D2.2.1 – Specification of Modelica extensions and interfaces for Bayesian networks, Fault trees and hybrid stochastic models

WP 2.2: Safety analysis methods: Bayesian networks, Fault trees and hybrid stochastic models

Work Package 2: Properties modelling and Safety

MODRIO (11004)

Version 1.0

Date 25/04/2016

Authors

Claire Campan	Dassault-Aviation
Eric Thomas	Dassault-Aviation

Summary

Executive summary

This document takes part of D2.2.1 – Specification of Modelica extensions and interfaces for Bayesian networks, Fault trees and hybrid stochastic models (specification of how to efficiently model, visualize, parameterize and analyze Bayesian networks, Fault trees and hybrid stochastic models in Modelica environments. Design of needed Modelica extensions and a flattened Modelica representation, possibly XML based, for the purpose of allowing different back ends for specialized analysis.)

It also gives results obtained during the project.

For Dassault-Aviation, the work is focus on Fault trees generation and analysis.

1. SUMMARY

This document defines the context of the sub-work package 2.2 and delivery objectives.

For Dassault-Aviation, the main challenge is to be able to define a process, as standard as possible, to generate Fault trees from aircraft architectures and make safety analyze on the whole system with all sub-systems.

The objective is then to

- Study if a process can be defined by:
  - Analyzing how fault trees can be generated from aircraft architectures, when the architecture is described with models written in Modelica
  - Defining a formal process to move from a set of physical models described as continuous and discrete DAE associated to synchronous features to a pure digital (Boolean) model
  - Analyzing generation of fault trees and visualization of safety characteristics in the architecture, and its components
- Specify missing elements to enable the previous features, and specify requirements for the Modelica language, FMI (for models encapsulated within FMU) and for the tools.

Contribution: Dassault- Aviation contributes to study how to generate FTA models from Modelica models, participates in prototyping experiments and case studies. The WP results will be tested, if possible, in the demonstrators (WP8.4).

Delivery: Specification of Modelica extensions and interfaces for Fault trees: specification of how to efficiently model, visualize, parameterize and analyze Fault in Modelica environments. Design of needed Modelica extensions and a flattened Modelica representation (possibly XML based) for the purpose of allowing different back ends for specialized analysis.

Dependencies with other Work-Packages:

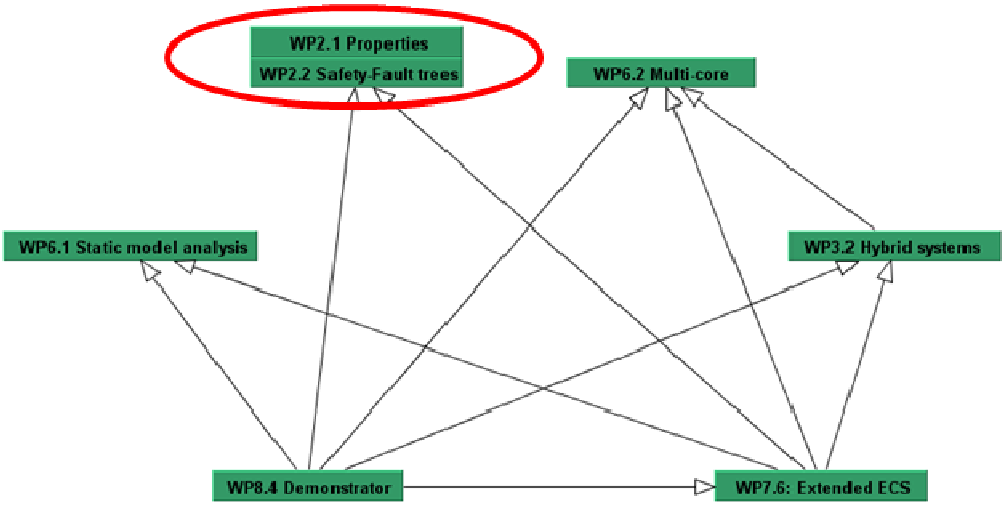


Figure 1 : Work-packages dependencies

1.1 Acronyms

- ASA/SSA : Aircraft / System Safety Assessment
- BAS : Bleed Air System
- BAMS : Bleed Air Management System
- CCA : Common Cause Analysis
- BIZJET : Business aircraft
- DAE : Differential Algebraic Equations
- ECS : Environmental Control System
- FHA : Functional Hazard Assessments
- FTA : Fault Trees Analysis
- FMI : Functional Mock-up Interface
- FMU : Functional Mock-up Unit
- LTS : Liebherr Aerospace Toulouse
- MSL : Modelica Standard Library
- PASA/PSSA : Preliminary Aircraft / System Safety Assessment
- UAV : Unmanned Aerial Vehicle

2. GENERAL

The need to be able to make easily different kind of analysis of architectures is a long term requirement from designers. But, even if modeling and simulation tools have evolved drastically within the last ten years, the gap to have consistent assessments and traceability from design architecture to simulation results is still large. It is particular true for safety analysis which uses very different models than physical ones.

Following chapters explain the context, current process and requirements.

2.1 Aircraft design, architecture and design process

Dassault Aviation designs business and military aircrafts like those represented in the following pictures:

BIZJET FAMILY (Falcon F7X ...)

UAV (nEUROn ...)

MILITARY AIRCRAFT (Rafale ...)

For Energy Aircraft Vehicle Systems, like those represented below for a conventional aircraft architecture, Dassault-Aviation works mainly as a sub-systems or equipment integrator within the structure of the aircraft.

A conventional aircraft is a network of more than one thousand equipment:

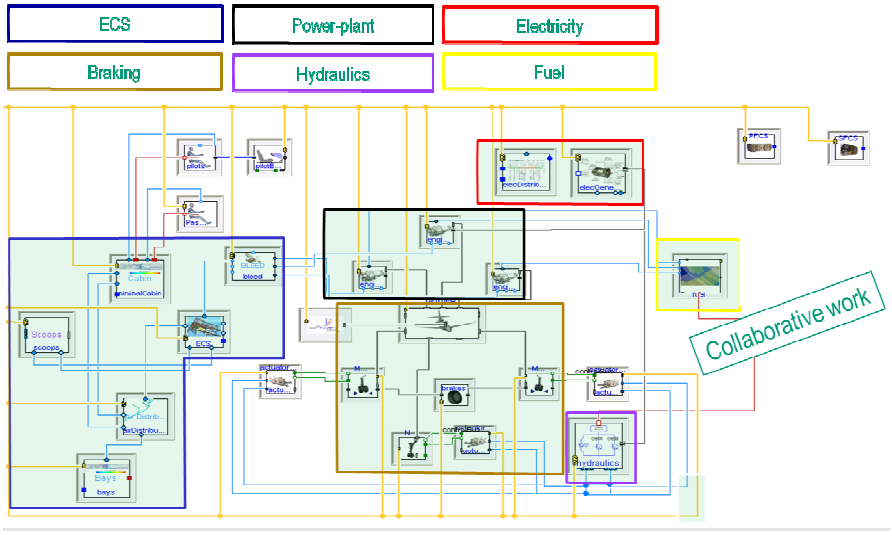


Figure 2 : Conventional Aircraft Architecture

The whole aircraft is made thanks to many partners. The design is then a collaborative work, from early to detailed phases.

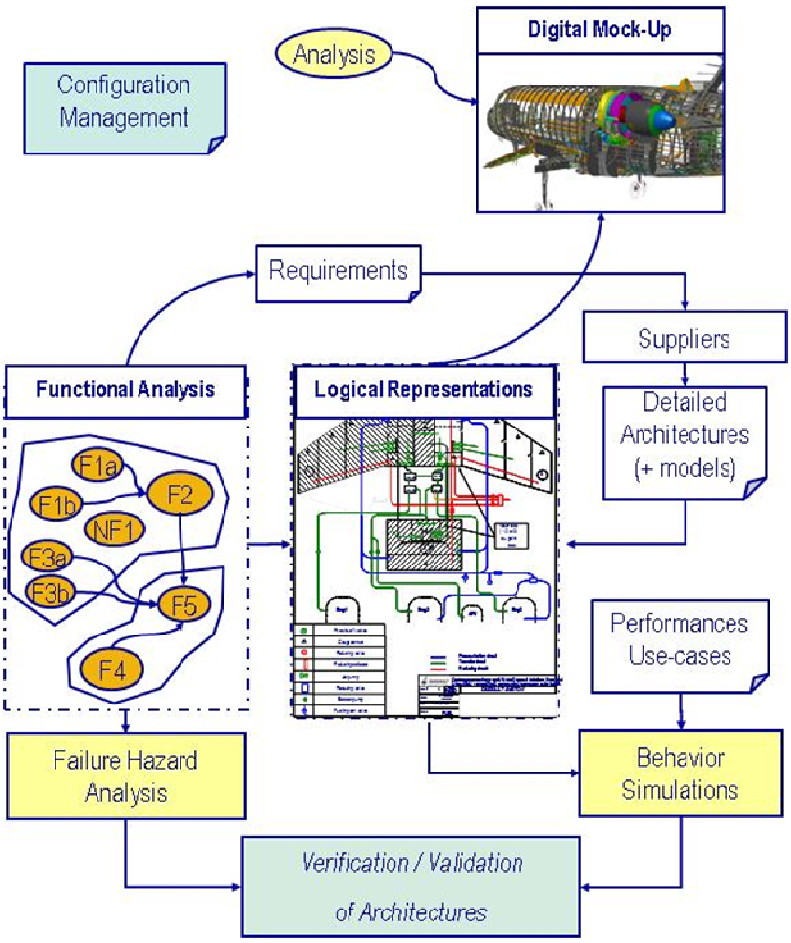


Figure 3 : Design process

The current trend is to use flexible tools able to deals with these various types of models and representations.

Several activities of the verification and validation process made during functional analysis, analysis which participates in justifications (e.g. FHA - Failure Hazard Analysis, behavioral analysis) are also sketched, connected to functional, logical and physical architectures.

The safety process is particular important for aircraft design and architecture selection to assure a high operational aircraft reliability. For aircraft, different guidelines define recommended process for aircraft and systems development, and in particular the safety assessment process during the development phase and during the in-service/operational phase (see [2]). As illustrated in fig. 2, the safety assessment process is tightly linked to the system development process. In the safety assessment process different type of analysis, called CCA, FHA, PSSA, SSA, are used.

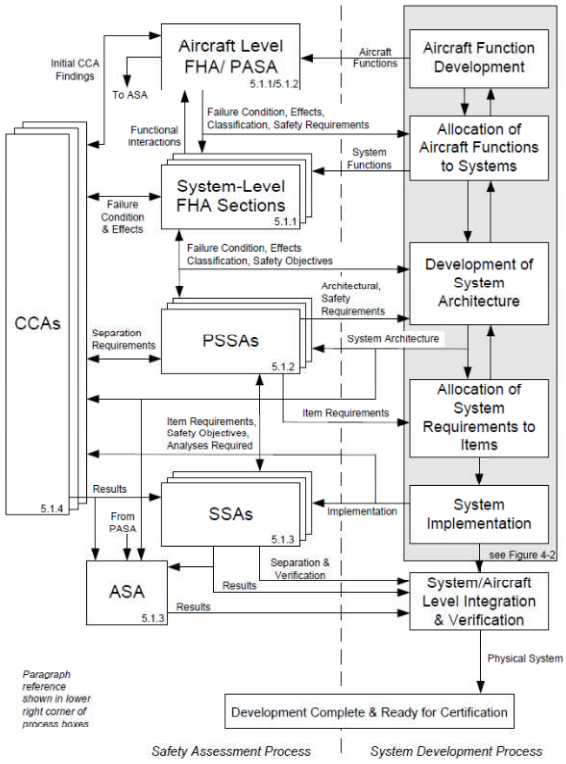


Figure 4 : SAFETY ASSESSMENT PROCESS MODEL (Extract from SAE ARP 4754-A)

Extract from ARP 4754 "The FTA takes part in "the Aircraft / System Safety Assessment (ASA/SSA)," which "is a systematic, comprehensive evaluation of the implemented aircraft and system(s) to show that requirements are satisfied … The ASA/SSA integrates the results of the various analyses to verify the safety of the overall aircraft/systems and to cover all of the specific safety considerations identified in the PASA/PSSA. The ASA/SSA process data includes results of the relevant analyses and substantiation. This may include the following information:

- e. Qualitative analyses for Failure Conditions (e.g. FTA, FMES, Markov Analysis, Dependence Diagrams),
- f. Quantitative analyses for Failure Conditions (e.g. FTA, FMES, Markov Analysis, Dependence Diagrams), …
- h. Safety related tasks and intervals (FTA, FMES, Markov Analysis, Dependence Diagrams), …"

2.2 Problem to be solved

The problem is here illustrated on an energy distribution system.

The topic of this document is to propose a method to analyze the propagation of the failures in such a network without being confronted to the algebraic loop problem.

The document presents the method applied to an electric distribution network, and then generalizes it to other types of energy.

A network of energy distribution consists of one or several sources and suppliers power to one or several consumers. Every consumer can be supplied through one or several paths joining it to one or several sources.

This method takes place within general methods for safety analysis by abstraction of the acausal continuous models which defines the physical behavior of the system.

2.3 Survey of existing studies

A survey of published existing studies around FTA has be done. Several references of previous have been already included in the paragraph REFERENCES at the end of the document.

Current status is that method used where applied on quite small systems with types of architectures of aircrafts.

3. PROPOSED PROCESS

The proposed process is based on three steps to move from an acausal continuous model (which defines the behavior of the architecture/system) to synchronous causal Boolean Model (which allows standard Failure Trees Analysis):

- Abstraction of the acausal continuous models
- Acausal hybrid model
- Synchronous Causal Boolean Model

All of these steps are described below.

Warning: the purpose of this work is to explore existing capabilities of Modelica, in particular including new synchronous features, for FTA generation from aircraft systems architectures. Its aim at finding current limitations of the method, language (Modelica and FMI), and tools to obtain industrial capabilities (able to make safety analysis on large and complex architectures as such we can find in aircraft systems).

3.1 Abstraction of the acausal continuous models

For a given level of decomposition, every item/component of the full system fulfills sets of functions. The validity of a set of functions is defined by a property, and its violation may define a failure for the safety analysis.

Stochastic parameters can be associated to the physical model which will characterize the probability of the failure event.

These failure events may involve changes of equations (change of DAE) system. To model these changes, we can make hybrid systems of equations, where every DAE is executed in a Modelica Synchronous state; and transitions between states are triggered by the failure event (violation of property).

We can make a first abstraction of this hybrid model in which we can find the same states corresponding in every DAE. All variables of this model are Boolean or enumeration variables. Such a model is therefore a causal model; the calculated output variables characterize the quality (in relation to the nominal actuation behavior) of components outputs of the system. For each DAE corresponds therefore a value of quality, and then an equation for the output variable. In the same way, the transitions of the hybrid model modify the DAE. A transition of the synchronous Boolean model corresponds to a change of equation for calculation of outputs.

The networks of energy distribution bring a particular problem for this model transformation. The topological loops can be solved by a stationary point calculation for the continuous variables. The "connect" statement between variables of type flow achieves the law of Kirchhoff to compute the flow balance (for the variables of type stream, a supplementary parameter is used for the calculation of the equilibrium).

The synchronous Boolean model allows calculating the stabilization point of the variables, characterizing the outputs of the component interconnected in a topological loop.

To break the instantaneous dependences between variables, we can introduce some delays (previous statement).

The point of stabilization is reached when no variable value of the topological loop changes anymore.

In the case that an algebraic loop is present in the system of equations, if the compiler detects that n variables are not calculable in this system, it will be necessary to put a delay ("previous") when using these variables. If all variables are interdependent, i.e. all variable need other variables of the system to be calculated, there nxn "previous" operators will be required in the final system of equations.

This system will reach its stabilization values after nxn calculations (otherwise the system would be unstable).

To transform the synchronous Boolean model which is time dependent into a pure Boolean time independent model, we have developed a method based on Shannon decomposition. This method allows transforming a system of Boolean equations with algebraic loops in a system of Boolean equations without algebraic loops. These two systems have the same solutions.

3.1.1 Application to an electric distribution network

To show how to get a time-independent causal Boolean, we use a simple example that makes appear a topological loop:

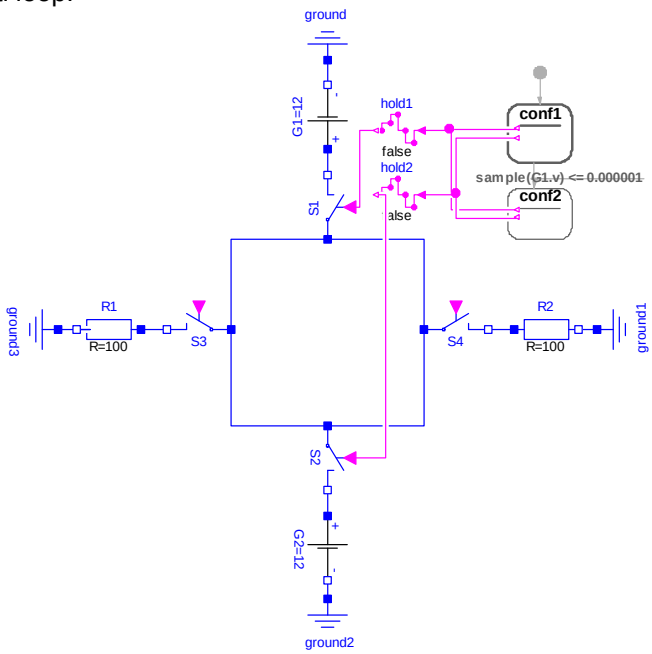


Figure 5: simple electric distribution network

The "hold" and "sample" operators allow interfacing the synchronous and continuous parts.

This model is composed of:

- 2 direct current generators G1 and G2
- 2 electrical resistances R1 and R2
- 2 electrical breakers S3 and S4 which protect G1 and G2 from short circuits of R1 or R2
- 2 electrical switches S1 and S2 that allow supplying power to the network by G1, or by G2. This choice is realized by a controller which detects that voltage at the generator boundaries becomes zero.

The components of this system are modified to add their failure modes. We get a hybrid acausal model; continuous for the computation of the physical variables; synchronous for the changes between modes.

3.2 Acausal hybrid model

In this model one can inject some failures and can visualize their propagation.

The switches are supposed to be the reliable components that don't break down.

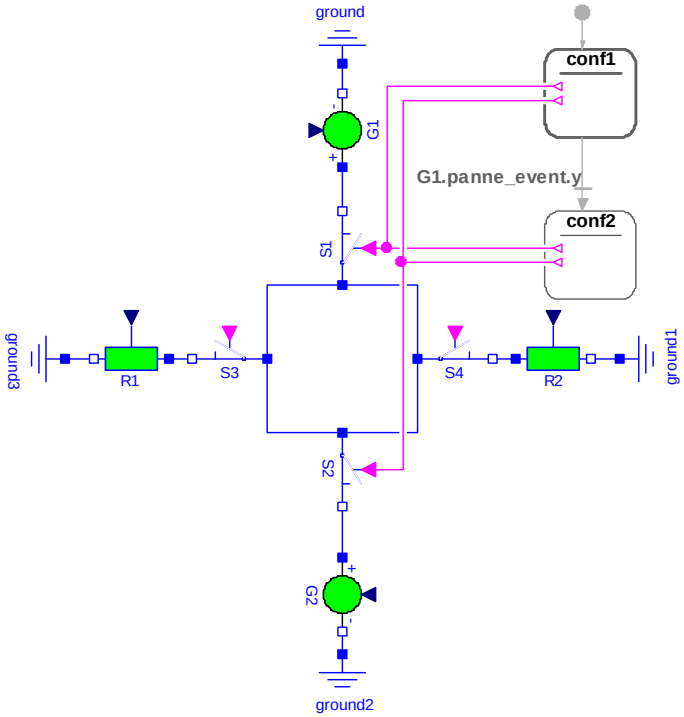


Figure 6: acausal hybrid model of the electric distribution network

3.2.1 Model of the generator

The generator is made from the MSL Electrical package components. It is a voltage generator controlled by an input signal. This signal is controlled by an automata which set the voltage to the nominal value in nominal mode and 10^{-6} in failure mode (we don't set the value to zero to avoid singularity division by 0).

2 conditions cause the crossing of the transition from nominal state to failure state:

- An internal failure ("failure_event") of the generator
- The presence of a short circuit (the current exceeds a threshold value).

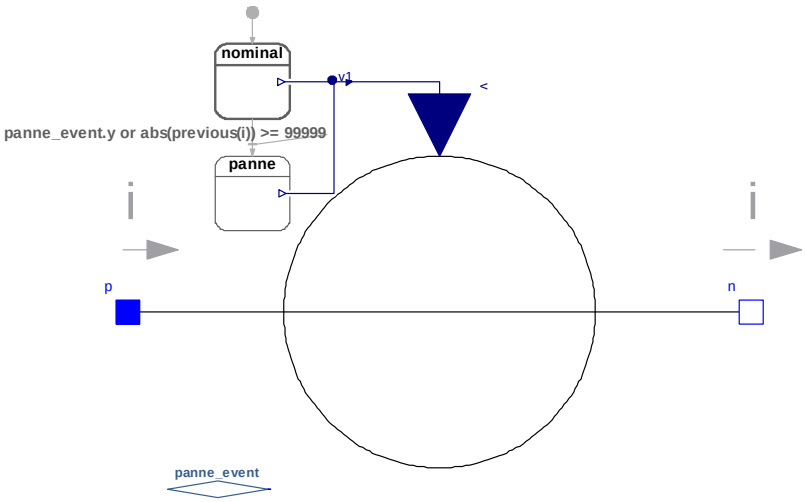


Figure 7: hybrid system of the generator

3.2.2 Model of Equipment, circuit breaker

In the acausal model, the equipment is supposed to operate correctly if a non-zero electrical current lower than a threshold value (maximum value allowable by the equipment) flows through it. The equipment has 2 failure modes, each caused by an event.

In the same way than previously, the components is built from the MSL Electrical package components. It has two modes with two resistance value according to the state ($R=10^{-6}$ if short circuit, $R=10^{6}$ in case of failure.)

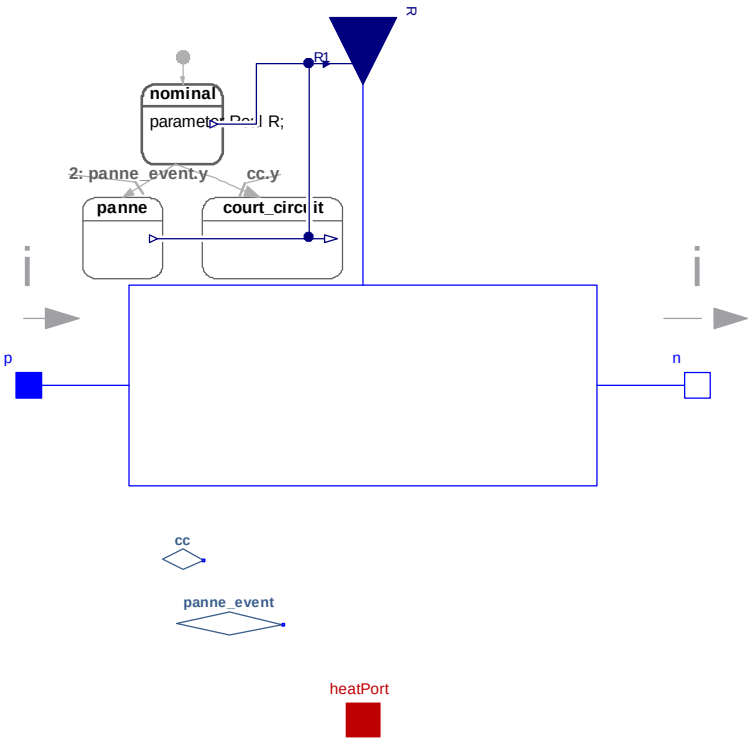


Figure 8: hybrid system of the equipment

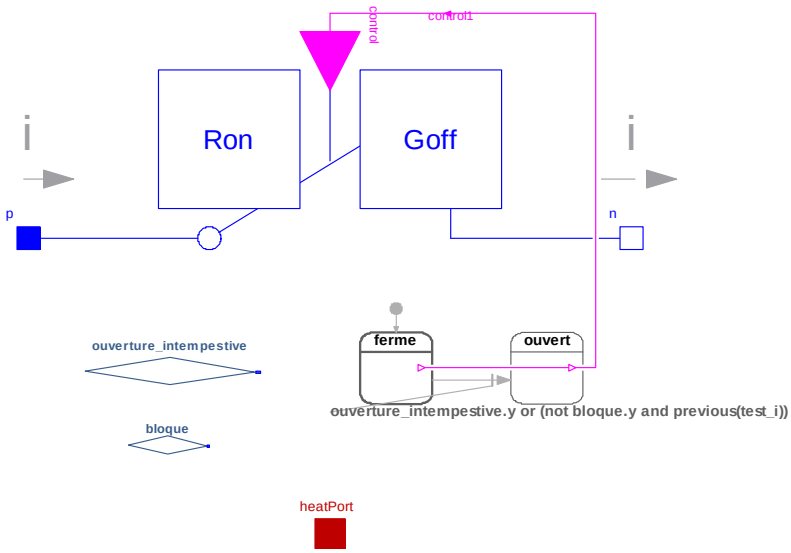


Figure 9: hybrid system of the circuit breaker

3.3 Synchronous Causal Boolean Model

To switch from the Acausal hybrid model to a Causal Boolean Model, following operations are required:

- All real variables are replaced by Boolean variables that characterize the quality (good=true / bad=false) of the real variable.
- The acausal ports of type flow are replaced by bi-directional causal ports that transmit the information Boolean in every sense.

But, the connect statements between Boolean variables cause algebraic loops because the Modelica compiler doesn't look for stationary point on the Booleans. To remove these loops a new component "Barre_T" has been created to establish the stability of the loop (it replaces the stationary point achieved by the compiler on the algebraic loop achieved by the connect statement on real).

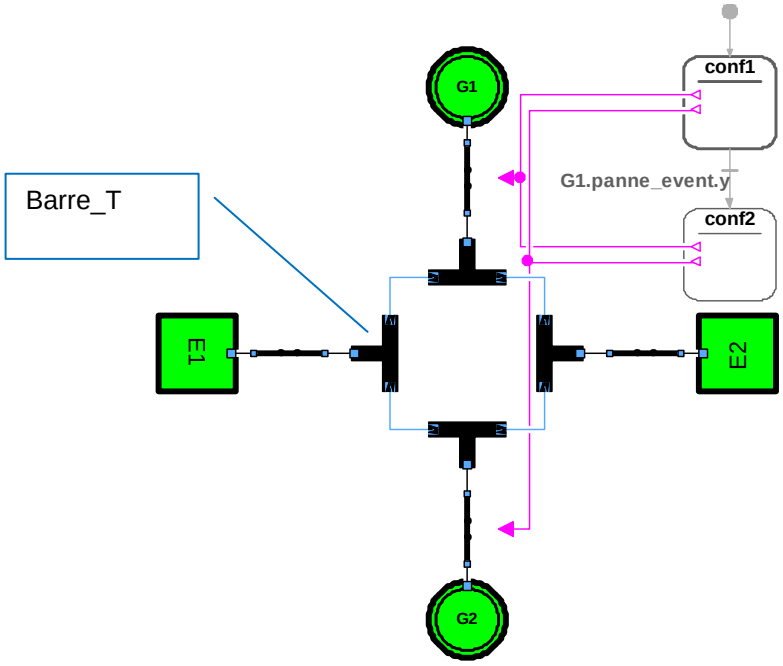


Figure 10: Synchronous causal Boolean Model of the electric distribution network

To break the loop, a component "Barre_T" introduces a delay (previous). When an event causes a change of the input value of one or several "Barre_T" components, this change is instantaneously propagated in the pipe-line made by the interconnected "Barre_T". When these changes are propagated in all pipe-lines, the system reaches a steady state. As this instantaneous propagation is not being endlessly, it doesn't cause an algebraic loop. The system is supposed to be steady when all variables of all "Barre_T" remain without changes between 2 consecutive calculation steps.

3.3.1 Model of the generator, Equipment and circuit breaker

The synchronous model synthesizes the hybrid model like a component which only provides information: it provides the electric power ("p_g"). This variable is a Boolean variable that is set to true if the generator is able to provide the power, false otherwise (failure).

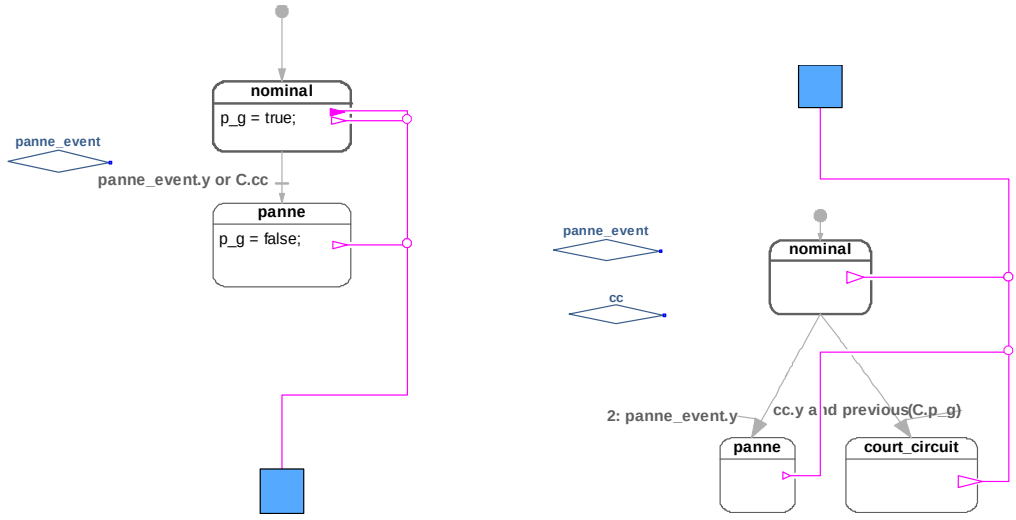


Figure 11: Models of the generator and equipment

These models have the same numbers of states and the same failure events that the hybrid models. The condition of the transition is equivalent to one of the hybrid model.

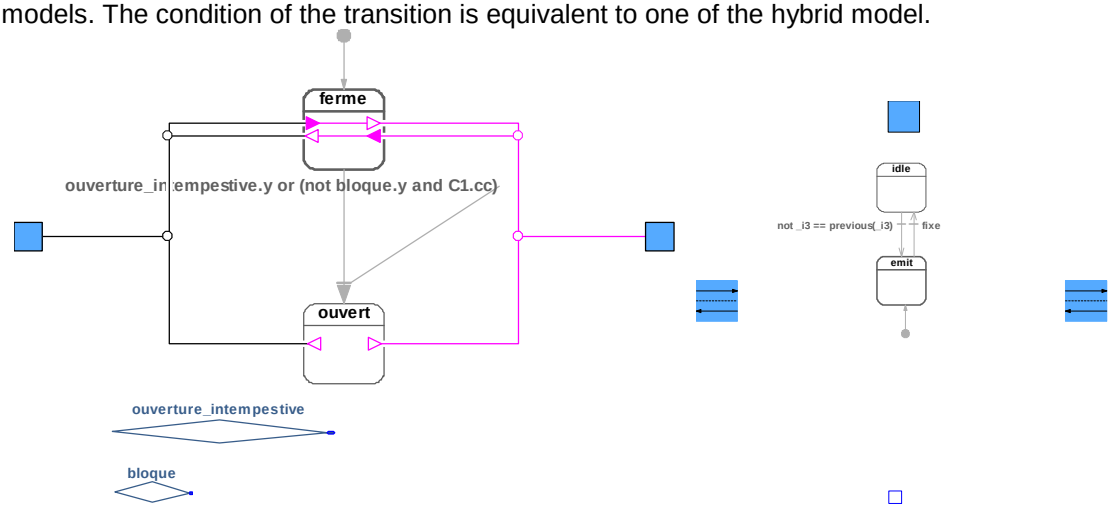


Figure 12: Model of the circuit breaker and Barre_T

3.4 Instantaneous causal Boolean model

This model only contains the Boolean equations and no time (no previous). This model is an abstraction of the synchronous model that only calculates the working modes of the component (corresponding to the states of the synchronous model).

Every component transmits in the network the information that is going to allow each to calculate its state of working, as in the synchronous model, the port is bidirectional.

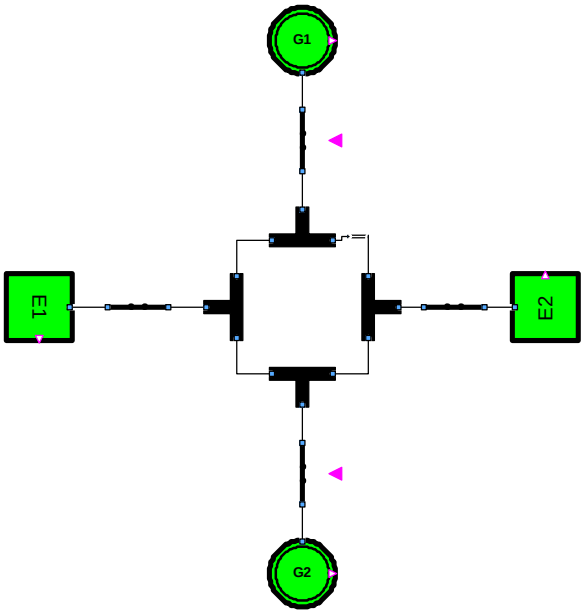


Figure 13: Instantaneous causal Boolean model of the electric distribution network

The causal dependence is thus broken between the capacity of the generator to provide the power and the state of short circuit of a device provided by this generator.

It remains to eliminate the topological loop by the method based on Shannon decomposition.

3.4.1 Models of the generator and equipment

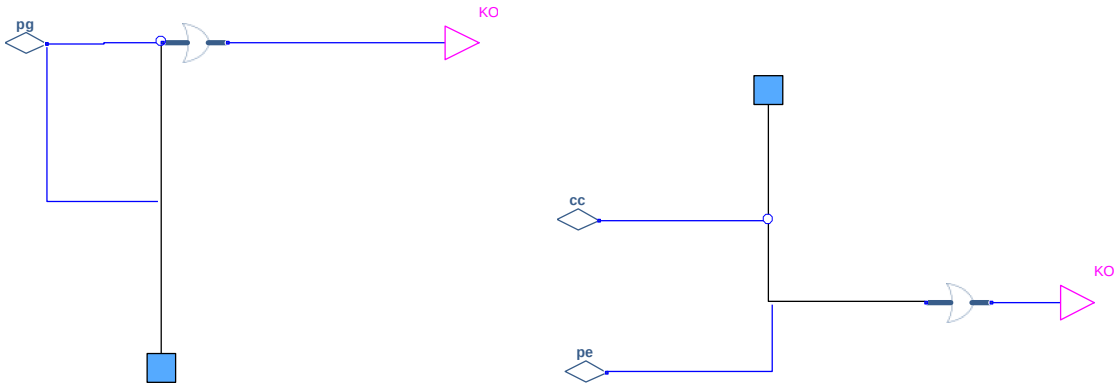


Figure 14: Models of the generator and equipment

The variable KO is a variable that qualifies the state of the generator or equipment from their internal failures propagated on the network.

4. EXPERIMENTATION OF THE COMPLET PROCESS ON THE ELECTRIC DISTRIBUTION EXAMPLE

4.1 Simulations

We verify that the simulation gives the same results with the different models (causal Boolean, causal synchronous and the hybrid acausal models).

Simulated scenario: a device causes a short-circuit when the breaker which isolates it in the circuit is blocked in closed position :

Simulation of the hybrid acausal model:

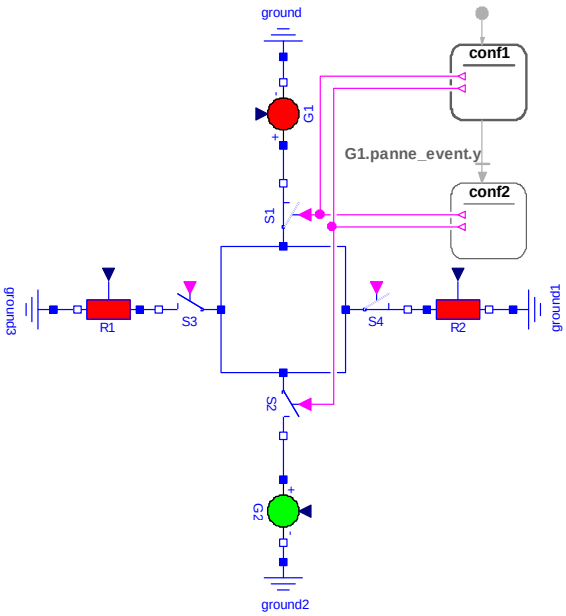


Figure 15: Simulation of the hybrid acausal model

Simulation of the causal synchronous model:

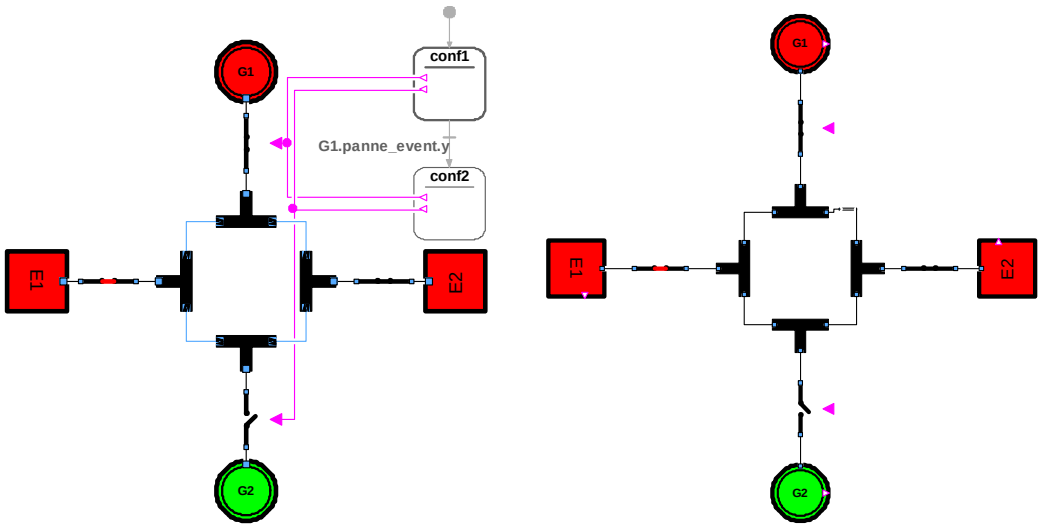
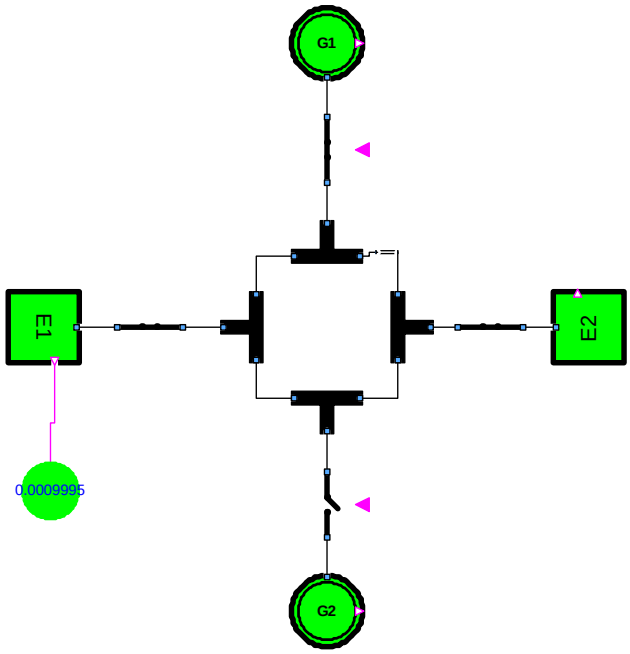


Figure 16: Simulations of causal synchronous and the hybrid acausal models

4.2 Safety Analysis

Probability calculation of the feared event "E1", set to KO:



Failure and cuts trees

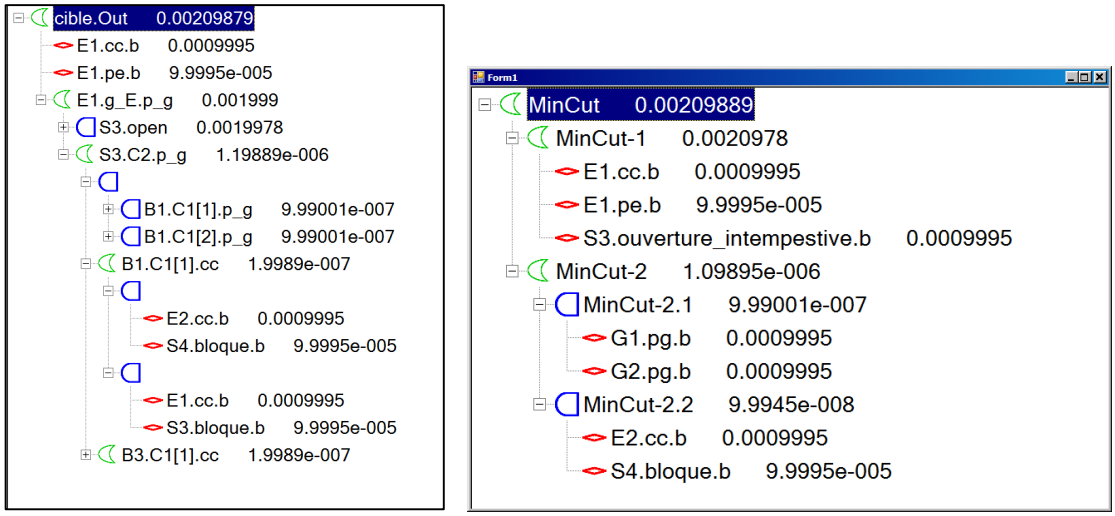


Figure 17 : Visualizes the failure tree and cuts of the feared event

5. TEST OF THE LIBRARY DASAFETY ON LARGE AIRCRAFT SYSTEMS

The capabilities of the developed library have been tested on larger and more realistic use-cases.

In the frame of the TOICA project, we studied the capability to handle larger models, in particular multi-system models.

5.1 Multi-Systems architecture

The following figure shows the Modelica library DASafety, and an application with multiple interconnected systems (Engines, BAS, ECS, Scoops, Air Distribution, and Anti-Icing … systems) with quite simple models. This application was successful, and allowed to get quickly fault trees, cuts … of the whole system.

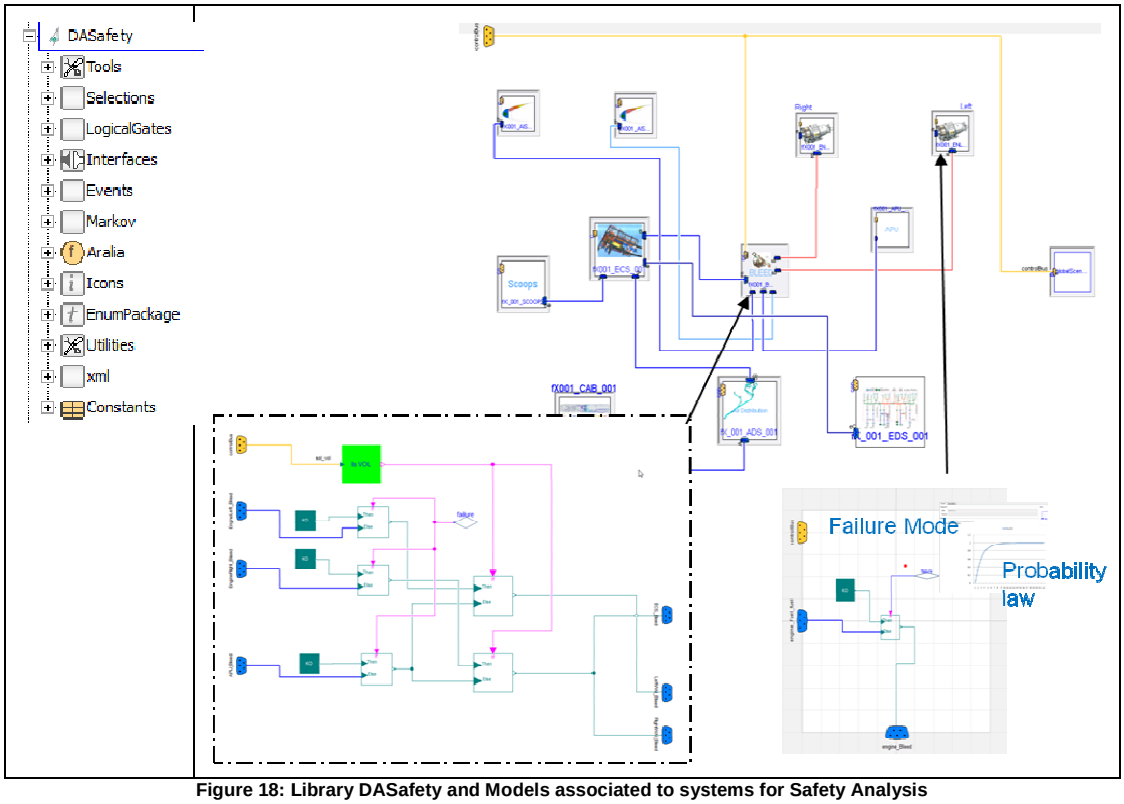


Figure 18: Library DASafety and Models associated to systems for Safety Analysis

From the above model, it is possible to define failure events (here failure of air supply to cabin), perform safety analysis (cut sets), and visualized failures of systems and connections as represented in the following figure.

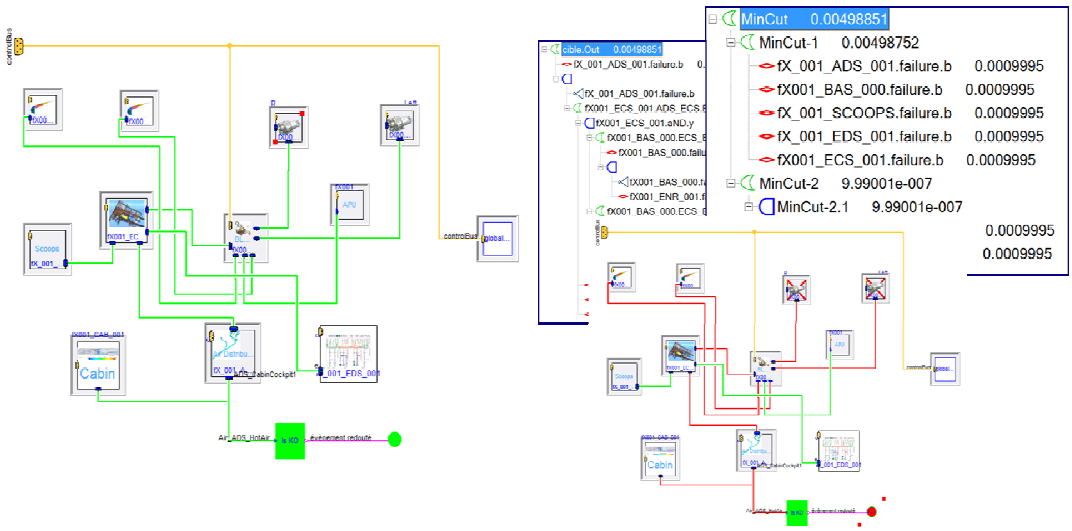


Figure 19: Examples of results obtained with the library

5.2 Safety analysis of a detailed Bleed Air System

The studied system is the complete Bleed Air System (BAS), with two Bleed Air Managed Systems (left and Right BAMS). The whole system is composed of pipes, valves and controllers ("controller_Bleed")

The following figure presents the overall BAS and A/I ducts architecture.

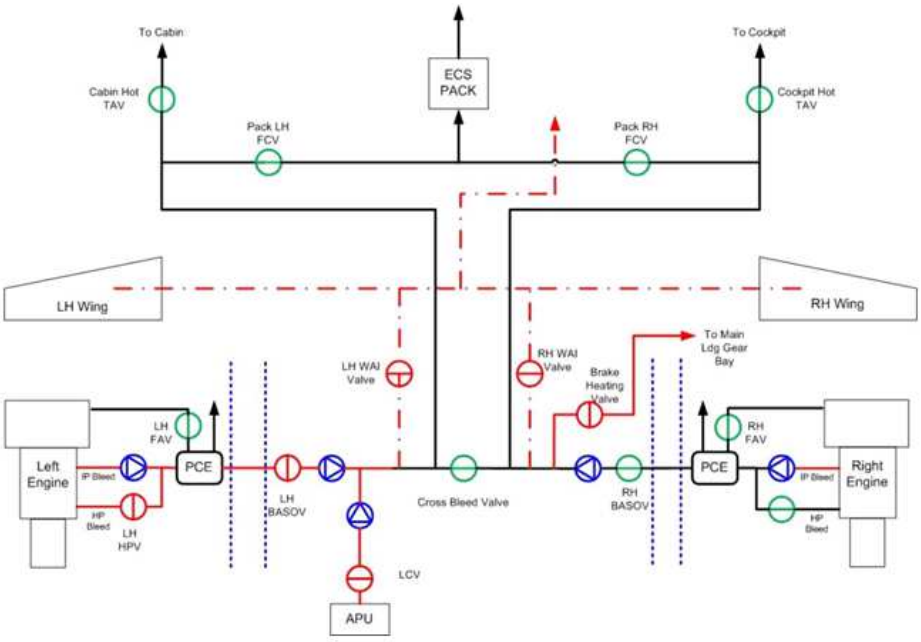


Figure 20: Overall BAS and A/I architecture, with 3 bleed valves and 1 pre-cooler per engine

- PCE stands for Pre-Cooler. It is a heat exchanger. LP air removed at engine fan level is used to lower temperature of the hot IP-HP air mix. LP valve is also called Fan Air Valve or FAV. FAV regulates bleed air's temperature.
- PRSOV, Pressure Regulating Shut-Off Valve, regulates bleed's air pressure.
- BASOV, BAS Shut-Off Valve, is a safety valve dedicated to isolation of manifold and loads from the BAS.

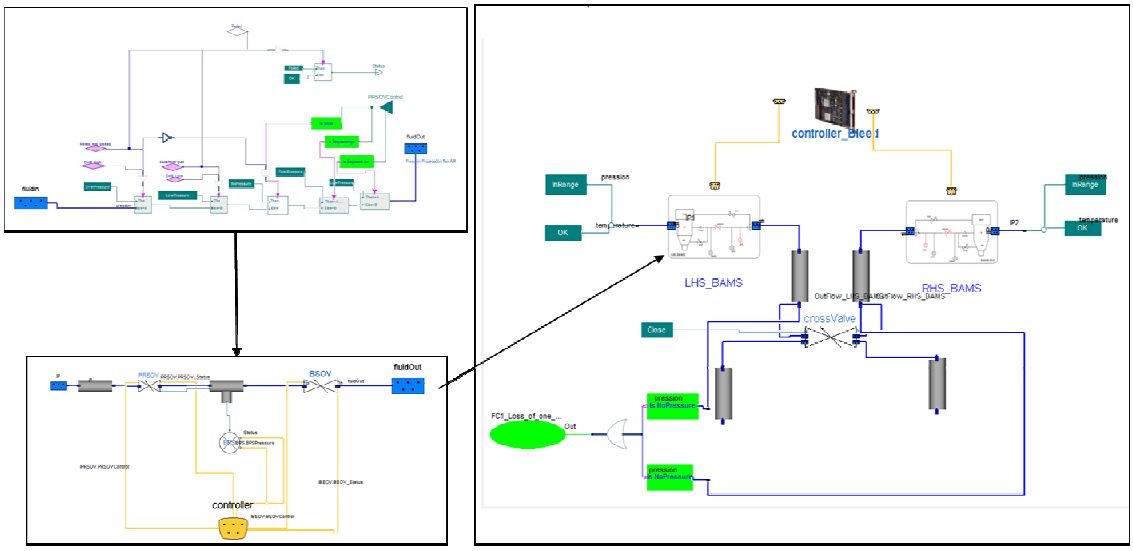


Figure 21: BAS architectures and sub-systems

The controller has two functions. It controls and monitors the left and right BAMS.

This model has 214 elementary components and 1678 scalar equations, for the failure condition "loss of one Bleed". The following figure shows the associated fault tree, with probabilities obtained for every gate.

Figure 22 : Fault tree of the BAS

6. CONCLUSIONS

Result of this work is a very important improvement.

- A promising transformation from physical models to safety models has been studied
- Tests of Safety Analysis of Modelica models have been performed on real models with success. In particular, the defined process allows making performance assessments and safety analysis based on the same modeling framework (common systems interfaces)

It appears that no Modelica language extensions are required to enable such safety analysis.

This new feature works correctly with Dassault Systems Dymola and FT9. Dassault Systems 3DExperience is under test within the project FP7 TOICA.

7. REFERENCES

[R01] Modelica Conference 2012 : "Collaborative complex system design applied to an aircraft system"; Eric Thomas, Michel Ravachol, Jean Baptiste Quincy and Martin Malmheden

[R02] EUROCAE ED-79A / ARP 4754-A "Guidelines for development for Civil Aircraft and Systems"

ARP 4761 "Safety Assessment Process Guidelines & Processes"

ARP 5150 / 5151 "Safety Assessment of Aircraft in Commercial Service"

[R03] SAE 2007 "A Novel Tool for the Conceptual Design of Aircraft Electrical Power Systems", Christian Schallert German Aerospace Centre (DLR), Institute of Robotics and Mechatronics, 82234 Wessling, Germany

[R04] MOET (More-Open Electrical Technologies) Project http://www.eurtd.com/moet/

[R05] 25th International Con-gress of the Aeronautical Sciences (ICAS), Hamburg (2006) "Generator Power Optimisation for a More-Electric Aircraft by Use of a Virtual Iron Bird" C. Schallert, A. Pfeiffer and J. Bals (DLR)

[R06] Modelica 2000 Workshop "Object-Oriented Modeling in Model-Based Diagnosis" Karin Lunde

[R07] "Dynamic Model Based Diagnosis for Combustion Engines in RODON", 2007 Master Thesis performed at Vehicular Systems and Sörman Information & Media AB by Joella Lundkvist StinaWahnström

[R08] "Evaluation of a diagnostic tool for use during system development and operations" Linköping 2007, Daniel Andersson, Patrik Sköld

[R09] "Structural Algorithms in RODON With a prototype implementation in Java", Master's thesis Linköping University, Oskar Särnholm

[R10] 2nd International Workshop on Equation-Based Object-Oriented Languages and Tools, 2008 "Supporting Model-Based Diagnostics with Equation-Based Object Oriented Languages", Peter Bunus, Karin Lunde[11] ITI SafetyDesigner