Using Behavior Models for the Specification of Software based automotive Systems – Challenges and practical Experiences

Dr. Henning Kleinwechter, Carmeq GmbH
Dr. Andreas Leicher, Carmeq GmbH

Behavior Models for Software Specification

Agenda

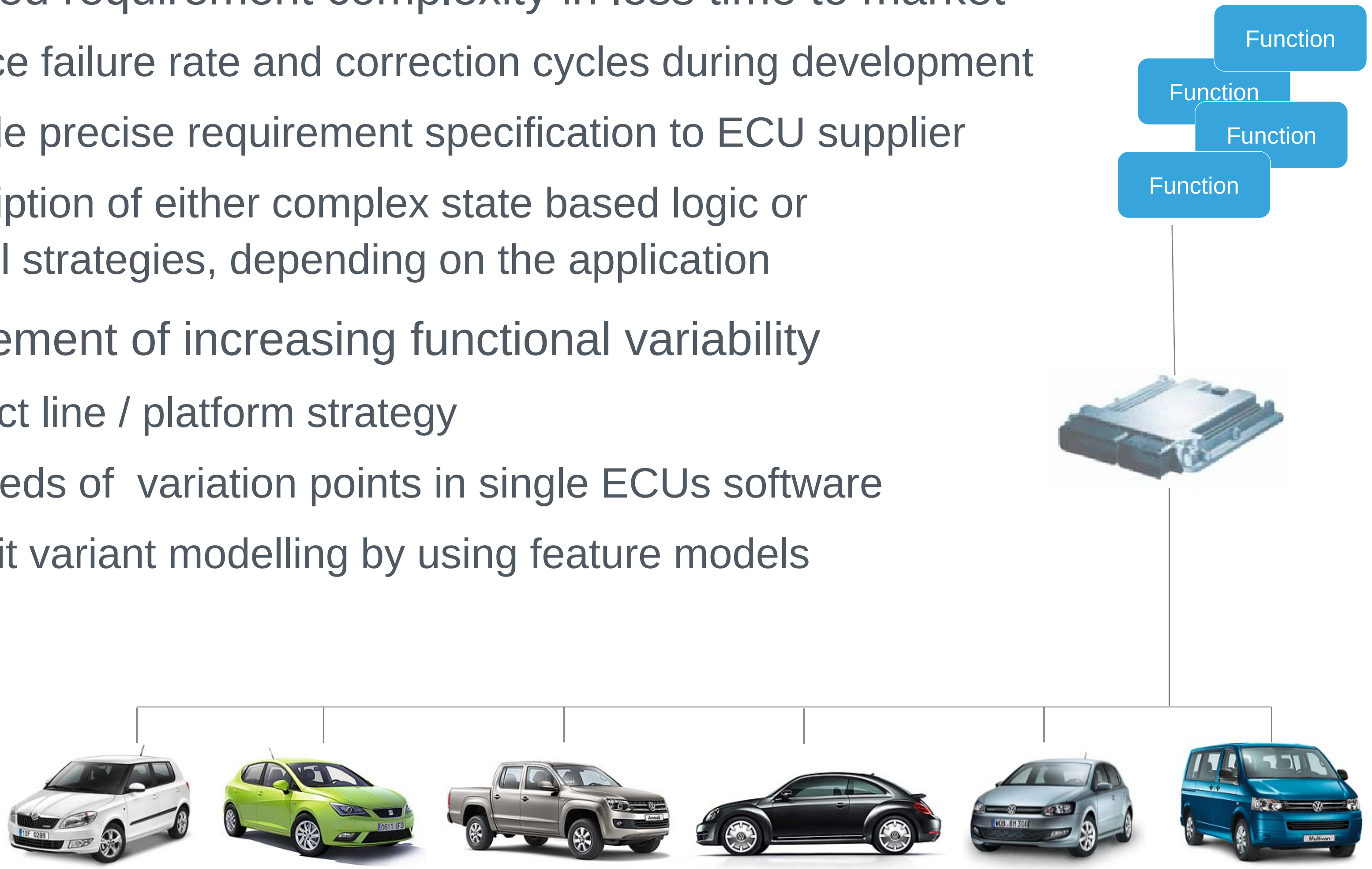
- **Challenges**
- Model supported Specification
- Variant Management

CARMEQ

Development of Electronic Control Units (ECU)

Challenges

- Increased requirement complexity in less time to market
  - Reduce failure rate and correction cycles during development
  - Provide precise requirement specification to ECU supplier
  - Description of either complex state based logic or control strategies, depending on the application
- Management of increasing functional variability
  - Product line / platform strategy
  - Hundreds of variation points in single ECUs software
  - Explicit variant modelling by using feature models



CARMEQ

Behavior Models for Software Specification

Agenda

- Challenges
- **Model supported Specification**
- Variant Management

CARMEQ

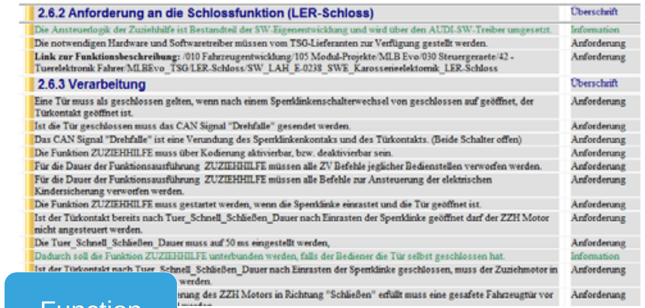
ECU Specification

Use of Models

- Requirements Specifications have to be precise and consistent
  - In order to get the correct functionality fast
  - In order to get the **same** functionality from different suppliers

    (especially true for multi supplier strategy)

textual requirements
(high degree of interpretation)

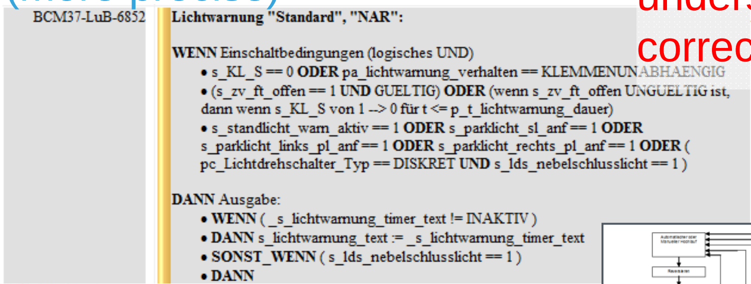


2.6.2 Anforderung an die Schlossfunktion (LER-Schloss)	Überschrift
Die Ansteuerlogik der Zuziehhilfe ist Bestandteil der SW-Eigenentwicklung und wird über den AUDI-SW-Treiber umgesetzt.	Information
Die notwendigen Hardware und Softwaretreiber müssen vom TSG-Lieferanten zur Verfügung gestellt werden.	Anforderung
Link zur Funktionsbeschreibung: /010 Fahrzeugentwicklung/105 Modul-Projekte/MLB Evo/030 Steuergeraete/42 - Tuerelektronik Fahrer/MLBEvo_TSG/LER-Schloss/SW_LAH_E-0238_SWE_Karosserieelektronik_LER-Schloss	Anforderung
2.6.3 Verarbeitung	Überschrift
Eine Tür muss als geschlossen gelten, wenn nach einem Sperrklinkenschalterwechsel von geschlossen auf geöffnet, der Türkontakt geöffnet ist.	Anforderung
Ist die Tür geschlossen muss das CAN Signal "Drehfalle" gesendet werden.	Anforderung
Das CAN Signal "Drehfalle" ist eine Verundung des Sperrklinkenkontaks und des Türkontakts. (Beide Schalter offen)	Anforderung
Die Funktion ZUZIEHHILFE muss über Kodierung aktivierbar, bzw. deaktivierbar sein.	Anforderung
Für die Dauer der Funktionsausführung ZUZIEHHILFE müssen alle ZV Befehle jeglicher Bedienstellen verworfen werden.	Anforderung
Für die Dauer der Funktionsausführung ZUZIEHHILFE müssen alle Befehle zur Ansteuerung der elektrischen Kindersicherung verworfen werden.	Anforderung
Die Funktion ZUZIEHHILFE muss gestartet werden, wenn die Sperrklinke einrastet und die Tür geöffnet ist.	Anforderung
Ist der Türkontakt bereits nach Tuer_Schnell_Schließen_Dauer nach Einrasten der Sperrklinke geöffnet darf der ZZH Motor nicht angesteuert werden.	Anforderung
Die Tuer_Schnell_Schließen_Dauer muss auf 50 ms eingestellt werden,	Anforderung
Dadurch soll die Funktion ZUZIEHHILFE unterbunden werden, falls der Bediener die Tür selbst geschlossen hat.	Information
Ist der Türkontakt nach Tuer_Schnell_Schließen_Dauer nach Einrasten der Sperrklinke geschlossen, muss der Zuziehmotor in [illegible] werden.	Anforderung
[illegible] erung des ZZH Motors in Richtung "Schließen" erfüllt muss eine gesafete Fahrzeugtür vor [illegible]	Anforderung

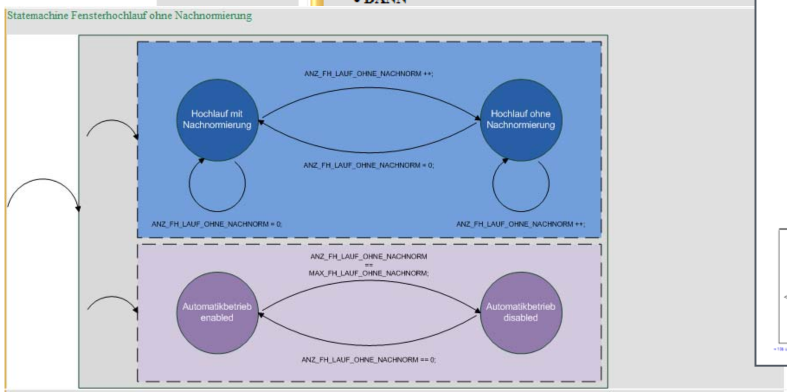
Function

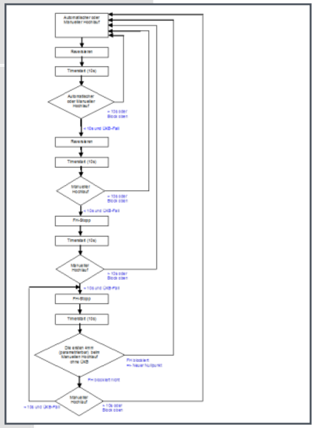
formalized text, informal models
(more precise)

understandable?
correct?



BCM37-LuB-6852	**Lichtwarnung "Standard", "NAR":** **WENN** Einschaltbedingungen (logisches UND) • s_KL_S == 0 **ODER** pa_lichtwarnung_verhalten == KLEMMENUNABHAENGIG • (s_zv_ft_offen == 1 **UND** GUELTIG) **ODER** (wenn s_zv_ft_offen UNGUELTIG ist, dann wenn s_KL_S von 1 --> 0 für t <= p_t_lichtwarnung_dauer) • s_standlicht_warn_aktiv == 1 **ODER** s_parklicht_sl_anf == 1 **ODER** s_parklicht_links_pl_anf == 1 **ODER** s_parklicht_rechts_pl_anf == 1 **ODER** (pc_Lichtdrehschalter_Typ == DISKRET **UND** s_lds_nebelschlusslicht == 1) **DANN** Ausgabe: • **WENN** (_s_lichtwarnung_timer_text != INAKTIV) • **DANN** s_lichtwarnung_text := _s_lichtwarnung_timer_text • **SONST_WENN** (s_lds_nebelschlusslicht == 1) • **DANN**





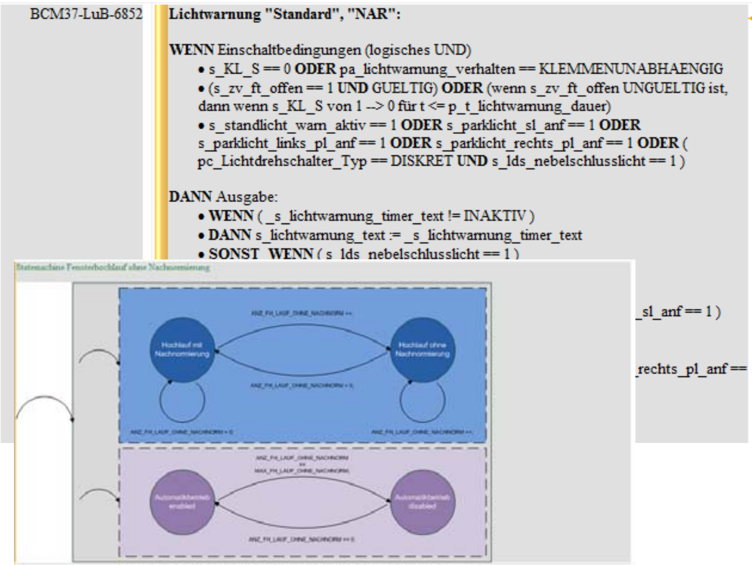
CARMEQ

Use of Behavior Models

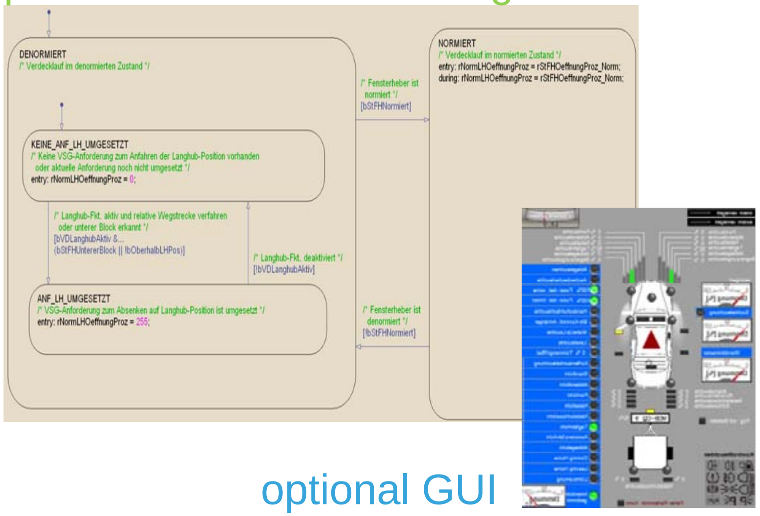
Model supported Specification

- Motivation
  - Early usage of models with defined semantic
  - Proof of correctness by simulation
  - Mainly Simulink/Stateflow used

formalized text, informal models



simulation model
defined semantic
proof of correctness through simulation

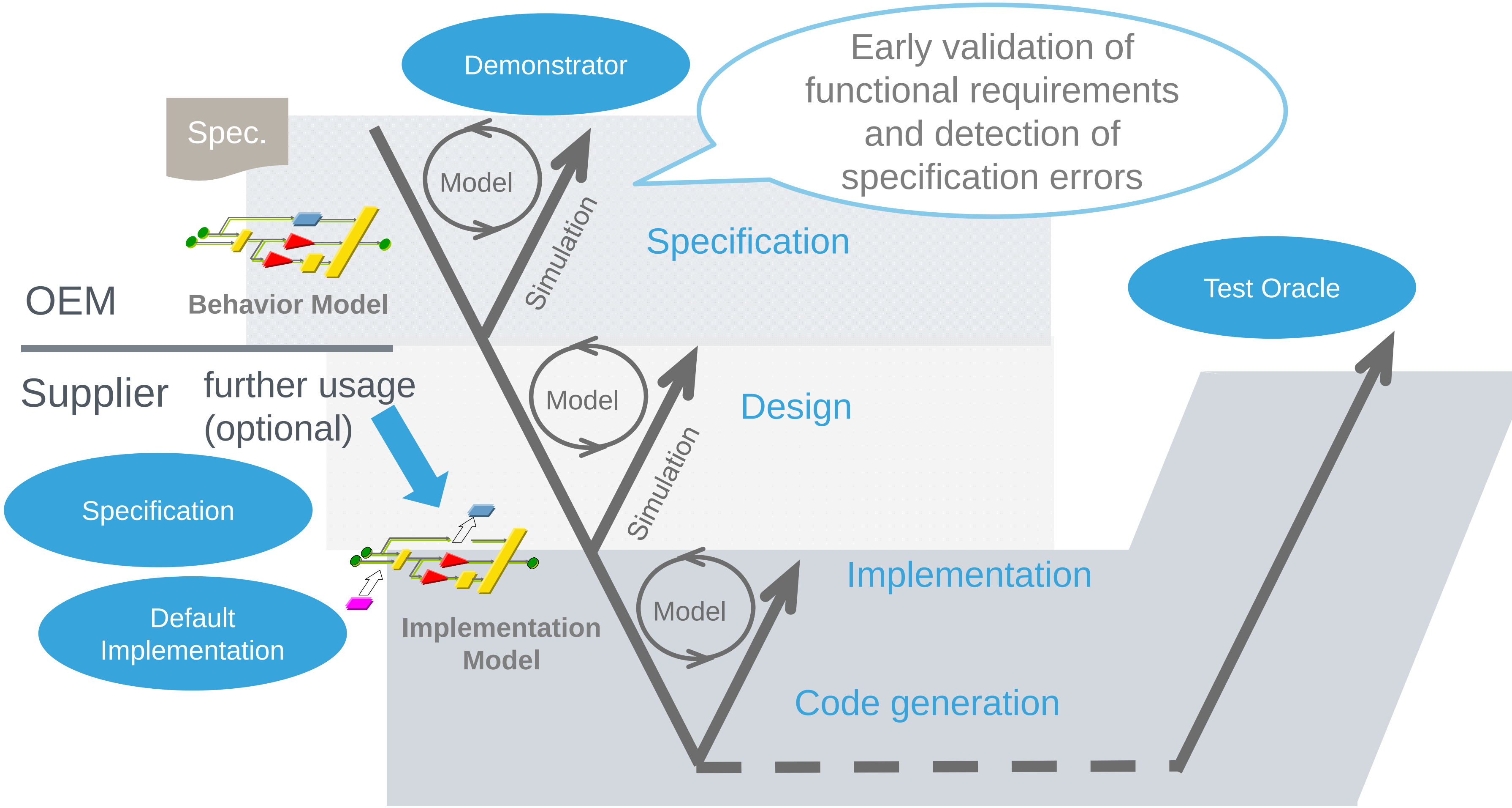


optional GUI
for demonstration purpose

CARMEQ

Use of Behavior Models

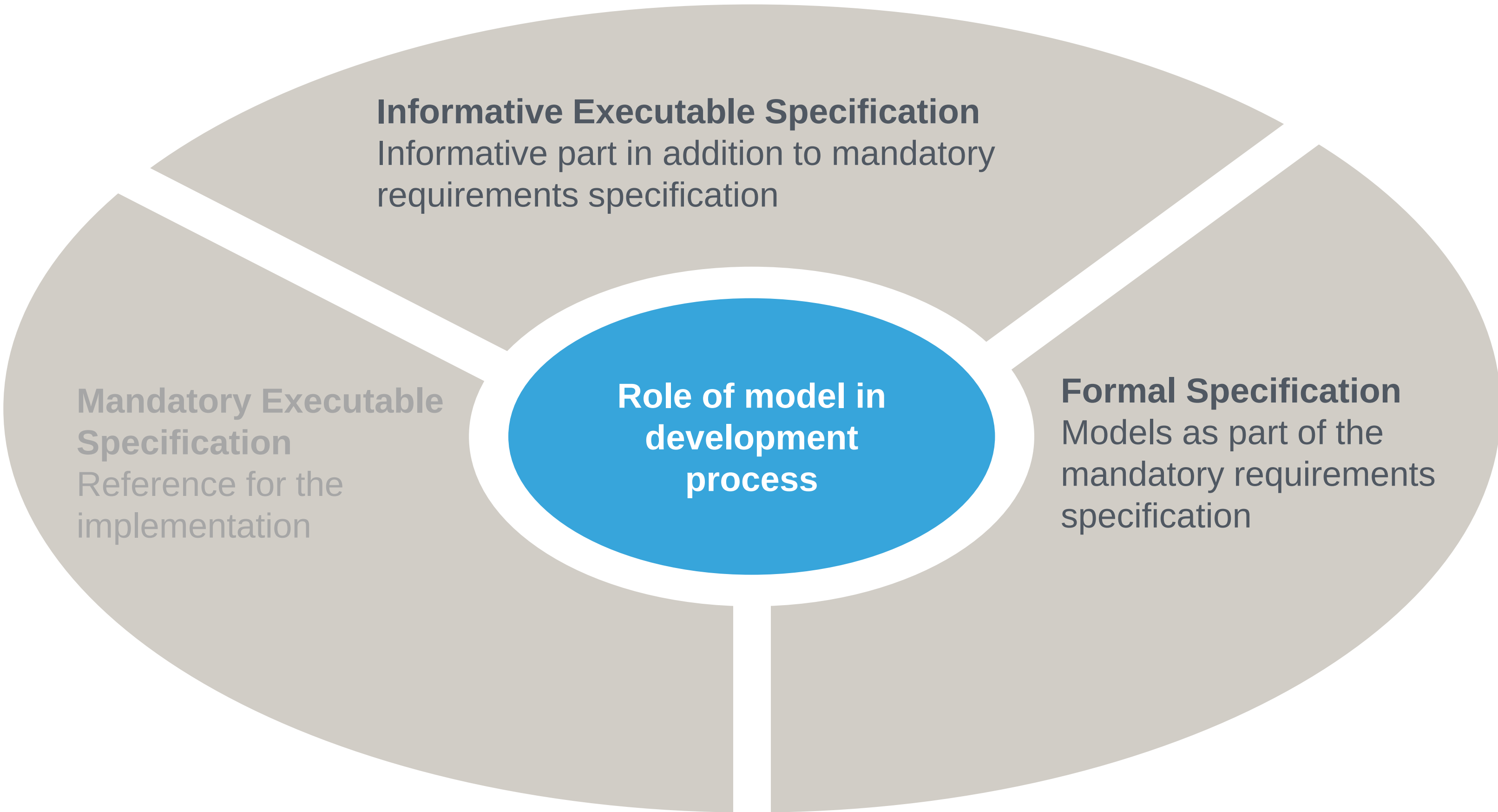
Process View and Benefit



CARMEQ

Model supported Specification

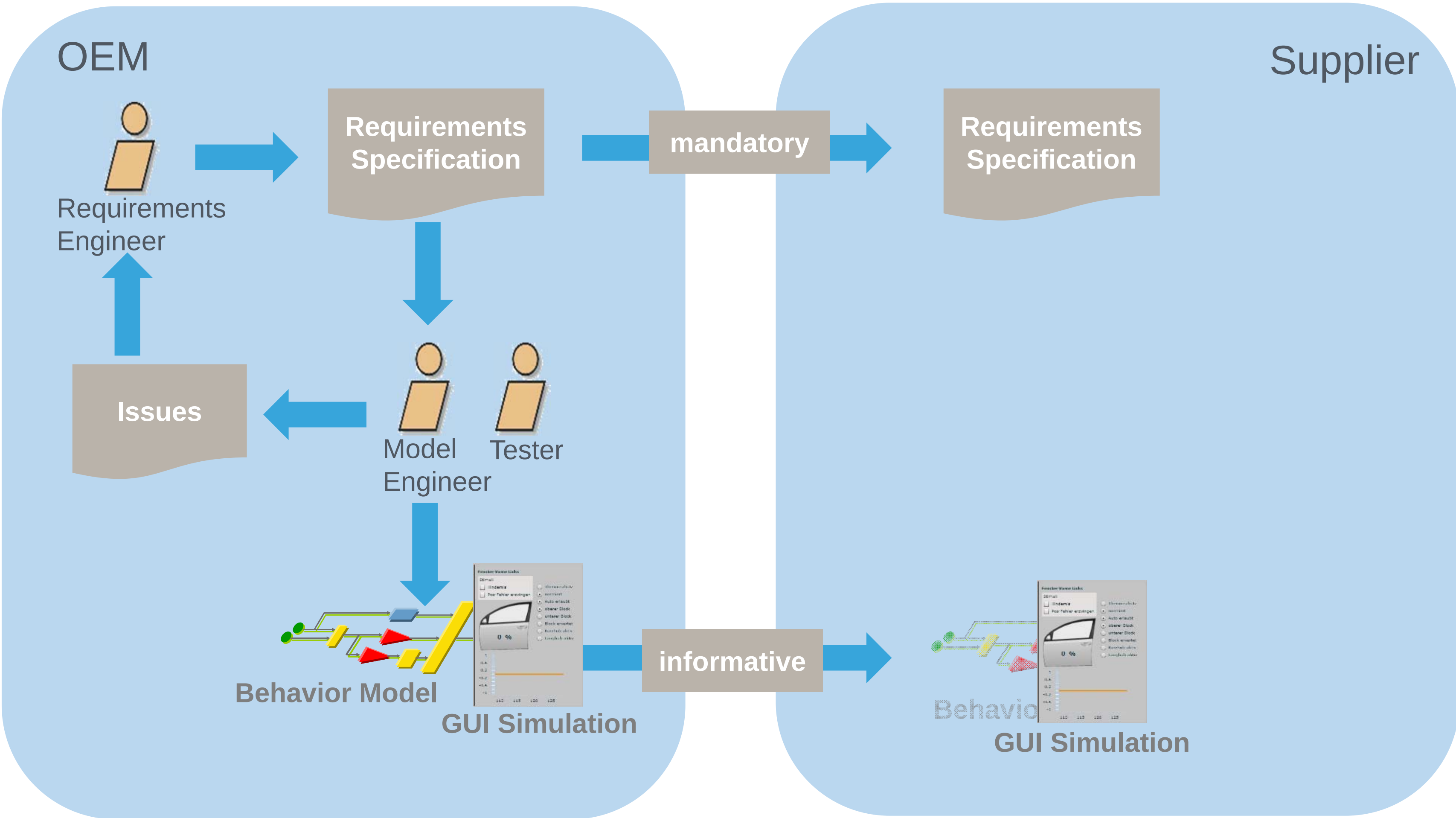
Role of models in development process



CARMEQ

Role of Models

Informative Executable Specification



CARMEQ

Informative Executable Specification

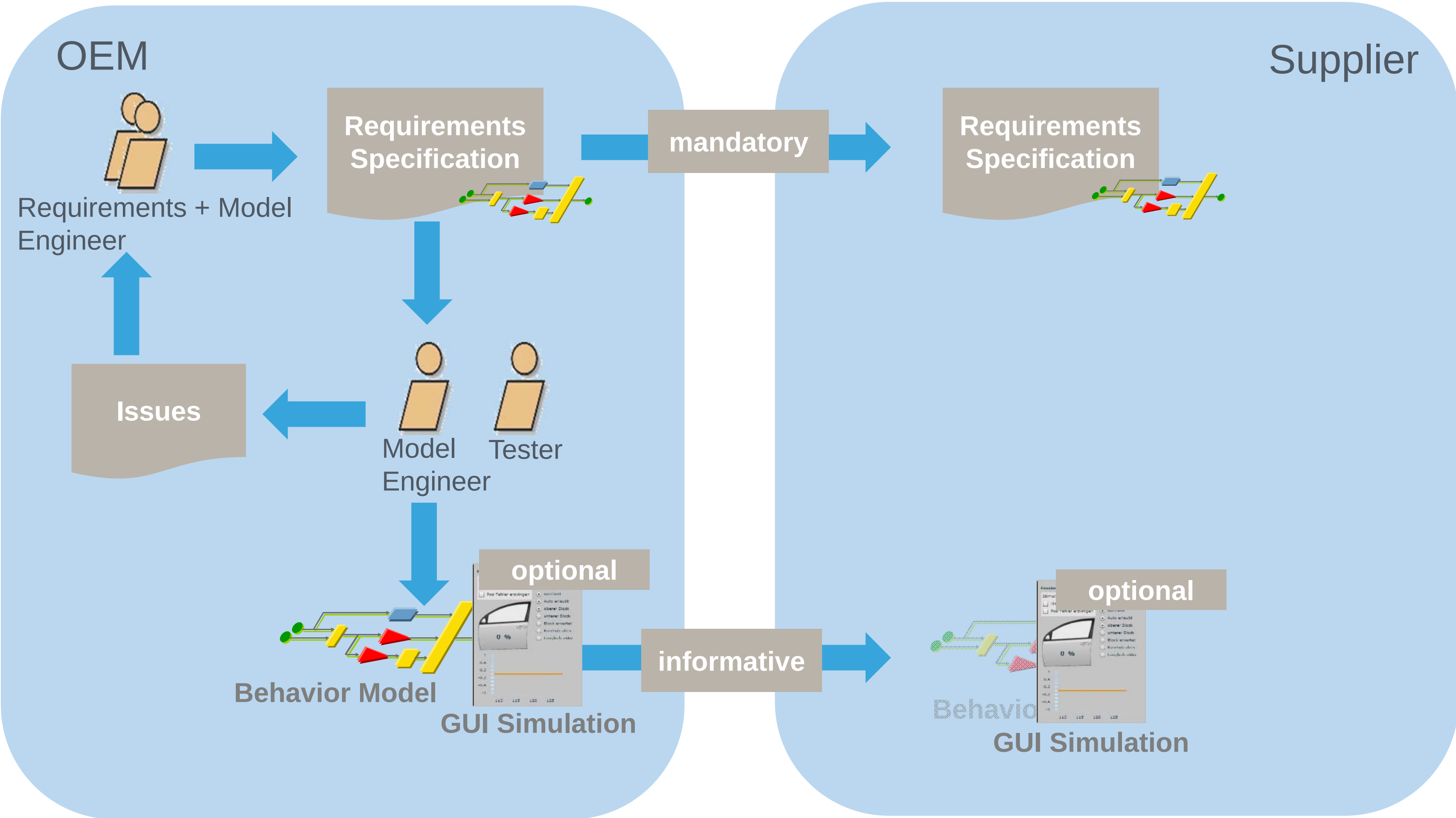
Experiences

advantage	disadvantage
› Improvement in the requirements specification › Better understanding about functionalities on OEM and supplier side › Reduction in development time, since less requirements have to be clarified › Reduction of costs for CRs	› High effort (and thereby cost) has to be spent in modelling activities › Requirements are still interpretable, since models do not have direct impact on specification method › Risk of inconsistency between requirements specification and model

- The improvement in the requirements specification (and the costs saved by this improvements) are usually compensated by the effort spend for modelling activities
- This strategy is profitable only, when models are further used for another purpose, e.g. rapid prototyping or series software implementation

Role of Models

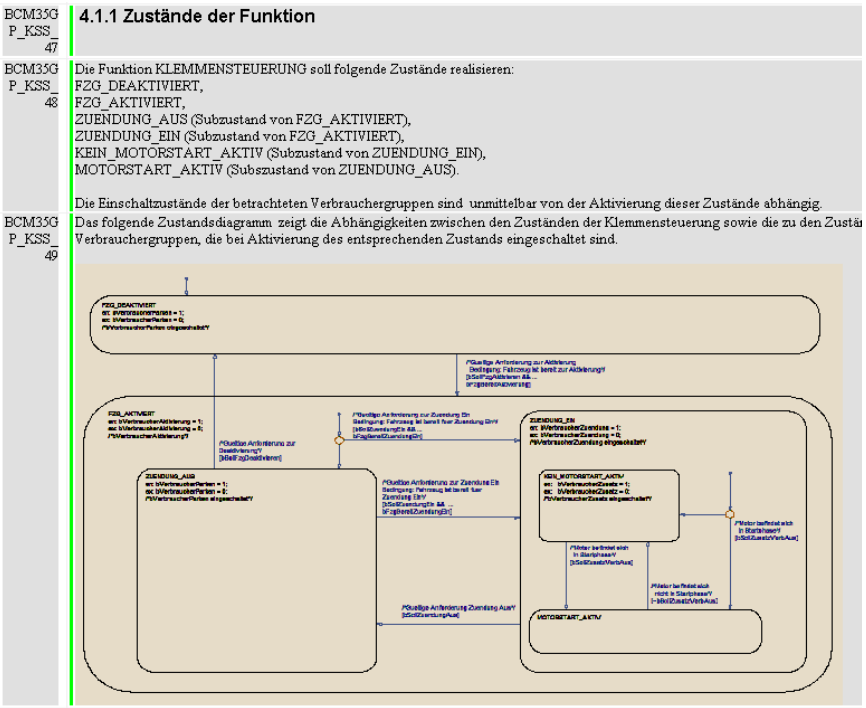
Formal Specification



Formal Specification

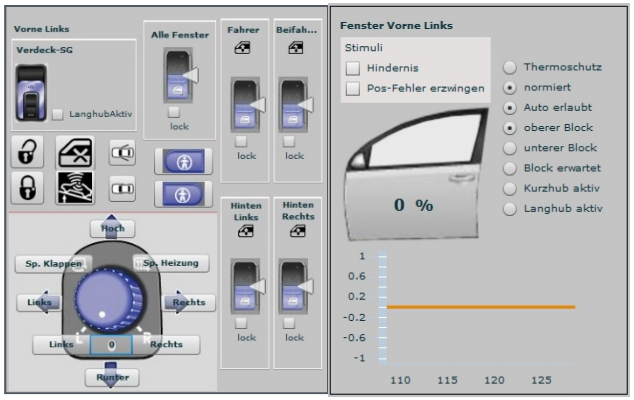
Characteristics

Requirements specification



BCM35G P_KSS_47	**4.1.1 Zustände der Funktion**
BCM35G P_KSS_48	Die Funktion KLEMMENSTEUERUNG soll folgende Zustände realisieren: FZG_DEAKTIVIERT, FZG_AKTIVIERT, ZUENDUNG_AUS (Subzustand von FZG_AKTIVIERT), ZUENDUNG_EIN (Subzustand von FZG_AKTIVIERT), KEIN_MOTORSTART_AKTIV (Subzustand von ZUENDUNG_EIN), MOTORSTART_AKTIV (Subzustand von ZUENDUNG_AUS). Die Einschaltzustände der betrachteten Verbrauchergruppen sind unmittelbar von der Aktivierung dieser Zustände abhängig.
BCM35G P_KSS_49	Das folgende Zustandsdiagramm zeigt die Abhängigkeiten zwischen den Zuständen der Klemmensteuerung sowie die zu den Zustä Verbrauchergruppen, die bei Aktivierung des entsprechenden Zustands eingeschaltet sind.

Optional: GUI based Simulation



Usually, only a part of the specification is replaced by models

Formal Specification

Experiences

advantage	disadvantage
› Improvement in the requirements specification › Better understanding about functionalities on OEM and supplier side › Low degree of interpretation › Reduction in development time, since less requirements have to be clarified › Reduction of costs for CRs	› Still initially high effort, but this is scalable, depending on the degree of modelling

- Improvement in the requirements specification with scalable effort
- Because only relevant parts of the specification are modeled, effort and cost for modelling activities are lower

Model supported Specification

Conclusions

- Model supported specification promises a variety of advantages
  - Less misinterpretation of requirements
  - Faster development cycles
  - Less communication effort
  - Better understanding of the underlying functionalities

- In order to obtain the most benefit
  - Models should play a mandatory role
  - Models should be exchanged with suppliers

- Initial high effort is notably justified in case of further usage of models

Behavior Models for Software Specification

Agenda

- Challenges
- Model supported Specification
- **Variant Management**

CARMEQ

Variant Management

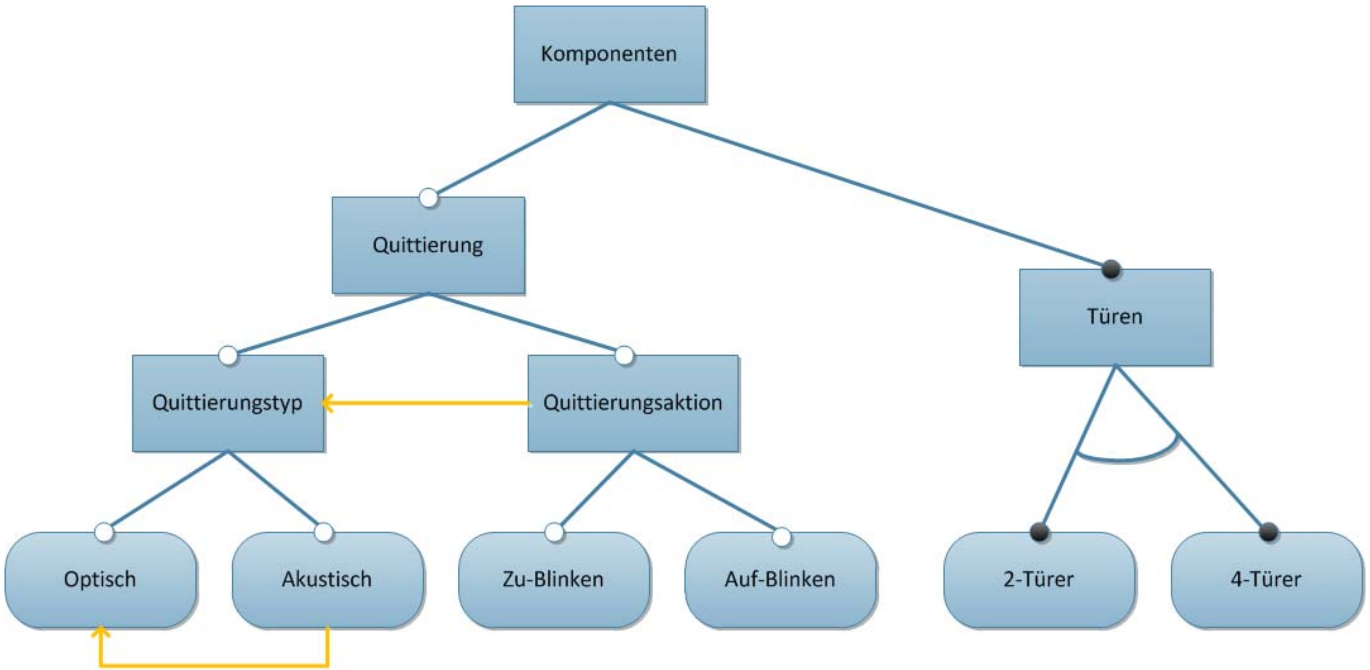
Motivation and Objectives

- Motivation
  - Software functions have to be deployed for different platforms and vehicles
  - Requirements differ slightly for each platform
  - Set of relevant functions differ for each platform
  - Software has to cope with these differences
- Objectives
  - Explicit variability management for models and requirements
  - Maintaining consistency between requirements and model variants
  - Provide a single software model which is able to handle defined variants
  - Management of similar software products (software product lines)

Variant Management

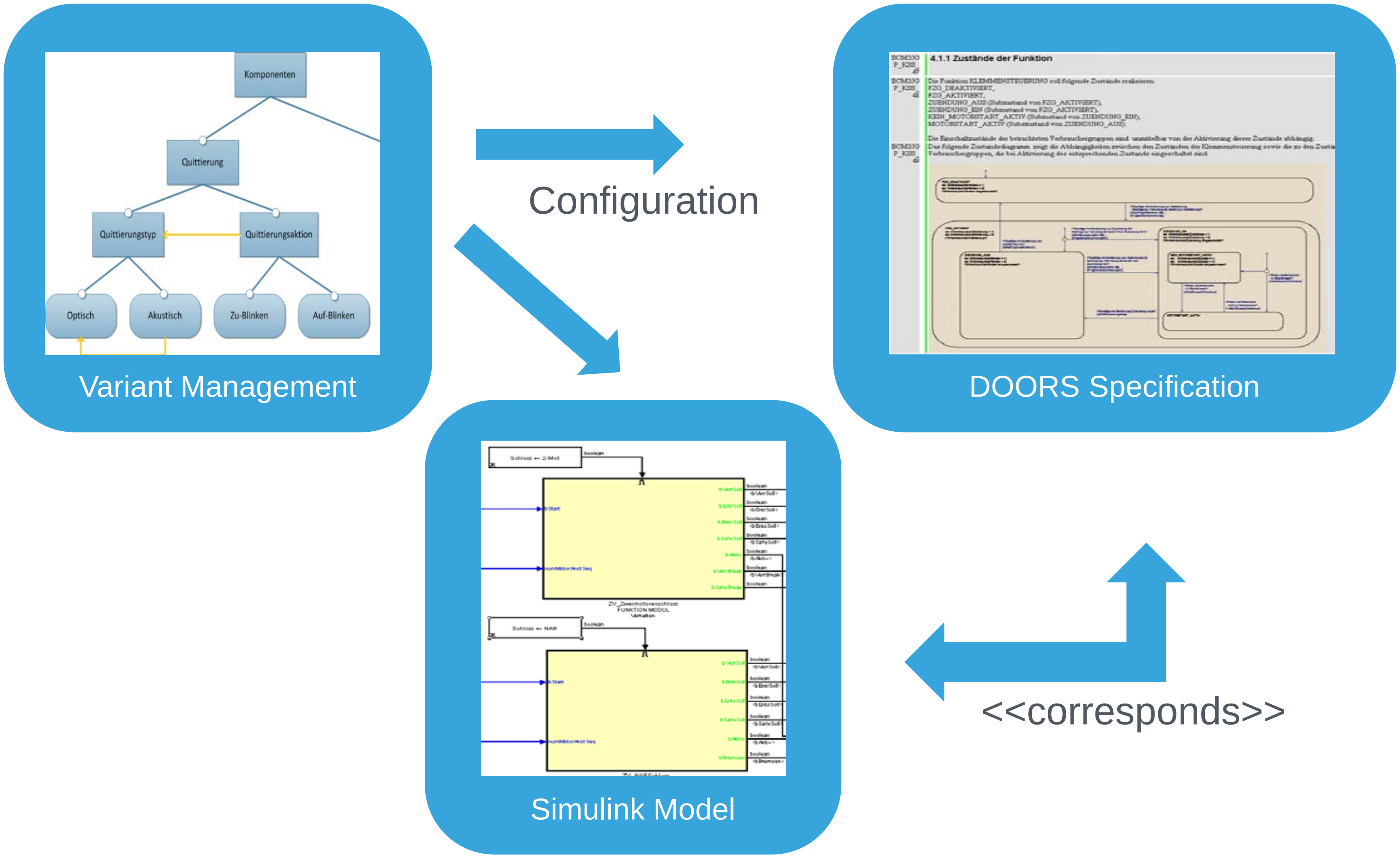
Feature Models

- Feature models explicitly describe variants in the software (Kang et.al., 1990)
- To reduce combinatorial multiplicity, relationships between features have to be defined



Overview Variant handling

Model Supported Specification

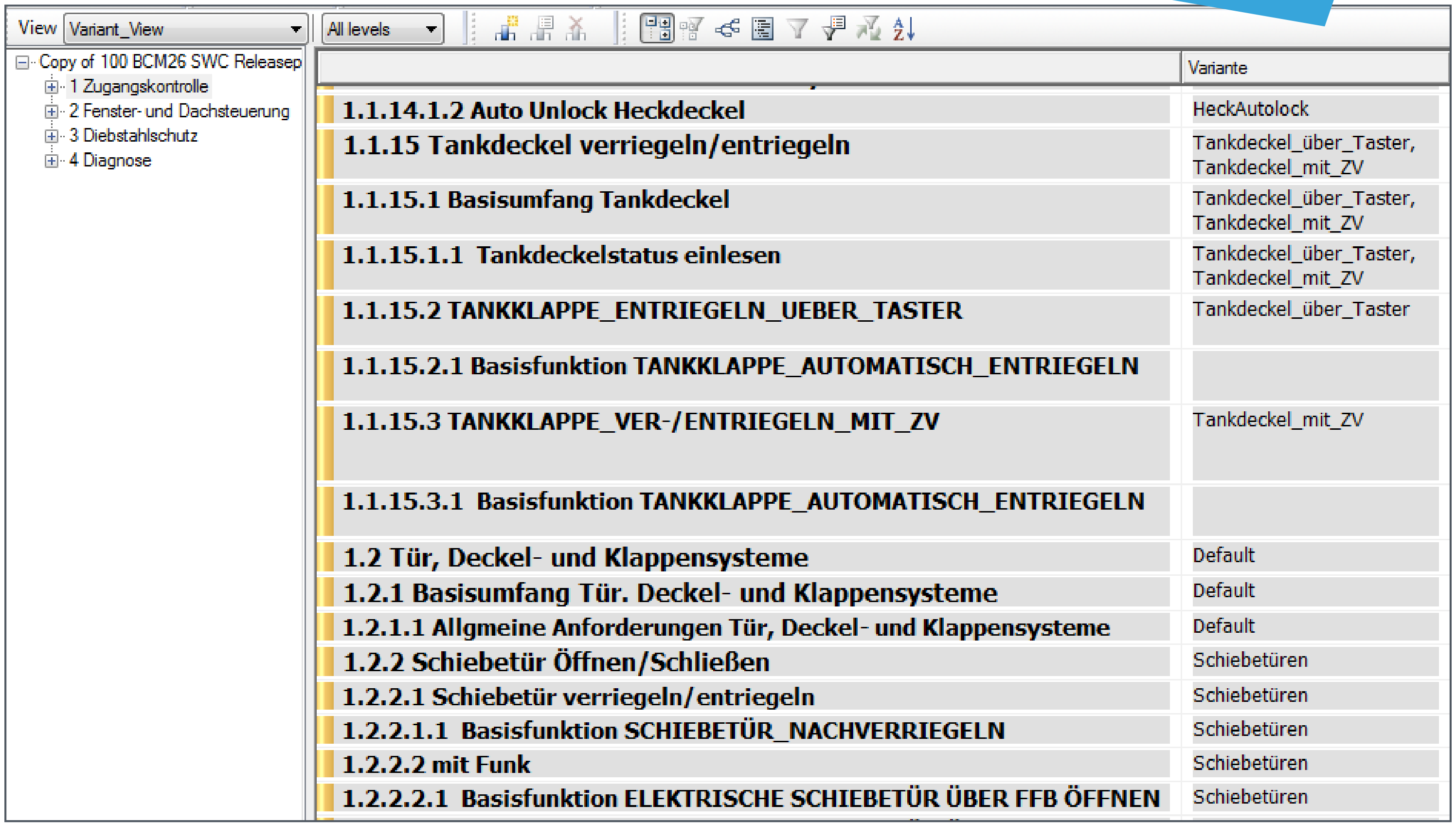


CARMEQ

Variant Management

DOORS Specification

Variants are maintained in feature models and are annotated to DOORS requirements. Feature selections are created by filters generated by the variability tool.

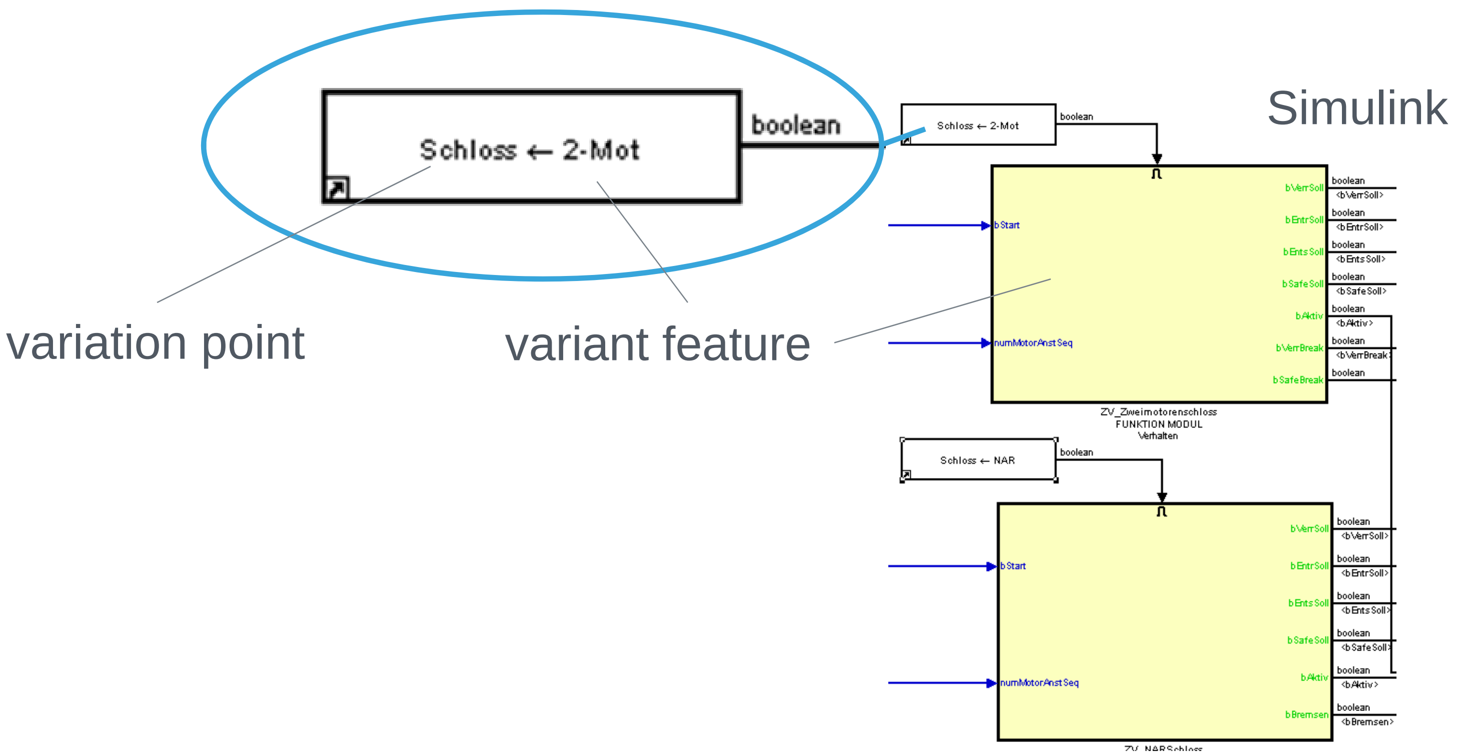


	Variante
1.1.14.1.2 Auto Unlock Heckdeckel	HeckAutolock
1.1.15 Tankdeckel verriegeln/entriegeln	Tankdeckel_über_Taster, Tankdeckel_mit_ZV
1.1.15.1 Basisumfang Tankdeckel	Tankdeckel_über_Taster, Tankdeckel_mit_ZV
1.1.15.1.1 Tankdeckelstatus einlesen	Tankdeckel_über_Taster, Tankdeckel_mit_ZV
1.1.15.2 TANKKLAPPE_ENTRIEGELN_UEBER_TASTER	Tankdeckel_über_Taster
1.1.15.2.1 Basisfunktion TANKKLAPPE_AUTOMATISCH_ENTRIEGELN	
1.1.15.3 TANKKLAPPE_VER-/ENTRIEGELN_MIT_ZV	Tankdeckel_mit_ZV
1.1.15.3.1 Basisfunktion TANKKLAPPE_AUTOMATISCH_ENTRIEGELN	
1.2 Tür, Deckel- und Klappensysteme	Default
1.2.1 Basisumfang Tür. Deckel- und Klappensysteme	Default
1.2.1.1 Allgmeine Anforderungen Tür, Deckel- und Klappensysteme	Default
1.2.2 Schiebetür Öffnen/Schließen	Schiebetüren
1.2.2.1 Schiebetür verriegeln/entriegeln	Schiebetüren
1.2.2.1.1 Basisfunktion SCHIEBETÜR_NACHVERRIEGELN	Schiebetüren
1.2.2.2 mit Funk	Schiebetüren
1.2.2.2.1 Basisfunktion ELEKTRISCHE SCHIEBETÜR ÜBER FFB ÖFFNEN	Schiebetüren

Variant Management

Simulink Model

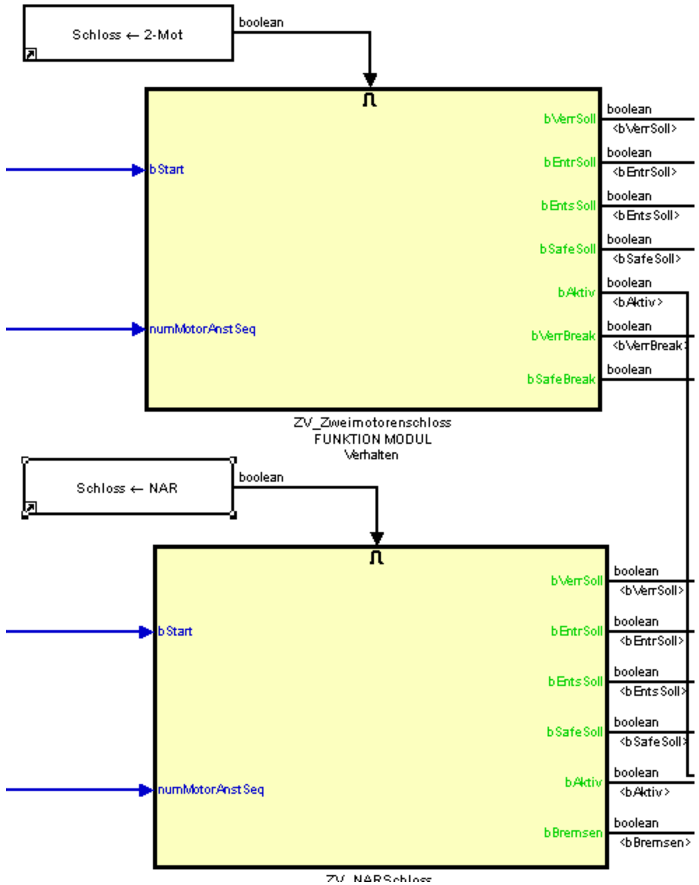
Variant features are modeled in different subsystems.
Correspondence to feature model is realized by special parameters.
For each variation point in the feature model a separate parameter exists.



Variant Binding Time

Simulink Model

- Design Time
  - Variants are selected for a specific project, e.g. they are fixed for given project
- Compile Time
  - Variants are determined at compile time
  - Deselected variants are removed from production code via specific storage classes
- Run Time
  - Variants are expressed via parameters that can be changed during run time.



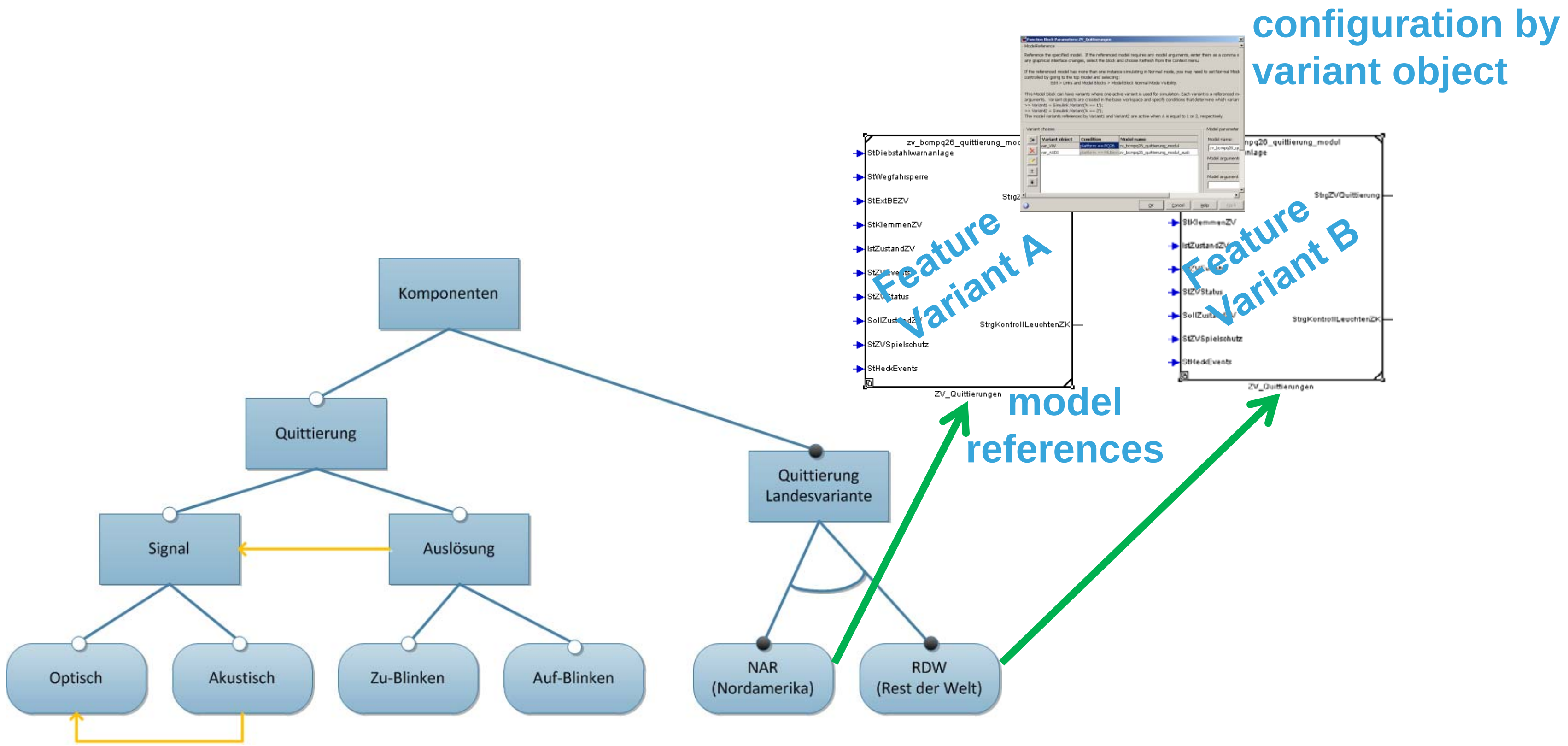
Compile Time and Run Time Variants are modelled as „Enabled“ Subsystems

CARMEQ

Variant Binding Time

Simulink Design Time Variants

- Are modelled as model references
- Variants are selected by variant objects
- Different implementations can be selected for specific projects.



CARMEQ

Variant Management

Conclusions

- We use a variant model that coordinates requirements in DOORS as well as the Simulink model
  - Variability is explicitly defined for all stakeholders
  - Dependencies between variation points are defined
  - Less errors in ECU software configuration
- We handle all variants in a single Simulink model hierarchy, which includes constructs for all kinds of binding time variants

- Further work
  - Improvements in tool integration needed: feature modelling to requirements management and modelling tools
  - Links between variant models of ECU functions and vehicle variant model

CARMEQ