Graphical simulation of the execution of DSL models

Kick-off presentation
by Rimco Boudewijns
Outline

• Background Information
• Problem Statement
• Related Work
  • Formal Verification Tools
  • Visualization Tools
• Approach

Disclaimer: Some slides have been changed or removed to protect confidential information.
Background Information

Project Setup

- User-oriented Project
- Project start: April 1\textsuperscript{st}
- Project end: October 1\textsuperscript{st}

- External company: ASML

- Supervisor: Tom Verhoeff
- Tutor: Ulyana Tikhonova
Background Information

**DSL Example**

### Action 1

1. **a**
   - **b**
     - **c**

### Action 2

2. **b**
Background Information

**DSL Structure (partial)**
Problem Statement
General DSL Problem

• High level of abstraction

• Debugging and impact analysis is hard
Problem Statement

**DSL User Problems**

- How is my model executed by the generated code?
- What is the impact of a certain change in my model?
Problem Statement

COREF Framework as Foundation
Related Work

Outline

- Formal Verification Tools
  
<table>
<thead>
<tr>
<th>Modeling Language</th>
<th>Formal Verification Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statemate</td>
<td>Statemate ModelChecker</td>
</tr>
<tr>
<td>Simulink/Stateflow</td>
<td>Reactis, EmbeddedValidator, Simulink DesignVerifier</td>
</tr>
<tr>
<td>SCADE</td>
<td>SCADE Design Verifier</td>
</tr>
<tr>
<td>UPPAAL</td>
<td>UPPAAL</td>
</tr>
<tr>
<td>Event-B</td>
<td>Rodin Framework + Plug-ins</td>
</tr>
</tbody>
</table>

- Visualization Tools
Related Work

Formal Verification Tools (Statemate)

- Hides the underlying verification technology
- Robustness checks
- Reachability analysis
- Graphical editor
- Graphical Simulation
- Automotive
Related Work

Formal Verification Tools (Simulink)

- Block diagram environment
- Graphical editor
- External mode simulation
- Scopes and data displays for viewing simulation results
- Electrical
Related Work

*Formal Verification Tools (SCADE)*

- Model diagram environment
- Model checking
- Formal verification
- Graphical editor
- Debugging and graphical simulation
- Automotive
Related Work

Formal Verification Tools (UPPAAL)

- Guarded command language
- Model checking
- Graphical editor
- Graphical Simulator
Related Work
Formal Verification Tools (Event-B)

- Set theory as a modeling notation
- Refinement
- Mathematical proving system
- Eclipse-based IDE
- Plug-ins
  - Composition
  - Model checking
  - Graphical Simulator
Summary

Formal Verification Tools

• Many formal verification tools are proprietary and focus on automotive or electrical industry
• Event-B is more general and can be integrated better
• Event-B is already used with the COREF Framework
Related Work

Outline

• Formal Verification Tools
• Visualization Tools for Event-B
  • B2EXPRESS
  • Brama
  • B-Motion Studio
Related Work

Visualization Tools for Event-B (B2EXPRESS)

• Translates to the EXPRESS Data Modeling Language
• Uses traces of events of the model
• Models are checked using a set of traces
• Offers the ability to run scenarios not conform to the used model
• Specific input values must be given manually
• Accepts external traces as input
Related Work

Visualization Tools for Event-B (Brama)

- Graphical animation of the transition system
- Uses Macromedia Flash for visualization
- Incorporates a scheduler mechanism
Related Work

Visualization Tools for Event-B (B-Motion Studio)

• Only finite set of values
• States can be visualized
• Includes a graphical editor
• Uses Event-B notation
• Controls and Observers
Summary
Visualization Tools for Event-B

- B2EXPRESS needs an intermediate language
- Brama needs external software
- B-Motion Studio is better integrated in Rodin
- Proof of concept in B-Motion Studio exists
Approach

Desired Result

• **Model-to-Model transformation**
  • Input:
    - DSL model
    - (Optional) layout settings
  • Output:
    - B-Motion Studio visualization
Approach

Plan

• Research related work
• Design transformation foundations (Layout etc.)
• Iteratively:
  • Manual DSL to B-Motion Studio translation
  • Check translation with end-users
  • Create model transformation (for a restricted input set)
• Generalize for other DSLs