Branimation

A proposal for tool development to assist animation of Event-B specifications with Brama

Atif Mashkoor

(atif.mashkoor@loria.fr)

(LORIA, France)

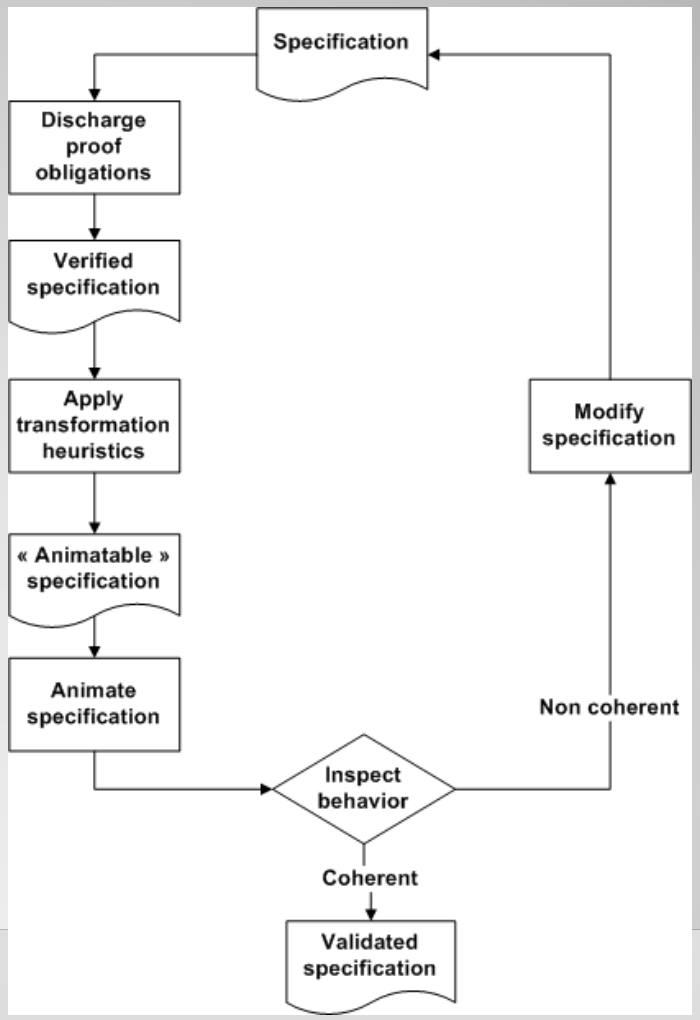
Rodin User and Developer Workshop
15-17 July 2009
Southampton, UK

Table of contents

- The aim
- The Brama
- The limitations of Brama
- The heuristics
- The Branimation tool
- The conclusion and future work

The main aim:

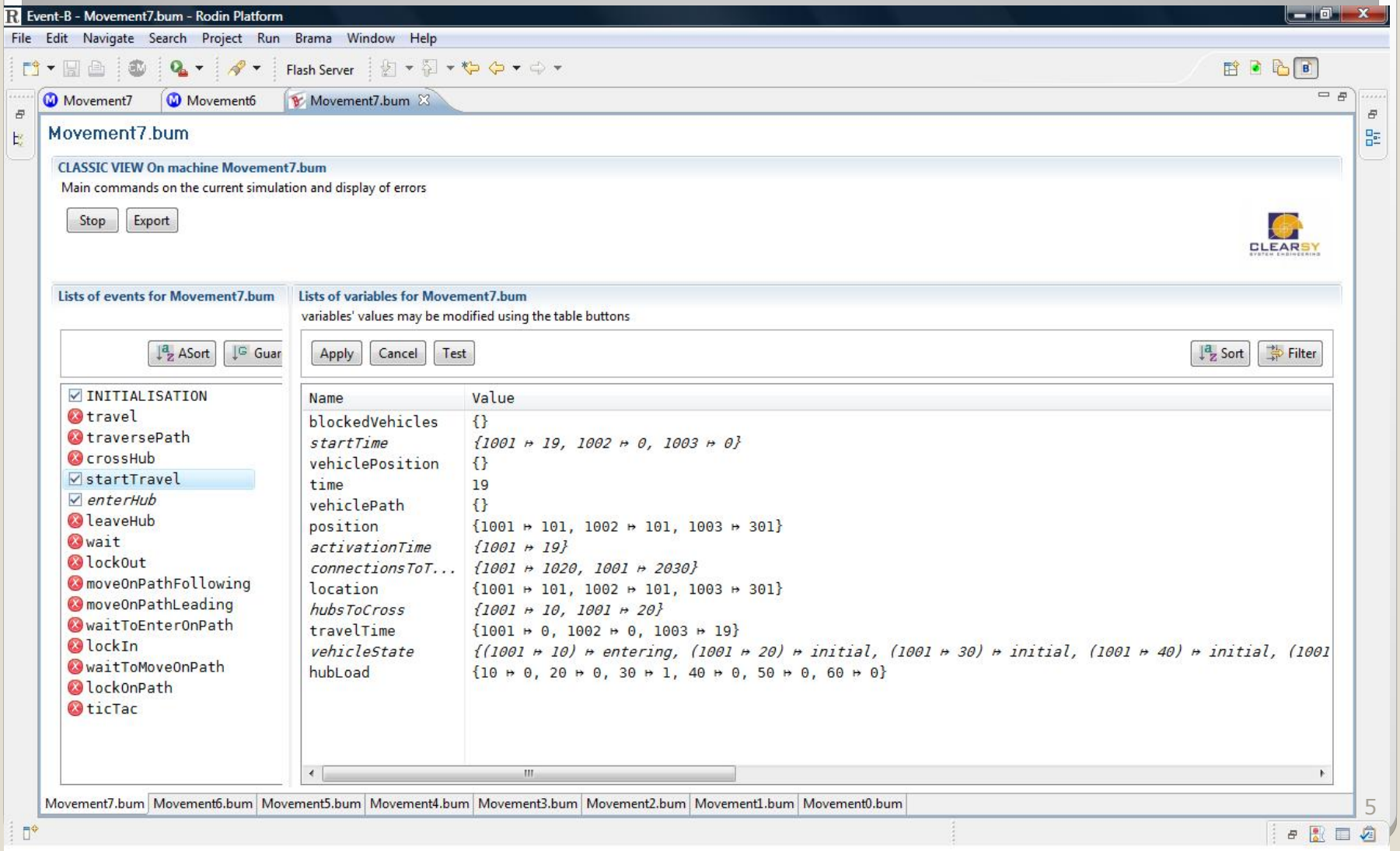
- A stepwise validation process complemented by animation
  - Help in design of complex specifications
  - Early requirement validation
  - Early user involvement
  - Quick and cheap



Brama

- Animator for Event-B specifications
- Eclipse based plug in for Rodin platform
- Graphical animation of specifications with Flash tools
- Demonstration of behavior of the system by firing of events (enabled/disabled)
- Supply of numerical values to constants

Brama snapshot



Limitations

1. "Finite" clause
2. Interpretation of quantifications as iterations
    1. Operation on finite lists
    2. Lack of assurance of animation despite list finiteness
    3. Typing information of sets involved in iteration
3. Dynamic bindings in substitutions
    1. Dynamic mapping of variables
    2. Dynamic function computation

Limitations (cont.)

1. Analytic function computation in Contexts
   1. Analytical function computation in events
   2. Case analysis functions expression in a single event
   3. Evaluation of invariants based on function computations
2. Limited communication with external graphical animation environment

The heuristic pattern

Heuristic Pattern

Symptom:	**What reveals the situation e.g. Brama error message**
Transform:	**The expression schema in the original specification and its transformed counterpart**
Caution:	**Description of the applicability conditions, possible effects, and precautions to follow**
Justification:	**A rigorous argument about the validity of the transformation**

The heuristics

1 Remove the axiom "finite" from the specification

2.1 Specify the finiteness of a quantified domain
2.2 Generalize expressions involving complex iterations
2.3 Explicitly provide the typing information of all sets used in an axiom

3.1 Avoid dynamic mapping of variables in substitutions

The heuristics (cont.)

3.2 Avoid dynamic function computation in substitutions

4.1 Use Inlining
4.2 Replicate events
4.3 Remove invariants

5. Introduce observation variables

2.2 Generalization of complex iterations

Symptom: Impossibility to build the iterators of the predicate

Pattern: Take super-set of the expression

- Original var = $\{x \mid \exists n \,.\, n \in \mathbb{N}1 \;\&\; x \in 1 \,..\, n \rightarrow y\}$
- Transformed var $\in \mathbb{P}(\mathbb{N} \text{ -+> } y)$

Caution:

- Some proof obligations may not be discharged
- Vigilant input of values

Justification:

- Vigilant input of values ensures same behavior
- Since original spec is verified, so transformed spec has the same properties

The Branimation tool

- To implement the proposed heuristics
- May not be fully automated
- Human intervention
- Simple tasks
  - removal of finite clause
  - provision of typing information
  - event replication

Conclusion and future work

- Animation constraints and heuristics
- Growing list of heuristics
- Needed development of supporting tool
- Checking of specification with ProB