Requirements Engineering for embedded systems

BOSCH ETAS Engineering Tools Telelogic TUM ZF OPEL

Modellbasierte Softwareentwicklung für automobilspezifische Steuergerätenetzwerke

Christian Schröder

Telelogic Deutschland GmbH

Bielefeld

http://www.forsoft.de/automotive/

„Requirements is a lot harder than it used to be (because we built all the easy systems a long time ago).“

2001 by Tom DeMarco: The Atlantic System Guild

Anforderungen an die automobile Systementwicklung

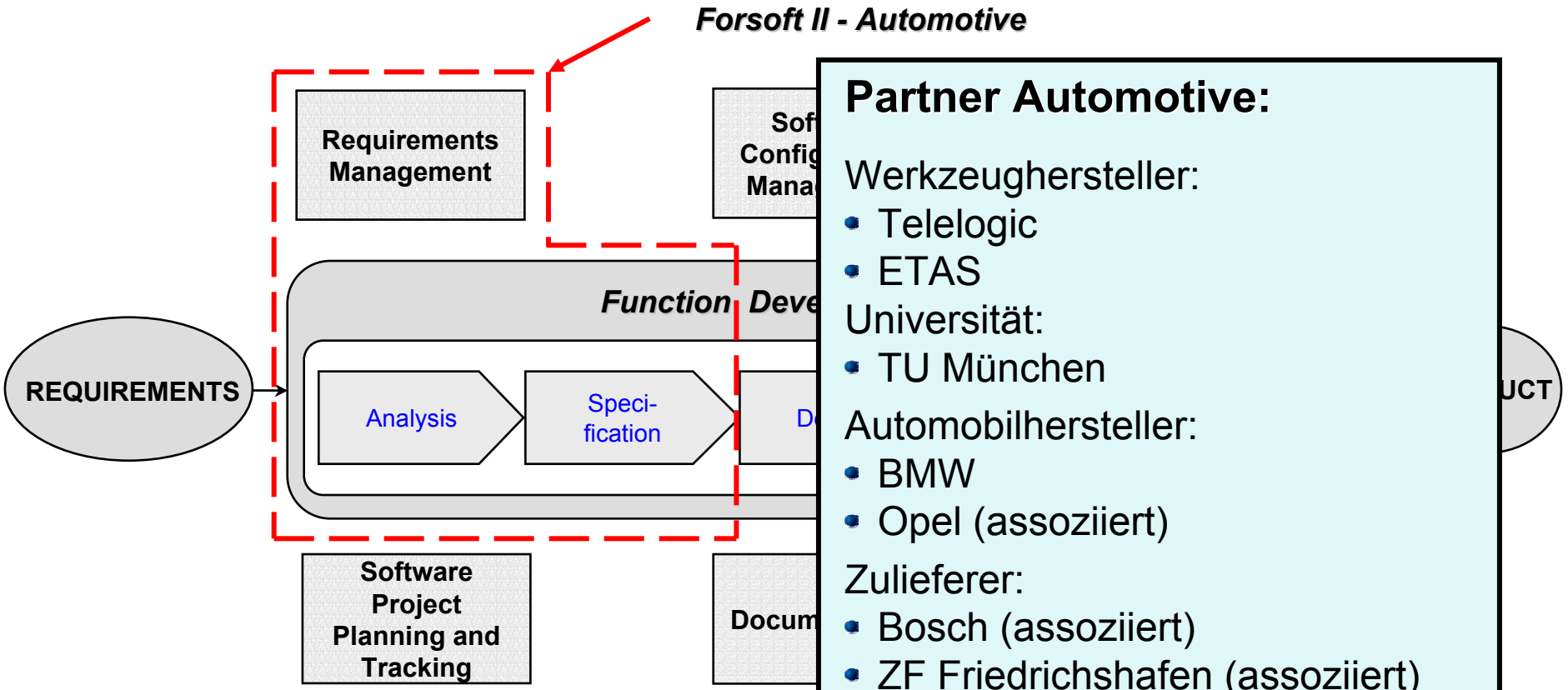
Verteiltheit und Interoperabilität in Steuergerätenetzwerken
- Verteilte Realisierung von Gesamtfunktionalitäten
- Zunahme der Abhängigkeiten

Kurze Entwicklungszeiten
Paralleles Entwickeln von
- Steuergeräten, und
- Funktionen innerhalb eines Steuergerätes

Arbeitsteiliges Vorgehen
- Anforderungskatalog vom Hersteller
- Realisierung vom Zulieferer
- Abnahme und Integration vom Hersteller

Wettbewerbsentscheidende Innovationen
- Zunahme des Stellenwerts eingebetteter Systeme für den Kunden (vgl. DSC, ACC)

Fokus Projekt Automotive



Partner Automotive:

Werkzeughersteller:
- Telelogic
- ETAS

Universität:
- TU München

Automobilhersteller:
- BMW
- Opel (assoziiert)

Zulieferer:
- Bosch (assoziiert)
- ZF Friedrichshafen (assoziiert)

Automotive Lösungsansatz

Unterstützung des Strukturierungsprozesses von Anforderungen:

„informell“ beschriebene Anforderungen ➡ modellbasierte Anforderungen

- Differenzierung in Informationsklassen
- Strukturierung der Funktionen
- Schrittweise Spezifikation der Funktionen

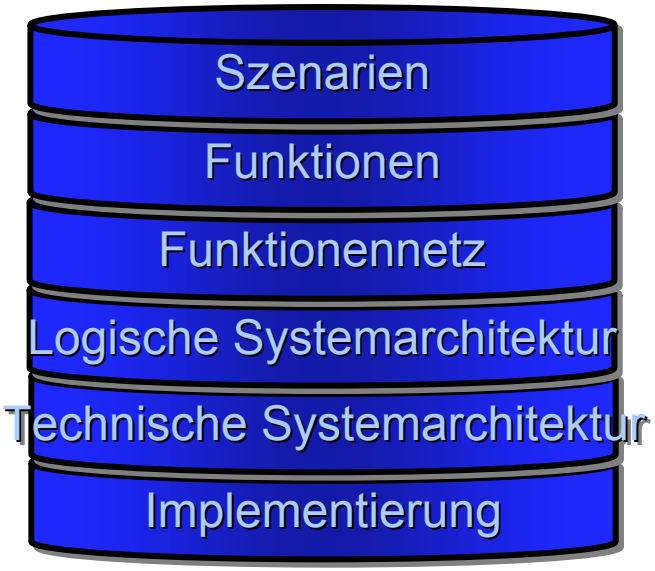
Übergang von Analysemodellen ins abstrakte Design

- Variantenbildung und Instanziierung von Funktionen zu Funktionennetzen
- Abbildung der Funktionen auf ein abstraktes Systemmodell

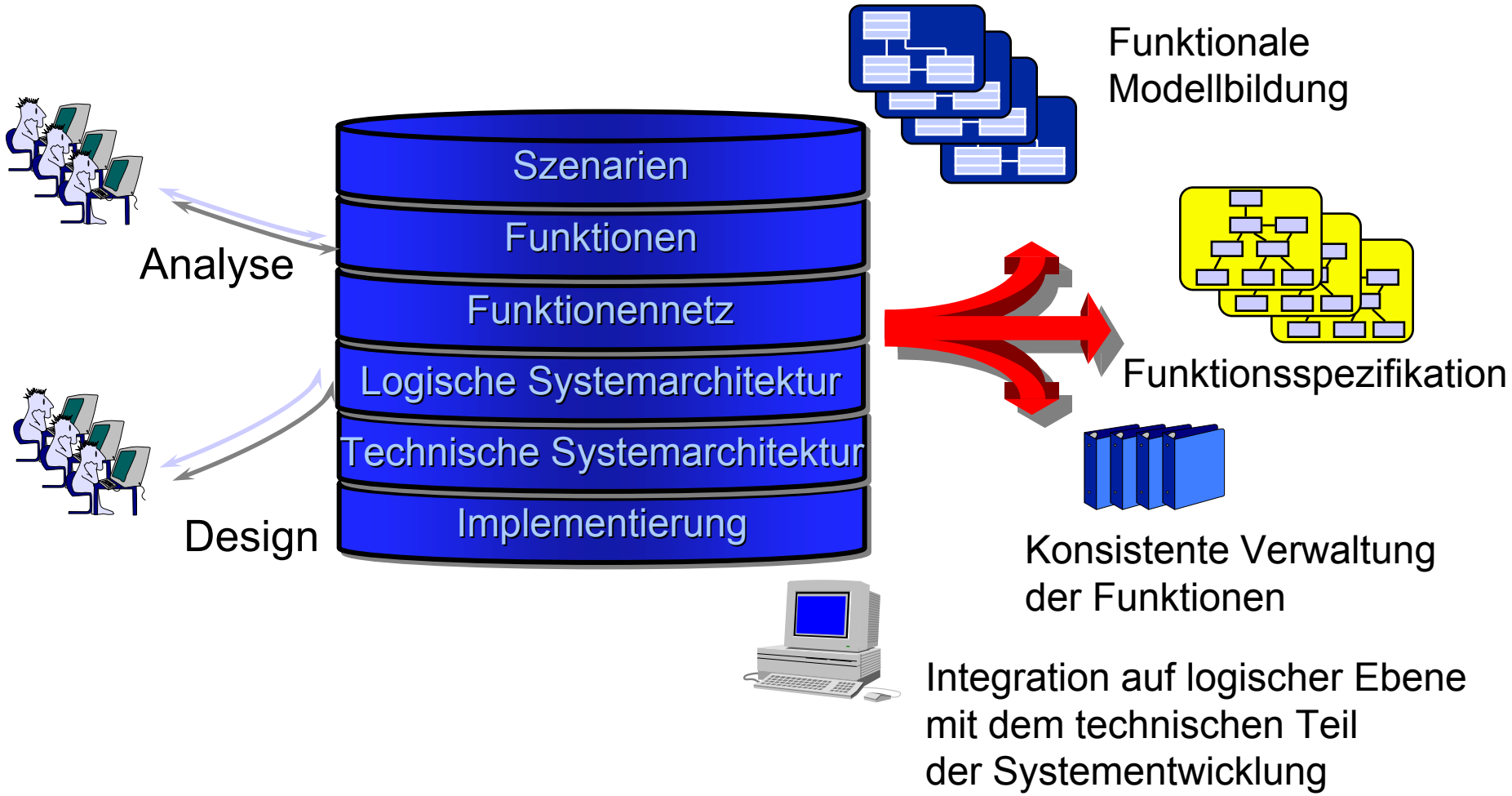
Konsistente Verwaltung der Anforderungen

- Traceability- und Impact Analysis
- Change Management
- Configuration Management

Automotive – integriertes Informationsmodell

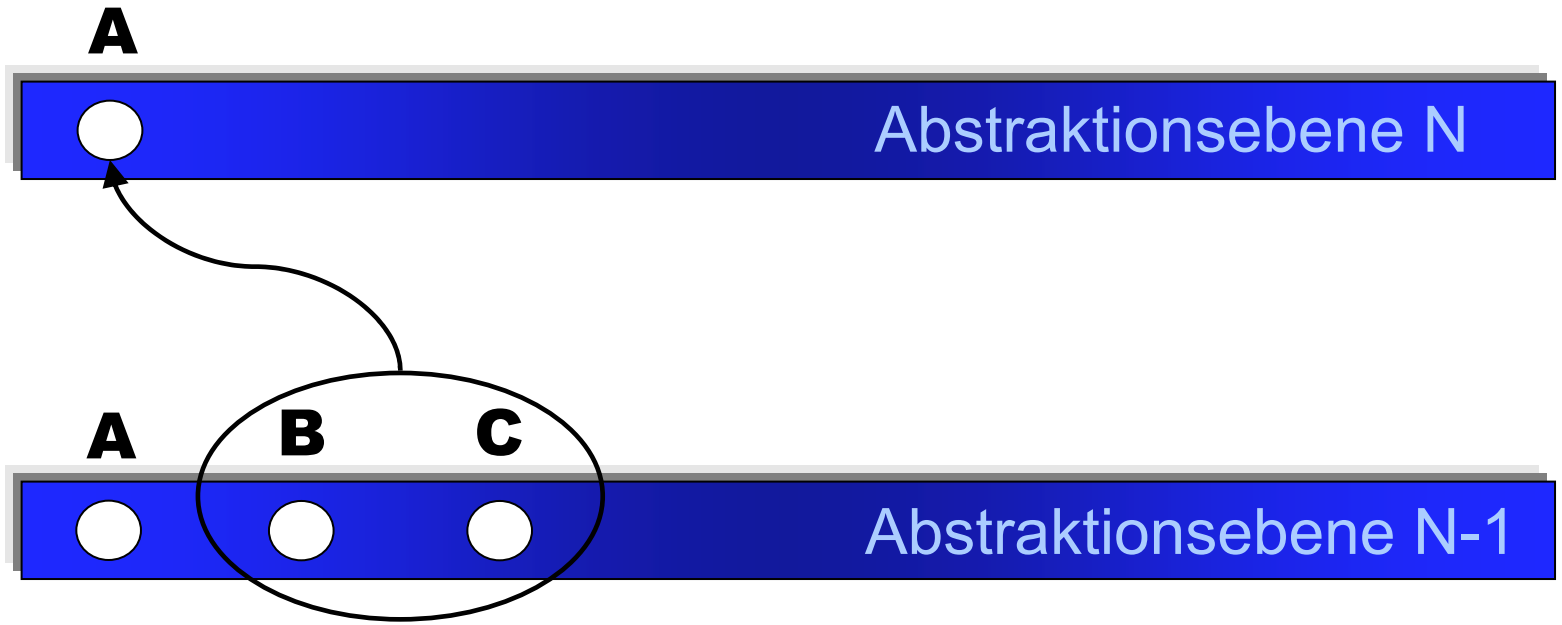


Automotive – Modellbasierte Systementwicklung



Abstraktionsebenen

Einschränkende Systemsichten



Konzepte B und C erklären Konzept A

Automotive – Werkzeugunterstützung

Szenarien	DOORS, UML-Suite
Funktionen	DOORS, UML-Suite, ASCET-SD
Funktionennetz	DOORS, UML-Suite, ASCET-SD
Logische Systemarchitektur	DOORS, UML-Suite, ASCET-SD
Technische Systemarchitektur	DOORS, ASCET-SD
Implementierung	DOORS, ASCET-SD

Automotive – Einsatz der UML

- Standard für objektorientierte Notationen
- „Multi purpose“ Modellierungssprache
- Spezialisierung durch Profilbildung
- kompatibel zum Konzept der Abstraktionsebenen
- gemeinsame Sprache für alle Beteiligten

Definition eines UML Automotive Profils

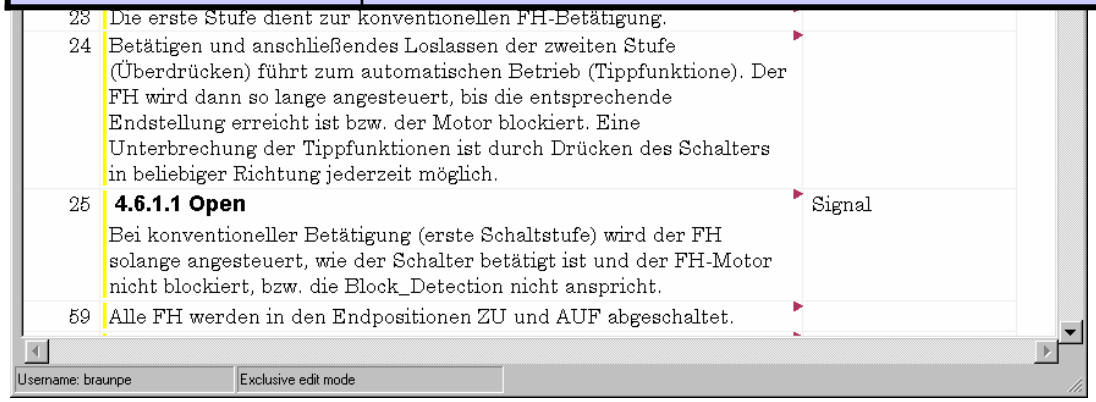
- *breite Akzeptanz in der Industrie*
- *„Automatische“ Integration in bestehende und zukünftige UML-basierte Entwicklungsprozesse*

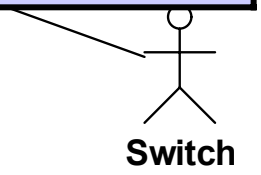
Definition der AML Spracharchitektur (Metamodell) mit UML

Ankopplung des AML Metamodells an den (neuen) UML Standard

Szenarien

Angabe von exemplarischen Systemabläufen (Szenarien)	
Diagrammart: *Use Case Diagram*	
Konstrukte	**Dargestellter Inhalt**
Use Cases	Szenarien
Dependency	Dekomposition von Szenarien
Actor	Definition von Schnittstellen

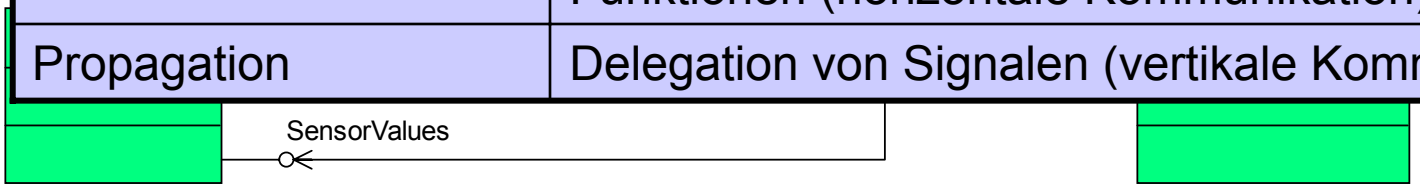




Funktionen

Hierarchische Dekomposition

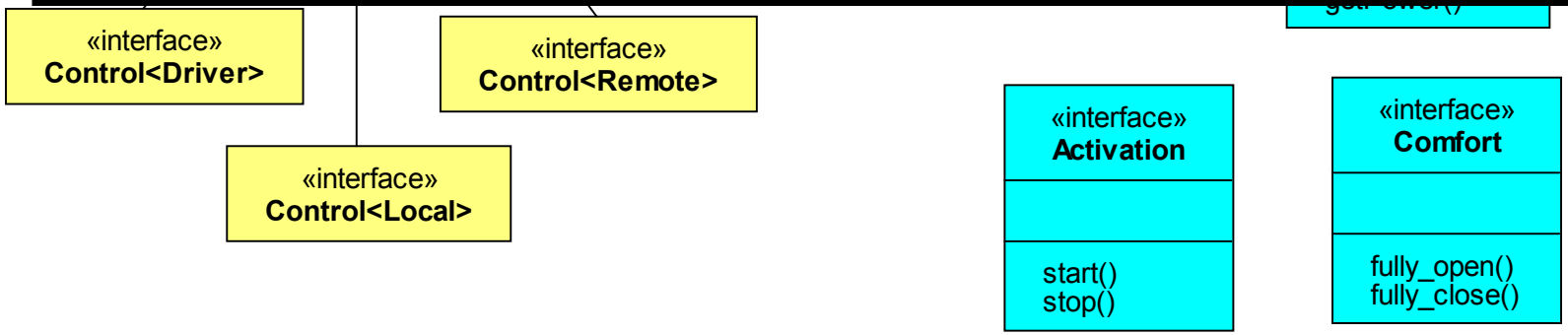
Darstellung von Funktionen und deren hierarchische Struktur	
Diagrammart: *Class Diagram*	
Konstrukte	**Dargestellter Inhalt**
Klassen	Funktionen
Hierarchische Dekomposition von Klassen	Dekomposition von Funktionen
Interfaces – Lollipops (Schnittstellenklassen)	Definition von Schnittstellen (Unterstützung der Wiederverwendung durch Schnittstellenvererbung)
Dependency	Signalabhängigkeiten zwischen einzelnen Funktionen (horizontale Kommunikation)
Propagation	Delegation von Signalen (vertikale Kommunikation)



Funktionen

Schnittstellendefinitionen

Darstellung von Schnittstellen und ihren Signalen	
Diagrammart: *Class Diagram*	
Konstrukte	**Dargestellter Inhalt**
Interface-Klassen	Spezifikation von Schnittstellen mit ihren Signalen
Interface-Klassen Templates	Generische Schnittstellen-Schablonen mit formalen Parametern (z.B. für die Sender-ID)
Dependency «bind»	Generierung von Schnittstellen aus Templates durch Bindung der formalen Parameter



Funktionennetz

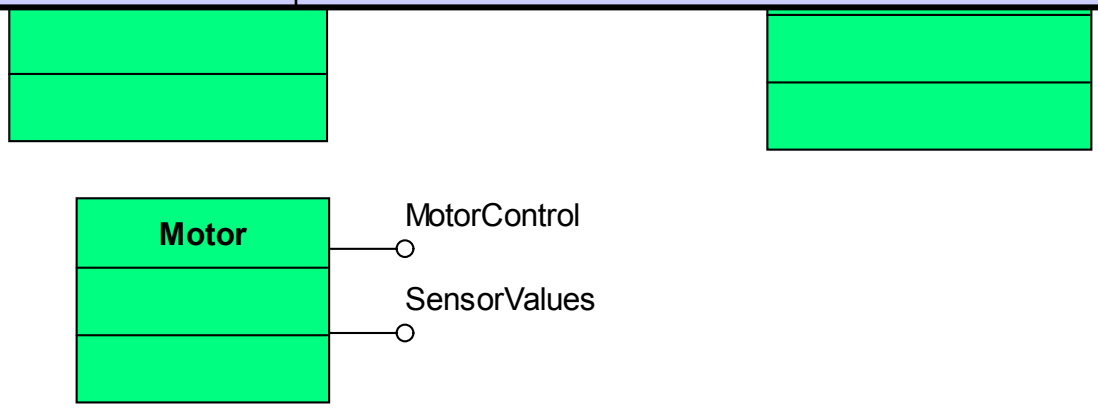
Variantenbildung

Darstellung von Funktionen und ihren Varianten	
Diagrammart: *Class Diagram*	
Konstrukte	**Dargestellter Inhalt**
Klassen	Funktionen und ihre Varianten
Hierarchische Dekomposition von Klassen	Dekomposition von Varianten
Dependency «variant of»	(Graphische) Darstellung der Variantenbildung aus Funktionen

Funktionennetz

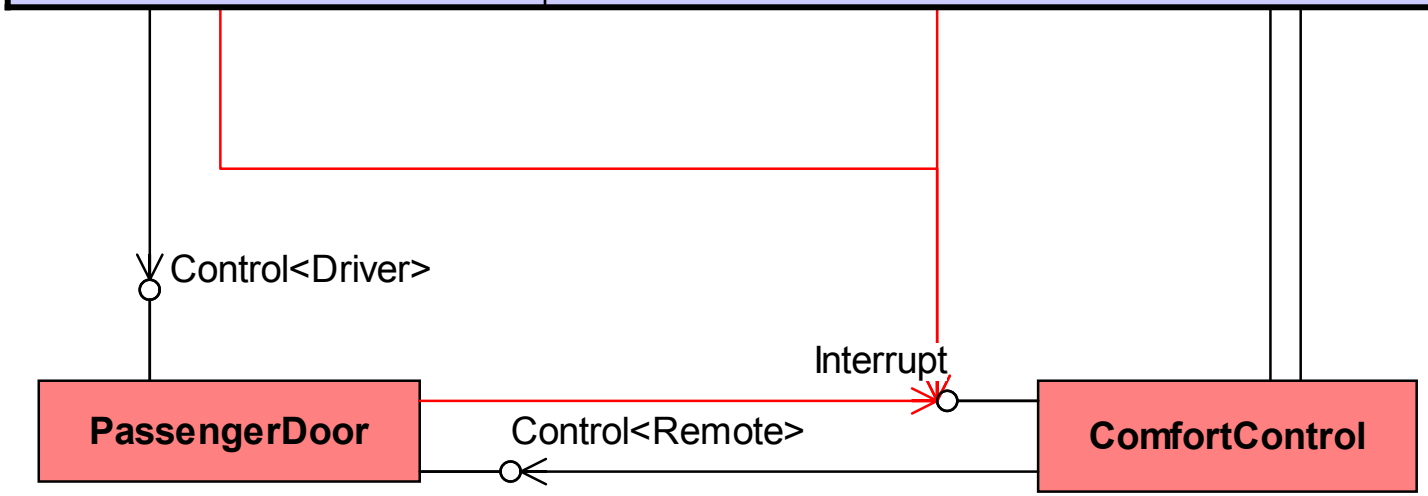
Hierarchische Dekomposition

Darstellung von Varianten und deren hierarchische Struktur	
Diagrammart: *Class Diagram*	
Konstrukte	**Dargestellter Inhalt**
gefaltete Klasse	Variante
Klassen	Sub-Funktionen, die in der Variante realisiert werden
Interfaces – Lollipops	Varianten-Schnittstellen mit Referenz zu den Schnittstellenklassen Definitionen



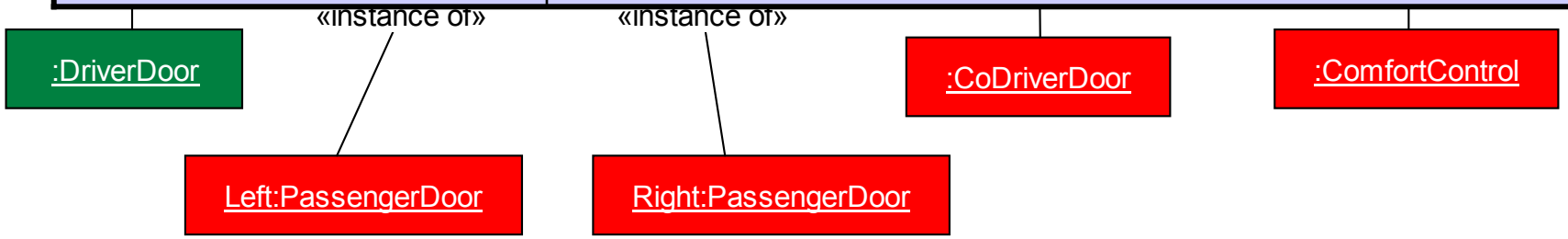
Funktionennetz

Darstellung der Kommunikation zwischen Varianten	
Diagrammart: *Class Diagram*	
Konstrukte	**Dargestellter Inhalt**
gefaltete Klassen	Varianten
Interfaces – Lollipops	Varianten-Schnittstellen mit Referenz zu den Schnittstellenklassen Definitionen
Dependency	Signalabhängigkeiten zwischen einzelnen Varianten



Logische Systemarchitektur

Darstellung der Instanziierungen von Varianten	
Diagrammart: *Class Diagram*	
Konstrukte	**Dargestellter Inhalt**
Klassen	Varianten
Objekte	Instanzen von Varianten
Dependency «instance of»	Darstellung der Instanziierung aus Varianten

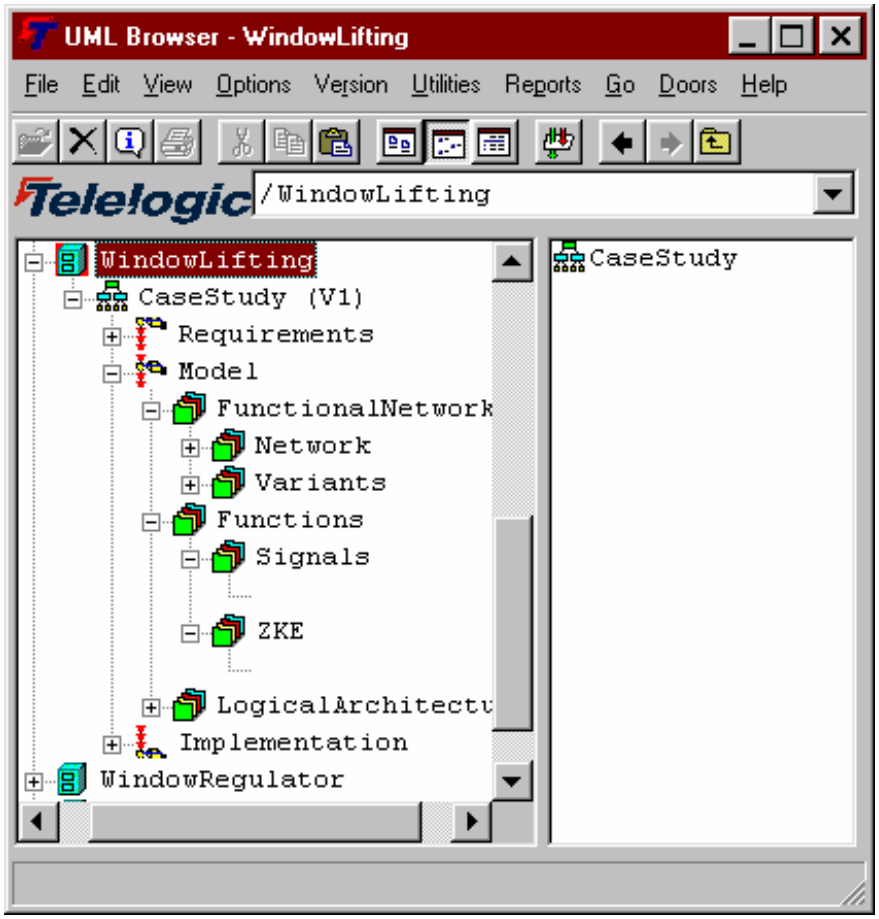


Logische Systemarchitektur

Verteilung der Funktionen auf Steuergeräte

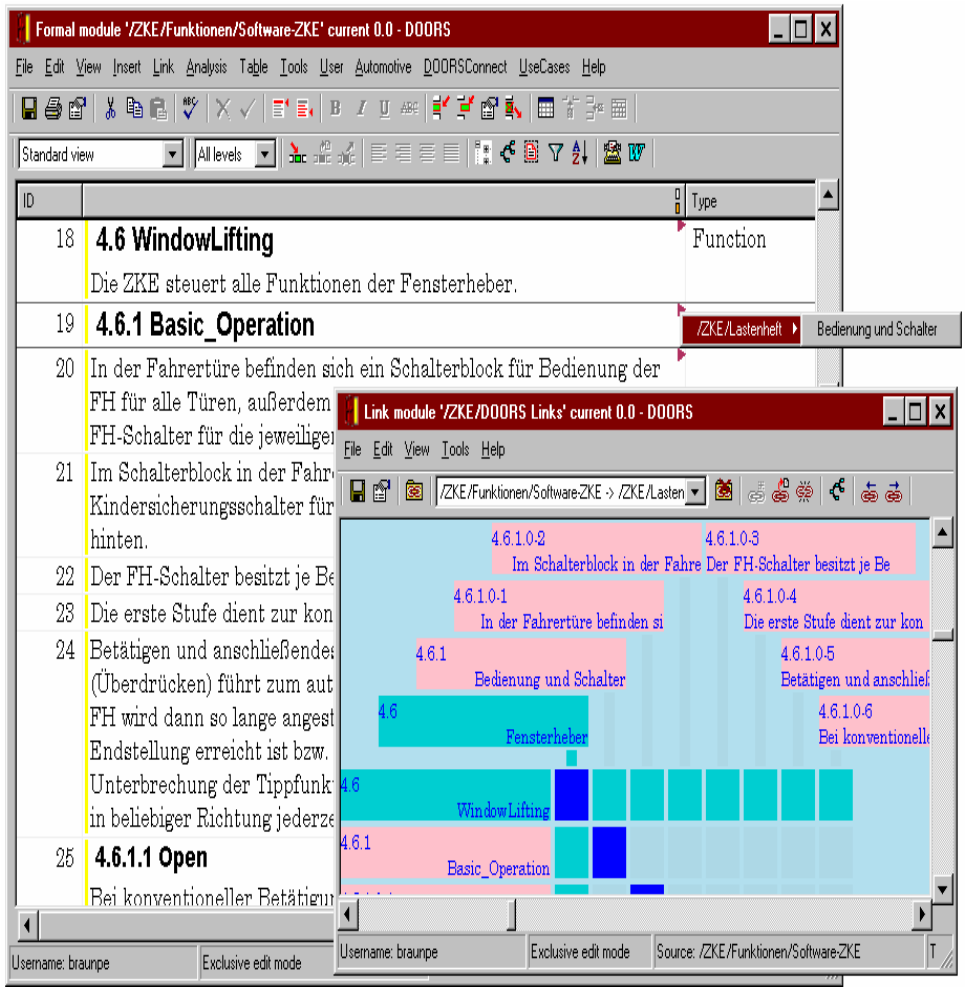
Verteilung der Funktionen auf den Steuergeräten	
Diagrammart: *Deployment Diagram*	
Konstrukte	**Dargestellter Inhalt**
Objekte	Varianten-Instanzen
Interfaces – Lollipops	Definition der Schnittstellen (inkl. Abhängigkeiten auf Instanzebene)
Dependency	Darstellung der Signalabhängigkeiten (horizontale Kommunikation)
Component	Logische Aggregation von Varianteninstanzen zu „Subsystemen“
Nodes	Abstrakte Darstellung von aktiven Rechnereinheiten (Steuergeräte)

Technische Unterstützung: UML-Suite



- Navigation zwischen Modellelementen und Diagrammen aller Abstraktionsebenen durch „Doppelklick“
- Übersichtliche Projektdarstellung durch konfigurierbaren Browser

Technische Unterstützung: DOORS



- Darstellung der Beziehungen im Modell (z.B. Szenarien-Funktionen, Variantenbildung) durch Links und Linkmodule zwischen Dokumenten
- Automatische Nachvollziehbarkeit
- Navigation von Anforderungen zu Modellelementen
- Verfolgung der Herkunft von Anforderungen

Laufende Untersuchungen

- Dynamische System-Modellierung mit UML
- Zur UML konsistente Modellierung mit ASCET-SD
  - Verhaltensbeschreibung von Funktionen mit Hilfe von Automaten
  - Darstellung von Instanzen und Signalen
  - Schnittstellenvererbung
- Untersuchung der gewonnenen Sprachkonstrukte
  - Anwendbarkeit
  - Effizienz
  - Ausdrucksstärke
- Implementation des zweiten Prototypen der Werkzeugkopplung
  - Konsolidierung des AML Metamodells
  - Anpassung der Import/Export Module der Werkzeuge

Ausblick

Automotive Modeling Language (AML):

- Modellierungssprache zur Spezifikation eingebetteter Systeme
- Gemeinsamer Standard für Hersteller und Zulieferer
- Basiert auf einer Untermenge von UML und ASCET-SD Notationen

Iterative Erarbeitung der Werkzeugkette

- Evaluierung des zweiten Prototypen
- Produktreife Werkzeugkette
- Support durch führende Werkzeughersteller

Fallstudien

- Durchführung realistischer Fallstudien bei den assoziierten Partnern
- Gemeinsame Evaluierung des Automotive Entwicklungsprozesses und der Werkzeugkette

Produktreife Werkzeugkette und konsolidierte AML: Feb. 2003