DO-178C Airborne Software Regulation Standard
What’s New Compared to DO-178B?
How to Comply with DO-178C?
Agenda

• Introduction to Esterel Technologies and SCADE
• Why DO-178C?
• Tool Qualification
• Model-Based Development
• SCADE and DO-178C
• Summary
• Questions & Answers
Introduction to Esterel Technologies and SCADE
Provide critical system and software developers with model-based development solutions that reduce cost, risk and time-to-certification
SCADE Product Family

Model-Based System Engineering

SCADE SYSTEM

Control Software Design

SCADE SUITE

Prototyping, Design, Verification, Qualified Code Generation

System Architecture, System Verification

HMI Software Design

SCADE DISPLAY

Prototyping, Design, Verification, Qualified Code Generation

System & Software Lifecycle Mgt

SCADE LIFECYCLE

Certification Plans, Metrics, Requirements, Configuration Management, Documentation Generation

© 2013 ANSYS, Inc.
SCADE Suite is Unique

• SCADE Suite has been developed specifically to address safety-critical applications

• The Scade language was formally defined with key safety objectives, in close connection with aircraft manufacturers and certification authorities

• The SCADE Suite KCG code generator has been qualified by EASA, FAA, Transport Canada, CAAC, etc, as a software development tool according to the DO-178B standard, for numerous aircrafts and engines projects

• SCADE Suite KCG is currently being qualified as a TQL-1 tool for DO-330
Modeling in SCADE Suite
Why DO-178C?
Process Structure of DO-178B
Summary of DO-178B

• DO-178B takes into account the inputs, constraints, requirements from all stakeholders
  – **Consensus** between Airframe manufacturers, Equipment suppliers, and Certification authorities

• DO-178B was written as much as possible as an **objectives-oriented** document
  – Try not to be prescriptive on the means
  – Less sensitive to technology evolution

• 20 years of use did not reveal major safety flaws
Why Changing It?

• **DO-178B was released in 1992**
  - In 1992, Software Engineering was **24 years old**
  - In 2012, Software Engineering is **more than 50% older**

• **New techniques have appeared since 1992, for example**
  - Model-Based Development and Verification (MBDV)
  - Formal Methods (FM)
  - Automatic Code Generation (ACG)

• **The apparent stability of DO-178B is partially an illusion:**
  - CAST (Certification Authorities Software Team) papers
  - IPs (Issue Papers) at FAA
  - CRIIs (Certification Review Items) at EASA
  ... have been accumulating and not always the result of a consensus!
DO-178C Requirements

• Document the intent of DO-178 more consistently

• Do not raise or lower the bar for certification

• Make the newer techniques, such as model-based development and verification, object-oriented technologies, and formal verification easier to apply through technology specific supplements

• Provide a document to better explain when and how to qualify tools (more reliance on qualified tools to guarantee robustness)
The New Documents

- OOT/RT (DO-332)
- Airborne (DO-178C)
- Ground (DO-278A)
- FM (DO-333)
- MBDV (DO-331)
- TOOLS (DO-330)
- FAQ, DP (DO-248C)
Tool Qualification

• Is tool qualification needed?
  – Yes, “when processes of this document (DO-178C) are eliminated, reduced, or automated by the use of a software tool without its output being verified as specified in section 6.0”

• The purpose of the tool qualification process is to obtain confidence in the tool functionality

• The higher the risk of the tool error adversely affecting system safety, the higher the rigor required for tool qualification
Tool Criteria

• There are 3 criteria:
  – Criteria 1 tool
    • A tool whose output is part of the airborne software and thus could insert an error
  – Criteria 2 tool
    • A tool that automates verification process(es) and thus could fail to detect an error, and whose output is used to justify the elimination or reduction of verification process(es) other than that automated by the tool, or development process(es) that could have an impact on the airborne software
  – Criteria 3 tool
    • A tool that, within the scope of its intended use, could fail to detect an error
Tool Qualification Levels

• 5 TQL defined
  - TQL-1 most rigor through TQL-5 least rigor
  - Determination of TQL outside scope of STQC document (it is described in Section 12.2 of Core document)

TQL-1 ≈ DO-178B Level A
TQL-2 ≈ DO-178B Level B
TQL-3 ≈ DO-178B Level C
TQL-4 ≈ DO-178B Level D
TQL-5 ≈ DO-178B Verification Tool
### Assigning the Tool Qualification Level

<table>
<thead>
<tr>
<th>(DO-178C) Software Level</th>
<th>Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>A</td>
<td>TQL-1</td>
</tr>
<tr>
<td>B</td>
<td>TQL-2</td>
</tr>
<tr>
<td>C</td>
<td>TQL-3</td>
</tr>
<tr>
<td>D</td>
<td>TQL-4</td>
</tr>
</tbody>
</table>
Stakeholders

• Tool Developer
  - Responsible for developing, verifying, documenting, and producing the tool
  - Satisfies development objectives for tool

• Tool User
  - Responsible for selecting, using, and qualifying the tool
  - Satisfies installation and use objectives for tool

• These roles were not identified as such in DO-178B
Model-Based Development and Verification
• Purpose of the Supplement:

– Provide guidance when models are used to represent SW life cycle data
Models Express Requirements or Architecture?

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

---

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

**Req_002:** The XX module ....

**Derived Req_003:** ...
Scope of MBDV Supplement

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

• **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** has to be described in **Model Standards**
Model Parent Requirements

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

- **Simulation**: an appropriate means to support model verification
Verification Process

Test (example)

Model = HLR + LLR

Model Parent
Requirements

Executable
Object Code
Model Coverage Analysis: a way to detect unintended functions in a model
SCADE and DO-178C
Safe Design with SCADE Suite

Modularity & Strong Typing

• The “Integrator” SCADE Node below is a functional module:

  o a formal interface:
    node Integrator
    (U: real ;
     TimeCycle: real)
    returns ( Y: real);

  o a set of intermediate variables:
    var
    delta : real;
    last_Y : real;

  o a set of equations:
    delta = u*TimeCycle;
    y = delta+last_Y;
    last_Y = fby(y, 1, 0.0);
Safe Design with SCADE Suite

**Time Operators**

- **pre**: delay one cycle
  \[ y = az^{-1}x \quad \Rightarrow \quad y = a \times pre \ (x) \]

- **- >**: data flow initial value

Translation completed with 0 semantic error(s), 0 semantic warning(s).

**WARNING**: NODE Count, VAR Counter, please verify the initialization of the variable(s).
SCADE LifeCycle Requirements Management Gateway

Integrated Requirements Management and Traceability
• The SCADE LifeCycle Reporter automatically generates the Design Documentation
void Button_ABC_N(inC_Button_ABC_N *inC, outC_Button_ABC_N *outC)
{
    /* ABC_N::Button::SM1::SSM_SM1_dispatch_sel */
    SSM_Button_SM1_ST SSM_SM1_dispatch_sel;
    if (outC->init)
    {
        outC->init = kcg_false;
        SSM_SM1_dispatch_sel = SSM_SM1_Unselected__ABC_N;
    }
    else
    {
        SSM_SM1_dispatch_sel = outC->M_pre_;
    }
    switch (SSM_SM1_dispatch_sel)
    {
        case SSM_SM1_Locked__ABC_N :
            outC->foreground = white_ABC_N;
            outC->background = green_ABC_N;
            if (inC->Unlock)
            {
                outC->M_pre_ = SSM_SM1_Preselected__ABC_N;
            }
            else
            {
                outC->M_pre_ = SSM_SM1_Locked__ABC_N;
            }
            break;
        case SSM_SM1_WaitUnlock__ABC_N :
            outC->foreground = black_ABC_N;
            outC->background = grey_ABC_N;
            if (inC->Unlock)
            {
                outC->M_pre_ = SSM_SM1_Unselected__ABC_N;
            }
            else
            {
                outC->M_pre_ = SSM_SM1_WaitUnlock__ABC_N;
            }
            break;
        [...]
    }
}
SCADE Suite KCG Certification Kit

• The SCADE Suite KCG certification kit provides all the artifacts produced by Esterel Technologies during the development of the tool, and **required by certification authorities in DO-178C** for a software tool qualified at TQL-1 for DO-330:
  o Tool Qualification Plan (TQP)
  o Tool Operational Requirements (LRM and KCG TOR)
  o Tool Requirements (TR)
  o Tool Installation Procedure (TIP)
  o Version Content (VC)
  o Tool Configuration Index (TCI)
  o Tool Accomplishment Summary (TAS)
If you are using models as defined in DO-331, section MB.1.0, as the basis for developing software, you should apply the guidance in DO-331:

(a) You should identify which of the objectives you propose to satisfy using model simulation.

(b) If you propose to use model simulation in combination with reviews and analysis to satisfy the objectives in MB.6.8.1, you should show that the errors detected include all errors that could be detected by reviews and analysis alone.”
Impact of SCADE Suite KCG Qualification

• When a code generator is qualified as a criteria 1 tool (DO-178C; section 12.2)
  - Conformance of the code to the input model is trusted
  - Verification activities related to the coding phase are eliminated.

• The SCADE Suite KCG automatic C code generators is qualifiable as a development tool at DO-178B level A and will be at TQL-1 for DO-330.
Model Test Coverage (MTC)

- Coverage *Analysis* at Model level:
  - Enables **Requirements-based** tests
  - Shows how **thoroughly** the SCADE model has been **tested**
  - Shows the **role of each test case** in covering operator instances of the SCADE model

- Coverage *Resolution* at Model level:
  - Provides **correction or justification for all uncovered features**, and reveals:
    - Shortcomings in Requirements-based test procedures
    - Inadequacies in System Requirements
    - “Dead” SW Requirements
    - “Deactivated” SW Requirements
MB.B.11 FAQ #11: May the applicant use the model coverage analysis activity to achieve the structural coverage analysis objectives?

These conditions should include at least the following:

- Model coverage analysis criteria hold the same properties as the applicable structural code coverage analysis criteria hold for the level of the software being developed, for example, MC/DC coverage for the level A.

- Qualification of the code generation tool chain with respect to objectives for which certification credit is sought (in particular Annex MB.A Table MB.A-7 (Annex MB.C Table MB.C-7 for DO-278A users)) should preserve the applicable coverage criteria.

- Any libraries used by code generated from the Design Model are verified according to DO-178C/DO-278A section 6, including structural code coverage activities in accordance with the required software level.
Compiler Verification Kit (CVK)  
Object Code Verification

Customer’s Development Project (SCADE Suite)

KCG → C code → Compiler → Object code

Customer’s Environment Certification

Integrate KCG in the Certification Process

SCADE Suite KCG Certification Kit

Verify Compiler in SCADE Suite environment

Compiler Verification Kit
SCADE LifeCycle Qualified Testing Environment (QTE)

- **Test Execution Engine (Q)**
- **MTC (Q)**
- **Target Test Harness Generator (Q)**
- **Qualified Test Environment**

**HOST**

- Conformity Report
- Model Coverage
- Target Tests

**TARGET**

- **Test Cases**
- **SCADE Model**
- **Target Tests**

Target tool suite (LDRA, VectorCAST, RTRT, ...)

Results
### SCADE Qualified Tools

<table>
<thead>
<tr>
<th>(DO-178C) Software Level</th>
<th>Criteria 1</th>
<th>Criteria 2</th>
<th>Criteria 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>TQL-1</td>
<td>TQL-4</td>
<td>TQL-5</td>
</tr>
<tr>
<td>B</td>
<td>TQL-2</td>
<td>TQL-4</td>
<td>TQL-5</td>
</tr>
<tr>
<td>C</td>
<td>TQL-3</td>
<td>TQL-5</td>
<td>TQL-5</td>
</tr>
<tr>
<td>D</td>
<td>TQL-4</td>
<td>TQL-5</td>
<td>TQL-5</td>
</tr>
</tbody>
</table>
Summary
Summary

• DO-178C and its supplements (MBDV, OO/RT, FM) make it easier to apply newer techniques

• DO-330 is a specific and independent document to handle tool qualification

• SCADE provides a number of appropriate (qualifiable) tools to apply DO-178C efficiently
SCADE DO-178 Methodology Handbooks

• Contents:
  - Development and verification steps
    • Model-based development with SCADE
    • Simulation and Model Test Coverage
    • Formal verification
    • Automatic code generation with KCG
    • C compiler verification activities
  - Set of guidelines for developing efficient models, generating efficient code, etc.
  - Two versions available for Display centric and Control centric applications

Download the handbook from www.esterel-technologies.com
Bernard Dion
Chief Technical Officer

White Papers, Datasheets, Methodology Handbooks,
Free Software, Webinars
www.esterel-technologies.com

Contact Us
salesscade@esterel-technologies.com
Europe (France) +33 1 30 68 61 60
U.S. +1 724-514-2997