Timing and Performance Analysis of Embedded Systems using Model Checking

Kim G. Larsen
CISS – Aalborg University
DENMARK
Embedded Systems

Tasks:
- Computation times
- Deadlines
- Dependencies
- Arrival patterns
- uncertainties

Scheduling Principles (OS)
- EDF, FPS, RMS, DVS, ..

Resources
- Execution platform
- PE, Memory
- Networks
- Drivers
- uncertainties

JTRES, York, September 26, 2011
Kim Guldstrand Larsen [2]
Timing Analysis

Model checking is fixpoint iteration without dynamic abstraction and using set union to collect states.

Abstract Interpretation is fixpoint iteration with dynamic abstraction using lattice join to combine abstract states.

Scheduling:
Assign resources to tasks

Verify no deadlines violated in higher level system for given schedule.

Tasks Resource System

principle

JTRES, York, September 26, 2011
Kim Guldstrand Larsen [3]
Overview

- Timed Automata
- Scheduling
  - Task Graph Scheduling
- Schedulability Analysis
  - Single Processor
  - Multi Processor
- WCET Analysis
- Performance Analysis
  - Statistical Model Checking

JTRES, York, September 26, 2011
Kim Larsen [4]
Timed Automata
Train Crossing

Communication via channels!

Stopable Area

[10,20]

[3,5]

Crossing

[7,15]

River

appr

stop

leave

go

list

enqueue()
dqueue()
front()

id-"parameter"

JTRES, York, September 26, 2011

Kim Larsen [6]
Timed Automata [Train] = Finite State Control + Real Valued Clocks

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

- **Cross**
  - $x \leq 5$
  - $x \geq 7$
  - $x = 0$

- **Start**
  - $x \leq 15$
  - $x = 0$

- **Stop**
  - $x \leq 10$
  - $\text{go[id]}$
  - $x = 0$

- **Appr**
  - $x \leq 20$

- **Resets**
  - $x = 0$

- **invariants**

- **Guards**

- **Synchronizations**

JTRES, York, September 26, 2011
Timed Automata \textbf{[Gate]} = Finite State Control
+ Real Valued Clocks
+ Discrete Variables

\begin{verbatim}
int len[N+1];
int[0,N] len;

// Put an element at the end of the queue
void enqueue(id_t element)
{
    list[len++] = element;
}

// Remove the front element of the queue
void dequeue()
{
    int i = 0;
    len -= 1;
    while (i < len)
    {
        list[i] = list[i + 1];
        i++;
    }
    list[i] = 0;
}
\end{verbatim}
Logical Specifications

- Validation Properties
  - Possibly: $E <> P$

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

- Liveness Properties
  - Eventually: $A<> P$
  - 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).
Scheduling
Composition

**Resource**

- **Idle**
  - Use?
  - \(x := 0\)
  - \(x \geq B\)
  - \(x \leftarrow B\)

- **InUse**

**Task**

- **Init**
  - use!
  - \(B := 6\)

- **Using**
  - done?

- **Done**

**Shared variable**

**Semantics:**

\[
(\text{Idle, Init, B}=0, x=0) \\
d(3.1415) \rightarrow (\text{Idle, Init, B}=0, x=3.1415) \\
\text{use} \rightarrow (\text{InUse, Using, B}=6, x=0) \\
d(6) \rightarrow (\text{InUse, Using, B}=6, x=6) \\
done \rightarrow (\text{Idle, Done, B}=6, x=6)
\]

JTRES, York, September 26, 2011

Kim Larsen [11]
Compute:
\[(D \times (C \times (A + B))) + ((A + B) + (C \times D))\]
using 2 processors

P1 (fast)
P2 (slow)

13 pico-sec !!
Compute:

\[(D \times (C \times (A + B)) + ((A + B) + (C \times D)))\]

using 2 processors

P1 (fast)

P2 (slow)

12 pico-sec

OPTIMAL!!
Compute:

\[(D \times (C \times (A + B))) + ((A + B) + (C \times D))\]
Optimal Scheduling – TIME

Compute:
\[(D \times (C \times (A + B)) + ((A + B) + (C \times D))\]
using 2 processors

A
B
C
D

\[\times\]
\[+\]
\[3\ 4\]
using 2 processors

P1 (fast)
P2 (slow)

C
\[\times\]
\[+\]
7ps
5ps
5ps

P

E<> (Task1.End and … and Task6.End)
## Experimental Results

<table>
<thead>
<tr>
<th>name</th>
<th>#tasks</th>
<th>#chains</th>
<th># machines</th>
<th>optimal</th>
<th>TA</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>437</td>
<td>125</td>
<td>4</td>
<td>1178</td>
<td>1182</td>
</tr>
<tr>
<td>000</td>
<td>452</td>
<td>43</td>
<td>20</td>
<td>537</td>
<td>537</td>
</tr>
<tr>
<td>018</td>
<td>730</td>
<td>175</td>
<td>10</td>
<td>700</td>
<td>704</td>
</tr>
<tr>
<td>074</td>
<td>1007</td>
<td>66</td>
<td>12</td>
<td>891</td>
<td>894</td>
</tr>
<tr>
<td>021</td>
<td>1145</td>
<td>88</td>
<td>20</td>
<td>605</td>
<td>612</td>
</tr>
<tr>
<td>228</td>
<td>1187</td>
<td>293</td>
<td>8</td>
<td>1570</td>
<td>1574</td>
</tr>
<tr>
<td>071</td>
<td>1193</td>
<td>124</td>
<td>20</td>
<td>629</td>
<td>634</td>
</tr>
<tr>
<td>271</td>
<td>1348</td>
<td>127</td>
<td>12</td>
<td>1163</td>
<td>1164</td>
</tr>
<tr>
<td>237</td>
<td>1566</td>
<td>152</td>
<td>12</td>
<td>1340</td>
<td>1342</td>
</tr>
<tr>
<td>231</td>
<td>1664</td>
<td>101</td>
<td>16</td>
<td>t.o.</td>
<td>1137</td>
</tr>
<tr>
<td>235</td>
<td>1782</td>
<td>218</td>
<td>16</td>
<td>t.o.</td>
<td>1150</td>
</tr>
<tr>
<td>233</td>
<td>1980</td>
<td>207</td>
<td>19</td>
<td>1118</td>
<td>1121</td>
</tr>
<tr>
<td>294</td>
<td>2014</td>
<td>141</td>
<td>17</td>
<td>1257</td>
<td>1261</td>
</tr>
<tr>
<td>295</td>
<td>2168</td>
<td>965</td>
<td>18</td>
<td>1318</td>
<td>1322</td>
</tr>
<tr>
<td>292</td>
<td>2333</td>
<td>318</td>
<td>3</td>
<td>8009</td>
<td>8009</td>
</tr>
<tr>
<td>298</td>
<td>2399</td>
<td>303</td>
<td>10</td>
<td>2471</td>
<td>2473</td>
</tr>
</tbody>
</table>

Symbolic A*  
Branch-&-Bound  
60 sec

Abdeddaïm, Kerbaa, Maler
Task Graph Scheduling – Revisited

Compute:
\[(D \times (C \times (A + B)) + ((A + B) + (C \times D))\]

using 2 processors

P1 (fast)
P2 (slow)

ENERGY:
P1
P2

1.39 nano-joule!!
Compute:

\[(D \times (C \times (A + B)) + ((A + B) + (C \times D)))\]

using 2 processors

**Energy:**

- **P1** (fast)
  - Idle: 10W
  - In use: 90W
- **P2** (slow)
  - Idle: 20W
  - In use: 30W

**Energy:** 1.32 nano-joule

**OPTIMAL!!**
Zones – from infinite to finite

State
(n, x=3.2, y=2.5 )

Symbolic state (set)
(n, \[2 \leq x \leq 4 \land 1 \leq y \leq 3 \land y-x \leq 0\) )
Zones – Operations

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

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

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

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

\[ \text{Reset} \]

\[ \text{Extrapolation} \]

\[ \text{Convex Hull} \]

\[ + \text{ COST Information} \]

\[ C = 4 \cdot x - 2 \cdot y + 3 \]
Symbolic B & B Algorithm

Cost := \infty
Passed := \emptyset
Waiting := \{(l_0, Z_0)\}

while Waiting \neq \emptyset do
    select \((l, Z)\) from Waiting
    if \(l = l_g\) and minCost(Z) < Cost then
        Cost := minCost(Z)
    if minCost(Z) + Rem(l, Z) \geq Cost then break

    if for all \((l, Z')\) in Passed: \(Z' \notin Z\) then
        \(\leq\) is a well-quasi ordering which guarantees termination!

\(Z' \leq Z\)
\(Z'\) is bigger & cheaper than \(Z\)

Competitive to MIPL on a number of benchmarks, e.g. Aircraft Landing
Schedulability Analysis
Task Scheduling

utilization of CPU

\[ P(i), [E(i), L(i)], \ldots : \text{period or earliest/latest arrival or} \ldots \text{for } T_i \]

\[ C(i) : \text{execution time for } T_i \]

\[ D(i) : \text{deadline for } T_i \]

\( T_1 \)

\( T_2 \)

\( T_n \)

Scheduler

\{ T_4 , T_1 , T_3 \} \text{ ready ordered according to some given priority: (e.g. Fixed Priority, Earliest Deadline,..)}
Utilisation-Based Analysis

• A simple sufficient but not necessary schedulability test exists

\[ U = \sum_{i=1}^{N} \frac{C_i}{T_i} \leq N \left( 2^{1/N} - 1 \right) \]

\[ U \leq 0.69 \quad \text{as} \quad N \to \infty \]

Where \( C \) is WCET and \( T \) is period

Response Time Equation

\[ R_i = C_i + \sum_{j \in \text{hpt}(i)} \left[ \frac{R_j}{T_j} \right] C_j \]

Where \( \text{hpt}(i) \) is the set of tasks with priority higher than task \( i \)

Solve by forming a recurrence relationship:

\[ w_i^{n+1} = C_i + \sum_{j \in \text{hpt}(i)} \left[ \frac{w_j^n}{T_j} \right] C_j \]

The set of values \( w_0, w_1, w_2, \ldots, w^n \ldots \) is monotonically non decreasing. When \( w_0 = w_0^* \) the solution to the equation has been found, \( w_0^* \) must not be greater than \( R_i \) (e.g. 0 or \( C_i \)).

Basic WCRT Analysis

• "Classical" scheduling analysis technique
• For all tasks \( i \): \( WCRT_i \leq \text{Deadline}_i \)

\[ R_i = D_i + C_i + \sum_{j \in \text{hpt}(i)} \left[ \frac{R_j}{T_j} \right] C_j \]

Blocking times for priority inheritance protocol (BSW):

\[ \text{Blocking}(i) = \sum_{r=1}^{R_i} \text{usage}(r,i) \cdot \text{WCET}_{\text{critical section}}(r) \]

Blocking times for priority ceiling protocol (ASW):

\[ \text{Blocking}(i) = \max \text{usage}(r,i) \cdot \text{WCET}_{\text{critical section}}(r) \]

Simple to perform

- Overly conservative
- Limited settings
- Single-processor

Tomorrow: Sanjoy Baruah
Modeling Task

Scheduler

T₁ ready done
T₂
Tₙ stop run

JTRES, York, September 26, 2011 Kim Larsen [25]
Modeling Scheduler

Scheduler

T1 ready done
T2 stop run
Tn done

len > 0
run[front()]!
len == 0
ready[e]? enqueue(e)

Occ

Free
e : id_t
len == 0
done[e]?
dequeue()
e : id_t
run[front()]!

e : id_t
ready[e]? enqueue(e)
Modeling Queue

In UPPAAL 4.0
User Defined Function

```c
// 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 = Safety Property

May be extended with preemption

\[ A \neg (\text{Task0.Error or Task1.Error or ...}) \]
Preemption – Stopwatches!

Scheduler

Task

Defeating undecidability 😊
Multi-Processor

JTRES, York, September 26, 2011  Kim Larsen [30]
Handling realistic applications?

Smart phone:

JPEG Encoder

JPEG Decoder

GSM Decoder

MP3 Decoder

GSM Encoder

Jan Madsen
Aske Brekling
Michael R. Hansen/ DTU

[Application from Marcus Schmitz, TU Linkoping]

JTRES, York, September 26, 2011

Kim Larsen [31]
Timed Automata for a task
Smart phone

- Tasks: 114
- Deadlines: [0.02: 0.5] sec
- Execution: [52 : 266.687] cycles
- Platform:
  - 6 processors, 25 MHz
  - 1 bus
  - Verified in 1.5 hours!
bool resource[N];

bool available(id_t id)
{
    return !resource[need[id]];
}

void take(id_t id)
{
    assert(!resource[need[id]]);
    resource[need[id]] = true;
}

void release(id_t id)
{
    assert(resource[need[id]]);
    resource[need[id]] = false;
}
ESA Missions

- Solar System, cold dust clouds and cores, star and galaxy formations, cataloging galaxies, gravitational lensing, cosmic microwave background, topology of the universe...

- Terma: Develop software for Attitude and Orbit Control System
Application software (ASW)
- built and tested by Terma:
- does attitude and orbit control, telecommanding, 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
UPPAAL Model

- One template for CPU scheduler:
  - maintains a queue of ready tasks
  - schedules tasks with highest priority in the queue
  - reschedule if higher priority task arrives to the queue

- One template per each ASW task.

- Two templates for BSW tasks: 1 plain, 1 using resource.

- One template for “idle” task to count CPU utilization.

- WCET is modeled by stopwatches and lower bound.

- WCRT is modeled by stopwatches.

- Deadline is enforced as guard on WCRT.

- System is schedulable if no deadline violated.

- CPU load is \( \frac{\text{used time}}{\text{total time}} \).
Modeling in UPPAAL

UPPAAL 4.1 Framework
ISoLA 2010

JTRES, York, September 26, 2011
Kim Larsen [38]
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

<table>
<thead>
<tr>
<th>ID</th>
<th>Task</th>
<th>Specification</th>
<th>Blocking times</th>
<th>WCRT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Period WCET Deadline</td>
<td>Terma UPPAAL Diff</td>
<td>Terma UPPAAL Diff</td>
</tr>
<tr>
<td>1</td>
<td>RTEMS_RTC</td>
<td>10.000 0.013 1.000</td>
<td>0.035 0.035</td>
<td>0.050 0.037</td>
</tr>
<tr>
<td>2</td>
<td>AswSync_SyncPulseIsr</td>
<td>250.000 0.070 1.000</td>
<td>0.035 0.035</td>
<td>0.120 0.083</td>
</tr>
<tr>
<td>3</td>
<td>Hk_SamplerIsr</td>
<td>125.000 0.070 1.000</td>
<td>0.035 0.035</td>
<td>0.120 0.070</td>
</tr>
<tr>
<td>4</td>
<td>SwCyc_CycStartIsr</td>
<td>250.000 0.200 1.000</td>
<td>0.035 0.035</td>
<td>0.320 0.103</td>
</tr>
<tr>
<td>5</td>
<td>SwCyc_CycEndIsr</td>
<td>250.000 0.100 1.000</td>
<td>0.035 0.035</td>
<td>0.220 0.113</td>
</tr>
<tr>
<td>6</td>
<td>Rt1553_Isr</td>
<td>15.625 0.070 1.000</td>
<td>0.035 0.035</td>
<td>0.290 0.173</td>
</tr>
<tr>
<td>7</td>
<td>Bc1553_Isr</td>
<td>20.000 0.070 1.000</td>
<td>0.035 0.035</td>
<td>0.360 0.243</td>
</tr>
<tr>
<td>8</td>
<td>Spw_Isr</td>
<td>39.000 0.070 2.000</td>
<td>0.035 0.035</td>
<td>0.430 0.313</td>
</tr>
<tr>
<td>9</td>
<td>Obdh_Isr</td>
<td>250.000 0.070 2.000</td>
<td>0.035 0.035</td>
<td>0.500 0.383</td>
</tr>
<tr>
<td>10</td>
<td>RtSdb_P_1</td>
<td>15.625 0.150 15.625</td>
<td>3.650 3.650</td>
<td>4.330 3.533</td>
</tr>
<tr>
<td>11</td>
<td>RtSdb_P_2</td>
<td>125.000 0.400 15.625</td>
<td>3.650 3.650</td>
<td>4.870 0.933</td>
</tr>
<tr>
<td>12</td>
<td>RtSdb_P_3</td>
<td>250.000 0.170 15.625</td>
<td>3.650 3.650</td>
<td>5.110 1.103</td>
</tr>
<tr>
<td>14</td>
<td>FdirEvents</td>
<td>250.000 5.000 230.220</td>
<td>0.720 0.720</td>
<td>7.180 5.153</td>
</tr>
<tr>
<td>15</td>
<td>NominalEvents_1</td>
<td>250.000 0.720 230.220</td>
<td>0.720 0.720</td>
<td>7.900 5.873</td>
</tr>
<tr>
<td>16</td>
<td>MainCycle</td>
<td>250.000 0.400 230.220</td>
<td>0.720 0.720</td>
<td>8.370 6.273</td>
</tr>
<tr>
<td>17</td>
<td>HkSampler_P_2</td>
<td>125.000 0.500 62.500</td>
<td>3.650 3.650</td>
<td>11.960 5.380</td>
</tr>
<tr>
<td>18</td>
<td>HkSampler_P_1</td>
<td>250.000 6.000 62.500</td>
<td>3.650 3.650</td>
<td>18.460 11.615</td>
</tr>
<tr>
<td>19</td>
<td>Acb_P</td>
<td>250.000 6.000 50.000</td>
<td>3.650 3.650</td>
<td>24.680 6.473</td>
</tr>
<tr>
<td>20</td>
<td>IocCyc_P</td>
<td>250.000 3.000 50.000</td>
<td>3.650 3.650</td>
<td>27.820 9.473</td>
</tr>
<tr>
<td>21</td>
<td>PrimaryF</td>
<td>250.000 34.050 59.600</td>
<td>5.770 0.966 4.804</td>
<td>65.470 54.115 11.355</td>
</tr>
<tr>
<td>22</td>
<td>RCSControlF</td>
<td>250.000 4.070 239.600</td>
<td>12.120 0.120</td>
<td>76.040 53.994</td>
</tr>
<tr>
<td>23</td>
<td>Obh_P</td>
<td>100.000 1.100 100.000</td>
<td>9.630 0.963</td>
<td>74.720 2.503</td>
</tr>
<tr>
<td>24</td>
<td>Hk_P</td>
<td>250.000 2.750 250.000</td>
<td>1.035 0.103</td>
<td>6.800 4.953</td>
</tr>
<tr>
<td>25</td>
<td>StsMon_P</td>
<td>250.000 3.300 125.000</td>
<td>16.070 0.822 15.248</td>
<td>85.050 17.863 67.187</td>
</tr>
<tr>
<td>26</td>
<td>TmGen_P</td>
<td>250.000 4.860 250.000</td>
<td>4.260 0.426</td>
<td>77.650 9.813</td>
</tr>
<tr>
<td>27</td>
<td>Sgm_P</td>
<td>250.000 4.020 250.000</td>
<td>1.040 0.104</td>
<td>18.680 14.796</td>
</tr>
<tr>
<td>28</td>
<td>TeRouter_P</td>
<td>250.000 0.500 250.000</td>
<td>1.035 0.103</td>
<td>10.310 11.806</td>
</tr>
<tr>
<td>29</td>
<td>Cmd_P</td>
<td>250.000 14.000 250.000</td>
<td>26.110 1.262 24.848</td>
<td>114.920 94.346 20.574</td>
</tr>
<tr>
<td>30</td>
<td>NominalEvents_2</td>
<td>250.000 1.780 230.220</td>
<td>12.480 0.1248</td>
<td>102.760 65.177</td>
</tr>
<tr>
<td>31</td>
<td>SecondaryF_1</td>
<td>250.000 20.960 189.600</td>
<td>27.650 0.2765</td>
<td>141.550 110.666</td>
</tr>
<tr>
<td>32</td>
<td>SecondaryF_2</td>
<td>250.000 39.690 230.220</td>
<td>48.450 0.4845</td>
<td>204.050 154.556</td>
</tr>
<tr>
<td>33</td>
<td>Bkgnd_P</td>
<td>250.000 0.200 250.000</td>
<td>0.000 0.000</td>
<td>154.090 15.046</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>5</td>
<td>474.6 58796 178432</td>
<td>545900 955270 1500000 1501170 0.636847</td>
</tr>
<tr>
<td>6</td>
<td>912.3 58856 352365</td>
<td>727110 1279260 2000000 2000070 0.636480</td>
</tr>
<tr>
<td>7</td>
<td>507.7 58796 186091</td>
<td>1181855 2069385 3250000 3251240 0.636734</td>
</tr>
<tr>
<td>8</td>
<td>1759.0 58728 704551</td>
<td>1454220 2545850 4000000 4000070 0.636463</td>
</tr>
<tr>
<td>9</td>
<td>541.9 58112 200364</td>
<td>2363640 4137530 6500000 6501170 0.636543</td>
</tr>
<tr>
<td>10</td>
<td>3484.0 75520 1408943</td>
<td>2908370 5091700 8000000 8000070 0.636463</td>
</tr>
<tr>
<td>11</td>
<td>583.5 7468 214657</td>
<td>3545425 6205745 9750000 9751170 0.636487</td>
</tr>
<tr>
<td>12</td>
<td>7030.0 91776 2817704</td>
<td>5816740 10183330 16000000 16000070 0.636458</td>
</tr>
<tr>
<td>13</td>
<td>652.2 74768 257582</td>
<td>7089680 12411420 19500000 19501100 0.636483</td>
</tr>
<tr>
<td>14</td>
<td>14194.9 141448 5635227</td>
<td>11633480 20366590 32000000 32000070 0.636456</td>
</tr>
<tr>
<td>15</td>
<td>789.4 91204 343402</td>
<td>14178260 24821740 39000000 39000000 0.636455</td>
</tr>
<tr>
<td>16</td>
<td>23219.4 224440 11270279</td>
<td>23266890 40733180 64000000 64000070 0.636456</td>
</tr>
<tr>
<td>17</td>
<td>1824.6 124892 686788</td>
<td>28356520 49643480 78000000 78000000 0.636455</td>
</tr>
<tr>
<td>18</td>
<td>49202.2 390428 22540388</td>
<td>46533780 81466290 128000000 128000070 0.636455</td>
</tr>
<tr>
<td>19</td>
<td>3734.7 207728 1373560</td>
<td>56713040 99286960 156000000 156000000 0.636455</td>
</tr>
</tbody>
</table>
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

% : difference

In BCET and WCET

limit: no of 250ms cycles

<table>
<thead>
<tr>
<th>limit</th>
<th>0%</th>
<th></th>
<th></th>
<th>5%</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>states</td>
<td>mem</td>
<td>time</td>
<td>states</td>
<td>mem</td>
<td>time</td>
</tr>
<tr>
<td>1</td>
<td>1300</td>
<td>51.2</td>
<td>1.47</td>
<td>485077</td>
<td>82.0</td>
<td>000.1</td>
</tr>
<tr>
<td>2</td>
<td>2522</td>
<td>53.7</td>
<td>2.45</td>
<td>806914</td>
<td>82.0</td>
<td>000.1</td>
</tr>
<tr>
<td>4</td>
<td>4981</td>
<td>54.5</td>
<td>4.62</td>
<td>1499700</td>
<td>250.3</td>
<td>000.8</td>
</tr>
</tbody>
</table>

6 Days

<table>
<thead>
<tr>
<th>limit</th>
<th>10%</th>
<th></th>
<th></th>
<th>14%</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>states</td>
<td>mem</td>
<td>time</td>
<td>states</td>
<td>mem</td>
<td>time</td>
</tr>
<tr>
<td>1</td>
<td>1481162</td>
<td>124.1</td>
<td>4962.8</td>
<td>3348246</td>
<td>186.9</td>
<td>23986.5</td>
</tr>
<tr>
<td>2</td>
<td>2414679</td>
<td>139.7</td>
<td>7755.5</td>
<td>5253778</td>
<td>198.7</td>
<td>33299.2</td>
</tr>
<tr>
<td>4</td>
<td>4421630</td>
<td>138.3</td>
<td>13720.0</td>
<td>9231399</td>
<td>274.6</td>
<td>51176.6</td>
</tr>
<tr>
<td>8</td>
<td>9093562</td>
<td>156.5</td>
<td>3112.3</td>
<td>18240030</td>
<td>364.6</td>
<td>102932.4</td>
</tr>
<tr>
<td>16</td>
<td>17798572</td>
<td>176.0</td>
<td>6012.4</td>
<td>35432003</td>
<td>520.4</td>
<td>158816.7</td>
</tr>
<tr>
<td>∞</td>
<td>181869652</td>
<td>1682.2</td>
<td>530604.9</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1 Day

error may be reachable
WCET Analysis
WCET: Worst Case Execution Time

In general: hard or impossible to predict

Determine tight upper time bound instead

Kim Larsen [45]
METAMOC
Modular Execution Time Analysis using MOdel Checking

with
Andreas Dalsgaard
Mads Christian Olesen
Martin Toft
René Rydhof Hansen

Abstract process model and value analysis

Abstract hardware model with caching and pipelining

Timed automata models for hardware components and process functions:

WCET 42 cycles
Overview of METAMOC
Modeling using Timed Automata
Analysis in UPPAAL
GUI for METAMOC

http://metamoc.martintoft.dk
Status

- Started out with ARM9 support
  - Five stage pipeline, instruction cache, data caches, simple main memory

- Now
  - .. support for ARM7, ARM9 and ATMEL AVR 8-bit
  - .. with modest effort
Experiments

- Evaluation using WCET benchmark programs from Mälardalen Real-Time Research Centre
  - Applicability
  - Performance
- Discarded a number of programs
  - Floating point operations handled by software routines
  - Dynamic jumps
  - Some programs do not compile
- 21 programs for ARM and 19 programs for AVR
- Manually annotated loop bounds
Experiments

<table>
<thead>
<tr>
<th>ARM9, 21 benchmarks</th>
<th>ATMEL AVR 8-bit, 19 benchmarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysable without caches</td>
<td>Analysable</td>
</tr>
<tr>
<td>Unanalysable, state space explosion</td>
<td>Unanalysable, state space explosion</td>
</tr>
<tr>
<td>Analysable with instruction cache</td>
<td></td>
</tr>
<tr>
<td>Unanalysable, state space explosion</td>
<td>1</td>
</tr>
<tr>
<td>Analysable with data and instruction cache</td>
<td>20</td>
</tr>
<tr>
<td>Unanalysable, state space explosion</td>
<td>1</td>
</tr>
<tr>
<td>Manual modification of e.g. data cache size</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Relative improvement in WCET for ARM9.

Analysis times in minutes for AVR and ARM9.
Constructing UPPAAL models
TetaJ Control Flow Graphs (TCFGs)
  Forms the interface for model construction
  Achieves reusability:
    - Java Bytecode to TCFG
    - AVR to TCFG
CFG analyses are applied yielding annotated TCFGs
  Loop detection
  Condition optimisation
  Progress measures in UPPAAL
TCFGs are transformed to models
The models are combined to one model
UPPAAL conducts the WCET analysis on the final model
# Evaluation of TetaJ

## PERFORMANCE

<table>
<thead>
<tr>
<th>Optimisation</th>
<th>Analysis time</th>
<th>States explored</th>
<th>Max memory usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>No optimisations</td>
<td>14h 51m 17s</td>
<td>41,854,143</td>
<td>3,905 MB</td>
</tr>
<tr>
<td>Only progress measures</td>
<td>4d 12h 7m 8s</td>
<td>408,223,029</td>
<td>589 MB</td>
</tr>
<tr>
<td>Only state space reduction</td>
<td>13h 33m 21s</td>
<td>41,854,143</td>
<td>2,426 MB</td>
</tr>
<tr>
<td>Only condition optimisation</td>
<td>1m 16s</td>
<td>53,732</td>
<td>294 MB</td>
</tr>
<tr>
<td>Only template reduction</td>
<td>4h 46m 41s</td>
<td>41,854,143</td>
<td>3,851 MB</td>
</tr>
</tbody>
</table>

## ACCURACY

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Measured WCET</th>
<th>TetaJ WCET</th>
<th>Pessimism</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iterative fibonacci</td>
<td>46,642 clock cycles</td>
<td>46,933 clock cycles</td>
<td>0.6%</td>
</tr>
<tr>
<td>Factorial</td>
<td>39,726 clock cycles</td>
<td>40,939 clock cycles</td>
<td>3.1%</td>
</tr>
<tr>
<td>Reverse ordering</td>
<td>64,436 clock cycles</td>
<td>81,919 clock cycles</td>
<td>27.1%</td>
</tr>
<tr>
<td>Bubble sort</td>
<td>907,103 clock cycles</td>
<td>2,270,401 clock cycles</td>
<td>150.3%</td>
</tr>
<tr>
<td>Binary Search</td>
<td>54,430 clock cycles</td>
<td>99,301 clock cycles</td>
<td>82.4%</td>
</tr>
<tr>
<td>Insertion Sort</td>
<td>849,353 clock cycles</td>
<td>3,740,769 clock cycles</td>
<td>440.4%</td>
</tr>
</tbody>
</table>
Program Slicing & Model-Checking

Jean-Luc Béchenec
Franck Cassez
IRCCyN

ARM920T

Program P

Slicing

Reduced $\text{Aut}(P')$ generates $\mathcal{L}(P)$ Finite Automaton

Synchronization $\text{Aut}(P') \parallel \text{Aut}(H)$

$\text{UPPAAL}$

$\text{WCET}(H,P)$

Hardware $H$

$\text{HDL, ...}$

$\text{Aut}(H)$ Timed Automaton

JTRES, York, September 26, 2011
Kim Larsen [56]
<table>
<thead>
<tr>
<th>Program</th>
<th>loc</th>
<th>UPPAAL Time States Explored</th>
<th>Computed WCET (C)</th>
<th>Measured WCET (M)</th>
<th>(\frac{(C-M)}{M} \times 100)</th>
<th>Slice</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single-Path Programs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fb-O0</td>
<td>74</td>
<td>1.74s/74181</td>
<td>8098</td>
<td>8064</td>
<td>0.42%</td>
<td>47/131</td>
</tr>
<tr>
<td>fb-O1</td>
<td>74</td>
<td>0.61s/22332</td>
<td>2597</td>
<td>2544</td>
<td>2.0%</td>
<td>18/72</td>
</tr>
<tr>
<td>fb-O2</td>
<td>74</td>
<td>0.3s/9710</td>
<td>1209</td>
<td>1164</td>
<td>3.8%</td>
<td>22/71</td>
</tr>
<tr>
<td>jarne-complex-O0*</td>
<td>65</td>
<td>1.15s/38014</td>
<td>4264</td>
<td>4164</td>
<td>2.4%</td>
<td>78/173</td>
</tr>
<tr>
<td>jarne-complex-O1*</td>
<td>65</td>
<td>0.48s/14600</td>
<td>1715</td>
<td>1680</td>
<td>2.0%</td>
<td>30/89</td>
</tr>
<tr>
<td>jarne-complex-O2*</td>
<td>65</td>
<td>0.46s/13004</td>
<td>1557</td>
<td>1536</td>
<td>1.3%</td>
<td>32/78</td>
</tr>
<tr>
<td>fdct-O1</td>
<td>238</td>
<td>1.67s/60418</td>
<td>4245</td>
<td>4092</td>
<td>3.7%</td>
<td>100/363</td>
</tr>
<tr>
<td>fdct-O2</td>
<td>238</td>
<td>3.24s/55285</td>
<td>19231</td>
<td>18984</td>
<td>1.3%</td>
<td>166/3543</td>
</tr>
<tr>
<td>Single-Path Programs‡ with MUL/MLA/SMULL instructions (instructions durations depend on data)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fdct-O0</td>
<td>238</td>
<td>2.41s/85007</td>
<td>[11242,11800]</td>
<td>11448</td>
<td>3.0%</td>
<td>253/831</td>
</tr>
<tr>
<td>matmult-O0*</td>
<td>162</td>
<td>5m9s/10531230</td>
<td>[502850,529250]</td>
<td>511584</td>
<td>0.1%</td>
<td>158/314</td>
</tr>
<tr>
<td>matmult-O2*</td>
<td>162</td>
<td>43.78s/1780548</td>
<td>[122046,148299]</td>
<td>116844</td>
<td>5.4%</td>
<td>75/288</td>
</tr>
<tr>
<td>jfdcdint-O0</td>
<td>374</td>
<td>2.79s/100784</td>
<td>[12699,12699]</td>
<td>12588</td>
<td>0.8%</td>
<td>159/792</td>
</tr>
<tr>
<td>jfdcdint-O1</td>
<td>374</td>
<td>1.02s/35518</td>
<td>[4897,4899]</td>
<td>4668</td>
<td>7.0%</td>
<td>25/325</td>
</tr>
<tr>
<td>jfdcdint-O2</td>
<td>374</td>
<td>5.38s/175661</td>
<td>[16746,16938]</td>
<td>16380</td>
<td>3.4%</td>
<td>56/2512</td>
</tr>
<tr>
<td>Multiple-Path Programs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bs-O0</td>
<td>174</td>
<td>42.6s/1421474</td>
<td>1068</td>
<td>1056</td>
<td>1.1%</td>
<td>75/151</td>
</tr>
<tr>
<td>bs-O1</td>
<td>174</td>
<td>28s/1214673</td>
<td>738</td>
<td>720</td>
<td>2.5%</td>
<td>28/82</td>
</tr>
<tr>
<td>bs-O2</td>
<td>174</td>
<td>15s/655870</td>
<td>628</td>
<td>600</td>
<td>4.6%</td>
<td>28/65</td>
</tr>
<tr>
<td>cnt-O0*</td>
<td>115</td>
<td>2.3s/76238</td>
<td>9028</td>
<td>8836</td>
<td>2.1%</td>
<td>99/235</td>
</tr>
<tr>
<td>cnt-O1*</td>
<td>115</td>
<td>1s/27279</td>
<td>4123</td>
<td>3996</td>
<td>3.1%</td>
<td>42/129</td>
</tr>
<tr>
<td>cnt-O2*</td>
<td>115</td>
<td>0.5s/11540</td>
<td>3065</td>
<td>2928</td>
<td>4.6%</td>
<td>39/263</td>
</tr>
<tr>
<td>insertsort-O0*</td>
<td>91</td>
<td>10m3s/24250737</td>
<td>3133</td>
<td>3108</td>
<td>0.8%</td>
<td>79/175</td>
</tr>
<tr>
<td>insertsort-O1*</td>
<td>91</td>
<td>7m2s/11455293</td>
<td>1533</td>
<td>1500</td>
<td>2.2%</td>
<td>40/115</td>
</tr>
<tr>
<td>insertsort-O2*</td>
<td>91</td>
<td>11.5s/38/292</td>
<td>13/1</td>
<td>1344</td>
<td>2.0%</td>
<td>43/108</td>
</tr>
<tr>
<td>ns-O0*</td>
<td>497</td>
<td>83.4s/3064315</td>
<td>30968</td>
<td>30732</td>
<td>0.8%</td>
<td>132/215</td>
</tr>
<tr>
<td>ns-O1*</td>
<td>497</td>
<td>11.3s/368719</td>
<td>11701</td>
<td>11568</td>
<td>1.1%</td>
<td>61/124</td>
</tr>
<tr>
<td>ns-O2*</td>
<td>497</td>
<td>29s/1030746</td>
<td>7343</td>
<td>7236</td>
<td>1.4%</td>
<td>566/863</td>
</tr>
</tbody>
</table>
WCA: IPET versus MC

- Benedikt Huber, Martin Schoeberl

Checking

JTRES, York, September 26, 2011
### WCA: IPET versus MC

<table>
<thead>
<tr>
<th>Problem</th>
<th>IPET</th>
<th>UPPAAL Breadth First</th>
<th>UPPAAL Progress</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Verify</td>
<td>Search</td>
</tr>
<tr>
<td>Prol DCT</td>
<td>0.00</td>
<td>0.09</td>
<td>1.21</td>
</tr>
<tr>
<td>Prol GCD</td>
<td>0.00</td>
<td>3.10</td>
<td>47.65</td>
</tr>
<tr>
<td>Prol MatrixMult</td>
<td>0.01</td>
<td>0.21</td>
<td>3.52</td>
</tr>
<tr>
<td>Prol CRC</td>
<td>0.01</td>
<td>1.93</td>
<td>16.22</td>
</tr>
<tr>
<td>Prol BubbleSort</td>
<td>0.00</td>
<td>7.63</td>
<td>197.91</td>
</tr>
<tr>
<td>CRC LineFollower</td>
<td>0.00</td>
<td>0.11</td>
<td>1.07</td>
</tr>
<tr>
<td>CRC Lift</td>
<td>0.01</td>
<td>0.01</td>
<td>2.07</td>
</tr>
<tr>
<td>CRC Dudp</td>
<td>0.03</td>
<td>8.43</td>
<td>84.69</td>
</tr>
<tr>
<td>CRC Udplp simplified</td>
<td>0.04</td>
<td>0.64</td>
<td>7.28</td>
</tr>
<tr>
<td>CRC Kfl (no cache)</td>
<td>0.04</td>
<td>92.19</td>
<td>1229.42</td>
</tr>
<tr>
<td>CRC Kfl simplified (no cache)</td>
<td>0.04</td>
<td>33.81</td>
<td>444.73</td>
</tr>
<tr>
<td>CRC Kfl (8 blocks)</td>
<td>0.13</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>CRC Kfl simplified (8 blocks)</td>
<td>0.13</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

**Table 3. Analysis example**

**SUP: EXPR in UPPAAL**
Symbolic B & B Algorithm

Cost := $0$
Passed := $\emptyset$
Waiting := $\{(l_0, Z_0)\}$

while Waiting $\neq \emptyset$ do
  select $(l, Z)$ from Waiting
  if $l = l_g$ and $\maxCost(Z) \leq \text{Cost}$ then
    Cost := $\maxCost(Z)$
    if $\maxCost(Z) + \text{Rem}(l, Z) \leq \text{Cost}$ then break
  if for all $(l, Z')$ in Passed: $Z' \not\subseteq Z$ then
    add $(l, Z)$ to Passed
    add all $(l', Z')$ with $(l, Z) \rightarrow (l')$

return Cost

Upper bound on remaining cost to goal from $(l, Z)$

$Z' \leq Z$
$Z'$ is bigger & more costly than $Z$ ($\Rightarrow$ prog msr)

All states reaches eventually goal guarantees termination!

Kim Larsen [60]
Symbolic B & B Algorithm

Cost := 0
Passed := ∅
Waiting := \{(l_0, Z_0)\}

while Waiting ≠ ∅ do
    select (l, Z) from Waiting
    if \(\text{Cost}(Z) + \text{Rem}(l, Z) \leq \text{Cost}\) then break
    if for all \((l', Z')\) in Passed: \(Z' \not\in Z\) then
        add \((l, Z)\) to Passed
        add all \((l', Z')\) with \((l, Z) \rightarrow (l', Z')\)
    return Cost

Upper bound on remaining cost to goal from \((l, Z)\)

TO BE IMPLEMENTED!

All states reaches eventually goal guarantees termination!
More..

- **Schedulability Analysis**
  - TIMES tool [TACAS 2002]
  - Multitasking applications under OSEK [RTS 2008]
  - CREOL Modular schedulability [FSEN09]
  - SARTS Java byte code on FPGA [JTRES08]
  - Schedulability Analysis Using UPPAAL 4.1 [Model-Based Design for ES, CRC Press 2010]
  - ARTS: MPSoC Schedulability [Model-Based Design for ES, CRC Press 2010]

- **Worst Case Execution Time Analysis**
  - WCA [Schoeberl 2010]
  - METAMOC [WCET10, NSF11]
  - TetaJ [AAU 2011]
  - Combining AI & MC for Timing Analysis of MultiCore [Yi et al, RTSS10]
  - WCET analysis of Multicore using UPPAAL [Petterson et al]

sarts.boegholm.dk
metamoc.martintoft.dk
tetaj.dk
www.irccyn.fr/franck/wcet/
Performance Analysis using Statistical Model Checking

Collaborators:
Peter Bulychev, Alexandre David
Axel Legay, Marius Mikucionis
Wang Zheng
Jonas van Vliet, Danny Poulsen

CAV 2011, PDMC 2011, FORMATS 2011
A[] forall (i : id_t) forall (j : id_t) Train(i).Cross && Train(j).Cross imply i == j

E<> Train(0).Cross and Train(1).Stop

Train(0).Appr --> Train(0).Cross

A<> .. E[] ..

sup: .. inf: ..

Limited quantitative analysis

Pr[ <> Time ≤ 500 and Train(0).Cross] ≥ 0.7
Pr[Train(0).Appr --> Time ≤ 100 Train(0).Cross] ≥ 0.4

Performance properties

State-space explosion

Safety

Reachability

Liveness
Performance properties

- $\Pr[\leq 200](\diamond \text{Train(5).Cross})$
- $\Pr[\leq 100](\diamond \text{Train(0).Cross}) \geq 0.8$
- $\Pr[\leq 100](\diamond \text{Train(5).Cross}) \geq \Pr[\leq 100](\diamond \text{Train(1).Cross})$

State-space explosion

Generate runs
Stochastic Semantics of TA

Exponential Distribution

Safe

appr[id]!

x=0

Aprr

x<=20

x<=10

stop[id]?

x=0

x>=3

leave[id]!

Uniform Distribution

Cross

x<=5

x>=7

x=0

Start

x<=15

go[id]?

x=0

Composition =
Repeated races between components

Input enabled

JTRES, York, September 26, 2011
Kim Larsen [66]
Queries in UPPAAL SMC

\[ \text{Pr[ } \leq 200\text{]}(\text{<> Train(5).Cross}) \]

Probability Clopper-Pearson CLs

Parameters: \( \alpha = 0.01, \varepsilon = 0.01 \), bucket width=0.587972, bucket count=200.
Runs: 26492 in total, 26492 displayed, 0 remaining.
Probability sums: 1 displayed, 0 remaining.
Average: 40.843.
Pr[ <= 100](<> Train(0).Cross) >= 0.8

Pr[ <= 100](<> Train(0).Cross) >= 0.5
Queries in UPPAAL SMC

\[ \Pr[\leq 100](\langle\rangle \text{Train(5)} \cdot \text{Cross}) \geq \Pr[\leq 100](\langle\rangle \text{Train(1)} \cdot \text{Cross}) \]

Probability comparison

Value 0.0 means less-than is true.
Value 0.5 means probabilities are indistinguishable.
Value 1.0 means greater-than is true.
Analysis Tool: Plot Composer
SMC in UPPAAL 4.1.4

- Constant Slope Timed Automata
  - **Clocks** may have different (integer) **slope** in different locations.
  - **Branching edges** with discrete probabilities (weights).
  - **Beyond** Priced TA, Energy TA. Equal LHA in (non-stochastic) expressive power.
  - **Beyond** DTMC, beyond CTMC (with multiple rewards)
- All features of UPPAAL supported
  - User defined functions and types
  - Expressions in guards, invariants, clock-rates, delay-rates (rationals), and weights.
- New GUI for plot-composing and exporting.
- Distributed SMC, 64bits.
Case Studies

FIREWIRE

BLUETOOTH

10 node LMAC

100 x 100 ROBOT

JTRES, York, September 26, 2011

Kim Larsen [72]
Conclusion & Future Work

- TA Model Checking provides highly modular, flexible and accurate timing analysis.
- Using static analysis, slicing, optimized models and verification condition yields significant performance improvements.
- Implementation of B&B for maximum cost reachability.
- Abstraction–Refinement
- Generation of concrete traces
- Use of abstract caches in model checking (UPPAAL with lattice-types).
- Support of Distributed Model Checking
- Support for 64 bit.
Thank You!