Preprints of the 18th IFAC World Congress
Milano (Italy) August 28 - September 2, 2011

Smart architecture for highly available, robust and autonomous satellite

X. Olive*, S. Clerc*, D. Losa*

**Thales Alenia Space, 100 Boulevard du Midi, BP 99, 06156 Cannes La Bocca FRANCE (Tel: +33 4 92 92 66 79; e-mail: {xavier.olive, sebastien.clerc, damiana.losa}@ thalesaleniaspace.com).*

Abstract: European leader for satellite systems and at the forefront of orbital infrastructures, Thales Alenia Space is a joint venture between Thales (67%) and Finmeccanica (33%) and forms with Telespazio a Space Alliance. Thales Alenia Space is a worldwide reference in telecoms, radar and optical Earth observation, defense and security, navigation and science. Thales Alenia Space has 11 industrial sites in 4 European countries (France, Italy, Spain and Belgium) with over 7,200 employees worldwide.

Satellite evolution and wish to design more autonomous mission imply an enhancement of satellite architecture to enable the smart control of spacecraft. New system architecture needs to be defined to permit the decision-taking, and special attention has to be paid to FDIR (Fault Detection, Isolation and Recovery) and the way action can be engaged. Nevertheless the constraints on architecture and related technique composing it, stay still invariant: robustness, high availability, industrially viable and cost efficient. This paper gives first some elements about a decisional architecture defined in joined work by Thales Alenia Space and CNES, then the current context of FDIR is briefly described and then a new FIDR strategy permitting smart decision is introduced, finally the way decision can be engaged on-board for the next generation of autonomous satellites is presented.

Keywords: Aerospace applications; Health monitoring and diagnosis, Autonomous systems, Decision making and autonomy, sensor data fusion.

1. INTRODUCTION

Satellites are today designed to find an equilibrium between three major axis: Cost, Performance and Availability. Cost is driven by commercial constraints; Performance is driven by industrial competitiveness; Availability is driven by the mission.

Satellites have evolved since the last 50 years from pre programmed automata performing a priori known tasks and unable to react against unforeseen events, to smart embedded system able to take pre-programmed decision on event occurrence or able to react against context changes. Following this trend, tomorrow, satellites will become autonomous embedded system able to achieve mission goals with a limited Ground interaction.

In such an evolution dedicated system's architecture should be designed to answer to the need of on-board decision and decision execution. This architecture is made of the current way a satellite is designed, completed by 2 levels dedicated to the autonomy management.

Furthermore FDIR systems are part of the critical path of satellite design. FDIR has a direct impact on the satellite's cost and availability; and indirectly on the performance. It is a cornerstone for the satellite's autonomy enhancement.

Like all the embedded elements on a satellite, FDIR needs to be designed and validated before being integrated into a satellite. Usually this validation is time-consuming and complex to perform, leading to the introduction of risks on the planning and to cost overtaking. Definition of adequate FDIR strategies can lead to decrease these risks without reducing the FDIR functionalities and/or perimeter.

Availability is directly depend from the FDIR system, and mainly the capability it has to recover quickly. Reaction time for detection and fault isolation, robustness and ability to recover from a failure are sizing elements of the satellite availability. For mission requiring a high level of availability (like telecommunication ones), FDIR are often very sophisticated and structured with different levels in order to minimize the effect of a fault w.r.t. the whole satellite, leading to a reduction of mission outage. On the other hand, when availability level is medium (Scientific Earth Observation mission requires availability but short mission interruptions are allowed and have no consequence), satellite saving will be prefer than mission follow-on. Nevertheless Space exploration mission requires an advanced autonomy capability due to the

lack of visibility from Ground. In this case, ability to decide on-board and perform the selected activities is a cornerstone.

Satellite's autonomy is more and more increasing on-board. This need comes from new kind of missions: Exomars mission requires to be able to land; in the next Martian missions generation, it is foreseen to perform in-orbit rendezvous. In the frame of Earth Observation mission, Ground- Board commandability loop will require to be shorten in order to increase the reactivity: by instance, when fire or pollution detections occur, some more detailed observations are quickly needed. Satellite will be able to take this kind of decisions, shortening the reaction loop between detection and action.

This paper gives first some elements about a decisional architecture defined in joined work by Thales Alenia Space and CNES, then the current context of FDIR is briefly described and then a new FIDR strategy permitting smart decision is introduced, finally the way decision can be engaged on-board for the next generation of autonomous satellites is presented.

2. ARCHITECTURE FOR ENABLING ON-BOARD DECISION

The development of a complex autonomous system relies on the organization of the components in a closed loop architecture. Many existing architectures embedding decisional capabilities are based on hierarchical layers. In the robotics domain for instance, the "3T architecture" (Bonasso et al., 1997) is based on three layers consisting of three components, which are a reactive feedback control system, a reactive plan execution system and a time-consuming deliberative system. The "LAAS architecture" (Alami et al., 1998) is also decomposed into three levels: a functional level contains a set of modules encoding the basic functionalities, a decisional level is responsible for both plan generation and plan execution, and the in-between execution control level has a fault protection role and filters the commands sent to the functional level by the decisional level.

In the space domain, the architecture deployed for the Remote Agent experiment (Muscettola et al., 1998) integrates three hierarchical components: a planning system responsible for back-to-back mission planning, an executive used to refine the activities in the plan, and a diagnosis and reconfiguration system.

2.1 Decisional architecture definition

Similarly, we propose to organize the architecture along three hierarchical levels (see Fig. 1). These levels are characterized by different reaction times, they handle more or less abstract data representations and have a different knowledge of the system state (a global or a local one). They interact through commands sent to the lower level and reports / alarms sent to the upper level.

The **decisional level** is in charge of programming the activities of the platform and/or the payload, and monitoring the execution of the activity plan. This level can be active (with or without planning) or inactive.

The **operational level** is in charge of the execution of operations (decomposition and routing of commands, monitoring of the global state of the system).

The **functional level** gathers the procedures and control loops of the operational subsystems.

2.2 Knowledge's organization

The 3 levels of the decisional architecture allow one to organize the knowledge by abstraction level with different time constraints. The decisional level is the most abstract and has the longest reaction time (order of magnitude of an hour). The functional level is the most time constrained (real time) in the architecture. The equipment level (not represented here) is a hard real time one. It handles the data and the drivers of on-board units.

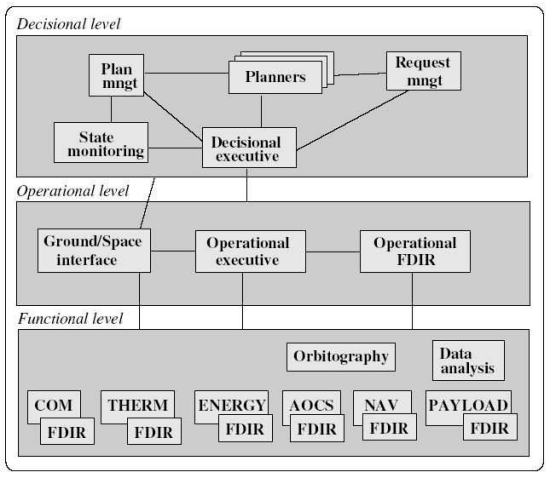

Fig. 1: Decisional architecture overview

The knowledge organization is very close to the one described by Chittaro (Chittaro et al., 1993). The decisional level is in charge of managing the goals of the mission which are received from the Ground. The decisional data are part of the teleological knowledge. Decisional executive determines the action to engage to achieve the goals of the mission and the state monitoring component ensures that the operational conditions are filled.

The functional knowledge (in the Chittaro's sense) is shared by the decisional and the operational levels. The first one is in charge of distributing the activities to be performed by the operational components. The second one executes the requested activities and ensures the consistency between all the requested (from Board and Ground) commands.

The functional level of the decisional architecture is related to the structural and the behavioural knowledge. It is composed by all the subsystems of the satellite. The links between them are not represented on the architecture due to the fact they are dependent of the mission, but they are handled at this level. The behavioural knowledge of the subsystem is partly

represented at this level. The model is related to the performance of the subsystems and does not contain the details of the units. The equipment level contains only behavioural knowledge and the detailed behaviour of the on-board units.

Following this classification the data handled by the decisional and the operational levels are mainly discrete and can be represented by qualitative values and processed through discrete automata. On the opposite, the data of the functional level are close to the continuous world. Data can be represented by quantitative, qualitative abstracted model, continuous or discrete / hybrid models. The following sections give details on the FDIR or diagnosis techniques foreseen to be used in the different layers. First the diagnosis strategy is defined, then the operational FDIR component is detailed and finally some interesting techniques are reported for the functional level.

3. DIAGNOSIS STRATEGY

3.1 Currently used FDIR strategy

The current strategy used by Thales Alenia Space is based on a set of hierarchical levels allowing a graduated reaction with regard to the criticality of faults.

Such a hierarchical structure permits to recover a fault with a quick reaction time and to minimize the perimeter of the fault's effects. Isolation is guaranteed by a local mechanism. Each failure is recovered at the lowest layer as possible to limit the impact on the mission.

The hierarchy is composed by 4 levels, which have different reaction times and are activated successively by order of criticality. The faults are filtered in each level: a higher level can only be called when the lower level has been activated several times (or when a fault depending on this level occurs).

Level 0 deals with failures having no impact of the satellite's subsystem performances and represents faults which can be recovered by local correction (bit flip, cyclic redundancy check (CRC), …). Detection is performed internally in the units. The recovery is autonomous and local to the unit.

Level 1 deals with failures requiring to switch an unit to its redundant one. Detection is performed outside the unit and the recovery is done by the subsystem in which the unit is involved. The effect of such a failure can lead to a temporary degraded mode without any effect on the mission goals. The recovery is autonomous and has an effect on the subsystem.

Level 2 and 3 are often mixed due to the fact they have the same kind of detection and recovery action. They deal with performance lost for a subsystem. Level 2 is strictly related to the occurrence of several alarms coming from the lowest level. This means that the recovery action which has been engaged has not corrected the anomaly and that the fault has to be considered more globally at subsystem level or platform level. Level 3 is related to faults on the FDIR units like software or processor module (PM). These faults are recovered by switching on a redundant PM. Level 2/3 recovery is still considered autonomous for most of these faults even if some main functions can be impacted.

The most critical level is the level 4, which is activated in case of several alarms from level 2/3 or from hardware alarms. A level-4 of alarm is the last one, which can be raised by the satellite and is consequent to a critical breakdown leading to the safe mode. The recovery in this case is done by the Ground and the mission is interrupted.

Through these 4 levels the isolation is seldom considered due to the fact that detection is sufficiently precise to perform isolation at the same time. Most of the detection and recovery actions are of software type except in level 4.

Figure 2 shows the hierarchical FDIR strategy. The higher the level, the more critical is the fault but the lower is the occurrence probability of the faults. The main advantage of such an architecture is the possibility to activate or deactivate one or several levels depending on the mission requirements.

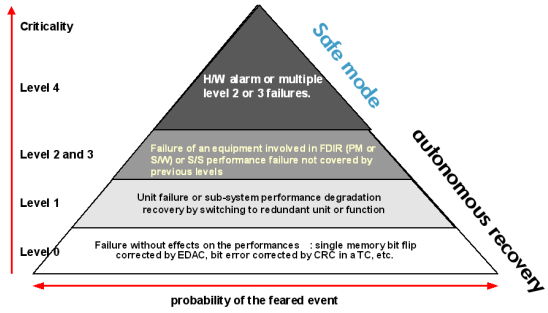

Fig. 2: Hierarchical FDIR strategy

3.2 Decisional FDIR strategy

The FDIR strategy presented previously is oriented to the telecommunication satellite market which requires a strong robustness and minimization of mission outage. The trend for this kind of missions is to keep on with the existing FDIR strategy by improving the detection phase. On the other side, Observation and Scientific missions required more and more advanced autonomy in terms of reaction time reduction and on-board decision. The next generation of European space missions should offer an opportunity to introduce new FDIR concepts on-board, related to the advanced autonomous missions.

Next generation of FDIR strategies is foreseen to be a centralized and distributed one. Centralized in the sense that on-board decision requires to have on a single location a synthetic view of the satellite state. Distributed aspect will allow to keep a local FDIR at the subsystem and the unit levels, similar to the one presented in the previous section for Level 0, 1 and 2. Designing such a FDIR architecture is a way of defining some generic FDIR concepts allowing one to reuse the centralized part and, from an industrial point of view to better master the planning and to reduce the cost. In addition decoupling the global (at satellite level) FDIR and the local ones allows one to relax the constraints on the local FDIR in term of development, and could permit to introduce for specific needs some advanced FDIR techniques.

Exchanges between centralized and distributed parts are under the form of discrete data or events (command, residual status switch, mode change, …), the centralized part (i.e. operational FDIR) handling all these discrete events through some kind of automaton (Bayoudh *et al.*, 2008).

3.3. Operational FDIR

Operational FDIR is related to functional knowledge at satellite / system level and intends to be a common and single location to get the current satellite's state. In case of on-board decision making, this architecture is mandatory in order to interact with the decisional applications.

At system level, data are coming from all the functional chains and are considered a priori to be hybrid, discrete or continuous. The 3 kinds of data must be proceeded by the operational FDIR in order to track the satellite's state. But due to the limited on-board resources and in order to reduce the FDIR algorithms computation's need, an abstraction of the continuous and hybrid data to discrete is performed. Abstracted discrete data are easier to compute and often require less computing resource. In addition, by using such an abstraction into the discrete level it is possible to cope with the complexity of the whole system.

From a theoretical point of view, the discrete events used for performing the diagnosis and the monitoring of the satellite's state are based on a diagnoser concept (Pencole et al., 2006). The diagnoser represents the different states that can be reached by the satellite. Only observable variables (event, alarm, residual switches, ...) are used in the diagnoser, allowing one to remove the non observable part of the design model but leading to ambiguity in some cases.

In addition, the diagnoser is used to determine the diagnosability degree of the faults set. By checking the non discriminable faults present in a diagnoser's state, we can determine the observables which have to be added to achieve a complete faults isolation.

The diagnoser uses only discrete events and commands as input. So all the data coming from the functional subsystems have to be abstracted to discrete events. The next section gives some illustration of FDIR techniques dedicated to a given kind of functional subsystems and the way the data are abstracted to events.

4. PROCEEDING ON-BOARD DECISION

The figure 1 shows an architecture designed for allowing to take and execute on-board decision. 2 executive components are present. At decisional level, its role is to engage, at the right time, the actions decided by the planners (mission activity, downlink, …). At operational level, its role is to refine the activity to perform, in a set of elementary actions (similar a TC from Ground).

OBCP (On-Board Control Procedures) (Garcia *et Al.*, 2004) are a mean to define some control procedures, to permit the execution of some activities. The engine used to run this procedure has a behaviour close to real-time. It constitutes an opportunity to enhance the autonomy of the satellite.

Another way is to use some basic functionalities of current satellite. It is based on a way to temporally execute a sequence of telecommands. This kind of sequences permits to have some reflex-actions, as an answer to an event.

Whatever the means used to engage some actions, the on-board decision permits to increase the control of the satellite, by raising the actions on a time where it will have the less influence on the current actions plan. This can only be the case is the satellite is able to provide its current belief state. It is coming from operational FDIR, which constitutes a unique place for this knowledge.

5. CONCLUSIONS

Future autonomy capabilities will require the integration of planning systems with FDIR and control loop. We have presented an example of architecture embedding several planners to take on-board decision, and propose a generic executive to achieve their coordination between on-board decision to engage and current belief state.

The proposed FDIR strategy is compliant with the current one used by Thales Alenia Space, which has been proven by flight, and with the next generation of architectures for autonomous satellites. The centralized part is mandatory to offer the opportunity to have on-board decision making. The distributed part allows one to take into account the industrial constraints dealing with the minimization of mission outage and the required robustness.

In addition, the use of a diagnoser allows us to give flexibility in the selection of the FDIR algorithms for the functional subsystems. This is permitted by making an abstraction of the data in the form of discrete events.

Such a strategy allows us to cope with the complexity of autonomous satellites. This kind of satellites embeds more functionalities and has a larger state space than the classical one. This latter point is due to the increase of the autonomous on-board reactions, and thus the number of possibilities for the satellite contexts.

As a perspective, the next work we intend to perform will deal with the decisional level. Currently, the FDIR is mainly curative. It is activated after a fault occurrence and covers the operational and functional levels. At decisional level, the FDIR is a preventive one. Its role is to prevent the occurrence of faults on-board. To achieve this goal, several monitoring have to be set up: monitoring of command consistency (commands can come from Ground or be decided on-board by planners, for instance) and anticipation of fault occurrences. The current state of the satellite inside the diagnoser can have only paths leading to faulty states. In this case it seems interesting to introduce commands in order to avoid reaching the faulty states. This technique is often called active diagnosis. In addition is can be used to inject commands to highlight the symptoms and has for goal to make easier the isolation phase.

Finally some new mission features are requiring a particular attention with regard to the control, like the rendezvous mission phase, the docking and the landing. Both first deals with collision avoidance and target acquisition, and the last one with hazard avoidance. The new mission features require some more accurate control, and some quicker control loop to manage the most critical mission phase, like vision based landing.

REFERENCES

Bonasso R.P., Firby J., Gat E., Kortenkamp D., Miller D.P. and Slack M.G. (1997), Experiences with an Architecture for Intelligent, Reactive Agents, *Journal of Experimental & Theoretical Artificial Intelligence,* Vol. 9 (2/3).

Alami R., Chatila R., Fleury S., Ghallab M. and Ingrand F. (1998), An Architecture for Autonomy, *International Journal of Robotics Research, Special Issue on Integrated Architectures for Robot Control and Programming*, Vol. 17(4).

Muscettola N.,Nayak P.P., Pell B. and Williams B. (1998) , Remote Agent : To Boldly Go Where No AI System Has Gone Before, *Artificial Intelligence*, Vol. 103.

Chittaro L., Guida G., Tasso C., and Toppano E. (1993), Functional and Teleological Knowledge in the Multimodeling Approach for Reasoning About Physical Systems: A Case Study in Diagnosis, *IEEE Transactions on Systems, Man, and Cybernetics,* Vol. 23, N° 6.

Bayoudh M., Travé-Massuyès L. and Olive X. (2008), Hybrid Systems Diagnosis by coupling Continuous and Discrete event Techniques, *Proceedings of the 17th World Congress The International Federation of Automatic Control,* Seoul, Korea, July 6-11.

Pencolé Y., Kamenetsky D. and Schumann A. (2006), Towards low-cost fault diagnosis in large component based systems, *6th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (SAFEPROCESS-06),* Beijing (P.R. China).

Garcia G., Roubion C. and Prunier S. (2004), Java as a standardized on-board control procedures platform ?, *Actes de DAta Systems In Aerospace (DASIA*), Nice, France.