SFAssist 2021-2023: Automatic Structural Improvements of Stateflow Automata

Funded by Pro FIT - Project Financing of Investitionsbank Berlin (IBB)
Project partners: Technische Universität (TU) Berlin Department of Software and Embedded Systems Engineering (formerly: Programming of Embedded Systems)
Lead partner: Prof. Dr. Sabine Glesner

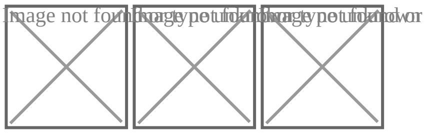
Image not found or type unknown
Image not found or type unknown
Image not found or type unknown

Starting point

Model-based software development (MBD) is an established approach for developing electronic control systems in the automotive sector. Developers model the desired behavior of the overall system and use these models, among other things, as a starting point for the automatic generation of code that is used directly or with adaptations in the vehicle. Simulink and Stateflow are the quasi-standard for model-based development in the automotive sector. In Simulink, functionality is developed using block diagrams, while in Stateflow state machines are created. The ISO 26262 standard (Road vehicles - Functional safety) defines strict quality standards for electronic control systems in the automotive sector. A central requirement of the standard is to maintain low complexity for safety-relevant software components.

There has been a steady rise in the application of embedded systems in vehicles in the past few years. Consequently, the interactions between these applications increase in a superlinear way, because execution and behavior of the applications are highly interdependent. As a result, the related Stateflow automata are also becoming more complex, with the associated drawbacks of poor traceability and error-proneness. Additionally, attaining complete test coverage of the model with reasonable effort, as required by the security standards, is barely viable.

One way out of the complexity trap of a pure Stateflow automata is to decompose the state models to dataflow models.

Objective

Our goal is to develop a method that can transform Stateflow automata into structured automata with explicit interfaces and mutual dependencies in a fully or partially automated way using Simulink subsystems:

- Resulting automata must describe functionality of the initial model in a much more structured way, i.e., by models that meet the complexity specifications of established safety standards.
- Basis: collection of syntactic transformations that decompose Stateflow automata and structure them via Simulink subsystems. In particular, define transformation sequences that assist developers in resolving specific anti-patterns in Stateflow.
- Subgoal: determine which transformations can be automated or if application knowledge is required.

Benefits

To evaluate the effectiveness of the transformation we will use a complexity metric for Stateflow automata which already has a well-established market presence:

- Problematic automata have a high complexity value.
- The method is successful in transforming these automata into models with lower complexity.

This project demonstrates practical benefits by carrying out procedures and transformations for Stateflow in combination with Simulink as functional models. Models from customer projects that are at MES' disposal will form the basis of the project and the evaluation thereof.

Project Partner

The Department of Software and Embedded Systems Engineering at the Technische Universität Berlin, headed by Prof. Dr. Sabine Glesner, investigates issues related to the software engineering of embedded and cyber-physical systems. In particular, the department is concerned with the development of methods and tools with which embedded and cyber-physical applications can be created correctly, reliably, securely, and efficiently. This includes specification, verification, and generation methods for embedded software, model-based development (especially Simulink/Stateflow), hardware/software co-design, and quality assurance for mixed hardware/software systems.

- Image not found or type unknown sfassist_auto_structural_improvement.pdf (105.5 KiB)