The Project: Wireless Fire Alarm System

Main functionality:

• Veri (display non-operational sensors at central unit).
• alarm notification

Timing constraints:

• regulated

Self-Monitoring: Sensor Requirements

Option 1

• teach DC (usually not economic).
• serve as translator / mediator.

Option 2

• FM expert
• electr./comm.
• compares expected outcome and real outcome.

Formal Methods: Requirements Validation Cont'd

Verification:

• Formal Methods in the Development Process

Model Decomposition, Resource Consumption

Looking Back (and Forward: Exam)

Lecture 21: Code Generation
From DC Frame to Generic Deployment

<table>
<thead>
<tr>
<th>Queries</th>
<th>seconds</th>
<th>MB</th>
<th>States explored</th>
<th>Detection possible</th>
</tr>
</thead>
<tbody>
<tr>
<td>E&lt;&gt; switcher.DETECTION</td>
<td>10,205.13</td>
<td>557.00</td>
<td>26,445,788</td>
<td>No message collision</td>
</tr>
<tr>
<td>A[] not deadlock</td>
<td>38.21</td>
<td>55.67</td>
<td>1,250,596</td>
<td>Detect</td>
</tr>
<tr>
<td>A[] (switcher.DETECTION imply switcher.timer &lt;= 300*Second)</td>
<td>38.21</td>
<td>55.67</td>
<td>1,250,596</td>
<td>Detect</td>
</tr>
<tr>
<td>A[] !center.ERROR</td>
<td>38.21</td>
<td>55.67</td>
<td>1,250,596</td>
<td>Detect</td>
</tr>
</tbody>
</table>

Repetters as slaves, N = 10.

<table>
<thead>
<tr>
<th>Queries</th>
<th>seconds</th>
<th>MB</th>
<th>States explored</th>
<th>Detection possible</th>
</tr>
</thead>
<tbody>
<tr>
<td>E&lt;&gt; switcher.DETECTION</td>
<td>368.58</td>
<td>250.91</td>
<td>9,600,062</td>
<td>No message collision</td>
</tr>
<tr>
<td>A[] not deadlock</td>
<td>231.84</td>
<td>230.59</td>
<td>6,009,120</td>
<td>Detect</td>
</tr>
<tr>
<td>A[] (switcher.DETECTION imply switcher.timer &lt;= 300*Second)</td>
<td>3.94</td>
<td>10.14</td>
<td>144,613</td>
<td>NoSpur</td>
</tr>
<tr>
<td>A[] !center.ERROR</td>
<td>3.94</td>
<td>10.14</td>
<td>144,613</td>
<td>NoSpur</td>
</tr>
</tbody>
</table>

(Opteron 6174 2.2Ghz, 64GB, U/P.sc/P.sc/A.sc/A.sc/L.sc 4.1.3 (64-bit), options -s -t0 -u)
Models and Corresponding Sizes

<table>
<thead>
<tr>
<th>Location</th>
<th>Clocks</th>
<th>Sensors as slaves</th>
<th>Repeaters as slaves</th>
</tr>
</thead>
<tbody>
<tr>
<td>Self-Monitoring</td>
<td>9 137 1040 6</td>
<td>6 16 101 16</td>
<td>5 16 108 12</td>
</tr>
<tr>
<td>Alarm</td>
<td>6 16 101 16</td>
<td>6 25 200 15</td>
<td></td>
</tr>
</tbody>
</table>

From DC Formulae to Queries: Alarm

- Queries:
  - A[[]] !Center.ALARMED imply time < 10*Second: " exactly one alarm displayed within 10 s"
  - A[[]] (!Sensor0.DONE || !Sensor1.DONE) imply time <= 10*Second: " exactly two (simultaneous) alarms displayed within 10 s"
  - A[[]] (!Sensor0.DONE || !Sensor1.DONE || ... || !Sensor9.DONE) imply time <= 100*Second: " exactly ten (simultaneous) alarms displayed within 100 s"

Verification Results: Alarm

<table>
<thead>
<tr>
<th>Query</th>
<th>MB</th>
<th>States expl.</th>
</tr>
</thead>
<tbody>
<tr>
<td>A[[]] !Center.ALARMED imply time &lt; 10*Second</td>
<td>38.±1</td>
<td>14.±1</td>
</tr>
<tr>
<td>A[[]] (!Sensor0.DONE</td>
<td></td>
<td>!Sensor1.DONE) imply time &lt;= 10*Second</td>
</tr>
<tr>
<td>A[[]] (!Sensor0.DONE</td>
<td></td>
<td>!Sensor1.DONE</td>
</tr>
</tbody>
</table>

Testing the Real System

<table>
<thead>
<tr>
<th>Test scenario</th>
<th>Measured</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Alarm</td>
<td>2.14 s</td>
<td>2.79 s ± 0.53 s</td>
</tr>
<tr>
<td>All 10 Alarms</td>
<td>27.08 s</td>
<td>29.65 s ± 3.26 s</td>
</tr>
</tbody>
</table>
Verifying "a whole system design" (i.e., every bit and detail of: car, plane, even WFAS) can be very expensive, gaining confidence into "the core design ideas" (or crucial aspects of the design) can be much more feasible.

One approach:
- fix a budget (time, effort, ...),
- identify and formalise core requirements (balance priority and budget),
- validate using positive / negative examples,
- model as far as possible, on an appropriate level of abstraction (balance level of detail and budget),
- validate using simulation of example runs,
- verify as far as possible (if infeasible: limit considered scenarios, at least simulate).

Other way round:
- fix the goal of the formal analysis.

In my opinion, everybody in this room (or on the "broadcast receiver" at home) has been exposed to all the knowledge and experience that it takes to do the WFAS project. What's your opinion?
Looking Back (and Forward: Examination)

Lecture 21: Code Generation

Model Decomposition, Resource Consumption

Verification

Conclusion

Characterizing "Dependency on Global Scheduler"

Another Example:

The Rendezvous Transition Rule may Block Senders

Example:

The Rendezvous Transition Rule may Block Senders

Code Generation from TA in the Literature

One Last Thing

•

trifft gar nicht zu

• trifft voll zu

40%

20%

30%

10%

0%

0%

6

I have improved my capabilities in scientific problem solving.

(Ich habe meine Fähigkeiten im wissenschaftlichen Problemlösen verbessert.)

• task (in own words), (2) solution (in full sentences), (3) correctness argument. That's already "half of the story" ;-)