RTCA SC-205: DO-178C
Model-Based Development & Verification

Mark Lillis, Co-Chair Sub Group 4
Agenda

► Model Introduction
► Foundation Concepts
► SG-4 Approach & Draft Status
► Outlook & Questions
Introduction

- RTCA Special Committee-205 and EUROCAE Working Group-71
- Working on updates to DO-178B/ED-12B
- Sub-Group 4 is working on Issues related to Model-Based Development & Verification (MBD&V) of Airborne Software
- Eurocaee Co-Chair is Pierre LIONNE, System & Software Process Assurance at EADS APSYS in Toulouse
Models

A Summary of the Topic
Models

- This term *model* should be thought about without restrictions
  - Models define behavior
  - The behavior may be for:
    - Dynamic control
    - State-based logic
    - Sequential control logic
    - Combinational control logic
    - Entire system architectures
    - System and/or software verification
Not Everyone’s Model is Codable

► Plant Model: Represent the Engine, Airplane, or other Real-World “Item” that must be controlled

► Models may be able to represent the “Item” in various environments, i.e. altitude, airspeed, ambient temperature, etc… and how they affect the Item’s behavior

► Some Plant models may be used to develop “Control” models

► Model might be used just to create “Expected Results” for verification
Models vs. Tools

►► There are Models
►► There are Tools
►► There are Modeling Tools
►► Model ≠ Modeling Tool ≠ Tools

Some Modeling Tools are part of a Qualified Development Environment

Different issues exist for each of these
Where do models come from within the requirements hierarchy?

► Models: Are they high level requirements (HLR) or low level requirements (LLR)?
  - *The answer is “YES!”*

► Different levels of models may be considered to be either high or low level requirements based on how they are developed and used.
  - This affects how the Annex Table Objectives are applied.

► High Level Requirements come from the Systems Process that is governed by ARP4754.
System Design Model Allocated to Software and Hardware

System Design/Architecture Model

Software Model

Hardware Model

Plant/Environment Model

System Requirements

Validate

Compliance Traceability

Verify

Conformance

Compliance Traceability

System requirements allocated to software as HLR model (DO-178)

System requirements allocated to hardware as HLR model (DO-254)

See Examples #2 and #3

ARP4754
System Design Model Allocated to Software and Hardware

- The previous figure shows how a system design model can be used to allocate portions or sub-models to software and hardware functions.

- Two examples will be shown later where the models allocated to software become HLR directly from the systems process.
SG4 Methodology

The Pathway to an MBD&V Supplement
Since the team had a wide array of perspectives on models, early consensus was impossible. Considerable effort was expended in generating *Foundation Papers* to help shape our common understanding of the model usage issues. The final result was a distilled version of our understanding and agreement on these foundation concepts. Issue Paper #420 serves to hold these concepts. SG4 divided into groups to tackle each section of DO-178 and make proposed wording changes for a supplement in accordance with the IP420 concepts.
1: MODELLING TECHNIQUES MAY BE SUITABLE TO EXPRESS...

15: MODEL SIMULATORS MAY BE...

Foundation Concept
IP420 - Foundation Concepts

► Concepts are a basis to be used to write a complete document in which more precise guidance should be elaborated
  ▪ 15 separate & complementary concepts were formed

► The aim of these concepts is to make everybody aware of the principles adopted by SG-4 in the writing of the MBD&V guidance

► It is obvious that some aspects of the MBD&V issues are not yet addressed in this paper

► To be tailored according to the design assurance level
IP420 - Foundation Concepts

**Usage of models**

► Modelling techniques may be suitable to express requirement and/or design and/or architecture lifecycle data items. (#1)

OR

► Modelling techniques may be suitable to express HLR and/or LLR and/or architecture lifecycle data items. (#1)
IP420 - Foundation Concepts

**System vs. Software (1)**

- MBD&V aspects may be in both the system and the software area. However, the MDB&V guidance only address the models that are involved in the production of the software development lifecycle items. This does not exclude the models representing system lifecycle data items. The guidance is not applicable to models representing system and only system lifecycle data items. (#4)
System vs. Software (2)

- The guidance contained in the supplement should apply regardless of who performs the modelling activities: system engineers or software engineers. The job title of the engineer shall not determine the process they are required to follow. (#5)

- System requirements, which may exist in model form, should be validated in accordance with the applicable guidelines (ARP4754). (#13)
**Decomposition (1)**

- The modelling process may produce several models, having different levels of abstraction. (#3)

- At a given level of abstraction, a model can express a set of requirements and/or design and/or architecture:
  - At times, several modelling techniques may be required to express the complete set of requirements or design.
  - At times, the used modelling techniques may not be suitable to express the complete set of requirements or design. (#8)
Decomposition (2)

Whenever modelling techniques are not suitable as specified in item 8, complementary information should be given by another means to augment that which is represented in the model. (#9)
IP420 - Foundation Concepts

Standards

- Modelling standards shall be defined and used for each modelling technique.
  - The standard shall provide a means to satisfy all traceability expectations.
  - The standard shall provide the semantic and syntax description of the model used. (#2)

- At any level of abstraction, the used modelling techniques shall be appropriate to represent the requirements and/or design without any ambiguity. Modelling standards will address this expectation. (#7)
IP420 - Foundation Concepts

**Derived Requirements**

- At any level of abstraction, a model can contain derived requirements. (#10)

- All derived requirements shall be identified. (#11)

- All derived requirements shall be justified, validated and assessed for potential safety impact. (#12)
IP420 - Foundation Concepts

**Verification (1)**

- Each lifecycle data item output from the modelling process shall be verified.
  - Each successive level of abstraction shall be consistent with its parent requirements.
  - Each successive level of abstraction shall be compliant with the modelling standards.

The verification evidence burden shall not be reduced by the applicant’s choice to use MBD&V. Informal models can be created to validate requirements and produce validation cases for the delivered system, but shall not be used directly as development process artefacts without full verification.

(#14)
**Verification (2)**

- Model simulators may be used to support the demonstration of the correctness and completeness of the models. (#15)
Current Draft

IP-440
How did we get to the current draft?

► Drafts were presented in Vancouver and Toulouse and considerable feedback was received

► FORMAT ISSUES: Concerned with length, complexity of table format, handling of annex tables

► TECHNICAL ISSUES: Concerned about applicability, need for a supplement, scope
IP-433 & IP-436

► Considerable comments on IP-433 Draft encouraged a sub-team to look at a new approach
► Parallel efforts on both new & old document approaches occurred for several months
► The simplified approach incorporated into IP-436 was adopted in Paris and refined in Toulouse as IP-436
► A go-forward approval was given in Toulouse and work on a further-simplified approach is now called IP-440
► IP-440 is an evolutionary change to IP-436 that, hopefully, will address all major issues to both format & technical approach
### Example of Table Changes

#### Table MBDV A-1

<table>
<thead>
<tr>
<th>Objective</th>
<th>Applicability by SW Level</th>
<th>Output</th>
<th>Control Category by SW level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Description</td>
<td>Ref.</td>
<td>A</td>
</tr>
<tr>
<td>5</td>
<td>Model development standards are defined.</td>
<td>MBDV 4.1</td>
<td>⚫</td>
</tr>
</tbody>
</table>

**LEGEND:**
- ⚫ The objective should be satisfied with independence.
- ● The objective should be satisfied.
- Blank Satisfaction of objective is at applicant’s discretion.
- ☑ Data satisfies the objectives of Control Category 1 (CC1).
- ☑ Data satisfies the objectives of Control Category 2 (CC2).

#### Applicability of the Supplement objectives with respect to the Core Document

<table>
<thead>
<tr>
<th>DO-178C Core Document Annex A Table Objectives</th>
<th>Models Used to Define High Level Requirements</th>
<th>Models Used to Define Low Level Requirements</th>
<th>Models Used to Define Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-1 Objective 1</td>
<td>Substitutions: None</td>
<td>Additions: None</td>
<td>Substitutions: None</td>
</tr>
<tr>
<td>A-1 Objective 2</td>
<td>Substitutions: None</td>
<td>Additions: None</td>
<td>Substitutions: None</td>
</tr>
<tr>
<td>A-1 Objective 3</td>
<td>Substitutions: None</td>
<td>Additions: None</td>
<td>Substitutions: None</td>
</tr>
<tr>
<td>A-1 Objective 4</td>
<td>Substitutions: None</td>
<td>Additions: None</td>
<td>Substitutions: None</td>
</tr>
<tr>
<td>A-1 Objective 6</td>
<td>Substitutions: None</td>
<td>Additions: None</td>
<td>Substitutions: None</td>
</tr>
<tr>
<td>A-1 Objective 7</td>
<td>Substitutions: None</td>
<td>Additions: None</td>
<td>Substitutions: None</td>
</tr>
</tbody>
</table>
## Example of Annex Tables

### Table MBDV A-2

<table>
<thead>
<tr>
<th>Objective</th>
<th>Applicability by SW Level</th>
<th>Output</th>
<th>Control Category by SW level</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td><strong>Ref.</strong></td>
<td><strong>A</strong></td>
<td><strong>B</strong></td>
</tr>
<tr>
<td>1 Models are developed.</td>
<td>MBDV 5.6.1a MBDV 5.6.2a MBDV 5.6.2b MBDV 5.6.2c MBDV 5.6.2d MBDV 5.6.2e MBDV 5.6.2f</td>
<td>☐ ☐ ☐ ☐</td>
<td>Software Requirements Data</td>
</tr>
<tr>
<td>2 Derived model elements are provided to the system safety assessment.</td>
<td>MBDV 5.6.1b MBDV 5.6.2c</td>
<td>☐ ☐ ☐</td>
<td>Software Requirements Data</td>
</tr>
<tr>
<td>3 Models are developed.</td>
<td>MBDV 5.6.1a MBDV 5.6.2a MBDV 5.6.2b MBDV 5.6.2c MBDV 5.6.2d MBDV 5.6.2e MBDV 5.6.2f</td>
<td>☐ ☐ ☐</td>
<td>Design Description</td>
</tr>
</tbody>
</table>
New Modeling Glossary Terms

► Model – An abstract representation of a given set of aspects of a system that is used for analysis, simulation and/or code generation and that has an unambiguous, well defined syntax and semantics.
  ▪ Note 1: If the representation is just presented (visualized) as a drawing with no well defined language, the development is not considered to be in the scope of this document.
  ▪ Note 2: The language used to represent the model may be defined as a meta-model but is not part of the model.
  ▪ Note 3: The given set of aspects of a system may contain all aspects of the system or only a subset.

► Model Architecture – The structure of the Model selected to implement its Higher-Level Requirements.

► Model-Based Development – A development process in which the primary software artifacts are represented by models from which other artifacts, for example source code, are generated.

► Model Category – The classification of a model according to the life cycle data the model represents.
  ▪ Example: Software High-Level Requirements Model, Software Architecture Model.
New Modeling Glossary Terms

► Software Architecture Model – A Model representing part or all of the Software Architecture.
  ▪ Example: UML Class Model.

► Software High-Level Requirements Model – A Model representing part or all of the Software High-Level Requirements.

► Software Low-Level Requirements Model – A Model representing part or all of the Software Low-Level Requirements.

► Software Verification Model – A Model representing Software Verification Cases, Procedures and Results or components of verification tools.
More Glossary Terms Introduced

► Model Simulator – A device, computer program or system which enables the execution of a model to demonstrate its behavior in support of verification and/or validation.
  - *Note:* This simulator may be executing code that is not representative of the target code.

► Model Transformation – The Process of transforming a Model into another Model
  - *Example:* Transformation of a dataflow Model designed in one modeling tool into a dataflow Model designed in another modeling tool.

► Model View – A subset or all of a Model for a specific purpose.
  - *Example:* A View of an Entity Relationship Model would select a set of entities and their relationships to other selected entities for the specific purpose.

► Model Execution – The process of exercising a Model on a Model Simulator to verify that it satisfies specified requirements and to detect errors.
Auto-coding Glossary Terms

► **Auto-Code Generator** – A Software Tool that transforms a Model into Code.
  
  *Note:* Automated code generation may take place in different software processes. If automated code generation is applied in the software coding process then the process activities defined in DO-178C/ED-12C section 5.3 are automated. If automated code generation is applied in Processes other than the software coding process, some or all of the model-based Process inputs may be used to generate Simulation Code. For example, a High-Level Requirements Model could be subject to an automated code generation resulting in Simulation Code for Execution of the Model.

► **Auto-Model Transformer** – A Software Tool implementing Model Transformation.

► **Auto-Model Transformer / Auto-Code Generator Setup** – The activity/result of setting up an Auto-Model Transformer or Auto-Code Generator for the intended purpose.
  
  *Note 1:* The term is used to synonymously identify the activity and the result.
  
  *Note 2:* In contrary to the setup of other tools within the software development/verification environment the Auto-Model Transformer /Auto-Code Generator Setup is an activity which is part of the software process automated by the application of Auto-Model Transformer /Auto-Code Generator, as setup decisions directly correspond to decisions made by a software developer when performing the software process manually.
FAQ Highlights

► FAQ: What should be the basis for test cases for executable object code for which the low level requirements are in a model?

► Answer:

- In this case, models are expressions of how a given set of software higher level requirements should be implemented. Therefore, test cases for the executable object code generated from a model should not only be based on the model but also on the higher level requirements for the model.
FAQ Highlights

FAQ: Is it required to test the executable object code generated from the model on the target?

Answer:

- If the tool chain that generates executable object code from the model is not qualified for the target computer, then all testing must be performed on the target or a target emulator as specified in DO-178C.
- If the tool chain that generates or verifies executable object code from the model is qualified for the target computer, then at least the following testing must be performed on the target or a target emulator as specified in DO-178C:
  - *Requirements-based hardware/software integration testing.*
FAQ: What is the benefit of performing coverage analysis on a model?

Answer:
- The benefit of performing coverage analysis with the associated coverage analysis resolution on a model is to supplement the requirements-based coverage analysis.
- Examples of supplementary analysis, with corresponding verification objectives are as follows:
  - Provide evidence of the thoroughness of model’s requirements-based testing
  - Provide evidence that the complete model structure was verified
Wrap Up

Where We Are Going
Fall 2008

► The Supplement Format finalizing for SG-4
► Chicago meeting scheduled for September to finalize draft based on IP-440
► Changes to document format may affect our plans (EARLY October chair’s meeting)
► November Plenary meeting will present latest draft
Thank You

Questions???
Acknowledgements

This presentation contains the work of the entire SG4 committee, thank you to all that contributed.

SG4 Website
http://forum.pr.erau.edu/SCAS/

SG4: Model Based Design and Verification
SG4 Model Based Design and Verification: Working Materials and Discussions

Contact Information
MARK.LILLIS@HS.UTC.COM