Structuring Requirements for Better Model Integration

Michael Jastram (Heinrich-Heine Universität Düsseldorf)

Overview

Requirmements management and modeling are complimentary — this is nothing new. But creating a good integration between requirements and models is challenging. We see some promising approaches in practice, for instance the successful SysML modeling notation. SysML support requirements and the traceability between requirements and model elements. Unfortunately, requirements are just arbitrary text blocks.

We present an approach that restructures the natural language requirements to allow better traceability to models. We also present tool support for our approach, based on the ProR requirements engineering platform. We use it for integrating of requirements with the Event-B formal notation via the Rodin platform, and with SysML via the Topcased platform.

Structuring Requirements with WRSPM

WRSPM [GJGZ00] is a reference model for applying formal methods to the development of user requirements and their reduction to a behavioral system specification. It distinguishes between artefacts and phenomena. Phenomena describe the state space (and state transitions) of the domain and system, while artefacts represent constraints on the state space and the state transitions. The artefacts are broadly classified into groups that pertain mostly to the system versus those that pertain mostly to the environment. These are Domain Knowledge (W), Requirements (R), Specifications (S), Program (P) and Programming Platform (M).

Once the requirements are structured according to WRSPM, a number of properties can be verified with regard to the model.

Traceability to Models

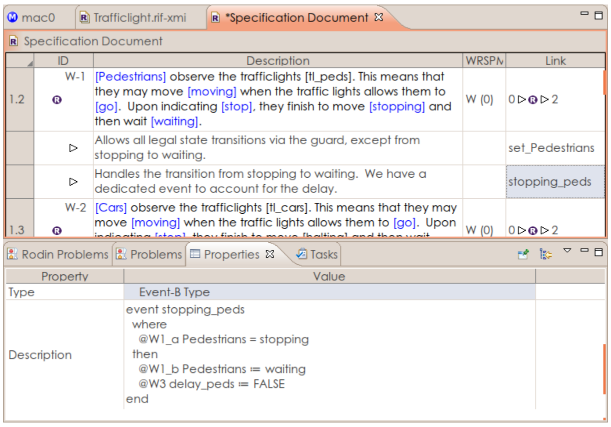
Once structured according to WRSPM, it becomes much easier to build a model that corresponds to the requirements. In our work, we focus on state-based modelling and provide an example using Event-B. Using Event-B we can also discuss limitations of requirements tracing: not all requirements can be formalized within the core Event-B formalism. Formal and informal reasoning need to be combined in a sensible way.

Our approach is not tied to Event-B, and we demonstrate it with a SysML model. However, formal modeling allows much more sophisticated reasoning. Our approach for establishing and maintaining the traceability is described in [JHLR10] and [JHL11].

Tool Support

A major contribution of our work is the development of a platform for managing natural language requirements, called ProR. ProR is now part of the Eclipse Requirements Modeling Framework (RMF).

We implemented prototypical integrations of ProR with Rodin, a platform for Event-B modeling (see figure). We also experimented with Topcased, an Eclipse-tool that supports SysML, amongst other notations.



	ID	Description	WRSPM	Link
1.2	W-1 Ⓡ	[Pedestrians] observe the trafficlights [tl_peds]. This means that they may move [moving] when the traffic lights allows them to [go]. Upon indicating [stop], they finish to move [stopping] and then wait [waiting].	W (0)	0 ▷ Ⓡ ▷ 2
	▷	Allows all legal state transitions via the guard, except from stopping to waiting.		set_Pedestrians
	▷	Handles the transition from stopping to waiting. We have a dedicated event to account for the delay.		stopping_peds
1.3	W-2 Ⓡ	[Cars] observe the trafficlights [tl_cars]. This means that they may move [moving] when the traffic lights allows them to [go]. Upon indicating [stop], they finish to move [halting] and then wait	W (0)	0 ▷ Ⓡ ▷ 2

Property	Value
Type	Event-B Type
Description	event stopping_peds where @W1_a Pedestrians = stopping then @W1_b Pedestrians ≔ waiting @W3 delay_peds ≔ FALSE end

Integration of ProR with an Event-B formal model. Model elements are highlighted in the requirement text.

Conclusion

We believe that our approach makes a number of contributions that are useful individually, but make a large impact if used together. As it supports various WRSPM-based approaches for requirements structuring and various modeling notations, we believe that it is flexible enough to be tailored to a wide range of situations.

References

[GJGZ00] Carl A. Gunter, Michael Jackson, Elsa L. Gunter, and Pamela Zave, *A reference model for requirements and specifications*, IEEE Software **17** (2000), 37–43.

[JHL11] Michael Jastram, Stefan Hallerstede, and Lukas Ladenberger, *Mixing Formal and Informal Model Elements for Tracing Requirements*, Automated Verification of Critical Systems (AVoCS), 2011.

[JHLR10] Michael Jastram, Stefan Hallerstede, Michael Leuschel, and Aryldo G Russo Jr, *An approach of requirements tracing in formal refinement*, VSTTE, Springer, 2010.