Validating Failure Detection Isolation and Recovery Strategies using Timed Automata

Ana-Elena Rugina, Jean-Paul Blanquart, Raymond Soumagne

Validating Failure Detection Isolation and Recovery Strategies using Timed Automata

Ana-Elena Rugina, Jean-Paul Blanquart
Astrium Satellites, 31 rue des Cosmonautes,
31402 Toulouse Cedex 4, France
{firstName.lastName}@astrium.eads.net

Raymond Soumagne
CNES, 18 avenue Edouard Belin,
31401 Toulouse Cedex 9, France
Raymond.Soumagne@cnes.fr

Abstract

The complexity and increased autonomy required for the next-generation space systems call for the deployment of automated techniques for the system validation prior to its implementation. This paper focuses on the approach adopted for validating the Failure Detection Isolation and Recovery strategy (FDIR) in the context of two innovative space projects: formation flying satellites and AGATA (autonomous satellite demonstrator). The validation activities coupled simulation and model-checking based on timed and synchronised automata. Two slightly different approaches have been experimented in order to overcome the state space explosion problem related to model-checking. The first one consisted in performing simulation on a detailed model and model-checking on several abstract models, representative for the target properties. The second one consisted in performing both simulation and model-checking on a less detailed but well focused untimed model. Our conclusions regarding the adopted validation approaches and the validation results are also summarized.

1. Introduction

Space systems become more and more autonomous and complex, which considerably increases the difficulties related to their validation. Powerful and highly automated analysis techniques should help validating such systems' specifications and designs prior to their actual implementation.

Traditionally, the operation of space systems is informally expressed in design documents as a set of modes representing functions, equipments and monitoring mechanisms. Usually the mode dynamics is validated by exhaustive manual (opposed to automatic) consistence checks and cross-reading. These activities become humanly impossible due to the high number of combinations to be analyzed. Powerful computer-aided analysis techniques are expected to help overcoming this issue.

This paper focuses on the approach adopted for validating the Failure Detection Isolation and Recovery strategy (FDIR) in the context of two next-generation space systems: formation flying satellites and AGATA (autonomous satellite demonstrator). Further details are presented in the final study reports [1], [2].

The paper is structured as follows. Section 2 gives an overview of the new challenges generated by the formation flying and AGATA contexts. Section 3 states our validation objectives and shows the resulting validation approaches chosen for formation flying and for AGATA. Section 4 sketches our general validation results. Section 5 presents our conclusions with respect to the chosen validation approach and its results.

2. Context

Formation flying is of increasing interest in the space domain, as it offers the possibility to distribute complex instruments over several spacecrafts. Indeed, the performance of today's space science missions is restricted by limitations on instrument size. The capabilities of telescopes, interferometers, coronagraphs, etc. are directly related to the size of their optics. Thus, several European Formation Flying missions are under preparation, e.g., Proba 3, SIMBOL X, and Pegase. The payload of Proba 3 is a solar coronagraph. The formation is composed of 2 spacecrafts: one spacecraft acts as occulter facing the Sun and the other one is the coronagraph satellite in the shadow of the occulter. SIMBOL X aims at obtaining a new generation of X-ray telescope with large focal length. The telescope is distributed on two satellites flying in formation. The Pegase mission is a space interferometer composed of 3 satellites flying in formation, 1 Beam Combiner spacecraft and 2 identical Siderostat satellites. Formation flying requires specific techniques to ensure flight coordination and the safe behaviour of the spacecrafts in case of anomaly. Thus, satellites flying in formation require increased autonomy and complex decision-making mechanisms.

The CNES and ONERA (the French Aerospace Laboratory) launched a joint programme for investigating the feasibility of an autonomous satellite, as well as innovative technologies for the validation of the underlying concepts. The project aims at developing a demonstrator, called AGATA, which must allow implementing the autonomous satellite architecture and simulating the behaviour of the global system (formed of the ground and space segments) in order to validate on ground the design and operation of such systems.

ASTRIUM supports the French Space Agency (CNES) in the definition and specification of control-command architectures, including FDIR (Failure Detection, Isolation and Recovery), for formation flying for missions with two or three satellites. ASTRIUM was also in charge of defining the FDIR principles for the AGATA demonstrator. These activities included the validation of the newly-defined architectures, both for formation flying missions and for AGATA.

The focus on FDIR is particularly relevant both for formation flying and for autonomous satellites:

- Formation flying introduces yet another layer in the FDIR hierarchical organisation where some actions must be coordinated between the spacecrafts constituting the formation, including distributed diagnosis and reconfiguration, and functions as critical as collision avoidance;
- Autonomy increases both the complexity and the criticality of FDIR; moreover the hierarchical modular architecture of AGATA enforces the design of FDIR as an organisation of collaborative local entities, rather than as a centralised monolithic application;
- FDIR validation is particularly difficult, especially with "traditional approaches" because of the large number of interactions and of situations; moreover these situations are not nominal, increasing the difficulty to analyse them; a large variety of points of view is necessary to fully analyse the behaviour of FDIR and its impact on dependability properties, including to be complete an explicit representation of the entities handled by FDIR: architecture, faults, time, etc.

3. Choice of a Validation Approach

The choice of a validation approach depends on the validation objectives and on the availability and effectiveness of validation technologies. The objectives in the context of formation flying and AGATA were similar. However, we adopted slightly different validation approaches in order to evaluate them.

Paragraph 3.1 motivates our validation choices. Paragraph 3.2 presents the validation strategy for formation flying, while paragraph 3.3 presents the validation strategy for AGATA. Paragraph 3.4 discusses the differences between the two strategies, with their advantages and drawbacks.

3.1 General Considerations

In previous studies, we attempted to validate similar specifications by modelling them under the form of interacting automata in UML and using simulation. Simulation proved to be useful for consolidating the specifications but did not guarantee that the global properties related to the behaviour of the system were satisfied. Indeed, the identification of specification errors relies on the intuition and experience of the person who chooses the simulation scenarios.

As a consequence, we chose to couple simulation with model-checking in the contexts of formation flying and AGATA. Among the available modelling techniques, timed and synchronized automata seem to have the most suitable expressive power for our target domain.

The expected behaviour of the control-command architecture for formation flying and of the FDIR strategy for AGATA was validated by simulation. However, simulation is based on relevant scenarios chosen by the user and does not cover all possible executions allowed by the specifications. Thus, model-checking was used to complement simulation. Model-checking consists in the traversal of the entire state space of the model in order to prove that some given logical or timing properties hold in all states. This coupling allows taking advantage of the strengths of simulation and model-checking while neutralizing their drawbacks (scenario-based analysis for simulation and state space explosion constraint for model-checking).

The general validation strategy based on coupling simulation and model-checking was instantiated in two different ways, both of them aiming at solving the state space explosion problem while being the most representative possible.

For formation flying, a detailed model was built to represent the entire control-command architecture. This model was used for simulation and for extracting several smaller models that were used for model-checking. The extraction process was based on decoupling the hierarchical levels of the detailed model, thus abstracting away details that should not impact each specific property subject to verification. Simulation was also performed on the smaller models in order to validate their representativeness with respect to the expected extracted behaviour.

For AGATA, the choice was to perform simulation and model-checking on the same model. In order to avoid state space explosion, this model focused on a significant part of the system and abstracted away the rest of it. In addition, the model was un-timed: rather than using clock variables for describing the dynamics of the model, a relatively small number of integer variables identifying orders of events drove it. Indeed, timed model-checking algorithms start by partitioning the state space in a finite number of regions with the "same behavior" with respect to the property to be checked. The region graph mainly depends on the number of clocks and the constants occurring in the guards. One of the main drawbacks of timed model checking is that the size of the region graph is exponential in the number of clocks [3].

For pragmatic reasons (timed automata modeller, graphical interface, tool maturity, graphical simulator and user-friendliness of the model-checker), we chose the Uppaal environment [4] to support our validation approaches. Uppaal was jointly developed by the University of Uppsala (Sweden) and the University of Aalbord (Denmark) to support the verification of real-time systems. It includes a graphical editor for timed automata (extended with integer variables, data structures, synchronizations and priorities), an interactive simulator and a model-checker that records the trace of the counter-example to be visualized in the simulator. Uppaal has already been used in the space domain by NASA, to validate the scheduling algorithms for the autonomous control system deployed in Deep Space 1 [5].

3.2 Validation Approach for Formation Flying

The objective was the validation of the entire control-command strategy:

- completeness of the specifications;
- consistency of the modes characterising the different elements of the formation;
- correctness of the telecommand processing;
- completeness / correctness of the FDIR strategy.

The validation approach consisted in performing simulation on a detailed model, including the equipments, functions, satellite and formation mode automata. On the other hand, this model is too large (49 automata ranging from 2 to 10 states) to be entirely analysed by model-checking due to state space explosion. Thus, model-checking was performed on several smaller models, obtained by reduction and abstraction from the detailed model so as to keep representativeness with respect to the properties to be verified (e.g., consistency between function modes and equipment states, between satellite and function modes, between formation and function).

The detailed model was expressed as a set of interacting **timed automata** representing the states of equipments, the function-level operational modes and the satellite and formation modes. Figure 1 shows the modelling approach adopted for capturing the hierarchy of elements and interactions necessary for the validation activities:

- the equipment models capture their temporal dynamics, redundancies and failures caused by temporary and permanent faults;
- the function models represent on-board functions that have an impact on formation flying (GNC – Guidance/Navigation/Control; Command/Control, Payload) and the surveillances related to the formation;
- satellite and formation modes defined as combinations of the function modes.

The ground-satellite and inter-satellite links are also modelled, as the delays generated by them may impact the consistency of the view each satellite has about the state of the formation. In order to be able to validate the FDIR strategy, a fault injection automaton models the faults that may impact equipments.

The function level is the nominal entry point for telecommands from the ground. Functions drive the equipments by ordering their switch-on/off.

As explained above, the models used for the property verification activities were derived from the detailed model by choosing pertinent simplifications; For example, for verifying the properties related to the consistency between function modes and equipment states, the following simplifications were performed:

- the only equipment automata extracted are those representing the equipments necessary in the function mode subject to analysis;
- the mode automata representing the functions were extracted and modified by removing guards referring to no longer modelled equipments;
- the mode automata for telecommand sending and fault injection were extracted.

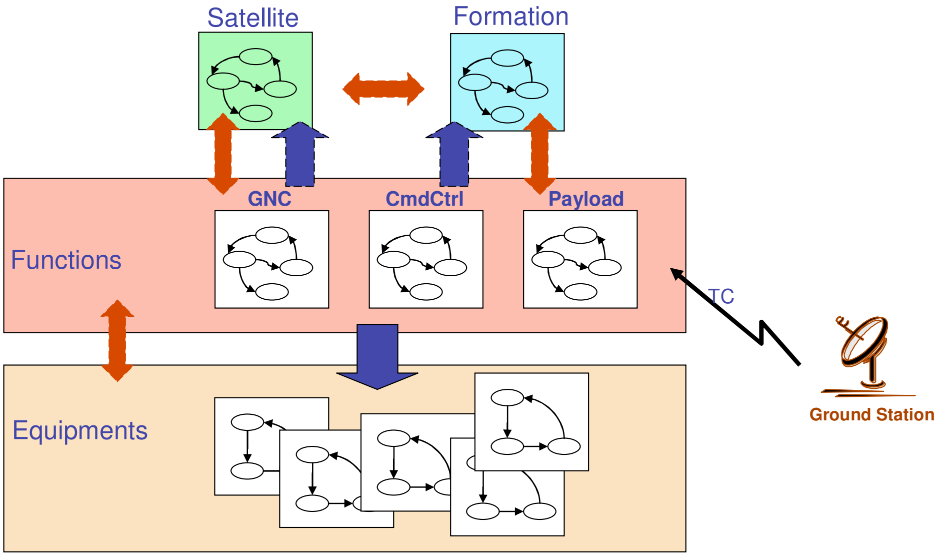

Figure 1. Modelling approach for formation flying

The extracted models for verifying the properties related to the consistency between function modes and equipment states are formed of 8-12 automata ranging from 2 to 10 states.

On the other hand, for verifying the properties related to the consistency between formation and function modes, the following automata were extracted:

- the mode automata representing the functions, the automaton modelling the formation evolution (initialization, formation flying, collision avoidance,…);
- the mode automata modelling the inter-satellite link and the telecommand sending.

The following modifications were also performed in order to simplify the model:

- the guards referring to equipments were removed;
- delays were added in the function automata to represent the equipment switch on/off (equipments are no longer modelled);
- the degraded modes of functions were suppressed (verification with respect to degraded modes is performed when checking the consistency between function modes and equipment states);
- the fault injection automaton was modified to synchronize directly with the surveillance automata.

The extracted model for verifying the properties related to the consistency between formation and function modes is formed of 14 automata ranging from 2 to 10 states.

3.3 Validation Approach for AGATA

The objective was to validate the FDIR strategy by demonstrating required dependability properties (e.g., that no single equipment failure leads the satellite in Safe mode[1], or that the loss of several equipments always leads correctly the satellite in Safe mode).

The validation approach was to perform simulation and model-checking on the same model (12 automata ranging from 2 to 10 states). Thus, the model had to be small enough to be processed by model-checking. In practice, we focused on one function, the AOCS – Attitude and Orbit Control System, of the on-board software. The correct operation of this function depends on the operation of three sub functions (guidance, navigation and control), which in turn make direct use of equipments (GPS, reaction wheel, magnetometer,…). We also chose to use an **untimed model**. On the other hand, we used integer variables to identify the order of certain events relevant to the FDIR strategy. Figure 2 shows the modelling approach adopted for capturing the hierarchy of elements and interactions necessary for validating the FDIR principles of AGATA in the context of the AOCS function:

[1] The Safe mode is the ultimate control degraded mode, preserving the spacecraft survival properties (e.g., sun pointing to charge the batteries). In Safe mode, the spacecraft waits for intervention (telecommands) from ground operators.

- the equipment models capture their dynamics, redundancies and failures caused by temporary and permanent faults;
- equipment monitoring surveys each equipment and manages the local reconfigurations;
- the function models represent sub functions that contribute to the correct operation of the AOCS. They capture the nominal and degraded modes;
- function monitoring is a higher-level decision-making layer, which aims at localising an anomaly once several equipments' surveillances have been triggered. This implies that there is coordination between the equipment and function monitoring for the reconfiguration actions.

As for the formation flying case study, a fault injection automaton models the faults that can affect equipments.

3.4 Discussion

The validation approach used in the context of formation flying allowed us to analyse thoroughly the behaviour of the system and led to the identification of some specification problems. However, it has several drawbacks. The first one is that the properties that hold for the simplified models are not guaranteed to hold for the detailed model, even though we paid particular attention to the relevance of the simplifications with respect to the targeted properties. The second one is that a significant effort is necessary to build the multitude of models, and especially to ensure their consistency: if a specification problem is identified by verifying a property on a simplified model, the modifications performed on this model must be propagated to the other models and all the property verifications, as well as simulations must be re-executed. The extraction of the simplified models depends on the architecture of the detailed model and thus on the architecture of the modelled system. Thus, it seems difficult to consider automating this process in a sufficiently generic and efficient manner. Moreover, it would be too difficult and time-consuming to specify an automated modification process, given the number of possible modifications and the lack of physical traceability between the multitude of models.

The validation approach used in the context of AGATA is less sensitive to modifications in the specification, as a single model is to be updated. However, the model is inherently less detailed (so that the model remains analysable by model-checking), which may cause a representativeness problem. Indeed, we have validated the FDIR strategy for one function only. This implies that possible dependencies between functions with respect to the use of equipments are not taken into account by the analysis.

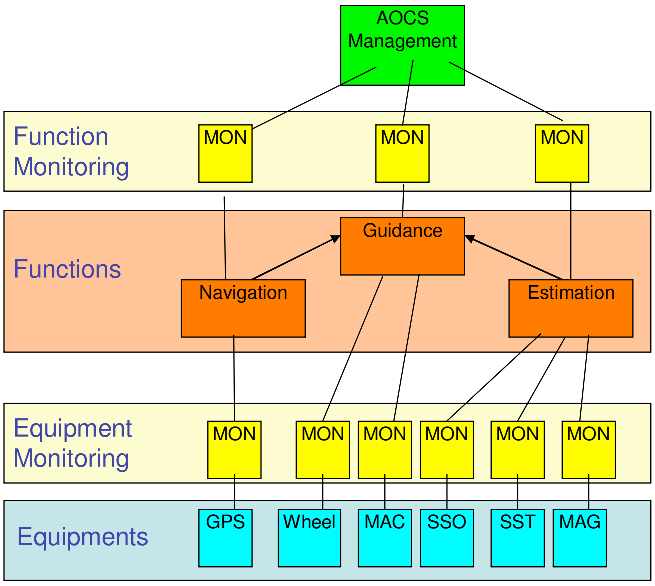

Figure 2. Modelling approach for AGATA

4. Validation Results

Both for formation flying and for AGATA, simulation was performed by executing several types of relevant scenarios:

- nominal scenarios: they correspond to possible system behaviours without taking into account the possible occurrences of equipment failures. Executing such scenarios allows to identify possible inconsistencies or omissions of the specifications with respect to dynamic aspects related to interactions between different parts of the system;
- scenarios for reaching degraded modes: the purpose of such simulations is to validate the management of equipment failures that should lead to a degraded mode;
- coordination of several equipments for complex reconfiguration schemes: the reconfiguration of one equipment sometimes requires the reconfiguration of other equipments. Such scenarios allow making sure that the sequence is not disturbed by third party events;
- satellite and formation reconfiguration in the case of formation flying: the satellite and formation automata evolve according to the function modes. The visibility of function modes of one satellite to the other satellite is delayed by the inter-satellite link. Thus, it is necessary to validate the consistency between the view that each satellite has on the formation.

Several types of properties (logical and timing for formation flying, only logical for AGATA) were verified by model checking:

- the failure of an equipment that is necessary in a given nominal mode leads the function in the appropriate degraded mode immediately;
- mode invariants related to satellite and function / formation modes;
- freedom of deadlocks;
- a single failure does not lead to Safe mode;
- Safe mode of sub function implies Safe mode of function;
- Occurrence of several failures leads to Safe mode;
- In Safe mode, only the equipments necessary in Safe mode are switched on.

The adopted validation approach, based on coupling simulation and model checking, allowed the early identification of some specification problems, as follows:

- Contradictions in the specifications of different function behaviours (identification of unexpected combinations of function modes);
- Sequences of telecommands that should be forbidden;
- Oversights in the surveillance specifications (e.g., surveillance that may be triggered in nominal mode and initiate a collision avoidance manoeuvre in the case of formation flying);
- Behaviour inconsistencies during non-observable transient modes (procedures may be initiated during such a non-observable mode, which may lead to an unexpected combination of function modes).

It is noteworthy that feedback coming both from simulation and property verification impacted the models (e.g., in the case of formation flying, the detailed and the extracted models were updated iteratively) and hence led to modifications in the initial specification. The adopted validation approach allowed for the identification of some specification problems very early in the development cycle, problems that would otherwise be detected only during the implementation phase.

5. Conclusion

Even though the state space explosion is a problem in the context of real-life systems, we believe that formalizing and validating the specifications through simulation and model-checking has several strong advantages. On the one hand, modelling during the specification phase forces the designer to formalise and clarify the specifications. Simulation is useful for validating the model against the specifications and for identifying behaviour inconsistencies based on relevant user-defined scenarios. Such inconsistencies are difficult to identify in a classical purely paper-based specification process. Last, formal verification proves that none of the possible execution scenarios violates the system properties.

Such approaches are useful not only for validating an architecture or FDIR strategy once defined, but also for tuning its parameters during the definition phase.

6. Acknowledgements

The work reported in this paper was performed by EADS ASTRIUM in the context of two studies supported by the French Space Agency (CNES) in 2008. We thank all persons who participated in these studies, both on the ASTRIUM and CNES sides.

7. References

[1] A.E. Rugina, “Etude Détaillée d’une Architecture de Commande/Contrôle pour le Vol en Formation – Rapport Final”, ASTRIUM report ACF-TN-ASF-008, 2008.

[2] J.P. Blanquart, D. Boudet, F. Laval, and Y. Mulet, “AGATA – Architecture d’autonomie et FDIR – Rapport Final”, ASTRIUM report ASG7.TN.12794. ASTR, 2008.

[3] C. Daws, S. Yovine, “Reducing the Number of Clock Variables of Timed Automata”, *17th IEEE Real-Time Systems Symposium*, Los Alamitos, CA, USA, 1996, pp. 73-81.

[4] K.G. Larsen, Paul Pettersson, and W. Yi, “Uppaal in a Nutshell”, *Int. Journal on Software Tools for Technology Transfer*, Springer-Verlag, Vol.1, 1997, pp. 134-152.

[5] L. Khatib, N. Muscettola, and K. Havelund, “Verification of Plan Models Using UPPAAL”, *LNCS Vol. 1871, Formal Approached to Agent-Based Systems*, 2000, pp. 114-122.