FDIR Strategies For Autonomous Satellite Formations - A Preliminary Report

Charles Castel, Jean-François Gabard and **Catherine Tessier**
ONERA-DCSD, Toulouse, France - {castel,gabard,tessier}@onera.fr

Bertrand Laborde and **Raymond Soumagne**
CNES, Toulouse, France - {laborde,soumagne}@cnes.fr

Abstract

This paper focuses on FDIR (Fault Detection, Isolation and Recovery) strategies for autonomous satellite formations. Anomalies that impact the formation geometry, the scientific mission and communications are considered. Three centralised, one mixed and two distributed strategies are characterised in terms of knowledge, algorithm and communication requirements. Preliminary Petri net - based simulations allow the dynamics of the spacecraft states and the FDIR task progress to be shown and the spacecraft communication links to be displayed. First results on the robustness of the strategies wrt an anomaly that would affect the FDIR itself or the communication links are presented.

Introduction

The autonomous formation flying of multiple spacecraft to replace a single large satellite will be an enabling technology for a number of future missions. Potential applications include synthetic apertures for surveillance and high-resolution interferometry missions, or for taking widespread field measurements for atmospheric survey missions (Cranfield).Very precise autonomous coordination and control differentiate formations from constellations. The challenge is to develop both the software and the hardware to allow separate, unconnected spacecraft to function as if they were a single, solid structure (Nasa). Spacecraft within a formation may be different from one another and the different parts of one instrument may be distributed among several spacecraft.

FDIR (Fault Detection, Isolation and Recovery) is the means to detect off-nominal conditions, isolate the problem to a specific subsystem/component, and recover of vehicle systems and capabilities (NASA 2005). Formation flying brings a new concept in FDIR, i.e. the *formation* has to be considered as an entity in itself. Indeed the scientific mission is performed by the formation (and not by the individual spacecraft). Therefore specific FDIR strategies have to be considered in order to deal with formation specific failures e.g. instrument failure, problems with the formation geometry, inter-spacecraft communication failures.

The paper is organised as follows : after a short section on typical anomalies in spacecraft formations, some hints to design FDIR strategies are suggested. Then the strategies we have designed are described with their requirements in terms of knowledge and algorithms, and communications. Simulations are then presented with first robustness results.

Typical Anomalies in Spacecraft Formations

This work focuses on anomalies stemming from the fact that autonomous spacecraft are organised as a formation. Consequently, individual spacecraft anomalies are not considered as such. Besides we will limit our study to a functional analysis of single anomalies affecting the main functions of the formation (though chains of consequences are considered).

Anomalies are considered in the various phases of the formation life: injection into the escape orbit (spacecraft in a stack or separately), transfer to the operational orbits, formation deployment with low then high accuracy and operational use. For the paper, let us consider anomalies in operational use. There are three classes of them:

- formation geometry anomalies : spacecraft collision risks, altered relative positions, altered orientations;
- degradation or loss of parts of the instruments;
- degradation or loss of the communication within the formation, or between the formation and the ground.

These anomalies do not impact the mission in the same way: geometry anomalies may be catastrophic for the spacecraft and immediate reactions have to be considered. Instrument anomalies may not affect the formation safety but may jeopardize the scientific goals of the mission, therefore replanning has to be considered. Inter spacecraft communication anomalies may have consequences on the scientific mission and on the safety of the formation, especially in two-spacecraft formations. What is more is that FDIR itself may be affected by communication anomalies.

In order to face different kinds of anomalies that may occur in different cases of formations, e.g. two-spacecraft formations and more-than-two-spacecraft formations, we have considered various FDIR strategies.

How to Design FDIR Strategies

State of the art

Very little literature concerning FDIR for spacecraft formation is available. Nevertheless supervision and agent modelling are two key elements that are considered.

(Zetocha 2000) focuses on onboard supervision for formation situation assessment, resource optimisation and formation - ground interface. Several questions are raised: should supervision be centralised or distributed? How should knowledge be shared between spececraft? How should spacecraft data be aggregated? How should a semantic protocol be designed between spacecraft and between the formation and the ground? Formation supervision may be implemented on a single spacecraft, on a subset of spacecraft or on all spacecraft. This is close to the ObjectAgent cluster manager in (Sherwood *et al.* 2001). Centralised supervision is easier to implement but may create a single point of failure. Moreover each spacecraft must be able to communicate its state to the supervisor spacecraft and conversely to get and execute the orders.

A distributed approach is suggested in (Guettier & Poncet 2001) with each spacecraft considered as an agent with its own perception - decision - reaction loop. A leader agent is designated by the formation for mission management and goal allocation, and the other agents manage their local plan to meet their own goals. Such a distributed supervision is more difficult and expensive to implement, but it is more fault-tolerant. The same kind of approach is also considered in (Bonnet-Torrès & Tessier 2005) for replanning after the occurrence of a disruptive event in a robot team.

A whole multiagent architecture is described in (Schetter, Campbell, & Surka 2003): at the formation level, agent `DecMakFailAgent` monitors the spacecraft health, detects failures and triggers formation reconfiguration. Agent `PlanReconfAgent` optimises spacecraft positions after a failure to maximise scientific return.
At the spacecraft level, four agent levels are defined:

- I4: the spacecraft can receive and execute orders and tasks from the other agents and from the ground; this is the state of the art for most spacecraft today;
- I3: the spacecraft can plan and re-plan locally; it is the case for EO-1 (Tran *et al.* 2005);
- I2: the spacecraft can interact with the other spacecraft within the formation (e.g. to coordinate or negotiate in case of conflictual requests), which needs at least a partial knowledge on the other agents;
- I1: the spacecraft can manage and plan for all the others within the formation, which needs a full knowledge on the others; for example, the spacecraft may compute a new formation configuration and allocate new relative positions.

Four possible architectures are suggested:
(1) master-slave coordination (an I1-agent plans for all the formation and the other I4-agents execute orders);
(2) centralised coordination (an I1-agent selects the best plans among those suggested by I2 or I3 - agents);
(3) distributed coordination (several I1-agents are implemented, the others are I2 or I3 - agents);
(4) each agent is an I1-agent.
These architectures are a basis for the strategies we are decribing in this paper.

Finally an important issue is dealt with in (McQuade *et al.* 2001), i.e. the fact that a spacecraft can or cannot perform reconfiguration after a failure. For example, in case of a collision risk between two spacecraft, the case when both of them can manoeuvre and the case when only one of them can are quite different.

Basic Features

Several basic features are worth highlighting when designing FDIR strategies for autonomous spacecraft formations:

- spacecraft collaborate within the formation; indeed they have been designed to fulfil a common scientific goal;
- consequently, the formation in itself can be regarded as a decision agent for FDIR;
- FDIR must be embedded in the global autonomy architecture; it is not an independent function and must be part of the various autonomy functions: Detection (D) belongs to situation monitoring, Isolation (I) to situation assessment and Recovery (R) to reaction and planning;
- the FDIR strategy must be designed in accordance with the type of formation: master-slave formation, homogeneous formation, etc.

Strategies

Centralised, mixed and distributed strategies are put forward, so as a reference individual strategy. Each one is characterised by the D, I and R knowledge and algorithms that are necessary and where they have to be implemented, and the communication requirements.
The following notations will be used:
F: the set of the spacecraft within the formation, $F = \{S_1, ..., S_N\}$;
N: the number of spacecraft within the formation, $N = |F|$;
A: the set of the formation anomalies a_k,
$A = \bigcup A_{T_n} = A_{T_{geom}} \cup A_{T_{instr}} \cup A_{T_{comm}} \cup A_{T_{others}}$ with:
$A_{T_{geom}}$ the subset of the geometry anomalies,
$A_{T_{instr}}$ the subset of the instrument anomalies,
$A_{T_{comm}}$ the subset of the communication anomalies,
$A_{T_{others}}$ the subset of the possible other anomalies;
A_i : the subset of the anomalies that spacecraft S_i may detect;
$D_{S_i}^{a_k}$: Detection function on spacecraft S_i for anomaly a_k (same definitions for $I_{S_i}^{a_k}$ (Isolation) and $R_{S_i}^{a_k}$ (Recovery));
Protocol: protocol for common situation awareness within the formation;
C_{gr} : communications between the current spacecraft and the ground;
C_{S_i}: communications between the current spacecraft and spacecraft S_i.

Individual Strategy

S-indiv: each spacecraft performs its own FDIR without communicating with the others (figure 1).
Requirements:

$\forall S_i \in F, S_i \rightarrow \left(\bigcup_{a_k \in A_i} D_{S_i}^{a_k}, \bigcup_{a_k \in A_i} I_{S_i}^{a_k}, \bigcup_{a_k \in A_i} R_{S_i}^{a_k}, C_{gr} \right)$: each spacecraft S_i carries all the knowledge and algorithms for the D, I and R functions, and each spacecraft communicates with the ground (arrows toward ground on figure 1).

Indeed it is not really a formation FDIR since there is no

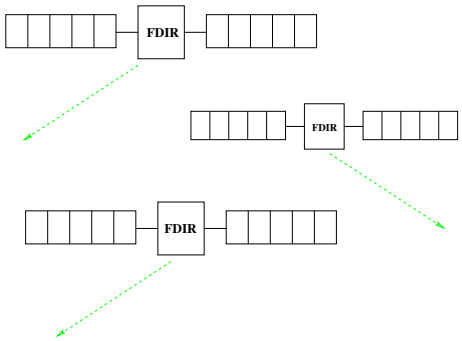

Figure 1: Individual strategy

communication between spacecraft. Nevertheless we keep it as the reference strategy.

Centralised Strategies

Master Centralised Strategy **S-centr-mast**: one of the spacecraft (always the same one) within the formation - S_{FDIR} - performs FDIR for the whole formation (figure 2).
Requirements: $\exists S_j \in F, S_j = S_{FDIR}, S_j \rightarrow$

$\left(\bigcup_{a_k \in A, S_i \in F} D_{S_i}^{a_k}, \bigcup_{a_k \in A, S_i \in F} I_{S_i}^{a_k}, \bigcup_{a_k \in A, S_i \in F} R_{S_i}^{a_k}, C_{gr}, C_{S_i, \forall i \neq j} \right)$

and $\forall S_i \in F, i \neq j, S_i \rightarrow (C_{S_j})$: S_{FDIR} carries all the spacecraft knowledge and algorithms for the D, I and R functions, S_{FDIR} communicates with all the other spacecraft and conversely (double arrows on figure 2), and S_{FDIR} communicates with the ground.

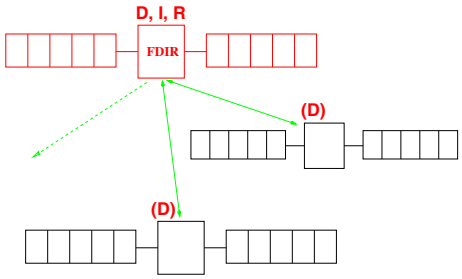

Figure 2: Master centralised strategy

Opportunistic Centralised Strategy **S-centr-opp**: one of the spacecraft (the spacecraft that can better deal with the current anomaly) - S_{FDIR} - performs FDIR for the whole formation (figure 3). An example is **S-centr-opp-expert**: S_{FDIR} is the spacecraft that is skilled at dealing with the type of the current anomaly.

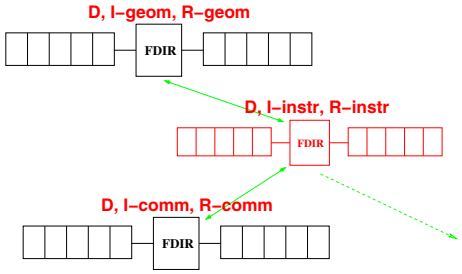

Figure 3: Opportunistic centralised strategy

Requirements: let $S_{FDIR}^{T_n}$ be a spacecraft that carries the expertise for T_n-type anomalies (e.g. $S_{FDIR}^{T_{geom}}$ is an expert for geometry anomalies). Other spacecraft are simply noted S_i. $\forall T_n, S_{FDIR}^{T_n} \rightarrow$

$\left(\bigcup_{a_k \in A_{T_n}} D_{S_{FDIR}^{T_n}}^{a_k}, \bigcup_{a_k \in A_{T_n}} I_{S_{FDIR}^{T_n}}^{a_k}, \bigcup_{a_k \in A_{T_n}} R_{S_{FDIR}^{T_n}}^{a_k}, C_{gr}, C_{S_i, \forall S_i} \right)$:

$S_{FDIR}^{T_n}$ carries all the FDIR knowledge and algorithms for anomalies it is an expert of, and $S_{FDIR}^{T_n}$ communicates with the ground and with all the other spacecraft.

$\forall S_i, S_i \rightarrow \left(\bigcup_{a_k \in A_i} D_{S_i}^{a_k}, C_{S_{FDIR}^{T_n}} \right)$: S_i carries the D knowledge and algorithms for anomalies it is sensitive to. Moreover, it must be able to characterise the type of the anomaly and send it to the expert.

Global Centralised Strategy **S-centr-glob**: S-centr-mast or S-centr-opp is applied only at the global *formation* level and each spacecraft manages the local projections of the global FDIR individually (figure 4).
Example: a global reconfiguration order *reset instrument* from S_{FDIR} is interpreted at the local S_i level as e.g. *reset detector mirror position*.

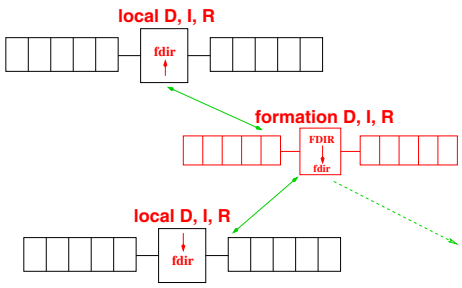

Figure 4: Global centralised strategy

Mixed Strategy

S-mix: FDIR is integrated at each equipement or functional level. FDIR is therefore hierarchical and is performed "close to" the anomaly: the first level is the equipment level, then the spacecraft function level, then the formation sub-set functional level and finally the global formation level (figure 5).
Requirements:
(1) for equipements and functions that are implemented on only one spacecraft: the requirements are the same as for S-centr-opp;
(2) for equipements and functions that are shared between several spacecraft: any centralised or distributed strategy may be implemented on this sub-formation.

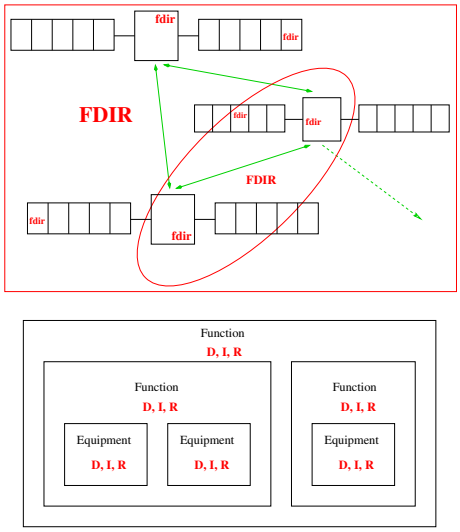

Figure 5: Mixed strategy and hierarchical FDIR

Distributed Strategies

Common Distributed Strategy S-distrib-comm: data are shared - according to a predefined *protocol* - so as to elaborate a common situation assessment and a common recovery decision from local points of view (figure 6). For example, a common conjecture elaboration process such as (Fiorino 1998) may be implemented.
Requirements: this strategy inherits the reference strategy S-indiv. At best each spacecraft carries all the knowledge and algorithms for D, I, R:
$\forall S_i \in F, S_i \rightarrow$

$$\left(\bigcup_{a_k \in A_i} D_{S_i}^{a_k}, \bigcup_{a_k \in A_i} I_{S_i}^{a_k}, \bigcup_{a_k \in A_i} R_{S_i}^{a_k}, Protocol, (C_{gr}), C_{i+1} \right),$$

or some spacecraft have less comprehensive knowledge and it will be expanded during common situation and solution elaboration. At least each spacecraft must be able to communicate with its neighbour and at least one spacecraft within the formation must be able to communicate with the ground.

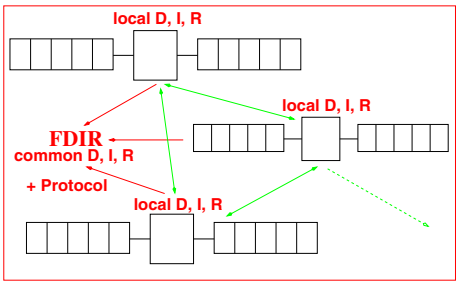

Figure 6: Common distributed strategy

Individual Distributed Strategy S-distrib-indiv: each spacecraft performs its own FDIR from its local point of view with the goal of a global formation FDIR. A protocol (e.g. a voting protocol) has to be implemented to check the consistency of the locally elaborated FDIR (figure 7).
Requirements: this strategy also inherits the reference strategy S-indiv but in this case, each spacecraft must carry all the knowledge and algorithms for the three FDIR functions, $\forall S_i \in F, S_i \rightarrow$

$$\left(\bigcup_{a_k \in A_i} D_{S_i}^{a_k}, \bigcup_{a_k \in A_i} I_{S_i}^{a_k}, \bigcup_{a_k \in A_i} R_{S_i}^{a_k}, Protocol, (C_{gr}), C_{i+1} \right).$$

At least each spacecraft must be able to communicate with its neighbour and at least one spacecraft within the formation must be able to communicate with the ground.

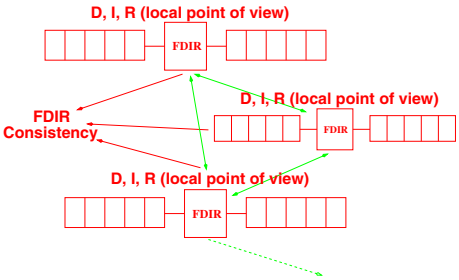

Figure 7: Individual distributed strategy

Simulations

Approach

Preliminary functional simulations have been carried out with ProCoSA (see Appendix), a Petri net - based supervision and control tool dedicated to highly autonomous vehicles. Simulations aim at (1) checking that the strategies cannot reach deadlocks and (2) showing the consequences of FDIR failures (i.e. spacecraft failures that would damage the FDIR function itself) and inter-spacecraft communication failures. The approach is as follows:

- three states are represented for each spacecraft: nominal, anomaly, recovery (*state-S_i* Petri nets);
- FDIR is represented in *FDIR-S_i* Petri nets by states: nominal-formation, D (detection), I (isolation), R (recovery elaboration) and a reaction state allowing the formation to be secured (e.g. with a distancing manoeuvre) while recovery is elaborated;
- communication links between spacecraft are represented.

Examples

Master Centralised Strategy **S-centr-mast** is simulated with a two-satellite formation without loss of generality. Let S_1 be S_{FDIR}.

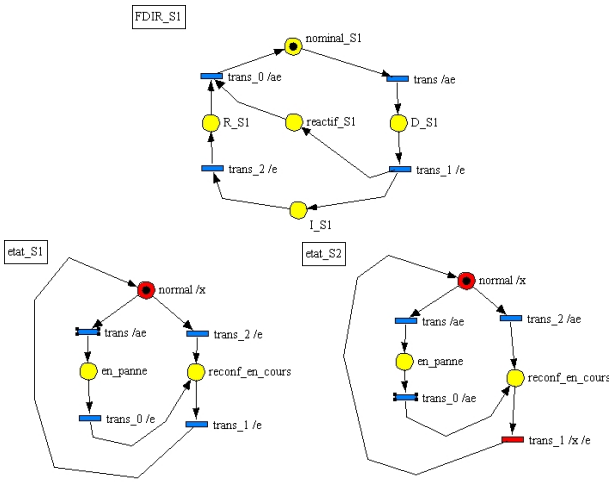

Figure 8: S-centr-mast with ProCoSA

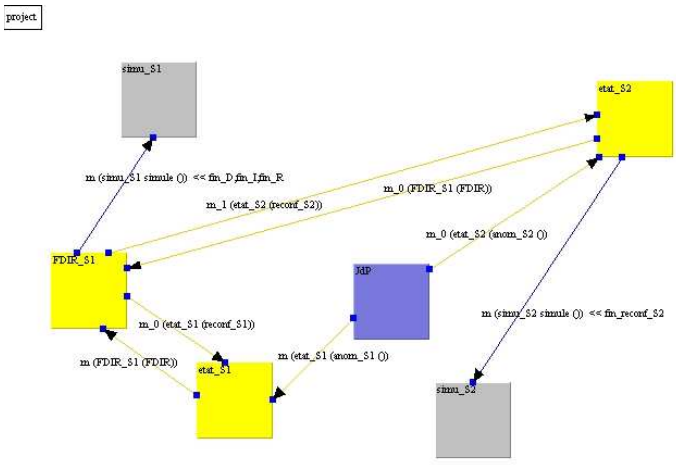

Figure 9: Communication links for S-centr-mast

(See figure 8) An anomaly within the formation makes one of the *state-S_i* (*etat-S_i*) Petri net pass from the nominal (*normal*) to the anomaly state (*en_panne*) whereas *FDIR-S_1* net passes from the nominal to the D_S1 state. Then *FDIR-S_1* passes to state $(I\_S1, reactifS1)$ meaning that a security reaction and Isolation are performed at the same time. Indeed, as *FDIR-S_1* carries all the FDIR knowledge and algorithms, only *FDIR-S_1* can perform a reaction, even if the anomaly is detected on another spacecraft. *FDIR-S_1* then passes to state $(R\_S1, reactifS1)$ meaning that the actions necessary to recovery are computed while the reaction goes on. Then each *state-S_i* net passes to state *on-going reconfiguration* (*reconf_en_cours*), then back to nominal, while *FDIR-S_1* goes back to the nominal state.

The global view of the ProCoSA project (figure 9) shows the communication links between *FDIR-S_1* and *state-S_i*: S_i sends anomaly events to *FDIR-S_1* and *FDIR-S_1* sends reconfiguration orders to S_i. The grey boxes represent the state simulation processes of S_i.

Common Distributed Strategy **S-distrib-comm** is simulated with a three-satellite formation without loss of generality. Each spacecraft has the same role for this strategy therefore they are represented the same way (figures 10 and 11):

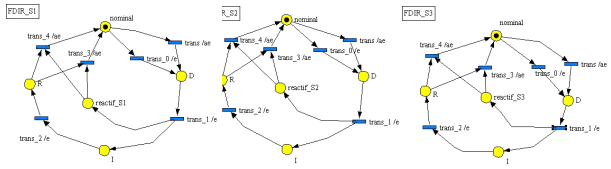

Figure 10: FDIR nets for S-distrib-comm

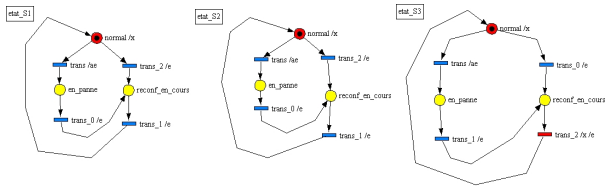

Figure 11: state nets for S-distrib-comm

For each spacecraft S_i, the protocol is implemented with two Petri nets (figure 12):

- *FDIR-S_i-niv-1* always listens to anomaly (transition *recoit_anom*) or protocol (transition *recoit_mess*) messages;
- *FDIR-S_i-niv-2* allows to: deal with a new anomaly or with a new protocol message; postpone the processing of an anomaly if a protocol is already running until it receives the new situation assessment; deal with protocol message arrivals when another protocol is running; end protocol when a consensus is reached (Fiorino 1998).

The global view of the ProCoSA project (figure 13)

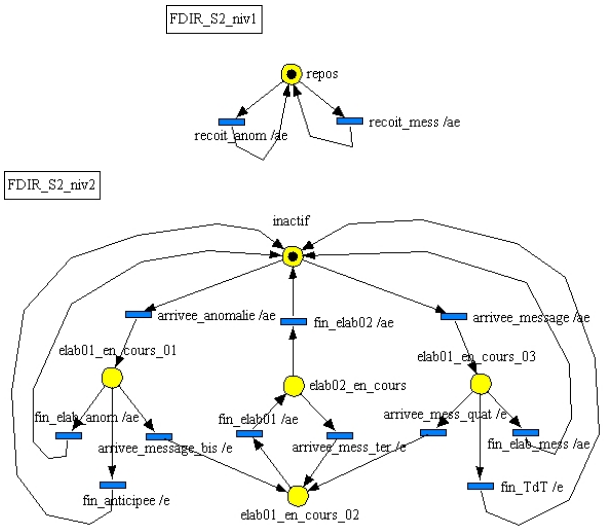

Figure 12: Protocol for S-distrib-comm - S_2

shows the messages that are sent and received by spacecraft S_i:

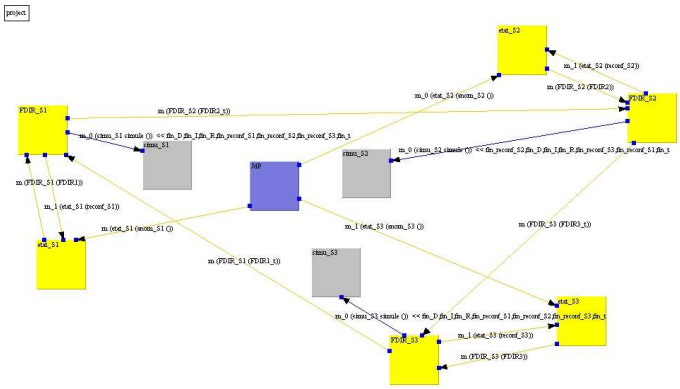

Figure 13: Communication links for S-distrib-comm

Some Results

First analyses and simulations have given some hints about strategy robustness wrt (1) failures that would damage the FDIR function itself and (2) intra-formation communication failures.

Robustness wrt FDIR Failure Table 1 shows that centralised strategies are less robust to FDIR failures than distributed strategies. Nevertheless robustness could be enhanced if each S_i carries redundant FDIR knowledge and algorithms and can communicate with the other spacecraft. In return the cost would be higher.
As far as distributed strategies are concerned, they are all the more robust to FDIR failures as they are implemented in larger formations, e.g. a eight-satellite formation like Darwin (Bagnasco & El Hamel 2001).

Strategy	FDIR failure	Effect
S-indiv	on S_i	S_i cannot perform its own FDIR
S-centr-mast	on S_{FDIR}	formation FDIR is lost
S-centr-opp	on $S_{FDIR}^{T_n}$	formation FDIR for T_n anomalies is lost
S-centr-glob	on S_{FDIR}	no more global FDIR, local OK
S-mix	on subsystem	implicit FDIR redundancy at lower or higher levels
S-distrib-comm	on S_i	OK if protocol allows for missing S_i
S-distrib-indiv	on S_i	OK if protocol allows for missing S_i

Table 1: Effects of an FDIR Failure

Robustness wrt Intra-Formation Communication Failures In table 2, $S_i \leftrightarrow S_j$ means "communication between spacecraft S_i and S_j".
As far as centralised strategies are concerned, degraded cases exist beside total loss of FDIR: (1) S_{FDIR} cannot receive one S_i telemetry, then D and I are performed without S_i data; (2) one S_i cannot receive S_{FDIR} telecommand, then R is performed without S_i.
As far as distributed strategies are concerned, the worst case is when no more communication is available: as spacecraft carry their own knowledge and algorithms, those strategies are degraded in the reference strategy S-indiv.

Strategy	Comm failure	Effect
S-indiv	-	-
S-centr-mast	$S_{FDIR} \leftrightarrow S_i$	formation FDIR is lost
S-centr-opp	$S_{FDIR}^{T_n} \leftrightarrow S_i$	formation FDIR for T_n anomalies is lost
S-centr-glob	$S_{FDIR} \leftrightarrow S_i$	no more global FDIR, local OK
S-mix	$S_i \leftrightarrow S_j$ shared subsystem	implicit FDIR redundancy at lower or higher levels
S-distrib-comm	$S_i \leftrightarrow S_{i+1}$	OK if $S_i \leftrightarrow S_j$ OK or protocol allows for missing S_i
S-distrib-indiv	$S_i \leftrightarrow S_j$	OK if protocol allows for missing S_i

Table 2: Effects of Communication Failures

Conclusion

FDIR strategies for autonomous satellite formations have been designed and simulated and first results on the robustness of the strategies wrt a failure that would affect the FDIR itself or the intra-formation communication links have been presented.

On-going work focuses on assessing the different strategies on different cases, i.e. different kinds of anomalies within different kinds of formations (e.g. two satellites, three satellites and four aligned satellites), during different flight phases. The aim of this work is to determine the best suited strategies for triplets (type of formation, type of anomaly, flight phase) so as to select the "best" strategies to be embedded in a global autonomy architecture. Indeed Detection and Isolation belong to the situation monitoring and assessment process and Reconfiguration is part of the planning-replaning process therefore FDIR should not be implemented as a separate function. A specification of FDIR strategies within a whole autonomy architecture for spacecraft formation is currently being considered.

Appendix: ProCoSA

A Petri net $< P, T, F, B >$ is a bipartite graph with two types of nodes: P is a finite set of places; T is a finite set of transitions (David & Alla 2005). Arcs are directed and represent the forward incidence function $F : P \times T \rightarrow \mathbb{N}$ and the backward incidence function $B : P \times T \rightarrow \mathbb{N}$ respectively. The marking of a Petri net is defined as function $\mathcal{M} : \mathcal{P} \rightarrow \mathbb{N}$: tokens are associated with places. The evolution of tokens within the net follows transition firing rules. Petri nets allow sequencing, parallelism and synchronization to be easily represented. An *interpreted Petri net* is such that conditions and events are associated with transitions.

ProCoSA (Barbier *et al.* 2006) is a software environment meant for controlling and monitoring highly autonomous systems. System autonomy is usually obtained by putting together various functions, among which: data analysis (sensor data, monitoring data, operator's inputs), nominal mission monitoring and control (vehicle and payload control actions), decision (management of disruptive events, replanning). These functions, which are often developed as separate subsystems, have to co-operate in order to fulfil the autonomous system behaviour requirements for the specified missions. More precisely, the needs are the following:

- off-line tasks: specification of the co-operation procedures between subsystem software; subsystem coding for embedded operation;
- on-line tasks: procedure monitoring, event monitoring, and management of the dialog with the operator.

ProCoSA includes the following components:

- EdiPet, a graphical interface for Petri nets which is used both by the developer for procedure design and by the operator for execution monitoring;
- JdP, the Petri net player, that executes the procedures, fires the event-triggered transitions of the Petri nets and synchronises the activation of the associated sub-system functions; a socket-based communication protocol allows data to be exchanged with external subsystem software;
- Tiny, a Lisp interpreter dedicated to distributed embedded applications.

The Petri nets used by ProCoSA are interpreted Petri nets: triggering events such as activation or event generation requests are attached to the transitions. Timers can be programmed: a special activation request enables a timer variable to be instantiated, which allows actions with a limited duration to be modelled.

The ProCoSA procedures are used to model the desired behaviours of the autonomous system; the hierarchical modelling features offered by ProCoSA enable to structure the whole application in a generic way: at the highest description level, generic behaviours can be described, regardless of the characteristics of a given vehicle; at the lowest level, they specify the sequences of elementary actions to be performed by the vehicle or the payloads; this modular approach enables a quick adaptation to system changes (e.g. taking into account a new payload).

An important feature of ProCoSA lies in the fact that there is no code translation step between the Petri net procedures and their execution: they are directly interpreted by the Petri net player, thus avoiding any supplementary error causes.

ProCoSA finally includes a verification tool, which makes use of the Petri net analysis techniques to check that some "good" properties are satisfied by the procedures, both at the single procedure level and at the whole project level (that is to say taking into account inter-net connections); the following properties are checked: place safety (not more than one token per Petri net place), detection of dead markings (deadlocks), detection of cyclic firing sequences (loops).

References

Bagnasco, G., and El Hamel, Z. 2001. Technology R&D programme support to future ESA science mission. *ESA Bulletin* 18:49–57.

Barbier, M.; Gabard, J.-F.; Vizcaino, D.; and Bonnet-Torrès, O. 2006. ProCoSA: a software package for autonomous system supervision. In *CAR'06 - First Workshop on Control Architectures of Robots*.

Bonnet-Torrès, O., and Tessier, C. 2005. From teamplan to individual plans: a Petri net-based approach. In *AAMAS'05*.

Cranfield, U. Satellite formation flying for an interferometry mission
`.http://www.cranfield.ac.uk/soe/space/flying.htm`.

David, R., and Alla, H. 2005. *Discrete, continuous, and hybrid Petri nets*. Springer.

Fiorino, H. 1998. Principles for a cooperative conjecture elaboration in a multiagent context. In *ECAI'98*, 315–316.

Guettier, C., and Poncet, J.-C. 2001. Multi-levels planning for spacecraft autonomy. In *ESA International workshop on planning and scheduling for space*.

McQuade, F.; Ward, R.; Ortix, F.; and McInnes, C. R. 2001. The autonomous configuration of satellite formations using generic potential functions. In *IAF congress*.

Nasa. Technology - formation flying . http://planetquest.jpl.nasa.gov/technology/formation_flying.cfm.

NASA. 2005. Glossary - NASA Crew Exploration Vehicle, SOL NNT05AA01J, Attachment J-6. http://www.spaceref.com/news/viewsr.html?pid=15201.

Schetter, T.; Campbell, M.; and Surka, D. 2003. Multiple agent-based autonomy for satellite constellations. *Artificial Intelligence* 145:147–180.

Sherwood, R.; Chien, S.; Burl, M.; Knight, R.; Rabideau, G.; Engelhardt, B.; and Davies, A. 2001. The Techsat-21 autonomous sciencecraft constellation demonstration. In *iSAIRAS*.

Tran, D.; Chien, S.; Sherwood, R.; Castano, R.; Cichy, B.; Davies, A.; and Rabideau, G. 2005. Demo: the autonomous sciencecraft experiment onboard the EO-1 spacecraft. In *AAMAS'05*.

Zetocha, P. 2000. Satellite cluster command and control. In *IEEE Aerospace 2000 Conference*.