Feng Sheng, Liang Dou, Zong-yuan Yang, 2017. Mechanized semantics and refinement of UML-Statecharts. *Frontiers of Information Technology & Electronic* Engineering, **18**(11): 1773-1783. http://dx.doi.org/10.1631/FITEE.1601196

Mechanized semantics and refinement of UML-Statecharts

Key words: UML-Statecharts; Coq; Refinement; Structured operational semantics

Corresponding author: Liang Dou
E-mail: ldou@cs.ecnu.edu.cn

ORCID: http://orcid.org/0000-0003-3044-3841

Motivation

- The Unified Modeling Language (UML) has been developed as a standard object-oriented modeling notation in MDE (Model-driven engineering) and is well accepted in the industry.
- UML cannot provide the formal specification of refinement in software design, which makes it difficult to formalize the refinement relations and verify their desired properties.
- Although formal methods and techniques are expensive, it is worth exploring how formal methods can be applied within MDE.
- Due to its very considerable expressive power and industrial strength support, there has been much effort in using Coq to perform verification.

Main idea

- We use the Coq proof assistant to state specifications, create implementations, and build proofs for UML-Statecharts.
- The properties of refinement relations are described as lemmas using extra auxiliary functions, and the provability of lemmas indicates the correctness of the properties..

Method

1. Formalize the structured operational semantics of the UML-Statecharts in Coq, including abstract syntax, semantic auxiliary functions, and semantics.
2. Prove the desired properties of the UML-Statecharts, e.g. determinacy, transitivity and reflexivity.
3. Define and formalize the refinement relations between UML-Statecharts.
4. Prove the desired properties of the refinement relations, e.g. the incremental refinement must be behavior-preserving.

Major results

- Provide a formalization of the semantics of the UML-Statecharts and the refinement relations in Coq.

- Prove that the one-step refinements are behavior-preserving, and multi-step refinements are reflexitive and transitive.

Conclusions

- The advantage of our approach is an intuitive and graph-based representation of the structured operational semantics and the refinement relations.
- This work provides a way to possibly obtain certified fault-free modeling and refinements.
- Our future work will be directed toward the extension of the semantics and refinement relations considering the initial state, final state, and data dependencies during transitions.