

# SFAssist 2021-2023: Automatische Strukturverbesserungen von Stateflow-Automaten

Gefördert durch die Pro FIT - Projektfinanzierung der Investitionsbank Berlin (IBB)  
Projektpartner: TU Berlin Fachgebiet Software and Embedded Systems Engineering (früher: Programmierung Eingebetteter Systeme)  
Ansprechpartnerin: Prof. Dr. Sabine Glesner

Dieses Projekt wird kofinanziert durch den Europäischen Fonds für regionale Entwicklung [EFRE].



## Ausgangssituation

Die modellbasierte Software-Entwicklung (MbE) ist ein etablierter Ansatz zur Entwicklung elektronischer Steuerungs- und Regelungssysteme im Automobilbereich. Entwickelnde modellieren das gewünschte Verhalten des Gesamtsystems und nutzen diese Modelle u.a. als Ausgangspunkt für die automatische Generierung von Code, der direkt oder mit Anpassungen im Fahrzeug eingesetzt wird. Den quasi-Standard für modellbasierte Entwicklung im Automobilbereich bilden Simulink und Stateflow. In Simulink wird die Funktionalität mittels Block-Diagrammen entwickelt, während in Stateflow Zustandsautomaten erstellt werden. Die ISO-Norm 26262 (Road vehicles – Functional safety) definiert strenge Qualitätsstandards für Steuerungs- und Regelungssysteme im Automobilbereich. Eine zentrale Anforderung im Standard ist eine niedrige Komplexität für sicherheitsrelevanten Softwarekomponenten.

In den letzten Jahren beobachten wir eine immer stärkere Zunahme der Applikationen in Fahrzeugen. Als Folge steigen die Wechselwirkungen zwischen diesen Anwendungen überlinear, weil Ausführung und Verhalten der Applikationen in hohem Maß voneinander abhängen. Dadurch werden auch die zugehörigen Stateflow-Automaten immer komplexer, mit den verbundenen Nachteilen der schlechten Nachvollziehbarkeit und Fehleranfälligkeit. Außerdem kann eine vollständige Testüberdeckung des Modells, wie in den Sicherheitsstandards gefordert, kaum noch mit vertretbarem Aufwand erreicht werden.

Ein Ausweg aus der Komplexitätsfalle eines reinen Stateflow-Automaten besteht in der Zerlegung der Zustandsmodelle auf Datenflussmodelle.

## Zielsetzung

Unser Ziel ist es, ein Verfahren zu entwickeln, das Stateflow-Automaten unter Einsatz von Simulink-Subsystemen in strukturierte Automaten mit expliziten Schnittstellen und wechselseitigen Abhängigkeiten voll- oder teilautomatisiert transformieren kann:

- Resultierende Automaten müssen Funktionalität des Ausgangsmodells in deutlich besser strukturierter Form beschreiben, d.h. durch Modelle, die den Komplexitätsvorgaben der etablierten Sicherheitsstandards entsprechen.
- Basis: Sammlung syntaktischer Transformationen, die Stateflow-Automaten zerlegen und über Simulink-Subsysteme strukturieren. Insbesondere sollen Transformationssequenzen definiert werden, die Entwickelnden bei der Auflösung spezifischer Anti-Pattern in Stateflow unterstützen.
- Teilziel: feststellen, welche Transformationen automatisiert vorgenommen werden können oder ob Anwendungswissen erforderlich ist.

## Nutzen

Die Wirksamkeit der Transformationen wird durch eine Komplexitätsmetrik für Stateflow-Automaten evaluiert, die bereits am Markt etabliert und akzeptiert ist:

- Problematische Automaten weisen einen hohen Komplexitätswert auf.
- Das Verfahren ist erfolgreich, wenn es diese Automaten in Modelle mit geringerer Komplexität transformiert.

Dieses Projekt weist den praktischen Nutzen nach, indem Verfahren und Transformationen für Stateflow in Kombination mit Simulink als Funktionsmuster realisiert werden. Die Basis der Arbeiten und der Evaluierung werden Modelle aus Kundenprojekten sein, die MES zur Verfügung stehen.

## Projektpartner

Das Fachgebiet Software and Embedded Systems Engineering an der Technischen Universität Berlin, geleitet von Prof. Dr. Sabine Glesner, untersucht Fragen, die im Zusammenhang mit dem Software Engineering eingebetteter und cyberphysikalischer Systeme auftreten. Insbesondere beschäftigt sich das Fachgebiet mit der Entwicklung von Methoden und Werkzeugen, mit denen eingebettete und cyberphysikalische Anwendungen korrekt, zuverlässig, sicher und effizient erstellt werden können. Dazu zählen Spezifikations-, Verifikations- und Generierungsmethoden für eingebettete Software, modellbasierte Entwicklung (insbesondere Simulink/Stateflow), Hardware/Software Co-Design und Qualitätssicherung für gemischte Hardware/Software-Systeme.



- [sfassist\\_auto\\_strukturverbesserung.pdf \(111,4 KiB\)](#)