Model Checking, Performance Evaluation, Synthesis and Optimization of Cyber–Physical Systems

Kim G. Larsen
Aalborg University, DENMARK
From Computer Science to Embedded Systems
From Computer Science to Cyber Physical Systems

New Foundation

Discrete Models
(Boolean correctness)

→

Quantitative Models
(time, resources, probabilistic, stochastic, continuous,..)
(Quantitative correctness)

Discrete
Real Time

Resources

Stochasticity

Hybrid
Topics & Task Overview

- Specification and Modeling
- Validation and Analysis
- Compositionality versus Global Features
- Cross-Level Preservation
- Mini Cases
- Prototype Tools
Models

Discrete

Probabilistic

Hybrid

ALW( OFF \Rightarrow EVEN ON)

ALW( T<3 \Rightarrow EVEN_{t<20, p>0.90} T>20)
UPPAAL Tool Suit

- **Ve**
- **CLASSIC**
- **CORA**
- **SMC**
- **ECDAR**
- **TRON**
- **TIGA**
- **STRATEGO**

Optimization

Performance Analysis

Component Testing

Synthesis

Optimal Synthesis

EMSIG Autumn School 2015

Kim Larsen [6]
Contributors

@UPPsala
- Wang Yi
- Paul Pettersson
- John Håkansson
- Anders Hessel
- Pavel Krcal
- Leonid Mokrushin
- Shi Xiaochun

@AALborg
- Kim G Larsen
- Alexandre David
- Marius Mikucionis
- Gerd Behrman
- Arne Skou
- Brian Nielsen
- Jacob I. Rasmussen
- Thomas Chatain

@Elsewhere
- Emmanuel Fleury, Didier Lime, Johan Bengtsson, Fredrik Larsson, Kåre J Kristoffersen, Tobias Amnell, Thomas Hune, Oliver Möller, Elena Fersman, Carsten Weise, David Griffioen, Ansgar Fehnker, Frits Vandraager, Theo Ruys, Pedro D’Argenio, J-P Katoen, Jan Tretmans, Judi Romijn, Ed Brinksma, Martijn Hendriks, Klaus Havelund, Franck Cassez, Magnus Lindahl, Francois Laroussinie, Patricia Bouyer, Augusto Burgueno, H. Bowmann, D. Latella, M. Massink, G. Faconti, Kristina Lundqvist, Lars Asplund, Justin Pearson...
Overview

- **Timed Automata** / UPPAAL
  - Verification

- **Stochastic Priced Timed Automata** / UPPAAL SMC
  - Performance Evaluation
  - SMC in a Nutshell
  - Stochastic Hybrid Automata

- **Timed Games** / UPPAAL TIGA
  - Controller Synthesis

- **Stochastic Priced Timed Games** / UPPAAL STRATEGO
  - Optimal & Safe Syntheses

- Conclusion
Timed Automata
A Dumb Light Controller

Off -> press? -> Light

press? -> Light

Light -> press? -> Bright

Bright -> press? -> Bright

Off -> press? -> Off
Timed Automata

[Alur & Dill’89]

x: real-valued clock

ADD a clock x
A Timed Automata (Semantics)

States:
( location, x=v) where v∈R

Transitions:

( Off, x=0 )
delay 4.32  → ( Off, x=4.32 )
press?  → ( Light, x=0 )
delay 2.51  → ( Light, x=2.51 )
press?  → ( Bright, x=2.51 )
Intelligent Light Controller
Intelligent Light Controller

Transitions:

- (Off, x=0)
- delay 4.32 → (Off, x=4.32)
- press? → (Light, x=0)
- delay 4.51 → (Light, x=4.51)
- press? → (Light, x=0)
- delay 100 → (Light, x=100)
- τ → (Off, x=0)

Note:

(Off, x=0) delay 103 → (Light, x=0)

Invariants ensure progress
Train Crossing

Safe → Approaching → Crossing → Safe

River

Bridge

tracks

0 20 3–5

Time

EMSIG Autumn School 2015 [15]
Train Crossing

Safe → Approaching → Crossing → Safe
Safe → Approaching → Crossing → Safe

Stop the train while it still stoppable!
Train Crossing

Safe → Approaching → Crossing → Safe
Safe → Approaching → Crossing → Safe

Stopped → Restarted

River

Bridge

tracks

0 10 20

3 – 5

7 – 15

EMSIG Autumn School 2015
Timed Automata [Train] = Finite State Control + Real Valued Clocks

**Invariants**
- Safe
- Cross
  - $x \leq 5$
- $x = 0$
- $x \geq 3$
- $\text{leave[id]}$

** Guards**
- $x \geq 10$
- $x = 0$
- $x \geq 7$
- $x = 0$
- Start
  - $x \leq 15$
- Appr
  - $x \leq 20$
- $x = 0$
- $\text{appr[id]}$
- $x = 0$
- $\text{Appr}$

**Synchronizations**
- $\text{id}$
- $\text{go[id]}$
- $x = 0$
- $\text{stop[id]}$
- $x = 0$
- $\text{Stop}$

**Resets**
- $\text{Safe}$
- $\text{Cross}$

**Semantics**
- $(\text{Appr}, x=0)$ $\rightarrow$ $5.2$
- $(\text{Appr}, x=5.2)$ $\rightarrow$ `stop?`
- $(\text{Stop}, x=5.2)$

EMSIG Autumn School 2015

Kim Larsen [19]
DEMO
Logical Specifications

- **Validation Properties**
  - Possibly: $E \leftrightarrow P$

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

- **Liveness Properties**
  - Eventually: $A \leftrightarrow P$
  - Leadsto: $P \rightarrow Q$

- **Bounded Liveness**
  - Leads to within: $P \rightarrow_{\leq t} Q$

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

Only references to integer variables, constants, clocks, and locations are allowed (and arrays of these).
THE "secret" of UPPAAL

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]
Zones – From Finite to Efficiency

A zone $Z$:

\[
1 \leq x \leq 2 \quad \land \\
0 \leq y \leq 2 \quad \land \\
x - y \geq 0
\]

**Theorem**

The number of regions is $n! \cdot 2^n \cdot \prod_{x \in C} (2c_x + 2)$. 
Zones – Operations

\[(n, 2 \leq x \leq 4 \land 1 \leq y \leq 3 \land y-x \leq 0 )\]

\[(n, 2 \leq x \land 1 \leq y \land -3 \leq y-x \leq 0 )\]

\[(n, 2 \leq x \land 1 \leq y \land x \leq 0 )\]

\[(n, x=0 \land 1 \leq y \leq 3 )\]

\[(n, 2 \leq x \leq 4 \land 1 \leq y )\]

\[(n, 2 \leq x \land 1 \leq y \land y-x \leq 0 )\]

Delay

Delay (stopwatch)

Reset

Extrapolation

Convex Hull
Datastructures for Zones

- Difference Bounded Matrices (DBMs)
- Minimal Constraint Form [RTSS97]
- Clock Difference Diagrams [CAV99]
Stochastic Timed Automata
Stochastic Semantics of TA

Exponential Distribution

Safe

x = 0

appr[id]!

x = 0

Appr

x <= 20

x = 10

Stop

x = 0

Stop

x = 10

Start

x <= 15

x = 0

Cross

x <= 5

x = 7

x >= 10

x = 0

x >= 3

leave[id]!

Uniform Distribution

Composition = Repeated races between components for outputting
Stochastic Semantics of Timed Automata

Delay Density Function
\[ \mu_s : \mathbb{R} \rightarrow \mathbb{R} \]
Output Probability Function
\[ \gamma_s : \Sigma_o \rightarrow [0, 1] \]

- \( \mu_s \) uniform on \([d_{\text{min}}, d_{\text{max}}]\)
- \( \gamma_s \) uniform over enabled outputs
Composition of STA

Composition = Race between components for outputting

\[ \Pr[\text{time} \leq 2](\nleq T \cdot T3) \]

\[ \int_{t_1=0}^{1} 1 \cdot \left( \int_{t_2=t_1}^{2} \frac{1}{2} dt_2 \right) dt_1 = \frac{3}{4} \]

\[ \Pr[\text{time} \leq T](\nleq T \cdot T3) \]
Stochastic Semantics of TA

Assumptions:
Component TAs are:
• Input enabled
• Deterministic
• Disjoint set of output actions

\[ (s, a_1, a_2, \ldots, a_n) : \] the set of maximal runs from \( s \) with a prefix
\[ t_1 a_1 t_2 a_2 \ldots t_n a_k \]
for some \( t_1, \ldots, t_n \in \mathbb{R} \).

\[ \mu_s : \mathbb{R} \rightarrow \mathbb{R} \]
• Delay Density Function
\[ \gamma_s : \Sigma_o \rightarrow [0,1] \]
• Output Probability Function

\[
\mathbb{P}_A(\pi(s, a_1a_2\ldots a_n)) = \int_{t \geq 0} \mu_{s_c}(t) \cdot (\prod_{j \neq c} \int_{\tau > t} \mu_{s_j}(\tau) \, d\tau) \cdot \gamma_{s_c} (a_1) \cdot \mathbb{P}_A(\pi(s^t a_1, a_2 \ldots a_n)) \, dt
\]
where \( c = c(a_1) \), and as base case we take \( P_A(\pi(s), \varepsilon) = 1 \).
Beyond Uniform / Exponential Dist.

Includes all Phase-Type Distributions.
Can encode any distribution with arbitrary precision.
Statistical Model Checking

M

Generate random run π

Validate π ⊨ φ?

Core Statistical Algorithm

Pr_M(φ) ≥ p

at significance level α

Pr_M(φ) ∈ [a−ε,a+ε]

with confidence θ

Inconclusive

Hypothesis testing

Confidence Interval

EMSIG Autumn School 2015
Kim Larsen [32]
Queries in UPPAAL Syntax

- **Evaluation**
  \[ \text{Pr}[^{<=100}] (<> \text{expr}) \]

- **Hypothesis testing**
  \[ \text{Pr}[^{<=100}] (<> \text{expr}) \geq 0.1 \]
  \[ c^{<=100} \#^{<=50} \square \text{expr} \leq 0.5 \]

- **Comparison**
  \[ \text{Pr}[^{<=20}] (<> \text{e1}) \geq \text{Pr}[^{<=10}] (<> \text{e2}) \]

- **Expected value**
  \[ \text{E}[^{<=10;1000}] (\text{min: expr}) \]
  Explicit number of runs. Min or max.

- **Simulations**
  \[ \text{simulate 10} [^{<=100}] \{\text{expr1,expr2}\} \]
Schedulability & Performance Analysis
Task Scheduling

Scheduler

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

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_1
T_2
T_n

T_2 is running
ready
run
stop
done
Modeling Task

Scheduler

T₁ ready done
T₂ stop run
Tₙ

EMSIG Autumn School 2015
Kim Larsen [37]
Modeling Scheduler

Scheduler

T₁ ready done

T₂ stop run

Tₙ

Scheduler

- T₁ ready done
- T₂ stop run
- Tₙ

Diagram:

- Free
- Occ
- C

Transition:

- len > 0 run[front()]!
- e : id_t ready[e]?
- enqueue(e)
- done[e]?
- dequeue()

States:

- C
- Free
- Occ
- stop!
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 && list[i-1]>list[i])
    {
      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*Task0Blocked, 
  Task1.Ready + 2*Task1.Running + 3*Task1Blocked + 4, 
  Task2.Ready + 2*Task2.Running + 3*Task2Blocked + 8, 
  Task3.Ready + 2*Task3.Running + 3*Task3Blocked + 12 } 

A[] not (Task0.Error or Task1.Error or Task2.Error or Task3.Error) 😊
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,  
: 1 : (Task0.Error or Task1.Error or Task2.Error or Task3.Error)
Performance Analysis

sup : Task2.r, Task3.r

Image of a Petri net with transitions and places labeled with conditions and actions.
Performance Analysis

- $D=400$
- $D=200$
- $D=100$

Images show probability distribution graphs for different $D$ values.
Herschel–Planck Scientific Mission at ESA

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

- **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 communication buses, sensors and actuators.

**Requirements:**
Software tasks should be schedulable.
CPU utilization should not exceed 50% load.
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.
<table>
<thead>
<tr>
<th>ID</th>
<th>Task</th>
<th>Specification</th>
<th>Blocking times</th>
<th>WCRT</th>
<th>Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Period</td>
<td>WCET</td>
<td>Deadline</td>
<td>Term</td>
</tr>
<tr>
<td>1</td>
<td>RTEMS_RTC</td>
<td>10.000</td>
<td>0.013</td>
<td>1.000</td>
<td>0.035</td>
</tr>
<tr>
<td>2</td>
<td>AswSync_syncPulseIsr</td>
<td>250.000</td>
<td>0.070</td>
<td>1.000</td>
<td>0.035</td>
</tr>
<tr>
<td>3</td>
<td>Hk_SamplerIsr</td>
<td>125.000</td>
<td>0.070</td>
<td>1.000</td>
<td>0.035</td>
</tr>
<tr>
<td>4</td>
<td>SwCyc_CycStartIsr</td>
<td>250.000</td>
<td>0.200</td>
<td>1.000</td>
<td>0.035</td>
</tr>
<tr>
<td>5</td>
<td>SwCyc_CycEndIsr</td>
<td>250.000</td>
<td>0.100</td>
<td>1.000</td>
<td>0.035</td>
</tr>
<tr>
<td>6</td>
<td>Rt1553_Isr</td>
<td>15.625</td>
<td>0.070</td>
<td>1.000</td>
<td>0.035</td>
</tr>
<tr>
<td>7</td>
<td>Bc1553_Isr</td>
<td>20.000</td>
<td>0.070</td>
<td>1.000</td>
<td>0.035</td>
</tr>
<tr>
<td>8</td>
<td>Spw_Isr</td>
<td>39.000</td>
<td>0.070</td>
<td>2.000</td>
<td>0.035</td>
</tr>
<tr>
<td>9</td>
<td>Obdh_Isr</td>
<td>250.000</td>
<td>0.070</td>
<td>2.000</td>
<td>3.650</td>
</tr>
<tr>
<td>10</td>
<td>Rtsdb_P_1</td>
<td>15.625</td>
<td>0.150</td>
<td>15.625</td>
<td>3.650</td>
</tr>
<tr>
<td>11</td>
<td>Rtsdb_P_2</td>
<td>125.000</td>
<td>0.400</td>
<td>15.625</td>
<td>3.650</td>
</tr>
<tr>
<td>12</td>
<td>Rtsdb_P_3</td>
<td>250.000</td>
<td>0.170</td>
<td>15.625</td>
<td>3.650</td>
</tr>
<tr>
<td>13</td>
<td>FdirEvents</td>
<td>250.000</td>
<td>5.000</td>
<td>230.220</td>
<td>0.720</td>
</tr>
<tr>
<td>14</td>
<td>NominalEvents_1</td>
<td>250.000</td>
<td>0.720</td>
<td>230.220</td>
<td>0.720</td>
</tr>
<tr>
<td>15</td>
<td>MainCycle</td>
<td>250.000</td>
<td>0.400</td>
<td>230.220</td>
<td>0.720</td>
</tr>
<tr>
<td>16</td>
<td>HkSampler_P_2</td>
<td>125.000</td>
<td>0.500</td>
<td>62.500</td>
<td>3.650</td>
</tr>
<tr>
<td>17</td>
<td>HkSampler_P_1</td>
<td>250.000</td>
<td>6.000</td>
<td>62.500</td>
<td>3.650</td>
</tr>
<tr>
<td>18</td>
<td>Aeb_P</td>
<td>250.000</td>
<td>6.000</td>
<td>50.000</td>
<td>3.650</td>
</tr>
<tr>
<td>19</td>
<td>IoCyc_P</td>
<td>250.000</td>
<td>3.000</td>
<td>50.000</td>
<td>3.650</td>
</tr>
<tr>
<td>20</td>
<td>PrimaryF</td>
<td>250.000</td>
<td>34.050</td>
<td>59.600</td>
<td>5.770</td>
</tr>
<tr>
<td>21</td>
<td>RCSControlF</td>
<td>250.000</td>
<td>0.070</td>
<td>239.600</td>
<td>12.120</td>
</tr>
<tr>
<td>22</td>
<td>Obt_P</td>
<td>100.000</td>
<td>1.100</td>
<td>100.000</td>
<td>9.630</td>
</tr>
<tr>
<td>23</td>
<td>Hk_P</td>
<td>250.000</td>
<td>2.750</td>
<td>250.000</td>
<td>1.035</td>
</tr>
<tr>
<td>24</td>
<td>StsMon_P</td>
<td>250.000</td>
<td>3.300</td>
<td>125.000</td>
<td>16.070</td>
</tr>
<tr>
<td>25</td>
<td>TmGen_P</td>
<td>250.000</td>
<td>4.860</td>
<td>250.000</td>
<td>4.260</td>
</tr>
<tr>
<td>26</td>
<td>Sgm_P</td>
<td>250.000</td>
<td>4.020</td>
<td>250.000</td>
<td>1.040</td>
</tr>
<tr>
<td>27</td>
<td>TcRouter_P</td>
<td>250.000</td>
<td>0.500</td>
<td>250.000</td>
<td>1.035</td>
</tr>
<tr>
<td>29</td>
<td>NominalEvents_2</td>
<td>250.000</td>
<td>1.780</td>
<td>230.220</td>
<td>12.480</td>
</tr>
<tr>
<td>30</td>
<td>SecondaryF_1</td>
<td>250.000</td>
<td>20.960</td>
<td>189.600</td>
<td>27.650</td>
</tr>
<tr>
<td>31</td>
<td>SecondaryF_2</td>
<td>250.000</td>
<td>39.690</td>
<td>230.220</td>
<td>48.450</td>
</tr>
<tr>
<td>32</td>
<td>Bkgnd_P</td>
<td>250.000</td>
<td>0.200</td>
<td>250.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>
## Effort and Utilization

<table>
<thead>
<tr>
<th>Cycle limit</th>
<th>Uppaal resources</th>
<th>Herschel CPU utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CPU, s Mem, KB States, #</td>
<td>Idle, μs Used, μs Global, μs Sum, μs Used, %</td>
</tr>
<tr>
<td>1</td>
<td>465.2 60288 173456</td>
<td>91225 160015 250000 251240 0.640060</td>
</tr>
<tr>
<td>2</td>
<td>470.1 59536 174234</td>
<td>182380 318790 500000 501170 0.637580</td>
</tr>
<tr>
<td>3</td>
<td>461.0 58656 175228</td>
<td>273535 477705 750000 751240 0.636940</td>
</tr>
<tr>
<td>4</td>
<td>474.5 58792 176266</td>
<td>363590 636480 1000000 1000070 0.636480</td>
</tr>
<tr>
<td>6</td>
<td>474.6 58796 178432</td>
<td>545900 955270 1500000 1501170 0.636847</td>
</tr>
<tr>
<td>8</td>
<td>912.3 58856 352365</td>
<td>727110 1272960 2000000 2000070 0.636480</td>
</tr>
<tr>
<td>13</td>
<td>507.7 58796 186091</td>
<td>1181855 2069385 3250000 3251240 0.636734</td>
</tr>
<tr>
<td>16</td>
<td>1759.0 58728 704551</td>
<td>1454220 2545850 4000000 4000070 0.636463</td>
</tr>
<tr>
<td>26</td>
<td>541.9 58112 200364</td>
<td>2363640 4137530 6500000 6501170 0.636543</td>
</tr>
<tr>
<td>32</td>
<td>3484.0 75520 1408943</td>
<td>2908370 5091700 8000000 8000070 0.636463</td>
</tr>
<tr>
<td>39</td>
<td>583.5 74568 214675</td>
<td>3545425 6205745 9750000 9751170 0.636487</td>
</tr>
<tr>
<td>64</td>
<td>7030.0 91776 2817704</td>
<td>5816740 10183330 16000000 16000070 0.636458</td>
</tr>
<tr>
<td>78</td>
<td>652.2 74768 257582</td>
<td>7089680 12411420 19500000 19501170 0.636483</td>
</tr>
<tr>
<td>128</td>
<td>14149.4 141448 5635227</td>
<td>11633480 20366590 32000000 32000070 0.636456</td>
</tr>
<tr>
<td><strong>156</strong></td>
<td>789.4 91204 343042</td>
<td>14178260 24821740 39000000 39000000 0.636455</td>
</tr>
<tr>
<td>256</td>
<td>23219.4 224440 11270279</td>
<td>23266890 40733180 64000000 64000070 0.636456</td>
</tr>
<tr>
<td>312</td>
<td>1824.6 124892 686788</td>
<td>28356520 49643480 78000000 78000000 0.636455</td>
</tr>
<tr>
<td>512</td>
<td>49202.2 390428 22540388</td>
<td>46533780 81466290 128000000 128000070 0.636455</td>
</tr>
<tr>
<td><strong>624</strong></td>
<td>3734.7 207728 1373560</td>
<td>56713040 99286960 156000000 156000000 0.636455</td>
</tr>
</tbody>
</table>
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=\(54.1\)ms (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.
<table>
<thead>
<tr>
<th>limit</th>
<th>( f=100% )</th>
<th>( f=95% )</th>
<th>( f=90% )</th>
<th>( f=86% )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>states</td>
<td>mem</td>
<td>time</td>
<td>states</td>
</tr>
<tr>
<td>1</td>
<td>1300</td>
<td>51.2</td>
<td>1.47</td>
<td>485077</td>
</tr>
<tr>
<td>2</td>
<td>2522</td>
<td>53.7</td>
<td>2.45</td>
<td>806914</td>
</tr>
<tr>
<td>4</td>
<td>4981</td>
<td>54.5</td>
<td>4.62</td>
<td>1499700</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>1481162</td>
<td>124.1</td>
<td>4962.8</td>
<td>3348246</td>
</tr>
<tr>
<td>( \infty )</td>
<td>181869652</td>
<td>1682.2</td>
<td>530604.9</td>
<td></td>
</tr>
</tbody>
</table>

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

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

Herschel deadline violation with $f = 50\%$:
LMAC
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
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.
SMC of LMAC with 4 Nodes

- Wait distribution:
  - geometric
  - uniform

- Network topology:
  - chain
  - ring

- Collision probability

- Collision count

- Power consumption

Pr[energy <= 50000] (<> time>=1000)

Pr[collisions<=50000] (<> time>=1000)
10–Node Star

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

- A Bouncing Ball

![Graph showing simulations of a bouncing ball](image)

Player 1

Player 2

Simulations (1)

- \(x=0\)
- \(x \leq 3\)
- Hit!
- Hit!
Stochastic Hybrid Systems

- A Bouncing Ball

\[ \Pr[\leq 20](<> (time\geq 12 && Ball1.p>4)) \geq \Pr[\leq 20](<> (time\geq 12 && Ball2.p>4)) \]
simulate 1 [<=100]{Temp(0).T, Temp(1).T}

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

Pr[<=100](<> Temp(1).T<=5 and time>30) >= 0.2

Pr[<=100](<> Temp(0).T >= 10)
Hybrid Automata

\[ \mathcal{H} = (L, I_0, \Sigma, X, E, F, \text{Inv}) \]
where
- \( L \) set of locations
- \( I_0 \) initial location
- \( \Sigma = \Sigma^d \cup \Sigma^o \) set of actions
- \( X \) set of continuous variables
  valuation \( \nu: X \rightarrow \mathbb{R} \)
  \((= \mathbb{R}^X)\)
- \( E \) set of edges \((l, g, a, \phi, l')\) with \( g \subseteq \mathbb{R}^X \) and
  \( \phi \subseteq \mathbb{R}^X \times \mathbb{R}^X \) and \( a \in \Sigma \)
- For each \( l \) a delay function
  \( F(l): \mathbb{R}_+ \times \mathbb{R}^X \rightarrow \mathbb{R}^X \)
- For each \( l \) an invariant
  \( \text{Inv}(l) \subseteq \mathbb{R}^X \)
Hybrid Automata

Semantics

- **States**
  \[(l, \nu) \quad \text{where} \quad \nu \in \mathbb{R}^x\]

- **Transitions**
  \[(l, \nu) \to_d (l, \nu') \quad \text{where} \quad \nu' = F(l)(d, \nu)\]
  provided \(\nu' \in \text{Inv}(l)\)

\[(l, \nu) \to_a (l', \nu') \quad \text{if there exists} \quad (l, g, a, \phi, l') \in E\]
with \(\nu \in g\) and
\[(\nu, \nu') \in \phi \quad \text{and} \quad \nu' \in \text{Inv}(l')\]
Stochastic Hybrid Automata

Stochastic Semantics
For each state $s = (l, v)$

**Delay density function**

$\mu_s : \mathbb{R}_{>0} \rightarrow \mathbb{R}$

**Output Probability Function**

$\gamma_s : \Sigma_o \rightarrow [0,1]$  

**Next-state density function**

$\eta_{a,s} : \mathbb{S}_t \rightarrow \mathbb{R}$  

where $a \in \Sigma$.

- **Player 1**
  - $5:2$
  - hit!

- **Player 2**
  - $x=0$
  - hit!
  - $x \leq 3$

- $Pr_1[\text{hit!} \ldots] = \int_{0}^{t=1.43} 2.5 \cdot e^{-2.5t} \, dt$
  - $= [-e^{-2.5t}]_{t=0}^{t=1.43} = 0.97$

- $Pr_2[\text{hit!} \ldots] = \int_{0}^{t=1.43} \frac{1}{3} \, dt$
  - $= [\frac{1}{3} \cdot t]_{t=0}^{t=1.43} = 0.48$

* Dirac's delta functions for deterministic delays / next state.
Stochastic Semantics of NSHAs

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

\[ \pi(s, a_1 a_2 \ldots a_n) := \text{the set of maximal runs from } s \text{ with a prefix } t_1 a_1 t_2 a_2 \ldots t_n a_k \text{ for some } t_1, \ldots, t_n \in R. \]

\[ \mathbb{P}_A(\pi(s, a_1 \ldots a_n)) = \int_{t \geq 0} \mu_{s_c}(t) \cdot \left( \prod_{j \neq c} \int_{\tau > t} \mu_{s_j}(\tau) d\tau \right) \cdot \gamma_{s_c t}(a_1) \cdot \int_{s'} \left( \prod_j \eta_{s_j}^{a_1}(s_j') \cdot \mathbb{P}_A(\pi(s', a_2 \ldots a_n)) ds' \right) dt \]

where \( c = c(a_1) \), and as base case we take \( P_A(\pi(s), \varepsilon) = 1 \).
Stochastic Hybrid Systems

**UPPAAL SMC**

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

**MITL**

**Networks**

- Repeated races between components for outputting
Energy Aware Buildings

With Alexandre David, Dehui Du, Marius Mikucionis, Arne Skou

Fehnker, Ivancic. Benchmarks for Hybrid Systems Verification. HSCC04
Rooms & Heaters – MODELS

\[ T'_i = \sum_{j \neq i} a_{i,j} (T_j - T_i) + b_i (u - T_i) + c_i h_i \]

(a) Template for Room temperature.

\[
T[i]' = (cvec[id]*h[id] + bvec[id]*(u+T[id]) + \sum(j:rid_t(Amat[id][j]*(T[j]+T[id])))/scale
\]

(b) Template for Heater control.

(c) Heating vector c.
Control Strategies – MODELS

<table>
<thead>
<tr>
<th>room</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>off</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
</tr>
<tr>
<td>on</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
</tr>
<tr>
<td>get</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>17</td>
<td>16</td>
</tr>
<tr>
<td>low</td>
<td>15</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>15</td>
</tr>
<tr>
<td>dif</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>imp</td>
<td>1</td>
<td>30</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>pow</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

(a) Strategy 1 (as in [FI04b]).
Weather & User Profile – MODELS

[Image of a complex diagram with graphs and nodes labeled with conditions and transitions.]

EMSIG Autumn School 2015

Kim Larsen [71]
Results – Simulations

simulate 1 [<=2*day] { T[1], T[2], T[3], T[4], T[5] }

simulate 1 [<=2*day] { Heater(1).r, Heater(2).r, Heater(3).r }
Results – Discomfort

\[ \text{Pr}[\leq 2 \text{ day}](\text{<> time} > 0 \land \text{Monitor.Discomfort}) \]

Figure 16: Monitor for comfort and energy.

(c) Daily weather, Static user.

(f) Daily weather, Dynamic user.
Results – Comfort

\[ \Pr[\text{comfort} \leq 2\times \text{day}] \ (<> \ \text{time} \geq 2\times \text{day}) \]

(d) Flat weather, Dynamic user.

(f) Daily weather, Dynamic user.
Results – Energy

\[ \text{Pr}[\text{Monitor.energy} \leq 1000000](\text{time} \geq 2 \times \text{day}) \]

(c) Daily weather, Static user.

(f) Daily weather, Dynamic user.
Result – User Profile

\[
\Pr[\text{Monitor.energy} \leq 1000000](\text{time} \geq 2 \text{ day})
\]
Other Case Studies

- FIREFIRE
- BLUETOOTH
- 10 node LMAC
- Schedulability Analysis for Mix Cr Sys

- Smart Grid Demand / Response
- Energy Aware Buildings
- Genetic Oscillator (HBS)
- Passenger Seating in Aircraft
- Battery Scheduling

EMSIG Autumn School 2015  Kim Larsen [77]
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.

License

The UPPAAL tool is free for non-profit applications. For information about commercial licenses, please email sales(at)uppaal(dot)com.

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 forum page. Bugs should be reported using the bug tracking system. To email the development team directly, please use uppaal(at)list(dot)it(dot)uu(dot)se.

Download

The current official release is UPPAAL 3.4.11 (Jun 23, 2006). A release of UPPAAL 3.6 alpha 3 (Dec 20, 2008) is also available. For more information about UPPAAL version 3.4, we refer to this press release.
THANKS!