The MathWorks™

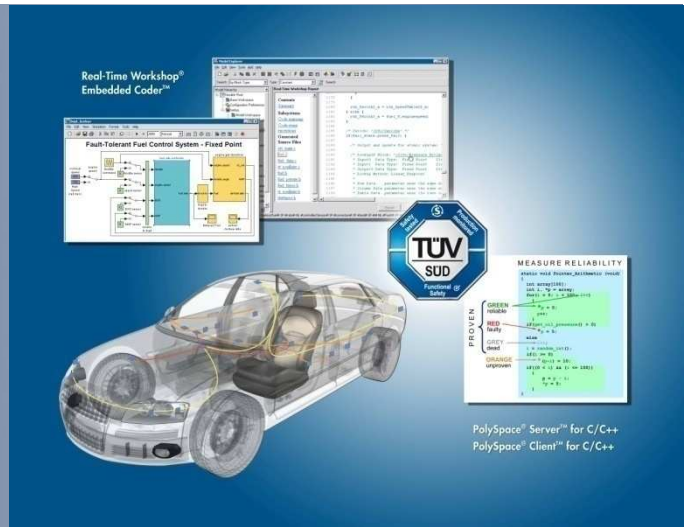
MATLAB®&SIMULINK®

Towards Certification

Paolo Bizzarri

TheMathWorks

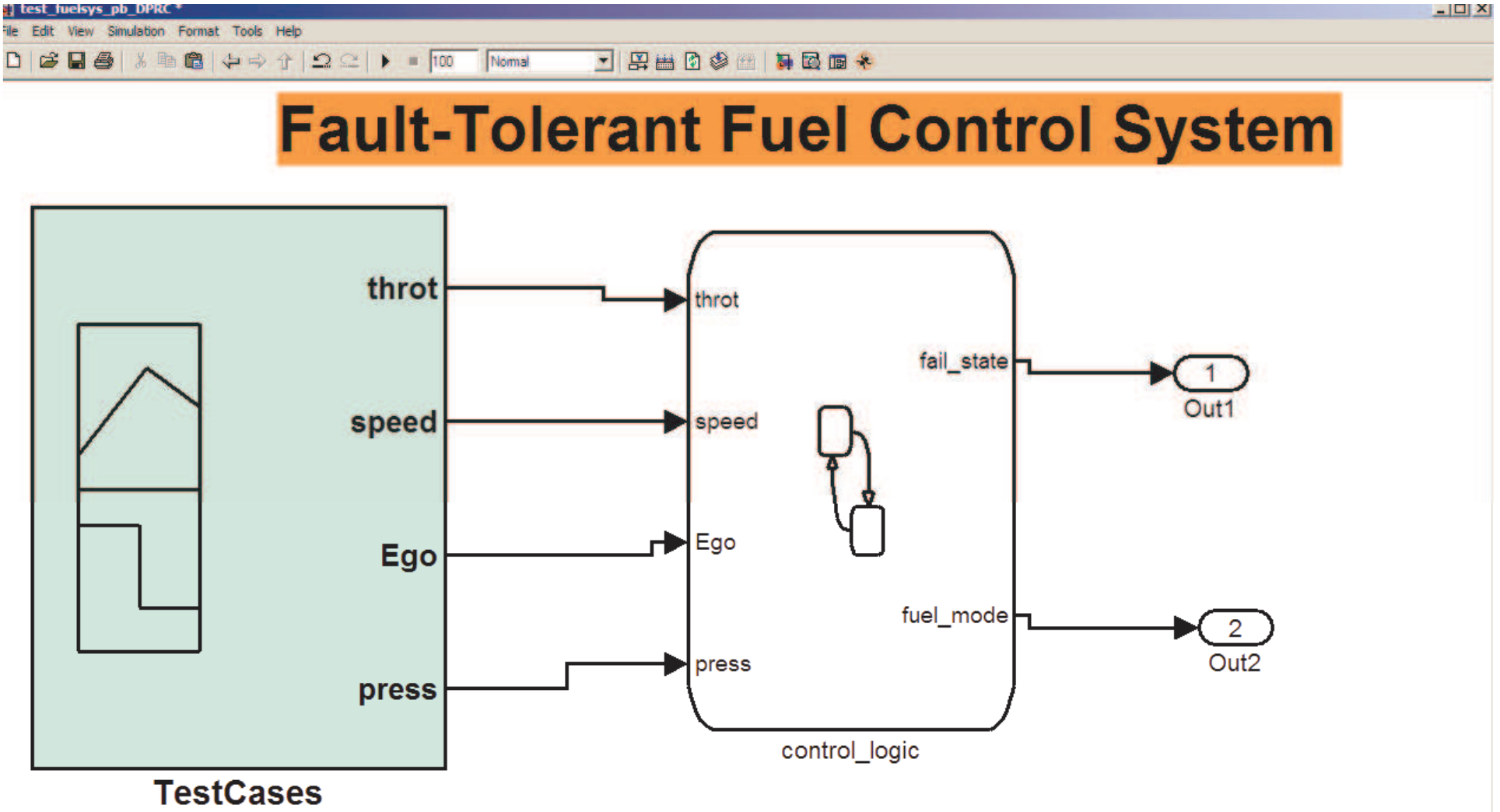
Italy



High-Integrity Applications

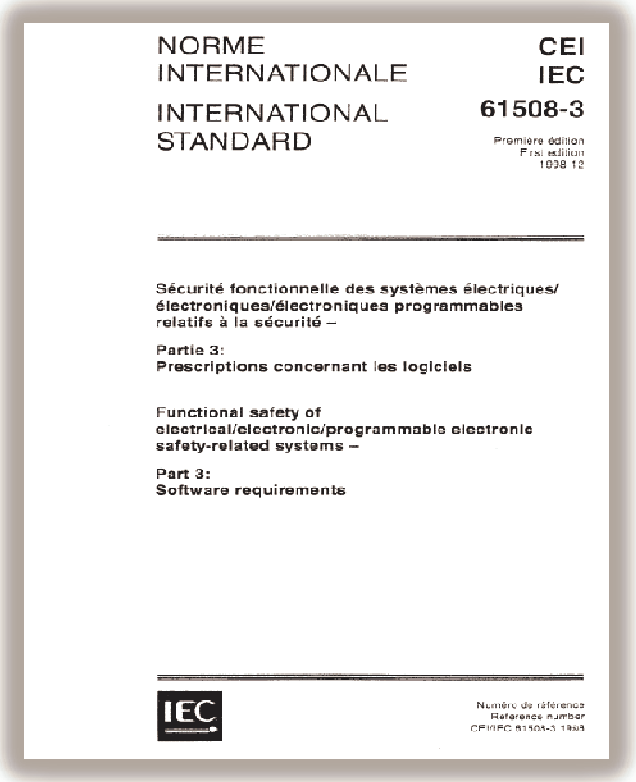
Software-based systems that are designed and maintained so that they have a high probability of carrying out their intended function

Definition: cf. Buncefield Investigation Glossary
http://www.buncefieldinvestigation.gov.uk/glossary.htm


test_fuelsys_pb_DPRC*
File Edit View Simulation Format Tools Help
100
Normal
Fault-Tolerant Fuel Control System
throt
speed
Ego
press
TestCases
throt
speed
Ego
press
fail_state
fuel_mode
1
Out1
2
Out2
control_logic

Standards Landscape

- Aerospace Standards
  - DO-178B (= JAA EUROCAE ED-12B)
  - DO-254
- Generic Standards
  - IEC 61508* (= EN 61508)
- Automotive Standards / Guidelines
  - ISO 26262
  - MISRA-C
  - MAAB Guidelines


NORME INTERNATIONALE
CEI IEC
INTERNATIONAL STANDARD
61508-3
Première édition
First edition
1998-12

Sécurité fonctionnelle des systèmes électriques/électroniques/électroniques programmables relatifs à la sécurité –

Partie 3:
Prescriptions concernant les logiciels

Functional safety of electrical/electronic/programmable electronic safety-related systems –

Part 3:
Software requirements

IEC

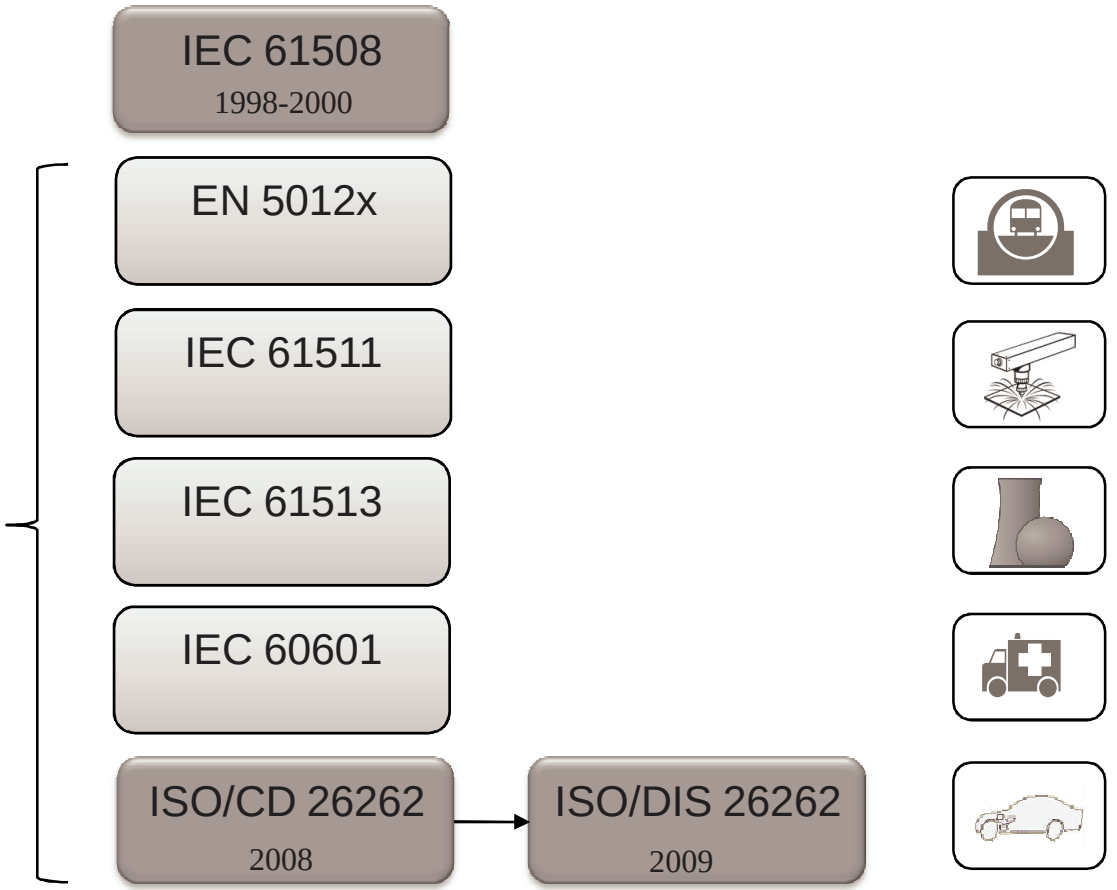
Numéro de référence
Reference number
CEI/IEC 61508-3: 1998

* Used e.g. in automotive and industrial automation

IEC 61508 Derivative Standards

Generic safety standard

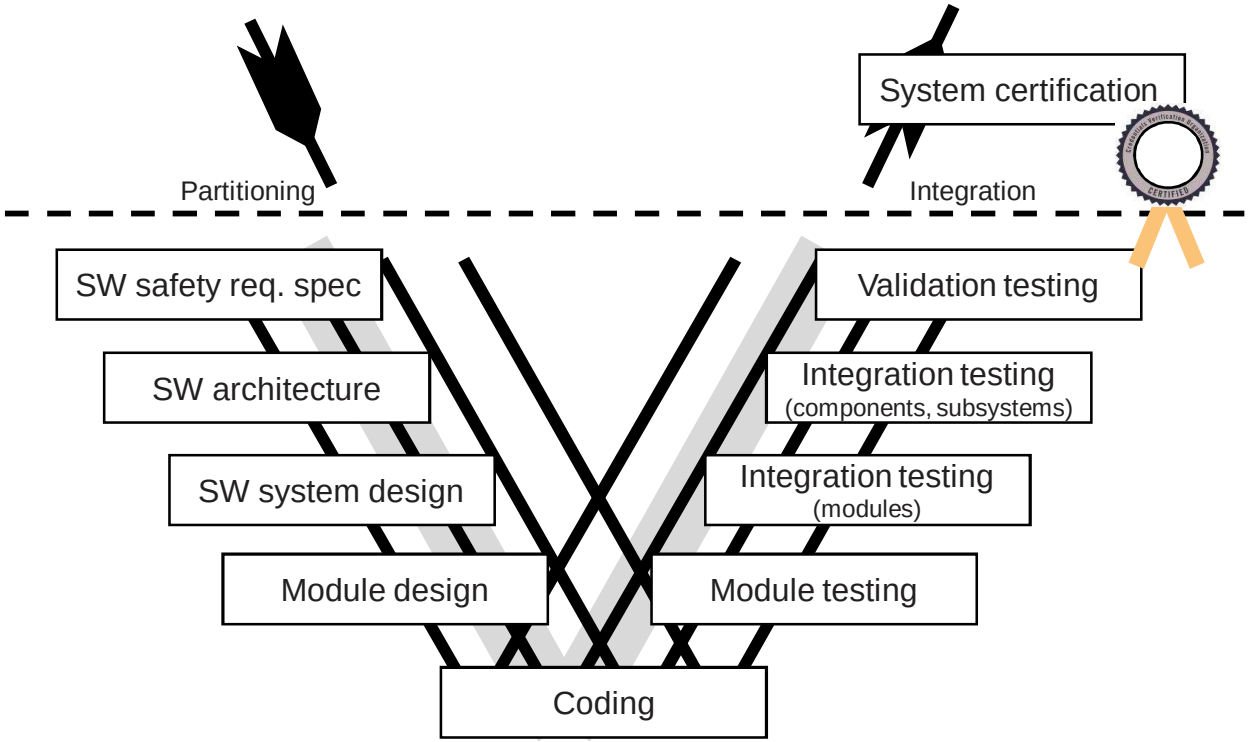
Derivative standards



IEC ... International Electrotechnical Commission
ISO ... International Organization for Standardization

Development Process for High-Integrity Applications: System Certification

- Recognition by a certification authority (e.g. TÜV) that an in-vehicle system complies with the requirements of a standard



IEC Certification Kit (for IEC 61508 and ISO 26262)

- **Provides certification artifacts, including:**
  - TÜV SÜD certificates
  - TÜV SÜD certificate reports
  - Verification and validation workflow documentation and guidance
- **Support includes the following tools:**
  - Real-Time Workshop Embedded Coder (R2008a, R2008b, R2009a, R2009b)
  - PolySpace Verifier for C (R2007a+)
  - PolySpace Client / Server for C/C++ (R2008a, R2008b, R2009a, R2009b)

www.mathworks.com/products/iec-61508/


СЕРТИФИКАТ ◆ CERTIFICADO ◆ CERTIFICAT

CERTIFICATE

No. Z10 08 05 67052 001

Holder of Certificate: The MathWorks, Inc.
3 Apple Hill Drive
Natick MA 01760-2098
USA

Factory(ies): 67052

Certification Mark:

Product: Software Tool for Safety Related Development

Model(s): Real-Time Workshop® Embedded Coder™

Report
on the
Certificate
Z10 08 05 67052 001

Software Tool for Safety Related Development

Real-Time Workshop® Embedded Coder™

Manufacturer:
The MathWorks, Inc.
3 Apple Hill Drive

СЕРТИФИКАТ ◆ CERTIF

Factory(ies): 63726

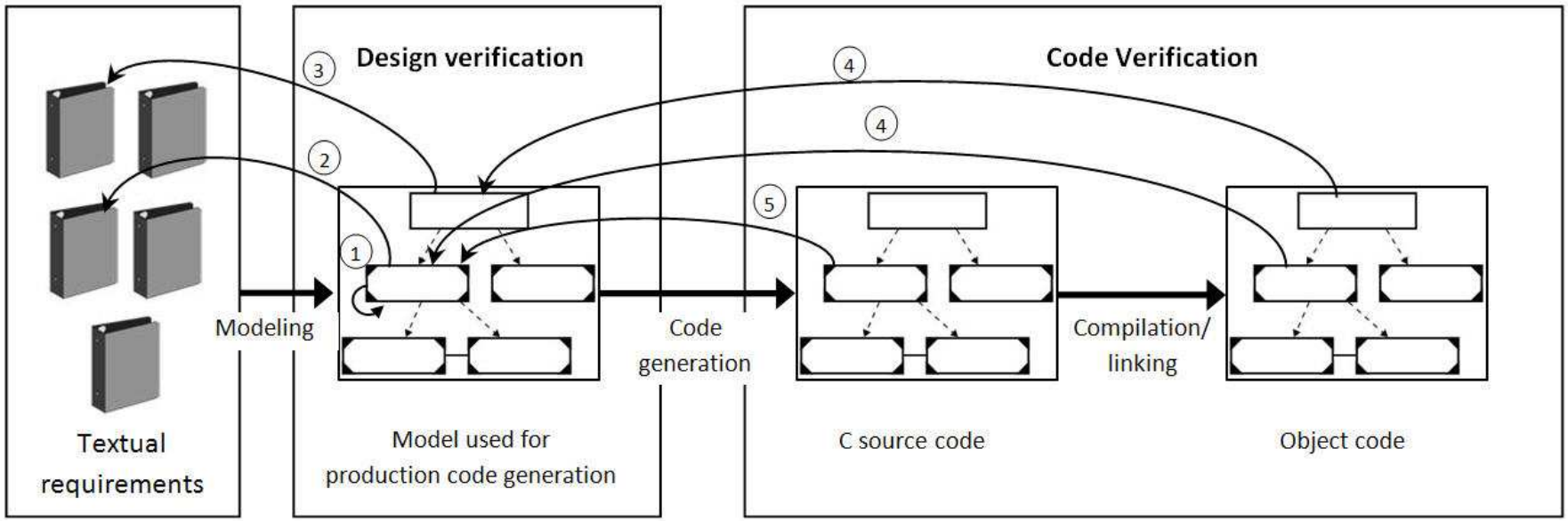
Certification Mark:

Product: Software Tool for Safety Related Development
Software Verification Tool

Model(s): Polyspace Verifier R2007a+

Note: Real-Time Workshop Embedded Coder and PolySpace products for C/C++ were not developed using certified processes.

Workflow for Verification and Validation of Models and Generated Code



Verification & Validation at the Model Level

- Goal of design verification is to gain confidence in the model, which is used for production code generation

- Following IEC 61508-3, the design verification part of the reference workflow comprises a combination of
  - Reviews & static analyses, and
  - comprehensive functional testing activities at the model level

Review and Static Analysis

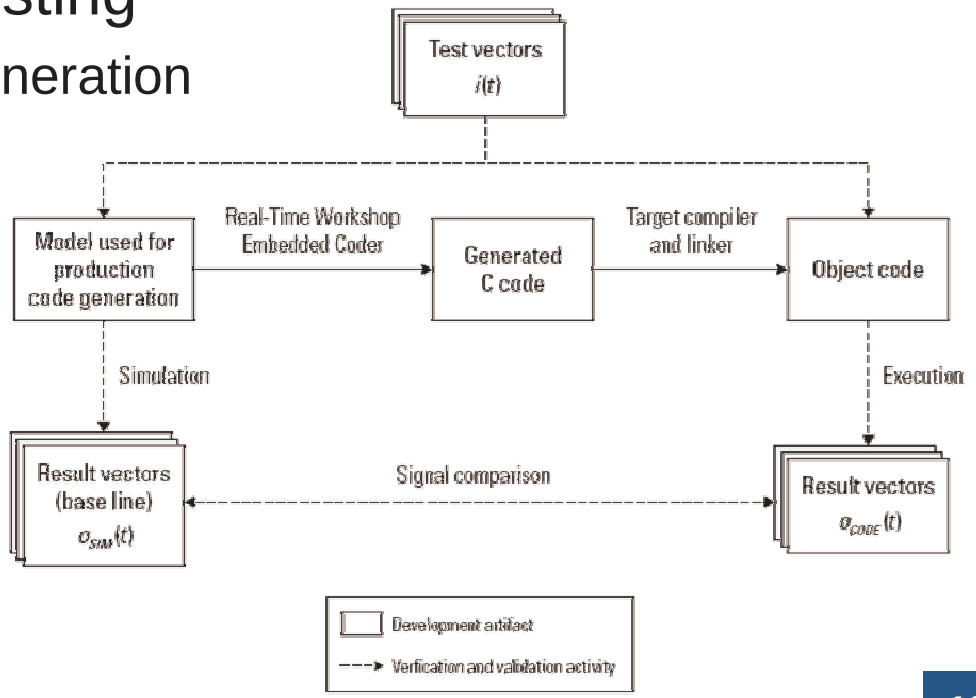
- Model components considered as modules should be reviewed
- If feasible, manual model reviews should be supported by automated static analyses of the model
- Modeling guidelines should be used, and adherence with the guidelines should to be assessed

Module and Integration Testing

- Model components should be functionally tested using systematically derived test vectors.
- Objective of module testing is
  - demonstrate that each model component performs its intended function and does not perform any unintended functions
- As module testing is completed, module integration testing should be performed with predefined test vectors
  - model integration stages should be tested in accordance with the specified integration tests.
  - Tests should show that all model modules and model subsystems interact correctly to perform their intended function and do not perform unintended functions.

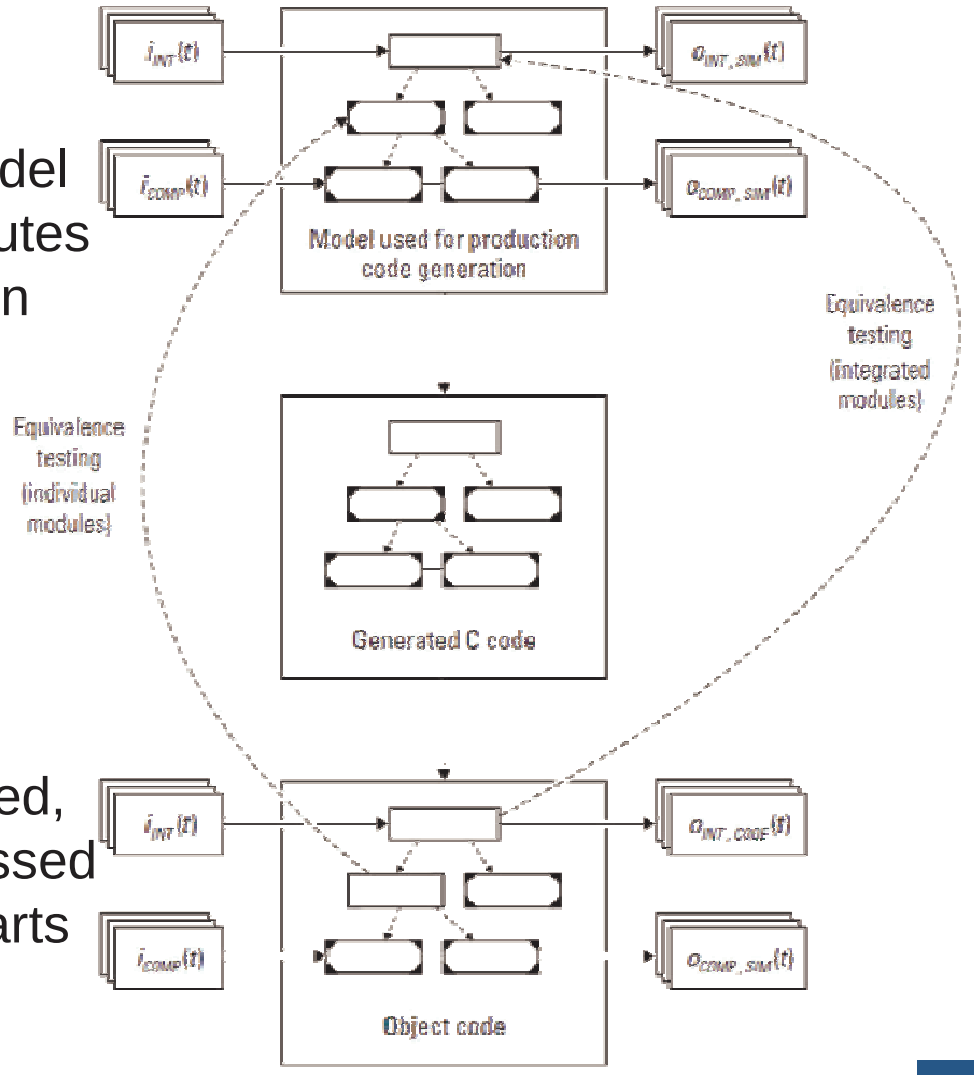
Verification & Validation at Code Level

- The workflow uses
  - translation validation through systematic testing
  - to demonstrate that the execution semantics of the model is being preserved during code generation, compilation, and linking
- Numerical Equivalence Testing
  - Equivalence Test Vector Generation
  - Equivalence Test Execution
  - Signal Comparison



Equivalence testing of individual and integrated modules

- Equivalence testing between model and resulting object code constitutes a core part of the code verification workflow
  - comparative testing or
  - back-to-back testing
- If the coverage is not sufficient, additional test vectors should be created.
- If full coverage cannot be achieved, uncovered parts should be assessed and justification for uncovered parts provided.



Prevention of unintended Functionality

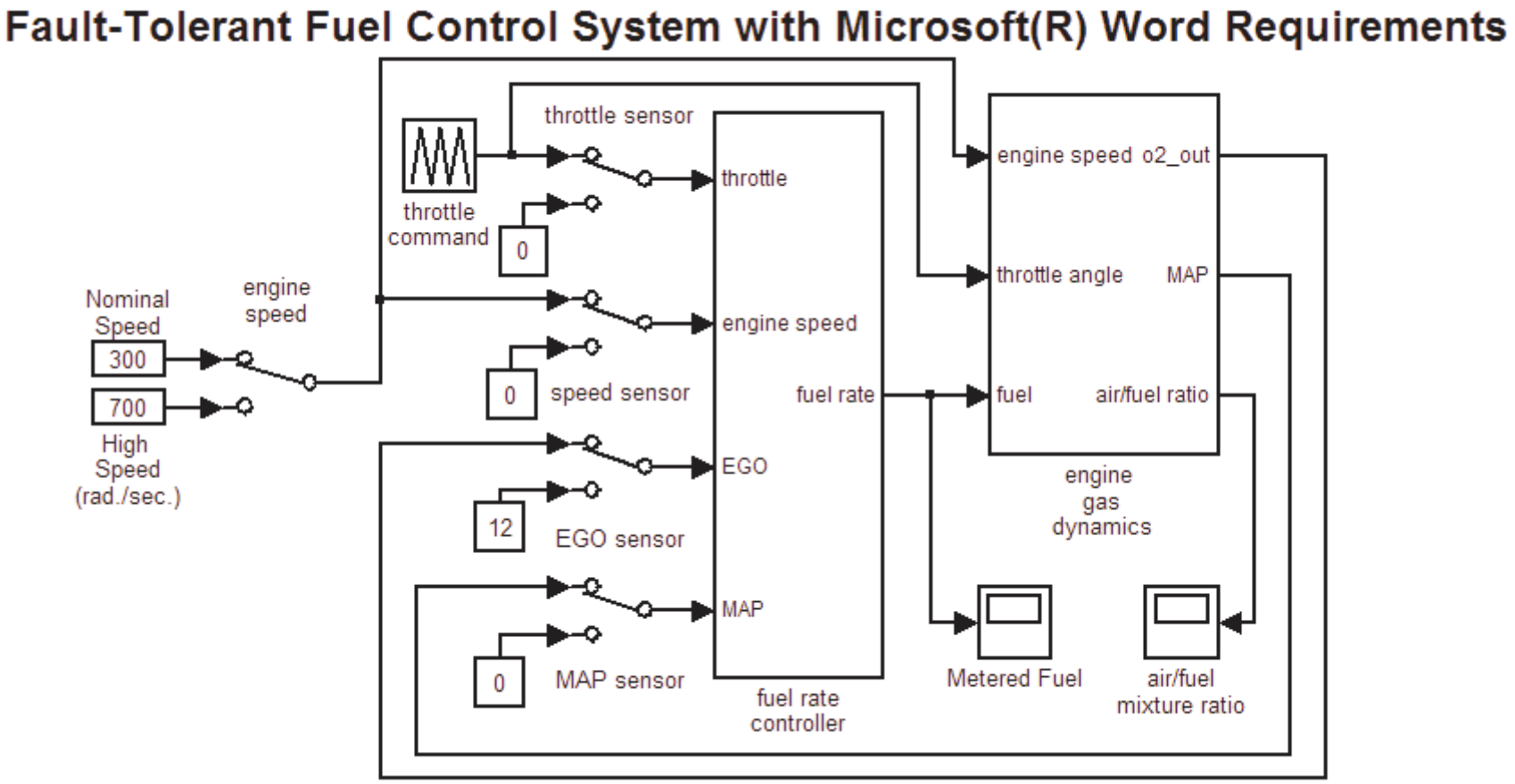
- Model versus Code Coverage Comparison
  - structural coverage metrics should be used on the model and code level respectively
  - decision coverage at the model level and branch coverage (C1) at the code level can be used in combination
  - Discrepancies between model and code coverage shall be assessed.

  ➢ If the code coverage achieved is less than the model coverage, unintended functionality could have been introduced

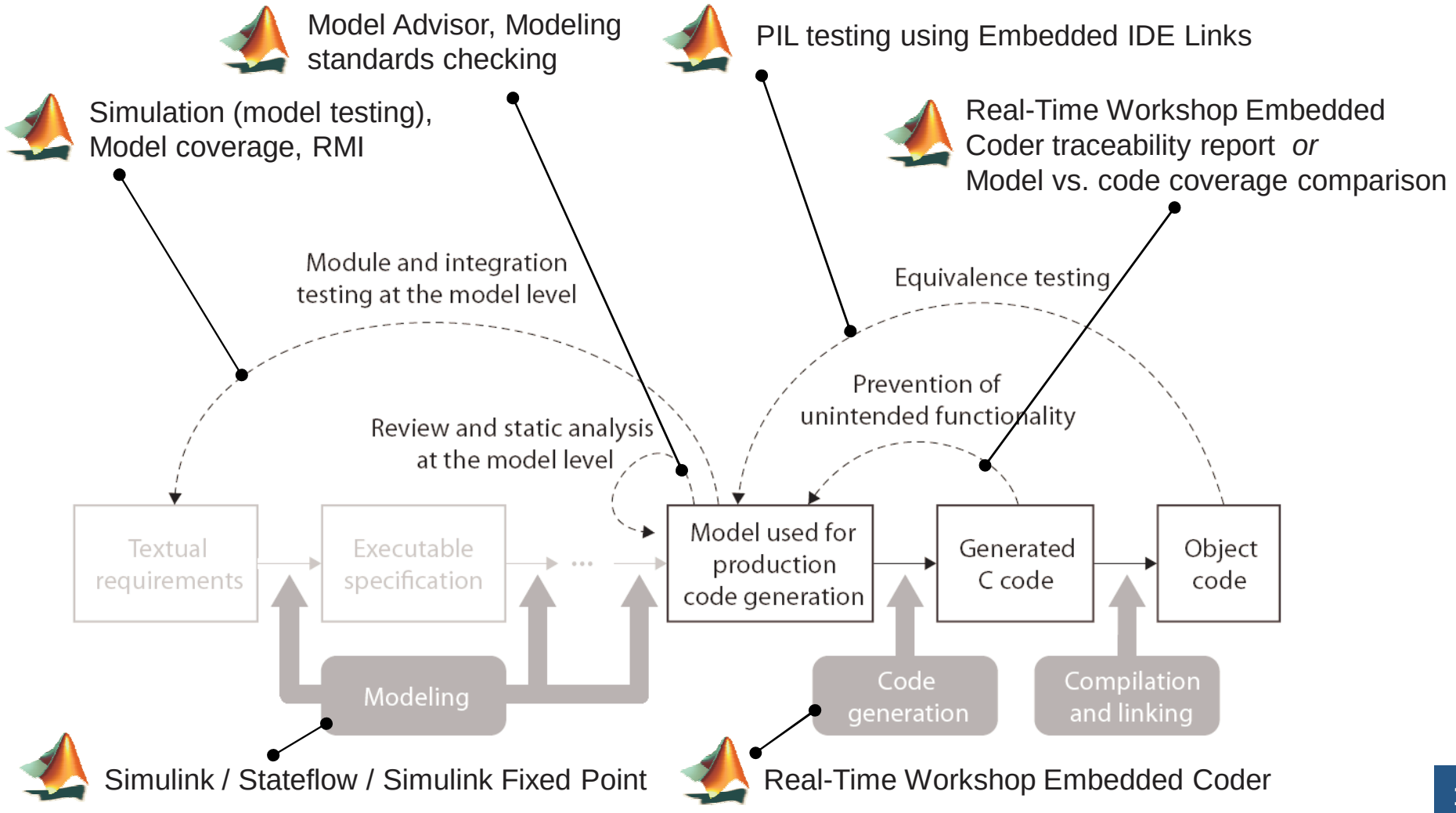
Prevention of unintended Functionality

- Traceability Review
  - Traceability analysis of the generated C source ensures that all parts of this code can be traced back to the model used for production code generation
  - The generated code is subjected to a limited review that exclusively focuses on traceability aspects
  - Non-traceable code shall be assessed

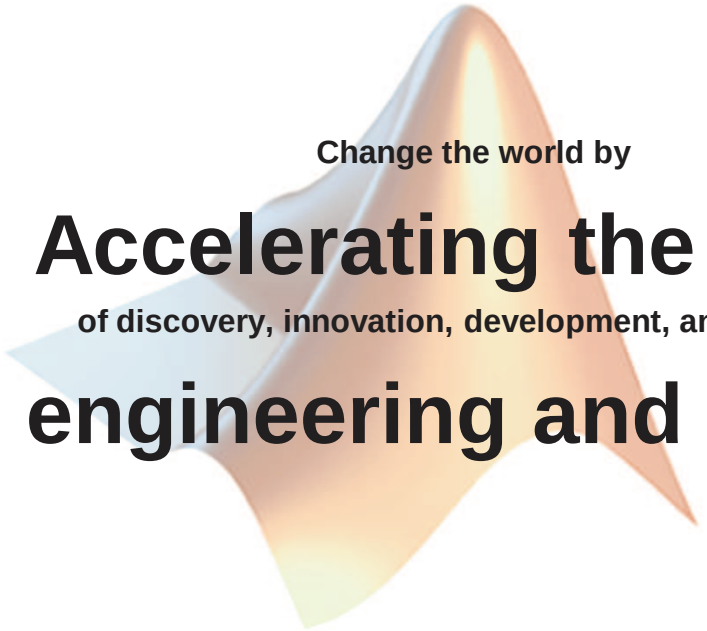
WORKFLOW DEMO DETAILS



Example IEC 61508 Workflow for Model-Based Design with MathWorks Products



The MathWorks



Change the world by

Accelerating the pace

of discovery, innovation, development, and learning

in engineering and science