Towards Probabilistic Mode Automata for Adaptable, Resource-Aware Component-Based Systems Design

Heinz W. Schmidt†, Ian Peake†, Hüsseyin Ayhan Aysan††, Sasikumar Punnekkat††, Radu Dobrin††

† School of Computer Science and Information Technology, RMIT University, Melbourne, Australia
†† MRTC, Mälardalen University, Sweden

ISSEC 2012, Melbourne, Australia
Overview

Resource-aware component-based software design:

- **design-time quality assurance**: reliability, timing, energy, etc.
- **concurrency**: (parallel/distributed software)
- **context uncertain**: blocks accurate, scalable analysis

How to model/analyse concurrent systems under uncertainty?

Cloud/grid, services, cyber-physical, nano...

Proposed approach

- separation of concerns...
- probabilistic, sequential global model: *mode+weighted* automata
- global states refined by concurrent models: *FSA-Nets*
- parameterised, generic cost analysis at mode and global levels

Contribution:

- new weighted (costed) interpretation of mode automata
- new time/performance tradeoff analysis
- concept demonstration
Motivating Example: Digital Camera

- Battery life, performance (shutter lag), robustness/reliability?
- Design time: choose components, structure, resource allocation
- Maintainability / reusability (cf product lines)
- How to model Usage?
Mode Automata (e.g. Camera Modes)

- IDLE
- Single-frame shooting mode
- High-speed multiple-frame shooting mode
- Low-speed multiple-frame shooting mode
- Multiple-frame shooting mode

Global, sequential view; mutual exclusion; hierarchical (scalability); model *dynamic reconfiguration* [Hirsch06]
Weighted Automata (Global Usage/Cost)

**Cost Model**

<table>
<thead>
<tr>
<th></th>
<th>seq</th>
<th>alt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probability</td>
<td>×</td>
<td>+</td>
</tr>
<tr>
<td>Best case Time</td>
<td>+</td>
<td>min</td>
</tr>
<tr>
<td>Worst case Time</td>
<td>+</td>
<td>max</td>
</tr>
<tr>
<td>Worst case Energy</td>
<td>+</td>
<td>max</td>
</tr>
</tbody>
</table>

### Weight states/transitions with “costs” e.g.: probability, time, energy

- Cost Model: sequence e.g. $\Pr(Idle \rightarrow HS \rightarrow Idle) = 0.1 \times 0.1 = 0.01$
- vs alternative e.g. $\Pr(Idle \rightarrow Single \rightarrow Idle) = 0.2 \times 0.1 = 0.02$

### Existing methods for e.g. steady-state costs (c.f. Markov chain)

**How to derive accurate mode costs?**

**Difficult to model concurrency (state explosion)**
Mode Refinement (e.g. HS)

*FSA-Decomposable* Petri Net (FSA-Net) [Schmidt03]:

```
HS_SHOOT HS_IMGPROC

1 do AS
2 do IB
3 do IP
4 ref AUTO
5
6 Shoot_sync
7
8
9 BF_sync
10 no Shoot
11 on Shoot
12
13
14
15
16
17
18
19
20
21

Schmidt, Peake, et al (RMIT Univ ©2012)
```
FSA-Net $\rightarrow$ Trace Set

Trace: choice-free Net (HS$\rightarrow$HS):

Weighted FSA-Nets

- Refined cost model: \textit{seq}, \textit{alternation}, \textit{parallelism}, up-to-\textit{k} repetitions

<table>
<thead>
<tr>
<th>Cost Model</th>
<th>seq</th>
<th>alt</th>
<th>par</th>
<th>( k ) rep</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best Case Time</td>
<td>+</td>
<td>min</td>
<td>max</td>
<td>0</td>
</tr>
<tr>
<td>Worst Case Time</td>
<td>+</td>
<td>max</td>
<td>max</td>
<td>( \times k )</td>
</tr>
<tr>
<td>Worst Case Energy</td>
<td>+</td>
<td>max</td>
<td>+</td>
<td>( \times k )</td>
</tr>
</tbody>
</table>

Per-configuration (e.g. hardware specs, user settings)
Per-configuration (e.g. hardware specs, user settings), per trace
Per-configuration (e.g. hardware specs, user settings), per trace:

- Per action/state: assign resource, throttle (→ local cost)
FSA-Net Cost and Resource Allocation

Per-configuration (e.g. hardware specs, user settings), per trace:
- Per action/state: assign resource, throttle (→ local cost)
- Per resource: order (unordered) actions
- Calculate cost
- Choose allocation with best cost tradeoff

<table>
<thead>
<tr>
<th>Act.</th>
<th>Abbreviates</th>
<th>GPP</th>
<th>DSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>AF</td>
<td>Autofocus</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>AE</td>
<td>Autoexposure</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>IP</td>
<td>Image processing</td>
<td>-</td>
<td>✓</td>
</tr>
<tr>
<td>IB</td>
<td>Image buffering</td>
<td>-</td>
<td>✓</td>
</tr>
<tr>
<td>IS</td>
<td>Image storage</td>
<td>✓</td>
<td>-</td>
</tr>
<tr>
<td>AS</td>
<td>Activate shutter</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>BC</td>
<td>Buffer check</td>
<td>-</td>
<td>✓</td>
</tr>
</tbody>
</table>

**Modes**
- SB  Standby | - | ✓ |
- SF  Single-frame | ✓ | ✓ |
- MF  Multi-frame | ✓ | ✓ |

**Submodes**
- FE  AF&AE | ✓ | ✓ |
- F   AF    | ✓ | ✓ |
- E   AE    | ✓ | ✓ |
- 0   -     | ✓ | ✓ |
Fix static configuration

Choose weights via e.g. *mode refinement FSA-Net analysis (cost)*, vs observation (probability)

*Analyse at global level* using e.g. probabilistic model checker (PRISM) where applicable

(Note separation: mode refinement vs global cost analysis)

Refine modes for accuracy
Valient, Yusuf: assumption about abstraction over concurrency

SHARP (Medvidovic, genericity, Nets); Palladio (Reussner); HMSCs

Scalability, compositionality, complexity at model intersection...
  ...E.g. Allocation/scheduling
  ...E.g. is HS → HS constant-bounded, or probability, or both?

Possible restrictions on models, cost models, analysis

Framework for extensions/research: refinement, cost-aware scheduling

Benchmarks on more systems (c.f. Yusuf, Grid FT)
Conclusion

- Propose weighted/probabilistic interpretation of mode automata
- Propose new time/performance cost tradeoff analysis method
- Extends our Rich Architecture Description Language (RADL)
- Combine previously distinct RADL methods:
  - probabilistic global sequential behaviour; reliability
  - concurrent behaviour; timing/energy


## Allocation tradeoffs

<table>
<thead>
<tr>
<th>RAW</th>
<th>Size</th>
<th>AUTO</th>
<th>WCT</th>
<th>WCE</th>
<th>AS</th>
<th>AUTO</th>
<th>IB</th>
<th>IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>4</td>
<td>-</td>
<td>17</td>
<td>51</td>
<td>D,50</td>
<td>G,100</td>
<td>D,100</td>
<td>D,50</td>
</tr>
<tr>
<td>Y</td>
<td>4</td>
<td>-</td>
<td>27</td>
<td>38</td>
<td>D,50</td>
<td>G,100</td>
<td>D,50</td>
<td>D,50</td>
</tr>
<tr>
<td>Y</td>
<td>10</td>
<td>-</td>
<td>32</td>
<td>90</td>
<td>G,100</td>
<td>G,50</td>
<td>D,100</td>
<td>D,50</td>
</tr>
<tr>
<td>Y</td>
<td>10</td>
<td>-</td>
<td>57</td>
<td>58</td>
<td>G,100</td>
<td>G,100</td>
<td>D,50</td>
<td>D,50</td>
</tr>
<tr>
<td>-</td>
<td>4</td>
<td>-</td>
<td>64</td>
<td>175</td>
<td>G,50</td>
<td>G,50</td>
<td>D,100</td>
<td>D,100</td>
</tr>
<tr>
<td>-</td>
<td>4</td>
<td>-</td>
<td>74</td>
<td>162</td>
<td>G,50</td>
<td>G,50</td>
<td>D,50</td>
<td>D,100</td>
</tr>
<tr>
<td>Y</td>
<td>4</td>
<td>Y</td>
<td>74</td>
<td>210</td>
<td>G,50</td>
<td>D,100</td>
<td>D,100</td>
<td>D,50</td>
</tr>
<tr>
<td>-</td>
<td>4</td>
<td>Y</td>
<td>236</td>
<td>179</td>
<td>G,50</td>
<td>D,50</td>
<td>D,50</td>
<td>D,50</td>
</tr>
<tr>
<td>-</td>
<td>10</td>
<td>Y</td>
<td>265</td>
<td>479</td>
<td>G,100</td>
<td>D,50</td>
<td>D,100</td>
<td>D,100</td>
</tr>
<tr>
<td>-</td>
<td>10</td>
<td>-</td>
<td>271</td>
<td>245</td>
<td>D,100</td>
<td>G,50</td>
<td>D,100</td>
<td>D,50</td>
</tr>
<tr>
<td>-</td>
<td>10</td>
<td>Y</td>
<td>271</td>
<td>376</td>
<td>G,100</td>
<td>G,50</td>
<td>D,100</td>
<td>D,50</td>
</tr>
<tr>
<td>-</td>
<td>10</td>
<td>-</td>
<td>296</td>
<td>213</td>
<td>D,100</td>
<td>G,100</td>
<td>D,50</td>
<td>D,50</td>
</tr>
<tr>
<td>-</td>
<td>10</td>
<td>Y</td>
<td>296</td>
<td>344</td>
<td>G,100</td>
<td>G,50</td>
<td>D,50</td>
<td>D,50</td>
</tr>
<tr>
<td>-</td>
<td>10</td>
<td>Y</td>
<td>385</td>
<td>324</td>
<td>G,100</td>
<td>D,50</td>
<td>D,100</td>
<td>D,50</td>
</tr>
<tr>
<td>-</td>
<td>10</td>
<td>Y</td>
<td>410</td>
<td>292</td>
<td>G,100</td>
<td>D,50</td>
<td>D,50</td>
<td>D,50</td>
</tr>
</tbody>
</table>