter Katoen (RWTH)

- How to accurately, effective and efficiently (algorithmically) analyze quantitative models?
  - Design of data structures for representing and exploring behavior of (combined) quantitative models.
  - Support for interrelating different quantitative models preserving properties.
  - Methods for allowing partial analysis of very complex models (size or quant. aspects considered).

T2.1: Statespace Representation and Model checking

T2.2: Abstraction, Refinement and Composi-tionality

T2.3: Approximate Analysis Techniques

WP2: Analysis

- Discrete-time Markov decision chains:
  - Trace equivalence for labeled MDPs
  - Decision algorithms for (strong and weak) probabilistic simulation on DTMCs
  - Probabilistic CEGAR for MDPs
  - Regular expressions for PCTL counterexamples
- Continuous-time Markov chains:
  - Discrete-event simulation for CSL model checking
  - Parameter synthesis for time-bounded reachability
  - Infinite-state CTMC model checking
  - Advances in three-valued abstraction
  - Minimization of acyclic phase-type distributions

WP2: Analysis

- Probabilistic timed automata:
  - Complexity results for 1-2C PTA model checking
  - Probabilistic semantics of timed automata
  - Probabilistic time-abstract bisimulation
  - Uppaal-PRO tool for PTA model checking
  - Undecidability of CBPR in priced 3C PTA
  - Decidability and data structures for Pareto-optimal reachability in multi-priced TA
  - Decidability for optimal infinite scheduling for priced TA using mean pay-offs and discounting metrics
  - Heuristic guided search for TA using Russian Doll principle

WP3: Implementation

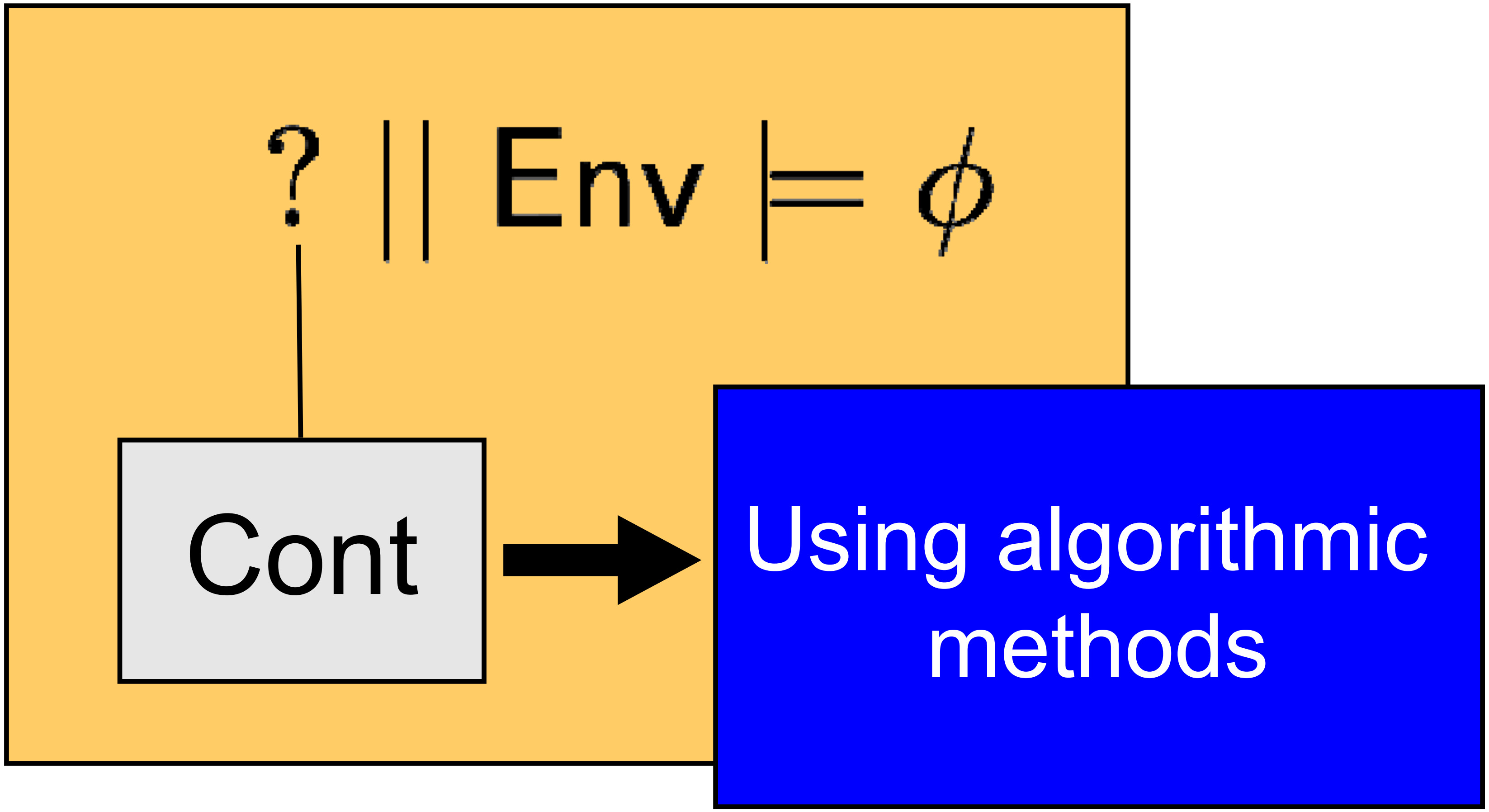
Leader: Jean Francois Raskin (CFV/ULB)

- How to algorithmically synthesize correct implementations of (continuous time) models on imperfect (digital) hardware/ software components?
  - Controller and scheduler synthesis
  - Property preserving code generation

T3.1:
Controller synthesis and scheduling

T3.2:
Implementa-bility and code generation

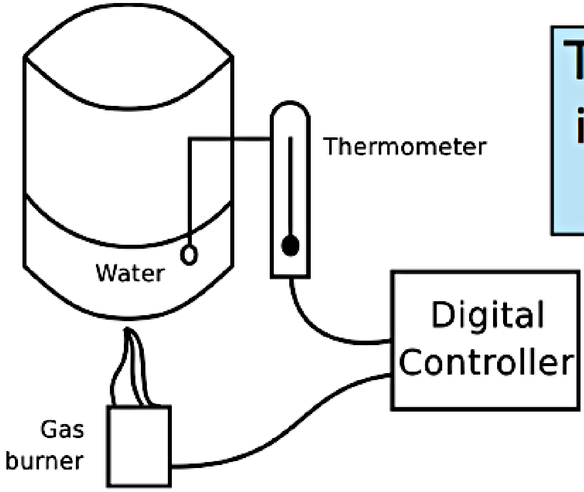
Controller Synthesis & Scheduling



Controller synthesis and scheduling

Highlight: imperfect information

Finite precision = imperfect information

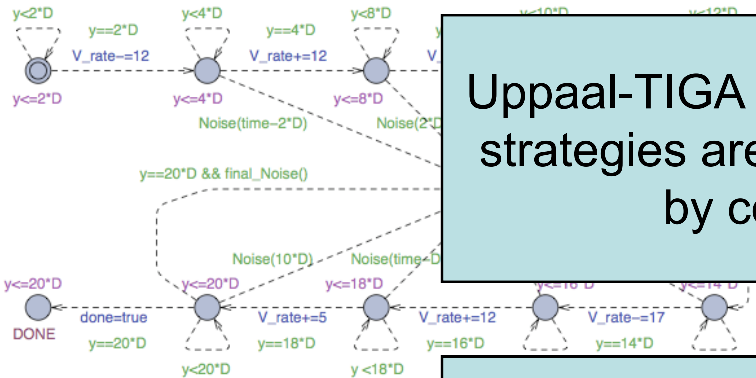


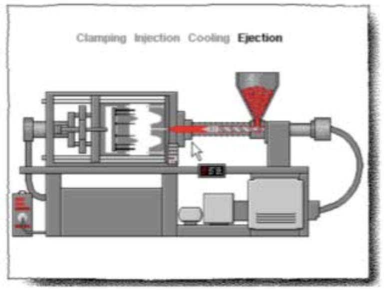
The temperature is in the interval $(c-1, c+1)$

Typical hybrid system

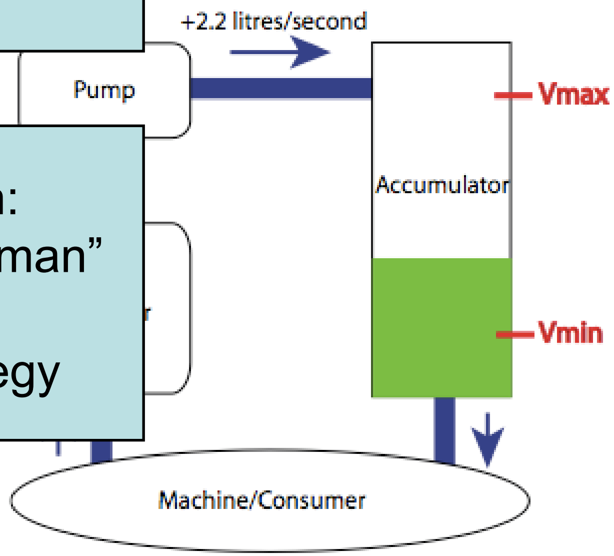
Case Studies
HYDAC: Accumulator Charge Controller

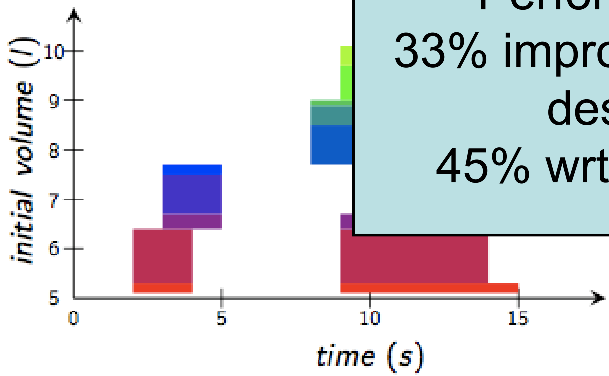
Accumulator Charge Controller (HYDAC)





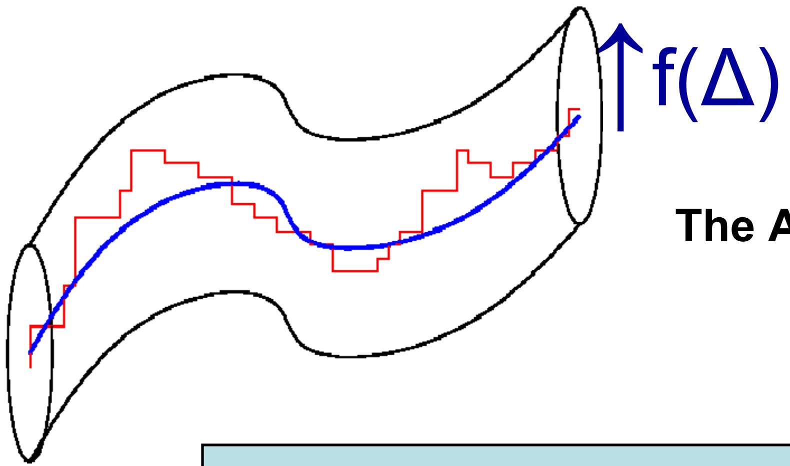
Uppaal-TIGA controller synthesis: strategies are correct and robust by construction





Performance simulation:
33% improvement over "human" designed strategy
45% wrt bang-bang strategy

Implementability and code generation



The AASAP semantics define a tube a strategies

- AASAP semantics define a "tube" of strategies instead of a unique strategy in the ASAP semantics.
- This tube can be refined into an implementation while preserving all LTL properties

Highlight: robustness analysis in UppAal

- **Robust** semantics is available within UppAal;
- Original results were formulated on **regions**;
- For implementation, we need a **zone based** formulation;

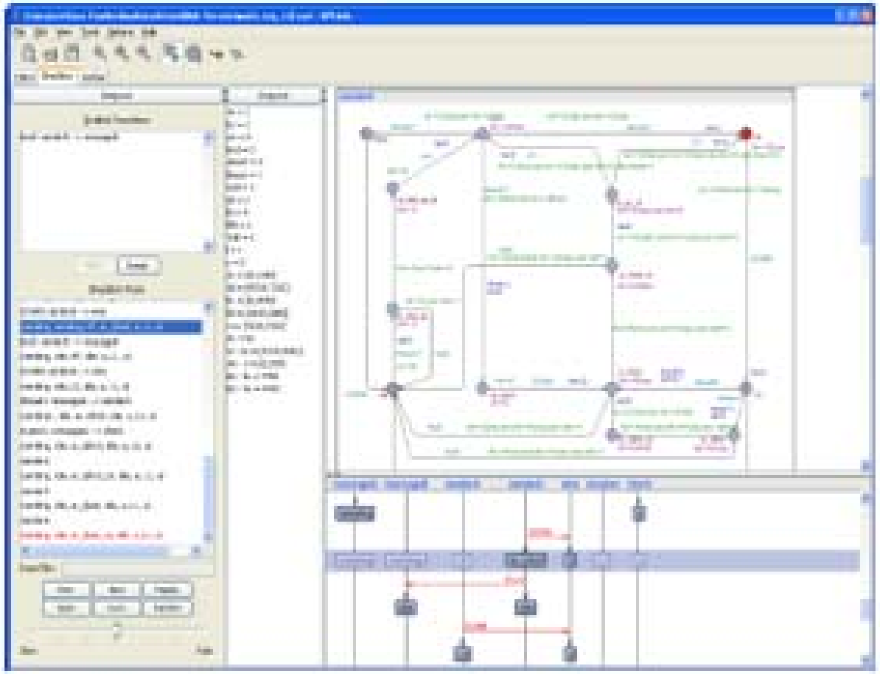


Figure 1: UPPAAL on screen.

WP4: Testing

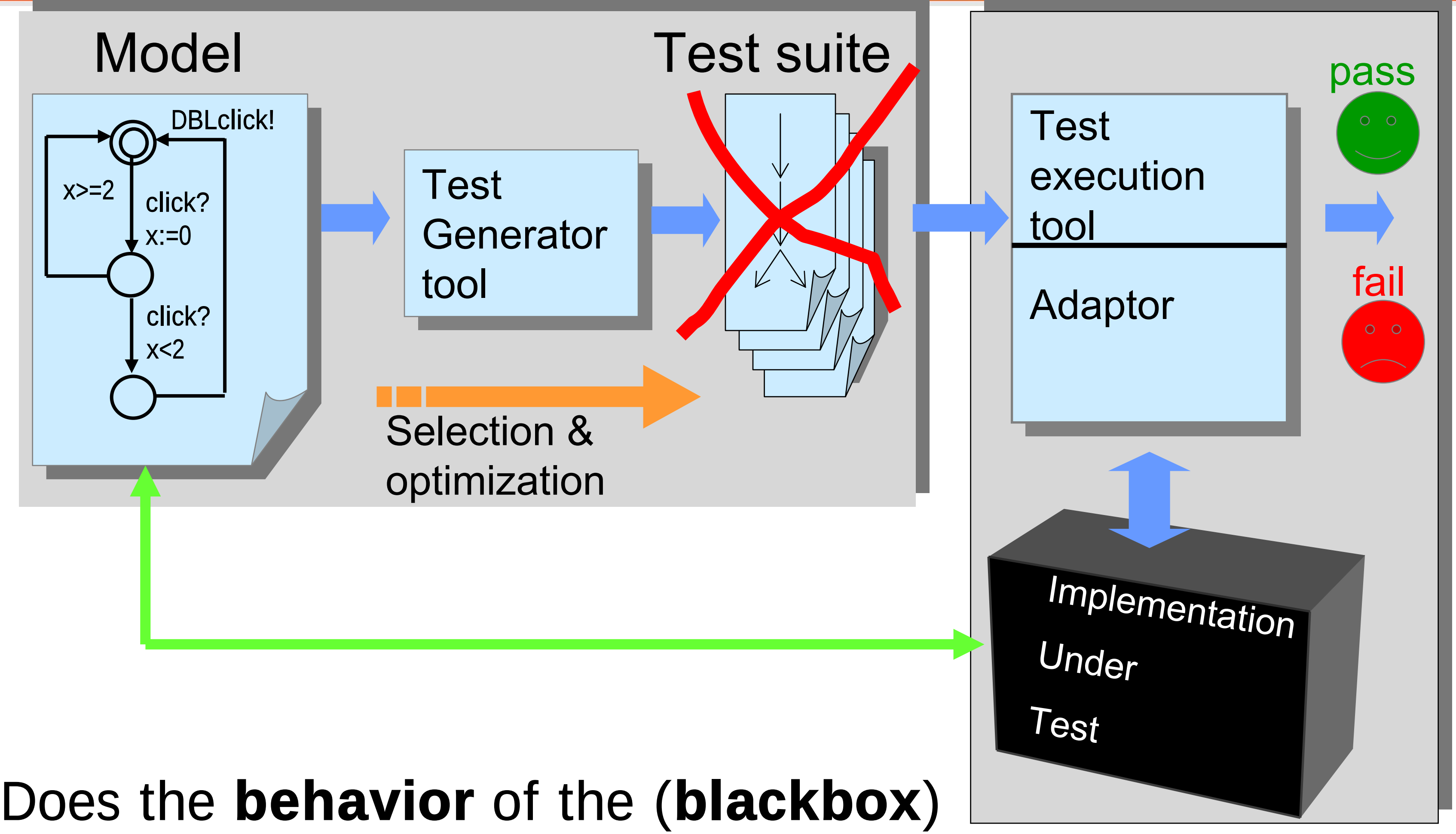
Leader: Arne Skou (AAU)

- How to automatically generate test cases from models with quantities?
  - Theory, algorithms, tools
  - Test selection methods, guidance and heuristics for optimal testing, and to define coverage metrics
  - Approximative testing methods to handle measurement and observation imprecisions

T4.1:
Test
generation

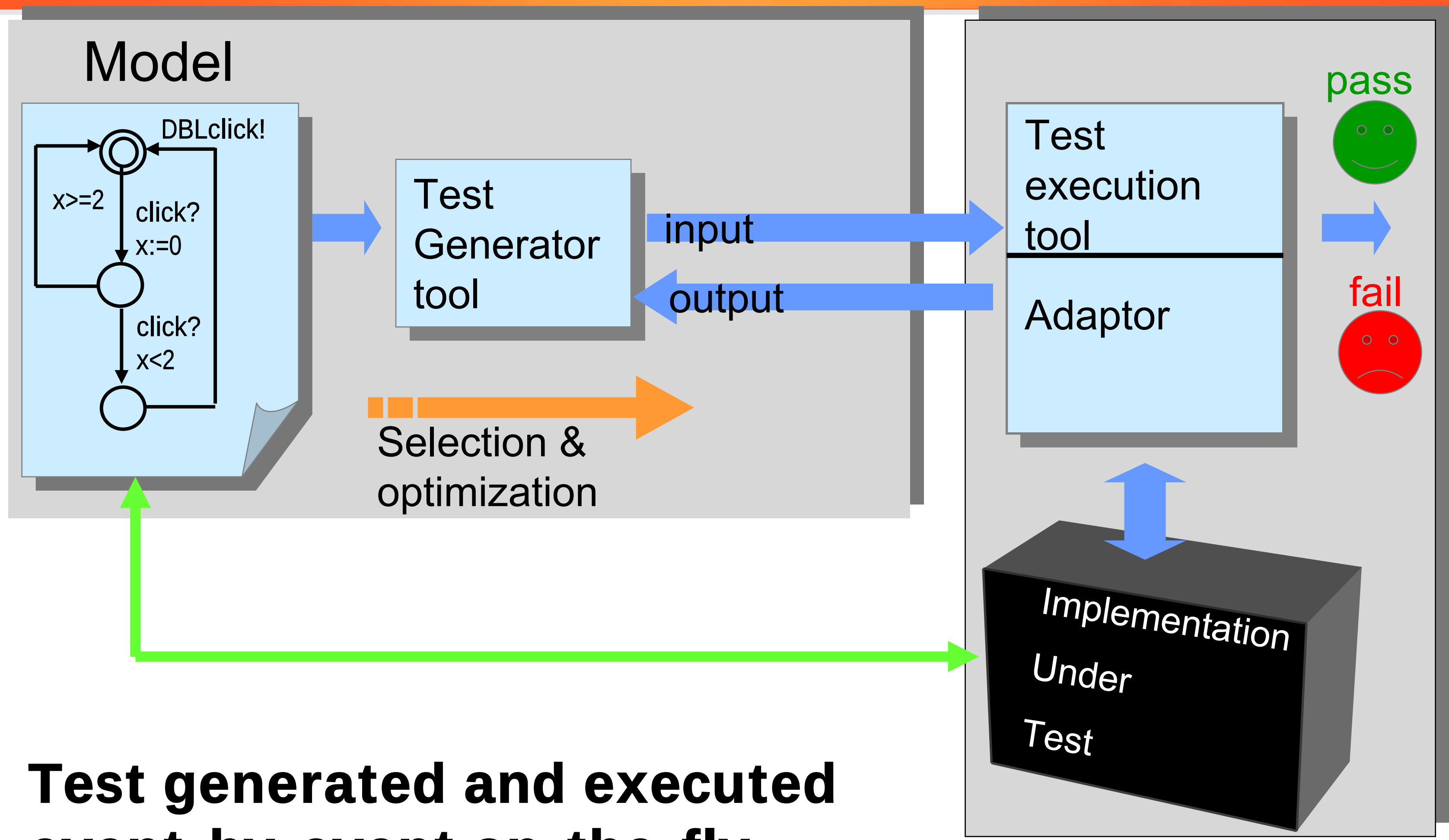
T4.2:
Approximate
Testing

WP4: Model Based Conformance Testing – offline



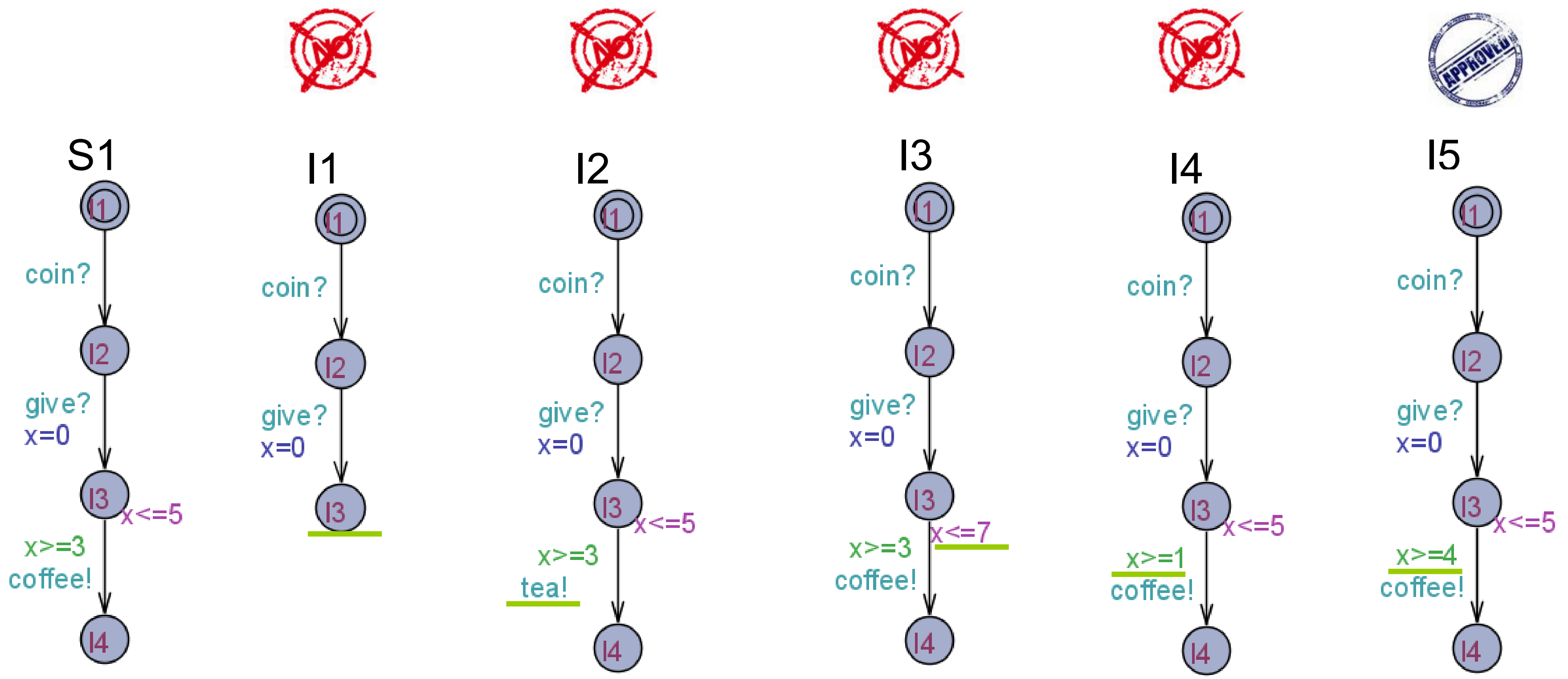
Does the **behavior** of the (**blackbox**) implementation ***comply*** to that of the specification?

WP4: Model Based Conformance Testing – online



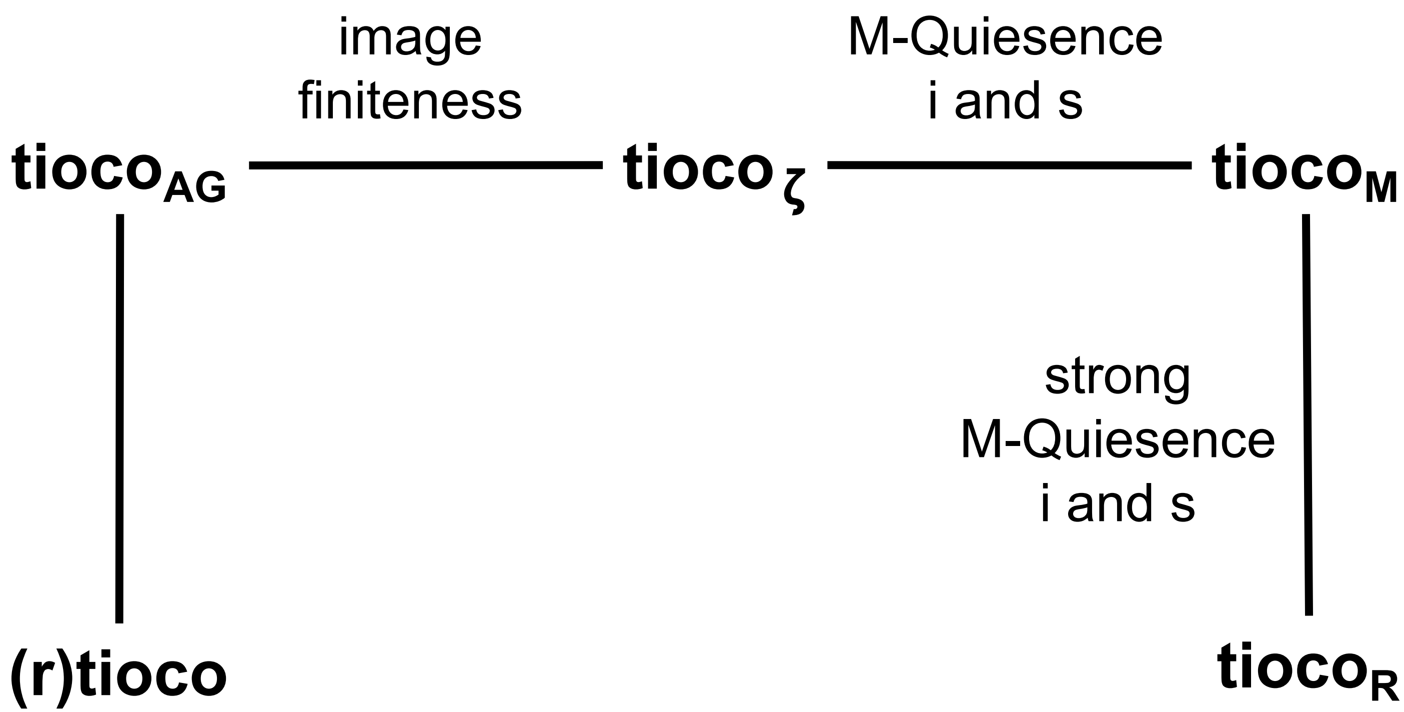
Test generated and executed event-by-event on-the-fly

Does I_n (rt-ioco) conform-to S_1 ?



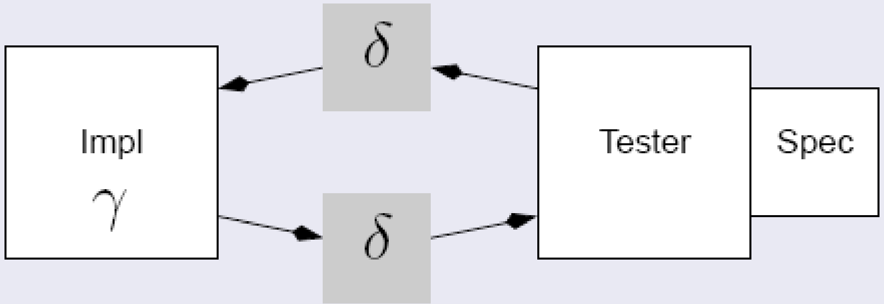
- Further work on timed conformance:
  - Conformance wrt. quiescence:
    - tioco_{ζ}: Unbounded delays are observable
    - tioco_{M}: Delays up to M are observable
    - tioco_{R}: Delays up to M and tau-moves are observable

Timed Conformance



Quantitative Testing Theory

The General Picture for Quantitative Systems



- δ: noise in communication
- γ: inherent imprecision due to faulty implementation

Results:

- **A new conformance relation defining the 'distance' from Spec**
- **Sketch of algorithms for testing the 'distance'**

Quantitative Testing Theory

$cof?(150\text{ml})$

$cof!(150\text{ml})$

$espr?(37\text{ml})$

Quantitative Testing Theory

cof?(150ml)

cof!(150ml)

espr?(37ml)

espr!(36.5ml)

WP5: Case Studies, Tools, Dissemination and Exploitation

- **Leader: Jan Tretmans**
  - How to evaluate and disseminate research
    - Case Studies
    - Integration of tool components into industrial tools or tool chains
    - Continuous dissemination to industry
    - Demonstrating the applicability of our approach

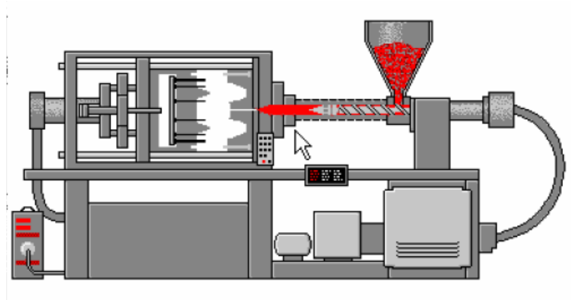
T5.1:
Case Studies

T5.2:
Tool Plug-ins and Tool Chain Integration.

T5.3:
Dissemination and Exploitation

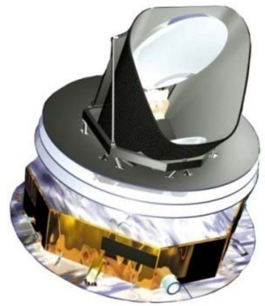
Main Achievements Y1
WP5 Case Studies

- **Accumulator Charge Controller** (HYDAC)
  - Simulink & Stateflow Models
  - Tool Chain: UPPAAL Tiga, Phaver, Simulink



- **Wireless Sensor Network** (CHESS)
  - gMAC protocol analyzed using UPPAAL (minumum waiting time for synchr.) and MODEST (prob. Of collision rates)

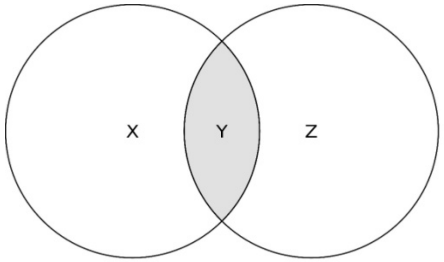
- **Control Software for satellites Hershel and Planck** (TERMA)
  - Recently started. A UPPAAL for schedulability analysis partially complete.



- **Self-Balancing Scooter** (CHESS)

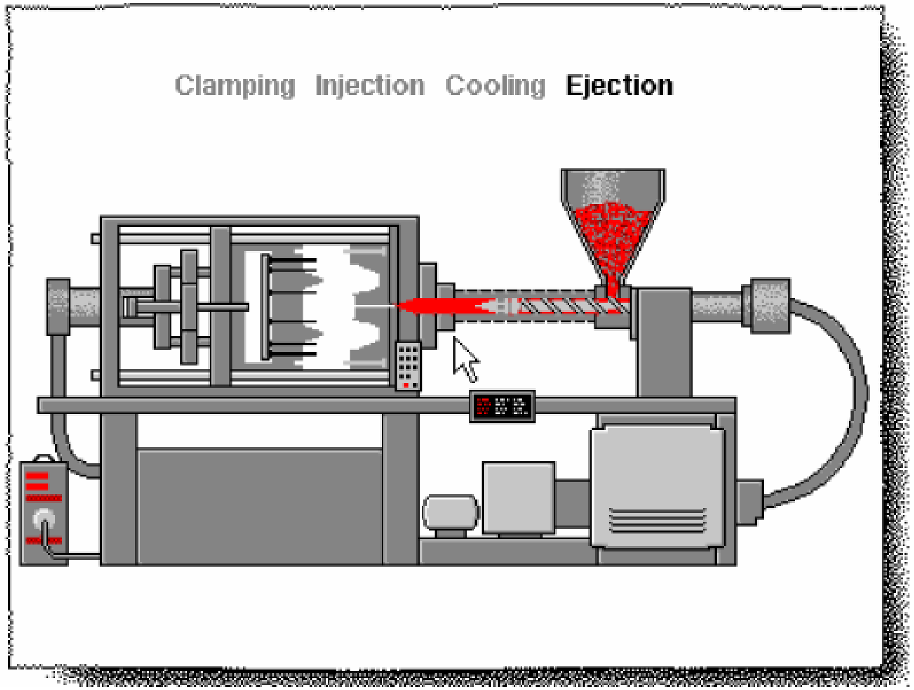
CHESS

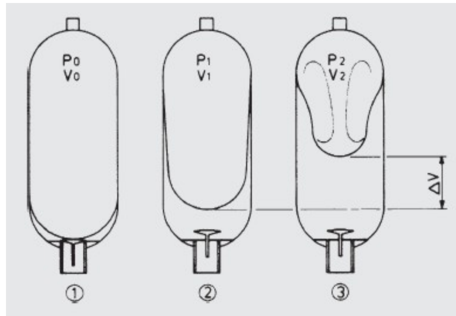
- Wireless Sensor Network (CHESS)
  - gMAC protocol analyzed using UPPAAL (minumum waiting time for synchr.) and MODEST (prob. Of collision rates)

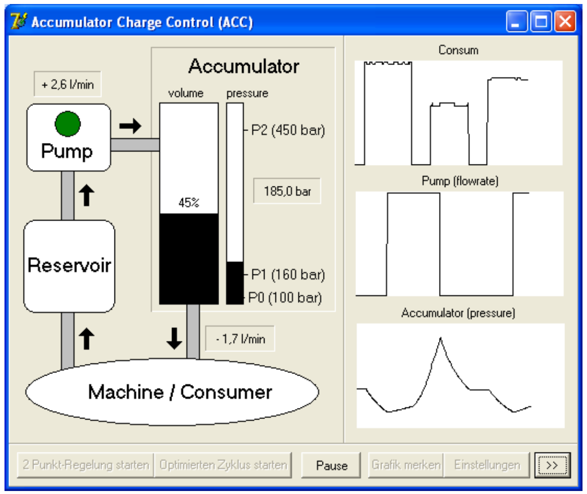


Hydac

- Accumulator Charge Controller (HYDAC)
  - Simulink & Stateflow Models
  - Tool Chain: UPPAAL Tiga, Phaver, Simulink

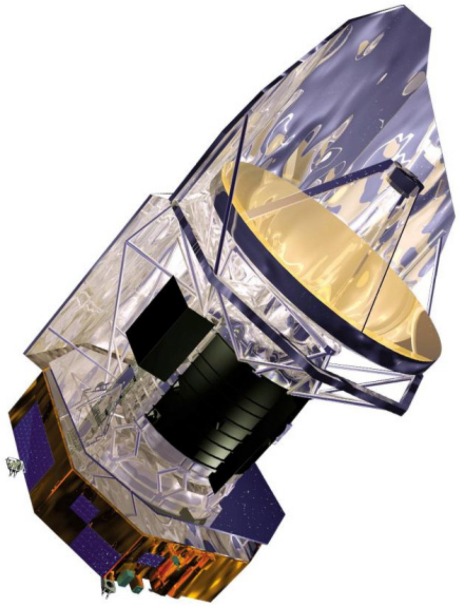


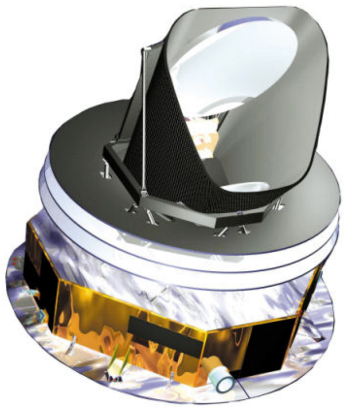




Terma

- Control Software for satellites Hershel and Planck (TERMA)
  - Recently started. A UPPAAL for schedulability analysis partially complete.

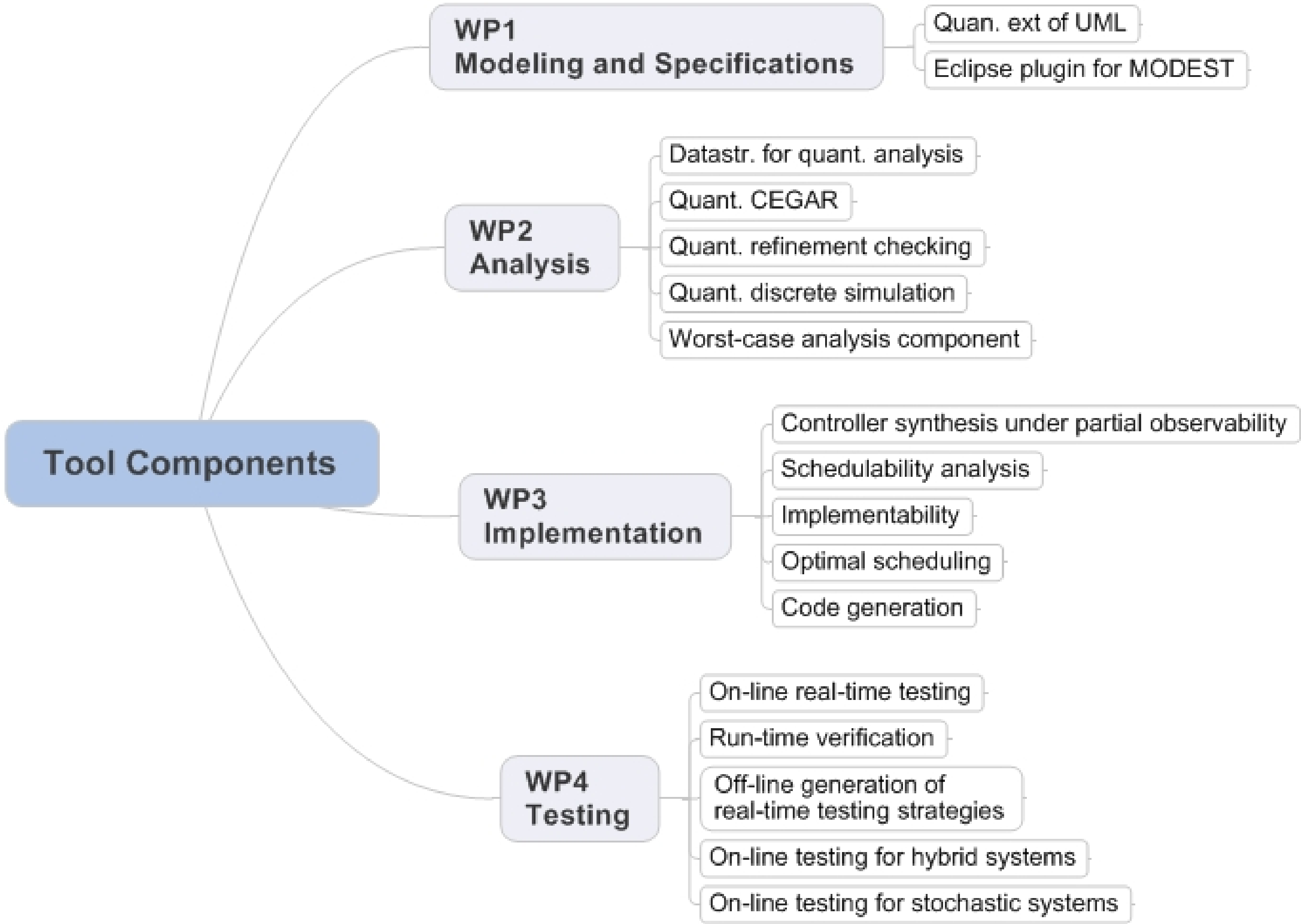




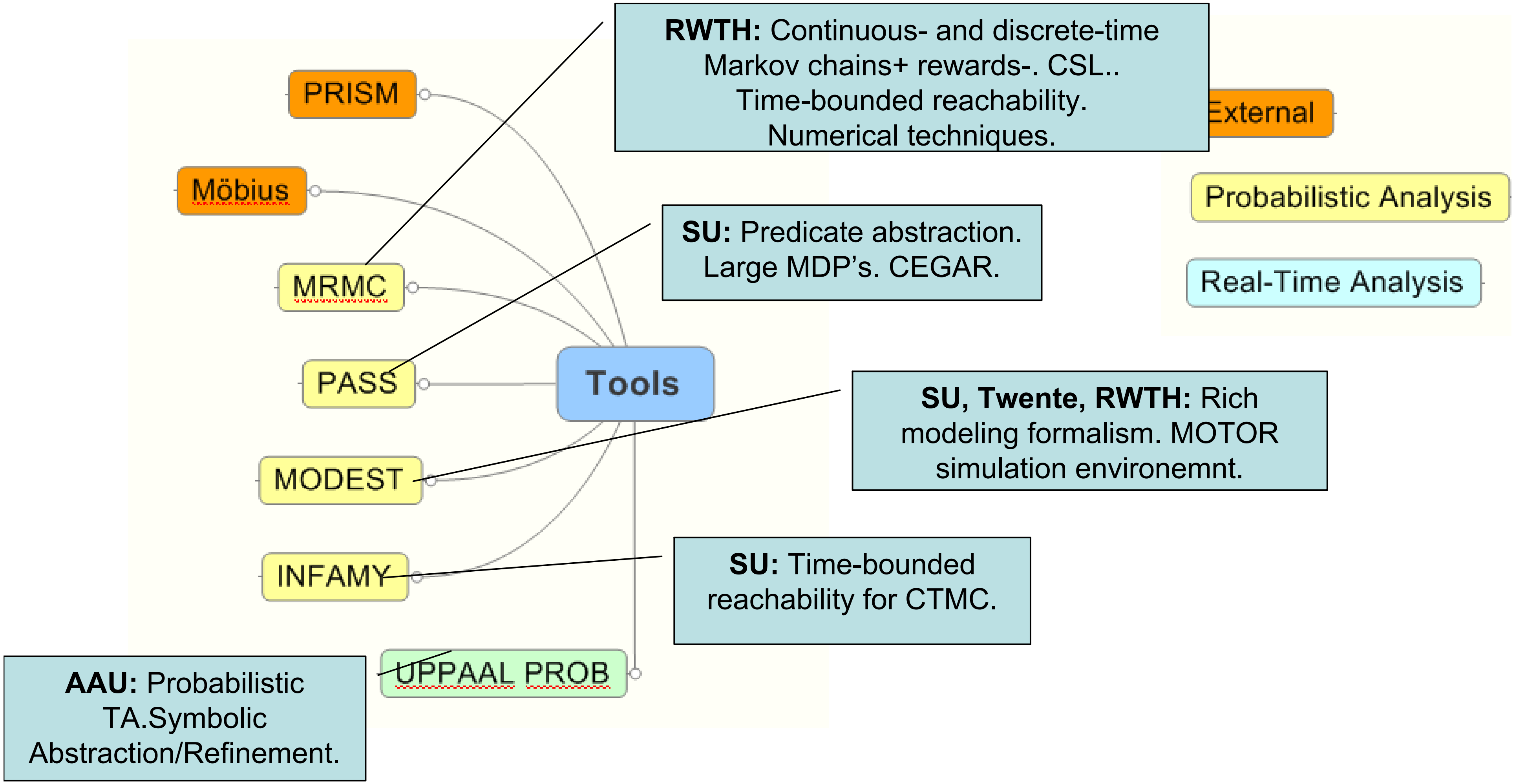
Chess

- Self-Balancing Scooter (CHESS)

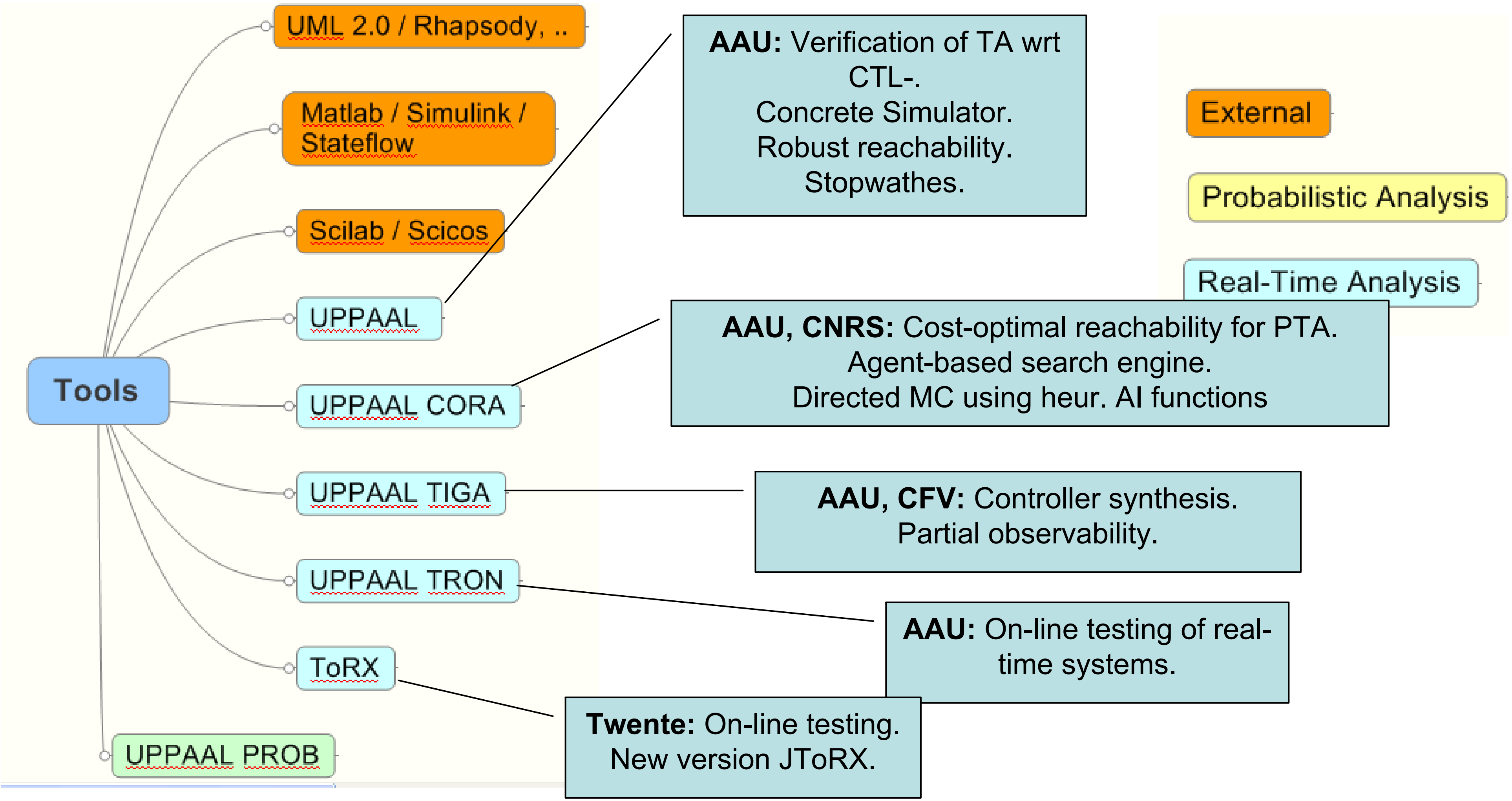
Tool Components



Tools for Probabilistic and Stochastic Model Checking

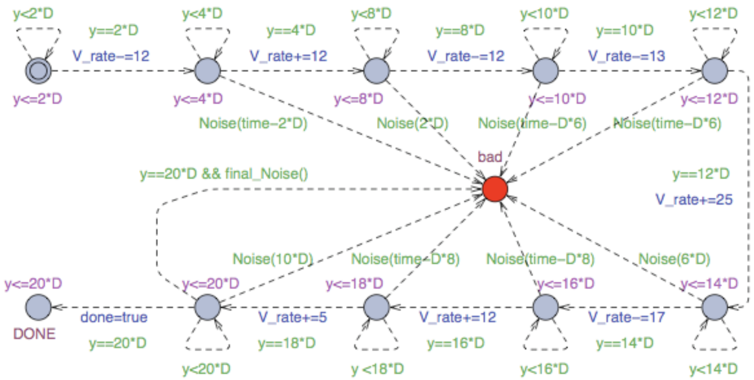


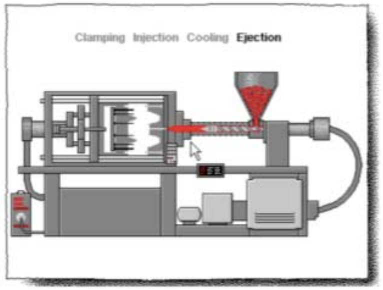
Tools for Real-Time Analysis

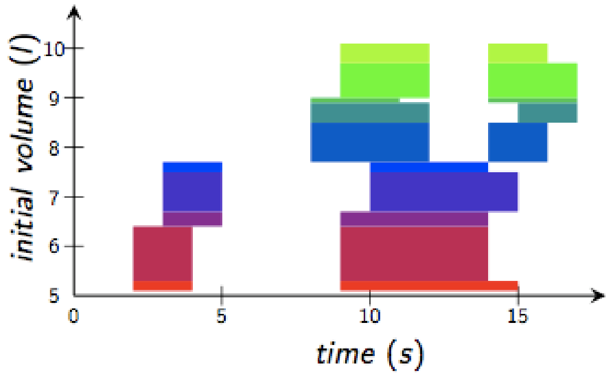


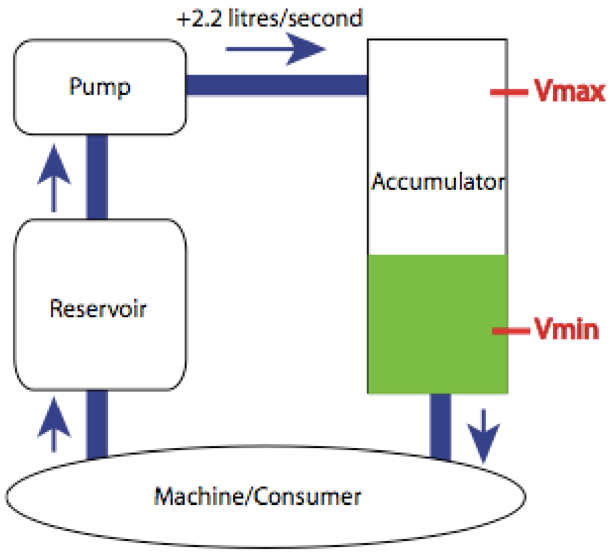
WP5-T1: Case Studies
HYDAC: Accumulator Charge Controller

Accumulator Charge Controller (HYDAC)





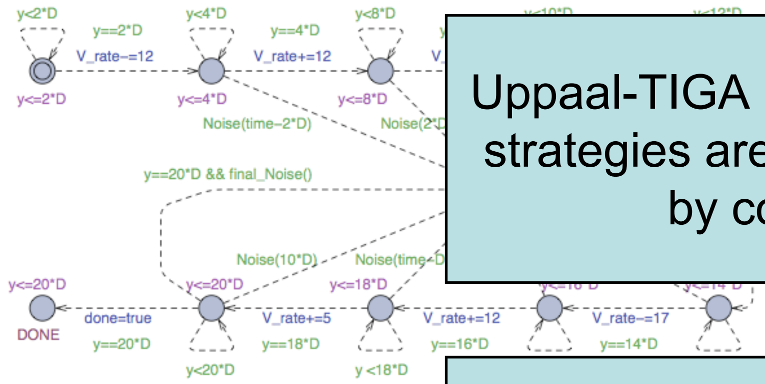


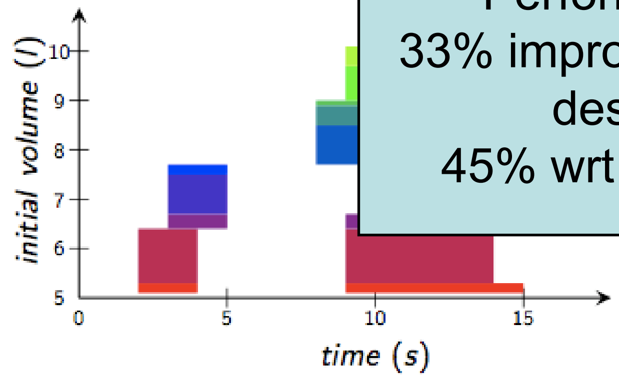


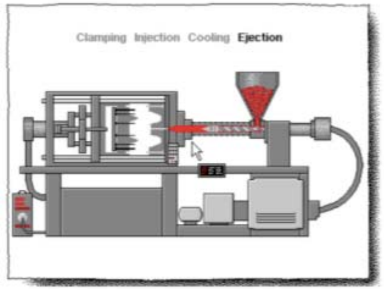
WP5-T1: Case Studies
HYDAC: Accumulator Charge Controller

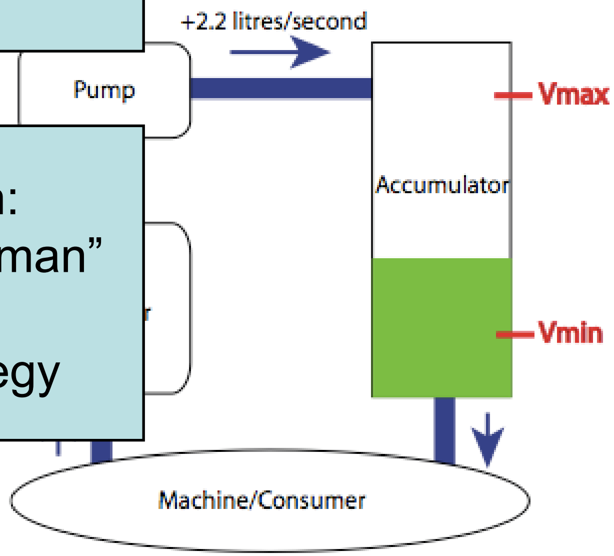
Quasimodo

Accumulator Charge Controller (HYDAC)









Uppaal-TIGA controller synthesis: strategies are correct and robust by construction

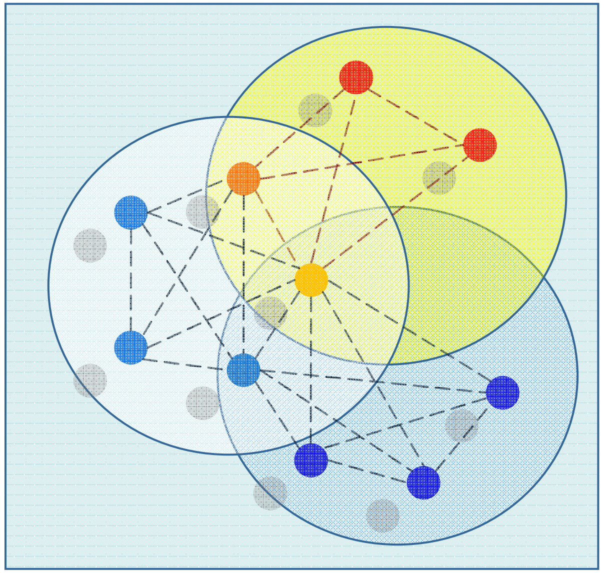
Performance simulation:
33% improvement over "human" designed strategy
45% wrt bang-bang strategy

WP5-T1: Case Studies
HYDAC: Accumulator Charge Controller

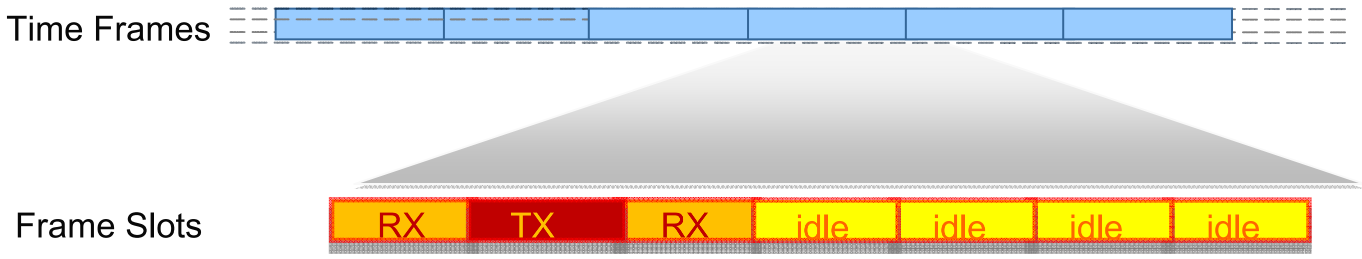
- Timed-Gamed Automata (UPPAAL-Tiga)
  - controller synthesis
  - robust (unsettled for HYDAC controllers yet)
- PHAVer
  - analysis and verification
  - HYDAC controllers are safe: pressure within margins
- Simulink and Stateflow
  - simulation, experimental validation (test of model, no proof):
    - HYDAC Smart Control uses less energy than Bang Bang Control
  - performance of synthesized controllers provide improvement
    - over HYDAC Smart Control: 33%; over Bang Bang Control: 45%

Uppaal modeling

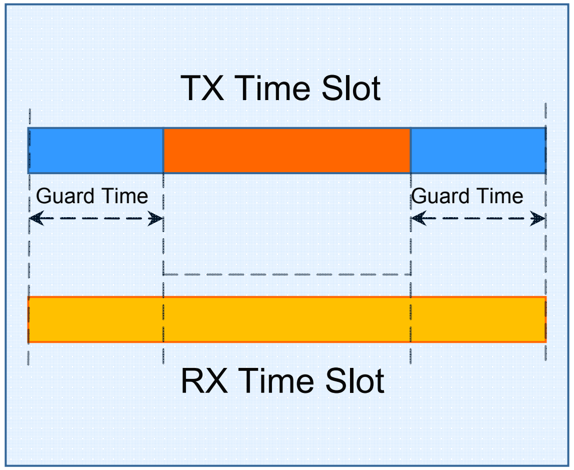
- Spatially distributed autonomous nodes
- Mobility
- Clock synchronization
  - synchronization in gMAC protocol
- Different kinds of applications



WP5-T1: Case Studies
CHESS: Wireless Sensor Network

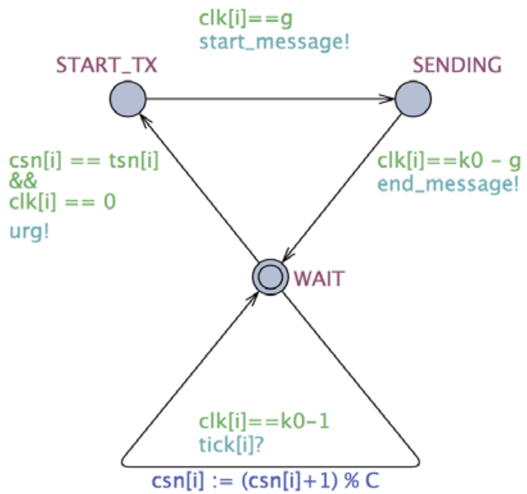


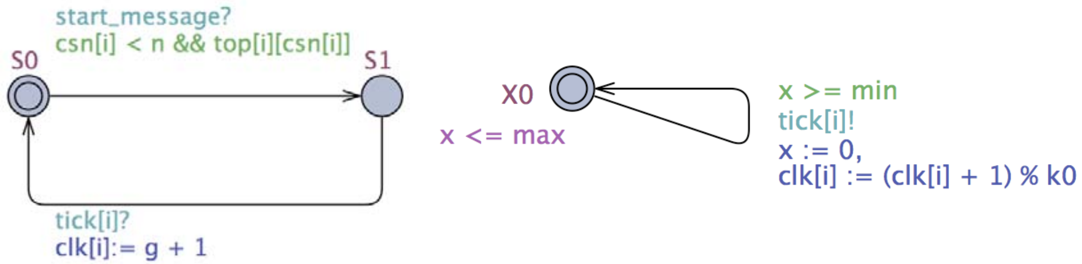
- Each node owns a slot to send (TX)
- Each node listens to other slots (RX)
- In TX slot: sender waits guard time *g* before and after sending a message to allow for synchronization



Which *g* ?
critical for functional correctness
and for energy consumption

WP5-T1: Case Studies
CHESS: Wireless Sensor Network





$$\frac{(C-d)k_0 - g}{(C-d)k_0 - 1} < \frac{\min}{\max} \qquad (d = tsn_{\max} - tsn_{\min})$$

- UPPAAL models and formalization of network synchronization:
  if node n is sending in slot s, all other nodes are also in slot s:

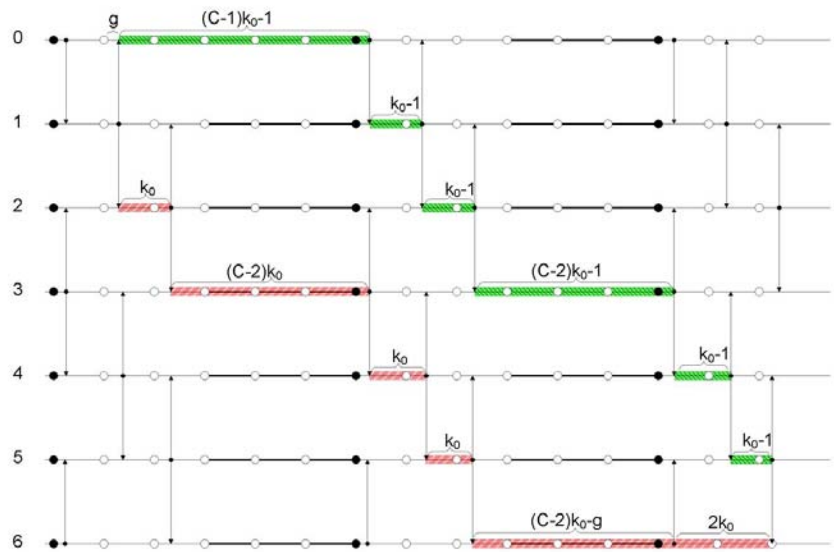
A[] forall(i:Nodes) forall(j:Nodes) (WSN(i).SENDING imply csn[i] = csn[j])

- Uppaal proves whether *g* leads to a synchronized network
- Generalized to formula for synchronization of connected networks

WP5-T1: Case Studies
CHESS: Wireless Sensor Network

another result:

every network will eventually fail with increasing #nodes



Future
- non-fully connected networks
- different topologies
- dynamic networks in UPPAAL

- because nodes get out of sync
- fixed guard time, #active slots, clock drift, ...
- generalizes Fan&Lynch, Meier&Thiele with decreasing clocks
- result of mathematical nature, manually proven; Uppaal used to generate counter-examples that were generalized
- practical implications?

WP5-T1: Case Studies
CHESS: Wireless Sensor Network

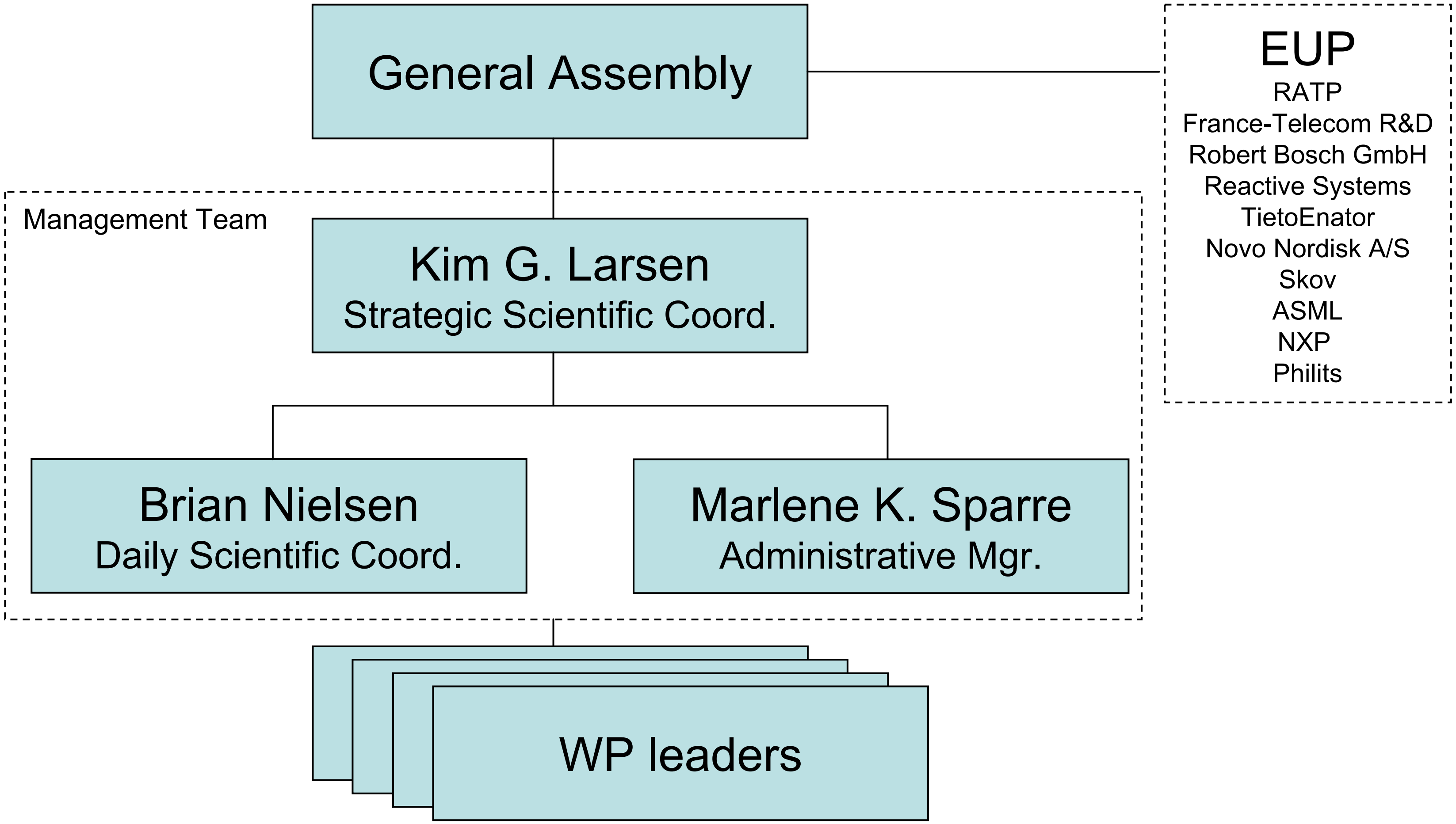
MoDeST modeling

- discrete event simulation
  - vary #active slots (now synchronization, and others, assumed)
  - determine probability of collisions vs. #active slots
  - analysis of effectiveness of collision detection mechanism
  - how this affects performance and energy consumption
- results

$$\begin{tikzcd} \text{\#active slots} \arrow[r, leftrightarrow] \arrow[d, leftrightarrow] & \text{\#collisions} \arrow[d, leftrightarrow] \\ \text{energy consumption} \arrow[r, leftrightarrow] & \text{latency} \end{tikzcd}$$

WP0: Management

- Efficient, flexible and reactive project management



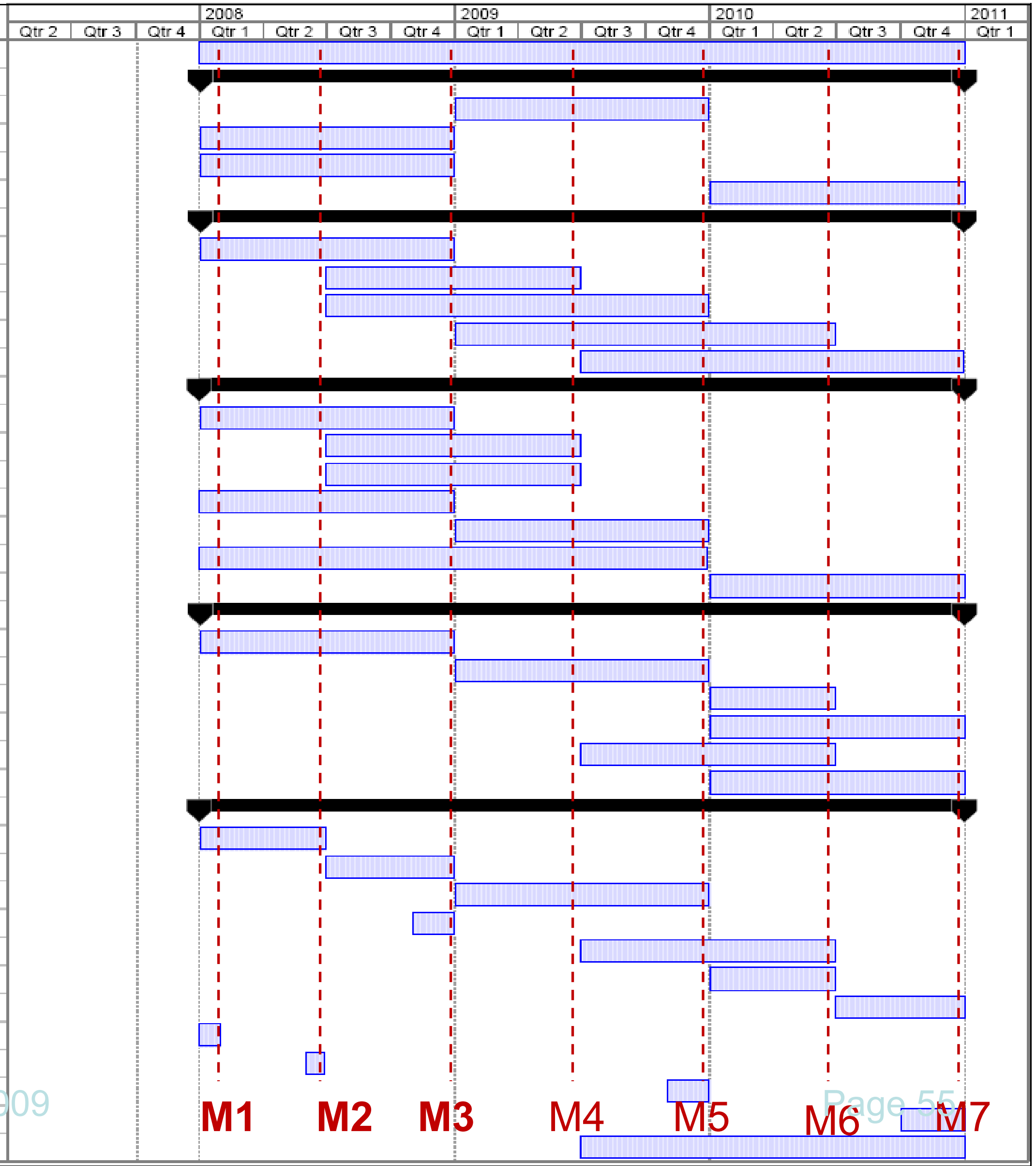
Resources

- 1.995 M€ EC contribution of a total 2.6 M€
- ~7.5 person year/year

	WP0	WP1	WP2	WP3	WP4	WP5	Total
AAU	18	2	6	6	6	8	46
ESI	0	16	11	6	14	25	72
CNRS	0	0	8	13	0	7	28
RWTH	0	4	14	0	0	10	28
SU	0	5	11	0	4	8	28
CFV	0	0	4	20	0	4	28
Terma	0	2	0	0	3	5	10
Chess	0	2	0	0	0	8	10
Inchron	0	1	0	2	2	5	10
Hydac	0	2	0	0	2	6	10
Total	18	34	54	47	31	86	270

Schedule

ID	Task Name
1	**WP0: Management**
2	**WP1: Modelling and specification**
3	T 1.1, D 1.3: Model process improvement
4	T 1.2, D 1.1: Modelling quantitative system aspects
5	T 1.3, D 1.2: Design notations
6	T 1.3, D1.4: Modelling tools
7	**WP2: Analysis**
8	T 2.1, D 2.1: Model checking real-time probabilistic models
9	T 2.1, D 2.2: Symbolic data structures and analysis of multiple quantitative models
10	T 2.2, D 2.3: Abstraction
11	T 2.2, D 2.4: Abstraction-refinement
12	T 2.3, D 2.5: Approximate analysis
13	**WP3: Implementation**
14	T 3.1, D 3.1:Transfer of correctness properties from model to implementation
15	T 3.1, D 3.2: Tool for implementability checking
16	T 3.1, D 3.5: Extended timed automata for scheduling
17	T 3.2, D 3.3: Model checking of controllability properties
18	T 3.2, D 3.4: Synthesizing controllers with bounded resources
19	T 3.2, D 3.6: Code generation from untimed specifications
20	T 3.2, D 3.7: Code generation from timed specifications
21	**WP4:Testing**
22	T 4.1, D 4.1: Quantitative testing theory
23	T 4.1, D 4.2: Algorithms for off- and on-line quantitative testing
24	T 4.1, D 4.3: Test selection and coverage
25	T 4.1, D 4.5: Final algorithms and evaluation
26	T 4.1, D 4.6: On-line hybrid/stochastic testing
27	T4.2, D 4.4 Approximate testing
28	**WP5: Coherence, Application, Dessimination and Exploitation**
29	T 5.1, D 5.2 Preliminary description of case studies
30	T 5.1, D 5.5: Case studies: models
31	T 5.1, D 5.7: Case studies: validation
32	T 5.2, D 5.4 Plan for integration of tool components
33	T 5.2, D 5.8: Tool components and tool integration
34	T 5.2, D 5.9: Tool components
35	T 5.1, T 5.2, D 5.10: Final report on case studies and tool integration
36	T 5.3, D 5.1: Quasimodo web site
37	T 5.3, D 5.3: Dissemination and use plan
38	T 5.3, D 5.6: Dissemination and exploitation
39	T 5.3, D 5.11: Final report on dissemination and exploitation
40	T 5.3, D 5.12: Industrial handbook



Quasimodo

Progress for Y1

s

Kim G. Larsen & Brian Nielsen
Aalborg University, DK

Objectives Y1

- Consolidate quantitative modeling formalisms.
  - Theoretical foundation
  - Precise semantics (QLTS)
  - Formal refinement relations
- Industrial Case Studies.
  - Initial Descriptions
  - First Models.
- A first iteration of research loop:
  - Application→Theory →Tools→Application

Deliverables Y1

No	Deliverable name	Due Date
D1.1	Modeling quantitative system aspects	**12**
D1.2	Design notations	**12**
D2.1	Model checking real-time probabilistic models	**12**
D3.1	Transfer of correctness properties from model to implementation	**12**
D3.3	Model checking of controllability properties	**12**
D4.1	Quantitative testing theory	**12**
D5.1	Quasimodo website	**1**
D5.2	Preliminary description of case studies	**6**
D5.3	Dissemination and use plan	**6**
D5.4	Plan for integration of tool components	**12**
D5.5	Case studies: models	**12**

Milestones Y1

No.	Name	Date	Means of verification (Check Availability of:)
M1	Project Start	1	Kick-off Meeting
M2	Definition phase	6	1. Precise descriptions of case studies. 2. Plan for tool components and their integration in industrial tool chain.
M3	Modeling formalisms	12	1. Semantic foundation of quantitative models in terms of labelled transition systems including semantics of composition of models, refinements between models. 2. Formal definition of conformance and robustness between quantitative models and implementations. 3. First models of case studies. 4. Quantitative extensions identified by the needs of case studies.

Main Achievements Y1
WP1 Modeling

- Quantitative Modeling Formalisms
  - Extensive use of Timed Automata in Industrial Case Studies
  - Probabilistic Timed Automata
  - Priced Timed Automata
  - Probabilistic Priced Timed Automata
- Design Notations
  - AADL → Probabilistic Automata
  - UML → Timed Automata
  - UML → Markov decision processes

Main Achievements Y1
WP2 Analysis

- Improved search engines for TA
  - Agent based | directed using Heur. Funct.
- Probabilistic Timed Automata
  - discretization | symbolic abstr./refin.
- Multi-Priced Timed Automata
- Refinement for Timed Automata as Games
- CEGAR for probabilistic programs
- Discrete-event simulation used in MRMC

Main Achievements Y1
WP3 Implementation

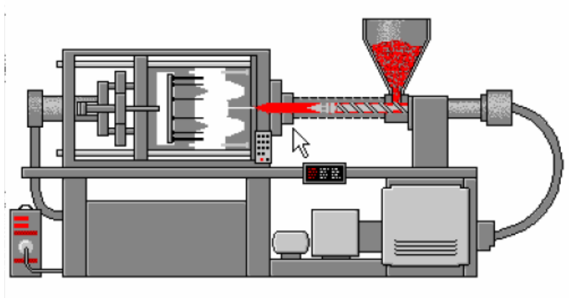
- Controller Synthesis
  - Fundamental results for ATL*
  - Synthesis for games with imperfect information. Implemented in UPPAAL-Tiga.
  - Application to HYDAC case study.
- Implementability & Code Generation
  - Robustness for Timed Automata and transfer for LTL. Implementation in UPPAAL.
  - New notion of robustness for Timed Automata based on probabilistic semantics

Main Achievements Y1
WP4 Testing

- Various timed ioco conformance relations defined and related.
- General quantitative ioco conformance relation provided.
- Advances on on-line testing tools for (timed) systems (ToRX, UPPAAL Tron).
- Off-line test generation as game problems using UPPAAL Tiga.

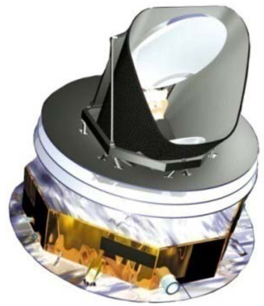
Main Achievements Y1
WP5 Case Studies

- **Accumulator Charge Controller** (HYDAC)
  - Simulink & Stateflow Models
  - Tool Chain: UPPAAL Tiga, Phaver, Simulink



- **Wireless Sensor Network** (CHESS)
  - gMAC protocol analyzed using UPPAAL (minumum waiting time for synchr.) and MODEST (prob. Of collision rates)

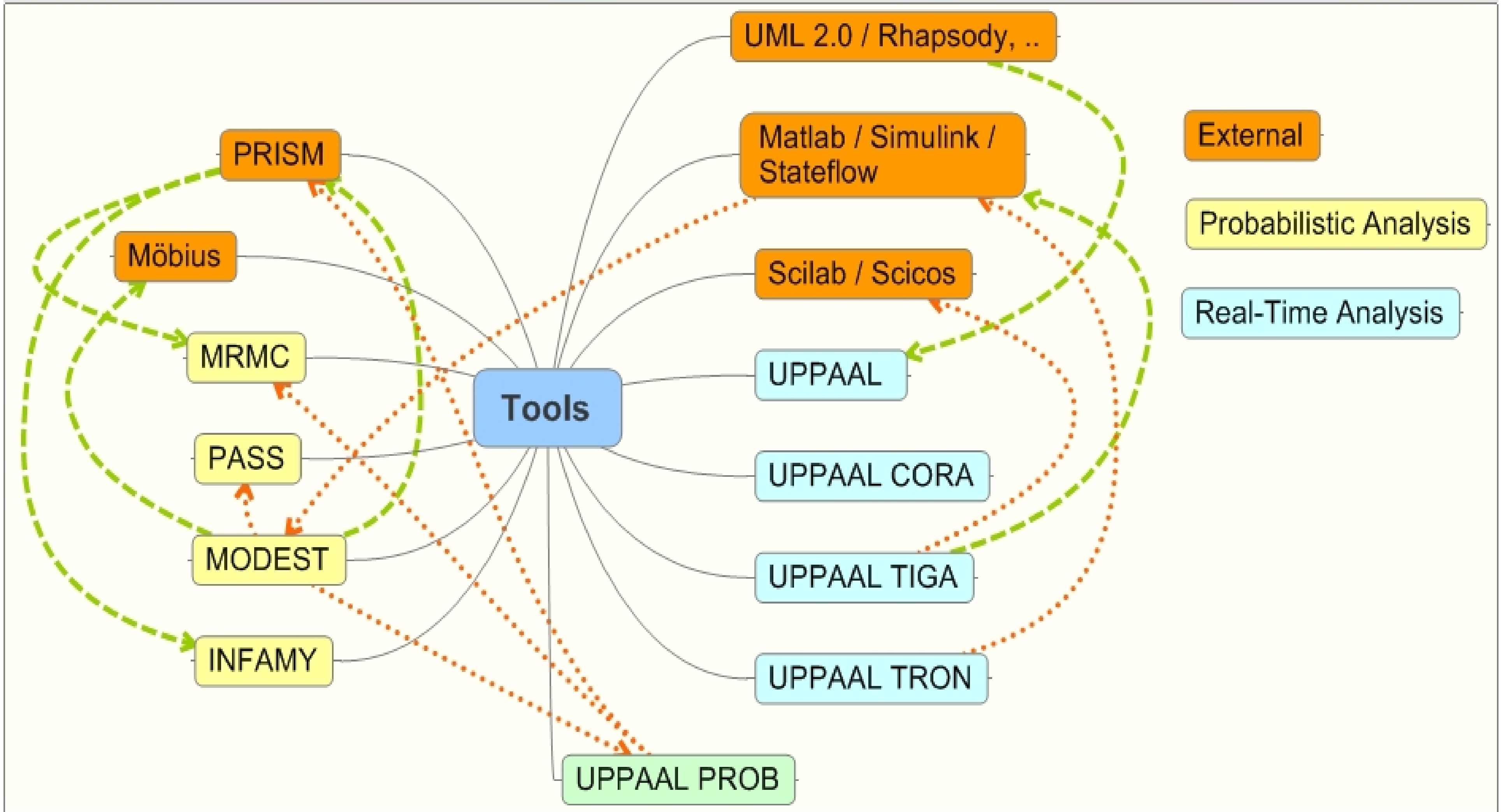
- **Control Software for satellites Hershel and Planck** (TERMA)
  - Recently started. A UPPAAL for schedulability analysis partially complete.



- **Self-Balancing Scooter** (CHESS)

Main Achievements Y1

WP5 Tools & Tool Integration



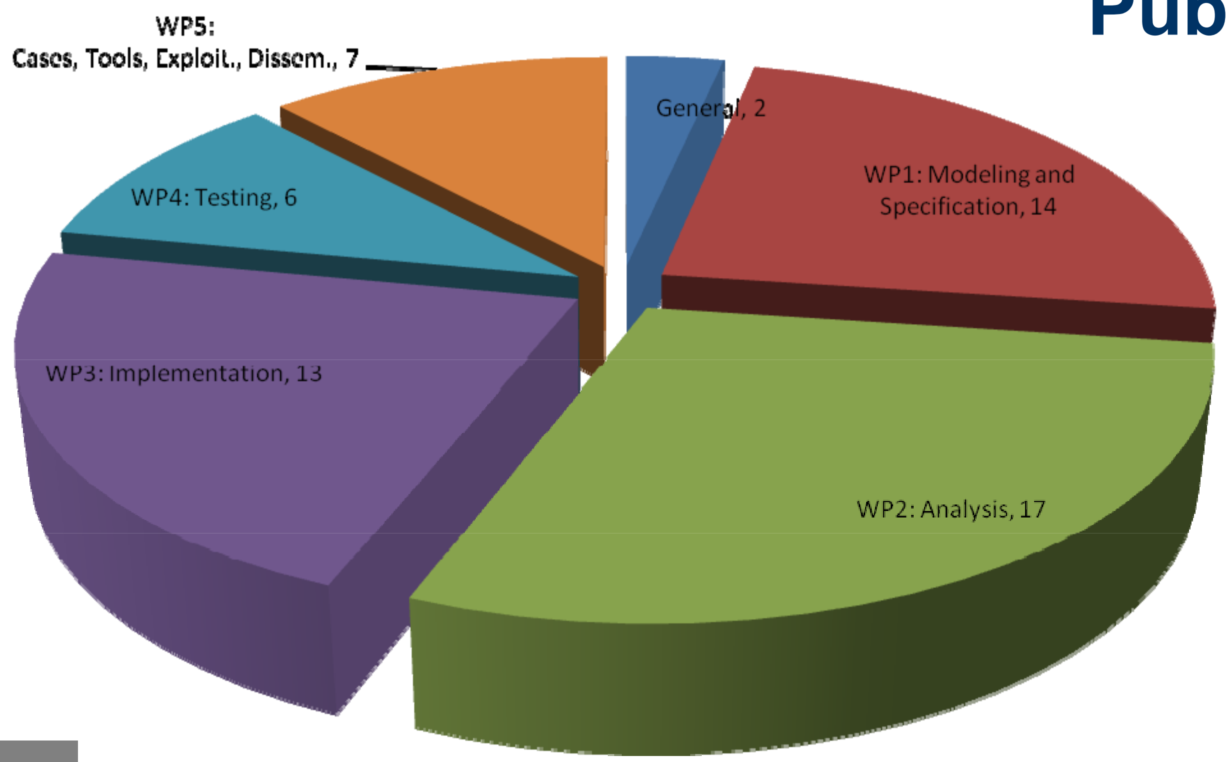
Main Achievements Y1

WP5 Tool Components

- Tool Components
  - WP1 Modeling and Specifications
    - Quan. ext of UML
    - Eclipse plugin for MODEST
  - WP2 Analysis
    - Datastr. for quant. analysis
    - Quant. CEGAR
    - Quant. refinement checking
    - Quant. discrete simulation
    - Worst-case analysis component
  - WP3 Implementation
    - Controller synthesis under partial observability
    - Schedulability analysis
    - Implementability
    - Optimal scheduling
    - Code generation
  - WP4 Testing
    - On-line real-time testing
    - Run-time verification
    - Off-line generation of real-time testing strategies
    - On-line testing for hybrid systems
    - On-line testing for stochastic systems

Main Achievements Y1
WP5 Dissemination

Publications



49 papers
107 authors
22 invited talks

Main Achievements Y1
WP5 Dissemination

- **Organisation & Contribtuion**
  - Conferences, Workshops,
  - PhD Schools, Courses
- **European Projects**
  - E.g. Design, FP7 NoE
- **National Projects**
- **Coming Events**
  - FM Week Industry day at Nov. 2009, Eindhoven
  - QUANTLOG 2009 - Workshop on Quantitative Logics. Rhodes, Greece, July 5-12 at ICALP 2009
  - GASICS, Workshop on Games for Design, Verification and Synthesis. at CAV 2009, Grenoble, June 26-July 2
  - Quantitative Model Checking, PhD School, Copenhagen, December 2009 (with ARTIST Design)
  - Quantitative Models: Expressivenss & Analysis, Dagstuhl Workshop, January 17-22, 2010.