Embedded Success dSPACE

The TargetLink® Ecosystem:

A Tool chain for Model-Based, Safety-Critical Software Development

Ulrich Eisemann

Senior Product Manager · Product Management

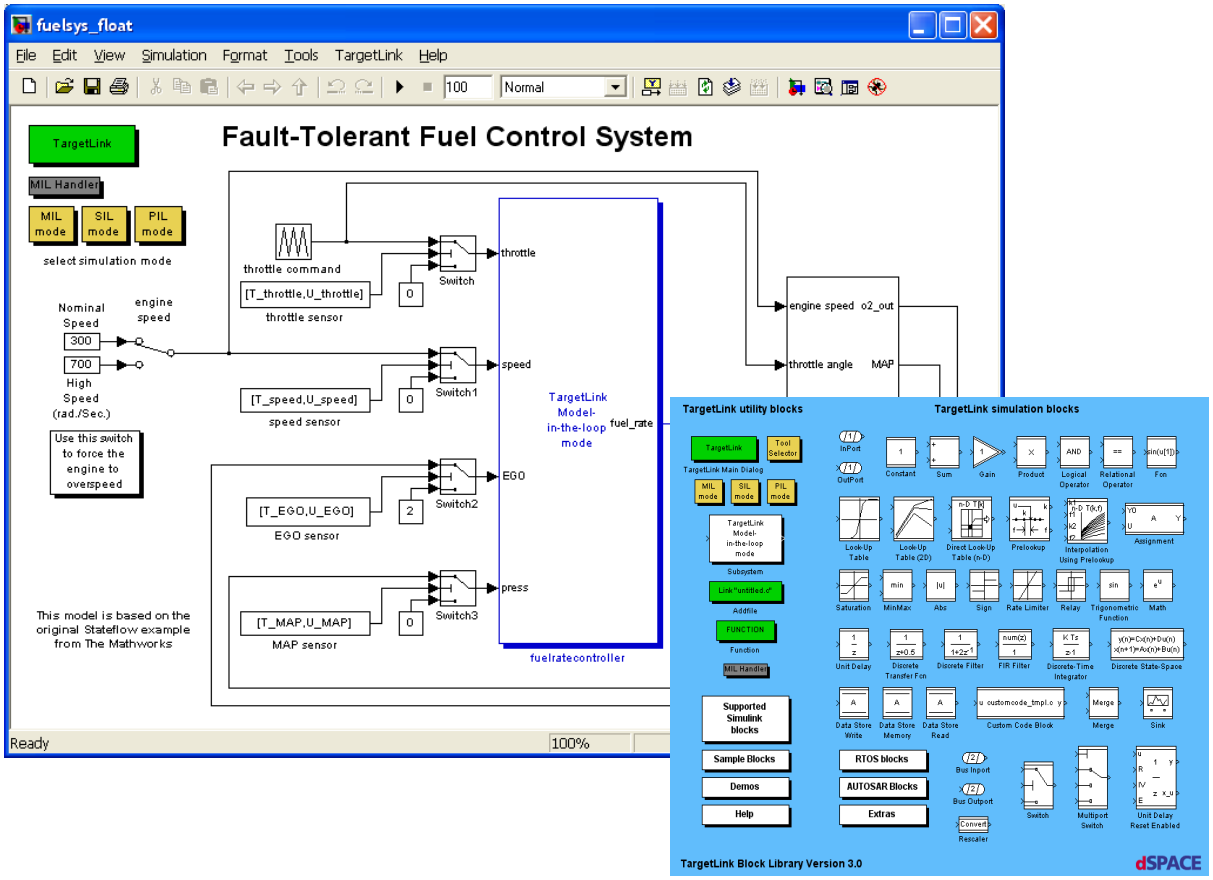
November 20, 2014 · Forum Mécatronique dSPACE

Agenda

- **TargetLink in a Nutshell**
- ISO 26262 and Model-Based Design
- The TargetLink ECO-System and ISO 26262
- Miscellaneous

TargetLink

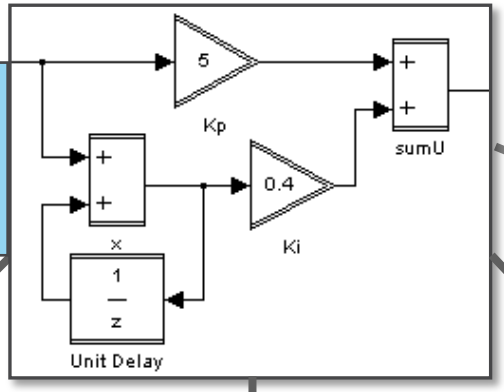
- Production-quality code generation straight from Simulink/Stateflow
- Fully integrated into MATLAB/Simulink
- Dedicated block library for controller development



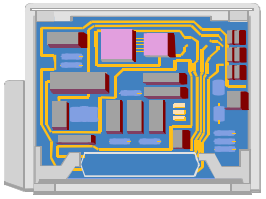
TargetLink Functionality in a Nutshell

ECU

Simulink/TargetLink model



Configure Code Generation Properties



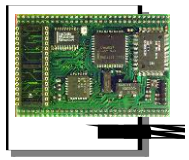
Model-in-the-Loop Simulation on the Host PC

- Checking the algorithm
- Validation using RCP
- Logging references

Software-in-the-Loop Simulation on the Host PC

- Checking fixed-point behavior
- Checking code behavior

Processor-in-the-Loop Simulation on evaluation board



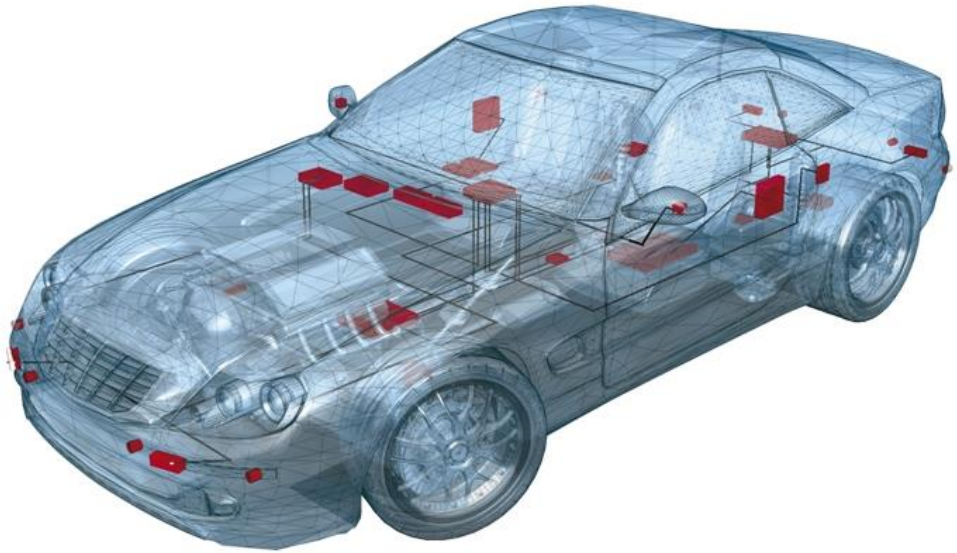
- Code validation
- Profiling RAM / ROM etc.
- Final verification

TargetLink Code is …

Proven in Practice	▪ In production for more than 14 years … ▪ … and in millions of vehicles
Highly optimized	▪ For floating-point and fixed-point applications ▪ Best-in-class fixed point support → Small software foot print saves ECU resources
Flexible and configurable	▪ For function interfaces, access functions, AUTOSAR, etc. → Easy integration with other software
Readable and Traceable	→ Well suitable for safety-relevant applications

TargetLink – Established and Successfully Deployed

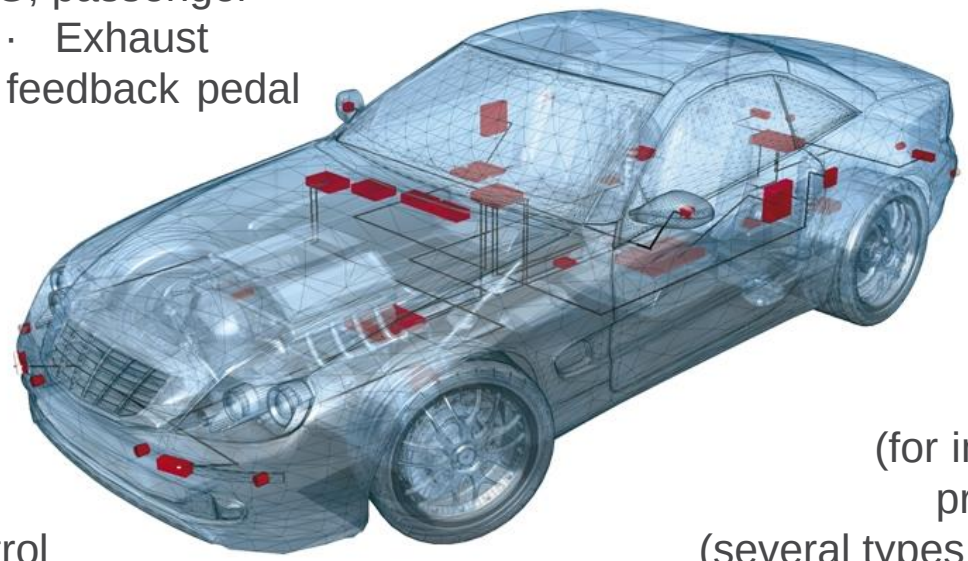
- **Worldwide**
  - Germany & Europe
  - USA
  - Japan
  - Korea, India, China, …



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

TargetLink – Successfully used for …

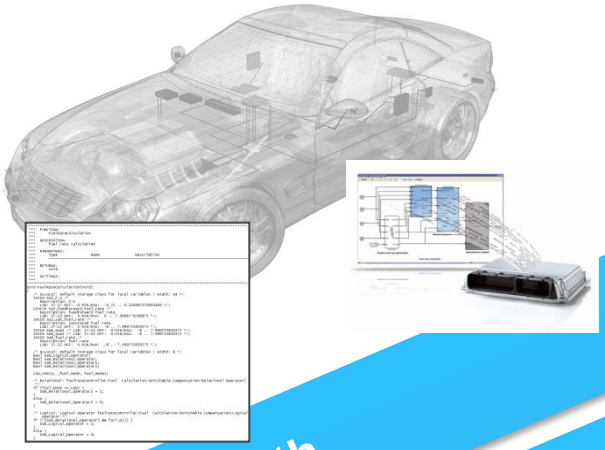
Active suspension management · Airbag firing control · Air suspension · Audio playback · Back door control · Battery management · Brake support · Central body controller · Continuous variable transmission · Collision warning · Damper control · Dynamic bend lighting · Electric Assisted Power Steering · Engine control (full and partial, gasoline, diesel, hybrid, CNG, passenger cars and trucks) · ESP · Exhaust gas reduction · Force feedback pedal control · Hydrogen fuel control · Integrated chassis controls · Lane Departure Warning · Occupant sensing · Park assistance · Reversible belt pretensioner · Rollover detection · Roof module (for interior lightning incl. Gateway) · Tire pressure monitoring · Transmission control (several types, for passenger cars and trucks) · Transfer case controls · Starter generator for hybrids · Steering control · Vertical dynamic control · … and more



Agenda

- TargetLink in a Nutshell
- **ISO 26262 and Model-Based Design**
- The TargetLink ECO-System and ISO 26262
- Miscellaneous

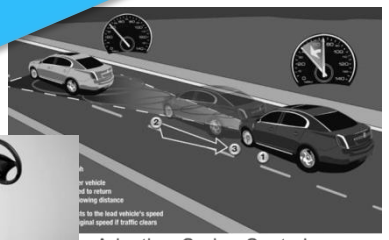
Motivation



Software Growth

Model-Based Software Development

Safety-Related Vehicle Functions



Adaptive Cruise Control

Active Front Steering

... and many more

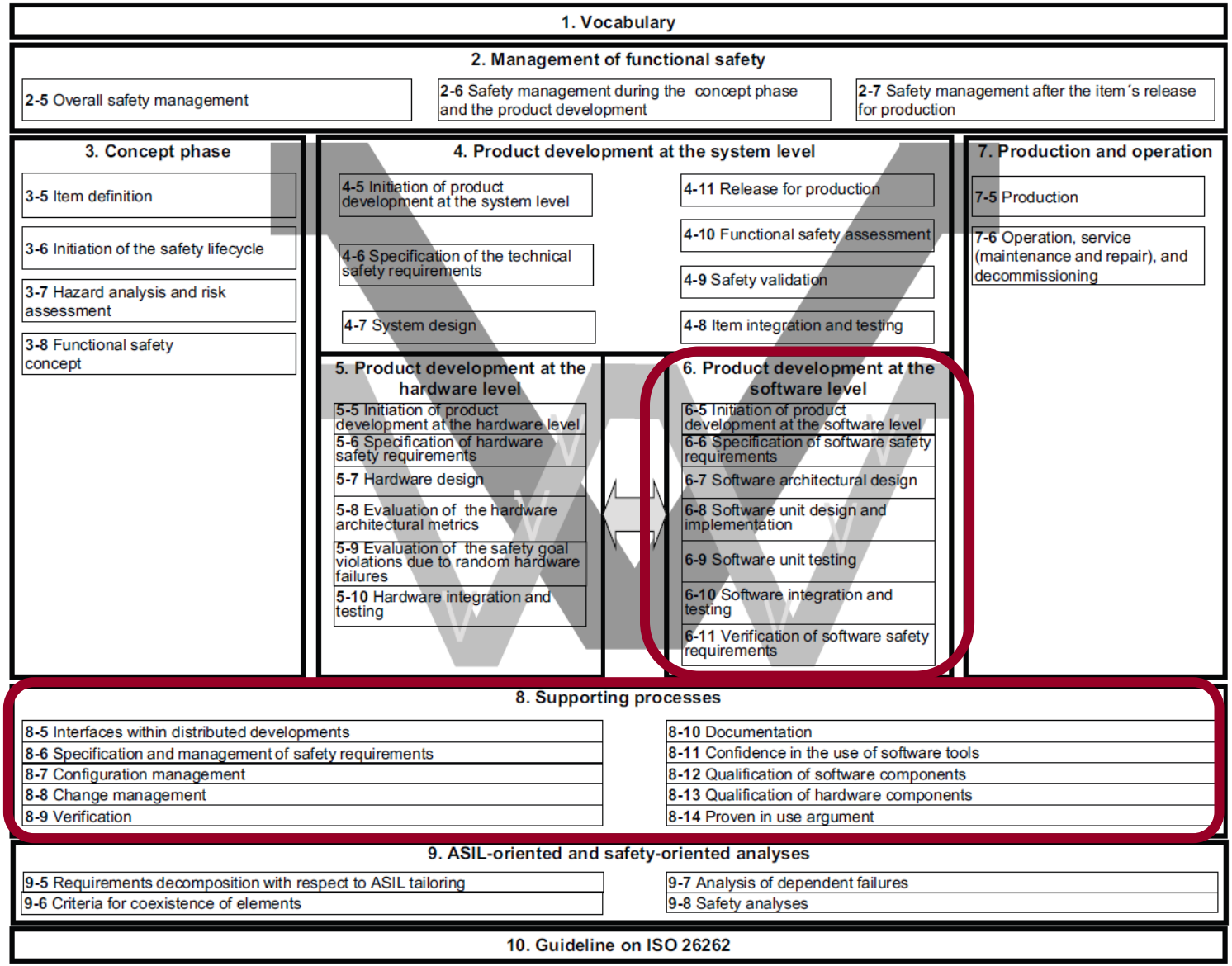
November 2011

ISO International Organization for Standardization

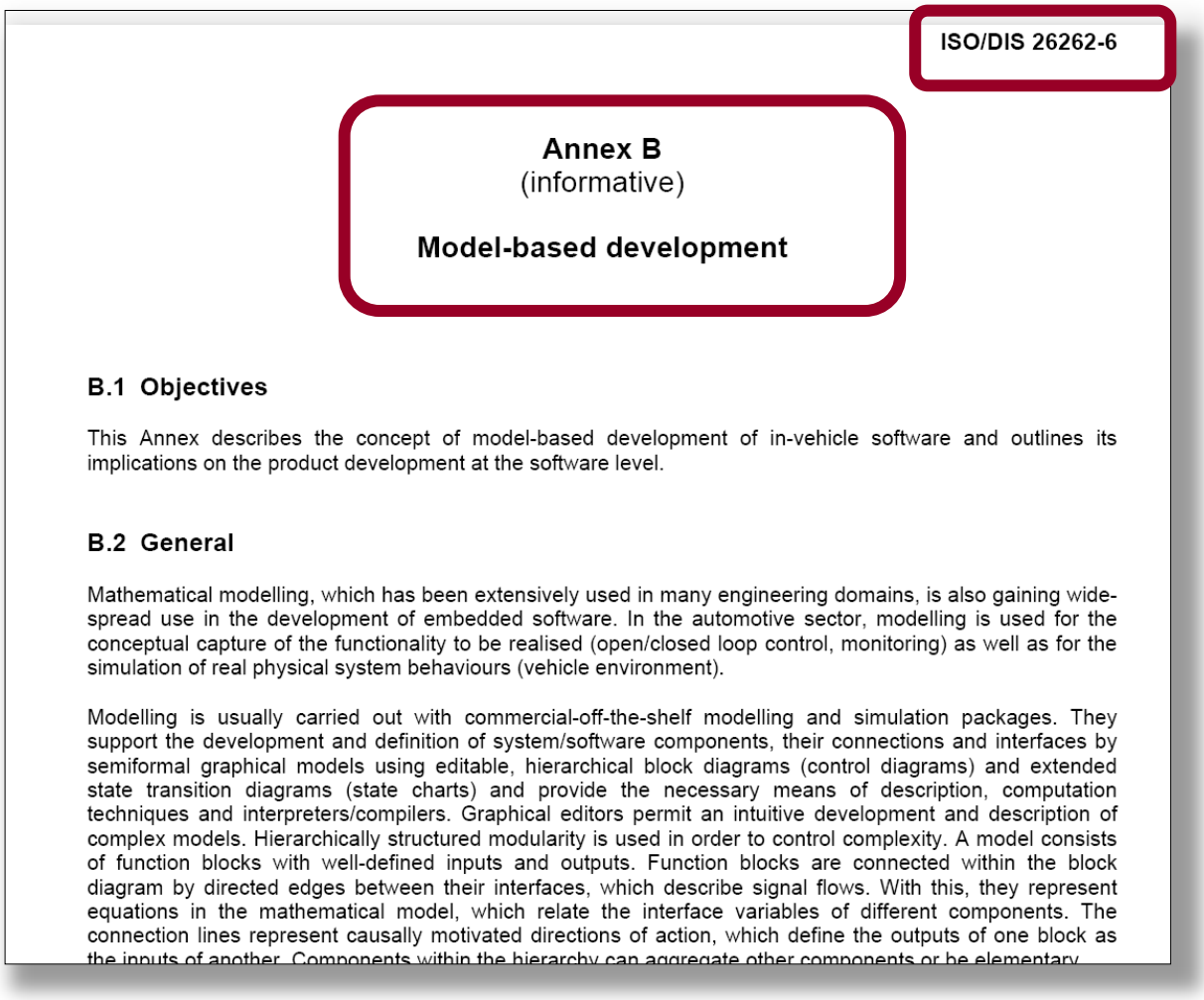
ISO 26262

Model-Based Software Development of Safety-Related Software According to ISO 26262.

ISO 26262 – Overview



ISO 26262 and Model-based Development



ISO/DIS 26262-6

Annex B
(informative)

Model-based development

B.1 Objectives

This Annex describes the concept of model-based development of in-vehicle software and outlines its implications on the product development at the software level.

B.2 General

Mathematical modelling, which has been extensively used in many engineering domains, is also gaining wide-spread use in the development of embedded software. In the automotive sector, modelling is used for the conceptual capture of the functionality to be realised (open/closed loop control, monitoring) as well as for the simulation of real physical system behaviours (vehicle environment).

Modelling is usually carried out with commercial-off-the-shelf modelling and simulation packages. They support the development and definition of system/software components, their connections and interfaces by semiformal graphical models using editable, hierarchical block diagrams (control diagrams) and extended state transition diagrams (state charts) and provide the necessary means of description, computation techniques and interpreters/compilers. Graphical editors permit an intuitive development and description of complex models. Hierarchically structured modularity is used in order to control complexity. A model consists of function blocks with well-defined inputs and outputs. Function blocks are connected within the block diagram by directed edges between their interfaces, which describe signal flows. With this, they represent equations in the mathematical model, which relate the interface variables of different components. The connection lines represent causally motivated directions of action, which define the outputs of one block as the inputs of another. Components within the hierarchy can aggregate other components or be elementary

ISO International Organization for Standardization

ISO 26262

dSPACE

ISO 26262-6 specifically addresses model-based development!

ISO/DIS 26262-6

Annex B
(informative)

Model-based development

One characteristic of the model-based development paradigm is the fact that the functional model not only specifies the desired function but also provides design information and finally even serves as the basis of the implementation by means of code generation.

…

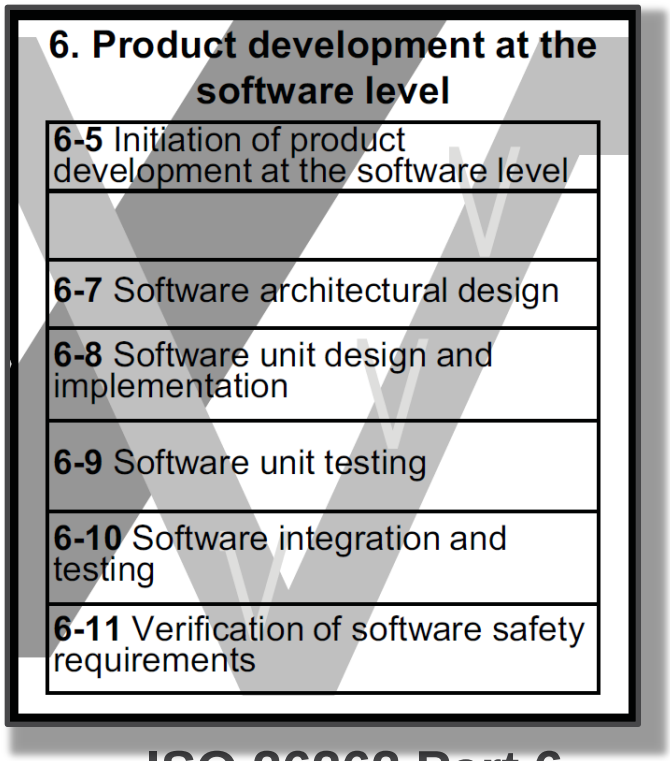
In contrast to code-based software development with a clear separation of phases in model-based development a stronger coalescence of the phases Software Safety requirements, Software Architectural Design, and Software unit design and implementation can be noted. Moreover, one and the same graphical modelling notation is used during the consecutive development stages. Testing activities are also treated differently since models can be used as a useful source of information for the testing process (model-based testing). The seamless utilisation of models facilitates a highly consistent and efficient development.

ISO International Organization for Standardization

ISO 26262

Software Safety Standards for Automotive and Aerospace

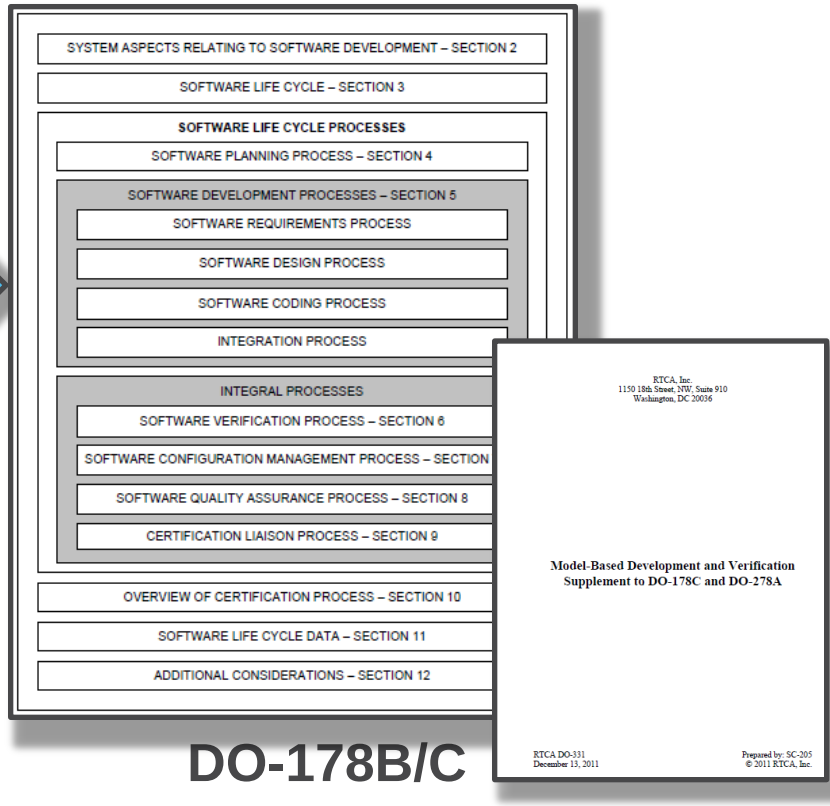
Automotive



ISO 26262 Part 6

Model-Based development is part of ISO 26262 Part 6

Aerospace



DO-178B/C

Supplement DO-331 is specifically about Model-Based Development for Aerospace

Agenda

- TargetLink in a Nutshell
- ISO 26262 and Model-Based Design
- **The TargetLink ECO-System and ISO 26262**
- Miscellaneous

TargetLink Certification

ZERTIFIKAT ◆ CERTIFICATE ◆ 認証書 ◆ СЕРТИФИКАТ ◆ CERTIFICADO ◆ CERTIFICAT

Product Service

CERTIFICATE

No. Z10 12 04 71483 002

Holder of Certificate: **dSPACE GmbH**
Rathenaustraße 26
33102 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 26262.

The report no. DP76881aC is a mandatory part of this certificate.

Tested according to: IEC 61508-3:2010
ISO 26262:2011

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.: DP76881aC

(Andreas Bärwald)

Date, 2012-05-07

Page 1 of 1

TÜV SÜD Product Service GmbH · Zertifizierstelle · Ridlerstraße 65 · 80339 München · Germany

TÜV®

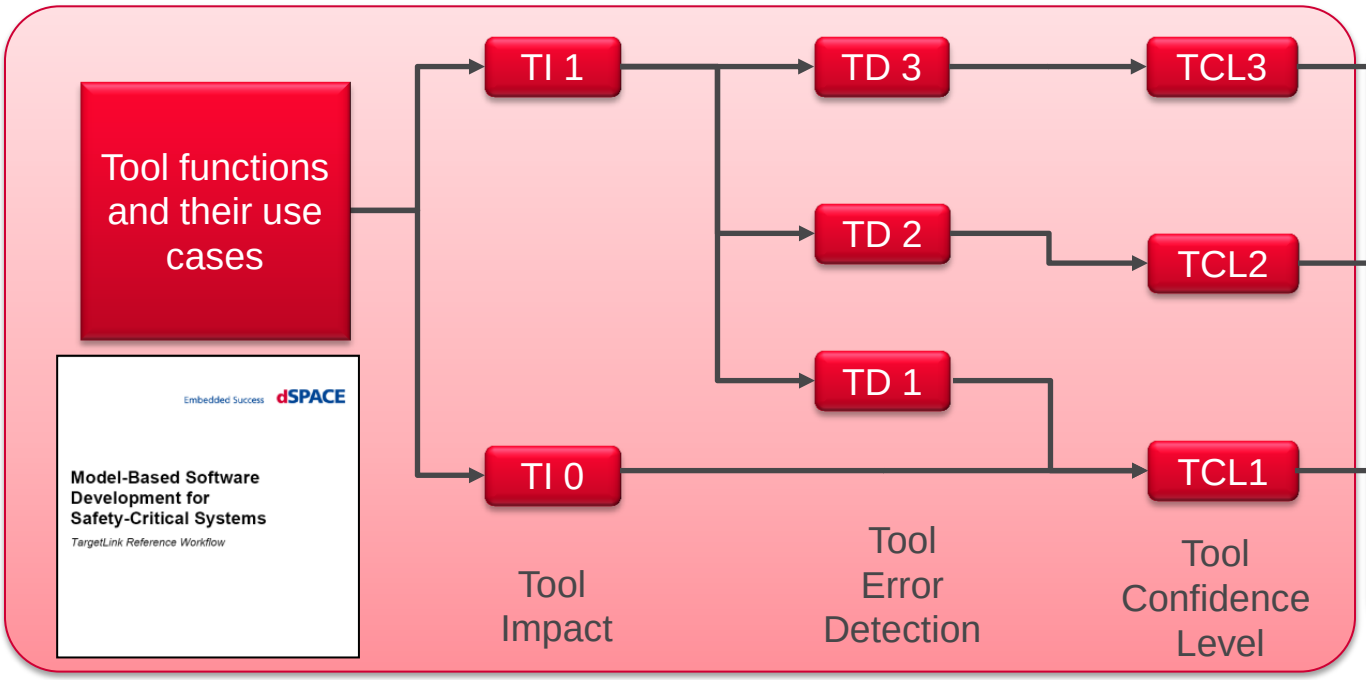
A1 / 04.11

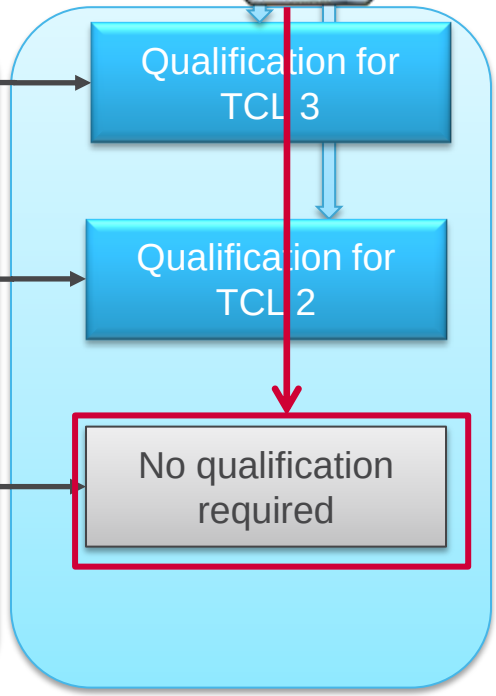
- TÜV SÜD Automotive GmbH, German certification authority, evaluated TargetLink
- Result:
  TargetLink is fit for purpose to develop safety-related software according to ISO 26262 (up to ASIL D), IEC 61508 (up to SIL 3), and derivative standards
- Certificate(s) granted for
  - TL 2.3.1 (2009)
  - TL 3.1 (2010)
  - TL 3.2 (2011)
  - TL 3.3 (2012)
  - TL 3.4 (2013)
  - TL 3.5 (2014)

Provides required confidence in using TargetLink.

ISO 26262 – Tool Classification and Qualification

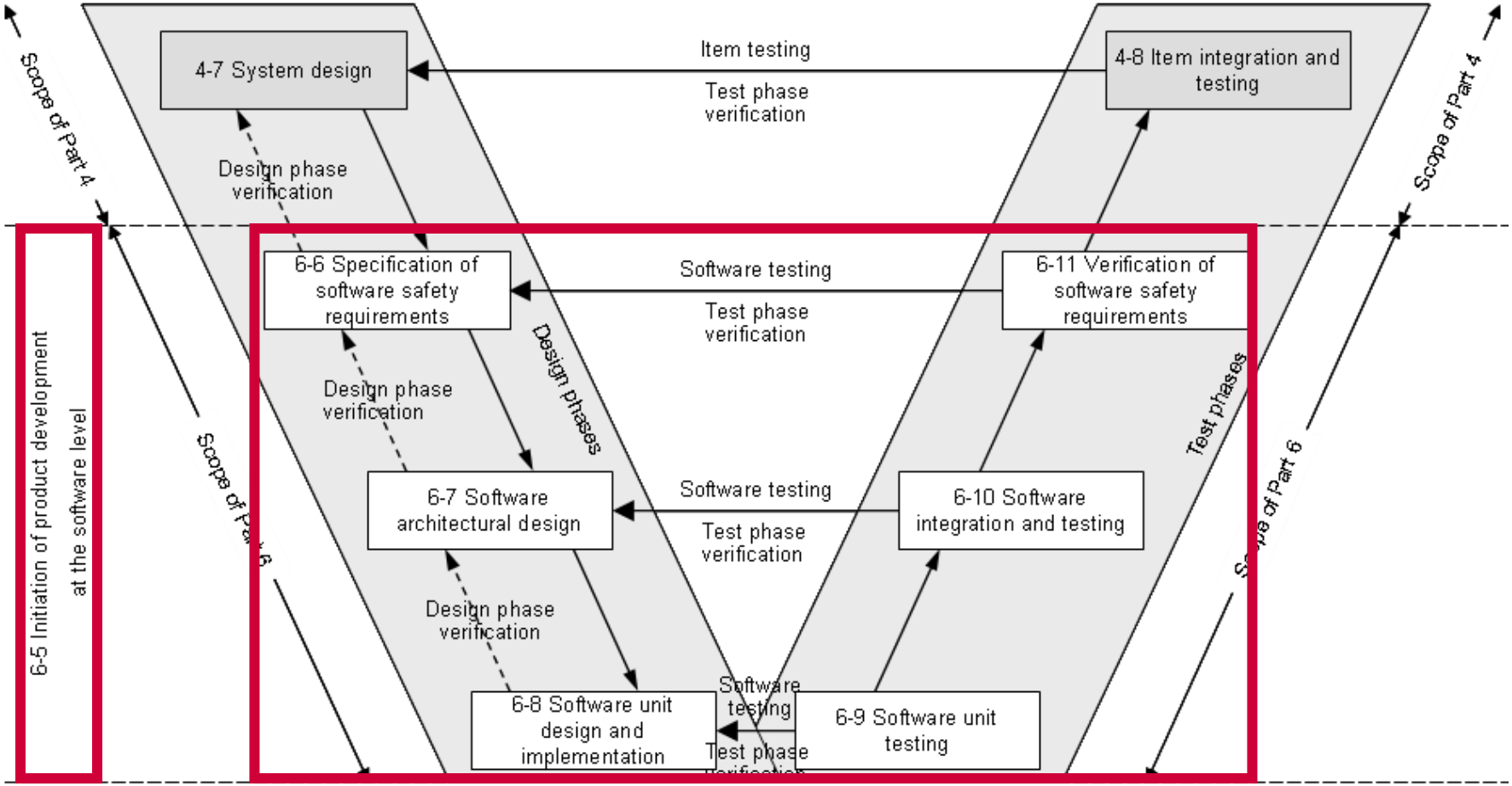
- TargetLink
  - TCL based on a TargetLink Reference Workflow
  - “Fit-for-Purpose” Certification anyway





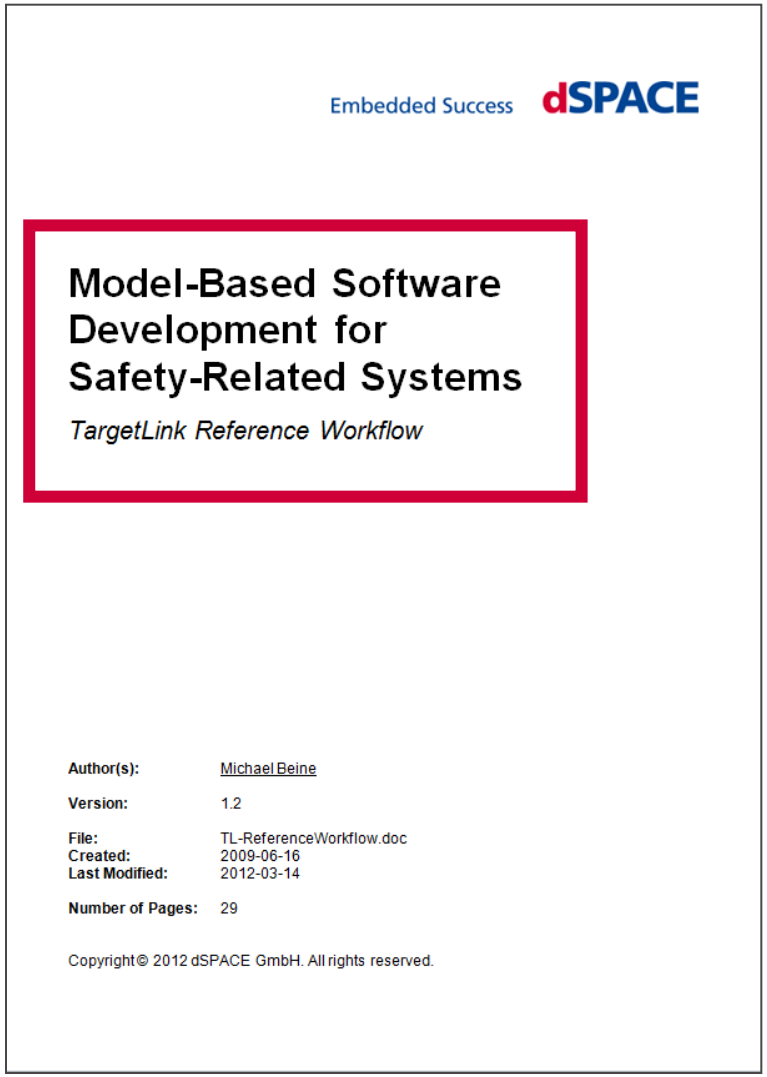
ISO 26262 requires tool classification and – depending on the classification - qualification

ISO 26262 – 6: Product Development at Software Level



Reference phase model for the software development

Reference Workflow for Safety-Related Systems


Embedded Success dSPACE

Model-Based Software Development for Safety-Related Systems

TargetLink Reference Workflow

Author(s): Michael Beine

Version: 1.2

File: TL-ReferenceWorkflow.doc
Created: 2009-06-16
Last Modified: 2012-03-14

Number of Pages: 29

- Concrete advice on how to develop according to ISO 26262
- **Covers the whole Model-Based Development Tool Chain, not just TargetLink**

Facilitates setting up ISO 26262 development process

- Based on industry best practices
- Approved by TÜV Süd

Confidence in development process

Note: The Reference Workflow conforms to ISO 26262, IEC 61508 and derivative standards.

The TargetLink Ecosystem in ISO 26262 Projects

dSPACE

Requirements Tracing

MathWorks Simulink V & V

dSPACE TargetLink

BTC EmbeddedTester®

Requirements Coverage Analysis

BTC EmbeddedTester®

Guideline Checking

MES MXAM

WCET & Stack Analysis

AbsInt aiT

AbsInt StackAnalyzer

Model Analysis & Review

MES M-XRAY

Run-Time Error Analysis

AbsInt Astrée

TargetLink

Model Coverage Analysis

BTC EmbeddedTester®

Code Coverage Analysis

BTC EmbeddedTester®

Test Vector Generation

BTC EmbeddedTester®

Formal Verification

BTC EmbeddedSpecifier®

BTC EmbeddedValidator®

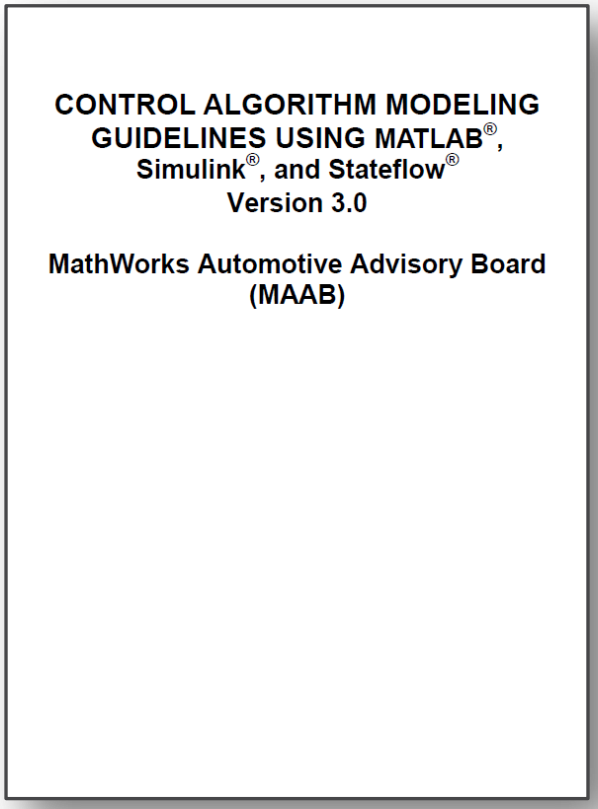
Model-Based Testing

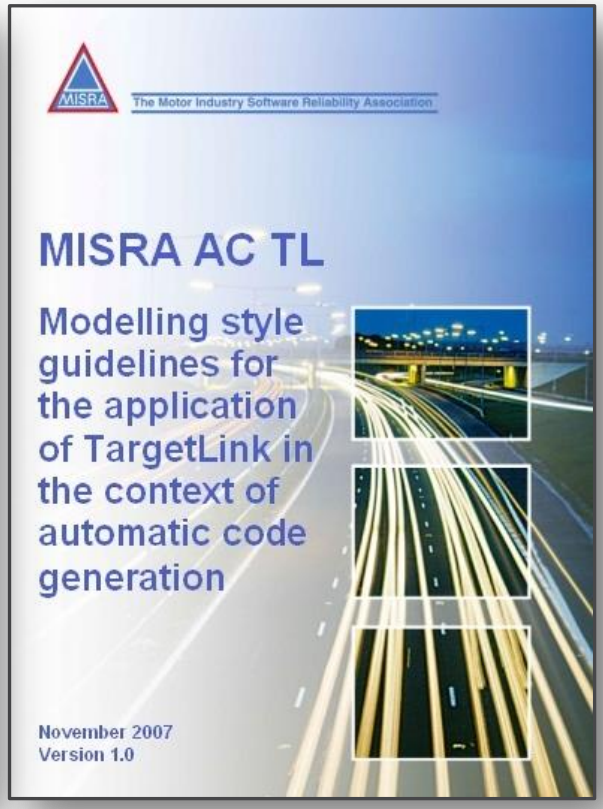
BTC EmbeddedTester®

Modeling Guidelines for TargetLink

Standard guidelines are available for TargetLink

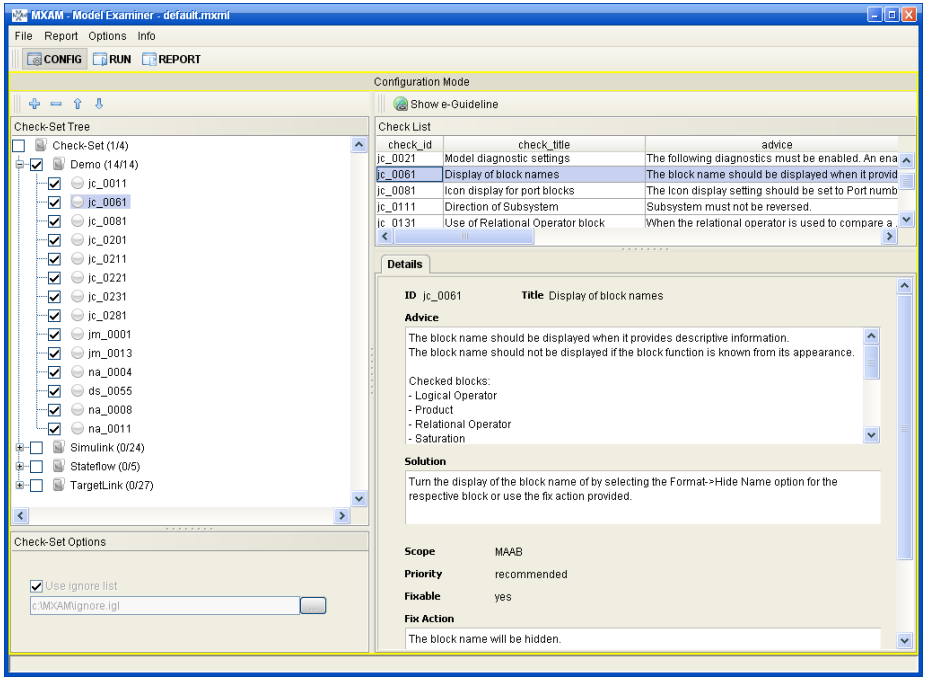
- Ensuring good design quality of models
- Can be used as the basis for project/company-specific guidelines





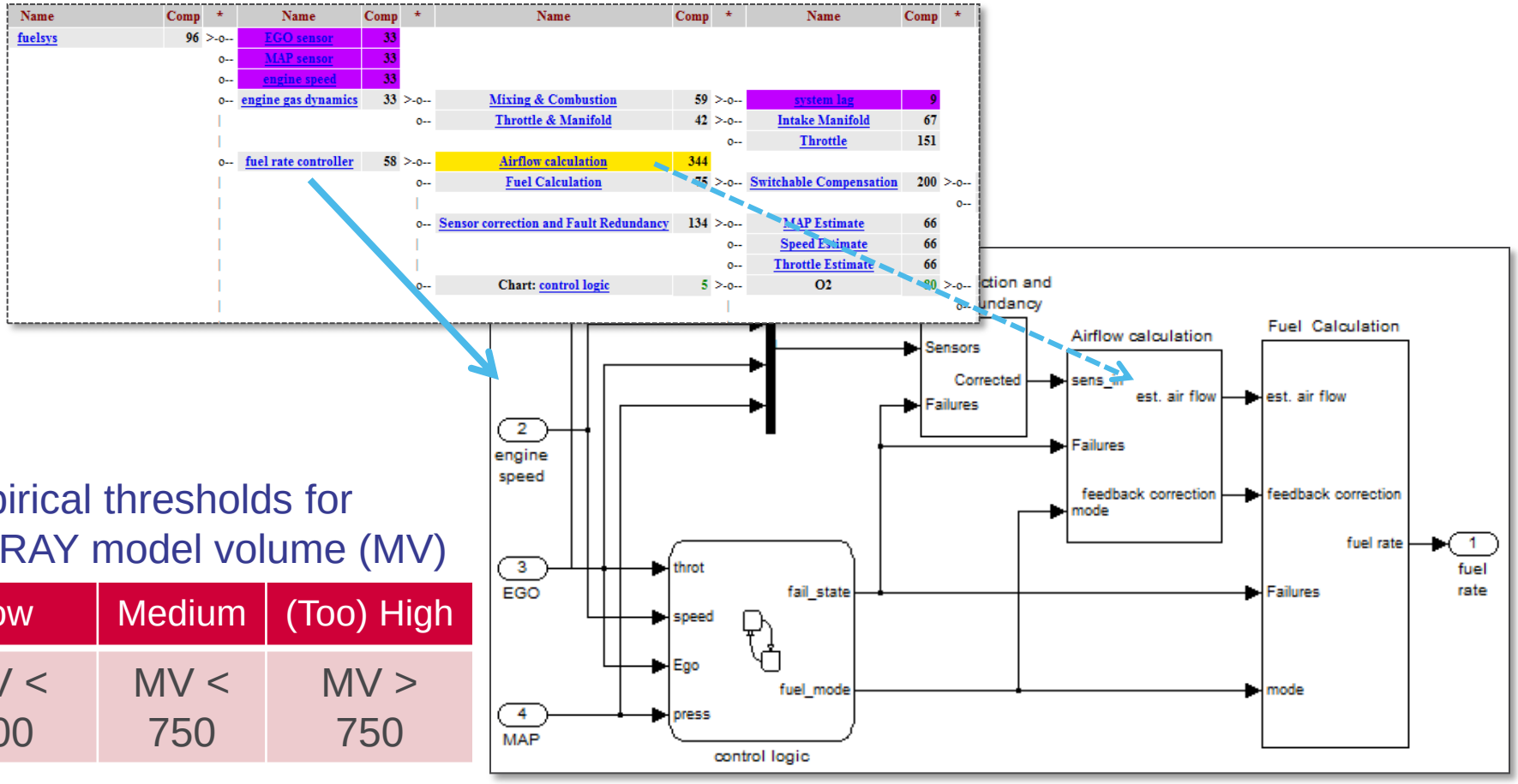
Model Examiner (MXAM) for Automated Rule Checking

- Cooperation between MES and dSPACE
- MES provides Model Examiner (MXAM) for automated rule checking
  - Automated checking of MAAB, TargetLink and MISRA AC TL style guidelines
  - Automated repair of guideline violations
  - Clear reporting of all detected and corrected guideline violations
  - Easy integration into project-specific development environments
  - Fully-functional API for developing and integrating your own model checks



M-XRAY: Model Architecture and Complexity Analysis

- Model architecture analysis to identify complex model parts easily



Empirical thresholds for M-XRAY model volume (MV)

Low	Medium	(Too) High
MV < 300	MV < 750	MV > 750

The TargetLink Ecosystem for ISO 26262 Projects

dSPACE

Requirements Tracing
- MathWorks Simulink V & V
- dSPACE TargetLink
- BTC EmbeddedTester®

Requirements Coverage Analysis
- BTC EmbeddedTester®

Guideline Checking
- MES MXAM

WCET & Stack Analysis
- AbsInt aiT
- AbsInt StackAnalyzer

Model Analysis & Review
- MES M-XRAY

TargetLink

Run-Time Error Analysis
- AbsInt Astrée

Model Coverage Analysis
- BTC EmbeddedTester®

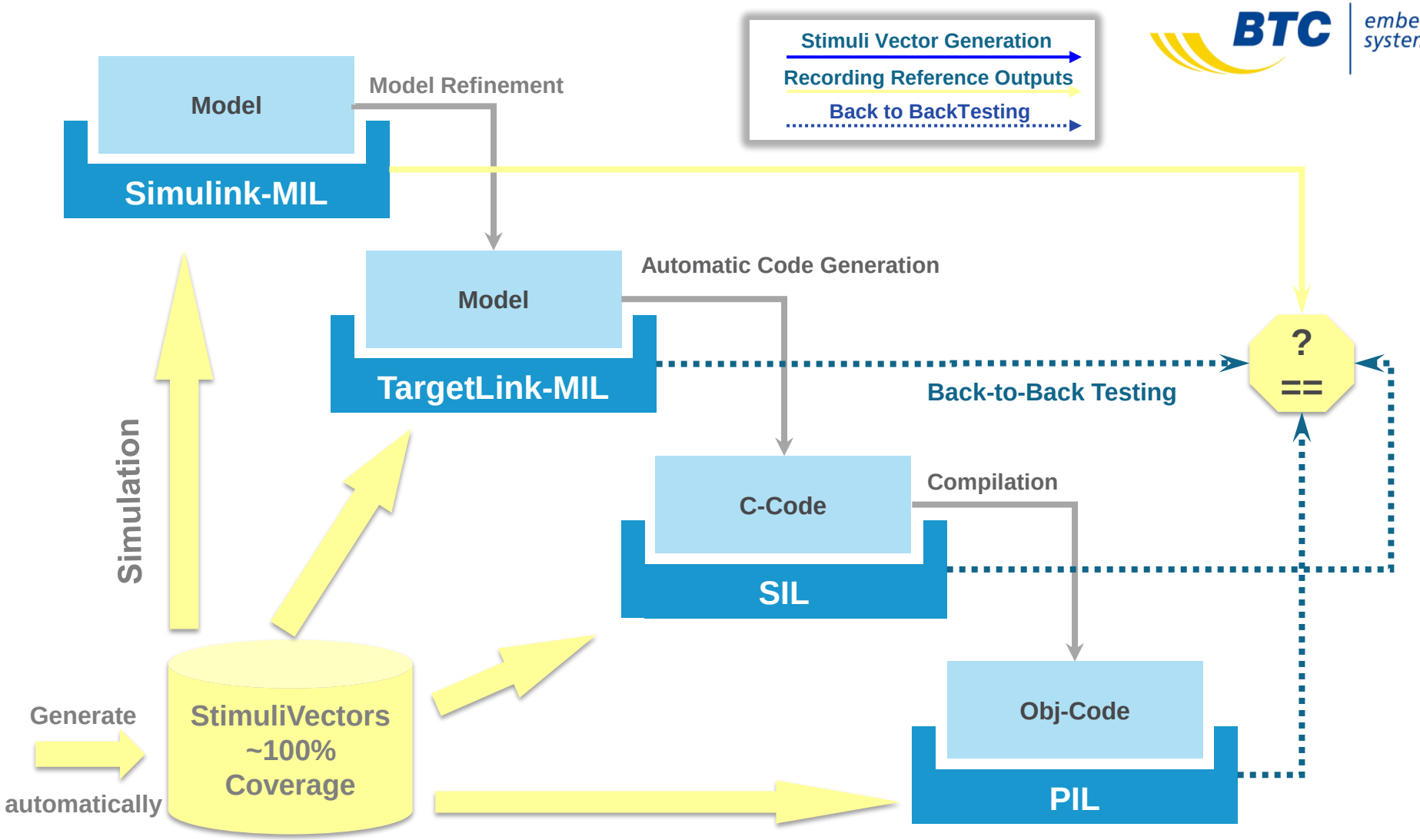
Code Coverage Analysis
- BTC EmbeddedTester®

Test Vector Generation
- BTC EmbeddedTester®

Formal Verification
- BTC EmbeddedSpecifier®
- BTC EmbeddedValidator®

Model-Based Testing
- BTC EmbeddedTester®

EmbeddedTester – Back-to-Back Testing



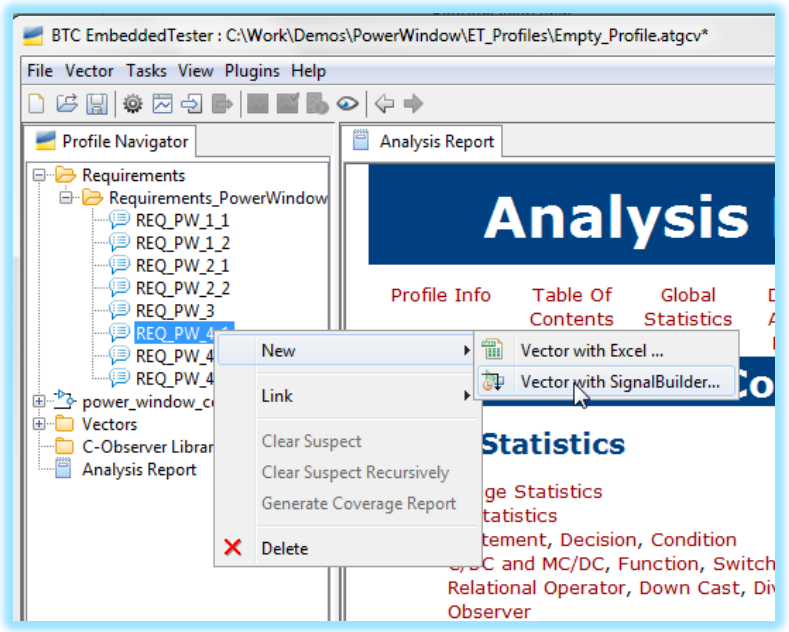
EmbeddedTester – Requirement-based Testing

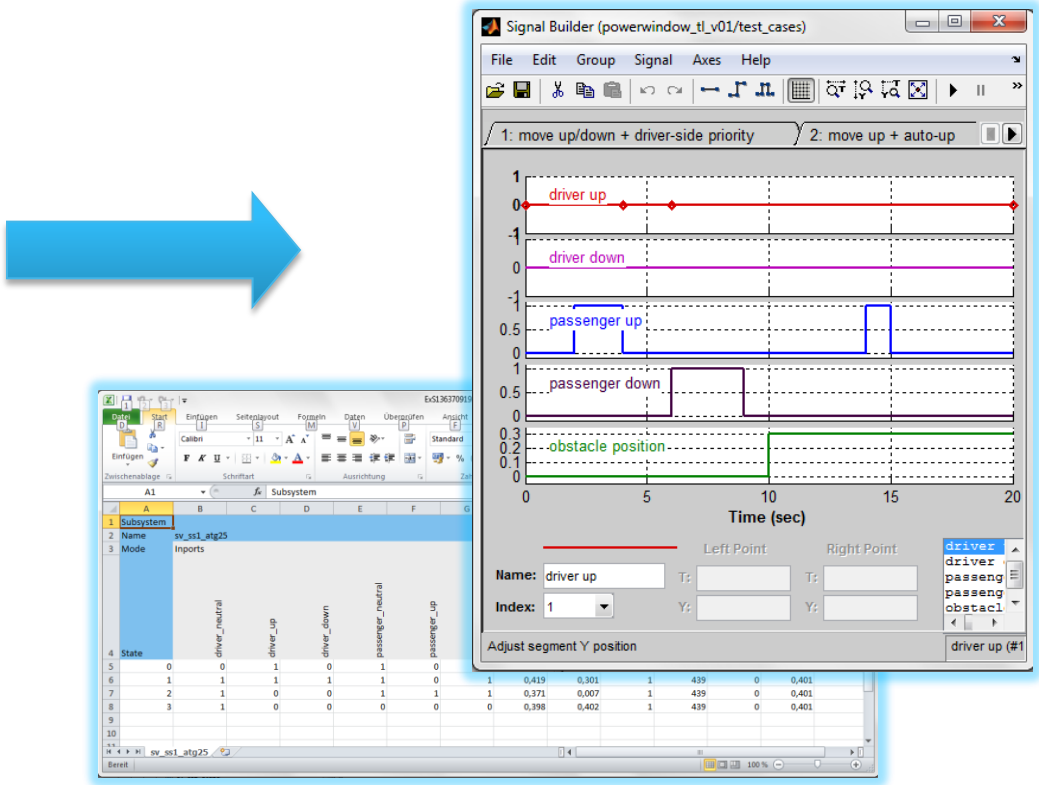
dSPACE

Create Test Cases — Execute Test Cases — Calculate Coverage

BTC embedded systems

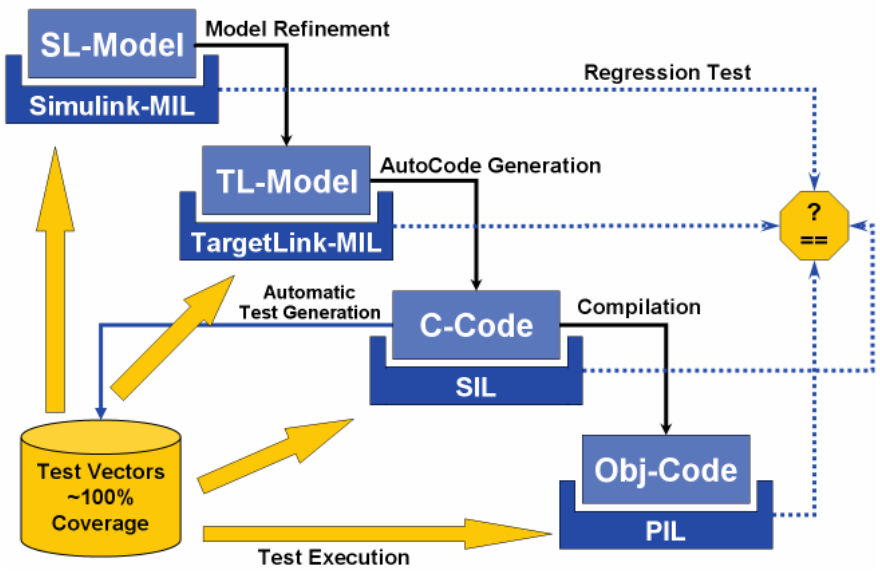
Create or import/export functional tests using existing industry formats (Signal Builder, Excel, TLDS, CTE..) or user defined ones (XML, CSV, MAT…)





Back-to-back Testing – Tool Support

- **BTC Embedded*Tester***
  - Automatic Test Execution & Regression
  - Automatic Test Evaluation
  - Automatic Regression Reporting
  - **Code Coverage Measurement**
  - **Automatic Testvector Generation**
  - Certified for IEC 61508 and ISO 262622



The TargetLink Ecosystem in ISO 26262 Projects

dSPACE

Requirements Tracing

MathWorks Simulink V & V

dSPACE TargetLink

BTC EmbeddedTester®

Requirements Coverage Analysis

BTC EmbeddedTester®

Guideline Checking

MES MXAM

WCET & Stack Analysis

AbsInt aiT

AbsInt StackAnalyzer

Run-Time Error Analysis

AbsInt Astrée

Model Analysis & Review

MES M-XRAY

TargetLink

Model Coverage Analysis

BTC EmbeddedTester®

Code Coverage Analysis

BTC EmbeddedTester®

Test Vector Generation

BTC EmbeddedTester®

Formal Verification

BTC EmbeddedSpecifier®

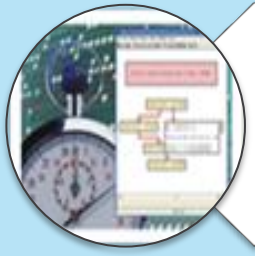
BTC EmbeddedValidator®

Model-Based Testing

BTC EmbeddedTester®

TargetLink Tool Coupling with AbsInt Static Analyzers

ISO 26262 requires to take non-functional safety properties into account, like worst case execution time, stack consumption etc.



- Proving the **correct timing behavior**
- Safe upper bounds on the worst-case execution time of **tasks**
- Working on **binary code**

aiT WCET Analyzer

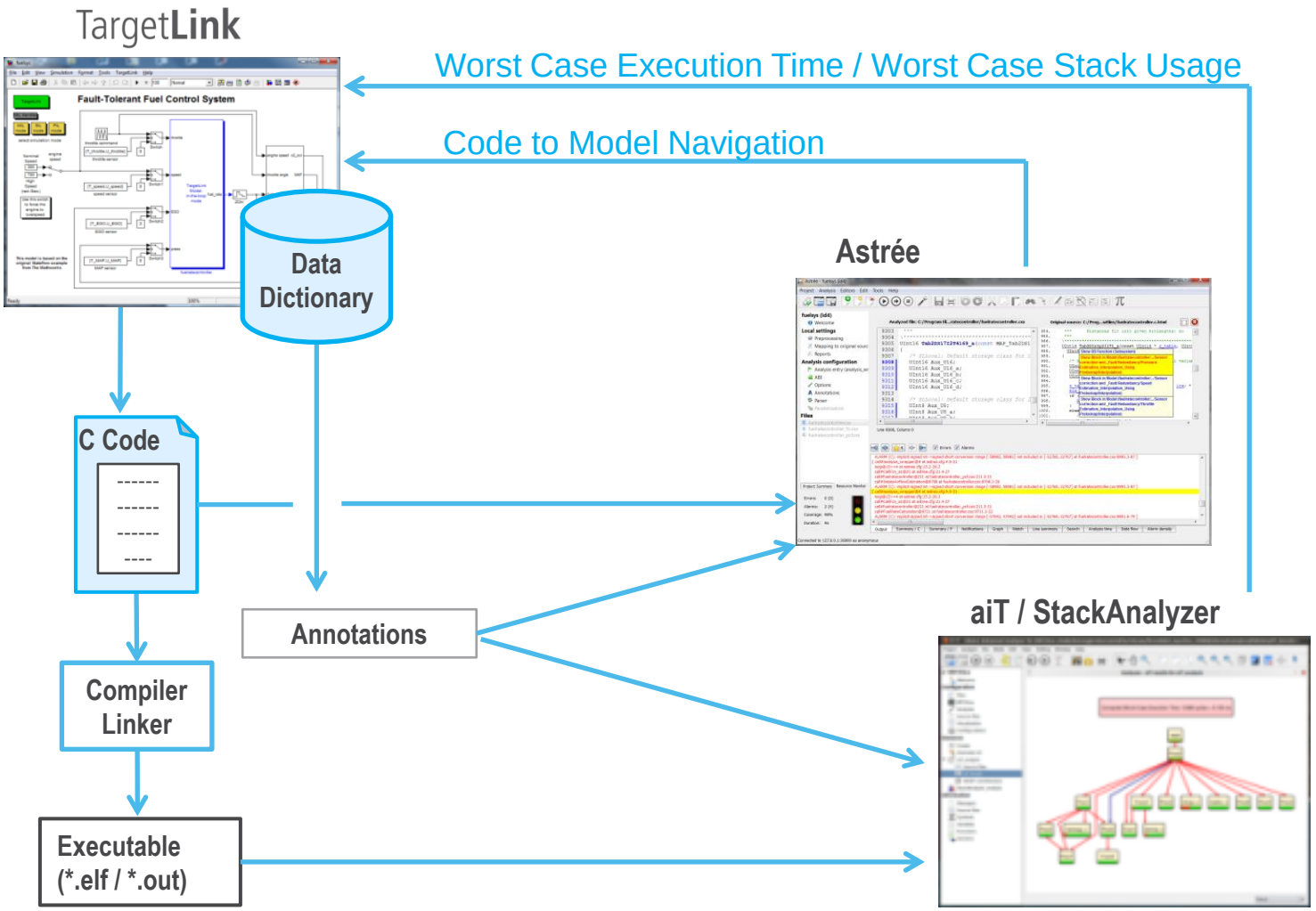
- Excluding **stack overflows**
- Safe upper bounds on maximal stack usage of **tasks**
- Working on **binary code**

StackAnalyzer

- Proving the **absence of runtime errors** (division by zero, arithmetic overflow, invalid pointer accesses, etc.) in **C programs**
- Working on **source code**

Astrée

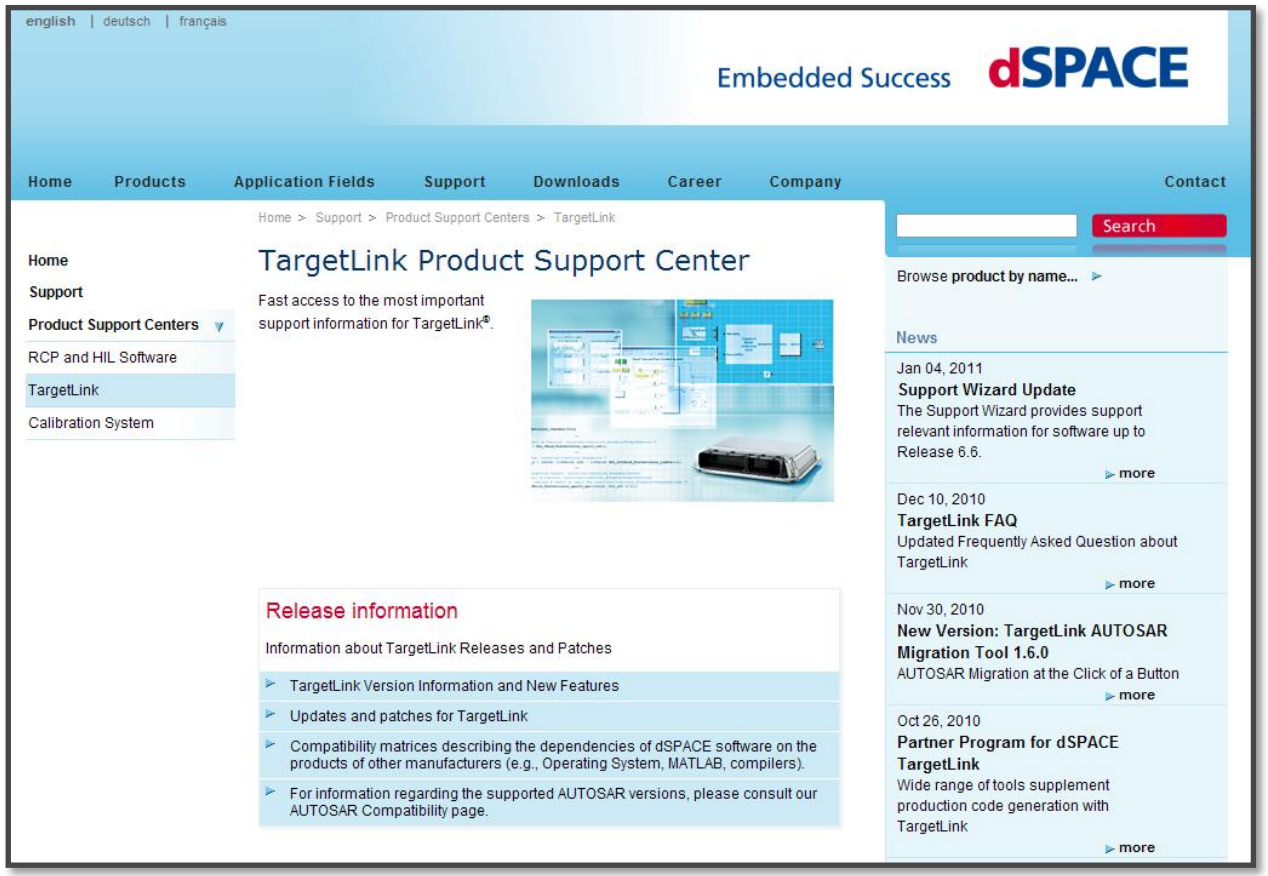
TargetLink Tool Coupling with AbsInt Static Analyzers



Agenda

- TargetLink in a Nutshell
- ISO 26262 and Model-Based Design
- The TargetLink ECO-System and ISO 26262
- **Miscellaneous**

TargetLink Product Support Center – Online Resources

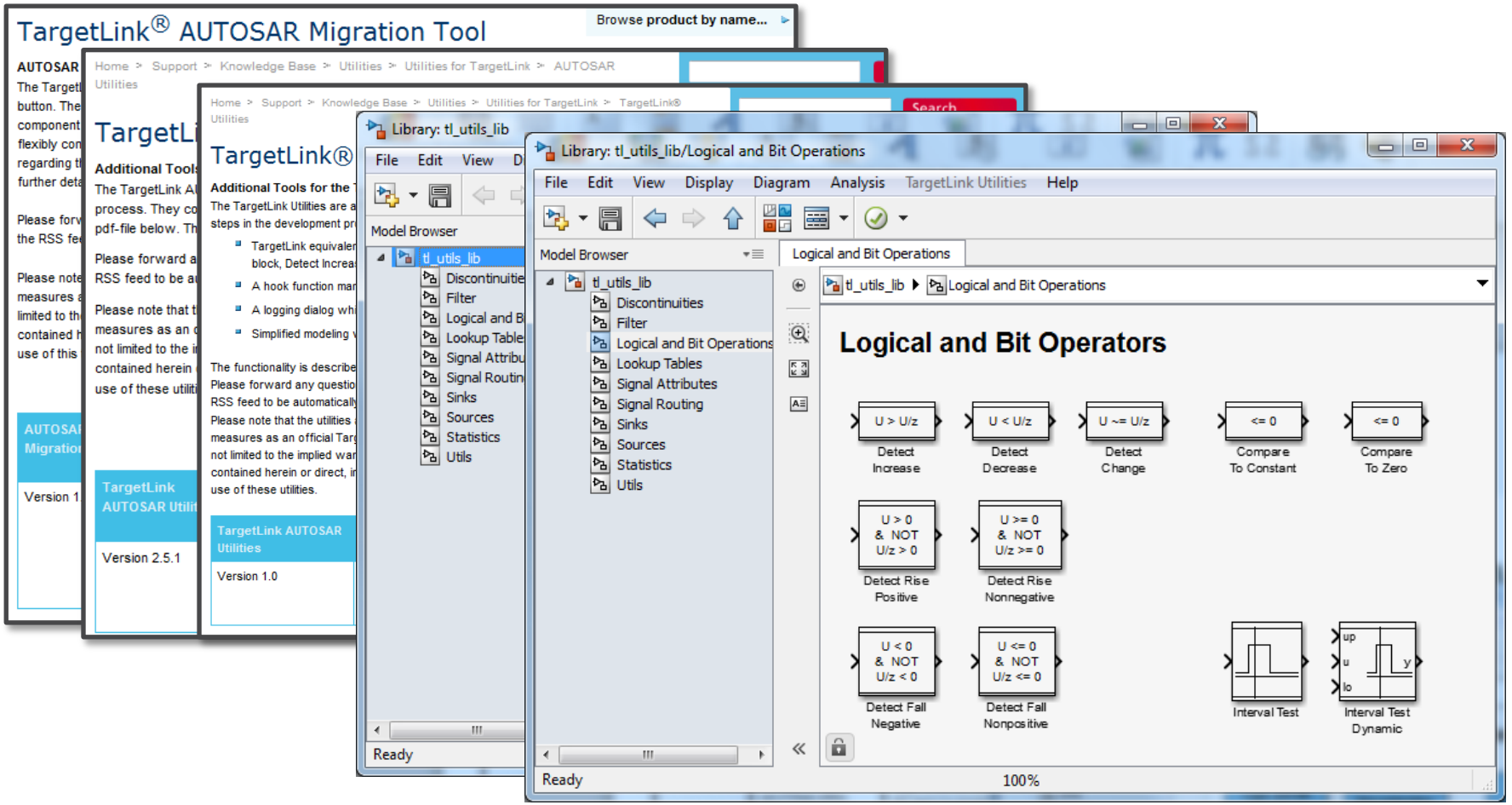


Access to:

- TargetLink patches
- Release information
- Known Problem Reports
- Application Notes
- Modeling Guidelines
- Add-on Tools
- Webinars
- …

- The URL for all developers working with TargetLink
  www.dspace.com/tlpsc

Add-on Utilities for TargetLink



Add-on Tools:

- AUTOSAR Migration Tool: Automatically migrating TargetLink models to AUTOSAR
- TargetLink AUTOSAR Utilities: AUTOSAR Add-ons, additional RTE calls etc.
- TargetLink Utilities: Additional TargetLink blocks, GUIs etc.

Thanks for listening!