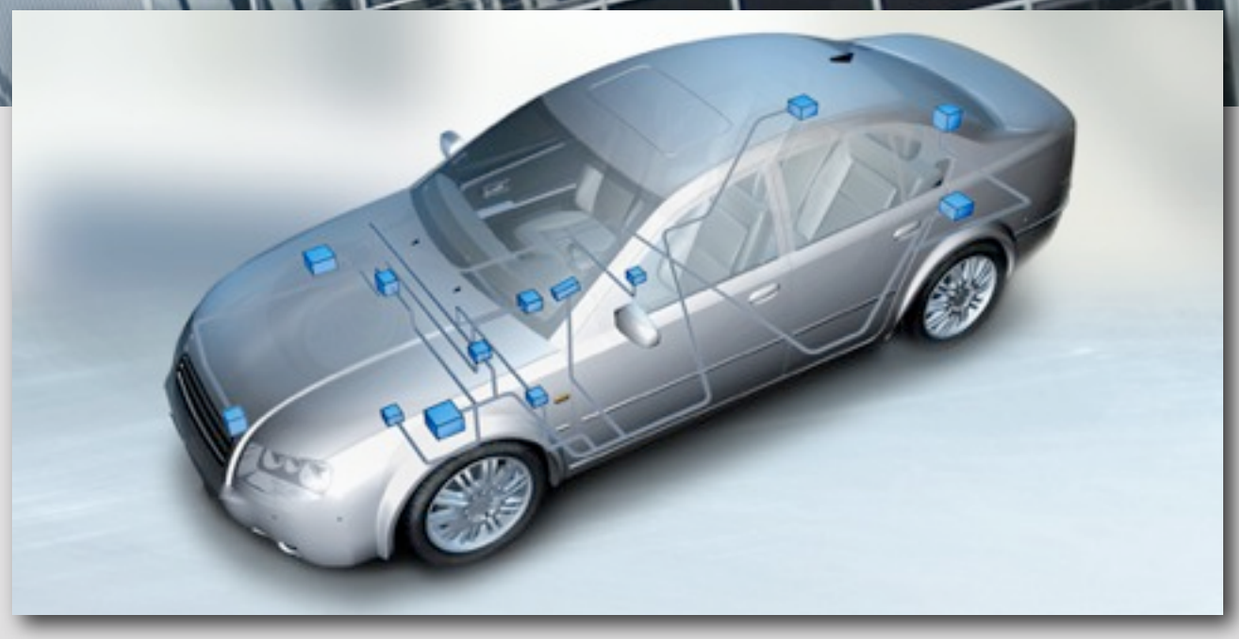
fortiss
innovation in software and systems

Model-Based Development of Automotive Software

From the State of the Art to Future Possibilities



SimPar 2010
Darmstadt
15. November 2010

Bernhard Schätz
fortiss gGmbH, München

fortiss - Who we are

- Private non-profit transfer and research institute of Technische Universität München
- Participators
  - Technische Universität München
  - LfA, Förderbank Bayern
  - Fraunhofer Gesellschaft
- Sponsored by Bavarian Ministery for Economics, Transportation, and Technology since January 1st, 2009
- Focus: Innovations in research and development of software intensive systems for technical and business applications

Development of embedded software: Challenges

- Complexity of domains
  - Application domains, e.g., power-train, body, infotainment
  - Engineering domains, e.g., mechanics, electric/electronics, informatics

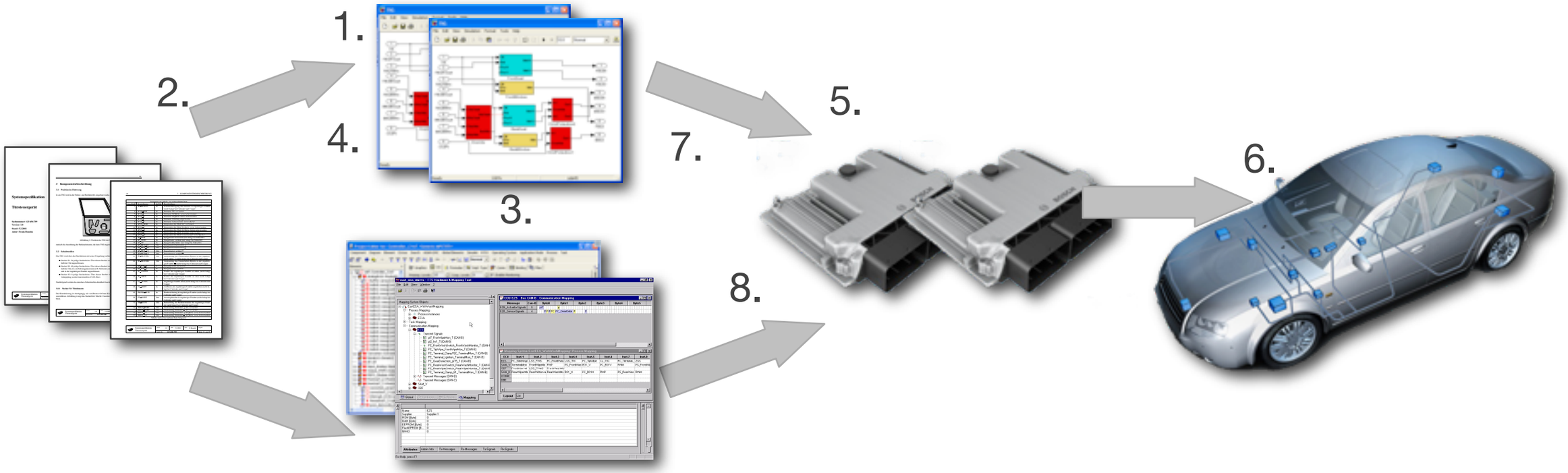
- Complexity of technology
  - Platform, e.g., memory limitations, bandwidth limitations
  - Distribution, e.g., sensors, control units, busses

- Complexity of functionality
  - Increase in size, e.g., comfort electronics, infotainment
  - Increase in interconnection, e.g., electronic stability, adaptive cruise control

- Complexity of development
  - Supply chain, e.g., middleware, applications
  - Product line, e.g., equipment, country specifics

➡ **Complexity reduction through use of models and tool support**

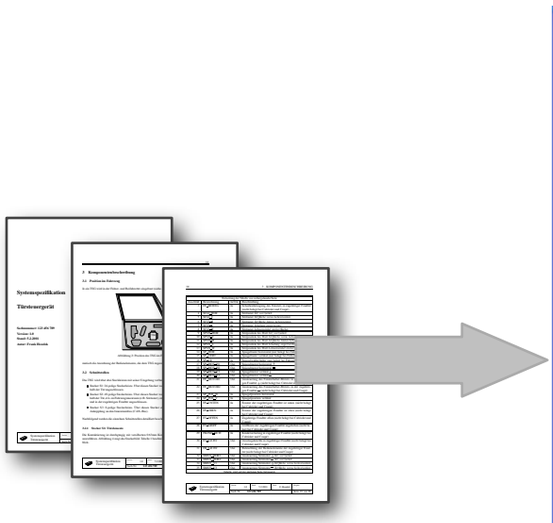
Automotive Software Development: Process

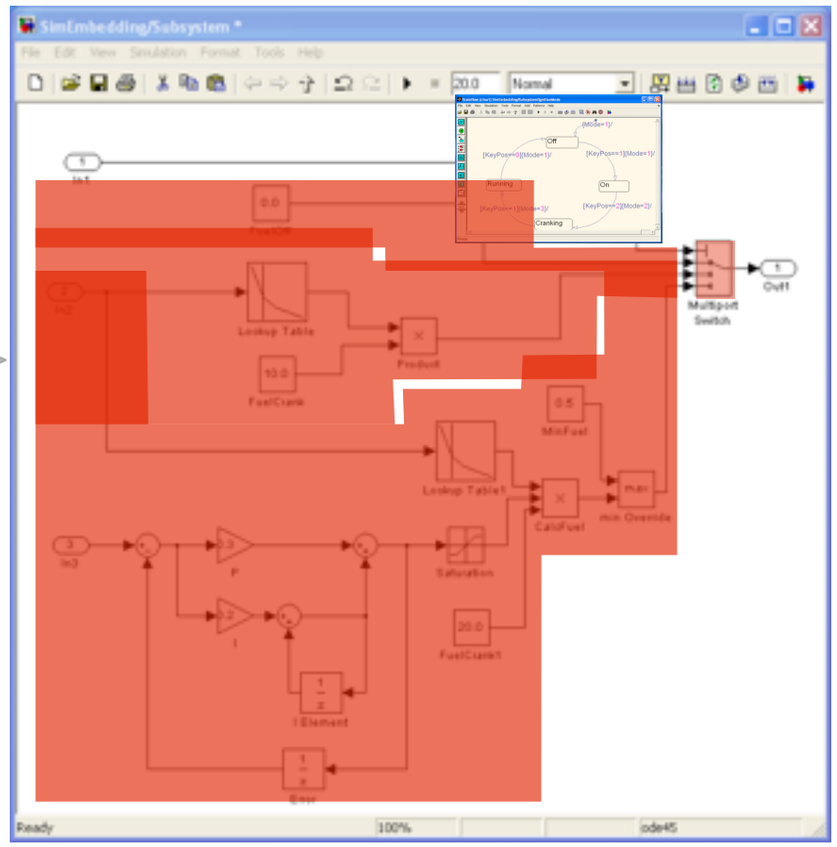


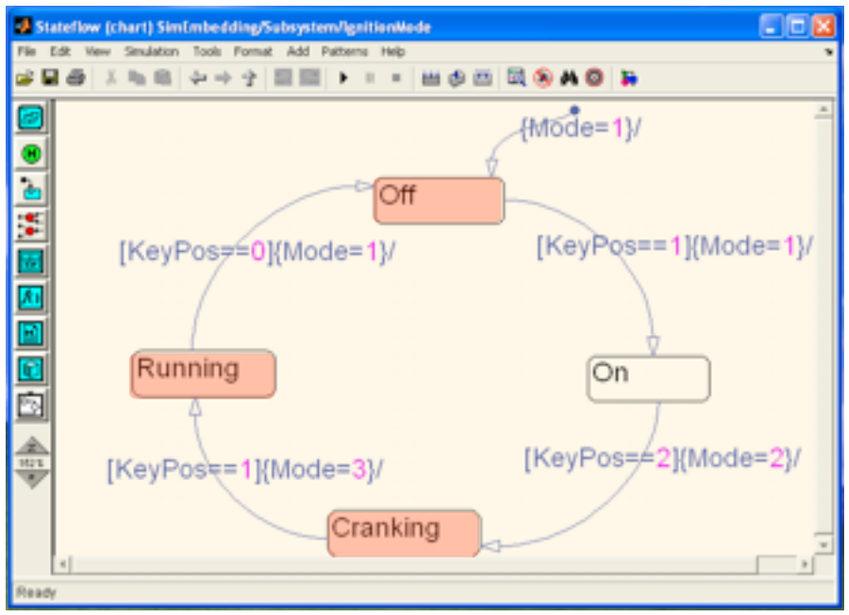
State-of-Art Development: Deficit by lacking models or inadequate use

1. Example component specification: Lacking domain concepts
2. Example system specification: Lacking functional models
3. Example implementation: Lacking view integration
4. Example component design: Lacking consistency analysis
5. Example release planning: Lacking plan generation
6. Example verification: Lacking test generation
7. Example system maintenance: Lacking quality assurance
8. Example system integration: Lacking design space exploration

Example 1: Component specification



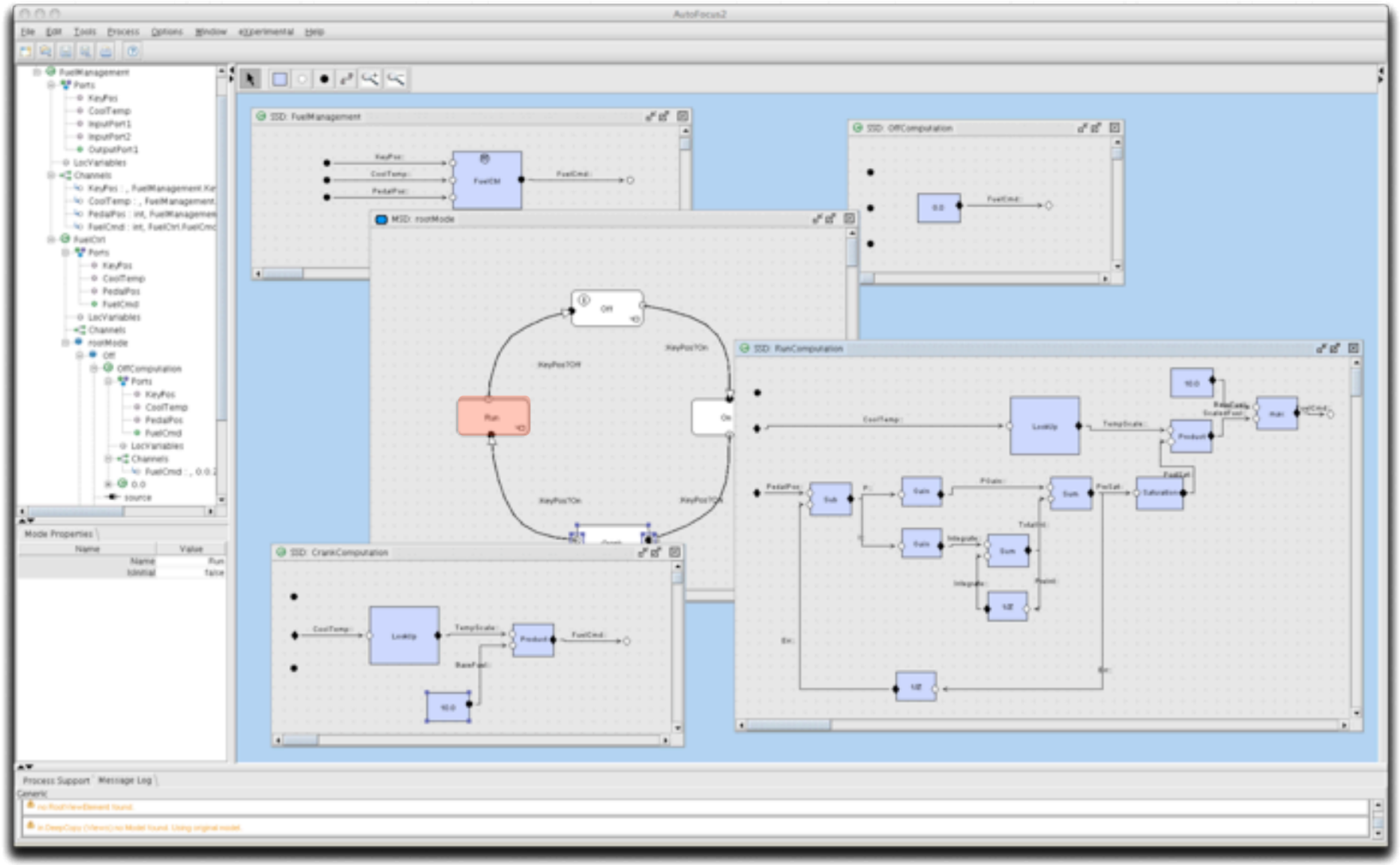




State-of-art: Inadequate models - Models capture technical solution

- Problem: Lacking concepts of problem domain
- Consequence: Inadequate models of functionality of application

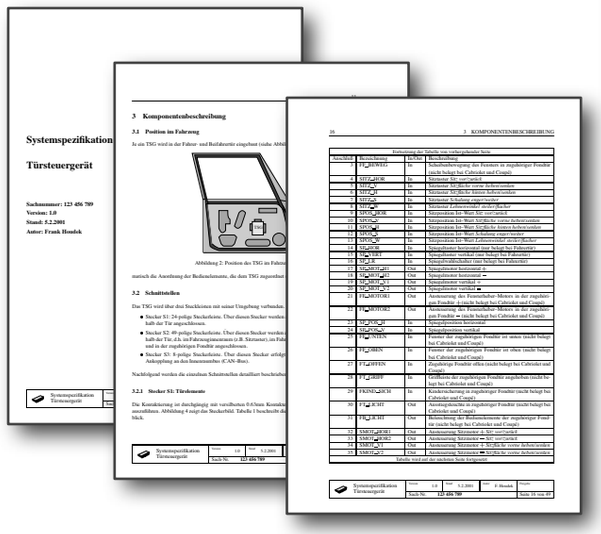
Domain-specific models: Modes of operation

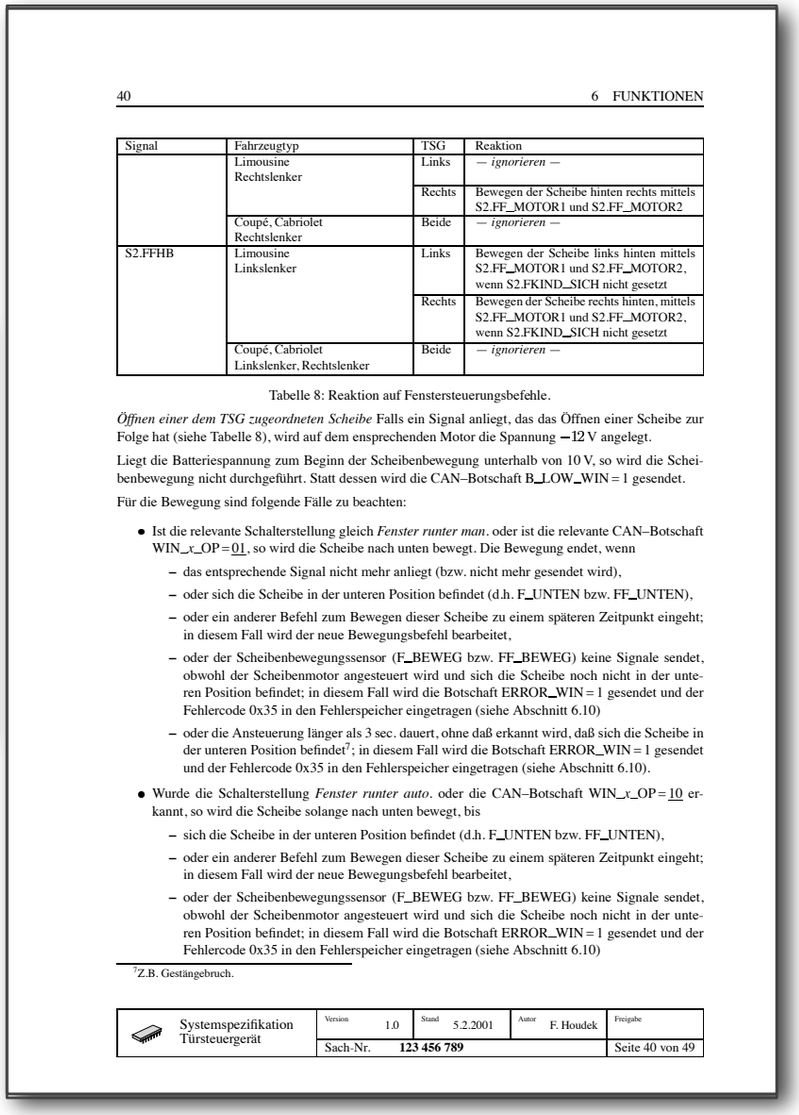


Improvement: Adequate models - Application domain vs technology domain

- Method: Extension of modeling techniques by domain concepts
- Further examples: Signal classes (state/event signals), timing constraints
- Improvement: More effective modeling, more targeted use

Example 2: Functional specification





40 6 FUNKTIONEN

Signal	Fahrzeugtyp	TSG	Reaktion
	Limousine Rechtslenker	Links	— *ignorieren* —
		Rechts	Bewegen der Scheibe hinten rechts mittels S2.FF_MOTOR1 und S2.FF_MOTOR2
	Coupé, Cabriolet Rechtslenker	Beide	— *ignorieren* —
S2.FFHB	Limousine Linkslenker	Links	Bewegen der Scheibe links hinten mittels S2.FF_MOTOR1 und S2.FF_MOTOR2, wenn S2.FKIND_SICH nicht gesetzt
		Rechts	Bewegen der Scheibe rechts hinten, mittels S2.FF_MOTOR1 und S2.FF_MOTOR2, wenn S2.FKIND_SICH nicht gesetzt
	Coupé, Cabriolet Linkslenker, Rechtslenker	Beide	— *ignorieren* —

Tabelle 8: Reaktion auf Fenstersteuerungsbefehle.

Öffnen einer dem TSG zugeordneten Scheibe Falls ein Signal anliegt, das das Öffnen einer Scheibe zur Folge hat (siehe Tabelle 8), wird auf dem ensprechenden Motor die Spannung −12 V angelegt.

Liegt die Batteriespannung zum Beginn der Scheibenbewegung unterhalb von 10 V, so wird die Scheibenbewegung nicht durchgeführt. Statt dessen wird die CAN–Botschaft B_LOW_WIN = 1 gesendet.

Für die Bewegung sind folgende Fälle zu beachten:

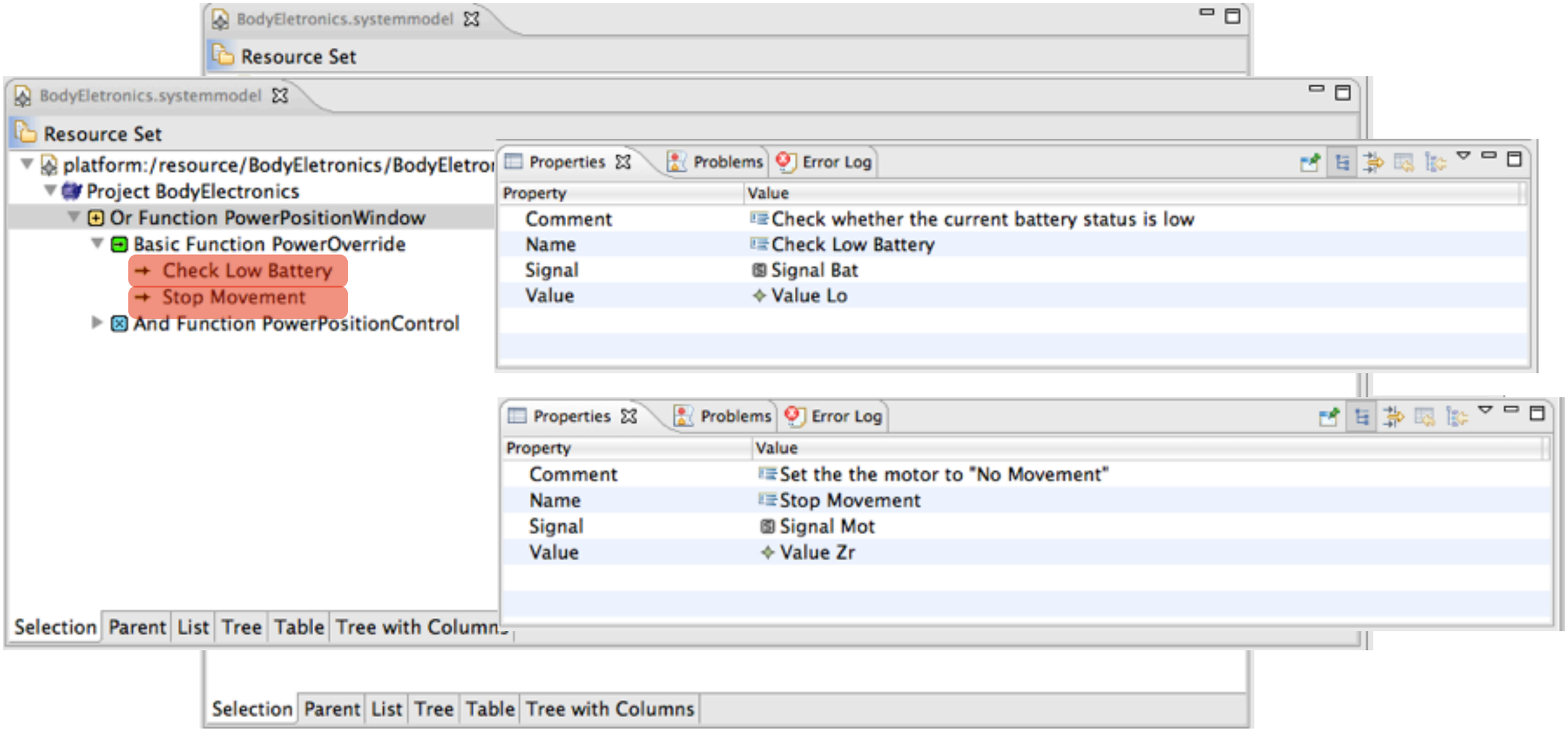
- Ist die relevante Schalterstellung gleich *Fenster runter man.* oder ist die relevante CAN–Botschaft WIN_*x*_OP = 01, so wird die Scheibe nach unten bewegt. Die Bewegung endet, wenn
  - das entsprechende Signal nicht mehr anliegt (bzw. nicht mehr gesendet wird),
  - oder sich die Scheibe in der unteren Position befindet (d.h. F_UNTEN bzw. FF_UNTEN),
  - oder ein anderer Befehl zum Bewegen dieser Scheibe zu einem späteren Zeitpunkt eingeht; in diesem Fall wird der neue Bewegungsbefehl bearbeitet,
  - oder der Scheibenbewegungssensor (F_BEWEG bzw. FF_BEWEG) keine Signale sendet, obwohl der Scheibenmotor angesteuert wird und sich die Scheibe noch nicht in der unteren Position befindet; in diesem Fall wird die Botschaft ERROR_WIN = 1 gesendet und der Fehlercode 0x35 in den Fehlerspeicher eingetragen (siehe Abschnitt 6.10)
  - oder die Ansteuerung länger als 3 sec. dauert, ohne daß erkannt wird, daß sich die Scheibe in der unteren Position befindet[7]; in diesem Fall wird die Botschaft ERROR_WIN = 1 gesendet und der Fehlercode 0x35 in den Fehlerspeicher eingetragen (siehe Abschnitt 6.10).
- Wurde die Schalterstellung *Fenster runter auto.* oder die CAN–Botschaft WIN_*x*_OP = 10 erkannt, so wird die Scheibe solange nach unten bewegt, bis
  - sich die Scheibe in der unteren Position befindet (d.h. F_UNTEN bzw. FF_UNTEN),
  - oder ein anderer Befehl zum Bewegen dieser Scheibe zu einem späteren Zeitpunkt eingeht; in diesem Fall wird der neue Bewegungsbefehl bearbeitet,
  - oder der Scheibenbewegungssensor (F_BEWEG bzw. FF_BEWEG) keine Signale sendet, obwohl der Scheibenmotor angesteuert wird und sich die Scheibe noch nicht in der unteren Position befindet; in diesem Fall wird die Botschaft ERROR_WIN = 1 gesendet und der Fehlercode 0x35 in den Fehlerspeicher eingetragen (siehe Abschnitt 6.10)

[7] Z.B. Gestängebruch.

State-of-art: Lacking models - No models of functional architecture

- Problem: Lacking models for usage view
- Consequence: No precise description of overall functionality

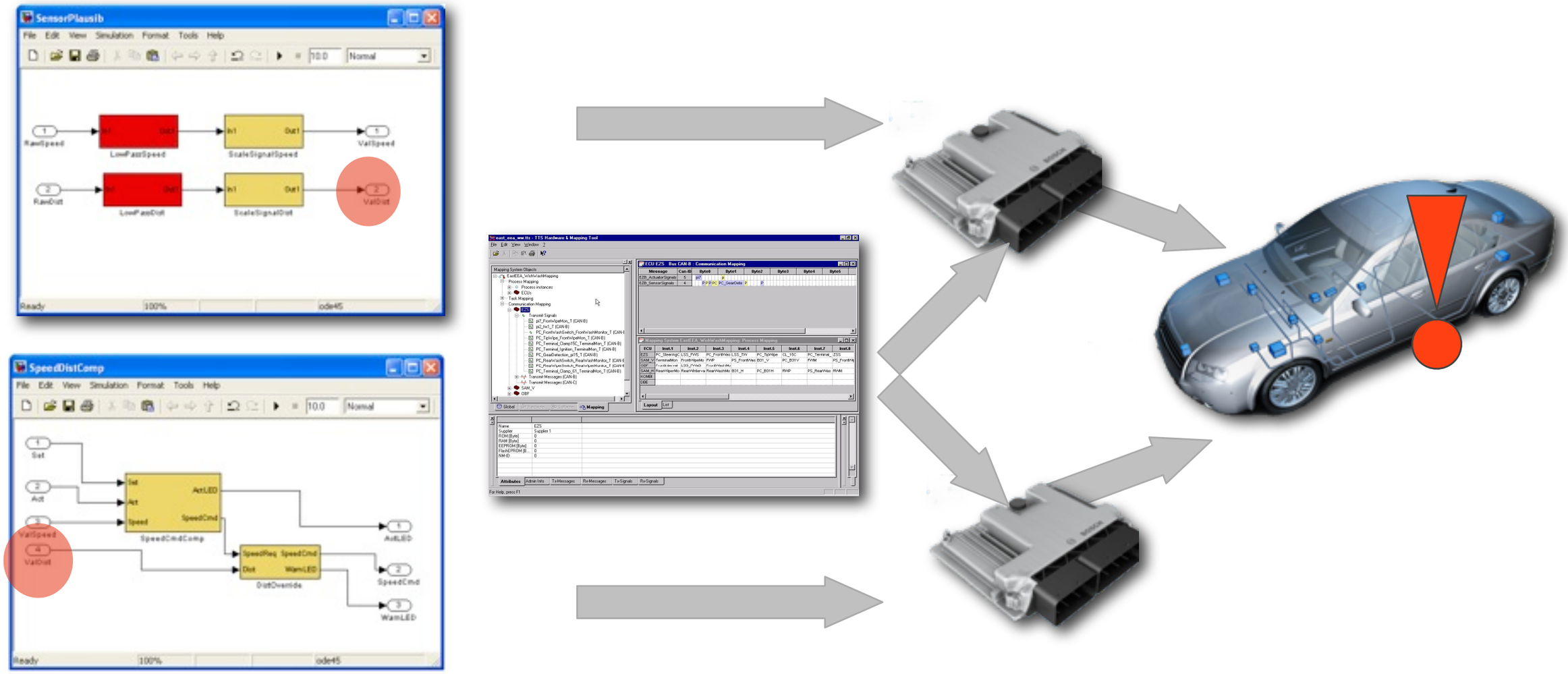
Models of early phases: Functions



Improvement: Explicit models - Structured instead of unstructured information

- Method: Extension of architectural modeling by functional level
- Further Examples: Product lines, timing requirements
- Improvement: More effective modeling and gezieltere Nutzung

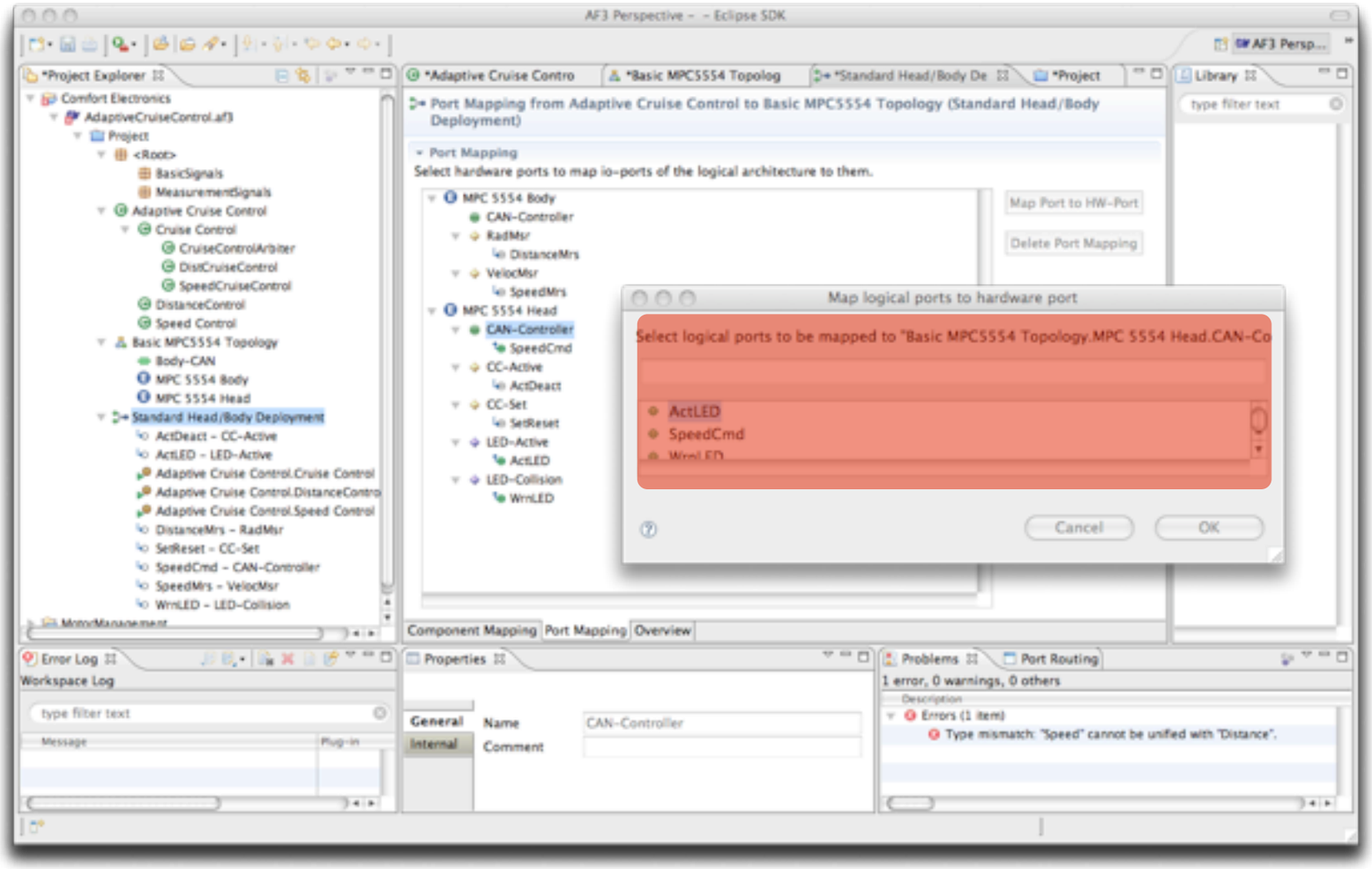
Example 3: Implementation



State-of-art: Isolated models - Models for individual steps of development processes

- Problem: Gaps in development process
- Consequence: Defects by contradicting decisions in later phases

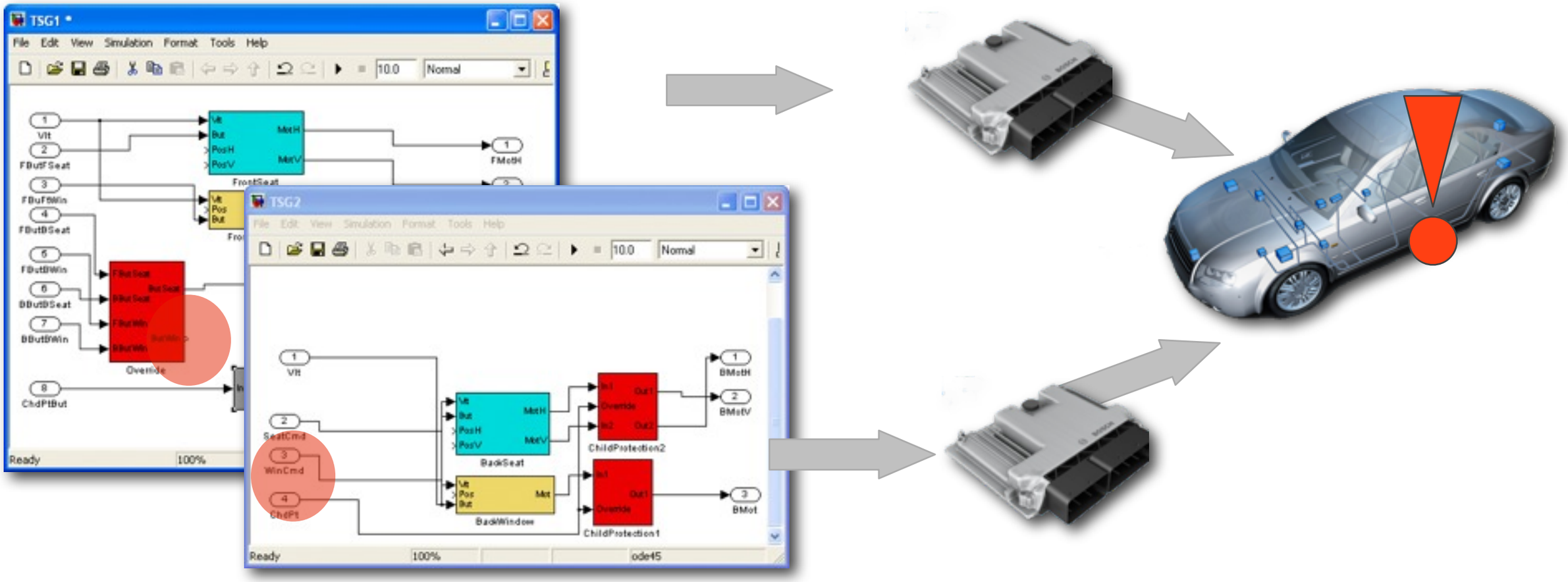
Integrated models: Component-topology-mapping



Improvement: Integrated models - Coherent description of system

- Methods: Integrated formalization of all system views
- Further Examples: Test case refinement (design-level/implementation level)
- Improvement: Avoidance of redundancy/inconsistency

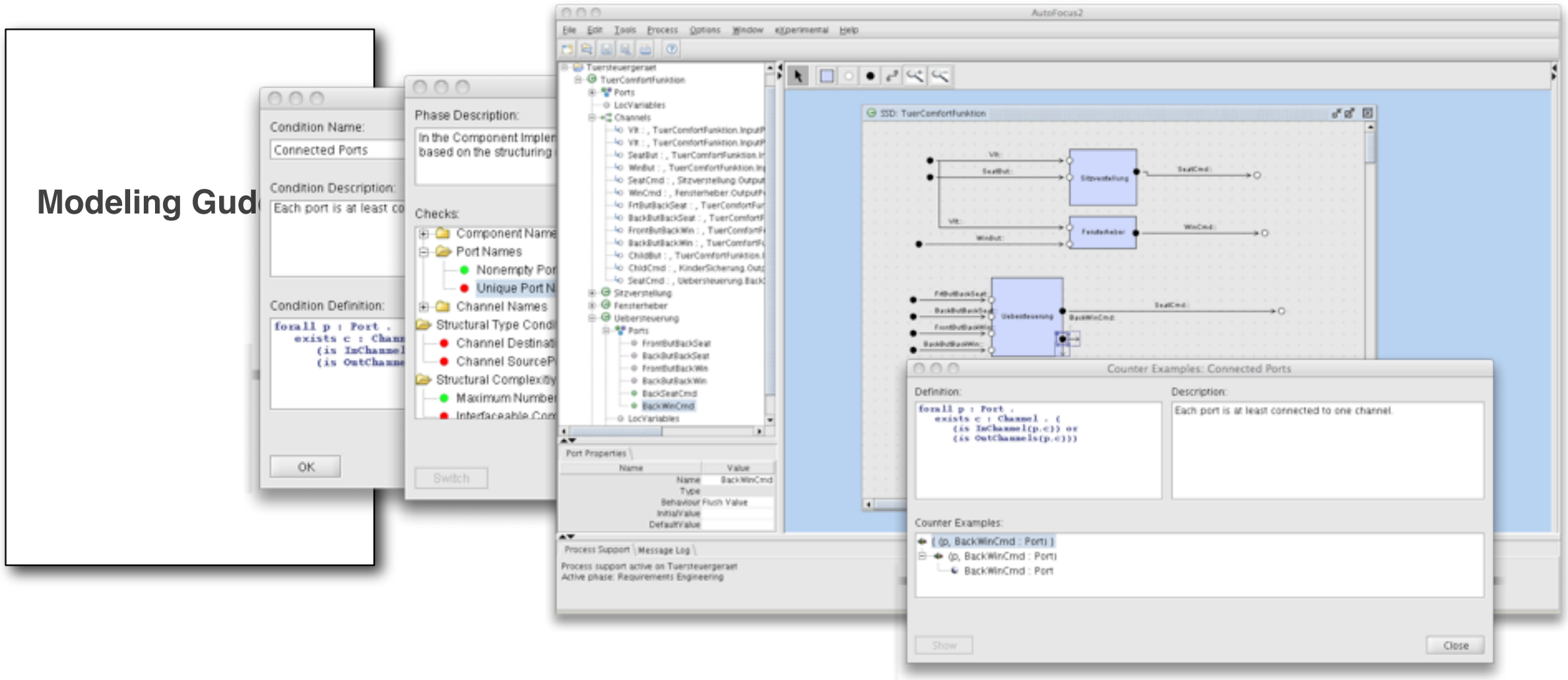
Example 4: Component desgin



State-of-art: Late quality assurance - Late detection of defects

- Problem: Lacking analysis support for early development steps
- Consequence: Conservation of early defects throughout the development process

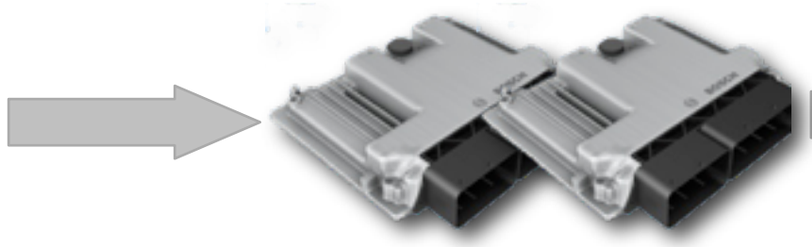
Analysis method: Checking of models

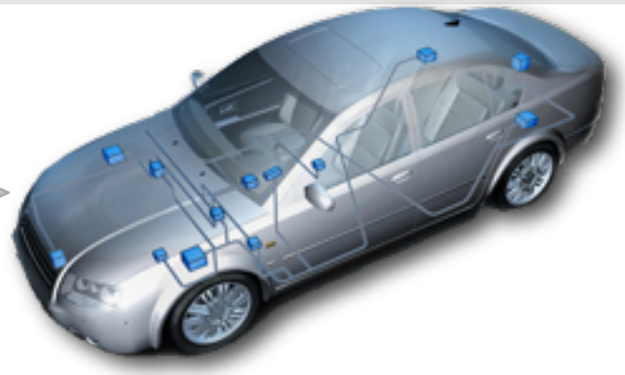


Improvement: Early Quality assurance - Early correct models via front-loading QS

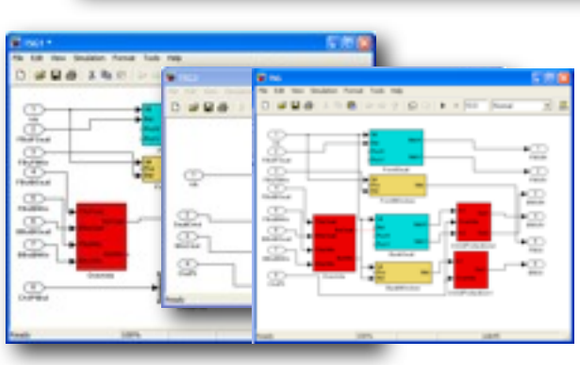
- Method: Automatic checks for model structure/model content
- Further examples: Checking of signal periods, checking of determinism
- Improvement: Early defect elimination

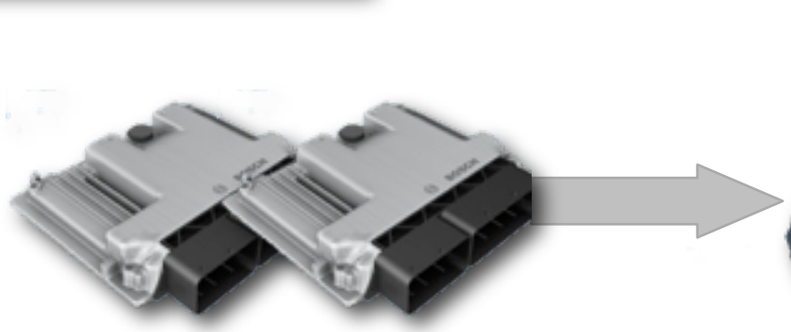
Example 5: Release planing

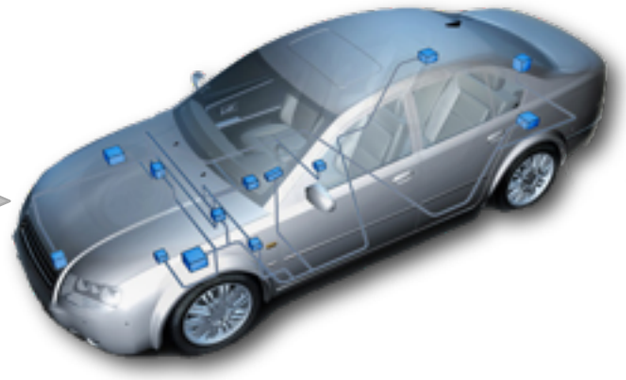




				Dependab.	Matrix			Integration Points		
Funktion	**PowerCtrl**	**ErrMgmt**	**DiagMgmt**	**LockDrv**	**LockPass**	**LightCtrl**	**InstCluster**	**I-Step 02**	**I-Step 02**	**I-Step 03**
PowerCtrl								x	x	x
ErrMgmt			X					**X**	**X**	**X**
DiagMgmt		X						x	x	x
WindowDrv	X							**X**	**X**	**X**
WindowPass	X					X		**X**	**X**	**X**
LockDrv	X					X			x	x
LockPass	X			X					**X**	**X**
MirrorDrv	X									**X**
MirrorPass	X									**X**
LightCtrl	X	X					X		x	x
InstCluster	X								x	x



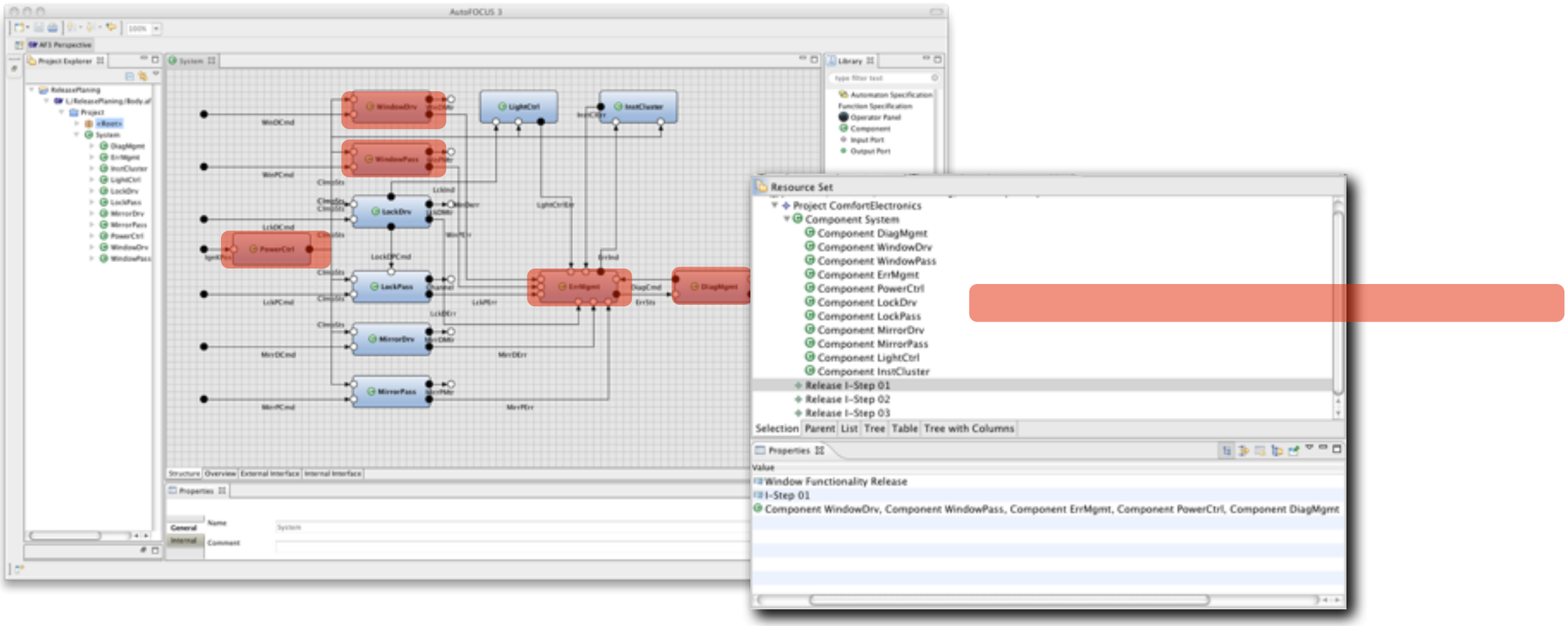




State-of-art: Manual release planing - Construction of complete sub-architectures

- Problem: Lacking automatization of release construction
- Consequence: Complication of release planing

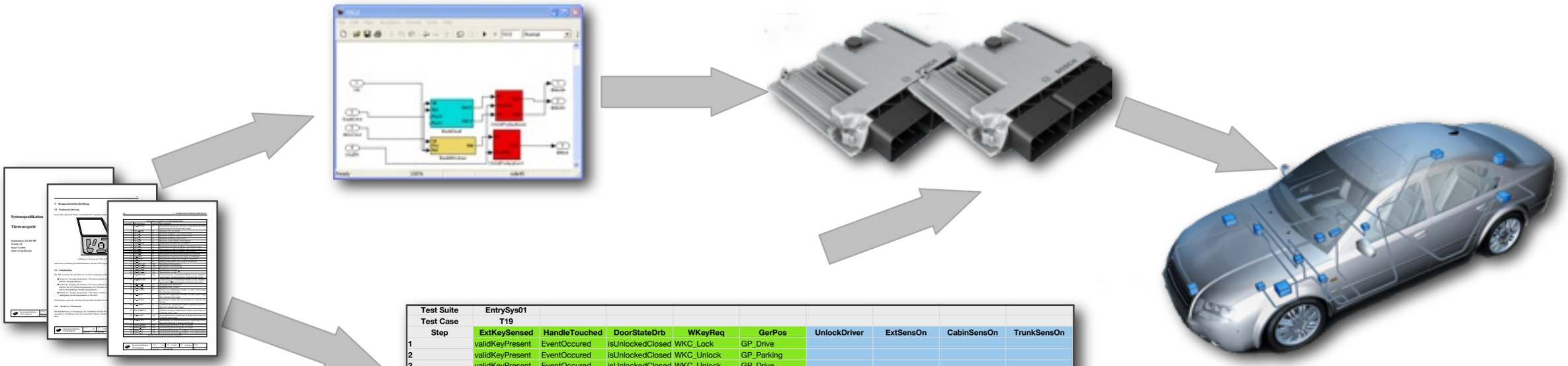
Synthesis method: Release plan generation



Improvement: Generation of release plans - Systematic incremental development

- Method: Automated model construction by generative search
- Further examples: Cable harness construction
- Improvement: Avoidance of miss-planing

Example 6: Verification

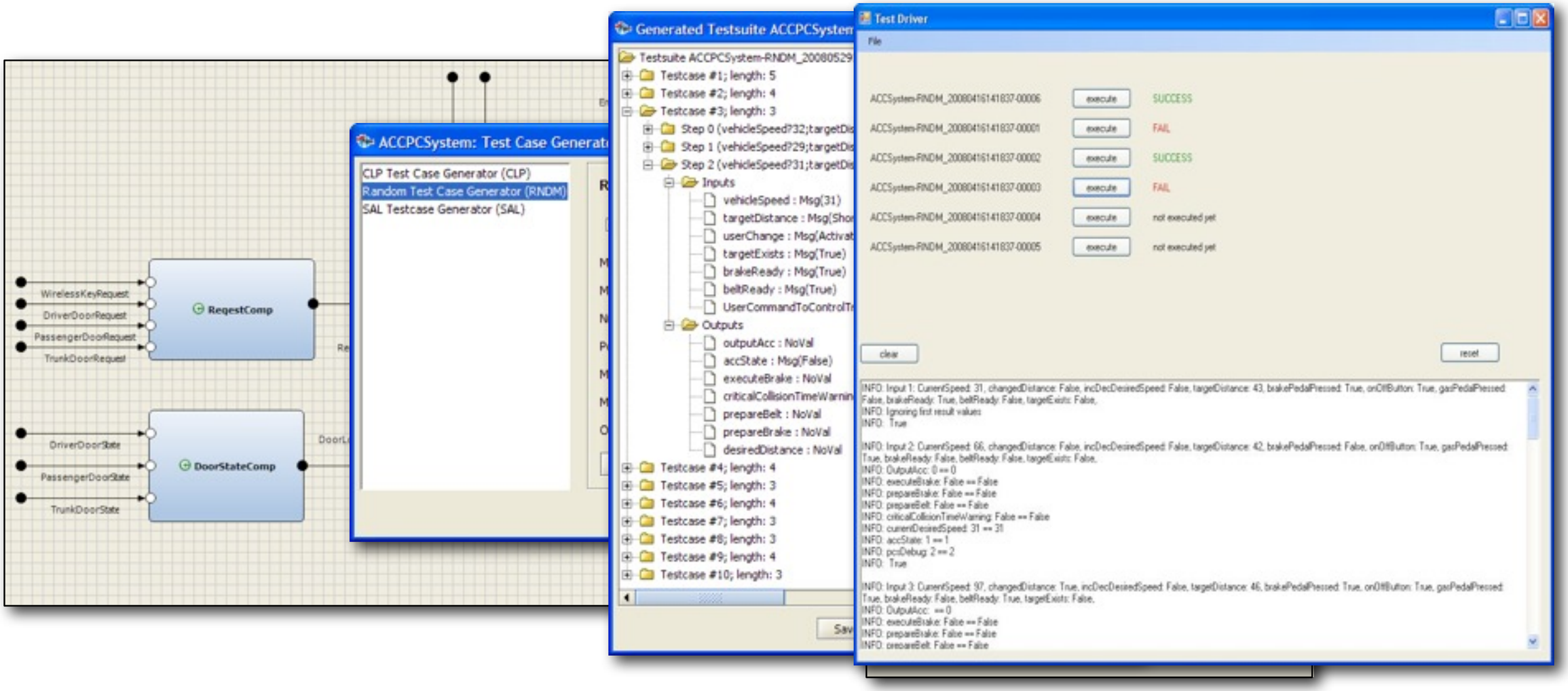


Test Suite	EntrySys01								
Test Case	T19								
Step	ExtKeySensed	HandleTouched	DoorStateDrb	WKeyReq	GerPos	UnlockDriver	ExtSensOn	CabinSensOn	TrunkSensOn
1	validKeyPresent	EventOccured	isUnlockedClosed	WKC_Lock	GP_Drive				
2	validKeyPresent	EventOccured	isUnlockedClosed	WKC_Unlock	GP_Parking				
3	validKeyPresent	EventOccured	isUnlockedClosed	WKC_Unlock	GP_Drive				
4	validKeyPresent	EventOccured	isLocked	WKC_Lock	GP_Parking	LC_Unlock			
5	invalidKeyPresent		isUnlockedClosed	WKC_Unlock	GP_Drive		SA_Off		
6	invalidKeyPresent		isUnlockedClosed	WKC_Unlock	GP_Drive			SA_On	SA_On
7	validKeyPresent	EventOccured	isUnlockedClosed	WKC_Lock	GP_Drive				
8	invalidKeyPresent	EventOccured	isLocked	WKC_Lock	GP_Neutral				
9	keyAbsent	EventOccured	isUnlockedClosed	WKC_Lock	GP_Parking				
10	keyAbsent	EventOccured	isUnlockedOpen	WKC_Lock	GP_Neutral				
11	validKeyPresent		isLocked	WKC_Lock	GP_Neutral	LC_Lock			
12	invalidKeyPresent	EventOccured	isUnlockedOpen	WKC_Unlock	GP_Parking		SA_On	SA_Off	SA_Off
13	validKeyPresent	EventOccured	isUnlockedClosed	WKC_Unlock	GP_Drive				
14	validKeyPresent	EventOccured	isUnlockedClosed	WKC_Lock	GP_Parking				
15	validKeyPresent	EventOccured	isUnlockedClosed	WKC_Unlock	GP_Neutral				
16	invalidKeyPresent		isLocked	WKC_Unlock	GP_Parking				
17	validKeyPresent	EventOccured	isUnlockedOpen	WKC_Unlock	GP_Neutral				
18	invalidKeyPresent	EventOccured	isLocked	WKC_Unlock	GP_Parking				
19	validKeyPresent	EventOccured	isLocked	WKC_Lock	GP_Parking				
20	keyAbsent	EventOccured	isUnlockedClosed	WKC_Unlock	GP_Neutral	LC_Unlock			
21	validKeyPresent	EventOccured	isLocked	WKC_Lock	GP_Neutral		SA_Off		
22	keyAbsent	EventOccured	isLocked	WKC_Lock	GP_Parking	LC_Unlock			
23	keyAbsent	EventOccured	isUnlockedClosed	WKC_Lock	GP_Neutral	LC_Lock			
24	validKeyPresent		isUnlockedOpen	WKC_Unlock	GP_Drive		SA_On		
25	invalidKeyPresent		isLocked	WKC_Unlock	GP_Drive				
26	keyAbsent	EventOccured	isLocked	WKC_Lock	GP_Neutral				
27	validKeyPresent	EventOccured	isLocked	WKC_Unlock	GP_Neutral				
28	keyAbsent	EventOccured	isLocked	WKC_Unlock	GP_Neutral				

State-of-art: Manual test case definition - Expensive test suite construction

- Problem: Lacking automatization of test case definition
- Consequence: Expensive and un-systematic verification of functionality

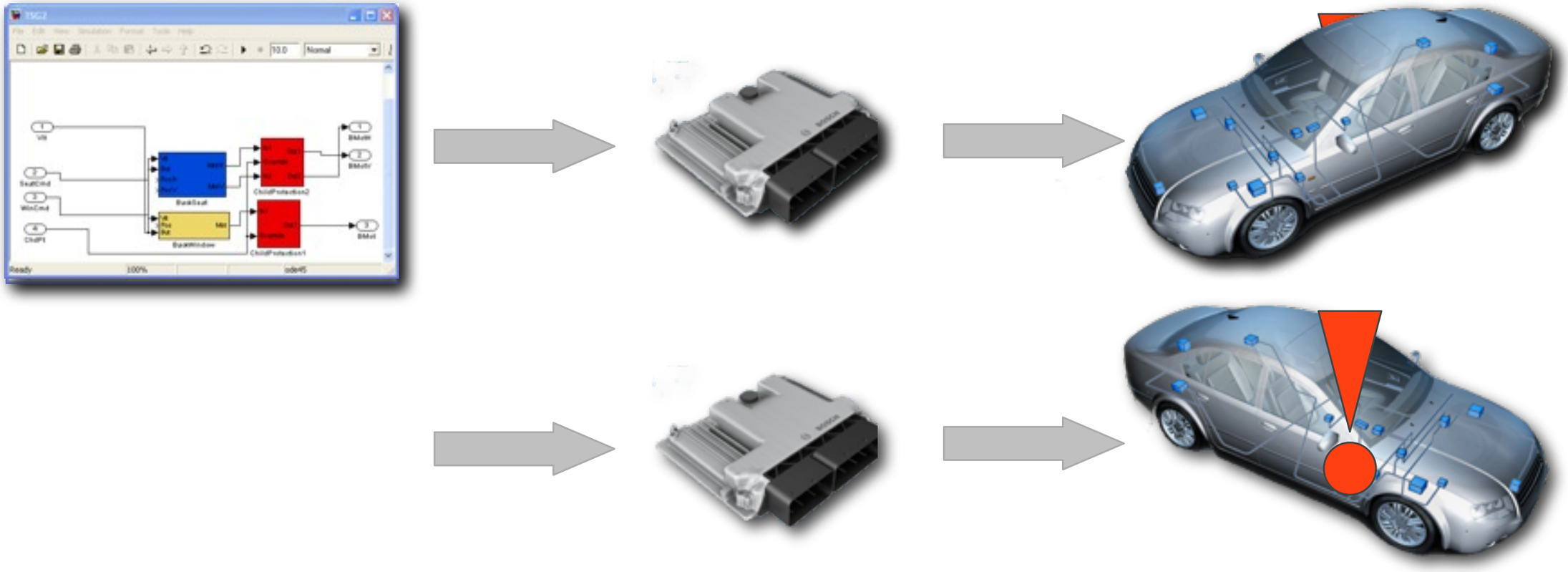
Synthesis method: Test case generation



Improvement: Synthesis of models - Optimization of test suit construction

- Method: Automatic model construction by generative search
- Further examples: Load tests, robustness tests, model validation
- Improvement: Improved efficiency of system verification

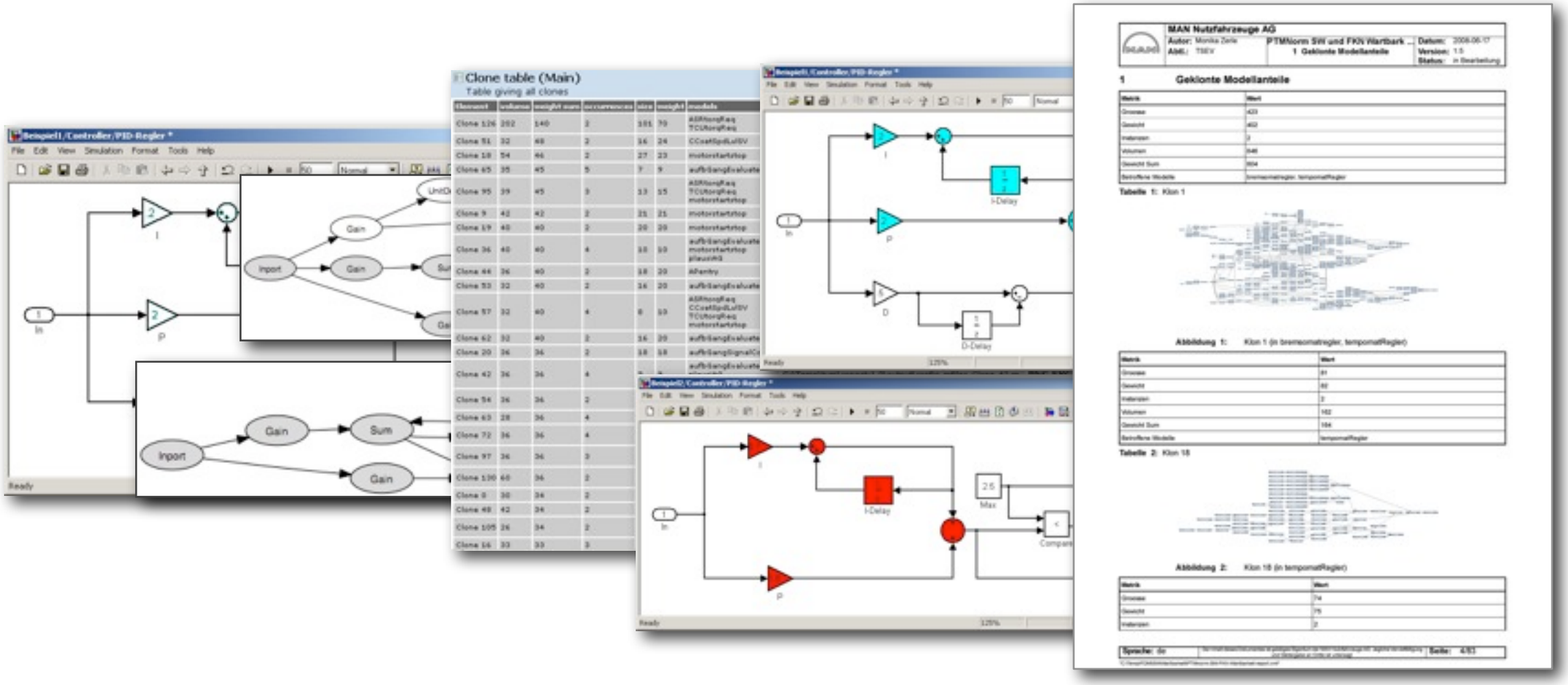
Example 7: System maintenance



State-of-art: Unsystematic use of identical parts - Introduction of change defects

- Problem: Lacking analyse support for maintenance
- Consequence: Complicated and expensive maintenance

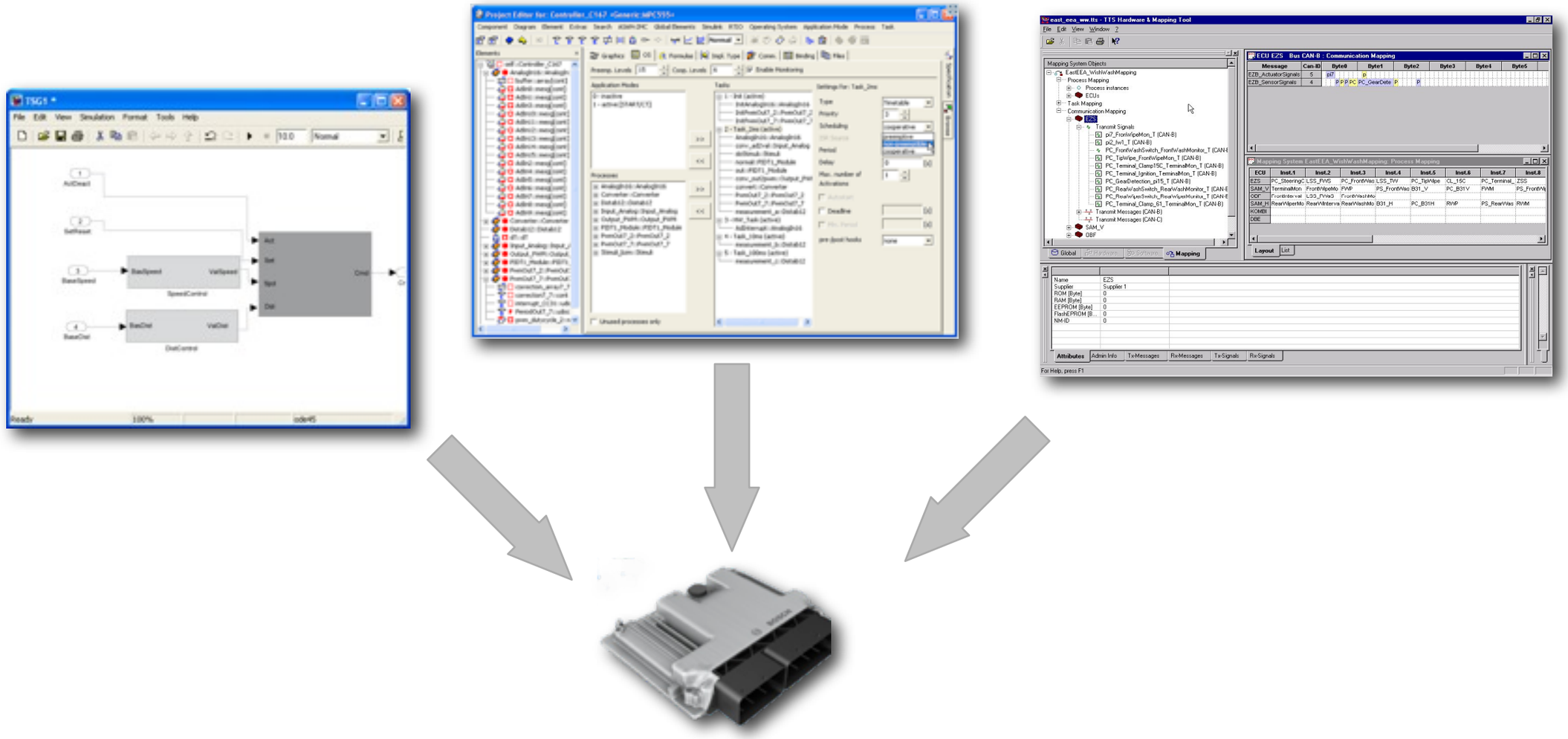
Analysis method: Clone detection



Improvement: Identification of identical parts - Avoiding change anomalies

- Method: Automated checks of architectural structure
- Further examples: Requirements clones, IP-libraries, product lines
- Improvement: Simplification maintenance

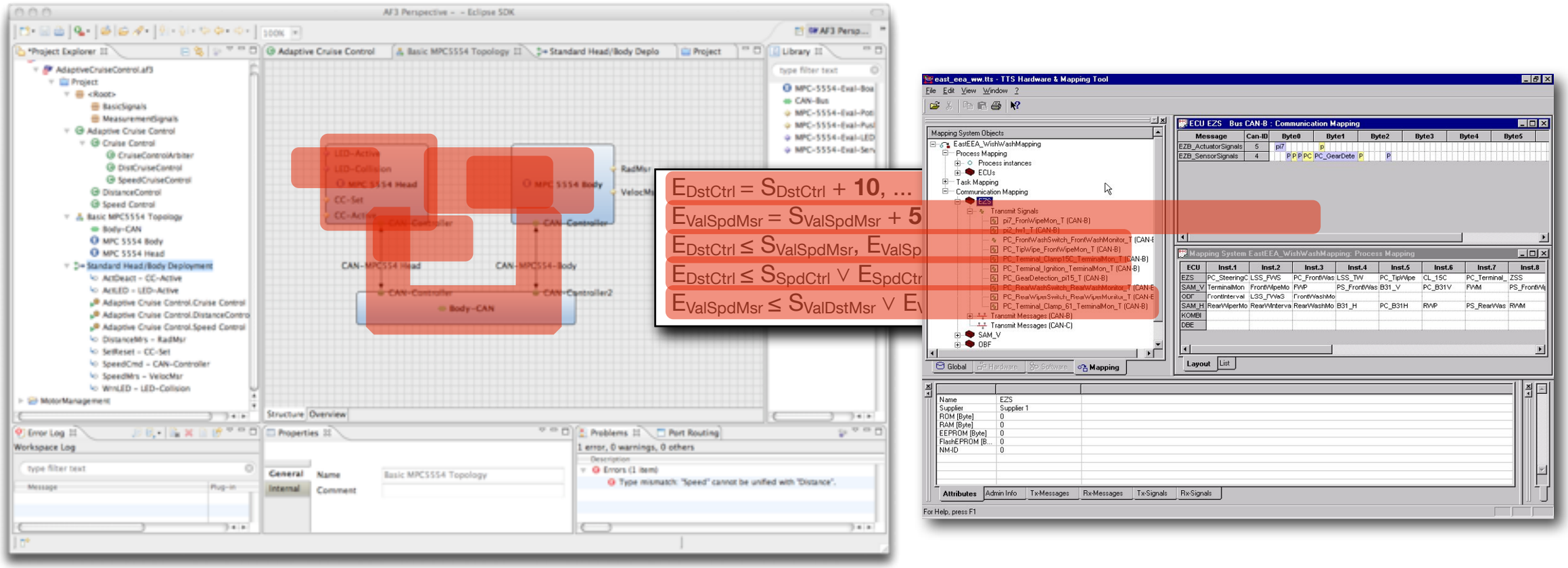
Example 8: System integration



State-of-art: Manual integration steps - Constructed models

- Problem: Lacking extensive automatization of design steps
- Consequence: Limited support for design space exploration

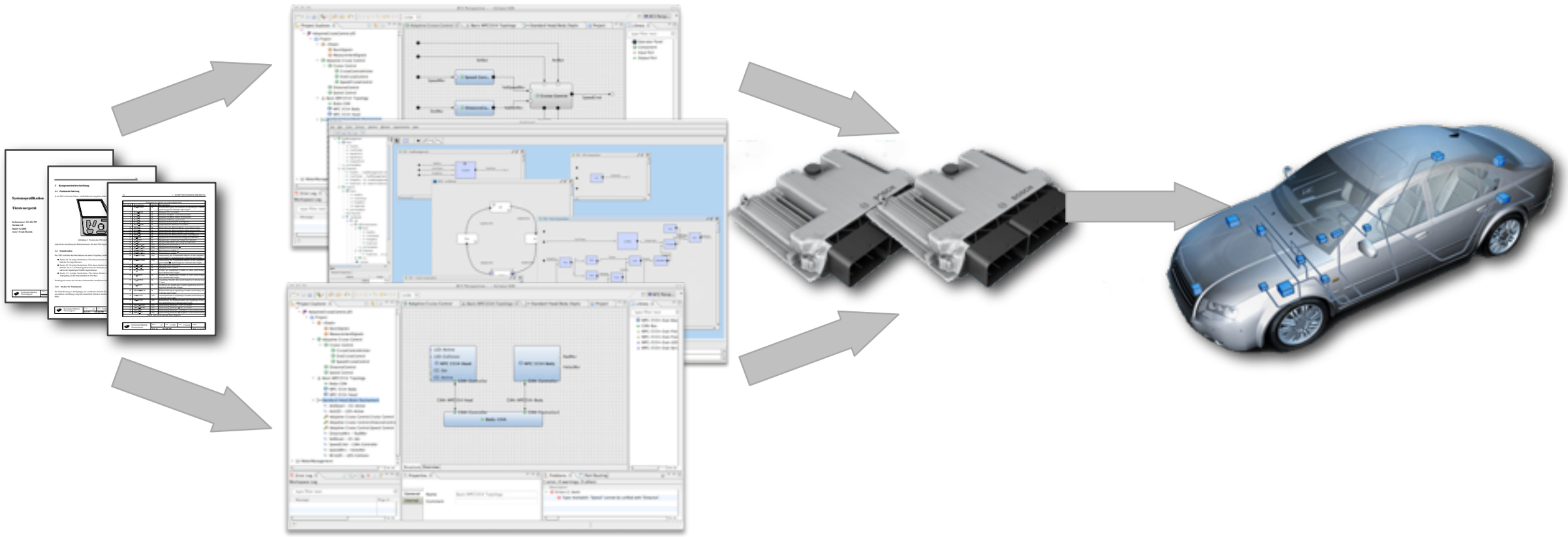
Synthesis method: Schedule generation



Improvement: Automated development steps - Generated modes

- Method: Automated model construction by generative search
- Further examples: (Mixed-criticality-)deployment generation, code generation
- Improvement: Accelerated exploration of design alternatives

Models in the development process: Principles



Improvement of effectivity and efficiency: Models support

1. Adequate modeling
2. Phase-integrating development
3. Front loading analyse methods
4. Synthesis of models

➡**Implementation of tools for an integrated development process**

Models in application: Examples

Test via application development:

- Driver-Assistance: Adaptive cruise control, parking assistant
- Infotainment: Use of entertainment system
- Body electronics: Power window, keyless entry system