

## Production Code Generation in the Model Based Development Process

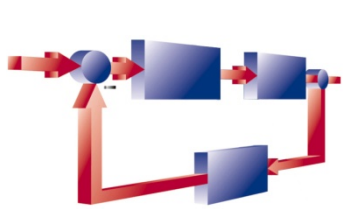


Frank Hein

dSPACE GmbH · Rathenastr. 26 · 33102 Paderborn

8. June 2010

# Today's Development Process - The V Cycle



**Control Design**

## TargetLink

C code generation from MATLAB/Simulink

Generated C code is as efficient as hand code

High reliability, readability and reproducibility of generated C code



**Calibration**



**Rapid Control Prototyping**

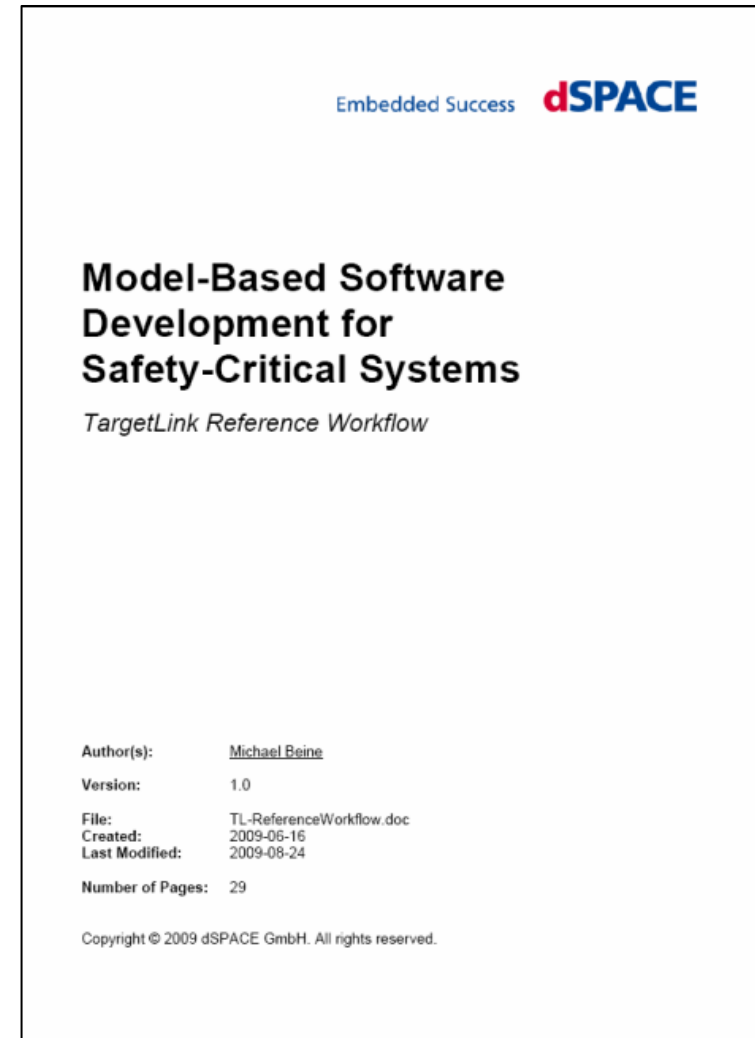


**Hardware-in-the-Loop Simulation**

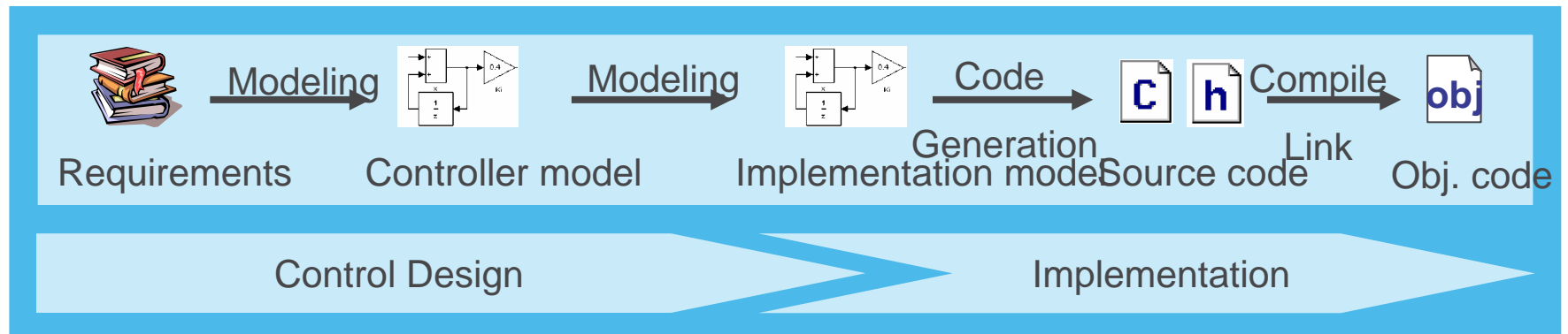


**Automatic Production Code Generation**

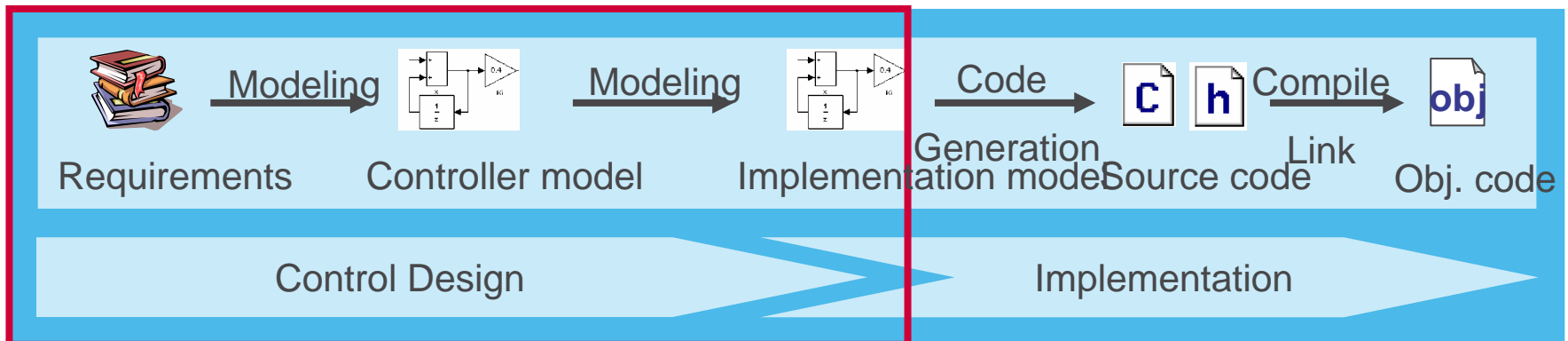
- TargetLink Reference Workflow document is available from dSPACE
- This TÜV **approved** reference workflow
  - is based on **best practices** already **proven** in safety-related projects
  - addresses various aspects relevant for the development of safety-related software with a special focus on verification and validation
  - provides guidance on how to fulfill functional safety requirements with model-based development methods and tools
- conforms to IEC 61508 and ISO 26262
- supports argumentation to justify model-based development to quality/safety personnel



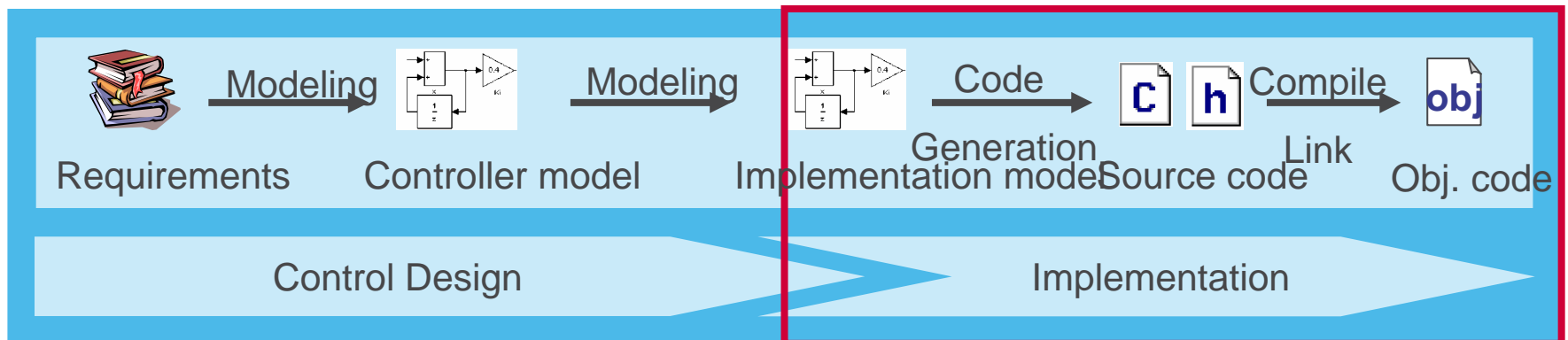
- Model-based software development from textual requirements via executable specifications and code generation to executable object code



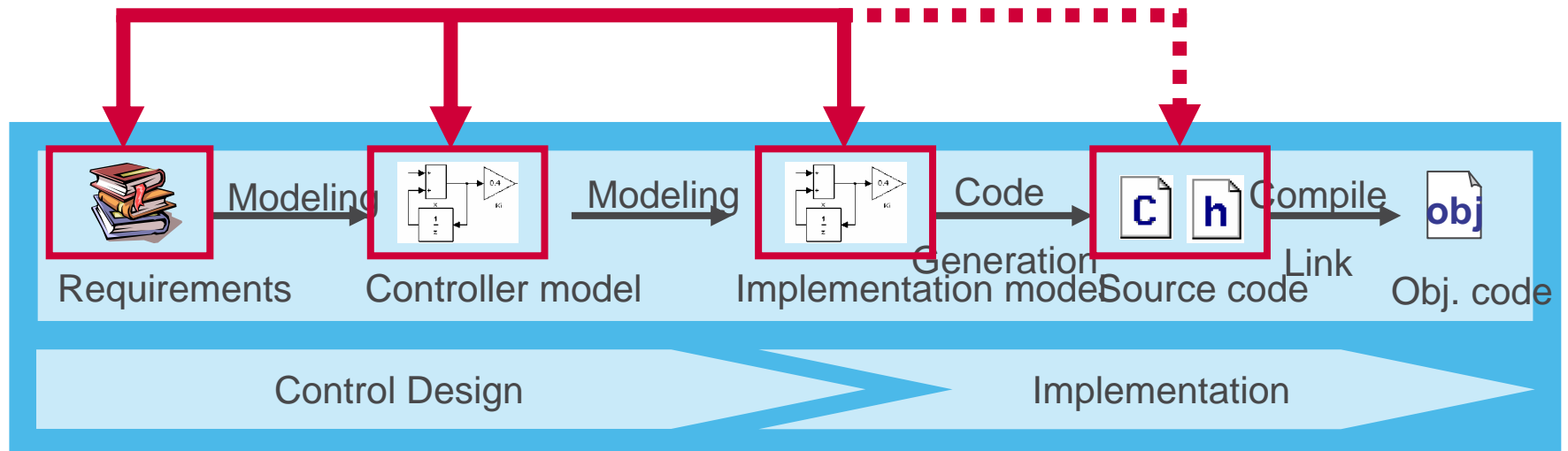
- Design Verification
  - Requirements traceability
  - Modeling guidelines (MAAB, MISRA AC TL) and guideline checking
  - Formal verification
- Simulation
- Model coverage



- Code Verification
  - Back-to-back testing between simulation model and code
  - Code coverage
  - Code correctness (PolySpace, EmbeddedTester)
  - Code review
  - Coding standards (MISRA C) and conformance checks

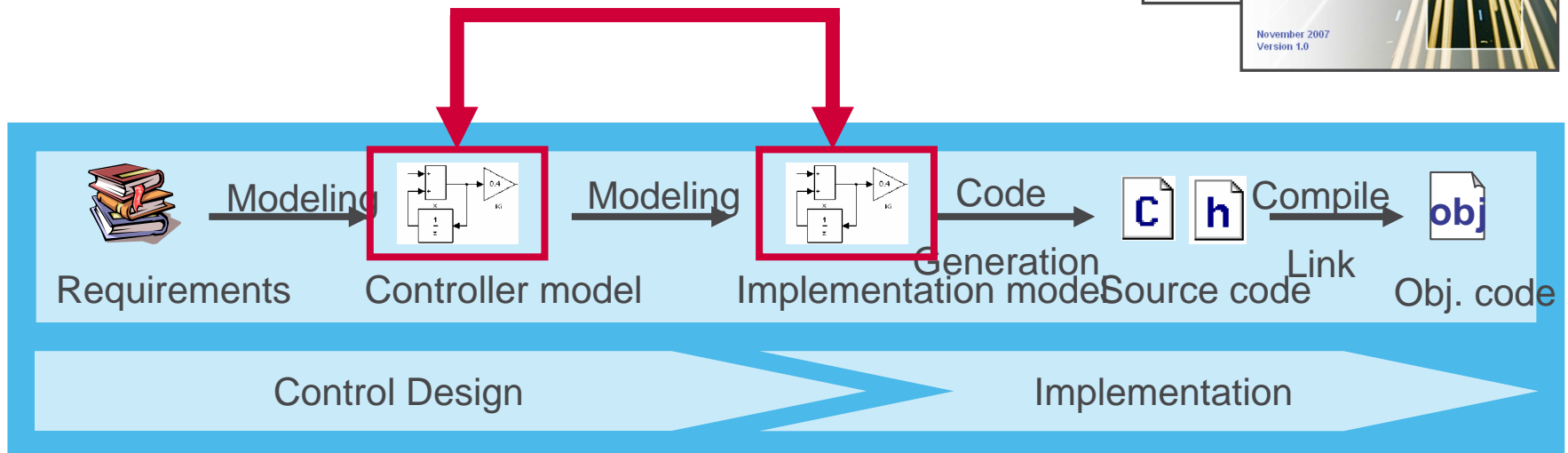
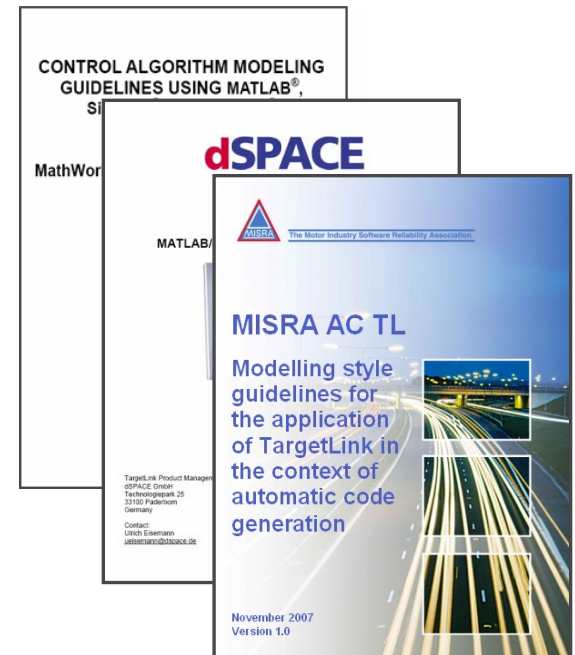


- Requirements traceability
  - Via MathWorks' Requirements Management Interface (RMI) (part of Simulink Verification & Validation)
  - Requirements from DOORS, Word, etc. can be linked with model



- Modeling Guidelines

- Guidelines providing best practices for safe modeling
- MAAB Controller Guidelines, MISRA Modeling Guidelines for TargetLink (MISRA AC TL), TargetLink Guidelines
- Check that guidelines are followed





- TÜV SÜD Automotive GmbH, German certification authority, evaluated TargetLink, with the following result:
- TargetLink code generator is fit for purpose to develop safety-related software according to
  - IEC 61508
  - ISO DIS 26262
  - derivative standards such as EN 50128<sup>1</sup>.
- TÜV granted the following certificate

**CERTIFICATE**  
No. Z10 09 08 71483 001

**Holder of Certificate:** dSPACE GmbH  
Technologiepark 25  
33100 Paderborn  
GERMANY

**Factory(ies):** 71483

**Certification Mark:**

**Product:** Software Tool for Safety Related Development

**Model(s):** TargetLink

**Parameters:** The code generator is fit for purpose to develop safety related software according to IEC 61508 and/or ISO DIS 26262.  
The report no. DP74761C is a mandatory part of this certificate.

**Tested according to:** IEC 61508-3:1998 (Fitness for purpose)  
ISO DIS 26262:2009 (Fitness for purpose)

The product was tested on a voluntary basis and complies with the essential requirements. The certification mark shown above can be affixed on the product. It is not permitted to alter the certification mark in any way. In addition the certification holder must not transfer the certificate to third parties. See also notes overleaf.

**Test report no.:** DP74761C

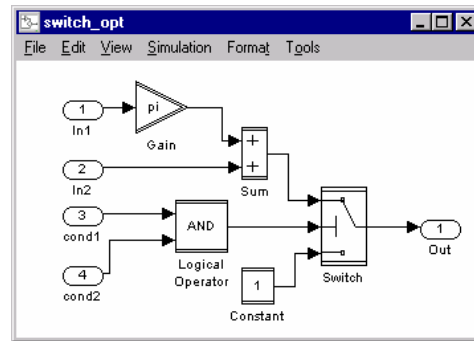
**Date,** 2009-08-20

Page 1 of 1

TÜV SÜD Product Service GmbH - Zertifizierstelle - Ridlerstrasse 65 - 80339 München - Germany

TÜV®

<sup>1</sup> EN 50128, standard for safety-related railway software, for example, is considered as a sector-specific standard derived from IEC 61508.



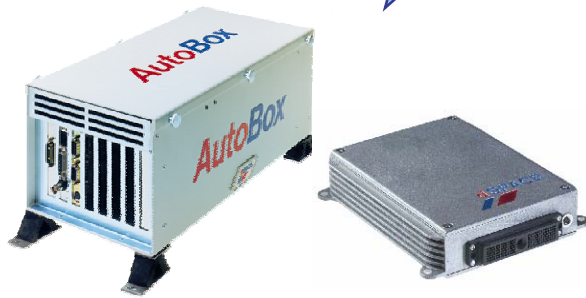
MATLAB / Simulink

TargetLink

Optimized production code directly from Simulink model

- Less memory consumption
  - Processor optimized code
- Generation of fixed-point and floating-point code

Rapid prototyping code generator  
(Real-Time Workshop)



Prototyping hardware



Electronic control unit

## Problem

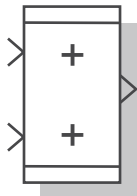
- Many different implementations for one algorithm exist
- Optimal implementations differ between compiler/processors

## Solution: Carefully designed code patterns

- Code pattern library
- Contains best code pattern for each compiler/processor combination
- Optional for:
  - ANSI C only
  - Assembly language and compiler specific C

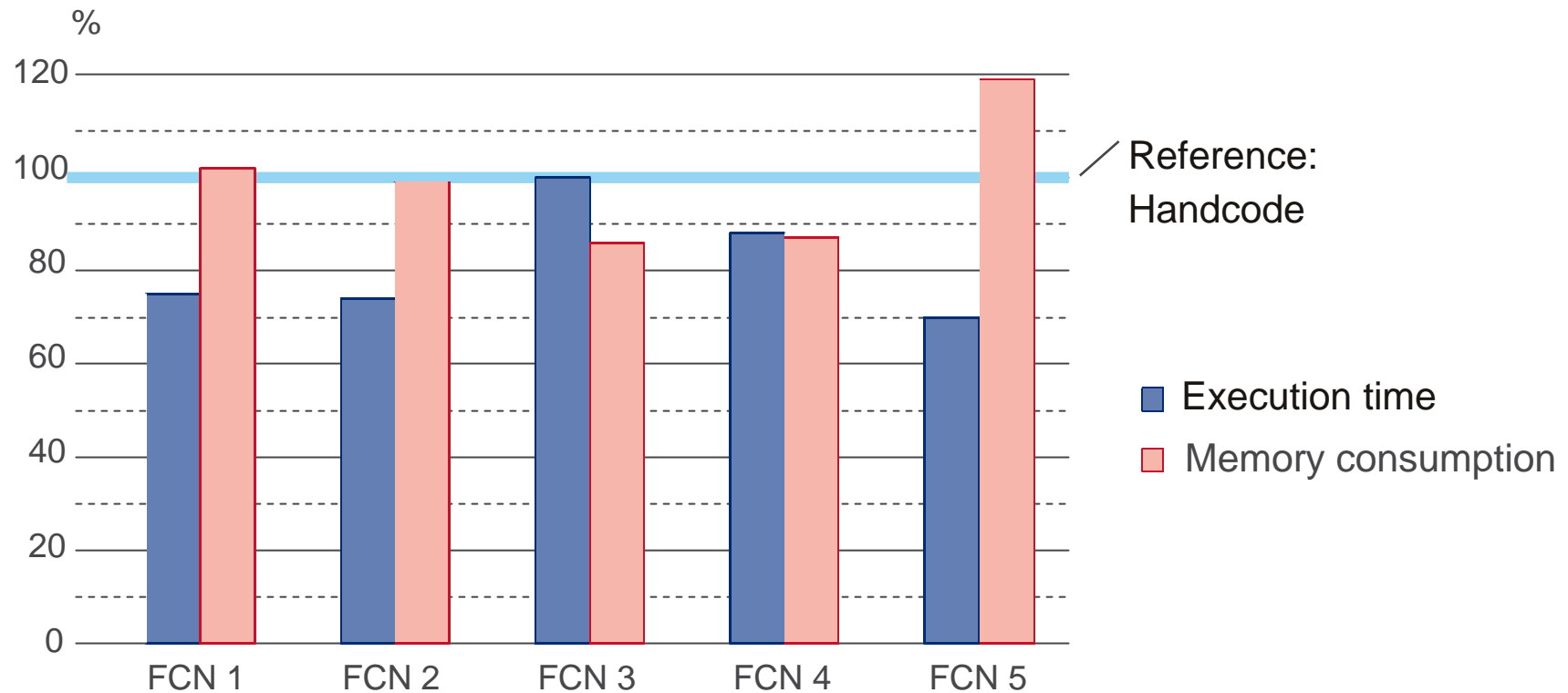


## Example:

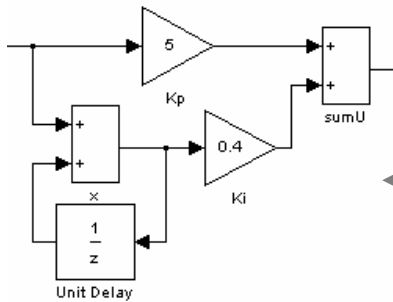


16 bit Addition with saturation

TargetLink generated code compared with manually written c

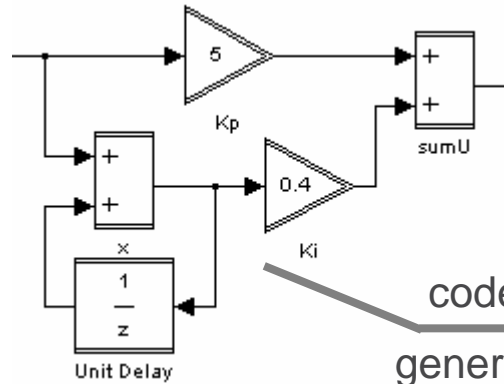


## Simulink Blockset



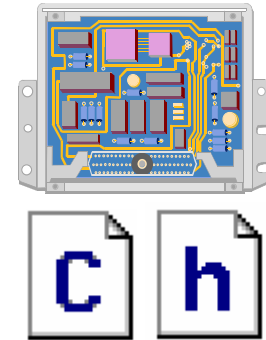
automatic  
replacement

## TargetLink Blockset



code  
generation

## ECU



**Host  
floating-point  
simulation (MIL)**

### algorithm design

- behavior validity checks
- scaling (auto or manual)
- overflow detection
- reference traces

**Host  
Prod. Code  
simulation (SIL)**

### fixed-point effects

- quantization errors
- saturation and overflow
- enter implementation options

**Target  
Prod. Code  
simulation (PIL)**

### code validation

- profile execution time
- measure stack size
- measure RAM / ROM
- final verifications

- **Highly efficient**
  - Best Code Generator on the market  
(that's what numerous customer evaluations have consistently shown, what customers consistently state)
  - No matter if **Fixed-Point** or **Floating-Point**
  - Dedicated Fixed-Point code generation
  - Optimized Code Generation for numerous processor/compiler combinations
- Very well **readable** code
- Very good **traceability** between model and code
- High Quality, MISRA Compliance

- StARS Micro-hybrid concept developed with dSPACE aid
- Reversible, belt-driven starter-alternator to reduce fuel consumption and CO2 emissions
- Code generation with TargetLink
- Validation bench with dSPACE Simulator



---

*“For validation, the service provided by dSPACE is not only hardware and software, but also intellectual, with the creation of a solution specific to our application.”*

**Sébastien Roue, Valeo**

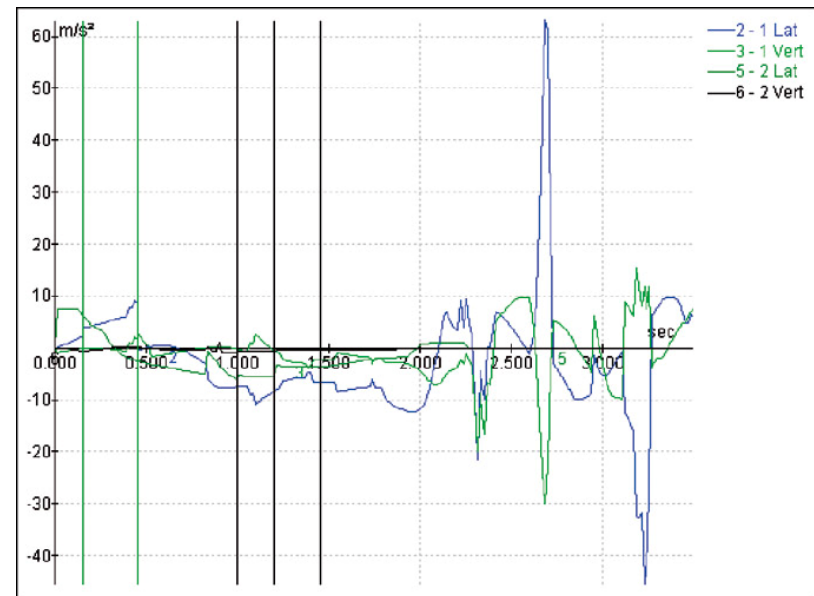
---



- Diesel engines emit soot particles and nitrogen oxide ( $\text{NO}_x$ )
- Idea: Reducing the emissions by burning off the soot
- Solution: Controlled continuously regenerating trap (CCRT)
- Result: Particle emission reduced by up to 90%



- Delphi's developments for passenger safety systems
- A rollover detection algorithm was implemented and is now in production
- Significant code improvements were achieved using TargetLink's code profiling techniques



Simulated sensor signals show the lateral and vertical accelerations to be analyzed by a rollover detection algorithm

## Autocode Success Story

- Code-Profiling led to handcode-efficiency RAM and throughput (execution time) down by 75%
- Integrated in 1.5 days
- In production
- Statements from software engineers:

*“The generated code was easy to understand. Every comment and variable name was a great help.”*

*“In my opinion it saved a lot of time. It is a good base for developing target C code. The main backbone of code was almost unchanged.”*



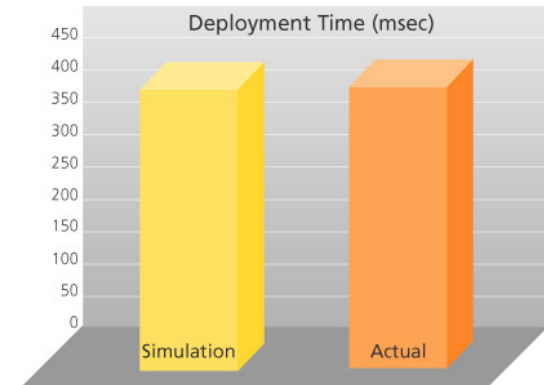
TargetLink was used to generate efficient fixed-point C code from Simulink and Stateflow models

## Equipment and Methods

- TargetLink
- Target Optimization Module
- Motorola HC12 Evaluation Board for code profiling
  - Throughput (execution time)
  - RAM (including stack)
  - ROM
- Back-to-Back-Tests (MIL, SIL and PIL simulations) at earliest stage

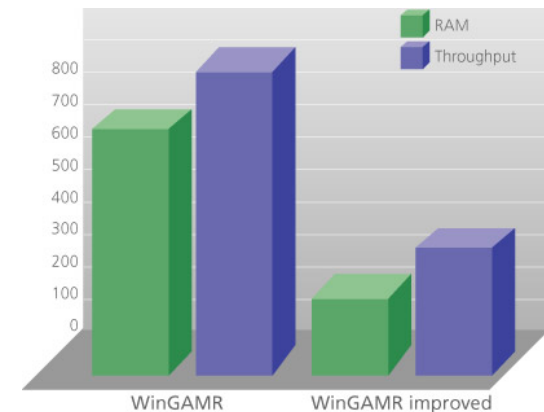
## Autocode reduces risks

- No transcription errors
- No spec misinterpretation
- Match with model performance

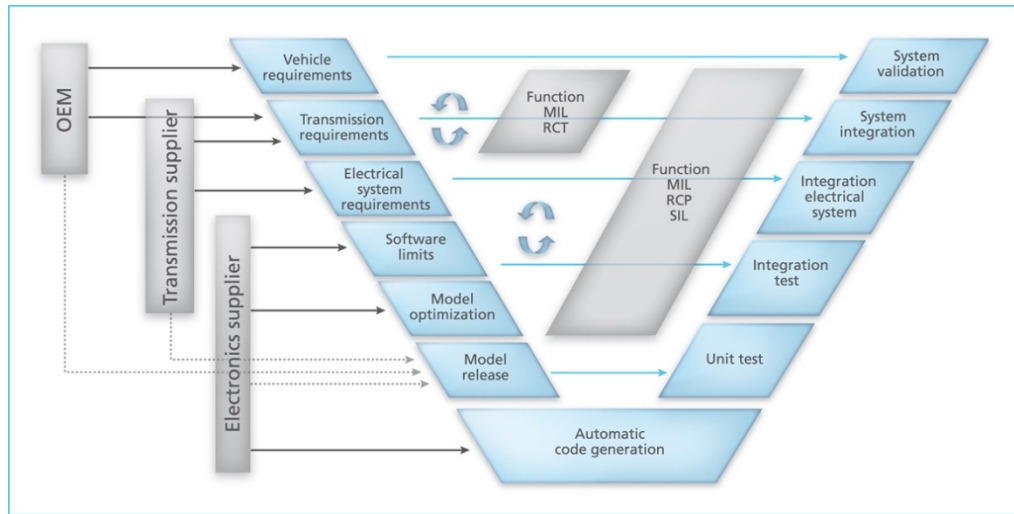


The deployment time measured for the autcoded algorithm matched the simulation

Insights into the algorithm gained from using TargetLink's code analysis tools and an evaluation board led to huge improvements in RAM consumption and throughput (execution time)



# Continental: Processes and Methods for Using TargetLink in Model-Based Development of Transmission Function Software



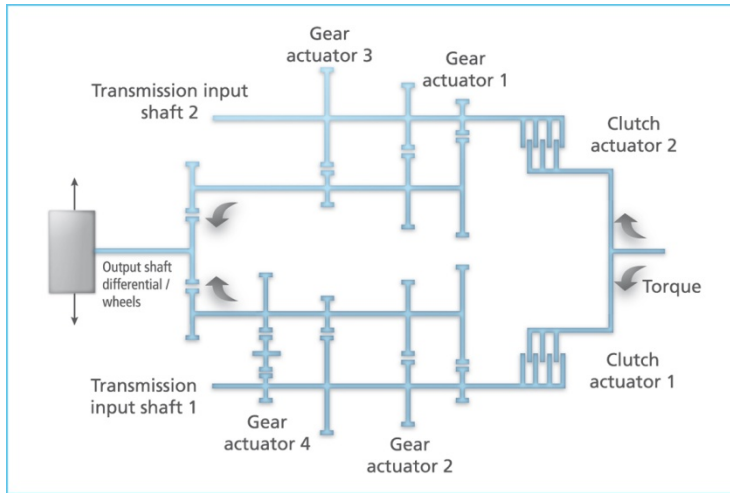
*The development cycle in the Business Unit Transmission.*

*“Within the framework of an in-house study by the OEM, the result was that the handwritten code cannot measure up to the code generated by TargetLink in terms of freedom from error.”*

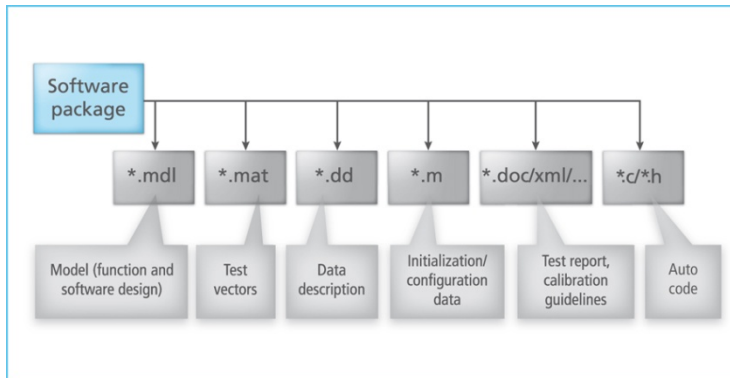
**Georg Grassl, Continental AG**

- Function development and Rapid-Prototyping with AutoBox and MicroAutoBox
- Demonstrating new functionalities in demo vehicles in a short space of time was decisive to the success of the project
- Front-loaded tests made it possible to assess the customer's requirements quite early
- Automatic code generation with TargetLink
- Quality assurance by model-based tests
- Structural coverage tests supported by tools for test vector generation
- The quality goal of 100% coverage of the fixed-point code (highest test coverage level: modified condition/decision coverage, MC/DC) was achieved

# Continental: Processes and Methods for Using TargetLink in Model-Based Development of Transmission Function Software



- Function reuse
- Adapted code generation for device-specific calibration methods
- Partitioned design in a multi-user environment



- Model, data description, test vectors (functional and structural), test areas, and fine-tuning guidelines as a package in the configuration management

- Development objectives of the first production project
  - Maximum CPU utilization of 15 %
  - Maximum ROM assignment of 100 kByte

With TargetLink, the values stayed far below these levels, whereas the size of the automatically generated function software was:

- Maximum 10 % CPU assignment
- 60 kByte compiled code
- 50 modules
- 20 % ROM assignment of the entire application software

- Results and important findings from the second production project (DCT)
  - 100 % of function software for the DCT project was developed by model-based design and autocoded with TargetLink
- Size of function software for the high volume project
  - 250 kByte compiled code
  - 120 model libraries

*“For the project performed with TargetLink, the time-to-market was even shortened, which our customer applauded.”*

**Georg Grassl, Continental AG**

*“Using model-based development and automatic code generation led to an exceptionally high number of software deliveries precisely on schedule.”*

**Georg Grassl, Continental AG**



“The C files generated by TargetLink are perfect in quality, the C code is well commented and optimized for the processor type used in the ECU.”

*Jochen Diehm and Dr.-Ing. Stefan Günther, Conti Temic*

„We don't find coding errors – we find modeling errors.“

User Quote

„The generated code fulfills the high aviation requirements.“

*Andreas Alaoui (Nord-Micro)*

„Our requirements with respect to code quality, readability and integration capability were fully met. ...

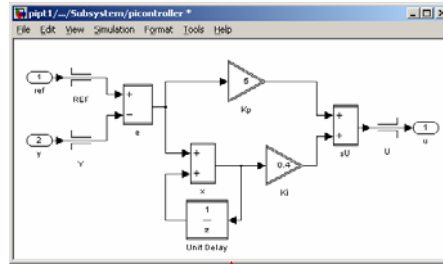
... TargetLink code consistently proved to be reliable.“

*Andreas Wilken, Claas KGaA*

“The code is reliable. As if it was handcoded.”

*Philipp Queins, DaimlerChrysler*

**dSPACE**  
TargetLink



**dSPACE** DD Manager

Property	Value
Description	"Motordrehzahl"
NameTemplate	"n_mot"
Kind	Plain
StructComponent	0
Class	<DISP>
Type	<UInt16>
Scaling	<S_N_MOT_U16>
Width	[]
Value	900
Min	0
Max	8000
Address	-

Save  
Load  
Include



Import

Export

ASAP2



DBC

XML

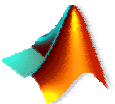
OIL



XLS



M  
MAT  
ML  
WS



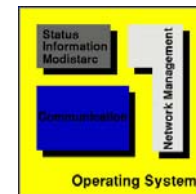
SW-C  
Desc.



cal\_vars.dd    disp\_vars.dd    global\_vars.dd    scalings.dd

- dSPACE is an automotive company
- **TargetLink addresses and focuses on automotive needs**
- Automotive  $\mu$ C support:
  - e.g. Renesas SH2, M32R, NEC V850, MPC55xx, TriCore
- Early and competent support of relevant standards  
Examples:

- ASAM
- OSEK
- MISRA
- AUTOSAR





- MISRA published official MISRA modeling guidelines for TargetLink
  - Special focus on functional safety
  - Avoid pitfalls on both, the modelling level as well as the code level
- TargetLink is the only code generator for which official MISRA guidelines exist
- Version 1.0, published November 2007
- MISRA AC TL guidelines are available from MISRA web store see [www.misra.org.uk](http://www.misra.org.uk)

- dSPACE strategy: early and competent support of automotive standards
- **AUTOSAR might not yet be interesting for you, but at the time it will, TargetLink will offer a mature and proven-in-practice solution!**
- First production projects, e.g. at



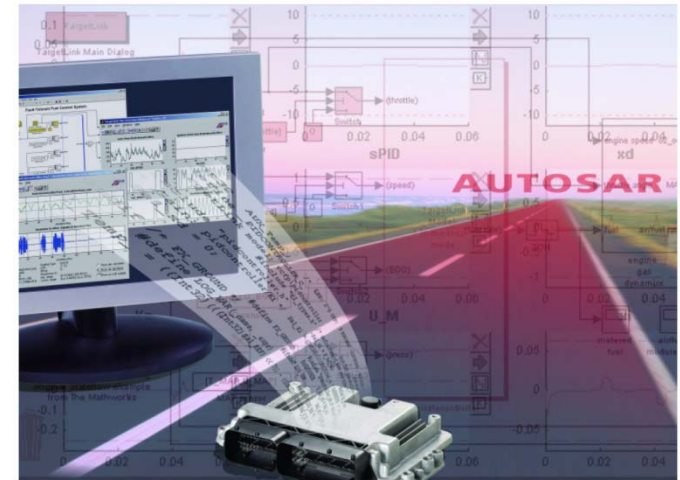
- Daimler AG, Body,  
refer to

AUTOSAR in the development process – procedure for introducing model-based AUTOSAR function development into production projects, Christian Dziobek, Dr. Florian Wohlgemuth, Dr. Thomas Ringler, Daimler AG, dSPACE Magazine, 01/2008

- AUDI AG, Chassis,  
refer to

Systematic AUTOSAR-Migration, Frank Gesele, Dr. Karsten Schmidt, AUDI AG, dSPACE NEWS 2008/1, Feb 2008

- Product level support since end of 2006
- AUTOSAR is the upcoming standard
- TargetLink support for AUTOSAR Software components includes
  - Modeling AUTOSAR SW Components directly on the block diagram level
  - Generate AUTOSAR-compliant code with corresponding RTE macros
  - Generate or import xml-File according to AUTOSAR SW Component Template
  - **SystemDesk and TargetLink** – System and function development – **dSPACE AUTOSAR tool chain**

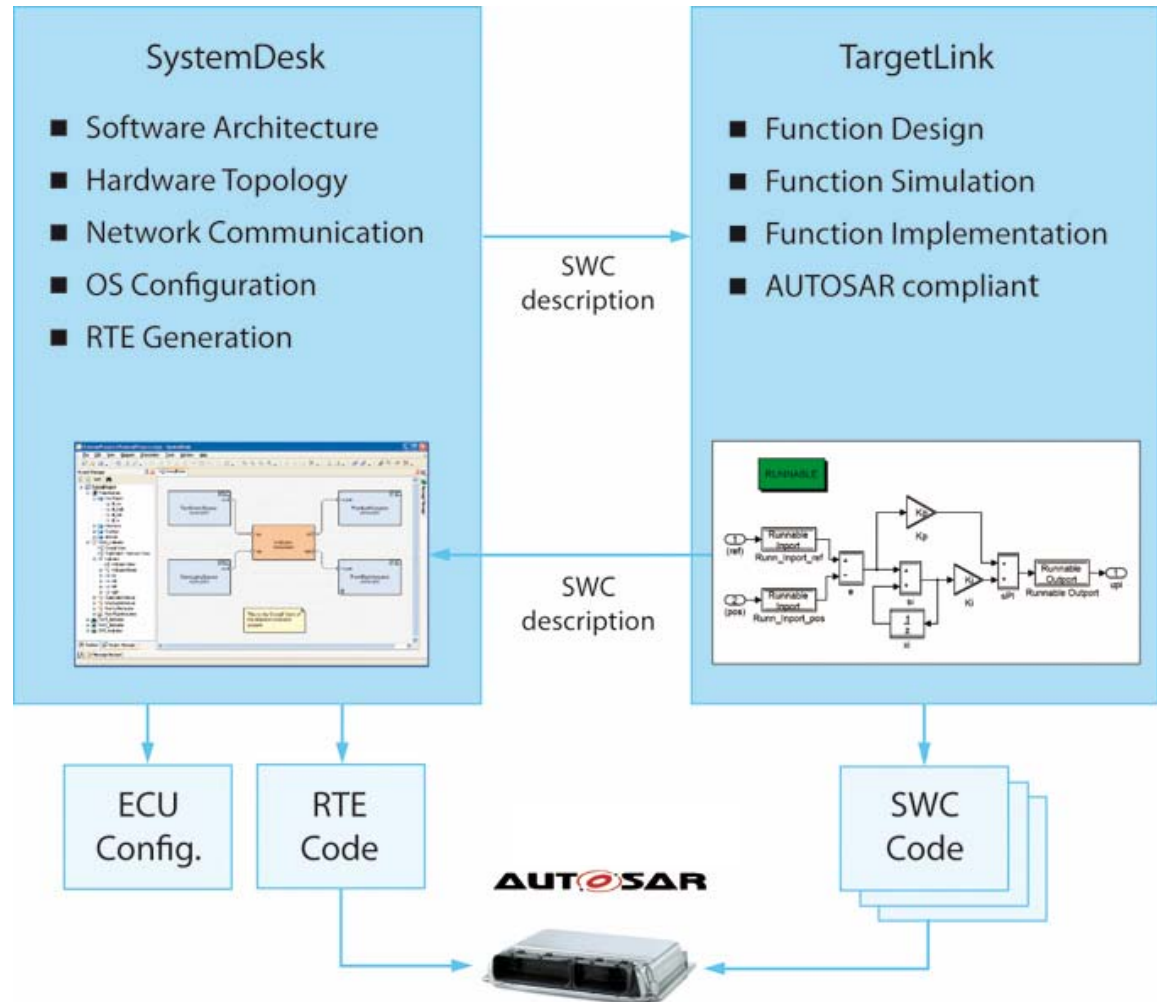


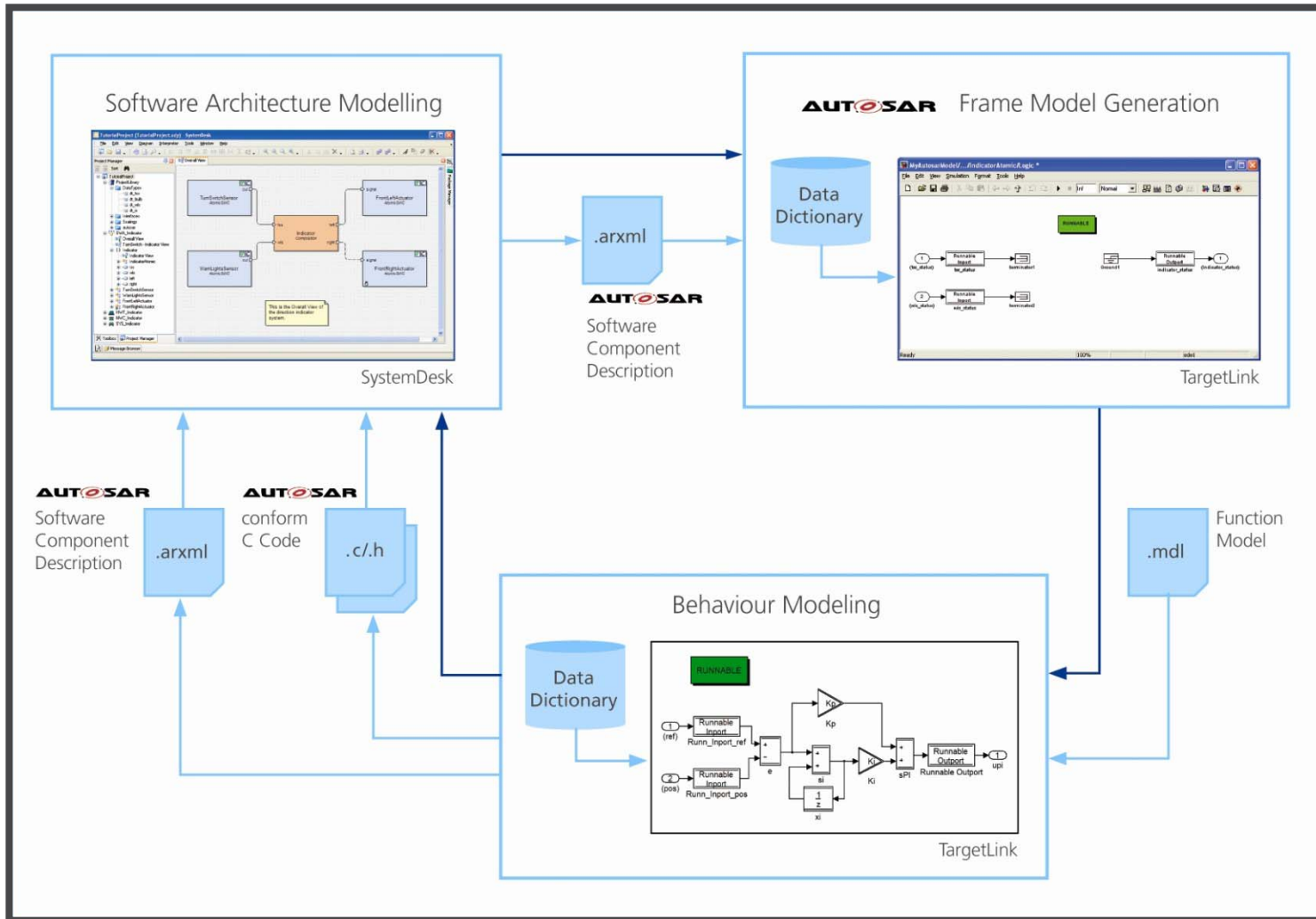
**AUTOSAR**

- Important: smooth integration of model-based design on system and function level for an efficient AUTOSAR-compliant development process

- **dSPACE solution**

- TargetLink for function development and implementation
- SystemDesk for Software Design and Architecture on system level





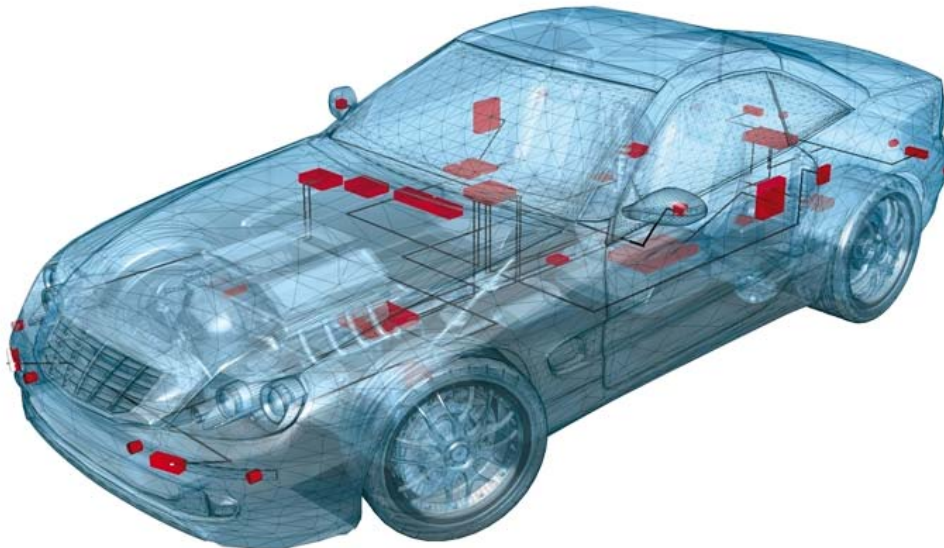
**Huge benefit: Smooth interaction between TargetLink and SystemDesk**



# TargetLink – Established and Successfully Deployed



- Worldwide
  - Germany & Europe
  - USA
  - Japan
  - Korea, India, China, ...



- Across domains
  - Powertrain
  - Chassis
  - Body
  - Driver Assistance
  - Active and passive safety

# TargetLink in Production Projects for example at ...

**dSPACE**



Thanks for listening!



© Copyright 2010, dSPACE GmbH

All rights reserved. Written permission is required for reproduction of all or parts of this publication. The source must be stated in any such reproduction.

This publication and the contents hereof are subject to change without notice. Brand names or product names are trademarks or registered trademarks of their respective companies or organizations.