

## Model Driven Engineering of Embedded Software (MDE)

### Homework (second part)

Relying on the previously defined TCL language (cf. first part), we will focus in this second part on Validation & Verification (V&V) of TCL models. To provide these features, we propose to implement a model simulator (as a model interpreter) and a code generator (i.e., a model compiler) to a formalism supported by a model-checker.

#### Task 1: Simulator for TCL

We propose to define an operational semantics for TCL and a model simulator like an interpreter.

- 1) Propose an architecture using AOM to implement the simulator
- 2) Describe an operational semantics of TCL with Kermeta (non-determinism will be dealt by a random choice of transition or an interaction with the end-user)
- 3) [optional] Implement a simulator with Kermeta

#### Task 2: Model-checker for TCL

We propose to define a translational semantics for TCL, implemented like a compiler to a formal domain supported by a model-checker. In this homework, we propose to define a compiler from TCL to the Petri net formalism<sup>1</sup>, and then to use the Tina toolkit<sup>2</sup> that provides a model-checker.

- 1) Define a mapping to translate TCL models into Petri nets
- 2) According to this mapping, give the Petri net that correspond to the TCL model of the car indicators, and propose the following LTL<sup>3</sup> properties:
  - a. *An active light flashes according to the clock,*
  - b. *[optional] When the warning is active, the right and left lights flash simultaneously.*
- 3) [optional] Implement a code generator with Kermeta, and use the Tina toolkit to formally check the property by model checking.



*Note that both the interpreter and the compiler implement the semantics of TCL. Similarly, other code generators could be defined for specific target execution platforms and also implement the semantics of TCL. Verify their equivalence is an important issue. Could you suggest the ways to achieve this verification?*

**Reminder:** this homework will be realized by group of two students. Each part will be evaluated by an oral (10min per group).

**Important:** during the evaluation of the second part, a different student will present each task.



- **21st October 2009:** oral evaluation of the first part (16:15-17:15) and a lecture (17:15-18:15) on "Formal Foundations of MDE for Model V&V" (giving more details for the second part of the homework)
- **4th November 2009 (16:15-18:15):** end of the lecture on "Formal Foundations of MDE for Model V&V"
- **18th November 2009:** oral evaluation of the second part (16:15-17:15) and quiz (17:15-18:15)

<sup>1</sup> See [http://en.wikipedia.org/wiki/Petri\\_net](http://en.wikipedia.org/wiki/Petri_net) for the syntax and semantics of Petri nets.

<sup>2</sup> The toolkit and the documentation are available at <http://www.laas.fr/tina/>

<sup>3</sup> See [http://en.wikipedia.org/wiki/Linear\\_temporal\\_logic](http://en.wikipedia.org/wiki/Linear_temporal_logic)