MISSION MANAGEMENT SYSTEM FOR PACKAGE OF UNMANNED COMBAT AERIAL VEHICLES

J. Baltié*, E. Bensana, P. Fabiani**, J.L. Farges**, S. Millet***, P. Morignot*, B. Patin***, G. Petitjean*, G. Pitois* and J.C Poncet***

** Axlog Ingéniérie, ** ONERA, ***Dassault Aviation*

Abstract: This paper presents the development and the assessment of a mission management system (MMS) for a package (group of aircraft assuming together a common mission) of Unmanned Combat Aerial Vehicles (UCAV). The mission is carried out in an environment including a safe area and a dangerous area, no-flying-zones, a command and control centre, the terrain, threats and targets. The MMS architecture presents reactive and deliberative layers. It includes a planner that distributes the plan computation effort among the UCAV. This plan takes into account the co-ordination and collaboration necessary for the package and its associated assets to implement the mission. Simulation results indicate that the MMS is able to carry on missions, to activate contingent behaviours, to decide whether or not to plan and to re-plan.

Keywords: Aircraft operations, architectures, autonomous vehicles, constraint satisfaction problems, decomposable searching problems, management systems, distributed artificial intelligence, planning, robotics, simulation.

INTRODUCTION

Some early tests with uninhabited aerial vehicles carrying and delivering weapons were conducted in the sixties and the seventies. However, Unmanned Combat Aerial Vehicles (UCAV) have been widely studied only since 1980. UCAV can be used for locating, identifying and destroying enemy targets. Multiple operators remotely operate currently one UCAV. UCAV demonstrators such as X-45 (Wise, 2003), X-47A and NEURON are developed. Nowadays, such studies as the NEURON demonstrator project envision leveraging effects by mixing these UCAV platforms with inhabited ones like RAFALE. For example, the sketch on Figure 1 shows a RAFALE collaborating with a package of two NEURON.

Such a vision lead researches towards an increase of UCAV and package of UCAV autonomy. For instance, Mehra *et al*. (2000) design a control scheme that supports autonomous co-ordinated flight of multiple UCAV. Other examples are the proposition by Li *et al*. (2002) of a hierarchical control scheme for a package of UCAV that provides path planning, trajectory generation and formation keeping and the development by Beard *et al.* (2002) of a target assignment method for a package of UCAV.

Fig. 1. A RAFALE collaborating with a package of two NEURON

UCAV decisional autonomy implies not only path planning and target assignment functions but also functions for planning and executing target and weapon selection, system reconfiguration, synchronisation actions, etc. Moreover, all those functions have to be integrated inside an UCAV sub-system dedicated to the management of the mission.

Barrouil *et al.* (1999) defined the scope and goals of what could a Mission Management System (MMS) do inside a mixed air patrol. Grounded on this work and European studies such as MISURE (Avalle and Patin, 2007), this paper presents the development and the assessment of a MMS giving decisional autonomy to a package of UCAV. This MMS includes a planner that allows on-line distributed computation of a new plan when a disruptive event occurs.

The first section of this paper is devoted to the presentation of the requirements for the development of the architecture. It includes the description of the environment of the mission and the presentation of the UCAV equipment the software has to interact with. Then the problems linked with architectural choices are addressed in the second section and the global architecture of the MMS is provided. This architecture includes reactive and deliberative layers that are described in the third and fourth sections respectively. The deliberative layer aims at solving, a planning problem including target selection, target assignment, weapon selection, path planning, collaboration and co-ordination aspects for a package of UCAV. Moreover, the problem is solved through its distribution over the UCAV. The issue of control of MMS computation time is addressed in the fifth section. The sixth section gives some experimental results about the plan computation and the MMS behaviour. Finally some conclusions are presented.

1. ENVIRONMENT AND REQUIREMENTS

1.1 The system

The environment of the package is presented on Figure 2. It includes a friend (or safe) area where the normal traffic management rules apply and a foe (or dangerous) area where the military authorities provides the air orders, No Flying Zones (NFZ), a Command and Control (C2) centre that supervises the package and provides eventually new information or new orders. It also includes the terrain, known and unknown threats, primary targets to be processed by the mission and secondary targets that are processed on an opportunity basis. This environment is dynamical and the perception of the environment by the package is also dynamical: threats may be discovered during the course of the mission. Tactical assets like targets may also be redefined by the C2.

Each UCAV of the package get a specific payload that may include several types of equipment:

- Weapons include bombs, missiles (WEAP) and laser designators (LD). Some type of weapons need to be guided using LD assets either on the same or another vehicle. Some other need the LD only to achieve a precision level.
- Localisation devices include satellite (GPS), radio (RS) and inertial (IN) positioning systems. The localisation information should be merged to produce continuously an accurate localisation of the aircraft.
- Flight management devices include engine control (ENG), autopilot (AP), fuel level sensors (FUEL) and flight control systems (FCS). They are developed and should be organised in order to provide relevant level of safety for such aircraft.
- Communication devices implementing intra-formation (IF), low bandwidth (LBW) and high bandwidth (HBW) data-links. The IF data-link is grounded on the medium frequency band and provides highest discretion but presents a limited range. The LBW data-link is grounded on lowest frequency band and presents the highest range but provides poor discretion. Finally the HBW data-link is grounded on highest frequency band and provides medium discretion but is dedicated only to upload of image necessary to designate a target by C2. The use of these links are constrained by mission dependant characteristics such as the will not to be seen or the safety of the package.
- Auto-protection devices include missile approach warners (MAW), radar warning receivers (RWR), chaffs (CHAFF), flares (FLARE), jammers (JAMM.) and active electronic counter measures (ECM). There is also the capability to control the Radar Cross Section (RCS) of the aircraft.
- Sensors include synthetic aperture radars (SAR) and electro-optical sensors (EO).

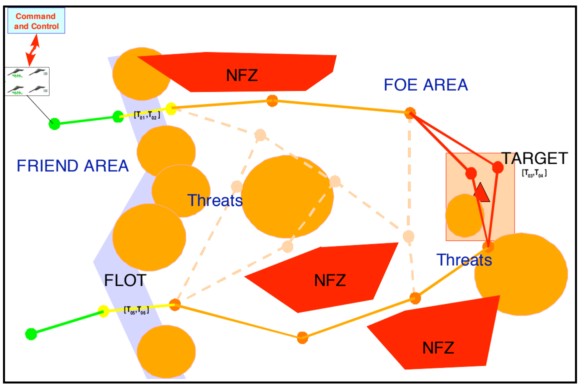

Fig. 2. Environment of the package of UCAV

Moreover, each UCAV is subject to flight dynamic constraints. Those constraints include:

- Maximum altitude,
- Speed limits,
- Load factor limits and
- Fuel consumption.

Finally, constraints appear due to the package work. Those constraints include:

- distance to respect in order to use discrete communications,
- relative positions of aircraft to gain benefit of the use of one auto-protection device,

- synchronisation issues and relative angles when laser designation is delivered by one aircraft and used by another aircraft to deliver weapons.

1.2 The problem

A mission plan is defined before take-off and the aims of the MMS are to follow the mission plan, to ensure safety, to ensure survivability and to ensure the success of the mission. Some requirements are deduced from those aims:

- the plan must be applied,
- disruptive events must be detected and analysed,
- if needed, reactive actions must be carried out and,
- if needed, the mission plan must be recomputed on-line.

Moreover, the flight dynamic constraints of the UCAV must be respected.

The package mission management problem includes more than the replication by the number of UCAV of a mission management problem for each agent. The set of actions that can be performed by a package of UCAV is larger than the one for a single UCAV. For instance the package can split; some UCAV fly to a convenient place to perform detection, identification and localisation of targets and other UCAV fly to another place to perform the strike itself. After the action the package can merge.

2. ARCHITECTURAL CHOICES

2.1 Approach

As pointed out by Findeisen and Lefkowitz (1969), there is two main ways of approaching control hierarchies:

- Control hierarchies with several levels: The system is decomposed in sub-systems. A controller is developed for each sub-system considering local criterion and local model. Higher level controllers integrate actions of lower level controllers in order to fulfil system objectives.
- Control hierarchies with several layers: The control problem is decomposed in sub-problems, for instance: reaction, optimisation, adaptation, auto organisation. Each sub problem is treated by a suited technique and the integration is made by the higher layers in order to solve the global problem.

Figure 3 present the application of the two approaches to the mission of a package UCAV. The decomposition of the system considered here could consist in splitting the mission in package and environment. The package is decomposed in different UCAV and the UCAV in different types of equipment. The decomposition of the problem in sub-problems highlights execution, disruptive event detection, disruptive event analysis, reaction and planning.

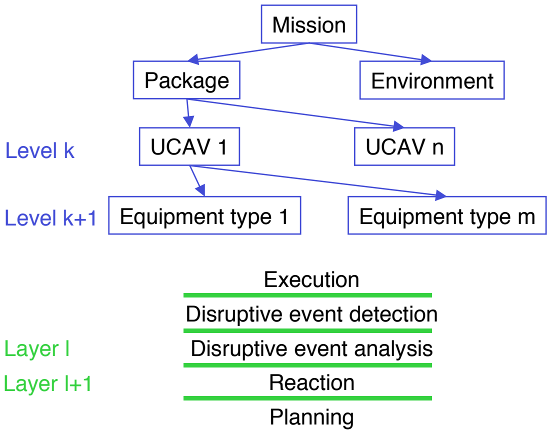

Fig. 3. Possible approaches to control hierarchies for the mission of a package of UCAV

The approach considered is grounded on both decompositions and also takes into account the fact that each UCAV has its own on-board computer and that there are communication constraints between UCAV. This approach stems from preceding studies (Riedel *et al.* , 2006; Avalle and Patin, 2007) where its efficiency was shown.

2.2 Levels of the hierarchy

As shown on Figure 4, the control hierarchy is organised in three levels that can be found on each UCAV. Each level is able to work with the corresponding level in the other UCAV. At the higher level, the MMS component is distributed among the UCAV and interacts with the other architecture components. The group of UCAV elaborates the different coupled plans using this distributed component. At an intermediate level, a component (TACSIT) is devoted to elaboration of a tactical situation. It is also distributed among the UCAV and merges different source of information to create the shared tactical situation. MMS and TACSIT have only information processing capability.

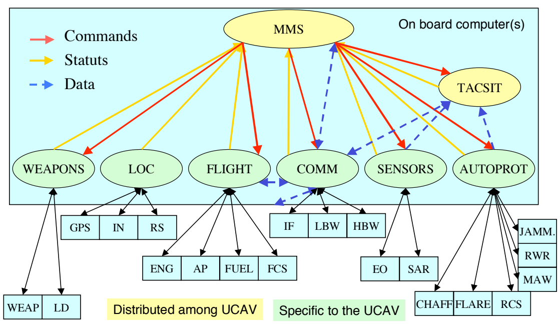

Fig. 4. Control hierarchy with three levels for the mission of a package of UCAV

At the lower level architecture components are able to control a specific part of the hardware of a specific UCAV. The functions of lower level components are localisation (LOC), flight management (FLIGHT), communication (COMM), self-defence (AUTOPROT), sensor management (SENSORS) and weapon management (WEAPONS). The details of the equipment controlled by each component are given in section 1.1. Equipment are only controlled by these components. There is no direct control by the MMS on the equipment. The rationale behind this decomposition is linked to the reality of the industrial organisation where aeronautic companies delegate components realisations to other companies. It also must be stated that the functional description of the lower level of the architecture is very similar to real military aircraft functional description.

From a practical point of view components are connected inside each UCAV through a software bus. Flows of information required to be able to implement a control of the package are numerous and of different types:

- Commands from the MMS to the other aircraft components,
- Status from the other aircraft components to the MMS,
- Data exchanged between components and,
- One very important asset, data between one UCAV communication component to another UCAV one.

2.3 Layers of the MMS

Existing architectures for MMS of autonomous vehicles are reactive or deliberative or both. Reactive architectures are not able to support problem solving but react quickly to events while deliberative architectures are fully based on problem solving and usually react slowly. The use of at least two layers to allow both behaviours is common in the literature since the work of Bresina and Drummond (1990). Practical intelligence of an UCAV consists in a mix of reactive and deliberative behaviours. The architecture is designed according to three principles:

1. The on-line planning shall be activated only when the current plan is invalidated by the current situation.
2. The reactive level shall not only execute the plan but also handle emergency situations.
3. Sensor inaccuracy is managed through pre-defined behavioural procedures for inaccuracy reduction.

Figure 5 presents the actual MMS architecture with reactive and deliberative layers. This architecture is organised using an underlying database that stores information about:

- the vehicle and the other vehicles,
- the known threats in the environment,
- the targets of the mission,
- the plan to carry out the mission and
- the configuration parameters and thresholds.

When considering the MMS database, it must not be confused with the TACSIT component. In fact, there is no semantics linked to the tactical situation described. It is influence map, list of objects, list of path (foe and friends) and the like. The MMS database is filled by its different parts interpreting data issued of the TACSIT component and giving sense to these data with respect to the concept manipulated by the MMS and its planning part.

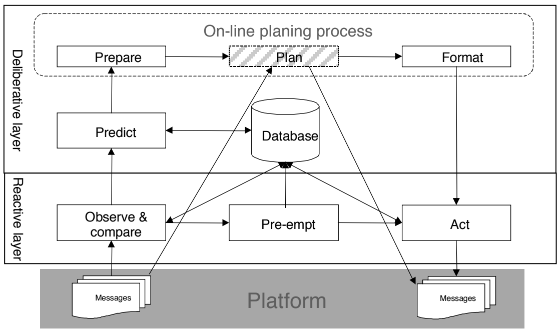

Fig. 5. Mission management system architecture with two layers

In this architecture, the purpose of the reactive layer is to execute the plan and to implement contingent behaviours for disruptive events that need quick reaction. The purpose of the deliberative layer is to determine if a new plan is needed and, in that case, to compute it.

3. REACTIVE LAYER

The reactive layer includes the following functions: “observe and compare”, “pre-empt” and “act”.

3.1 Observe and compare

The “observe and compare” function receives the messages from the platform, from other platforms and from C2 through the communication component. It updates the database information according to those messages and performs tests about the short-term situation. It tests the possible disruptive events such as unexpected exposure to a threat, approaching missile, presence in a NFZ, loss of communication and failure of a vehicle component. Most of the time, this function does not send messages in direction to “pre-empt” and “predict”. For the failure of a vehicle component a message is sent to “predict”, for other disruptive events a message is sent to “pre-empt”.

3.2 Pre-empt

The “pre-empt” function determines if necessary contingent behaviours and their unit actions. This function verifies the UCAV situation and uses a set of rules to determine the suited contingent behavior. If necessary, it computes the parameters of a contingent behaviour and activates it. Finally, the function computes the actions associated with the active contingent behaviours. Four behaviours may be activated by the function:

- The behaviour for new radar threat detection implements update of the radar list and radar locations, active electronic counter measure under specified conditions, manoeuvre for

avoidance or information gathering under other specified conditions.
- The behaviour for approaching missile detection implements unconditional use of active electronic counter measures, chaff and flare decoy and evasive manoeuvres.
- The behaviour for loss of communication between neighbouring vehicles consists in commanding the altitude of the vehicles to different pre-determined flight levels in order to ensure the absence of collision between them. The secured altitude slots are attributed to each aircraft at plan generation time, ensuring their uniqueness for each aircraft of a formation.
- The behaviour for NFZ violation avoidance is implemented in two parts. First, a modification of current trajectory is computed, attempting to avoid incoming NFZ, going round it by the shortest way. If this fails or if the situation evaluation reports that aircraft suddenly appears to be inside a NFZ, solution consists in a fast trajectory computing that will attempt to exit NFZ by crossing the closest frontier point.

All behaviours have a date parameter, a timeout and a homing way-point parameter. They are updated each time the "pre-empt" component is activated. This update can be null according to the current aircraft situation, meaning that the system can return to nominal plan execution. Otherwise, the contingency ends once its associated timeout has been passed out or when homing point is reached. All behaviours, except loss of communication, have a Boolean parameter indicating if the global path can be modified. Finally, the behaviour for new radar threat detection has an additional parameter indicating the origin way-point.

Four types of information represent the behaviours:
- The warning processing information indicates what should be done in terms of knowledge management.
- The system management information indicates what should be done in terms of auto-protection actions.
- The flight plan modification information indicates what should be done in terms of navigation actions.
- The end of contingency information indicates conditions for terminating the behaviour.

3.3 Act

The "act" function carries out the unit actions of the "format" or of the "pre-empt" function. It also determines if an action is correctly performed or not and if it is performed in time. For this, the function analyses the status of the sub-systems stored in the database. When an action is assumed finished, it triggers the next actions. In order to allow the flight management component to perform trajectories with appropriate turns, navigation actions are not treated individually, but by couples of successive actions.

Whenever actions of the UCAV are linked to actions of other UCAV they are to be done on conditions. This is the way to implement collaborative actions such as fire to shooter sequences.

Actions are scheduled according to their execution criteria:
- At a given instant,
- At a given location or,
- At an instant defined by the execution of another action:
  - Simultaneously,
  - Immediately after or,
  - After a given interval of time.

When actions of the plan and from "pre-empt" are conflicting, it always gives the priority to pre-emption actions. This mechanism corresponds to a suppressor function in the sense of Brooks (1986). Finally the function sends messages to the platform sub-systems.

4. DELIBERATIVE LAYER

The deliberative layer includes the following functions: "predict", "prepare", "plan" and "format".

4.1 Predict

The "predict" function assesses the feasibility of the on going plan and decides to compute a new plan or not. Most of the time, this function does not send message in direction to "prepare". The feasibility of the on going plan is assessed with respect to several criteria:
- Probabilities of UCAV survival and of target killing are updated and checked against a threshold.
- The possibilities of fulfilling time constraints at some waypoints and of having enough fuel to finish the mission are checked.

It should be noted that the "predict" component faces a dilemma. On one hand, deciding to re plan all the time leads the agent to an erratic behaviour, always starting the beginning of new unrelated plans, and therefore not leading to any goal at all (too often replanning). On the other hand, deciding to plan too scarcely leads the agent to follow unusable plans, since the behaviour of the agent does not adapt to what actually happens in the environment (too scarce replanning). The solution we propose for this "predict" component is a medium term on the previous spectrum, by using variables representing states of the agent. When the mean of these variables is above some threshold, then replanning decision is taken and replanning occurs. This solution is not satisfactory in principle, since it does not solves the problem of the continuity of behaviour of the agent over successive replanning activities. But at least it provides a practical and simple (but not elegant) solution, even if these variables and thresholds need careful tuning for realistic replanning frequency to be adopted. Moreover, the occurrence of a replanning request while replanning has to be managed. This management is performed using priorities on replanning reasons. If the priority of the reason of the present replanning request is lower than the one of the on going replanning, the request is ignored. Otherwise, the on going replanning activity is stopped and the replanning is started with a context including the present request.

4.2 Prepare

The "prepare" function gathers and generates data for the "plan" function. The data describes a planning context for a package and includes:

- The time at which the plan should began because problems are not stationary.
- The vehicles participating to be considered and their predicted state in terms of geometry and resources at the time at which the plan should began. Three classes of vehicles are to be distinguished:
  - The vehicles participating to the communication cluster in which the planning is carried on.
  - The vehicles not participating to the communication cluster but presenting a plan assumption.
  - The vehicles not participating to the communication cluster and assumed out of order.
- Relevant characteristics of the environment including threats, targets and NFZ. It includes the description of the airspace threatened by each threat.
- A graph, with nodes and edges, including possible paths for acquisition, attack and return to base. This graph is built in two steps. An initial graph is deduced from the initial mission plan by associating mission waypoints to graph nodes and transitions between these waypoints to graph edges. Nodes and edges are tagged according to their strategic properties for acquisition, shooting, etc. The second step is done each time the component is activated. It consists in the generation of different alternative paths for each strategic action, including Return To Base. These paths are generated using a potential field algorithm in which threats and NFZ are associated to repulsing potential while targets and base airport are associated with attractive potential.
- Time intervals and altitude constraints at waypoints.
- The goal-action prototypes in terms of resources to be used by the vehicles at specified places in the space and at specified times in order to achieve each goal.

4.3 Plan

The "plan" function allows a multi-UCAV distributed planning. It can exchange directly messages with other UCAV. It takes a planning context given by "prepare" and provides a plan to "format".

The resulting plan includes for each vehicle participating to the communication cluster a timed sequence of actions. There are two kinds of actions:

- Navigation actions and
- Use of resources by vehicles in order to fill a goal-action prototype.

Constraint programming approach: The planning problem for the package involves different aspects: selection of goals, selection of an action mode for each goal, assignment of UCAV and their resources to each selected action mode, path planning and scheduling for each UCAV. Constraint programming is a powerful approach for integrating those different aspects. Indeed this approach is efficient for path planning (Allo *et al.*, 2002; Strady-Lécubin and Poncet, 2003) as well as for planning problems expressed by means of a propositional representation (van Beek and Chen, 1999). Thus, the problem is modelled using a constraint programming approach. Variables are associated to UCAV and nodes. For instance the Boolean variables $P_{i,j}$ and $I_{i,j}$ indicate respectively that UCAV i will pass by node j and is involved in a target attack at that node, $Q_{k,i,j}$ indicates that it uses resource k. The integer variable $T_{i,j}$ gives the arrival time of the UCAV at the node. Other variables are associated to UCAV and edges and to goals. For instance, for a target o, the Boolean variables A_o, $A_{o,j}$ and $A_{o,j,m}$ indicates respectively that the target will be attacked, that the attack takes place at node j and that it is done in mode m. The integer variables $Tobj_o$ and Eff_o indicate the attack time and efficiency respectively. Constraints describe navigation possibilities and conditions for goal achievement. In the model, the link between the path planning and the propositional planning parts of the problem is ensured by constraints of the type:

$$I_{i,j} \leq P_{i,j} \quad (1)$$

$$I_{i,j} \leq \sum_k Q_{k,i,j} \quad (2)$$

$$Q_{k,i,j} \leq K_{k,i} I_{i,j} \quad (3)$$

$$\sum_i Q_{k,i,j} \geq R_{k,m} A_{o,j,m} \quad (4)$$

$$\sum_m A_{o,j,m} = A_{o,j} \quad (5)$$

$$\sum_j A_{o,j} = A_o \quad (6)$$

Equations 1 and 2 indicate that pre-conditions for an UCAV to participate to an attack at a node are to pass by that node and to have some resource to use at that node. Equation 3 bound the resources usage by zero if the UCAV does not participate and by the available quantity, $K_{k,i}$, otherwise. Equation 4 indicates that the pre-condition for the package to attack a target in a given mode is to have at least the resource amount requested for that mode, $R_{k,m}$. Equations 5 and 6 indicates that a single mode and a single node are selected for the attack of the target.

Distribution of the computation: The graph of variables and constraints associated to a multi-vehicle mission presents a star structure: the variables associated to goal achievement are connected by constraints to the variables associated to the different UCAV but there is no direct constraint between the variables of two UCAV. This structure allows the decomposition of the initial problem into a problem associated to each UCAV and a goal achievement problem. The decomposition of the initial problem has the advantage of permitting the use of the computing resources of all UCAV. Several techniques are available to conduct the distributed search of a solution. The technique used works in three steps:

1. Sets of solutions are searched for the problems associated to each UCAV.
2. A co-ordination problem, including the goal achievement problem and the selection of one solution per set, is solved.
3. The solution is refined for each UCAV.

Implementation: This technique is implemented using JADE (Bellifemine *et al.*, 1999), a FIPA compliant agent framework, and the CHOCO (Laburthe, 2000) tree searching constraint solver. Special attention has been given to the problem of the control of the time spent by the solver to solve the sub-problems; selection of variables to be assigned, selection of values for those variables, interruption of a tree search and time assignment to each step of the resolution.

4.4 Format

The “format” function refines the macro actions of the plan into sequences of unit actions. For instance the macro action “launch bomb 1 on target 101 at time t” is refined in the sequence of unit actions:

- “select resource type bomb 1 at time t-d” then
- “initialise selected resource with target 101 features at time t-e” then
- “ask to C2 go/no go at time t-f” then
- “if C2 answer is go fire bomb 1 at time t-g”.

Finally, “format” send a message to “act” to inform it about the existence of a new plan.

5. EXECUTION AND COMPUTATION TIME

5.1 Activation logic

The activation logic of the MMS modules is presented on figure 6. The MMS is activated every 0.1 seconds. For most of the cycles only three functions are activated: “observe and compare”, “predict” and “act”. For those cycles, the result of the analysis of messages from the platform by “observe and compare” and “predict” indicates that no pre-emptive behaviour has to be activated and no new plan has to be computed. The “act” function continues carrying on actions of the current plan.

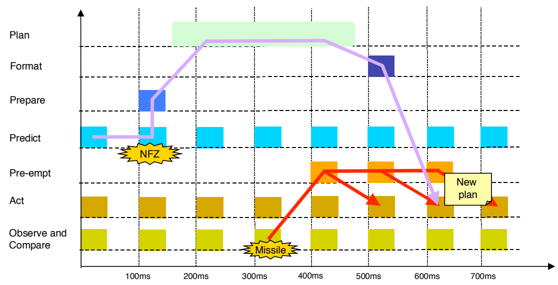

Fig. 6. Activation logic of the modules of the MMS

For cycles where “observe and compare” indicates that a pre-emptive behaviour has to be activated, the “pre-empt” function is additionally activated. This function may remain activated for several consecutive cycles until the behaviour is finished. Meanwhile the “act” function applies the actions issued from the behaviour. For instance, if the disruptive event is the detection of a missile firing, the function remains activated until the end of the escape manoeuvre.

For cycles where “predict” indicates that a new-plan has to be computed, the “prepare” function is additionally activated. In the same cycle the “plan” function is activated in the background. Several cycles after the “plan” function provides a plan and activates the “format” function just for a single cycle.

5.2 Taking into account constraints

Special attention is given to the way the computation time constraints are taken into account. First, different priorities are assigned to the input messages and only a bounded number of messages are processed at each cycle. Second, long processing is performed over several cycles, for “predict”, or in a separate thread, for “plan”. Moreover, “plan” solves the co-ordination problem by searching first a solution without attack. This solution is obtained very quickly providing an almost anytime feature for “plan”. The conjunction of those techniques ensures that a computation time bound for a time cycle of the mission management system can be predetermined.

6. EXPERIMENTAL RESULTS

Experimental results are given in the context of suppression of enemy air defence and strike scenarios.

6.1 Unit test of the planning function

In order to assess the performance of the planning function, tests are conducted on a single Sun Blade 1500 computer. A planning request for four UCAV with 4 targets conducts to the generation of 46 solutions for the first UCAV after 1.0 seconds, 46 solutions for the second UCAV after 1.6 seconds, 46 solutions for the third UCAV after 1.7 seconds and 138 solutions for the fourth UCAV in 2.4 seconds.

Figure 7 illustrates the second step of the planning method. A solution with no target attacked is found in about 1.0 seconds. Then as resolution time increases the efficiency of the solution, in terms of average number of targets destroyed, is improved and more targets are attacked. Finally the refinement of the solution is given in 0.05 seconds for the first UCAV, 0.09 seconds for the second UCAV, 0.14 seconds for the third UCAV and 0.24 seconds for the fourth UCAV. Note that the resolution times for the first and third step of the method are over estimated because the tests are conducted using a single computer. Finally, it can be observed that if optimality is not required the computation time for the co-ordination step can be reduced to few seconds.

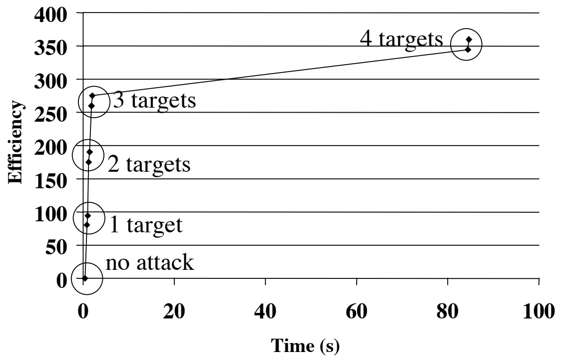

Fig. 7. Efficiency (%) of the solution of the co-ordination problem in function of the computation time

6.2 Simulation experiments

A simulation program was developed in order to test the behaviour of the MMS for a complete mission execution. In order to run these assessments there was an incremental complication of the scenarios starting from only one aircraft doing a flight path without any action and ending with a complete eight UCAV scenario including all the sequence of events possible.

Figure 8 presents the result of an experiment conducted with this program. It shows how the package, here two UCAV, reacts to the introduction of a new threat and a simultaneous failure in its jamming capabilities.

The first step is the detection by different UCAV of the package of the presence of the new threat. After having localised this threat the package decides to replan because of a too high probability of being killed if it follows the original path. The second step is then to prepare new segments on which the package will be able to find a new path and eventually assign actions. As there is no more jammers usable and as the fuel consumption used to avoid the new threat is compatible with the goal of the mission without endangering more the package, the solution kept is to go around this new threat changing flight level in order to use the terrain as a natural mask. This adaptation of the flight path to avoid the threat leads to the adaptation of how to attack the target because of the cooperation needs of the two UCAV. The new plan is then applied at a previously defined waypoint of application.

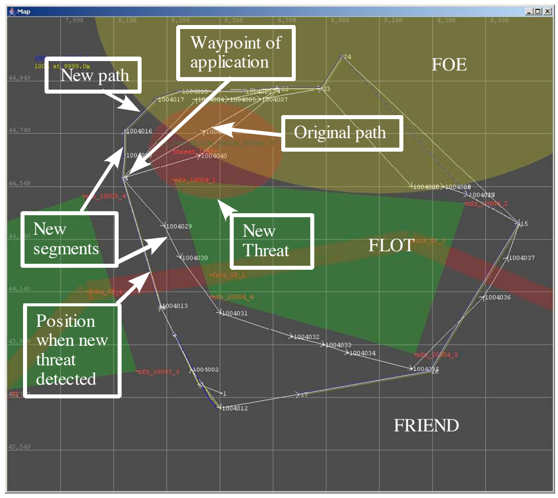

Fig. 8. Introduction of a new threat and package reaction

More experiments were done during the project in order to increase the complexity of the planning to be done and their results demonstrate the capability of the package integrating a distributed planing function through a reactive and deliberative architecture to achieve the goals specified. That is, the package is able to carry on nominal missions as specified, to activate the contingent behaviours on disruptive events, to decide whether or not to plan and, if necessary, to plan and to run in a bounded time. Moreover data link requirements for the functions of the MMS and performance of distributed planning are assessed.

Figure 9 shows how complex a tactical situation can become and what the planner had finally to manage as well in the nominal situation than in the presence of events.

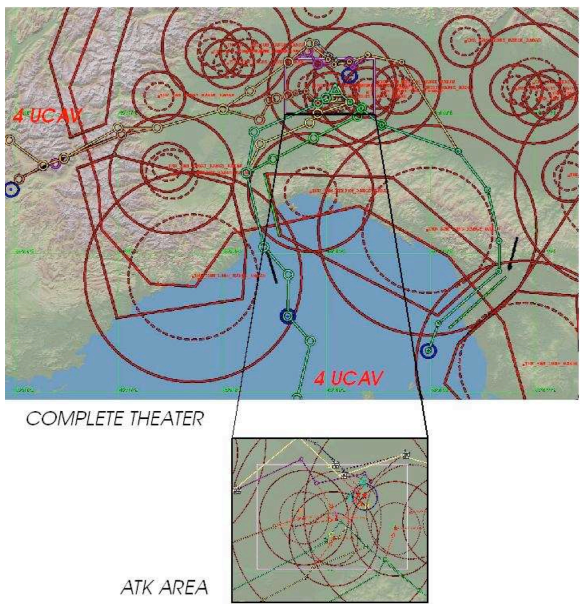

Fig. 9. Complexity of a tactical situation

CONCLUSION

The proposition of this architecture and a distributed planning method for package missions contributes to demonstrate the feasibility of vehicle intelligence and autonomy. Indeed, with the integration of on-line planning, disruptive events in absence of human intervention do not conduct necessary to the abortion of the mission.

However this area of application of robotic architectures is not fully covered. Some research directions are:

- Study of the link between the geometry of the flight path and the actions.
- Study of the way of taking into account uncertainty about the state of the vehicles and the environment as distance from current date increases.
- Study of the efficiency of other distributed methods.
- Study of mixed initiative planning for fleets with manned and unmanned vehicles.
- Study of the sustainability of the mission consistency despite the ability of computing several new plans while performing the mission.
- Study of the possibility of deriving proofs about the frequencies of pre-emptive behaves and re-planning activity.
- Study of the possibility of deriving proofs about the architecture safety and about the planner safety as a preliminary step to certification.

REFERENCES

Allo, B., C. Guettier, V. Legendre, J.C. Poncet and N. Strady-Lecubin (2002) Constraint model-based planning and scheduling with multiple resources and complex collaboration schema. In: *Proceedings AIPS'02*.

Avalle, M. and B. Patin (2007) Mission Management System for UCAV. Presented in: RTO meeting, AVT146, Platform Innovations and System Integration for unmanned Air, Land and Sea vehicles.

Barrouil, C., R. Demolombe, P. Fabiani, C. Tessier, P. Da Silva Passos, Y. Davaine, B. Patin and N. Prego (1999) TANDEM: an agent-oriented apoach for mixed system management in air operations. In: *RTO meeting proceedings 46, Advanced Mission Management and System Integration Technologies for improved Tactical Operations.*

Beard, R.W., T.W. McLain, M.A. Goodrich and E.P. Anderson (2002) Coordinated target assignment and intercept for unmanned air vehicles. *IEEE Transactions on Robotics and Automation*, **18(6),** 911-922.

van Beek, P. and X. Chen (1999) CPlan: A Constraint Programming Approach to Planning. Proceedings of AAAI, 585-590.

Bellifemine, F., A. Poggi and G. Rimassa (1999) JADE – A FIPA-compliant agent framework. In: *Proceedings of PAAM.*

Bresina, J.L. and M. Drummond (1990) Integrating planning and reaction. A preliminary report. Proceedings of the American Association of Artificial Intelligence Spring Symposium.

Brooks, R. (1986) A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, **2(1),** 14-23.

Findeisen, W. and I. Lefkowitz (1969) Design and Applications of Multilayer Control, 4th Congess of the International Federation of Automatic Control, Technical session 42, 3-22.

Laburthe, F. (2000) CHOCO: implementing a CP kernel. In: *CP'00 Post Conference on Techniques for Implementing Constraint programming Systems*. TRICS. Singapour.

Li, S., J.D. Boskovic, S. Seereeram, R. Prasanth, J. Amin, R.K. Mehra, R.W. Beard and T.W. McLain (2002) Autonomous Hierarchical Control of Multiple Unmanned Combat Air Vehicules (UCAVs). In: *Proceedings of the American Control Conference*. Anchorage.

Mehra, R.K., J.D. Boskovic, and S. Li (2000) Autonomous Formation Flying of Multiple UCAVs under Communication Failure. In: *Position Location and Navigation Symposium IEEE 2000*, 371-378, San Diego.

Riedel, J.E., S. Bhaskaran, S. Desai, D. Han, B. Kennedy, G.W. Null, S.P. Synnott, T.C. Wang, R.A. Werner and E.B. Zamani (2006) Autonomous Optical Navigation of Deep Space I - http://nmp-techval-reports.jpl.nasa.gov/.

Strady-Lécubin,N. and J.C. Poncet (2003) Mission Management System High Level Architecture, Report 4.3. MISURE/TR/4-4.3/AX/01, EUCLID RTP 15.5.

Wise, K.A. (2003) X-45 Program Overview and Flight Test Status. In: *2nd AIAA "Unmanned Unlimited" Systems, Technologies and Operations – Aerospac*. San Diego.

ACKNOWLEDGEMENT

The French "Délégation générale de l'armement", a part of the ministry of defence, has funded this work from 2003 to 2006 in the scope of the ARTEMIS project: thanks to this institution. The authors acknowledge the ARTEMIS partners for their contribution: thanks to Jean-Francois Gabard and Catherine Tessier.