Embedding heterogeneous levels of decisional autonomy in multi-robot systems*

Jeremi Gancet, Simon Lacroix

LAAS-CNRS, 7 avenue du colonel Roche, 31400 Toulouse
jgancet@laas.fr, simon@laas.fr

Summary. This paper presents an architecture for multi-robot (MR) systems in which robots exhibit heterogeneousness in terms of autonomous decisional capacities. This architecture enables various configurations of decision distribution among the components of the system, according both to the available decision making capabilities of the robots, and to the operational constraints related to the tasks to be performed. This architecture is instanciated in the context of a multi heterogeneous Unmanned Aerial Vehicles (UAV) project, and in this context, details related to a generic executive (supporting the versatility of the architecture) are provided.

1 Introduction

The architecture presented in this paper deals with multi-robot (MR) systems operations, considering a particular, understudied frame: contrary to most of the existing MR architectures, we do not make any definite assumption regarding the autonomous decisional capabilities of the robots composing the system. Different kinds of contexts or applications justify such an approach, like the two following ones :

- when a MR application requires to be able to add new (a priori unkwnon) robots in this MR system, with limited work/adaptation load
- when a MR system's user wants to be able to take control over robots, whatever their decisional capabilities.

We introduce in this paper an architecture concept and associated tools dedicated to the requirements of such *heterogeneous multi-robot systems*, the term "heterogeneous" being essentially related to decisional autonomy capabilities.

Problem statement. MR architectures embrace more concerns than single robot architecture: in particular, designing MR architectures requires to define the decision making scheme and to specify the interaction framework

* This work is partially supported by the Comets IST project (# 34304) of the 5th European Framework Program

among the different robots of the system, which of course influences the design of the individual robot architecture. For instance ALLIANCE [8] provides a behavior-oriented solution, enabling the design of totally distributed, fault tolerant multi-robot systems, whereas Simmons & al. [9] extend the three-layers architecture model [1, 4] within a MR framework, where interactions between robots may occur along the different layers.

These different MR architectures enable coordination and cooperation of several robots, but assume a given, homogeneous level of decisional autonomy for all the system's robots. Although these architectures may allow the integration of physically heterogeneous robots, they can not cope with heterogeneous robots in terms of *decisional capabilities*. Nevertheless, some MR applications clearly require the possibility to involve robots exhibiting quite different decisional abilities. This is typically the case in the COMETS project [7, 3]: COMETS aims at designing a multi-UAV framework enabling cooperative operations, in the context of forest fire monitoring and surveillance applications. In COMETS, some UAVs are directly controlled by an operator, some others are only endowed with *operational autonomy* (their tasks being planned and monitored by a central station), whereas others may have *decisional autonomy* capacities (hence should achieve given missions by themselves). Moreover, depending on the situation, the central station should be able to take control over UAVs endowed with decisional capabilities. COMETS hence requires an architecture integrating both *central decision making* and *distributed decision capabilities*; moreover, this architecture should provide with the possibility to configure dynamically the decisional scheme, depending on the available decisional capabilities of the robots and on the operational context.

Outline. Section 2 details the proposed multi-robot architecture: a taxonomy of robots decisional autonomy is introduced, and used as a foundation to state this architecture. In section 3, we focus on the design and implementation of a generic executive that is suited for various levels of decisional autonomy. Section 4 then provides a general scheme for the higher levels of robots autonomous decisional capabilities, as well as the main directions to implement these features.

2 Robot decisional autonomy architecture

2.1 A taxonomy of decisional autonomy capabilities

Within a multi-robot system, the "decision" encompasses several notions:

- **Supervision and execution**: The *executive* is a passive, reactive management of tasks execution, whereas *supervision* is an active process that manages all the decisional activities of the robot, whatever their extent.
- **Coordination**: It ensures the consistence of the activities within a group of robots. It defines the mechanisms dedicated to avoid or solve possible resource conflicts that may arise during operational activities. This is especially related to trajectories and multi-robot cooperative tasks execution (*e.g.* simultaneous perception of the same target by several robots).

- **Mission refinement, planning and scheduling**: These decisional activities are dedicated to plan building, taking into account on one side models of missions and tasks, and on the other side models of the world: robot's perception and motion abilities, current knowledge related to the environment, etc.
- **Task allocation**: This deals with the way to distribute tasks among the robots. It requires to establish a task assignment protocol in the system, and to define some metrics to assess the relevance of assigning given tasks to such or such robot.

These decisional components can be implemented according to different configurations: they can be gathered within a Central Decisional Node (CDN), or be partially (or even totally) distributed among the robots. We define the "level of autonomy" of a robot as the amount of decisional mechanisms it is endowed with, and consider the following five levels (see table 2.1):

		Supervision and execution	**Coordination**	**Planning**	**Task allocation**
Level 1	Central	X	X	X	X
	Distrib.	-	-	-	-
Level 2	Central	x (supervision)	X	X	X
	Distrib.	x (executive)	-	-	-
Level 3	Central	x (supervision)	x (high level coordination)	X	X
	Distrib.	x (executive)	x (low level coordination)	-	-
Level 4	Central	-	-	-	X
	Distrib.	X	X	X	-
Level 5	Central	-	-	-	-
	Distrib.	X	X	X	X

Table 1. Repartition of the decisional components between the central station and a given robot, regarding the five decisional autonomy levels defined.

- **Level 1**: no autonomy onboard the robot. The robot is only able to directly execute elementary tasks requested by the CDN.
- **Level 2**: executive capabilities (operational autonomy). The robot is able to manage partially ordered sequences of elementary tasks, and to return execution status of the tasks.
- **Level 3**: same as level 2, plus simple interacting capabilities. The robot may manage online simple interactions (synchronizations) directly with other robots endowed with at least the same level of decisional autonomy.
- **Level 4**: deliberative capabilities. High level tasks requests are managed (involving autonomous task planning/scheduling), and the multi-robot tasks coordination is autonomously ensured in a distributed way among robots endowed with at least the same level of autonomy.

- **Level 5**: same as level 4, plus tasks reallocation capabilities. The robot may opportunistically reallocate tasks and accept new tasks from other robots of the system also endowed with this level of autonomy.

This taxonomy is characterized by a large gap between levels 3 and 4: up to the level 3, a CDN is expected to ensure the global consistence of the system's activity: these levels are considered as **"low levels"** of decisional autonomy. Whereas levels 4 and 5 introduce the possibility to delegate coordination and mission refinement activities in a distributed way (**"high levels"** of decisional autonomy), and even, for level 5, to have the tasks allocation configurations dynamically updated.

2.2 Robots decisional architecture

Figure 1 depicts the overall architecture of the system: a CDN (interfaced with users) communicates with robots, exchanging messages which abstraction level depends on each robot's current level of autonomy. Robots are provided with a set of functional components and with a Distributed Decisional Node (DDN) possibly ranging from the simplest decisional autonomy level, up to the highest decisional capabilities.

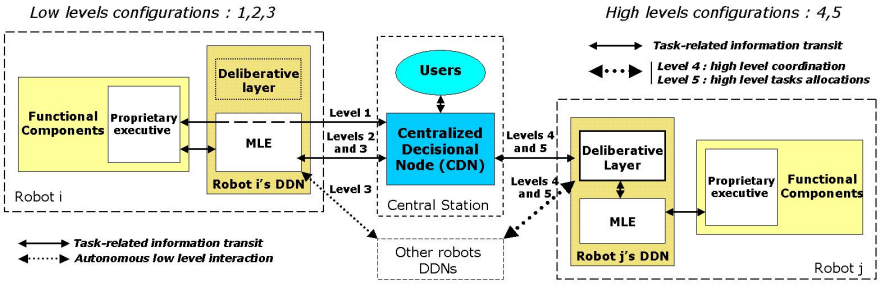

Fig. 1. Left: Low levels (1 to 3) configurations of decisional autonomy. Right: High levels (4 and 5) configurations of decisional autonomy

Regarding low autonomy levels, the DDN is restricted to an executive. This executive being actually common to all levels, we denote it as the *multi-level executive* (MLE). At level 1, the MLE behaves as a transparent connecting point between the CDN and the robot's functional components. However, the full MLE's features are required when considering upper levels: at levels 2 and 3, it manages tasks sequences execution, and at level 3 it enables simple interactions with other robots of the same level.

For higher levels, the MLE relies on the robot's Deliberative Layer (DL) instead of the CDN, tackling higher autonomous decisional capabilities. The DL deals with missions and tasks refinements, as well as coordination activities or task reallocation (when relevant, i.e for the level 5). Stakes and issues related to the DL are further discussed in section 4.

This architecture is currently exploited for the development of the COMETS multi-UAV system, where different, heterogeneous UAVs have to achieve missions related to surveillance and monitoring implying collaborative activities such as cooperative perception.

3 Multi-Level Executive (MLE)

We focus here on the MLE's features, relying on the COMETS project, as the application context: hence robots are UAVs in all the following.

3.1 General task model and assumptions:

Our task model is built around elementary events processing: these events are expected to occur whenever the states of tasks evolves. Events can also correspond to other noticeable activities evolution, such as the reception of a message, or the elapsing of a certain lapse of time. Tasks have a temporal extent: a task starts, then ends after a certain amount of time. The starting event is the only controllable event: all other kinds of events related to a task are contingent, *i.e.* the system can not guarantee that such an event will occur, neither exactly when it may occur. A task can give rise to several, partially ordered contingent events during its execution.

For the low levels of autonomy, the CDN is supposed to be able to elaborate a safe and consistent MR plan, and therefore to provide the UAV with the (already consistent) tasks to be processed, according to a task communication formalism. On the other side, the minimal requirement expected from an UAV is its ability to execute *elementary tasks*, *i.e.* unitary, "simple" tasks that can be handled by robot's functional components. In the COMETS project, the following tasks are expected to be processable by the functional components of a robot integrated in the system: take-off (TO), go-to (GT), take-shot (TS), wait (WT), and land (LD).

Integrating a given UAV in the whole system requires to provide this UAV with a basic interface enabling elementary tasks information transmission (requests, status, exec. results). For that purpose, an *elementary task formalism* has been developed (its specification is not detailed in this paper).

3.2 Executive's mechanisms for the low decisional levels

At the **first level** of decisional autonomy, the MLE only transmits the elementary tasks requested by the CDN to the functional components of the UAV, and then sends back execution status.

Regarding the **second level**, the MLE manages partially ordered sequences of tasks in a consistent way and in a timely and safe manner. Two main mechanisms are involved for this purpose:

- **dynamic tasks insertion:** this enables the possibility to request tasks insertion in the UAV's current task plan, according to an *insertion mode*

that characterize the relative order of the newly inserted task versus the current partial order of the tasks already scheduled. Four possible insertion modes are defined:

- **SEQuential (SEQ) mode:** The task has to be provided with a certain number of preconditions (in terms of expected events), which satisfaction can be specified either as *mandatory* or *optional*: in the first case, the satisfiability itself should be permanently satisfied, *i.e.* if the precondition happens not to be satisfiable anymore, then the task is aborted. On the contrary, an *optional* precondition is considered as satisfied (and hence removed from the task's list of preconditions) if it is actually satisfied *or* if it happens that its own satisfiability becomes **unsatisfiable**. In this case, the task is not aborted. Figure 2 illustrates these precondition mechanisms.
- **Very Urgent Task (VUT) mode:** this mode is a way to trigger a priority task, preventing any incompatible task to be executed during this time: the list of incompatible tasks to prevent should be provided as parameters of the task insertion. If an incompatible task is already running, it is interrupted. Otherwise, if an incompatible task is scheduled, then it can be either cancelled or only delayed (its preconditions are updated taking into account the task being inserted in VUT mode).
- **DEPendant (DEP) mode:** it allows to insert a task with as many preconditions as tasks currently scheduled: each precondition is satisfied when the corresponding task triggers its "end of task" event. Moreover, these are *mandatory* preconditions (*i.e.* as defined in the SEQ insertion mode).
- **Non Urgent Task (NUT) mode:** it allows to set as many preconditions as tasks currently scheduled: each precondition is satisfied when the corresponding task triggers its "end of task" event. However, contrary to the DEP mode, these are *optional* preconditions (*i.e.* as defined in the SEQ insertion mode).

The SEQ insertion mode is the most usual mode: it is a natural way to attach preconditions to a new task being inserted in the UAV's current plan. The VUT mode provides with a way to request urgent tasks execution, bypassing the current plan's constraints, but keeping it consistent. Finally, DEP and NUT modes are only shorcuts: a SEQ mode task may as well substitute itself to each of them.

- **dynamic tasks aborting:** this mechanism enables the possibility to request tasks abortions in the current plan. If the task is already running, then the abortion of the task is an interruption. If the task is not yet running, then the abortion is a cancellation (the task is descheduled). The abortion triggers a propagation mechanism, that checks which of the scheduled tasks depend on the aborted task (*i.e.* the tasks having a precondition expecting an event from the aborted task, like a "end-of-execution" event): if the dependence is a *mandatory* precondition, then this task is also aborted, and so on. If the dependence is an *optional* precondition,

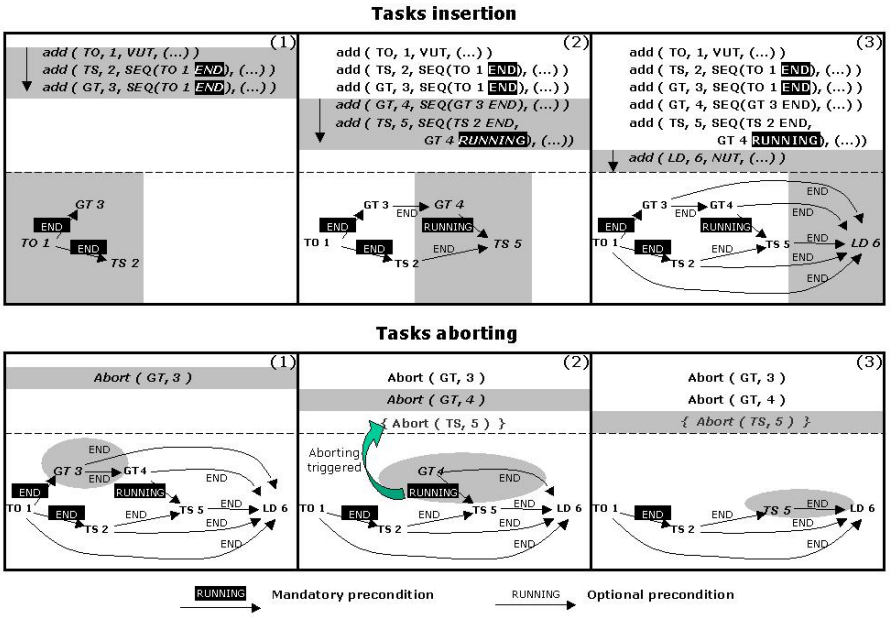

Fig. 2. Top: Examples of tasks insertion and illustration of the corresponding preconditions dependencies. (1): a VUT task and SEQ tasks with single mandatory precondition. (2): SEQ tasks with both mandatory and optional preconditions. (3): NUT task. **Bottom:** Examples of tasks aborting (1) and illustration of abortion propagation toward dependent tasks having mandatory preconditions (2) and (3)

then the dependence is removed as if the precondition was satisfied, and the corresponding task is not aborted.

The **third level** of decisional autonomy deals with an additional mechanism intended to enable autonomous synchronizations between several UAVs' MLEs. A synchronization can be requested to a given MLE as a particular task, that produces events (start, running, end...) in the same way as usual tasks do. It is also possible to insert a synchronization task with particular insertion modes as defined previously. Two "roles" are specified as parameters of a synchronization task: sender ($\mathcal{S}$), and receiver ($\mathcal{R}$): $\mathcal{S}$ and $\mathcal{R}$ are the sets of UAVs considered respectively as senders and receivers of the synchronization message. When a synchronization task is processed, the MLE checks whether its own ID is noticed in the $\mathcal{S}$ or $\mathcal{R}$ sets. Three situations may occur:

- ID $\in \mathcal{S}$ (only): the MLE has to send a synchronization signal to all UAVs which ID belongs to the set $\mathcal{R}$. This signal contains the synchronization task's ID, and also this UAV's ID. From this UAV's point of view, the task is considered achieved.
- ID $\in \mathcal{R}$ (only): the UAV expects to receive synchronization signals from all UAVs which IDs belong to the set $\mathcal{S}$. From the point of view of this

UAV, the synchronization task is considered achieved as soon as all signals are received.

- ID $\in \mathcal{S}$ and ID $\in \mathcal{R}$: the UAV should both send its own synchronization signal then wait for signals from all other UAVs specified in the set $\mathcal{S}$. The synchronization task is considered achieved as soon as all signals are received. If $\mathcal{S}$=$\mathcal{R}$, then the synchronization is a general "rendez-vous" between all UAVs.

Figure 3 illustrates this synchronization mechanisms.

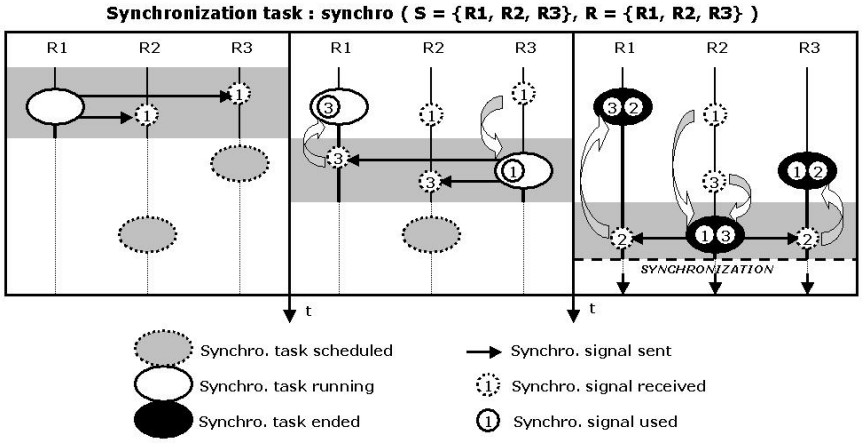

Fig. 3. Illustration of a synchro. task with 3 UAVs, in the case of a "rendez-vous"

4 High decisional autonomy levels

We briefly present here the general framework for the development of the DL, dealing with the higher autonomy levels. Figure 4 exhibits the general structure of the UAVs' DDNs: the main part is the *supervisor*, which is responsible for missions and tasks refinements. It elaborates plans with the support of a *tasks planner/scheduler* and the *specialized refiners*. It also triggers negotiation sessions for coordination purposes and tasks reallocations (when relevant, i.e. at level 5 only), through the *negotiation manager*.

- **Task planning / scheduling**: given a requested mission, the deliberative layer is expected to build executable plans, which execution would be straightforward in an ideal context. Actually, contingencies during missions execution may require to repair dynamically parts of the plans, hence a plan is never considered as definite: it should be reprocessable, online, taking into account latest contingencies. This is partially devolved to the *task planner / scheduler*, manipulating symbolic tasks models. The *specialized refiners* are intended to bridge the gap between abstract, symbolic plans and actual data of the world, known through various models (namely environment model, UAV and communication models): the "*path*" refiner provides the trajectories, expected times and resources consumptions in order

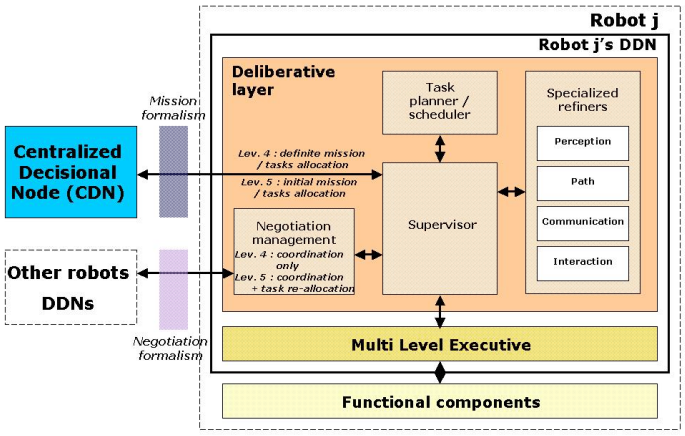

Fig. 4. Deliberative layer detail

to reach given points. The "*perception*" refiner computes the most relevant perceptions in the current context (mapping, tracking, or fire monitoring, for instance). The "*communication*" refiner computes the valid ranges of communications, taking into account UAVs and environment models. Finally the "*interaction*" refiner takes into account interactions models, and restricts/constraints operational areas according to these models. A preliminary version of these refiners has been developped, and on-going work deal with the integration of a HTN-oriented symbolic planner (D. Nau's SHOP2 [6]) with these refiners.

- **Coordination**: two kinds of coordination can be pointed out: spatial coordination, and interactions-related coordination. The spatial coordination with other UAVs is an activity that should continuously work in background, since it is a vital mechanism (prevents conflictual trajectories). It should overcome all other kinds of deliberative mechanisms (we are currently exploring the possibility to adapt the Plan Merging Protocol [2] to the particular purpose of UAVs spatial coordination). Regarding interactions-related tasks coordination, flexible plans produced by the planner/scheduler should be coordinated with the plans of other involved UAVs, for each given interaction: this mechanism should rely on a relevant negociation protocol, ensuring a consistency of the joint plans of the UAVs. Such a protocol definition and development, based on incremental assessment and exchange of critical plan's sections, is currently on-going.
- **Task reallocation**: task reallocation is a distributed mechanism intended to dynamically improve missions/tasks allocation. Task reallocation continuously requires to identify which tasks or parts of plans would be worth to be reassigned to other UAVs: this can be solved using an opportunistic mechanism, pointing out "expensive" tasks (metrics to be defined), or tasks looking consistent with regards to other UAVs plans (criteria to be defined). For the reallocation process itself, we intend to extend the Contract Net Protocol pardigm [10][5].

These directions are guidelines for our next developments related to the distributed decisional capabilities for the COMETS' UAVs: all the mechanisms and decisional features briefly introduced in this section deal with current, on-going investigations and developments. We plan to provide much more details related to these purposes in future publications.

5 Current results and future work

The MLE's low level mechanisms (levels 1 to 3) introduced in section 3 have been tested both in simulation and in actual experimentation, with several UAVs (up to 3 UAVs: 2 helicopters and 1 blimp). During these tests, a MLE was attached to each of the UAVs, and high level decision making was performed within a control center (the CDN) interfaced with human users. Furthermore, some preliminary simulated tests have also already been led for some of the DL components.

The design and development of higher decisional capabilities are on-going, to meet our objectives of applying the whole versatile architecture to an actual multi-UAV system (e.g. the COMETS' UAVs fleet).

References

1. Alami R., Chatila R., Fleury S., Ghallab M., Ingrand F.(1998). An Architecture for autonomy. In: Int. Journal of Robotics Research. Vol.17, p.315-337
2. Alami R., Ingrand F., Qutub S.(1998). A Scheme for Coordinating Multi-Robot Planning Activities and Plans Execution. European Conf. on Artificial Intelligence (ECAI'98)
3. COMETS project's official web site: http://www.comets-uavs.org
4. Gat E.(1991). Integrating planning and reacting in a heterogeneous asynchronous architecture for mobile robots. In SIGART Bulletin 2, 17-74
5. Lemaire T., Alami R., Lacroix S.(2004). A Distributed Tasks Allocation Scheme in Multi-UAV Context, to appear in Int. Conference on Robotics and Automation (ICRA'04)
6. Nau D., Au T.C., Ilghami O., Kuter U., Murdock W., Wu D., Yaman F. (2003). SHOP2: An HTN planning system. Journal of Artificial Intelligence Research 20:379-404
7. Ollero A., Hommel G., Gancet J., Gutierrez L.G., Viegas D.X., Forssen P.E., Gonzalez M.A. (2004). COMETS: A multiple heterogeneous UAV system. To appear in SSRR 04, Bonn, Germany
8. Parker L.(1998). ALLIANCE: An architecture for fault-tolerant multi-robot cooperation. In IEEE Transactions on R.& A. 14(2):220-240
9. Simmons R., Smith T., Dias M.B., Goldberg D., Hershberger D., Stentz A., Zlot R.M. (2002) A Layered Architecture for Coordination of Mobile Robots. In: Multi-Robot Systems: From Swarms to Intelligent Automata, Proc. of the 2002 NRL Workshop on Multi-Robot Systems, Kluwer Academic Publishers
10. Smith R.G.(1980). The contract net protocol: High-level communication and control in a distributed problem solver. IEEE Transactions on Computers
11. Volpe R., Nesnas I., Estlin T., Mutz D., Petras R., Das H.(2001). The claraty architecture for robotic autonomy. In: Proc. of the 2001 IEEE Aerospace Conference, Big Sky, Montana