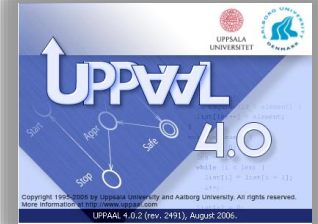


Validation and Performance Analysis of CPS in UPPAAL

Kim G. Larsen
Aalborg University, DENMARK



Cyber Physical Systems

OPTIMAL:

finding a **controller** for a given system such that a certain optimality criterion is achieved.

Discrete

Real Time

Resources

Stochasticity

Hybrid

DEPENDABLE:

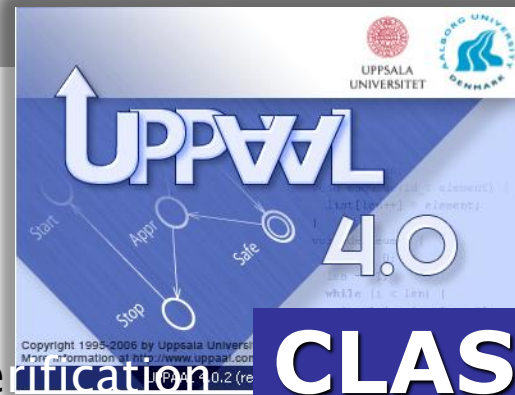
the ability of a **controller** to function (correct) under stated conditions for a specified period of time.



d IoT ng

Cyber-Physical Systems

UPPAAL Tool Suite



1995

CLASSIC

Verification

Optimization

CORA

2001

Testing

TRON

2004

Synthesis

TIGA

2005

Component

ECDAR

2010

Performance Analysis

SMC

2011

2014

Optimal Synthesis

STRATEGO

The image displays several screenshots of the UPPAAL tool suite. The top-left screenshot shows a state transition diagram for a 'Train' model with states like 'Safe', 'Appr', 'Cross', and 'Stop', and transitions labeled with guard conditions and actions. The top-right screenshot shows a multi-view simulation interface with several small diagrams for different train instances (Train(0) to Train(5)) and a 'Gate' component. The middle-left screenshot shows simulation results including a histogram of 'stop[id]?' values and a cumulative probability curve. The bottom screenshot shows a code editor with a query: `A[] forall (i : id_t) forall (j : id_t) Train(i).Cross && Train(j).Cross imply i == j` and a comment: 'There is never more than one train crossing the bridge (at any time instance)'. The interface includes buttons for 'Check', 'Insert', 'Remove', and 'Comments'.

Timed Automata

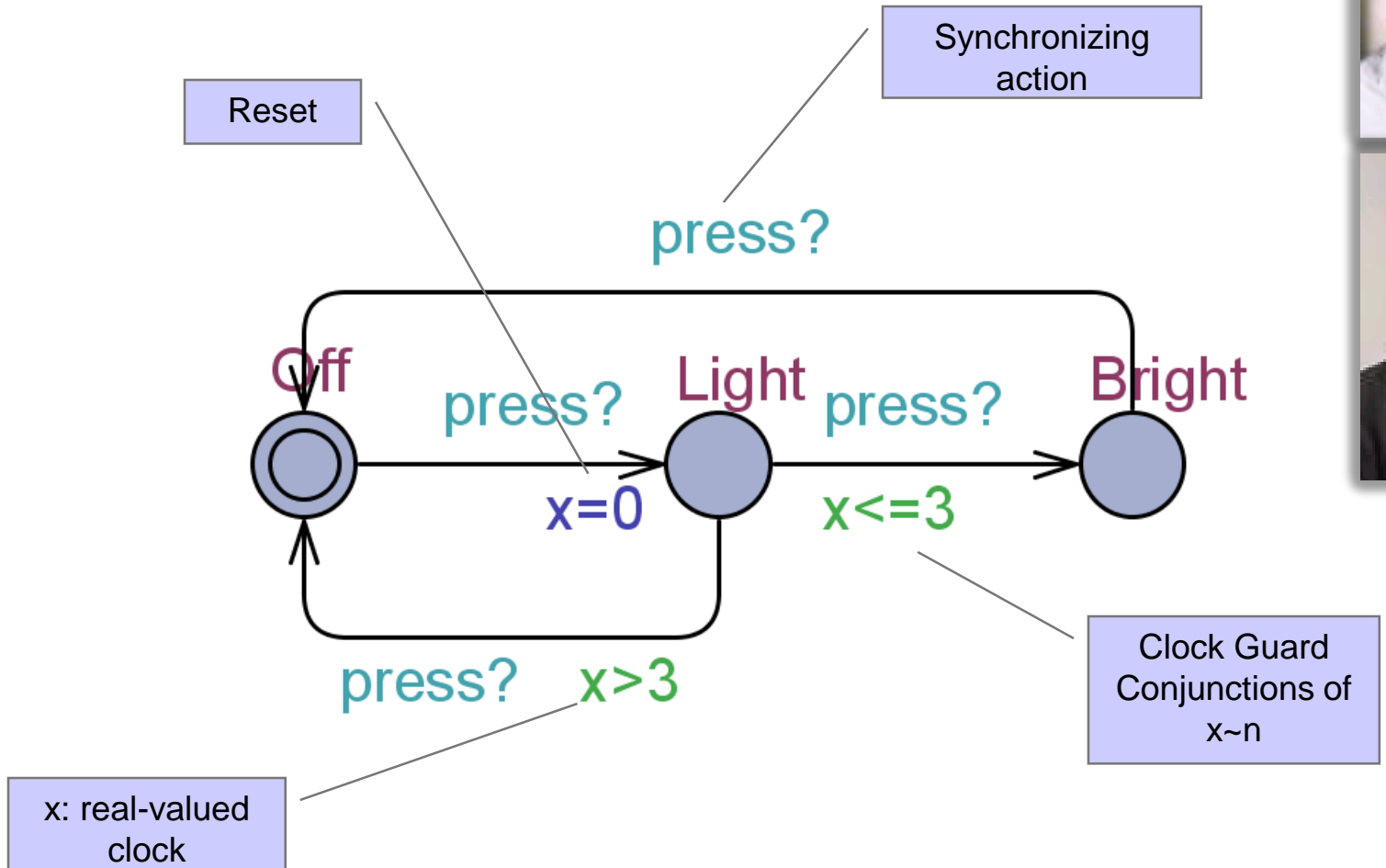
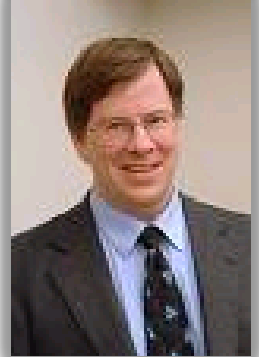


AALBORG UNIVERSITET



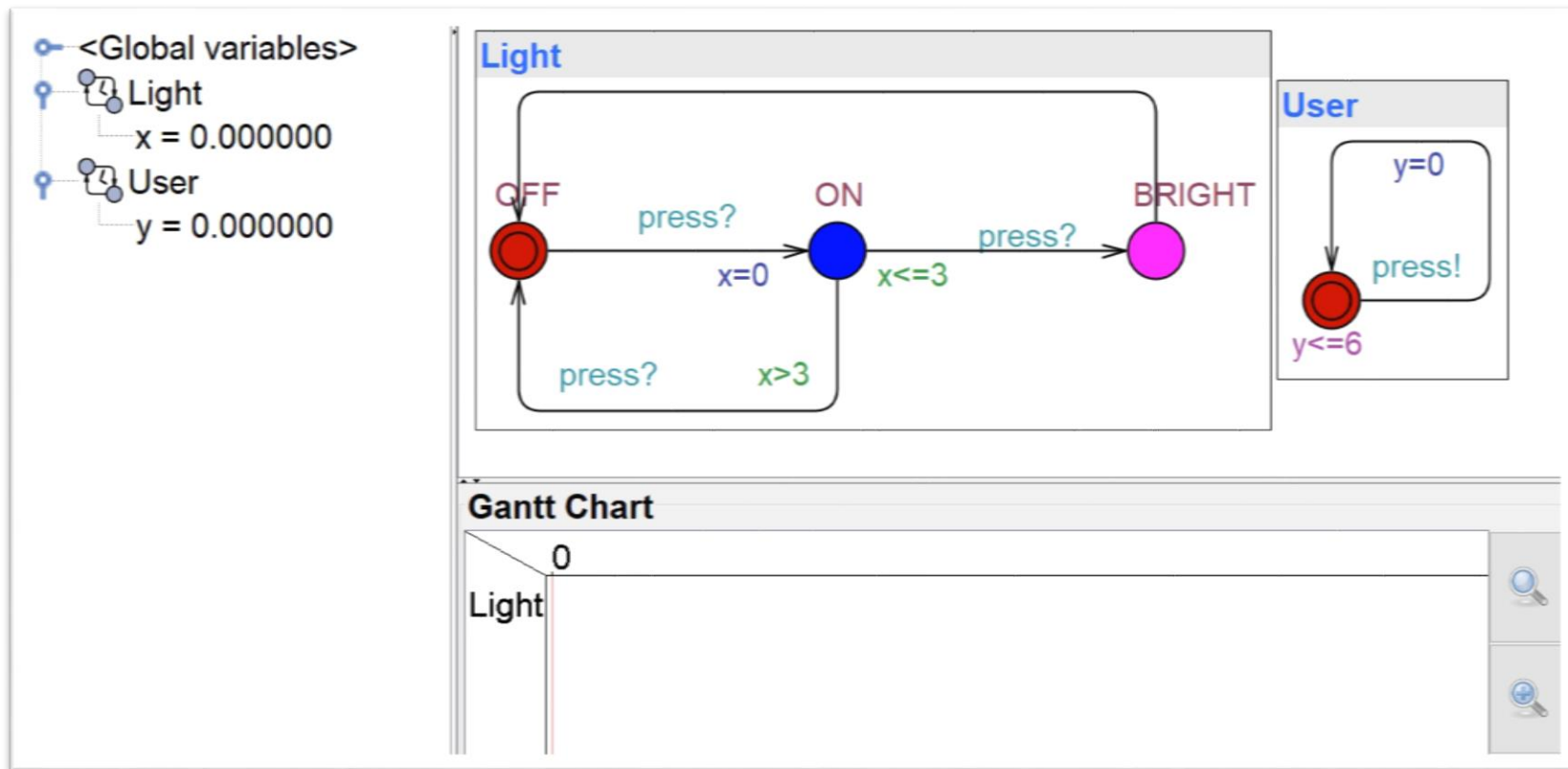
Timed Automata

[Alur & Dill'89]

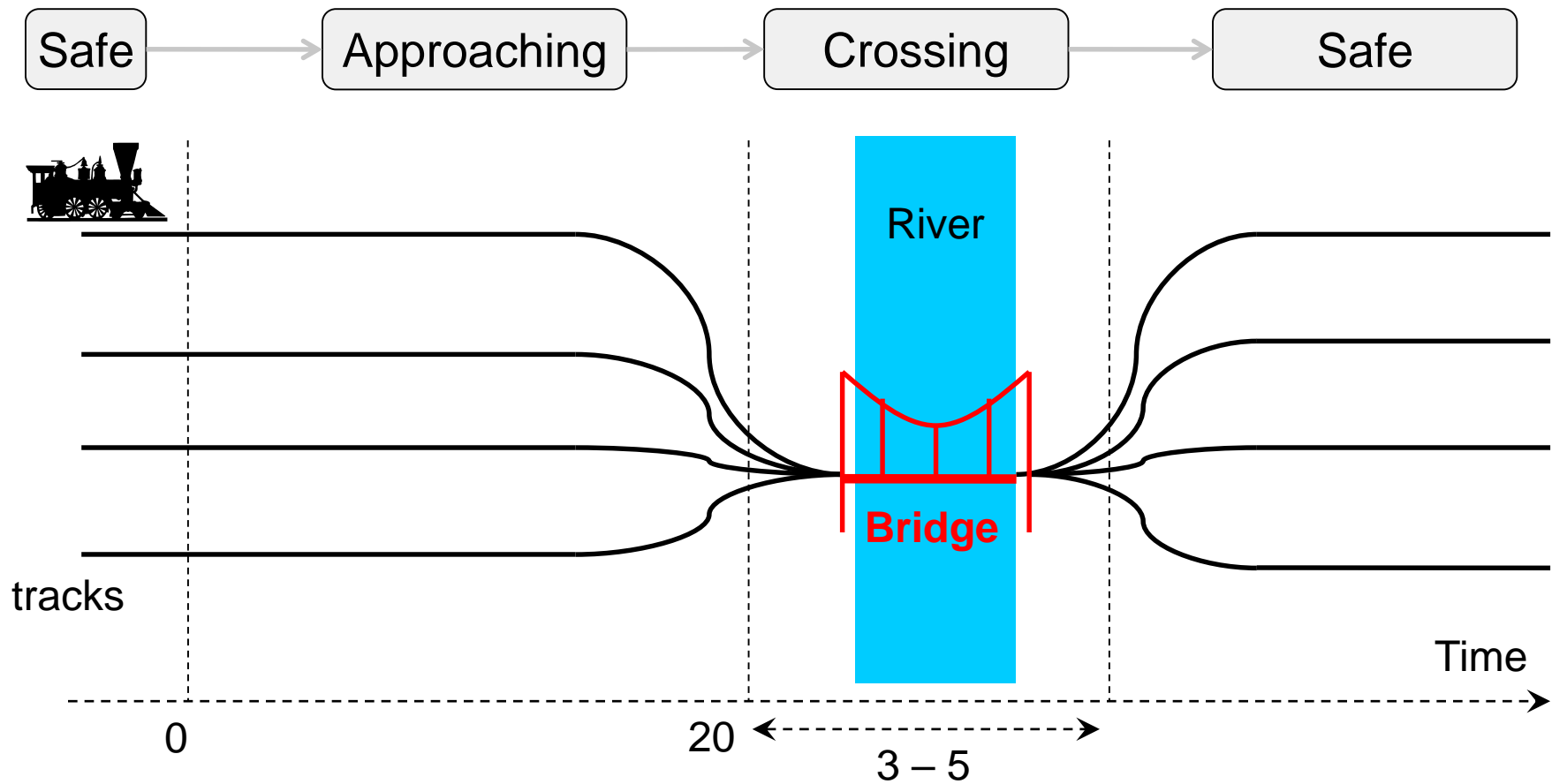


ADD a clock x

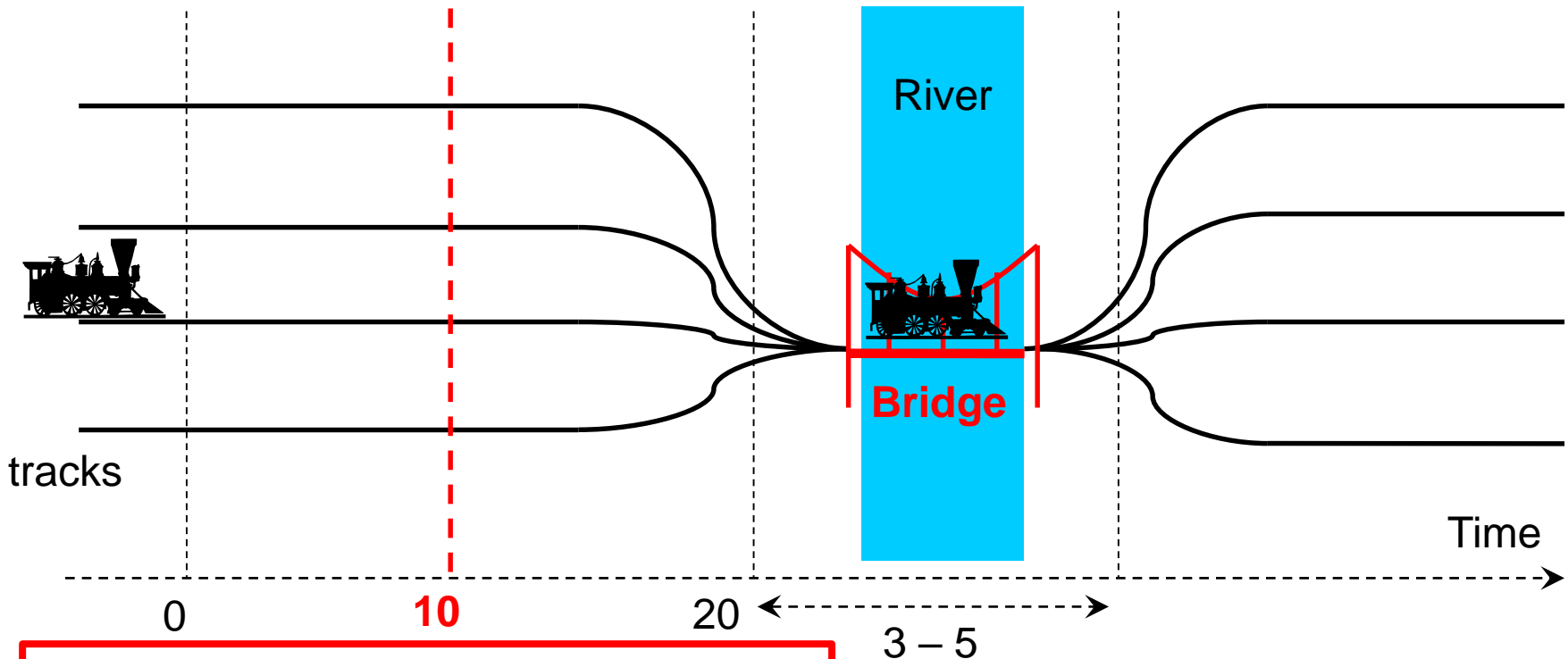
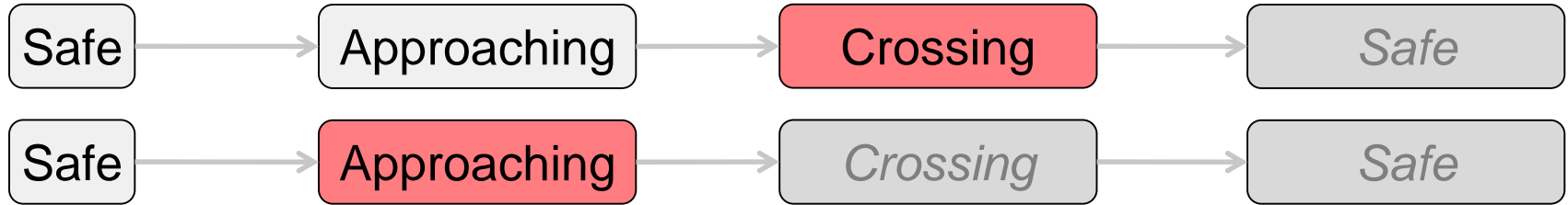
Semantics in UPPAAL



Train Crossing

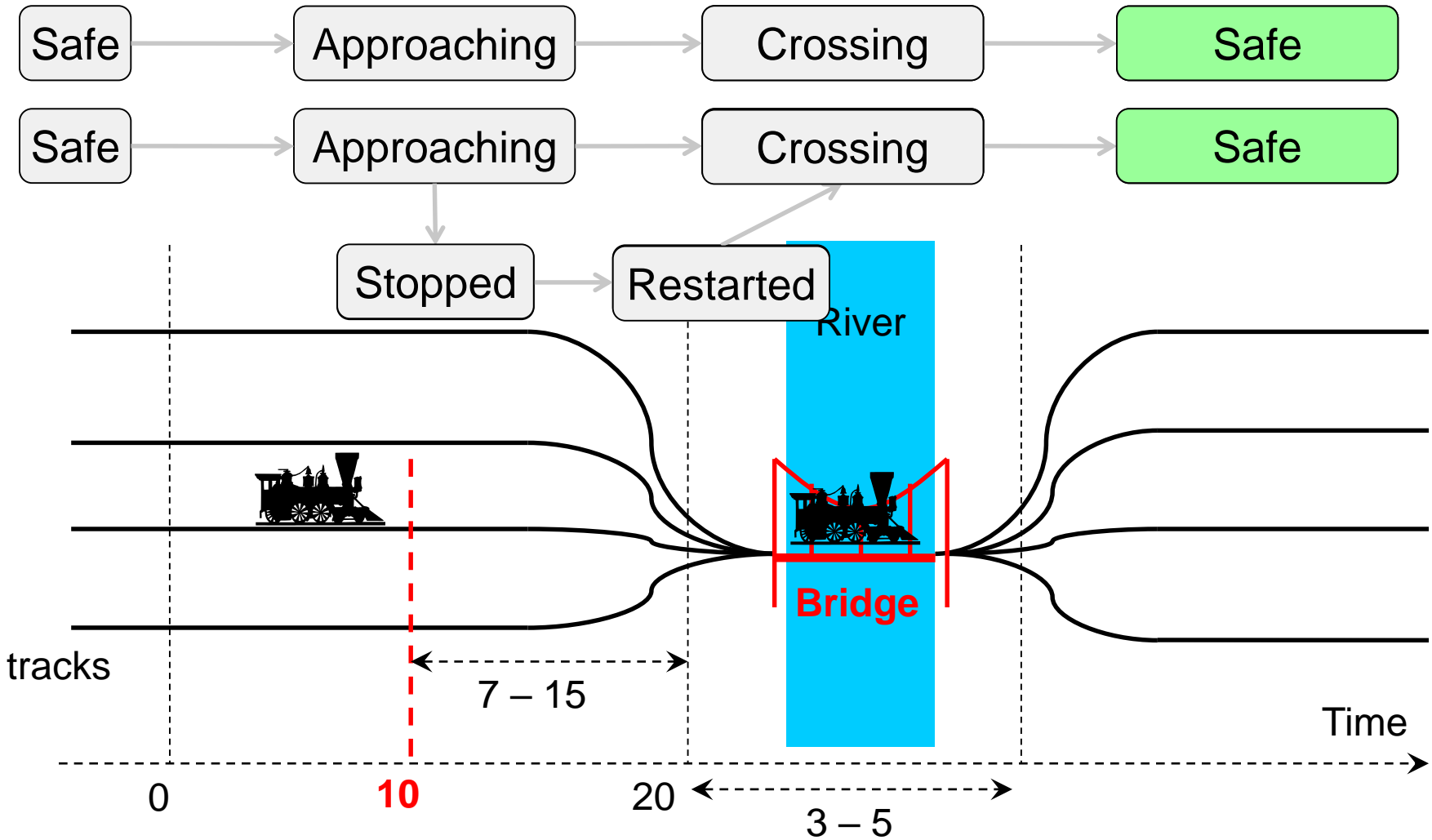


Train Crossing

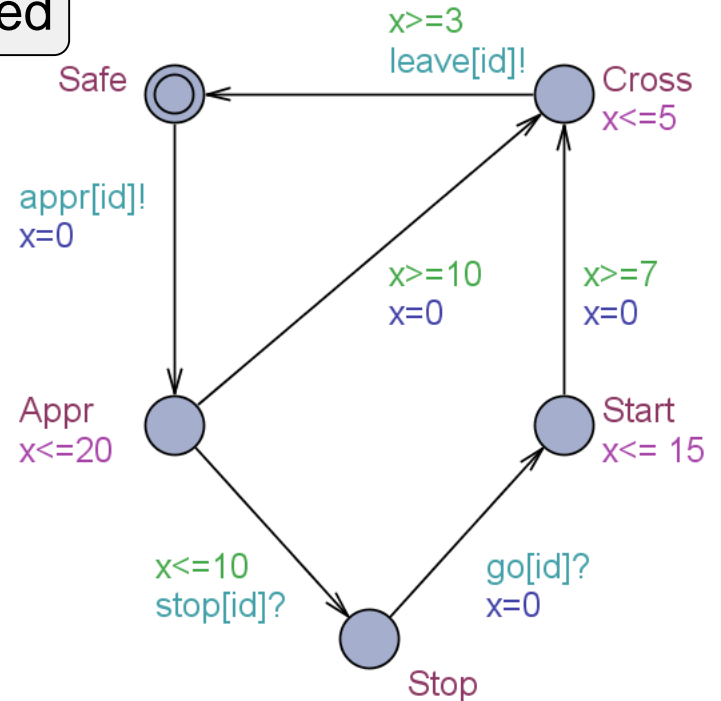
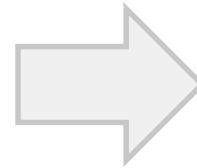
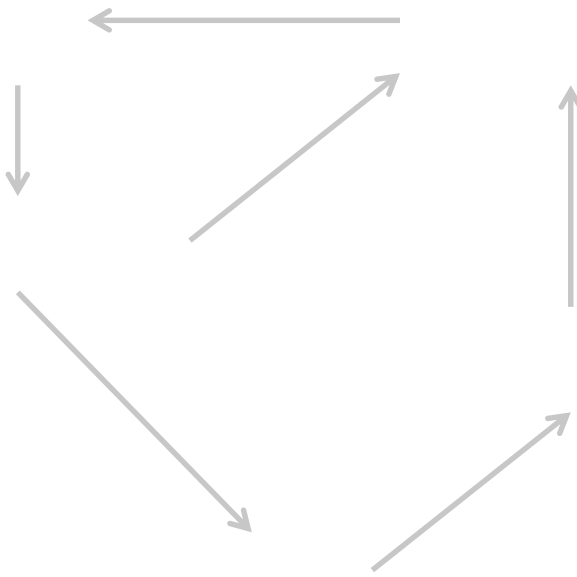
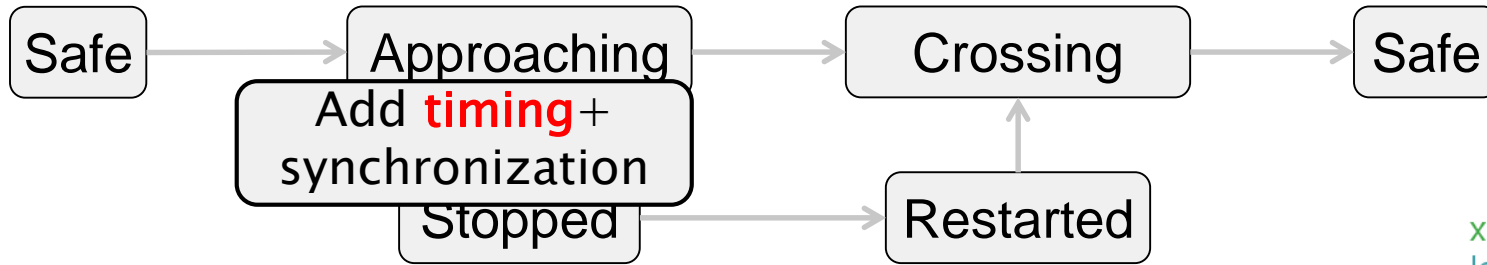


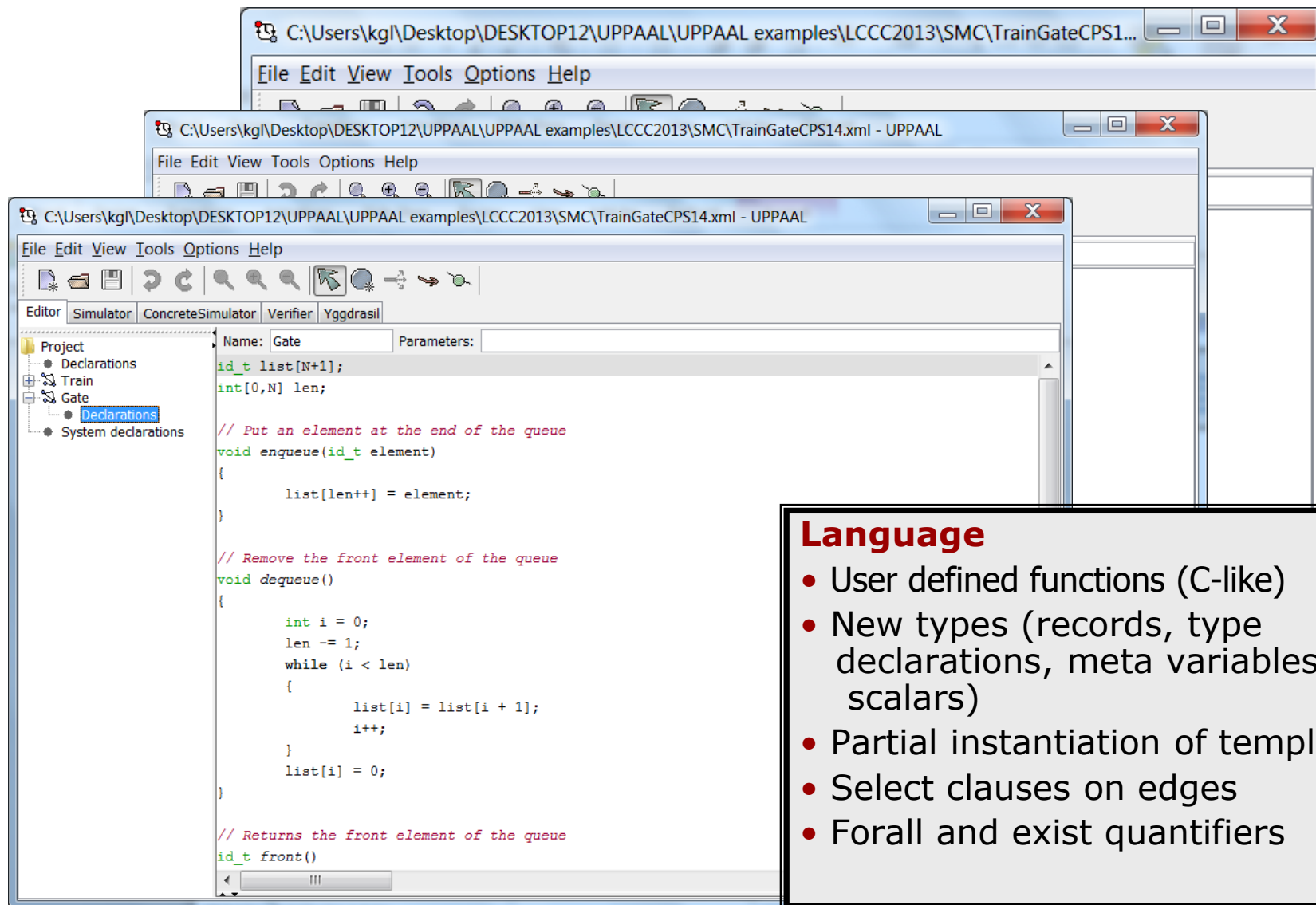
Stop the train while it still stoppable!

Train Crossing



Train Crossing





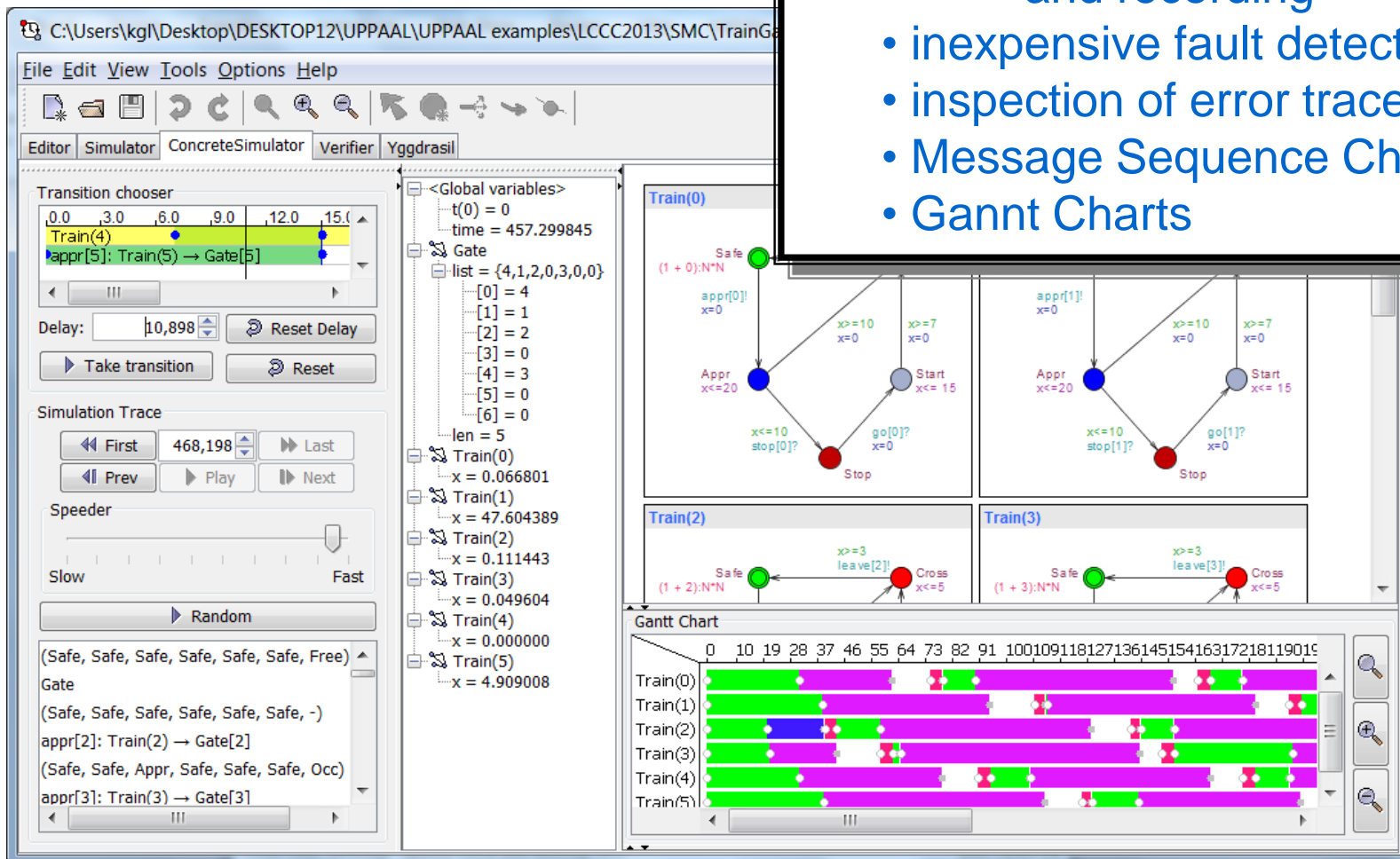
Language

- User defined functions (C-like)
- New types (records, type declarations, meta variables, scalars)
- Partial instantiation of templates
- Select clauses on edges
- Forall and exist quantifiers

Concrete Simulator

Graphical Simulator

- visualization and recording
- inexpensive fault detection
- inspection of error traces
- Message Sequence Charts
- Gantt Charts



Symbolic Simulator



Graphical Simulator

- visualization and recording
- inexpensive fault detection
- inspection of error traces
- Message Sequence Charts
- Gantt Charts

File Edit View Tools Options Help

Editor Simulator ConcreteSimulator Verifier Yggdrasil

Enabled Transitions

go[front()]: Gate → Train(5)

Next Reset

Simulation Trace

Train(1)

(Safe, Cross, Stop, Stop, Stop, Stop, Occ)

leave[1]: Train(1) → Gate[1]

(Safe, Safe, Stop, Stop, Stop, Stop, Free)

go[front()]: Gate → Train(5)

(Safe, Safe, Stop, Stop, Stop, Start, Occ)

appr[0]: Train(0) → Gate[0]

Trace File:

Prev Next Replay

Open Save Random

Slow Fast

Gate

list = {5,3,4,2,0,0,0}

[0] = 5

[1] = 3

[2] = 4

[3] = 2

[4] = 0

[5] = 0

[6] = 0

len = 4

Constraints

time ≥ 63

Train(0).x ∈ [13,25]

Train(1).x ∈ [3,5]

Train(2).x ∈ [10,25]

Train(3).x ∈ [30,65]

Train(4).x ∈ [23,60]

Train(5).x ∈ [30,65]

Train(0).x - time ≤ -50

Train(0).x - Train(1).x ∈ [10,20]

Train(0).x - Train(2).x ∈ [0,5]

Train(3).x - Train(0).x ∈ [17,40]

Train(4).x - Train(0).x ∈ [10,35]

Train(2).x - Train(1).x ∈ [7,20]

Train(3).x - Train(5).x ∈ [-5,0]

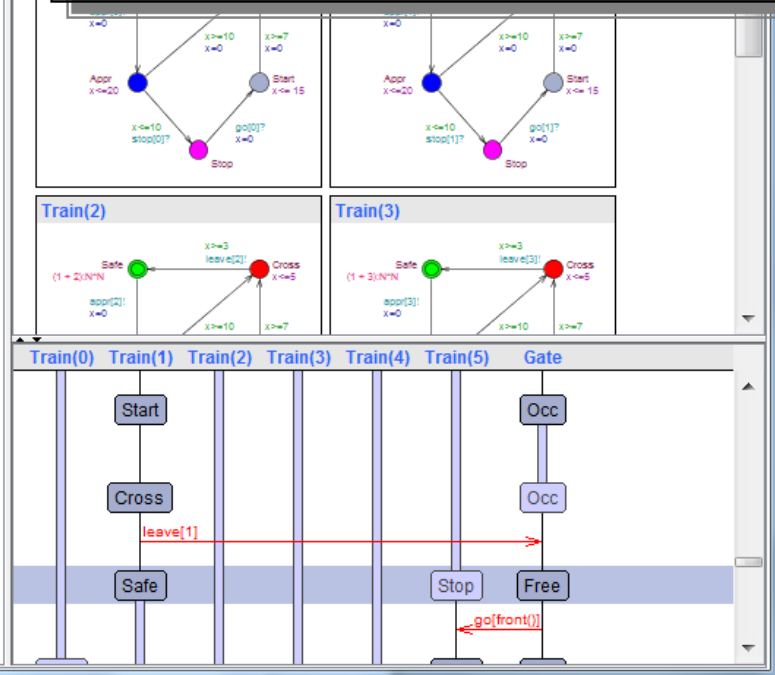
Train(4).x - time ≤ -33

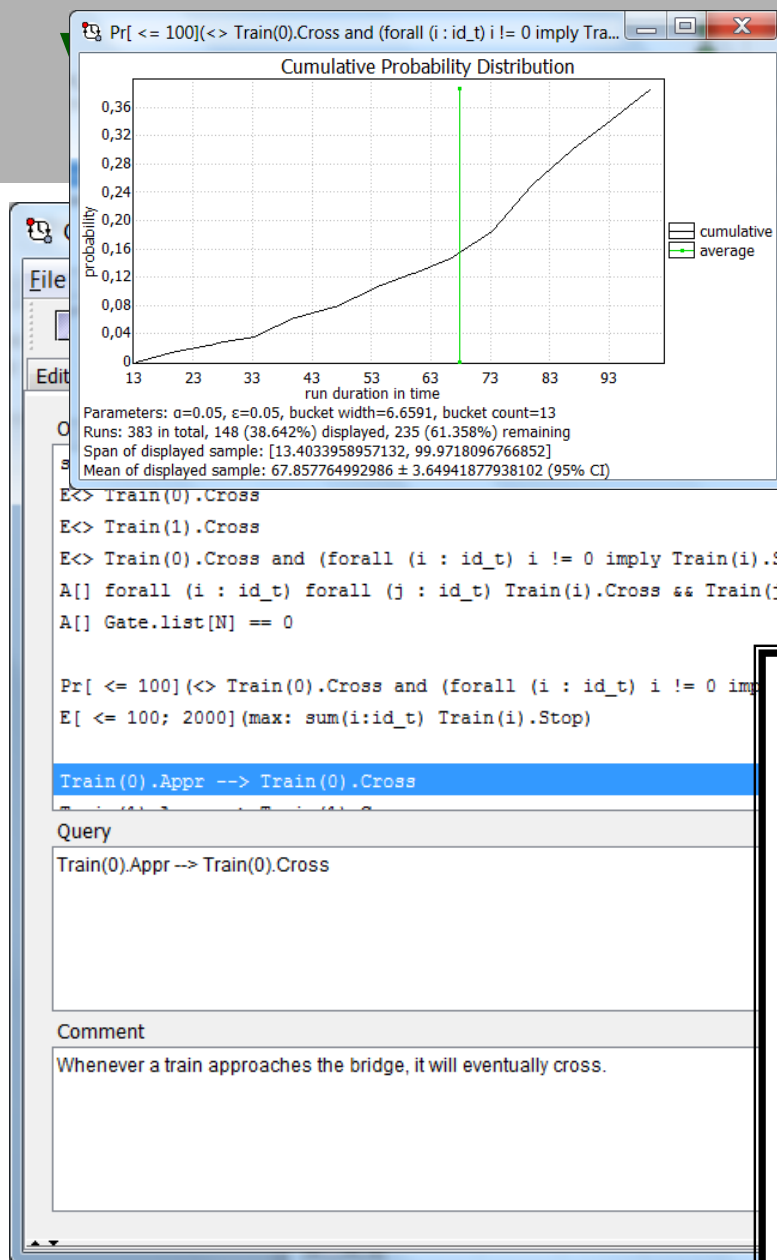
Train(4).x - Train(3).x ∈ [-20,0]

Train(5).x - time ≤ -30

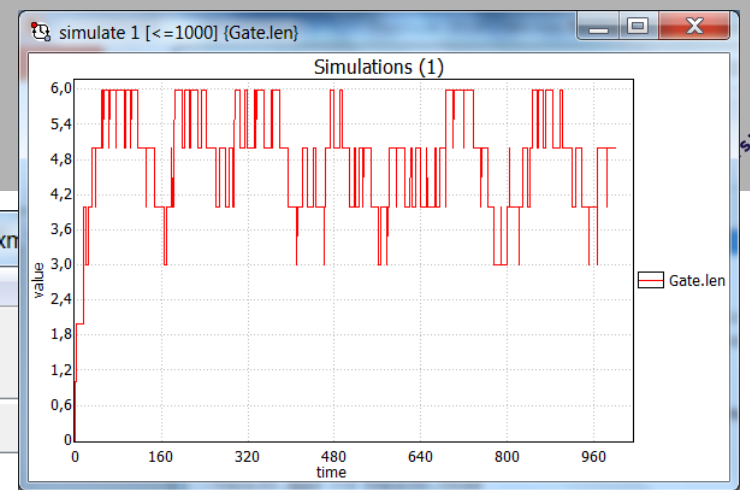
Train(5).x - Train(0).x ∈ [17,40]

Train(5).x - Train(4).x ∈ [0,20]





013\SMC\TrainGateCPS14.xm

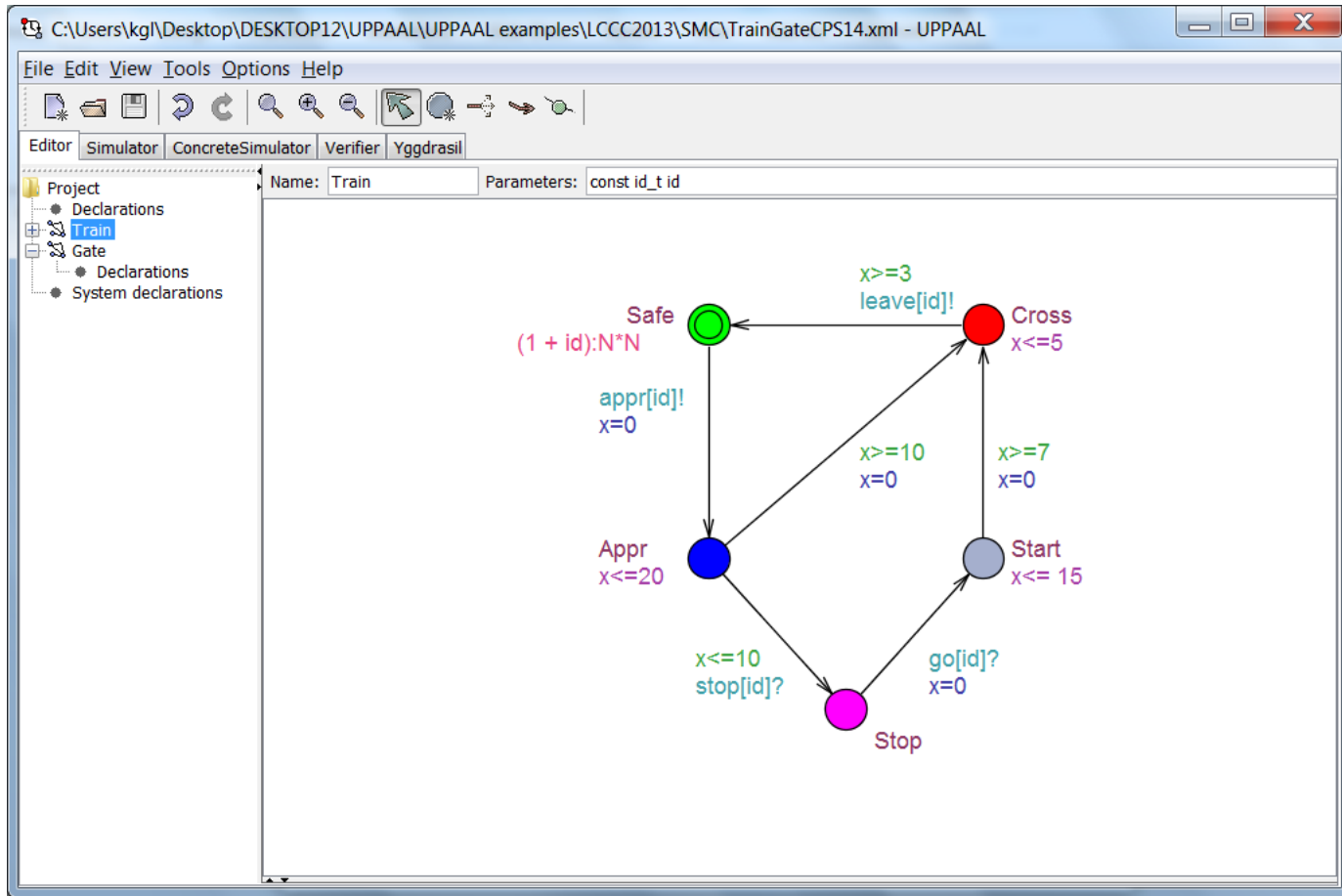


Verifier

- Exhaustive & automatic checking of requirements
- .. including validating, safety, liveness, bounded liveness and response properties
- .. performance properties, e.g probabilistic and expectation.
- .. generation of debugging information for visualisation in simulator.
- .. plot composer

SITET

Demo 1



UPPAAL Examples
SSFT15/UPPAAL SMC/

■ Validation Properties

- Possibly: $E \langle \rangle P$

■ Safety Properties

- Invariant: $A [] P$
- Pos. Inv.: $E [] P$

■ Liveness Properties

- Eventually: $A \langle \rangle P$
- Leadsto: $P \dashrightarrow Q$

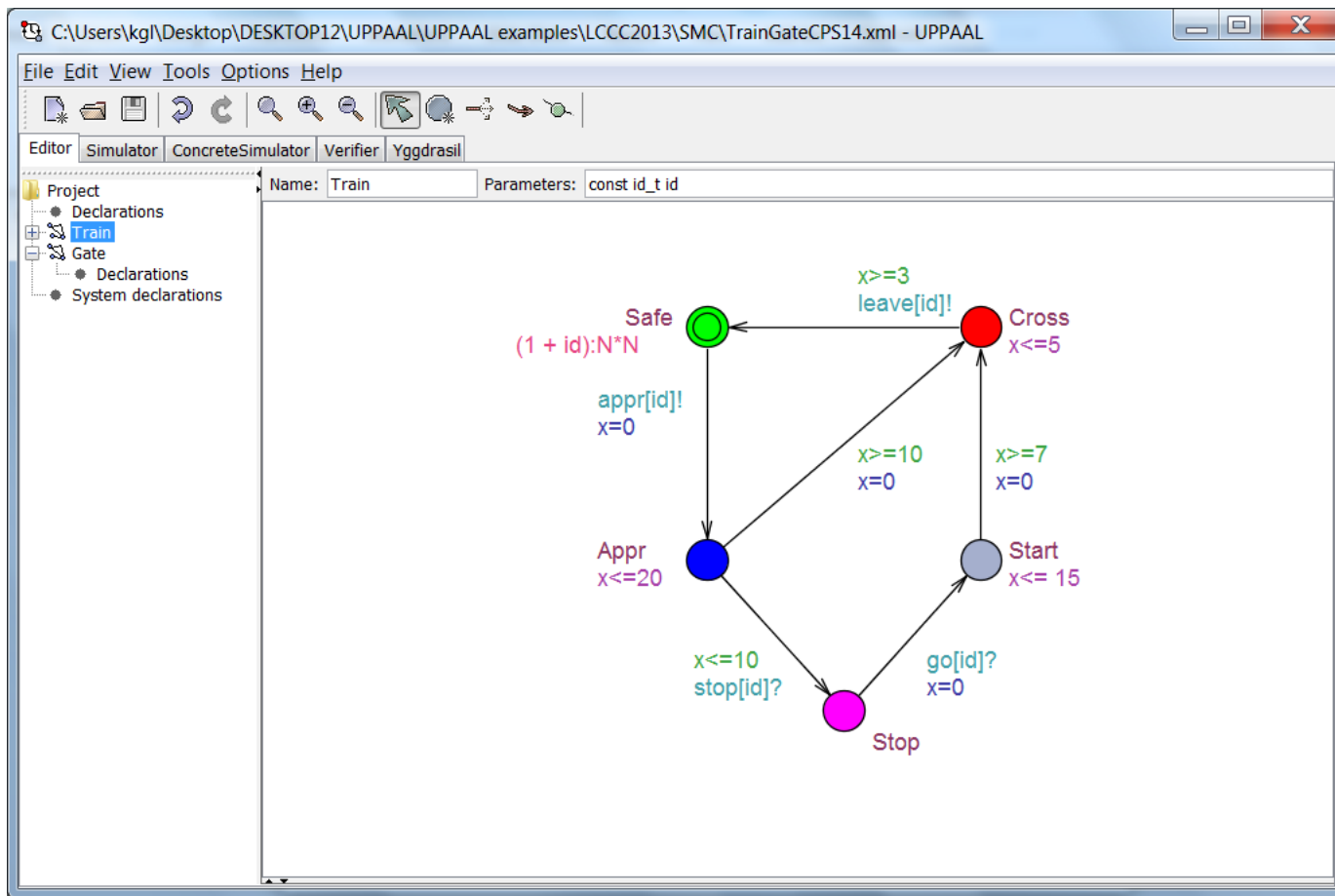
■ Bounded Liveness

- Leads to within: $P \dashrightarrow_{\leq t} Q$

The expressions P and Q must be type safe, side effect free, and evaluate to a boolean.

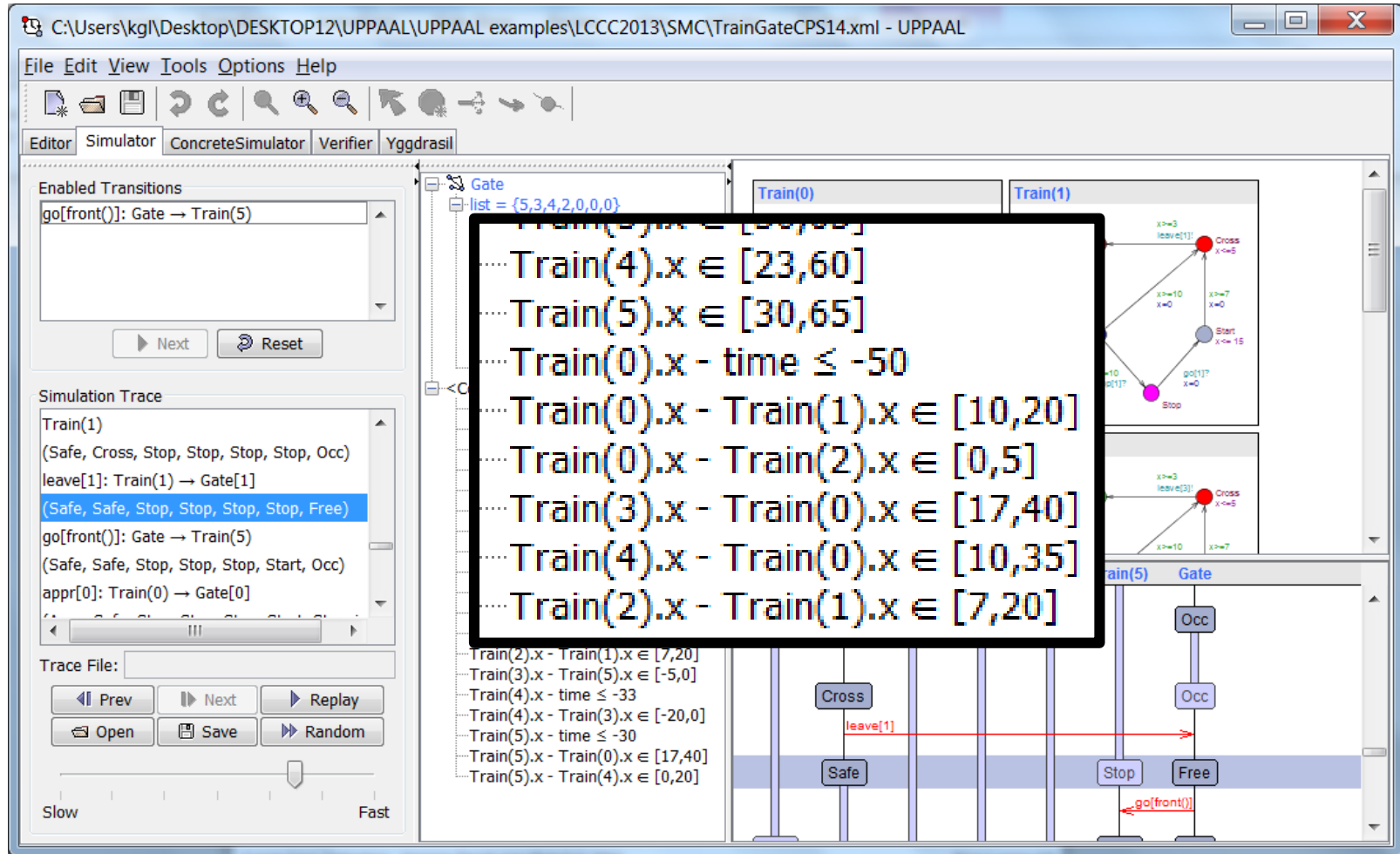
Only references to integer variables, constants, clocks, are allowed (and arrays of these).

Demo 2



UPPAAL Examples
SSFT15/UPPAAL SMC/

THE "secret" of UPPAAL



The screenshot shows the UPPAAL simulator interface. The main window displays a simulation trace and a list of constraints. A black box highlights the following constraints:

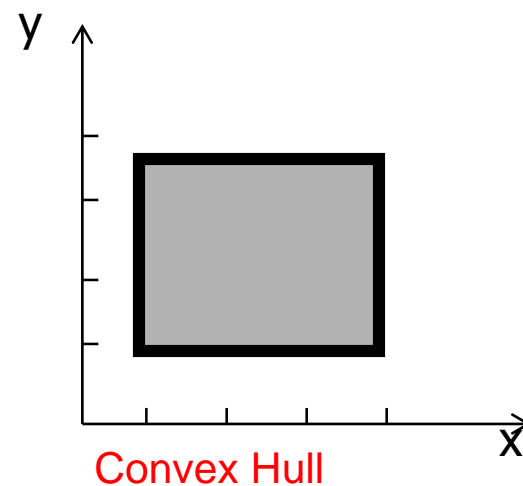
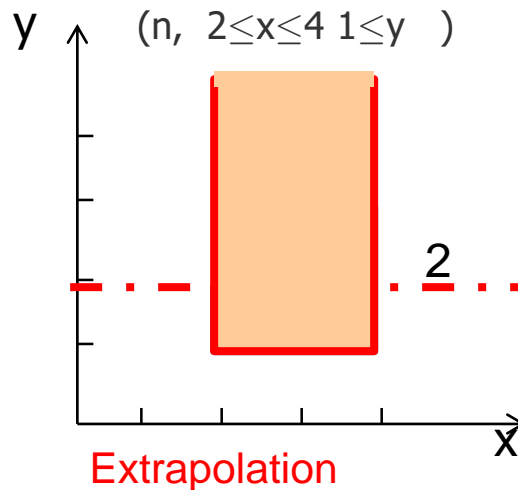
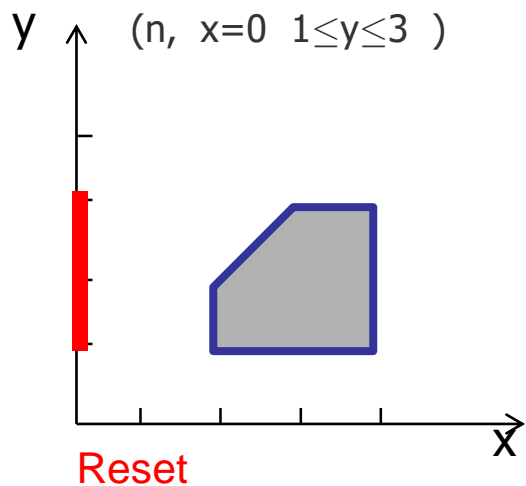
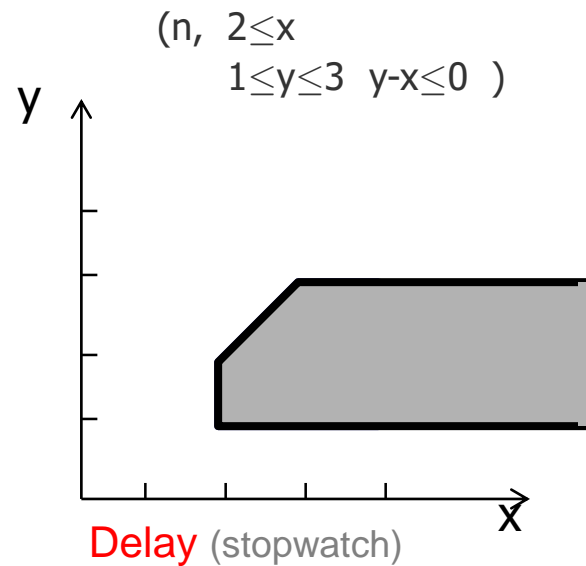
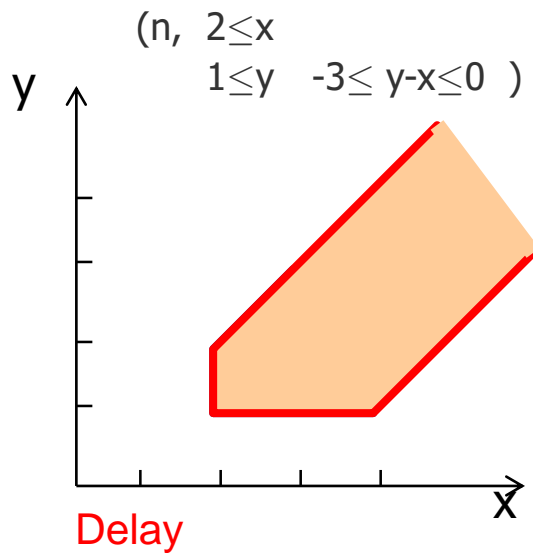
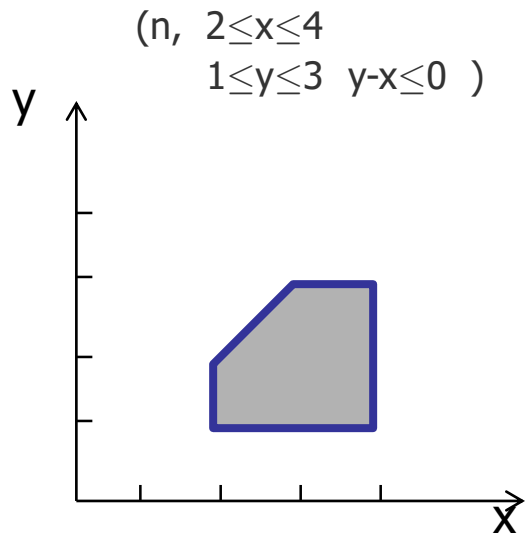
- Train(4).x \in [23,60]
- Train(5).x \in [30,65]
- Train(0).x - time \leq -50
- Train(0).x - Train(1).x \in [10,20]
- Train(0).x - Train(2).x \in [0,5]
- Train(3).x - Train(0).x \in [17,40]
- Train(4).x - Train(0).x \in [10,35]
- Train(2).x - Train(1).x \in [7,20]

The simulation trace shows the following sequence of events:

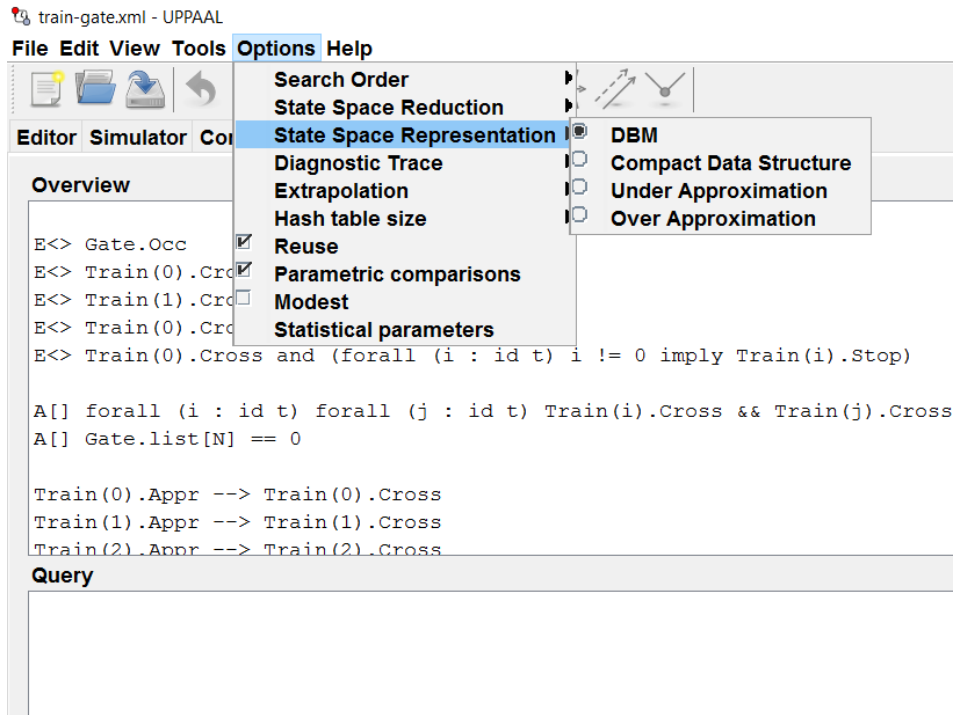
- Train(1) (Safe, Cross, Stop, Stop, Stop, Stop, Occ)
- leave[1]: Train(1) \rightarrow Gate[1]
- (Safe, Safe, Stop, Stop, Stop, Stop, Free)
- go[front()]: Gate \rightarrow Train(5)
- (Safe, Safe, Stop, Stop, Stop, Start, Occ)
- appr[0]: Train(0) \rightarrow Gate[0]

The interface also shows a diagram of the train gate system with various states (Cross, Safe, Stop, Free) and transitions (leave[1], go[front()]).

Zones – Operations



Verification Options



- **Search Order**
 - Depth First
 - Breadth First
 - Random Depth First

- **State Space Reduction**

- None
- Conservative
- Aggressive
- Extreme

- **State Space Representation**

- DBM
- Compact Form
- Under Approximation
- Over Approximation

- **Diagnostic Trace**

- **Extrapolation**

- Automatic
- None
- Difference
- Local
- Lower/Upper

Stochastic Timed Automata



With

Peter Bulychev, Alexandre David,
Marius Mikucionis

Dehui Du, Axel Legay, Guangyuan Li,
Danny B. Poulsen, Amalie Stainer, Zheng Wang



AALBORG UNIVERSITET

FORMATS11+12, CAV11,
RV12, HSB12, QAPL12,
NaSA12+13, SCIENCE CH13, STTT15



Safety ✓

$A[] \text{ forall } (i : \text{id_t}) \text{ forall } (j : \text{id_t})$
 $\text{Train}(i).\text{Cross} \ \&\& \ \text{Train}(j).\text{Cross} \ \text{imply } i == j$

Reachability ✓

$E \leftrightarrow \text{Train}(0).\text{Cross} \ \text{and} \ \text{Train}(1).\text{Stop}$

Liveness ✓

$\text{Train}(0).\text{Appr} \ \text{-->} \ \text{Train}(0).\text{Cross}$

$A \leftrightarrow \dots E[] \dots$ ✓

Limited quantitative analysis ✓

sup: .. inf: ..

~~Performance properties~~

~~State-space explosion~~



Editor Simulator Verifier

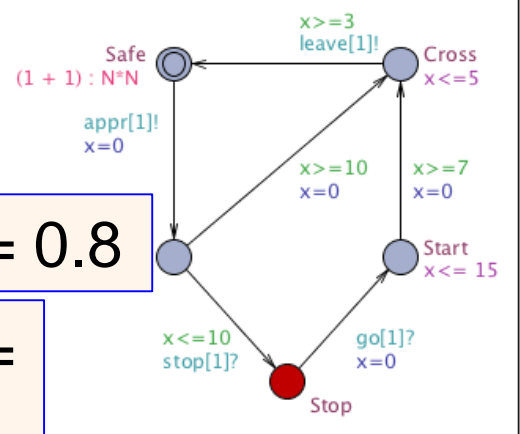
Performance properties ✓

$$\Pr[\leq 200](\langle \rangle \text{Train}(5).\text{Cross})$$

$$\Pr[\leq 100](\langle \rangle \text{Train}(0).\text{Cross}) \geq 0.8$$

$$\Pr[\leq 100](\langle \rangle \text{Train}(5).\text{Cross}) \geq$$

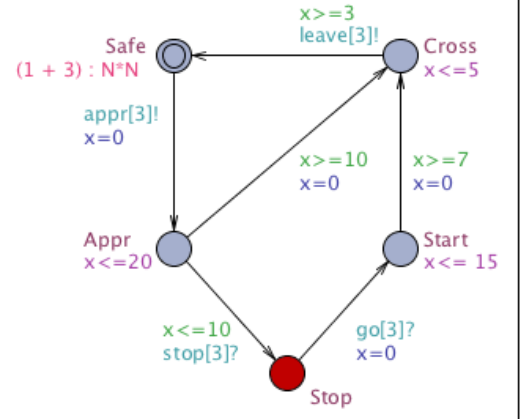
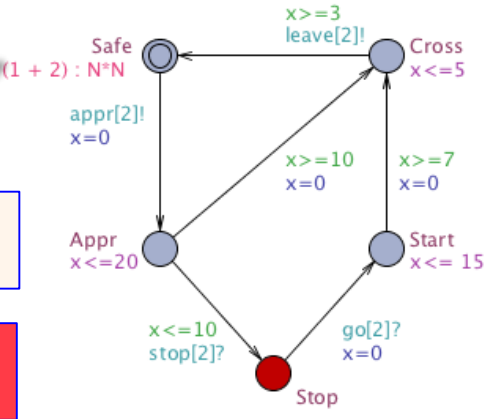
$$\Pr[\leq 100](\langle \rangle \text{Train}(1).\text{Cross})$$



Train(5)
 appr[0]: Train(0) --> Gate
 Next Reset
 Simulation Trace
 appr[3]: Train(3) --> Gate
 (Safe, Stop, Safe, Appr, Stop, Start,
 stop[tail0]: Gate --> Train(3)
 (Safe, Stop, Safe, Stop, Stop, Start, Occ)
 appr[2]
 (Safe,
 stop[tail0]: Gate --> Train(2)
 (Safe, Stop, Stop, Stop, Stop, Start,

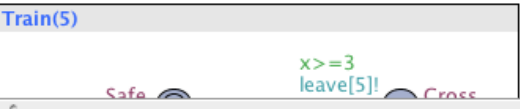
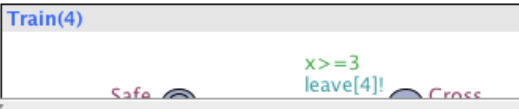
State-space explosion ✓

Generate runs



Performance properties

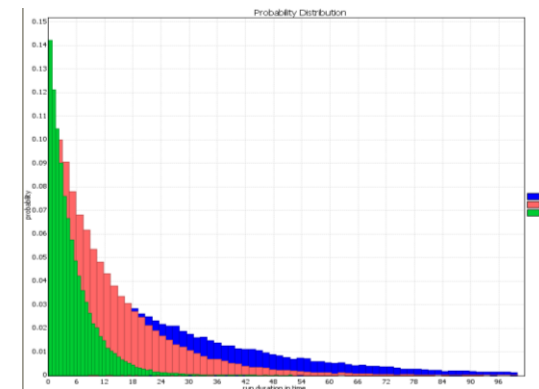
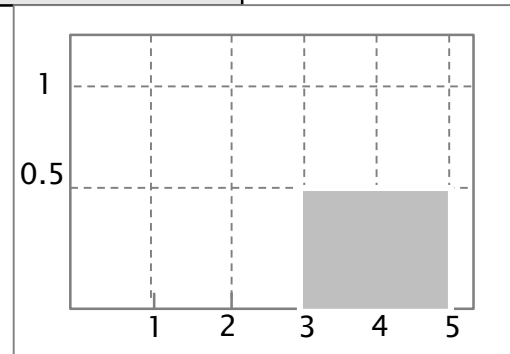
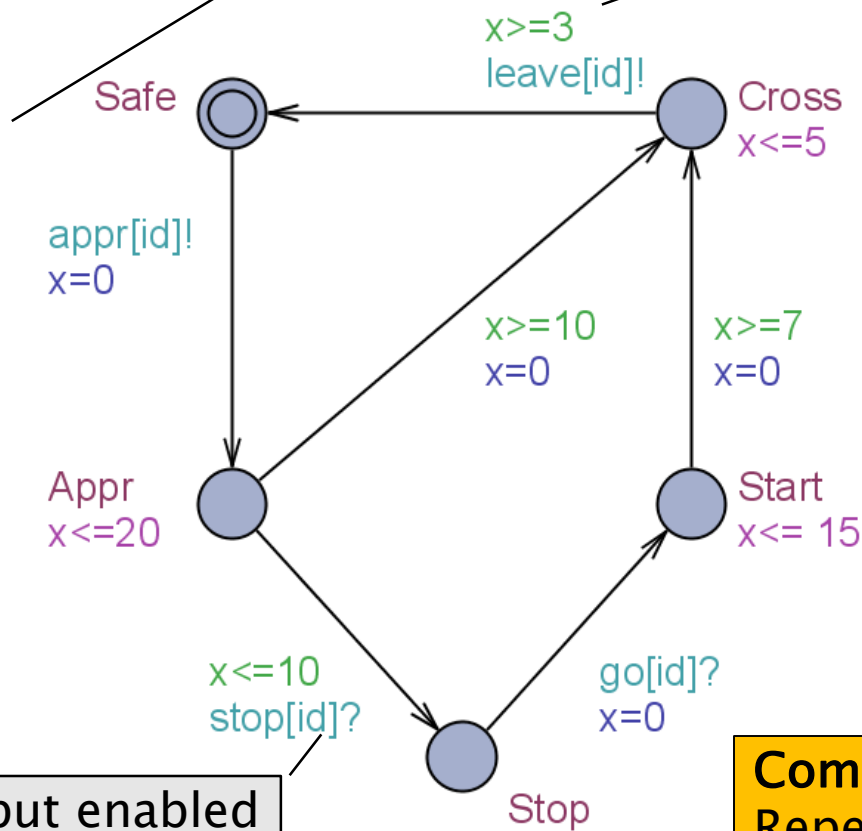
State-space explosion



Stochastic Semantics of TA

Exponential Distribution

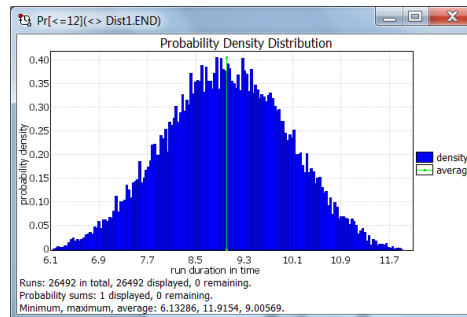
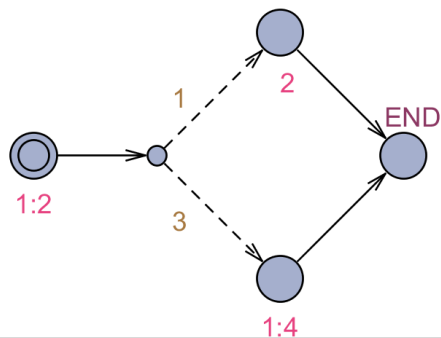
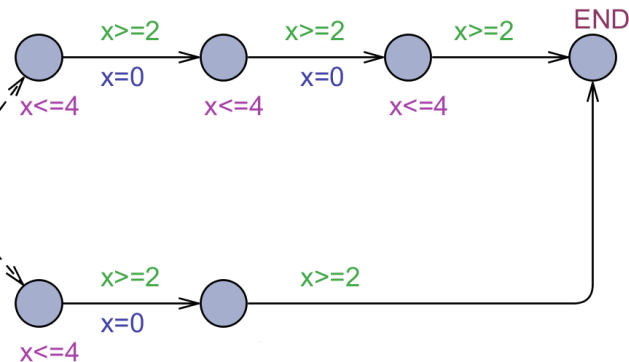
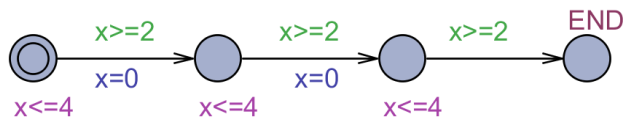
Uniform Distribution



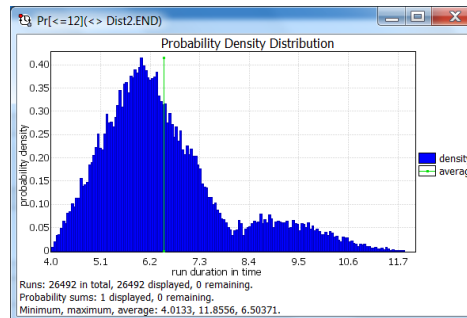
Input enabled

Composition = Repeated races between components for outputting

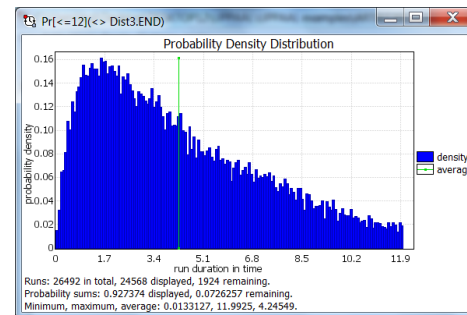
Composed Distributions



$$\Pr(\langle \rangle_{\leq 9} \text{ END}) = \frac{1}{2}$$



$$\Pr(\langle \rangle_{\leq 7} \text{ END}) \geq \frac{1}{2}$$

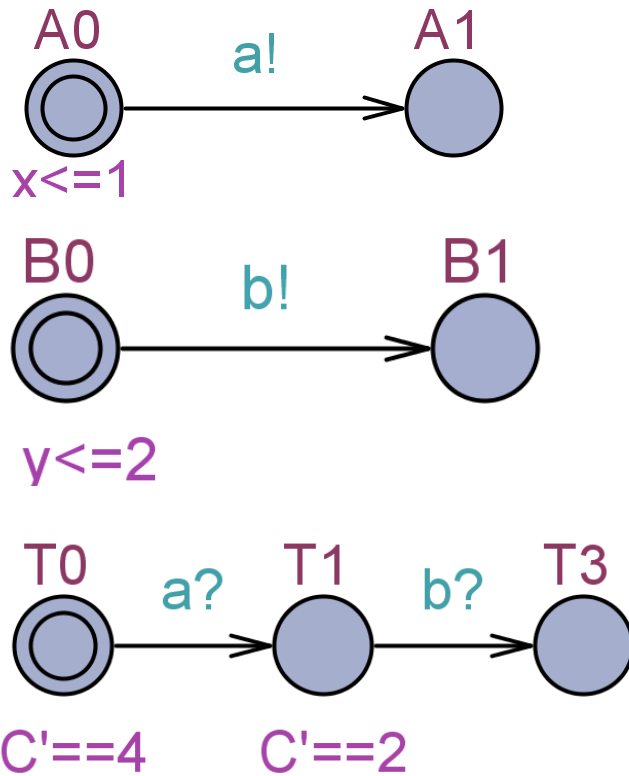


Includes all Phase-Type Distributions.

Can encode any distribution with arbitrary precision.

σ -algebra with prob. measure from cylinders $C(I_0 \ell_0 I_1 \ell_1 I_2 \dots I_n \ell_{n+1})$

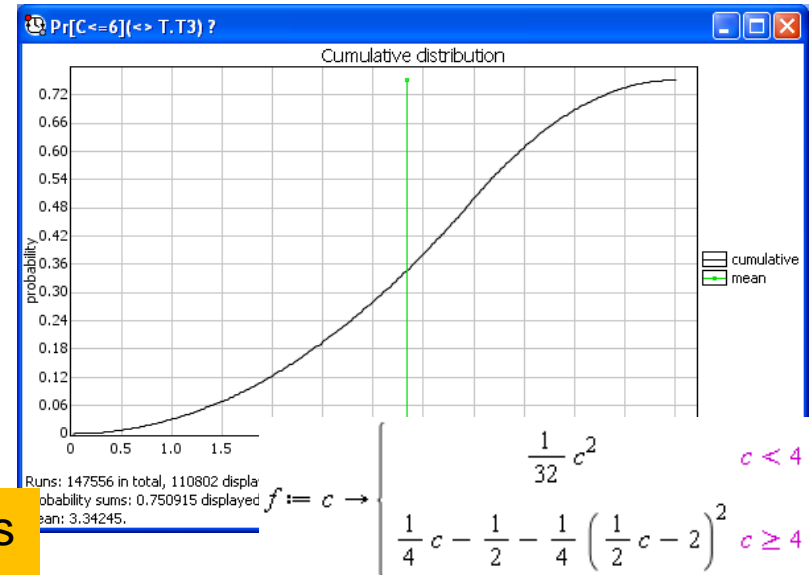
Composition of STA



$\Pr[\text{time} \leq 2](\langle \rangle T.T3) ?$

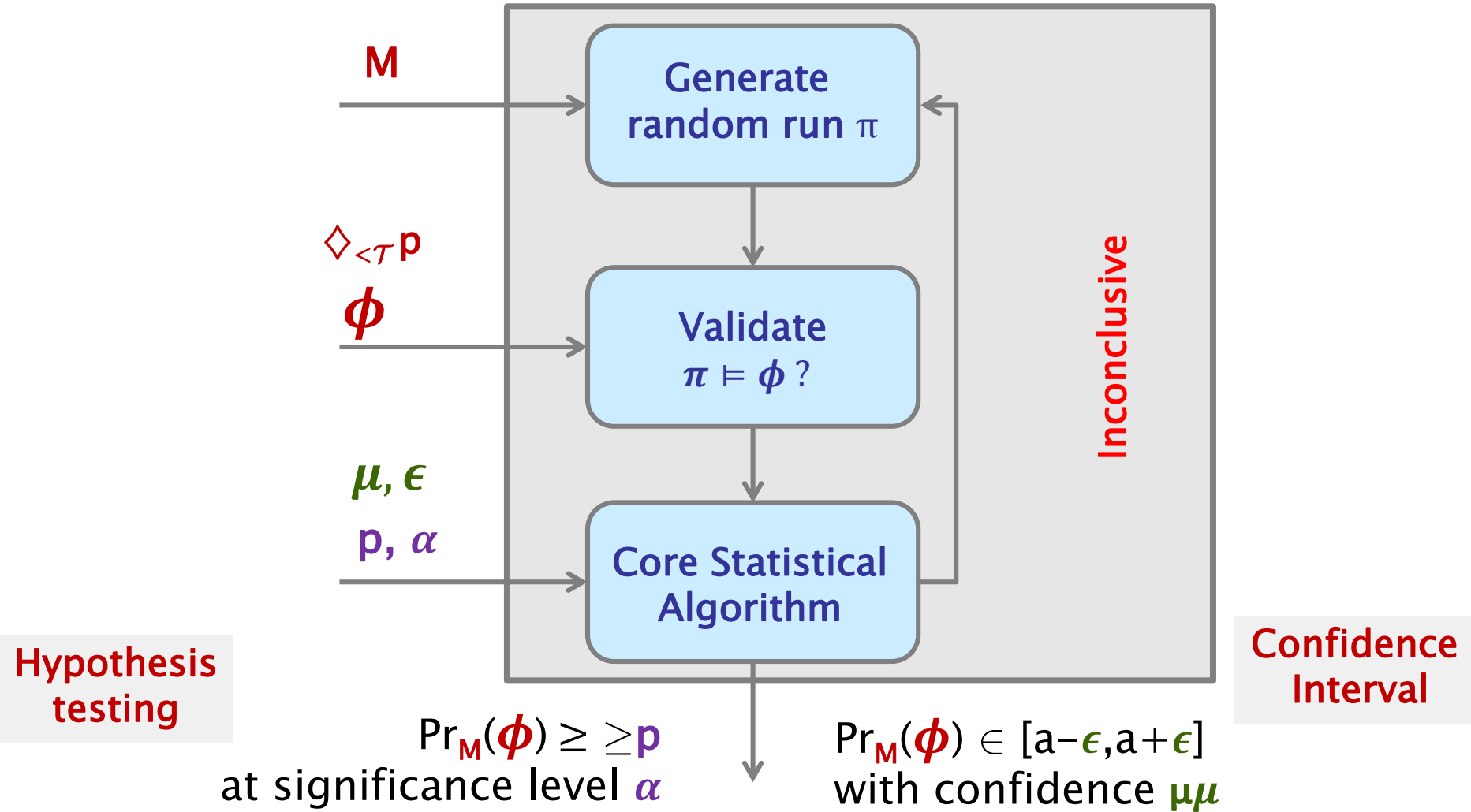
$$= \int_{t_a=0}^1 1 \cdot \int_{t_b=t_a}^2 \frac{1}{2} dt_b dt_a = 3/4$$

$\Pr[C \leq 6](\langle \rangle T.T3) ?$



Composition = Race between components for outputting

Statistical Model Checking



Hypothesis testing

$\Pr_M(\phi) \geq p$
at significance level α

Confidence Interval

$\Pr_M(\phi) \in [a-\epsilon, a+\epsilon]$
with confidence μ

- Evaluation

$\text{Pr} [\leq 100] (\langle \rangle \text{expr})$ $\text{Pr} (\Phi): \Phi \in \text{MITL}$

Hypothesis testing

$\text{Pr} [\leq 100] (\langle \rangle \text{expr}) \geq 0.1$

`c<=100 #<=50 [] expr <=0.5`

- Comparison

$\text{Pr} [\leq 20] (\langle \rangle e1) \geq \text{Pr} [\leq 10] (\langle \rangle e2)$

- Expected value

$\text{E} [\leq 10; 1000] (\text{min}: \text{expr})$

Explicit number of runs. Min or max.

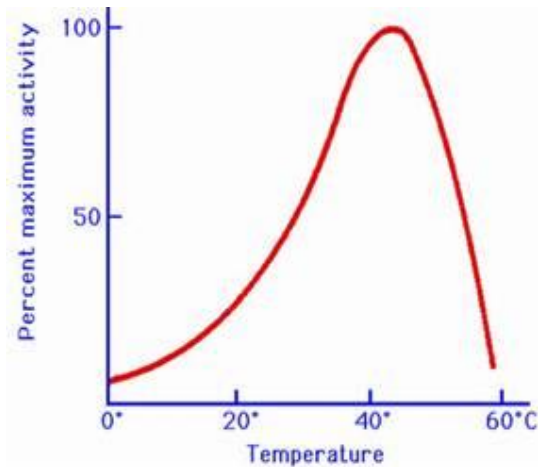
- Simulations

`simulate 10 [<=100]{expr1,expr2}`

DEMO



Stochastic Hybrid Automata



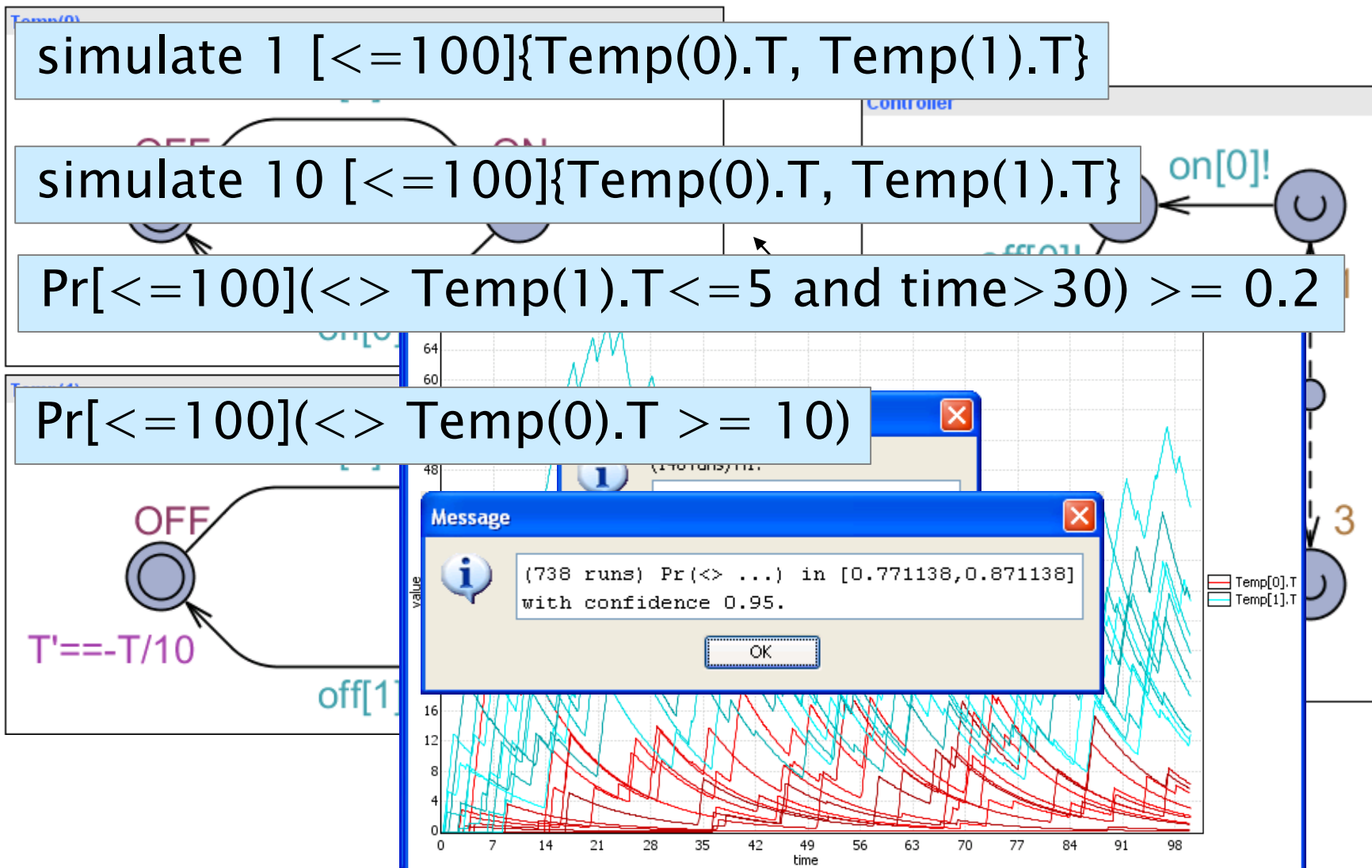
Stochastic Hybrid Systems

simulate 1 [≤ 100]{Temp(0).T, Temp(1).T}

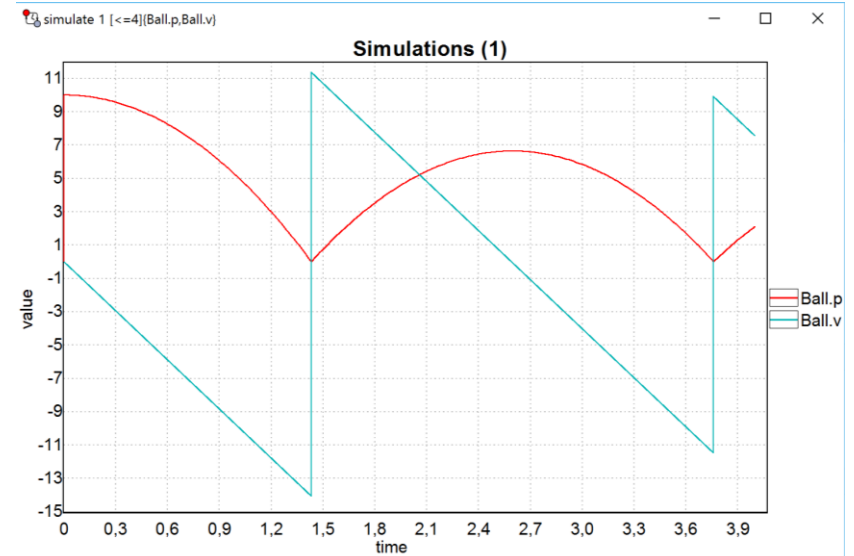
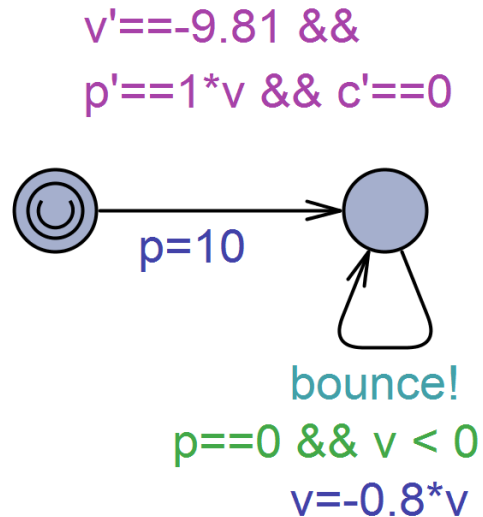
simulate 10 [≤ 100]{Temp(0).T, Temp(1).T}

$\Pr[\leq 100](\langle \rangle \text{Temp(1).T} \leq 5 \text{ and time} > 30) \geq 0.2$

$\Pr[\leq 100](\langle \rangle \text{Temp(0).T} \geq 10)$

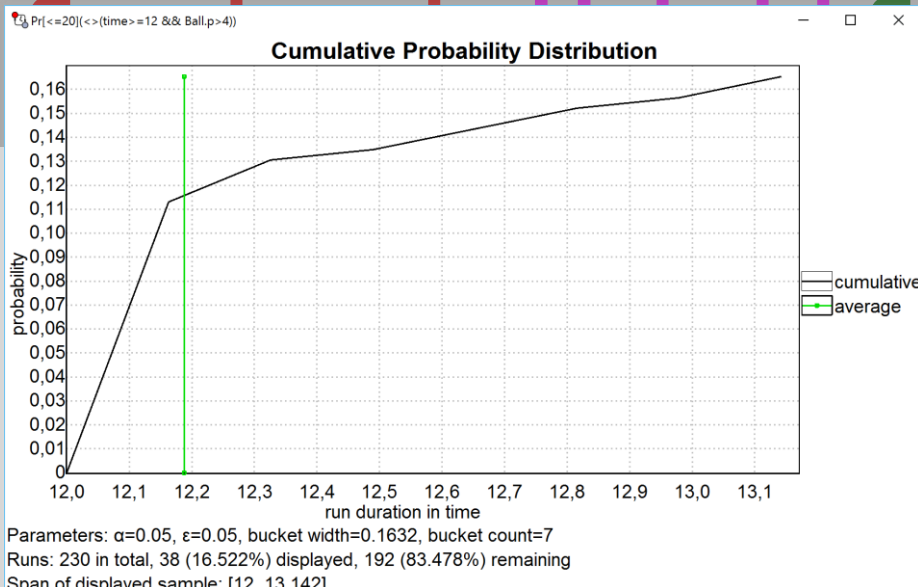


■ A Bouncing Ball



```
simulate 1 [ $\leq 4$ ]{Ball.p, Ball.v}
```


Player 1

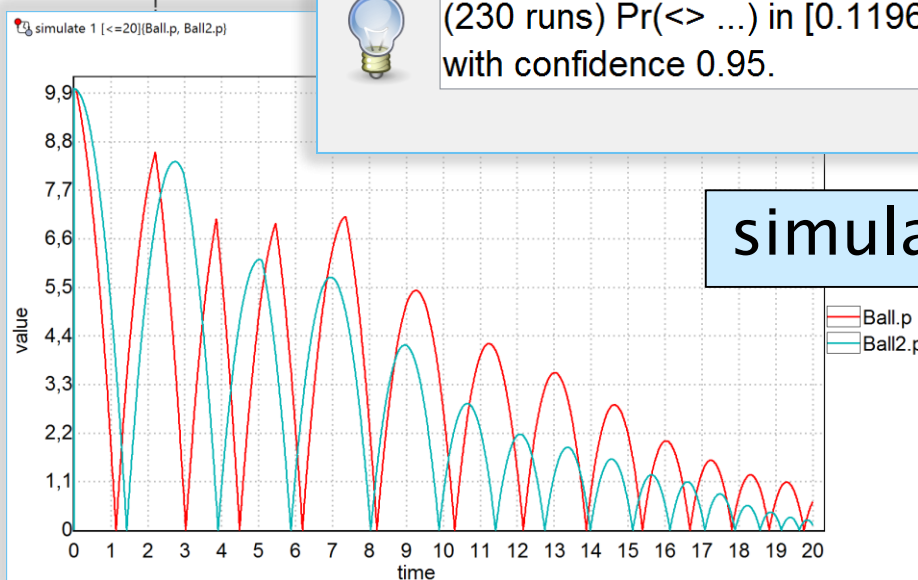


$\Pr(<> (time \geq 12 \ \&\& \ Ball1.p > 4))$



(230 runs) $\Pr(<> \dots)$ in [0.119649, 0.219647]
with confidence 0.95.

OK



simulate 1 [≤ 20]{Ball1.p, Ball2.p}

LMAC

node	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	d	d	d	d	d	3	3	3	3	3	3	3	3	3
1	i	d	d	d	d	d	1	1	1	1	1	d	d	d	d	d	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	i	d	d	d	d	d	1	1	1	1	1	d	d	d	d	d	2	2	2	2	2	d	d	d	d	d	3	3	3	3	3	3	3	3	3
3	i	i	i	i	i	i	i	w	w	w	w	w	d	d	d	d	d	0	0	0	0	0	0	d	d	d	d	4	4	4	4	4	4	4	4
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34



Lightweight Media Access Control



- Problem domain:
 - communication scheduling
- Targeted for:
 - self-configuring networks,
 - collision avoidance,
 - low power consumption
- Application domain:
 - wireless sensor networks
- **Initialization** (listen until a neighbor is heard)
- **Waiting** (delay a random amount of time frames)
- **Discovery** (wait for entire frame and note used slots)
- **Active**
 - choose free slot,
 - use it to transmit, including info about detected collisions
 - listen on other slots
 - fallback to Discovery if collision is detected
- Only neighbors can detect collision and tell the user-node that its slot is used by others

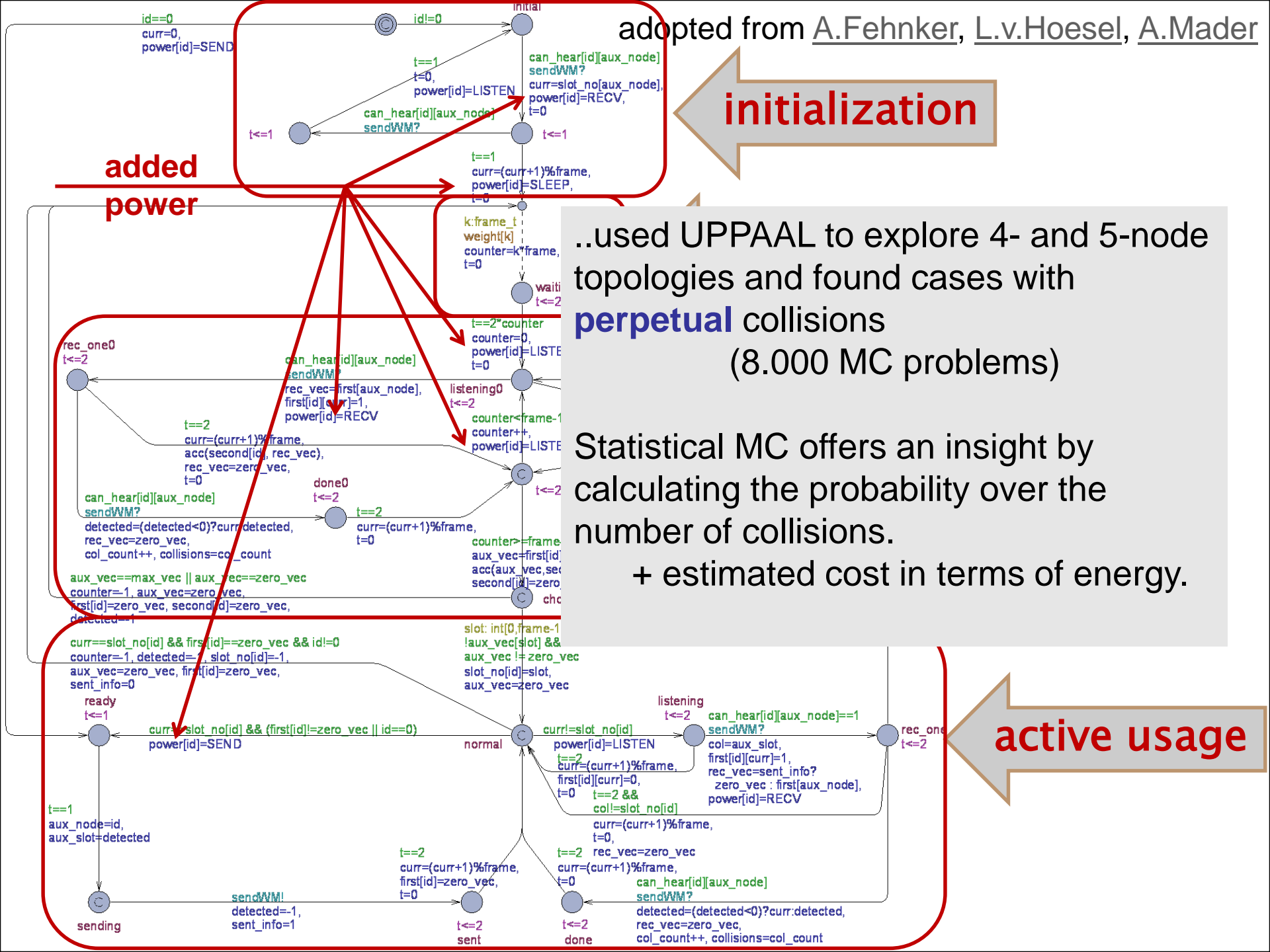
initialization

..used UPPAAL to explore 4- and 5-node topologies and found cases with **perpetual collisions** (8.000 MC problems)

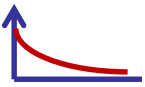


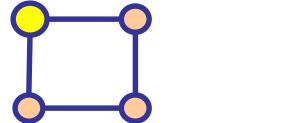
Statistical MC offers an insight by calculating the probability over the number of collisions.

+ estimated cost in terms of energy.

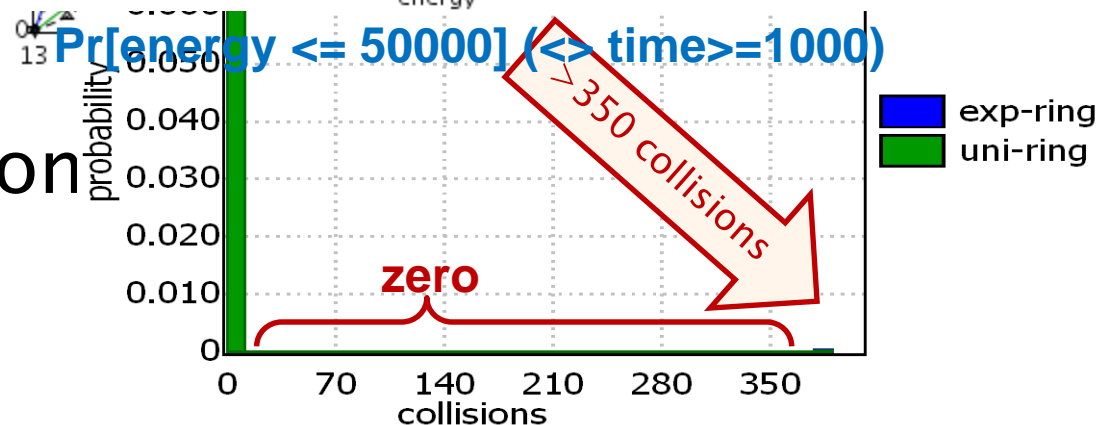
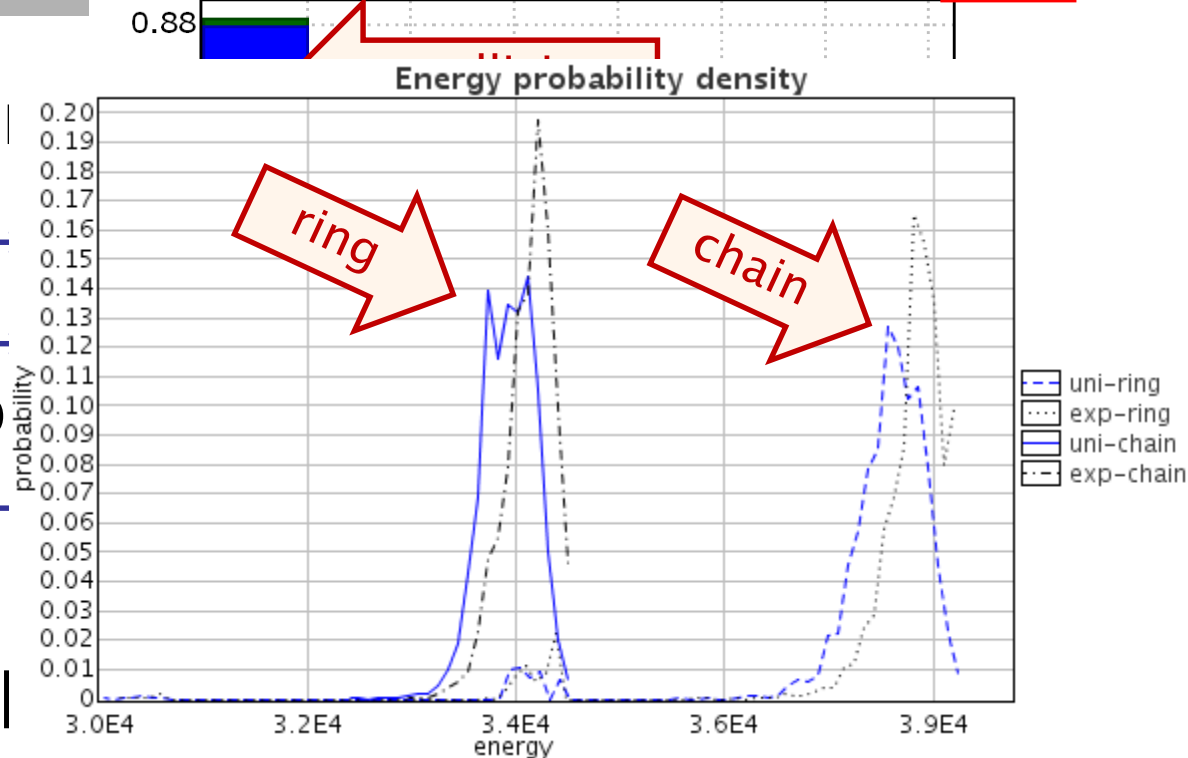
active usage



SMC of LMAC with 4 Nodes

- Wait distribution
 - geometric 
 - uniform 
- Network topology
 - chain 
 - ring 
- Collision probability
- Collision count
- Power consumption

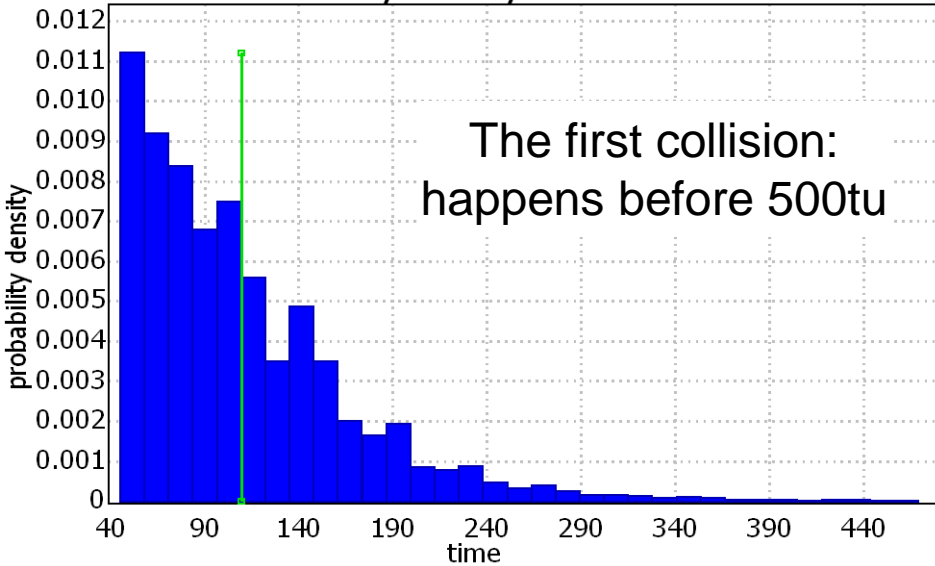
Probability density of Collision Count in a Chain



$Pr[\text{collisions} \leq 50000] (\langle \text{time} \rangle = 1000)$

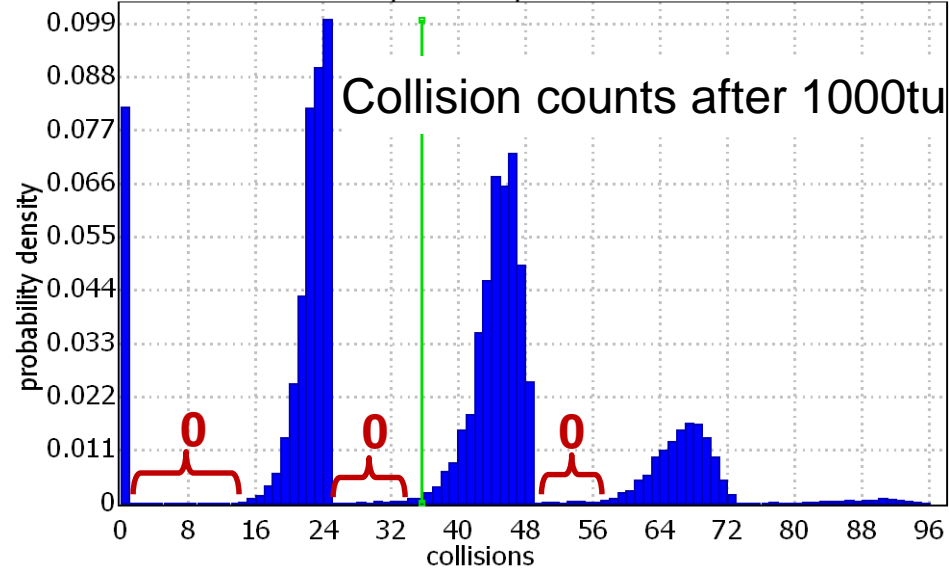
10-Node Star

Probability Density Distribution

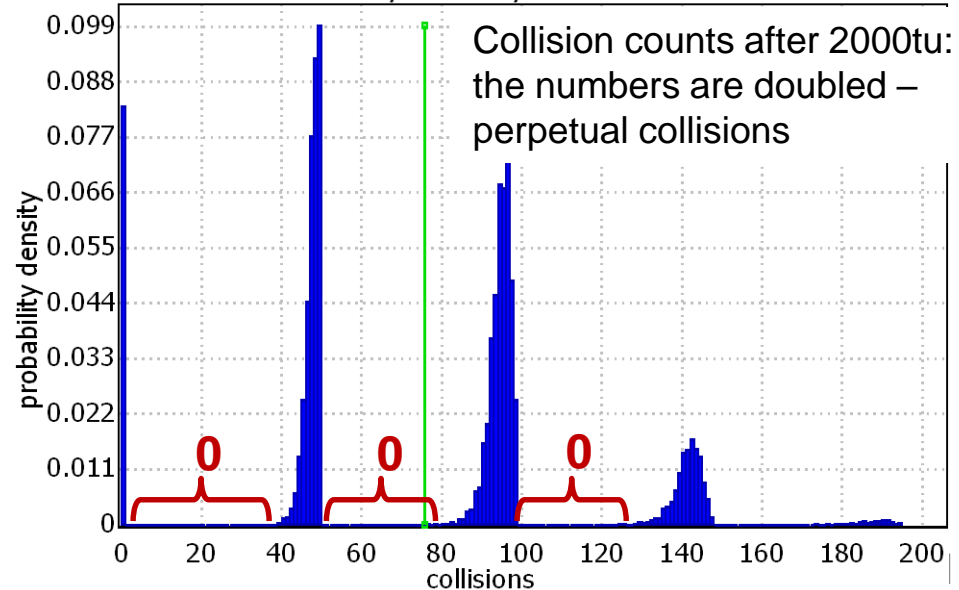


- The first collisions happen before **500tu**.
- It is unlikely (**8.2%**) that there will be **0** collisions.
- And if they happen, they are perpetual.

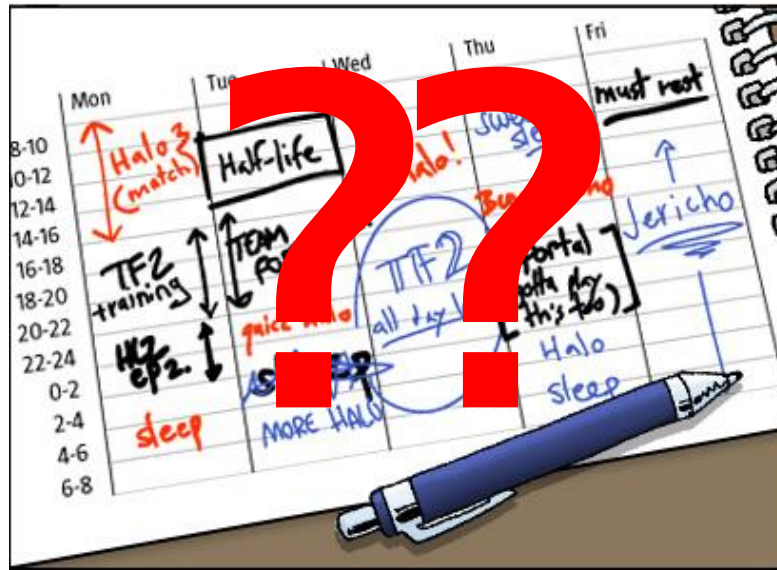
Probability Density Distribution



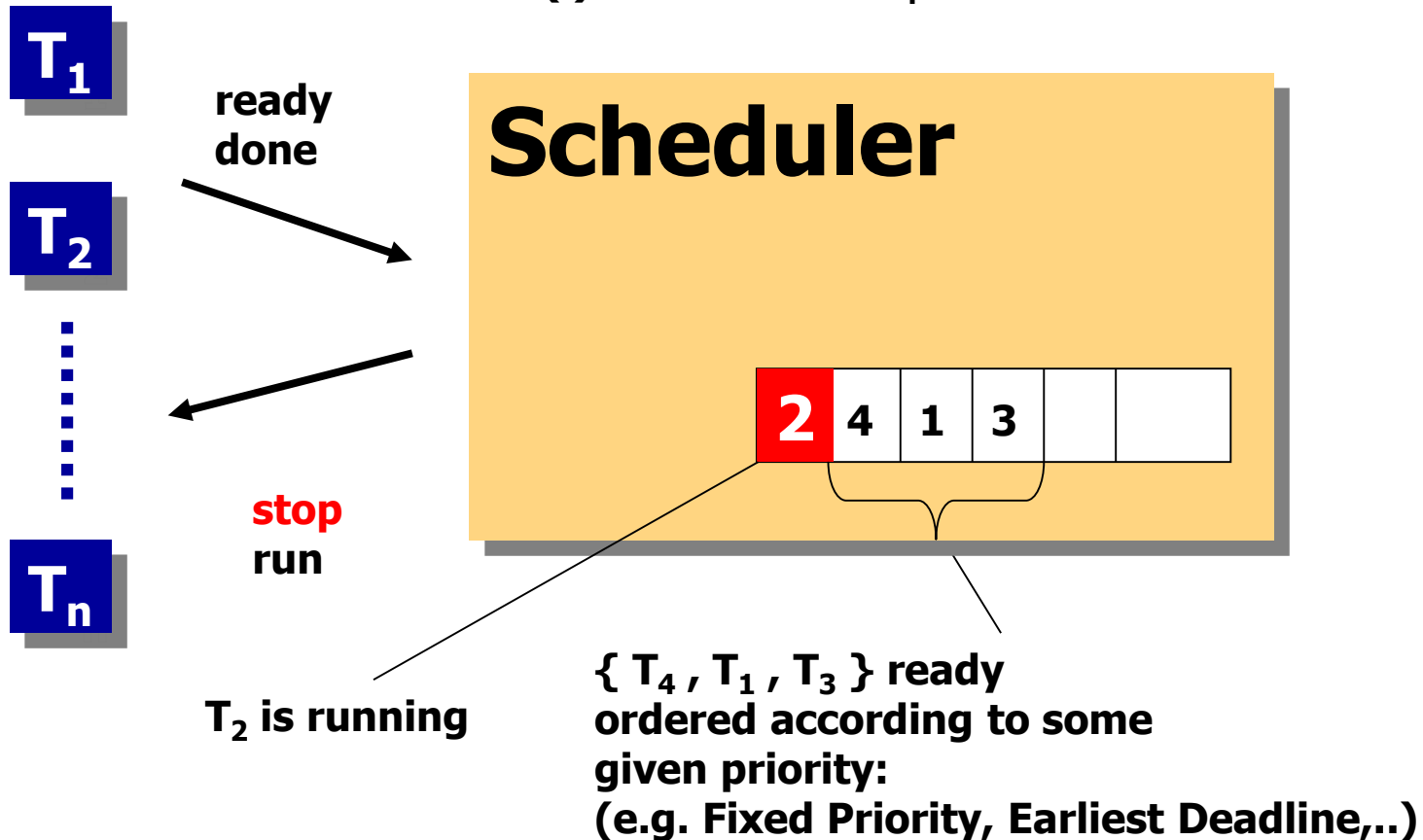
Probability Density Distribution



Schedulability & Performance Analysis



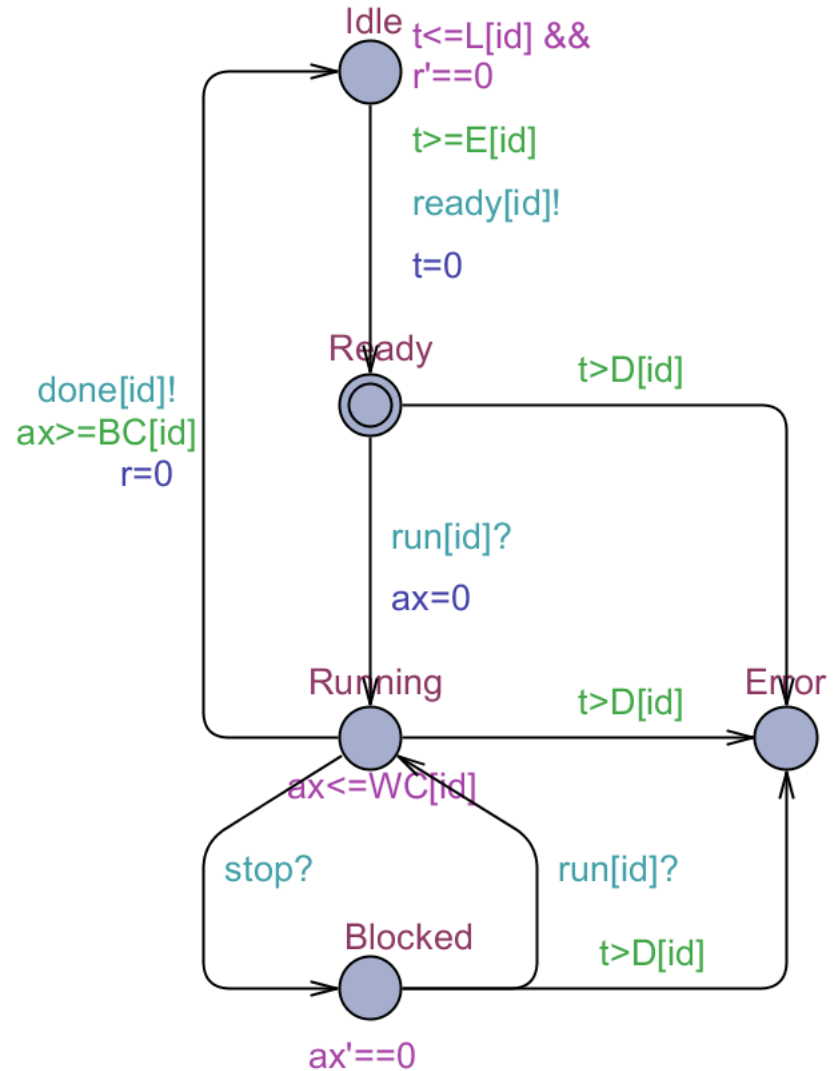
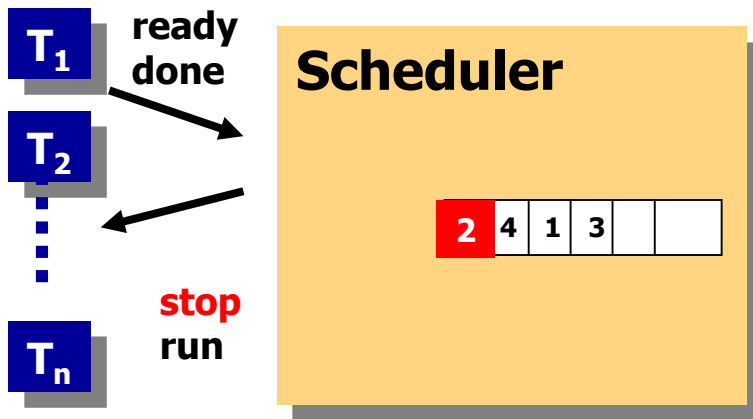
$P(i)$, **UNI**[$E(i)$, $L(i)$], .. : period or earliest/latest arrival or .. for T_i
 $C(i)$, **UNI**[$BC(i)$, $WC(i)$] : execution time for T_i
 $D(i)$: deadline for T_i



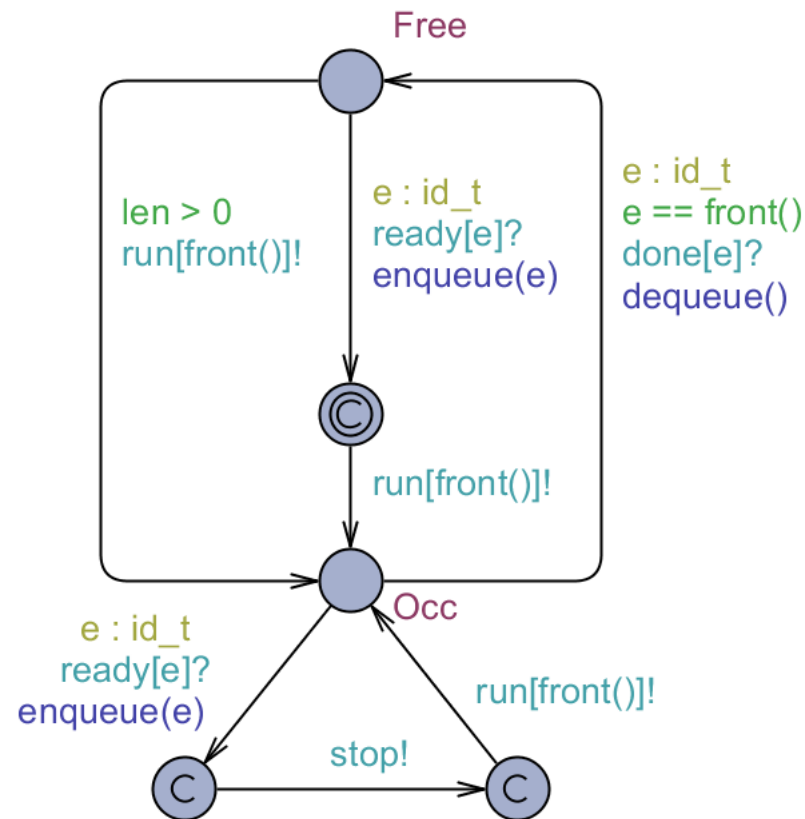
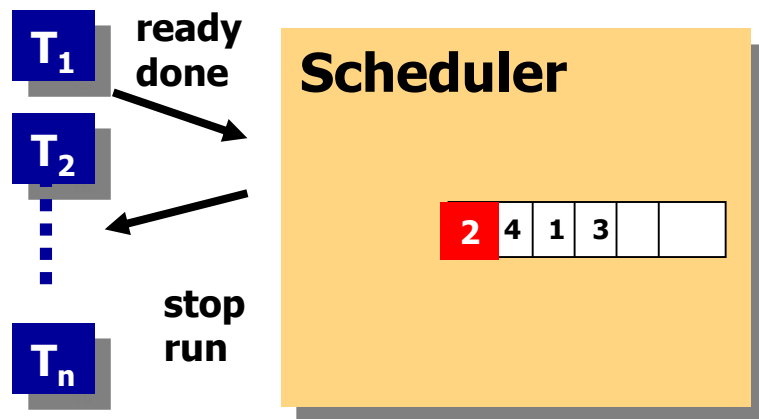
T_2 is running

{ T_4, T_1, T_3 } ready ordered according to some given priority:
(e.g. Fixed Priority, Earliest Deadline,..)

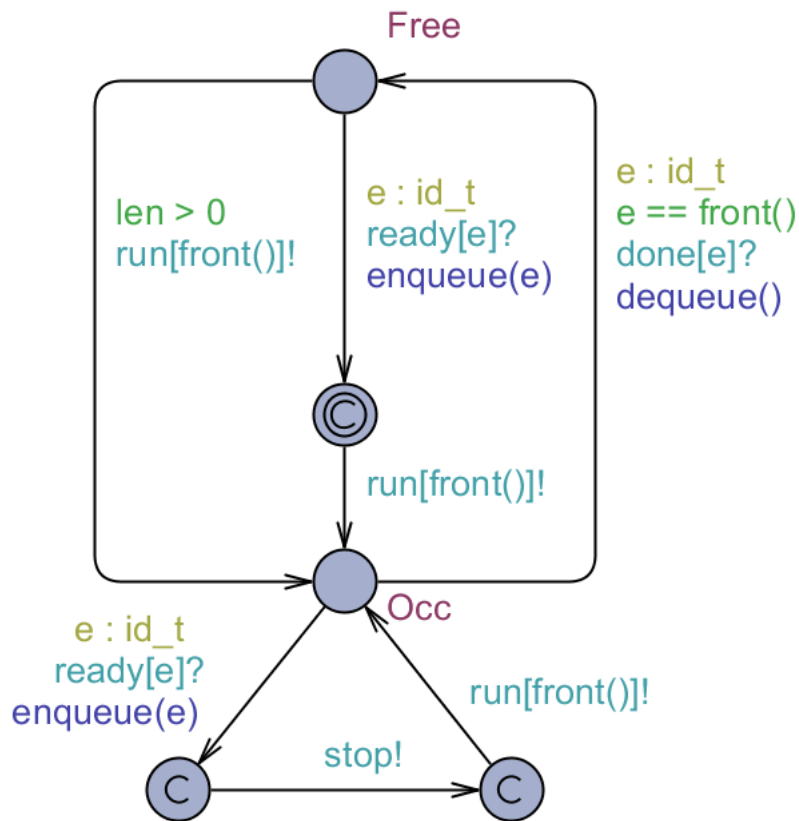
Modeling Task



Modeling Scheduler



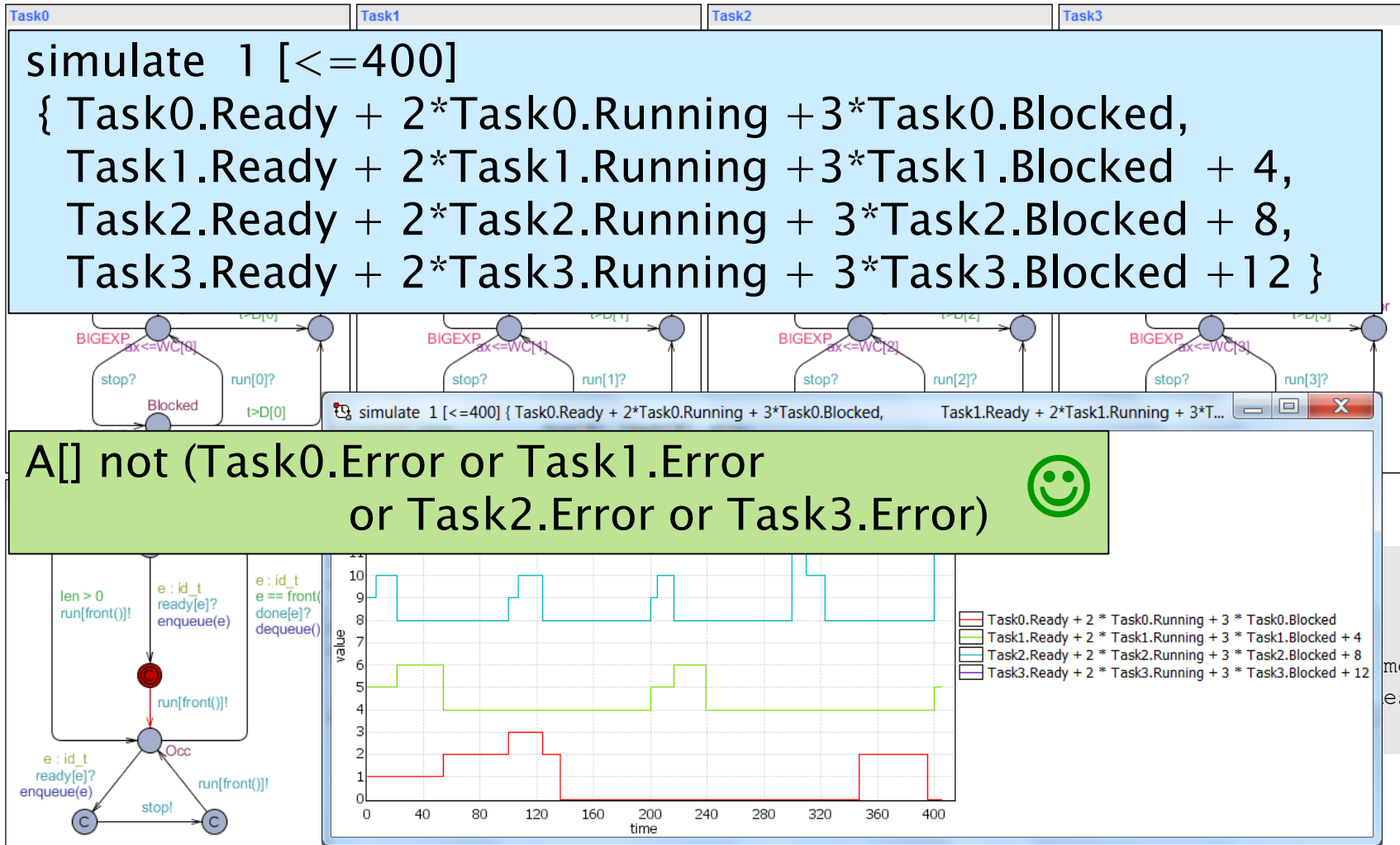
Modeling Queue



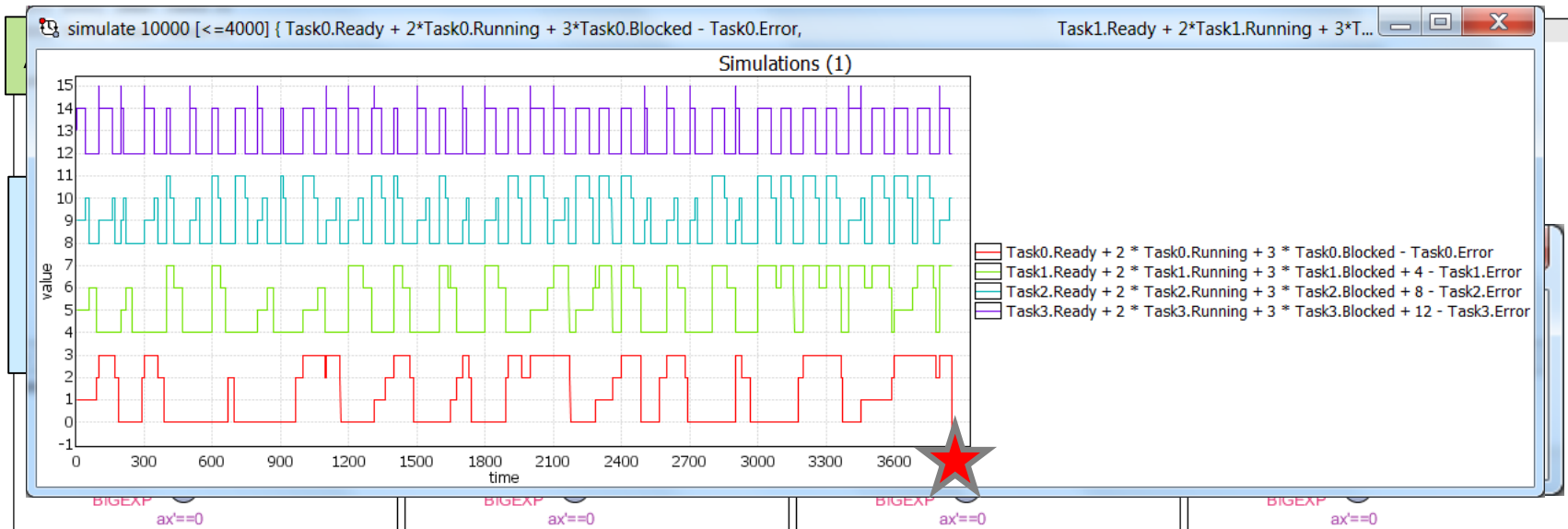
```
// Put an element at the end of the queue
void enqueue(id_t element)
{
    int tmp=0;
    list[len++] = element;
    if (len>0)
    {
        int i=len-1;
        while (i>1 && P[list[i]]>P[list[i-1]])
        {
            tmp = list[i-1];
            list[i-1] = list[i];
            list[i] = tmp;
            i--;
        }
    }
}
```

```
// Remove the front element of the queue
void dequeue()
{
    .....
}
```

Schedulability Analysis

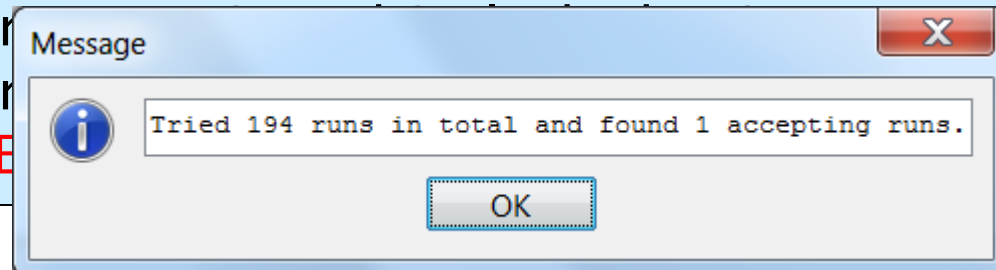


Schedulability Analysis

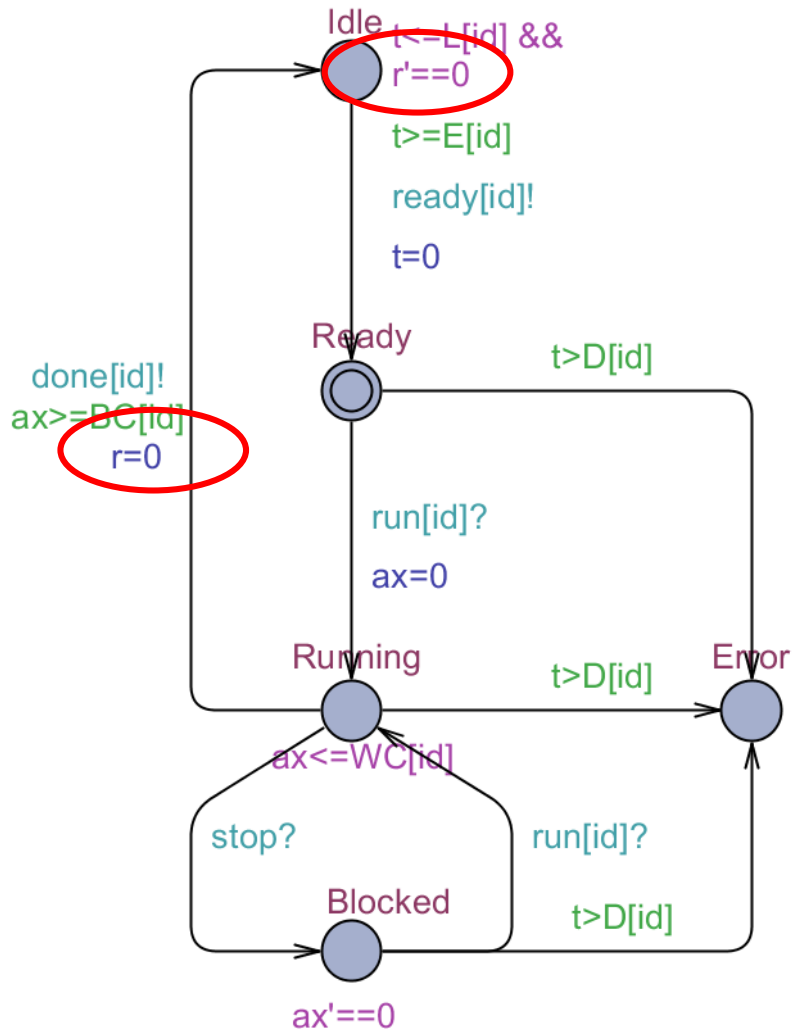


simulate 10000 [<=400]

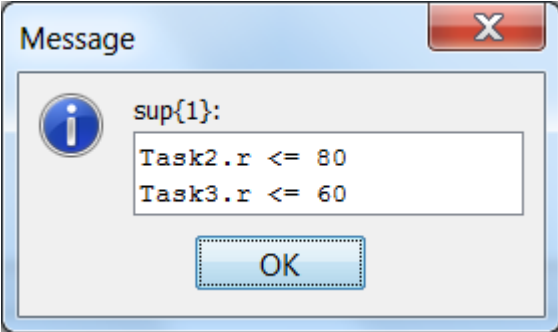
```
{ Task0.Ready + 2*Task0.Running + 3*Task0.Blocked,  
  Task1.Ready + 2*Task1.Running + 3*Task1.Blocked + 4,  
  Task2.Ready + 2*Task2.Running + 3*Task2.Blocked + 8 - Task2.Error,  
  Task3.Ready + 2*Task3.Running + 3*Task3.Blocked + 12 - Task3.Error  
: 1 : (Task0.Error or Task1.Error)
```



Performance Analysis



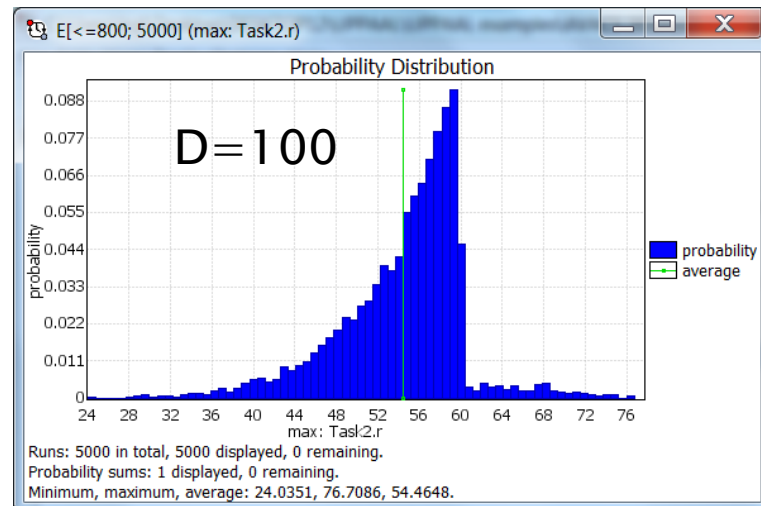
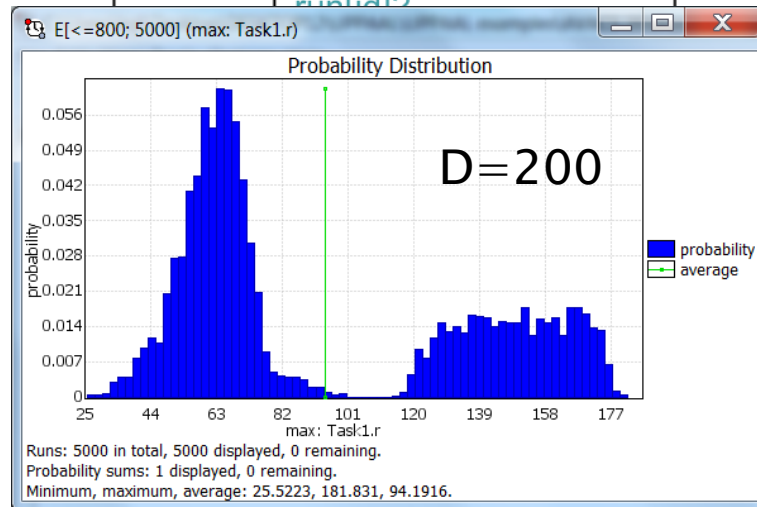
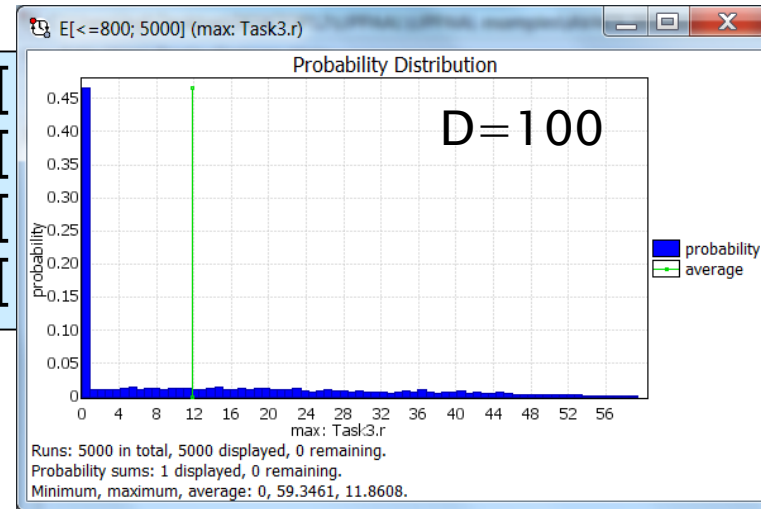
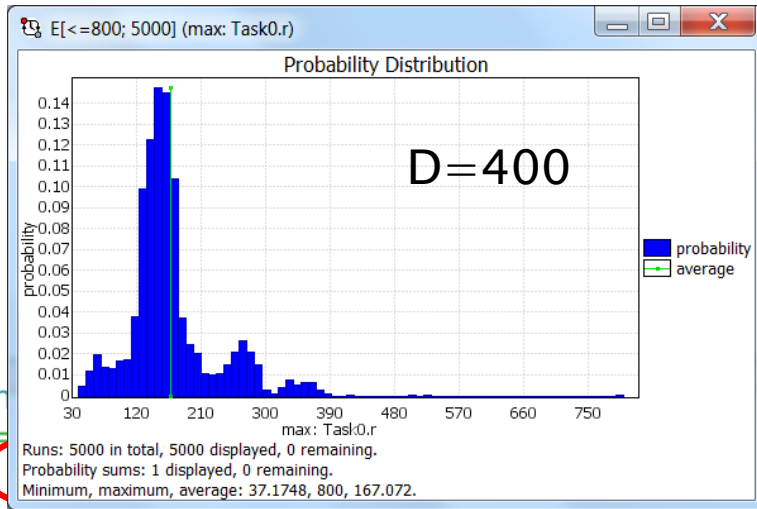
sup : Task2.r, Task3.r



Performance Analysis

don
ax>

E[
E[
E[
E[

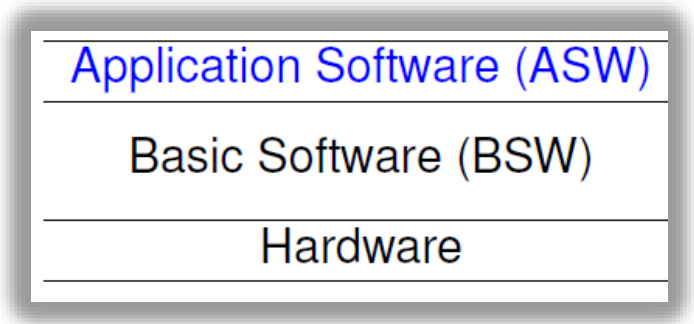


Herschel–Planck Scientific Mission at ESA



Attitude and Orbit Control Software
TERMA A/S Steen Ulrik Palm, Jan Storbak Pedersen, Poul Hougaard

- **Application software (ASW)**
 - built and tested by Terma:
 - does attitude and orbit control, tele-commanding, fault detection isolation and recovery.
- **Basic software (BSW)**
 - low level communication and scheduling periodic events.
- **Real-time operating system (RTEMS)**
 - Priority Ceiling for ASW,
 - Priority Inheritance for BSW
- **Hardware**
 - single processor, a few buses, sensors and actuators

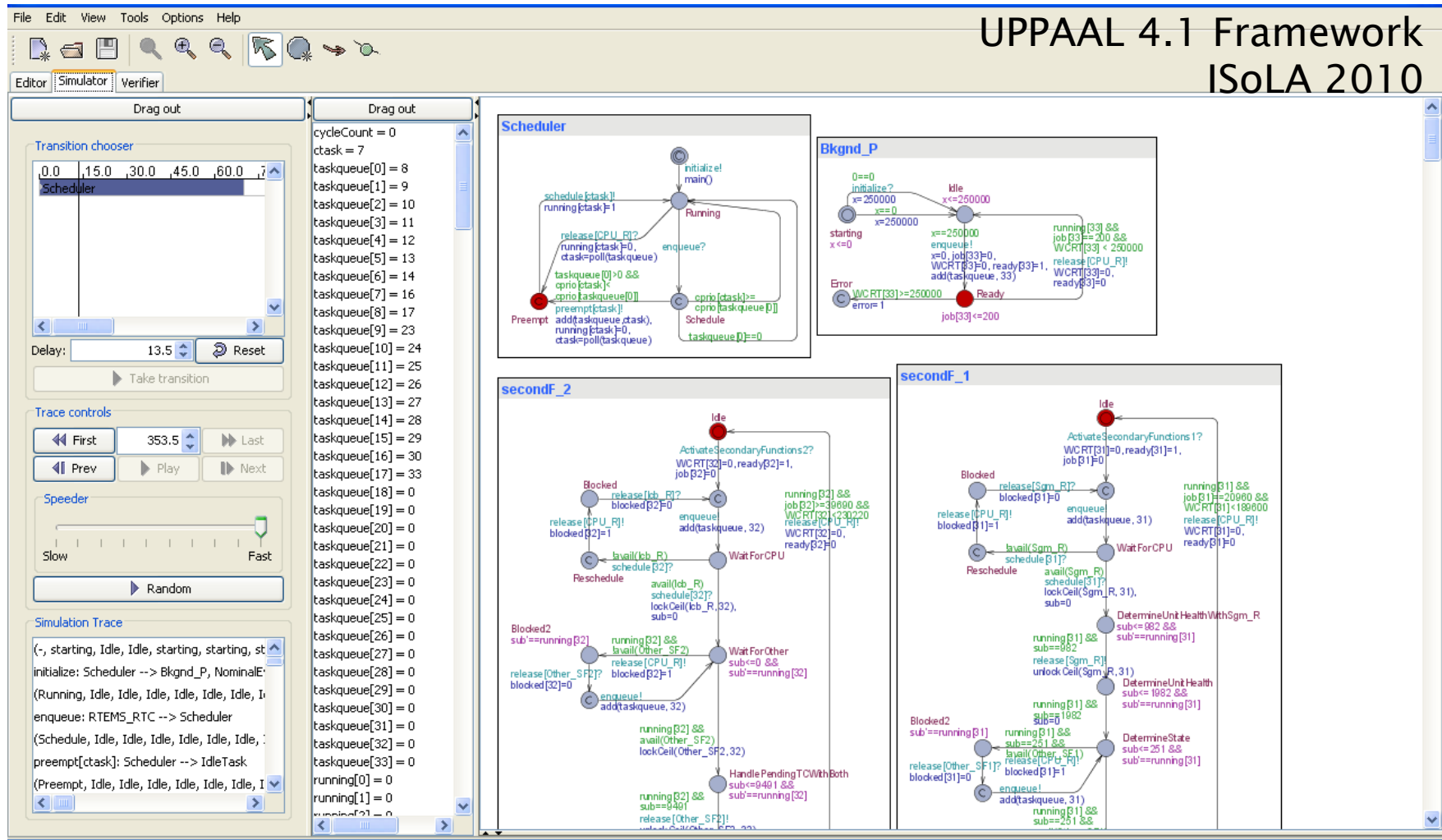


Requirements:

Software tasks should be schedulable.
CPU utilization should not exceed 50% load

Modeling in UPPAAL

UPPAAL 4.1 Framework ISoLA 2010



The screenshot displays the UPPAAL 4.1 Framework interface, which is used for modeling and simulating real-time systems. The interface is divided into several panels:

- Transition chooser:** A list of transitions with a search bar and a delay field (set to 13.5). A "Reset" button and a "Take transition" button are also present.
- Trace controls:** Buttons for "First", "Prev", "Play", "Next", and "Last", along with a "Speeder" slider ranging from "Slow" to "Fast" and a "Random" button.
- Simulation Trace:** A log showing the sequence of transitions and states, such as "initialize: Scheduler --> Bkgnd_P, NominalE", "(Running, Idle, Idle, Idle, Idle, Idle, Idle, I", "enqueue: RTEMS_RTC --> Scheduler", "(Schedule, Idle, Idle, Idle, Idle, Idle, Idle, I", "preempt[ctask]: Scheduler --> IdleTask", and "(Preempt, Idle, Idle, Idle, Idle, Idle, Idle, I".
- Drag out:** A list of variables and their values, including "cycleCount = 0", "ctask = 7", and a list of "taskqueue" arrays from [0] to [33].
- Scheduler:** A state transition diagram showing states like "initialize! main()", "Running", "enqueue?", "cprto [task]=0", "Schedule", "taskqueue [i]=0", "Preempt", and "running [task]=1".
- Bkgnd_P:** A state transition diagram showing states like "starting x<=0", "Idle x<=250000", "x=250000", "enqueue!", "ready [i]=0", "WCRRT [i]=0", "ready [i]=0", "Error error=1", and "Ready job [i]=200".
- secondF_2:** A state transition diagram showing states like "Idle", "Blocked", "Wait For CPU", "Wait For Other", "Handle Pending TCWithBoth", "Reschedule", "Reschedule", "Blocked2", "running [i] && avail [Other_SF2]", "lockCell [Other_SF2, 32]", "enqueue!", "add [taskqueue, 32]", "release [CPU_R]", "blocked [i]=0", "release [CPU_R]", "blocked [i]=1", "release [job_R]", "schedule [i]?", "avail [job_R]", "lockCell [job_R, 32]", "sub=0", "ActivateSecondaryFunctions2?", "WCRRT [i]=0, ready [i]=1, job [i]=0", "running [i] && job [i]=30000 && WCRRT [i]<30000", "release [CPU_R]", "WCRRT [i]=0, ready [i]=0", "running [i] && avail [Other_SF2]", "lockCell [Other_SF2, 32]", "sub=0", "running [i] && sub=9401 && sub==running [i]", "release [Other_SF2]", "sub=0", "Blocked2 sub==running [i]", "running [i] && sub=1982 && sub==running [i]", "release [Sgm_R]", "unlockCell [Sgm_R, 31]", "DetermineUnitHealthWithSgm_R", "sub<=982 && sub==running [i]", "running [i] && sub=251 && sub==running [i]", "release [CPU_R]", "WCRRT [i]=0, ready [i]=0", "DetermineState", "sub<=251 && sub==running [i]", "release [Other_SF1]", "blocked [i]=0", "enqueue!", "add [taskqueue, 31]", "running [i] && sub=251 && sub==running [i]", "release [CPU_R]", "WCRRT [i]=0, ready [i]=0", "running [i] && job [i]=20000 && WCRRT [i]<20000", "release [CPU_R]", "WCRRT [i]=0, ready [i]=0", "Error error=1", "Ready job [i]=200".
- secondF_1:** A state transition diagram showing states like "Idle", "Blocked", "Wait For CPU", "DetermineUnitHealthWithSgm_R", "DetermineState", "Reschedule", "Blocked2", "running [i] && sub=1982 && sub==running [i]", "release [Sgm_R]", "unlockCell [Sgm_R, 31]", "DetermineUnitHealthWithSgm_R", "sub<=982 && sub==running [i]", "running [i] && sub=251 && sub==running [i]", "release [CPU_R]", "WCRRT [i]=0, ready [i]=0", "DetermineState", "sub<=251 && sub==running [i]", "release [Other_SF1]", "blocked [i]=0", "enqueue!", "add [taskqueue, 31]", "running [i] && sub=251 && sub==running [i]", "release [CPU_R]", "WCRRT [i]=0, ready [i]=0", "running [i] && job [i]=20000 && WCRRT [i]<20000", "release [CPU_R]", "WCRRT [i]=0, ready [i]=0", "Error error=1", "Ready job [i]=200".

Blocking & WCRT

ID	Task	Specification			Blocking times			WCRT		
		Period	WCET	Deadline	Terma	UPPAAL	Diff	Terma	UPPAAL	Diff
1	RTEMS_RTC	10.000	0.013	1.000	0.035	0	0.035	0.050	0.013	0.037
2	AswSync_SyncPulseIsr	250.000	0.070	1.000	0.035	0	0.035	0.120	0.083	0.037
3	Hk_SamplerIsr	125.000	0.070	1.000	0.035	0	0.035	0.120	0.070	0.050
4	SwCyc_CycStartIsr	250.000	0.200	1.000	0.035	0	0.035	0.320	0.103	0.217
5	SwCyc_CycEndIsr	250.000	0.100	1.000	0.035	0	0.035	0.220	0.113	0.107
6	Rt1553_Isr	15.625	0.070	1.000	0.035	0	0.035	0.290	0.173	0.117
7	Bc1553_Isr	20.000	0.070	1.000	0.035	0	0.035	0.360	0.243	0.117
8	Spw_Isr	39.000	0.070	2.000	0.035	0	0.035	0.430	0.313	0.117
9	Obdh_Isr	250.000	0.070	2.000	0.035	0	0.035	0.500	0.383	0.117
10	RtSdb_P_1	15.625	0.150	15.625	3.650	0	3.650	4.330	0.533	3.797
11	RtSdb_P_2	125.000	0.400	15.625	3.650	0	3.650	4.870	0.933	3.937
12	RtSdb_P_3	250.000	0.170	15.625	3.650	0	3.650	5.110	1.103	4.007
14	FdirEvents	250.000	5.000	230.220	0.720	0	0.720	7.180	5.153	2.027
15	NominalEvents_1	250.000	0.720	230.220	0.720	0	0.720	7.900	5.873	2.027
16	MainCycle	250.000	0.400	230.220	0.720	0	0.720	8.370	6.273	2.097
17	HkSampler_P_2	125.000	0.500	62.500	3.650	0	3.650	11.960	5.380	6.580
18	HkSampler_P_1	250.000	6.000	62.500	3.650	0	3.650	18.460	11.615	6.845
19	Acb_P	250.000	6.000	50.000	3.650	0	3.650	24.680	6.473	18.207
20	IoCyc_P	250.000	3.000	50.000	3.650	0	3.650	27.820	9.473	18.347
21	PrimaryF	250.000	34.050	59.600	5.770	0.966	4.804	65.470	54.115	11.355
22	RCSControlF	250.000	4.070	239.600	12.120	0	12.120	76.040	53.994	22.046
23	Obt_P	1000.000	1.100	100.000	9.630	0	9.630	74.720	2.503	72.217
24	Hk_P	250.000	2.750	250.000	1.035	0	1.035	6.800	4.953	1.847
25	StsMon_P	250.000	3.300	125.000	16.070	0.822	15.248	85.050	17.863	67.187
26	TmGen_P	250.000	4.860	250.000	4.260	0	4.260	77.650	9.813	67.837
27	Sgm_P	250.000	4.020	250.000	1.040	0	1.040	18.680	14.796	3.884
28	TcRouter_P	250.000	0.500	250.000	1.035	0	1.035	19.310	11.896	7.414
29	Cmd_P	250.000	14.000	250.000	26.110	1.262	24.848	114.920	94.346	20.574
30	NominalEvents_2	250.000	1.780	230.220	12.480	0	12.480	102.760	65.177	37.583
31	SecondaryF_1	250.000	20.960	189.600	27.650	0	27.650	141.550	110.666	30.884
32	SecondaryF_2	250.000	39.690	230.220	48.450	0	48.450	204.050	154.556	49.494
33	Bkgnd_P	250.000	0.200	250.000	0.000	0	0.000	154.090	15.046	139.044



Marius Micusionis

TERMA Case Follow-Up

limit	f=100%			f=95%			[f*WCET, WCET]
	states	mem	time	states	mem	time	
1	1300	51.2	1.47	485077	82.0	0.0	
2	2522	53.7	2.45	806914	82.0	0.0	
4	4981	54.5	4.62	1499700	82.0	0.0	
8							
16							
∞							

	f=90%			f=86%		
	states	mem	time, s	states	mem	time
1	1481162	124.1	4962.8	3348246	186.9	23986.5
2	2414679	139.7	7755.0	5253778	198.7	33299.2
4	4421630	138.3	13720.0	9231399	274.6	51176.6
8	9093562	156.5	31120.3	18240030	364.6	102932.4
16	17798572	176.0	60174.5	35432003	520.4	158816.7
∞	181869652	1682.2	530604.9			

1 Day

6 Days

error may be reachable

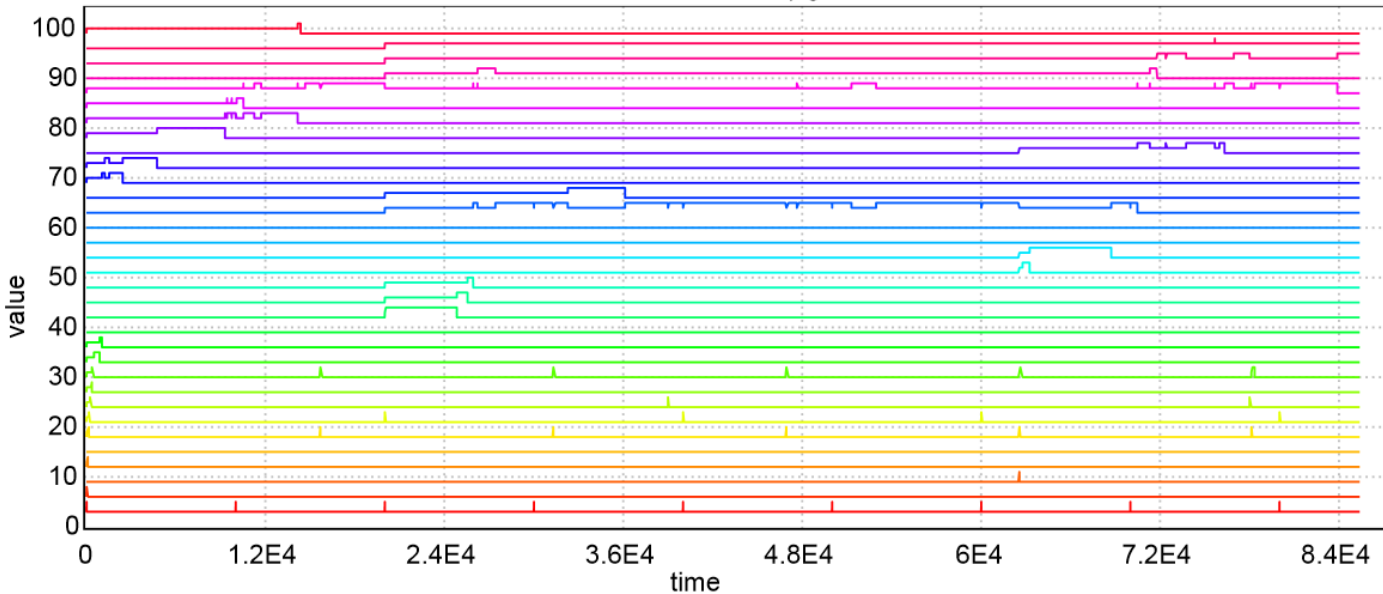
TERMA Case – Statistical MC



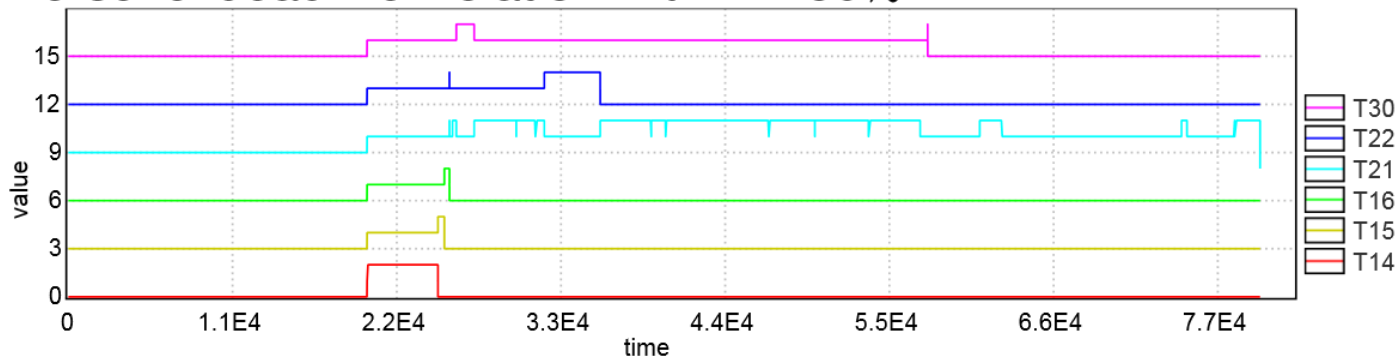
Limit cycles	f %	α	ε	Total traces, #	Error traces #	Error Probability	Earliest cycle	Error offset	Verification time
1	0	0.0100	0.005	105967	1928	0.018194	0	79600.0	1:58:06
1	50	0.0100	0.005	105967	753	0.007106	0	79600.0	2:00:52
1	60	0.0100	0.005	105967	13	0.000123	0	79778.3	2:01:18
1	62	0.0005	0.002	1036757	34	0.000033	0	79616.4	19:52:22
160	63	0.0100	0.05	1060	177	0.166981	0	81531.6	2:47:03
160	64	0.0100	0.05	1060	118	0.111321	1	79803.0	2:55:13
160	65	0.0500	0.05	738	57	0.077236	3	79648.0	2:06:55
160	66	0.0100	0.05	1060	60	0.056604	2	82504.0	2:62:44
160	67	0.0100	0.05	1060	26	0.024528	1	79789.0	2:64:20
160	68	0.0100	0.05	1060	3	0.002830	67	81000.0	2:67:08
640	69	0.0100	0.05	1060	8	0.007547	114	80000.0	12:23:00
640	70	0.0100	0.05	1060	3	0.002830	6	88070.0	12:30:49
1280	71	0.0100	0.05	1060	2	0.001887	458	80000.0	25:19:35

TERMA Case - Conclusion

Herschel simulation run with $f = 90\%$:



Herschel deadline violation with $f = 50\%$:



- Frits Vaandrager: A first introduction to UPPAAL
- Alexandre David, Kim G Larsen: More features in UPPAAL
- Alexandre David, Kim G. Larsen, Axel Legay, Marius Mikucionis, Zheng Wang: Time for Statistical Model Checking of Real-Time Systems. CAV 2011: 349–355.
- Alexandre David, Kim G. Larsen, Axel Legay, Marius Mikucionis, Danny Bøgsted Poulsen: Uppaal SMC tutorial. STTT 17(4): 397–415 (2015)
- Kim Guldstrand Larsen: Validation, Synthesis and Optimization for Cyber-Physical Systems. TACAS (1) 2017: 3–20
- Alexandre David, Peter Gjør Jensen, Kim Guldstrand Larsen, Marius Mikucionis, Jakob Haahr Taankvist: Uppaal Stratego. TACAS 2015: 206–211

Applications

(some)



Bang & Olufsen IR-Link

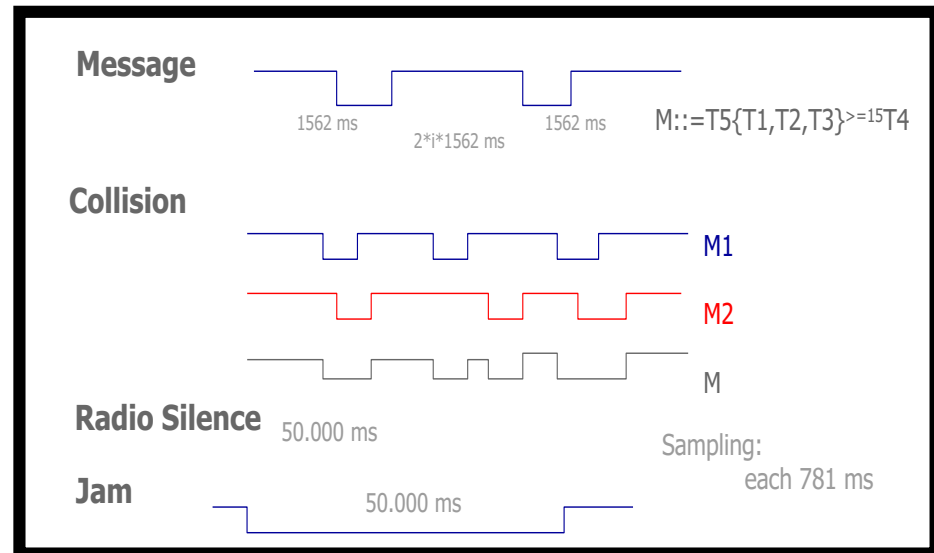
- Bug known to exist for 10 years
- Ill-described:
 - 2.800 lines of assembler code + 3 flowchart + 1 B&O eng.
- 3 months for modeling.
- UPPAAL detects error with 1.998 transition steps (shortest)
- Error trace was confirmed in B&O laboratory.
- Error corrected and verified in UPPAAL.

Arne Skou, Klaus Havelund



Bang & Olufsen IR-Link

- Bug known to exist for 10 years
- Ill-described:
 - 2.800 lines of assembler code + 3 flowchart + 1 B&O eng.
- 3 months for modeling.
- UPPAAL detects error with 1.998 transition steps (shortest)
- Error trace was confirmed in B&O laboratory.
- Error corrected and verified in UPPAAL.



1st RTSS'97 talk, Klaus Havelund

Philips Bounded Retransmission Protocol



Pedro D'Argenio
Joost-Pieter Katoen
Theo Ruys
Jan Tretmans

```
Science):  
> I should tell you that I am quite dissappointed with this new  
> release of Uppaal ;). You take all the fun out of it!!. With this  
> new releas I could verify everything in a couple of minutes,  
> including a couple of properties that where impossible before!!!  
> Moreover, I was playing with the simulator and I found a silly  
> deadlock in the specification.  
>  
> ...  
>  
> I found this new Uppaal a quite huge leap from the previous version.  
> As a user, I have had a really good first impression. I will  
> compile a list of comments for tomorrow afternoon.  
Regards,
```

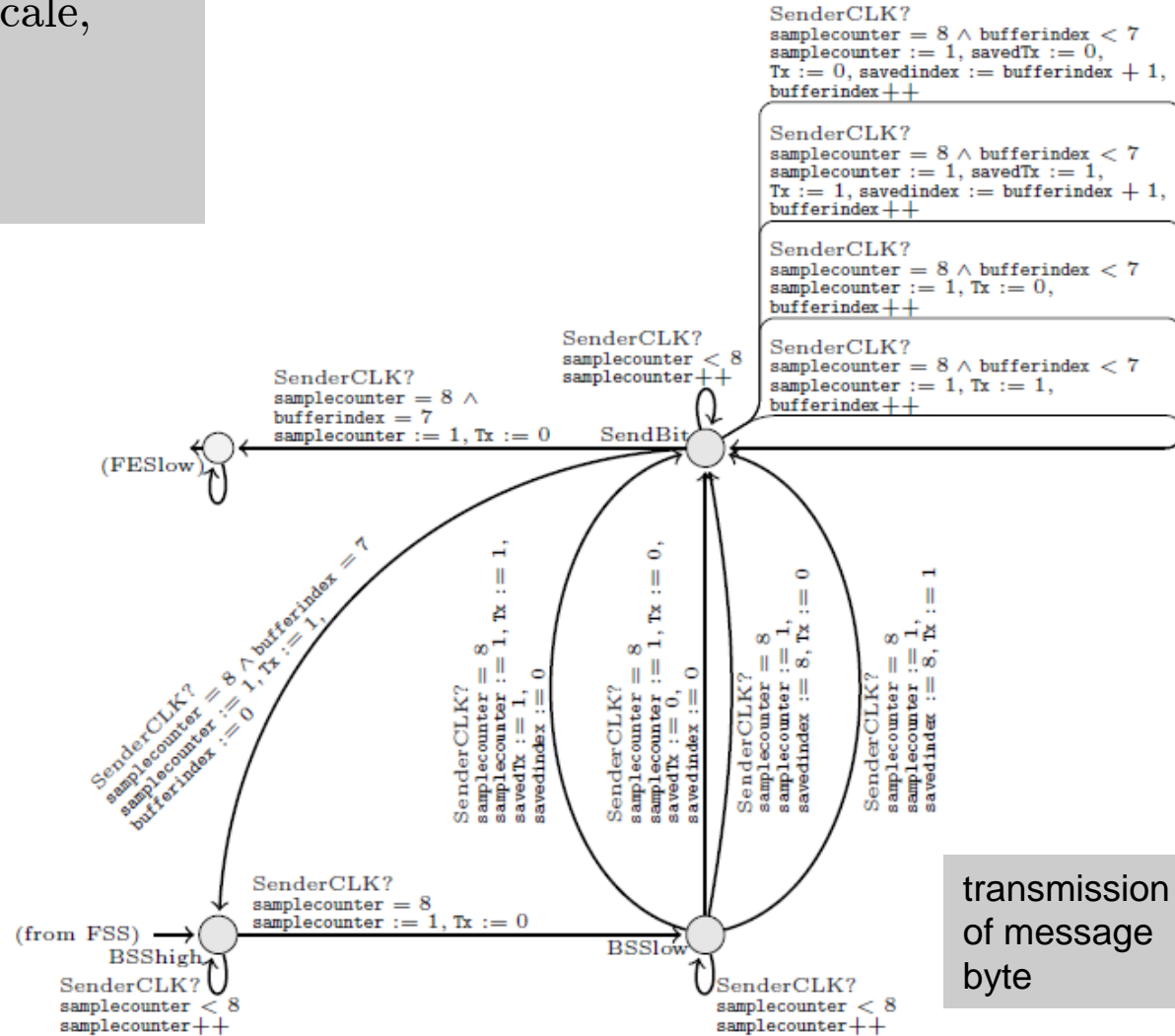
FlexRay

[Gerke, Ehlers, Finkbeiner, Peters, 2010]



BMW, Bosch, Daimler, Freescale,
General Motors, NXP
Semiconductors, and
Volkswagen

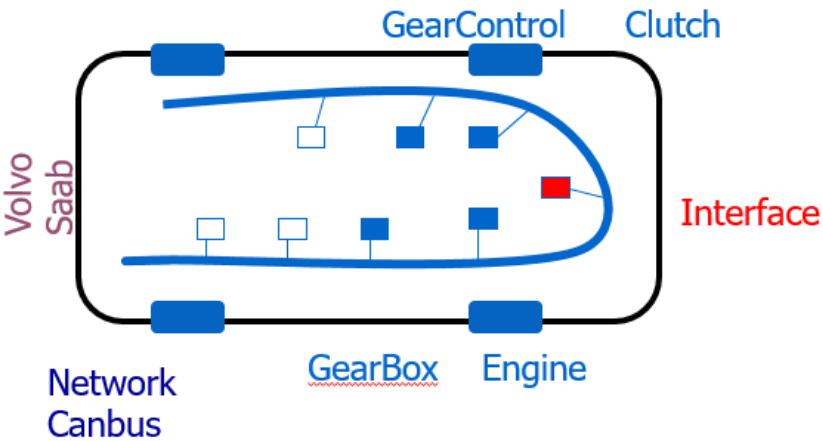
Fault-tolerance
Timed hardware model
Parameterized error models
(glitches, jitter)
Voting & bit-clock alignment



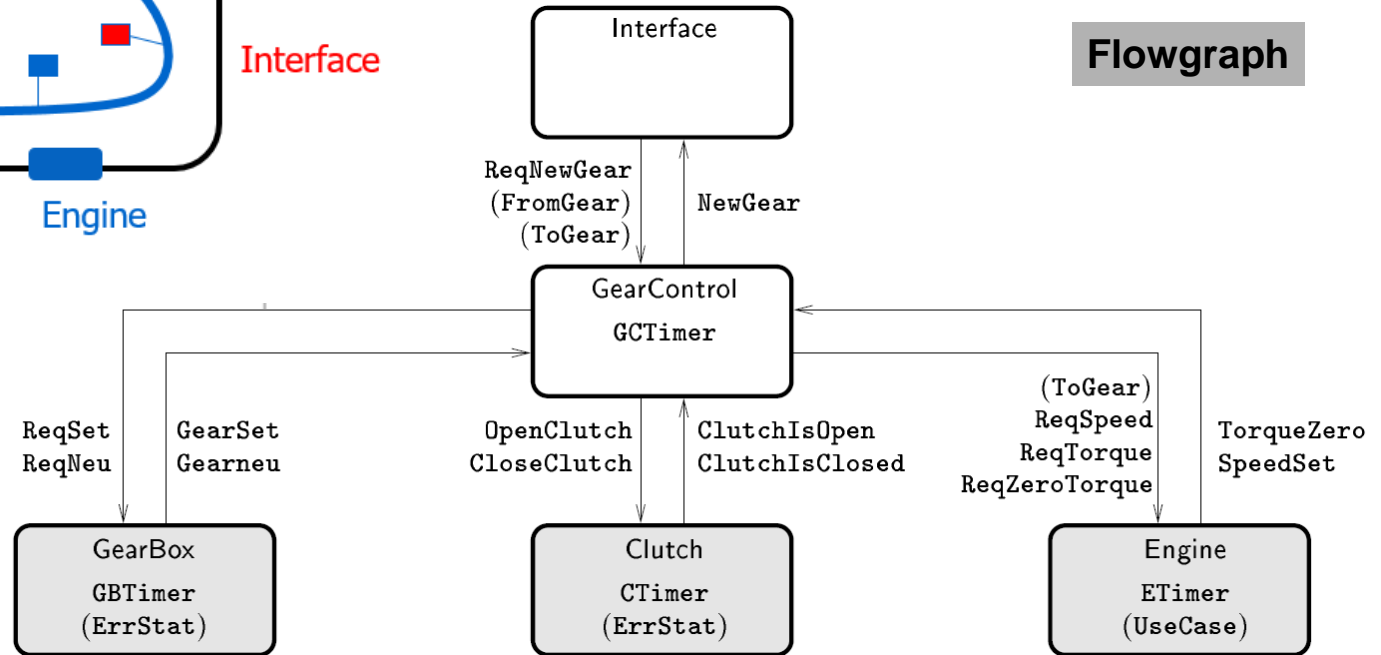
transmission
of message
byte

Gear Controller

with MECEL AB



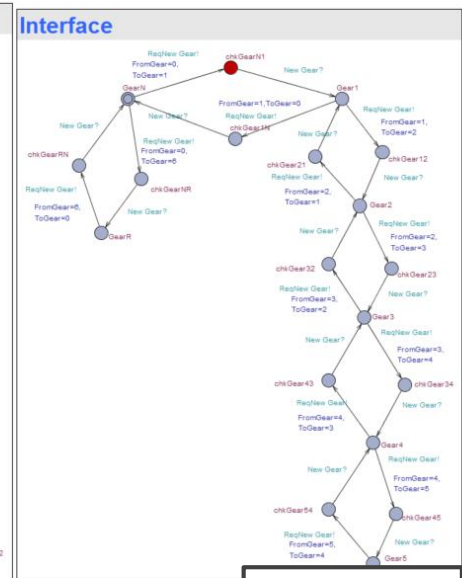
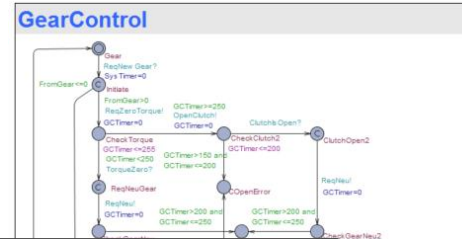
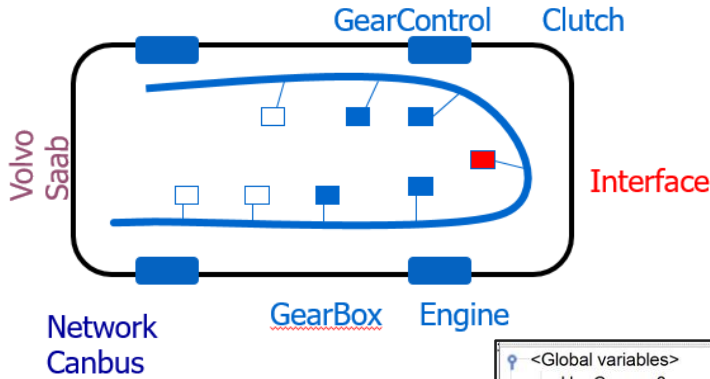
Flowgraph



Magnus Lindahl
Paul Pettersson
Wang Yi
2001

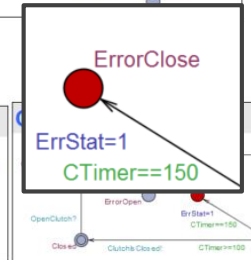
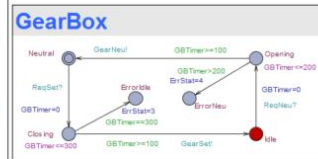
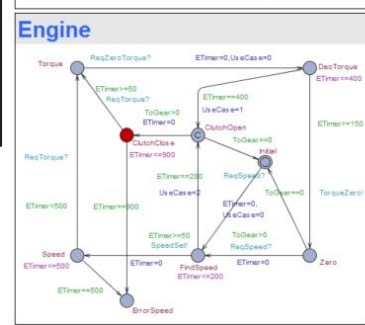
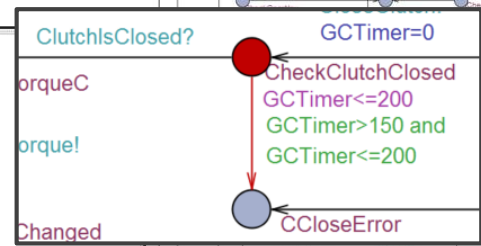
Gear Controller

with MECEL AB



```

<Global variables>
  UseCase = 0
  FromGear = 0
  ToGear = 0
  ErrStat = 0
<Constraints>
  CTimer ≥ 0
  ETimer ≥ 0
  GBTimer ≥ 0
  GCTimer ≥ 0
  SysTimer ≥ 0
  GearControl.GCTimer ≥ 0
  CTimer = ETimer
  ETimer = GBTimer
  GBTimer = GCTimer
  GCTimer = SysTimer
  SysTimer = GearControl.GCTimer
  GearControl.GCTimer = CTimer
    
```

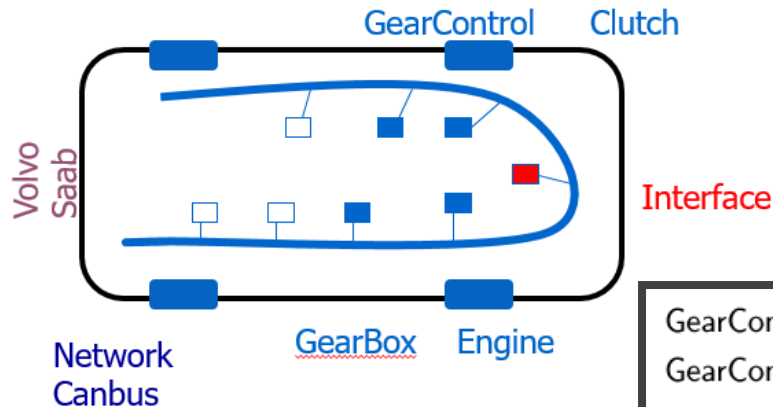


Magnus Lindahl
Paul Pettersson
Wang Yi
2001

Timed Automata Models

Gear Controller

with MECEL AB



Requirements

```
GearControl@Initiate  $\rightsquigarrow_{\leq 1500}$  ( ( ErrStat = 0 )  $\Rightarrow$  GearControl@GearChanged )
GearControl@Initiate  $\rightsquigarrow_{\leq 1000}$ 
    ( ( ErrStat = 0  $\wedge$  UseCase = 0 )  $\Rightarrow$  GearControl@GearChanged )
Clutch@ErrorClose  $\rightsquigarrow_{\leq 200}$  GearControl@CCloseError
Clutch@ErrorOpen  $\rightsquigarrow_{\leq 200}$  GearControl@COpenError
GearBox@ErrorIdle  $\rightsquigarrow_{\leq 350}$  GearControl@GSetError
GearBox@ErrorNeu  $\rightsquigarrow_{\leq 200}$  GearControl@GNeuError
Inv ( GearControl@CCloseError  $\Rightarrow$  Clutch@ErrorClose )
Inv ( GearControl@COpenError  $\Rightarrow$  Clutch@ErrorOpen )
Inv ( GearControl@GSetError  $\Rightarrow$  GearBox@ErrorIdle )
Inv ( GearControl@GNeuError  $\Rightarrow$  GearBox@ErrorNeu )
Inv ( Engine@ErrorSpeed  $\Rightarrow$  ErrStat  $\neq$  0 )
Inv ( Engine@Torque  $\Rightarrow$  Clutch@Closed )
```

Magnus Lindahl
Paul Pettersson
Wang Yi
2001

UPPAAL Model Checking – Demo



engine-classic.xta - UPPAAL

File Edit View Tools Options Help

Editor Simulator ConcreteSimulator Verifier

Overview

```
E<> GearControl.GearChanged
E<> ( Interface.Gear5 )
E<> ( Interface.GearR )
E<> ( GearControl.GearChanged and ( SysTimer<=1000 ) )
A[] not ( GearBox.Neutral and ( Interface.Gear1 or Interface.Gear2 or Interface.Gear3 or In...
A[] not ( GearBox.Idle and Interface.GearN )
A[] ( Interface.GearN imply GearBox.Neutral )
A[] ( ( ErrStat==0 and UseCase==0 and SysTimer>=900 ) imply ( GearControl.GearChanged...
E<> ( ErrStat==0 and UseCase==0 and SysTimer>899 and SysTimer<900 and not ( Gea...
A[] ( ( ErrStat==0 and UseCase==0 and ( SysTimer<150 ) ) imply not ( GearControl.Gear...
```

Check
Insert
Remove
Comments

Query

```
E<> GearControl.GearChanged
```

Comment

```
P1. It is possible to change gear.
```


UPPAAL Model Checking – Demo



engine-classic.xta - UPPAAL

File Edit View Tools Options Help

Editor Simulator ConcreteSimulator Verifier

Overview

```
A[] ( ( ErrStat==0 and UseCase==2 and SysTimer>=1205 ) imply      ( GearControl.GearChanged... ^
E<> ( ErrStat==0 and UseCase==2 and SysTimer>1204 and          SysTimer<1205 and          not ( Ge...
A[] ( ( UseCase==2 and ( SysTimer<450 ) ) imply not ( GearControl.GearChanged or GearContro...
E<> ( UseCase==2 and GearControl.GearChanged and ( SysTimer==450 ) )
A[] ( ( ErrStat==0 and UseCase==2 and FromGear>0 and ToGear>0 and SysTimer<750 ) imply not ...
E<> ( ErrStat==0 and UseCase==2 and FromGear>0 and ToGear>0 and GearControl.GearChanged and...
A[] ( ( Clutch.ErrorClose and ( GearControl.GCTimer>200 ) ) imply GearControl.CCloseError )
A[] ( GearControl.CCloseError imply Clutch.ErrorClose )
A[] ( ( Clutch.ErrorOpen and ( GearControl.GCTimer>200 ) ) imply GearControl.COpenError )
A[] ( ( GearControl.COpenError ) imply Clutch.ErrorOpen )
```

Check
Insert
Remove
Comments

Query

```
A[] ( ( Clutch.ErrorClose and ( GearControl.GCTimer>200 ) ) imply
      GearControl.CCloseError )
```

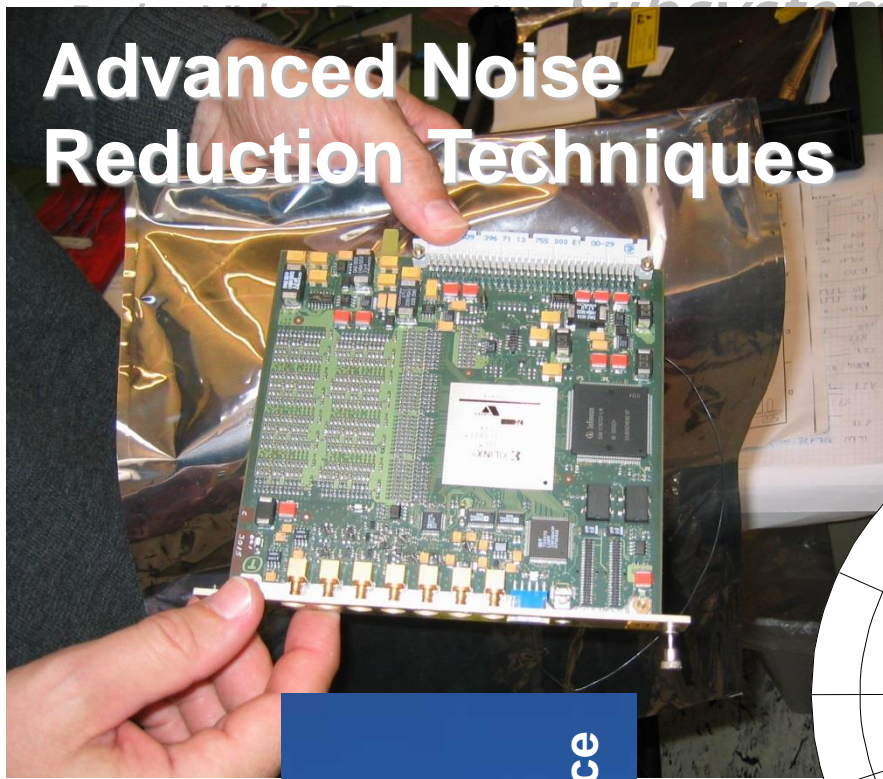
Comment

P9. Clutch Errors.

a) If the clutch is not closed properly (i.e. a timeout occurs) the gearbox controller will enter the location CCloseError within 200 ms.

TERMA A/S (2004)

Memory Management for Radars



Advanced Noise Reduction Techniques

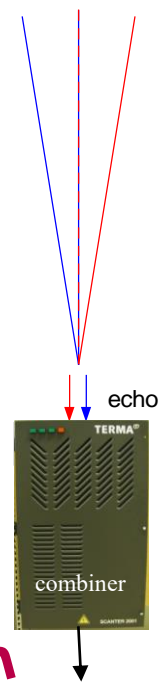


Coastal Surveillance

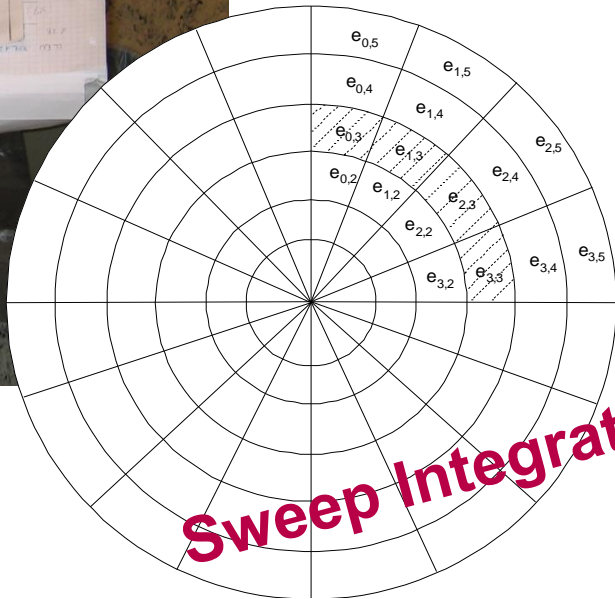


Airport Surveillance

9.170 GHz
9.438 GHz



Frequency Diversity



Sweep Integration

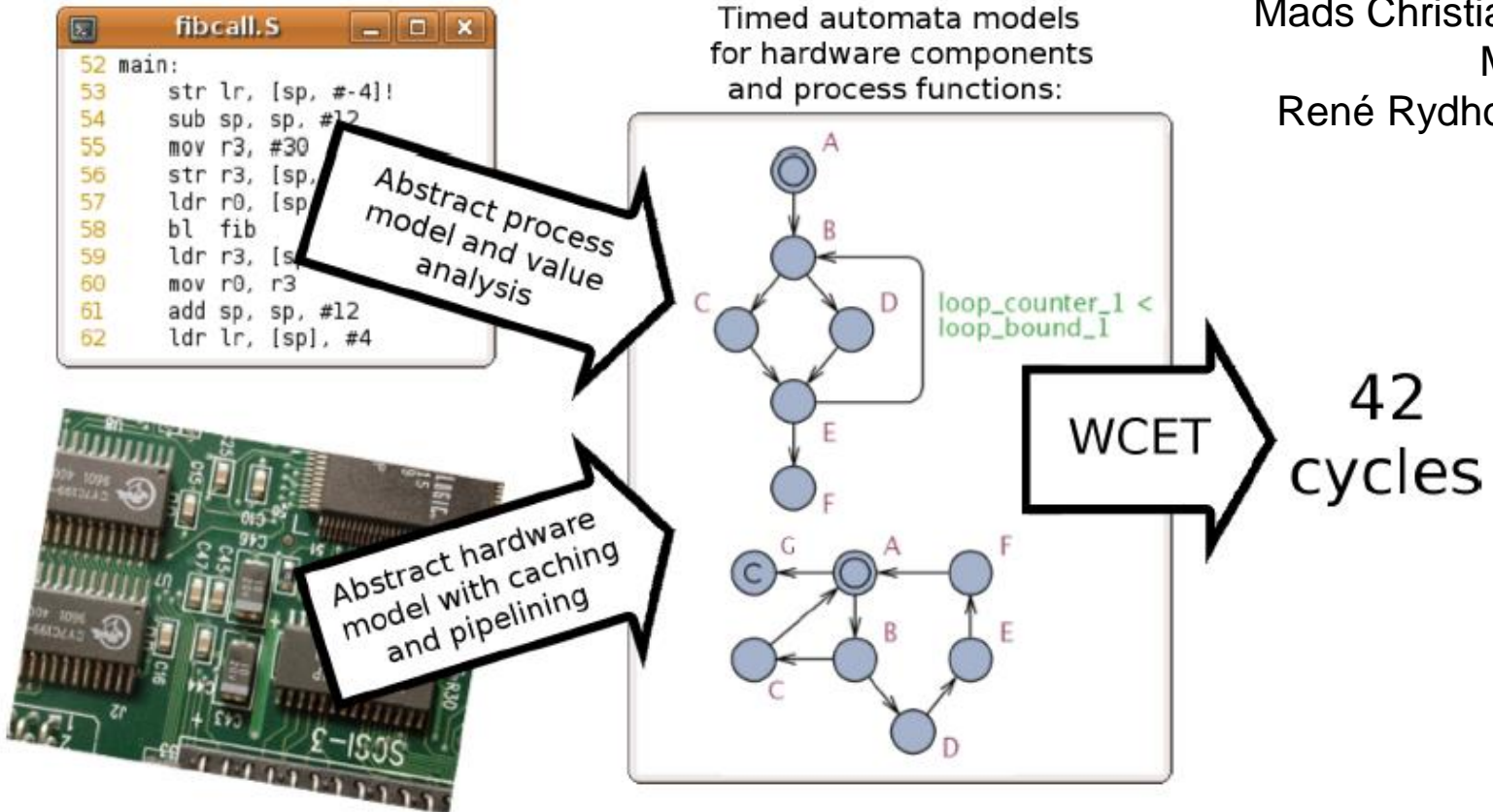




Attitude and Orbit Control Software

TERMA A/S Steen Ulrik Palm, Jan Storbak Pedersen, Poul Hougaard

with
Andreas Dalsgaard
Mads Christian Olesen
Martin Toft
René Rydhof Hansen



Controllers in UPPAAL



- Gearbox Controller [TACAS'98]
- Bang & Olufsen Power Controller [RTPS'99, FTRTFT'2k]
- SIDMAR Steel Production Plant [RTCSEA'99, DSVV'2k]
- Real-Time RCX Control-Programs [ECRTS'2k]
- Terma, Verification of Memory Management for Radar (2001)
- Scheduling Lacquer Production (2005)
- Memory Arbiter Synthesis and Verification for a Radar Memory Interface Card [NJC'05]
- Adapting the UPPAAL Model of a Distributed Lift System, 2007
- Analyzing a χ model of a turntable system using Spin, CADP and Uppaal, 2006
- Designing, Modelling and Verifying a Container Terminal System Using UPPAAL, 2008
- Model-based system analysis using Chi and Uppaal: An industrial case study, 2008
- Climate Controller for Pig Stables, 2008
- Optimal and Robust Controller for Hydraulic Pump, 2009

(Wireless) Protocols in UPPAAL



- **Bang & Olufsen IR Link**
- **Philips Audio Protocol**
- Collision-Avoidance Protocol
- **Bounded Retransmission Protocol**
- TDMA Protocol
- Multimedia Streams
- ATM ABR Protocol
- Lamport's Leader Election Protocol
- ABB Fieldbus Protocol
- IEEE 1394 Firewire Root Contention
- Bluetooth Protocol
- Distributed Agreement Protocol
- **FlexRay**
- **CHES MAC Protocol**
- **Proprietary WSN, Other Big Danish Company**
- **MESH Protocol (MAC & Routing), NEOCORTEC**

UPPAAL as a Back-End



- Voodoo: verification of object-oriented designs using Uppaal, 2004
- Moby/RT: A Tool for Specification and Verification of Real-Time Systems, 2000
- Formalising the ARTS MPSOC Model in UPPAAL, 2007
- Timed automata translator for Uppaal to PVS
- Component-Based Design and Analysis of Embedded Systems with UPPAAL PORT, 2008
- Verification of COMDES-II Systems Using UPPAAL with Model Transformation, 2008
- METAMOC: Modular WCET Analysis Using UPPAAL, 2010.

← → ↻ ⓘ www.uppaal.org 🔍 ☆ ⋮

UPPAAL
Home

RELATED TOOLS: TIMES | Stratego | CORA | TRON | TIGA | SMC | COVER | PORT | PRO ▲

Home | About | Documentation | Download | Examples | Web Help | Bugs

UPPAAL is an integrated tool environment for modeling, validation and verification of real-time systems modeled as networks of timed automata, extended with data types (bounded integers, arrays, etc.).

The tool is developed in collaboration between the [Department of Information Technology](#) at Uppsala University, Sweden and the [Department of Computer Science](#) at Aalborg University in Denmark.

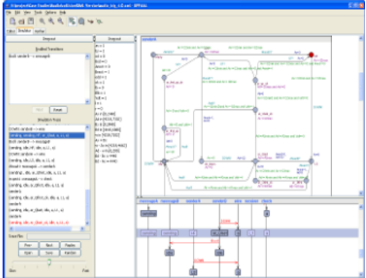


Figure 1: UPPAAL on screen.

Download

News: The current official release is UPPAAL 4.0.13 (Sep 27, 2010). Compared to version 3, the 4.0 release is the result of over 2.5 years of additional development, and many new features and improvements are introduced (see also this [release note](#) and the web help section [new features](#)). To support models created in previous versions of UPPAAL, version 4.0 can convert most old models directly from the GUI (alternatively it can be run in 3.4 compatibility mode by defining the environment variable `UPPAAL_OLD_SYNTAX`, see also item 2 of the [FAQ](#)).

Since Feb 26 2008, we also distribute development snapshots of the tool. The current version is

License

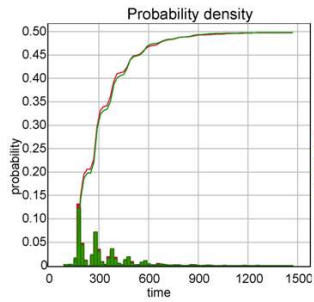
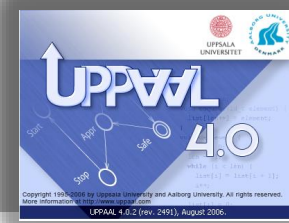
The UPPAAL tool is free for non-commercial applications in academia **only**. For commercial applications a commercial license is required. Please see the [Download](#) section or www.uppaal.com for more information.

To find out more about UPPAAL, read this short [introduction](#). Further information may be found at this web site in the pages [About](#), [Documentation](#), [Download](#), and [Examples](#).

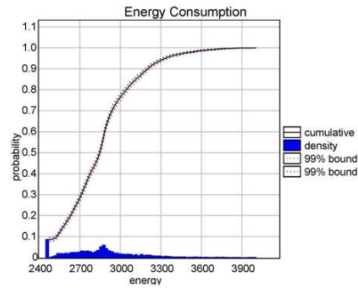
Mailing Lists

UPPAAL has an open [discussion forum](#) group at Yahoo!Groups intended for users of the tool. To join or post to the forum, please refer to the information at the [discussion](#)

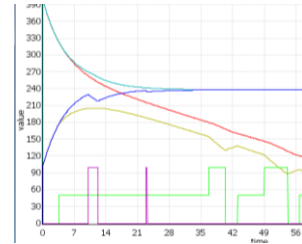
More Applications



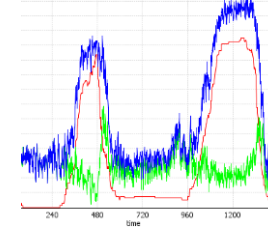
FIREWIRE



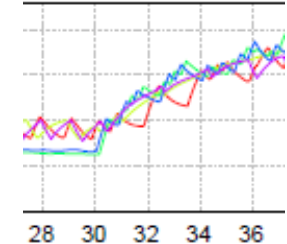
BLUETOOTH



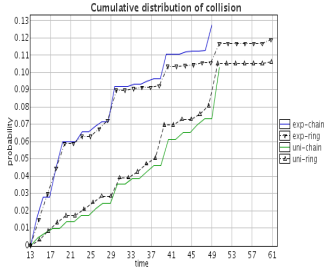
Battery Scheduling



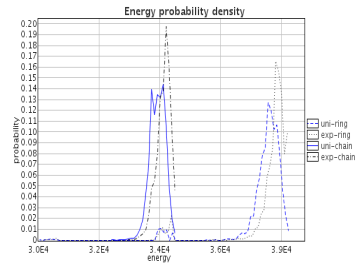
Smart Grid



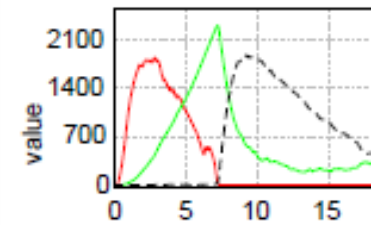
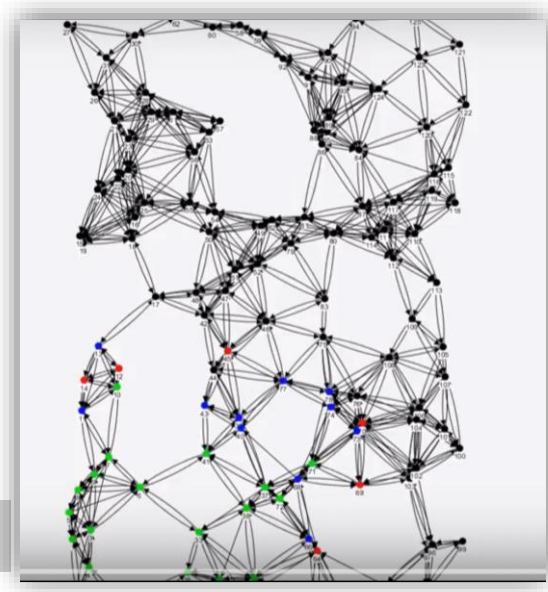
Energy Aware Buildings



10 node LMAC



Mesh Network



Genetic Oscillator