Stepwise Validation of Formal Specifications

Atif Mashkoor, Jean-Pierre Jacquot

Stepwise Validation of Formal Specifications ⋆

Atif Mashkoor, Jean-Pierre Jacquot

LORIA – DEDALE Team – Nancy Université
Vandoeuvre-Lès-Nancy, France
{firstname.lastname}@loria.fr

1 Motivation

Formal methods are recommended for the development of software controlled safety critical systems because they support formal verification. However, they raise the issue of validation. Mostly due to poor readability of formal texts, validation is difficult.

The key idea of stepwise development process, such as B method [1] is to break the general verification into a sequence of smaller steps, called proof-obligations, associated with each refinement. We aim at "breaking" validation in the similar way. Technology allows us to animate specifications so users can validate them.

Verification and validation impose very different constraints on the specification text: the former prefers abstract, non constructive, legible (for specifiers!) texts, the latter requires concrete, constructive, computable texts, for instance. These constraints come from the processes (proofs) or the tools (provers, animators). Validation through animation implies to modify the text: using it on each refinement therefore requires a cost-effective transformation technique.

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

Fig. 1. The Heuristic Pattern

2 Stepwise validation

Experimenting with Brama[1] on two Event-B[2] safety critical systems–a formal domain model of land transportation [2] and a situated multi-agent platooning system [3]–we have dressed up a typology of five general cases:

1 Brama forbids finite clause in axioms
2 Brama interprets quantifications as iterations
3 Brama cannot compute dynamic functional bindings in substitutions
4 Brama does not compute analytically defined functions

⋆ Work partially supported by ANR under project ANR-06-SETI-017 TACOS (http://tacos.loria.fr), and by Pôle de Compétitivité Alsace/Franche-Comté under CRISTAL project (http://www.projet-cristal.org).

[1] http://www.brama.fr
[2] http://www.event-b.org

5 Brama has limited communication with its external graphical environment

This lead us to design 10 transformation heuristics expressed following the pattern shown by figure 1.

We designed the heuristics to preserve the behavior of the specification, *not* its formal properties. In particular, the transformed specification may not be provable. The correctness of the transformation is then a crucial issue.

Since heuristics cannot be "proven" within B formal logic system, we relied on the mathematical tradition of *rigorous arguments*. For this to work, we need a basic assumption: the initial specification text must have been formally verified. Most of the arguments given in the **Justification** clause of heuristic rely on this hypothesis.

So, to be used confidently, the heuristics must be part of the rigorous process illustrated in figure 2. Its tenet is "first verify, then validate." Problems revealed during the animation should lead to correction into the *initial* text, whose proof-obligations should be proved again, and heuristics are re-applied before the next validation round.

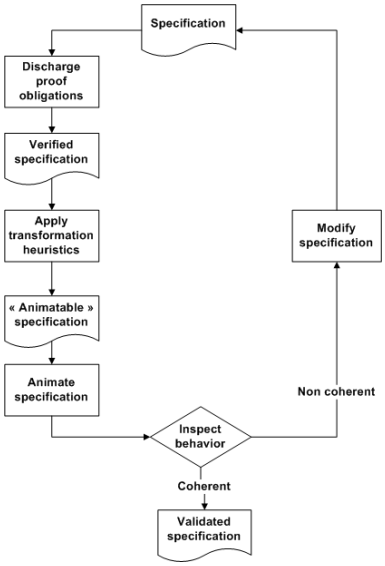

Fig. 2. The Transformation Process

3 Conclusion

The process and heuristics presented here are correct and cost-effective. In the case-study of the land transport domain model development, setting up animation with the process was cheap enough so that we could use animation as a form of "prototyping" to help design and fix a refinement.

The technique was also successfully used on the platooning system. Although the initial specification has been declared "correct", animations showed intriguing and interesting behaviors that were not expected.

Some heuristics entail major modification of the text, such as event duplication. Their manual application is tedious and error-prone. Implementing the heuristics, which does not seem to raise big problems, is mandatory to make our technique a routine exercise for specifiers.

References

1. Abrial, J.R.: The B Book. Cambridge University Press (1996)
2. Mashkoor, A., Jacquot, J.P., Souquières, J.: Transformation Heuristics for Formal Requirements Validation by Animation. In: 2nd International Workshop on the Certification of Safety-Critical Software Controlled Systems (SAFECERT 2009), York (UK) (2009) 16 pp.
3. Lanoix, A.: Event-B specification of a situated multi-agent system: Study of a platoon of vehicles. In: 2nd IFIP/IEEE International Symposium on Theoretical Aspects of Software Engineering (TASE), IEEE Computer Society (2008) 297–304