A Formal Co-Simulation Approach for Wireless Sensor Network Development

Adisak Intana, Michael R. Poppleton, Geoff V. Merrett
{ai1n10,mrp,gvm}@ecs.soton.ac.uk
Electronics and Computer Science

AVoCS14
25 September 2014
Research Challenges of WSN Development

- Software engineering is needed by WSN development [pic10].
- Traditional “Code-and-Fix” implements code on the real node under the constraints of display-less, low-level specific platform and low-power design [pic10].
Research Challenges of WSN Development

- “Simulation” provides the higher level of abstraction [KM07, ISH10]. - but unexplored scenarios may still contain flaws [IPM13].

- “Formal Methods” give the strong verification - but require long running simulation with more realistic environments for reliability and performance analysis.
Contribution

• **Goal of this research:** “to construct the infrastructure for co-simulation which provides an integrated set of methodologies for WSNs.”
Outline

- Vision of co-simulation approach for WSNs
- Case Study
- S-style vs F-style Modelling
- C-style co-modelling
- Conclusion and Future Work
Vision

** extended from [PM12]
This work focuses on the network level of development.
Vision
Vision
Vision
Case Study

- **SensorScope** project, an environmental monitoring system [BIS+08]
S-Style Modelling

- Simulation tool: MiXiM based on OmNet++
Strength and Weakness of S-Style

- **Strength:**
  - Network algorithms can be analysed with the performance evaluation.

- **Weakness:**
  - Absence of faults such as Loop problems cannot be guaranteed.
  - Complexity in development
    - the controller representing each specific protocol has to be completed together with environment elements at the same time.
    - All functionalities of each protocol algorithm have to be implemented into one single model.
F-Style Modelling

- WSN development in Event-B:
  - Each WSN protocol algorithm is layered by through refinement steps.
F-Style Modelling

- shared event decomposition is used to separate software controller from environment [But09, SPHB11].
F-Style Modelling (2)

- Model Verification:
  - Route tree safety properties [IPM13] :

```
@inv3_1 nodes ⊆ ND
@inv3_2 cRouteTree ∈ (nodes \ {Sink}) → nodes
@inv3_3 completedRoute = TRUE ⇒ nodes = ND
@saf3_1 ∀ S · S ⊆ cRouteTree⁻¹[S] ⇒ S = ∅

theorem @mth3_1 ∀ T(Sink∈T ∧ cRouteTree⁻¹[S] ⊆ T ⇒ nodes ⊆ T)
theorem @mth3_2 nodes ⊆ {Sink} ∪ (tcl(cRouteTree))⁻¹{{Sink}}
theorem @mth3_3 nodes \ {Sink} ⊆ (tcl(cRouteTree))⁻¹{{Sink}}
```

- Proof Obligation (POs) verification: 330 Pos were discharged by RODIN, 85% of these were proved automatically.
F-Style Modelling (3)

- Model Animation and Validation:
  - ProB were used to animate the develop formal model with testing scenarios affecting the different link quality ratios.
Strength and Weakness of F-Style

- **Strength:**
  - Strong guarantee for the absence of faults.

- **Weakness:**
  - Long running simulation is needed to give full confidence about reliability and performance requirements.
C-Style Co-modelling

- *FoCosim-WSN* framework for WSNs
Co-Simulation Case Study Modelling

- Co-simulation models

(a) SensorApp co-models

(b) MintRoute co-models
Engineering Process for C-Style Development

- Starting From F-style development
Engineering Process for C-Style Development

- Starting From S-style development
# Conclusion

<table>
<thead>
<tr>
<th></th>
<th>S-Style</th>
<th>F-Style</th>
<th>C-Style (FoCoSim-WSN)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proof of absence of certain classes of faults e.g. safety properties</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Validation &amp; Testing Techniques</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Performance Analysis &amp; Evaluation via (co-)simulation</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
</tr>
</tbody>
</table>
Conclusion

- *FoCoSim-WSN* co-simulation provides:
  - Strong formal verification in the controller.
  - Performance evaluation with stochastic environment from simulation engine.
  - Complexity reduction in development through multiple refinement levels.
  - Flexibility to integrate between F-, S- and C-Style modelling.
Future Work

- Fault Injection including node failure, unreliable connection scenarios will be considered.

- “Killer” traces will be sought to validate formal Event-B models.

- Dense network models to explore the network congestion problems will be exercised.
Open Research

- Code generation experiments from node controller for the real node.

- The node level co-simulation development and the extension for multi-co-simulation.
Thank You!

and Any Question?
References


References (2)


