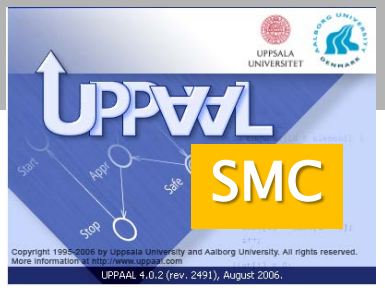
Statistical Model Checking for Stochastic Hybrid Systems



Kim G. Larsen

Alexandre David, Marius Mikucionis,

Peter Bulychev, Axel Legay, Dehui Du, Guangyuan Li,
Danny B. Poulsen, Amélie Stainer, Zheng Wang

AALBORG UNIVERSITY DENMARK

Cyber-Physical Systems

- Complex systems that tightly integrate m ed c ts (hardware and software) with no computing physical elements such as e m components.

Real Time

Resources

Hybrid Systems

Stochasticity

Overview

- Stochastic Hybrid Systems
- Metric Interval Temporal Logic
- UPPAAL SMC

- Schedulability and Performance Analysis of Mixed Critical Systems

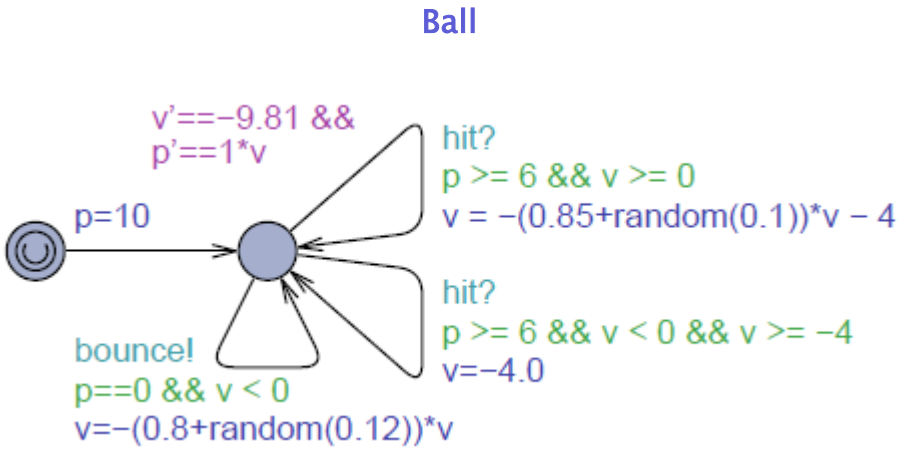
- Energy Aware Buildings

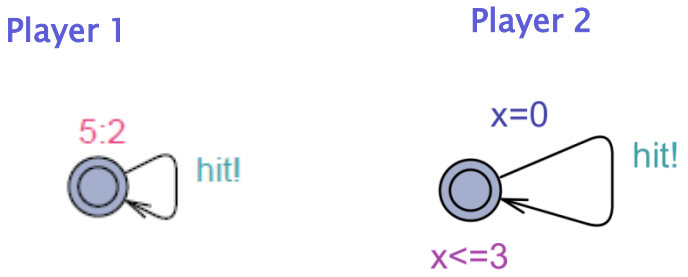
- Conclusion

Hybrid Automata

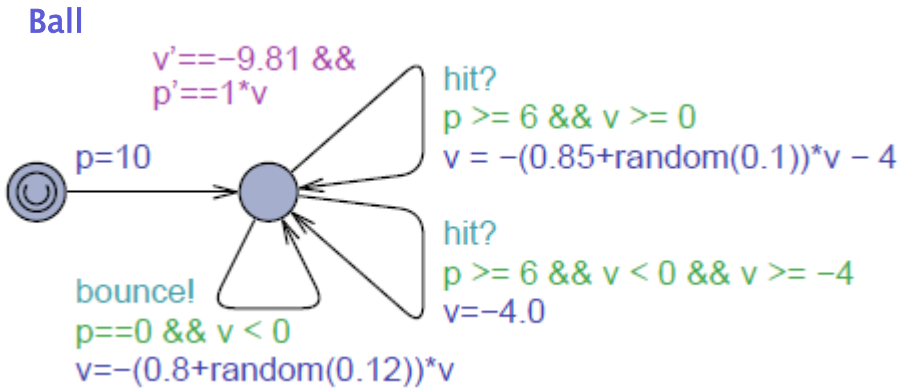
H=(L, l_0,§ , X,E,F,Inv)
where

- L set of locations
- l_0 initial location
- § =§ $_i$ [§ $_o$ set of actions
- X set of continuous variables
  valuation °: X! R $(=R^X)$
- E set of edges (l,g,a,Á,l') with gμR^X and Á$\mu R^X \times R^X$ and a2§
- For each l a delay function
  F(l): $R_{>0} \times R^X$! R^X
- For each l an invariant
  Inv(l)μR^X





Hybrid Automata



(p = 10; v = 0) $\xrightarrow{d}$ (p = 10 ¡ 9:81=2d^2; v = ¡ 9:81d)

bounce! $\xrightarrow{}$ (p = 0; v = 14:02 ¢0:83) at d = 1:43

$\xrightarrow{d}$ (p = 6:92; v = 0) at d = 1:18

$\xrightarrow{d}$ (p = 0; v = 11:51) at d = 1:18

bounce! $\xrightarrow{}$: : :

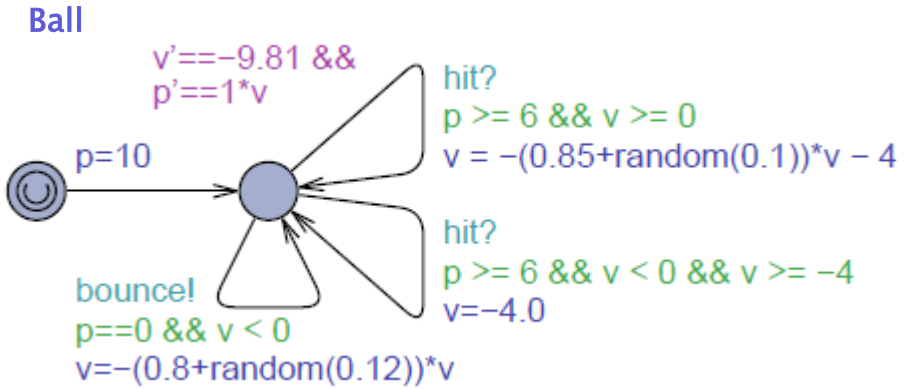
Semantics

- States
  - (l,ν) where $\nu \in R^X$
- Transitions

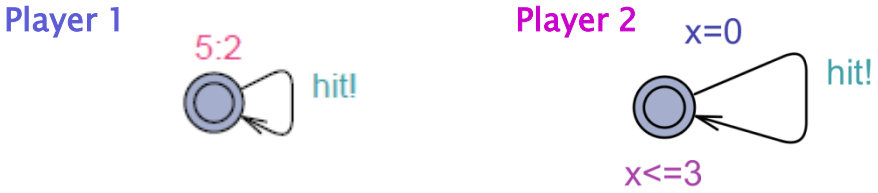
$(l,\nu) \xrightarrow{d} (l,\nu')$ where
$\nu' = F(l)(d,\nu)$
provided $\nu' \in Inv(l)$

$(l,\nu) \xrightarrow{a} (l',\nu')$ if
there exists $(l,g,a,Á,l') \in E$
with $\nu \in g$ and
$(\nu,\nu') \in Á$ and
$\nu' \in Inv(l')$

Stochastic Hybrid Automata



(p = 10; v = 0) !^{d} (p = 10 ¡ 9:81=2d^2; v = ¡ 9:81d)
bounce!
! (p = 0; v = 14:02 ¢0:83) at d = 1:43



Stochastic Semantics

For each state s=(l,º)

Delay density function*

$^{1}{}_{s}$: $R_{>0}$! R

Output Probability Function

$^{\circ}{}_{s}$: § $_{o}$! [0,1]

Next-state density function*

$^{´}{}_{a\,s}$: St! R

where a2§ .

$$Pr_1[hit!\ bounce!] = \int_{t=0}^{t=1.43} 2.5\, e^{-2.5t}\, dt$$

$$= [-e^{-2.5t}]_0^{1.43} = 0.97$$

$$Pr_2[hit!\ bounce!] = \int_{t=0}^{t=1.43} {}^{1}/_{3}\, dt$$

$$= [{}^{1}/_{3}\ t\ ]_0^{1.43} = 0.48$$

* Dirac's delta functions for deterministic delays / next state

Stochastic Hybrid Automata

UPPAAL SMC

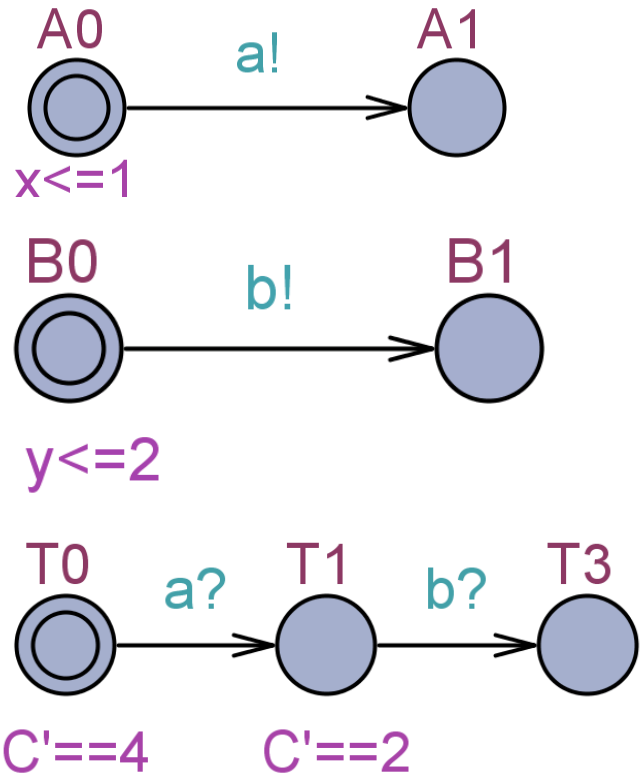
Uniform distributions (bounded delay)
Exponential distributions (unbounded delay)
Syntax for discrete probabilistic choice
Distribution on next state by use of `random`
Hybrid flow by use of ODEs
+ usual stuff (structured variables, user-defined types
user-defined functions, ….)

Networks

Repeated races between components for outputting

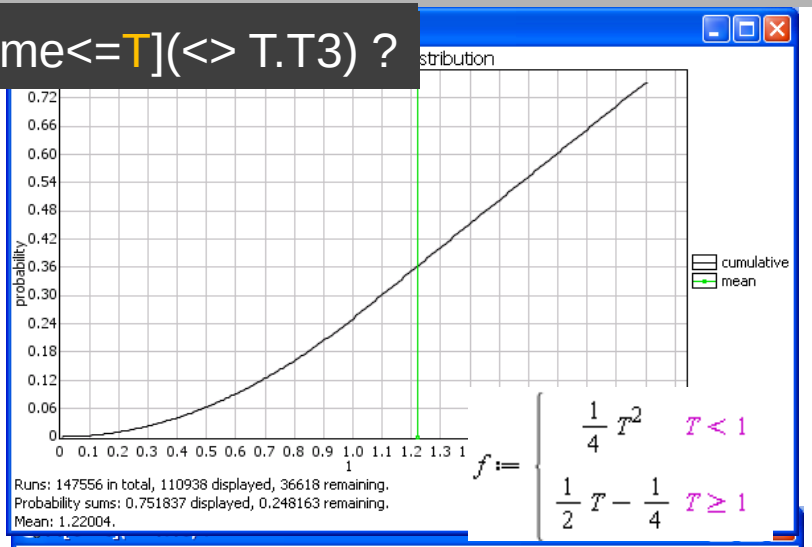
* Dirac's delta functions for deterministic delays / next state

Stochastic Semantics NTAs

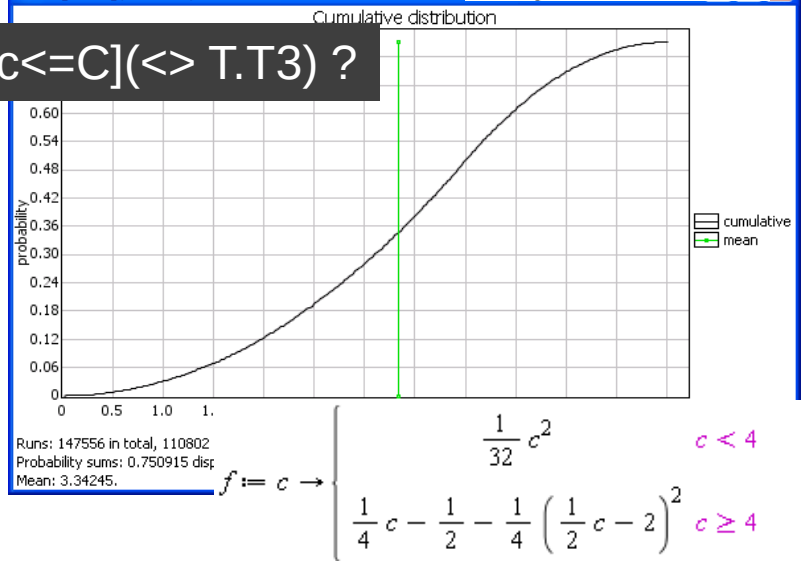


Composition = Race between components for outputting

Pr[time<=T](<> T.T3) ?

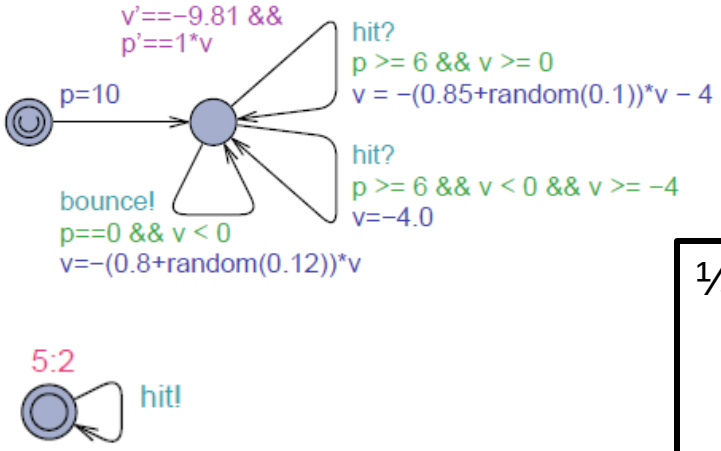


Pr[c<=C](<> T.T3) ?



Stochastic Semantics of NHAs

$\mathcal{A}$



Assumptions:
Component SHAs are:
- Input enabled
- Deterministic
- Disjoint set of output actions

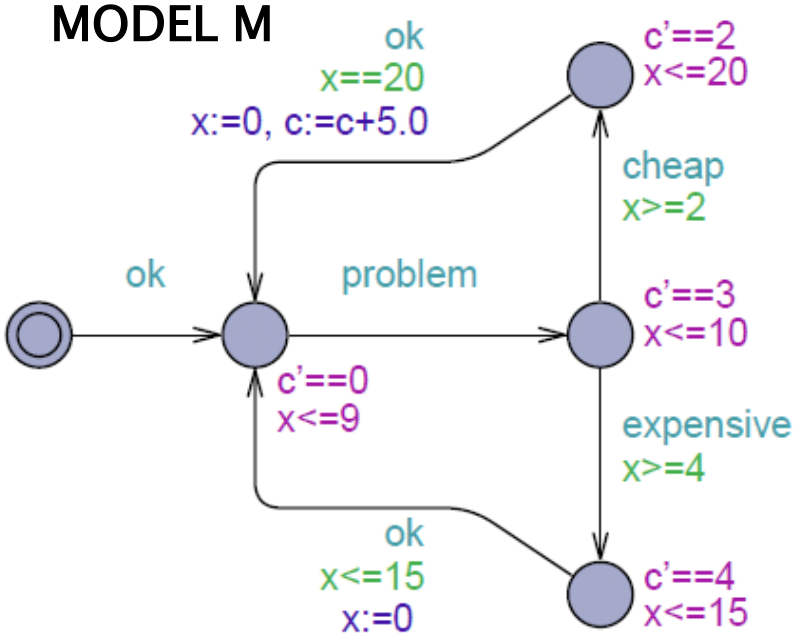
¼(**s** , a_1 a_2 …. a_n) :
the set of maximal runs from **s** with a prefix
t_1 a_1 t_2 a_2 … t_n a_k
for some $t_1,\ldots,t_n$ 2 **R**.

$$\mathbb{P}_{\mathcal{A}}(\pi(\mathbf{s}, a_1 \ldots a_n)) =$$
$$\int_{t\geq 0} \mu_{s_c}(t) \cdot \Big(\prod_{j\neq c} \int_{\tau>t} \mu_{s_j}(\tau)d\tau\Big) \cdot \gamma_{s_c{}^t}(a_1) \cdot \int_{\mathbf{s}'} \Big(\prod_j \eta^{a_1}_{s_j^t}(s'_j) \cdot \mathbb{P}_{\mathcal{A}}(\pi(\mathbf{s}', a_2 \ldots a_n))d\mathbf{s}'\Big)\, dt$$

where $c = c(a_1)$, and as base case we take $P_{\mathcal{A}}(\pi(\mathbf{s}), \varepsilon) = 1$.

Logical Properties- WMITL

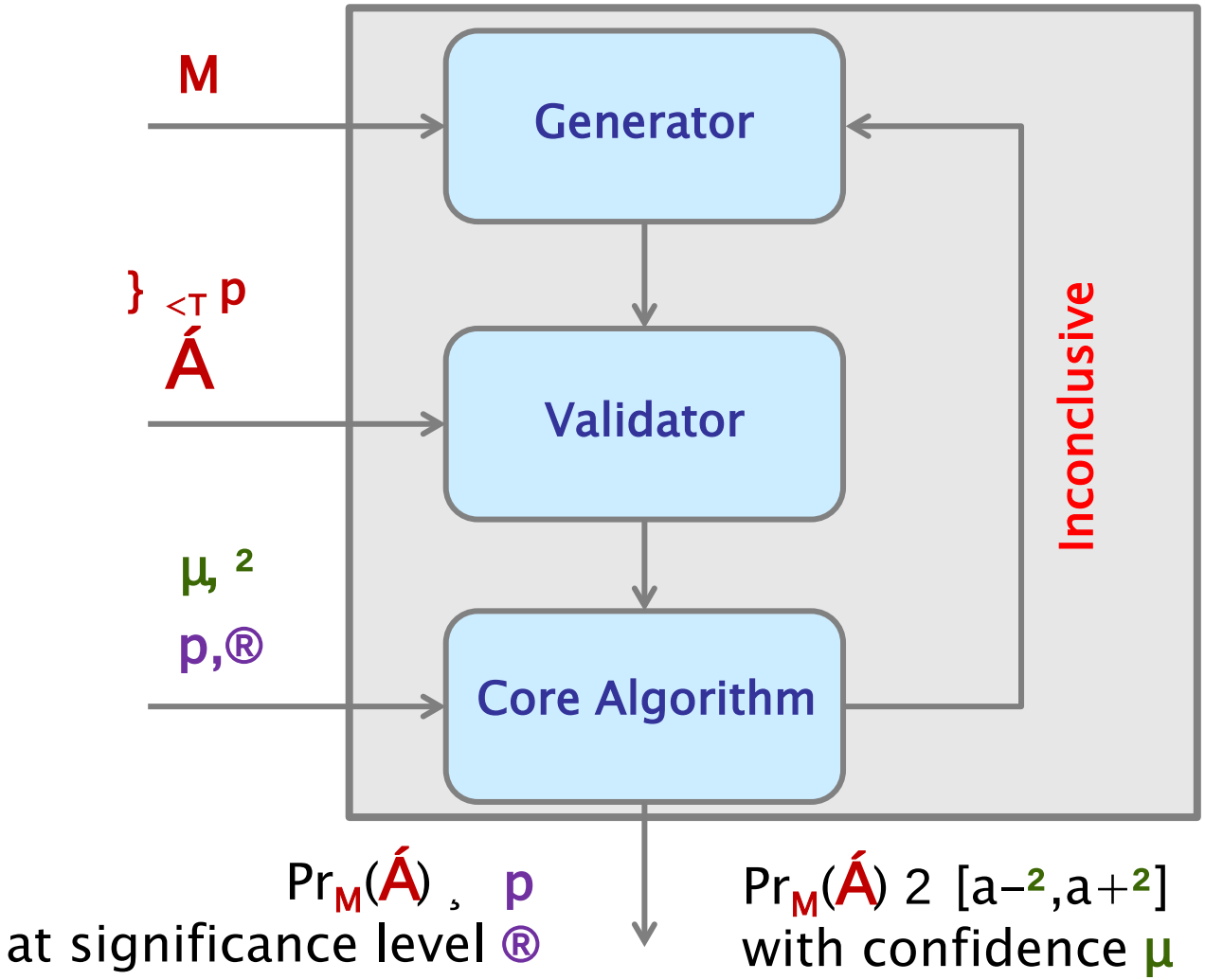
$$\text{Á} = \text{ok}\, U^{\tau}_{\leq 9}(\text{problem} \wedge (\neg\text{ok}\ U^{\tau}_{\leq 10}\ \text{ok}) \wedge (\neg\text{ok}\ U^{c}_{\leq 40}\ \text{ok}))$$



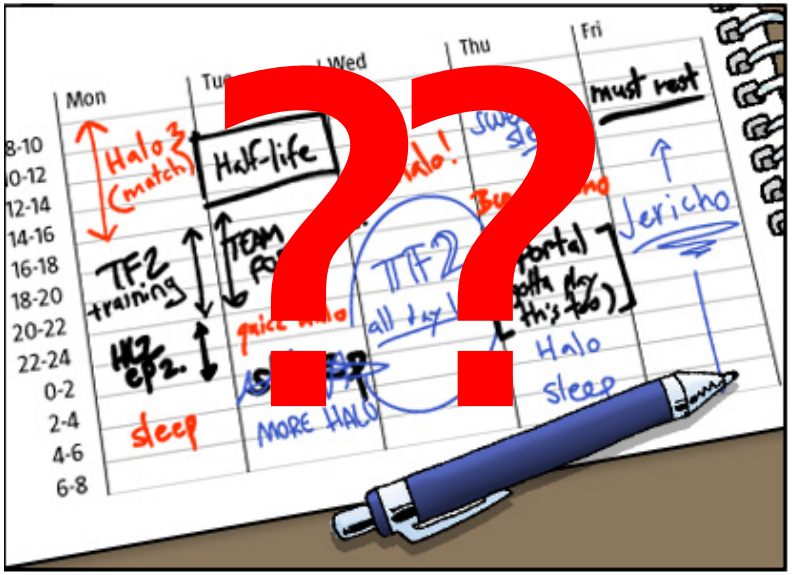
$$\Pr_M(\text{Á}) = ??$$

Statistical Model Checking

[FORMATS11, LPAR12, RV12]



Schedulability & Performance Analysis



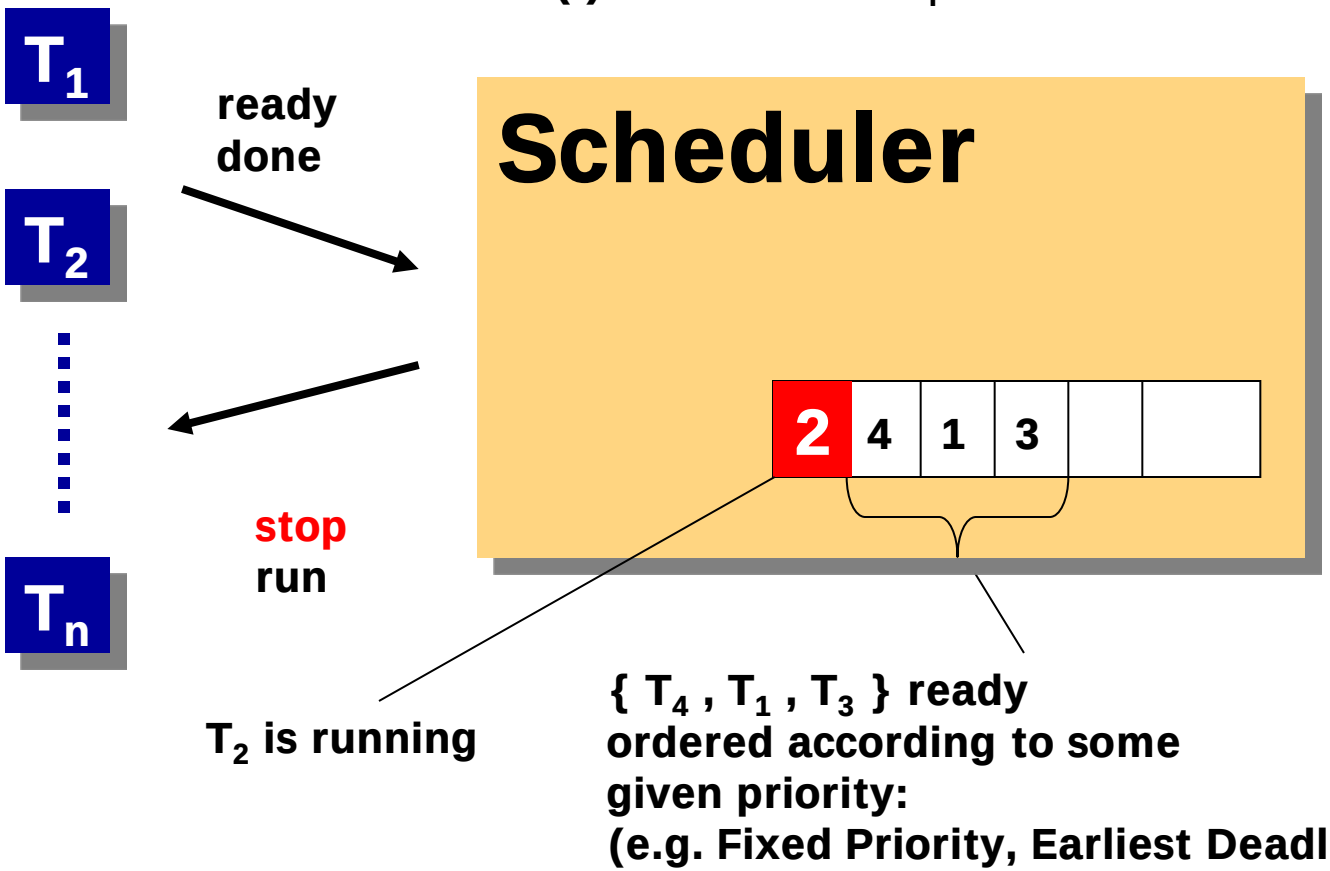
Task Scheduling

utilization of CPU

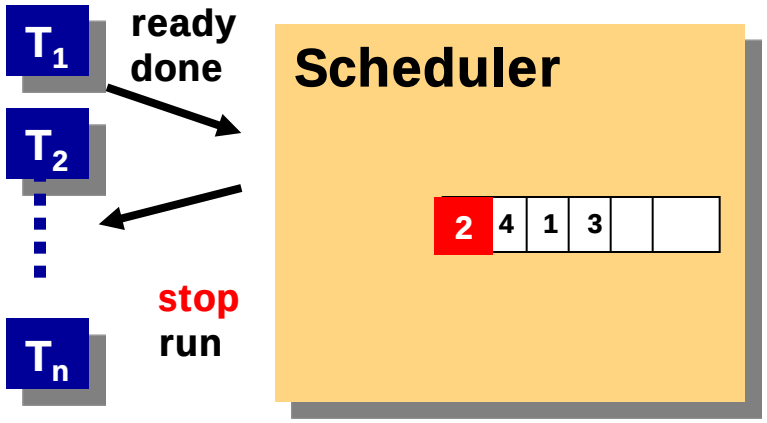
P(i), UNI[E(i), L(i)], .. : period or earliest/latest arrival or .. for T_i

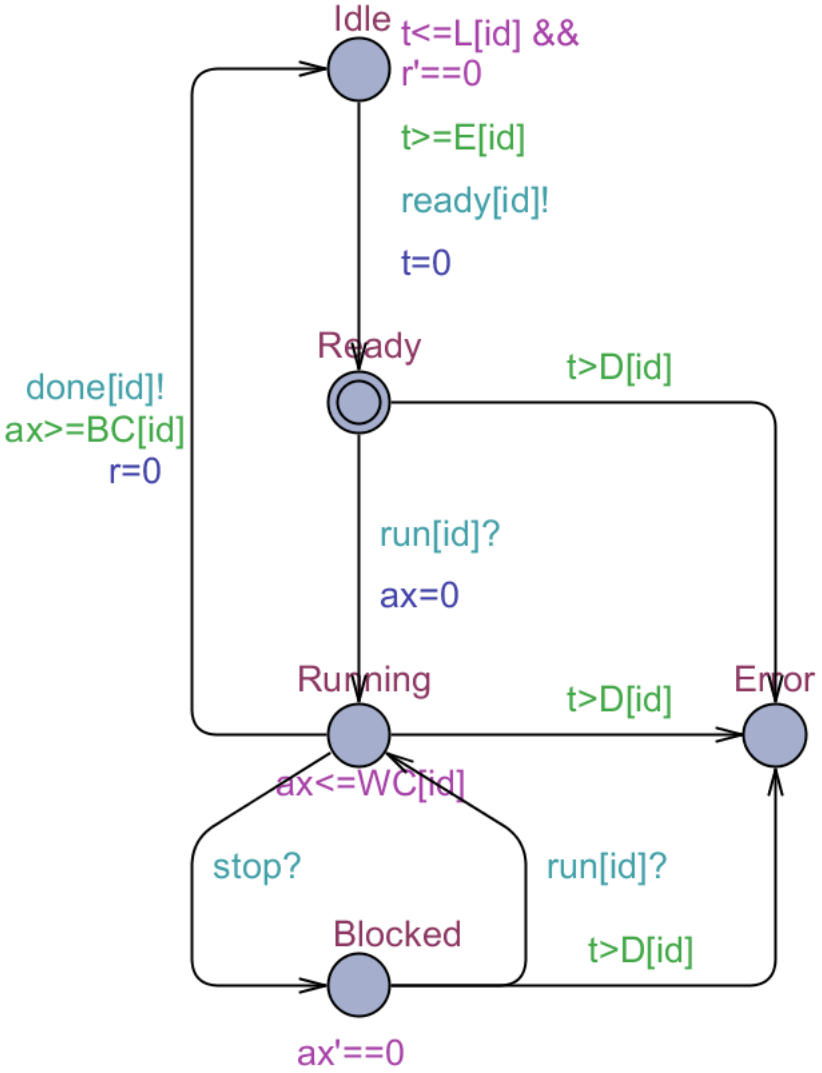
C(i), UNI[BC(i),WC(i)] : execution time for T_i

D(i): deadline for T_i

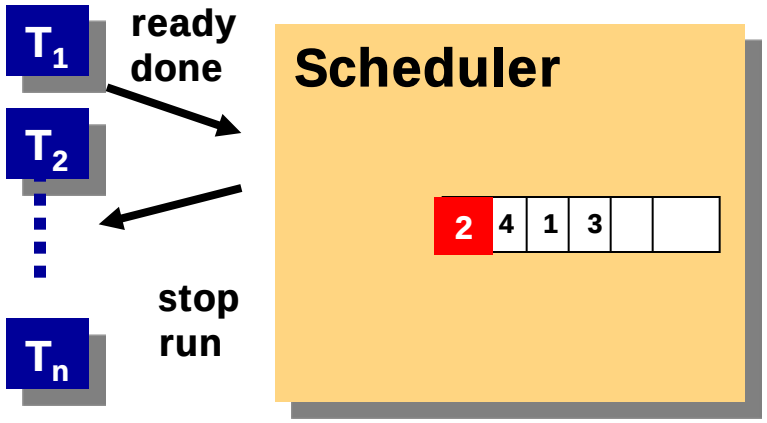


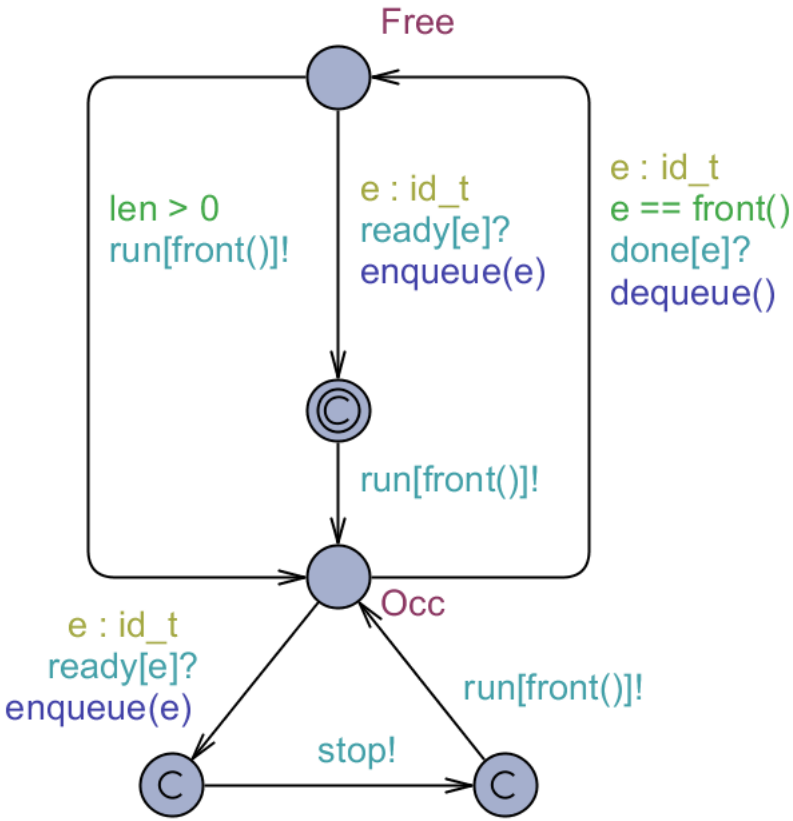
Modeling Task



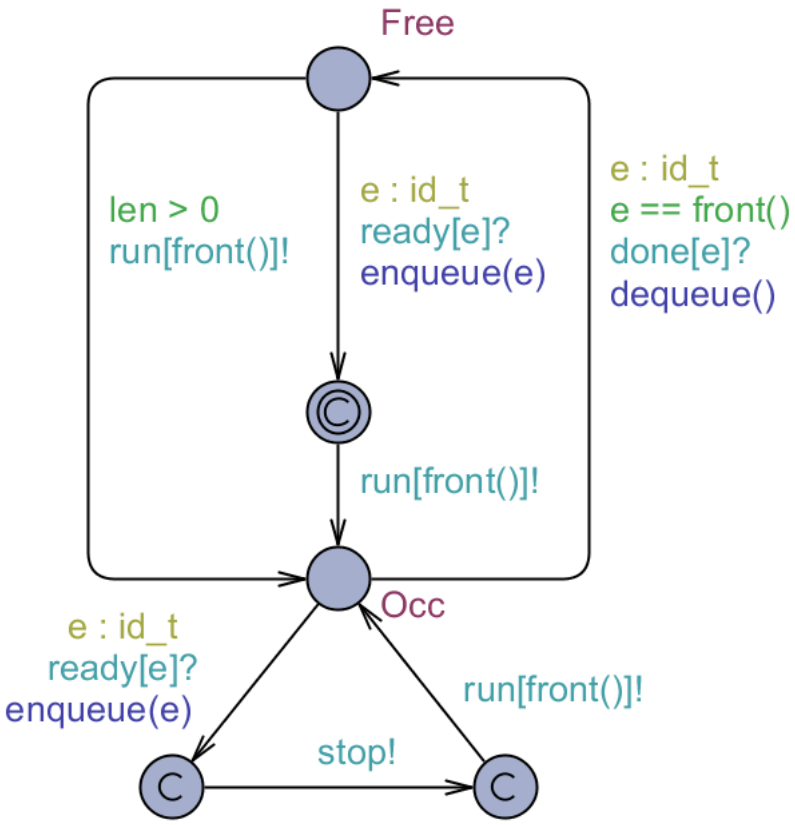


Modeling Scheduler





Modeling Queue



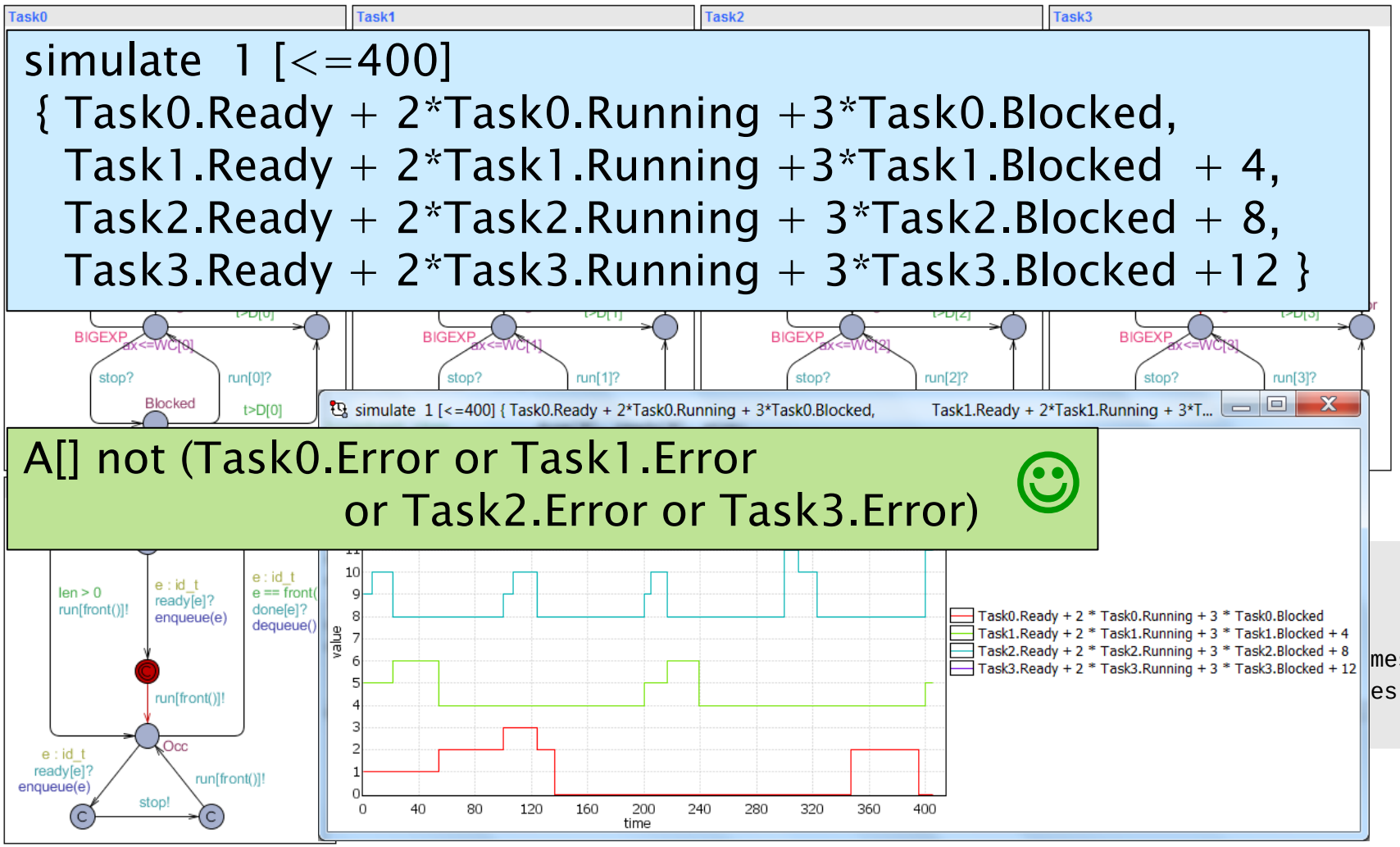

```
// Put an element at the end of the queue
void enqueue(id_t element)
{
int tmp=0;
list[len++] = element;
if (len>0)
{
        int i=len-1;
        while (i>1 && P[list[i]]>P[list[i-1]])
        {
                tmp = list[i-1];
                list[i-1] = list[i];
                list[i] = tmp;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

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

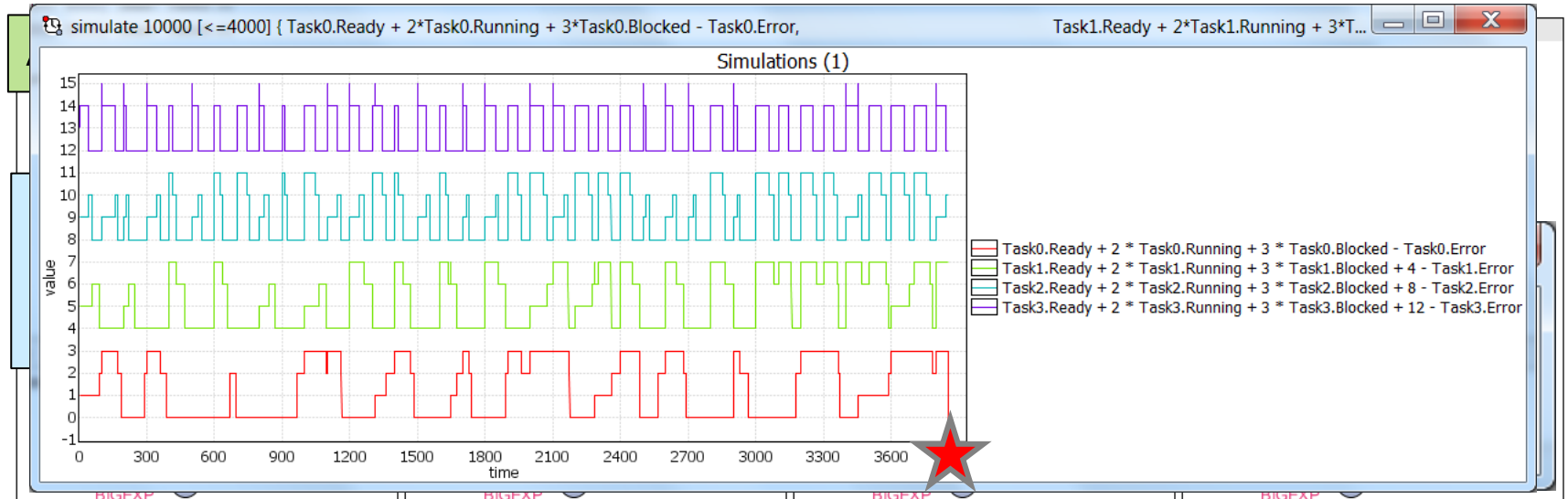
Schedulability Analysis

simulate 1 [<=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,
Task3.Ready + 2*Task3.Running + 3*Task3.Blocked +12 }

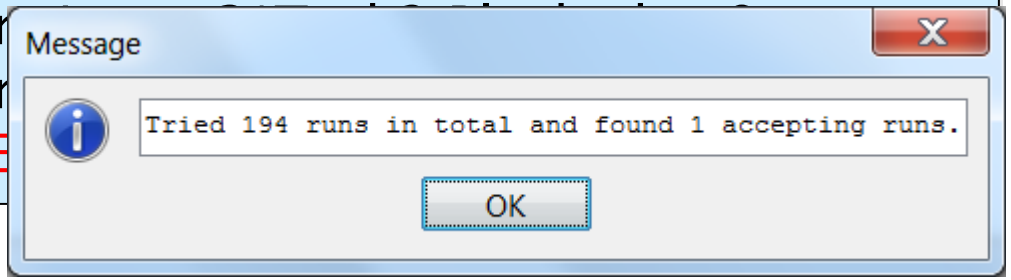
A[] not (Task0.Error or Task1.Error
or Task2.Error or Task3.Error)



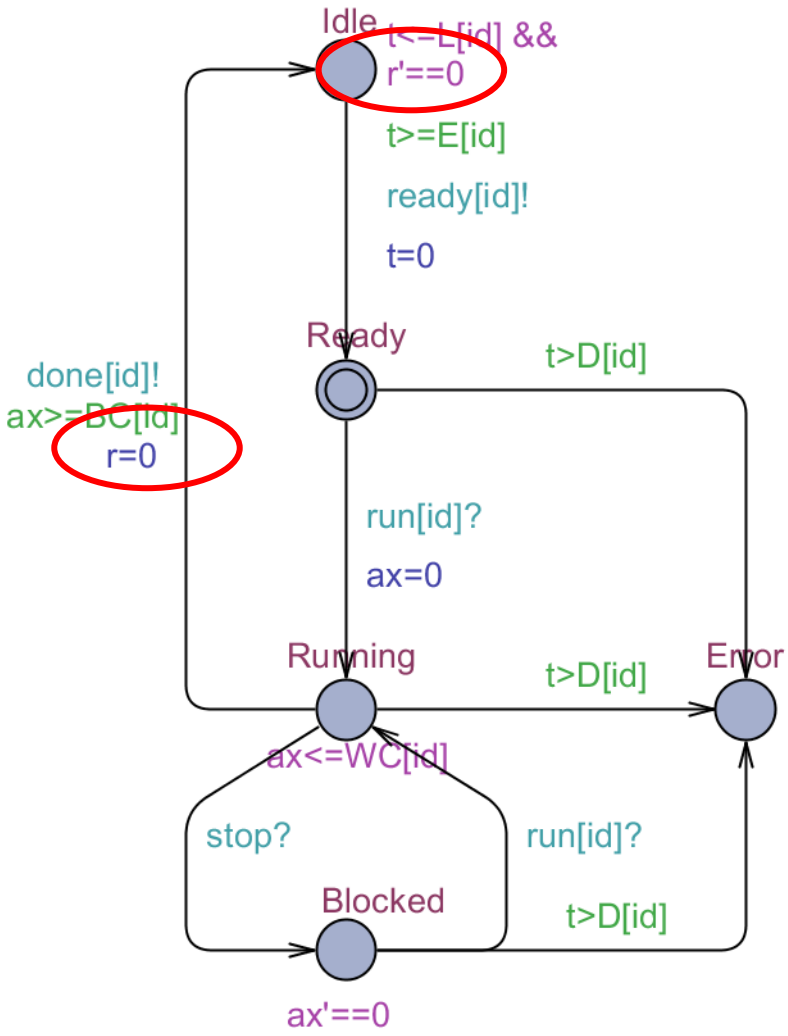
Schedulability Analysis



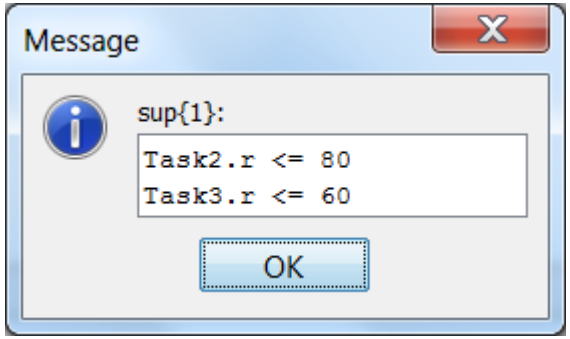
simulate 10000 [<=400]
{ Task0.Ready + 2*Task0.Running +3*Task0.Blocked,
Task1.Ready + 2*Task1.Running +3*Task1.Blocked + 4,
Task2.Ready + 2*Task2.Ru
Task3.Ready + 2*Task3.Ru
: 1 : (Task0.Error or Task1.E



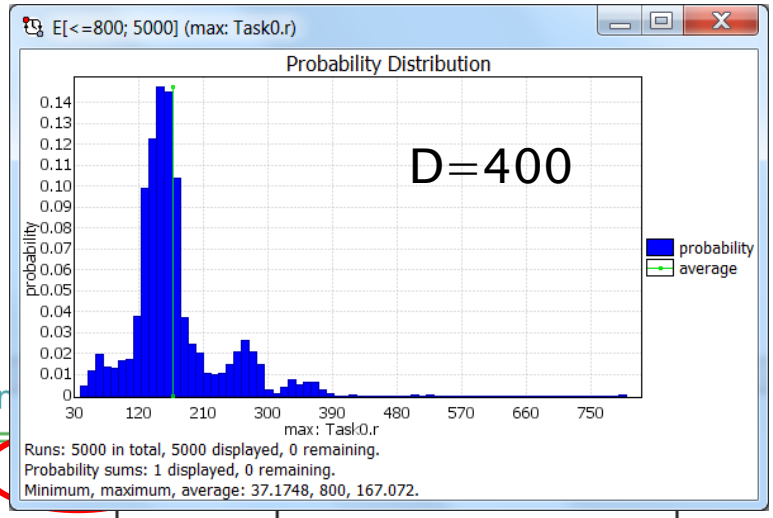
Performance Analysis

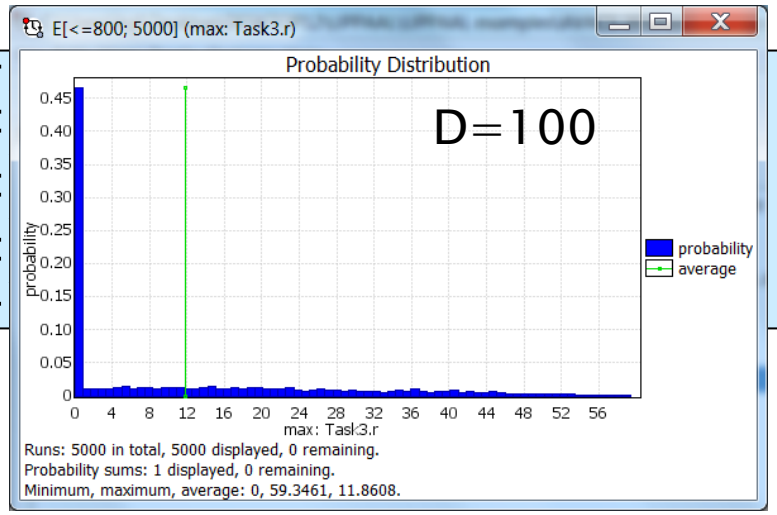


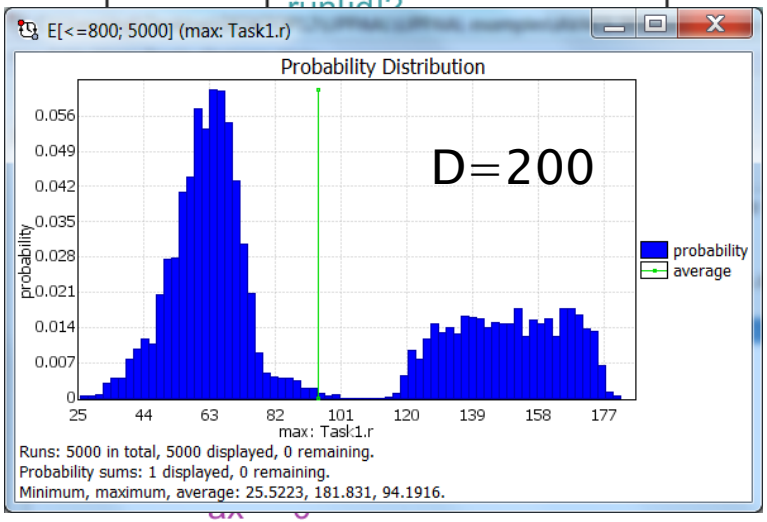
sup : Task2.r, Task3.r

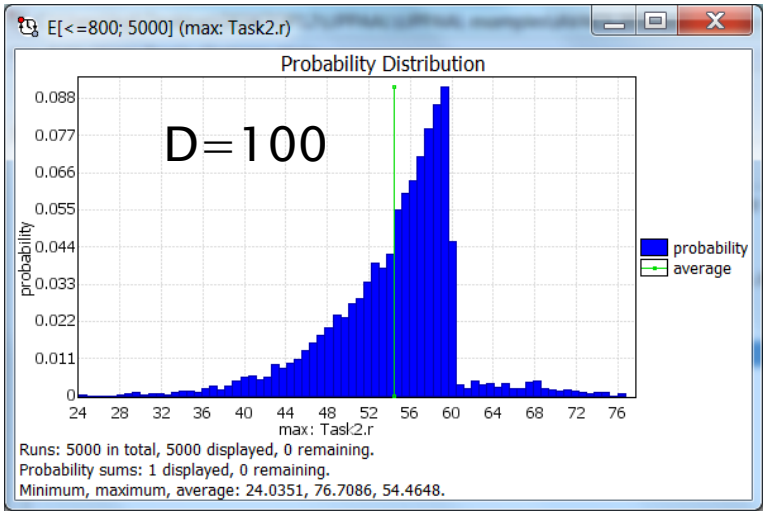


Performance Analysis









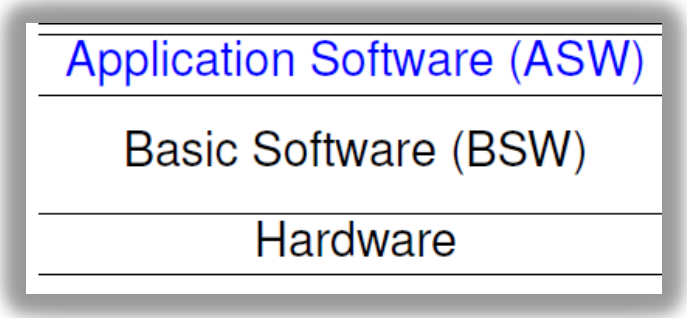
Herschel-Planck Scientific Mission at ESA

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

Herschel & Planck Satelites

TERMA

- 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 act

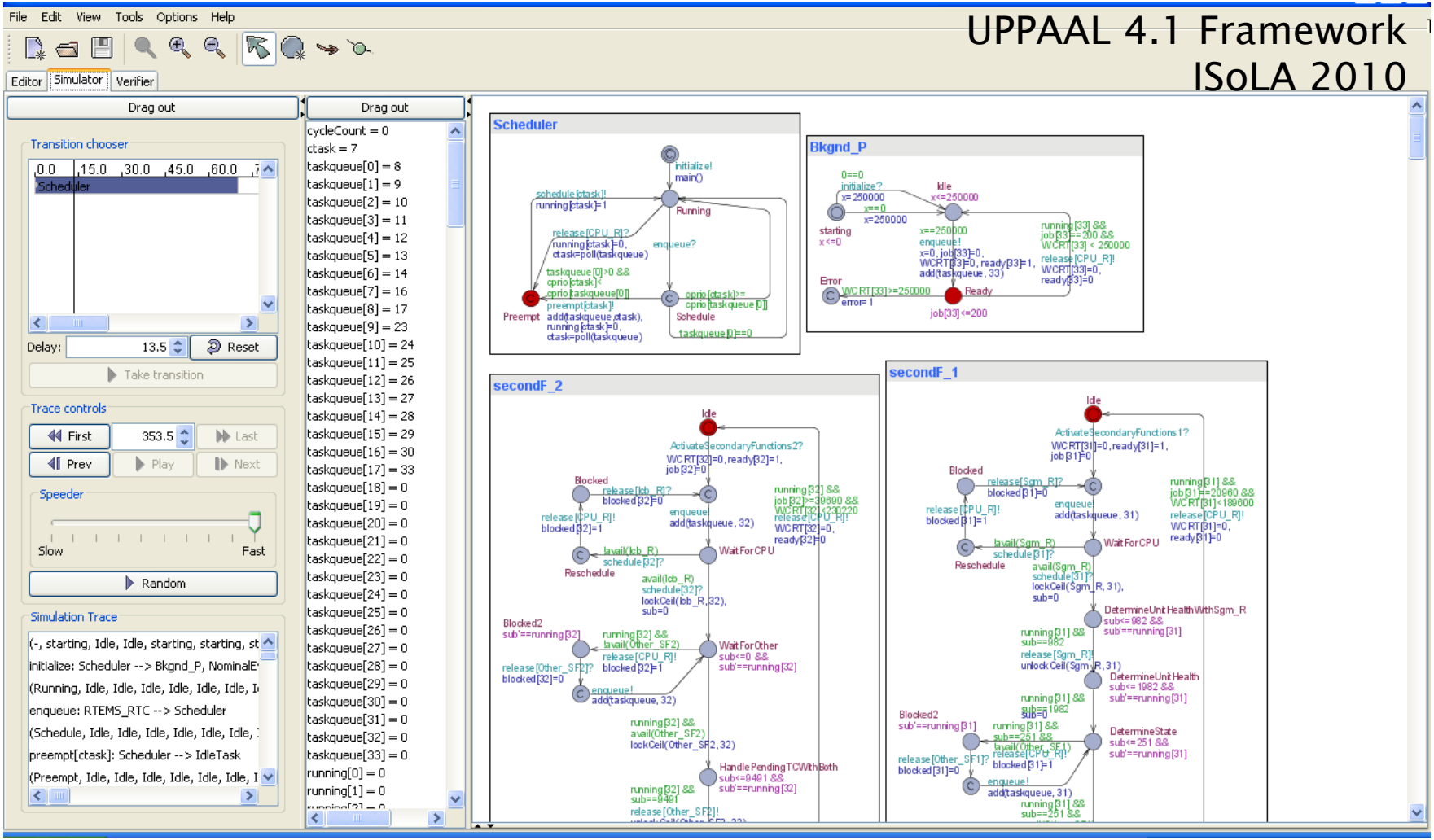


Requirements:

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

Modeling in UPPAAL

TERMA



UPPAAL 4.1 Framework
ISoLA 2010

Gantt Chart 1. cycle

TERMA

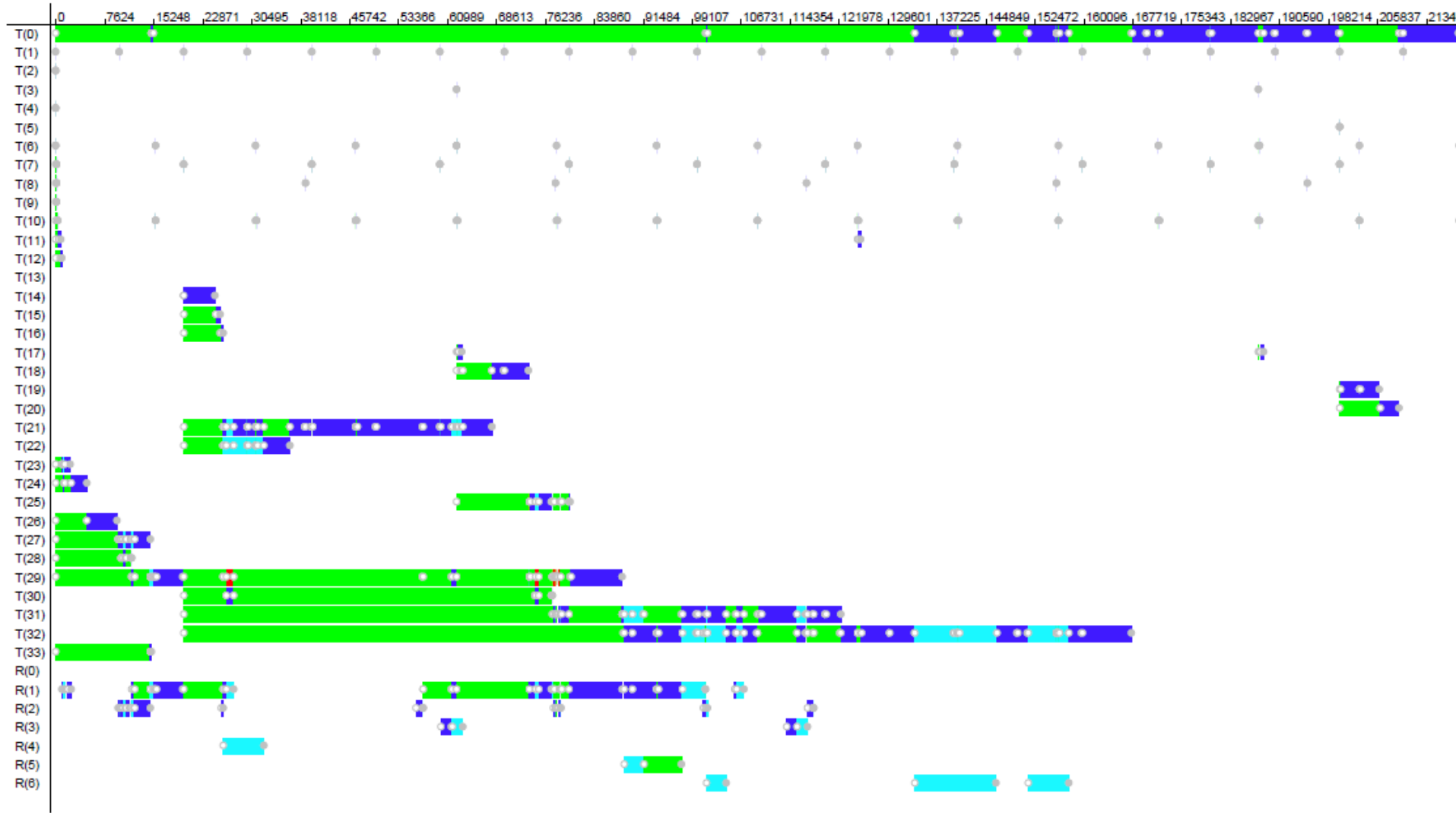


Fig. 11. Gantt chart of a schedule from the first cycle: green means ready, blue means running, cyan means suspended, red means blocked. R stand for resources: CPU_R=0, Icb_R=1, Sgm_R=2, PmReq_R=3, Other_RCS=4, Other_SF1=5, Other_SF2=6.

Blocking & WCRT

TERMA

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

Effort and Utilization

TERMA

cycle limit	Uppaal resources CPU, s	Mem, KB	States, #	Herschel CPU utilization Idle, μs	Used, μs	Global, μs	Sum, μs	Used, %
1	465.2	60288	173456	91225	160015	250000	251240	0.640060
2	470.1	59536	174234	182380	318790	500000	501170	0.637580
3	461.0	58656	175228	273535	477705	750000	751240	0.636940
4	474.5	58792	176266	363590	636480	1000000	1000070	0.636480
6	474.6	58796	178432	545900	955270	1500000	1501170	0.636847
8	912.3	58856	352365	727110	1272960	2000000	2000070	0.636480
13	507.7	58796	186091	1181855	2069385	3250000	3251240	0.636734
16	1759.0	58728	704551	1454220	2545850	4000000	4000070	0.636463
26	541.9	58112	200364	2363640	4137530	6500000	6501170	0.636543
32	3484.0	75520	1408943	2908370	5091700	8000000	8000070	0.636463
39	583.5	74568	214657	3545425	6205745	9750000	9751170	0.636487
64	7030.0	91776	2817704	5816740	10183330	16000000	16000070	0.636458
78	652.2	74768	257582	7089680	12411420	19500000	19501100	0.636483
128	14149.4	141448	5635227	11633480	20366590	32000000	32000070	0.636456
156	789.4	91204	343402	14178260	24821740	39000000	39000000	0.636455
256	23219.4	224440	11270279	23266890	40733180	64000000	64000070	0.636456
312	1824.6	124892	686788	28356520	49643480	78000000	78000000	0.636455
512	49202.2	390428	22540388	46533780	81466290	128000000	128000070	0.636455
624	3734.7	207728	1373560	56713040	99286960	156000000	156000000	0.636455

Marius Micusionis

TERMA Case Conclusion

- Schedulability analysis using UPPAAL:
  - Reusable and customizable task templates.
  - *Blocking* times and WCRTs can be derived from the model.
  - WCRTs of all tasks are more optimistic than in RTA.
  - There are very few blocking times and they are short.
  - PrimaryF meets deadline (59.6ms) with WCRT=54.1ms (65.5ms in RTA).
  - Herschel event mode is **schedulable**.
- UPPAAL verification for schedulability:
  - can be **scaled** using sweep-line method,
  - takes up to **2min** to verify schedulability of 32 task system,
  - takes up to **8min** to find all WCRTs and CPU utilization.
- In addition, it is possible to:
  - simulate the system model and examine details,
  - render a Gantt chart, validate and inspect visually.

TERMA Case Follow-Up

[f*WCET, WCET]

limit	f=100%			f=95%		
	states	mem	time	states	mem	t
1	1300	51.2	1.47	485077		
2	2522	53.7	2.45	806914		
4	4981	54.5	4.62	1499700		.8
8						
16						
∞	1					

	f=90%			f=86%		
	states	mem	time, s	states	mem	time
	1481162	124.1	4962.8	3348246	186.9	23986.5
	2414679	139.7	7755	5253778	198.7	33299.2
	4421630	138.3	13720	9231399	274.6	51176.6
	9093562	156.5	3112.3	18240030	364.6	102932.4
	17798572	176.0	6014.5	35432003	520.4	158816.7
	181869652	1682.2	530604.9	error may be reachable		

1 Day

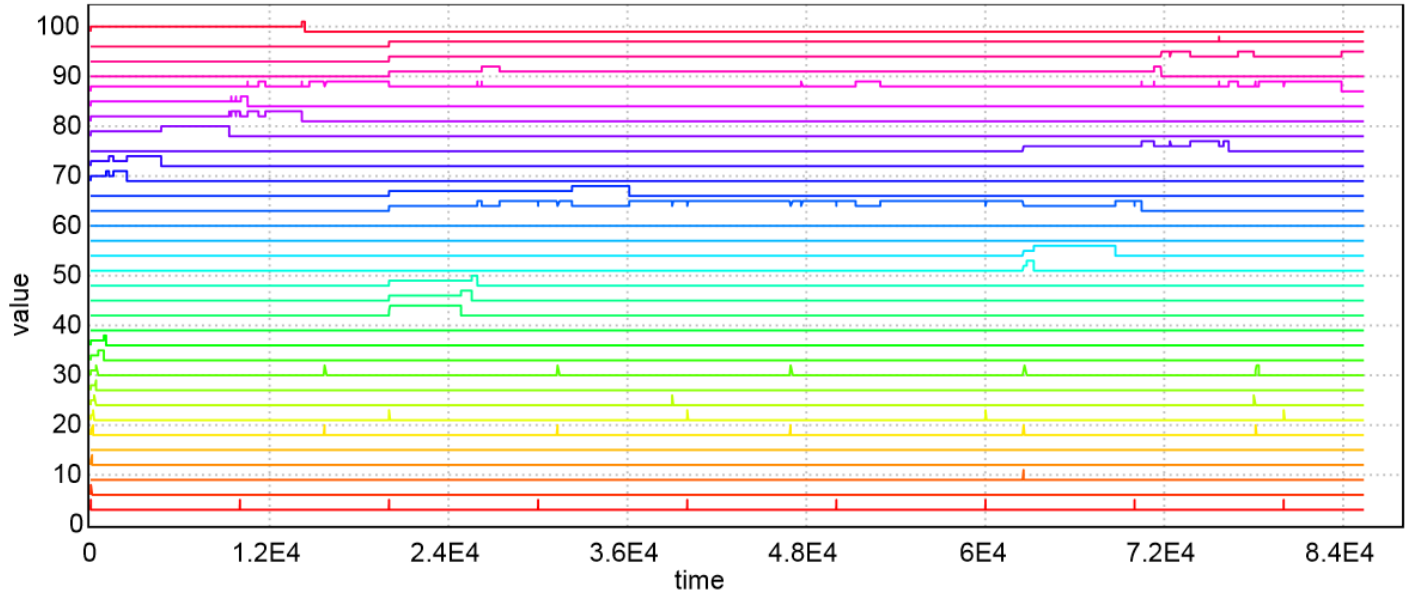
6 Days

TERMA Case – Statistical MC

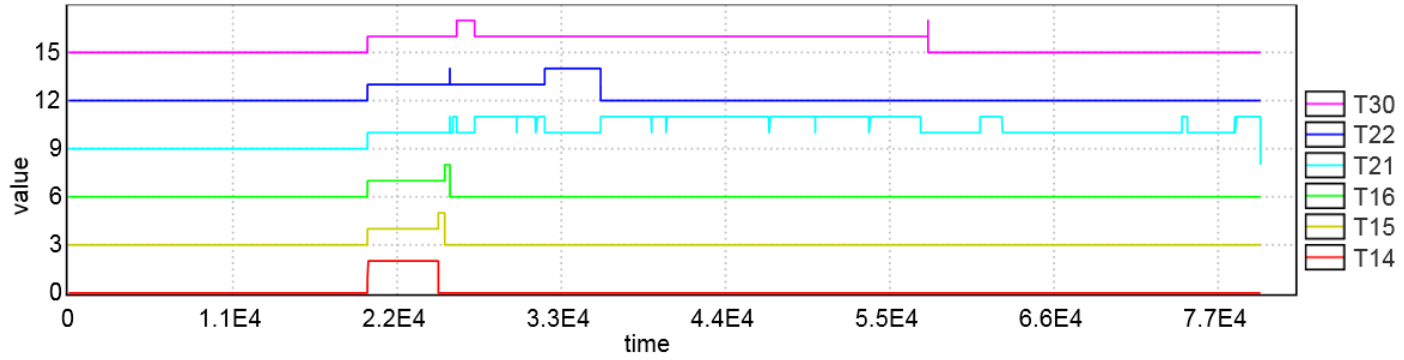
Limit cycles	f %	α	ε	Total traces, #	Error traces #	Error traces Probability	Earliest Error cycle	Earliest Error offset	Verification time
1	0	0.0100	0.005	105967	1928	0.018194	0	79600.0	1:58:06
1	50	0.0100	0.005	105967	753	0.007106	0	79600.0	2:00:52
1	60	0.0100	0.005	105967	13	0.000123	0	79778.3	2:01:18
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

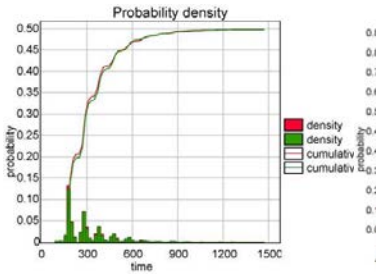
Herschel simulation run with $f = 90\%$:



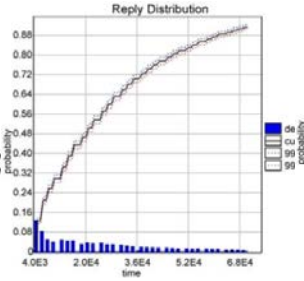
Herschel deadline violation with $f = 50\%$:

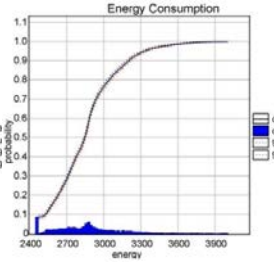


Other Case Studies

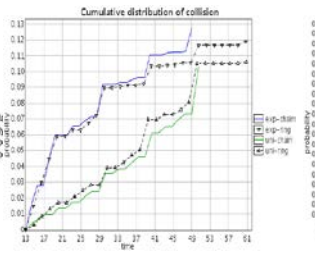


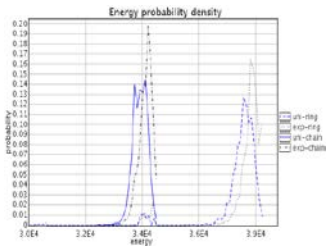
FIREWIRE



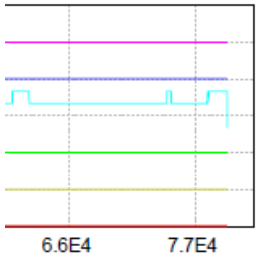


BLUETOOTH

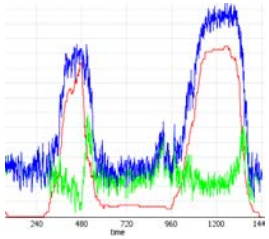




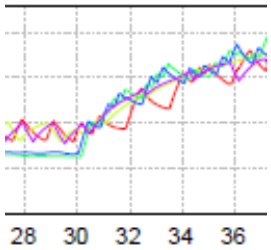
10 node LMAC



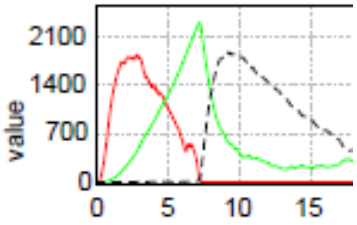
Schedulability Analysis for Mix Cr Sys



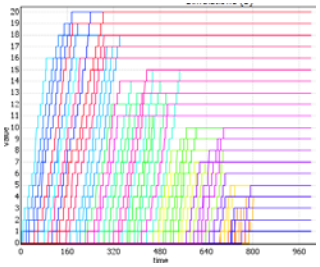
Smart Grid Demand / Response



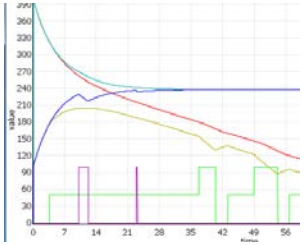
Energy Aware Buildings



Genetic Oscilator (HBS)



Passenger Seating in Aircraft



Battery Scheduling
(SENSATION)
Erik Wogensen

www.uppaal.{org,com}