Model-Level Unit Testing of UML-RT Models

Reza Ahmadi

Supervised By: Dr. Juergen Dingel

May 2017, Montreal
Our Model-Level Unit Testing Framework

• **JUnit for UML-RT Capsules**
  - Automatically Generating Test Inputs/Test Verdicts
  - Capsules can be Composite
  - Using Symbolic Execution

• **Targeted Test Case Generation by Exploring the State Machine Partially**
  - *Test Purposes: testing part of the state machine.*
  - *Safety Properties: checking properties on the state machine*
  - *Test Goals: simulating slight changes on the state machine and see its consequence*

• **Test Case Execution**
  - To execute generated test cases on the model level
Our Model-Level Unit Testing Framework

Fig 0. “CruiseController” Capsule
Our Model-Level Unit Testing Framework

Fig 0. “CruiseController” Capsule

Capsule Under Test (CUT)
Our Model-Level Unit Testing Framework

Fig 0. “CruiseController” Capsule

“CruiseSafety” Property

Capsule Under Test (CUT)

May 2017, Montreal
Our Model-Level Unit Testing Framework

Fig 0. “CruiseController” Capsule

Property to Check on the CUT

“CruiseSafety” Property

Capsule Under Test (CUT)
Example: A “Cruise Control System” components:

- Sensor Scan
- Input Speed
- Throttle
- Cruise Controller & Speed Control (Control capsule in the model below)
Specifying a Safety Property for a “Cruise Control System”

Example: A “Cruise Control System” components:

- Sensor Scan
- Input Speed
- Throttle
- Cruise Controller & Speed Control (Control capsule in the model below)

Fig 1. “Cruise Control System” model in UML-RT, Top Capsule
Specifying a Safety Property for a “Cruise Control System”

Example: A “Cruise Control System” components:

- Sensor Scan
- Input Speed
- Throttle
- Cruise Controller & Speed Control (Control capsule in the model below)

Fig 1. “Cruise Control System” model in UML-RT, Top Capsule
An Important Check!

In this system how to make sure that:

Once the Cruise Control system is enabled by “on” or “resume” signals, then pressing “off”, “brake” or “acc” will indeed disable the Cruise Control?!
Specifying a Safety Property for a “Cruise Control System”

Fig 2. “Cruise Safety” Safety Property modelled in UML-RT using Protocol State Machines (PSM)
Test Goals

Fig 3. Parcel Router Problem [1]
Test Goals

Note:
What happens if generator generates with some delay? Will we lose any parcel?

Fig 4. Top Capsule of a Parcel Router Model [Model by Nicolas Hili, Queen’s]
Our solution:
Specify a test goal, which “simulates” that extra delay, and we generate a set of tests to “exhibit” the effect of that extra delay.

Fig 4. Top Capsule of a Parcel Router Model [Model by Nicolas Hili, Queen’s]
Future, Ongoing Work, and Challenges

• **Symbolic Execution (Ongoing)**
  • For Test Input/Outputs (not verdicts)
  • We have customized an existing SE, that now works in Papyrus-RT
  • I Need to extend our Symbolic Execution tool to support more advanced scenarios in terms of supported action code and model features (Future Work)

• **Test Selection and Execution (Ongoing)**
  • Various Eclipse plugins
  • Works with Papyrus-RT models

• **TraceCompass (Future use)**
  • For visualizing test execution traces
Thanks for your attention! Any questions?