# BE SCADE
# Study of the Longitudinal Flight System


Frédéric Camps
**LAAS/CNRS**
fcamps@laas.fr

# 1 Objective of the BE

## *1.1 General objective*

The general objective of the TP is to specify and validate an embedded system.

- Specification: switching from a high-level specification (textual) to a detailed specification (SCADE formalism).
- Traceability between the high-level specification (textual) to a detailed specification (SCADE formalism) using the gateway "Requirements Management"
- Validation using simulation tools and formal proof.
- Verification of the validation using the model test coverage tool (MTC)

A final report is requested, with the following content:

1. Description of each function of the system (SCADE graphics board, and explanations as to the design choices)
2. Traceability matrix report between the textual specification and the detailed specification
3. Output state of the simulations (graphics)
4. Description of the formal proof test (SCADE graphics boards of the properties, and output state of the Prover tool)
5. Output report for the model coverage

## *1.2 Specification constraints*

The specification has to be done in graphics format. The specification has to be compact but legible.

# 2 Specification

The high-level textual specification is in the SpecificationVL.doc document

# 3 Documents provided

- SpecificationVL.doc: high-level textual specification of the Longitudinal Flight system
- SystemVL.etp: Scade project to be completed
- Appendix Bibliotheque.pdf: Document listing the operators provided in the BIB_SVL library

# 4 Work to be performed

## *4.1 Design*

- Open the SystemVL.etp project, this project contains in terms of libraries:
    - o a BIB_SVL.etp project containing basic operators such as Sine, Confirmer, Derivative, etc. (see doc. on the library),
- Create the **"System_PA"** node of the highest level that corresponds to the Longitudinal Flight system,
- Create its interface (inputs/outputs) using the detailed specification (comply with the order given),
- Create the **"Auto Pilot"** node that corresponds to the "Automatic Pilot" subsystem of the detailed specification, its interface and do the graphics implementation in the form of state machines,
- Create the **"Alarms"** node corresponding to the "Alarms and Indicators" subsystem of the detailed specification, its interface and do the graphics implementation,
- Create the **"Controls"** node corresponding to the "Flight controls" subsystem of the detailed specification, its interface and do the graphics implementation,
- Create the **"CalculSlope"** node corresponding to the "Calculation of the flight parameters/Req 20" subsystem of the detailed specification, its interface and do the graphics implementation (see formula in the appendix),
- Create the **"CalculAltitude"** node corresponding to the "Calculation of the flight parameters/Req 18" subsystem of the detailed specification, its interface and do the graphics implementation (see formula in the appendix),
- Create the **"CalculDensity"** node, its interface and do the graphics implementation (see formula in the appendix),
- Create the **"CalculSpeed"** node corresponding to the "Calculation of the flight parameters/Req 19" subsystem of the detailed specification, its interface and do the graphics implementation (see formula in the appendix),
- Create the **"Calculs"** node, its interface and do the graphics implementation by re-using the CalculAtitude, "CalculSpeed" and "CalculDensit" and CalculSlope nodes that you have just created,
- Finally, do the graphics implementation of the "System_PA" node by re-using the nodes "AutoPilot", "Controls", "Calculs", "Alarms",
- Connect the DISPLAY (horizon and spolier to the System_PA operator)
- Generate the automatic documentation by using the Reporter.

## *Traceability*

- Produce a traceability matrix

## *4.2 Validation via the Simulation*

In the SystemVL.etp project

- Simulate the "System_PA" node according to the following 2 scenarios:

**Req. 15** Ground test: Pa=P0=101325, gear extended, other inputs 0 or false. The outputs must be 0 or false.

**Req. 16** Flight test: Load the provided scenario TestScenario1.in and analyze the output state of the simulator.

## *4.3 Validation via Formal Proof*

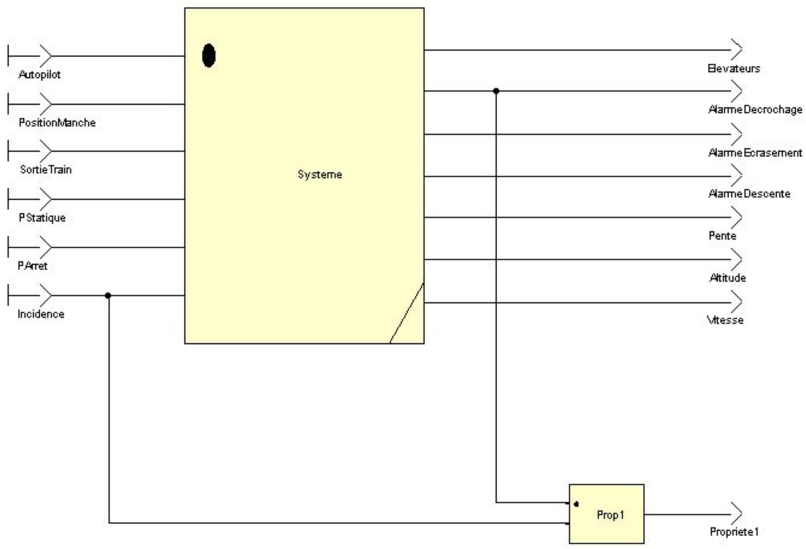
The SCADE PROVER module makes it possible to apply the Proof techniques to the SCADE specifications. These techniques consist in automatically testing all of the possible combinations of inputs in order to prove that a given Boolean output, representing a property of the system, can never be false. If the property can be falsified, the tool provides a counter-example.

In the SystemVL.etp project:

- ➢ Create an Observer note, and copy in its interface the I/O of the Longitudinal Flight system.
- ➢ Call the "System_PA" node of the VF system in the observer
- ➢ Create a node for the property to be proven and connect this node to the node "System_PA"

  **Req. 17** Prove that it is not possible to have an untimely stall alarm, i.e. it cannot be triggered if the incidence is less than or equal to 12° (additional proof of the requirement Req. 1)

- ➢ The Observer node must look like this:




Propose a formal proof for each critical alarm of the system: acceleration, roll, yaw, pitch, temperature, sensor failure, heating...

- ➢ Provide the reports obtained by the proof engine

## 4.4 *Verification of the validation using model test coverage*

In the SystemVL.etp project:

- ➢ Instrument the project
- ➢ Open the instrumented project and add an MTC Result project
- ➢ Run the acquisition on this resulting project
- ➢ Run the scenario provided for the simulation
- ➢ By analyzing the coverage obtained, one realizes that the "CalculDensity" node is not entirely covered.
- ➢ Create a simulation scenario in such a way as to fully cover this "CalculDensity" node
- ➢ Once this node is covered, generate the automatic report on the functional coverage.

| |
|---|
| **Warning**: the scenario provided does not fully cover the specifications. Provide the modification of the scenarios in order to highlight all of the operators. The PA is not used in the initial scenario. |

## 4.5 *Additional specifications*

- ➢ See "Flight parameter calculation – cont.", 'flight controls – cont".

-oOo-