An example-based study of the verification of model transformations

Hans Vangheluwe, Levi Lucio, Joachim Denil
Presentation Overview

• Background concepts
• The PowerWindow case study
• More background – verifying model transformations
• Analysis of the transformations in the PowerWindow case study
• Future Work
Presentation Overview

• **Background concepts**
• The PowerWindow case study
• More background – verifying model transformations
• Analysis of the transformations in the PowerWindow case study
• Future Work
Model-Driven Development (MDD)

... is a technique that aims at automating software (and systems) development. It is based on:

- **Models**, do express requirements, architecture, design, tests, code...

- **Transformations**, to explicitly express how models are related to each other and to automate certain software engineering steps.
Models and Metamodels

statechart metamodel

statechart model

Conforms To
Model Transformations

a statechart model

a Petri net model
Domain Specific Modeling using Domain Specific Languages (DSLs)

general Purpose?
Multi-Paradigm Modeling (MPM)

• Use the right formalism(s) to model at the right level(s) of abstraction;

• Enablers:
  – modelling language engineering
  – model transformation

• Model everything (explicitly)!
Verification and Simulation

• **Verification** consists of checking/proving, under certain assumptions, whether a property of interest holds for a model;

• **Simulation** consists of producing a behavior (trace) of a model (given initial conditions and parameters).
Presentation Overview

• Background concepts
• **The PowerWindow case study**
• More background – verifying model transformations
• Analysis of the transformations in the PowerWindow case study
• Future Work
The PowerWindow Case Study

An example of developing software to control window movement using an MDD approach
Domain-Specific PowerWindow Models

- Powerwindow environment (Environment DSL)
- Powerwindow control (Control DSL)
- Powerwindow plant (Environment DSL)
- Powerwindow network (Network DSL)
Plant metamodel (DSL)
PowerWindow Plant model
Control metamodel (DSL)
PowerWindow control
PowerWindow “environment”
Network metamodel (DSL)
PowerWindow network
S5. Automatic Reversal Systems

A power-operated window, partition, or roof panel system that is capable of closing or of being closed under any circumstances other than those specified in S4 shall meet the requirements of S5.1, S5.2, and, if applicable, S5.3.

S5.1 While closing, the power-operated window, partition, or roof panel shall stop and reverse direction either before contacting a test rod with the properties described in S8.2 or S8.3, or before exerting a squeezing force of 100 newtons (N) or more on a semi-rigid cylindrical test rod with the properties described in S8.1, when such test rod is placed through the window, partition, or roof panel opening at any location in the manner described in the applicable test under S7.

S5.2 Upon reversal, the power-operated window, partition, or roof panel system must open to one of the following positions, at the manufacturer’s option:

(a) A position that is at least as open as the position at the time closing was initiated;

(b) A position that is not less than 125 millimetres (mm) more open than the position at the time the window, partition, or roof panel reversed direction; or

(c) A position that permits a semi-rigid cylindrical rod that is 200 mm in diameter to be placed through the opening at the same location as the rod described in S7.1 or S7.2(b).

S5.3 If a vehicle uses proximity detection by infrared reflection to stop and reverse a power-operated window, partition, or roof panel, the infrared source shall project infrared light at a wavelength of not less than 850 nm and not more than 1 050 nm. The system shall meet the requirements in S5.1 and S5.2 in all ambient light conditions from total darkness to 64 500 lux (6 000 foot-candles) incandescent light intensity.
R1. Window must be fully opened or closed within 5[s].

R2. The motor must shut off after 5[s] of continuous movement in any direction, as a fail safe protection for the window mechanism, motor, and drive.

R3. The window must start moving no later than 0.2[s] after the command is issued.

R4. The window must stop when it reaches a fully opened or fully closed position.

R6. If the up or down command is issued for a duration of 0.2[s] - 1[s], the window must be fully opened or closed, unless interrupted by a new window command or an obstacle. This requirement represents the automatic-up and automatic-down capability of the power window.

R7. The window must be able to detect an obstacle with a force less than 100[N].

R8. The window must be lowered by approximately 10[cm] if an obstacle is detected.

R9. Obstacle detection has priority over both driver-side and passenger-side controls.

R10. Driver-side controls have priority over passenger-side controls.
The PowerWindow Case Study
Verification using Petri Nets
Required Transformations

- Powerwindow environment (Environment DSL)
- Powerwindow control (Control DSL)
- Powerwindow plant (Environment DSL)
- Powerwindow network (Network DSL)

- Powerwindow environment (Petri Net)
- Powerwindow control (Petri Net)
- Powerwindow plant (Petri Net)
- Powerwindow Petri Net network (Network DSL)

Composed powerwindow (Petri Net)
Example transformation: PowerWindow plant into Petri nets
Example transformation: (PowerWindow DSL level) network into (Petri net level) network
Example transformation: PowerWindow Petri net composition
PowerWindow Case Study

• Quantitative simulations/optimizations (taking into account power effects)

• Calibration
The PowerWindow Case Study Simulation using Causal Block Diagrams (CBDs) (aka Synchronous Dataflow or Simulink)
Required Transformations

- Powerwindow environment (Environment DSL)
  - Powerwindow environment (CBD)
  - Powerwindow control (Control DSL)
  - Powerwindow plant (Environment DSL)

- Powerwindow network (Network DSL)
  - Powerwindow CBD network (Network DSL)

- Composed powerwindow (CBD)
Example transformation: PowerWindow plant into CBD
Example transformation: PowerWindow CBD composed model
The PowerWindow Case Study
Code Generation
The powerwindow Case Study

Code Generation
Presentation Overview

• Background concepts
• The PowerWindow case study
• More background – verifying model transformations
• Analysis of the transformations in the PowerWindow case study
• Future Work
Verifying Model Transformations
A Taxonomy

Collaboration with: M. Amrani, Y. LeTraon (U. Luxembourg, Luxembourg)
G. Selim, J. Dingel (Queens U., Canada)
B. Combemale (U. Rennes / IRISA, France)
Transformations
Form and Intention

• Heterogeneity (endogenous vs exogenous)
• Abstraction (add/remove/same level of, detail)
• Model Conservation (inplace vs. outplace)
• Input/Output model arity
Properties of a Transformation

• Properties related to the transformation’s computation:
  – Termination
  – Confluence

• Properties related to the transformation’s input (and expected) output models:
  – Model syntax relations
  – Model semantics relations
Verification Techniques

• Transformation-independent and input-independent
  – “paper-pencil” proofs
  – computer assisted proofs;

• Transformation-dependent and input-independent
  – computer assisted proofs (e.g. PVS, KIV)
  – model checking (e.g. SPIN, GROOVE)

• Transformation-dependent and input-dependent
  – “lightweight” formal methods (e.g. Alloy)
  – post-condition checking
  – transform the transformation into verifiable formalisms (e.g. Petri Nets)
  – model checking
Presentation Overview

• Background concepts
• The Powerwindow case study
• More background – verifying model transformations
• **Analysis of the transformations in the PowerWindow case study**
• Future Work
Example: powerwindow Petri Net composition transformation
Example: powerwindow Petri Net composition transformation
Example: powerwindow Petri Net composition transformation
Example: powerwindow Petri Net composition transformation
Powerwindow composition transformation: properties to verify

- Structural preservation of the input Petri Nets;
- Reachability is preserved in each of the preserved component nets;
- All structurally preserved component nets remain 1-safe;
- Network links are transformed into the right composition patterns at Petri net level.
Presentation Overview

• Background concepts
• The Powerwindow case study
• More background – verifying model transformations
• Analysis of the transformations in the Powerwindow case study
• Future Work
Future Work

• Include Requirements/Property languages!
• Include transformations for testing!
• Refine/validate PowerWindow transformation requirements (classification)
• Implement PowerWindow transformations
• Transformation verification techniques taxonomy
• Relate techniques to problems (based on our analysis experiences)
• Include fault tolerance?
• More representative case study (ACC)?