Verifying the Adaptation Behavior of Embedded Systems

Klaus Schneider\textsuperscript{1} \quad Tobias Schuele\textsuperscript{1} \quad Mario Trapp\textsuperscript{2}

\textsuperscript{1} Reactive Systems Group, University of Kaiserslautern
Gottlieb-Daimler-Straße 48, 67663 Kaiserslautern, Germany

\textsuperscript{2} Fraunhofer Institute for Experimental Software Engineering (IESE)
Fraunhofer-Platz 1, 67663 Kaiserslautern, Germany

SEAMS 2006
May 21-22, Shanghai, China
Outline

1. Introduction
2. Modeling Adaptation Behavior
3. Verifying Adaptation Behavior
4. Tool Demonstration
5. Summary and Conclusion
### Improve Quality and Functionality

- changing environment (car enters a tunnel, aquaplaning)
- reliability and dependability (get safely to next garage)
- personalization for specific needs (different drivers)
Adaptation in Embedded Systems

Improve Quality and Functionality

- changing environment (car enters a tunnel, aquaplaning)
- reliability and dependability (get safely to next garage)
- personalization for specific needs (different drivers)

Reduce Costs

- concurrent systems that consist of several parts
- depending on situation not all parts active at the same time
- dynamically adapt according to currently required needs
- share parts that are not used simultaneously
Adaptation in Embedded Systems

Challenges

- embedded systems are reactive real–time systems
  - verification of functional and temporal behavior
- hybrid systems (interacting analog and digital parts)
  - verification requires abstraction to discrete domains
- safety–critical systems (aviation, automotive industry)
  - legal aspects (“it wasn’t me who pushed the brakes”)

Klaus Schneider, Tobias Schuele, Mario Trapp
Verifying the Adaptation Behavior of Embedded Systems
Adaptation in Embedded Systems

Challenges

- embedded systems are reactive real–time systems
  - verification of functional and temporal behavior
- hybrid systems (interacting analog and digital parts)
  - verification requires abstraction to discrete domains
- safety–critical systems (aviation, automotive industry)
  - legal aspects (“it wasn’t me who pushed the brakes”)

What about adaptation?

- adaptation has become increasingly complex part
- may trigger further adaptations in other components
- chain reaction of adaptations (up to 80% affected)
- can cause inconsistent and unstable configurations
- verification of adaptation behavior is a crucial concern
Modeling Adaptation Behavior

Services and Quality Descriptions

- Configuration
  - Parameters
  - Influences

- Configuration
  - Parameters
  - Influences

Type

- Mode
  - Quality

- Mode
  - Quality
Modeling Adaptation Behavior

Services and Quality Descriptions

Configuration
- Parameters
- Influences

Type
- Mode
- Quality

Variable
- Mode
- Quality

Configuration Rules

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Priority</th>
<th>Guard</th>
</tr>
</thead>
<tbody>
<tr>
<td>OccupancyDetection</td>
<td>4</td>
<td>cameramage[available]</td>
</tr>
<tr>
<td>TransponderDetection</td>
<td>3</td>
<td>transponderID[available]</td>
</tr>
<tr>
<td>MotionDetection</td>
<td>2</td>
<td>motion[available(r_point&gt;0)]</td>
</tr>
<tr>
<td>Off</td>
<td>1</td>
<td>true</td>
</tr>
</tbody>
</table>
Modeling Adaptation Behavior

Example

Service
Occupancy-Detection
OccupancyDetection

Service
Light-Control
AdjustLight
tOff := 0;

Service
Lamp
DimLights

MotionDetection

occupancy->motion
 omissionrate = 0

dimmerValue->unavailable

SwitchLights

tOff := 5;
Synchronous Languages (Quartz)

- precise notion of concurrency, communication, and time
- detailed formal semantics (structural operational semantics)
- specifications: temporal logics $\Rightarrow$ symbolic model checking
Verifying Adaptation Behavior

Synchronous Languages (Quartz)

- precise notion of concurrency, communication, and time
- detailed formal semantics (structural operational semantics)
- specifications: temporal logics $\Rightarrow$ symbolic model checking

Example

```plaintext
module ABRO:
    input a,b,r: event;
    output o: event;
    loop
        [await a || await b];
        emit o;
        each r;
    spec
        safe: A G (o -> a | b);
    end
```
Tool Demonstration

Introduction
Modeling Adaptation Behavior
Verifying Adaptation Behavior
Tool Demonstration
Summary and Conclusion

Klaus Schneider, Tobias Schuele, Mario Trapp

Verifying the Adaptation Behavior of Embedded Systems
### Adaptation in Embedded Systems

- react on changes in the environment (failure of sensors)
- reduce costs and increase dependability (graceful degradation)
- can cause chain reaction of adaptations in other components
Summary and Conclusion

### Adaptation in Embedded Systems
- react on changes in the environment (failure of sensors)
- reduce costs and increase dependability (graceful degradation)
- can cause chain reaction of adaptations in other components

### Modeling and Verification
- modeling adaptation behavior at an abstract level
- augmenting data flow with quality descriptions
- configuration rules to describe potential adaptations
- translation to synch. languages $\Rightarrow$ symbolic model checking
  - can a certain configuration be reached at all?
  - can a system be caught in such a configuration?
  - can a certain configuration be reached infinitely often?
  - how long will it take to complete an adaptation?