Modular Performance Analysis

Lothar Thiele
Simon Perathoner, Ernesto Wandeler
ETH Zurich, Switzerland
Embedded Systems

Computation/Communication ⊕ Resource Interaction
Models of Computation

How can we classify and compare them?

- stepwise refinement
- concurrency
- hierarchy
- incremental design
- simple
- tools
- formal
- efficient
- implementation
- beauty
- modular
- expressive
- execution

- accuracy
- compositional
- easy to use
- scope
- scalable
Why is it difficult?

• Many aspects can not be quantified.
• Models cover different scenarios:
Intention

• Compare models and methods that analyze the timing properties of distributed systems.
Approach

• Define a set of benchmark examples that cover common area (obligatory)

• Define benchmark examples that show the power of each method (free style)

• Leiden Workshop on Distributed Embedded Systems: http://www.tik.ee.ethz.ch/~leiden05/
Wait another 20 minutes ...
SymTA/S

• Based on classical RT analysis (periodic, jitter).

• Simplified relations and adaptors in order to achieve modularity.

• Computation and Communication
SymTA/S

- Based on classical RT analysis (periodic, jitter).
- Simplified relations and adaptors in order to achieve modularity.
• Models are based on Timed Automata.

periodic stream

fixed priority scheduling
Modular Performance Analysis (MPA)
Abstract Stream Model

Event Stream

number of events in
in $t=[0 .. 2.5]$ ms

Arrival Curves

maximum / minimum
arriving events in any
interval of length 2.5 ms
Load Model - Examples

- **periodic**
- **periodic w/ jitter**
- **periodic w/ burst**
- **complex**
Process Abstraction

Formal Specification
Program Analysis
Data Sheets
...

Functional Unit Automaton

- triggering event
- min/max resource demand
- produced event

Task

Cache

* b[15,15]/e
b/[15,15]/e
b/[5,5]/e
b/[3,20]/d

a/[10,10]/d
a/[10,10]/d
a/[5,5]/d
a/[5,5]/d

b/[3,20]/d

Swiss Federal Institute of Technology
Computer Engineering and Networks Laboratory
Service Model (Resources)

Resource Availability

Available service in $t=[0..2.5]$ ms

Service Curves

Maximum/minimum available service in any interval of length 2.5 ms
Service Model - Examples

- **Full resource**
  - $\beta^u$
  - $\beta^l$

- **Bounded delay**
  - $\beta^u$
  - $\beta^l$

- **TDMA resource**
  - $\beta^u$
  - $\beta^l$

- **Periodic resource**
  - $\beta^u$
  - $\beta^l$
What kind of resources can be modeled?

- Memory (buffer space)
- Delay (end-to-end delay / processing and waiting)
- Computation
- Communication
- Energy
Processing Model (HW/SW)

HW/SW Components
- Processing semantics and functionality of HW/SW tasks

Abstract Components
- $\alpha'(\Delta) = f(\alpha, \beta)$

Diagram:
- HW/SW Task
- Predicate $\Psi$
- RTC
- Variables a, b, a', t, D
Scheduling and Arbitration Components

FP/RM

\[ \beta \]

GPC

\[ \alpha_A \rightarrow \alpha_A' \]

GPC

\[ \alpha_B \rightarrow \alpha_B' \]

\[ \beta' \]

GPS

\[ \beta' \]

EDF

\[ \beta \]

GPC

\[ \alpha_A \rightarrow \alpha_A' \]

GPC

\[ \alpha_B \rightarrow \alpha_B' \]

\[ \beta' \]

TDMA

\[ \beta' \]

EDF

\[ \beta \]

GPC

\[ \alpha_A \rightarrow \alpha_A' \]

GPC

\[ \alpha_B \rightarrow \alpha_B' \]

\[ \beta' \]

RR

\[ \beta \]

GPC

\[ \alpha_A \rightarrow \alpha_A' \]

GPC

\[ \alpha_B \rightarrow \alpha_B' \]

\[ \beta' \]
What kind of resource usage can be modeled?

• Different resource sharing strategies
  – EDF
  – TDMA
  – Fixed Priority
  – GPS

• Different processing semantics
  – Greedy Processing
  – Greedy Shaper
  – Blocking
Complete System Composition

\[ \beta_{CPU} \]
\[ \alpha \]
\[ \alpha' \]

\[ \beta_{BUS} \]
\[ TDMA \]
\[ GPC \]
\[ GPC \]
\[ GPC \]
\[ GPC \]

\[ \sigma \]
\[ \beta_{DSP} \]

RM → CPU

CPU → BUS

BUS → DSP
Basic Concepts for Describing Activations

\[ \text{AND} \]
\[ \alpha_1 \rightarrow \alpha_2 \]
\[ \alpha_1 \rightarrow \nabla \]
\[ \alpha_2 \rightarrow \]
\[ \text{max} \left\{ \min \left\{ \alpha_1^u \otimes \alpha_2^l + B_1^0 - B_2^0, \alpha_2^u \right\}, \right. \]
\[ \left. \min \left\{ \alpha_2^u \otimes \alpha_1^l + B_2^0 - B_1^0, \alpha_1^u \right\} \right\} \]

\[ \text{OR} \]
\[ \alpha_1 \rightarrow \alpha_2 \]
\[ \alpha_1 \rightarrow \nabla \]
\[ \alpha_2 \rightarrow \]
\[ \alpha_1 + \alpha_2 \]
6 Real-Time Input Streams
- with jitter
- with bursts
- deadline > period

3 ECU’s with own CC’s

13 Tasks & 7 Messages
- with different WCED

2 Scheduling Policies
- Earliest Deadline First (ECU’s)
- Fixed Priority (ECU’s & CC’s)

Hierarchical Scheduling
- Static & Dynamic Polling Servers

Bus with TDMA
- 4 time slots with different lengths
  (#1,#3 for CC1, #2 for CC3, #4 for CC3)

Total Utilization:
- ECU1  59 %
- ECU2  87 %
- ECU3  67 %
- BUS   56 %
... and its Abstract Component Model

CPU

ECU1

PS

T1.1

T1.3

S1

ECU2

CPU

T4.1

T5.1

S4

S5

ECU3

CPU

PS

T2.2

T3.2

T4.2

T5.2

PS

BUS

TDMA

T1.2

C1.1

C1.2

C2.1

C3.1

C3.2

CC1

CC2

CC3

T3.3

T6.1

S2

S3

S6

ECD

ECU3
Buffer Requirements
Interface-Based Design

• MPA is suited for interface-based design
  – Stepwise refinement
  – Inverse relations because of min+ algebra
  – Assume/Guarantee by means of partial order

\[ \begin{align*}
  F & \quad \psi_F \\
  x_1 & \quad x_2 \\
  y_1 & \quad y_2 \\
  \end{align*} \]

\[ \begin{align*}
  F' & \quad \psi_F' \\
  x_1^G & \quad x_1^A \\
  x_2^G & \quad x_2^A \\
  y_1^G & \quad y_1^A \\
  y_2^G & \quad y_2^A \\
  \end{align*} \]
Intention

- Compare models and methods that analyze the timing properties of distributed systems.
Benchmarks

• Pay Burst Only Once
• Cyclic Dependencies
• Variable Feedback
• AND/OR task activation
• Intra-context information
• Workload Correlation
• Data Dependencies
Benchmark 1

• Pay Bursts Only Once

<table>
<thead>
<tr>
<th>Input stream H1</th>
<th>periodic with burst (P=10ms, J=50ms, d=1ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task WCETs</td>
<td>T1: 1ms, T2: 4ms, T3: 8ms</td>
</tr>
</tbody>
</table>

0 \leq J \leq 70ms
Benchmark 1

simulation (10000 events)

end-to-end delay [ns]

< 1s

100s
Benchmark 2

- Cyclic Dependencies

<table>
<thead>
<tr>
<th>Input stream I1</th>
<th>periodic with burst (P=10ms, J=50ms, d=0ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource sharing</td>
<td>CPU1: FP preemptive</td>
</tr>
<tr>
<td>Task WCETs</td>
<td>T1: 1ms, T2: 4ms, T3: 4ms</td>
</tr>
<tr>
<td>Scheduling param.</td>
<td>1) priority T1: high, priority T3: low</td>
</tr>
<tr>
<td></td>
<td>2) priority T1: low, priority T3: high</td>
</tr>
</tbody>
</table>

$0 \leq J \leq 50\text{ms}$
Benchmark 2-1 : T1 high

end-to-end delay [ms]

jitter T1 [ms]

- MPA-RTC
- Uppaal
- SymTA/S
- PESIMDES 90-5-5
- MAST
- PESIMDES random

T1 → T2 → T3

β₁ → β₂
Benchmark 2-2 : T3 high

- MPA-RTC
- MAST
- SymTA/S
- Uppaal
- PESIMDES 90-5-5
Benchmark 3

- Variable Feedback

\[ 4 \leq J \leq 30\text{ms} \]

| Input streams | I1: periodic (P=4ms)  
I2: periodic with burst (P=100ms, J=200ms, d=0ms) |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource sharing</td>
<td>CPU1: FP preemptive, CPU2: FP preemptive</td>
</tr>
<tr>
<td>Task WCETs</td>
<td>T1: 20ms, T2: 15ms, T3: 3ms, T4: 20ms</td>
</tr>
</tbody>
</table>
| Scheduling param. | priority T1: high, priority T2: low  
priority T3: high, priority T4: low |
Benchmark 3 : T1 high

for each step in binary search: 50 minutes
(Expected) Results

• Understand the modeling power of different models and the relation between models and analysis accuracy.

• Improve methods by combining ideas and abstractions.

• Not: competition ... .
In models for timed systems abstraction matters

Knowledge about MoCCs that (also) talk about resource usage are far less understood