

ERFAHRUNGSBERICHT VON GIGATRONIK

# Modellbasierte Entwicklung

## – vom A-Muster bis zur Serien-ECU

Die modellbasierte Entwicklung von Steuergerätesoftware mit MATLAB/Simulink ist das Erfolgsmodell der Softwareentwicklungsmethoden der letzten Jahre. Trotzdem sind viele Einzelprobleme in der Softwareentwicklung nach einem Automotive-SPICE-konformen Prozess in der Praxis noch nicht gelöst. Anhand der Erfahrungen bei Gigatronik bei der Entwicklung von Getriebe- und Kupplungssteuerungssoftware werden Lösungen vorgestellt, die in der Projektzeit von fünf Jahren entwickelt wurden.

In der Entwicklung von Steuergerätesoftware mittels Model-Based Design werden die Werkzeuge MATLAB/Simulink und ein Autocodegenerator eingesetzt, z. B. MathWorks Simulink RTW EC bzw. dSPACE TargetLink, aber auch GENE-AUTO oder SCADE. Im Vergleich zu anderen Entwicklungsparadigmen ergeben sich dabei einige Unterschiede:

- Die Software wird signalflussbasiert modelliert (Blöcke und gerichtete Verbindungen).
- Einzelne Komponenten werden als Zustandsautomaten bzw. als Flussdiagramme modelliert.
- Die Software wird in Modellen und Modellbibliotheken organisiert.
- Die Werte von Signalen entsprechen, soweit anwendbar, physikalischen Größen.

Diese Unterschiede haben große Auswirkung auf die Arbeitsweise: Jedes Modell ist in der Simulationsumgebung MATLAB/Simulink direkt lauffähig, sodass die Funktion auf der abstrakteren Modellebene überprüft und die Integration mehrerer Modelle durchgeführt werden kann („Modell der zukünftigen Software“). Die Modelle werden im zweiten Schritt an die Zielplattform angepasst. Dabei werden die Signale skaliert (Zuordnung von Datentypen, Wortbreite, Auflösung und Offset) und Schnittstellensignale an die Schnittstellen im Steuergerät angebunden.

Mit dem Codegenerator wird aus den Modellen C-Code für das Zielsteuergerät generiert. Der C-Code kann auf dem PC als Software-in-the-Loop (SiL) und auf einem Zielprozessor

als Processor-in-the-Loop (PiL) simuliert werden. Abweichungen zwischen Modell, SiL und PiL sowie Skalierungs- und Schnittstellenfehler werden vor der Integration auf der Hardware entdeckt.

### Anwendung im Projekt

Das Zielsystem der Software ist ein Steuergerät für die Getriebeautomatisierung. Die Steuergeräte-Hardware und die Basissoftware werden von einem Tier-One-Lieferanten geliefert, die Applikationsschicht setzt sich zusammen aus Softwarekomponenten des OEM und aus Software, die von Gigatronik am Standort Stuttgart entwickelt wurde. Gigatronik liefert Softwaremodule für das Fehlermanagement, die Variantencodierung, die Kupplungslageregelung, die Regelung der Getriebeaktoren sowie die funktionale Diagnose und integriert die Gesamtsoftware.

Für die exakte Ansteuerung der Magnetventile der Getriebebesteuerung wird eine Simulation des pneumatischen Teilsystems (Ventile, Zylinder, Kammern) auf dem Steuergerät gerechnet. Für die Softwareentwicklung wird ein SPICE-Assessment Level 2 gefordert, um die Qualität der Software zu gewährleisten. Die geforderten, statischen Code-Analyse-Tests werden mit MathWorks Polyspace durchgeführt. Die einzelnen Funktionen wurden im Zeitraum Ende 2007 bis Mitte 2012 von Gigatronik mit einer Mannschaft von bis zu zwölf Projektmitarbeitern entwickelt, die sich aus den Bereichen Funktionsentwicklung, Testautomatisierung, Projektmanagement und Qualitätssicherung, Konfigurationsmanagement und Buildmanagement zusammensetzten.

## Prozessintegration

Für effektives und prozesssicheres Arbeiten muss die Entwicklungsumgebung in die Werkzeuge für Anforderungs- und Konfigurationsmanagement integriert werden.

Die Toolbox Simulink V&V (Verification and Validation) bietet eine Basisintegration zum Anforderungsmanagement in Microsoft Word und Excel, PDF, HTML und IBM Rational DOORS. Die Gigatronik-Prozesslandschaft setzt als Anforderungs- bzw. Konfigurationsmanagement-Werkzeuge Borland CaliberRM und Borland StarTeam ein. Über ein im Rahmen des Projekts entwickeltes Plug-in für CaliberRM können Anforderungen in CaliberRM bidirektional und interaktiv mit Blöcken in Simulink verlinkt werden (**Bild 1**).

## Konfigurationsmanagement und Tests

In einem Gesamtmodell (**Bild 2**) werden alle Module mit einer Nachbildung der Schnittstelle und einer Umweltsimulation integriert und bilden eine Simulation der Gesamtsoftware im Steuergerät.

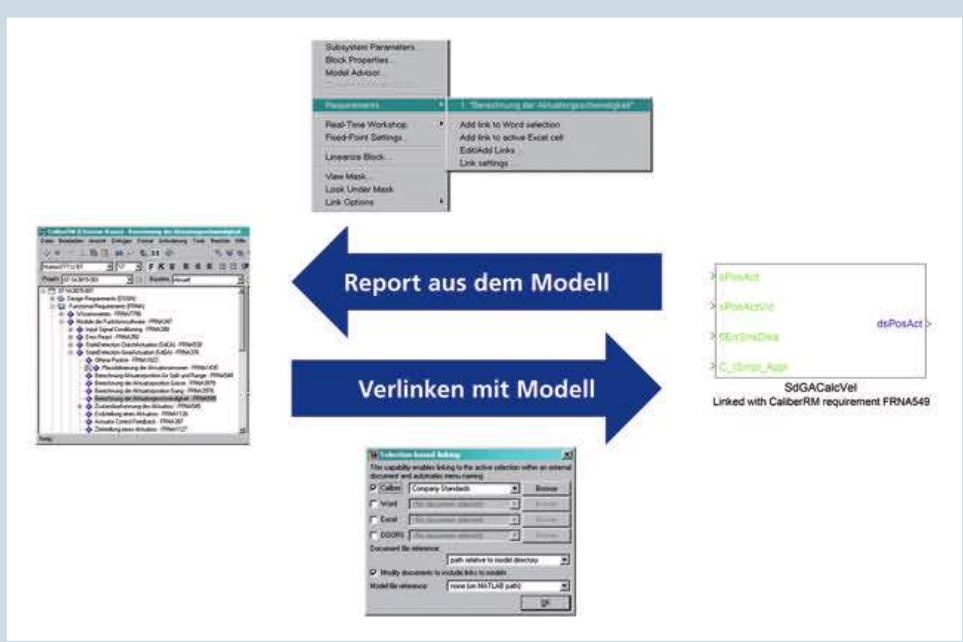
Simulink-Modelle haben eine hierarchische Struktur, d. h. sie bestehen aus Subsystemen. Indem man die verschiedenen Subsysteme auf verschiedene Bibliotheken verteilt, die als eigene Dateien im Konfigurationsmanagementsystem verwaltet werden, lässt sich das Gesamtmodell nachvollziehbar konfigurieren: Funktionale Änderungen der Module können einzeln versioniert werden.

Damit kann bereits in der Modellentwicklung die Integration aller Softwaremodule im Gesamtsystem überprüft werden und Fehler auf Modellebene gefunden werden, bevor die Modelle zur Codegenerierung freigegeben werden. Alle auf Softwareentwicklungsebene (Software construction ENG.6) und Softwareintegrationsebene (Software integration test ENG.7) von SPICE geforderten Tests werden in der MATLAB/Simulink-Umgebung durchgeführt.

- Unit-Tests und Simulation sind in der Entwicklungsumgebung jederzeit möglich.
- Einzelne Modelle können untereinander integriert und somit vor der Gesamtintegration überprüft werden.
- In der Simulationsumgebung können Regressionstests vom Modell, dem skalierten C-Code und dem Target-C-Code ohne Änderung der Infrastruktur ausgeführt werden.

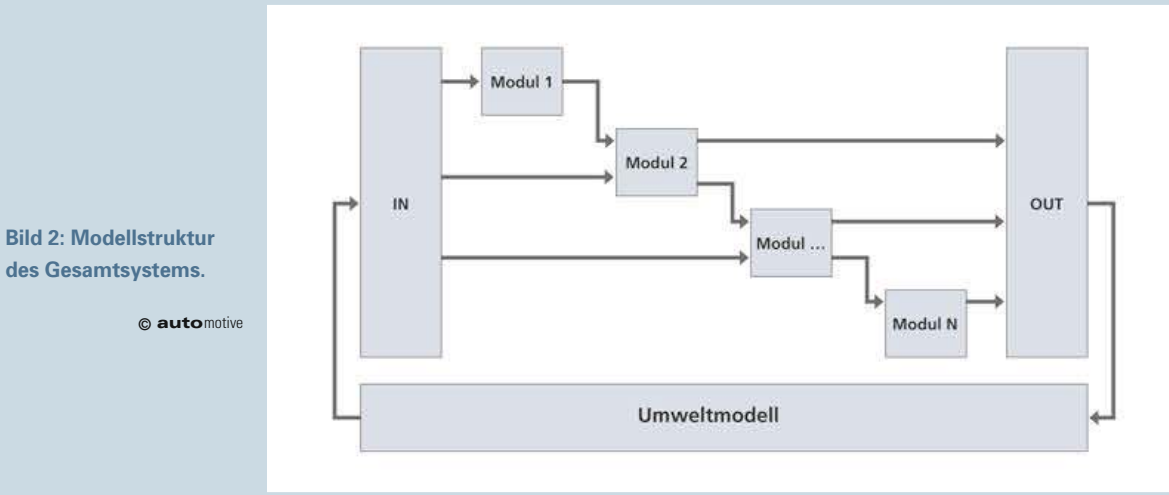
Realistische Testfälle werden modulweise in der Gesamtsystemsimulation generiert. Der Testfall wird automatisiert nach bestimmten Erfolgskriterien ausgewertet. Danach werden die Ergebnisse der Simulation mit den Ergebnissen einer SiL- und einer PiL-Simulation verglichen, die automatisiert über dSPACE TargetLink mit einem Evaluationsboard mit dem Zielprozessor durchgeführt werden. In den SiL-Modultests wird zusätzlich eine Abdeckungsanalyse (C1) durchgeführt.

Für die Vereinfachung des Testprozesses wurde von Gigatronik das Werkzeug GT-moteto entwickelt, das die Testobjekte mit intelligenten Stimulationsblöcken instrumentiert, die Testfälle aufnimmt und verwaltet sowie die SiL- und PiL-Tests automatisiert durchführt. Für die Entwicklung von Steuergerätesoftware ist die Einhaltung von Programmierungs- und Modellierungsrichtlinien gefordert. Unter anderem sind dies: MISRA Guidelines for C, MISRA Guidelines for AC TL, kundenspezifische und Gigatronik-interne Richtlinien. Die einzelnen Richtlinienkataloge müssen untereinander abgeglichen und ihre Prüfbarkeit bewertet werden. Die Einhaltung der nicht automatisiert prüfbareren Regeln wird dann im Rahmen von Modell-Reviews geprüft, die automatisiert prüfbareren Regeln werden mit dem Werkzeug MXAM (MES Berlin) getestet.



**Bild 1: Verlinkung Anforderung und Modell.**

© automotive



**Bild 2: Modellstruktur des Gesamtsystems.**

© automotive

**Anforderungen an die Toolchain**

Die Toolchain MATLAB/Simulink und dSPACE TargetLink ist – im Vergleich zu einer Entwicklungsumgebung wie zum Beispiel „Eclipse“ – schlecht in die Tool-Welt eingebunden. Jedoch bildet die Plattform aus MATLAB/Simulink und Autocodegenerator zusammen mit einem integrierten Modultestwerkzeug und einem integrierten Konfigurationsmanagement eine vollständige, ausgereifte Entwicklungsplattform. Ein Versionsupdate ist nur dann notwendig, wenn sich das Architekturparadigma ändert, z. B. bei der Einführung von AUTOSAR. Eine Evaluierung neuerer Versionen wird erst beim Start neuer Projekte durchgeführt.

**Fazit**

Der Einsatz der modellbasierten Softwareentwicklung hat in diesem Projekt die Implementierungszeiten reduziert und den initialen Softwarereifegrad erhöht. Im weiteren Projektverlauf hat sich gezeigt, dass verschiedene Qualitätsprozesse noch nicht in die Simulink-Umgebung integriert sind. Mit der Programmierung eines Interfaces zwischen dem Requirements Engineering Tool und Simulink und der Programmierung des Modultestwerkzeugs GT-

moteto wurden deutliche Lücken in der Toolchain geschlossen. Da das Modell plattformunabhängig entwickelt wird, kann die Software vom A-Muster (auf einem Rapid-Control-Prototyping-System) bis zum Seriensteuergerät eingesetzt werden. Durch inkrementelle Codegenerierung werden die Generier- und Kompilierzeiten klein gehalten. (oe)

.....  
**Dipl.-Ing. Stefan Harms** war von 2002 bis 2012 bei der GIGATRONIK-Gruppe beschäftigt und verantwortete seit 2009 im Bereich Komponentenentwicklung am Standort Stuttgart das Team Funktionsentwicklung.

.....  
**Dipl.-Ing. Jochen Lüling** ist Projektleiter und Leiter der Abteilung Prozess- und Projektmanagement am Stuttgarter Standort der GIGATRONIK-Gruppe.

.....  
**Dipl.-Ing. Christian Bauer** ist seit 2006 in der Abteilung Funktionsentwicklung & Simulation der GIGATRONIK-Gruppe am Standort Stuttgart beschäftigt.