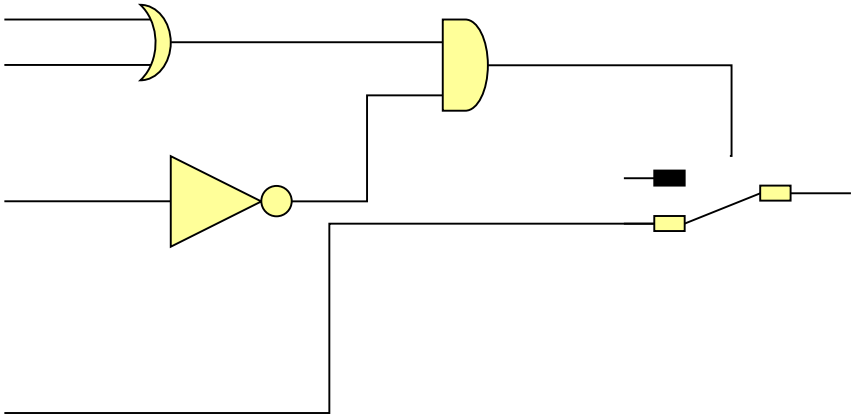
DO-178C / ED-12C
Model Based Supplement



Pierre Lionne,
SC-205 / WG-71 SG-4 Co-Chairman

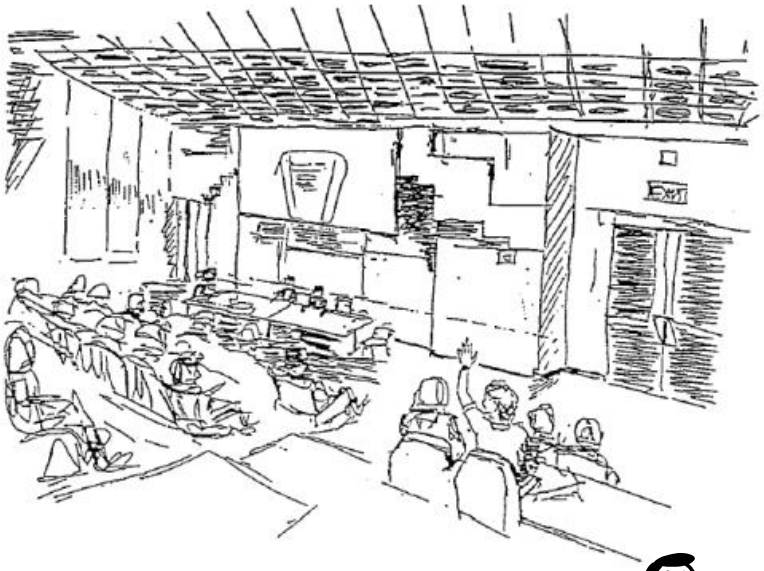
1 Nov. 2011

Summary

- Introduction
- Foundations Concepts
- Highlights
- Conclusion

Introduction

Introduction



Issues

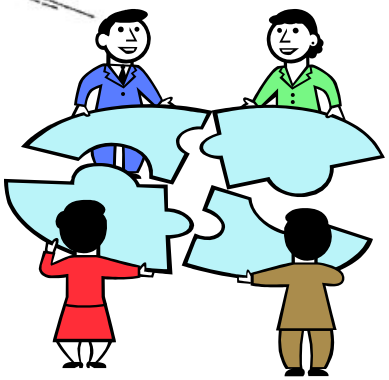
TOR

DO-178C

ED-94C

Supplement X

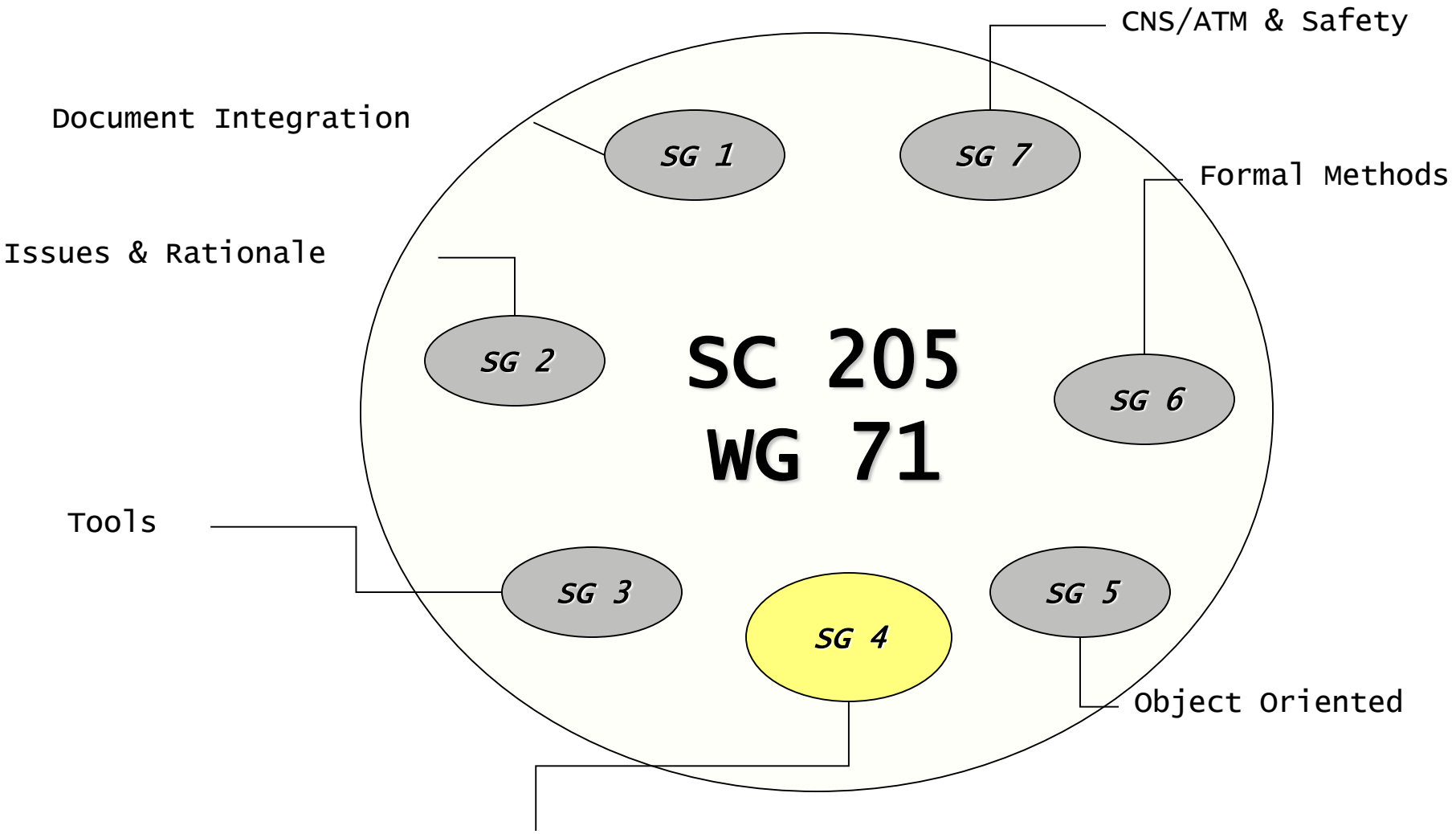
Supplement Y



ED-94B

DO-178B

Introduction



Foundation Concepts

Foundation Concepts

- Models to express requirements
- Scope of supplement
- Modeling Technique
- Model “Parent” Requirements
- Simulation

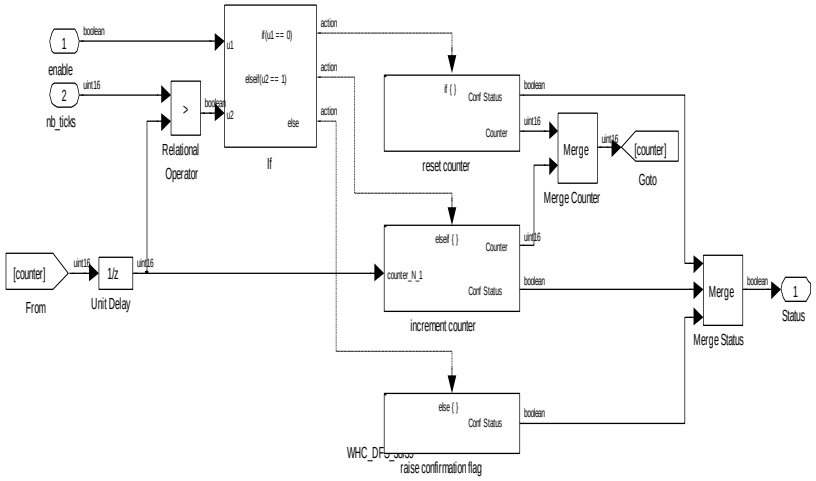
Concept #1

- Model is an acceptable means to express **completely** software **requirements or architecture**

Req_001: ***The XX module shall Wait 10ms before entering in blabl state***

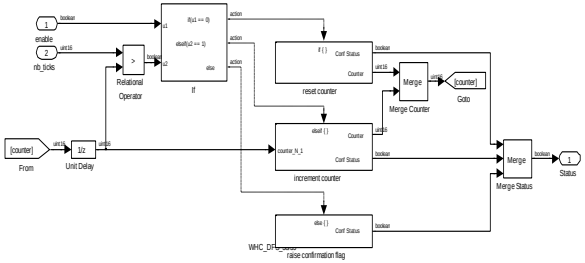
Req_002: ***The XX module ….***

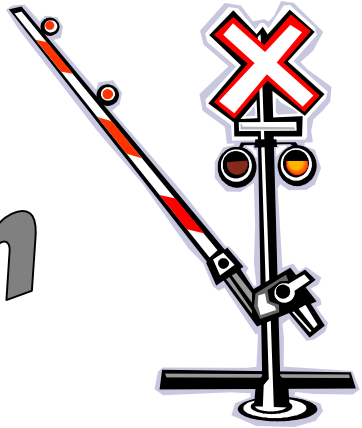
Derived Req_003: ***…***

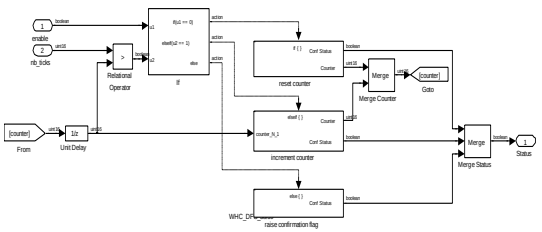


Concept #2

- The supplement applies to any model that is used to define software artifacts **whatever the process that produced it**







Concept #3

- **Modeling Technique** =
  - A Modeling Language

    AND
  - A manner of using this language

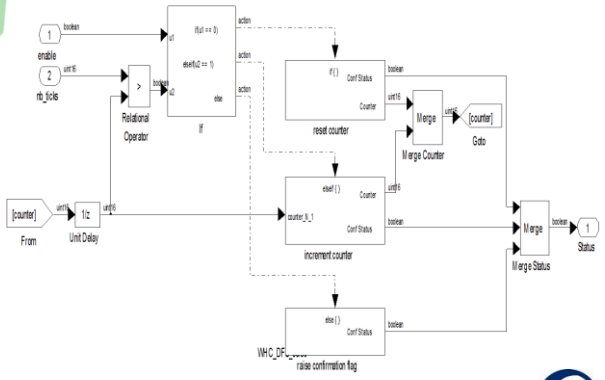
- Modeling Technique has to be **suitable** to the type and to the level of abstraction of the information to be expressed

- Modeling Technique have to be described in **Model Standards**

Concept #4

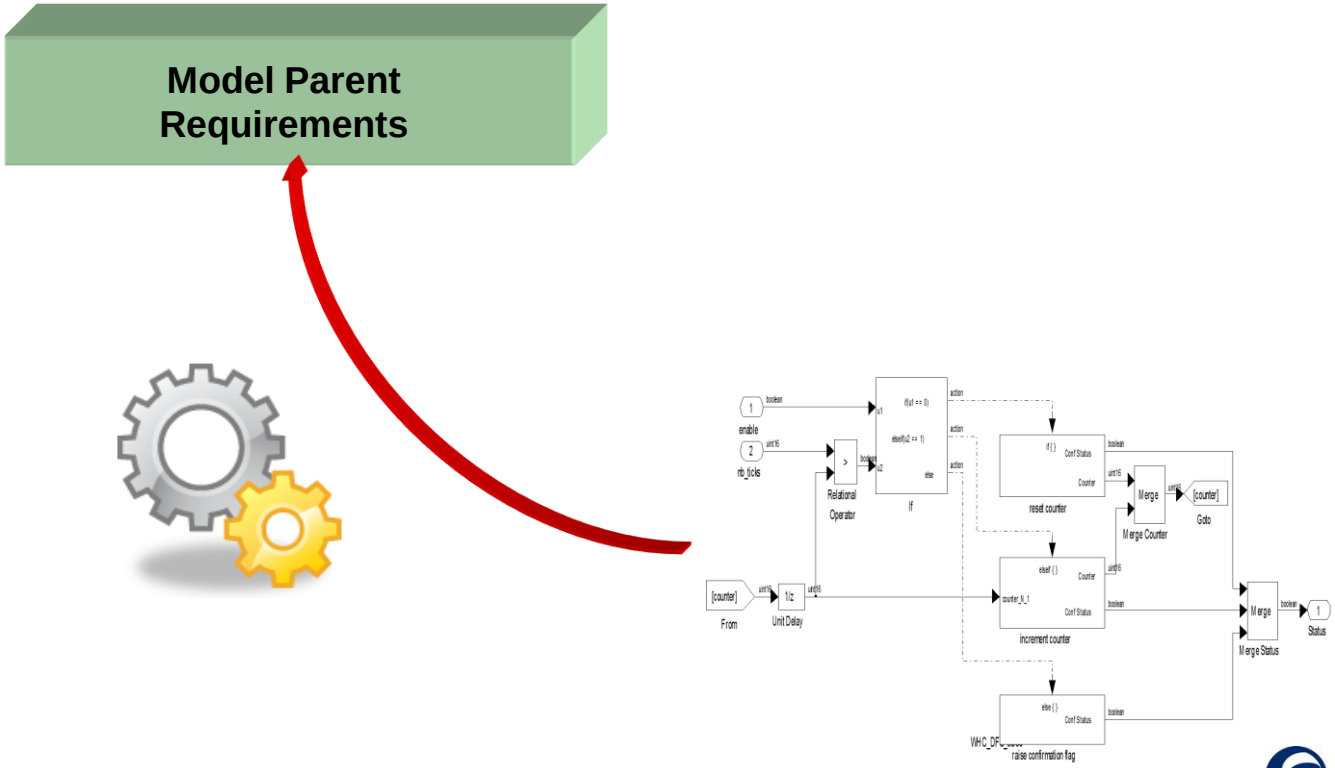
- Model should be developed from a complete set of requirements and constraints **external** to it

Model Parent Requirements



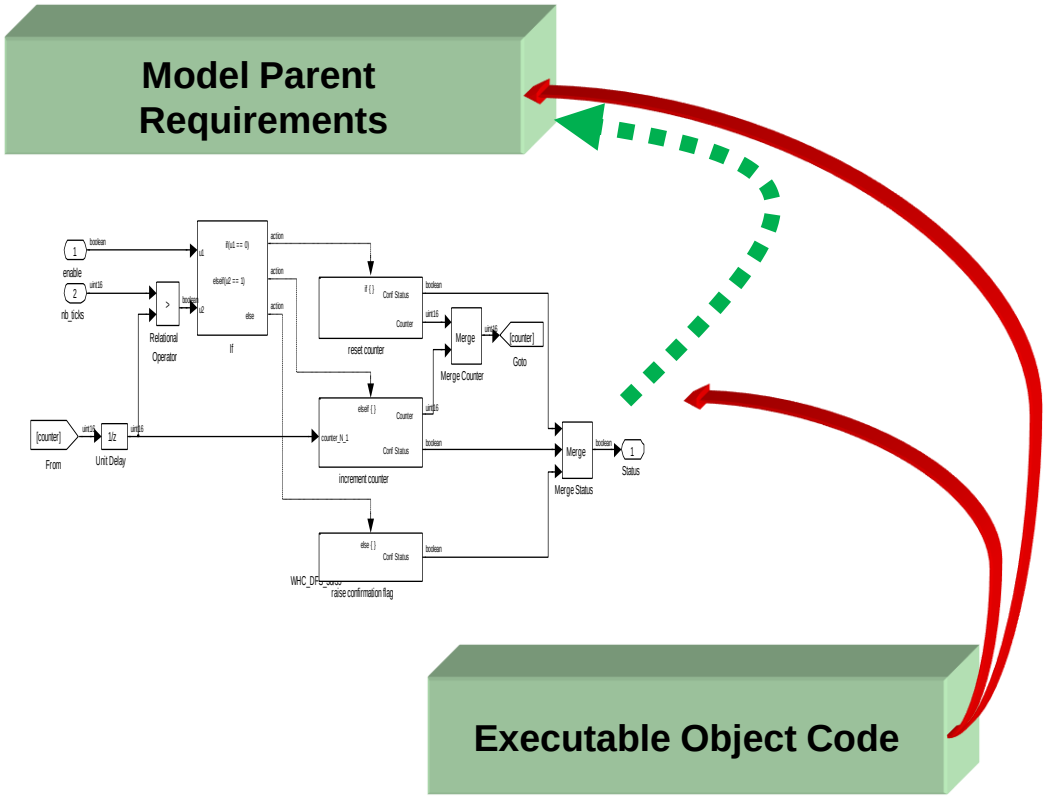
Concept #5

- **Simulation**: appropriate means to support model verification



Concept #6

- **Simulation** may be used to support the testing effort



Highlights

Highlights

- System / Software
- Planning Process
- Development Process
- Verification Process
- Tools

System / Software

- Interfaces between System and Software processes updated to address the case where **system team** produces a **software model**

- Introduction of **Model Standards**
  - Syntax & Semantic of the language
  - Constraint on complexity
  - Means to identify Requirements
  - Derived requirements identification
  - Means to establish traceability
  - …

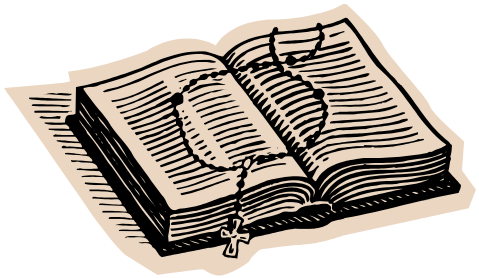
Development Process

- **Same guidance** apply for requirements expressed in a model

- **Model elements** which do not represent requirements should be identified

Guidance from DO-178C / ED-12C

Core Document remains applicable



Simulation & model verification:

- New means => New artifacts:
  - Simulation Cases & Procedures
  - Simulation Results
- Simulation Cases **based on Model Parent Requirements**

Verification Process

Model Parent Requirements

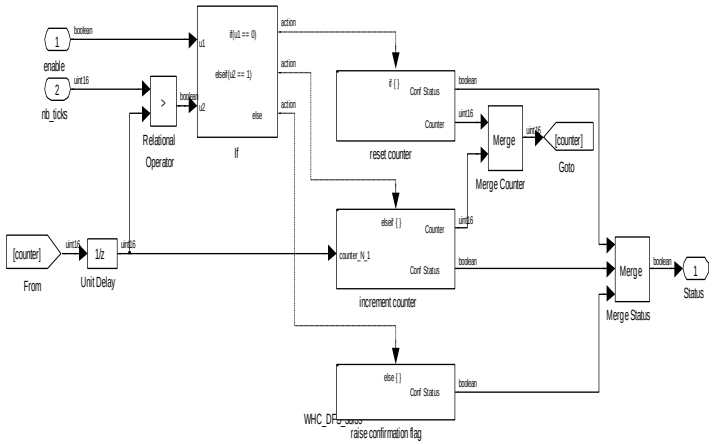
Simulation Cases

Simulation Procedures

Simulation Results

Development

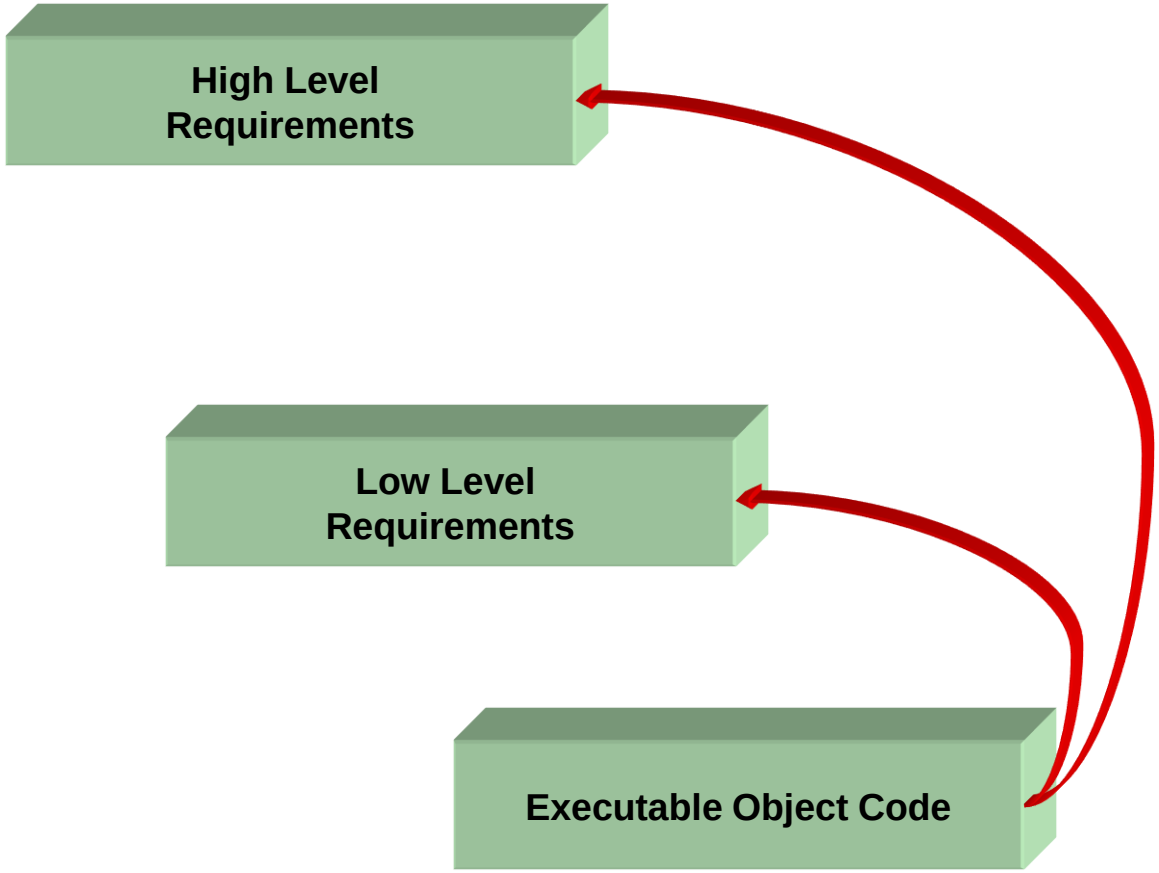
Verification



Test:

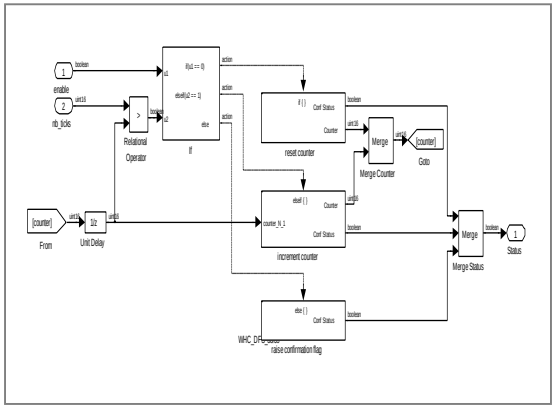
- Same guidance than in DO-178B / ED-12B:
  - Compliance & Robustness with LLR
  - Compliance & Robustness with HLR

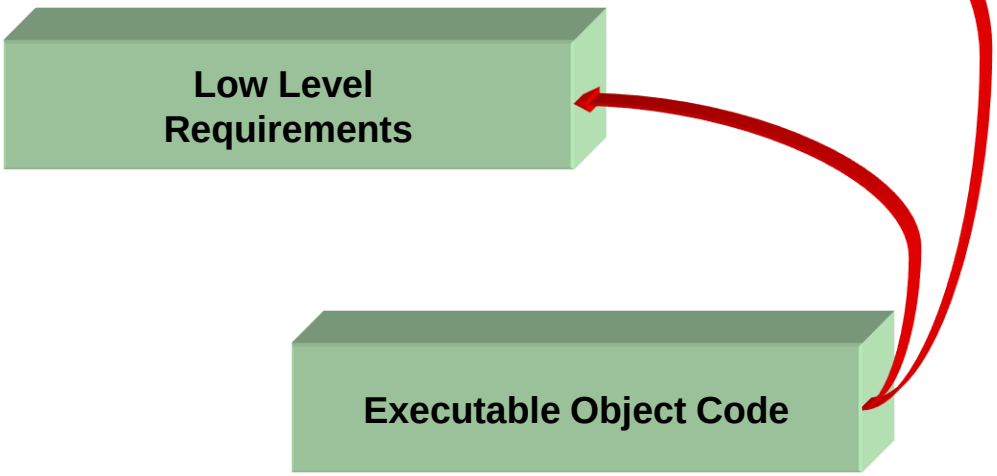
Test (classical)



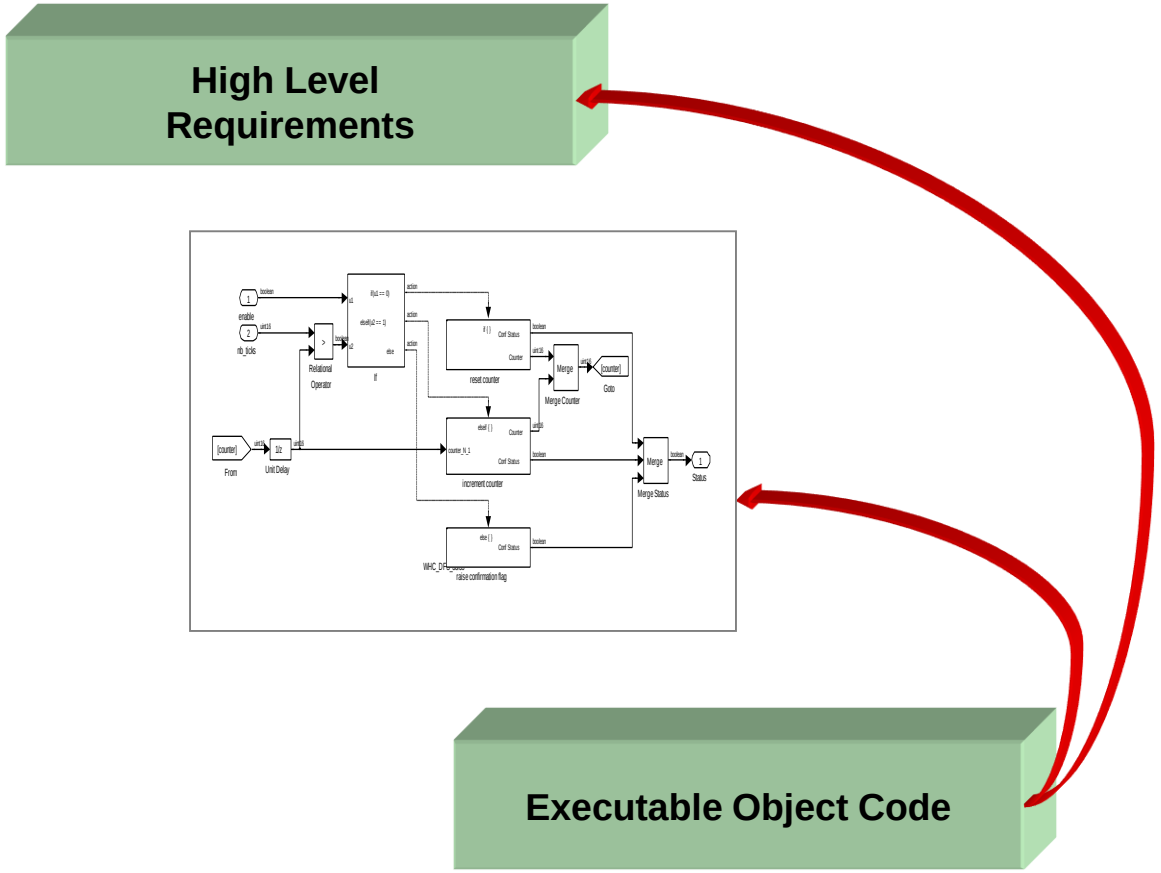
Test (example #1)

Model = HLR



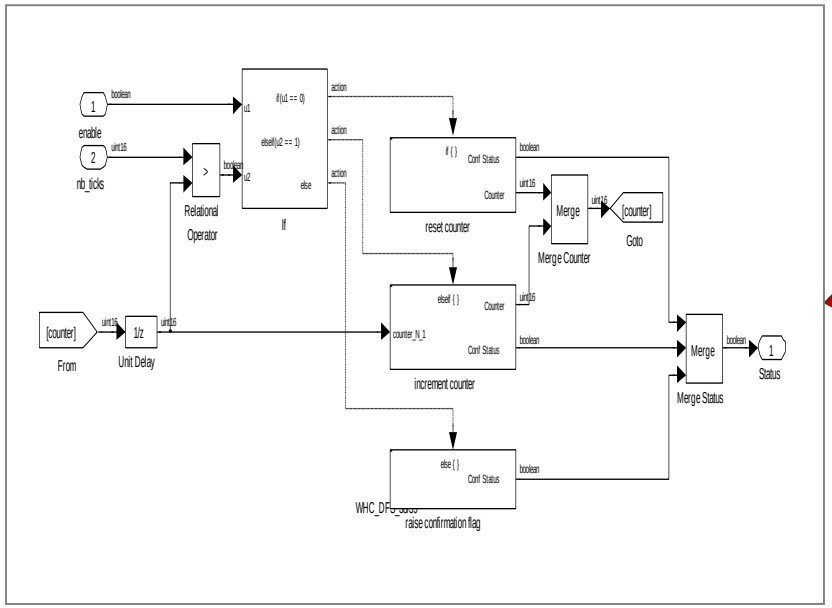


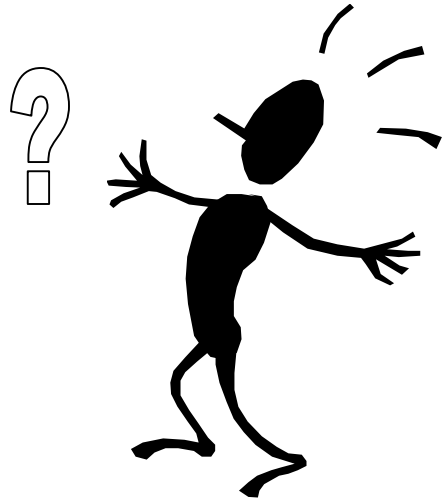
Test (example #2)

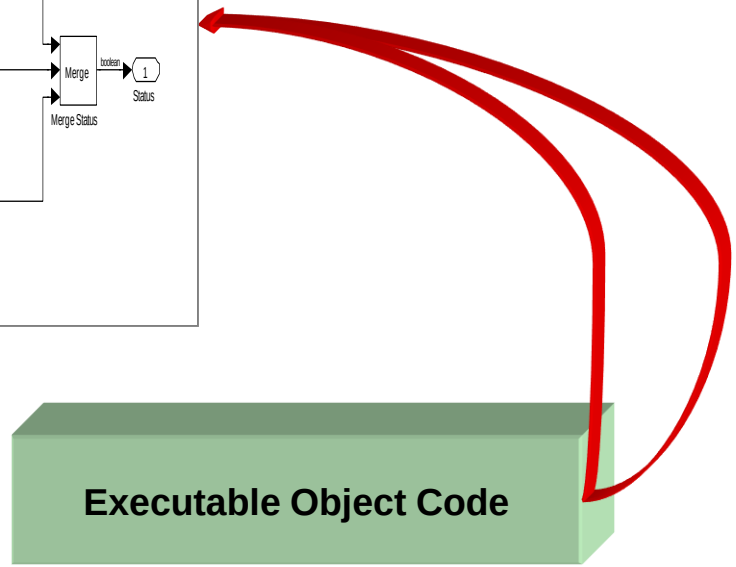


Test (example #3)

Model = HLR + LLR

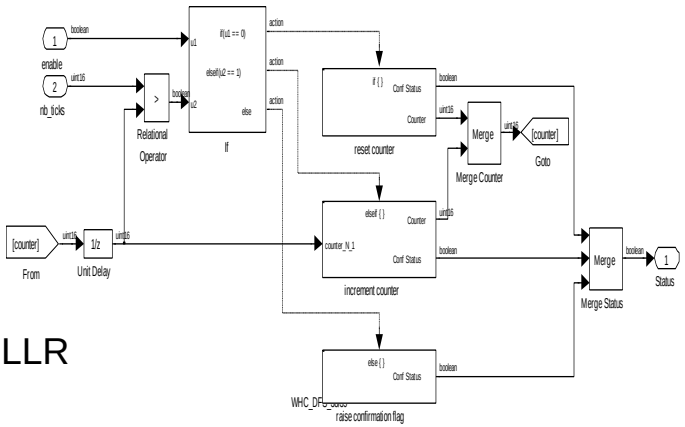






Test (example 3)

Model Parent Requirements



Model = HLR + LLR

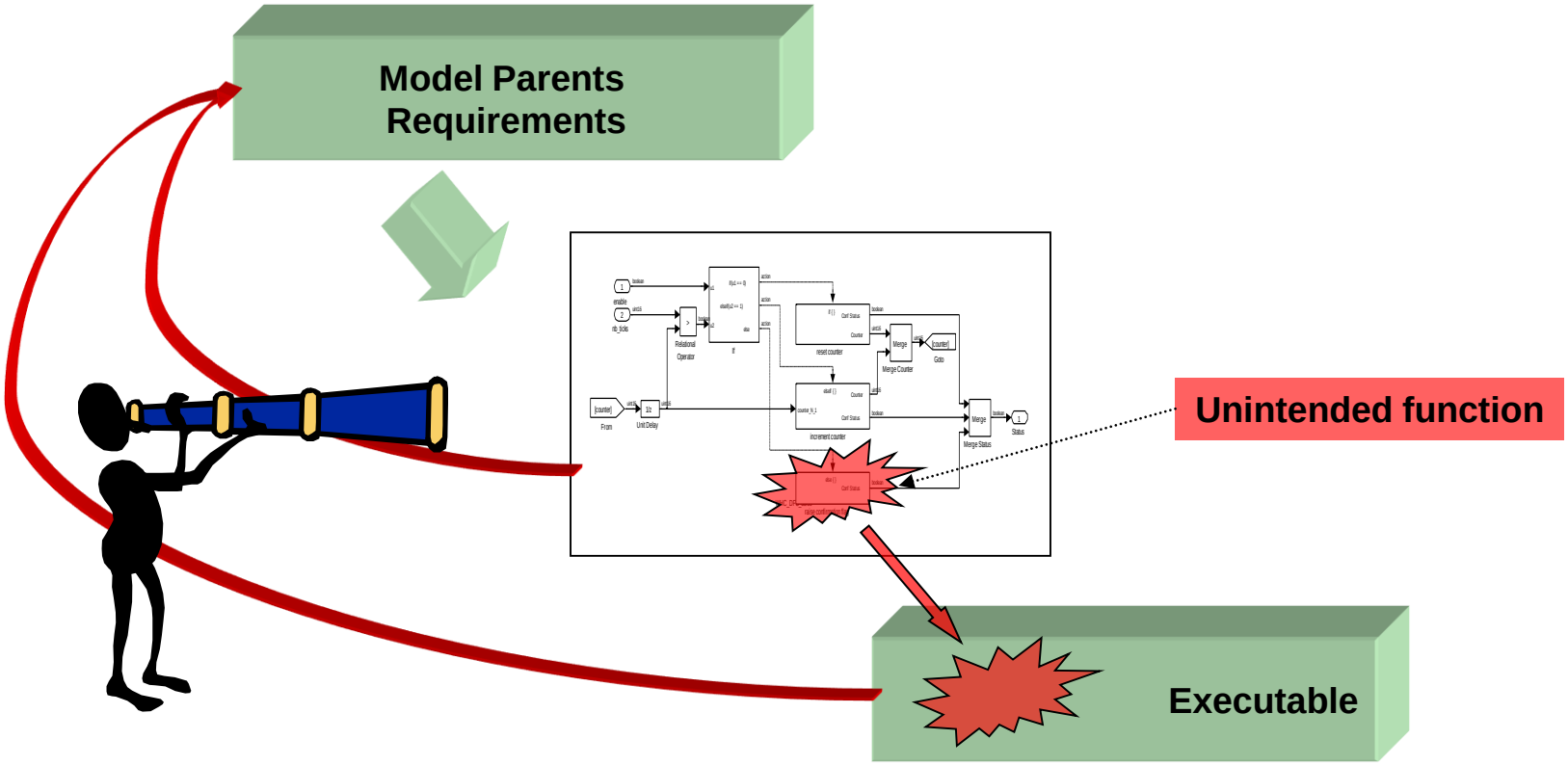
Executable Object Code

Test (example 3)

When model express both LLR and HLR, it is required to show:

- Compliance & Robustness of **EOC with Model**
- Compliance & Robustness **of EOC with Model Parent Requirements** (whatever the process that produced it)

Model Coverage Analysis: Detect unintended functions in a model

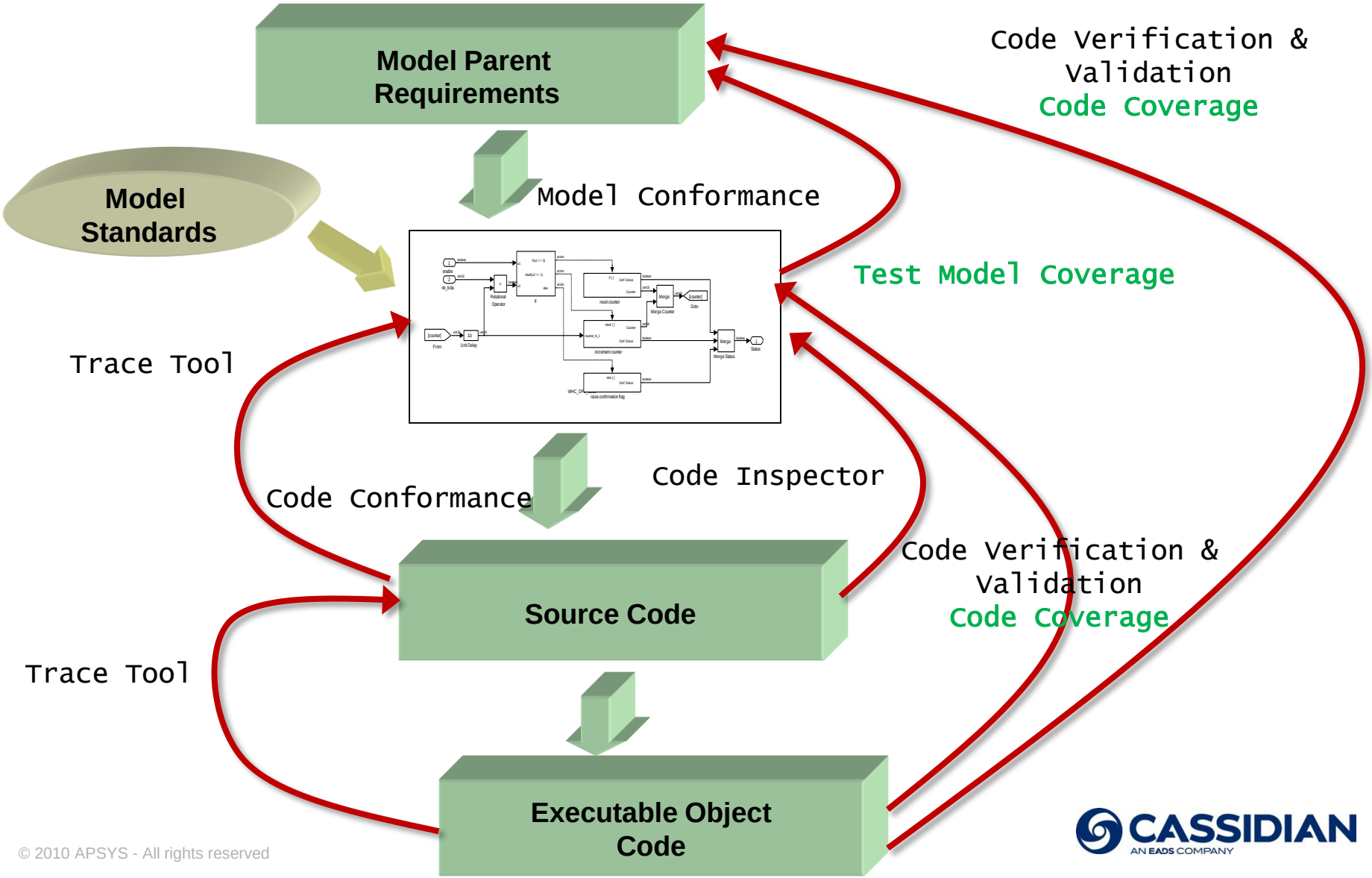


Simulation & Test:

- **Some testing objectives** can be achieved by a **combination** of simulation and other traditional means.

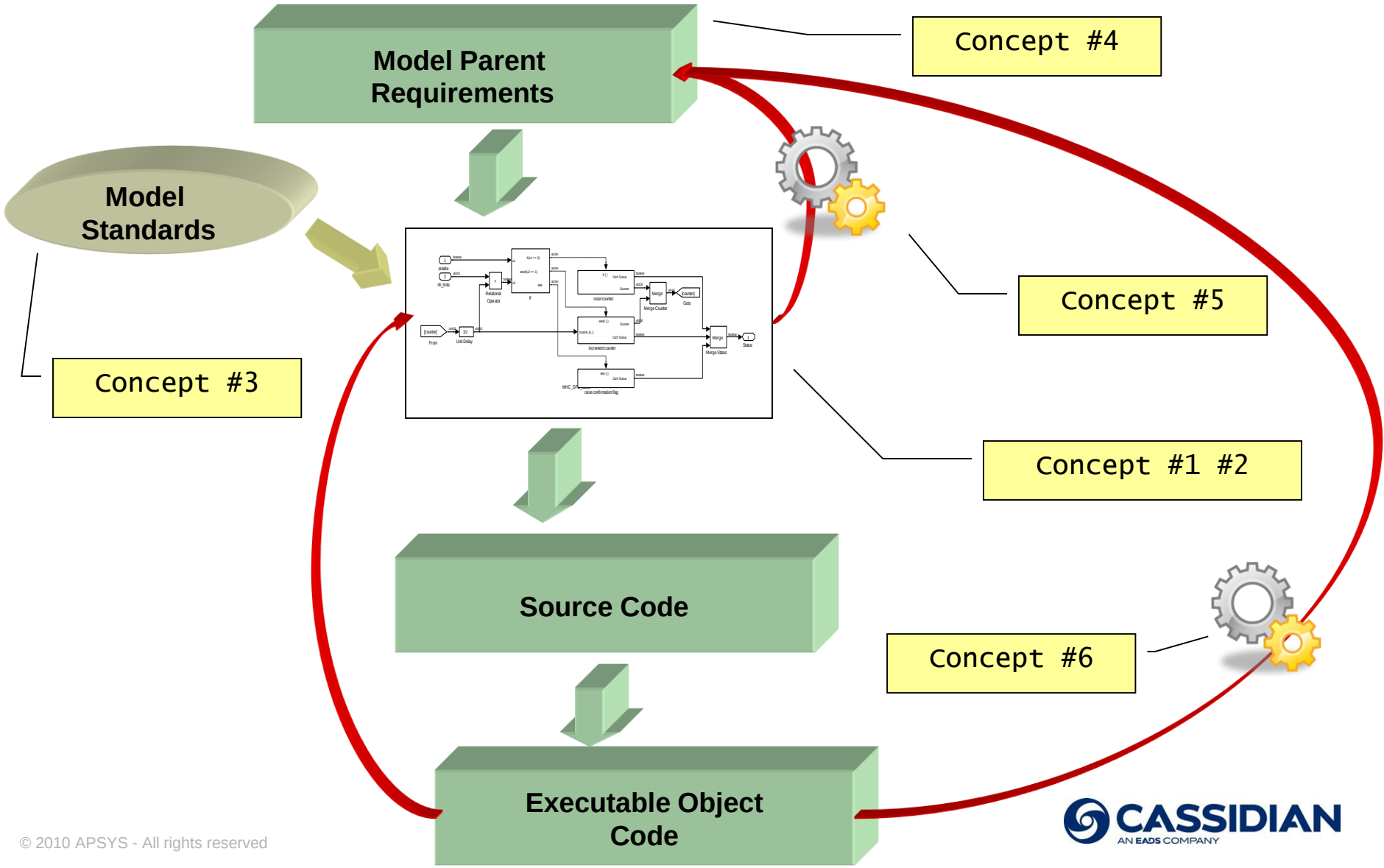
- **HW/SW Integration** test objectives **cannot** be achieved by **simulation**.

Tools



Conclusion

Highlights



Conclusion

- In the continuity of existing rules
- Consistent with current practices
- Try to anticipate future trends

Thank you for your attention!