Presented by

Jean-Charles DALBIN
Airbus Operations SAS
&
Laurent DUFFAU
Airbus Operations SAS

GENE-AUTO
Status of new Airbus case Studies

Agenda

1/ OBSYS (EADS research demonstrator)

- Primary Flight Control case study :
  - Part of laws function (using vector based controller)
  - Part of logics function (using state machines)

2/ AIRBUS internal research activities :

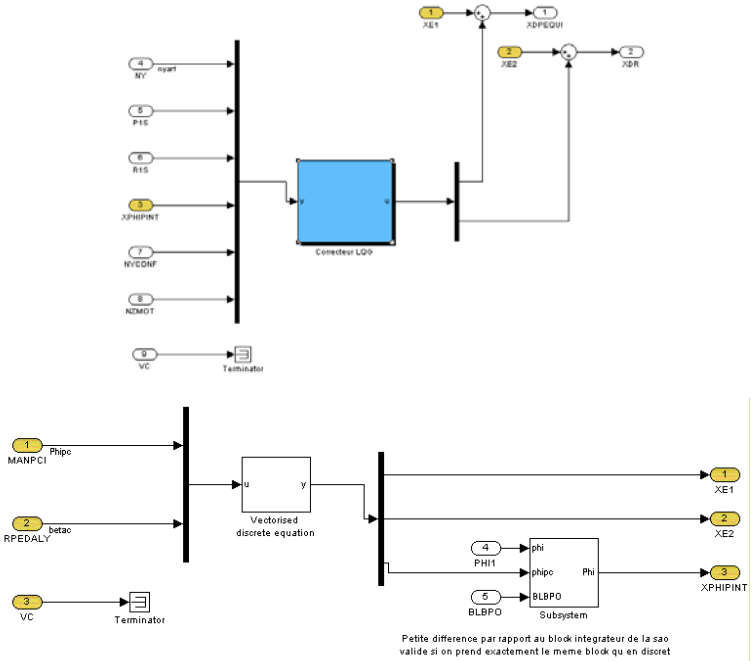
- “Real life” Avionics case study **:**
  - Weight and Balance Backup Computation Function (Experimentation on a complete Simulink specification (equiv to 100 SCADE nodes)
- Gene-auto evaluation to produce AP2633 code for simulation

3/ Feedback on Code Customization

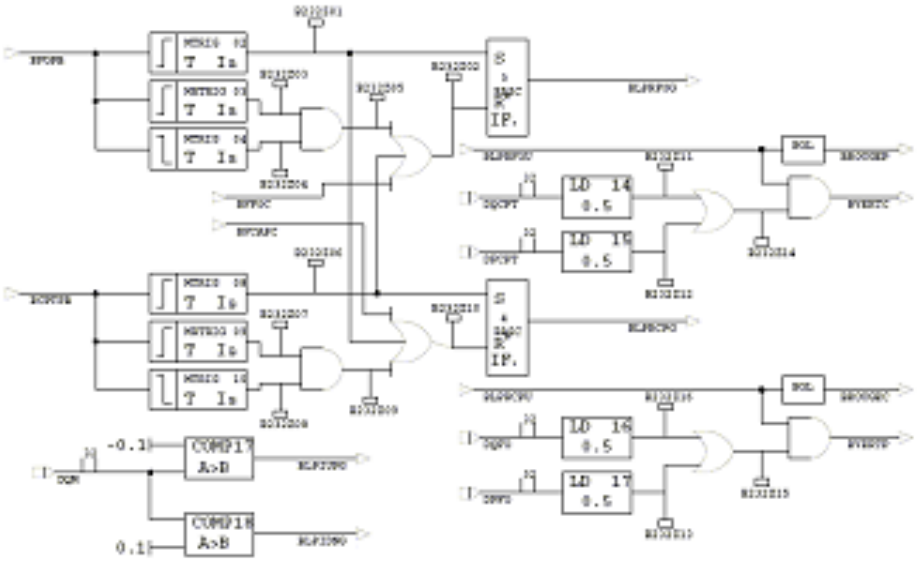
4/ Global Status

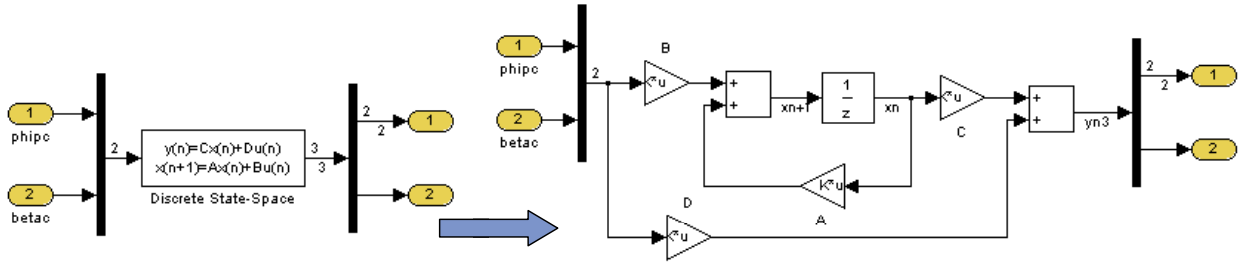
OBSYS : Flight control laws function - Overview

- **Aim :**
  - **Use a vector based approach for the laws function to enhance the capability of the design**
  - **Test the capability to generate certified code with the discrete vector based model, and compare it to the one from the current AIRBUS tools**
- **Context :**
  - **Part of flight control laws function** (**F**light **C**ontrol **P**rimary **C**omputer A340-600)
  - Target cpu Intel 486
  - Multi-rate : 10 ms & 40 ms
  - Simulink model : Vectors, 30 blocks, 2 levels of hierarchy



OBSYS : Flight control laws function - Re-Design





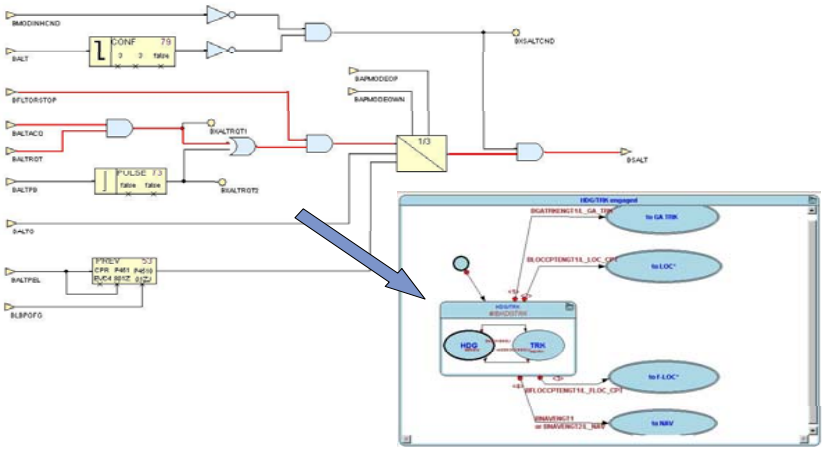
OBSYS : Flight control laws function - Status

	Generation (Gene-Auto without optimization tool) with Geneauto symbols	**Generation (Gene-Auto without optimization tool) with AIRBUS backends**
Design	Re-design part of laws function (25 SAO sheets) to a vector based Simulink (one model for C1, one model for C3)	
Code Generation	OK	39 backends have been developed to optimize symbols
Compilation	OK + Link Gene-Auto source code with the other part of the design (SAO sheets)	
Integration on target	Operation overflow detected after 9 steps of computation (cycles) => Investigation on going	
Functional verification on target	TO DO	TO DO
CPU Performance analysis	TO DO	TO DO

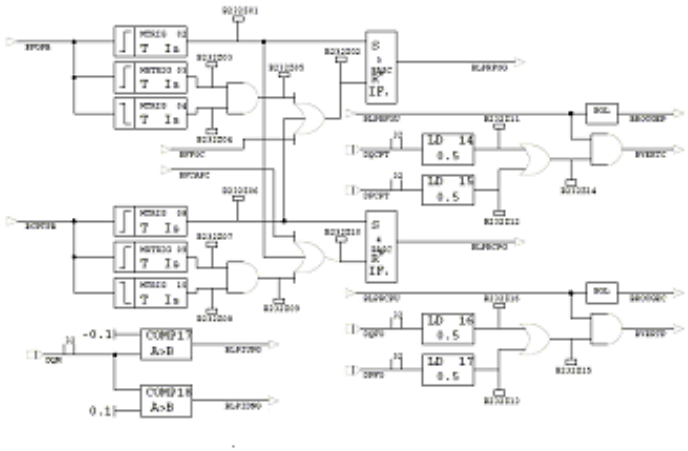
OBSYS : Flight control logics function - Overview

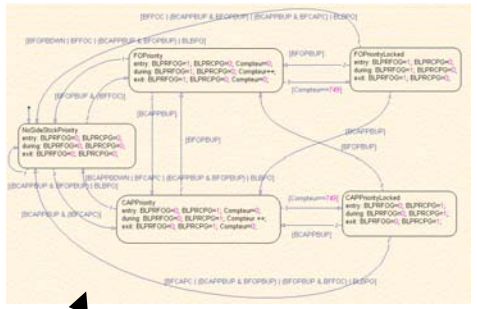
- **Aim :**
  - **Use state machines for the mode computation to split logics to enhance the capability of the design**
  - **Test the capability to generate certified code with model using state machine, and compare it to the one from the current AIRBUS tools**

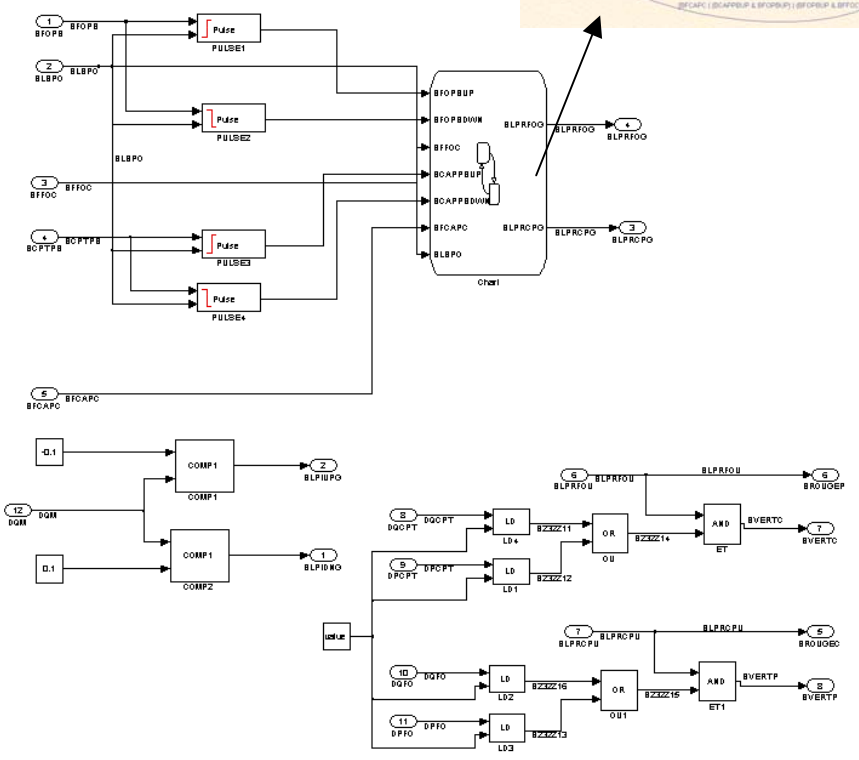
- **Context :**
  - **Part of flight control logics function** (**F**light **C**ontrol **P**rimary **C**omputer A340-600)
  - Target cpu Intel 486
  - Mono-rate : 40 ms
  - Simulink & Stateflow model : 75 blocks, 3 levels of hierarchy, 5 states



OBSYS : Flight control logics function - Re-Design







Symbols Mrtrig & Bascr have been replaced by symbols Pulse + 1 automaton with 5 states

OBSYS : Flight control logics function - Status

	Generation (Gene-Auto without optimization tool) with AIRBUS backends
Design	Re-design a part of logics function with state machine (Mrtrig & Bascr have been replaced by Pulse + 1 state machine with 5 states)
Code Generation	OK
Compilation	OK
Integration on target	OK
Functional verification on target	OK (same functional behaviour as SAO model reference)
CPU Performance analysis	=>Cpu time consumption measured : multiplied by 2 => Memory consumption measured : **TBC** => Waiting for i486 model from Ait for WCET computation and analysis

Agenda

1/ OBSYS (EADS research demonstrator)

- Primary Flight Control case study :
  - Part of laws function (using vector based controller)
  - Part of logics function (using state machines)

2/ AIRBUS internal research activities :

- “Real life” Avionics case study **:**
  - Weight and Balance Backup Computation Function (Experimentation on a complete Simulink specification (equiv to 100 SCADE nodes)
- Gene-auto evaluation to produce AP2633 code for simulation

3/ Feedback on Code Customization

4/ Global Status

“Real-life” Avionics function Case Study

- **Aim :**
  - **Evaluate the complete software application process using Simulink/Gene-Auto toolchain on a selected avionics function**
  - **Test the capability to generate certified code on a data flow Simulink design, and compare it to the one from the current SCADE tools used in AIRBUS**

- **Context :**
  - **Weight & Balance Backup Computation function** (CPIOM computer **A380**)
  - Target power PC 755
  - Mono-rate : 40 ms
  - Simulink model : 107 models, 197 blocks, 2 levels of hierarchy

“Real life” Avionics function Case Study - Status

	Generation (Gene-Auto without optimization tool) with AIRBUS backends
Simulink Model	Original Simulink model split in 107 elementary models (for sequencing purpose) *Code generation performance to be checked on the complete original model (>several hours ?)*)
Code Generation	OK But some issues have been detected during backend development : Structured data, empty mask, order attributes ⇒Trackers opened
Compilation	To Do
Integration on target	To Do
Functional verification on target	To Do
Performance analysis	To Do

Gene-Auto benchmark for Simulation AP2633 code

Context :

Simulation models : hydraulics, engine…

- For Aircraft 0 (simulation + real equipment), Aircraft –1 (virtual equipment), OCASIME (desktop simulation)
- Host PC Linux, PC windows (for local verification)
- Simulink models provided by several system vendors

Aim :

- **Replace RTW as much as possible… (lower licence costs, more flexible tool…)**
- **Customize Gene-Auto in order to produce AP2633 compliant source code without additional post-processing**

Status : On-going

Agenda

1/ OBSYS (EADS research demonstrator)

- Primary Flight Control case study :
  - Part of laws function (using vector based controller)
  - Part of logics function (using state machines)

2/ AIRBUS internal research activities :

- “Real life” Avionics case study **:**
  - Weight and Balance Backup Computation Function (Experimentation on a complete Simulink specification (equiv to 100 SCADE nodes)
- Gene-auto evaluation to produce AP2633 code for simulation

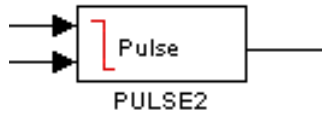
3/ Feedback on Code Customization

4/ Global Status

AIRBUS

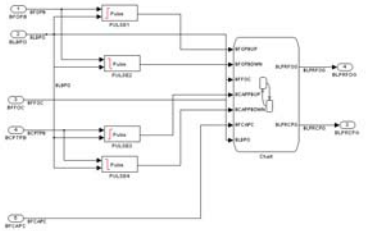
Feedback on code customization with Gene-Auto

- **Local customization** (at block level)



  - Call to external code by using lib.xml :
    - call to a C function `y=function(x)`
    - TBC for macros `macro_m(x,y)`
  - Develop backend :
    - use Macro instead of function (for using embedded symbol library)
    - compute constants (to reduce CPU consumption)
    - add instructions (ex: pragmas, data…)

- **Global code customization** (at model level)



  - The development of an additional tool is on-going.
    This tool is placed between CodeGenerator and Printer.
    - add external dependencies (includes)
    - add instructions for verification tools (WCET tool…)

Agenda

1/ OBSYS (EADS research demonstrator)

- Primary Flight Control case study :
  - Part of laws function (using vector based controller)
  - Part of logics function (using state machines)

2/ AIRBUS internal research activities :

- “Real life” Avionics case study **:**
  - Weight and Balance Backup Computation Function (Experimentation on a complete Simulink specification (equiv to 100 SCADE nodes)
- Gene-auto evaluation to produce AP2633 code for simulation

3/ Feedback on Code Customization

4/ Global Status

AIRBUS

Global status

- 16 Problem reports opened (Mantis tool)
- Good reactivity from Krates
- Sufficient level of maturity of Gene-Auto at this stage
- Toolset architecture (developers feedback) :
  - Several tools allow more flexibility (to add new tool…) and Model Driven Architecture gives a strong evolution potential

- **To be Done :**
  - **Solve the problem of access to data with separate models**
  - **Optimize source code with Gene-auto optimizer and use of cache memory)**
  - **Analyse cpu consumption for part of flight control logics function (StateFlow)**
  - **Look at structure of source code regarding certification issues**