



# Schnell geschaltet

Prozesse und Methoden für die modellbasierte Entwicklung  
von Getriebe-Funktionssoftware mit TargetLink



Das Ziel, die Entwicklungszeit von Prototypen und Serienprodukten deutlich zu reduzieren, erfordert innovative Methoden bei der Entwicklung eingebetteter Software. Daher entschloss sich die Continental AG vor einigen Jahren, die modellbasierte Entwicklung und Serienne-Generierung für die Entwicklung von Getriebesteuergeräten einzuführen. Dieser Artikel betrachtet die gesamte Vorgehensweise von der Ausarbeitung der Prozesse und Methoden bis zum erfolgreichen Einsatz des Serienne-Generators TargetLink in zwei Serienprojekten.

Das letzte Jahrzehnt hat enorme Herausforderungen für die Entwicklung eingebetteter Software mit sich gebracht. Zum traditionellen Prozess der manuellen Codierung und zum Test von Software sind strengere Anforderungen in Bezug auf Sicherheitsstandards und Entwicklungsmethoden hinzugekommen. Zusätzlich verschärft werden diese Herausforderungen durch eine ständig zunehmende Komplexität, insbesondere was die Steuerung der neuen Getriebegeneration mit Doppelkupplungsgetriebe (Double Clutch Transmission, DCT) und stufenlosem



Das Doppelkupplungsgetriebe verbindet die Energieeffizienz eines Schaltgetriebes mit dem Komfort einer Wandlerautomatik.

Getriebe (Continuously Variable Transmission, CVT) betrifft. Darüber hinaus sind die Zulieferer der Automobilindustrie unablässig auf der Suche nach Möglichkeiten, die Entwicklungszeiten – wie vom Markt gefordert – zu verkürzen. Aus diesem Grund führte der Geschäftsbereich Getriebe der Continental AG bereits vor einigen Jahren eine Untersuchung durch, die die Anwendung eines modellbasierten Entwurfs mit einer daraus resultierenden Seriercode-Generierung zum Thema hatte. Die sich aus dieser innovativen modellbasierten Vorgehensweise ergebende potenzielle Produktivitätssteigerung wurde ermittelt, indem das Verfahren zunächst in einem internen Projekt zum Einsatz kam. Parallel zueinander wurden der Prozess, die Methoden und die Tools für den modellbasierten Entwurf einschließlich der automatischen Code-Generierung festgelegt und an die spezifischen Bedürfnisse des Getriebebereichs angepasst. Nach dem erfolgreichen Einsatz in einem ersten Serienprojekt wurde die Toolkette schließlich für die Entwicklung der kompletten Anwendungssoftware eines Doppelkupplungsgetriebes eingesetzt. Die einzelnen Entwicklungs-

schritte und gewonnenen Erkenntnisse von der initialen Erprobung über ein erstes Serienprojekt bis hin zur vollständig mit TargetLink autocodierten DCT-Getriebesteuerung sollen nachfolgend dargestellt werden.

#### Ausarbeitung des Prozesses und der Methoden

Im Rahmen interner Projekte wurden zunächst die erforderlichen Prozessschritte ausgearbeitet, um den modellbasierten Entwurf und die automatische Seriercode-Generierung auf den Weg zu bringen. Mit dem Ziel, das komplette V-Modell für die Funktions- und Software-Entwicklung unter Berücksichtigung des Zusammenspiels von OEM, Getriebe- und Elektroniklieferanten abzudecken (Abbildung 1), wurden folgende Anforderungen an den modellbasierten Entwurf festgelegt:

- Verifizieren, Validieren und Archivieren von Modellen, Code und Skripten durch Anpassen an den traditionellen Prozess
  - Modellieren der physikalischen Anforderungen an das Steuergerät
  - Simulieren des echten Steuergeräteverhaltens (Tasks, Ausführungsreihenfolge, Betriebssystem)
  - Unterstützen einer verteilten Entwicklung durch Einrichten einer Multi-User-Umgebung
- Zunächst wurde die modellbasierte Entwicklung auf Funktionalitäten in Demo-Fahrzeugen angewendet. Dabei bestand die bevorzugte Lösung in der Verwendung etablierter Tools auf dem neuesten Stand der Technik. Für den Funktionsentwurf wurde daher MATLAB®/Simulink® gewählt. Der Demonstration neuer Funktionalitäten in Demo-Fahrzeugen in sehr kurzer Zeit kam für den Erfolg des Projekts eine entscheidende Bedeutung zu. Die Simulation von Konzepten zusammen mit der nachfolgenden Prüfung und Verifizierung erfolgte unter Verwendung der Model-in-the-Loop (MIL)-Simulation auf dem PC. Nach erfolgreicher Verifikation war die Funktion für Tests im Demo-Fahrzeug und auf dem Prüfstand unter Einsatz der Rapid-Control-Prototyping (RCP)-Umgebung bereit. Für eine agile Validierung und Feinabstimmung von Funktionen wurden der Real-Time Workshop® und die dSPACE AutoBox/MicroAutoBox verwendet. Durch Front-Loaded-

„Für das mit TargetLink durchgeführte Projekt konnte die Markteinführungszeit sogar verkürzt werden, wofür uns unser Kunde seine Anerkennung aussprach.“

Georg Grassl, Continental AG

Tests war eine frühzeitige Beurteilung der Kundenanforderungen möglich. Dem erfolgreichen Einsatz der modellbasierten Entwicklung in der Rapid-Prototyping-Phase folgte die Umstellung des nächsten Schritts im V-Modell von manueller C-Codierung auf automatische Seriencodierung. Hierfür wurde als Code-Generator dSPACE TargetLink ausgewählt.

Um den Nachweis der Machbarkeit einer automatischen und optimierten Code-Generierung für einen 16-Bit-Mikrocontroller zu erbringen, wurde ein Algorithmus zur Positionssteuerung für ein Verteilergetriebe genutzt. Dabei wurde zuerst die bereits vorhandene Funktion in einen modellbasierten Entwurf übertragen und anschließend mit Implementierungsinformationen für die Seriencodierung erweitert.

Abschließend erfolgten Integration und Validierung des automatisch generierten Codes auf dem Seriensteuergerät. Der Nachweis der Effizienz des generierten Codes erfolgte durch Vergleich des Ressourcenbedarfs mit der bereits in der Serienproduktion eingesetzten manuell codierten Software. Schließlich konnten anhand dieser Voruntersuchungen die wesentlichen Prozesselemente für die Generierung serienreifer Software unter Anwendung der neuen Entwicklungsmethodik identifiziert werden. Es folgt eine nähere Betrachtung dieser Prozesselemente, einschließlich einiger unterstützender Methoden:

#### Modellbasierter Funktionsentwurf

Eine schnelle Bewertung der Umsetzung der funktionalen Anforderungen wird durch die Simulation von Floating-Point-Modellen ermöglicht,

gefolgt von Rapid Prototyping zur Validierung der funktionalen Anforderungen in einem Demo-Fahrzeug. Eine frühzeitige Definition der Funktionsarchitektur mit einer daraus resultierenden Generierung von Modellbibliotheken ist notwendig, um die verteilte Entwicklung und die Konfigurationselemente zu erstellen. Darüber hinaus trägt eine Modellierungsrichtlinie für Funktion und Software dazu bei, kompatible modellbasierte Entwürfe zu erzielen. Industriestandards werden hinsichtlich ihrer Übereinstimmung mit hauseigenen Regeln bewertet und gegebenenfalls durch projekt- und anwendungsspezifische Vereinbarungen ergänzt.

#### Modellbasierter Software-Entwurf

Modellentwurfsmuster zur Generierung eines optimierten Zielcodes werden eingeführt und müssen bereits während des Funktionsentwurfs beachtet werden. Dies ist eng mit den von der MISRA (Motor Industry Software Reliability Association) und den Werkzeugherstellern vorgegebenen Standards verknüpft, etwa den MISRA-AC-T-Richtlinien. Software-Entwurf bedeutet Skalierung, Festlegung von Blockeigenschaften, Anpassung der Konfiguration

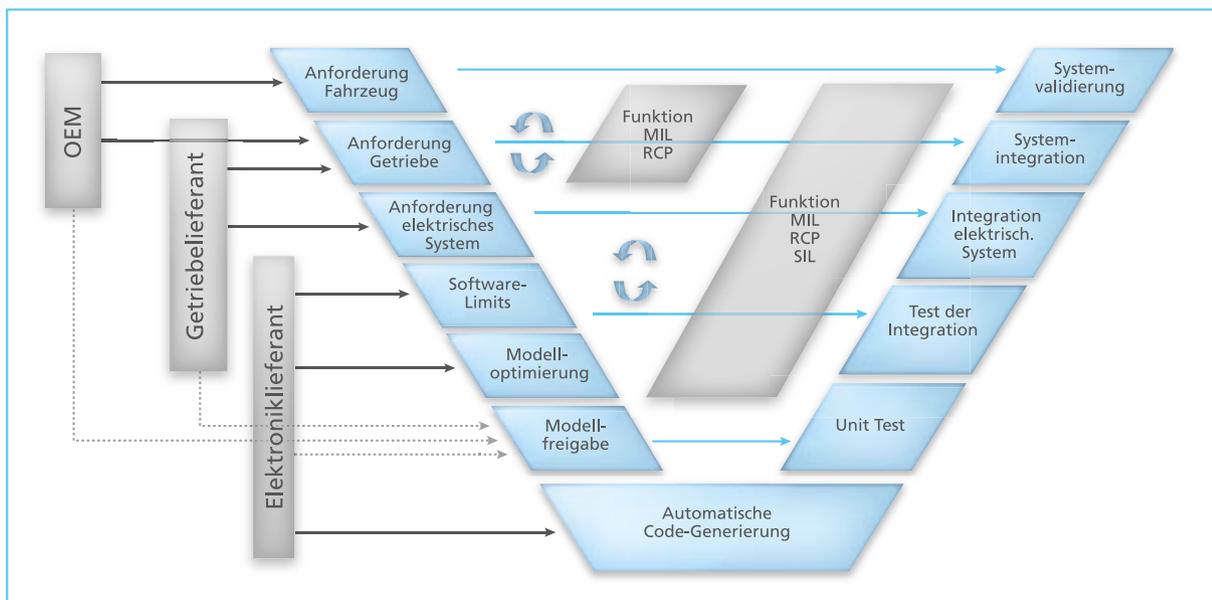


Abbildung 1: Der Entwicklungszyklus im Getriebebereich.

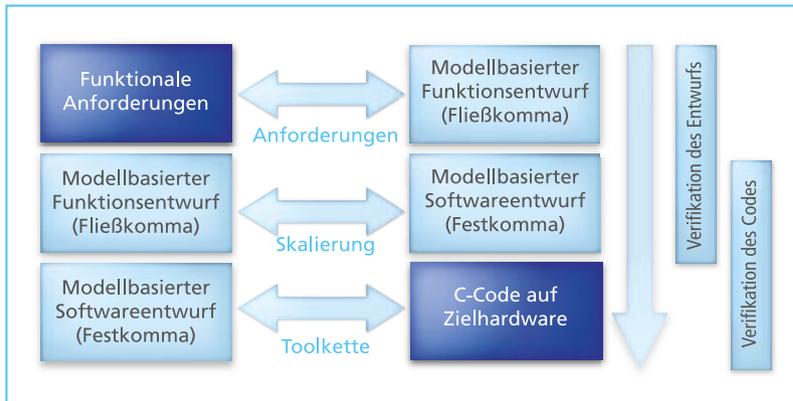


Abbildung 2: Modellbasiertes Testen – Entwurfs- und Code-Verifikation.

des Code-Generators und damit Unterstützung der automatischen Code-Generierung. Durch die Eingabe von Implementierungsinformationen (Speicherzuweisung etc.) ergibt sich ein sogenanntes Implementierungsmodell. An diesem Punkt muss eine frühzeitige Definition der Softwarearchitektur erfolgen, kongruent mit der Funktionsarchitektur mit einer daraus resultierenden Partitionierung in C-Module und C-Funktionen, um die Testbarkeit, Konfiguration (Archivierung) und Wartung zu gewährleisten.

### Modellbasiertes Testen

Zur Optimierung des Verfahrens, im Besonderen durch Automatisierung, ist die Einführung von modellbasiertem Testen und dazu passenden Qualitätsmaßnahmen notwendig. Die standardmäßigen Software-Qualitätsmaßnahmen werden modifiziert. Dies ermöglicht schnelle Mikro-V-Zyklen, um rasche Fortschritte in der Modellreife zu erzielen. Modellbasiertes Testen bietet die Möglichkeit eines schrittweisen Ansatzes (Abbildung 2).

„Im Rahmen einer hauseigenen Untersuchung durch den OEM wurde festgestellt, dass sich der handgeschriebene Code in puncto Fehlerfreiheit nicht mit dem Code-Generator TargetLink messen kann.“

Georg Grassl, Continental AG

### Modellbasierte Zusammenarbeit

Eine enge Zusammenarbeit aller beteiligten Parteien auf Modellebene ist notwendig, um einen konsistenten Modellentwurf und -austausch sicherzustellen. Komplexe Funktionalitäten werden in kleinere Einheiten zerlegt und in Modellbibliotheken abgelegt. Eine auf den Modellen basierende Dokumentation ersetzt die traditionelle in Schriftform erstellte Software-Anforderungsspezifikation.

### Erste Serienanwendung des neuen Prozesses und der Methoden

Basierend auf den Ergebnissen der obigen Voruntersuchungen wurde das erste Hochvolumenprojekt mit einer definierten und abgestimmten Zusammenarbeit zwischen den beteiligten Partnern für alle Themenbereiche im Zusammenhang mit modellbasiertem Entwurf und automatischer Code-Generierung gestartet. Bei dem

modellierten Algorithmus – einer Teilfunktionalität der Applikationssoftware – handelte es sich um eine Schaltstrategie für ein sechsstufiges Automatikgetriebe (Abbildung 3), das für eine 16-Bit-Plattform umgesetzt werden sollte.

Als Tools wurden aufgrund der erfolgreichen Voruntersuchungsprojekte MATLAB/Simulink und dSPACE TargetLink verwendet und entsprechend den zuvor entwickelten Prozessen und Methoden eingesetzt.

Der Test der Funktionssoftware erfolgte mit konventionellen Qualitätssicherungsmaßnahmen, die bereits für manuell geschriebenen C-Code definiert sind, jedoch an Sachverhalte der automatischen Code-Generierung angepasst wurden. Um die Kompatibilität mit hauseigenen Codierregeln zu gewährleisten, wurde eine statische Code-Analyse vorgenommen. Auf Modulebene wurden strukturelle Abdeckungstests durchgeführt und von Tools für eine Testvektorgenerierung unterstützt. Dadurch konnte das Qualitätsziel einer 100%-Abdeckung (höchste Testabdeckungsstufe: Modified Condition/Decision Coverage, MC/DC) des Festkomma-Codes erfüllt werden. Diese Modultests wurden mit einem Standard-Tool durchgeführt, das in der Lage ist, die Mikrocontroller-Plattform zu simulieren. Somit wurde die komplette Toolkette (Code-Generator, Compiler und Linker) erfolgreich verifiziert.

### Ergebnisse und Erkenntnisse aus dem ersten Serieneinsatz

Der von uns gewählte Ansatz gewährleistete ein zeitgerechtes Verfahren, um den festgelegten engen Terminrahmen zu unterstützen.

Als Resultat konnte für dieses Projekt die Markteinführungszeit sogar verkürzt werden, wofür der OEM seine Anerkennung aussprach. Des Weiteren musste die entwickelte Getriebesoftware strenge Beschränkungen hinsichtlich des Ressourcenbedarfs einhalten:

*Im Labor bei Continental Automotive Systems wird das Steuergerät für Doppelkupplungsgetriebe speziellen Qualifikationstests unterzogen.*

- eine maximale CPU-Auslastung von 15 % und
- eine maximale ROM-Belegung von 100 kByte.

Diese Anforderungen konnten mit TargetLink deutlich unterschritten werden, wobei sich der Umfang der automatisch generierten Funktionssoftware wie folgt darstellte:

- Maximal 10 % CPU-Auslastung
- 60 kByte kompilierter Code
- 50 Module
- 20 % ROM-Belegung der kompletten Applikationssoftware

Es wurde deutlich, dass ein Einfrieren der verwendeten Version des Code-Generators spätestens dann erfolgen sollte, wenn mit den Software-Qualitätsmaßnahmen beziehungsweise der Verifikation des Codes begonnen wird. Alle Aktualisierungen der Code-Generator-Version können zu einem geänderten Quellcode führen, was eine erneute Durchführung der Softwaretests und Qualitätsmaßnahmen erforderlich machen würde. Im Rahmen einer hauseigenen Untersuchung durch den OEM wurde festgestellt, dass sich der handgeschriebene Code in puncto Fehlerfreiheit nicht mit dem Code-Generator

messen kann. Durch den Einsatz der modellbasierten Entwicklung und automatischen Code-Generierung wurde eine außergewöhnlich hohe Anzahl terminlich präziser Softwarelieferungen ermöglicht, was dem strukturierten Prozess für diese neue Methodik zu verdanken ist. Dieser Ansatz wurde auf weitere Nutzer an verteilten Standorten als Standardansatz für Getriebeprojekte ausgeweitet.

#### Entwicklung der Funktionssoftware für ein Doppelkupplungsgetriebe

Aufgrund der positiven Erfahrungen beim Einsatz der neuen Methoden und Prozesse wurde 2006 mit der Serienentwicklung von Hardware und Software für ein Getriebesteuergerät eines DCT begonnen (Abbildung 4).

Die komplette Funktionssoftware wird dazu mittels modellbasierter Entwicklung und automatischer Code-Generierung mit TargetLink erstellt. Die Verantwortlichkeiten teilten sich wie folgt auf:

- Der OEM gibt die Funktionsanforderungen vor.
- Continental stellt den modellbasierten Funktionsentwurf zur Verfügung, validiert unter Verwendung von Rapid Control Prototyping.
- Continental als Lieferant des Getriebesteuergeräts stellt die Hardware und Software zur Verfügung (einschließlich des modellbasierten Softwareentwurfs, der automatischen Code-Generierung, des modellbasierten Testens und der Qualitätssicherung).

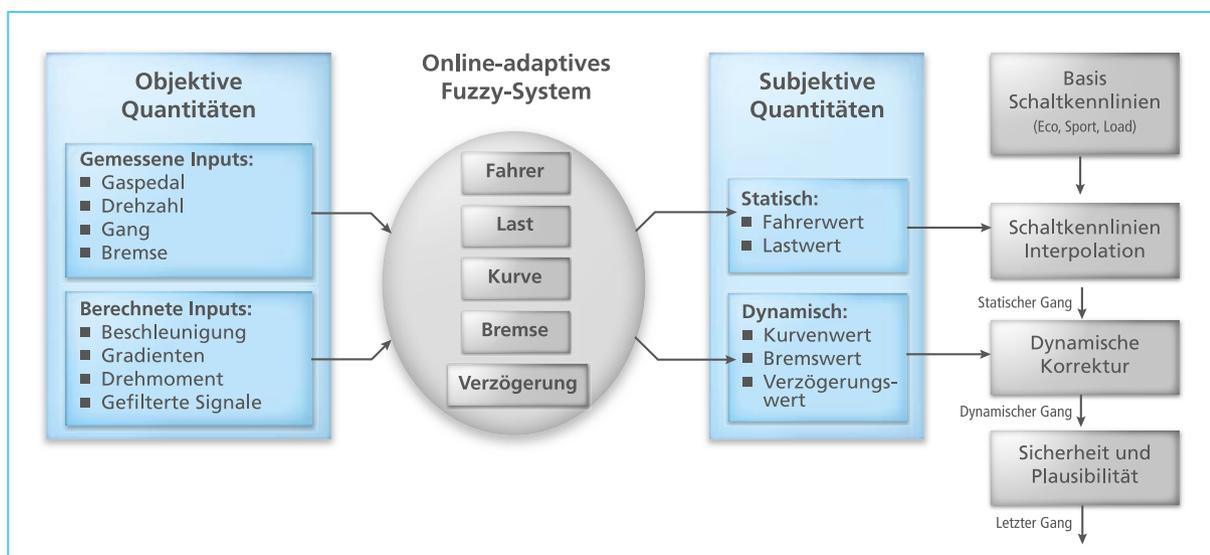


Abbildung 3: Schaltungsstrategie – Struktur und Signalfluss.

- Der OEM und Continental führen die Systemintegration und die Validierung der Funktion im Fahrzeug und mit HIL durch.

Die Handhabung der Prozesselemente, der entsprechenden Methoden zur modellbasierten Entwicklung und der automatischen Code-Generierung für dieses DCT-Projekt verliefen ähnlich dem ersten Serienprojekt. Während der Laufzeit des Projekts ergaben sich aufgrund komplexer DCT-Funktionen einige weitere Herausforderungen:

- Wiederverwendung von Funktionen
- Angepasste Code-Generierung für geräteeigene Kalibriermethode
- Verteilte Entwicklung in einer Multi-User-Umgebung

Die Umsetzung dieser Herausforderungen soll in den nächsten drei Abschnitten kurz beschrieben werden.

### Wiederverwendung von Funktionen

Analysiert man das DCT-System, wird deutlich, dass zwei Kupplungen und vier Gangkomponenten eine Wiederverwendung von Funktionen wünschenswert machen (Abbildung 4),

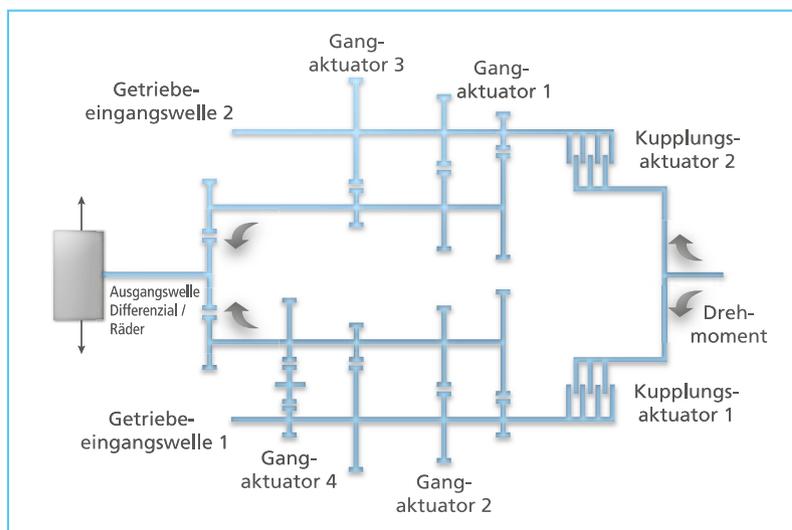


Abbildung 4: DCT-Schema.

„Durch den Einsatz der modellbasierten Entwicklung und automatischen Code-Generierung wurde eine außergewöhnlich hohe Anzahl terminlich präziser Softwarelieferungen ermöglicht.“

Georg Grassl, Continental AG

was durch begrenzte Hardwareressourcen noch verstärkt wird. Aktuatorfunktionen für zwei Kupplungen, zwei Wellen und vier Gangsteller müssen einen gemeinsamen Algorithmus nutzen. Im Projekt werden außerdem Simulink-Bibliotheken für Low-Level-Funktionen (z. B. Filterroutinen) eingesetzt, woraus eine „geschachtelte Wiederverwendung“ resultiert. Diese Anforderungen wurden durch das Function Reuse Feature von TargetLink erfüllt, wodurch der Ressourcenbedarf reduziert werden konnte. Das Bewusstsein für diese Wiederverwendung von Funktionen wurde zudem bereits für den Entwurf der Modellarchitektur zu einem sehr wichtigen Aspekt.

### Angepasste Code-Generierung für geräteeigene Kalibriermethode

In diesem Projekt sollte eine spezifische Kalibriermethode für Steuergerä-

te ohne Kalibriervorrichtung/ Speichererweiterungen implementiert werden, um Kosten bei der Entwicklung zu sparen. Dies hat zur Folge, dass ein Satz Kalibrierdaten als Strukturen organisiert ist und dass der Zugriff darauf zur Laufzeit erfolgt, indem die Pointer auf die Strukturen vom ROM auf das RAM umgeleitet werden. Zur Implementierung der hierfür benötigten Code-Pattern wurde einerseits auf TargetLink-Features wie Varianten-Codierung und Template-Mechanismen zurückgegriffen (Abbildung 5), andererseits in Zusammenarbeit mit dSPACE eine Änderung am Code-Generator vorgenommen. Die Umstellung des kompletten DCT-Modells auf diese spezifische Kalibriermethode erfolgte dann innerhalb einer einzigen Woche, was nochmals die Leistungsfähigkeit der von dem Code-Generator gelieferten offenen Programmierschnittstelle belegt.

### Verteilte Entwicklung in einer Multi-User-Umgebung

Für dieses Projekt erfolgte der funktionale Entwurf sowohl durch Continental als auch den beteiligten OEM. Die hohe Anzahl der involvierten Entwickler sprengte den bisher genutzten Single-User-Ansatz, weshalb die entwickelten Prozesse im Hinblick auf Multi-User-Fähigkeiten erweitert wurden. Konkret wurden umfangreiche Funktionalitäten in kleinere Modellfragmente zerlegt und in Simulink-Modellbibliotheken verwaltet. Um die Daten innerhalb der Umgebung konsistent zu halten, wurde auf die Multi-User-Unterstützung des dSPACE Data Dictionarys in Form von Include Files

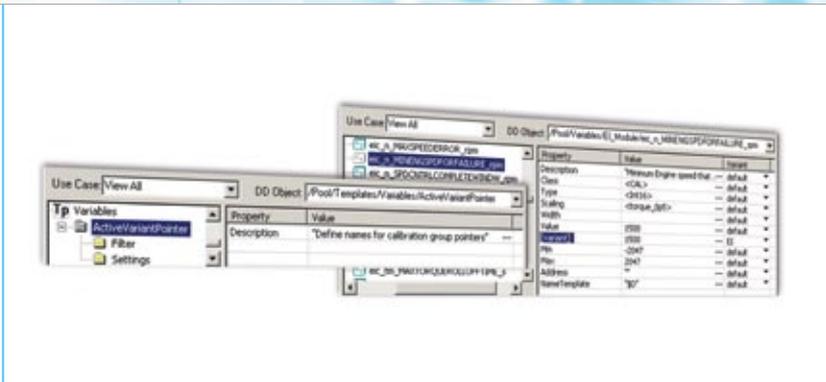


Abbildung 5: Definition eines Pointers auf Datenvarianten durch Verwendung eines Variablen-Templates und Zuweisung von Varianten zu Parametern mit TargetLink.

zurückgegriffen. Das Getriebe-Team entschied sich, das Modell, die Datenbeschreibung, die Testvektoren (funktional und strukturell), die Testberichte und die Richtlinien zur Feinabstimmung als Paket (Abbildung) im Konfigurationsmanagement bereitzuhalten. Auf diese Weise ist der Weg für eine Wiederverwendung in anderen Projekten und eine konsistente Entwicklung geebnet.

### Ergebnisse und wesentliche Erkenntnisse

Dank der zuvor entwickelten Prozesse und Methoden konnten 100 % der Funktionssoftware des DCT-Projekts modellbasiert entwickelt und mit TargetLink autocodiert werden. Der Umfang der in diesem Hochvolumenprojekt entstandenen Funktionssoftware belief sich dabei auf

- 250 kByte kompilierten Code und
- 120 Modellbibliotheken.

Unter Berücksichtigung der oben erläuterten Herausforderungen und entsprechenden Lösungen war eine vertrauensvolle Zusammenarbeit mit dSPACE als Lieferanten des Code-Generators wichtig. Dies galt insbesondere für spezielle Support-Anfragen vom Projektteam hinsichtlich Speicherzuweisung, Datenvarianten und Robustheit des Code-Generators. Außerdem war im Hinblick auf die Spezialanforderungen eine Synchronisierung von Updates und Patches für den Code-Generator mit den Softwareabgaben des Projekts notwendig.

### Ausblick

Aufbauend auf dem Know-how, das sich durch die Nutzung des projektspezifischen Ansatzes für einen modellbasierten Entwurf und die automatische Code-Generierung ergeben hat, ist das Getriebe-Team nun in der Lage, diese Methodik auf Low-Level-Software, also auf Aktuatorsteuerungen, zu erweitern. Darüber hinaus wird derzeit der Schritt von einem projektspezifischen Ansatz hin zu einem Plattformansatz bei gleichzeitiger Aufrechterhaltung agiler Methoden entwickelt. Auf diese Weise kann eine konzerninterne weltweite Einführung unterstützt werden, wobei konsistente Entwicklungsmethoden sichergestellt sind. ■

*Georg Grassl  
Business Unit Transmission  
Gerd Winkler  
Business Unit Engine Systems  
Continental AG  
Deutschland*

## Fazit

Der Geschäftsbereich Getriebe implementiert einen durchgängigen Ansatz für die modellbasierte Entwicklung und automatische Code-Generierung. Die Entwicklungsphasen sind durch eine ausführbare Spezifikation (ein Modell) miteinander verknüpft. Währenddessen wird die komplette Funktionssoftware für Getriebesteuerungen unter Verwendung eines modellbasierten Entwurfs und automatischer Code-Generierung realisiert. In Zusammenarbeit mit den OEMs ist der Geschäftsbereich Getriebe in der Lage, einen effizienten Prozess für die Entwicklung von Funktionssoftware einzurichten.

Die Toolkette für diese Methode hat sich bereits für Hochvolumenprojekte bewährt. Der Prozess basiert auf bewährten Elementen von Projekten mit konventioneller Entwicklung, die im Hinblick auf die Anforderungen des modellbasierten Entwurfs und der automatischen Code-Generierung angepasst wurden. Hierdurch wurde eine effiziente Einführung der neuen Methodik möglich gemacht. Diese erfolgte schrittweise vom ersten internen Projekt zur Implementierung der neuen Methoden über ein erstes Serienprojekt mit einem 20 %-Anteil der autocodierten Applikationssoftware bis zum hochkomplexen DCT-Getriebe mit 100 % automatisch generiertem Applikationscode.

Durch die Anwendung des neuen Verfahrens wurde eine sofortige Qualitätsverbesserung erzielt. Werden Projekte identischen Umfangs miteinander verglichen, wobei auf der einen Seite traditionelle Entwicklungsverfahren eingesetzt werden und auf der anderen Seite modellbasierte Entwicklung mit den sich daraus ergebenden Optionen (Rapid Prototyping, automatische Seriencode-Generierung, modellbasiertes Testen), so zeigt sich eine klare Überlegenheit der modellbasierten Entwicklung gegenüber den traditionellen Verfahren.

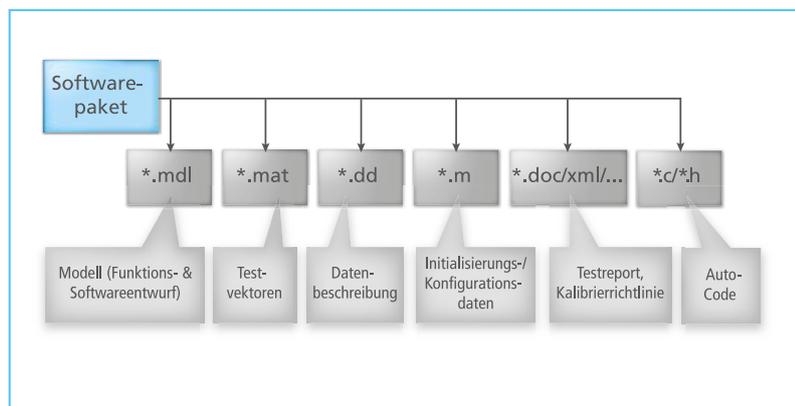


Abbildung 6: Softwarepakete für Multi-User-Umgebungen.