



Integration textueller Anforderungen und Modell-basiertem Testen mit SysML

Oliver Alt

oliver.alt@contiautomotive.com

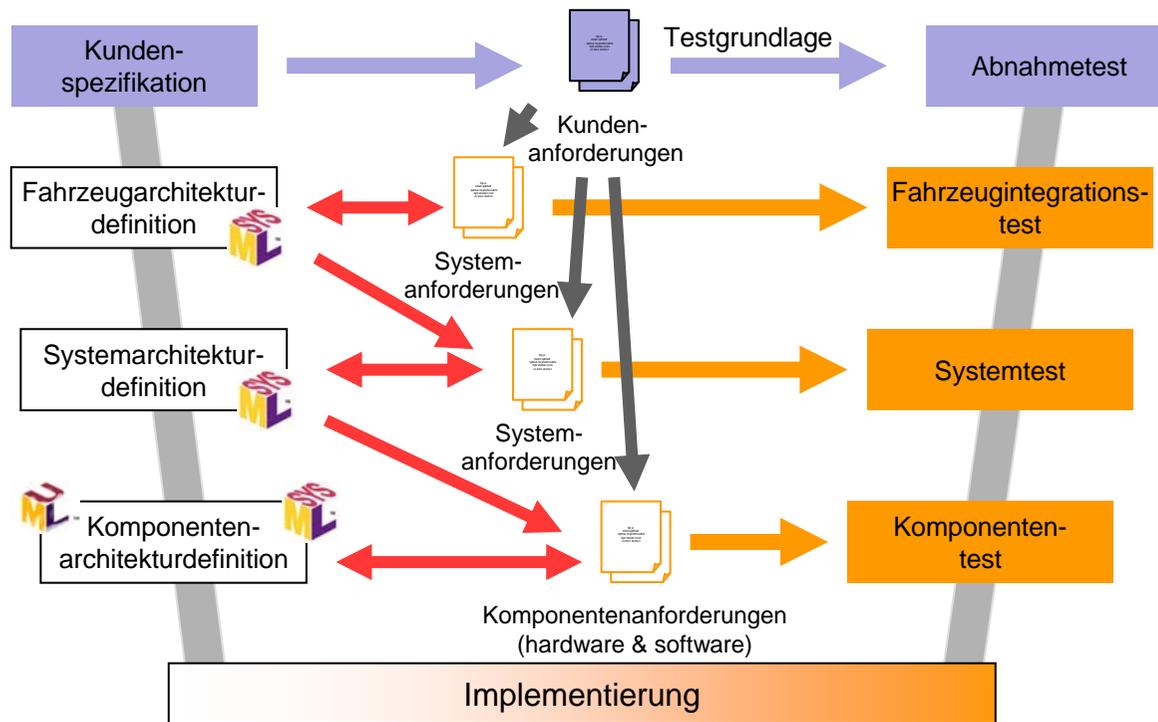
Divisionen und Geschäftsbereiche der Continental

Continental-Konzern					
Chassis & Safety	Powertrain	Interior	Pkw-Reifen	Nfz-Reifen	ContiTech
<ul style="list-style-type: none"> ▶ Electronic Brake Systems ▶ Hydraulic Brake Systems ▶ Sensorics ▶ Passive Safety & ADAS ▶ Chassis Components 	<ul style="list-style-type: none"> ▶ Engine Systems ▶ Transmission ▶ Hybrid Electric Vehicle ▶ Sensors & Actuators 	<ul style="list-style-type: none"> ▶ Body & Security ▶ Connectivity ▶ Commercial Vehicles & Aftermarket ▶ Instrumentation & Displays ▶ Interior Modules ▶ Multimedia 	<ul style="list-style-type: none"> ▶ Erstausrüstung ▶ Ersatzgeschäft Europa ▶ Ersatzgeschäft The Americas ▶ Ersatzgeschäft Asien ▶ Zweiradreifen 	<ul style="list-style-type: none"> ▶ Lkw-Reifen Europa ▶ Lkw-Reifen The Americas ▶ Lkw-Reifen Asien ▶ Industriereifen 	<ul style="list-style-type: none"> ▶ Air Spring Systems ▶ Benecke-Kaliko Group ▶ Conveyor Belt Group ▶ Elastomer Coatings ▶ Fluid Technology ▶ Power Transmission Group ▶ Vibration Control ▶ Other Operations

Division Chassis & Safety – Geschäftsbereiche

Electronic Brake Systems	Hydraulic Brake Systems	Sensorics	Passive Safety & ADAS	Chassis Components
<ul style="list-style-type: none"> ▶ Hydraulisch-elektronische Steuerungen (HECU) <ul style="list-style-type: none"> - ABS - ESC ▶ Funktionen: <ul style="list-style-type: none"> - Abstandsregelung - Regeneratives Bremsensystem - Aktive Vorderachslenkung - Berganfahrhilfe, Bergabfahrhilfe - Reifendruck-Kontrollsystem - Hydraulischer Bremsassistent - Anhängerstabilisierung 	<ul style="list-style-type: none"> ▶ Bremssättel und -scheiben ▶ Trommelbremsen ▶ Bremsschläuche ▶ Bremskraftverstärker ▶ Tandem-Hauptzylinder ▶ Elektrische Parkbremsen ▶ Pedalmodule ▶ Bremsdruckregler 	<ul style="list-style-type: none"> ▶ Lenkwinkel- & Lenkmomentsensoren ▶ Raddrehzahlsensoren ▶ Motor-/Getriebedrehzahl- und Positionssensoren ▶ Gurtschlosssensoren ▶ Fahrwerksensoren ▶ Sensorcluster 	<ul style="list-style-type: none"> ▶ Airbag-Einheiten ▶ Hybrid-Gateways ▶ Pre-Crash-Sensoren ▶ Kraftsensoren-schnittstellen ▶ Insassenklassifizierungssysteme (OCS) ▶ Fahrerassistenzsysteme <ul style="list-style-type: none"> - Radar - Lidar - Kamera 	<ul style="list-style-type: none"> ▶ Fahrwerkelektronik (Tier 1-Kunden) ▶ Elektrische Servolenkung ▶ Scheibenwaschsysteme ▶ Federungssysteme ▶ Fahrwerkelektronik 

Anforderungen sind die Brücke zwischen Entwicklung und Test



► Anforderungen in Form von Lasten- und Pflichtenheften bilden heutzutage das verknüpfende Element innerhalb der Systementwicklung

► Neben der Anforderungen existieren noch weitere, teils formale Entwicklungsdokumente

- Architekturmodell
- Verhaltensbeschreibungen
- Modellierete Testfälle

Modell-basierte Entwicklung

- ▶ Im Rahmen einer Modell-basierten Entwicklung steht ein (formales) Modell im Mittelpunkt der Entwicklung
 - ▶ Modell ist das zentrale Element
- ▶ Ein solches Modell dient dann als Grundlage weiterer Entwicklungsartefakte
 - ▶ Code (Modell-basierte Codegenerierung) oder
 - ▶ Testfälle (Modell-basierter Test)

- ▶ Beispiele für formale Systemmodelle:

- ▶ Matlab/Simulink Modelle



- ▶ Statecharts

- ▶ Specification and Description Language (SDL)



- ▶ ...

Lücke zwischen Anforderungen und formalen Modellen

- Im Rahmen der Modell-basierten Entwicklung gibt es eine Lücke zwischen Anforderungen und den formalen Modellen
- Die Anforderungen müssen durch den Entwicklungsingenieur formalisiert werden
- Aufgrund des informellen Charakters der Anforderungen keine automatische Formalisierung möglich



Systems Modelling Language (SysML)

▶ SysML ist eine standardisierte, grafische Sprache zur Systemmodellierung (*Systems Engineering*)

▶ Relativ neu: Version 1.0 verabschiedet im Mai 2007



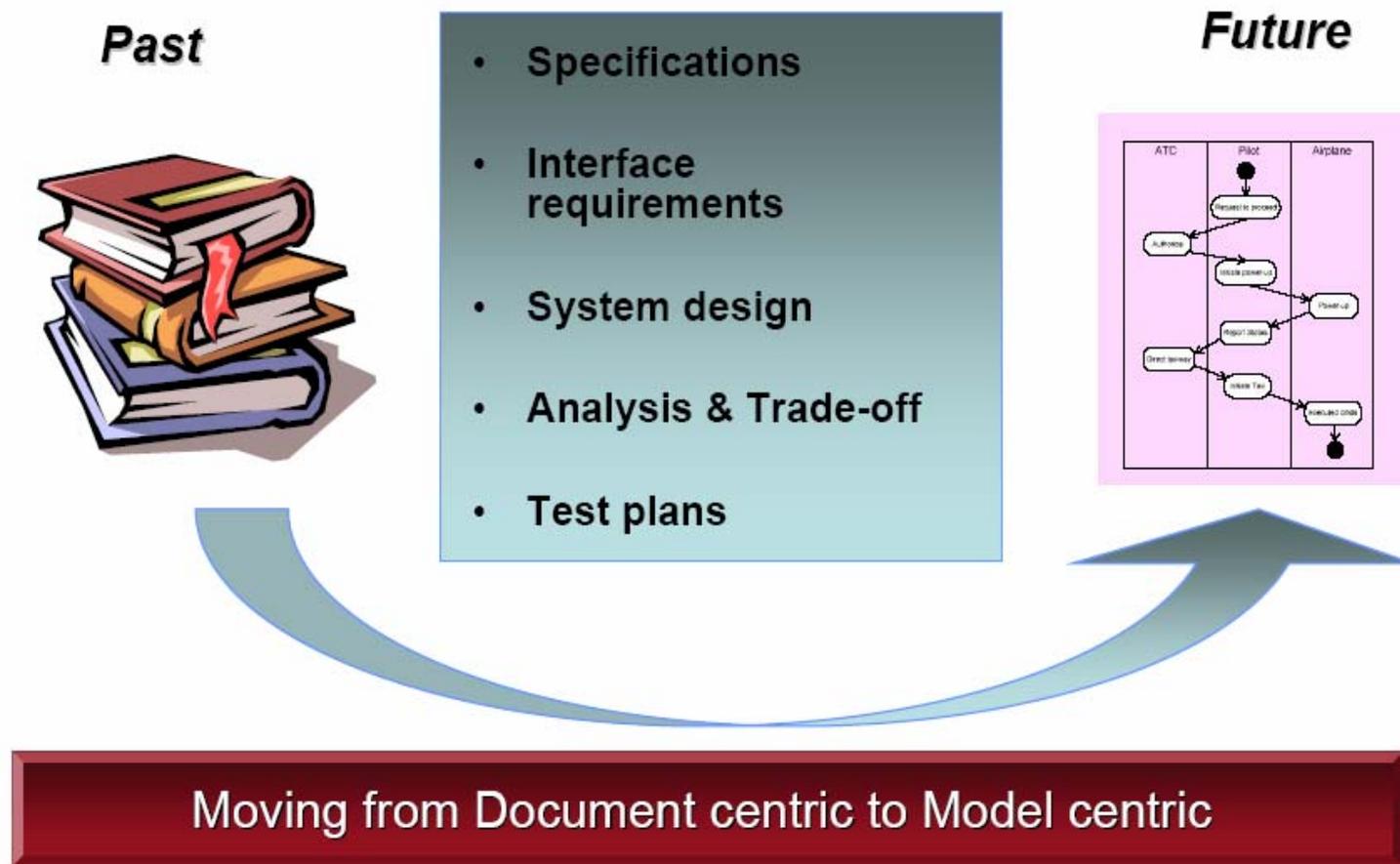
▶ Benutzt große Teile der Softwaremodellierungssprache UML und erweitert diese an einigen Stellen

▶ Bewährte Konzepte aus der Systemtechnik fanden Eingang in die Sprache

▶ Im Gegensatz zu UML mit 13 Diagrammarten besitzt SysML nur 9 und ist damit leichter anwendbar

▶ Durch Spracherweiterung, so genannte Profile, an domänenspezifische Aufgaben anpassbar

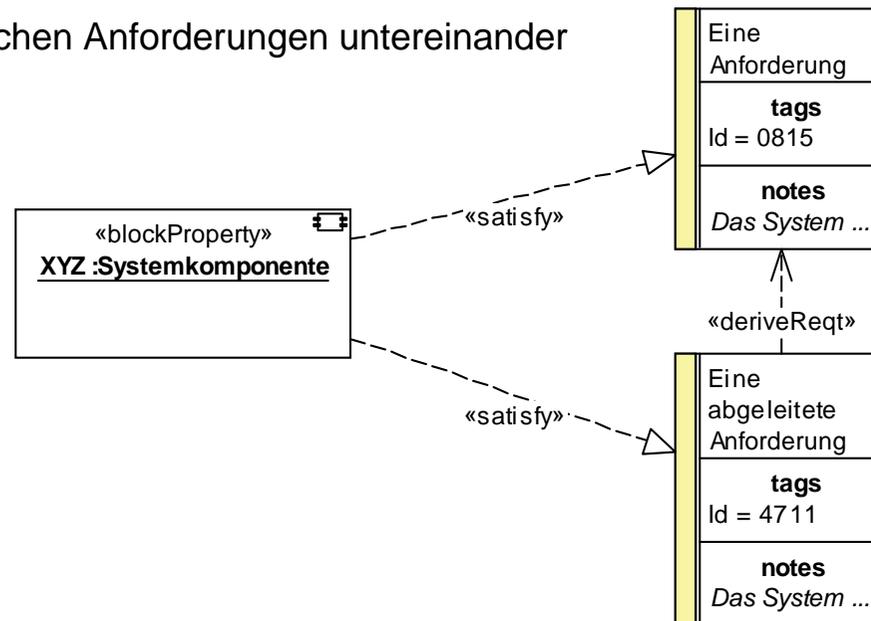
▶ Werkzeugunterstützung vorhanden



Quelle: OMG

Lose Kopplung zwischen Anforderungen und Modell durch SysML

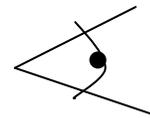
- ▶ SysML standardisiert erstmals ein Modellelement zur Repräsentation von Anforderungen im Modell
 - ▶ Neuer Diagrammtyp: *Requirements Diagram*
- ▶ Damit lassen sich Anforderungen mit beliebigen andern Modellelementen in Beziehung setzen
 - ▶ Traceability zwischen Architektur und Anforderungen
 - ▶ Traceability zwischen Anforderungen untereinander



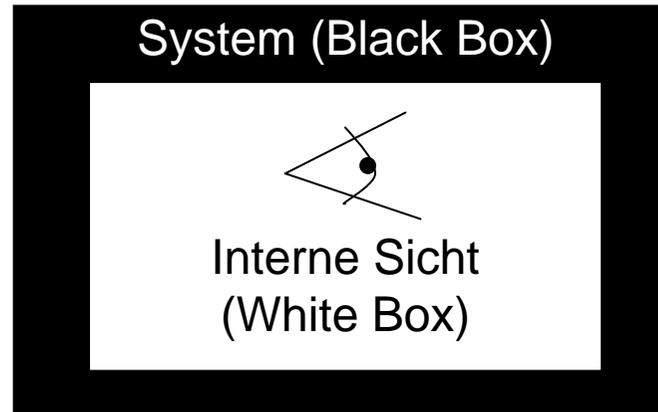
Stärkere Kopplung durch Verhaltensmodellierung

- ▶ Funktionale Anforderungen beschreiben Systemverhalten
- ▶ SysML bietet gleichfalls die Möglichkeit Verhalten zu modellieren
- ▶ Wie können Anforderungen/Anwendungsfälle möglichst einfach in formale Modelle überführt werden ?
- ▶ Verfahren aufgrund einer Analyse von Anforderungen, Anwendungsfallbeschreibungen und Systemmodellen entwickelt
 - ▶ Untersuchung von Abhängigkeiten und Gemeinsamkeiten
- ▶ Tempomat als Beispielsystem
 - ▶ Eine durch den Fahrer eingestellte Wunschgeschwindigkeit konstant halten

Externe und interne Systemsicht



Externe
Sicht

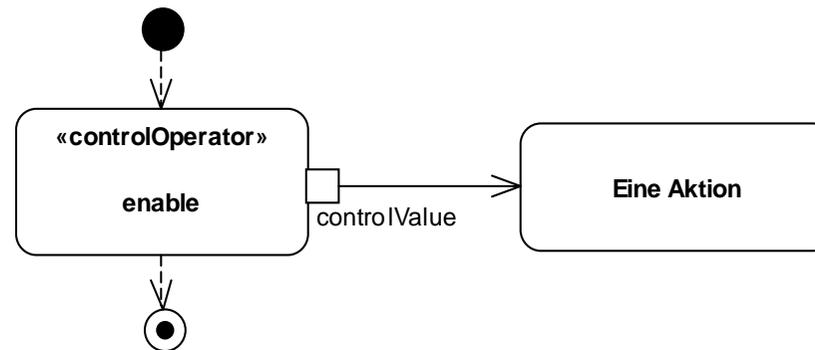


- ▶ **Externe Sicht** beschreibt das System aus Benutzersicht
 - ▶ Kommunikation nur über Schnittstellen
 - ▶ Beschreibung was der Benutzer wann mit dem System machen darf ?
- ▶ **Interne Sicht** beschreibt das interne Systemverhalten
- ▶ **Verknüpfungen zwischen beiden Sichten**
 - ▶ Welchen Einfluss hat externes Verhalten auf internes und umgekehrt

Verhaltensmodellierung mit SysML-Aktivitätsdiagrammen

- ▶ Zur Verhaltensmodellierung werden SysML-Aktivitätsdiagramme verwendet
 - ▶ Implizite Darstellung von Zuständen – nicht explizit wie bei Zustandsmodellierung
 - ▶ Anforderungen und Anwendungsfälle nutzen Aktivitäten um das Verhalten von Systemen zu beschreiben
- ▶ Die Aktivitätsmodellierung in SysML erweitert die UML-Aktivitätsdiagramme noch einmal
- ▶ Der so genannter Kontrolloperator ermöglicht nun die Beeinflussung des Laufzeitverhaltens anderer Aktionen von außen

- ▶ Starten
- ▶ Stoppen
- ▶ Anhalten
- ▶ Fortsetzen
- ▶ ...

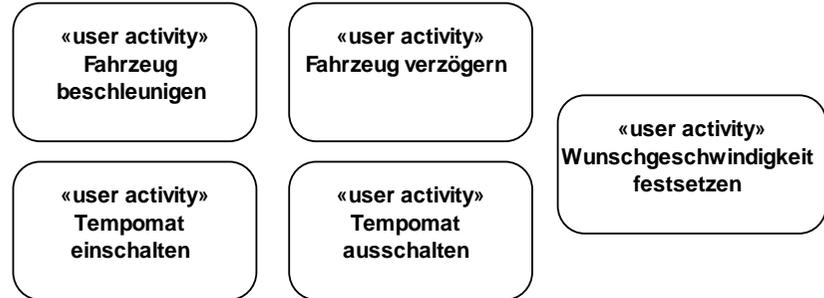


Benutzer- und Systemaktionen

Definition von zwei Arten von Aktionen

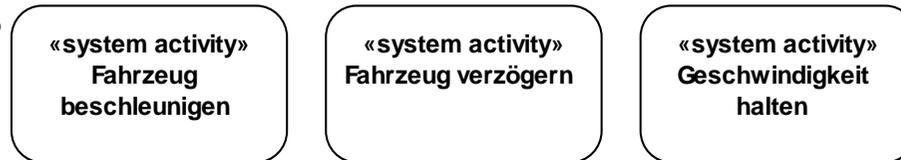
Benutzeraktionen

Was macht der Systembenutzer mit dem System ?



Systemaktionen

Was macht das System intern ?



Weiterhin halten **Systemgrößen** aktuelle Zustandswerte des Systems fest

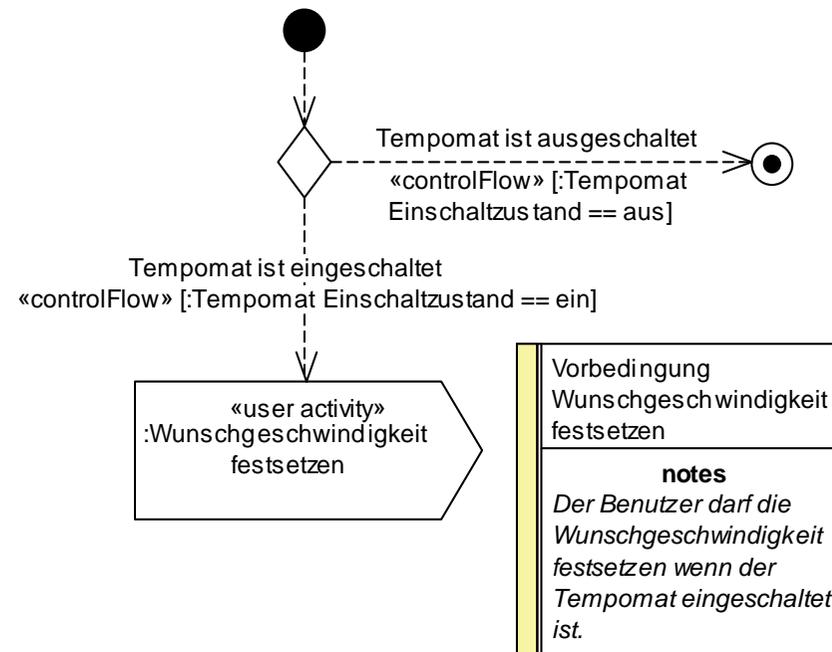
«valueType» Fahrzeuggeschwindigkeit	«enumeration» Tempomat Einschaltzustand	«valueType» Wunschgeschwindigkeit
	«enum» ein aus	

Modellierungsmuster

- ▶ Mit Hilfe der definierten Modellelemente lässt sich nun das Systemverhalten modellieren
 - ▶ **Vorbedingungen** für die Ausführung der Benutzeraktionen (z.B. Der Tempomat muss eingeschaltet sein bevor die Wunschgeschwindigkeit eingestellt werden kann)
 - ▶ Eine **Verknüpfung zwischen Benutzer- und Systemaktionen** (Wie wirkt sich die Ausführung einer Benutzeraktivität auf das System aus ?)
 - ▶ Die **interne Systembeschreibung** durch die Systemaktionen (internes, eigenständiges Verhalten des Systems)
- ▶ Die Modellierung kann dabei immer nach dem gleichen Schema mit Hilfe von Modellierungsmustern erfolgen
- ▶ Jedem Muster lassen sich dann bestimmte Anforderungen zuordnen oder Anforderungen daraus ableiten

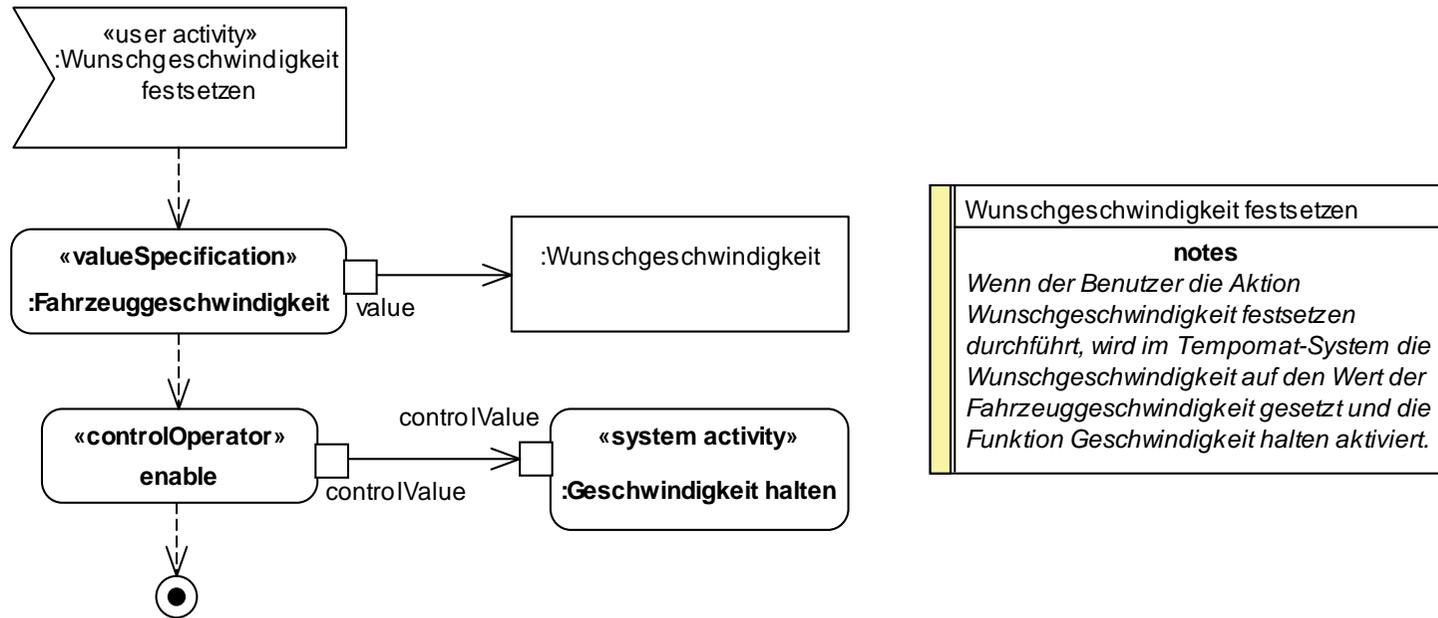
Muster 1: Modellierung von Vorbedingungen

- ▶ Vorbedingung zur Benutzeraktion **Wunschgeschwindigkeit festsetzen**



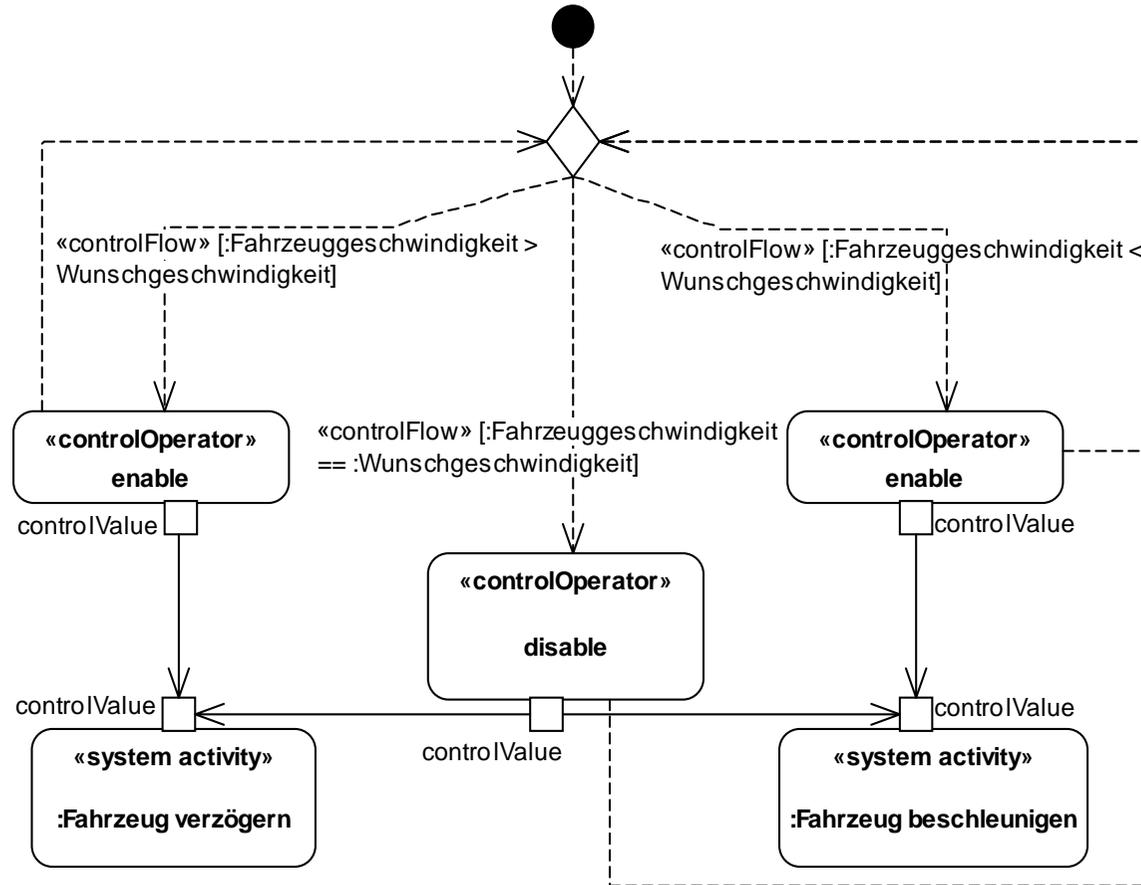
- ▶ Beschreibung der Vorbedingung durch bedingte Kontrollflüsse als Aktivitätsdiagramm
- ▶ Verwendung der **SendSignal**-Action zur Auslösung der Benutzeraktivität bei erfüllter Vorbedingung

Muster 2: Verknüpfung von externem und internen Verhalten



- ▶ Ereignisempfänger reagieren auf die durch die Vorbedingungen geschickten Signale
- ▶ Der neue SysML-Kontrolloperator ermöglicht die Beeinflussung des Laufzeitzustandes von Aktivitäten von außen
- ▶ Verknüpfungen beeinflussen Aktivitäten und Objekte der **internen** Systembeschreibung

Muster 3: Internes Systemverhalten



Fahrzeuggeschwindigkeit unter Wunschgeschwindigkeit
notes <i>Liegt die Fahrzeuggeschwindigkeit unter der Wunschgeschwindigkeit, wird die Funktion Fahrzeug beschleunigen aktiviert.</i>

Fahrzeuggeschwindigkeit über Wunschgeschwindigkeit
notes <i>Liegt die Fahrzeuggeschwindigkeit über der Wunschgeschwindigkeit, wird die Funktion Fahrzeug verzögern aktiviert.</i>

Fahrzeuggeschwindigkeit gleich Wunschgeschwindigkeit
notes <i>Ist die Fahrzeuggeschwindigkeit gleich der Wunschgeschwindigkeit, werden sofern aktiv die Funktionen Fahrzeug verzögern und Fahrzeug beschleunigen deaktiviert.</i>

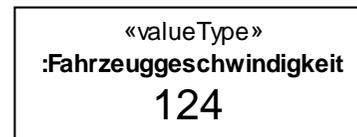
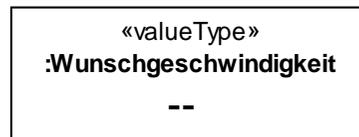
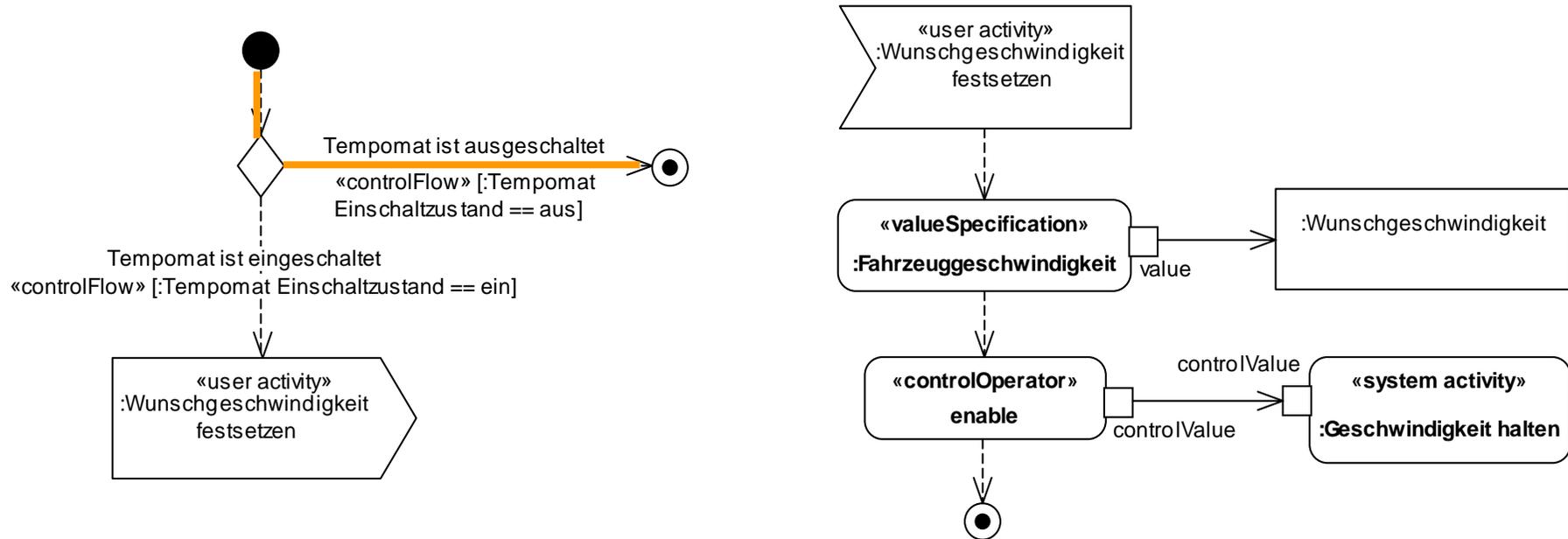
- ▶ Interne Beschreibung der Systemaktion **Geschwindigkeit halten**
- ▶ Modellierung durch Objekte und Aktionen

Anmerkungen zum Verfahren

- ▶ Die gezeigten Anforderungen sind etwas idealisiert formuliert um die Äquivalenz zu den Verhaltensdiagrammen zu verdeutlichen
- ▶ Jedoch lassen sich mit einiger Übung im Umgang mit den Mustern auch anders formulierte Anforderungen leicht umsetzen
- ▶ Mit Hilfe der sukzessiven Formalisierung der Anforderungen lassen sich oftmals auch Lücken und Ungenauigkeiten innerhalb der Anforderungen erkennen und dadurch die Qualität der Anforderungen verbessern

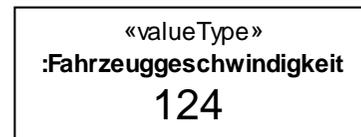
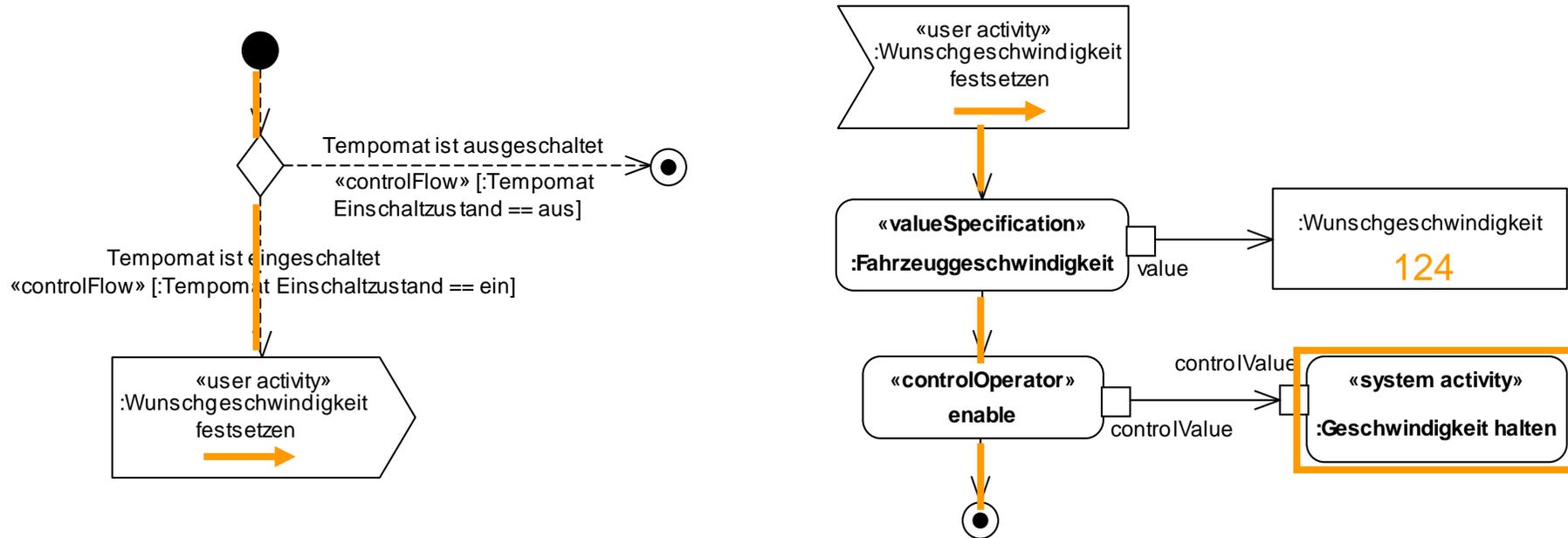
Modell-basierte Testfallgenerierung

- ▶ Aus einem derartig aufgebauten Modell lassen sich automatisch der Zustandsraum des Systems aufbauen und letztendlich Testfälle generieren



Modell-basierte Testfallgenerierung

- Aus einem derartig aufgebauten Modell lassen sich automatisch der Zustandsraum des Systems aufbauen und letztendlich Testfälle generieren



Aufbau des Zustandsraumes

- ▶ Sukzessive Ausführung der Benutzeraktionen
- ▶ Vergleich des Systemzustandes vor und nach der Ausführung
 - ▶ Ein Systemzustand setzt sich aus den Werten der Systemgrößen und den Laufzeitzuständen der Systemaktionen zusammen
- ▶ Wird durch eine Ausführung einer Benutzeraktion ein neuer Zustand erreicht wird dieser durch eine Transition an den vorherigen angefügt
- ▶ Die Ausführung wird rekursiv mit dem neuen Ausgangszustand fortgesetzt

Zusammenfassung

- ▶ Durch die Integration von textuellen Anforderungen in die SysML-Sprache wird die Lücke zwischen Requirements Engineering und Modell-basierter Entwicklung ein Stück weit geschlossen
- ▶ Die vorgestellten Modellierungsmuster erlauben darüber hinaus eine strukturierte Formalisierung von Anforderungen mit Hilfe von SysML-Aktivitätsmodellierung
- ▶ In früheren Arbeiten wurde bereits gezeigt wie aus solchen Verhaltensmodellen Testfälle automatisch generiert werden können
- ▶ Auch die Ableitung von Anforderungen aus dem Modell ist denkbar
- ▶ Aktuell wird an einem Prototyp eines solchen Werkzeuges im Umfeld des verwendeten SysML-Werkzeuges Sparx Systems Enterprise Architect gearbeitet



Fragen...

