SAVOIR AUTOCODE

space avionics open interface architecture

AUTOCODE Working Group

Automatic Code Generation for AOCS/GNC Flight SW

[ACG-WG] Index

- Working Group key driving factors
- Working Group objectives
- *Modelling Guidelines for Automatic Code generation Handbook*
  - Draft ToC
  - Content and Examples
- Working group schedule of activities
- Wrap Up
- Q&A

[ACG-WG] Key driving factors

- **“autocoding process”** has been used on several project in last years and it became the **baseline for future projects**
  - The autocoding process needs to be identified in all its steps and fitted in the existing SW development process
- **Model Based Design with MATLAB/Simulink + Embedded Coder** using restricted set of Simulink blocks (predictable behavior of generated code & clarity of implementation)
  - MATLAB/Simulink modelling standard needs to be identified and implemented through checking tools
- The **Verification & Validation process** requires a lot of **manual steps** to be performed in combination with autocoding.
  - Identification of the V&V steps that can be automated (testing, reporting)
  - AOCS and SW expertise shall work closely together

[ACS-WG] Objectives

- The Working Group objectives are:
  1. to provide **MODELLING RULES AND GUIDELINES** to develop AOCS & GNC models using MATLAB/Simulink in order to ensure the generated code being functionally correct, compliant with the existing standards as well as readable, reusable and maintainable.
  2. to provide **DVV RULES AND GUIDELINES** to ensure the DVV process for the AOCS automatic generated code fitting the existing development, verification and validation process (i.e. being compliant to the applicable standard) considering it as part of the FSW product
- The WG will prepare the **HANDBOOK**:

Guidelines for the Automatic Code Generation for AOCS/GNC Flight SW

- As per SAG recommendation, the handbook could also address the architecture (interface between AOCS and the rest of the software), the EDS to interface the equipment and the redundancy.

Autocoding Handbook *table of content*

[ACG HB ToC] Process description

4. INTRODUCTION to AOCS FSW development PROCESS

- **a. The classical process: Manual Coding**
- **b. Introduction to Autocoding**
- **c. Comparison and key differences**

The classical process: Manual Coding

Description of the model development steps, overview the different tools available, list of the applicable standards, the implications and roles of teams and models during development.

Introduction to Autocoding

Description of the model development steps, overview of the different tools available, list of the applicable standards, the implications and roles of teams and models during development.

Comparison and key differences

Differences wrt Autocoding in terms of models final scope, in terms of documentation and deliverables, in terms of verification and validation steps and expertise.

Autocoding Handbook *table of content*

[ACG HB ToC] Modelling Guidelines

5. MODELLING GUIDELINES FOR CODE GENERATION

- **a. Modelling guidelines**
  - **i. General modelling guidelines**
  - **ii. Modelling with Matlab**
  - **iii. Modelling with Simulink**
  - **iv. Modelling with Stateflow**
- b. Code generation guidelines
  - i. Coder configuration settings
  - ii. Generated Code structure
  - iii. Reuse of legacy code

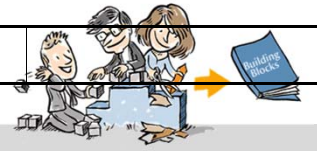
This chapter will go through the AOCS model development and verification for automatic flight code generation and for each step the guidelines are defined.

The guidelines are categorized per applicability:

- Mandatory
- Strongly recommended
- Recommended

The applicability describes the importance of the guideline and determines the consequences of violations.

Mandatory	Strongly Recommended	Recommended
Definition		
• Guidelines that are absolutely essential. • Guidelines where 100% compliance shall be required.	• Guidelines that are agreed upon to be a good practice, but use of legacy models preclude from being compliant at 100% • Models should conform to these guidelines to the greatest extent possible; however 100% compliance is not required	• Guidelines that are recommended to improve the appearance of the model diagram, but are not critical to running the model • Guidelines where conformance is preferred, but not required
Consequences – If the guideline is violated		
• Essential items are missing • The model might not work properly	• The quality and the appearance deteriorates • There may be an adverse effect on maintainability, portability, and reusability	• The appearance will not conform with other projects
Waiver Policy – If the guideline is intentionally ignored		
• The reasons must be justified through RfW	• The reasons must be documented	



[ACG HB ToC] Modelling Guidelines

General Modelling Guidelines	Approaches everything that has to do with the environment, in which the user models the system
Simulink	Rules regarding the Simulink blocks

5. MODELLING GUIDELINES FOR CODE GENERATION

- **a. Modelling guidelines**
  - **i. General modelling guidelines**
  - **ii. Modelling with Matlab**
  - **iii. Modelling with Simulink**
  - **iv. Modelling with Stateflow**
- b. Code generation guidelines
  - i. Coder configuration settings
  - ii. Generated Code structure
  - iii. Reuse of legacy code

ID	ESA-SL-0[illegible]
Title	Block[illegible] recommended for C/C++ code production
Priority	Mandatory
Description	The model should not have any kind of blocks that are not suitable for code production. The list of such blocks can be found is in annex **Error! Reference source not found.**. Automatic Testing: mathworks.do178.PCGSupport mathworks.maab.jm_0001 mathworks.maab.hd_0001
Rationale	Using blocks compatible with code generation is essential for the process.

EXAMPLE

[ACG HB ToC] Code Generation Settings

5. MODELLING GUIDELINES FOR CODE GENERATION

a. Modelling guidelines

i. General modelling guidelines

ii. Modelling with Matlab

iii. Modelling with Simulink

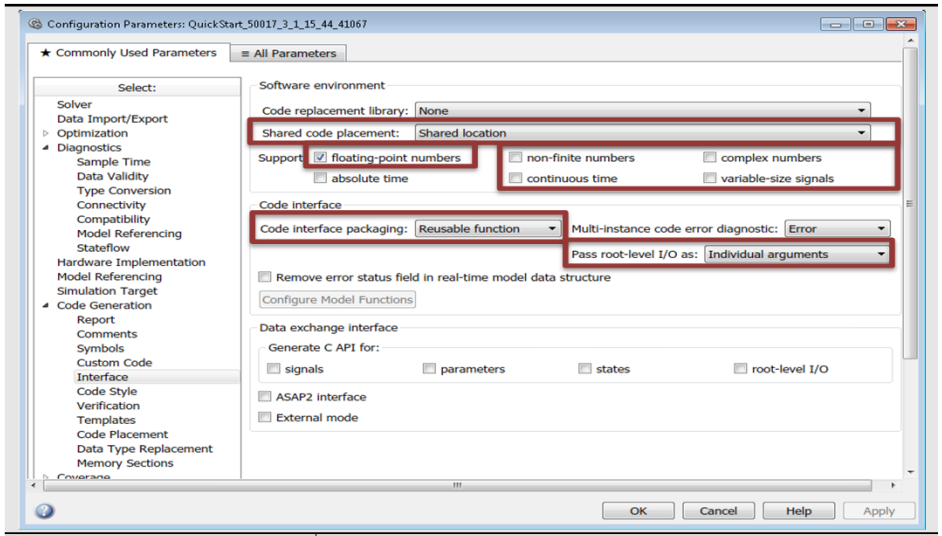
iv. Modelling with Stateflow

b. Code generation guidelines

i. Coder configuration settings

ii. Generated Code structure

iii. Reuse of legacy code



Parameter	Value	Description
Shared code placement:	**Shared location**	**Directs code for utilities to be placed within the slprj folder in your working folder.**
floating-point numbers	**Selected**	**If the processor supports floating point numbers**
non-finite numbers	**Unselected**	**Not used, so it should not be supported**
continuous time	**Unselected**	**The system is discrete.**
complex numbers	**Unselected**	**Not used, so it should not be supported**
variable-size signals	**Unselected**	**Not used, so it should not be supported**
Code interface packaging	**Reusable function**	**Produces more efficient code.**
Pass root-level I/O as	**Individual arguments**	**Increases readability, and an extra structure is not created.**

[ACG HB ToC] Use of Legacy Code

5. MODELLING GUIDELINES FOR CODE GENERATION

a. Modelling guidelines

i. General modelling guidelines

ii. Modelling with Matlab

iii. Modelling with Simulink

iv. Modelling with Stateflow

b. Code generation guidelines

i. Coder configuration settings

ii. Generated Code structure

iii. Reuse of legacy code

Introducing Legacy Code

Legacy code as Simulink blocks

Define the legacy code function as Simulink S-function, compile it and generate the code.

→ This section will define the rules for coding and impact on the FSW by reusing code through S-Function

Legacy code as Library function

If programming in Embedded Matlab (Matlab Function block) a C-Function can be called

→ This section will define the rules for coding and impact on the FSW by reusing code through 'ceval' Matlab function

Autocoding Handbook *table of content*

[ACG HB ToC] V&V Guidelines

Automatic model and code verification tools

The use of automatic tools for the verification process is analyzed and for each tool description and examples on how to configure it and what are the advantages and limitations.

- Reference cases definition (test harness) and execution for AOCS Performance verification
- Model and Code Standard analysis
- Model and Code Complexity analysis
- Model and Code coverage analysis
- Model and Code unit testing

Code Validation process MIL/SIL/PIL

Test harnesses shall be used to validate the Code with respect to technical specification and requirements baseline.

ECSS TRACEABILITY

Clause / subclause	Covered by Rule	Comment

Clause / subclause	Rule	Comment
Clause 6.2.8 Automatic code generation		

Autocoding Handbook *table of content*

[ACG HB ToC] Reporting

This section provides recommendation and highlights on how to set the tools to generate appropriate documentation for each development and verification phase in order to ensure visibility and traceability to requirements.

MATLAB Report Generator is able to create Microsoft Word, Microsoft PowerPoint, HTML, or PDF reports that present results from your MATLAB programs and applications.

The prebuild, customizable templates can be used (and customized in order to comply with any company standard template), or a new template can be created from scratch.

8. GENERATION OF REPORTING

- **a. Design reports**
- **b. Test plans**
- **c. Test reports**
- **d. Verification control documents**

Model Generation Advisor Report
Based on template:
This document goes through all the modelling established rules. The Model Advisor Report contains each modelling guideline with: • the result of the check • the description of each test • if the result is a fail, it describes where the error is and the user justification for it The report can be customizable to: Include the manual checks ant the user input to them. To control the order of the displayed checks in the document; additional user remarks at will.

Autocoding Handbook *table of content*

[ACG WG] Schedule of activities

- **Nomination of WG members (AOCS, GNC, SW, SW PA)**
  - All interested disciplines has been involved
- **WG objectives definition**
  - Refer to previous slides
- **Collection of supporting material from previous WG's and studies**
  - Final reports collected from previous studies
  - Projects experience as presented at ADCSS 2016

- **Schedule for the Handbook production (each task is 6 months)**

FEB 2018

- Phase I: **ESA internal working group** between **SW / SW PA / AOCS / GNC experts** to agree on the scope, table of contents and first draft of the handbook contents

[ACG WG] Schedule of activities

- **Nomination of WG members (AOCS, GNC, SW, SW PA)**
  - All interested disciplines has been involved
- **WG objectives definition**
  - Refer to previous slides
- **Collection of supporting material from previous WG's and studies**
  - Final reports collected from previous studies
  - Projects experience as presented at ADCSS 2016

- **Schedule for the Handbook production**

  - Phase I: **ESA internal working group** between **SW/SW PA/AOCS/GNC experts** to agree on the scope, table of contents and first draft of the handbook contents

AUG 2018

  - Phase II: **Extended working group** (including **industry and other agencies/institutions**) to prepare a second draft, harmonized with external stakeholders

[ACG WG] Schedule of activities

- **Nomination of WG members (AOCS, GNC, SW, SW PA)**
  - All interested disciplines has been involved
- **WG objectives definition**
  - Refer to previous slides
- **Collection of supporting material from previous WG's and studies**
  - Final reports collected from previous studies
  - Projects experience as presented at ADCSS 2016

- **Schedule for the Handbook production**

  - Phase I: **ESA internal working group** between **SW / SW PA / AOCS / GNC experts** to agree on the scope, table of contents and first draft of the handbook contents

  - Phase II: **Extended working group** (including **industry and other agencies / institutions**) to prepare a second draft, harmonized with external stakeholders

MAR 2019

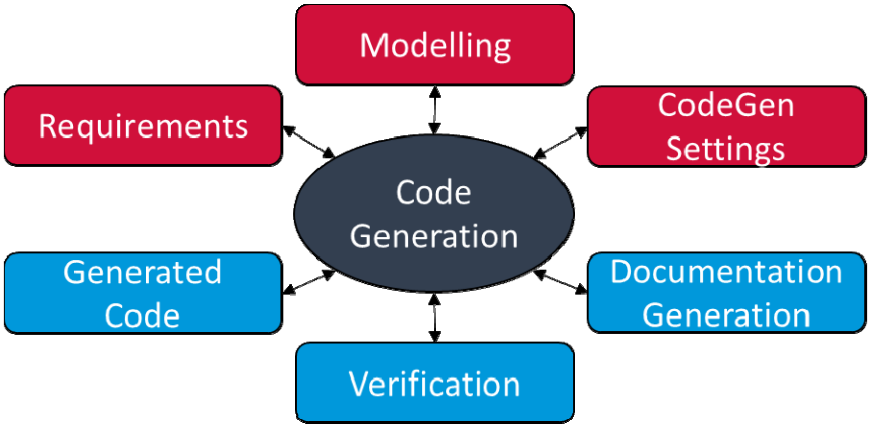
  - Phase III: **Public review** for final rework up to the formal release in the frame of SAVOIR

Autocoding Working Group Wrap-Up

The objectives of the ACG-WG: provide **modelling rules and guidelines** to ensure good auto-code quality**,** provide **DVV rules and guidelines** to comply and fit within existing standard FSW DVV process

The HB will cover the following main process aspects:

- Tools, Development Environment
- Code Generation Process / Settings
- Modelling Guidelines for Code Generation
- Verification and Validation Guidelines
- Report Generation



The schedule foresees the following steps:

- **ESA internal [02/2018]**
- **Extended working group** (including industry and other agencies/institutions) **[08/2018]**
- **Public review** up to the **release [03/2019]**

Questions

SAVOIR coordinator:

Jean-Loup.Terraillon@esa.int

ACG WG coordin

davide.odder