# Model-based approaches for an improved FDIR development and validation process

*Jean-Paul Blanquart, Pierre Valadeau*[1]
*Astrium Satellites. 31, avenue des cosmonautes, F-31 402 Toulouse Cedex 4, France*
*Email contact: Jean-Paul.Blanquart@astrium.eads.net*

An important and difficult issue in the development and validation of avionics of modern spacecrafts, concerns the FDIR.

Advanced model-based approaches for system and software engineering are more and more widely used to overcome such difficulties. However they still suffer some limitations to address FDIR principally due to the difficulties to represent complex fault dynamics and complex interactions between a large number of failure and recovery processes spread all over the system and across all abstract levels. Moreover to be useful a FDIR model must appropriately combine structural and behavioural views, as well as discrete logic and temporal behaviour.

We propose to present the most recent results of our investigations and elaboration of model-based methods and tools in this area, as well as their integration into an improved FDIR process.

The approach we propose is based on timed automata and their validation through a combination of interactive simulation and formal proofs for instance thanks to modelling tools such as UPPAAL. However whereas it is technically possible to model at the appropriate abstract level the behaviour of a system taking into account the presence of faults and the behaviour of the various FDIR mechanisms, producing, mastering, manipulating and maintaining such models for complex systems remains difficult (and error-prone). Conversely architectural modelling languages such as AADL seem much more appropriate and natural to represent the various components of a system and their organisation and especially most of the information relevant for an easy and natural incorporation of dependability aspects (faults, errors, and failure occurrence and propagation, detection and recovery mechanisms). Moreover they are very close to, and even could be the same as, models used for system and software engineering, thus avoiding duplication of modelling activities and inconsistencies between models. However their capabilities to represent and validate the functional and dysfunctional behaviour, in particular when time related properties are important are quite limited.

Therefore we investigated and elaborated an approach where architectural models are enriched with formally described dependability related features so that timed automata representing the impact of faults and FDIR mechanisms on their behaviour can be automatically extracted and simulated or analysed to formally demonstrate properties (safety or dependability goals or detailed properties of expected FDIR behaviour).

We have in particular exploited and preserved the compositional nature of the AADL model, proposing a library of model entities for fault and failure occurrence and propagation and for FDIR mechanisms. These elementary models are then attached in a compositional way to the entities of the basic architectural model.

We have also investigated and elaborated promising solutions to overcome the well known state explosion problem that limits the capability to demonstrate properties, especially temporal properties when using model-checking techniques. We experimented in particular solutions (based for instance on a limitation of the faults or number of faults to consider, or on the search for stable points in the propagation chain so as to “cut it” into separate parts easier to validate) that can be implemented before the automatic transformation into the timed automata, without having to produce manually ad-hoc reduced models (neither architectural nor timed automata models).

[1] Now with EADS Apsys