AUTOCODING WORKING GROUP
Automatic Code Generation for AOCS Flight SW

Davide Oddenino
15th ADCSS, 16/11/2021
Outline of the presentation

- Background – Autocoding Working Group
- Vol I - Process: Autocoding generation process
- Vol. II - Technology: Modelling Guidelines
- Conclusions
Outline of the presentation

- **Background – Autocoding Working Group**
- **Vol I**  - Process: Autocoding generation process
- **Vol. II** - Technology: Modelling Guidelines
- **Conclusions**
Autocoding Working Group - Objectives

The purpose of the Autocoding Working Group is to **prepare the ESA Autocoding Handbook** to define for the **AOCS Flight SW Cat. B**:

- the **technology** (modelling guidelines, impact of the code generator, etc…)
- the **process** (GNC algorithm development process and application software process covering all the lifecycle up to V&V)

To be used when creating models and generating flight code with the objective to ensure the generated code is **correct, reliable, readable, sharable/reusable** and **maintainable**.

*The intended use of the guidelines are the following ones:*
- **Use in support to projects** providing an harmonized ESA position across the Agency.
- **Use in R&D technology** activities.
- **Promotion of the use of this type of methodology across the phases of a development.**
- **Contribution to the assessment of the quality** of the final software product
SAVOIR Autocoding Working Group

I. Vol I development and verification process:
   • Verification & Validation process review wrt manual coding
   • Automatic steps and available tools (possible improvements, customization) guidelines
   • Deliverables (comparison SW dev flow)
   • Test reporting guidelines (templates, content description)

   Compliance with existing standards:
   • Cross check/mapping of ECSS requirements to the Autocoding V&V process
   • Comparison with Autocoding processes as proposed/implemented on different projects

II. Vol II Modelling guidelines (AOCS modelling):
   • Define modelling guidelines
   • Configuration of code generation toolboxes
   • Classification of guidelines
Outline of the presentation

• Background – Autocoding Working Group

• **Vol I – Process: Autocoding generation process**

• Vol. II - Technology: Modelling Guidelines

• Conclusions
Process - Code generation development & Verification
Autocoding process overview
Development tasks – AOCS

AOCS SW Development Process with Autocoding

1. AOCS SW Requirements-Definition
   - AOCS SW Requirements
   - AOCS Requirements

2. AOCS Algorithm Development
   - AOCS SW Requirements
   - AOCS Algorithm Development

3. AOCS SW Development
   - AOCS SW Development
   - AOCS SW Validation

4. AOCS SW Verification
   - AOCS SW Verification
   - AOCS SW Verification

5. AOCS SW Integration
   - AOCS SW Integration
   - AOCS SW Integration

6. AOCS SW Testing
   - AOCS SW Testing
   - AOCS SW Testing

7. AOCS SW System Functional Verification

8. AOCS SW System Verification

9. AOCS SW System Functional Verification

AOCS Analysis Tools

- FB5 Model in the Loop - MIL
- FB5 Software in the Loop - SIL
- SW Testing Tools (SIL) SW Verification Facility - SVF
- MWL Avionics Test Bench - ATB
Development tasks - SW

AOCS SW Development Process with Autocoding

AOCS Requirements Definition → ADC Algorithm development → ADC Code Generation and ADC Performance Validation → ADC Code Verification and Integration → System Functional Verification

1. AOCS Requirements
2. ADCS Algorithm Development (ADC Model)
3. ADCS Model Unit Testing (UT)
4. ADCS Model Verification (FULL AC campaign - ADC Model)
5. ADCS Data Generation
6. ADCS Data Verification
7. SW Integration and Testing
8. System Integration and Testing
9. SW Qualification

AOCS Analysis Tools
- F5 Model in the Loop - MIL
- SW Testing Tools (SW) - Verification Facility - SVF
- MWL Avionics Test Bench - ATB
Verification facilities
**Requirements definition**

**A. From system level to AOCS Subsystem:**
- Derivation of requirements: functional, performance, FDIR
- Justification of AOCS design architecture

**B. From system level to SW Subsystem:**
- Derivation of SW requirements (non AOCS)
- Definition of AOCS SW Interface Requirements Document

→ AOCS team shall start modelling following AOCS SW IRD
A. AOCS performance verification MIL
- AOCS model developed following Autocoding HB guidelines
- AOCS performance (pointing, stability, sensitivity...) verified on MIL wrt AOCS requirements baseline
- Justification of AOCS design

B. Code generation:
- Code generation process configured as per HB
- Definition of AOCS Performance verification reference test cases (subset from MC campaign)
- Comparison of results MIL vs SIL → equivalence expected at last digital bit

→ The Proof of Equivalence (PoE) is mandatory to confirm performance verification on Model covers the verification of Code
→ The PoE can be achieved with accurate modelling and use of mathematical library

→ If PoE at last digit is not achieved, option 2 (see next slide shall be used)
AOCS Performance Verification – Option2: SIL

A. AOCS development MIL
• AOCS model developed following Autocoding HB guidelines
• Justification of AOCS design

B. Code generation:
• Code generation process configured as per HB

C. AOCS Performance verification SIL:
• Verification of AOCS performance (pointing, stability, etc..) running full MC campaign

→ The Code (final product) is used to verify AOCS Performance, no need of equivalence vs Model
→ Any iteration shall not allow Code modifications (changes through Model and Code generation)
→ Synchronization with reviews to be considered
The developed algorithms are subjected to Unit Testing where they are checked against modelling standard guidelines (generation of C-code) and open loop tests performed to be used as reference cases. Coverage tests are also performed to ensure sufficient model coverage with selected test harness. It needs to be reminded that model coverage does not imply coverage of the generated code. Note: the AOCS Models are used as Technical Specification (TS) for the automatic SW code generation, in the sense that SW behaviour will be validated against the Model behaviour.
SW Unit Testing

The generated and verified Code is delivered to the SW team to undergo SW Unit Tests and SW performance tests as per standard SW development plan and the integrated in the On Board SW (OBSW) for further qualification and acceptance testing before final delivery to system for functional verification (as per standard process).

It is noted that unit test a SW level have a different scope than unit test at model level (3). The UT at model level are intended to address modelling aspect while the one at SW level are intended to address code coverage and robustness approach as per E40C 5.5.3.2. Between the two set of UT there is a certain level of overlap and the scope of the SW UT can be reduced if it can be shown that certain aspects are addressed in the model UT.

It is also noted that current process allocate the unit testing at SW level to the SW team, but depending on team expertise and tool chain it could also be conceivable to allocate this activity to the AOCS team.
Delivery of the generated code

The generated and verified Code is delivered to the SW team to undergo the SW Unit Tests and SW performance tests as per standard SW development plan and the integrated in the On Board SW (OBSW) for further qualification and acceptance testing before final delivery to system for functional verification (as per standard process).

It is noted that unit test a SW level have a different scope than unit test at model level (9). The UT at model level are intended to address modelling aspect while the one at SW level are intended to address code coverage and robustness approach as per E40C 5.5.3.2. Between the two set of UT there is a certain level of overlap and the scope of the SW UT can be reduced if it can be shown that certain aspects are addressed in the model UT.

It is also noted that current process allocate the unit testing at SW level to the SW team, but depending on team expertise and tool chain it could also be conceivable to allocate this activity to the AOCS team.
Outline of the presentation

• Background – Autocoding Working Group

• Vol I  - Process: Autocoding generation process

• **Vol. II - Technology: Modelling Guidelines**

• Conclusions
Technology: modelling guidelines, code generation

Guidelines SAVOIR-01 SAVOIR-02

ECSS E-40 Q-80 compliant
Modelling Guidelines objectives

A. Modelling guidelines scope
• Review the guidelines category, mandatory only when no alternative exists
• Express guidelines in terms of objectives, without providing implementation details
• Identify guidelines linked to toolbox used and those independent

C. Coder Configuration settings
• The code configuration settings shall be defined and applied at each generation to avoid differences

D. Configuration control
• The model versioning shall be kept in configuration control. Several methods exist to trace changes into the model (model block, each subsystem, etc.)

Each guideline will be composed by the following fields:

Guideline ID:
• The ID is a unique reference for each guideline (between brackets reference to existing guideline from where it has been derived)

Guideline Title:
• The title is a short, but unique description of the guidelines area of application (e.g. ‘length of names’).
• The title is used for the Prerequisites field and for custom checker-tools.
• The title text should appear with a hyperlink that links to the guideline.

Priority:
• Mandatory / Strongly recommended / Recommended

Description:
• This field contains a detailed description of the guideline.
• If needed, images and tables are included.

Rationale:
• Reasons the rule should be applied
• Mapping to improved quality index (e.g. readability, ..)
• Consequence/disadvantage in case the guideline is not applied
<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAVOIR-SE-050 (ID_ref:AUTO-MCR-070)</td>
<td>Model reference blocks shall be used at the unit test boundaries of model components with explicitly specified interfaces for data type and dimension. Model reference with explicitly specified interfaces should also be used whenever a component (and its generated code) should be functionally independent of the parent model. Changes made to the SIMULINK model within the model reference are guaranteed not to functionally change the generated code outside of the model reference. Changes made to the SIMULINK model outside of the model reference are guaranteed not to functionally change the generated code inside of the model reference. This guarantee cannot be asserted for other modelling components (including library blocks), because of the way Embedded Coder attempts to optimize generated code within a model.</td>
</tr>
</tbody>
</table>
Modelling guidelines classification

<table>
<thead>
<tr>
<th>Classification</th>
<th>Requirement</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct</td>
<td>The code generation process shall work properly and free from errors, generated code shall correspond to the model.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>Reliable</td>
<td>The generated code shall be fully equivalent to the model to ensure verification is valid</td>
<td>Strongly Recommended</td>
</tr>
<tr>
<td>Readable, Reusable, Sharable</td>
<td>Despite no manual changes to the generated code are foreseen, the readability shall be kept for code inspection, debugging, etc... The possibility to reuse or share generated code (industrial organizations) shall be foreseen by code readability</td>
<td>Recommended</td>
</tr>
<tr>
<td>Maintainability</td>
<td>The maintenance of auto-generated code is meant to be achieved without manual intervention, but acting on the model and re-generating the code. This possibility shall be maximized by modular architecture and traceability</td>
<td>Recommended</td>
</tr>
</tbody>
</table>
Outline of the presentation

• Background – Autocoding Working Group

• Vol I - Process: Autocoding generation process

• Vol. II - Technology: Modelling Guidelines

• Conclusions
Guidelines for the Automatic Code Generation for AOCS/GNC Flight SW Handbook: Volume 1 - General concepts


The handbook is available in the ESA Space Software Repository
https://essr.esa.int
Future European Projects

Thank you in different languages:
- danke
- спасибо
- dank je
- gracias
- obrigado
- merci
- teşekkür ederim
- go raibh maith agat
- grazie
- grazie
- merci
- thank you